6/4/2021
Characters (with spaces): 189079

User Experience in FLOSS Projects:

Challenges, Considerations and Possible Solutions

Written By Daniel Valentin
IT University of Copenhagen

Abstract

Free/Libre and Open Source Software (FLOSS) projects are steadily producing more and more software that is used by non-technical end-users. These projects have since their inception in the early 2000's been plagued by the mantra that they have little concern for their user experience, and as a result their user adoption is poor compared to their proprietary rivals. The field of HCI has made numerous studies on these projects to understand the mindset, characteristics and the fast-paced development that is believed to be responsible for this phenomenon. So far, generalisation and success in introducing contemporary HCI methodology into these projects has proven difficult. Making matters worse, their distributed online nature makes the use of traditional HCI methodology difficult. This case study of two contemporary FLOSS projects, utilizes contextual inquiry and subsequently design interventions to 1) gauge the validity of existing findings, 2) create a prototype to create awareness about the underlying issues while 3), arguing for the continued experimentation with how to best collect knowledge and thereby make an impact in these projects. The study discusses the implications of this endeavour and concludes that FLOSS projects have the potential for strong UX and that the designers' role in these projects increasingly are going to be shifting towards supporting and intervening in the processes of system, user, and community meaning-making.

Table of Contents

1. Introduction

1.1. Topic and Context

When a service, product or technological artefact is created, its designer must actively choose how to inform themselves about a great many things. Combined with the ever-increasing pace at which technologies shape society, and vice versa it is often a detriment that "(...) prevents us from experimenting with and learning about all the new possibilities created by new technology and new knowledge." (Löwgren & Stolterman, 2004, p. xii). This is especially the case in Free/Libre and Open Source Software (FLOSS) projects where communities freely create and share software that is using the frameworks and technological opportunities that are available today. These projects have since the early 2000's impacted the way that software is shared and made available to users around the world, from the consumer level all the way up to how services and applications are being managed on the enterprise level in multimillion dollar companies. Many of these applications and projects are installed on millions of computers around the world and if asked, most computer literate people have encountered projects such as Linux, VLC, LibreOffice, Firefox, or GIMP.

These projects, however, are not without their own unique issues. They are, and always have been spearheaded by developers and technically inclined people. This means that ever since the conception of FLOSS projects as we know them today, their widespread adoption has been halted by the fact that these projects often consider design activities to be of either low priority or non-essential to the goals of the project (Cheng & Guo, 2018; Masson et al., 2017). This, in addition to the fact that many studies found that the community aspect and developing for others is a huge motivator in these projects, a discrepancy is quite clearly visible in this rather unique field. How do these projects come to terms with their dual ontological nature, namely that they are both social and technical, and allow room both for developers to have fun, create value for the average user and drive strong user experiences that rival proprietary software? This study seeks to unravel and understand the problems that exist in the crossroads between contemporary user experience design and Free/Libre and Open Source Software projects. It does so via a contextual inquiry and subsequent design intervention into two relatively new FLOSS projects, 'Jellyfin' and 'Taskcafe'.

1.2. Focus and Scope

This paper seeks to investigate how FLOSS projects address their user experience and design considerations via a case study of two such projects, and does so by using the methods of contextual inquiry, and design intervention to inspire a new way in which to continuously create awareness about the issues of the field. As largely supported by previous studies, the field of FLOSS represents a unique challenge for designers in that these communities are decentralized, distributed, and composed by mostly technically inclined developers. As the impact of FLOSS projects are increasingly gaining in significance, it is clear that the motivation for designers to intervene into these projects are rising in parallel. The focus of this paper is to bridge the gap between existing research and current FLOSS projects, and empirically cement what findings from these previous studies that are not technologically outmoded, by the fast-moving nature of this field. The hope in doing this, is to generally improve user experience in FLOSS projects overall and further the normalization of having design discussions as a high priority in FLOSS.

As mentioned, these empirical findings of this study are based on an inquiry into two projects, one small and run by a single developer, and another project that has quickly grown quite large and is including more and more contributors into their efforts. The study looks into and unpacks four primary categories of findings, namely 1) motivation for contributing, 2) resource and project management and its impact on design choices, 3) the perception of design and 4) design methodology. This study shows that these four categories are very much aligned with previous studies, but also help in understanding the contemporary state of affairs in current FLOSS projects with respect to the currently used platforms, modes of communication and how design choices are being made. Finally, a design research prototype is created to investigate how these projects react when faced with this new knowledge, the implications of it and the future challenges thereof to inspire future research in the field.

1.3. Question and Objectives

To come to meaningful insights regarding how FLOSS communities can be made aware about UX considerations, there were a lot of different dimensions to discuss and challenges to tackle. While collecting and analysing the data and figuring out possibilities for design, the following research question was used as a compass to navigate the complexity of this field and its problems, and eventually in creating a design research prototype with the goal of providing continuous valuable insights applicable both in the field of HCI, but also for these FLOSS projects.

How can researchers through enculturation more effectively engage with and build upon what is known about the user experience practices of FLOSS projects, to inspire the communities and future research towards action?

To answer this question, data was collected from FLOSS projects with the goal of establishing a holistic understanding of the current practices related to user experience design. After analysing the qualitative data, key areas of interest were discovered, and a design research prototype was used to experimentally engage with these projects with the goal of creating more discussions, awareness and enculturation surrounding design processes.

1.4. Glossary

The following list of words and their meanings are provided to the reader as much of the terminology of these projects are highly specialized and code centric. This list can be referred to whilst reading the study in order to clarify meanings and implications of this terminology throughout the paper.

Free/Libre and Open Source Software - (FLOSS)

In this context the Spanish word ‘Libre’ is used to “(...) emphasize that “free software” refers to freedom and not to price” (Stallmann, 2016). While other abbreviations exist (such as OSS for Open-Source Software, or FOSS for Free and Open-Source Software), and that the lines between them are somewhat blurry this paper opts to use the abbreviation ‘FLOSS’ to include both of these political camps, in order to assume a neutral position on the matter

Emby
- The project that became the codebase for Jellyfin, after the project moved from a FLOSS model to a proprietary model (*Beta Source Code Missing · Issue #3479 · MediaBrowser/Emby*, n.d.)

Open-source Bounty Sites
- Websites that facilitate a system wherein users can pay developers to implement certain features in FLOSS projects (*IssueHunt - A Bounty Platform for Open Source Projects*, n.d.)

Version Control Systems and Frameworks

- Systems often used by developers to support the hosting and merging of code. Examples of popular version control systems are for instance GitHub and Bugzilla. Certain terminology is often used with regards to these platforms.

A 'fork'
- Referring to 'forking' involves copying the source code of one project into another, either for the purpose of creating a new project altogether or for the purpose of re-emerging any changes back into the original branch later.

A 'Pull Request'
- Submitting a pull request refers to when a developer submits a request for the inclusion of some new code into an existing repository of a project.

Vue and Vuetify
- Vue and Vuetify is a JavaScript framework that supports building easily exchangeable components. Vuetify is a plugin for Vue that allows building components based on the material design guide made by google.

'Docker pulls'
- Docker is a containerization technology that in recent years has become popular for hosting applications such as Jellyfin, both in the enterprise space and in the home

Fider
A platform used to collect and allow users to vote on wanted features for software projects.

Communication Platforms

- Various communication platforms are used in FLOSS projects. Below some of the common ones utilized by the project are found.

Matrix
- Matrix is "(...) an open network for secure, decentralized communication" (*Matrix.Org*, n.d.), used by many FLOSS projects to support their instant chat needs.

Discord
- Is an instant messaging and video conferencing platform that is targeted at gaming, but has also seen popularity as a general-purpose platform for chatting (*About Discord | Our Mission and Values*, n.d.)

2. Literature Review

Here follows the review and walkthrough of the literature that is preliminary to discussing the inclusion of design considerations into FLOSS projects. The section goes into what existing research exists that cover the UX practices of FLOSS projects, and endeavours to synthesize the key findings, assessing strengths and weaknesses of the studies and uncovering gaps in the literature. The section also covers what type of contribution this study seeks to make to the field of HCI.

2.1. Existing FLOSS Research

Even though FLOSS projects are still a relatively young phenomenon, especially in an academic context, studies into its being has been conducted since the early 2000's when various projects started having larger user bases. Since then, design considerations such as usability has been a detriment to the mainstream success of these projects (Benson et al., 2004; Frishberg et al., 2002; Nichols & Twidale, 2006), which seems to still be the case over a decade later (Cheng & Guo, 2018; Wang et al., 2020). The overall body of FLOSS studies is still relatively small, however in recent years it seems that interest in understanding the design processes that goes on in FLOSS projects has risen and as a result more studies that seek to address these design-centric issues has appeared (Bach et al., 2007; Cheng & Guo, 2018; Rajanen et al., 2012). Below a table summarizing selected FLOSS studies can be found based on the seven types of research contributions often seen in HCI (Wobbrock & Kientz, 2016). These studies are not to be understood as an exhaustive list of all studies that might involve findings on the crossroad between FLOSS and design, instead this list represents the different types of studies that are representative of current efforts to understand the important concepts in the field.

Contribution type	Year and author
Opinion	(Frishberg et al., 2002)
Empirical	(Benson et al., 2004) (Andreasen et al., 2006) (Iivari, 2008) (Terry et al., 2010) (Iivari, 2010) (Iivari et al., 2014) (Rajanen et al., 2015) (Cheng & Guo, 2018)
Methodological	(Nichols & Twidale, 2006) (Bach et al., 2007) (Rajanen et al., 2012) (Masson et al., 2017) (Llerena, Rodríguez, Castro, et al., 2018) (Llerena, Rodríguez, Llerena, et al., 2018)
Survey	(Despalatovic, 2013)
Theoretical	(Rajanen & Iivari, 2015b)
Dataset	(Wang et al., 2020)

Table 1 - The studies referenced in this paper and their HCI contribution type based on often used types of HCI research (Wobbrock & Kientz, 2016).

2.1.1. Key Empirical Findings in FLOSS Studies

As can be seen from the table above, while these categories overlap greatly, most of the major HCI studies targeting the FLOSS environment are empirical in nature. Empirical studies "(...) provide new knowledge through findings based on observation and data gathering." (Wobbrock & Kientz, 2016, p. 40). As a lot of data has been collected and analysed since the beginning of FLOSS studies in the early 2000's, a summary of key findings is provided below.

2.1.1.1. An Ongoing Need to Support FLOSS Projects

The first notion that is consistent between the studies in this category is that fact that there "(...) is an imperative need for supporting OSS communities to collaboratively identify, understand, and fix UX design issues in a distributed environment." (Cheng & Guo, 2018, p. 1). This largely stems from the fact that while "(...) Open Source Software developers have produced systems with a functionality that is competitive with similar proprietary software developed by commercial software organizations" (Andreasen et al., 2006, p. 303), it is still largely developed for power users, which per definition creates friction when it is used increasingly by non-technical users (Nichols & Twidale, 2006). Another factor in this is the lack of general theory and methods for actively supporting the communities in their unique distributed setting (Terry et al., 2010).

2.1.1.2. The Perception of Design

In addition, it also seems consistent throughout the findings that the perception of design concepts such as usability and therefore its importance varies in different projects. With this in mind it was by one study suggested that "(...) projects for diverse users [...] tend to have a greater focus on the current state compared to projects for a more homogeneous audience" (Cheng & Guo, 2018, p. 6). This suggests that a community made of, for instance, 3D artists might be inherently more visual and engage in relevant discussions about a desired state more than diverse environments of users where it might be more difficult to establish consensus. In support of this, it was found in studies that some developers contribute the lack of design work to the FLOSS or "hacker" mentality in several ways, and as such is not directly a problem that needs solving. As such it seems that perception of design is a key finding across most of the studies in that it might be enforced in certain projects, downplayed in others, and possibly even discounted completely in some.

2.1.1.3. A Lack of Methodology

It also seems that many of the empirical studies found that while the project members argue that the importance of design decisions such as usability is high, "(...) in practice most of the efforts are based on common sense" (Andreasen et al., 2006, p. 311). This can be attributed to the lack of HCI methods that are tailored to this specific environment. In other words, to mitigate making decisions based on common sense there is "(...) a need to reconceptualize HCI methods to better fit this culture of practice and its corresponding value system" (Terry et al., 2010, p. 999). In line with this, it was also found that terminology in the projects differ greatly for words such as 'usability' (Andreasen et al., 2006, p. 308). This seems to suggest that while some project members might have specific knowledge "(...) which collectively mirror definitions commonly found in HCI textbooks" (Terry et al., 2010, p. 999), this knowledge is not shared by all members and as such it might be difficult to argue for certain design driven changes. It was found with this in mind that human interaction guidelines (HIGs) were used to solve this problem in certain situations allowing "(...) the community to orient itself towards 'action'" (Yamauchi et al. 2000 in Nichols & Twidale, 2006, p. 153).

2.1.1.4. A Lack of Resources

Furthermore, it was found that a lack of resources in the projects were a challenge (Masson et al., 2017), and this was especially evident given the nature of design discussions as "(...) interface (re)design may not be amenable to the same approaches as functionality (re)design" (Nichols & Twidale, 2006, p. 150). This was both a challenge in that a lack of market pressure generally results in less of a push to 'sell' the product to users, and that most resources are already being expended just from the sheer amount of communication that goes on between users, developers and involved parties in between (Terry et al., 2010). Also connected to this shortage of resources it was found that resource management problems also can emerge from certain platform specific choices, resulting in a loss of design rationale (Nichols & Twidale, 2006). Specifically, more archaic version control software had issues supporting design discussions as they often took a very visually rich form not suited for these platforms meant for resolving code-centric issues. This in some cases made "(...) the basic act of reporting a usability bug [...] more complex than reporting a functionality bug" (Nichols & Twidale, 2006, p. 151). Emerging solutions to this problem were also discovered in for example the *design-by-blog* approach used by some FLOSS projects to allow for discussions and the inclusion of visual aids outside the confines of their version control software (Despalatovic, 2013; Nichols & Twidale, 2006; Terry et al., 2010). It was also clear however, that offloading design work to different platforms is not without risk as it can cement its place as something external, out of view of the developers, and thereby of a lower priority (Andreasen et al., 2006).

This lack of resources was found to be tightly coupled with the inherent complexity of design work. Namely, in how design issues cluster in complex ways and span multiple components and interfaces, while also requiring longer than usual discussions involving the need for discussing “(...) design rationales, multiple, perhaps contradictory goals and design trade-offs” (Nichols & Twidale, 2006, p. 153). As such, design changes are difficult to make under the already very busy and resource depleted projects. A supporting factor in this would be that properly educating project members takes time (Terry et al., 2010), and that if time is already short in the projects joining, convincing and training project members becomes a very difficult task.

2.1.1.5. Gate-Keeping Tactics

Another key finding from the empirical studies was rooted in the long-term observation of projects where contributing design changes had previously been attempted (Rajanen et al., 2015). It was found that even if a designer manages to get to a place in the project where they might exert direct influence over the project direction, their efforts might slowly be changed, ignored, or outright reverted. This suggests that long-term change in a project differs from single design changes, which might on the surface appear as successful and meaningful contributions to a project. As such the successful intervention of a designer into a FLOSS project must be looked at with analytical scrutiny and must be allowed time in order for a realistic assessment to be made. With this fact in mind, the power structures in these projects are not the pure democracies that some participants might proclaim. Instead, FLOSS projects were found to be meritocracies, here-by-meaning, based on the merits of the involved.

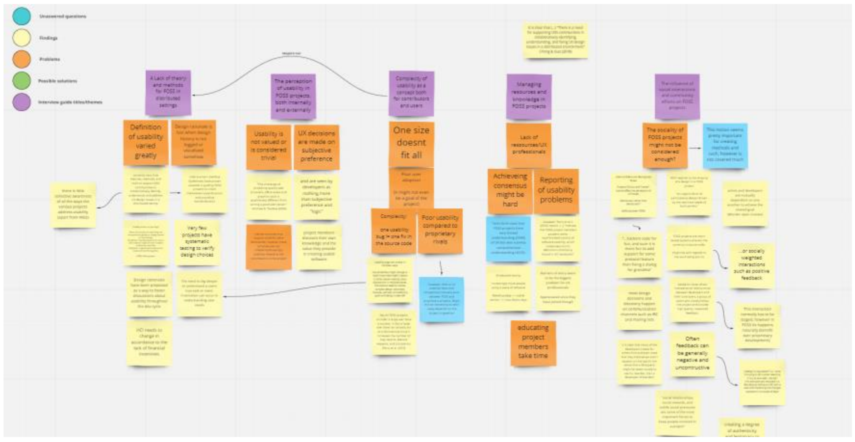

Unanswered questions
Findings
Problems
Possible solutions
Interview guide titles/themes
A Lack of theory and methods for FOSS in distributed settings
The perception of usability in FOSS projects both internally and externally
Complexity of usability as a concept both for contributors and users
Managing resources and knowledge in FOSS projects
Definition of usability varied greatly
Usability is not valued or is considered trivial
UX decisions are made on subjective preference
One size doesnt fit all
Lack of resources/UX professionals
The sociality of FOSS projects might not be considered enough?
Achieveing consensus might be hard
Reporting of usability problems
Very few projects have systematic testing to verify design choices
Complexity
one usability bug != one fix in the source code
Poor usability compared to proprietary rivals
...or socially weighted interactions such as positive feedback
HCI needs to change in accordance to the lack of financial incentives
educating project members take time
Often feedback can be generally negative and unconstructive

Figure 1 - Working with and uncovering the key findings

2.2. Limitations of Existing Research

In mapping the current research, the limitations of the outlined studies must be considered. Firstly, inquiries into the field of FLOSS is still as an academic entity relatively small and very much is its own small branch in the overarching field of HCI. Some studies critiquing the user interface and design of what can be considered the precursors of the modern FLOSS movement, were published as far back as the 1980’s (Norman, 1981). However, generally speaking the field is still young, and due to the increasing growth of technology FLOSS itself is constantly changing, making continuous academic efforts important. This consequently means that the body of empirical knowledge is small, compared to the amount of work done in more traditional HCI settings. These small sample sizes then mean that the reader must consider that each FLOSS project is highly specialized, and as such might have methods, terminology or otherwise unique characteristics not covered by existing literature.

With this in mind the fast technological turnover that defines this field can drastically change the validity of studies, the further one goes back. One such example is the change of version control software that has happened since the early 2000’s to now. Bugzilla, a version control software produced by Mozilla (*Bugzilla.Org*, n.d.), was used primarily in the 2000’s and therefore many of the papers of that era deal with findings specifically targeted at that platform or similar platforms. Accordingly, in this field that is increasingly at the forefront of technological opportunity one must be cautious not to conclude that the findings pertaining to one complex system necessarily translate 1:1 to a different system, even though their desired core function might be the same. Many of the key papers in the field do base most of their observations on these somewhat, compared to the contemporary landscape, archaic systems and as such this fast paced evolution that drastically changes research must be considered as a limitation as well (Liu et al. 2014 in Lazar, 2017).

Another consideration that is a product of technological change is the proliferation of code frameworks that abstract away some of the minutia user interface choices. 20 years ago, frameworks like ReactJS, Vue or Angular were not the norm, but now UI libraries (*Bootstrap*, n.d.; *Material-UI: A Popular React UI Framework*, n.d.; *Vuetify — A Material Design Framework for Vue.Js*, n.d.) make it easier for developers to create and cooperate on projects, and have software built on established design principles from companies with proven design frameworks such as Google and Apple (Apple, n.d.; Google, n.d.). These very same companies, and even large FLOSS projects, have easily accessible human interaction guidelines available that also serve to inspire these projects to create better user experiences (*GNOME Human Interface Guidelines*, n.d.; *KDE Human Interface Guidelines — Human Interface Guidelines Documentation*, n.d.). In other words, the expectations for what constitutes good user experience has changed quite drastically from the early 2000's, and this quite clearly plays into how projects perceive design goals and the use of for example design resources such as HIGs, design oriented code frameworks and inspirations from existing FLOSS or proprietary software. In summary, as these new frameworks and resources are still a relatively new tool in the field of FLOSS projects, their impact is not yet fully understood which both creates room for new research and critical analysis of existing research.

2.3. Goals of This Study; A Summary

Now, having assessed the current HCI studies that look into FLOSS projects, the contribution of this study can be expanded upon. As more and more were revealed about the practices of the involved projects, it became more apparent where most efforts are needed to both create stronger UX in FLOSS projects, but also where more knowledge is needed. Firstly, one goal of this study is to verify existing findings within HCI as to accommodate for the rapid change of technologies in the field. Secondly, there is a discrepancy between the findings of these studies and the actual impact they have on the overall quality of UX in FLOSS projects. In addition, even though there has been a growing interest in creating stronger UX in FLOSS in the last couple of years throughout various communities within the field (*Design & UX*, n.d.; Esparza, 2019; Open Source Design, n.d.), there is still a need for researchers to further this agenda. Therefore, another goal of this study unequivocally becomes to bridge the gap between the findings of HCI studies targeting the FLOSS environment, and the practical way in which these findings are communicated to these projects.

3. Theoretical Background

In the following section the theoretical framework is presented. Key concepts from the research are explored and overlapping findings are related to each other. Specifically, it is argued that one of the defining characteristics of FLOSS is that it warrants a complete rethinking of what defines use, and by extension the term ‘user’ in that FLOSS participants are both designers, users and indirectly the ones responsible for proliferating a ‘solution’ mindset. Additionally, the social shaping of FLOSS projects and their products, their mindset and their properties as a niche group within the realm of online communities is expanded upon through the introduction of selected parameters that aid in distilling these unique attributes.

3.1. Use and Users in FLOSS; Blurred Boundaries

In thinking about what constitutes the concepts of ‘use’ and ‘user’ Johan Redström presents a paper in which he “(...) explore[s] a different perspective than the role-based account of relations between designers and users” (Redström, 2008, p. 410). The reasons for this conceptual exercise lie in that certain problems arise when the concept of ‘users’ interfere with the actual use of a design. Specifically, he argues that “(...) referring to ‘users’ during design seem to assume that there already are users of things not yet designed, thus obscuring the complexity of what actually happens as someone starts using a thing, as someone becomes a user.” (Redström, 2008, p. 410). Additionally, emerging tendencies from more open-ended design scenarios wherein adaptability and extensive user participation takes place, suggests that intended use and actual use becomes increasingly difficult to harmonize. This phenomenon is also linked to the increasing pace at which technology constantly changes the fabric in which designers work, creating new possibilities for interaction. This increased interaction means that the form of a technological artefact is not only the product of the “(...) work of the designer but also, increasingly, the actions of the user.” (Redström, 2008, p. 411).
While Redström does not specifically base these notions in FLOSS communities themselves, they are a strong example of many of the very same tendencies he describes as part of open-ended design situations. The notion that the users are increasingly affecting the design of artefacts, or in this case digital artefacts, has been observed back to the very beginning of FLOSS development (Benson et al., 2004). FLOSS projects are without a doubt at the forefront of technological possibility and this specific group of users, namely the developers themselves,

are in many ways a double edged sword according to existing theory all which further illustrates the need for increased diversity in these communities. Studies make it clear that developers are not typical end users. It has many times been argued in previous studies that FLOSS software is "(...) designed for and by power-users" (Andreasen et al., 2006, p. 303) and theory has shown that they are power-users with a strong sense of priorities when it comes to the creation of a piece of software (Andreasen et al., 2006; Nichols & Twidale, 2006). While it has been argued more recently that FLOSS communities are "(...) now starting to acknowledge that 'we are not our users'" (Iivari, 2008, p. 4), and that non-developer user innovation does occur (Iivari, 2010), it seems that even in recent times this blurring of developer and user affect the day-to-day mindset in these projects and continue to impact the widespread adoption of this software (Wang et al., 2020). Further complicating this situation, some projects outright dismiss any interest in adhering to user experience principles out of their informal nature. Some projects are hobbies, others are solely for the fun of coding and others yet have no interest in attracting non-technical users to the community (Andreasen et al., 2006**).** To summarize, it would seem that while FLOSS projects in their best-case scenarios can embody a similar dynamic to that of participatory design (Rajanen et al., 2012; Terry et al., 2010) and allow for a bottom-up and democratic approach to user experience design, this is still rarely the case. One could say that the "(...) root of the problem seems to lie in the open source development process itself, where the focus is not placed on the user or usability, but on the system or the programmer" (Despalatovic, 2013, p. 958). As such, approaching the context of FLOSS "(...) on [the] basis of terms such as 'designer' and 'user' might not only be difficult, but potentially also misleading for design methodology." (Redström, 2008, p. 411).

3.1.1. Ontology of FLOSS Projects; 'The Solution Mindset'

With the above sentiment in mind, it is clear that the act of design covers both what Redström calls 'thing-design', but also the concept of 'use-design' (Redström, 2008). This notion aligns with how it many times has been argued in the field of socio-technology that technical artefacts are value-laden, and that the shaping of technology and society is intertwined in many ways. Designers, with this in mind are then obviously important as they are "(...) instrumental in societal development by designing digital artifacts that become part of large and small sociotechnical systems." (Löwgren & Stolterman, 2004, p. 142). This dual ontological nature (Kroes, 2001, p. 1 in Redström, 2008), namely that technology is social and technical, is a most important principle in socio-technical activities generally, but especially with FLOSS projects in mind.

In FLOSS projects the embodiment of this idea is what is described in several studies under different names. It is called the ‘FLOSS mindset’, the ‘solution mindset’, ‘rational culture’ or a ‘system-centered’ mindset (Benson et al., 2004; Despalatovic, 2013; Iivari et al., 2014; Yamauchi et al., 2000 in Nichols & Twidale, 2006; Wang et al., 2020), which describes the tendency to focus on solving problems, creating fixes and generally results in that projects allocate “(...) resources on other competing goals” (Wang et al., 2020, p. 4). These goals are often related to the talents of the developer and involve code-centric tasks and while this obviously differs from project to project, it was found that for the proliferation of user-centered approaches to be accepted it often relies on a top-down pursuit and initiative from the core project members (Wang et al., 2020). While it seems counterintuitive that “(...) OSS culture might be at odds with usability work” (Iivari et al., 2014, p. 2) it is partly related to the very methodical approach traditional design work often takes (Sharp et al., 2019), which does not harmonize easily with the voluntary work that goes on in these projects. In existing theory, this is one of the major challenges in creating stronger user experiences throughout FLOSS and is largely a problem of the lacking diversity of people in these projects. The existence of this mindset is largely responsible for why much of the existing theory argues that FLOSS projects are not democratic as one might think, instead they are meritocracies wherein this system-oriented approach thrives and non-code contributions are inherently harder to argue for as a contributor (Andreasen et al., 2006, p. 2014; Iivari et al., 2014; Masson et al., 2017; Rajanen & Iivari, 2015b). In summary, engaging with FLOSS projects as a designer becomes difficult when there is an inherent bias towards non-code contributions. It must be emphasized that while it can generally be said that FLOSS projects share this mindset, it is without a doubt a sliding scale and part of what makes engaging with these projects complex is that they are very much unique independent sub-communities with different ingrained values and participants. In other words, just as there is no one-size-fits-all in traditional design projects these projects reflect that very same quality, namely that the context that they exist in define how solutions are best approached (Cheng & Guo, 2018; Iivari et al., 2014).

3.2. Online Communities

Undeniably, a big part of what defines FLOSS projects as unique is the fact that they almost exclusively are online entities. Online communities have been the subject of much research since the conception and proliferation of the internet (Baym, 2015), and accordingly findings from this area of study can undoubtedly aid in understanding some of the unique attributes of such communities.

3.2.1. Properties of Online Communities

Nancy Baym outlines five properties that are central in the study of online communities. These five properties are space, shared practise, shared resources and support, shared identities and interpersonal relationships. Interestingly, these all resonate with aspects of FLOSS communities, even at this introductory and non-exhaustive stage. 'Space' refers to how groups of online communities are disconnected from any geographical place, yet that its members think of these places as shared (Baym, 2015). The idea of shared practises refers to how "(...) community also can be found in the habitual and usually unconscious practices - routinized behaviours - that group members share" (Baym, 2015, p. 86). These unconscious practises reveal a great deal about the underlying logic of such groups, which is a valuable tool in understanding how FLOSS projects might empower their design process. The concept of 'Shared resources and support' concerns "(...) the supportive exchange of resources" (Baym, 2015, p. 91) that often define the interactions in online communities. 'Shared identities' entails that online communities often "(...) include a shared sense of who 'we' are that may be pre-existing or develop within a group" (Baym, 2015, p. 96). These roles, defined which are defined by their consistent and systematic behaviour, could for example be the roles of the 'local expert', 'conversationalists' or 'trolls' (Baym, 2015). Lastly, online communities embody the possibility for creating interpersonal relationships. These interpersonal relationships are the result of consistent engagement within an online group and "(...) provide a social mesh that underlies and helps to connect the broader web of interconnection within the group more closely" (Baym, 2015, p. 100). In summary, these five interconnected attributes are a suitable starting point for a discussion about the online FLOSS communities with the goal of "(...) re-search[ing] new HCI methods that operate in the culture and value system of the FOSS community." (Terry et al., 2010, p. 1008).

These socially driven and unique identifiers immediately appear like attributes in opposition of the notion of a ‘rational designer’. Namely, in how rationality of design choices usually are defined by if “(...) (1) [...] it is possible to understand - that is, when we can see why the process has been enacted in a specific way, and (2) when the enactment is in line with our own values.” (Löwgren & Stolterman, 2004, p. 49). In other words, while rationality is an elusive and often shallow concept in everyday situations, one could “(...) argue that rationality is the intentional reason and motivation that makes an action understandable, and that motivation is a result of (consciously or unconsciously) chosen values.” (Löwgren & Stolterman, 2004, p. 50). How does one then counter this complexity? Can rationality even be achieved from this outlook? One way to at least support rational design work is to expose the process specifics to all involved parties. It stands to reason that “(...) externalizing an idea, making it ‘visible,’ also makes it accessible for criticism, development, expansion, revision, and possible discard.” (Löwgren & Stolterman, 2004, p. 51). In summary, this suggests that especially these decentralized projects need to take special care to how they share the ongoing processes as a means to foster stronger design especially in the light of some of the gate-keeping tactics that previous research has suggested to be present in FLOSS projects (Rajanen et al., 2015).

3.2.2. FLOSS Resource and Platform Specifics

In relation to the five properties listed above, while Baym’s model at first glance infers the idea of separate properties, their interconnectedness is undeniable. This interconnectedness is especially evident observing the platforms on which they live, namely in that the platforms used in FLOSS projects encourage many of the same habitual practises that are analogous to the system-centred mindset as presented above.

3.2.2.1. Space

FLOSS projects usually employ several textual spaces in their work. Firstly, the code being created requires structuring to handle both merging, sharing and discussions. Additionally, this code needs to be shared somehow, while allowing any interested party to review and download said code. This means that a versioning control system or platform is usually involved in this task, such as the very popular platform GitHub (*Github ‘About’*, n.d.). As these systems are highly textual, existing studies have found their properties to be an influential factor in the design of FLOSS. One of these findings, a method of design in FLOSS called ‘design-by-blog’, describes how it is increasingly becoming more and more common for projects to host a blog that can support longer and more detailed rationales for changes, ideas and choices made in the project in question (Nichols & Twidale, 2006). Several reasons have been discovered for why design work is being offloaded of these central spaces in FLOSS projects. Firstly, it was found that there was a desire to “(...) minimize the acknowledged complexity of Bugzilla by removing certain design debates until a solution can be devised that can be returned for public review and critique” (Nichols & Twidale, 2006, p. 156). Secondly, it was discovered that there was a wish to only use the version control platform for more formal reviews. Lastly, it was found that the text-centric nature of these platforms is “(...) less than ideal for the brainstorming, design and debate of candidate solutions.” (Nichols & Twidale, 2006, p. 156). In summary, much of the more deliberate design work that involves debate and visually rich discussions were found to be “(...) occurring in face-to-face meetings, use of Mozilla newsgroups, Internet Relay chats, emails, instant messaging, etc.” (Nichols & Twidale, 2006, p. 156).

Whether or not design work is being done in a design-by-blog scenario, mostly in chat groups or on social media, certain risks were involved with moving design work off the central project repository. By having most of the work being carried out in chat between few people “(...) can mean that the record of how the design evolved and how the decisions were made is lost” (Nichols & Twidale, 2006, p. 156). Making this notion more noteworthy is the fact that “(...) by far, the most commonly cited means of discovering and discussing usability issues was through the project’s communication channels” (Terry et al., 2010, p. 1004). Also, a risk of moving such design work off the main project repository, is that design work might be labelled as a second thought, as something external to the project thereby failing to be properly integrated into the development cycle as seen in other studies (Rajanen et al., 2015).
In any case, it is clear that current research shows that the textual nature of these systems often does not support design work properly, and simply moving design work elsewhere entails managing other types of complexity such as a lack of awareness about design goals or a loss of rationale.

3.2.2.2. Shared Practices

Shared practises in FLOSS are very dependent on the platforms they use day-to-day because of their distributed setting. While it has been found that the characteristics of the specific communities being studied "(...) greatly affected the focus and style of its discussion" (Cheng & Guo, 2018, p. 6), some shared practises were found to be present among most of existing theory. Firstly, as mentioned in the literature review it was observed that the diversity of the project participants usually by itself reveals a great deal about what sort of practices might be prevalent in a FLOSS community (Cheng & Guo, 2018). Specifically, it was argued that the more diverse the users of the software were, the more the community was focused on the current state of the project in solving emerging issues, which involved looking at several factors. Among these were the urgency of the problem, symptoms, their cause and finally the complexity of solving the issue. In other words, one way to gauge the impact of characteristics between communities, is to establish how diverse the community is, which in turn defines how likely it is that an inquiry into the project will be successful. It was found however that even in communities for diverse users, developers still based UX discussions on "(...) personal opinions and experiences, with a lot of over-generalized assumptions" (Cheng & Guo, 2018, p. 6) which also coincides with what has been found generally in the field (Iivari, 2008; Nichols & Twidale, 2006; Rajanen et al., 2015). Another, in some ways unintended shared practice that relates to this, is that a product of using these textual and technical version control systems such as GitHub, it was "(...) reported that non-technical users may be intimidated or incapable to use the bug reporting systems" (Iivari, 2008, p. 5). In other words, a big hindrance to achieving a higher diversity is that the systems being used to support their day-to-day work in these projects are fairly technical.

Among the shared practices, it was also found that FLOSS projects work in short intensive cycles, which generally does not align with how design projects materialize (Andreasen et al., 2006). For instance one study trying to appropriate usability testing into a FLOSS project found that "(...) usability evaluation was slower, and therefore some of the usability team's findings were obsolete by the time the report was ready" (Rajanen et al., 2015, p. 11).
Lastly, language as a shared practice was also a barrier between developers and users, and while the body of knowledge looking into FLOSS as an "online speech community" (Baym, 2015, p. 86) is very small it is a noteworthy observation nonetheless (Llerena, Rodríguez, Castro, et al., 2018).

3.2.2.3. Shared Resources, Support and Identities

With shared resource and support in mind it was found that projects, specifically employ the adaptation of support called ‘informational support’, namely that the primary shared resource and support is “(...) advice or possible solutions to a problem” (Cutrona & Russell, 1990, p. 322 in Baym, 2015, p. 94). This coheres with the notion of the ‘solution mindset’ and means that developers are likely to argue for why “(...) the requested feature [might] already [be] available somehow in the OSS” (Iivari, 2008, p. 10) as opposed to engaging in discussion about design possibilities. Naturally proposing such solutions through some technical means, can be a detriment to non-technical users which is analogous to previously mentioned mantra that FLOSS is built for and by power users.

Shared identities, while more elusive than some of the others of Bayms properties have appeared throughout existing theory. One developer for instance said that “(...) hackers code for fun, and sure it is more fun to add support for some protocol feature than fixing a dialog for grandma” (Andreasen et al., 2006, p. 307). This touches on the perception of design activities, but while this sentiment was only propagated by few developers it might be representative of a somewhat elitist mindset that exists in some projects, comparable to other occurrences in hacker culture where a battle rages between “(...) elitism and populism that continues to be a defining mark of hacker sociality” (Coleman, 2010). It has however with this in mind been noted that generally there is a “(...) lack of research on organizational culture in open source software (OSS) development” (Rajanen & Iivari, 2015a, p. 2). While inconclusive some findings suggests that this sense of shared identity might be a detriment for designers as “(...) usability intervention succeeded only in the OSS projects showing resemblance with the adhocratic type of culture, while in the unsuccessful cases the culture types identified were hierarchical and group culture type” (Rajanen & Iivari, 2015a, p. 10). In other words, adhocratic projects with a large degree of freedom might be better suited to support UX work as more types of contributions might be accepted.

3.2.2.4. Interpersonal Relationships

The notion of interpersonal relationships was found to be somewhat discounted in current research, especially with motivational factors in mind. It was found in some studies that the most important motivational factor in participating in FLOSS projects were “(...) direct interpersonal relationships between developers and their core users, a group of users who closely follow the project and provide high quality, respected feedback” (Terry et al., 2010, p. 999). In other words, designers must keep in mind that “(...) the more important currency for motivating usability work in [a] FOSS community is social rewards – praise and positive feedback from end-users whose opinions are valued” (Terry et al., 2010, p. 1000). This means that in order for designers to effectively engage these communities existing theory points to “(...) that enculturation of usability specialists is essential for this to happen” (Iivari et al., 2014). While the prospect of fostering interpersonal relationships to generate social rewards surely seems like an interesting way to approach these unique issues, it also by its very nature means that the most active users likely are technically inclined and able to appreciate the work of the developers and understand for example change logs, or developers' blog posts. Just how the designer is supposed to involve end-users more actively, while fostering these kinds of rewards likely require “(...) a more in-depth analysis of the enculturation process [...] to understand the variety of meanings, attitudes and practices that can be associated [with] it” (Iivari et al., 2014, p. 9). It is however with this notion clear that a major barrier for designers is the barrier of entry, and reaching a place where contributions are valued enough to somehow effectively engage with the community and possibly even a larger context of users (Rajanen & Iivari, 2015b; Wang et al., 2020).

3.3. ‘Design’ After Design; A FLOSS Model

Another relevant concept to discuss is the difference between ‘use’ before use and ‘design’ after design. The fundamental difference lies in that traditional HCI typically employ “(...) the by now classic ambition [...] to test and try out ‘use’ during the design process and in advance of actual use - what we might call a ‘use’ before use approach” (Redström, 2008, p. 421). At the other end of this spectrum lies the ‘design’ after design approach, that entails a more open-ended approach. This, what we might call a ‘design’ after design approach allows for “ (...) attempt[s] to create a larger space of possibilities for acts of defining use through use” (Redström, 2008, p. 421), and is interestingly very analogous to what is done currently in most FLOSS projects. While this is not necessarily a choice made with design processes in mind, the system-centered mindset of the participants indirectly establishes this way of working, namely by developing something that works first and iterating on design and other aspects later.

As alluded to in the literature review, it was found that FLOSS projects vigorously employ a method of graduated testing (Terry et al., 2010), which is quite similar if not identical to this exact approach.

First, reference users and bleeding edge users test development versions of the software. Then, a larger group of users use the software and provide feedback when it is released as a stable version. Finally, the last significant group of users is reached when the software is included in Linux distributions. (Terry et al., 2010, p. 1005)

Now, this layered approach to testing features and stability is at the very core of FLOSS mentality, and while the above quote pertains specifically to usability testing, this notion of graduated design or testing is an unmistakable opportunity for designers to better understand and navigate the risks of this way of working.

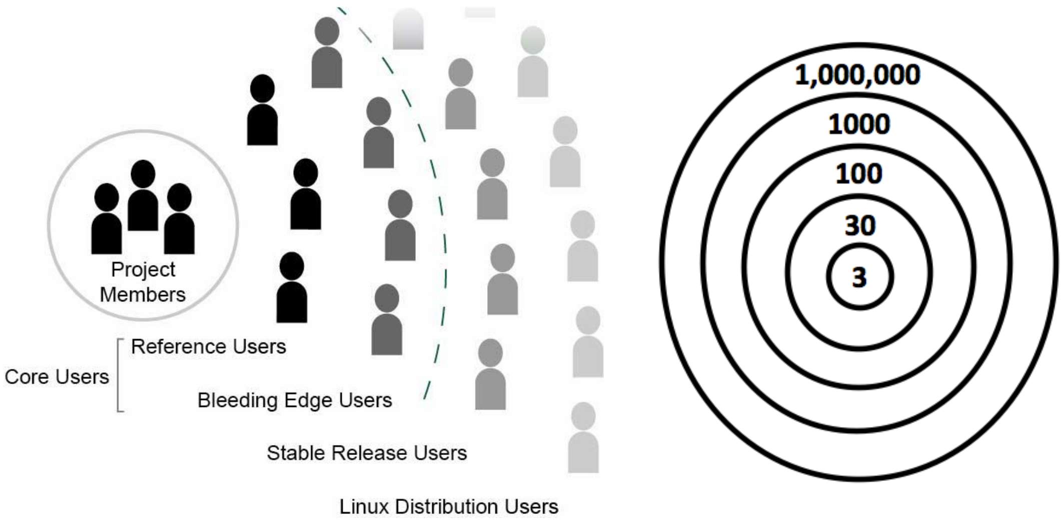


Figure 2 - "Strata of Users" and "The Onion Metaphor" - (Figure 1 in Masson et al., 2017, p. 1136; Figure 1 in Terry et al., 2010, p. 999)

Several pitfalls were found in researching this 'design after design' approach in FLOSS. Firstly, it seems that this way of working heavily relies on users to actively report their findings and issues with the software. The users most likely to report these bugs were found to be those closest to their respective projects which becomes a culprit in "(...) that it can potentially lead to software tuned to the needs of users close to the project, rather than the larger user base" (Terry et al., 2010, p. 1006). A supporting factor in this is the fact that the vast majority of

FLOSS end-users are passive and do not contribute or consider any of the community aspects of these projects (Nichols & Twidale, 2006). This bolsters the notion that projects can quickly enforce their own world-views and indirectly enforce this system-centric mindset, which often means that the end result is software that favors many features, instead of learnability or user experience (Terry et al., 2010). Further complicating the effects of this cultural echochamber is the fact that all project resources are tied up into this process, labelling bugs, fixing bugs and discussing changes. When only the same people participate in this process and occupy most resources, little momentum exists to seek out and understand more diverse users that might reside outside the immediate context of the project, especially in the absence of financial pressures to compete with others (Terry et al., 2010).

In summary, while it has been assumed that "(...) it is quite likely that a usability infrastructure will evolve, just as one did for functionality bugs" (Nichols & Twidale, 2006, p. 156) it seems that this evolution is slower than first assumed in the early 2000's. While initiatives such as design-by-blog, has shifted this somewhat towards a more user centered mindset that mirrors methods commonly used in UX work, it is clear that very rarely this graduated 'design after design' approach yields the results that might be possible were more diverse users involved.

3.4. Summary

In summary, in synthesizing key findings and ascertaining their connections some general categories of interest can be found. Firstly, a lot of findings deal with the perception of design activities, looking into how the participants in these projects view and act on design challenges with regards to how their opinions determine action.

Motivation is a topic covered by much of the literature and increasingly deals with how to best utilize "(...) social relationships, social rewards, and subtle social pressures" (Terry et al., 2010, p. 1006) to further the acceptance of design work in FLOSS projects.

Furthermore, much of the literature deals with the resources, and the managerial aspects of the projects, which obviously is intertwined with the methodologies of these FLOSS participants. With methodology in mind a general lack of design experts, design focused platforms, and shared design methodology is the biggest challenge throughout the existing theory.

4. Case Description

To understand the various FLOSS projects that have been included in this paper a preliminary case description is required for further understanding the used methods and the content of the analysis. A historical account for the projects is essential due to the highly specialized nature that FLOSS projects can have, be it in their target audience, community or something else entirely. Therefore, it is critical to differentiate the Taskcafe project and the Jellyfin project as two separate projects as presented below.

4.1. The Jellyfin Project

4.1.1. History and Connection to Emby

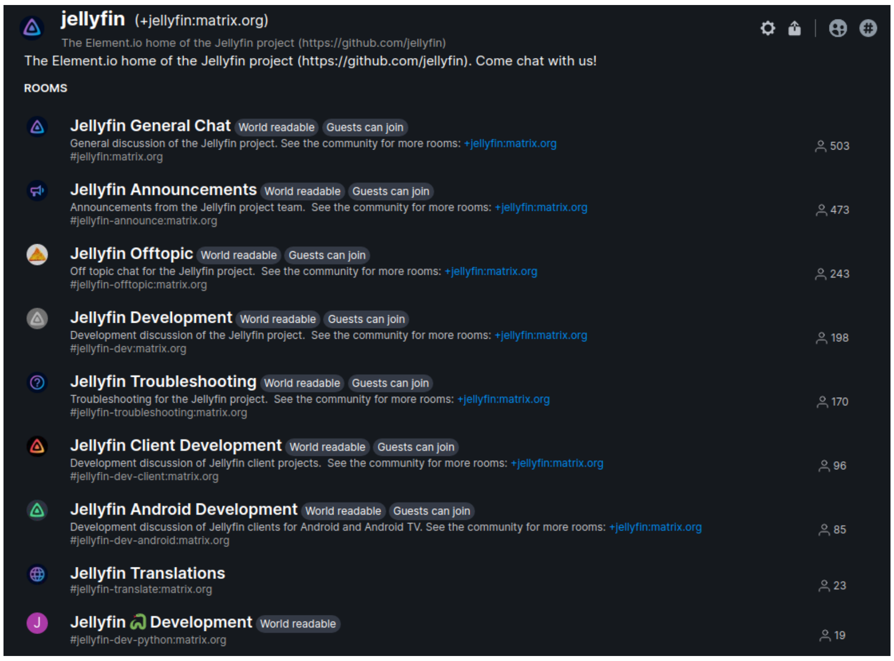
The Jellyfin project is a centralized media server solution that is available completely free and hosted on Github. Jellyfin's source code "(...) is descended from Emby's 3.5.2 release" (*Home | Documentation - Jellyfin Project*, n.d.) which is also a centralized media server solution that seeks to fulfill the same functionality as Jellyfin, however has since its conception turned from a fully FLOSS model to a freemium business model with a closed and proprietary source code. In other words, Jellyfin is built on the latest version of Emby where the project's code was still open and available to everyone. This has meant that Jellyfin has inherited a large amount of code and bugs from this older code base, along with all of the design choices made in the creation of Emby. The time consuming process of cleaning up this legacy code, replacing parts of it with newer solutions and creating their own identity in the field of media solutions is still very much ongoing and is according to many of the participants a very complex overarching process with many intertwined and complex parts.

4.1.2. The Community

The Jellyfin community was established shortly after it was announced that Emby would close their source code (*Beta Source Code Missing · Issue #3479 · MediaBrowser/Emby*, n.d.) and transition into a proprietary model in order to allegedly mitigate financial issues in the organization. Members of the Emby community that found this change unacceptable voiced their opinion and eventually the decision to formally fork the Emby project was made by several contributors. The project is now being driven by 24 project members (*Jellyfin 'People'*, n.d.), however this number is always expanding as more and more people are invited to join the Jellyfin project. To manage these community efforts several channels and modes of communication are being used in their work and activities.

4.1.3. Jellyfin Communication Channels

Their internal communication takes place on the matrix network which is "an open network for secure, decentralized communication" (*Matrix.Org*, n.d.). Matrix supports instant chat messaging features such as different chat-channels, voice chat, video chat and administrative tools for project owners. In addition to the channels found in the picture below, there are also some internal closed matrix groups that only the current project members are allowed to participate in.



jellyfin (+jellyfin:matrix.org)
The Element.io home of the Jellyfin project (https://github.com/jellyfin)
The Element.io home of the Jellyfin project (https://github.com/jellyfin). Come chat with us!

ROOMS

Jellyfin General Chat World readable Guests can join
General discussion of the Jellyfin project. See the community for more rooms: +jellyfin:matrix.org
#jellyfin:matrix.org
503

Jellyfin Announcements World readable Guests can join
Announcements from the Jellyfin project team. See the community for more rooms: +jellyfin:matrix.org
#jellyfin-announce:matrix.org
473

Jellyfin Offtopic World readable Guests can join
Off topic chat for the Jellyfin project. See the community for more rooms: +jellyfin:matrix.org
#jellyfin-offtopic:matrix.org
243

Jellyfin Development World readable Guests can join
Development discussion of the Jellyfin project. See the community for more rooms: +jellyfin:matrix.org
#jellyfin-dev:matrix.org
198

Jellyfin Troubleshooting World readable Guests can join
Troubleshooting for the Jellyfin project. See the community for more rooms: +jellyfin:matrix.org
#jellyfin-troubleshooting:matrix.org
170

Jellyfin Client Development World readable Guests can join
Development discussion of Jellyfin client projects. See the community for more rooms: +jellyfin:matrix.org
#jellyfin-dev-client:matrix.org
96

Jellyfin Android Development World readable Guests can join
Development discussion of Jellyfin clients for Android and Android TV. See the community for more rooms: +jellyfin:matrix.org
#jellyfin-dev-android:matrix.org
85

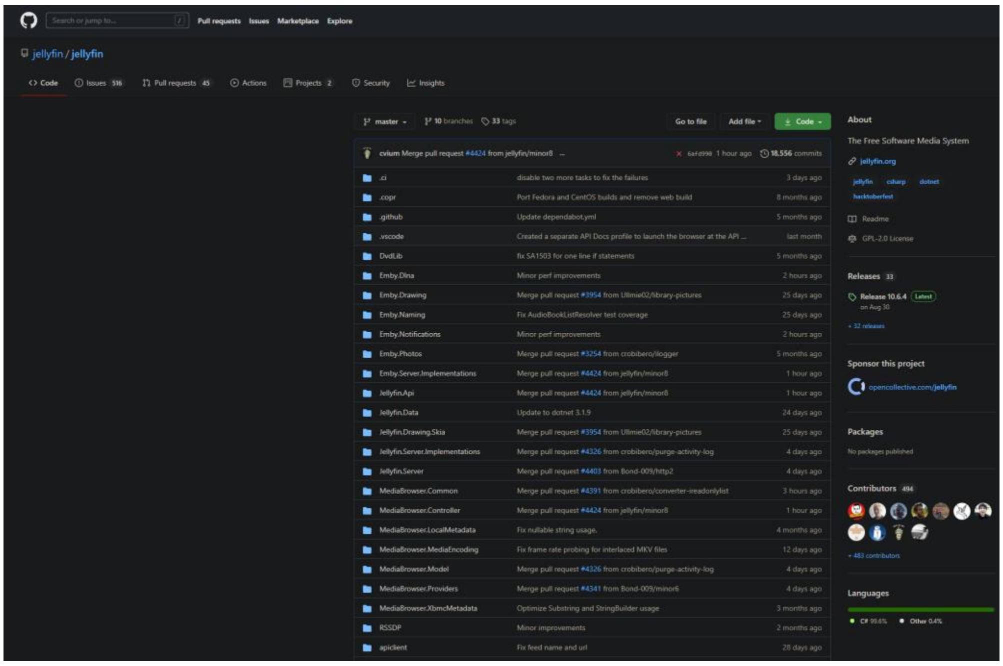
Jellyfin Translations
#jellyfin-translate:matrix.org
23

Jellyfin Development World readable
#jellyfin-dev-python:matrix.org
19

Figure 4 - The community channels on the Matrix network

The Jellyfin subreddit is another channel that is available to the public as a means to discuss, ask about and get help with using Jellyfin. Since December 10, 2018 it has quickly grown and is currently very active and has, at the time of writing, nearly 16.000 members. Here many of the developers in the Jellyfin project are also active, in both providing information about the latest releases, helping in solving issues and participating in general discussions about the project.

GitHub is also a major channel not only for communication, but also for a lot of the actual work that goes on in the project. GitHub is an organization that through its website, of the same name, provides hosting for source code, software development and enables version control using the tool "Git" (*Github 'About'*, n.d.).
Much of the contribution work that is being done on GitHub is done through their "Pull request" system that allows for someone to review and approve changes before they become final and merged in the "master" or main branch. Additionally, issues can be opened and managed on GitHub using labels such as "Design", "Enhancement" or "Feature request". Duplicate Issues and relations are then managed by mentioning other issues that are related or duplicates of each other (*GitHub Features*, n.d.). In these issues communication back and forth seek to resolve the issue, figure out its origin and resolve it immediately or for future releases. As Jellyfin is a media solution, multiple ways in which one might consume media are considered via the different clients, also found in GitHub. These clients include Android, AndroidTV, FireTV, Roku and many others (*Clients - Jellyfin: The Free Software Media System*, n.d.). Each of these clients are hosted in their own repository under the Jellyfin project.



cvium Merge pull request #4424 from jellyfin/minor8 ... 6afd998 1 hour ago 18.556 commits

.ci	disable two more tasks to fix the failures	3 days ago
.copr	Port Fedora and CentOS builds and remove web build	8 months ago
.github	Update dependabot.yml	5 months ago
.vscode	Created a separate API Docs profile to launch the browser at the API ...	last month
DvdLib	fix SA1503 for one line if statements	5 months ago
Emby.Dlna	Minor perf improvements	2 hours ago
Emby.Drawing	Merge pull request #3954 from Ullmie02/library-pictures	25 days ago
Emby.Naming	Fix AudioBookListResolver test coverage	25 days ago
Emby.Notifications	Minor perf improvements	2 hours ago
Emby.Photos	Merge pull request #3254 from crobibero/ilogger	5 months ago
Emby.Server.Implementations	Merge pull request #4424 from jellyfin/minor8	1 hour ago
Jellyfin.Api	Merge pull request #4424 from jellyfin/minor8	1 hour ago
Jellyfin.Data	Update to dotnet 3.1.9	24 days ago
Jellyfin.Drawing.Skia	Merge pull request #3954 from Ullmie02/library-pictures	25 days ago
Jellyfin.Server.Implementations	Merge pull request #4326 from crobibero/purge-activity-log	4 days ago
Jellyfin.Server	Merge pull request #4403 from Bond-009/http2	4 days ago
MediaBrowser.Common	Merge pull request #4391 from crobibero/converter-ireadonlylist	3 hours ago
MediaBrowser.Controller	Merge pull request #4424 from jellyfin/minor8	1 hour ago
MediaBrowser.LocalMetadata	Fix nullable string usage.	4 months ago
MediaBrowser.MediaEncoding	Fix frame rate probing for interlaced MKV files	12 days ago
MediaBrowser.Model	Merge pull request #4326 from crobibero/purge-activity-log	4 days ago
MediaBrowser.Providers	Merge pull request #4341 from Bond-009/minor6	4 days ago
MediaBrowser.XbmcMetadata	Optimize Substring and StringBuilder usage	3 months ago
RSSDP	Minor improvements	2 months ago
apiclient	Fix feed name and url	28 days ago

The Free Software Media System
jellyfin.org
jellyfin csharp dotnet hacktoberfest
Readme
GPL-2.0 License

Release 10.6.4 Latest
on Aug 30
+ 32 releases

opencollective.com/jellyfin

No packages published

+ 483 contributors

● C# 99.6% ● Other 0.4%

Figure 5 - The Jellyfin server repository on GitHub

Finally, the Jellyfin website also hosts a blog that serves as a one-way communication platform for new initiatives and topics from the work being done in the project (*Blog - Jellyfin: The Free Software Media System*, n.d.). The blog usually has in depth posts with screenshots and videos to show progress being made across the different client and server repositories, and elaborate in length on either why code changes are made, what needs to be done in the future or simply talk about some of the logistical choices being made on various platforms.

4.2. Taskcafe

Taskcafe is a self-hosted project management tool. It uses the Kanban framework as its inspiration and allows the user to do task sorting and filtering, add colours and labels, due dates, assign members and manage checklists in a convenient setting. There is currently only one main contributor to the project, namely the founder of Taskcafe. In contrast to Jellyfin he is doing all the implementation work by himself but does have a Discord group sorted into different channels at his disposal.

4.2.1. Communication Channels in Taskcafe

Discord is an instant messaging service similar to Matrix and has many of the same features such as channels, mentions, video and voice chat and the like. Here the project has several channels set up for the community to interact in.

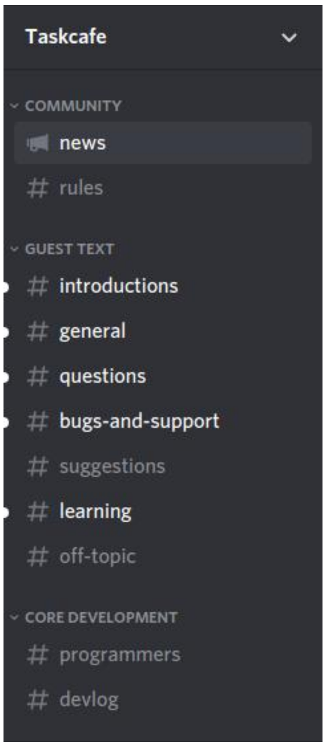


Figure 6 - The channels in the Taskcafe Discord server

The project is hosted on GitHub (*Taskcafe Github*, n.d.), and this is where issues and pull requests live. In contrast to Jellyfin Taskcafe has no legacy code, no history of a fork or anything of that nature. Instead it is one of the many sprouting seeds that exist in the FLOSS world, a relatively new project still in its infancy.

Even though the Taskcafe project is not large enough as of yet to warrant a creation of its own subreddit, it has been posted on many subreddits engaged in FLOSS activity (thenightwolf51, 2020), and has as a result seen the aforementioned discord channel grow in members. This has created a unique scenario where a solo developer sharing is code suddenly is also managing design suggestions and discussions.

5. Methods

This section covers how empirical data was collected for this study with the goal of creating a holistic understanding of the involved FLOSS projects. The data collection methods, namely interviews, are presented. Then the distinction of doing interviews in a distributed setting is touched upon, while also introducing the concepts of contextual inquiries and design interventions. Lastly, how the analysis of the qualitative data was done is presented and a list of data sources is provided to act as a general overview.

5.1. Data Collection Methods

The data collection methods used to inform this case study draw on established methodology from the field of HCI. The core ideology of these methods is based in user centred design, and by extension on the concept that “everyone is designing most of the time — whether they are conscious of it, or not.” (Nelson & Stolterman, 2012, p. 1). It is this notion that to a large degree drove the intervention into these FLOSS projects, in other words to uncover how these imperceptible design choices are made and how they impact their projects and products. Here, it must be said that calling this paper a case study is premeditated as it follows Lazars’ model of exploration, explanation, description, and demonstration (Lazar, 2017, p. 159).

Starting with exploration, to uncover these ingrained values and project specifics, it was clear from the very beginning of the project that a qualitative approach was well suited to the task. Firstly, it was scientifically prudent to verify the previous studies and their validity given the fast-paced technological evolution and nature of the field. Secondly, given that these findings were proven to still be correct in the contemporary landscape the validation was required to meaningfully address possible gaps in the literature and to inform a possible research prototype to drive further findings. Lastly, given the considerable difference from project to project qualitative enquiry was seemingly the only way to ascertain the particulars of each project, in a holistic way.

Eight semi-structured interviews were planned, conducted, and transcribed for further analysis. Seven of these interviews were conducted with the Jellyfin project members, and one with the sole developer of the Taskcafe project. The semi-structured interview format was optimal for this scenario as it numerous times resulted in “(...) the possibility of exploring topics in a depth and breadth that may [have been] harder to achieve with fully structured interviews" (Lazar, 2017, p. 199). These two projects were selected as they both represent two unique scenarios, and as such would provide a more nuanced view on the diversity of these projects in order to "(...) understand the needs and challenges presented by [this] particular situation" (Lazar, 2017, p. 190). An interesting consideration for FLOSS projects in this case, was that the need for focused and contextual interviews were lowered by the fact that the technology choices and process stack of both projects were fully open and readily available for gathering examples to support the interview findings, a fortunate side effect of their free/libre ideology.

Additional data was collected during the prototyping from the same developers that participated initially, and in addition the prototype was posted on selected social media channels, and a forum to gauge the initial reaction to the prototype. The drawback of this qualitative, interview heavy approach was the increased requirements for analysis, as "(...) transforming raw notes and recordings of open-ended responses to broad questions can take a great deal of time" (Lazar, 2017, p. 188). Additionally, interviews by definition rely on the interviewee's recollection of experiences and thoughts. By having their replies "(...) one step removed from reality" (Lazar, 2017, p. 188) one must consider the possibility that the answers would look completely different under for example the lens of an ethnographic observation of the described situation. While an ethnographic approach to understanding FLOSS design habits could reveal interesting discrepancies between said and done, the decentralized nature of the project would make direct observation difficult. Furthermore, while it is an abductive mode of reasoning it would seem that ethnographic observation might also impede too much on the projects, as the contributors already were busy with numerous daily tasks both on their day job and in their respective projects. While the logistics of direct ethnographically inspired observation would be difficult, a more important consideration was the ideological ramifications and the privacy concerns of these individuals. FLOSS participants are, in personal experience with the field, often ideologically driven and hold in high regard the user's freedom and privacy when it comes to software. With that in mind, ethnographic inquiry would likely need recontextualization and rethinking in order to work in this context.

5.1.1. Online Interviews and Methodology

As mentioned above, the online nature of these communities has both opportunities and challenges tied to the process of studying them. While “(...) exploration of online content and activity can provide deep insight into how people communicate, create communities, learn, and interact online” (Lazar, 2017, p. 416) certain challenges also occurred during the interviews and data collection that warrant methodological reflection. While “(...) working
with domain experts who are geographically distant, face-to-face conversations may simply be too expensive to arrange” (Lazar, 2017, p. 216) and as such the affordances of this type of interaction must still be included in the holistic image of this study. In the case of this study, 4 of the 8 online interviews, and all of the feedback was collected using various instant message (IM) platforms, while the remaining four were conducted using voice-chat. This led to a varying degree of answers, some answers taking several minutes to formulate and send, while in some cases warranting asking contributors to expand on their answers. One such example could be “if a participant in an instant-messaging conversation provides a short answer after having been typing for quite some time, [in which case] you might consider asking them to clarify their thinking” (Voida et al., 2004 in Lazar, 2017, p. 218). In addition, in certain cases pacing issues meant that interviews could span days, or even weeks which obviously impacts the quality of answers. Yet, in the grand scheme of FLOSS and its online nature, the general results from the interviews were that the IM based interviews resulted in “(...) more openness in responses” (Walston and Lissitz, 2000 in Lazar, 2017, p. 219) as also found in previous studies in similar conditions.

5.1.2. Contextual inquiry & Design interventions

One approach to supporting FLOSS communities would be to recontextualize the role of the designer as someone who fosters discussion and drives involvement. The question then becomes how best to engage in these practises and how to effectively understand and intervene when needed. Here, inspiration can aptly be drawn from theory supporting contextual inquiry as one possible way of doing this. Additionally, theory on design interventions reveal certain tropes that will need designer attention if the goal of both learning about and creating stronger end user experiences is to be achieved.

Contextual inquiry is “(...) interviews combined with observations, where the goal is to construct a rich picture of the actual work situation: roles, responsibilities, problems with the work and existing tools, and so on” (Löwgren & Stolterman, 2004, p. 66). Specifically, three principles are used to guide contextual inquiries namely context, partnership, and focus. ‘Context’ is related to understanding what people actually do when they work. ‘Partnership’ deals with the inclusion of future users, as they are “(...) experts in their work and should be jointly responsible for the inquiry” (Löwgren & Stolterman, 2004, p. 67). Lastly, ‘Focus’ deals with how subjectivity implies that everyone will notice different things in the same situation, and that awareness about this fact is most important to extending the “(...) total focus of the inquiry” (Löwgren & Stolterman, 2004, p. 67).

Somewhat related to contextual inquiry, design interventions “(...) are increasingly seen as a research method, not to test a prefigured solution to a defined problem, but to enable new forms of experience, dialogue and awareness about the problematic to emerge” (Halse & Boffi, 2014, p. 2). One of the primary ideas of design interventions is that the quicker an idea becomes tangible, it can be critiqued and analysed which will aid in approaching a good solution. In other words, the goal of design interventions lies in “(...) appreciating a problem through attempts at solving it” (Andersen et al., 2011, p. 1). Both of these methods are quite clearly beneficial to understanding and working with FLOSS projects. Not only do these approaches result in empirical findings to support the field of HCI in general, they also allow for a rethinking of previous studies while supporting the creation of a design research prototype to further drive new and continuous findings.

5.2. Data Analysis

In analysing the data, some sort of categorizing was required to meaningfully extract the core findings. For this “(...) the process of coding, which assigns labels to observations from text or other qualitative data forms.” (Lazar, 2017, p. 299) was a fitting choice. Specifically, the data was sifted through several iterations of coding which resulted in “(...) developing themes and then key concepts” (Lichtman, 2013, p. 244). As qualitative research is inductive in nature, analysis of qualitative data also embodies these inductive and iterative attributes (Lichtman, 2013). As such, the analysis was conducted in parallel with the data collection as described above, which meant that new findings and new questions to ask naturally emerged from the analysis. In other words, the border at which the data collection stops, and the analysis starts are somewhat blurred by this "(...) process that moves between questions, data, and meaning" (Lichtman, 2013, p. 255).

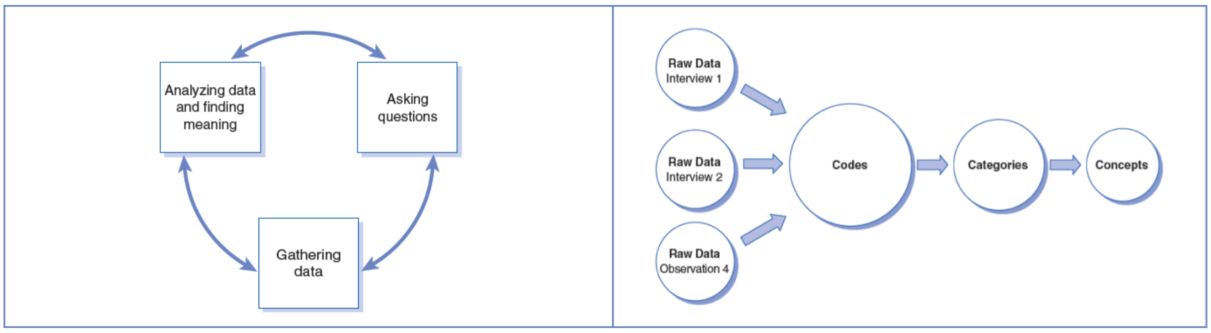
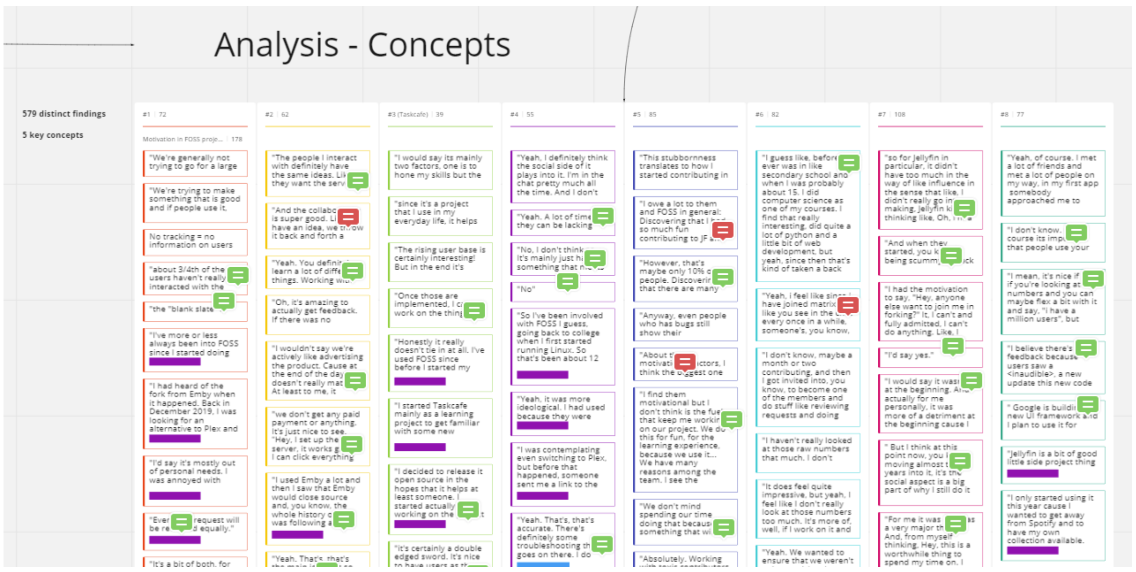


Figure 7 - Relationship between questions, data, and meaning and the "three Cs" of data analysis (Lichtman, 2013, p. 252 - 255)

Practically, the online whiteboard platform "Miro" was used to work with the data as it allows for assigning labels, comments and colours to cards of text. This enables a workflow wherein one can easily move around findings with comments and labels attached describing its properties, without disrupting the surrounding findings or the layout. Additionally, this allows the user to search for both contents in the quotes themselves but also comments, and labels. Specifically, a Kanban board was used to do this, and while this is not the intended use of a Kanban board, its structure worked very well for handling a lot of data, while retaining meta-information about prior categories or attachments while narrowing down categories and findings into concepts. As an example, after the initial coding where one goes "(...) from responses to summary ideas of the responses" (Lichtman, 2013, p. 252) 10 distinct categories were initially found. However, after moving from categories to concepts only 4 remained, yet all of these findings maintained their meta-data and as such serve as a rationale or log for the choices made during the analysis.



Analysis - Concepts

579 distinct findings

5 key concepts

#1 | 72

Motivation in FOSS proje... | 178

"We're generally not trying to go for a large

"We're trying to make something that is good and if people use it,

No tracking = no information on users

"about 3/4th of the users haven't really interacted with the

"the "blank slate

"I've more or less always been into FOSS since I started doing

"I had heard of the fork from Emby when it happened. Back in December 2019, I was looking for an alternative to Plex and

"I'd say it's mostly out of personal needs. I was annoyed with

"Ever request will be re d equally."

"It's a bit of both, for

#2 | 62

"The people I interact with definitely have the same ideas. Lik they want the serv

"And the collabo is super good. Li have an idea, we throw it back and forth a

"Yeah. You definit learn a lot of diffe things. Working wit

"Oh, it's amazing to actually get feedback. If there was no

"I wouldn't say we're actively like advertising the product. Cause at the end of the day doesn't really mat At least to me, it

"we don't get any paid payment or anything. It's just nice to see. "Hey, I set up the server, it works g I can click everything

"I used Emby a lot and then I saw that Emby would close source and, you know, the whole history o was following a

"Yeah. That's, that's

#3 (Taskcafe) | 39

"I would say its mainly two factors, one is to hone my skills but the

"since it's a project that I use in my everyday life, it helps

"The rising user base is certainly interesting! But in the end it's

"Once those are implemented, I c work on the thing

"Honestly it really doesn't tie in at all. I've used FOSS since before I started my

"I started Taskcafe mainly as a learning project to get familiar with some new

"I decided to release it open source in the hopes that it helps at least someone. I started actually working on the t

"it's certainly a double edged sword. It's nice

#4 | 55

"Yeah, I definitely think the social side of it plays into it. I'm in the chat pretty much all the time. And I don't

"Yeah. A lot of time they can be lacking

"No, I don't think It's mainly just h something that n

"No"

"So I've been involved with FOSS I guess, going back to college when I first started running Linux. So that's been about 12

"Yeah, it was more ideological. I had used because they were

"I was contemplating even switching to Plex, but before that happened, someone sent me a link to the

"Yeah. That's, that's accurate. There's definitely some troubleshooting th goes on there. I do

#5 | 85

"This stubbornness translates to how I started contributing in

"I owe a lot to them and FOSS in general: Discovering that I had so much fun contributing to JF a

"However, that's maybe only 10% o people. Discoverir that there are many

"Anyway, even people who has bugs still show their

"About t motivati ctors, I think the biggest one

"I find them motivational but I don't think is the fue that keep me workin on our project. We do this for fun, for the learning experience, because we use it... We have many reasons among the team. I see the

"We don't mind spending our time doing that becaus something that wi

"Absolutely. Working

#6 | 82

"I guess like, before ever was in like secondary school and when I was probably about 15. I did computer science as one of my courses. I find that really interesting, did quite a lot of python and a little bit of web development, but yeah, since then that's kind of taken a back

"Yeah, i feel like since i have joined matrix like you see in the every once in a while, someone's, you know,

"I don't know, maybe a month or two contributing, and then I got invited into, you know, to become one of the members and do stuff like reviewing requests and doing

"I haven't really looked at those raw numbers that much. I don't

"It does feel quite impressive, but yeah, I feel like I don't really look at those numbers too much. It's more of, well, if I work on it and

"Yeah. We wanted to ensure that we weren't

#7 | 108

"so for Jellyfin in particular, it didn't have too much in the way of like influence in the sense that like, I didn't really go in making, Jellyfin ki thinking like, Oh, I'm a

"And when they started, you k being scummy ck

"I had the motivation to say, "Hey, anyone else want to join me in forking?" It, I can't and fully admitted, I can't do anything. Like, I

"I'd say yes."

"I would say it was at the beginning. And actually for me personally, it was more of a detriment at the beginning cause I

" But I think at this point now, you l moving almost t years into it, it's the social aspect is a big part of why I still do it

"For me it was as a very major th And, from myself thinking, Hey, this is a worthwhile thing to spend my time on. I

#8 | 77

"Yeah, of course. I met a lot of friends and met a lot of people on my way, in my first app somebody approached me to

"I don't know. course its imp that people use your

"I mean, it's nice if if you're looking at numbers and you can maybe flex a bit with it and say, "i have a million users", but

"I believe there's feedback because users saw a <inaudible>, a new update this new code

" Google is buildi new UI framework and I plan to use it for

"Jellyfin is a bit of good little side project thing

"I only started using it this year cause I wanted to get away from Spotify and to have my own collection available.

Figure 8 - An excerpt of the data after sorting

5.2.1. Analysis of Non-Interview Data Sources

Some sources were not collected directly from the interviews, which meant that the subsequent analysis of these did not follow the same structure. However, they are intertwined within the process of creating concepts and act as supportive material throughout the analysis. This supportive material includes examples from the projects instant chat platforms such as Matrix and Discord, screenshots from the social media outlet Reddit and finally excerpts of the communication and development that is ongoing on GitHub. These data sources however, were also supportive throughout the data collection and contextual inquiry and aided in verifying the validity of the interviews, and whether the comparison between the statements of the participants and the practices on these various platforms had any interesting discrepancies that might subsequently lead to a better understanding of the problem area.

One limitation of using these textual platforms as additional data sources was that, it became apparent that for the full value and knowledge to be extracted from within these systems, less time had to be spent on the analysis of the interviews and the testing of the prototype. Accordingly, these data sources were only used as supportive material in this contextual inquiry, somewhat in opposition of the idea that such inquiries require a great deal of focus on "(...) demonstrations of how participants complete key tasks" (Lazar, 2017, p. 194)

6. Analysis - The 4 Uncovered Themes

Here follow the core concepts derived from the various data acquired from interviews, communication channels, and the projects specifics. The section is divided into 4 categories wherein each constitutes a core concept that has a high importance for cultivating, driving and implementing design decisions in FLOSS projects.

Concept	Sections	Content
Motivational factors in FLOSS	1. Ideology and FLOSS conviction 2. Social interaction as a motivational factor 3. User types in FLOSS 4. In-Project education 5. The developer playground	*Deals with the different motivational factors uncovered from the interviews with the two projects. Is supported by figures and screenshots that exemplify what was found.*
Resource and project management in FLOSS	1. Roles and skill sets in FLOSS 2. Project history and aligning goals 3. Knowledge management and platform considerations	*This section addresses the structuring of the project with emphasis on how decisions are made, how knowledge is handled and how different roles and skills are distributed in the projects.*
Perception of design in FLOSS projects	1. Barriers for design according to developers 2. Project terminology and external knowledge gathering 3. Subjectivity and design in FLOSS projects	*Deals with the contributors' perception of the design process and user experience changes in general. It seeks to uncover their views on its importance and relating concepts such as terminology and subjectivity.*
Design methodology in FLOSS projects	1. Democracy and consensus in FLOSS projects 2. Feedback mechanisms and testing in the projects 3. Iterations and prototyping 4. Complexity of design in FLOSS projects	*This part of the analysis addresses the explicit project methodology in the two projects. It seeks to unearth how the projects practically go about changing, testing and iterating on new features or issues.*

Table 2 - The 4 main concepts extracted from the data source from the two projects

6.1. Motivational Factors in FLOSS

The first section of the findings covers the various motivational factors that have to be considered as critical in understanding the complex landscape that is FLOSS projects. Firstly, Ideology and important convictions in the projects will be described. Secondly, social interaction and the sense of community will be considered with regards to its impact on the projects as a motivational notion. Lastly, the 'developer playground' is described as another important factor in understanding FLOSS contributors' motivational patterns.

6.1.1. Ideology and FLOSS Conviction

FLOSS projects are unique in the sense that most of the time participants are not compensated financially in any structured way or form. One could even argue that financial incentives might act as a demotivating factor in certain ideologically involved participants in such projects. The notion of financially driven development in a FLOSS project might even be considered an idea that suffers from "(...) mismatched incentives" (Interview #7, Jellyfin). Then, if financial incentives are not the driving factor for continuing to maintain these projects it might not be immediately clear what is.

FLOSS is often thought of as a very ideological entity, and while this is obviously a sliding scale unearthing what real world projects are motivated by is key to determining how to navigate these projects with design contributions in mind. When asked about ideology almost all of the participants in the projects agreed that they were ideologically involved with the Jellyfin project. Even most of the developers that were not primarily motivated by ideological reasons or to the idea of FLOSS itself, at least touched on the notion of helping others through their sharing of code and general openness. For instance the creator of Taskcafe noted that he "(...) decided to release [Taskcafe] open source in the hopes that it helps at least someone" not as the primary motivation, but as an added bonus to creating something that had no other reason to be private and closed. Others were heavily motivated by Emby's switch to a proprietary model. One might think this to be an anti-capitalist ideal yet it would seem that the distinction is more complex than what a surface level analysis would reveal.

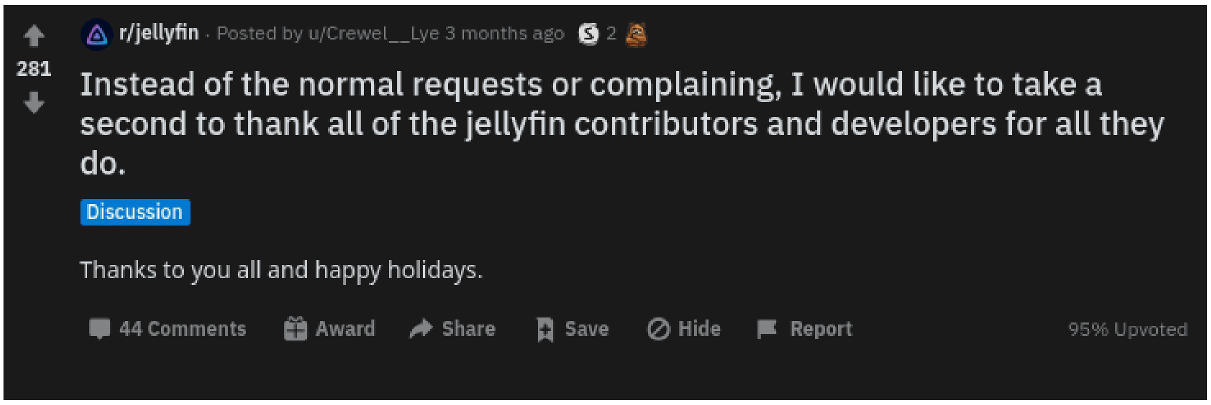
"(...) when they started, you know, being scummy for lack of a better word, I just kinda was like, okay, I've had enough of this." (Interview #7, Jellyfin)

The idea of mismatched incentives was touched upon by another Jellyfin contributor who argued that “(...) this whole war between open source software and closed source software is pointless: both of them can coexist perfectly.” (Interview #5, Jellyfin). This goes to show that the ideological lines are more blurred than one might assume. The idea of making money from software development is not then in itself in opposition to the FLOSS mentality. Rather, the worry is that financial incentives might become an assertive force in these projects. The Jellyfin core members have for instance taken a hard stance against FLOSS bounty sites to avoid these mismatched incentives. They have gone as far as to contact said sites to remove any active bounties that involved Jellyfin. One interviewee argued that if contributors are financially driven then “(...) at least in my view, I don't want the contributor because they're gonna say.. Hey, what can I do that maximizes my money rather than what helps the project” (Interview #7, Jellyfin). In that sense the prevalent ideological mentality in the Jellyfin project is that financial incentives do not belong in FLOSS because it inevitably will damage the sense of community and the democracy of the project by favouring wealthy project members and external users and creating a lesser end product as a result.

6.1.2. Social Interaction as a Motivational Factor

The above sentiment suggests that the idea that maintaining a healthy community and the inevitable social interaction that comes as a result is a motivational factor in itself. Again, this might not be considered as an initial motivational factor in orienting project members towards action, especially in the early days of FLOSS projects. However, it is argued by practically all of the contributors that social interaction has gone on to become one of their primary motivations for contributing and maintaining in a FLOSS project. One contributor especially highlighted that he found it motivational to see the appraisal posts that often appear on reddit.

“We receive a lot of feedback on Reddit with titles like "Just wanted to thank the Jellyfin devs for what they do". It might seem like something silly, but the fact that somebody is actually spending their time in making a thread just for saying nothing but ‘thank you’ means a lot. That's something very difficult to see in commercial projects." (Interview #5, Jellyfin)


281

r/jellyfin · Posted by u/Crewel__Lye 3 months ago 2

Instead of the normal requests or complaining, I would like to take a second to thank all of the jellyfin contributors and developers for all they do.

Discussion

Thanks to you all and happy holidays.

Figure 9 - Example of social interaction as a motivating factor on Reddit - (Crewel__Lye, 2020)

This supports the idea that fostering a community is indeed high on the list of motivational factors. As alluded to above, it might not be the case at the early stages of the project. It takes time for a project to build a sizable community and for some projects it happens slowly and for others it happens overnight. Even in Jellyfin there still hasn't been any planned recreational activities as of yet, however the idea of having socially focused coding nights has been discussed internally (Interview #8, Jellyfin).

In other words, it seems that it differs greatly in the projects with regards to personal preferences and the wish to arrange activities that are planned as social calls. A project member in Jellyfin noted that he considered the necessary social interaction required in the project a detriment in the early stages as he himself had to transition from a more passive FLOSS user mindset, into a FLOSS project member mindset (Interview #7, Jellyfin). It is also noted by another member that the idea that a project must reach a certain size for this community feeling to arise might not be true. He argues that "(...) if a 100 people use it [in this case Jellyfin] and give good feedback and on your work and appreciate you, it doesn't make such a big difference if a million people use it." (Interview #8, Jellyfin). This statement supports the idea that the interactions that matter the most are the ones you experience first-hand, be it an appraisal post on reddit, an interaction in an issue thread or a message on Matrix.

This is also confirmed by looking at the interactions and statements made by the founder of Taskcafe, especially given the size difference of the two projects. Even though he measures the importance of social interaction to be low in the project so far, he also comments on how exciting the Discord chat is and how surprised he is that people stay, help out new users and thank him for his work (interview #3, Taskcafe). Assigning it a low importance might in this scenario also be related to the fact that no other developers has joined the project as a project

member yet. This seems to indicate that within these projects, or more specifically between Taskcafe and Jellyfin, that the closer you interact with other people in turn the more likely you are to be motivated by having social interaction and user feedback.
This might also explain how Taskcafe differs from Jellyfin in this regard. While Taskcafe has a small external community, their developer is doing the bulk of his work alone. Whereas Jellyfin both has the community aspect covered, while also having a separate internal community that supports these closer social relations, interactions, and rewards. As mentioned above a wish to be more social, for instance through dedicated coding nights, has been voiced and would be an interesting move towards creating a stronger sense of community especially internally in the project.

Social interaction is not always a motivational factor though. Several of the interviewees also brought forward the point that it often is a double-edged sword. First of all, as mentioned above scalability of these different interactions are important in this regard as well. A project can only take in so much feedback and after a certain point this information, be it negative or positive, will simply not be seen or considered. This is related to the fact that users of FLOSS projects come with different technical knowledge and backgrounds. One project member noted that he found feedback could be "(...) a bit annoying depending on how the feedback is made and the technical level of the user". (Interview #1, Jellyfin). He also argued that feedback for design often is subject "(...) to suggestions of things that are impossible to make or that would be fairly terrible from a UI/UX standpoint, which we usually have to shut down or rework." (Interview #1, Jellyfin).
This finding was also confirmed in Taskcafe, and it was noted that while the feedback is nice, it also involves an upfront cost of more time spent on bug solving and communication which can be mentally draining in the end (Interview #3, Taskcafe) especially in the case of solo projects as also brought up by other developers in the field (Fisher & Payne, 2021). It would seem that given this ratio between the size of the external community and project member team stays roughly the same, it can be argued that social interaction in general also will result in more time spent on activities that might be resource sinks for the project. Even though this is the case both projects, both agreed that the benefits outweigh the cost as of right now, and that they believe feedback generally greatly improves the end products, and helps in identifying areas of change, issues and generating motivation to continue maintaining the project.

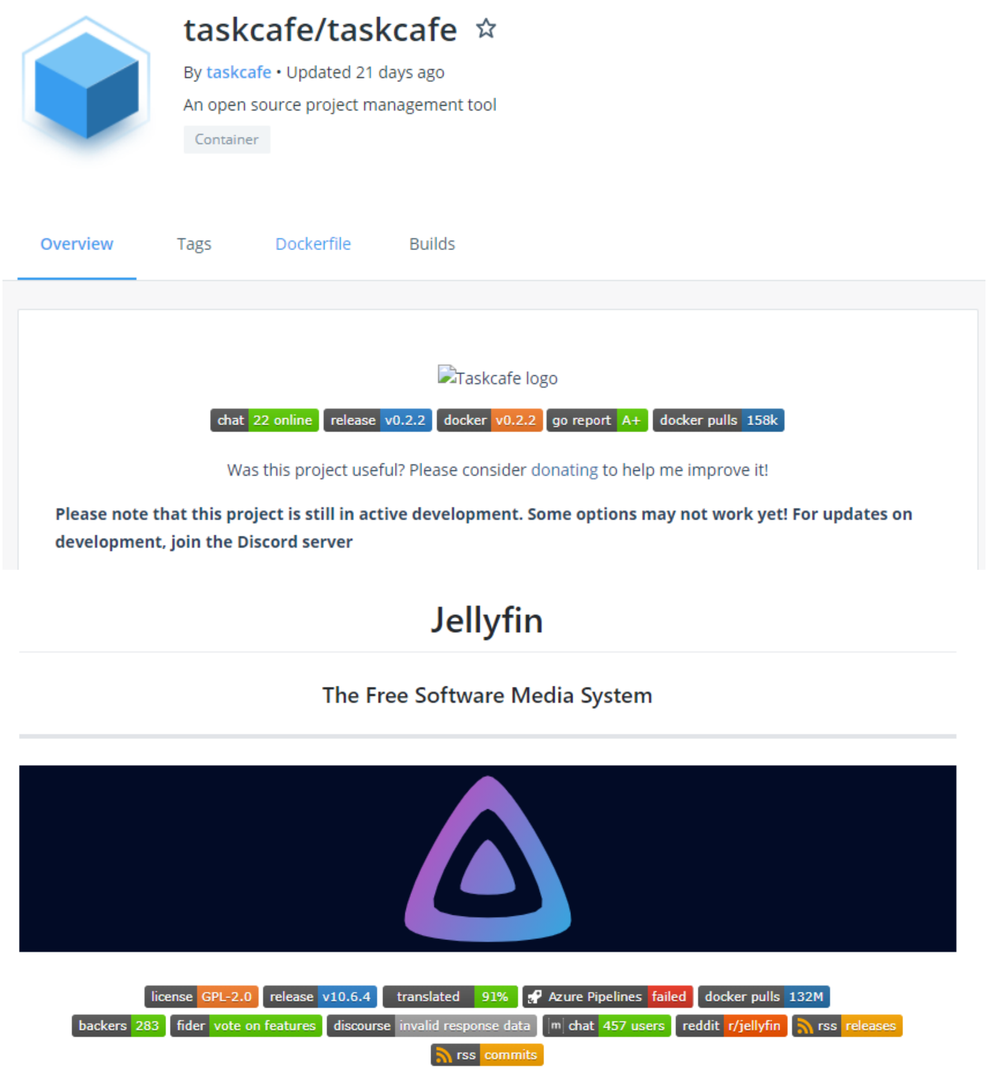
6.1.3. User Types in FLOSS

One finding that also appears to be consistent between the projects with regards to social interaction is the idea of different user types with different participation profiles. Especially the idea of “regulars” or core users are prevalent in both projects. This entails that certain users are more active than others, and provide more feedback, participate in more discussions, and help new users more than the rest of the user base and community. They are also the first users to test beta releases, provide feedback for the bleeding edge changes and so on. These core users are also the ones that are the closest to the project members while remaining external in some sense. One Jellyfin contributor said that “we recognize a bunch of our regulars and those of us who like it engage in off-topic talk with them” (Interview #1, Jellyfin). Another member noted that they do come and go occasionally, however that he did “(...) see a lot of the same names come up a lot, especially users who report bugs” (Interview #7, Jellyfin). This notion is also confirmed in interview #2 where the project member mentions that “(...) there are a couple of users that come back and work a lot of bugs and they do have very thorough bug reports”. (Interview #2, Jellyfin)

On the other end of the spectrum, Jellyfin has seen users who vandalize the project through various means. One project member recounted how the community powered translations that are being done from time to time sees vandalization attempts where false translations that are hard to spot are added. He also disclosed that he had encountered people selling illegal Netflix-like services using Jellyfin as their distribution platform (interview #5, Jellyfin). He alluded to the fact that these users demand or expect features of Jellyfin that are unreasonable for a product meant for home use, without any interest in contributing to the project directly themselves. To this he says that "they're free to do what they want, it's free software, but sometimes it's a bit frustrating to see people earning a lot of money and complaining so hard when we don't earn anything." (Interview #5, Jellyfin).

Some users are also what could be called “one timers”. These users are “(...) people that just turn up, make one pull request and then leave” (Interview #6, Jellyfin). These contributors are mostly motivated by issues found in their own daily use, and initially contribute just with the intention of resolving that specific issue or two that they might struggle with. This description also is in accordance with how some of the now project members joined the project as will be explained in the next section.

The most prevalent user type to emerge from the findings is the passive user. According to a survey done in Jellyfin 75% of all users are simply passive users of Jellyfin (Interview #1, Jellyfin). These users are simply people who use the product without contributing or engaging in any of the social interaction. This suggests that a lot of people are not providing any actual feedback or opinion into the project. This finding especially makes sense when looking at some of the impressive numbers that can be found when searching for Jellyfin or Taskcafe.


taskcafe/taskcafe ☆

By taskcafe • Updated 21 days ago

An open source project management tool

Container

Overview Tags Dockerfile Builds

Taskcafe logo

chat 22 online | release v0.2.2 | docker v0.2.2 | go report A+ | docker pulls 158k

Was this project useful? Please consider donating to help me improve it!

Please note that this project is still in active development. Some options may not work yet! For updates on development, join the Discord server

Jellyfin

The Free Software Media System

license GPL-2.0 | release v10.6.4 | translated 91% | Azure Pipelines failed | docker pulls 132M

backers 283 | fider vote on features | discourse invalid response data | chat 457 users | reddit r/jellyfin | rss releases

rss commits

Figure 10 - Jellyfin and Taskcafe in numbers

As seen above, both projects have relatively impressive numbers. Namely, Taskcafe with 158.000 docker pulls and Jellyfin with 132 million docker pulls at the time of writing. It must of course be considered that these numbers very likely do not represent the actual user numbers, rather the amount of people that have at least tried the software. Even so, reaching 130 million people in some way or form is no small feat. Given this many people at least try the product it resonates with the finding that few people consider going into the project and report bugs or create pull requests.

6.1.4. In-Project Education

Another finding based in and supported by the idea that these numerous types of people are interacting with each other is the concept of in-project education. Several of the project members argued that best practices are considered when implementing code, and here educational conversations are taking place. For instance, one participant said that “#1 is much more experienced than me. So normally it's him educating me about, you know, a better way to do something. But that's alright. I feel that that's really good." (Interview #6, Jellyfin). This quote seems to indicate that it's not necessarily code quality or similar technical aspects being discussed and taught in the project.

However, it does seem that design discussions are predisposed to be more subjective and routed in the contributor in question’s own view on the matter. One such example comes from one of the primary contributors on the android client that recounts a discussion regarding how notifications were handled with regards to music (Interview #8, Jellyfin). Here he describes that while music is playing the corresponding notification should not be removable which made sense for him both from a logical standpoint, but also since this design decision was mentioned in Google's material guidelines. Regarding the more technical aspects, in-project education is also mentioned by several contributors in some shape or form. Many of these findings from the interview mentioned a constant back and forth and discussion on features, problems and the like in different ways and at different times. One such quote is the following from a contributor regarding his experience with people asking about the inner workings of the different projects. He said that “there is a lot of that sort of back and forth between people, but someone will come in and be like, ‘hey, I'm not really sure how this works or how that works’. and there will usually be a couple of people who do know how it works.” (Interview #7, Jellyfin). This is also touched upon when a contributor said that he “get[s] a lot of help in these parts. And well, I kinda think that it also helps us with doing things correctly." (Interview #8, Jellyfin). This notion of doing things quote-on-quote ‘correctly’ seems to be prevalent in the project, especially with regards to making sure that the code that is merged into the main branches is of high quality. Yet, it does seem that design decisions are subject to more discussion and in general it seems that contributors are likely to foster strong opinions on the design choices even though it allegedly is not their strongest suit. This however ties into the general perception of design decisions and will be discussed in the section aptly named “Perception of design in FLOSS projects”.

6.1.5. The Developer Playground

Finally, one of the motivational factors that was discovered in the interviews are inspired by what can be called “the developer playground”. Several of the contributors noted that they mainly were driven by fixing issues they themselves had with the product. One contributor said that he “(...) started with the intention to fix only a few things. However, after fixing something, something else appeared that I wanted to get sorted out.” (Interview #5, Jellyfin). Another said that he contributed in the beginning “(...) mostly out of personal needs. I was annoyed with some things at first, then it sort of evolved from there” (interview #1, Jellyfin). Others noted that they had a lot of fun “(...) contributing to Jellyfin and pushing my code in GitHub helped me realize that I wanted to go into software engineering." (interview #5, Jellyfin). With the above in mind, it seems that a lot of the developers in the project are entering into FLOSS as, 1) a way to empower their own private use of the software, while 2) learning new skills and 3) having fun doing so. These 3 things seem key to “the developer playground” and might aid in understanding why design contributions might not as easily fit into this context.

To confirm this, the contributors were asked about the user numbers presented above in figure 10. Here participant #5 put it as follows: “I find them motivational, but I don't think it is the fuel that keeps me working on our project. We do this for fun, for the learning experience, because we use it. We have many reasons among the team”. To summarise, it would seem that developer motivations in this regard are also very driven by the fact that FLOSS is a place that can support developer interests, give access to new knowledge while maintaining a no-commitment attitude towards the amount of work being done by each participant. This especially rings true when thinking about how software development might in organizations be a solo activity as put by one of the backend developers (interview #2, Jellyfin). This mindset also constitutes a departure from more traditional software development processes where software developers likely handle most of the execution and implementation, while the actual requirements for the software might come from a client, design team or some other organization entity. In other words, as put by one of the developers “(...) "the “blank slate’ is very exciting” (Interview #1, Jellyfin).

With the above findings in mind it would seem to suggest that most of the contributors follow a similar pattern getting into FLOSS projects. First, a project is approached or even created to fulfil one's own needs, to learn, to have fun or for ideological reasons. Then as more people join the project the sense of community, cooperation and the resulting social interactions or team that is created continue to create a stronger incentive for maintaining the project. The idea that certain people might leave at different stages is supported by for instance what the contributor from interview #1 said. He noted that “(...) usually the people that only care about some annoyances they have in their daily use do ‘hit and run’ pull requests. They fix or implement something they want, get it merged, then we never see them again.” (interview #1, Jellyfin). This might also ring true with regards to the ideological commitment FLOSS projects can embody. People who only are interested in the “it’s free”-mantra might simply not be invested enough in this model for them to stick around a keep working on and engaging in a project community as suggested by interviewee #5.

To summarize, it would seem that while a lot of the developers do voice their opinion that they surely are developing for themselves, as they are users of the software, it would seem that as a project goes on their invest in developing for others grow to be larger if not the largest factor when considering motivation forces in a FLOSS project.

6.2. Resource and Project Management in FLOSS

One of the categories to emerge from the findings has to do with the handling of resources, the project management being done and how decisions are reached, executed, and conceptualized for further development. Also uncovered here was the distribution of different roles in the projects, and the varying skill sets that came to light in the discussions with the project members.

6.2.1. Roles and Skill Sets in FLOSS Projects

Contributor	Role and skill set
#1	The primary maintainer for Jellyfin-vue, the new all-inclusive client developed on the Vue framework. He is noted as the closest thing the project has to a designer (Interview #2, Jellyfin). He is a frontend developer by trade and is also very active in the Jellyfin-web repository.
#2	A backend developer responsible for documenting API calls which has helped a lot of new contributors in understanding the underlying code and its possibilities. He is very hands off with regards to design work but does occasionally share his opinion when asked.
#3 (Taskcafe)	The founder of Taskcafe is a full stack developer. capable of handling both frontend and backend bugs and implementation. He manages the whole project by himself, but he interacts with his community through Discord.
#4	This Jellyfin contributor is an implementation engineer and does both front-end and backend development. He is primarily involved with the client-side development, specifically AndroidTV and IOS. Currently he is "(...) just kind of bouncing around between [...] them." (Interview #4, Jellyfin)
#5	Contributor #5 is also involved mostly in the client-side development. He is a student studying software engineering and has always been interested in all aspects of working with computers.
#6	This project member is also a student and is self-taught when it comes to coding and information technology in general. He is primarily working on the clients, more specifically on the Vue-client with contributor #1.
#7	This project member is one of the founding members of Jellyfin and is an active project manager. He oversees releases, and keeps a broader view on what to include, what needs to be finalized or prioritized in order to get releases finished. He is a system administrator and systems architect. His

	day job is to host different services for internet service providers.
#8	This project member is an android developer, who accordingly mainly works on the android client. He is also a student studying computer science, where he also primarily develops for the Android platform.

Table 3 - The roles of the interview participants

It was with the above table in mind, immediately clear that all of the contributors in the project are developers or at very least technically inclined. Their core competencies vary, however all either work with code related jobs, education or are self-taught individuals with interest in the technologies involved with creating software. One example is participant #1 who specializes "(...) in frontend development, so website integrations, JavaScript and such.". Others are more focused on the backend, such as participant #2 who is "(...) a software developer for a telecom company." (interview #2, Jellyfin). The founder of Taskcafe is similarly "(...) a developer first [and as such] designing components from scratch is not [his] strong suit" (Interview #3, Taskcafe). Interview participant #7 however, holds an interesting role that according to himself has raised a few eyebrows in the Jellyfin project. He accounts that he "(...) kind of calls [himself the] project manager and people are like, 'what do you mean by that?', and I'm like, think like your enterprise workplace." (Interview #7, Jellyfin). According to him the role of project manager is a nebulous term in the day to day activities. He does not do much in the way of code however he "(...) watch what other contributors are doing with an eye towards kind of keeping the project moving towards like the goal of, you know, being better" (Interview #7, Jellyfin). He also manages more actively when the project is nearing a release, where he looks at the existing and new features nearing completion, and as time passes take a more active role in encouraging certain features to be finalized and made ready for inclusion in the stable release candidate (interview #7, Jellyfin). He however, made the point that he holds no real power in the sense that he can mandate what project contributors chose to work on. In a similar sense he does not get to reject changes based on his judgement of their importance. He verifies that he is subject to the back-and-forth discussion process in that if he likes a given change "(...) "it will get my check mark or if I don't like the change, I'll put my feedback in." (interview #7, Jellyfin). In summary, it is clear that the Jellyfin or Taskcafe projects currently do not have anyone doing design contributions that outside of the project work professionally with design activities.

6.2.2. Project History and Aligning Goals

Another consideration that emerged from the findings was that a project's history will affect its resource allocation and focus, especially in what could be considered the early life of the project. A difference can be seen between Taskcafe and Jellyfin with this in mind. Jellyfin has inherited a lot of code from Emby that still is a major challenge to fully document, understand and update. Even though the project size is vastly different between the two as well, it does seem like this notion would hold true for most projects. Several contributors in Jellyfin voiced their opinion that for instance the underlying database had been an issue since day one (Interview #2, Jellyfin; Interview #7, Jellyfin). Even those who did not mention the database specifically mentioned and verified that the legacy code is currently the largest overarching task in the project (interview #8, Jellyfin). This notion also holds true for design elements. One participant noted that "(...) "unlike the new client, where we're building everything from the ground up, the old client still has all the design aesthetics that we inherited from Emby." (Interview #5, Jellyfin). This will no doubt limit the options of the developers with regards to improving the user experience and might also be limiting their own ideas for design.

Taskcafe which is a completely new project does not suffer from this and has as a product never been in the maintenance and repair phase that Jellyfin is facing. This finding seems to verify the importance of having discussions about the choice of legacy code, and its inevitable impact on how easy it is for new contributors to join the project and for adding new features. This especially rings true for design contributions that might not necessarily contemplate the situation in the underlying code, and as a result might be considered a change too difficult to implement or consider. With the above in mind it can be seen how the founder of Taskcafe has the freedom to completely overhaul the user interface between releases as a product of iteration. When asked what exactly had changed, the developer from Taskcafe said that "The entire app 'shell' has changed from the original design" (Interview #3, Taskcafe).

In summary, whether a project is a fork determines the need for good documentation as for instance Jellyfin focused on by documenting the API (Interview #2, Jellyfin). Without the necessary documentation and understanding of the inherited code, getting support from new contributors is increasingly more difficult as a lot of time must be spent reverse engineering the code base. In addition, any possible design contributions or discussions being had in a project state where general maintenance is the current goal will likely impose on what is currently possible with regards to the backend.

6.2.3. Knowledge Management and Platform Considerations

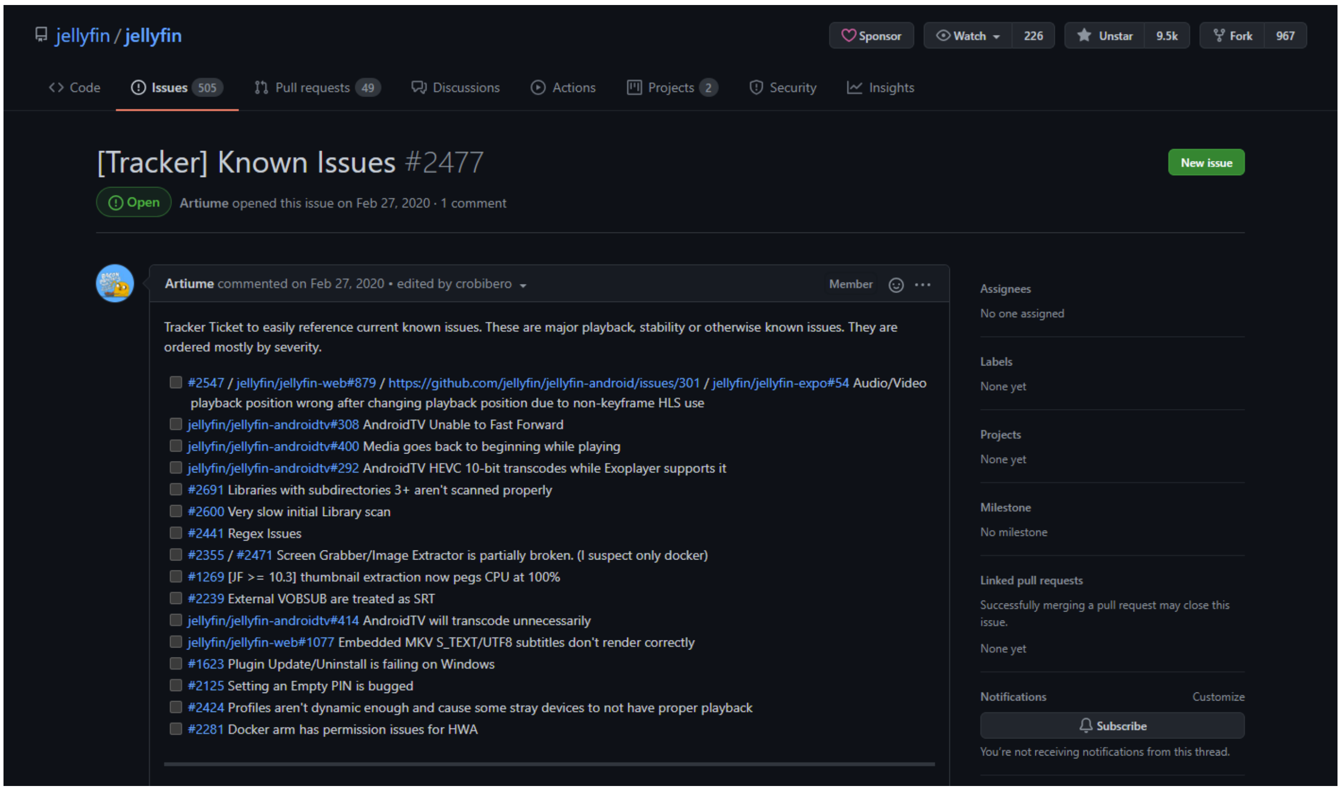
As FLOSS projects receive and create large amounts of data such as discussion, feedback, bug reports and pull requests to name a few. How this data is stored, accessible and worked with determines a lot about what knowledge is available to the contributors in the project, and how effectively the team solves future problems. As this is a major consideration for how the project is managed several of the contributors in the projects had thoughts on the matter when asked.

Type/format of knowledge	**Knowledge platform/location**	**Knowledge use and value**	**Potential knowledge management issues**
Discussions in text, back and forth	Discord, Matrix (Internal/External groups)	Provides possible ideas and acts as a rationale for changes, and future enhancement. Also contains the answers to many questions that are likely asked over and over again.	The type of data is difficult to search and easily create an overview for as it primarily exists in chat channels.
Issue reporting	Github (In Issue templates)	Acts as a centralized location for issues. Avoids duplicate issues, while allowing for gauging the severity of the issue and its possible solution while being easily accessible. Also supports archiving bugs that might not be easily reproducible on unique hardware.	The issues might not always actually be issues and require time to correctly label and address. Also, managing huge numbers of issues are increasingly difficult, which hurts the ability to extract useful information.
Feature requests	Fider	Acts as a larger overview of currently most wanted features.	Fider might not be found at first glance of the project and as such might require

		Provides an easily read and uses a democratic voting system that shows the wishes of the community at the given time.	manual intervention from project members.
Download counts (Jellyfin)	Various app stores	The use of this data is rather limited as it only accounts for limited numerical data retaining to how many users has at the very least tried or downloaded Jellyfin.	It remains clear that this is not seen as an important motivational factor in the various projects, and as such has a limited impact.

Table 4 - Sources of knowledge in the projects, the challenges involved and their importance

A portion of these findings were specifically tied to the characteristics of the various platforms used in the project. One contributor said that “we try to avoid long pieces of feedback in chats like the Matrix rooms, because we'd like it to be archived somewhere for future references.” (Interview #1, Jellyfin). The idea of information being lost is confirmed in that Matrix is not always as easy to search through as for instance GitHub.



[Tracker] Known Issues #2477

Open Artiume opened this issue on Feb 27, 2020 · 1 comment

Artiume commented on Feb 27, 2020 • edited by crobibero

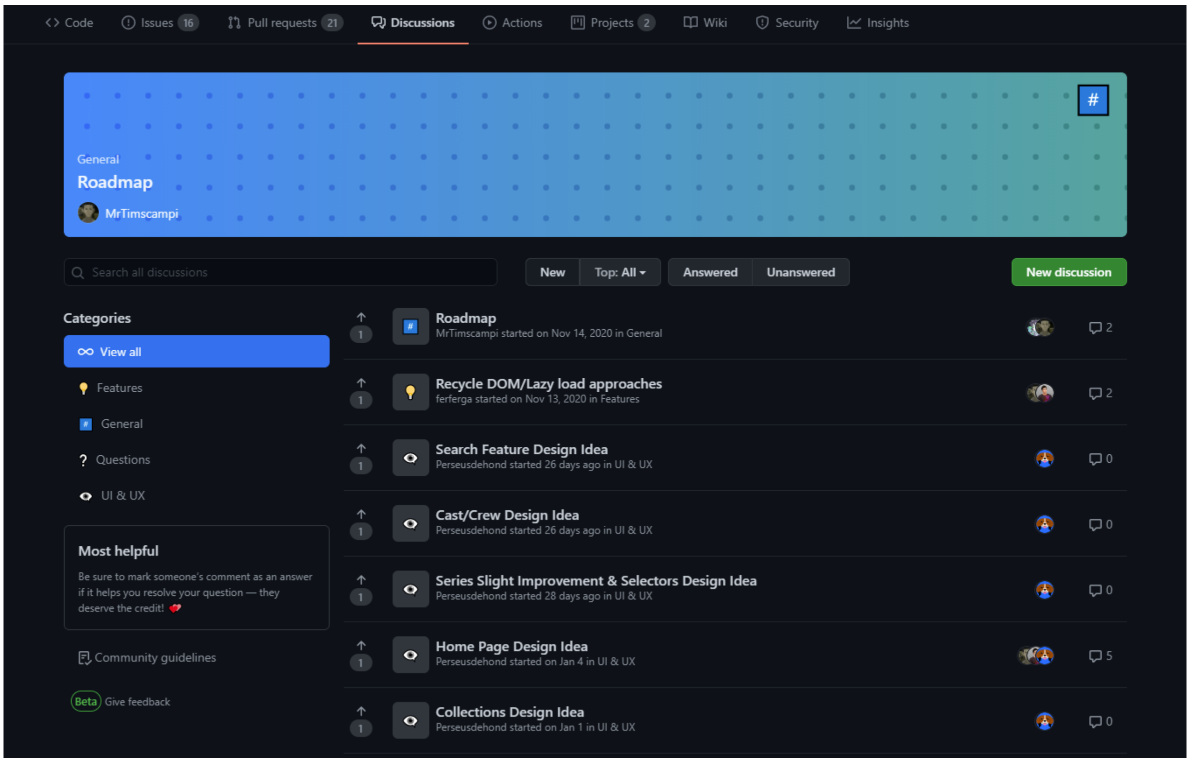
Tracker Ticket to easily reference current known issues. These are major playback, stability or otherwise known issues. They are ordered mostly by severity.

- [ ] #2547 / jellyfin/jellyfin-web#879 / https://github.com/jellyfin/jellyfin-android/issues/301 / jellyfin/jellyfin-expo#54 Audio/Video playback position wrong after changing playback position due to non-keyframe HLS use
- [ ] jellyfin/jellyfin-androidtv#308 AndroidTV Unable to Fast Forward
- [ ] jellyfin/jellyfin-androidtv#400 Media goes back to beginning while playing
- [ ] jellyfin/jellyfin-androidtv#292 AndroidTV HEVC 10-bit transcodes while Exoplayer supports it
- [ ] #2691 Libraries with subdirectories 3+ aren't scanned properly
- [ ] #2600 Very slow initial Library scan
- [ ] #2441 Regex Issues
- [ ] #2355 / #2471 Screen Grabber/Image Extractor is partially broken. (I suspect only docker)
- [ ] #1269 [JF >= 10.3] thumbnail extraction now pegs CPU at 100%
- [ ] #2239 External VOBSUB are treated as SRT
- [ ] jellyfin/jellyfin-androidtv#414 AndroidTV will transcode unnecessarily
- [ ] jellyfin/jellyfin-web#1077 Embedded MKV S_TEXT/UTF8 subtitles don't render correctly
- [ ] #1623 Plugin Update/Uninstall is failing on Windows
- [ ] #2125 Setting an Empty PIN is bugged
- [ ] #2424 Profiles aren't dynamic enough and cause some stray devices to not have proper playback
- [ ] #2281 Docker arm has permission issues for HWA

Figure 11 - Example of the "Known issues" pinned issue

To take feedback and troubleshooting as an example, on GitHub the troubleshooting steps, the current progress on a certain issue and related issues can with relative ease be found, shared or referenced. Often, if a certain issue has a fix it will likely be found in the corresponding issue as part of the discussion. This allows both project members and users to be equally aware of what major issues might be going on, even outside of their own silo of code. Furthermore, a status label will show whether the issue is open or closed which mitigates the creation of duplicate issues, thereby avoiding polluting communication channels. Comparing this functionality to the troubleshooting channel in the Matrix group, while information is searchable this requires quite some creativity to find exactly what a user might be looking for as the layout is chronologically linear resulting in a lot of searching for information. This is especially true if a lot of time has passed since the discussion in question was initially finalized, since a lot of back and forth will likely have happened since. That being said, the Matrix channels do exist for a reason. The contributor also noted "(...) that it gets confusing for everyone when discussions and legitimate issues are mixed. It creates noise for both purposes and, as such, leads to issues or discussions getting lost." (Interview #1, Jellyfin).

With regards to GitHub certain things have been done to try and mitigate these issues of knowledge getting lost or being unstandardized. First and foremost, issue templates are being used to standardize the required information to create an issue. Secondly, the use of the platform Fider has been made the default way to handle feature requests in Jellyfin "(...) mostly just because of the upvote system that Fider gave us as well as to try to avoid a very major, like backlog of feature requests in our issues board." (Interview #7, Jellyfin). Lastly, the new addition of 'GitHub discussions' has just recently been implemented into the Jellyfin project.



General
Roadmap
MrTimscampi

Most helpful
Be sure to mark someone's comment as an answer if it helps you resolve your question — they deserve the credit!

1	Roadmap MrTimscampi started on Nov 14, 2020 in General	2
1	Recycle DOM/Lazy load approaches ferferga started on Nov 13, 2020 in Features	2
1	Search Feature Design Idea Perseusdehond started 26 days ago in UI & UX	0
1	Cast/Crew Design Idea Perseusdehond started 26 days ago in UI & UX	0
1	Series Slight Improvement & Selectors Design Idea Perseusdehond started 28 days ago in UI & UX	0
1	Home Page Design Idea Perseusdehond started on Jan 4 in UI & UX	5
1	Collections Design Idea Perseusdehond started on Jan 1 in UI & UX	0

Figure 12 - The discussions feature on GitHub in use in Jellyfin-Vue

The discussions tab is a relatively new feature in GitHub that was implemented while the interviews with the project was ongoing. The feature can be used "(...) for conversations that span across projects or repositories and don't belong in a specific issue or pull request. Instead of opening an issue in a repository to discuss an idea, you can include the entire team by having a conversation in a team discussion." (*About Team Discussions - GitHub Docs*, n.d.). This feature had sparked interest in the Jellyfin project when it was still in beta, and now having implemented it the project members hope that it can help in having "(...) everything [...] be centralized in one place, which is helpful." (Interview #5, Jellyfin). In summary it seems that this new tab in GitHub could serve as a place for archiving these discussions, and thereby make the knowledge readily available to all project members and users equally. Moreover, it would reduce the need for extra resources spent on managing topics opened as issues that might benefit from a discussion instead. These discussions are being had anyway, which is why archival, and centralization is of general interest to the project.

Interestingly, the Taskcafe project has done something similar in spite of its smaller size. The founder of the project said that “one of the main reasons I made the Discord, was for a place for users to ask questions on how Taskcafe works without polluting the GitHub issues, which has worked pretty well” (Interview #3, Taskcafe). It would then, with project size in mind, make sense that Taskcafe would be able to make due with just the Discord group whereas Jellyfin’s back and forth communication could benefit from this type of centralization and archival, especially for design decisions where the rationale behind a change might not be immediately clear to another project member, user or new contributor. Even with that in mind, the developer in Taskcafe did express interest in the notion of archiving design rationales. To being asked about whether or not he records his decision making process in designing he said that “I don't, though looking back I kind of wished I had if for no other reason than curiosity” (Interview #3, Taskcafe).
In summary, it would seem that archiving discussions in a place where they are easily accessible for everyone in the project is an overall beneficial idea that is still only getting traction in Jellyfin, and might be absent in many FLOSS projects due to the ad-hoc nature of the work being done.

6.3. Perception of Design in FLOSS Projects

To uncover how one might integrate a diverse design methodology and mindset into FLOSS projects, uncovering the project members views on the concept is fundamental for informing such a design inquiry. This section goes into the developers of the two projects, and what constitutes their perception of design as a whole.

6.3.1. Barriers for Design According to Developers

Firstly, the project contributors were asked about their general perception of design, its importance, and barriers for creating more design influence in FLOSS projects. It was clear from the beginning that almost all the developers agreed that it was a very important aspect of software. One developer said that “In my eyes, it's of equal importance with the quality of the code.” (Interview #1, Jellyfin). However, by extension he also argued that design “(...) takes a low priority for a lot of projects, unfortunately”. This developer argued that a barrier for strong design choices in FLOSS would be to ensure that “(...) more developers take up an interest in this. Perhaps stress the importance of getting some UX concepts when training developers. Getting developers to understand that UX is just as important as code and should be treated with the same focus and care” (interview #1, Jellyfin). Interestingly he still argued that the most important aspect for him in FLOSS projects is ensuring high code quality. He argues that having code of high-quality means that it's easier to approach, which will result in more contributors and a more active project overall. This notion is also confirmed by the founder of Taskcafe that said that he does not [...] ever plan a design with those things (for instance usability) specifically in mind” (Interview #3, Jellyfin). His first priority was also implementing the basic features and such before planning new goals for the project. He does however say that getting, for instance, UX designer feedback would be helpful for his project. (Interview #3, Taskcafe). Contributor #4 also argued that “it would be nice to kind of have a push through that and get to a point where we're happy with the architecture and to have a stable, solid, or release where we can kind of get to that point where we aren't making huge architectural changes” (Interview #4, Jellyfin.). In other words, these findings again confirm the idea that code quality is the perceived underlying basis for project success.

Some of the other developers outright said that “(...) it's not really a thing that I really think about.” (Interview #2, Jellyfin). This developer argued that his concept of design was more technical in nature. One such example would to be the ongoing database issues in Jellyfin that very much becomes as user experience problem in that movies and other media is slow to appear in the frontend when added (interview #2, Jellyfin). The very same developer did however voice the opinion that even though his understanding of such design work is limited he “(...) feel[s] like it would be really interesting to see what they had to offer. [...] I think it would be really neat to have, say all of our applications once they eventually stopped being web wrappers to have like a semi cohesive UX feel to them." (Interview #2, Jellyfin). Interestingly, one of the developers from the Android client argued that their current design efforts are "(...) currently doing fine, but maybe it would get better if you had such people, I can't really imagine it." (Interview #8, Jellyfin). The very same developer did however also argue for well-known design concepts such as the need for coherence, visibility and furthermore the increasing need for such principles in the light of project growth. This seems to suggest that what he has trouble imagining is not the need for design per say, rather how the contribution would work in practice.

Another finding with regards to developer perception of design seems to be influenced by their experiences in their workplace. This has to do with how a traditional organizational trope is that designers provide materials such as prototypes to the developers for implementation, which in some sense excludes developers from making these decisions. Some developers in Jellyfin voiced a concern that design contributions could be seen as "being told what to do".

"Some don't like the organization that having a UX/Usability team provides. A lot of developers see FLOSS projects as a playground after work. Like you work on things that aren't interesting for money, but then you work for fun in the evening. Having more organization and a process [...] might feel too rigid for some" (Interview #1, Jellyfin)

This does seem like it would be a valid concern, however it can also be argued that this is only true if the potential designers in the project engage with the project in a way that confirms that rhetoric or expects full control over the outcome. As seen in the above paragraph developers have shown their interest in seeing exactly what having designers on board the project could offer (Interview #2, Jellyfin). Confirming this, another developer said that he thinks that "(...) nobody, at least in Jellyfin, will ever feel looked down upon by [virtue of] their ideas. Every hand in FLOSS is helpful." (interview #5, Jellyfin). A notion to support this idea, is the fact the Jellyfin already has other positions in the project, for instance the project manager(s) (Interview #7, Jellyfin), that provide non-code contributions. As such, having design contributions coexist alongside these other contribution types does not seem that far away from reality. What seems to be the important discovery with regards to maintaining developer freedom while supporting designers in engaging with FLOSS projects, is that project participants make sure to discuss this dynamic and reflect on how such design contributions are to be benefitted from the most. In this way, everyone is included, and the project can avoid the idea that design contributions are assertive in nature.

In summary, it would seem that the developers' perception of design concepts are overall that they are very important concepts to consider when developing software. However, it does also seem that they do prioritize code quality over for instance design choice, and it might also look like some developers might not understand how designers could benefit the project, or are afraid that their ideals could interfere with what they chose to work on.

6.3.2. Project Terminology & External Knowledge Gathering

Another finding that was uncovered with regards to the perception of design, was that of the terminology used in the project, which informed the study about the knowledge of the developers with regards to external reference and inspirational materials. Firstly, it must be said that a general tendency among the developers were that they had a relatively strong understanding of design and mentioned concepts such as usability, accessibility and user experience as a natural part of the conversation. For instance, one developer said that "(...) usability, I'd say it's about how easy something is to use for the first time. Ideally, it should make sense and be presented in a logical way." (Interview #1, Jellyfin). The same developer also accounted for his views on the difference between user interface (UI), user experience (UX) and the aforementioned usability which he explained by saying "(...) UI for me would be how a screen looks visually. UX would be the different paths the user can take in that screen (and into it or out of it). Usability would be how easy the screen is to understand". Some of the other developers were not as articulate in discerning the subtle differences between these design concepts however, touched on key concepts in how they related in their work. Regarding usability one developer said that he uses his gut feeling to assess a design and that he knows that "(...) "If it's too complicated for me, I know most people will not be able to follow it. I'm a pretty technical user." (Interview #2, Jellyfin). Others had even "(...) read some more general pieces on user experience and user interface design" (Interview #4, Jellyfin) to inform their decision-making process. In summary, it seems that while the terminology is not agreed upon in these projects it is rather advanced for a group of people not working with it professionally in their daily jobs. In addition, it would also appear that the more specific and comprehensive answers in relation to design came from the developers working mostly in, and with the development of the Jellyfin clients. This also seems logical given that the clients support more design work, than the other project repositories.

In relation to uncovering what external knowledge there might be used in these projects, several of the developers noted that they do use best-practise minded design guidelines such as google-material design or the android guidelines (*GNOME Human Interface Guidelines*, n.d.; Google, n.d.), or even human interaction guidelines from other FLOSS projects, for example the Gnome project (Interview #8, #6, #7, Jellyfin). From these mentions of using various guidelines for informing design decisions it can also be seen that there is no clear consensus from a project point of view on terminology, and that any and all terms or concepts derived from these guidelines are only used on an individual user basis. This means that currently these guidelines and concepts such as UX, UI, and accessibility to mention a few might be used differently in each silo of code, or between each contributor. In a similar fashion the goals of Jellyfin itself is also up to interpretation with regards to design, which especially is true given the lack of any real major design overhauls as of yet.

6.3.3. Subjectivity and Design in FLOSS Projects

As alluded to above, subjectivity does play a role in these design decisions. A developer even noted his own initial distaste for a user interface change but found that he might have been biased since he also said that "(...) I've used it for a while now and I'm kind of, you know, over it." (Interview #7, Jellyfin). He also said about design that "It's a very each to their own and it's kind of caused some problems in the past internally in terms of like people, some people want to change some aspects of the UI, some don't" (Interview #7, Jellyfin). The project members were asked about their feelings about how subjectivity takes part in making these design decisions, and if they felt that design is more subjective in nature. One contributor said that "I think there's a mix of subjectivity and agreed upon rules for all of this." (Interview #1, Jellyfin). This subjectivity is even a concern for some developers, for instance in interview #4 where the contributor said that "(...) I think a lot of it just comes down to needing a strong UX and design resource as far as guiding the project in a good direction that focuses on UX and design, as opposed to having many developers with their own sense of what is, or isn't good design." (Interview #4, Jellyfin). He continues to say that he feels the project is lacking some guidelines to align developer views and goals in the design department. One way to address making design decisions based on personal opinion is how the Vue client repository supports their decision-making process. The project makes use of the Vuetify plugin that has prebuilt templates and modules that are built with industry best practises in mind (*Vuetify — A Material Design Framework for Vue.Js,* n.d.). One contributor that works on the Vue client expressed his positive experiences with this, as this way of working insures a baseline of modern and good UX (Interview #6, Jellyfin).

In summary, it seems that Jellyfin could use a more coherent approach to avoid making decisions in each developer silo to support at coherent design experience, which the developer is interested in establishing between their clients (Interview #2, Jellyfin). Also, it would seem that this notion is not really brought up with regards to Taskcafe, however it makes sense as the friction described in Jellyfin is a product of developer to developer discussion which does not take place in Taskcafe as of yet.

6.4. Design Methodology in FLOSS Projects

With the above section addressed, it is now clear what the developers of FLOSS projects think about design efforts as a worthwhile task. However, it is not clear what they do to incorporate design efforts into their activities. This section looks at the project contributor's methodology when it comes to making design decisions.

6.4.1. Democracy and Consensus in FLOSS Projects

One finding that was very consistent between all developers in Jellyfin and Taskcafe from the interviews was the idea that FLOSS and decision making therein is wholly democratic. One developer said that the actual changes are decided by a "majority rules" mindset. He said that it is "(...) completely democratic. Majority wins. Although we always try to discuss every opinion given so many times that, if a minority thought of something but it was convincing enough for the majority, that minority's opinion ends up implementing." (Interview #5, Jellyfin). He provided an in-depth explanation for how they approach these decisions and said the following:

"When there are different opinions among us, we generally follow two mantras:

A) How do alternatives/similar software accomplish this? [...] or B) The contributor who is proposing the changes is the one who choses how to do it. This mostly happens when we see that Netflix, Spotify, Amazon Prime, Plex, etc. Do the same thing in a completely different way and there are a lot of different opinions among us" (Interview #5, Jellyfin)

These quotes support the idea that any external party can influence the project in any way independently given the argument is strong enough. However, rather than a true democracy there is some merit involved in the decision making as it happens. One developer called the project a "(...) sort of a "do-ocracy", as we call it. Meaning that usually, if you really want something done, we expect you to open a pull request for it." (Interview #1, Jellyfin). This would then inherently limit non-code contributions as they, as a currency in the project, do not hold the same value as code contributions. At the same time however, project management contributions have already shown to be accepted in the project which suggests that other types of contribution do exist. Other contributors did however confirm that code is likely the most understood and valued type of contribution. One developer argued that "if you want a feature, don't just be like, 'oh, excuse me. Would you mind doing this?' Go out and code it and, you know, learn what you need to learn to implement it" (Interview #6, Jellyfin).

In other words, it would seem that non-code contributions are accepted in these FLOSS projects. This could include translation work, helping new users or otherwise. Yet, the consideration that code itself is the main type of contribution needed to advance the project might be a detriment to designers and might also be in conflict with the "if you want it, make it" mantra that is present in the FLOSS projects. This is also supported by the fact the developers still argue that the project direction is still mostly influenced by internal members, as they have the best understanding of the underlying code and what is possible (Interview #7, #8, Jellyfin). The internal project members' Matrix channels are also something that is still a factor and to what extent the discussion there shapes the project is not immediately clear. To summarize, it seems that considering the foundation for contributions is important in these projects especially when receiving non-code contributions. Discussions are necessary for project members to agree on how this bias is handled, and designers need to be wary of how they make suggestions and argue for changes as their contribution will be met with the "if you want it, make it" mantra.

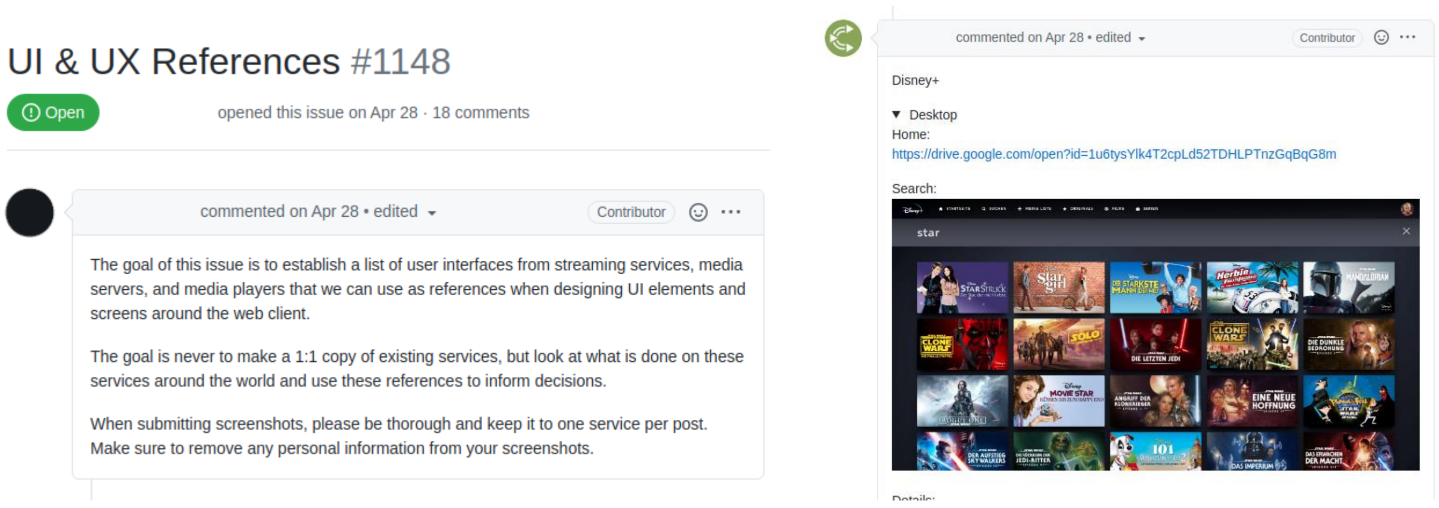
6.4.2. Feedback Mechanisms and Testing in the Projects

In the two projects it seems that there is no formal testing being done. However, after some initial discussion on the topic some findings did suggest that testing is going on, just not in an organized manner. One contributor in Jellyfin said that "We don't have A-B testing or any sort of usage tracking to get information on how users do things, so we mostly fly blind until someone raises an issue." (Interview #1, Jellyfin). This makes sense when considering the open nature of the projects. However, when considering this, it would seem that FLOSS projects are rather unique in that testing is an ongoing process where anyone can file a bug report. This of course requires that the testers are running the latest version, which most won't if the product is used in daily use where the goal most often is stability. However, it does seem like most of the developers all are running the latest build in their day to day use to do real-time iterative testing. One developer accounted how he does not "(...) do any specific testing. It's more just, you know, me using [Jellyfin] at home and then, you know, seeing what I think about it" (Interview #5, Jellyfin). Another said that after testing, developers in between "(...) a wider user base who uses unstable see the changes right away and can also provide feedback." (Interview #5, Jellyfin). In other words, it would seem that a cycle of testing starts with a change being made, the developers testing the change by deploying it in their home systems, then merging to the main branch and then getting feedback from the users running the most recent unstable release. Then of course, these changes eventually end up in the stable release where, yet another user group gets to do a round of bug finding and reporting.

In rare cases developers have also gotten outside help when testing. One developer explained how "(...) for usability. I know we had some people reach out to their colleagues and say, 'hey, can you take a look at this?'" (Interview #2, Jellyfin). Additionally, domain experts have been used briefly for specialized problems such as for making sure that colour-blind users could operate the software without issue (Interview #2, Jellyfin). It does also seem, with regards to getting testers on board, that the project does publicly say when they are needed. One developer explained how “(...) we do sort of posts like before every, before every major release saying, inviting testers to come on and test.” (Interview #6, Jellyfin). There are also examples of certain features that are heavily dependent on outside help, and the log files and feedback from users. One such example happened when “(...) we (The Jellyfin developers) apparently broke live TV, but we don't really have a lot of our, like the core developers using live TV.” (Interview #2, Jellyfin). In this case live TV functionality required specific hardware and therefore requires testers that own said hardware, meaning that this feedback mechanism described above does work. However, it also indicates that it does come with quite large risks such as completely breaking functionality without any single developer knowing.

6.4.3. Iterations and Prototyping

With regards to how these projects handle reiteration and prototyping, various methods have been discovered in the two projects. While the projects largely make it seem like there are no structured prototyping methods in place, and that no agreed upon methodology is in use with regards to for instance mock-ups some structured activities do already take place. Firstly, Jellyfin’s primary way of informing design decisions seems to be by using existing design as inspirational material. This way of informing design was also mentioned by the developer from Taskcafe when he said that he “(...) research the different ways others have designed similar components” (Interview #3, Taskcafe). The way this is structured in Jellyfin is mostly by using pinned issues that contain a large amount of inspirational material such as screenshots or video examples from popular media platforms such as Netflix, Disney+ or Amazon Prime (*UI & UX References - Jellyfin-Web*, n.d.).


UI & UX References #1148

Open opened this issue on Apr 28 · 18 comments

commented on Apr 28 • edited · Contributor

The goal of this issue is to establish a list of user interfaces from streaming services, media servers, and media players that we can use as references when designing UI elements and screens around the web client.

The goal is never to make a 1:1 copy of existing services, but look at what is done on these services around the world and use these references to inform decisions.

When submitting screenshots, please be thorough and keep it to one service per post. Make sure to remove any personal information from your screenshots.

commented on Apr 28 • edited · Contributor

Disney+

▼ Desktop

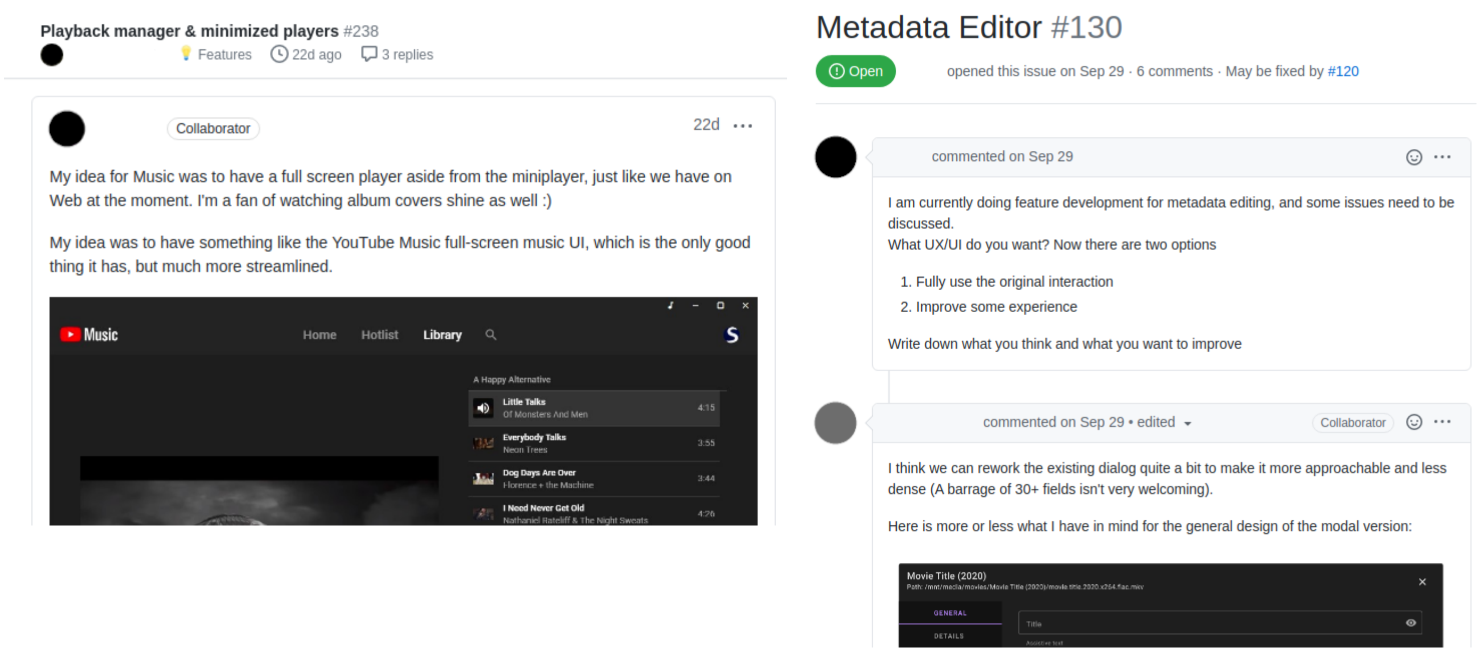
Home:
https://drive.google.com/open?id=1u6tysYlk4T2cpLd52TDHLPTnzGqBqG8m

Search:

Figure 13 - The UI & UX reference pinned issue in Jellyfin-web with an example from Disney+

This way of working shows that the project sees the need for keeping such design references stored and accessible in an organized way, however this requires using the GitHub issue functionality in an unintended way. In other words, this way of working supports the idea that GitHub as a platform could be more powerful in certain ways as confirmed by many of the developers (Interview #7, Jellyfin; Interview #3, Taskcafe; Interview #5, Jellyfin). As mentioned earlier, the GitHub discussions module is a new addition to the project that seeks to solve some of these issues. With the above way of working it does look like the intended use of the discussions module would solve some of these roundabout ways of working with GitHub.

One such way would be to record design rationale and foster discussions surround design as proposed by some of the developers (Interview #1, Jellyfin; Interview #6, Jellyfin). From a methodological standpoint, the new Vue client is spearheading this new more design centric way of working. Here, the developers already are proposing design changes, while presenting reasons and ideas for the change, while providing screenshots to show before and after the change has been implemented. It does appear however as this is a very new way of working that some people are yet to discover the functionality and its use as seen when one user commented “I wasn't aware of these ‘discussion’ areas of GitHub. This is good.” (*Jellyfin-Vue - ‘Lazy Loading #272’*, n.d.). Additionally, it seems that not every design issue is being discussed here, as some still exist in the issues tab, which seems to suggest that the definition of what constitutes a discussion rather than an issue is still not clearly defined in the projects.


Playback manager & minimized players #238

Features · 22d ago · 3 replies

Collaborator · 22d

My idea for Music was to have a full screen player aside from the miniplayer, just like we have on Web at the moment. I'm a fan of watching album covers shine as well :)

My idea was to have something like the YouTube Music full-screen music UI, which is the only good thing it has, but much more streamlined.

Metadata Editor #130

Open · opened this issue on Sep 29 · 6 comments · May be fixed by #120

commented on Sep 29

I am currently doing feature development for metadata editing, and some issues need to be discussed.
What UX/UI do you want? Now there are two options

1. Fully use the original interaction
2. Improve some experience

Write down what you think and what you want to improve

commented on Sep 29 • edited · Collaborator

I think we can rework the existing dialog quite a bit to make it more approachable and less dense (A barrage of 30+ fields isn't very welcoming).

Here is more or less what I have in mind for the general design of the modal version:

Figure 14 - Example of design issues being discussed both in issues (right) and discussions (left)

In summary, structured activities are already taking place in the Jellyfin project especially, however it does seem that jellyfin-vue is the testing ground for this new way of working and that adoption is still something that is being worked out as what constitutes a discussion or an issue is being worked out between the contributors.

6.4.4. Complexity of Design in FLOSS Projects

At this point the methodology of these FLOSS projects are largely uncovered and it seems that while many findings have emerged with regards to how they perceive, approach and implement design changes there still are some concepts that live in the seam between perception and methodology. Namely, if there is any perceived and methodological difference in how FLOSS projects approach design as a whole with regards to the inherent complexity of design work. One example that could support this idea is that maybe design work is just inherently more complex to approach and requires more planning as specified by a project member when he accounted that “(...) people usually don't report them (design and user experience issues) enough, unless they're really egregious. So, it's harder to keep track of in the sense that it happens less often, so we might have some that we don't know." (Interview #1, Jellyfin). In other words, the issues themselves might be harder to report as they might strictly be thought about as “enhancement” rather than a critical issue. Compared to for instance backend issues that very often are very clearly defined problems where a certain unit within a program fails or otherwise needs fixing to allow a certain use, design is more defined by the individuals existing knowledge and preferred approach to a certain task. As it has already been uncovered that both knowledge and subjectivity are factors that are present in the projects this again lends credence to the idea that design issues might be harder to approach and in this case report, resulting in an uneven balance of reporting for design related issues.

As mentioned above, there is also an inherent “solution mindset” in the FLOSS projects. This appears to affect the attempted resolution of design issues. One of the developers explained it as “(...) people generally try to provide solutions instead of explaining the issue at hand. It makes it harder for us to understand their issue and decide if their solution is really suitable or if it would be another issue in itself.” (Interview #1, Jellyfin). Explained this way, reporting design issues or discussions does involve a certain complexity and more often than code-centric issues have a need for further discussion and planning.

These considerations are in themselves complex in that it seems like a paradox exists regarding design, between what is thought and done. On one hand the developers recognize the value and importance of design decisions and admit that they often take a low priority in FLOSS. However, as shown with regards to their design methodology it seems that they do have several initiatives that take place that seem to align with processes normally associated with industry design best-practices. This suggests that a worthwhile exercise would be, as already mentioned by project members, to increase the amount of discussions being had surrounding design in general. Something that has already started during the inquiry of this study, namely in the Jellyfin-Vue repository, but has yet to spread to the rest of the repositories and largely FLOSS in general. Through the use of frameworks that support having design front and centre when developing, such as the Vuetify framework, some of these concerns are mitigated and as a result “(...) the choice becomes how you combine them (design elements) together rather than how you like deal with what a button should look like” (Interview #7, Jellyfin). With this initiative being taken in the Jellyfin-Vue repository it appears that the project is trying to move the discussions being had to a higher level, and avoid the smaller UI specific discussions that in this case is largely mitigated through the use of Vue and Vuetify. In other words, the efforts to address the complexity of design through discussion is well under way, in some cases unbeknownst to the developers themselves.

7. ‘FLOSS-UX’; Proposed Design and Testing

This section covers the steps in creating a design research prototype for addressing the issues uncovered in the analysis. The first part of the section describes how the findings from the analysis and existing theory were addressed and thought into the prototype. The second part of this section covers the testing of the prototype and the resulting changes that might benefit the effectiveness of the prototype in the future, both as a vessel for generating more knowledge about UX practices in FLOSS projects but also to help create awareness about the root causes of these issues throughout the field.

7.1. Presenting the Prototype

The initial steps for creating the prototype was rooted in looking at existing ventures into the field, too see how both the available research had addressed issues and where possible improvements and knowledge might be gained most effectively. With solutions in mind some general categories for these inquiries were available from previous studies. The conclusions and design prospects of the existing research can loosely be structured in the following categories and sub-categories:

1. *The Shift from the System-Centred Design to the User-Centred Design*
2. *Developers and HCI Experts’ Motivation for Participation in FLOSS Projects*
3. *The Importance of Automatic Usability Testing Tools*
4. *Suggestions on How to Improve Usability*
    a. *The Commercial Approach*
    b. *Creating a Usability Discussion Infrastructure*
    c. *Education and Evangelism*

(Despalatovic, 2013, p. 962)

As “(...) designers rarely work alone on design projects” (Löwgren & Stolterman, 2004, p. 32) some of these design opportunities were naturally out of reach for a single designer, and out of scope for this study. However, some suggestions from the corpus of the existing knowledge did immediately appear possible to implement and use to uncover more knowledge and inspire future work in the field. Namely, the notion of creating a design discussion infrastructure proposed as a possible solution for fostering more UX awareness in FLOSS projects (Terry et al., 2010). The use of dedicated platforms for handling design activities, such as the design-by-blog approach, has undoubtedly risen in popularity since this methodology was uncovered in the 2000’s. However, calling this an exclusive design activity might be somewhat misleading given that these blogs also support more system-centric topics and might often be used not necessarily due to their affinity for handling larger and more visually heavy work. Additionally, not all of these project blogs support comments and therefore any discussion surrounding these posts will have to be moved to a separate forum such as Reddit or Matrix, thus decreasing the effectiveness of the design centric benefits of the format. With this in mind, it seems there is still plenty of room in this field for rethinking the format, and subsequently how to support both the gathering of knowledge for the field of HCI and in the transmission of user-centred ideals into these projects.

Closely related to this, as found in the analysis is that most of the developers that engage in user interface design already utilize design resources that cover best-practices for contemporary UI design, such as Google’s Material Design, Apple's HIG and inspiration from competing proprietary products. In other words, this means that “(...) one of the most immediate ways the FOSS community can improve its practices is simply for it to be more aware of what others are already doing to improve software usability.” (Terry et al., 2010, p. 1007). The prototype presented in this section is just that, a platform to support design interventions in gathering and facilitating discussions and creating awareness about important concepts for creating stronger UX in FLOSS projects.

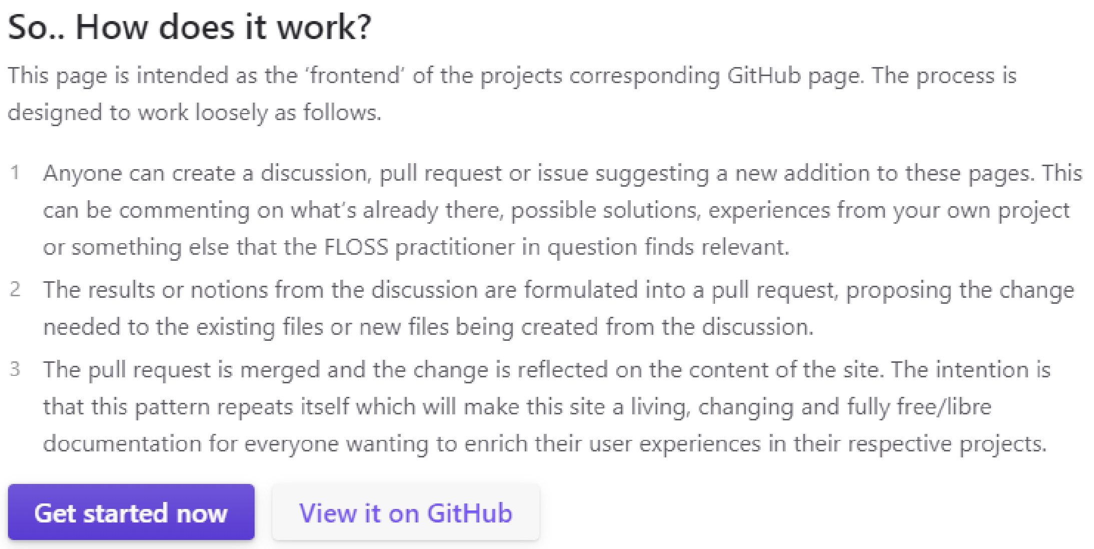
7.2. Striving for Awareness; Walkthrough of the Prototype

This section covers the process and the various sites and systems involved in the prototype and its functioning. Given the iterative nature of this design research prototype, both its visual appearance and content might have slightly deviated from the content of this paper as a result of more contributions being added to the project at a later date. Below a link, and a QR code to the prototype can be found.

Link to the Prototype

7.2.1. The Goals of the Process; 'Use before Use'

The prototype is both a FLOSS project in and of itself, which is hosted on GitHub and acts as a documentation piece that allows FLOSS participants to contribute their knowledge and experiences from their respective projects with regards to design, while learning about knowledge from this study and other projects. The site is based on the 4 categories from the analysis and attempts to convey the findings of the field in a condensed and easily accessible manner, that is familiar to the participants of these FLOSS projects. Specifically, this is done through inviting FLOSS projects to participate via the GitHub discussions functionality, which facilitates communication and attachment sharing while remaining open, approachable and constitutes a simple way to manage and reference these discussions. After a new discussion has been contributed to the project, the key points are added onto the main site and the unique discussion number is referenced to visualize the immediate impact to the body of the documentation. In summary, this prototype seeks both to present, discuss and support enculturation and cross-project knowledge sharing in one continuously maintained designer driven project.



So.. How does it work?

This page is intended as the 'frontend' of the projects corresponding GitHub page. The process is designed to work loosely as follows.

1. Anyone can create a discussion, pull request or issue suggesting a new addition to these pages. This can be commenting on what's already there, possible solutions, experiences from your own project or something else that the FLOSS practitioner in question finds relevant.
2. The results or notions from the discussion are formulated into a pull request, proposing the change needed to the existing files or new files being created from the discussion.
3. The pull request is merged and the change is reflected on the content of the site. The intention is that this pattern repeats itself which will make this site a living, changing and fully free/libre documentation for everyone wanting to enrich their user experiences in their respective projects.

Figure 15 - The process as described on the FLOSS-UX website

The intended outcome of this process is manifold. For one, it is clear that both the existing theory and the analysis suggests that there still is a strong need for the field of HCI to generate and collect empirical knowledge in this field. The unique characteristics of each project makes generalisation difficult, which means that obtaining knowledge requires empirically founded case studies or design inquiries. Looking at FLOSS projects and how much they have changed just in the past decade it is clear that one risk is that the pace of the field might be moving faster than these inquiries can sustainably generate holistic knowledge, both for the sake of the general field of HCI but also for the projects themselves. As a response to this, this prototype is in and of itself a 'design after design' approach to the situation, a project that is meant to follow the evolution of the field and reflect the challenges within both for the designers and researchers looking to understand the challenges, but also for the participants that currently practice a mostly system-centric mindset.

Secondly, there is a need to experiment with how to practically create awareness about the issues outlined in the analysis and existing theory, especially while staying within the field itself and on the relevant platforms such as GitHub. While a rising number of initiatives are appearing in the field (Esparza, 2019; Fox, 2020/2021; Littauer et al., 2021; Open Source Design, n.d.; openusability.org, n.d.; *SustainOSS*, n.d.; teaserbot-labs, n.d.), it still stands to reason that what is realistically needed for FLOSS projects is more designer driven intervention and inquiry. In current studies there is still an apparent lack of initiatives that leave the realm of academia and position itself as a continued effort to understand and thereby improve the situation, as opposed to efforts limited to a finite timespan.

7.2.2. Addressing the 4 Themes

In designing the prototype the four themes uncovered in the analysis acted as a compass for making decisions for both what was possible as a single designer, but also for what were the most consistent and important findings throughout existing theory and in the findings of this study.

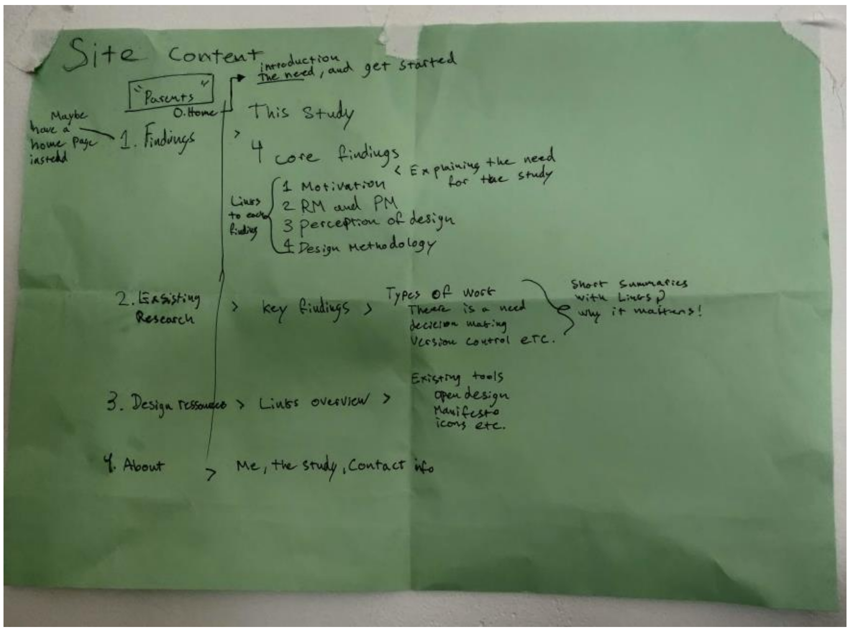


Figure 16 - An early sketch and brainstorming of the prototype content and layout

7.2.2.1. Motivation for Participation

The motivational factors uncovered in the analysis had many implications for the prototype. While the motivational factors discovered in the analysis refers to motivation for generally participating in FLOSS, the line between motivation for general FLOSS participation in the projects and participation in the use of this prototype are somewhat overlapping.

As such, ensuring that developers were familiar with the platform was important. As noted by one of the Jellyfin contributors ventures into more design centric platforms had been difficult in the past as the developers were simply not trained in that way of working, and learning a new platform naturally was difficult with their limited resources (Interview #7, Jellyfin). As a result, the option of using GitHub for this prototype would ensure that the core functionality, such as opening a pull request or filing an issue would be second nature for the participants. While system-specific usage was an important consideration for establishing the best possible participation from the developers, its ideological implications were equally important. The analysis showed that participation in these projects are also generally driven by ideological conviction to a varying degree. In order for this prototype to fit within this ideological frame of reference, it made sense to replicate the space in which these projects usually work which promotes openness, sharing and community driven efforts. As such, the project on GitHub was released with the GNU General Public License v3.0 (*Dani763f/FLOSS-UX*, n.d.) which is another tangible way to ensure the prototypes compliance with FLOSS ideals. Additionally, it was already established that the developers frequently used documentation of various kinds. Therefore, it was decided that the presentation of the findings was to mirror the type of documentation that these developers were already familiar with, and if possible, employ open-source technologies to do so. In this manner, every choice in this prototype would reflect a commitment to the ideology and hopefully increase the likelihood of recognition both in the projects, but also in the wider context of the field. In accordance with the idea of design interventions and the methodology of the study, it was important to reach a certain level of tangibility. Historically, as described by existing theory and in the analysis, pure design contributions have some complicated connotations in FLOSS, and it was therefore believed that presenting a functional prototype would be more likely to engage the developers. This also aligned with the notion of enculturation, namely in that enculturated UX design initiatives are likely to be more successful and seen less as external and are therefore less likely to be alienated (Iivari et al., 2014). In accordance with this idea of presenting this documentation piece in a way that was recognizable and familiar to developers it was decided to utilize the GitHub Pages functionality, which allows hosting of a website based on the content of the corresponding sites repository (*GitHub Pages*, n.d.). This was made possible by utilizing Jekyll, an open-source blog-aware static site generator, wherein a Jekyll theme for documentation with built-in search called ‘just-the-docs’ was used (*Jekyll • Simple, Blog-Aware, Static Sites*, n.d.; Marsceill, 2017/2021).

Secondly, with regards to ensuring motivation for actually reading and engaging with the content of the site, it was clear that a general limitation of the existing theory was that the academic format, namely that of long and often very detailed papers, did not fit with the developers expectations for how to easily obtain information. They argued that while academic ventures into the field were advantageous generally speaking, there is a need for “(...) distilling it down into a way that someone who is not an academic could easily understand” (Interview #7, Jellyfin). As a product of this consideration, the prototypes main content page, namely the “Key Findings Overview” was presented with summarizing images and bullet points to maintain searchability, and the general sense of overview that the FLOSS developers gravitate towards in their knowledge gathering efforts.

One apparent limitation of the platform with regards to motivation could be that end-users, or so-called passive users who are completely removed from the FLOSS context likely won't share their experiences on this platform. While GitHub discussion is more approachable for non-technical users as it requires no knowledge of some of the more code-centric features of the platform, it still warrants uncertainty surrounding the participation of users that are far removed from this context. One way to build on the prototype to gather experiences from passive-users could be based on designer driven approaches, more akin to traditional UX methodology wherein the goal is to discover, define, develop and deliver some product (Sharp et al., 2019). However, due to the uniqueness of each FLOSS project this effort might harmonize better with efforts rooted in the user experience of a given FLOSS project and would likely generate more usable data were some specific UX examples then introduced to the end-user. In other words, engaging with non-technical end-users that are currently passive is largely outside of the scope of this prototype.

7.2.2.2. A New Resource for Diversity, Terminology and Decision Making

Much as HIGs were found to be an external tool that helped in aligning projects toward actions with regards to for example the placement of buttons, margins, padding and other user-interface oriented decisions, this prototype is meant to serve a similar purpose but for more high level and open-ended concepts. While it is without a doubt simpler to convey rules for UI best-practices as they often are quite definitive in their recommendations, there is an apparent lack of platforms to meaningfully collect and create awareness about the higher level issues in the field. This focus on UI specific guidelines that was found in both this and previous studies, suggest that it might be bi-product of the system-centric mindset and by extension a contributing factor in proliferating the notion that design activities are only about ‘making things pretty’ as some developers were found to believe in previous research (Iivari, 2008). As mentioned above, conveying academic findings to FLOSS projects is no easy task just as passing on a user-centered mindset has proven difficult. This prototype is an attempt at filling that gap, both in the academic context where there are few initiatives that work towards enculturation on the platforms that are familiar to FLOSS participants, but also as an accessible tool that can aid these projects in understanding where and when they need to discuss their approach to design activities within their specific process.

An additional resulting benefit of this approach would be the potential for a partial mitigation of the barriers for designers entering into the field. While no designers were interviewed as part of the analysis, the findings in the literature suggests that one of the largest issues is that designers "(...) face obstacles to joining a FOSS project" (Terry et al., 2010, p. 1006). After joining a given project, designers were generally found to be quite appreciated, which seems to coincide with the notion of making UX concepts visible, and their benefits understandable to the developers. This idea of designer confusion, and fear of entering into the field has also been brought-forward by designers with experience in the field (SustainOSS, 2021), which confirms this as a detriment to the proliferation of design in FLOSS projects.

There continues to be an interesting duality in how the developers all found design and UX efforts to be of high importance. As mentioned, they did not have any collective terminology for their efforts and in general were very driven by their personal opinions and in the use of 'logic' for making decisions. While perception of a topic is somewhat elusive and difficult to change, the concept of enculturation and awareness again seems like the long-term solution to facilitate a change by inviting designers to join, thereby showing developers the value of having design contributions and initiatives. In summary, a hope of this prototype is that it might inspire more new designers to enter the field and be mindful of how they approach these projects, while normalizing and showing developers that these discussions and activities have value.

7.2.2.3. Methodology in FLOSS; Elevating Developer Efforts

One of the key findings with regards to methodology, was that graduated testing, and by extension that the 'design after design' approach was the most commonly used distinct method across the projects. While the developers had diverging opinions on the importance, placement and definition of design activities they did all consistently discount their own efforts, even though some of these activities seemingly do lead to better UX. With the goal of creating awareness, this is another central aspect of the prototype, namely, to promote the existing design methods that work within these projects. Much of the current research upholds the importance of recontextualization of existing HCI methodology into these projects, but not as many of these studies accredit the projects for having created a design methodology of their own. Both Jellyfin and Taskcafe mirror design choices made by proven products in the same space, and while it is still early for both projects in terms of lifespan, it is also early for their respective methodologies to properly manifest themselves. In other words, the graduated testing that was used in the both projects are similar to how designers think about iterations, and feedback. One of the goals of the prototype then becomes to accredit the method where it is needed, and to challenge the notion that graduated testing is inherently a system-centric methodology. Hopefully, this will aid in demystifying design as a concept and lead the projects to "(...) intentionally chang[ing] the world" (Nelson & Stolterman, 2012, p. 258) as opposed to absence of intention.

7.3. Testing the Prototype; 'Design after Design'

In testing the prototype, 4 of the interview participants were briefly interviewed again and were invited to create a discussion on the site and follow the process to discover potential areas of improvement and reiteration. The 4 remaining developers were informed of the process and invited to try out the process, without participating in an interview. In testing the prototype a few goals were especially important. As the prototype is holistic in nature and targets the field of FLOSS rather than one specific project, participation was a topic of interest for iterating on the prototype. In other words, the preliminary focus was on whether the process itself was understood by the developers. Another aspect of the testing was rooted in establishing whether or not the platforms chosen for the prototype would support the process and provide a foundation for collecting and creating the desired knowledge and awareness. Lastly, an inherent goal of this experimental effort was to verify whether or not this prototype would be a suitable method for gathering data about more projects in the future, and to subsequently present, discuss and foster enculturation as envisioned at the outset of the design process.

7.3.1. Overview of Responses

The following table contains an overview of the participants and public forums where the prototype was shared and potentially discussed, for convenience each platform in the below table has a link to the corresponding post on the given forum. Additionally, the table also covers the summarized reactions of the participants and whether they contributed to the prototype via knowledge or some other resource. The "Community sharing" section covers the places in which participants deemed the sharing of the prototype important, and links to the forums where it was reposted. GitHub is included here even though its primary function is to facilitate the process itself, as it has some measurements for gauging interest in projects, namely stars and watchers. The Reaction and contribution fields are left empty as all possible contributions converge on the GitHub platform.

Forum/Platform	**Participants**	**Reaction**	**Contribution**
Matrix/ Telegram/Discord	7 Jellyfin developers 1 Taskcafe developer	Positive and interested	No contributions
Reddit (r/opensource)	131 Upvotes 8 Commenters	Positive and supportive	2 discussions on Github
opensourcedesign.net	117 Views 6 Commenters	Positive Commenters engaged in sharing resources	Prototype was shared on other forums.
GitHub	22 Stars 1 Watcher		
Community Sharing			
Humane Tech Community			No contributions
Fediverse	15 Favorites 21 Reblogs		No contributions

Table 3 - Overview of the responses and outcome of the testing

7.3.2. Feedback and Testing; Synthesizing the Key Findings

7.3.2.1. Jellyfin and Taskcafe

The first distinct findings that emerged from testing the prototype was that the participation of the involved developers, namely the very same developers that participated in the initial interviews, was very low. While almost all of them displayed positivity towards the process and stated that it was aligned with their expectations all of them failed to actually participate and create a discussion as intended. This was the case both in the group that was interviewed about the feasibility of using the prototype, and in the group of developers simply invited to the process without further interaction. Some of the participants stopped responding, after being asked again to create a discussion for the prototype. While this is reminiscent of some of the gate-keeping tactics discovered in previous studies (Rajanen et al., 2015), it goes without saying that the process of testing this prototype would have no direct implications for the involved developers' projects and that the goals of this prototype was quite different from that which has previously been tried in the field. As such it is not certain why the developers did not follow the process, and one must assume that it was either due to a lack of time or resources, the lack of content to contribute or them not understanding the process itself. It could also be argued that the developers were too intimidated to feel comfortable following the process, which further supports the notion that more solutions in this space are needed to be designer driven and maintained. To further validate the feasibility, and reasoning for not participating in the envisioned process more projects must be involved to gauge exactly when and why the process fails to support further discussion.

7.3.2.2. Reddit; r/opensource

Interestingly the prototype did attract more attention in the larger context of FLOSS projects. The subreddit r/opensource was the most active of the forums on which the prototype was posted. 8 unique commenters engaged with the post and all generally shared the belief that FLOSS projects needed initiatives like this one. One of the commenters said that they “hope you can make an impact with this! As much as I absolutely love the philosophy of FLOSS and a lot of FLOSS projects, UX and UI is often not the main strength of applications in this category.”. Another commenter shared an interesting example of this lack of attention to UX, which happened to be an interesting showcase of just how important the specific characteristics of these communities can be in creating a good user experience (Tantacrul, 2019). In this specific example the open-source music application Musescore, had received feedback from the UX Youtube creator “Tentacrul” which resulted in him becoming the head of design of the project, and ultimately becoming the product owner (Potey, 2019; Tantacrul, 2021). This example was interesting and supported the idea that enculturation, through whatever means, was a requirement to reach an influential place that allows for decisions to be made. From this interaction on r/opensource one commenter provided his thoughts on the ideological section of the prototype.

Found the ideological discussion interesting. FLOSS was never intended to be anti-capitalist, it's free as in freedom, as in a free market of ideas. Entrepreneurship in a sense where people can try out new and different things from existing projects and see if there are benefits from doing that ... (PlayerDeus)

While this commenter was invited to copy this discussion into the GitHub repository for inclusion in the prototype, eventually it was moved manually and marked as a copy of a comment on r/opensource.

Lastly, a developer of an open-source application called BookStack (*BookStack*, n.d.) provided his findings after maintaining his respective project for 6 years. This developer followed the process and provided interesting perspectives on what can be considered part of the system-centric mindset. He argued that the sense of ownership is much greater in FLOSS projects, and argued that "(...) if you make changes to the design [of a project], you are making changes to their platform, their instance, their investment" (Brown, 2021). He also argued that this desire for options, referred to as "feature-creep" by the developers of Jellyfin (Interview #8, Jellyfin), is an effect that increases as a project grows and as the requests for new features grows. By extension it can be argued that this discussion supports the idea that the GitHub platform is somewhat complicit in creating this dynamic, and by extension the 'solution mindset'.

7.3.2.3. Feedback from Designers in the Field; opensourcedesign.net

One avenue that has yet to be explored both in the initial interviews and largely in the general field is the viewpoint of the designers already actively working in FLOSS projects, and by extension with these issues. As the open source design forum is a well-established community it was an obvious way to collect, explore and further bolster the existing findings for possible modifications to the prototype. The participants in this community, maybe unsurprisingly, were very supportive of the efforts of creating awareness about these issues. The primary contribution from the designers was that several resources were shared which all mirror many of the same findings that have been presented in this study. One such project was the open design project which is ‘a methodology for distributed, asynchronous design contributions to software projects’ (Fox, 2020/2021). In this vein it was also proposed that this prototype was added via pull request to the ‘delightful humane design’ repository (teaserbot-labs, n.d.) as the maintainer of that project argued for the importance of resources such as this prototype.

Lastly, the prototypes GitHub page was subsequently shared to several open-source outlets. It was posted in the humane tech community, which is a community that specialized in “(...) all things regarding the intersection between tech and humanity” (Humane Tech Community, n.d.). The project was also posted to Fediverse, which is a decentralized social networking site similar to twitter, yet with emphasis on privacy (*Mastodon & The Fediverse Explained*, n.d.).

8. Concluding Discussion

This section presents the interpretations and implications of the study uncovered throughout the sections of the study, namely in the analysis, theory and in the testing of the prototype. It seeks to cross examine these findings in order to both discuss the contributions of the study but also to inspire future work in the field. Among the most important findings of this study is that design interventions by themselves appear to be able to facilitate change, and that it is a suited approach for externalizing an idea, and in doing so making it 'visible' to the involved parties. This paper by extension argues for more continuous designer driven interventions, and the need for design activities in FLOSS to shift from "(...) deciding on and communicating an interpretation to supporting and intervening in the processes of designer, system, user, and community meaning-making" (Sengers and Gaver, 2006, p. 102 in Redström, 2008, p. 412).

8.1. Intervention Creates Action

One of, if not the most influential findings from this study certainly is that design interventions and inquiries by themselves seem to be able to orient projects towards action and inspire discussion. Several findings point to this fact, and while some part of this may also depend on the timely introduction of the GitHub discussions functionality, its emergence and functionality seems in line with what is required to further support these projects in diversifying their understanding of their own design choices.

At the time when the initial empirical findings were being researched, GitHub discussions was a limited feature that was possible to enable via invite from GitHub. Only a few of the developers were aware of the feature and it was not available either in Jellyfin or Taskcafe. While it cannot be definitively stated with the given sample size, there seems to be a correlation between the initiation of this study and that both projects enabled GitHub discussions shortly after the interviews, and has subsequently moved UX initiatives from the issues tab to the discussions tab (*UI & UX References · Discussion #3 · Jellyfin/Jellyfin-Meta*, n.d.). It was clear in the interviews and conversations with the developers that internal discussions were being had about the subject matter of the study, and that this sharing of experiences with the interview did have an impact both on the amount of participation in the initial empirical gathering but also in the feedback of the prototype. For example, after the first interview with a developer working on Jellyfin-Vue he had shared that the topic of the study was interesting in the Jellyfin internal chat, which resulted in five of the other developers reaching out and confirming their interest in participating in the study. A similar situation occurred in the testing of the prototype where all the involved developers were aware of the prototype after the first participant shared the GitHub repository to his peers, in a similar manner. In other words, even though the project was approached with no tangible method or platform simply intervening in the project with curiosity was enough to affect the design considerations of the projects.

This seems to suggest that a possible misconception in the field, especially in the early theory, is that specialized usability methodology is required to foster change. If such methodology was implemented, it might interfere with the adhocratic nature of these projects and as argued by the developers would largely depend on the size of the project as "(...) the bigger a project gets eventually you do need guidelines or a better focus" (Interview #8, Jellyfin). This notion seems to correlate with how the largest FLOSS projects all seem to have a stronger sense of their design language, and even employ traditional methods that fall under the umbrella of UX such as user testing, upholding accessibility requirements, using detailed mock-ups and even creating descriptive human interface guidelines of their own (*Firefox UX*, n.d.; *GNOME Human Interface Guidelines*, n.d.; *GNOME Shell & Mutter*, n.d.; *KDE Human Interface Guidelines — Human Interface Guidelines Documentation*, n.d.).

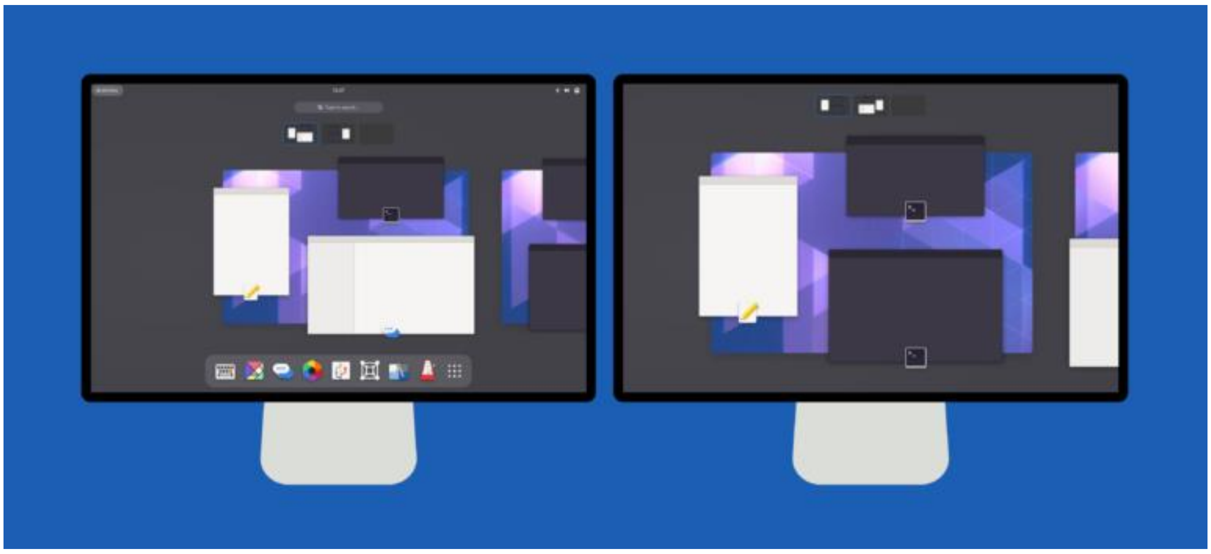
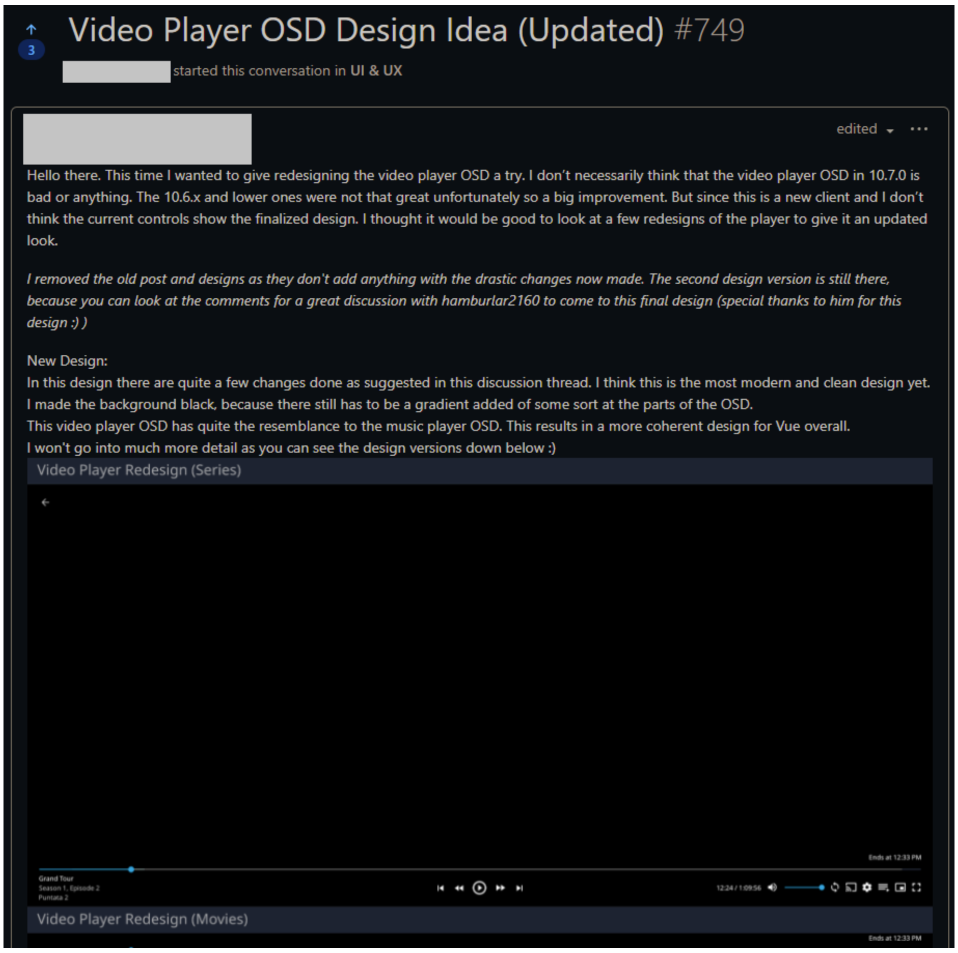


Figure 17 - Example of the GNOME project using mock-ups and user research consistently

In relation to the findings of this study, it seems that contemporary research on the matter will benefit from further investigating practical methods that keep in mind that for many outcomes, roughly 80% of consequences come from 20% of the causes (Bhowmick, 2019). While it is not definitive if such a principle will translate into these projects, it raises questions and possible options for discovering what minimum effort will result in the most measurable changes in projects such as these. The feasibility of the prototype in this study, is without a doubt tied to this idea through it being recommended as the most obvious and therefore simple way to improve UX in previous studies (Terry et al., 2010). In striving for awareness however, this prototype is not the only way. As shown earlier other equally experimental ways of doing this has succeeded in this space (Potey, 2019; Tantacrul, 2019) and more ways in which to create awareness about these discussions are still emerging (Esparza, 2019; Littauer et al., 2021; teaserbot-labs, n.d.). In summary, it would seem that the adhocracy of these projects could allow future research to experiment even further with the format and the platforms used to create awareness and collect knowledge about these issues.

8.1.1. Rise in Design Contributions

Another interesting discovery that emerged after GitHub discussion was enabled, is the apparent rise in design contributions in both projects. At the outset of the study, many of the participants argued that they did not receive many pure design contributions such as mock-ups or wireframes as part of their process. However, after the GitHub discussions feature was enabled it would appear that these types of contributions now have appeared in a structured manner, both in Jellyfin and in Taskcafe just as it was hypothesized that they would early on in the prototyping and in some form in the theory (Nichols & Twidale, 2006). Additionally, it appears that the developers themselves have started using the functionality to gauge, collect and expand on ideas.



3

Video Player OSD Design Idea (Updated) #749

[redacted] started this conversation in **UI & UX**

[redacted]

edited

Hello there. This time I wanted to give redesigning the video player OSD a try. I don't necessarily think that the video player OSD in 10.7.0 is bad or anything. The 10.6.x and lower ones were not that great unfortunately so a big improvement. But since this is a new client and I don't think the current controls show the finalized design. I thought it would be good to look at a few redesigns of the player to give it an updated look.

I removed the old post and designs as they don't add anything with the drastic changes now made. The second design version is still there, because you can look at the comments for a great discussion with hamburlar2160 to come to this final design (special thanks to him for this design :))

New Design:
In this design there are quite a few changes done as suggested in this discussion thread. I think this is the most modern and clean design yet.
I made the background black, because there still has to be a gradient added of some sort at the parts of the OSD.
This video player OSD has quite the resemblance to the music player OSD. This results in a more coherent design for Vue overall.
I won't go into much more detail as you can see the design versions down below :)

Video Player Redesign (Series)

Video Player Redesign (Movies)

Figure 18 - Example of design contributions after GitHub discussions have been enabled in Jellyfin-Vue

While it is impossible to draw definitive conclusions from this apparent increase in design contributions, it does seem to support the notion that creating awareness and utilizing a platform that makes it easier to make design contributions, will improve the overall quality and support multiple interpretation and possibilities for the given software. While it is unclear whether the amount of these types of discussions has risen compared to before GitHub discussions, they are without a doubt easier to both reference, and find for newcomers and veterans of the projects alike. It must also be said that the scope of these two projects are vastly different. The number of participants, their potential end-users and the project's history all differ greatly which further makes it difficult to conclude on whether this increase is in fact causation rather than simply correlation.

8.1.2. Lack of Participation; Discussing the Prototype

Several findings point to the fact that for FLOSS projects to truly improve their UX processes more designers need to take action and join these projects. Not only do more designers or researchers need to engage with these communities, they simultaneously need to understand the specifics of the project they are joining, but also the ideological beliefs and the platforms that are being used to successfully change the status quo. In the case of the FLOSS-UX prototype, all these requirements were accounted for in the initial stages of the design. However, as evident from the low participation of the developers, designing for "(...) 'users' during design seem to assume that there already are users of things not yet designed, thus obscuring the complexity of what actually happens as someone starts using a thing, as someone becomes a user" (Redström, 2008, p. 410). As alluded to in the design section it was not immediately clear why participation was so low, while the general reception of the prototype in the surrounding context was very positive.

The first possibility that comes to mind in discussing this lack of participation is that the holistic nature of the prototype might have been a hindering factor in its use, specifically for Jellyfin and Taskcafe. Early on in the design process, it was clear that in order for the process to work as intended it would be needed to engage with several communities to recognize their unique characteristics. As such, with the goal of cross-community awareness, it was clear that such an infrastructure had to be more generalized than specialized. This corresponded with the fact that developers were already using more UI specific documentation, that while useful, rarely talks about the 'why' and the reasoning behind its principles. It can be imagined that this generalization and abstract nature of the presented findings might have obscured the direct relevance to the projects, making the topics for discussion more difficult to relate to for these developers that are not normally examining their own practices and concepts for creating stronger UX. This raises questions about how such a discussion platform might support both generalization and specialization, while still supporting more cross-community awareness about the FLOSS specific concepts that both support and withhold developers in creating better UX.

Additionally, there is a possibility that the developers in these projects simply had already shared all that they could. A different approach in the testing of the prototype might have been to involve completely different projects to verify the validity of the findings and bolster them with perspectives from other projects with different characteristics. One finding in support of this is that most of the interest and participation came from the more generalized field of FLOSS, namely the developers and enthusiasts on sites such as Reddit. This might also be related to them seeing the project for the first time, thereby having an increased interest in its working compared to the developers that had been involved over a longer period. One thing of note related to this, is that both Jellyfin and Taskcafe are very busy projects and are both resource depleted. Additionally, both projects are still in the early stages of their life which means that this lack of participation might also coincide with the increased work that is required when releasing a new version, which both projects did around the time the feedback. Especially in the case of Jellyfin, this means that once the tech-debt of Emby is addressed in full, and the project reaches a "(...) point where [they] aren't making huge architectural changes" (Interview #4, Jellyfin) design activities such as this process might attract more available resources.

Another possibility is that some indirect gate-keeping tactics are at play as were also found to be the case in other studies (Rajanen et al., 2015). While all study participants have been willing to participate all the way through this project, this lack of participation is somewhat reminiscent of previous studies wherein design "(...) activities and their results [quickly were] buried beneath other discussions and news" (Rajanen et al., 2015, p. 11). The implication of this is that there is a need for design activities to be continuously shared, updated, pushed and discussed in the project's development cycle. Doing this might be difficult, especially when it is done as a single designer driving these discussions simply due to the amount of required work and the fast pace of the development cycle. By this logic it is clear how designer enculturation is difficult, and currently rare. The designer has to be able to understand the subtle social values and convictions found in FLOSS, the usage of the platforms and cope with the high speed at which changes are made in these projects. In summary, designers clearly have three main priorities in FLOSS projects if they want to increase their chances of success.

1. *Understanding the product under development*
2. *Finding motivation for [design activities] and*
3. *locating and targeting the decision-makers.*

(Iivari et al., 2014, p. 1)

It is difficult to conclude whether or not the prototype in this study reached a decision-making arena, even though it did incorporate all of the above points. Furthermore, its perceived success to generate momentum in the general FLOSS context suggests that the continued use of the prototype will reveal more about what needs to be considered in order to normalize design considerations in FLOSS. This success might also be affected by the fact that this author has spent many years learning and exploring the field of FLOSS and understands its subtle dynamics to a relatively advanced degree. This might not be the case for most designers, which further cements the need for the continued gathering of data from various projects.

8.2. Adhocratic Design in FLOSS; Disregarded Initiatives?

In the study it was clear from the onset of the study that both the theory and the developers agreed that very little emphasis was put on methodically planning design activities in FLOSS projects. However, it was also clear from existing research that while the adhocratic nature of these projects might obfuscate what is being done, even on a micro-scale for instance between repositories, some common approaches were discovered that were a net benefit for UX in the projects. Furthermore, it was found that these beneficial initiatives that mirror simplified methods and concepts from the field of design often are discounted by their organizers (Terry et al., 2010).

The first of these approaches to design was, as mentioned in the analysis, based in the stockpiling of inspiration from existing products that serve a similar purpose. In the case of Jellyfin this was platforms such as Netflix, Amazon prime and Disney+ that without a doubt has the financial capacity and interest for investing in UX on a big scale. Similarly, Taskcafe drew inspiration on solutions such as Trello (*What Is Trello? - Trello Help*, n.d.), to establish what the required base functionality was. Can FLOSS projects then simply copy the aesthetics, proven models of interaction and in essence the user experience of these projects and juxtapose their essence with the FLOSS philosophy? Interpreting the findings from these two projects, there seems to be a correlation between the success of this method and in the project's ability

to foster discussions about multiple interpretations of these ‘best-practices’. Jellyfin can certainly draw on established functionality and interaction principles, and in fact this way of working is not totally alien to contemporary UX work (Apple, 2014; Carlson, n.d.). In the case of Jellyfin, they arguably support this need for supporting multiple interpretations of design through having a large quantity of different sources for inspiration, and even state that “(...) the goal is never to make a 1:1 copy of existing services, but look at what is done on these services around the world and use these references to inform decisions” (*UI & UX References · Discussion #3 · Jellyfin/Jellyfin-Meta*, n.d.). This initiative is quite well organized and appears to have positive results for the continued improvement of the UX of the project, especially given how these projects simultaneously need to balance the ability for these developers only working in their free time and with no obligation to complete a fixed amount of work. In addition to this, some developers even explained how they occasionally ask for feedback from friends, family and colleagues in order to understand where to improve UX aspects next. Collectively, it can be argued that these projects then mirror UX and design principles more than what the stereotype about FLOSS might suggest.
This again reaffirms that there are grounds to believe that an important part of this act of striving for awareness is not only to mitigate the effects of the so-called ‘solution mindset’ that the developers employ most of the time, but also to spread awareness about the initiatives that are working as intended such as this ‘fake it till you make it’ approach between FLOSS communities. Somewhat reinforcing this interpretation is the fact that very little actual comparisons exist that pit proprietary solutions against FLOSS alternatives in a structured and analytical manner (Nichols & Twidale, 2006). This makes the claim that the characteristics of the project might be as large a component as its methodology seem more plausible even accounting for the consensus in current theory.

In summary, while Jellyfin and Taskcafe might not be the most extreme cases of poor UX decisions in the field of FLOSS, they are representative of modern and heavily developing projects that are seeing a lot of work and progress. Both the projects have, with the above in mind, achieved relatively impressive results from just using a ‘design after design’ approach wherein numerous iterations over time propel the projects closer to having a coherent user experience. This might also simply be related to the ever-changing expectation for what constitutes good UX, and while there most definitely are different levels of abstraction for these expectations, the general increase in the popularity of UX is seemingly bleeding into these community driven projects.

8.3. Limitations and Threats to Validity

Some limitations and considerations of validity can also be discussed in this closing part of the paper. The first limitation of the paper is time, especially in the case of the prototype. Part of adopting the ‘design after design’ approach to the prototype makes it susceptible to drastic changes over time. This is the case in many FLOSS projects where looking at the initial starting point or first releases of a piece of software is quite significantly different from the current iteration. In this regard FLOSS is not too different from traditional proprietary software, but it can be hypothesized that the adhocratic nature of the projects and their system-centric mindset might result in more drastic and fast paced changes as a result of, for instance, changing to a new codebase. This obviously has implications for the prototype as it also relies on that very same concept, namely that time and active use will shape it accordingly. As such it is very possible that the timespan of this study is not sufficient in realising the full potential of the prototype, at least as described in this paper. This is a somewhat double-edged sword as the prototype is inherently meant to be a continuous effort, as a response to the lack of such initiatives in the field of HCI. In summary, the reader of this study might find it relevant to draw comparisons between the prototype as described in this paper, and the iteration of the prototype that is accessible on GitHub as this version represents a continuous effort to collect knowledge about FLOSS practices and create awareness about their implications for better UX.

Similarly, to the other studies in this field, the sample size of this paper does not necessarily reflect the holistic truth as characteristics of these projects might diverge more than what is represented in this study. This is increasingly true moving towards the findings provided by the prototype in this design intervention, as this is the most experimental and open-ended part of the study. The 4 categories of concepts as outlined in the analysis do however very likely apply to the majority of FLOSS projects in some way shape or form, as they act as both verification and as an addition to the existing studies in the field. As such, any new research into FLOSS projects should be able to build at least their initial efforts on these principles given that the roles, platforms and overall similarities of the studies projects are somewhat similar.

This author, as alluded to earlier in this section, has since a young age been interested in these communities and has spent a great deal of time learning and using software produced by these socio-technical systems. As such, a limitation of the study might be that some of these ingrained predisposed experiences have shaped this paper in some way shape or form. Great care has been taken to factually and theoretically confirm any and all abductive reasoning that has taken place throughout the study, however as it often happens "aiming to be comprehensive, such approaches in fact often lead to problematic oversimplifications" (Nelson & Stolterman, 2012, p. 144). While this might be common knowledge in academic circles, this author makes no claim that the findings presented here represent the full truth or the perfect solution for creating knowledge and awareness about these practices. In summary, the cross section where FLOSS and UX design practices meet and intermingle is without a doubt an example of a problem area where a finite study such as this might simply be unable "(...) to deal with the full richness and complexity of reality" (Nelson & Stolterman, 2012, p. 144).

8.4. Future of the Field; A Summary

While it is unlikely that anything can be concluded unquestionably from these findings, as the field itself is a moving target, some new key findings and possibilities have emerged for the future research in this field.

Firstly, it was proposed that a platform to support design activities would very likely appear "(...) just as one did for functionality bugs" (Nichols & Twidale, 2006, p. 156). Now, 15 years later, either the natural emergence of such a platform simply will not happen, or alternatively, that it takes longer to emerge than first imagined. New initiatives such as GitHub Discussions, while it is still too early to tell, show great promise to partly support the types of discussions and the way of working that is required and expected in design work. Additionally, more initiatives that promote the need for awareness on UX-centric discussions exist than ever before, and their form is increasingly becoming more and more experimental which suggests that a multitude of undiscovered ways in which to support this awareness through enculturation exists, especially ones that depart from the traditional academic format.

There are reasons to believe that while FLOSS projects are adhocratic and generally unstructured when it comes to their design and UX methodology that the methods that have emerged in these socio-technical systems have some significance and are, depending on the characteristics of the project, capable of producing surprisingly strong user experiences. It stands to reason that this might simply be that expectations for UX are changing generally, and that FLOSS projects then follow. However, there are still FLOSS projects that suffer from famously poor UX choices, and while it at this point is conjecture, there might be some correlation between the age, community, or some other characteristic in making these poor decisions. In other words, the continued collection of data from such projects are key to form an understanding of these correlations and their causations.
With this in mind, it would seem that one future interest of studies is to gauge the effectiveness of methods, and establish an understanding of how to leverage the most obvious ways in which to create awareness, as supported by the fact that the inquiry of this study and the apparent benefit of its corresponding prototype seemingly are inseparable in terms of what was most effective in supporting change.

Related to this, there is also room for further bolstering the findings of this study with more attention to facilitating demonstrations of daily tasks in FLOSS projects as briefly mentioned in the analysis of non-interview data sources. This focus of contextual inquiry is difficult in FLOSS projects due to their online and distributed nature. However, their openness and the ease of access to platforms on which they work suggests that, while it was outside the scope of this study, it might be a beneficial endeavour in order to examine the correlation between what is said, and what is done in these projects.

With the above in mind it is clear that in order to understand which '80% of consequences that result from 20% of the causes', more time is required to reiterate on the prototype of this study. As mentioned previously, the prototype will continue to be used to collect, present, discuss and hopefully establish a cycle of enculturation into the future. Accordingly, having read this paper the reader should visit the FLOSS-UX prototype and access the design rationale of the project. In summary, to reflect on the question posed at the outset of this report it appears one possible way to engage and build upon what is known about the user experience practices of FLOSS projects, is firstly to adhere to the 4 uncovered findings of this study. Secondly, researchers will benefit from utilizing frameworks, such as the 5 properties of online communities as proposed by Nancy K. Baym to discover and define which characteristics that define the FLOSS project that is subject to study. Lastly, there is ample room in the field for experimental efforts that set out to discover new ways in which to drive awareness, community meaning-making and [to] approach this shift in the designers role towards "(...) supporting multiple interpretations as a central goal" (Sengers and Gaver, 2006, p. 102 in Redström, 2008, p. 412).

9. References

About Discord | Our Mission and Values. (n.d.). Discord. Retrieved 6 April 2021, from https://discord.com/company

About team discussions—GitHub Docs. (n.d.). Retrieved 14 November 2020, from https://docs.github.com/en/free-pro-team@latest/github/building-a-strong-community/about-team-discussions

Andersen, T., Halse, J., Moll, J., Dk, D., Dk, D., & Dk, J. (2011). *DESIGN INTERVENTIONS AS MULTIPLE BECOMINGS OF HEALTHCARE*. 10.

Andreasen, M., Nielsen, H. V., Schroder, S., & Stage, J. (2006). *Usability in open source software development: Opinions and practice*. https://doi.org/10.5755/J01.ITC.35.3.11776

Apple. (n.d.). *Human Interface Guidelines—Design—Apple Developer*. Retrieved 24 February 2021, from https://developer.apple.com/design/human-interface-guidelines/

Apple. (2014). *Prototyping—Fake It Till You Make It*. https://www.youtube.com/watch?v=3lqh-A5Jy4Q

Bach, P. M., Kirschner, B., & Carroll, J. M. (2007). Usability and free/libre/open source software SIG: HCI expertise and design rationale. *CHI '07 Extended Abstracts on Human Factors in Computing Systems*, 2097–2100. https://doi.org/10.1145/1240866.1240957

Baym, N. K. (2015). *Personal connections in the digital age* (Second edition). Polity Press.

Benson, C., Müller-Prove, M., & Mzourek, J. (2004). *Professional Usability in Open Source Projects: GNOME, OpenOffice.org, NetBeans*. 2.

Beta source code missing · Issue #3479 · MediaBrowser/Emby. (n.d.). GitHub. Retrieved 5

April 2021, from https://web.archive.org/web/20181212104719/https://github.com/MediaBrowser/Emby/issues/3479

Bhowmick, A. (2019, September 30). *The 80/20 Rule in User Experience*. Medium. https://medium.com/design-ibm/the-80-20-rule-in-user-experience-1695de32aaae

Blog—Jellyfin: The Free Software Media System. (n.d.). Retrieved 6 November 2020, from https://jellyfin.org/posts/

BookStack. (n.d.). BookStack. Retrieved 2 April 2021, from https://www.bookstackapp.com/

Bootstrap. (n.d.). Retrieved 24 February 2021, from https://getbootstrap.com/

Brown, D. (2021, February 28). *Thoughts on design after developing/maintaining BookStack · Discussion #10 · dani763f/FLOSS-UX*. https://github.com/dani763f/FLOSS-UX/discussions/10

Bugzilla.org. (n.d.). Retrieved 24 February 2021, from https://www.bugzilla.org/

Carlson, B. (n.d.). *Copying an Existing UI to Learn How It Was Designed | Wireframing Academy | Balsamiq*. Retrieved 5 April 2021, from https://balsamiq.com/learn/articles/copying-to-learn/

Cheng, J., & Guo, J. L. C. (2018). How Do the Open Source Communities Address Usability and UX Issues? An Exploratory Study. *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*, 1–6. https://doi.org/10.1145/3170427.3188467

Clients—Jellyfin: The Free Software Media System. (n.d.). Retrieved 6 November 2020, from https://jellyfin.org/clients/

Coleman, G. (2010, September 21). *The Anthropology of Hackers*. The Atlantic. https://www.theatlantic.com/technology/archive/2010/09/the-anthropology-of-hackers/63308/

Crewel__Lye. (2020, December 16). *Instead of the normal requests or complaining, I would like to take a second to thank all of the jellyfin contributors and developers for all they do.* [Reddit Post]. R/Jellyfin. www.reddit.com/r/jellyfin/comments/keg00z/instead_of_the_normal_requests_or_complaining_i/

Dani763f/FLOSS-UX. (n.d.). GitHub. Retrieved 30 March 2021, from https://github.com/dani763f/FLOSS-UX

Design & UX. (n.d.). A Space for Open Source Sustainers. Retrieved 1 March 2021, from https://sustainoss.org/working-groups/design-and-ux/

Despalatovic, L. (2013). *The Usability of Free/Libre/Open Source Projects*.

Esparza, G. (2019, July 15). *Open Collective Design*. Medium. https://medium.com/open-collective/open-collective-design-ec6bd42da79

Firefox UX. (n.d.). Firefox UX. Retrieved 4 April 2021, from https://blog.mozilla.org/ux

Fisher, C., & Payne, W. (2021, March 30). *No PRs Please*. LINUX Unplugged. https://linuxunplugged.com/399

Fox, E. (2021). *Erioldoesdesign/opendesign*. https://github.com/Erioldoesdesign/opendesign (Original work published 2020)

Frishberg, N., Dirks, A. M., Benson, C., Nickell, S., & Smith, S. (2002). Getting to know you: Open source development meets usability. *CHI '02 Extended Abstracts on Human Factors in Computing Systems*, 932–933. https://doi.org/10.1145/506443.506666

Github 'About'. (n.d.). GitHub. Retrieved 6 November 2020, from https://github.com

GitHub features: The right tools for the job. (n.d.). GitHub. Retrieved 6 November 2020, from https://github.com/features

GitHub Pages. (n.d.). GitHub Pages. Retrieved 4 April 2021, from https://pages.github.com/

GNOME Human Interface Guidelines. (n.d.). Retrieved 24 February 2021, from https://developer.gnome.org/hig/stable/

GNOME Shell & Mutter. (n.d.). Retrieved 4 April 2021, from https://blogs.gnome.org/shell-dev/

Google. (n.d.). *Material Design*. Material Design. Retrieved 24 February 2021, from https://material.io/design

Halse, J., & Boffi, L. (2014). *Design Interventions as a Form of Inquiry*. The Design Anthropological Futures Conference. https://adk.elsevierpure.com/da/publications/design-interventions-as-a-form-of-inquiry

Home | Documentation—Jellyfin Project. (n.d.). Retrieved 6 November 2020, from https://jellyfin.org/docs/

Humane Tech Community. (n.d.). *About Humane Tech Community*. Humane Tech Community. Retrieved 2 April 2021, from https://community.humanetech.com/about

Iivari, N. (2008). Usability in Open Source Software Development: An Interpretive Case Study. *ECIS*.

Iivari, N. (2010). Usability Innovations in OSS Development—Examining User Innovations in an OSS Usability Discussion Forum. *OSS*. https://doi.org/10.1007/978-3-642-13244-5_10

Iivari, N., Rajanen, M., & Hedberg, H. (2014). *Encouraging for Enculturation – An Enquiry on the Effort of Usability Specialists Entering OSS Projects*. https://doi.org/10.13140/2.1.2714.8483

IssueHunt—A bounty platform for open source projects. (n.d.). Retrieved 6 April 2021, from https://issuehunt.io/

Jekyll • Simple, blog-aware, static sites. (n.d.). Jekyll • Simple, Blog-Aware, Static Sites.

Retrieved 4 April 2021, from https://jekyllrb.com/

Jellyfin 'People'. (n.d.). GitHub. Retrieved 6 November 2020, from https://github.com/orgs/jellyfin/people

jellyfin-vue—"Lazy loading #272". (n.d.). GitHub. Retrieved 19 November 2020, from https://github.com/jellyfin/jellyfin-vue

KDE Human Interface Guidelines—Human Interface Guidelines documentation. (n.d.). Retrieved 24 February 2021, from https://hig.kde.org/

Lazar, J. (2017). *Research methods in human computer interaction* (2nd edition). Elsevier.

Lichtman, M. (2013). *Qualitative research in education: A user's guide* (3rd ed). SAGE Publications.

Littauer, R., Fox, E., Flory, J. W., Ojemeh, P., Esparza, M., & Bullen, G. (2021). *Sustain Open Source Design Podcast*. Sustain Open Source Design. http://sosdesign.sustainoss.org/

Llerena, L., Rodríguez, N., Castro, J. W., & Acuña, S. (2018). How to Incorporate a Usability Technique in the Open Source Software Development Process. *SEKE*. https://doi.org/10.18293/seke2018-006

Llerena, L., Rodríguez, N., Llerena, M., Castro, J. W., & Acuña, S. (2018). *Applying a Usability Technique in the Open Source Software Development Process: Experiences from the Trenches*. https://doi.org/10.5772/INTECHOPEN.74862

Löwgren, J., & Stolterman, E. (2004). *Thoughtful Interaction Design: A Design Perspective on Information Technology*. The MIT Press. https://doi.org/10.7551/mitpress/6814.001.0001

Marsceill, P. (2021). *Pmarsceill/just-the-docs* [SCSS]. https://github.com/pmarsceill/just-the-docs (Original work published 2017)

Masson, A. L., Lalanne, D., & Amstutz, T. (2017). A Usability Refactoring Process for

Large-Scale Open Source Projects: The ILIAS Case Study. *CHI Extended Abstracts*. https://doi.org/10.1145/3027063.3053345

Mastodon & The Fediverse Explained. (n.d.). Savjee.Be. Retrieved 2 April 2021, from https://www.savjee.be/videos/simply-explained/mastodon-and-fediverse-explained/

Material-UI: A popular React UI framework. (n.d.). Retrieved 24 February 2021, from https://material-ui.com/

Matrix.org. (n.d.). Matrix.Org. Retrieved 6 November 2020, from https://matrix.org

Nelson, H. G., & Stolterman, E. (2012). *The design way: Intentional change in an unpredictable world* (Second edition). The MIT Press.

Nichols, D., & Twidale, M. (2006). Usability processes in open source projects. *Softw. Process. Improv. Pract.* https://doi.org/10.1002/SPIP.256

Norman, D. (1981). The trouble with UNIX: The user interface is horrid. *Datamation*, *27*, 139–150.

Open Source Design. (n.d.). *Opensourcedesign.net*. Retrieved 2 March 2021, from https://opensourcedesign.net/

openusability.org. (n.d.). *Open Usability*. Retrieved 2 September 2020, from http://openusability.org/

Potey. (2019, December 8). *Tantacrul, creator of "Music Software &amp; Interface Design: MuseScore" on YouTube, is announced as the new Head of Design for MuseScore* [Reddit Post]. R/Composer. www.reddit.com/r/composer/comments/e7xpjn/tantacrul_creator_of_music_software_interface/

Rajanen, M., & Iivari, N. (2015a). Examining Usability Work and Culture in OSS. *OSS*. https://doi.org/10.1007/978-3-319-17837-0_6

Rajanen, M., & Iivari, N. (2015b). Power, Empowerment and Open Source Usability. *CHI*.

https://doi.org/10.1145/2702123.2702441

Rajanen, M., Iivari, N., & Keskitalo, E. (2012). Introducing usability activities into open source software development projects: A participative approach. *NordiCHI*. https://doi.org/10.1145/2399016.2399120

Rajanen, M., Iivari, N., & Lanamäki, A. (2015). Non-response, Social Exclusion, and False Acceptance: Gatekeeping Tactics and Usability Work in Free-Libre Open Source Software Development. *INTERACT*. https://doi.org/10.1007/978-3-319-22698-9_2

Redström, J. (2008). RE:Definitions of use. *Design Studies*, *29*(4), 410–423. https://doi.org/10.1016/j.destud.2008.05.001

Sharp, H., Preece, J., & Rogers, Y. (2019). *Interaction Design—Beyond Human Computer Interaction* (5th ed.). John Wiley & Sons, Inc.

Stallmann, R. (2016, November 18). *FLOSS and FOSS*. https://www.gnu.org/philosophy/floss-and-foss.html

SustainOSS. (n.d.). A Space for Open Source Sustainers. Retrieved 1 March 2021, from https://sustainoss.org/

SustainOSS. (2021, March 22). *Eriol Fox on Open Source Design and Sustain*. Sustain. https://podcast.sustainoss.org/72

Tantacrul. (2019, March 19). *Music Software & Interface Design: MuseScore*. https://www.youtube.com/watch?v=4hZxo96x48A

Tantacrul. (2021, January 15). *How I Designed a Free Music Font for 5 Million Musicians (MuseScore 3.6)*. https://www.youtube.com/watch?v=XGo4PJd1lng&t=79s

Taskcafe Github. (n.d.). [TypeScript]. Retrieved 6 November 2020, from https://github.com/JordanKnott/taskcafe (Original work published 2020)

teaserbot-labs. (n.d.). *Delightful-humane-design*. Codeberg.Org. Retrieved 27 March 2021, from https://codeberg.org/teaserbot-labs/delightful-humane-design

Terry, M. A., Kay, M., & Lafreniere, B. J. (2010). Perceptions and practices of usability in the free/open source software (FoSS) community. *CHI*. https://doi.org/10.1145/1753326.1753476

thenightwolf51. (2020, December 26). *Taskcafe 0.3.1—An open source project management tool* [Reddit Post]. R/Opensource. www.reddit.com/r/opensource/comments/kke9y5/taskcafe_031_an_open_source_project_management/

UI & UX References · Discussion #3 · jellyfin/jellyfin-meta. (n.d.). GitHub. Retrieved 5 April 2021, from https://github.com/jellyfin/jellyfin-meta/discussions/3

UI & UX References—Jellyfin-web. (n.d.). GitHub. Retrieved 19 November 2020, from https://github.com/jellyfin/jellyfin-web

Vuetify—A Material Design Framework for Vue.js. (n.d.). Vuetify. Retrieved 6 October 2020, from https://vuetifyjs.com/

Wang, W., Cheng, J., & Guo, J. L. C. (2020). How Do Open Source Software Contributors Perceive and Address Usability? Valued Factors, Practices, and Challenges. *ArXiv:2007.06654 [Cs]*. http://arxiv.org/abs/2007.06654

What is Trello? - Trello Help. (n.d.). Retrieved 6 April 2021, from https://help.trello.com/article/708-what-is-trello

Wobbrock, J. O., & Kientz, J. A. (2016). *Research Contributions in Human- Computer Interaction*. 7.

10. Appendix

10.1. Interview Guide

What is your day job?

What does the participant do for a living?

What FOSS project are you associated with?

When and how did you get involved with FOSS in general, and the project?

How? What was your primary motivation for joining X FOSS project?

What do you do in the project?

What have you done historically?

What is your motivation for developing/Contributing in a FOSS project?

Social rewards? Developing personal skill? Scratching an itch?

Are user numbers important? Is a large user base the goal of the project?

Are you ideologically driven?

What is your take on feedback? Is it mostly negative or positive, or maybe not detailed enough (frustrating to understand/create value from?)

What are your general thoughts on how it shapes the project?

Are there many passive users? Do you feel that amount of external reporting/feedback is enough?

10.1.1. Usability/UX/Design perception

What is your perception of usability?

What terms do you use in the project? UX/Usability? Is there a difference to you? What do they entail?

Is it a trivial task? Is it very important?

Do you feel that subjectivity is a factor in usability decisions?

Do you have any UX/Usability guidelines in your project?

What tools are being used? HIGs?

Do you employ systematic testing? Do you have a definition of usability you all follow?

Do you keep track of design rationale?

Do you have any people contributing solely on UX/Usability?

Are there any resources allocated only for UX? Are there any UX professionals in the project?

Do you feel the project could use UX pro's? Or is it doing fine without them?

Have you ever been approached by any UX pro's? If so what happened?

Do you employ any methods in the project between contributors/end users with a focus on usability?

for instance a blog, or A/B testing?

How do you approach user needs? You get a feature request, what happens? How do you map what the user wants/is trying to do

Do you feel that [bug tracking platform here] is sufficient for all types of issues?

Are usability issues especially hard to keep track off? Complexity?

Are usability issues the same as bugs in the code to you?

How do you manage usability bugs that might require large code rewrites?

10.1.2. Social aspects and resource management of the project

Do you ever take your work offline?

How are decisions made? Is consensus required?

Is the project totally democratic or are there certain people with more pull than others?

Do you find that social bonds are important in your work in the project?

Social bonds between contributors or between devs

Do you have certain users that are very active in testing and providing feedback with the most recent code?

How much change in the project would you say is driven or influenced by completely external users?

What do you assign the greatest value in the project? What do you assign the most resources for?

Figuring out how different projects manage this differently

Do you spend time educating other project members? Can it be UX, code or otherwise?

How do you manage resource allocation? Is it first come first served or do you have roles. If so, who distributes work to whom?

Freeing up resources?

Where do you manage resource allocation? Discord, IRC?

10.1.3. Future improvement

Do you ever look up any academic sources/papers/books on how to manage something you are unsure of? With regards to the many roles you/contributors take in the project?

What would you say is the largest challenge in having strong usability/UX in FOSS projects?

How would you address it?

10.2. Interview #1, Jellyfin

Researcher

Ready! First of all some formalities. You will of course be completely anonymous in the project and will receive a draft of the thesis (if you are interested) to verify that everything is in order. You can of course always reach me as well, if there are any concerns regarding something of that nature.

#1

Sounds good :)

Researcher

Great! First of all... What do you do for a living? And how and when did you get involved with FOSS in general, and the project for that matter?

#1

I'm a developer by trade. I work for Odoo, a company in Belgium that makes a partly open source ERP. I work in the services department, making custom apps for clients. I specialize in frontend development, so website integrations, Javascript and such.
I've more or less always been into FOSS since I started doing dev work when I was around 13. For Jellyfin specifically, I had heard of the fork from Emby when it happened. Back in December 2019, I was looking for an alternative to Plex and remembered that it existed, so I tried it and wanted to fix some visual annoyances in the web client.

Researcher

Okay. So would you say that your motivation for joining was your own usage/needs or where there other motivational factors as well?

#1

I'd say it's mostly out of personal needs. I was annoyed with some things at first, then it sort of evolved from there

Researcher

Do you find that most of the project members have a similar story (if you know that is) or are people generally also ideologically driven, or something else entirely?

#1

It's a bit of both, for most, I think. Usually the people that only care about some annoyances they have in their daily use do "hit and run" pull requests. They fix or implement something they want, get it merged, then we never see them again. The people that stay and eventually join the team usually have some ideological reason for continuing to contribute.

Researcher

What about user numbers? Is a large user base a goal in itself for the project? Maybe even as a source of motivation?

or user adoption i should say

#1

We're generally not trying to go for a large number of users. We're trying to make something that is good and if people use it, we're happy, but trying to overtake other alternatives is not really something we even think about. We don't even know for sure how many users we have, since we don't have any tracking anywhere. The only numbers we have are from stores where we can't disable data collection (Like Android installs, iOS installs, etc), but that's not really representative of the number of users we have, I think

Researcher

With that in mind, i can imagine most users are "passive" users. How much feedback do you get (on github/reddit etc) that shape the project in some way? And do you find that interaction with users motivating?

#1

Yeah, from a recent survey we conducted, about 3/4th of the users haven't really interacted with the project in any meaningful way (reporting issues, requesting a feature, etc). We get some user feedback, usually in the form of issues, comments on PRs or feature requests. We try to avoid long pieces of feedback in chats like the Matrix rooms, because we'd like it to be archived somewhere for future references.

Personally, I do like interacting with users when designing something, but I tend to find it a bit annoying depending on how the feedback is made and the technical level of the user (not really in regards to Jellyfin, but in general). There's often suggestions of things that are impossible to make or that would be fairly terrible from a UI/UX standpoint, which we usually have to shutdown or rework.
The project works as sort of a "do-ocracy", as we call it. Meaning that usually, if you really want something done, we expect you to open a pull request for it. We're always available to help in whatever way is necessary, but since developer time is very limited, that tends to be the way things go.

Researcher
Very interesting! You mention some different terminology here i would like to poke at a bit. UI/UX/Usability for instance. Do you in the project have a agreed upon definition/standardization on what exactly these things entail? Is there a difference as far as you are concerned?

#1
The consensus around the main contributors is that UI is how things look, while UX is how things "feel". It's sort of difficult to explain, because none of us really have any design background (We're all either devs or sysadmins). For usability, I'd say it's about how easy something is to use for the first time. Ideally, it should make sense and be presented in a logical way.
To give an example, UI for me would be how a screen looks visually. UX would be the different paths the user can take in that screen (and into it or out of it). Usability would be how easy the screen is to understand
(Not sure that's much clearer 😀)

Researcher
It makes perfect sense! To take a step back again...
What is your perception of these notions? Is it a trivial task in the grand scheme of things? or is it very important? Also, do you feel that subjectivity is a factor in the decisions concerning these notions? If so, how do you deal with it in the project? Do you have any authoritative guidelines to fall back on if a decision cannot be made regarding a design decision?

#1

In my eyes, it's of equal importance with the quality of the code. But I think it takes a low priority for a lot of projects, unfortunately (Talking FOSS in general, here). On the projects I work on and get to design from the get go, I try to think about it at least a little before diving into code (See the new jellyfin-vue client I'm leading)

I think there's a mix of subjectivity and agreed upon rules for all of this. Some of the subjective things would be which case to use for titles in a UI, while an objective thing would be to make the primary action more visible or give feedback for user actions as quickly as possible

Regarding the way the project handles it, it depends. I can only really talk about jf-web and jf-vue. For jf-web, there were a few things in place that we had to keep in for consistency. We tend to do changes in batches, and having something that acts differently in two places would feel bad. Usually, it's up to the main contributors for jf-web, and then it falls back to the core team (redacted)

My "authoritative" guideline would be this: https://material.io/design/guidelines-overview/

Mostly because it has sensible advice, it's simple to use and understand and seems based on good ideas

In that case though, it's less about following the material design visual language, but more about trying to understand why they do things the way they do

Researcher

Understood. With that in mind when you change something do you employ systematic testing of some kind? Do you have some of the same cores users that are always running the bleeding edge that report back?

Also, do you keep track of design rationale? By that i mean the decision for making a certain design change, a design history if you will. For instance on the blog?

*core users

#1

There is nothing really formal. Some of the team (myself for example) runs the bleeding edge version in production. We use it daily and usually some feedback will come up in internal talks. We notice a small handful of users that provide feedback on things as they are implemented or when they are merged, but that number is usually very small.

We don't have A-B testing or any sort of usage tracking to get information on how users do things, so we mostly fly blind until someone raises an issue.
For the design rationale, it's currently not being tracked or enforced anywhere. We have a basic branding page with some guidelines for colors, logo usage and such, but nothing else.
With the Vue client, I'm trying to slowly bring it to a state where we can have these discussions somewhere and have the history available. Usually though, we just start implementing and go from there, since none of us are designers. We iterate with feedback during the review process and sometimes do multiple passes on things (See the rework of the details page that was done during 10.5 and 10.6 on the web client)

Researcher
I see. Will definitely check out the changes you mentioned.
You already touched a bit on this next question but i would like to expand a bit upon it. So as I understand you do not have any people contributing solely on UX/Usability, as in wireframes/mock-ups? or do you have people allocated only for UX/Usability?
And do you have any UX professionals in the project?
Do you feel the project could use UX pro's? Or is it doing fine without them?
Have you ever been approached by any UX pro's? If so what happened?

#1
We don't have anyone doing solely UX/usability or any professional at hand. We'd be very open to it, but I think it's a bit hard to get involved that way in most projects, Jellyfin included. We'd love to have someone with real UX/usability experience to do wireframe and mock-ups for us, as they'd probably do a better job than us and save us devs some time in figuring how stuff should behave and the general look of it, but none have manifested so far.
We don't generally do any mock-up or anything and haven't been approached by someone who would do it.
I've made one for a potential initial setup redesign using Figma, a few months ago, but that's it, as far as I know

Researcher

Okay, that seems consistent with my observations then. You mentioned that with the new jf-vue client you are trying to make a space for discussions. With that in mind do you feel that bug tracking in GitHub is sufficient for all types of issues?
Are usability issues especially hard to keep track off in terms of complexity?
Are usability issues the same as bugs in the code to you (in the sense that they may be descriptive of a series of issues). Also, if one such issue is found how do you manage usability bugs that might require large code rewrites?

#1
With that in mind do you feel that bug tracking in GitHub is sufficient for all types of issues?

I don't think it is. It gets confusing for everyone when discussions and legitimate issues are mixed. It creates noise for both purposes and, as such, leads to issues or discussions getting lost.

Are usability issues especially hard to keep track off in terms of complexity?

People usually don't report them enough, unless they're really egregious. So it's harder to keep track of in the sense that it happens less often, so we might have some that we don't know. But also, people generally try to provide solutions instead of explaining the issue at hand. It makes it harder for us to understand their issue and decide if their solution is really suitable or if it would be another issue in itself.

Are usability issues the same as bugs in the code to you (in the sense that they may be descriptive of a series of issues). Also, if one such issue is found how do you manage usability bugs that might require large code rewrites?

They are. I've had plenty of issues with how the current web client is structured and behaves, which are usually issues we inherited from Emby. They act very much like regular code issues, for me.

As for how to handle issues that require large code rewrites, usually I'm for doing these in "baby steps". For example, if dialogs require deep changes, I'd prefer to change one, gauge how it's received for a while, then expand these changes to more dialogs.

Researcher
Makes sense. Now.. going into more of the social aspects and resource management of the project.

You already mentioned the "do-ocracy". Does that mean that the project is totally democratic (in the sense that pull requests are created when needed) or are there certain people with more pull than others?
Also, do you ever take your work offline to resolve something face to face (if thats an option)? Alternatively you might engage in other social activities in the project that can be considered "channels" where things are discussed or resolved "casually" (such as video games or other recreational activities)
And in general (sorry for the wall of text), are social bonds important in your work in the project? Either between other members or even end users and devs?

#1
Every pull request will be reviewed equally. They may not be accepted though, but that's usually discussed among the team and if a consensus can't be reached, the decision falls to the core team (usually specifically to [redacted]).
We have internal matrix channels, one of which is a more lighthearted channel where we share memes, links to music or random stuff. Only Anthony and [redacted] know each other offline and live near each other. The rest of us are scattered around the world and we haven't had the occasion to meet yet.
For social bonds, it depends. Some team members are more into it and discuss non-Jellyfin stuff daily. Others don't. For users, there's less familiarity, since most of them aren't here for that kind of thing. We do try to keep a community aspect, though. We recognize a bunch of our regulars and those of us who like it engage in off-topic talk with them, but it can be harder to do. In general, it's up to each person how they wish to interact with either the team or the users
They may not be accepted though

This is very rare though. Usually it will be because we don't think that the feature fits in the core of Jellyfin and would be better suited for a client or as a plugin. But it's only happened a handful of times

Researcher
I see. A bit on that same note. What do you assign the greatest value in the project? What would you assign the most resources for here and now (if it was completely up to you)?

#1
For me, it would be code quality. I think that having good, readable and easy to approach code leads to more contributors, which then leads to more features and a more active project. Everything sort of falls into place if the project is easy to maintain, I think.

Researcher
Okay, that makes a lot of sense.
Finally, the last couple of questions i have for you are a bit more general on the project and maybe even FOSS in general.

You mentioned the material design guidelines as something external you might use to verify your own decision making. With that in mind do you ever look up any academic sources/papers/books on how to manage something you might be unsure of? (With regards to the many roles you/contributors take in the project, one day you might be a UX designer, developer or even project manager)

What would you say is the largest challenge in having strong usability/UX in FOSS projects? And how would you address it?

#1
I don't really reference academic papers, but I do some Googling for things I'm not sure about (One I saw this morning would be "Should you load a modal if it's content is empty, then load the content, or should you wait until the content is loaded?"). It's basically the process devs use

to figure out stuff we don't know, so I apply the same methods when dealing with UX, Usability or even UI design.
I've been trying to get more into the theory behind UX, so I'm looking to pick up a few books in the coming months, but haven't yet.

I think the biggest challenges for strong usability/UX in FOSS are developers themselves. There are multiple things at play:

We develop for us, not for users. This can lead to bad patterns and hard to use software.
Some don't like the organization that having a UX/Usability team provides. A lot of developers see FOSS projects as a playground after work. Like you work on things that aren't interesting for money, but then you work for fun in the evening. Having more organization and a process (RFCs or a design process) might feel too rigid for some
A lot of developers aren't trained for UX/Usability. They do their best, but without a particular interest in it, the opportunity to do it with some guidance or proper training, it's hard to grasp.
Ideally, I'd like more developers to take up an interest in this. Perhaps stress the importance of getting some UX concepts when training developers. Getting devs to understand that UX is just as important as code and should be treated with the same focus and care

Researcher
Very interesting ideas! I think thats all i have left 🙂
Its very rewarding to have your findings verified, and its interesting that the challenges in the field (according to my studies so fare) hasn't changed much since the early 2000's. More stuff to investigate for the thesis! 😀

#1
Glad I could help :)

Researcher
Again, huuuge thanks for your time! I can see how scarce it must be in the project. And on a totally different note, thank you for saving me from buying a plex pass. I have a Dell t30 sitting in my closet that is just waiting to become my Jellyfin server 👏

#1

Thank you for using it 🙂

And for considering us for your thesis

It's very interesting stuff 😛 And FOSS needs less focus on devs and more on UX, documentation, translators and such

Researcher

I found the project very interesting also because your UX/UI is very much up in the air after the fork from Emby i can imagine, defining where you want to go and such. And yeah totally agree. So much powerful software that is not seeing enough love in my opinion.

#1

I agree, the "blank slate" is very exciting :)

I've been the one spearheading some of the large UI changes like the current details page, so I'm taking full advantage of not being tied to Emby, for sure 🙂

Researcher

My goal is to create some design that can help devs from projects of all sizes approach these issues somehow. Maybe a redesign of github to facilitate the considerations you have to ensure that these discussions are being had, they can be long and feel unnecessary but they always create better software in the end. Very nice to see that the ideas and the motivation for it is there! One of my key findings from the academic side of things indicates that devs also usually discount their own efforts to develop better UX. It sounds like you might be underselling yourself as well ;) Again, huge thanks. I have an interview with redacted lined up as well and hope some of the other project members also want to participate.

#1

Underselling yourself is a classic developer move, I think lol

Good to know that I wasn't the only one to answer :)

I hope others will as well

You're working on something good, I'd love to get a copy of your thesis to give it a read when you're ready :)

Researcher

Sure! It will likely go through many iterations but i really hope it can actually be used by FOSS projects for considering what needs to be discussed with regards to UX. Signing off for now. Thanks again, will let you know when i have something interesting 😁

#1

You're welcome 🙂 Have a good evening

Researcher

You too!

10.3. Interview #2, Jellyfin

Researcher 0:02

Great. So yeah, just some formalities first, I guess, obviously this is totally anonymous and of course it won't be shared anywhere in your name. It's totally up to you also, you'll receive a copy of my thesis. If you want to read the whole 120 pages, that's up to you, but at least you can confirm the transcript is correct if you want. Yes.

#2 0:43

Okay.

Researcher 0:43

Let's see here. So, great. I'll just find my notes here. Okay. So a bit about the project. So what I'm basically doing is I have a specialization in UX design from the, or I'm getting it currently, from the IT university of Copenhagen, and I have been involved in various FOSS communities over the years, mostly as just some bug reporting and stuff like that. But I would really like to merge my interest for UX with FOSS communities. And there's already some preliminary studies that have been done, but a UX or usability I should say in FOSS is kind of a new field still. So I'm pretty much just interviewing different projects and yeah. To, to get some, some knowledge to kind of verify previous findings from studies and then also possibly create some new findings and yeah, no, that's where you're come in. So, so first of all, what is, what is your day job? What do you do for a living?

#2 2:01

I'm a software developer for a telecom company.

Researcher 2:05

Okay. And you're associated with Jellyfin. And how were you involved or how did you get involved in the first place with Jellyfin?

#2 2:17

I used Emby a lot and then I saw that Emby would close source and, you know, the whole history of it. It was following along and was making some plugins for it. And then all of a sudden they're like, “Hey, we need server developers”. And I was like, “I can do that”.

Researcher 2:34

Okay. So that was.. So partly for your own, again, I'm guessing you're a user as well.

#2 2:42

Yeah. I use it all the time.

Researcher 2:44

Yeah. Great. But also for, I can almost tell the kinda ideological standpoint on having it being open?

#2 2:55

Yeah. That's, that's the main idea. It’s so much better. Just to be able to go in there.

Researcher 3:02

Definitely. I definitely agree as well. Would you say you're primarily ideologically involved or is it more your own for your own gain that you contributed in the first place?

#2 3:21

I primarily would say it's ideological. I don't fully use Jellyfin yet because the apps aren't there, but I'm excited about the prospect of a fully open source and free and all the great things about Jellyfin that are coming out. Yeah.

Researcher 3:41

And you historically, you, you said you were a backend developing, so primarily I'm guessing the server side of things is where you're residing. So about motivation for developing and contributing in a FOSS project in general, would you say that people in the project, would you say they, they are like you ideologically driven mostly? Or is it more like a, let's say a social, a need to interact with people or something else entirely?

#2 4:15

The people I interact with definitely have the same ideas. Like they want the server to be as good as it can be and they want to work with the community to make it happen. And the collaboration is super good. Like we have an idea, we throw it back and forth a couple of times. And then all of a sudden it's, it's implemented and everybody's like “Oh, that was really cool”. But my at my day job, I'm basically a solo developer. So I, I really liked the collaboration of working open source.

Researcher 4:48

Okay. That's a, that's also one of the findings I'm kinda, I'm kind of poking at a bit since like, so the, let's say the social reward of interacting with different people all the time might be, it might be something I can, I can hear being a motivation for you then.

#2 5:04

Yeah. You definitely learn a lot of different things. Working with different groups of people.

Researcher 5:08

Would you say it's mostly developers you're interacting with, or are you let's say not maybe not equally, but almost as much maybe talking to people who report bugs or like end users?

#2 5:20

I would say it's mostly developers, but though there are a couple of users that come back and work a lot of bugs and they do have very thorough bug reports. So come back and we interact with those people very distantly.

Researcher 5:36

Okay. Okay. So let's say like “core users” or whatever we could call them.

#2

Yeah

Researcher 5:40

Okay. So the feedback, that you mentioned, the feedback they gave these core users, is that a, so that's very detailed. I can hear they might be technical users themselves and able to provide detailed reports.

#2 5:56

Yeah. Either that, or they've already reported something and they have the knowhow of how to do it again to make it easier for everyone.

Researcher 6:04

Do you ever find the feedback to be not detailed enough then? I can imagine though.

#2 6:09

Oh yeah. All the time.

Researcher 6:11

And is that, how, how do you, how do you deal with that when you approach issues that are not detailed enough?

#2 6:17

When we specifically, when I go to it and I see if there's the bare minimum of information I say, “Hey, can you send a log? What were you doing? What version were you on?” Because normally there, if there's not enough information, it's basically, “Hey, I can't say I can't play videos back.” You're like, okay, that doesn't mean anything to me. So I need the rest of the info from them.

Researcher 6:39

Right. So you need something to kind of create value from the, whatever they post in a sense.

#2 6:45

Yeah. Some way to reproduce it.

Researcher 6:48

So how would you say that shapes the project, this, this feedback you said, is it a burden or is it a general, like generally a positive thing that you were getting feedback at all in the first place?

#2 7:00

Oh, it's amazing to actually get feedback. If there was no feedback, we would be building inside of a bubble. And then we probably wouldn't, we wouldn't know that things were breaking as constantly or as quickly as they were. Cause yeah. It's, there's a lot of parts of Jellyfin that not necessarily everyone has access to be able to use. So like, like for instance, live TV. We apparently broke live TV, but we don't really have a lot of our, like the core developers using live TV. So it really wasn't tested until a bunch of people came out and said, “Hey, it's not working anymore. Here's what I did”.. to yeah. All that stuff.

Researcher 7:49

Yeah. Obviously that requires some hardware that actually can support that I can imagine. That's really interesting. A really interesting thought. What about, let's say passive users. I don't know if you have, I don't think you have any metrics. If I remember correctly in terms of like user base, do you, apart from obviously like a Docker pulls and numbers like that?

#2 8:11

Well, we ran a survey. Was it a month or two ago? And we have the results from that. I didn't look too detailed into it, but it looked like a lot of, we had, like, I think I might be able to pull it up.

Researcher 8:26

Yeah. I think, I think I actually recall. I think [redacted] and also told me, I think it was like three fourths are passive users if I remember correctly. So like 75% I think were passive

#2 8:40

Sounds correct.

Researcher 8:44

Do you feel, do you feel it with that in mind that there's enough reporting then, or do you feel that there might be some need for having more people to report or let's say crash reporting or something else to fill in the blanks?

#2 8:59

I think the amount of active users that we have, like the ones that report things is a good amount because we do get a lot of duplicate reports as well. So I, I think we cover most of the issues with the active users. So if we had more passive or converted to active. It might be more confusing for us or more, more work.

Researcher 9:22

It might simply be an extra burden. Interesting. Okay. So a bit about usability. So what is your perception of it just as a, as a backend developer, the concept of usability, or let's say it doesn't even have to be that term. You can also call it UX, or is there any difference to you between those terms? That's a UX and usability and UI, like how to, how do they, how do they let's say feel to you?

#2 9:56

Well, to me, it's not really a thing that I really think about. Like, I, I spent a bunch of months working on converting the API to something that was actually usable and the, like the front end people, the client developers can actually look at the, the end points and be like, “Oh, we have 700 end points that we didn't know we had”. And that's, to me, that's the usability that I focus on. But I do pay every once in a while, we'll post some, one of the client developers were posting in the chat being like, “Hey, I'd like some feedback on this app I'm working” on and I'll, I'll go through it and say, cause I am not a front end person at all. I don't know any, any of that stuff, but I can say looking at the app to me, this doesn't make any sense on how it's implemented in the cause they were working in their own bubble and then they get outside and they're asking people, and it's always good to get feedback, especially from people who aren't also front end developers.

Researcher 11:03

Yeah. That's a, that's a good idea generally. Yeah. So you, you kinda mentioned it a bit here, but just so you'll basically try to just use like common sense logic to, to judge it.

#2 11:17

Try to try to make it easy for like my parents who don't know computers.

Researcher 11:22

Exactly. Okay. So is it, is it then like a trivial task, would you say, or is it a very important, how would you rate it as a, like let's say in the sphere that is Jellyfin where would you place it on a scale?

#2 11:36

I would say usability is very important, especially right now, because we're working on the Vue client, I'm sure you've seen.

Researcher 11:46

Yeah. Yeah.

#2 11:48

They're working on making sure it's completely feature complete with the previous or the current web client, and also looking at how the previous or the current web client is implemented in saying, why was it like this and trying to make it a little bit easier to use.

Researcher 12.10

Right. And then maybe a good follow up question to that is that if, if your usability is important, I'm guessing it's because there's this inherent wish that somebody will use the software. And also let's say as a personality profile to pick “mom and dad”, can actually sit down and use this somehow.

#2 12.34

Yeah.

Researcher 12:48

Would you say then that the user numbers or user adoption is then a goal of the project as well? Like, do you do strive to have more people use the, Jellyfin as a product?

#2 12:48

I wouldn't say we're actively like advertising the product. Cause at the end of the day, it doesn't really matter. At least to me, it doesn't really matter if a million million people use it or like a thousand people use it. Cause we don't, we don't get any paid payment or anything. It's just nice to see, “Hey, I set up the server, it works great”. I can click everything and it just works, but it's, it is nice to have some users so we can of course get the reporting back and reporting back on. But in the end, it's up to the user on what they, what software they choose to use.

Researcher 13:25

Okay. So just, just to verify if I'm hearing you correctly, so you find the interaction with the other developers and some of the core users more motivational or more motivating than let's say just the raw amount of people using the software.

#2 13:45

Oh yes, definitely.

Researcher 13:46

Yeah. Okay. That's, that's really interesting. A really interesting notion fits actually quite well with what Julian said as well. So something else I know you're not directly involved in usability, but do you, for instance, when you are, when you're judging these screenshots, you mentioned, do you have any, like, do you ever look up a usability guideline or something like an external source on that?

#2 14:17

I don't myself. I don't even know where I would look to see that I just go off of what my feeling is. If it's too complicated for me, I know most people will not be able to follow it. I'm, I'm a pretty technical user.

Researcher 14:34

Yeah. Yeah. I can imagine you are not the, not the average, Joe, trying to watch a movie, then maybe something else that's a bit more up your alley. Do you have any kind of systematic testing you do? Let's say you get to some, some issue you have to work out. Do you have the same procedure you test every time to reproduce some issue or do you just, it's like try to start in, in whatever area that makes sense to you?

#2 15:06

Normally I look at the logs to see exactly where it was throwing an error. Then I worked my way back to see if it was like from an API or from like an internal, like service kind of thing. Right. And then I start from there and see, and like step through and trying to find exactly how it got to that place.

Researcher 15:27

Right. Okay. Do you know if you have any people in the project contributing solely on, on usability? So like a resource or a tool that's just only doing UX or let's say a UX professional, even in the project?

#2 15:49

I would say, I don't think so. I think most people are just straight developers. Like the, the person that Julian you talked to would be probably the closest person. Okay. For usability. I know we had some people reach out to their colleagues and say, “Hey, can you take a look at this and see if you could …”

Researcher 16:07

Oh, okay. Okay. So you're using acquaintances from work or whatever.

#2 16:12

Yeah. We had some people for accessibility come in and give some pointers on making it more accessible for say color blind people or yeah, stuff like that.

Researcher 16:25

The, the, the people that, that looked at the accessibility who, who were they again?

#2 16:31

I, we never spoke directly. It was more like the, I don't even remember who asked for the assistance. I just remember we got like an Excel or a spreadsheet, or I guess it was a document with some pictures and some pointers on what to improve on.

Researcher 16:48

Oh, okay. That's interesting. So like an external person, someone provided some input there. Okay. Do you know if the project has ever been approached by any usability professionals? And if so, what happened?

#2 17:05

I can't say for sure. I know if, anyone were to approach, they would probably approach either [redacted] or [redacted]. They are, basically the face of the product every once in a while, they'll be like, “Hey, I got this really interesting email, but I don't think any of them have ever been a usability.”

Researcher 17:24

No. Okay. It is at least consistent with what I found, it seems that it's a pretty big barrier of entry for usability or UX professionals for that matter into FOSS projects is, that's why I'm asking if there's even someone contacting you. Do you feel the project could, could use a, let's say a project member that's solely doing UX contributions, let's say mock mockups or wireframes or whatever. Or do you have to, do you feel you're doing fine without any?

#2 17:58

I feel like it would be really interesting to see what they had to offer. It would, I think it would be really neat to have say all of our applications once they eventually stopped being a web wrappers to have like a semi cohesive UX feel to them.

Researcher 18:21

Yeah. Cohesion in between those different apps there. Yeah. It makes sense. Let's see. Do you know if you have any, let's say methods or our sites or pages or whatever in the project that has a focus on usability? Let's say you, you, I know you have a blog, for instance, that would be one. Do you have like, a thread or something where you discuss usability changes or where people contribute somehow?

#2 19:01

I don't think so. Normally it's just someone going through and being like, I want to make this a little bit easier to work with and a little bit nicer to look at. I know each like the web repo and the, the Android TV repo has a thread? Well, I don't know about the web repo, but at least Android TV has a thread on what they're going to be changing and all of that. And then what the goals are. I don't think there's a central location for that.

Researcher 19:35

Okay. Do you feel that that generally GitHub is sufficient for all types of issues, then

#2 19:44

It's pretty efficient yeah, but we're, we just started looking at GitHub discussions. Apparently, or a new thing, which could be very interesting to hookup. So we can have like longer discussions that aren't just issue based or in matrix chat. So I, that was an idea that was floated around a couple of times. I think they started working on that.

Researcher 20:08

And thats just an add-on you enable or?

#2 20:11

Yeah, I think so. I don't know really much about the GitHub administration side. I think it's just you, you hit a checkbox and all of a sudden you have a new tab to, to go into. We do have the forums as well, but those aren't really visited by many people, I wouldn't, I wouldn't go there for usability stuff.

Researcher 20:32

No. Especially if it's not getting too many yeah. Visits. So a bit about the complexity of usability bugs, would you say they are, let's say similar or the same as a standard bug in the code to you? What'd you say that they're identical? I can imagine some usability bugs might span multiple, multiple lines of code, or let's say multiple screens or frames or whatever we can call them. That has to be reworked.

#2 21:12

They're similar in complexity, especially with the current web, how it's written. I, I looked at it a couple of times and I'm no JavaScript developer, but even to me, that was way too complex to just make any changes in. And the way the, the API previously was, it was almost impossible to figure out where the entry point into the program was based on what you had. It was, it was a disaster, but now it's a little bit easier. And now we have more contributors actually looking at the API and going in and making some changes. So I guess, I guess the backend might feel a little bit more important because it's really hard to fix the front end if the backend is super broken.

Researcher 22:01

I can imagine using duct tape, everywhere to fix stuff. Also, I have a bit about like the social aspect of some of the resource management in the project. Do you ever take your work offline to, like, let's say there's a decision that requires some kind of consensus or something. How do you, how do you deal with that?

#2 22:26

I'm not sure what you mean by that.

Researcher 22:28

So, let's say there's a bug report filed or something that says that some kind of… Let's say the placement off the movie poster or something needs to be moved in order to, you know, make space for something else. Some user says that makes sense to them or whatever, or maybe another developer thinks so. How would you then let's say you, you don't agree in the project. How would you kind of decide on what to do?

#2 22:59

I've been pretty hands off with the web decisions or the actual design decisions. But as far as when I pay attention to chat, there's a lot of back and forth between the, the couple of members who really, really care about what the design of the website. And then eventually they, they just come to a consensus and, and one of it's usually like two against one, unless the, the one person has a really, really good reason. It usually moves towards the majority. And so, so far it's been working for us.

Researcher 23:38

So like a pretty traditional democratic approach to that then. Okay. That's interesting. Let's see here. We already talked a bit about this. Yeah. Eh, I kind of asked about this already, but just to clarify, what, what would you assign the greatest value in the project, if you, if you had to pick?

#2 24:08

In terms of what?

Researcher 24:09

Well, in terms of what, what to assign, let's say the most resources to fix immediately

#2 24:17

Right, right now I would definitely say the database and scanning. Because that's a very, it's a big pain point right now. It's very slow to scan and to do anything. And that's, it's actually a really big usability concern because when you add new media, it doesn't show up right away. And then it shows up an hour later and it has no poster and no details. And then eventually it has all that stuff.

Researcher 24:43

Okay. And that's a, just a, basically a database issue then.

#2 20:08

Yeah. But the database is a big mess and we have a couple of members working on that right now.

Researcher 24:46

Okay. So that's a, some backend stuff that kind of bleeds into the usability side of things.

#2 24:58

Yeah. Have you used Jellyfin before?

Researcher

I yeah, I tried it. Yeah. I've tried to, all of those media... I really liked it honestly, but I didn't, I didn't, I didn't use it long. My, my server is currently down, so I haven't had much time to play with it, but it's, it's, I'm thinking, gonna go into being my main centralized media solution, I guess.

#2 25:23

Cool.

Researcher 25:24

…Is my plan also, maybe something here. Let's see. Yeah. How do you manage resource allocation actually? Is it first come first served or do you have roles or certain issues where you always will jump in? Or how do you distribute the work?

#2 25:44

Primarily, we have a couple of areas of the code base that we each specialize in just by nature of what we started working on. And then we, we typically would be like, “Hey, if you have a moment, can you take a look at this?” And if they don't, if they don't have time to look at it or whatever, someone else will jump in, who's a little bit familiar with it. But the, the code base is so massive. It's hard to know it all

Researcher 26:12

Right. Yeah. I can imagine. So how'd you manage that? You just chat on matrix? I can imagine, or?

#2 26:22

Yeah, usually so far it's been pretty quick. Like something, a bug pops up and someone will just jump on it the day of, or next day or something. If it goes for a while, then we'll, we'll poke them and be like, “Hey, if you don't forget about this thing”, and if it's not something you can handle or if you know how to fix it, well, someone else that can a look

Researcher 26:47

And then a bit about future improvement. So, basically what I'm trying to do with my project, I kind of already gone over it a little bit, but I really want to see how some of these issues are being addressed. And then create some, let's just call that a design right now. Something that can be consumable enough for new or existing FOSS projects that they can maybe make more coherent design decisions across the board is my goal. And with that in mind, do you ever look up any, let's say papers, books, sources on anything you don't know how to manage, if you're completely, let's say unsure of, of the realm you're going into and, and if you do, what, what form do you most like to, to like consume that content from like video texts? What, what do you prefer?

#2 24:47

It really depends on the type of content. It is nice to have videos of, of, of some things, but sometimes the videos are just too long and slow that I give up and try to find the text, text version of it, but both are really nice. It's good to be able to just like copy and paste the code if you need it. Or, and it's also really nice to see how they work through it and how, how, why, why it is what it is. Right. It's this really nice.

Researcher 28:23

Yeah. Interesting. Okay. Yeah. That's, that's also kind of what I'm debating right now. What, how to best presents maybe someone who's never done any UX or usability to, to have that field and since every developer in FOSS projects, usually you have to kind of juggle roles a little bit, at least at least more than that, than what they do at their primary job usually. So yeah, it is, it is kind of a difficult thing to design for. How do you compress all that into some kind of consumable format? And then also, what would you say is the largest, challenge in having a strong usability in general and FOSS projects, have you ever used some FOSS project where you thought this is horrible to use?

#2 29:17

Yeah. Every once in a while they, the, the, the settings are buried too deep, or it was written by a developer for a developer. And then you're, you're not that kind of developer. So you have no idea what some of the words mean. Those are always fun to get.

Researcher 29:34

How would you, how would you address like that issue of, of developers developing for developers? How would you..

#2 29:46

So I would show it to nontechnical users, especially if it's not a specific technical, tactical based project. Sometimes like I'm using this, this program for Dungeons and dragons, character creation. And the way that I use it is very different than all the other people in my group use it because they aren't very technical and they read the, on how to use it and they have no idea what's going on. And then I explain it and they're like, Oh, okay. So I hit this button..

Researcher 30:25

Funny you should mention it. It is, it is oddly difficult to get into D and D. For some reason I found that as well.

#2 30:33

It's like, once you get into, it's really hard until you figure out your first little bit, and then it's so much easier.

Researcher 30:40

So, yeah. That's, let's see if I, I think we got pretty much around all my questions. Let's just see if I have any I missed. No, I don't think so. Yeah. I think actually, that's, that's all the questions I have if I have some, epiphany, all of a sudden than I, I remember some really important thing. Is it okay if I shoot you a question?

#2 31:17

Oh yeah

Researcher 31:19

I think I got it all here, but, but, but yeah, it seems, yeah, that was really helpful and really interesting. And I, I think a lot of the backend developers will, will likely, like you did also think at least, “Oh, I, I probably don't know a whole lot”, but one of my key findings from a lot of the research that has been done previously on kind of the same is that a lot of the developers are really doing a lot actually to improve usability. They just don't know, like the, let's say the method names or what you already mentioned, like, you know, user testing. So it's a basic concept, but it's such a huge requirement for so many things when you have to when you have to design a product.

#2 32:04

Interesting. it's funny.

Researcher 32:06

It is, but it's a, yeah. This project here is basically fueled by, by, by me thinking that it's such a shame that there's so much powerful software out there. That's just held back basically like by design. And, and then it's interesting since, you know, the user base might not always be like a huge user base might not be the goal in itself. And then how do you, you know, get people using it anyway. And then there's also the whole issue of why do usability professionals not join these projects in the first place? Yeah. So that's what I'm trying to kind of mitigate all those issues somehow.

#2 32:50

Interesting.

Researcher 32:51

Yeah. Thank you for your time and so awesome you wanted to participate.

10.4. Interview #3, Taskcafe

Researcher 21:34
Hey Im ready in a bit. Are you still free?

#3 22:11
Yes

Researcher 22:12
Great! First of all some formalities. You will of course be completely anonymous in the project and will receive a draft of the thesis (if you are interested) to verify that everything is in order. You can of course always reach me as well, if there is any concerns regarding something of that nature.

#3 22:12
Sounds good!

Researcher 22:12
Great!
So. First of all, what do you do for a living?

#3 22:13
I'm a web developer for a company that makes websites exclusively for manufacturers

Researcher 22:14
Okay, and how does that tie into how you get involved with FOSS in general, and Taskcafe? When and why did you start the project?

#3 22:19
Honestly it really doesn't tie in at all. I've used FOSS since before I started my current job. For me, Linux and vim (my preferred editor) is what got me into FOSS. I started Taskcafe mainly as a learning project to get familiar with some new technologies but also to create a project tool

that fit my needs and ideas of what such a tool should entail. I decided to release it open source in the hopes that it helps at least someone. I started actually working on the project about a year ago off and on, with much of the initial framework being written in December.

Researcher 22:20
Interesting! So what are you motivated by mostly? Would you say you are ideologically driven? Or is it more to hone you own skills?

#3 22:23
I would say its mainly two factors, one is to hone my skills but the other is that i just simply enjoy building things and taskcafe allows me an outlet to do so. And since it's a project that I use in my everyday life, it helps keep me motivated to continue working on it rather than switch to another project

Researcher 22:25
i see. You also mention that the project might help others, if its kept open. Would you then say that you are also motivated by an increasingly rising user base? Or is the raw number of users not that meaningful to you?

#3 22:26
The rising user base is certainly interesting! But in the end it's more of a side benefit and not one of my primary motivations for working on the project

Researcher 22:28
With a rising user base, i can imagine more feedback and issue reports might also be coming your way. What is your take on feedback? Is it a burden? Is it generally a nice thing?

#3 22:32
Hmm it's certainly a double edged sword. It's nice to have users as they can help catch things that I might have missed as well as suggest functionality that I might otherwise not have thought

about. But the cost is bugs and bug fixing, specifically dealing with hard to reproduce bugs that only happen on the users end. Those types of bugs are annoying and very time consuming to try and fix, not to mention mentally draining.

One of the other "burdens" would be the feature requests I have to say no to. Sometimes there will be requests that I just won't be able to easily support in my limited time to work on the project and saying no is sometimes hard to do when a user cares enough to suggest something.

Researcher 22:34
Okay, i see. What about the interactions themselves (with end users). Are they somewhat motivating to you as well? Or do you feel more that its a resource hog?
I can imagine the feedback might have 1) varying quality, but 2) i can also imagine that people might be super satisfied with the software. Have you had any experiences that stand out to you maybe?

#3 22:37
Since I'm not really flooded with interactions, I wouldn't say they're a major resource hog and the interactions give me an opportunity to kinda step away from the code aspect of the project and think about things from a users perspective.

The biggest experiences that stand out to me is the several people on my discord that have stuck around for a while and actually been helping others with questions about the tool - I thought that was really exciting

Researcher 22:39
A little community appearing out of nowhere. I can see how that would be really cool!

Now to go a little closer to the concept of usability. What is your perception of usability/ux/ui and all the other terms that might be used interchangeably in the industry. Is there a difference to you? What do these terms entail according to you?

#3 22:42

To me, UX and usability are essentially the same thing and they mean how a potential user can effectively use a tool with little training or help (and the process of getting a tool to this point). UI is just the visual elements of a tool and how they look - which plays an important aspect in UX as how a certain element looks can drastically affect how a user might go about interacting with it

Researcher 22:44
I see. And is planning a design with these things in mind a trivial task would you say? Compared to say fixing bugs in the code? How do you normally approach such a task?

#3 22:47
I'm a developer first so I wouldn't say I ever plan a design with those things specifically in mind, but instead I usually research the different way other's have designed similar complements and take the aspects I personally like and transform it into the design (which almost always gets further tweaked as I use it and encounter issues).

Researcher 22:47
Okay, so one could say that you use existing designs as a reference to inform your own

#3 22:48
Correct
Like I said I'm a developer first so designing components from scratch is not my strong suit

Researcher 22:48
Do you then also use any other external methods or tools? Like human design guidelines or something like an established design framework?
Aha i see
Lets say, material design from google or something similar

#3 22:49
Not really, mainly I just have a color scheme in place and some general numbers that I keep consistent (border radius, padding, margin, etc)

Researcher 22:50
Okay, makes sense. Do you keep track of design rationale? So, the history of the design and how it evolves?

#3 22:51
I don't, though looking back I kind of wished I had if for no other reason than curiosity

Researcher 22:51
Has the design changed alot from how you invisioned it then?
*envisioned

#3 22:52
It has! The entire app "shell" has changed from the original design

Researcher 22:53
Interesting. Do you remember how that came about more specifically?

#3 22:54
Mainly I just was never happy with the initial design and as I worked on it I got a better idea of how I wanted it look and function. For example there used to be a nav sidebar but I just never could get it to look right and it just never felt a natural part of the tool, so I removed it

Researcher 22:55
So it wasnt from any external feature request or something that drove that change? Just from your own testning.

#3 22:56
Correct - i would say 99% of changes were driven by my own testing (at least at this point in time)

Researcher 22:58

Right. One thing im a bit curious about. Would you say that you employ any methods in the project between contributors/end users with a focus on usability? I can see that your website has a blog sort of feel, also the suggestions channel here in discord. How are those factoring into the project?

#3 23:00
At the moment, I would no (at least nothing I'm aware of) mainly do to the fact that the project is still in alpha so many things are changing drastically and hopefully for the better. For example right now I'm redesigning how a user shares a project to make it a bit better

Researcher 23:02
Right. Would you then, with regards to the types of feature requests and bug reports that involve usability, say that GitHub is sufficient to handle all types of issues? Or are there certain things that are not historically awesome to handle on there?

#3 23:05
With the introduction of issue templates, that's certainly helped GitHub be a better tool for managing issues that relate to usability. I'm not aware of any others that do it better either

Researcher 23:07
Would you say usability issues are harder or more complex than other bugs? I can imagine a single issue spanning many required changes and such. How do you address those issues/feature requests that are to time consuming you mentioned earlier?
Where you had to potentially turn them down i mean

#3 23:11
They can certainly be more complex as many of the issues can be personal opinions. What works for some users, might not work for others (or me).

They also usually require many changes like. If it's a feature request or issue that I plan on fixing, then I'll add it to my backlog (unless it's something urgent in which case I usually work on next).

If it is something that is simply too time consuming to do for me personally but I could maintain such a feature, then I'll usually suggest writing a PR themselves (or marking it as help wanted so others can find it)

If it is both too time consuming to implement and maintain, then I thank them for their suggestion and let them know why I have to reject the suggestion

Researcher 23:15
I see.

I also have some questions regarding some of the more social aspects of the project i was wondering about, especially in light of the discord server. Would you say that social bonds are important factor in the project? Maybe certain users are more diligently reporting issues than others? Maybe there are even some friendships forming with other devs helping or even end users?

#3 23:18
There are some users that create more bug reports / suggestions than others which is nice to see. There hasn't really been any other devs that have stepped up to implement some other features (except for one who did a small PR), so I can't say the social bonds are important (at least at this stage in the project) but the social aspect of using the discord has been helpful mainly
for getting others ideas on how to go about implementing certain features. For example, one of the things I've looked to others for suggestions on in where to place the new "Share" button that I am working on and I get some helpful replies from other users.
One of the main reasons I made the Discord, was for a place for users to ask questions on how Taskcafe works without poluting the GitHub issues, which has worked pretty well

Researcher 23:20
Very interesting. And a nice workaround compared to github i can imagine! What would you then assign the greatest value in the project? Where would you put the most energy right here right now as it stands?

#3 23:21
My biggest focus on the project right now it implementing the "base" features - features that I can consider essential to any project management tool. Once those
are implementing, I can work on the things that make Taskcafe special
(and are generally more fun to implement)

Researcher 23:22
Understood. Makes a lot of sense.

Finally, my last couple of questions are about future improvement for the field of FOSS with usability/UX in mind.
What, if and when you look up external guidance for something what format do you prefer? Video, text or something more interactive?
Also, What would you say is the largest challenge in having strong usability/UX in FOSS projects? And what would you do to address it?

#3 23:25
I personally prefer text with some demos

The largest challenge at least for me is the time to ensure the various features in my project have it - and the only way I can address it right now is to slowly chip away at it when I can

Another challenge, at least as a solo dev, is getting others opinion (someone with fresh eyes) on how a feature looks to an outsider. That's a bit better for me now with the Discord having some members.

Researcher 23:26
Okay, so for instance being able to get ux designer feedback somehow would be of value to you?

#3 23:27
It would

Researcher 23:28

Great. I think we are all the way through all my questions. If i have an epiphany of something missing is it okay if i send a follow up question at some point? :slight_smile:

#3 23:28

Sure

Researcher 23:30

Great! Thank you so much for all your time. So awesome of you to participate in the project, great input all around. Interesting stuff. I have also talked to 4 devs at Jellyfin (a selfhosted media server solution if you are not aware) and the findings are so far very interesting all around.

#3 23:31

Happy to help!

Researcher 23:34

Have a nice day! Danish time says 23:30 so i will log off for tonight. Again huge thanks, and looking forward to setting up Taskcafe on my server. I have a bachelor that involves quite a bit of project management and it looks really nice, so super stoked on trying it out. :thumbsup:

10.5. Interview #4, Jellyfin

Researcher 1:21

So first of all, what is your day job? What do you do for a living?

#4 1:27

So I am an implementation engineer. So I do front-end and back-end development.

Researcher 1:34

Okay. And when, and how did you get involved, with FOSS in general and, and the Jellyfin project specifically?

#4 **01:43**

So I've been involved with FOSS I guess, going back to college when I first started running Linux. So that's been about 12 or 13 years. Jellyfin is probably the largest project that I've worked on. And I started with that right after the fork from Emby. That was in December of 2018.

Researcher **02:16**

And your motivation for joining, was it for your own gain? You had some issues with using it or was it more of an ideological, let's say, standpoint and it being open in terms of the whole Emby thing that happened?

#4 **02:34**

Yeah, it was more ideological. I had used because they were open-source and then they went closed source. I was contemplating even switching to Plex, but before that happened, someone sent me a link to the Jellyfin announcement and then I joined the chat there and it all kind of went downhill from there. (Laughing)

Researcher **03:00**

That's a good way to put it. Yeah. Okay. So, you're more of the ideological type by I'd take it then. Okay. And so, what do you do in the project now and what have you done historically? in Jellyfin?

#4 **03:14**

So I've mostly been involved with the client side development. I started out working on the web and Android clients and started working on the Android TV client and, created the iOS client. And now I just kind of bounce around between the four of them.

Researcher 03:41

So a bit more about the, or your motivation rather for developing and contributing. Let's see. Okay. So I was wondering how much of an interaction would you say you have first of all, with the other developers and end users, and also would you consider those social interactions a motivation in itself or a, I'm just thinking some of my, some of your, let's say colleagues from, Jellyfin has at least expressed a, let's say that there's a difference between developing for FOSS projects and then doing it for, for a normal job, let's say, do you feel, do you feel that way or what's your thoughts on those social interactions?

#4 04:37

Yeah, I definitely think the social side of it plays into it. I'm in the chat pretty much all the time. And I don't think I would be nearly as motivated if there wasn't that interaction, but as much time as I do to the project.

Researcher 04:56

Okay. What about end users? Do you have any end users that are specifically, let's say at dedicated to like reporting bugs or, whatever. Are you interacting at all with them?

#4 05:15

As far as like responding to bug reports that are, open?

Researcher 05:21

Yeah, exactly. That's, that's one that's one way, right. I'm just, I'm just thinking, just kind of just any interactions, really the I'm guessing, not all of the members of the matrix chat are actually let's say contributing necessarily. They might just be there for information or suggestions or something.

#4 05:43

Yeah. That's, that's accurate. There's definitely some troubleshooting that goes on there. I do participate some in that if it's something that I I know about as well as on Reddit and, and GitHub issues.

Researcher 06:00

Okay. And do you, do you find that motivational at all that there's this interaction or is it more of a burden in terms of, you know, I can imagine some of the bug reports or something might not be technically adequate or in terms of trouble shooting and all that.

#4 06:16

Yeah. A lot of times they can be lacking information that can be kind of annoying.

Researcher 06:27

Yeah. I can imagine trying to, I've worked a, like a basic it service desk job for, for a while. So I know that's that struggle trying to, yeah, get all the details you need. Again, on the end users. Would you say numbers are important? Is that a large goal of the project and itself having a large user base or?

#4 06:51

No, I don't think so. It's mainly just having something that meets the needs of the people who are working on it.

Researcher 07:02

Okay. So you don't find a, let's say yeah. Some of the numbers you do have, I can imagine you probably don't have a whole lot of numbers, maybe only from the app stores. I can imagine?

#4 07:15

Basically all we have is whatever that app stores provide us and then maybe some like, download counts.

Researcher 07:23

Exactly. Yeah. But you don't find that raw number there motivational?

#4 07:30

No.

Researcher 07:33

No. Okay. Interesting. Let's see. Yeah. Maybe going a bit into the, the notion of usability. What, what is your perception of, of that concept as a whole, do you have, for instance, a set of terms that you've agreed upon in the project let's say the difference between usability and UX or something?

#4 07:59

No, we don't really. I've always thought that we've kind of lacked on that side of things. Having agreed on guidelines.

Researcher 08:19

Okay. Is that something you would normally use when you're developing on your own? Or do you have some external guides that you use?

#4 08:31

Nothing specifically. No. So accessibility is something that I try to keep in mind just in general. And then I have read some more general pieces on user experience and user interface design.

Researcher 08:57

Right. Do you ever Feel then if there's no guidelines or necessarily any, let's say authoritative agreements on, on what to be done and in case of the differences on, on these, usability decisions. Do you feel that subjectivity is a factor then that people have different perceptions off a usability?

#4 09:21

Yeah, definitely. And that's something that we don't always as a group agree upon. In that case, it just tends to either fall chooses a majority or at least the most vocal majority.

Researcher 09:44

Okay. So I was also thinking about, do you have any kind of systematic testing in terms of your usability to do tests just yourself? I'm guessing you're running the latest builds. How do you go about testing new usability changes?

#4 10:10

Yeah. So the only testing that I would do would be manual testing. Right now I haven't been running the latest. But I've been trying to look into some issues, in 10.6 In the latest stable. So I haven't been running master recently.

Researcher 10:28

Okay. Do you know if you keep track of the design rationale? So let's say the historic perspective of the design, like how you changed something and why changed at some point? Do you know if you have any, I noticed the, the blog posts are kind of doing that a little bit, but I don't know if you have any like a discussion thread or something that might be...

#4 10:52

It. It just depends. We don't have really a formal process for that. Sometimes it will be discussed, in GitHub issues. Or as you mentioned, a blog post, whenever after changes have been made. But there's nothing formal for that.

Researcher 11:15

Also, do you have any people contributing solely on UX or usability? So maybe not even code just maybe like prototypes or mock-ups or something?

#4 11:29

No, not that I'm aware of.

Researcher 11:32

Okay. Do you feel the project could, could use UX professionals or do you think it's doing fine without them?

#4 11:37

Oh, I definitely think that's an area we could use help with.

Researchern 11:43

Okay. It seems, at least it's a finding I keep encountering is that it seems it's, it's kind of a big barrier or at least for, for entering into FOSS projects, as a UX or UI professional. Do you know if you or the project has ever been approached by anyone that's wanted to contribute with that?

#4 12:06

I can only think of maybe one or two times. And that was probably earlier in the project. We probably weren't as well equipped to handle that type of...

Researcher 12:22

Okay. Yeah. You're more organized now you would say?

#4 12:26

right. Yeah.

Researcher 12:29

Okay. Why, why would you think that there is a barrier, especially for that field, if you have any thoughts on that?

#4 12:42

I think part of it may be just not knowing how to get started as far as where to go to offer help and what kind of help the project could use. It could just be not being aware that this is something that we could use help with at all. It's not something that we've necessarily put out there.

Researcher 13:14

No, but again, I think it might.. I think developers might be more, I don't know, technically inclined to also take part in these projects. So I definitely agree with you there. That it's probably also that.

Researcher 13:45

Can you actually just get me all, I'll talk me through how let's say you get us a future request or a suggestion of something. How would you map and, and really what, how do you approach what a user wants or is trying to do with, with Jellyfin? How would you like that whole process from issue being filled out on GitHub to you potentially solving it or doing something else with it?

#4 14:17

So we have a separate site for feature request. So those don't typically go through GitHub.

Researcher 14:28

Yeah. I saw that site.

#4 14:34

Fighter, which is another open source project. We use to manage the feature request. So as far as things coming in as feature requests, I occasionally look through the list and tag them or what client they correlate with. So if it's a specific request for iOS or Android or Android

TV or whatever it is just so we can keep track of where those need to go. And as far as feature requests go, that we're always kind of aware of the more popular things and working towards those just mainly in the background.

Researcher 15:36

Let's see.

#4 15:37

I guess, as far as planning, when features go into what release, a lot of it just comes down to what developers want to work on at a given time. And then for the apps I generally manage and GitHub they have a projects feature. So the next release or two releases we'll have a project and then can have issues assigned to them with different statuses as to what's being currently worked on and plan to be worked on for that specific release. But as far as the web server go, it's much more fluid and just developers working on what they decide to pick up.

Researcher 16:26

Okay. So it's very much like a "who wants to do what" need basically. Also, maybe something I should have asked, how much time would you say you're spending on Jellyfin just out of curiosity, let's say compared to your, your job, is it just whatever free time you have or?

#4 16:50

Yeah, it's just freedom.

Researcher 16:54

Let's see. We already talked a bit about GitHub and something that can be quite interesting to consider as it is if you feel that it's sufficient for all types of issues, did you find there are certain types of issues that are more complex to handle on GitHub, for instance, usability issue?

#4 17:25

It's an interesting question. So you already mentioned handling features separately. And a lot of that is just because it gets hard to maintain large numbers, of GitHub issues. I'm wondering if maybe UX issues would fall under that same situation where there could be a large number of them and it would be hard to maintain that standard backlog of actual bug reports.

Researcher 18:06

I'm also thinking that certain usability or UX issues might, you know, not necessarily be one issue in the code, it might be a larger rewrite of something or..

#4 18:23

Right. And it could go across projects. It could not just be in a single silo of code. And then I guess just submitting an issue on GitHub that can be a barrier. You have to have an app account, you have to know markdown. How do I get images in that? I'm guessing UX issues would to be image heavy. So that process probably isn't as easy for those types of issues.

Researcher 19:04

And now that we're on the topic, how would, how do you normally in the project manage some of these issues that might require larger quote rewrites? How do you, how do you approach such an issue?

#4 19:17

I'd say that's still a process we are working out. Sometimes we will set up projects within GitHub to handle larger issues, track changes for those. And those can be used to go over multiple repos. So if it's something that affects different clients and that can be tracked all through a single project in GitHub.

Researcher 19:52

Okay. I see. But, that's still something you're figuring out. So how are you figuring that out? Is it a, just an ongoing discussion on where stuff should go or?

#4 20:10

Yeah, we have been having internal discussions about how the best way to handle proposing larger changes and the bringing it on a path forward, implementing them, documenting that process. That's still something that we're discussing. We do have a repository set up GitHub. I don't think we've actually used it for anything yet, but the thought is that those kinds of issues could live there eventually.

Researcher 20:45

Okay. So like a, a project specifically only for like the larger goals of changing something or?

#4 20:54

Right. Just in order to be able to track that and the discussion around it.

Researcher 20:58

There's also some talk on the GitHub discussions mode, or tap that you could enable. Do you have any thoughts on that or is that related to, to what you just mentioned?

#4 21:13

So that, that is another thing that's came up right now. That's only available by invitation from GitHub. I think we were going to apply for it, but I haven't heard anything back yet.

Researcher 21:26

Okay. And do you, do you think that would, you know, give some value to the project?

#4 21:32

Yeah, I think that could definitely help.

Researcher 21:37

Do you know a, I haven't actually seen it. You say that it's invite only, so it I'm guessing it's an early access feature that's still being...

#4 21:47

It's still early access.

Researcher 21:52

Okay. I'll just have to look into that. That's interesting. Let's see here. Okay. So i have some questions also, we already kinda touched upon it a little bit, but some of the social aspects and some of the resource management of the project, do you ever take your work offline, I don't know if you're nearby any of the other project members, but in terms of like establishing a consensus, if something maybe is, you know, discussed and a consensus cannot be found. Do you ever do any offline work?

#4 22:31
Not really, we're pretty spread out. Yeah. So I guess we do have some internal channels on matrix. That's probably the closest thing to that

Researcher 22:47

Some like off topic channel or something.

#4 22:55

Yeah. It's just closed groups. So only team members can discuss channels.

Researcher 23:03

Okay. Do you have any other, let's say activities that can be considered channels that might not necessarily be channels? I mean like, do you guys ever play like video games or do you have any other way of discussing and meeting as a project?

#4 23:26

We haven't to date. Well, I guess a couple people did eat up video game a while ago. But we are planning on having some type of online hang out in October.

Researcher 23:54

Interesting. it is kind of a, something I touch a bit about upon because it, it is so different in projects and can be quite an interesting source for motivation as well. So that's, that's why, that's why I ask. Also a question I have. How much would you say if, you compare external and internal influence on the project. I can almost gather from what you're saying, that you would say the project is mostly influenced from the inside as like you, for instance, would like something to be changed and then you, you open a pull request or something and you change it all or an external user make some change. but how much would you say it's influenced from those two sides? Is it mostly internal and mostly external?

#4 24:53

That's kind of a hard question. I would say only mostly internal cause once an external user has made a significant contribution and they get invited to the team. So then at that point, I mean, they were external, but now they're internal.

Researcher 25:17

Okay. Okay. So it's also, let's say a merit based a little, a little bit.

#4 25:21

Right.

Researcher 25:23

You get a little more say if you have contributed some code rewrite or something. would you say the project then is democratic? Or I heard the term, I think it was Julian. Who, who called it a "do-ocracy"? If that's a correct. Do I don't know if you know, what, what I'm talking about specifically, but would you say that the fits the #4, the, the idea of a do-ocracy?

#4 25:59

Yeah. I'd say that's largely true. It's the only time that that isn't necessarily true is if there's some disagreement about a change. Right. And generally it's just, we reach a consensus from the internal team as to how we should proceed.

Researcher 26:23

Let's see here. Yeah. So what would you assign the greatest value in the project? If you had to select like one thing you could assign all the resources for right now, what, what would that be?

#4 26:45

Something i hadn't really considered.

Researcher 26:47

Yeah. It might be, it's fine taking a little time to think if you need that, but yeah. It can be, I dont know, general code quality or something else.

#4 27:02

So there's a lot of effort right now to fix some of the larger architectural issues within the app. It would be nice to kind of have a push through that and get to a point where we're happy with the architecture and to have a stable, solid, or release where we can kind of get to that point where we aren't making huge architectural changes.

Researcher 27:47

So it feels to me what you're saying there is that you're still in a place where it's kind of like maintenance. So getting to a place where you have a core That works and then you can build new features upon that.

#4 28:02

Yeah. Yeah. I think that's a good way of summarizing that. Yeah.

Researcher

Mmm. Let's see. Well, to talk about this a bit, so a bit about future improvement. So what I mean by future improvement is to, to inspire my thesis and my, hopefully some design that I'll create an in the end of the thesis here, the idea is that all these findings will go into some kind of, I don't know what to call it yet. A consumable format of some kind that FOSS project members can draw upon somehow. And to make some of these maybe a bit harder decisions on design and, and what to do and what not to do. And with that in mind, do you ever look up any academic sources or papers of books or videos or something on, something you were unsure off? And if you do, what format do you prefer to consume that knowledge in?

#4 29:15

Yeah, I do. A lot of that, I guess, would depend on the platform that I'm developing. So for Android and Android TV has resources on that. iOS have all has references that are available as well as UX references. Personally, I prefer things in a written form. I don't generally want to watch videos for things like that. Okay. I would prefer written form with visuals to help.

Researcher 29:59

And also, what would you say is the largest challenge in having strong usability in UX, in FOSS projects and how would you address the issue?

#4 30:13

I think a lot of it just comes down to needing a strong UX and design resource as far as guiding the project in a good direction that, that focuses on UX and design, as opposed to having many developers with their own sense of what is, or isn't good design. I don't know that might not be a popular...

Researcher 30:58

Oh, that's a, a really inspiring, so some kind of, let's say authoritative thing that you could, you know, direct someone to or it might also be, you know, a way to settle discussions, I guess.If some guidelines existed that you could agree upon.

#4 31:22

and could provide a source of reference when making changes.

Researcher 31:30

Okay. Let's see here. I'll just quickly run through my list here. I might have a, we talked a bit about it, but more specifically, would you say that you or the project employ any methods between contributors and end users that has a focus on usability? It might be a bit of a broad question, but what I mean by it, is do you have some specific activity that you do that has a focus on usability? So for instance, the announcements channel might be one such thing. Where you are interacting outwards to end users. Any initiatives like that, that I might have missed? or end users might not know about that you, that you're doing can also just be the blog, which we already talked about.

#4 **33:08**

No, I don't think there's really much else.

Researcher **33:13**

You also, you mentioned Reddit a little bit, maybe a couple of words on, how does that differ from, from let's say matrix or the are the same users that post a posting on Reddit? Are they also posting a matrix or why do you think that is?

#4 **33:33**

Generally very different groups of users. So if somebody is on Reddit, then that will be their first method of reaching out. And a lot of the team are pretty active on Reddit too. So there's a lot of interaction in that channel also.

Researcher **33:58**

Okay. Yeah. That's exactly what ... it is just interesting why those two kind of silos they excist and they provide something completely different, but still, since I can imagine from a, like, let's say logical standpoint, that everyone would just go straight to a matrix and just chat there, but it does seem there is quite a lot of activity on Reddit as well, discussions and stuff. I think we are all the way through as far as I can tell. Is it okay if I have some epiphany in the middle of the night where I realize what I've missed is it okay if I send you some follow up questions or something at some point? It's not necessarily something that's going to happen, but if I remember something that I, that I missed,

#4 **34:46**

yes, that's fine.

Researcher **34:50**

Great. And yeah, again, thank you so much for wanting to participate and yeah, the next time you will hear from me hopefully is when I have some concrete thesis writing done and potentially something to share with the project or any FOSS project, really. So, yeah.

#4 35:10

That's great. I look forward to seeing what you come up with and yeah. Good luck with your thesis.

Researcher 35:17

Yeah. Thanks man.That's hopefully it'll all turn out all right. Great. Yeah. Thanks. And I'll let you know if there's any follow up questions or if there's any other things I need you to approve. Okay, great. Have a nice day.

#4 35:36

You too. Bye.

10.6. Interview #5, Jellyfin

28 September 2020
17:29
#5
Good afternoon

Saw your request for a chat in Matrix the other day. If you don't mind, we will keep talking here, as it gives less problems, at least for me.

Thank you very much for considering me for your interview. It's a surprise for me because, although I have some contributions to the Jellyfin project, I think other collaborators did more for Jellyfin than me, so it's great to see that a little bit of my work didn't pass by 😊. Of course I want to take the interview. Hopefully I can give you the information and answers you need for your thesis.

In case you need other people aside from me, I invite you to consider other contributors like [redacted] (core leader), [redacted] Machiels (MrTimscampi on GH), Niels, Thornbill, [redacted] Robibero... I'm sure that most of the people in the Jellyfin project will agree 👍🏻.

Regarding how to do the interview: does it need to be recorded? Because if we can do it here, just chatting, it would be perfect, as I'm not comfortable enough with a recorded one.

We can do it whenever you want/can: I start university next week so this week is perfect if you have time. I also promise you I won't take so long to reply — had many chats piled up and I wanted to dedicate the time yours needed and a comprehensive explanation, instead of giving you a quick reply.

Thank you very much again for your offer! 😊
17:36
Researcher
Sounds perfect! Your work didnt pass by at all :) we can do text, thats fine with me. Will just have a look at my calendar and write you later today with a time/date for the interview. Huge thanks for wanting to participate!
18:17
#5
Sticker
Not included, change data exporting settings to download.
👍, 12.1 KB
19:22
Researcher
How does Tuesday sound?
21:17
Researcher
Otherwise wednes is also possible. After 15. What time zone are you in?

21:25
*wednesday
22:41
#5
GMT+1 CEST
22:41
I'm in Spain just in case
22:41
Wednesday sounds good
22:43
Researcher
Great! At 15 my time so that should be 14 for you right?
22:59
#5
What's yours timezone? You never said 😆
22:59
Better, which country are you in?
23:00
With this whole thing of summertime added to the timezones, I get really confused 😆
23:02
Researcher
Oh my bad 😂 im in Copenhagen so UTC +2 according to google. Yeah med to, so confusing. Its 23.01 here now if that makes it easier
23:03
Or gmt +2.. which os the same.. hmm confusing is putting it mildly 🤷‍♀️
23:08
Timezone confusion aside. Im interviewing Bill also wednesday at 16 my time. So how does 17.30 Sound? (Meaning 18.30 for me)
23:17
#5
In reply to this message
I confirm you we're on the same timezone, so great! No confusions!
23:18
In reply to this message
17.30 sounds good (now for you and for me 😁)
23:41
Researcher
Great!
30 September 2020
17:21
Researcher
Hey **#5**! Ready in 10 minutes time? :)
17:37
#5
Hey! Sorry for the delay!
17:37

Yes, I am!
17:37
Fully for you!
17:38
Got out of work late than expected 😅
17:39
Researcher
I know the feeling! No worries :)
17:40
#5
Thank you very much for your consideration 😁
17:40
Researcher
Yeah sure :).
First of all some formalities. You will of course be completely anonymous in the project and will receive a draft of the thesis (if you are interested) to verify that everything is in order. You can of course always reach me as well, if there is any concerns regarding something of that nature.
17:41
#5
Perfect 👍
17:42
Researcher
Great :) First, a little background. What is your day job? and how does that relate to when and how you got involved with FOSS in general, and the project?
18:03
#5
I'm still an student, so that's my primary focus. However, before classes start, I'm working in my spare time in my old primary school, doing a bit of network fixes and fixing computers, as they were overwhelmed by the quarantine and remote working stuff. I'm also doing volunteering here during my free time in a young association that it has, so they're helping me a bit with the savings.

Two years ago I started a degree of Network Systems Administration, and this year I start software engineer. I've always had a passion for computers, in general. Whether it's photoshopping or video editing, everything that involves working with a computer is enjoyable for me.

Although I was always aware that I wanted to study something related to computers, I was never sure if I would go for Networking, Hardware or software engineering. Things started to get clear for me as soon as I started developing for the first time: I was a fan of The Sims 3 game and, for me, the interaction between the characters and the vampires was lackluster. I consider myself an stubborn and constant person, so I was decided that I wanted to get that done. It ended being up one of the most used vampire mods with over 1000 downloads the last time I saw it. It was all thanks to FOSS honestly: I developed it with only 14 years and without taking any classes, just by inspecting code from others and seeing how they accomplished what I wanted.

This stubbornness translates to how I started contributing in Jellyfin: I've always wanted to make a personal media server for myself and I never liked Plex. Never felt open for me. I discovered Jellyfin by chance in Google. I loved it since second one. However, there were a lot of things that weren't great: images were trash quality, profile pictures looked bad in the header... I started with the intention to fix only a few things. However, after fixing something, something else appeared that I wanted to get sorted out. That situation repeated constantly until one day, Vasily spoke to me to ask me if I wanted to join them. And that's how we're today! Building a media server with an amazing team that teached me a lot of things.

In this context, I owe a lot to them and FOSS in general: Discovering that I had so much fun contributing to JF and pushing my code in GitHub It helped me realize that I wanted to go into software engineering.
18:04
(btw, I tend to talk a lot, so if you need more concise answers, please let me know) 😊
18:04
Researcher
No thats exactly the point!! Talk all you want :D
18:06
Very interesting journey! Where do you mostly contribute in the project? Which repositories and such?
18:24
#5
Although the mod I mentioned before was developed in C#, the same language that the backend part of Jellyfin uses, I found that most of the bugs were present in the clients. So I started with that and, to this day, is where I contribute the most. I only knew the basics of Javascript, however it turned out to be easy.

I also like to help in internationalization, triage (sorting issues, etc) and Reddit. Internationalization is one of the areas that I think we're failing and we are not sure how to accomplish. Jellyfin is starting to be really widespread, more than we're even aware of. An example: it's only been a few months since we decided to set up CDNs for our repo, so people in Europe, for instance, don't have to reach Canada (where [redacted], our leader, is based). The 10.6 update generated so much traffic that it was difficult to get more than 50 KB/s of download speed the day it was released

There's an insane amount of chinese, spanish, french people... And (at least in Spanish) the translations we have are really bad. I think internationalization might be one of the biggest problems that FOSS projects might face: it's really difficult to have a contributor base from all over the world. This leaves, inevitably, some "gaps" aroind the usability and experience of the software to a lot of people. Also, most of us have our servers set up in English (much easier for troubleshooting and sharing stuff with the team) so this makes matters worse, as we might miss some issues that our language/region has.

Right now we're using a contributor approach where everyone can help, without moderation. This is leading to vandalism in some languages. I'm currently elaborating docs and manuals and I'm thinking in ways to organise this mess with ideas such moderation people. However, it's very difficult and needs consensus.

As for the "future-past" roadmap, I want to play a bit with Chromecast client, our newly Vue and server plugins. I wanted to tinker a bit in all of these areas during this summer, but I disconnected like no year so they'll have to wait. Hopefully this remote working/studying stuff will allow me to do it!
18:29
Researcher
Very interesting points! You mentioned the overflow of new users upon release. Do you find that motivational as a project member? What about the social interactions on reddit? In general, what are some of the motivational factors in FOSS projects as you perceive it?
19:14
Researcher
You still there? 😁
19:38
#5
Yes, it really is. Jellyfin started as an small community that wanted a free and open media server. There is an small fraction of people that doesn't care about the community and open source part, only about the "it's free" mantra. However, that's maybe only 10% of the people. Discovering that there are many people who love and believe in what you contribute is a really amazing sensation.

About Reddit, personally, I have mixed feelings: in one hand, it's great because you see that a lot of people love the software and, in the other hand, most of the posts are bug reports, which means that there is still a lot to do. Anyway, even people who has bugs still show their appreciation and love for Jellyfin!

About the motivational factors, I think the biggest one for me it's the community. I did a lot of cosmetic fixes for the client for example. However, those fixes are nothing if it doesn't have a backend behind that answers to the client queries. I know nothing about the server. And the guys who develop the server know so little about the client.

We all have the same objective: however, each one of us don't have all the puzzle pieces. However, if we all join, we make a nice little puzzle.
19:40
In reply to this message
Yes, sorry, received a family call I couldn't dismiss. I will also start with some house chores, so I might take some time to reply.
19:40
I don't mind if we keep going, just that maybe I can't answer right away
19:40
If that's a problem for you, we can continue tomorrow
19:40
For me, no problem 👍

19:41
So no worries!
19:42
Sorry for all the hassle, today is being an unexpected busy day
19:42
Researcher
No worries! Also, if you want to continue tomorrow or something we can do that as well. No need to rush :)
Very interesting none the less.
On that note. Do you find user numbers themselves motivational? I mean if you have any. I can imagine numbers like docker pulls or app store downloads could be something you would find impressive as a creator
19:42
Yeah totally understandable! Again, if you want to continue tomorrow or some other time its totally fine!
19:43
And what about the ideological side of the equation? Are you more of a pragmatist or are you also motivated by the FOSS mentality/ideology?
23:37
#5
I find them motivational but I don't think is the fuel that keep me working on our project. We do this for fun, for the learning experience, because we use it... We have many reasons among the team. I see the numbers more as a reason to do a more quality job: this is no longer the software that a dozen people use, so we need to make sure that what we do is well-thought and tested. Of course bugs will always slip through but, in my opinion, this is something we've been improving.

Quick example: before, Jellyfin had hardcoded sizes for scaling the images that you see while browsing the library so, for example, if the placeholder for a movie poster had a size of 1000x1500 on screen, the rendered image inside would be of 500x750 (for example, made up numbers). This looked terribly and I really hated it, so my second pull request to the project was precisely for removing those hardcoded limits. This meant that the server would return the original poster set by the user or data provider. This turned out to be great for UI and it worked in my devices perfectly. However, TVs broke with that. Some providers/users sets images with sizes as high as 5000x7XXX: loading 100 images with those sizes in memory was overkill for low-specced devices. It also meant bandwidth usage as high as 100 MB for some people, just for the images.

Back in the day, that PR took only one day. Don't remember now who reviewed it exactly but it was like "heh, it's great, let's merge it". This was something that wasn't bad back then, but today it's completely crazy.

About the ideological side, I position myself in the middle. I love closed source software and I love open source software. All my computers run Windows and I don't see myself using another OS as my daily driver. But I also have always open a VM or bash for Windows to use Linux. For servers, IoT, learning, etc... Linux has no rivals. UX, Windows is the winner in my opinion.

I personally think this whole war between open source software and closed source software is pointless: both of them can coexist perfectly. It's understandable that companies want to do business in something that they invest a lot of money. It's understandable that the community wants to get involved in projects that aren't profitable enough for big corporations. Google, RedHat, Intel and Microsoft earns a lot of money from Linux. Linux earns a lot of money from them as well. They're not enemies. My Windows PC wouldn't be as useful as it's right now without my router, which runs Linux. Computers in general wouldn't be as popular as it is today without Windows. Linux based distros earned a lot of users thanks to the missteps that Microsoft did through the history. That's something that many people argues about on the internet: Windows is bad, Linux is just better. If that is 100% accurate, why Windows is still dominating the market? Isn't true that Linux distros still need to make a lot of work to attract the people who don't know nothing about computers? At the end of the day, everything it's about freedom of choice, which seems to contradictory, given that closed source "limits" your freedom. In my opinion, it isn't: you're always free to choose what's best for you.
23:39
In reply to this message
100% honestly, I have no problems! This is the good thing about chatting, you can drop the questions whenever you can and I can drop my answers whenever I can as well, so you can focus in what to ask me and I can focus in giving you my best possible answer! 😁
23:40
I'm being so clear to you about this because I don't want you to think that I'm doing this quickly and without effort. Actually it's the other way around, lets me think even more about your question. This last answer was a product of all my thoughts while doing my household chores 😊
23:43
For your interest anyway, tomorrow morning I'll be mostly free, if nothing else shows up like today. I'll probably go at 8 am to buy a new chair for my desktop and later, at 13 more less, I'll probably do a bit of exercise. During that time I can answer you rightaway
23:45
But again, maybe I'm too repetitive with this, but don't feel bad about you "taking my time" or something like that, I'm really enjoying this interview and it's really nice you're bringing up these questions for me to think about. We can extend this as much as you want, even if you have questions that aren't for your interview. I'll happily answer.
23:47
Researcher
Sounds perfect. I will drop you some more questions during tomorrow then :) your answers are great, keep them coming 😁
1 October 2020
12:32
Researcher
Hey **#5**. Here is the continuation from yesterdays questions for whenever you have the time :)

Also related to my question on motivational factors, and the amount of users you have. What about feedback from outside the project? Do you find there is enough feedback? Is it generally motivational or is it more of a burden?
13:02
#5
Feedback is being great. We know that we're miles behind Plex or Emby in features, clients, reliability... However, people still love the development pace and the fact that they're fully under the control of what is running on their own servers. As a FOSS project, I think we have a really good advantage: Plex and Emby are businesses that need to take special care about what they introduce, how it's done... Business, by nature, are risk-averse. They can't tolerate the same amount of bugs that our software still ships while we're cleaning the old codebase, for instance. That's also why we had features like SyncPlay that appeared before major players started playing with similar stuff.

Users in general will be more forgivable with us, as they generally know that we do this on our free time and that we don't earn anything from this. We receive a lot of feedback in Reddit with titles like "Just wanted to thank the Jellyfin devs for what they do". It might seem something silly, but the fact that somebody is actually spending their time in making a thread just for saying nothing but thank you means a lot. That's something very difficult to see in commercial projects.

However, as this is the internet I guess, you'll always find some hostile people. Some people demands that certain issues must be fixes, without even caring to find in GitHub issues to realise that their issue is already being worked on. We end up knowing that some of those people are running and reselling Netflix-like services that are powered by Jellyfin, and they expect to get scability for thousands from a software designed to home users. They're free to do what they want, it's free software, but sometimes it's a bit frustrating to see people earning a lot of money and complaining so hard when we don't earn anything.

That's the worst thing we found so far. But generally, I think we have a really good and positive community where feedback is constructive and good to read. Personally, and I think I said this many times through this interview, is one of the aspects I love the most about this project. Because it doesn't matter if you don't like X or Y: if you Express those in a constructive way, everybody will always be positive with you. We actually accomplished a lot of changes in Jellyfin motivated by a lot of constructive criticism from the people.
13:17
Researcher
Wow i had no idea that people are exploiting it like that! But good to hear that its predominantly positive feedback!

Now.. Something a bit more focused on usability/UX. What is your perception of usability? What terms do you use in the project? UX/Usability? Is there a difference to you? What do they entail?
Is it a trivial task? Is it very important? And do you feel that subjectivity is a factor in usability decisions when it comes to creating consensus when something cannot be agreed upon in the project?
14:26

#5
This is a really complex topic. In my opinion, something has good usability when it works and behaves exactly how you expect it to be in the bare minimum. Small example: the music player UI we had before didn't have shuffle and repeat buttons in the mobile interface. Those two functions are present in probably every music player out there. That's bad usability, because something that should be expected to be there isn't nowhere to be seen.

Usability is one of the key roles that compounds UX, but it's not alone. Following the prior example, we could solve the usability problems just by adding repeat and shuffling features, right? However, we need to present those options in a friendly manner, so they're quickly accesible and don't prevent the user from distracting to the main topic of the interface: the playing track in this case.

(Continues after the pictures...)
14:26

Before
14:27

After
14:38
UX is a very important (but difficult thing) to accomplish in every software. The music player redesign, for example, was very well welcomed. However, some users preferred the old way of always displaying the queue, for example, as they needed another tap to manage the playlist. That's why UX is so subjective.

When there are different opinions among us, we generally follow two mantras:

A) How do alternatives/similar software accomplish this? Normally, all music players put the queue behind a button and leave the main player UI for the album cover and the controls only. That's why we chose that behaviour. This way of working is what I also want to implement in how we translate Jellyfin: we use a wording that it's not common in other services. Using familiar terminology always eases the learning curve and the "familiarity" of the software.

B) The contributor who is proposing the changes is the one who choses how to do it. This mostly happens when we see that Netflix, Spotify, Amazon Prime, Plex, etc etc... Do the same thing in a completely different way and there are a lot of different opinions among us and the users. If the contributor who proposed the changes decided and spend his/her time in designing it that way it's for something, so we follow that. As long as it gets two approving reviews and it's something that it's not inherintly bad and we're neutral about it, it's good
15:01
Researcher
Very interesting points. So you then dont have any UX/Usability guidelines in the project, someting external?
Do you normally use any external tools/methods for desiging?
Do you, with regards to design, employ systematic testing on any users or how do you test?

With regards to the screenshots you provided, do you keep track of design rationale (history of the design, and why it changed)? Or is that part of your own design process?
16:46

#5
We should but we don't. Until now that some contributors started to rewrite the web client in Vue, most of the new design challenges have been small and, as explained before, followed the logic of seeing what others do or doing what the contributor who started the redesign thinks. Unlike the new client, where we're building everything from the ground up, the old client has still all the design aesthetics that we inherited from Emby. We did small touches to parts of the UI, yes (like item details and music player) but none of them were big enough to justify the need of a big design process: most of them were sorted out just by looking at the practicity of stuff and common sense. Then, we exchange screenshots of how it looks like and everyone says what he likes or dislikes.

The good thing about being FOSS is that everyone is able to see what we're working on: you just need to head over the Pull Request tab in GitHub to see what's inside the oven. Aside from the internal discussion, in the pull requests we also receive some feedback from users. When everything is settled up and merged, a wider user base who uses unstables see the changes right away and can also provide feedback. It's rare that we receive feedback on design choices when when we're early in the development cycle but, before every major release, we do a call out for testing unstables so as many people as possible test and give feedback. That's the last step where we can polish things before release. Once it's release in stable, of course the feedback gets bigger. We then use minor versions (10.6.1, 10.6.2...) to sort out the minor issues. Major design or changes will never occur between minor versions.

I haven't done so many big design decisions aside from the music player but, in general, I would say that everyone that does something remembers why he did that and the reasons behind it. So we don't keep a track of it anywhere, if that's your question.

And about the external tools for design, we don't. We would like to have some people in the organization that knows about design, but we don't unfortunately
16:52
Researcher
That leads onto my next question. Do you have any people contributing solely on UX/Usability?
Is there any resources allocated only for UX?
Are there any UX professionals in the project?
Do you feel the project could use UX pro's? Or is it doing fine without them?
Have you ever been approached by any UX pro's? If so what happened?
3 October 2020
08:30
#5
No, we don't. We should but it's as simple that nobody wants to contribute solely to that purpose. Each of us try to do a bit of everything.

No. We only have contributors who either dev directly or others who do support and triage.

No, unfortunately.

I think we need them, and even more now that we're writing a new client from scratch which needs more decisions and thinking than the small redesigns we did in the old one. We're still writing the "backend" components of the UI and build utils, but probably all the form of UI design we do will be all based in experiences from other services or common sense, as we have been doing. Which is lame, but those are the resources we have.

We didn't, the only case I can think of is when someone designed a few icons for Gelli, the third party music client that one of our main contributors develop. But nothing else.
09:59
Researcher
Understood. It seems that very few UX pro's are engaged in FOSS projects. Why do you think that is?
16:51
#5
I'm not really sure. I guess it's because it's a very time-consuming thing to do if you're not getting paid.

Businesses pay a lot for graphic designers for UIs and such and I think they're more scarce than developers as well. That implies that there are less "potential" people who can get into this.

Also, the good thing about the people who write the code itself is that, if we see a design we like, we can make it real easily. We don't mind spending our time doing that because it's something that will be useful, not only for the community but for us, as we're users as well of the software. A designer who doesn't write code might see him/herself in a weak position because of that. They might use the software as we do, but the final choice of how to implement X feature or Y design is on the people who actually write the code. And this can be obviously frustrating.

Businesses like Facebook, Google, etc... have a more well-defined structure and a wide range of developers with a wide range of skills at their disposals. If design guys agree in one design and project manager do as well, devs must make it real. FOSS projects are, in that sense, "homemade". Some good design choices by UX experts might be unrealistic just because contributors don't have the time/experience/willing to work on that.

That said, this is based purely on my speculation and in my experience in Jellyfin, which is a rather small community compared to the people behind Linux, Gnome, Gimp, etc... Also, (and this is a "call" to UX designers who might read this) I explained what UI/UX designers might see designing for FOSS might be like. I don't think nobody, at least in Jellyfin, will ever feel looked down at their ideas. Every hand in FOSS it's helpful.
16:52
(sorry for the gif I sent before, gboard slip!)
19:32
Researcher

No worries on the GIF :) Yeah those points all really makes sense.

Now that you mentioned being able to write code, and that being a determining factor in FOSS projects...

Do you feel that github is sufficient for all types of issues?
Are usability issues especially hard to keep track off in terms of complexity?
4 October 2020
17:13
Researcher
Also, might be easier for you if i send you the remaining questions. Here they are (if you feel you already replied to some of them, just let them be. The structure is being thrown around a little in unstructured interviews anyway). I will let you know if i have any follow up questions based on your replies :)

Do you ever take your work offline? Do you have social activities in the project?
How are decisions made? Is consensus required?
Is the project totally democratic or are there certain people with more pull than others?

Do you find that social bonds are important in your work in the project? For instance social bonds between contributors or between devs and end users?
Do you have certain users that are very active in testing and providing feedback with the most recent code, so called "core users"?
How much change in the project would you say is driven or influenced by completely external users, compared to the internal influence from the project members?

What do you assign the greatest value in the project, and what would you assign the most resources for if you could decide what should be fixed here and now?
Do you spend time educating other project members on for instance UX, code or something else?

How do you manage resource allocation? Is it first come first served or do you have roles. If so, who distributes work to whom?
Do you ever look up any academic sources/papers/books on how to manage something you unsure of? With regards to the many roles you/contributors take in the project?

What would you say is the largest challenge in having strong usability/UX in FOSS projects? and how do would you address it?
17:28
#5
Great, I'll try to answer them this noon!
17:32
Researcher
Sure! ✌
5 October 2020
12:30
#5

In reply to this message
I think it is. We don't want to migrate to another platform, although some people were concerned at first about the Microsoft purchase, but so far, they like what they're doing with it.

The main problem we have with GitHub is the lack of granular permissions: you can be assigned admin, reviewer, triage, etc etc perms in a repo, but you can't customize those teams' permissions. For instance, I'm a member of triage but I can't move issues across repositories in the organization because we can't enable that permission without being full admins in the destination repo.

Also, with the implementation of GitHub discussions, everything can be centralized in one place, which is helpful.
7 October 2020
15:49
Researcher
Right. Sounds like it working out for you then. Any thoughts on the final questions? :)
15:59
#5
Do you ever take you work offline?

Not sure what do you mean, but if it's about having some kind of "holidays", yes, this whole summer I've been completely out without coding. A little bit active in our private chats and Reddit, to always be aware of what was going on, but 0 contributions during that period.

How are decisions made?

Completely democratic. Majority wins. Although we always try to discuss every opinion given so many times happened that, if a minority thought of something but it was convincing enough for the majority, that minority's opinion ends up implementing.

Do you find that social bonds are important in your work in the project?

Absolutely. Working with toxic contributors or users is like working in a toxic office, and I'm really glad that we don't have that there. Social bonds is what makes you realise that.

Do you have certain users that are very active in testing and providing feedback with the most recent code, so called "core users"?

As I mentioned many times through the interview, the time when we receive huge amounts of feedback is when we're nearing the release. At those times we have a lot of users who keep reporting bugs and stuff that needs improving. However, I don't remember having a fixed "team" of those people. In 10.5, one set of users contributed to giving feedback. In 10.6, another set did. We have them in our chats and so on, but not everybody participates with the same "intensity" through the course of time. Same applies to contributors.

How much change in the project would you say is driven or influenced by completely external users, compared to the internal influence from the project members?

I would say that everything is influenced by external people. Most of the stuff that gets implemented was mentioned or suggested by someone else before. Only the approaches or the way to do it is mostly decided by the contributor who wants to make it real.

Although we do it publicily, framework changes discussions (like the one we had for Vue), migrations in the codebase, etc etc... Tend to be contributor-only, although everyone can participate. Probably because people outside the project isn't as knowledgeable of the codebase as us, contributors, are.

How do you manage resource allocation?

Completely free. If during a period of time I want to work on clients we talk with [redacted] or [redacted], owners of the repo and they add us to the relevant repositories. So everybody does what he/she feels to do, no obligations.

What would you say is the largest challenge in having strong usability/UX in FOSS projects?

Inconsistency and lack of resources, as I mentioned in my previous answer of the graphic designer. The fact that FOSS is more "homemade" instead of commercially-driven, makes this a problem, as commercially-driven software must have good UI and UX. Bad UI/UX in those cases can be deadly for one app/project, as it can lead to bad sales and less profits.

That's why FOSS projects aren't that risk-averse or worried about losing users just because UX.

16:00
I omitted the questions that I think I already addressed, as you told me

16:00
Review everything and if you see that something else is missing, please let me know

16:12
Researcher
Read through the whole thing! Thank you so much for the very detailed answers and input. I think we got around all that i had planned anyway. If i realize something that i haven missed i will let you know down the line 👌
Again, huge thanks for wanting to participate

10.7. Interview #6, Jellyfin

Researcher 00:01
Yeah. And that should be going. Let's see. Yes. Great. Okay. So first of all, a little background, what is your day job? And then what do you do for a living?

#6 00:15
I've just started university going on my first year. Outside of that, I worked as a lifeguard and swimming teacher. Not really stuff related to Jellyfin.

Researcher 00:25
No, no, no, that's totally fine. So, how does that relate to how you are involved in Jellyfin and FOSS projects in general? How, does that come about?

#6 00:38
Yeah, so I guess sort of like beginning this year, I was using Plex and then I got a bit tired of Plex not being open source media software, Plex having ads and not showing me the media that I owned directly. It was showing unrelated stuff, podcasts, other stuff like that. And I didn't want to see that. I just wanted my media to be accessible through my phone, tablet, or whatever. I found out about Jellyfin started using that really liked it. And since then I've been contributing I've me and another guy we'd been doing the majority of the work in the new Vue client. And we hope to release maybe beginning of next year.

Researcher 01:26
Yeah. I just saw the, I saw the announcement on the channel there pop up a while back. Yeah. Okay. So what I can gather from that... are you also then ideologically driven in that sense? I mean you mentioned a Plex not being open and, and not really tailored to your needs specifically, or are you more of a pragmatist in that sense?

#6 01:49
I guess I just wanted the best possible experience for, for anyone to have, whether that be, you know, consuming their media and not have to pay a fee. And the Plex features such as downloading media to use, and the phone was you have to pay for that. And that's, I didn't really want that. I just wanted it to be free and open. Anyone can use it if they want something slightly different, they can, they're free to go and modify that, how they see fit.

Researcher 02:19
So both really in the, in that sense. Yeah. So a bit about the motivation for developing and contributing in a FOSS project, what would you, what would you say is your primary motivation for, for contributing in the project?

#6 02:37
I guess like, before i ever was in like secondary school and when I was probably about 15. I did computer science as one of my courses. I find that really interesting, did quite a lot of python and a little bit of web development, but yeah, since then that's kind of taken a back seat and then, so at the end of my level I took a gap year just before I went to university. And through that, I was like, I learned a skill. So I started to learn some web development. So all of this stuff i have been contributing, is self taught, using some courses online. I've got no formal, like education as to software engineering or anything like that.

Researcher 03:30
Okay. So you're very much a, let's say kind of scratching an itch a little bit and developing your own skills would be your motivation then

#6 **03:38**
Yeah. That's right.

Researcher **03:39**
Okay. What about something like the social rewards, maybe thats also a part of it, like communicating with the other project members or even end users, do you find that motivational at all?

#6 03:52
Yeah, i feel like since i have joined matrix, like you see in the chat every once in a while, someone's, you know, really appreciates the work we've done on it, that feels really good. And, so after, I don't know, maybe a month or two contributing, and then I got invited into, you know, to become one of the members and do stuff like reviewing requests and doing stuff about that. That made me feel really good about myself. Like I've been contribution to a point where they feel like I'm a valued member of the team. That felt really good. So, yeah.

Researcher 04:29
Yeah. Getting, getting acknowledged for the work you've done. Yeah. Totally makes sense. That that would be a nice pad on the shoulder. Up that alley as well. I mentioned the end users before. What is, what is your take on feedback? You also mentioned reviewing some of the issue, issue that's been opened on GitHub. Is it mostly negative or positive or frustrating, or like what, what are your thoughts on that? It can also be a motivational factor, I guess, you know, interacting with some of the users of Jellyfin.

#6 05:05
Yes. So fixing stuff like issues, it's a bit annoying when you, when you're trying to keep the issues on github, just to actually <inaudible> bugs and we have a separate, resource for requesting features. It can be a little bit frustrating when you've got a bunch of what you think are issues, but are actually feature requests in the github issues tab. But yeah. So if someone has an issue trying to solve it, and I find that quite rewarding, you get to

see them be happy at the end of it, that you've managed to solve that problem. They're more happier that they've got know more successful media setup that they're using.

Researcher 05:51
So do you ever, do you ever have any issues that are maybe not technically elaborated enough for you to help? Or how do you, how do you deal with that? Normally, if, if you're encountering some, let's say less technically inclined users?

#6 06:07
Yeah. So it's just, I'll just try and go through normal things and ask what software they're using whether they've got the right permissions, to see whether there's any logs and to see whether we can identify any errors that, are the core of the issue. And I'll ask questions like that to help try and sort of narrow down what the problem is.

Researcher 06:28
Okay. A bit up that alley as well, so regarding the users, do you find it motivational at all to look at some of the download numbers? Or I don't know how many metrics exactly you have for measuring this kind of stuff. I guess the app stores would be an obvious way of looking at that how many downloads you have or something, do you find those raw numbers motivational?

#6 06:54
I haven't really looked at those raw numbers that much. I don't know, on Docker hub, we've got something ridiculous. Like a hundred million.

Researcher 07:01
Yeah. I saw that. I was like, okay, that's kind of impressive. At least from an objective standpoint, right? Yeah.

#607:10
We know that thats definitely not the raw user number..

Researcher 07:16
Yeah. Everyone that's ever tried it on Docker. Yeah. Okay, great.

#6 07:20
It does feel quite impressive, but yeah, I feel like I don't really look at those numbers too much. It's more of, well, if I work on it and then I can see the work that I've done when I'm using Jellyfin that feels quite like good.

Researcher 07:38
Okay. Let's see. So a bit more pertaining to usability specifically. What is your perception of usability? Do you have any like terms or, let's say even a terminology you have

agreed upon in the projects? Really like what's your general perception of usability or UX and are these terms even the same to you? Or how, how do you go about?

#6 08:06
Yeah, obviously I guess, cause I'm, I'm self taught really. I don't really look much into usability, but for stuff about the new, Vue client, we are looking to follow some like material design guidelines to keep it reasonable and understandable for the user and stuff like the text. If you have a setting that's complicated, making sure that the description, the setting is really easy to understand. So I guess my sort of understanding of that is maybe having a system that makes sense it's easy to use and you're not having to search to find what you're looking for in the software.

Researcher 08:47
Right. So kind of a, like a first glance what's logical kind of logic to it then.

#6 08:53
Yeah. That's right.

Researcher 08:54
Okay. Also with that in mind, do you find it a trivial task? Is it something you find very important? How do you, how do you judge what's the most important to you in your work?

#6 09:12
I think it's quite important. I guess it was quite difficult on the old client to make a significant change that really improved usability, but, and in the new client definitely we are going to be thinking really hard about what's going to be the best decision. So every step in every design decision we make so that when it comes, to the users can come on and use it straight away, them not having to say, Oh, what does this button do? Oh, I don't understand stuff like that. They can immediately log on for the first time and understand exactly what they need to do.

Researcher 09:52
Right. Avoid that initial flood of questions on Reddit or something. Also with that in mind, do you feel that subjectivity is ever an issue? I can imagine, you know, part of usability is also subjectivity. One could argue, at least. Do you find that ever to be an issue in let's say finding a common ground on some problem in the project, or how do you deal with, let's say someone having a different opinion on a usability issue?

#6 10:26
I guess. Yeah. It can be difficult to find out, you know, what does the community think is the best idea? And one of the things we've debated doing internally is for decisions like the item details page, which you could take a number of different approaches to putting

a poll out or something on GitHub saying "what do you guys think is best? We've got these five options. What do you like the best?" But yeah, because obviously you can have, you know, people that just turn up, make one pull request and then leave, you know, people aren't going to be having the experience of like the pre.. Everything that's going on previously in, in the project. So I guess by discussing the issues and if, like there's a bit of friction, you know, getting someone else's opinion about it can really help to make sure your, you know, you're doing what's best, what everyone thinks is best. Not, you know, if you're the outlier, what you think is best.

Researcher 11:34
Right. Okay. So basically just involving more people until it kinda makes sense, or at least there's some kind of

#6 11:41
A group decision and what's, what's the best design point. Yeah.

Researcher 11:48
Do you ever, I can imagine the roles also are a bit blurred here, do you ever let's say involve any of the backend developers on usability decisions or something?

#6 12:01
Yeah. So I'd say, definitely like internally we'll, you know, someone might post a screenshot of what they're working on and, you know, just ask for people's opinions. I know someone did i think yesterday or a couple of days ago. Cause even though they're not doing that the front end day to day, they can, they still use, use the product. They're still gonna know that if they still are going to know what's a good design and what's, and what's not as a good decision.

Researcher 12:30
Right, right. Okay. Do you then have any UX or usability guidelines in the project you follow? You mentioned the material design guidelines so that they can serve kind of as an external guideline I can imagine. Do you have anything else that you, you specifically look at when you're developing?

#6 12:54
I don't think so. No. No. We do... We for, for the Vue client we are using Veautify, which has a bunch of premade effectively like templates that you can customize, however you like. So using that as like a baseline that already gives us a pretty good design from a usability standpoint designed to build upon.

Researcher 13:23
Okay. So they, they already relatively a good, best practice minded. I can imagine those templates then.

#6 13:31

Yeah. It's yeah.

Researcher 13:33
Right. Okay. Then with, with regards to testing, do you have any usability testing or how do you manage testing new changes? Do you then just run the version yourself for a while? Or how do you, how do you do that?

#6 13:50
So like I'm always running the unstable version of Jellyfin on my home, on my home server. So anytime I'm interacting with a client, it's always going to be the latest, the latest version. So if, if I come along, come across something that I don't think is quite right. I can always go back, ask other people's opinions and then make a change if other people agree with me. Right. Yeah. We don't do any like specific testing. It's more just, you know, me using our home and then, you know, seeing what I think about it, any changes or not.

Researcher 14:27
Okay. So primarily the changes are, would you say internally driven from your own use in the project? Right. And then there is of course the external issues and features and requests and all that stuff.

#6 14:42
Yeah. So one of the things we sometimes take inspiration of is the, I don't know whether you've seen or read it. We've got a couple of people who do, who do CSS files that you can import to the Jellyfin server and then it can basically completely redoes the whole design of the whole experience.

Researcher 15:07
Yeah. I saw some of the, some of the themes are yeah. That was being uploaded. Okay.

#6 15:10
Yeah. So that's one of the, that's one of the things, if, you know, there's a thing that people really like where you can, you know, take inspiration from that, right. When we're, you know, developing the next version

Researcher 15:22
Are some of those themes made by you or the project members or are they completely external to the project in that sense?

#6 15:31
I think most of the people that do it are just external to the project.

Researcher 15:42

So it's just for themselves just making a CSS file too, to change the layout or the look rather.

#6 15:43
Yeah. And then they basically gone and said, Oh, I think this is pretty good. I'll put it on, GitHub. So, you know, so if someone wants to use it, they can, it's really easily accessible for them to use.

Researcher 15:55
Different question here. Do you ever keep track of the design rationale? So let's say the historical changes of the design, it can be you or the project in general. So let's say you change a place like the placement of some button or something and you then write down why that was changed or?

#6 16:19
I dont think so if, if we, like, if I was to completely redesign a page and I think when I made the pull requests, I would say we've done the I've made these changes because.. It's in a more appropriate place. Or I think that the button is more suited to this location..

Researcher 16:37
And that, and that information would live in matrix then?

#6 **16:41**
Probably GitHub.

Researcher **16:42**
Okay. So that's just part of the pull request then. Okay. Do you know if you have any people in the project that are contributing solely on UX or usability?

#6 **17:01**
I dont think so.

Researcher **17:02**
Okay. That makes sense. Do you, do you feel the project could use like usability professionals that doesn't even, excuse me, doesn't even provide code necessarily. It could just be like a wire frames or mock ops or something, or is it...?

#6 17:22
I think sometimes that could be used to like, you know, the other day, as I mentioned earlier, someone posted a picture of one of the pages and they were like, "what do you guys think of this?" And some people posted a response and they were like, "yeah, that's nice". And then at the end of it, we were like, we all thought it looked nice, but would a UX professional be able to provide a better insight as to, to make it even more

accessible and like have more sensible design choices for everyone. And so I guess it would be useful. Yeah.

Researcher 17:57
I don't know if you know specifically, but if you have ever the project or you have you been approached by any UX professionals and if so, what happened?

#6 18:05
I don't think we have.

Researcher 18:10
Okay. It is, it is quite quite an interesting thing. I keep finding that very few projects have actually even been approached by any professionals. And it seems like it kind of is a big barrier of entry or there is a big barrier of entry into FOSS projects, maybe for usability professionals. It's kind of an interesting thing.

#6 18:32
I guess what would be really good is if one of the, one of the things that's been done on the Vue project is having an initial discussion about, the design decisions that they're going to make. So I think it's all in Jellyfin Vue, the metadata editor and one of the issues someone's posting, they said, "this is my initial design. What do you guys think?"

Researcher 19:00
Right. Yeah. I saw, I saw that exact post. It was really interesting actually.

#6 19:05
I think doing more of that for every single, you know, design decision we make, it could be really useful in getting UX designers in to give their opinion on it. Which is something I've never really thought about before, but yeah, that seems like it would be a good idea.

Researcher 19:24
So yeah. More discussion basically as would be a good.. yeah. Let's see. Where did we get to? Yeah. Yeah. Okay. So do, do you, I don't know if you know, and this is kind of a broad question. Keep that in mind, do you employ any methods in the project between contributors and end users with a focus on usability? This can be a blog or it could be AB testing or something completely different. For instance, I don't know if you know, Blender, the 3D suite they have, for instance, these open movies where it's like the project and then end users kind of working together to reach some common goal, being a movie in this case, do you have any, like, it can be totally anything here. Do you have like a monthly Reddit post or something?

#6 20:18

I dont think so. So we do sort of posts like before every, before every major release saying, inviting testers to come on and test. But I don't think we have any articulate methods that we use.

Researcher 20:32
Okay. It is also kind of a unique thing they have there in the Blender community, also a really huge community of people, of course, but still, it's kind of an interesting test off their own project, and product in that sense that always generates some new requests for usability and stuff and is really interesting. They are the only project so far that I found that has anything like that. Structured events in the, in that sort of way. I don't know what it would be in the Jellyfin sphere, but I can imagine like a viewing party for, for everyone or something.

#6 21:18
Maybe have like a monthly Reddit posts basically summarizing the work that had been done in that month. And then people comment say, I really like this, but what about changing, changing something else in that way to improve the usability and something like that. That might be a quite good idea.

Researcher 21:41
So also maybe, yeah, it gets, it gets some of the passive users on board for reporting as well. Or do you, I don't know if you find that there are many passive users?

#6 21:54
I'm not too sure. Actually.

Researcher 21:59
Again, it's not even necessarily a measurement of success for you. If, as you said, the, your primary motivation might even for some people in the project, just be your own usage. So in that sense, it's kind of a complex idea. How much should users be involved. It kind of becomes a balancing act. I think. Let's see. Okay. So, but specifically on GitHub, do you feel that it's sufficient for all types of issues? You mentioned already that you have a feature requests on a separate platform, and I can imagine there's a reason for that. Are certain issues more complex to keep track of, or what's, what's your thoughts on, on that?

#6 22:48
I guess some issues, it can be quite difficult to, to diagnose if there's not enough information that you've been given. I saw someone was asking about a couple of days ago, probably a week ago now by this point. And he was saying he wants he wanted it to be an option to be able to hide part of the user interface. And I was saying "which parts specifically" he wants to hide. So I felt like there wasn't enough information in following

that, I don't think the person has replied. So I can't really, you know, I can't go on and, you know, potentially implement that, which can be a bit frustrating. Yeah.

Researcher
What about for instance... Oh, did I interrupt? I can also imagine that certain usability issues might be hard to keep track of in terms of complexity. Like, you know, a pretty simple on the surface at least issue might be, let's say a complete overhaul of some flow or between certain things that in the code is actually quite a lot of rewriting. How do you, how do you manage usability bugs that are simply too large or might require rewrites that are completely unreasonable. Do you just close them? Or do you explain it or how do you..?

#6 24:17
I dont think we've had any like really, really huge, so that's, that's I guess a bit of a plus, but hopefully using the Vue framework, it's going to be easier to, to, you know, to improve on, on something because previously it was complete nightmare to, if you were trying to, if you're a first time contributor trying to dive into the code and it's a complete nightmare to try and find what you're looking for and work out how something is made, unless you will, unless you've previously spent maybe a month looking at the code and testing, finding stuff out about it. Yeah. That quite frustrating. And yeah, we don't get too many requests, like huge rewrites of part of the,

Researcher 25:08
And then just a, I'm not the, an expert by any means on, on programming.. Vue is that similar at all to like how React works? Is it more like a modular system and that sense or why would it make it easier to..?

#6 25:25
Yeah, it's basically like basically a different framework to react. It's basically basically the same thing.

Researcher 25:33
I just, I read about it and I've kind of, to me, it just kind of sounded like another React type thing, kind of having a modular pieces of, yeah. Okay.

#6 25:45
It's, you know, you've got Vue, React, Angular and in a community post probably half a year ago. Something like that, we said that we wanted to move the code base to a more recent, more up to date code base. We want to choose a framework what was going to be was going to be React, Vue, Angular. There were a couple of others I can't quite remember. And in the end community chose view these to go with. So yeah. That's why we picked it.

Researcher 26:17

Okay. Well, that's interesting how the, code base was also all the community was also involved in choosing the code base. That's interesting. Yeah.

#6 26:28
Yeah. We wanted to ensure that we weren't going to pick, take a framework that not many people knew cause that's less likely to inspire more first time contributors and, or to, you know, to pick up the framework and to, to start contributing to it. Yeah.

Researcher 26:47
Let's see. Yeah, we got through those. Okay. I have a, I have some questions about some of the social aspects and some of the resource management let's say in the project. I don't know if it's, I don't know if it's possible for you in terms of location and the global situation you are in, in the project, but do you ever take your work offline? I don't know if you are anywhere near anyone else?

#6 27:18
I don't think there's anyone else from the UK that's that I know of.

Researcher 27:27
Yeah. Mostly the ones I've been talking to are from Canada and the US and..

#6 27:37
Yeah, I think [redacted] Nevado and..

Researcher 27:42
Yeah, [redacted], I think live near each other. Yeah. That's what I heard. I haven't talked to any of them yet, but..

#6 27:50
They, I think they've occasionally hung out together, but no, I haven't hung out with anyone else we wanted to do. Hopefully at some point, this month, we're going to get round to doing, get everyone on a discord or a Skype call or something like that. And, you know, spend a couple of hours just doing Jellyfin and doing whatever we feel like doing. Just a Skype call to be open, you know, talk to each other stuff like that.

Researcher 28:21
Yeah. It sounds like a, like a nice idea getting to know each other when you're working in the project. So also a bit about the project structure here. Would you say the project is totally democratic then? Or are there certain people with more pull than others? Or how does that work?

#6 28:41
Yeah, so obviously you've got like the core members. There's about there's five core members. If there's any dispute, then obviously the decision is going to go to one of them, but most of the time, like in the decision of something like the user interface, it all

comes down to whoever wants to take the time to, they would, I guess, because I think it's unfair for, you know, the core members to just be like, you know, "I want this feature" Or "I want this design" when the, you know, the people that are actually writing contributions to the clients, don't really want that.

Researcher 29:27
Right. It is kind of a, an interesting thing, because on the surface, like FOSS projects, they're very, you know, open and there's this general notion that it's totally democratic. And I think [redacted], called it a "do-ocracy" If that makes sense. Does that resonate well with you?

#6 29:48
Yeah. That's a really good point. Like, if you want a feature, don't just be like, "Oh, excuse me. Would you mind doing this?" Go out and code it and, you know, learn what you need to learn to implement it.

Researcher 30:03
That's a, I thought that was a good quote from him. Let's see, we also talked a bit about this already, but do you find social bonds generally are important in the project? We already talked a bit about some of the end users and that interaction, but what, what about the, between between developers? I mean, is that also a motivational factor?

#6 30:36
Yes, a lot. We've got, we've got two internal internal messaging channels. We've got one that is just, that was meant to be on topic to do with Jellyfin to do with, you know, for request discussing features, bugs, stuff like that. And then another part, which is off topic, which is just talk about anything, something you found on Reddit. Yeah. That could just be anything that's always quite fun. Good to, you know you get into and talk about about how people are doing. Yeah. That's, that's nice.

Researcher 31:11
Also with that in mind, obviously there's these channels that are exclusive to, you the project members, but are there in turn, maybe some end users that are, let's say more connected to the project as like a regular tester or always providing great reporting is still like a couple of, like, let's say core users, or something we can call the?

#6 31:39
Yeah. So you've got a couple of, other of the matrix chats, a general one where people can just talk about anything, post stuff like how they're doing, or with the new, with like the new, is it Google Chrome cast? The one that just came out a couple of days ago, they were posting about that. Talking about whether it be possible to integrate our Android TV application with, you know, the applications from stuff like Netflix and Disney to, to not really improve the user experience and make it so your media can be on the home screen, alongside stuff from Netflix and Disney and those other apps.

Researcher 32:22
Right. Oh yeah. So filling up the like tiles or whatever they're called on the Android TV interface. I see. Okay. That's interesting. And just to get like a general notion of that, how much of the change and the project would you say is driven or influenced by completely external users? Is it most of it or is it mostly internal? How, how would you on like a scale? I don't know.

#6 32:53
I guess, quite a lot. Quite a lot of it is internal, I think. Yeah. How Julian put it is a really good description of it's a do-ocracy. Like if, if you want a feature the best way, the most likely way that is going to happen are going to appear is if you go out and implement it, cause there's nothing from the contributors, there's no, there's no actual commitment that we've made saying we are gonna do two pull requests a week or anything like that, it's sort of, yeah. Do whatever you want to. We've got the features that board where people can request stuff. And we do look at that and, it has a voting system so we can identify what, what stuff people really, really want, what stuff people know only one or two people want. So we do look at that, but at the end of the day, if I'm not going to implement a feature just for the sake of the sake of implementing it is. Yeah. Yeah. I feel that there's other stuff that I could work on would be more a better use of my time.

Researcher 34:09
Right. Yeah. Okay. That makes sense. Also, what would you assign the greatest value in the project in terms of like, what would you assign the most resources for? If you could right here and now

#6 34:23
What'd you mean like,

Researcher 34:24
Yeah. Like front end, back end the code quality, whatever is the big issue as you see it

#6 34:40
I'm not sure. Not sure. I think we've got, we've got quite a good balance of people. It'd be nice to have, be nice to have a couple more people working on the Vue client, but thats alright, it's just me and a another couple of people at the moment, but that's fine.

Researcher 34:56
But you feel the, you feel the balance is, is okay. It doesn't have to be like this one big thing. It was more a, if you had some gripe that was just..

#6 35:07
If I like some extra resources, it would be really nice to get a native iOS app. Cause I've got an iPad and I really, I like Jellyfin I use it on that, but having it, using it through the web browser or using it through the Jellyfin app, which is a implementation of a web view, isn't quite the same as isn't quite the same as having, having a native app. So

that's a lot, one of the things I've been doing in my spare time of being kind of teaching myself, React native to get, to get a native iOS app.

Researcher 35:46
Yeah. Yeah. That makes sense.

#6 35:53
eah. We've got a couple of people working on Android and Android TV, but as it being much iOS specific development going on.

Researcher 36:01
Yeah. It's a, it's actually a funny with the whole, the whole client situation since I didn't really consider that going into this project, but I can totally see how that's really specific to Jellyfin having to support that many players and platforms

#6 36:19
You've got so many different, different stuff. You've got iPads, Apple TV, iPhones. Yeah.

Researcher 36:31
Smart TVs and the Chromecast

#636:33
Yeah. People are trying to get, get a Jellyfin app up on the.. I don't know what the store's called, but the LG store

Researcher 36:41
Yeah. Trying to find someone to develop that is probably not going to be...

#6 36:45
Yeah. I think we're pretty close though. There's a list of guidelines on one of, I think LG's websites to get an app into the store. And we're trying to slowly but surely try and try to hit so we can get that in the store.

Researcher 37:03
And what would the Vue of client then help in terms of modularity and porting it to all these different systems?

#6 37:11
Yeah. Hopefully so that, so hopefully, so in one of the things that hopefully the Vue client will help me is make it really easy to switch out components on different, different platforms. So being able to switch out the layout, if someone is using it on a TV to make the TV experience better and meet those guidelines that LG whoever set out.

Researcher 37:44

Right. I have, I have a kind of a outlier question here that I missed. I just want to ask it anyway. I think you kinda already explained it a bit, but I just want to ask it anyway. Did you spend any time educating other project members? It can be UX, code or something else I'm thinking in terms of the do-ocracy you're probably all pretty self-driven, but..

#6 38:11
what do you mean.. If someone makes, makes a pull request, for example..

Researcher 38:16
Yeah. Let's say, let's say they implement something and you think maybe that's not the most of the most efficient way of doing it or something

#6 38:25
Yes. So yeah. I always try to, when I'm reviewing them always, I'll go and have a brief look out and see what else is on the web to see whether there is a better method of, of doing it. But as I said, there's there in that the front end, me and Julian.. Julian is much more experienced than me. So normally its him educating me about, you know, a better way to do something. But that's alright. I feel that that's a really good.

Researcher 38:56
Part of it. Yeah, totally. Let's see. Yeah. I think we have just two questions left or two like a general broad questions left. They were about like,future improvement. When I say future improvement I mean, not only Jellyfin, I mean, all FOSS projects, really. First of all, when you're, let's say have a question about some user interface issue or usability issue or something, do you ever look up any like academic sources, papers, books, or something on how to manage what you're unsure off? And if you do, what form should these things normally be stored in? What do you prefer? Like text? Is it video? Is it some other interactive format?

#6 39:50
I've never really looked up any academic papers or anything like that to look at these, these guidelines.

Researcher 39:55
They're not the most, not the most approachable things in the world.

#6 40:00
Well, one of the things I find quite interesting is actually apples like design. I think they call it human interaction guidelines. I'll occasionally look on their website. Cause, like maybe not quite to this extent, but you can give an iOS device to almost anyone and they will be able to pick it up quite quickly, understand how to use it, how to do the right things have to get to know it quite quickly. So I guess, because of, you know, my experience that I feel like Iphones and iOS devices are quite easy to pick up. I do occasionally go there. But the other thing we do is we've got an issue that's used to

basically collate information to other platforms. So stuff like Netflix, Plex and Spotify. So we can look at how they've implemented features and design.

Researcher 40:56
So just like a, oh, I saw it's the pinned issue on, on the web repo, right? Yeah. Yeah. So that's like if an inspiration, inspirational type of thing. Yeah.

#6 41:08
Yeah. So you can take the, basically the best of Netflix, Disney, Plex, whatever other clients there are and bring them all into, hopefully, you know, bring them all together into Jellyfin.

Researcher 41:21
Right. Do you, and now you're the, now that you mentioned HIGs or human interaction guidelines, could you see, Jellyfin having like a guideline, like GitHub page with like a template, let's say windows or placement and padding margins and all this stuff that kind of goes into making these decisions for like a centralized location for all of that.

#6 41:51
I think that'd be good to have, but at this point I think that like, it would be, I think far to formal. Whereas like, as I say, you can have people that come along and contribute. They don't want to have to be looking at guidelines and saying, have I got the right margin and stuff like that. I think it would be good to have, but..

Researcher 42:12
It might also scare people off if you think

#6 42:16
Yeah. Scare people off. If you've got really strict guidelines, you know, contributing guidelines for your first...

Researcher 42:24
Couple of small code rewrites or additions or whatever. Yeah. Okay.

#6 42:31
Yeah. That's all right. And then yeah. If they implement something, we can always pull that branch and you can look at it and see what we think, and what is our opinion. And then they can go away, change it and then we can yeah. End up with like a good implementation of that.

Researcher 42:53
Yeah. It is quite an interesting thing. You mentioned human design guidelines because they no human interaction guidelines. Rather some of the larger FOSS projects let's say a gnome or gnome. I don't know how the, the best it's probably depending on your location in the world. And they have like really strict guidelines and Mozilla also has

some pretty fleshed out like human interaction guidelines on what goes, where and this stuff. Do you think that's usually something that comes with the project simply being a lot larger? Or do you think it's, does it scale?

#6 43:31
Yeah, I think as the project gets, yeah, as it gets bigger and bigger and you have people implementing stuff, you're going to want it to be consistent across all the devices. To look the same or pretty similar, across everything. Like, "Hey, I've complained about Android for a second. I've got an Android phone." When you see stuff like Gmail or Google drive being redesigned, it looks really nice afterwards. But the effect of these... All the apps redesigns are being released at different times. I really hate it because you end up with some apps looking a completely different way. Whereas if I take iOS, they had a <inaudible>. They move in steps. So IOS 13, 14, et cetera. And the same thing with dark mode, I think that came in android nine. Yeah. I've got, I've havent got a pixel. I've got a oneplus device I'm on Android 10 and I've half got a dark mode, but not completely, not fully implemented. And it's not user friendly as I would like

Researcher 44:55
Yeah. It's not everywhere yet. no, I'm also on a, I'm on a one plus six. I have kind of the same. It's, it's almost everywhere, but not really. It's like some apps just don't..

#6 45:08
Yeah. Yeah. And not the other thing is when I'm on my iPad, I can quickly swipe down, tap and turn on dark mode. And I would have thought pixel devices can do that. Why can't my one plus?. And it's on guess not the latest anymore. Now that Android 11 is out, it was almost the latest versions. Why isn't that implemented? I have to go into some settings in appearance. So yeah, that can be a bit frustrating. Yeah.

Researcher 45:34
That is the eternal discussion on the more close something is the more, probably at least the design, how the it'll at least help the design and how easy it is. But again, you're also removing features in terms of, let's say comparing Apple and Android, it is kind of a..

#6 45:59
One of the other things that I've complained about in the in the off topic chat is messaging apps on Android or Google messaging apps. Cause at one point you had Hangouts, you got Gmail messages, Hello, wherever the other one is like, you've got a ridiculous number of messaging apps. And my Google home just started trying to talk to me and you've got a ridiculous number of apps. You know, if I was a user coming along, wanting to use it and someone said, Oh, download this Google app. And then there's five, 10 different messaging apps. It's not, it's not useful at all.

Researcher 46:42

No, I, I, that's also a, I finally gotten most of my friends to use Google duo now. And now I just learned that they're killing Google duo in terms of, instead they want to use Google meetup or meet or whatever it's called. Oh no, not again. Yeah.

#6 47:01
And stuff like YouTube now, is it like a, I don't know whether it's still there, they've removed it like a DM place at one point. I'm like, if you showed up at YouTubeI'm just going to go share the link

Researcher 47:18
Yeah. It gets quickly gets kind of convoluted. What should be like the primary, like the Imessage of Android. What's that really like? And also the like FaceTime of Android. That's not really standardized. And also it's really different from country to country. What do you, WhatsApp in the UK?

#6 47:40
Yeah. WhatsApp is, is very big. Yeah.

Researcher 47:43
Right. It's funny because in Denmark, nobody uses WhatsApp. I have no clue why. I think it's like Facebook messenger. It's just, everyone's using messenger. I think it just like Facebook was just really early on that whole market. And then now everyone's using messenger. It's kind of a weird thing. How that's so different.

#6 48:07
Yeah. If, if, if like you don't have WhatsApp, you're going to struggle with if like, not going to struggle, but like in a job it's going to be a pain. I've got like three or four group chats at various jobs. They can indicate when, you know, when you need to come into work, whether you need to do training and stuff like that. It's yeah. If you didn't have all that stuff, it would be a pain to get that information. Yeah.

Researcher 48:37
Yeah. It is. It is that you kind of rely on that one platform and then suddenly it's deleted or something. That's not the best. Yeah. Not just not to stray too much off topic here. I have one final question. So this is kind of a broad question as well. So according to you, what would you say is the largest challenge in having a strong usability or UX in FOSS projects and how would you go about addressing it, designing something or changing something?

#6 49:13
So like..

Researcher 49:15

It can be just to say some fancy words that cause you like lack of resources or like financial incentives or user adoption could be a lot of things, but, but just, if you had some thoughts on

#6 49:32
I guess the best the, if to like, I guess, to improve user interaction in, in like for example, Jellyfin the best thing we could do would be try and get more people onto Github and looking and then firstly getting on GitHub. And then secondly, before someone maybe as like a feature is designed and implemented, have a discussion, try and get some people off Reddit, get their opinions about how it should be implemented as an issue before, before the feature is done. So you can have people that might not necessarily know the code, but they use the app day in day out to get their opinions on what the best design would be. Yeah. I think that would be the best way forward to get, get more people in to discuss designs and stuff like that.

Researcher 50:35
Yeah. That makes sense. I think, discussion is definitely something that's kinda missing a bit with that in mind. I don't know if you've seen the discussions module or tap or plugin or whatever you want to call it on, on GitHub.

#6 50:49
Yeah. I think it's in currently in beta, we were trying to get it on a couple of our repos. I don't know whether we've got it yet, but hopefully yeah. Again, that would be good to have, you know, for discussions.

Researcher 51:05
Yeah. I think, I think we made it all the way through. Let's see if there's any obvious things I missed. Nope. I think that's, I think that's all for now. If I have any followup questions at some point, is it okay if I just send them on a matrix or discord or something?

#6 51:34
Yeah. Probably Matrix is the best, i dont use discord

Researcher 51:36
Yeah. No, totally fine. Yeah. I probably won't have any, I think we have got a pretty detailed discussion. really, really nice. Yeah. And, and again, thank you so much for wanting to participate in spend some time. Hopefully what I'm trying to do is kind of take all these findings. And first of all, verify existing studies that have been done in FOSS projects and then try to create some kind of space it's broad right now and a fussy, but like yeah, some kind of design to support FOSS communities in making design decisions somehow, or at least create some initial steps to do. So I think a lot of the findings currently are very academic and not very approachable, but yeah. And also a lot of the studies are like 15 years old, so technologically. A lot of it has happened since like 2006.

#6 52:36
Yeah, like if I wanted to look at reasearch something to do with user design, there's no way I would go an pick a paper, it it's just not very accessible. Easy to understand.

Researcher 52:46
No. And it's like, usually it's like one tiny niche part off like the field you're looking at and then all this, these 200 citations and then you need to read all those 200 papers and yeah, it's a whole, it's a whole thing going down that rabbit hole. I'm currently kind of reading through the whole, the whole field. So, but yeah. But, but yeah, again, thank you so much. And yeah. Let me know if, yeah, if you need the transcript or something in any case, I'll probably send either like the people who, who I interviewed or maybe even posted in the general chat or something and when I'm done sometime in probably like January or something when I'm completely done with the thesis.

#6 53:32
So yeah. That'd be quite cool actually. Just did I read a copy of that thesis when its done.

Researcher
Yeah. It is. You can, I can probably already say you can probably skip the analysis and like the mid part, but like the conclusion and the introduction and maybe some of the findings definitely interesting currently it says it's basically going to be on Jellyfin. And then I have one interview from a guy who was basically like soloing a, FOSS project. He has only like himself developing and then a couple of people in Discord, like coming up with with ideas. So that's kind of a comparison between the two, how, how that, how that differs. But yeah, hopefully it turns out pretty good, but so far the findings are really nice. So, so yeah. Thank you so much.

#6 54:23
no worries. Yeah. Best of luck for the rest of your thesis

Researcher 54:25
Yeah. Thank you. And, and you too with the, the whole Vue client going on can imagine the, the work that needs to go into that. Great. Yeah. In any case, have a, have a nice day.

#6 54:40
Yeah, you too.

10.8. Interview #7, Jellyfin

Researcher 00:01

There we go. Great. Okay. So yeah. Thanks for wanting to participate first of all, and yeah. So just a bit about the project. I already wrote a little summary of what I'm trying to do, but it is kind of a, big thing and might be a little fussy in terms of what exactly it's, about. But what I'm basically trying to do is merge my, let's say, "hobby" interest for FOSS and FOSS projects with my university degree in UX and trying to figure out how I can inform some kind of design that will potentially help FOSS projects in making consistent design decisions. That can be existing FOSS projects or new starting FOSS projects So, yeah. So that's where you come in, first of all what is your day job, but what do you do for a living?

#7 00:58

So I'm a primarily a system administrator and architect, so I run servers for internet service providers.

Researcher 01:07

Okay. And how does that, let's say tie in to how you got into FOSS in general. And Jellyfin specifically?

#7 01:17

In general, I was using Linux for my own purposes for a long time, but it kind of really pushed me that way. I'm using it every day at work. I started using it at home more too. And that kind of brought me more into the FOSS for its own ends, sort of ecosystem and way of thinking. That was seven or eight years ago now. So I just kind of immersed myself in it since then. And yeah, so for Jellyfin in particular, it didn't have too much in the way of like influence in the sense that like, I didn't really go into making, Jellyfin kind of thinking like, Oh, I'm a system administrator, I want this to work, but it does, it kind of tied in, in the way that I ran Emby because I wanted a video streaming, my own self hosted video streamer. And when they started, you know, being scummy for lack of a better word, I just kinda was like, okay, I've had enough of this. And I had the motivation to say, "Hey, anyone else want to join me in forking?" It, I can't and fully admitted, I can't do anything. Like, I don't know the code. I don't know any of that, but I'll manage the project and yeah. Went from there.

Researcher 02:33

Really interesting. Yeah. I can totally relate to a lot of that. I'm also kind of a being pushed to find alternatives for everything through using Linux. So I, I get that, that standpoint. So just to, just to confirm what I'm, what I'm hearing. So your primary motivation was, I'd say ideological, would you agree with that?

#7 02:55

I'd say yes.

Researcher 02:56

Yeah. Yeah. Since you called it, "scummy" behaviour.. And I agree totally. So you're, you're currently managing the project as well. And what, what does that entail? Just, just so I'm totally on the same page as you.

#7 03:11

So basically it's pretty nebulous on a date on a day to day. I don't tend to do a lot in terms of like code. Like I tend to just watch what other contributors are doing with an eye towards kind of keeping the project moving towards like the goal of, you know, being better, which is it's kind of nebulous term, but, but yeah. And then I'm also primarily around releases is when I do most of the work that I do personally is like, cause I act as the release manager as well. So as, as the, yeah, like master branch starts to get more and more solidified, "these are the features we have, it's getting ready". Then I start to take a more active day to day role in terms of like, "okay, can we get this bug fixed merged?" Can we get this feature finished? Can we do, you know this or that? And then when the day comes, you know, okay, we're cutting off new features. Okay. Testing phase and then okay. We're releasing today. And that kind of thing. I handle the <inaudible> unstable afterwards.

Researcher 04:24

Really interesting. Okay. So kind of a acting project manager or something like that would be a, if I'm going for the analogy of the tradional roles.

#7 04:37

I kind of call myself the project manager and people are like, "what do you mean by that?", And I'm like, think like your enterprise workplace.

Researcher 04:43

Yeah, I'm trying to equate it to some, you know, let's call it best practice for the sake of simplicity, but yeah.

#7 04:53

Yeah. It's a very.. I've had a lot of people ask me about that. And it's like, it's a very unusual kind of way of working for a FOSS project.

Researcher 05:02

Right. Yeah. And that is, that is one of the..

#7 05:06

Cause I like of the four or five people who were posting in my original issue, most of whom are still contributors. Only one of us knew anything about C-sharp code. So it kind of forced our hand in that regard to move towards the sort of, "okay, we will have a leadership structure who may not be writing code, but they're mostly for the ideological and like governance portion of it and then we'll find contributors" and we did.

Researcher 05:34

Right. Okay. That's quite the, quite the story and challenge I can imagine. So let's see, where did we go? Okay. So yeah. About the motivations for, for contributing. So you were definitely ideological. What about the, what about the social rewards? That I'm guessing is inherently a part of being in FOSS as well. Is that, would you say also a motivational factor for you?

#7 06:02

I would say it wasn't at the beginning. And actually for me personally, it was more of a detriment at the beginning cause I was never really involved in any sort of FOSS project in that way before I'd always been a user. And in the beginning it was a little weird to get used to. But I think at this point now, you know, moving almost two years into it, it's the social aspect is a big part of why I still do it because I love the community that we've built like of the contributors and just the wider community it's really fun to interact with.

Researcher 06:35

Right. And kind of related to that wider community, you mentioned there. Feedback, I can imagine, is quite a big value creating aspect of FOSS. What's your take on feedback? Is it mostly negative, positive, or is it frustrating and all, or what's your take on feedback as it is currently?

#7 06:56

I think for the most part it's been positive. There's a couple things that come up frequently specifically that I find really annoying. The major one being the "Jellyfin isn't polished or isn't like ready" or isn't like that one always kind of drives me nuts. Cause I'm like, what do you actually mean by that? But for the most part I've found we've gotten really good user feedback. Like people, people tend to suggest things that at least to me make me go, I never thought of that. I would have not saw it in my use case that that's important, but clearly a lot of people think it is. And that's something that, you know, helps influence where we take our decisions in terms of like, you know, what should we prioritize? In terms of what we should work on, kind of necessitates a very hands off approach. Like there's no, like I can't force anyone to do anything. I can't be like, "no, don't work on that, work on this instead". But it definitely helps, you know, if we can put like, if we have that kind of feedback, it tends to get everyone thinking, "okay, well maybe, yeah maybe my idea of what and where we should go or what we should do is not necessarily the same as what most of our users think". And sometimes we decide, maybe the users don't know what they really want other times we think, "okay, the users are definitely right here". It's right down the middle, but it definitely helps inform what we want to do either way.

Researcher 08:30

Right. So, so what I, what I gathered from that is it's kind of a balance right. Between creating value from the feedback and you know, maybe discarding, what's not, let's say really beneficial to the project in that sense.

#7 08:48

Yeah. It definitely it's very, very much a balance.

Researcher 08:52

Yeah. Really interesting. Yeah. That's, that's pretty much also the same finding I've been confirming so far, at least that it's very kind of, you know, difficult to sort out everything necessarily what'd you can get value from and what's more, you know, "why is it not like Netflix", that kind of, that kind of Reddit post.

#7 09:16

Good. We get a lot of that, but it's always why aren't you like Plex

Researcher 09:20

Yeah, yeah, yeah. That would probably be a better equivalent. Yeah. Yeah. Well, that's a good thing in my, in my book, but, also on the, on the topic of feedback, would you say that there's many passive users as well? And, and do you feel that the external reporting you get is enough as it is? Or would you like more reports coming in or?

#7 09:45

I think where we're at right now is pretty good. We, get a lot of feedback that we don't necessarily see just because of how, how big it's gotten. But I think we get a lot of useful feedback from mostly from like our Reddit community and our Fider as to what people want. And I think it's right at the edge of too much, but it's so far it's been pretty balanced. I think like there's a good healthy amount of good feedback that comes in and the bad feedback tends to be discouraged from at least from the official places we ask for feedback.

Researcher 10:29

Okay. Oh, that's interesting. I can imagine it being also kind of a balancing act between too much of a resource hog, and then you know really beneficial to the project. So yeah, again kind of a balancing act. Okay. So maybe moving on a bit into more usability or UX specific questions. What is your general perception of usability or UX, depending on what terms you use and maybe a relevant question there would also be, what terms do you use in the project? Do you have like a shared terminology you use or is it more, you know, each, each to their own? How do you manage manage that?

#7 11:05

It's a very each to their own and it's kind of caused some problems in the past internally in terms of like people, some people want to change some aspects of the UI, some don't, but generally I tend to take the opinion that, you know, if someone wants to make a change, I am all for it. And if the feedback we get in response to the change is positive, then great. Like, that's a good change. if we get people saying, "Hey, this is terrible. Why'd you do this?" We tend to at least rework it, if not entirely revert it. But thankfully that hasn't actually happened. Probably the biggest example of that, that I can think of in terms of like a major UI change was we, moved some of our buttons around, like we used to have, well, Emby had a lot more buttons along its top right. We moved a bunch of those into the sidebar. And I was personally against that. Cause I'm like, well, I liked them up there. But then the general feedback actually was quite positive in that regard. Cause it was like, "Hey, this looks less cluttered". So I was like, okay, well I've used it for a while now and I'm kind of, you know, over it.

Researcher 12:31

Yeah. Okay. So that was pretty democratic of you to not enforce the project managers strict rule on the project there.

#7 12:44

A lot of that comes from the fact that I'm not a UX person. I mean i tend to put it as I'll always try to adapt my workflow to the program I'm using rather than try to adapt the program to my workflow. So if the UX is absolutely terrible, but I need the thing work, I'll just accept what's wrong with it Which for something, I mean, GIMP.. We all know.

Researcher 13:17

Yeah. That's a, that's a good example of something that's extremely powerful code wise, but not really a...

#7 13:22

Yeah. Especially not if you are used to Photoshop.

Researcher 13:24

Well, no, then it's completely broken. Yeah. Yeah. That's a, I luckily started using a gimp and just never touched Photoshop though. So that was my solution to it.

#7 13:34

Same boat. I never really used Photoshop and it's like, but even learning how it's kind of like some of these things seem that we should be very obvious, but yeah. Like how do I draw a box?

Researcher 13:47

Yeah. That's a, that's one of the like big like dinosaurs in FOSS I think, and it's one of the really good examples on, you know why aren't more businesses not using this. It seems like such a huge license saver and can do like 90% of what Photoshop does most of the time and yeah. The reason, for this project basically kind of as. Oh, not to get too off track here. I could talk about FOSS projects all day, so let's see. Yeah. Okay. So do you feel that then since it is kind of a, you know, a subjective decision on what usability entails, what does UX entail to each contributor in that sense. Would you then say that this subjectivity is also kind of an issue in the project? Or do you ever, you know, how do you manage, if there cant be established kind of consensus on an issue, how do you, how do you resolve something like that?

#7 14:53

It's a very good question. Thankfully, it actually has really come up at what I mentioned about the moving some buttons actually the closest we've had to like a really big split and people not agreeing with things. And it was such a minor change that it's like, "Oh, ultimately this isn't a big issue", but thankfully so far we haven't had any big ones, mostly because I don't think we've really made any major UI changes. Like there's been a few, there was two releases ago, a fairly substantial rework. I'll use the word rework of how things look, but it wasn't like a significant change. Like we moved from having some like overlay buttons to buttons up top and, and things like that. And everyone kind of agreed with the change, and there were some comments and complaint and points from how things were before, but for the most part, yeah, we haven't actually come across a situation where it was like a real make or, "Hey, I want to change the UI to look like this" where a bunch of other people are like, "no, we don't like that change". It's been pretty seamless. I think if we did run into that situation, that would be a time where I would pretty much use my .." hey as a project leader, I really don't think there's consensus on this. We should defer this or", you know, make it sort of optional. And that's something we've tried to, we try to push. Again, it hasn't really happened yet. But if its something like that, like that choice between two major UI components, I'd say, "Hey, let's do them both and put a select box in. Or like if possible, put some sort of switch in there somewhere that says to me, "hey if you want this new way of doing it, turn it on". And actually kind of in a way, doing that right now with our new web UI, that's being worked on, there's a Vue client being worked on by a couple of the contributors, their reasons were more technical than like UI. I think so far it looks pretty similar, but it's different. And that's a kind of a place where I think, we can have more options, and we can have two ways of doing this. And I don't see that as a failing of the project. Cause I know some projects will typically be like, "we want to have a very unified UI, everything looks the same, all of our clients look the same" and i tend to take a more hands off approach to that, or <inaudible> (leaving in more options rather than excluding them)

Researcher 17:42

Right. So let's see. Okay. So yeah. So you mentioned this consensus thing here where you don't see having two things necessarily as an issue. Could you see yourself looking up any like external, let's say authoritative guidelines or something to, guide those decisions or?

#7 18:22

Yeah, I haven't, but I definitely would prefer to, if that situation came up, especially there's been, I actually don't know what happened with it because, you know, hands-off day to day for the most part, but there was, and an issue raised quite a while ago actually by I believe. Yeah. It was a, a visually impaired user who wanted, who was concerned with some aspects of it not being readable by a screen reader. And I believe there was actually some changes to just make sure everything was, everything had like, a tag on it that could be read by the screen reader. And I feel like that probably made it in, but that's the kind of thing where I was like, yeah, I I'll look at what are the actual, you know, general UX guidelines for this sort of thing, because obviously it's not an issue I face or anyone I know personally faces and the consensus was yeah. You know, make sure there's a label on everything make sure there is a tag on everything. Those are kind of the best practices in the industry. And we said, okay, we can do that.

Researcher 19:28

Right. Okay. That makes a lot of sense. Yeah. Accessibility is kind of one of those things where it's kind of difficult to design for without yeah. Without knowing exactly what to do. Found the same in my work. And then with, those changes in mind. Do you have any kind of systematic testing or how do you, how do you test those changes? Are you all just running the bleeding edge or how do you, how do you manage that?

#7 19:54

The process tends to be for any sort of UX change to just, we have the pull request open we all tend to build our own if that happens, we'll be like, okay, everyone build, you know, the latest version of this PR and let's see how it looks. And that's what I do personally. I'm not sure about everyone else on the team, but I tend to just build the deb package for myself with, you know, the pull request changes in it and I'll run it on one of my instances for a couple of days and just, you know, browse through it, see how it looks. And if I like, you know, if I like the change it will get my check mark or if I don't like the change, I'll put my feedback in. But yeah, there isn't really any sort of official testing system or policy in place for how we make changes like that. It tends to just be ad hoc. And like, I usually request people put screenshots, like before and after screenshots in their pull requests, if that, if there is like a major UI change, sometimes even the code is enough to just be like, okay, I can see the change and i like it.

Researcher 21:03

Right. Okay. You mentioned the, the pull requests there. I just want to ask, do you keep track of the design rationale, like the historic changes or are they simply stored in those and archived on GitHub?

#7 21:19

They're just stored on, on GitHub. I haven't been keeping track really of the major changes.

Researcher 21:27

Do you, then I also noticed the, the blog posts that are on there, the primary website, how does that.. I noticed that some of the, some of the stuff is more technical. Some of it is a bit more UI. Yeah. How does that tie into that? Is that meant to be some kind of like a, let's say history of the project or how do you, I don't know how much you personally have to do with the blog.

#7 21:54

Yeah. It's kind of interesting. It's very, it's, it's very much just anyone who wants to write a blog post on the team does, and then we all kind of look it over. Yeah. There were a couple, I believe [redacted] did a number of blog posts about UI changes and that was kind of his wheelhouse. And I said, "Hey, go for it". Like write up what you want. I'll read through it. I will make sure it sounds coherent and sounds good. But for the most part, yeah. That's, it's just, whoever wants to write one on the team writes one. And I liked that from the perspective of having those multiple, you know, opinions on things and multiple people on things and have someone who's more into the front end, right up on a frontend blog post versus me trying to write, "it looks shiny".

Researcher 22:42

Yeah. "It's pretty". Yeah. Yeah. That's not the, yeah..

#7 22:48

Well, it's my opinion on a lot of things. It's like "this looks, this looks pretty good". Okay. Let's go.

Researcher 22:52

Yeah. But I mean, that is a, to some, a valid usability measurement. Right. Also, actually on that note, I'll have to ask you, do you know if there's any resources in the project or contributors only allocated for UX? Like maybe not even doing code, just like wire frames or mock ups, something like that?

#7 23:26

No, there isn't. I know there was some talk. We talked a little while ago about using Figma for some mockups and that didn't really like, it's not that it didn't go anywhere, but I, I mean, I didn't really try it and I don't think anyone actually has, but for the most part, yeah. Right now our quote on quote front end team is quite small. Really. There's only like two people. Well, I think really focused on it. So I tend to just say, Hey, what did you want to do? And if everyone hates it. As I mentioned earlier, like we'll visit that when it comes up. But if not just, do it.

Researcher 24:12

Right, and then also kind of on the same topic there, have you ever been approached by any professionals? Like from external somewhere, it doesn't really matter, but to like help out the project or something?

#7 24:26

Yes. There was very early in the project. I have to, I say, I think like six or eight months in, someone did <inaudible> and said, "Hey, im a UX designer, I'd like to help out". And I was like, "great!". Here's all the resources. Like if you're interested, you know, we'd love to hear more from you. And then we never heard anything from them.

Researcher 24:43

Okay. Okay. Yeah. Okay. That's a..

#7 24:46

I thought about it later and I thought, "Oh, I'd really like it If this person had joined like at that point in the project. Cause I think that's, would be a major benefit to us is to have at least a couple people who are very skilled in like UX design to kind of hash out these things amongst themselves in presenting the project as a whole.

Researcher 25:07

Yeah. I mean, I can definitely imagine it being a kind of yeah. Taking some of the, the workload from some of the developers that maybe are capable of doing some of the technical stuff that some of those decisions could be, you know, kind of made for them so that they don't have to also do all that work. Okay. Let's see. Also kind of relates to what I asked earlier. I just want to get your take on it. Anyway, do you employ any methods in the project between like contributors or end users with a focus on usability? This can be like the blog posts would be one possible example, Or like, do you have like a monthly Reddit post or something?

#7 25:57

Pretty much nothing explicit. We have the blog posts,. And then we have the Fider to kind of take things from the opposite direction. And Reddit posts from users tends to be another way that we get feedback from users to us, but nothing official. It's all very ad hoc. When something comes up, we've had quite a few posts saying, "Hey, you know, I don't like this or Hey, I really liked this and do more of it or do less of it". And I, and given that everyone's pretty active, at least on the Reddit I believe all that feedback is taken into account by the people designing, but I've never really established any sort of formal process or like any timed, you know, feedback. And also in the future, the fact that you're asking these questions is great and I might ask to leverage you later, like we did our user survey a couple months ago, but it ended up being, I asked quite a few questions in that survey, but it ended up being basically kind of like, it wasn't useless in the sense that like, we got some really important information out of it, but it was very broad. And one thing I've wanted to do for awhile now is a more in depth survey, things like UX and specifics about what people like, and don't like about the project. So I guess my problem is writing the questions that are needed,

Researcher 27:30

Yeah. Yeah. That's not the..

#7 27:32

Is it pretty? Yes. No. Okay.?

Researcher 27:33

10 out of 10. Pretty. Yeah, exactly. That is, that is really interesting. And it's not easy doing surveys that are beneficial. Really. It is quite difficult, honestly. Let's see. And you mentioned before I forget, you mentioned Fider is it's the feature requests platform, right. Also that kind of ties into this question. Do you feel then that GitHub is sufficient for all types of issues or are certain issues like usability issues, maybe complexity wise, harder to keep track off would you say?

#7 28:10

I tend to flip flop on that. I think GitHub is very, very good for most things, we moved features off of GitHub, mostly just because of the upvote system that fighter gave us as well as to try to avoid a very major, like backlog of <inaudible> in our issues board. I think GitHub could be more powerful if it truly like split issues from features from bugs kind of thing. But since it's all tied into that one issues, tab it tends to get a little overwhelming. I think at one point we had like 600 or something open issues and that's where we decided, "Hey, we should probably split feature requests". So we can manage those separate. Mostly for our purpose, because we're still at the point where we're trying to like rebuild the backend code a lot and implementing new features really isn't our priority. So we didn't want that it was like feature requests at issue number 200 to get lost behind like 20 pages of, "Hey, there's a bug in like version 10.6" or whatever. So at least in their own place where they're, they can be thought of separately.

Researcher 29:35

Yeah. Also I can see that Fider has more of like an overview rather than like separate threads on something that doesn't work or would be nice to have kind of gives more of a democratic view on what's currently wanted. Let's see if I got all of those..Yeah, I did. Okay. So some questions I have also kind of go into more some of the social aspects and a bit of the resource management, we've all ready, discussed a little.. kind of a funny question, but I think somehow important. Do you ever take your work offline, like meet up with some of the other developers or like, are there any like social activities in the project?

#7 30:22

Right now? No. One of the other major contributors, [redacted] and I are our good friends in real life. So we, I actually kinda, for lack of a better word, brought him online very early on because I wanted someone I could talk to in real life about it. But for the most part, no, we're entirely online and we were planning on some get togethers in 2020 until you know, covid.

Researcher 30:47

Oh yeah, yeah. That pretty quickly stops...

#7 30:54

Like I've never been to Europe and that was one thing I was considering in 2020 before COVID was going to Europe and right around now. And I probably would have said, "Hey, Jellyfin contributors. I'm going to be in Europe, who wants to meet up for a beer or something?" But then yeah..

Researcher 31:09

Yeah. That's a lot of plans that were kind of stopped immediately when that happened here in Denmark. It's like a week ago we have had kind of increasing numbers for a while now we killed it really early on and then it kind of returned a bit because everyone was like, yeah, it's gone now. No problem.

#7 31:29

What happened to us the last, in the last two weeks, our case numbers we were doing so well, Ontario, Canada, and all of a sudden we were like, okay, let's open every, like almost everything back up. And then now the cases they're spiking again, they're like shut everything down again...

Researcher 31:46

It seems such a, like, I don't know if you have the same, but we have so many like a kind of weird restrictions like schools, no they're open, no problem. But a bars. Yeah. They close at 10. Museums? Hell no, forget about it.

#7 31:58

They make no sense. They're like restaurants and bars get totally open, but close at 11:00 PM. And you know, you have to keep social distancing inside of it. But again, it's in our school without getting too much into our local politics, our leader of our province, in my opinion, a goddamn idiot, who is like, yes, let's open all the schools. And instead of having smaller class sizes and more classrooms, he's like, let's just combine classes together.

Researcher 32:30

Well, might as well infect everyone now that we're at it. Yeah. That's great. Yep. Nice to, nice to hear that its a global effort in being stupid. So let's see here. Yeah. Okay. Also a social aspect question here. Do you find that the social bonds are important in your work, in the project then?

#7 33:03

We've made some there there's some good friendships that I think there's been online, friendships, whatever you may think of those have been made in this project, and there's like pretty much the entire team I think, has really bonded over this project in a lot of ways and found a lot of other common interests. And I think the social aspect has become a very major component of why a lot of people still stick around.

Researcher 33:26

And do you do then, I mean, one thing is of course from like contributor to contributor and project members in between, but what about some of the end users? Do you have, do you have, let's say very active, like certain let's call them "core users" for the sake of labeling them. Like do you have some like always like returning users that are like almost project members?

#7 33:53

We have seen quite a few of them. Yeah. They tend to come and go, which is, I mean, been unfortunate in a lot of ways. And I've noticed, yeah, I see a lot of the same names come up a lot, especially users who report bugs. Unfortunately I don't think there's really been too much of a like direct, like social dynamic between at least myself and them. Some of the other contributors may definitely. Kind of just as a side effect of how big we've gotten. We kind of shattered my expectations for how big this project would get very quickly. So I kind of imagined a more of a like close user community at the beginning. And then it kind of blew up. Cause I guess this was something people wanted and now it's so big that like, even if I go into like our general chat room on matrix, that tends to be very busy and there's always a lot of new names and I'm like. "Oh, hello new person"

Researcher 35:01

Yeah. I mean, yeah, it's a, seems like a saturated field, but again, not really like Plex cost money, Emby suddenly closed source and yeah.. And also costs money and yeah, it's a, it's kind of a good thing that you guys popped up on everyone's radar when you did.

#7 35:24

We were looking, a lot of us were looking to stream media as like our serious way to move, but then it was like, there really isn't... Yeah. We all thought there was so much more in this space, but really there wasn't because there were like 50 projects out there were nowhere near ready. And then the big ones.

Researcher 35:42

Yeah. How much also related to before I forget the question as well, how much of this like change in the project would you say is then driven like by external users or is it mostly internal still?

#7 35:54

I think it's mostly internal still. We tend to get a lot of, for lack of a better word feature requests from external users, but then getting them implemented is kind of hard because, you know, whether they know it or not, like the backend code that we're dealing with, it's like absolutely atrocious and trying to like our focus for the last 2 years and probably for at least the next year, just really been on making that backend code far more extensible, longterm so that we can easily implement features or that such some new developer can come in and say, "Hey, I want this feature". Oh, adding that it's easy rather than today, where it tends to be like, "Oh, you want to interface with the like database"? That's a nightmare. So we've kind of been driving most of our changes from within the team. And we also tend to take a very open approach to the team. We don't really have a hard and fast rule, but generally if someone comes in and makes, you know, three or four, five or six, somewhere in that neighborhood, like really good pull requests we will immediately reach out and be like, "Hey, would you like join the team?" If you're willing to keep contributing to this we'd love to have you". Our team has grown quite considerably via that method.

Researcher 37:15

Okay. Well, that's really interesting. You already kinda mentioned this next one here, but I just want to ask anyway. And so what, what would you assign the greatest value in the project right now from a place where you as a project manager manager could say, let's assign all the resources for this one thing to fix it immediately. Where would you place those resources right now?

#7 37:43

If there was one thing that I'd really like to get fixed, it is actually the backend database, which has been a long running kind of issue. We've been discussing replacing it, or like rebuilding it since literally day one and it's come a long, but nowhere near as.. <inaudible>, it's just a lot of work and having such a <inaudible> group of people try to work on it. It's been really hard. But other than that, I think the, like that would be one. And my other big one would be trying to like put some effort behind the transcoding sort of like actual playback part of the project, which I don't think anyone has actually really dug their heels into yet. just because its so very fragile. Very, let's just say there's a lot of decisions in there that don't make sense on face value until you try to change them and go, "Oh, that's why".

Researcher 38:45

Okay. Yeah. Also when, on the topic of like allocating resources, do you, do you ever, or is it something that happens in the project often, that contributors spend time educating each other, would you say maybe on the code or?

#7 39:03

Yeah. There's a lot of that. We kind of have a very.. because it's, I don't know how common this is in other FOSS projects, but we actually have like private channels for the team. And a lot of internal discussion goes on in those channels for, for better or worse. I know there's, there's a couple of contributors. Who'd never liked that and would rather it be public. And then there's other contributors, myself included who really like that. Cause I mean, I mean, negative connotations aside, we don't like the peanut gallery, like, "Oh, you guys are discussing this well, here's, here's my thoughts on this". And we're like, yeah, we considered that. Or, or this and that. Especially with what we are doing right now with the backend cleanup, it may be a negative that we're not putting other voices in, but at the same time getting other voices I found personally tends to just result in like devolving into like arguments over, you know, minutia of implementation details rather than, you know, trying to move towards like a grand, like action for change. So yeah, we tend to keep that internal. And because of that there is, a lot of that sort of back and forth between people, but like someone will come in and be like, Hey, I'm not really sure how this works or how that works. And they'll usually be a couple of people who do know how it works. We'll be like, "Oh yeah, this, this, this," or if no one knows how something works, there tends to be a little bit of interaction between people who like basically try to figure it out how well that works depends on the issue. The transcoding still, no one knows how it works. But actually getting the API documented was a very good pull at that. There was a couple of guys who were very focused on like, "Hey, I want to do this with the client. How do I do this?" And that resulted in this basically two or three of the guys really writing like an API spec and getting all the code really documented for that so that they could print it out and say, "Hey, your is a full API document." All the things that Emby never bothered to document.

Researcher 41:15

I mean, it's also on the topic of Emby. It's also nice to know that there are likely equally yeah. Screwed in terms of code base, I'm guessing. But yeah, when it costs money, I don't know if at the changes resource management it maybe does. What's your, what's your take on like financial incentives generally?

#7 41:40

I'm generally very against them, at least for this project. One thing I don't really like about FOSS projects when they try to go that route is basically what I saw happen to a couple of the projects that we were influenced by Subsonic and Emby being the major ones, where at least in my view.. (thinking pause) mismatched incentives! You know, two people who both want features and one of them has, you know, $300 to like throw at a project and the other person doesnt have any money. I don't think it's right to kind of privilege the person that has $300 to throw it at developer and be like implement my feature versus the, maybe it isn't one person, maybe it's one person with 300 bucks and 30 people with, you know, a buck each. And there's still a huge disconnect even though the one that helps the 30 people is probably better long term for like either usability, or the overall state of the project than the other one. And that's why I've tried to put money out of it as much as possible, we take donations. But the donations actually today don't do anything because <inaudible> is giving us a whole bunch of free servers, it was originally to for pay that. But then they were like, "Hey, have a bunch of free stuff". We're like, ah, boy, we have all this money. So these days it tends to just be like, if someone wants like a client we let them expense it, but that's pretty much the extent of money. And we've, we've actually taken a hard stance against, I can't remember the name of the site, the site that like collect bounties on features. And we actually have reached out to them and asked them to remove any bounties related to our project because we don't want it to be like, we don't want the incentive of money to come into any of our contributors, we'd want everyone to just volunteer. And if they don't think that they like think that, "Hey, I'll only contribute to Jellyfin if I get paid to do it", at least in my view, I don't want the contributer because they're gonna, they're gonna look at, "Hey, what, what can I do that maximizes my money rather than helps the project."

Researcher 44:15

Right. Yeah. I think that's a very good stance generally. Let's see. Oh, before I forget, I have one more question in the motivation section, I kind of forgot. You also mentioned the, the scope of the project that it had grown larger than you than you expected initially. And did you find that like factor motivational, like seeing the some hundred, I don't know how many thousand Docker pulls you have, or like downloads on the app store or whatever. Like, it looks very impressive on like paper, but I mean, comparing it to, let's say like the social value you mentioned, would you say that it's, it's very motivational seeing the project grow in that sense and it's user adoption or end user adoption and inherent goal?

#7 45:07

For me it was a, it was a very major thing. And, from myself thinking, Hey, this is a worthwhile thing to spend my time on. I don't think it's our major goal though. And I've always kind of taken the view that like, look, if you want to use our project for, for what you're doing, that's great. And I love that you want to use it, but if you're going to turn around and be like, well, I don't want to use it because i like Plex better. I'm like, okay, then you just use what works for you. Like it's, I don't think driving sheer contributor numbers, it should be the goal of our project. I definitely treat it more as a, "Hey, I'd rather build the best thing we possibly can for whatever definition of the best we can come up with". And if that gets a lot of people and if it beats the biggest media streamer and we are the main game in town. That's cool. That's great. If it means we're a niche project that has like a 2% market share versus the competition that's I think that's also fine. I was fully only expecting us to have like under a thousand users tops and that blew up very quick to, you mentioned the Docker pulls and that was kind of a big moment for that's when we hit a million Docker pulls on our second release, we were like, "what?"

Researcher 46:29

Yeah. And what is it now? Like a hundred...½

#7 46:31

and all these videos and all these like big podcast talking about us

Researcher 46:37

Yeah. I can imagine that being kind of, yeah, kind of crazy. Suddenly just blowing up like that. Let's see. Oh yeah. Okay. So the last couple of questions here are about like future improvement. And by that, I mean, not only for, for Jellyfin, I mean, for all FOSS projects and the industry in general. And, and with that in mind, the design im trying to create, or I just call it a design right now. It's very, not very descriptive of what it's going to be initially, but that's, that's what it is right now. Do you ever look up any like academic sources, papers, books, or something? And if you do, what's the, let's say form or format that you prefer to consume these, this knowledge. Is it video is texts. What do you prefer?

#7 47:30

I haven't actually looked up any real academic sources on this, mostly due to not really knowing where to look, but cause I'm not an academic. I, dropped out of university, so.

Researcher 47:42

I asked because it's not really, especially academic papers I think are not really that approachable and, there's actually quite a bit of interesting studies on FOSS, but you know, you have to be three pages in, on Google scholar and find three different citations and then like it's not really approachable.

#7 48:11

Yeah. And that's something I think like I like text, but in the sense of, like not so much like reading an academic paper, I've done it for other things that I find interesting, but like it's, I've never enjoyed it. But I think text i like summaries of it, like somewhere between the abstract and the massive wall of texts that tends to be an academic paper, like really quality blog posts, summing up. And I think I could make a comparison to like actually, cause I'm kind of interested in physics and such and I think of my, my favourite resource in that sense is these sort of YouTube channels guys like PBS digital studios spacetime, and that kind of thing. Where the, where the, or what's her name? Dr. Becky. I can't remember her actual name. I think it's just dr. Becky, but I'm like, they're absolutely fantastic. And taking like this academic paper that I could never read myself and breaking it down in a way that like, Oh, I understand what this is saying. And that's absolutely fascinating. So something like that I think would be a hugely valuable to the world in general, having whether they're like blog posts, written blog post, or a video saying like, "Hey, here's what this paper found about usability". And, you know, instead of the dry kind of black and white, like here's some boxes on a inside of another box, here's the like mock up with how this looks, that would be like super, super useful.

Researcher 49:50

Okay. So, so basically trying to like kind of distil the most important things from those walls of text

#7 49:59

just distilling it down into a way that someone who is not an academic could easily understand, and even in a way that a user could understand or should go a long way to, you know, helping users understand why some decisions being made and by others, they might seem good on the surface, but they don't work long term or that kind of idea.

Researcher 50:23

And then, the final question. So what would you say is the largest challenge right now and having strong usability or UX in general in FOSS projects and how would you address it?

#7 50:38

I think the biggest issue is finding a way to, I have two actually. Number one would be a way to really like make it easy to kind of play with changes, something that I've found lacking. And I know I mentioned Figma earlier, which seemed like a very good way to, to work with that. But I mean, they're kind of.. I didn't really understand how it worked, but some sort of way to be like typically take like a, I'm thinking like a tool that would take here's my UI today. How can I like see how it would look if I wanted to make this change or that change or, or this and that. And basically a way to quickly mock up changes. I mean, maybe something like that exists. I've just not found it. I haven't looked too hard, but I think that would be really powerful in terms of like a thing that would help people in the UX of like FOSS in general. And the other kind of big challenge, I think would be some sort of like, maybe not universal guidelines, but as much as I dislike Google, I loved the idea of the material design guidelines, because like, that's the sort of thing where it's like, everyone gets on the same page about, you know, the minutia elements and how, and then the choice becomes how you combine them together rather than how you like deal with what a button should look like. That would be a really powerful thing, I think, in the FOSS world. And like, I know it's true regardless of platform, but, I kind of hate the way a lot of programs, like, they have their own UX style and then you have like that superseded against like your, the rest of your system. Like, I'm trying to think of a good example right now of like where that's bad on my Linux system, but that I look at everything is kind of like themed to know for me. So there isn't really one, but like, think of all those like tools on like a windows system where you have like your X buttons in the top look, nothing like the windows X buttons and like the, the inner Y of the applications look nothing like any other app. And I think that's somewhere where Apple excels, where they enforce those design guidelines across all and any app on the platform. And that gives like that unified look and feel, which is something I think FOSS has in terms of like the, kind of like the known usability guidelines in that, but then there's still like a four or five way split between them. So it's kind of a half solved problem, but getting that the other half solved would be, you know, monumental improvement.

Researcher 53:26

Right. So, so some kind of guidelines basically would be a..

#7 53:31

More universal guidelines

Researcher 53:34

What are your, what's your take on something like a Gnome, depending on how you, they, they have pretty descriptive guidelines. Some, would call them restrictive, but I what's your take on something like that?

#7 53:52

I've always liked them because I like, I use Gnome 3 as my main DE and I always liked how their guidelines were. And once we got over the kind of initial pump of this looks nothing like what I'm used to. Like, I absolutely love it these days. Like, I can't use any other UI. It just, everything works and gels in a way that like, I can't even explain.

Researcher 54:13

Yeah. Right. I also use, I also use Gnome and it's, I think it's by far the best UI I've used ever

#7 54:22

KDE is good, but it's a little too cluttered is the wrong word. But like, it tries to be like, look at me rather than like, very, smooth which gnome is.

Researcher 54:34

I think the KDE widgets of old, like the red ball and all that stuff you could like put on the desktop. I think that summarizes perfectly KDE, why it's just like, there's so much stuff going on. And yeah, to me, it's like, after that, after those first couple of years with Linux and ditro hopping every second week to some new obscure, distro and spending 10 hours configuring your desktop, I'm just kinda like, yeah, pop os, yeah, that works. Doesn't have to change anything. Install.

#7 55:06

Yup. That's me with Debian. And I'm just like Debian stable. Okay. That's how I want it and it wont change in the next few years. Yeah.

Researcher 55:14

Yeah. I think the, the pragmatist is kind of taking over a bit, but yeah. It's like, yeah. Snap, it doesn't matter. Install flatpak? Fine. Yep. Sure. Installed and works. Its kind of the same thing. Yeah. I think we got all the way through, actually, let me just check real quick if I have any thing I forgot, right. Yeah. I actually think that's all the questions I have fun now. Is it okay if I come up with some question, all of a sudden that I just like send a single question or something down the line at some point. I definitely will. If I come up with something or I might possibly it's, it's not necessarily a thing and I don't know how I'll manage it, but I might have something to test or something for you to maybe look at down the line as well. I have my hand in this in January, so there's plenty of time still, but, yeah, it might be something to kind of for testing, whatever design I'm creating or getting some feedback, maybe something I can just bring to the oral exam or something would be awesome to have someone someone don't get. But yeah, it's at least far in the future. Still for me, but, but yeah. Thank you. Thank you so much for spending. What is it? An hour almost.

#7 56:47

Wow. Yeah, no problem at all

Researcher 56:51

Thanks for letting me pick your brain for all the good stuff you, you brought to the table here. Really awesome stuff all around.

#7 57:02

Good luck with the project. Of course you have any other questions always happy to answer.

Researcher 57:07

Great. I'll definitely let you know if I have something and then that case have a nice, what is it? Day, still for you. Yeah. Night for me now. Seven a bit over seven. So a little little day left. Yeah, exactly. Yeah. Great. Yeah. In that case, a yeah. I'll let you know if I need anything and otherwise have a nice day.

#7 57:30

All right. You too. Take care. Yeah, you too. Bye.

10.9. Interview #8, Jellyfin

Researcher **00:00**

There we go. Let's see. Yeah, that's rolling. Perfect. Great. Okay. So just first, just a little about the project. I already wrote you a little about it, but I just wanted to maybe expand a bit upon that. So basically what I am trying to do is take my, yeah, let's call it a hobby interest for, for FOSS and open source software in general, and then merge that with my, my specialization and my master's degree in UX design and trying to kind of bridge that gap and create some kind of a, currently I just call it a design, some kind of consumable format that can enrich FOSS projects in the sense that they can maybe make some more concrete and cohesive design decisions across the board. So, i'm trying to take some, all of those academic findings that exist and maybe, you know, turn them into something that's maybe a bit more approachable to FOSS projects in general and yeah, that's, that's where you come in and yeah. So, so first of all, what is, what is your day job and yeah. What do you do for living?

#8 **01:14**

Well, I'm still a student right now and studying computer science. Well, I'm probably in a similar position to you. And Jellyfin is a bit of good little side project thing as well. I have a student job working with a software company and do android development there, but Jellyfin I mean, I only started using it this year cause I wanted to get away from Spotify and to have my own collection available. For music mostly so, and I discovered the current Android just didn't work for me perfectly in the end. The problem then was the, it was still based on Cordova which is a web wrapper thats a bit outdated now. It was a huge hassle for developers to get into the project. The build environment is not something you're used to, if you do android development. So then I started rewriting that web-wrapper inside. And in the future, I already plan to build a really full native app. It's still early.

Researcher **02:27**

So you already kind of touched a bit upon it, but, but what would you say then is your primary motivation for joining Jellyfin? Was it to hone your own skills or was it more of a, you needed something to handle your music collection or where would you put yourself?

#8 **02:45**

Both. I'm very interested in music and consuming music. And Jellyfin just ticks all the boxes for me for listening to music, but where the app was lacking and I wanted to integrate it better. So I can personally use it better, but also I'm very into the FOSS ideas and I want to give back something to other people.

Researcher **03:08**

So you also let's say ideologically involved in the.. yeah. Okay. Okay. And then have you, have you always been a FOSS kind of guy or, or how did that in general come about? When did you start going into FOSS projects?

#8 **03:24**

Well, my first enrolled project was an applocker and it's basically was a fork of another one. And that got me a bit into this idea. I earlier already worked on FOSS stuff. So before I could even write any apps or do any development work. I started translating for some Android apps who are open-source and I liked the idea. I was always a Linux user because <inaudible> also uses Linux. My own app, because it was forked also was <inaudible>. So the GPL 3 and I continued. And the more I read into this, this whole topic, the more I liked it, I appreciative the idea of having code everyone kind of owns and everyone can make changes to it.

Researcher **04:19**

Yeah. Okay. That makes a lot of sense. And then would you say some, maybe a, let's say a reward of contributing in FOSS, that's maybe not at first is the, the primary goal, but maybe a, let's say a bonus of doing FOSS is maybe the social part of it. Would you say that's also a motivation to you, you know, interacting with other developers and the project and maybe even end users?

#8 **04:45**

Yeah, of course. I met a lot of friends and met a lot of people on my way, in my first app somebody approached me to write a website for my app. And so I don't have a lot of contact with them unfortunately, but was quite a long friendship, that formed out of a project. Also the

Jellyfin project is something I'm working on with other people. And we have quite a nice community and internal chats where you can talk about anything. And yeah, it's really nice.

Researcher 05:21

You mentioned the community here. Would you say that the raw user numbers are important then to the project? Would you say that like user adoption is a goal in itself in terms of, for instance, in terms of, for instance, the, I just noticed, that Docker pulls, that's a crazy number you have now and Jellyfin would you find that motivational or is it more that, you know, one-to-one or one to two group interaction you might, you might have?

#8 05:50

I don't know. Of course its important that people use your software because if nobody uses your software why even bother. I dont know it it scales so well. So because I mean, if a 100 people use it and give good feedback and on your work and appreciate you, doesn't make such a big difference if a million people use it. I mean, it's nice if you, if you're looking at numbers and you can maybe flex a bit with it and say, "i have a million users", but actually the closer the users are to you in such a project and better the feedback they give, it doesn't really matter how many users you have.

Researcher 06:27

You already mentioned the feedback here as something that's important to kind of, you know, have some kind of distinction between what's good feedback and maybe what's not so good. How is your take on feedback in general, currently in Jellyfin? Is it, are you getting enough feedback? Is it positive, negative, or maybe not detailed enough to, do you have some, some, some thoughts on that?

#8 06:51

Well, details are often a problem, I guess, because if you get a crash report or something, it's sometimes hard for users who are just not into this whole topic to get logs and something, because if they're not, you've got to <inaudible> themselves, but they don't necessarily know what these logs are even for. But generally the feedback on user experience or like they need another button, or the they don't like a feature can be very helpful already. So it's nice with the crash reports into the technical aspects, but also just the feedback or feature requests. So I believe the old android app before I got in was a bit negative and there wasn't a lot of work on it. Not so much feedback, but now that its rewritten its easier to get in for new developers, a lot of more contributions for, with code, but also I believe there's more feedback because users saw a <inaudible>, a new update this new code and this new feature. So they get into the project and also provide feedback on the features they want to see. So yeah, that's, that's really helpful.

Researcher 08:04

Okay. Okay. So with that in mind, do you feel that there's enough feedback then, or do you find that it's a good amount you currently have?

#8 08:13

It's a good amount. I mean, it's always a bit problematic if you get too much feedback or too much to many feature requests because you never can implement all of them. And I learned over the years in working with FOSS, that can also be harmful to listen to all users, you will lose focus and maybe implement too much. But generally the feedback is definitly helpful and a good thing.

Researcher 08:44

Okay. Yeah. That makes sense. Okay. So, a bit about the most specific usability stuff here and what your general perception of usability? Is it something you think about all the time, or what about some of these, you know, the terminology can also be a bit subjective, let's say UX, usability, UI, like, how do you kind of make sense of this when you're working with that side of code?

#8 09:16

Well, personally, i <inaudible> very subjective I think. I'm, if I'm using the app or any app, I instantly recognize if it's good or not good, the usability. If it's easy to use or, for some apps, I sometimes have to, I think that it's, it has problems like buttons that don't align perfectly, that you mis-click. A good example is the YouTube app. I mean, it got better in the recent times, but there was a timeframe where I always misclicked the full screen button because it was directly next to the progress bar. Now it's above. That's a huge change, so touch targets and touch target size is important. I believe for visibility. Also the app shouldn't crash. I mean, that's a given, but the rest is very subjective for me.

Researcher **10:21**

Okay. So you mentioned subjectivity here being, being kind of a factor, do you then find that can ever create some kind of, let's say friction in projects where everyone has their own view on, let's say your usability as a concept, and then like how, how do you, how do you then manage all of those kind of expectations of these things?

#8 **10:44**

Yeah. And of course we had some requests on the notification, the music notification for Jellyfin, so I don't know.. in the earlier app you could swipe away the notification if the track was playing, I think, but in my opinion, a music player, if it's currently playing, you shouldn't be able to swipe away. And I think also google does say in the guidelines that music notifications are ongoing events that shouldn't be cleared

Researcher **11:13**

Right, from the material guidelines?

#8 **11:17**

the Android guidelines.

#8 **11:22**

so I changed it up. So I said, okay, if you, if you want to move the notification you first have to pause the playlist. But then there were some people who said, okay, maybe at a stop button so that you can stop from notification, but I wasn't really a fan of it because an extra button just takes away more space and you already have the play/pause/seek so next or previous. And then a stop button would be a bit too much. I believe we implemented but put it away <inaudible>, also the swiping was also an option. And that's always a bit of a problem actually, because if you put to many targets and to many options it can result in a feature creep also with issues with the app in the future. If you have more combinations of wholly adequate work and got options, you have an exponential growth of problems that could also reside from different combinations you cant test. So I learned from the side project, I'm working on, I'm working with a launch, of a company, and there we discovered that, even with user request teachers, you also have to keep in mind, then you follow some core values or focus of your project and you cant listen to everyone or all users. And in FOSS projects, it's a bit different because everyone could contribute and work on it. You are kind of more open to implement features, even though you personally think they are, shouldn't be in the app because you don't have the same control in a sense. I mean, you could say, okay, I wont implement it, but if somebody else decides to do it, it's more likely to just say, okay, well fine. If you do this, maybe I <inaudible> Yeah.

Researcher **13:13**

Right. Okay. You already mentioned that there may be kind of a lack of control for a lack of a better word as well here in FOSS projects. Do you then in Jellyfin have any like usability guidelines that you use? I know for some other FOSS projects, they have those, Gnome for instance, has pretty strict actually usability guidelines, and i know Mozilla has some pretty actually descriptive ones. Do you have any like shared resource that you, you all go to?

#8 **13:46**

Not really well I'm, I'm pretty new still to the project only a few weeks, I guess. But mostly if you decide on adding stuff or if you want to include stuff, it's a discussion with maybe me and other developers. And <name of project member> is also working on the Android side. So im in close contact with him, but also inside the chat groups.

Researcher **14:12**

Yeah. So that's, that's that discussion primarily reside in, in matrix, I guess.

#8 **14:18**

Yes. Yes. So we don't have a document in that case. it's very, it's a democratic process, but not written down yet. I don't know if we would do something in the future like this.

Researcher **14:31**

Okay. Okay. Yeah. That makes sense. I think that's the case for many, many FOSS project s really that they kinda, yeah. Not necessarily follow any strict guidelines in, in that sense. I think maybe Gnome and some of the larger projects might have have those. Do you think that scales and that sense that the size of the project also kind of determines?

#8 14:57

I do think so because some enormous, such a huge FOSS project and so many users, even the Linux kernel as a lot of guidelines,

Researcher 15:06

Right? Yeah. That's another,

#8 15:08

The bigger a project gets eventually you do eventually need guidelines or a better focus, or the people who control it, I mean, in Linux you have to subsystem maintainers who decide what makes it in and what doesnt.

Researcher 15:21

I can imagine

#8 15:24

... it will spiral out of control. You need such guidelines. But for Jellyfin its still very early. We are also rewriting a lot of stuff, from before, from Emby.

Researcher 15:33

Yeah. Yeah. It makes sense that at some point the like the discussion is healthy, but then if it's too much discussion, you don't get anywhere. So yeah, that makes sense that it would scale probably let's see. Oh yeah. Do you have any kind of systematic testing in general and, and how do you go about testing any changes you make?

#8 15:56

Right now in Jellyfin, we don't have actually any tests, so unit tests, but the other clients does have some tests because <inaudible> Is easier to test. The problem in Android is generally that your <inaudible> tests and integration tests a bit of a pain, and always have been. So if you're, I believe a lot of projects don't do tests, but I've learned in the last year or two trough stuff i have learned at university, to write better or write tests at all and for another library i wrote, thats a preference library im also using in Jellyfin. I wrote a lot of tests now because I have a lot of regressions and bugs that shouldn't have, and tests help a lot also on the.. <inaudible>, i did a lot of testing because if such code breaks, with many users you get problems.

Researcher **16:50**

Yeah. Yeah. It makes sense that it's a, it's kind of a good, a good thing to go through. Yeah. You were saying

#8 **16:57**

Next problem is since it's a web-wrapper right now you can't have too many tests. I don't even know if the integration tests from Android work on the web content, i believe not, Jellyfin-web has some tests, I think. And also the, the Vue client, which was currently in development has few tests already. And we definitely plan to do more test driven development, but at such early stages it's probably something the maintainers just don't want to bother with.

Researcher **17:32**

Right. Do you think in, do you think in FOSS projects, like testing in a systematic way kind of inspired by like best practice testing in the industry. Do, do you think it's like too much of a resource hug in that sense? Or do you think it's really beneficial in general?

#8 **17:48**

It would be beneficial but it is a resource hug and you can't force people to work on it if they dont want to.

Researcher **17:54**

Right. Okay. Okay.

#8 **17:57**

If you're a bigger project probably, with the guidelines you can say, okay, we want tests or you can do nothing. I recently fixed a small bug inside AndroidX, so a google library. And they already, even if it was a one line change, wanted an extra test just for this part, just have less regressions in the future and they have very strict guidelines.

Researcher **18:20**

Yeah. I can imagine that seems, that seems really...

#8 **18:22**

But I kind of wonder how this bug even got in, because it was like a very simple list index bug, where it's reading off the list and checking the index wrong or something the tests just didn't come to this problem.

Researcher **18:39**

Yeah. That is, that is kind of an interesting, yeah. Let's say distinction between FOSS and other projects in that sense. Yeah. But I guess yeah. Financial incentives obviously also play a big part in that.

#8 **18:52**

Google is very interested in keeping it stable, as they, are suppliers for huge companies and other developers.

Researcher **19:01**

A kind of a different question. I don't know if you know, but do you have any people in Jellyfin that are contributing only on UX and usability? Or is it all developers currently?

#8 **19:20**

Mockups, not really, but we have some testers and some <inaudible> people I believe so they kind of find bugs and help a bit with the user communication. Also not people in the project, but outside of the project who help a lot if they give feedback or connect with users that maybe.. I'm not as good in such departments, but like mock ups or something we don't really have. It's more like a dev, creates something maybe even a UI design. And then we decide if we like that or not, most times we like it as we share a very common view on UI.

Researcher 20:06

Let's see here. You mentioned that you have some, some users that help as well with testing in mind, are those kind of the same returning core users or whatever we, we can call them?

#8 20:23

Yeah. You have some more active users. Of course. The metrics generally sheds for example, there wasn't people who often provide feedback or ask questions some are a bit to active sometimes, but generally most are <inaudible>

Researcher 20:40

Hmm. Well, I just forgot one question in relation to the previous one about if you have any UX professionals, do you feel that the project could use people contributing? Let's say only. What could be like Figma prototypes, for instance, do you feel that the project could use that or do you feel it's doing fine without them as it is?

#8 21:02

I think it's currently doing fine, but maybe it would get better if you had such people, I can't really imagine it.

Researcher 21:14

No. Yeah. Yes.

#8 21:15

If we had some people like this could it could help more with more input. I mean, even if they say, "okay, you could do this better" or maybe a little bit better and you agree to have the yes.

Researcher 21:31

Okay then. Yeah, that makes sense. Let's see here. Do you know currently, if in the Jellyfin project, is any let's call the methods being utilized between contributors and end users with a focus on usability? I know you have a blog for instance, that could be one way to do that. Or do you have like a monthly Reddit post or something where you, like, I don't know if you, if you're a, at all involved in that part of the project, but I don't know if you have any..

#8 22:04

Not too much, actually. So I do follow the blog a bit, but not as active and also the Reddit community. Sometimes you occasionally see a post if it is shared within our internal groups. I don't know follow that actively though. Dont know if i can help in this regard.

Researcher 22:22

No, that's totally fine. It's just, if you had any, it is kind of some, FOSS projects, I guess it also depends on what the FOSS project is really developing for and doing, for instance, blender has these open movies. I don't know if you've seen any of them or heard about them and they are, you know, kind of a way of testing the product or, you know, involving community and the creators of the, of the project and kind of creating something. And in that process, creating new issues and, and features and stuff that would be nice to have

#8 22:59

Also a showcase for the project

Researcher 23:01

Exactly. And also showcase in the same at the same time. Right. That is just..

#8

It always helps if the developers of the project also use it personally for such projects, for example. For Jellyfin we have the advantage that all devs use it themselves. It's a, it's a media system. So it's specialized in that regard.

Researcher 23:28

Yeah. I also think I, in my research so far, the only real like method of, or so with the focus on the usability and that sort of thing, I think Blender and the open movies are probably the best example I found it is kind of unique, but also again, Blender is very specialized and targeted at one or several things in the, in the same field.

#8 **23:51**

May I ask which projects you looked at as well?

Researcher **23:56**

So, so the ones I've, I've been kind of going through currently, it's primarily Jellyfin and I think that's very likely because I think it's in a very interesting place, after the whole fork from Emby and kind of establishing what the project now wants to be, really. And I know still you're fixing lot of spaghetti code leftover, leftover from Emby. And I can definitely see that in the in GitHub as it is, but I think it's a, it's in a really interesting place, Jellyfin and it gained so much traction so quickly, but yeah, I also looked at like Gnome and Mozilla and stuff, but they are really, you know, dinosaurs in the FOSS world. with a lot of experience. And also I find maybe the projects that might need some more usability kind of guidelines or help, since all developers in FOSS projects, kind of have to take a stance on it at some point almost. Is some of the smaller and newer projects. Maybe Mozilla and Gnome and all these big ones where they have really detailed UX guidelines and decisions are like really tightly regulated, I think maybe wont need the same kind of assistance. But yeah, I looked at the, yeah, at Mozilla and then of course, Firefox, Jellyfin, Gnome, Blender, some smaller projects that was posted on Reddit, like really small. It was like just a couple of, developers then have a single interview from a project called Taskcafe, which is a, like a Kanban boat kind of Trello type thing, self hosted as well. And also looked at the NextCloud and then LibreOffice. So yeah. And then I'm also including studies that have been done previously. And part of, part of this is also verifying those findings since it's like, some of them are from 2006 and I think Bugzilla, which was what was used then is quite different from GitHub

#8 **26:01**

I believe Gnome also migrated to Github. So yeah.

Researcher **26:04**

Yeah, yeah. They did. Yeah. I'm pretty sure they did, even though they were stuck on Bugzilla for, awhile or maybe it was Gitlab they migrated to. Yeah. I think it was GitLap. It is here as I said in my notes, but there's actually kind of a, it's a good segue-way to my next question. Do you feel that GitHub is sufficient for all types of issues? I noticed for instance, that you have feature requests separately to, GitHub. Do you find that GitHub has some, some limitations and especially with usability in mind, maybe certain issues are harder to keep track of or the complexity can be an issue.

#8 **26:48**

I personally like Github alot, using it for all my projects, actually. I have some experience with JIRA, also GitLab but in the end I prefer the UI and the usability of GitHub. We have another feature request platform, but I believe it's only used for the server and on Android, we share GitHub issues for both feature request and bugs. So, yeah. And the labels and stuff help a lot. So we also have templates now that users can fill out. And most times that works quite okay. That we don't have to ask the same questions over and over again, like they already say which implementation they are, they're using. Right now we have 3 of them actually, one is the web client implementation then a native one with Exoplayer, which is a bit bugged at the moment, and then external player support.

Researcher **27:55**

Okay. Do you then think that for instance usability issues, are they more complex than just normal, let's say a, you mentioned a simple fix that you made in the other project there, where you mentioned Google and that's that kind of bugs, it's like a simple, you know, in code bug, did you feel that usability issues are more complex then in the sense that they maybe have, you know, several places in the code that needs to be changed or several screens or a whole flow of, of some kind of usability issue needs to be changed?

#8 **28:32**

I mean, usability you would have to spend more time thinking about it, planning your changes. But if you have a clear view of them and understand what's needed, it's probably easier because you can verify and test yourself. The hardest bugs are those you just can't test on your own device, compatibility issues with other devices or some random stuff occurs, because they have certain media type or certain video file that doesn't play on their device, but something similar plays on yours, or bugs with libraries that you cant easily diagnose, so its maybe in between. It's not easy, but it's also not as hard as some, as a bugs.

Researcher 29:22

Right. And then if you have something that require like a larger code rewrite of something, how, how do you manage that or distribute that in the project?

#8 29:35

So you mean a cross mulitple..?

Researcher 29:37

Yeah. Yeah. I don't know really what's a good example, but, but some of the larger rewrites that needs to be done, I'm guessing for instance, one example could be the kind of ongoing process of trying to, you know, fix all the, the legacy code from, Emby. How do you, how do you, how do you manage that? Is that different or is it simply to spread out into a smaller issues then?

#8 30:06

I can't directly talk about server implementation because of what im working on.

Researcher 30:10

No, I'm not sure, but I don't know if you have any equivalent in the Android client. That's kind of like a larger overarching issue that you're trying to..

#8 30:19

The Android client, which was based on Cordova at first. It just didn't feel the right so i created a new project and stuff. Right now it's mostly smaller patches that we do. And each of those are only a few files that are changed.

Researcher 30:42

So, so the, the older client than was, it just didn't make sense to salvage anything at all. It just was quicker to..

#8 **30:49**

The big problem was that it just used another build system, Cordova, so it was build with Yarn and then that called Gradle, and importing into Android Studio was a pain because first you had to generate the sources for Android, then you could open it in Android Studio. And if you did changes to the code you first had to sync it back to the main repo, and then we reapply it to the Android project, and it was very complicated. And now we have a standard Gradle system and also we don't bundle the web-client anymore. So we load the web client from the server and directly inject custom code. That makes it a lot easier.

Researcher **31:34**

Interesting. Okay. I also have some questions here about kind of the social aspects. We already touched a bit upon those also a bit about the resource management. I don't know if it's even a possibility for you. I know most of the people are in the US and, and such, but have you ever taken your work offline or do you do some kind of social activities in the projects?

#8 **32:01**

No really actually. So as I said, I'm quite new. Most devs are from the US, but we have them spread around the world. People from asia, netherlands, US, Canada, even Japan. So yeah, a local meet up wouldn't really work, but we have talked internally about maybe doing something like online or whether it's something like Discord and do a coding night.

Researcher **32:41**

Okay. And also again, you mentioned that the social bonds are a motivational factor for you in the project and in FOSS in general. And you also mentioned that you have some of these external users that are not technically project members, but might be more invested than the average user would be. So providing feedback and such, and then kind of an overarching question regarding both of those things would be, how much of the project would you say is driven internally and then how much is external influence? Yeah, again, it's kind of like a large like broad question. It doesn't have to be super specific more of like, you're kind of feeling for the, for that.

#8 33:28

I personally think a lot is happening internally, just because these are the most active groups. If people are really active, you put them into those groups. I mean, it happened with with me and happened with other people, but I also can't really say because I'm not seeing the whole picture. I'm very focused on Android, and the Android TV client, but server and very client specific, and a lot of feedback comes GitHub, or Reddit. I'm not active in these spots, but I believe most is currently driven internally from my perspective.

Researcher 34:12

Yeah. And then if you had to assign all the resources you could into fixing one issue in the project, what would that be right now? If it was up to you?

#8 34:26

Well, regarding Android, I really want to get the ExoPlayer implementation working properly. So the problem is that our device profiles, we are generating, so which codecs and which container formats that are supported. It is taken from the old app. And that was, or not even the old app but a PR that was made to the old app and it was broken then and it's still a bit broken and we don't fully understand the code. So it's a bit complicated to debug these issues. So like, if you define that, you're trying to transcode or play it isn't actually supported by the form, but because the profile is wrong and it also happens that you get a transcode and then you get no audio, or other issues. So I personally would really like to fix those issues so that we can roll this out as a fully native final player and replace the webplayer, we also need it in the future. If you do the native client, because we're building a UI and the libraries and such stuff takes a lot of work, but it's easier to test, but supporting all the cortex and supporting containers and the video playback actually works, which is important for a media system. That would be something that I really like finished or done. Yeah.

Researcher 36:00

Yeah. It makes sense. I was also wondering, you mentioned, you mentioned for instance, if you'd take over some part of the project, let's say in the Android repo where you currently are working and you have part of the code, you don't understand, do you, how do you, how do you figure that out? Do you just like work on it, yourself, or do you seek help in the chats and are you maybe even educating other people in the project yourself as well?

#8 **36:27**

All of them actually. So mostly in the rewriting phase. I wasn't asking a lot of questions inside the groups. I wasn't really familiar with the internal of the server or webclient, and especially JavaScript is not my strongest suit so i was asking questions, how all these clients work and how i can inject my content property. And I get a lot of help in these parts. And well, I kinda think that it also helps us with doing things correctly. If I'm already reviewing the pull requests that we get, that this, I guess, is a strong indicator to better understand the code. Yeah. Core documentation also helps. Try to document stuff thats not immediately obvious.

Researcher **37:28**

Yeah. That makes a lot of sense. I think we also kind of talked about this already, but I just want to get the exact question out there anyway. Resource allocation, like how do you manage that precisely? Is it first come first served? Or for instance, you're a, you're a bit more of an expert on, on Android development. Is that just basically "you come in and say, Hey, I know Android, let me handle that". Or how do you manage resources? Is there someone in the project then that kind of allocate that if there's something that needs to be done or?

#8 **38:04**

So I was getting into project, by just rewriting the old client. Now, i pretty much have a maintainer spot. So I decide which stuff goes in, and which doesn't. With other people, as I said, Niels is also very active on Android client and helps me alot. <inaudible>. Personally I've worked on the parts that I want to work on or I see as important. So currently the idea is to get the stable version of 2.0 out so we can release it to everyone. Because right now that's <inaudible>. And once that's done, I want to structure the app a bit more so that we can then continue and move forward with Exoplayer and start with the native implementation.

Researcher **39:16**

Yeah, sure. So, so in that sense, it's also, I'm guessing when you were invited to be a project member that was of course based on your already existing contributions. So that kind of also determines what you end up doing in the project then.

#8 39:34

Yeah, i pretty much asked if it was planned to rewrite the client, or how we should continue with our current client and their voiced it and said "Okay". It would be nice if we got something different, but somebody <inaudible> then I just did it and said, "okay, here's the repo can you take it up-stream?" and well, merge it into the project and then it was pull it into the groups and internal groups shortly after. And i just said we have a new client and you can work now work on it.

Researcher 40:07

Okay. Yeah. That's really interesting. Yeah. Two questions left. If you have any, let's say concept, it can be for instance, usability. And you also mentioned you have some of these external guidelines for Android. You have, you've looked up once, once or twice, I guess. Do you ever look up any academic sources like papers or books or something? I know you're already a student, so you probably already have some available to you. And what form do you prefer to consume content? Is it video? Is it texts and, and really what, what would you prefer if you, when you look up something?

#8 40:49

For Jellyfin, I've only looked at the documentation. So like I've seen the google documentation, the material guidelines and such. And I personally prefer text for most of the things, because I can skim over them quickly and get the gist of the content or read through it, and get the details. With video it sometimes feels a bit well bloated to me, but theres also sometimes content, which is really nice if you need to see something like the new Jetpack Compose framework. I don't know if you seen it already. So Google is building a new UI framework and I plan to use it for Jellyfin and it's still unstable and in beta, but I kind of want to use it. So I was watching a video from Google on this topic. It explains it very well with graphics and shows it in practise. I do also like video, but generally I'm more of a text guy. Regarding the scientific papers, not necessarily for Jellyfin, but I did have some experience when writing a paper, a small paper for University, well now I'm more into this scientific thinking. And if I see a paper with which interests me i also read it.

Researcher **42:20**

Yeah, it is a, I think it is a, there's so much, really great knowledge in academic texts than papers, but it is not the most approachable format in the world. I'd say you'll have probably a benefit there of being a university student. You're kind of inclined to, to know where to go.

#8 **42:41**

I used some, some platform, which is only open for universities. To write my own paper was about AR and interaction. So I use a lot of papers and since I didn't do any research on my own, I had to look up a lot on this topic cause before I rarely look at such papers because they're just not so approachable and are often walls of texts

Researcher **43:15**

And you really only need the top three best points or whatever from the yeah. Yeah.

#8 **43:21**

And you don't think that you will find exactly what you need and its just too much text. But if you do give it a chance, most times helps a lot.

Researcher **43:34**

Yeah. I definitely agree. And that's also, I think one of my goals is try to condense the current. That's actually quite a bit of studies on usability and FOSS and how they kind of interact those two concepts and in real life. And, but again, I'm trying to kind of condense that into some resource that can actually aid in, in making these decisions. So yeah, but then yeah, my final question is which is quite kind of a, like a broad question. What would you say is the largest challenge in having strong usability and all UX design for that matter in FOSS projects currently? And how would you, how would you address it?

#8 **44:18**

Hmm. Well, to have to have a common taget across all developers thats important. If one person does one thing and another person does another thing it doesn't really fit together, but only creates issues. You need a common design language and a common experience language in a sense. So I'm heavily orienting myself for writing the new client at the current webclient, but a bit more touch friendly. And also the Vue client does something similar. So we try to have generally a similar appearance so that you have first, maybe the home screen with general overview of all your media and sub categories and sub collections and such stuff in a way structured that you can find what you need quickly, regardless of which platform you're on. It's very important because of the user, switches devices all the time and has almost a different experience. That's not helpful.

Researcher 45:31

Right. So, what I'm, so what I'm hearing you say there is so like a shared language shared terminology and a result of that would be a more like a coherent design then. Yeah. Yeah. Okay. Let's see if I missed anything. Just I think we got all of it. Yeah. I think that's actually, I think that's all the questions I have. Really interesting points. Very nice all around. And is it okay if I have like a follow up question at some point? I probably don't but, I'll let you know on Discord. If I have some, a specific question maybe for the repository you're most active in or something. It depends. Yeah.

#8 46:24

Also, if you, if you look at your recording transcipt, if you have more questions.

Researcher 46:29

Yeah, sure. I'll, I'll, I'll definitely do that. That's a really nice of you.

#8 46:32

If you have a result, i would also be interested in seeing that.

10.10. Feedback #1, Jellyfin

Researcher
So... Just about ready. You preferred text chat right?

#1
If possible, yes :)

Researcher
Totally fine with me. Saves me a lot of transcription 🙂
Okay. To start off, just to explain where I am right now with the project.

From the interview is have summarized a lot of information and the general most prominent take-away seems to be that in order for FLOSS projects to have stronger UX decisions being made, the most straightforward approach is simply to make more inquiries into projects (being more designers joining) and secondly, creating more documentation to support FLOSS projects with thinking about these decisions..
So..... How does one do that? I have tried to create a piece of documentation that is very much still a prototype (so excuse any spelling and missing formatting).

https://github.com/**Researcher**/FLOSSUX

The idea is that the findings from my study will fuel some discussions in FLOSS projects (i'm going to approach some more projects i think) and these discussions will be logged and become part of the documentation piece:

So think of a fully open-source UX documentation that goes into the high-level ideas and considerations that might help a project become more organized and oriented towards action when it comes to design.
(where anyone can contribute a discussion that will make it into the documentation)

#1
That sounds like a good idea 🙂 Kind of like those Material Guidelines, but more generic and high-level

Researcher
Exactly. My general finding is that there are plenty of technical documents that goes into "how many pixels this, how many pixels that", but almost none that equips one to think about these things on a higher level.

So first of all. Do you see any potential problems with the site? It can be the idea itself, the template or something else? Do you see developers using something like this?
https://**Researcher**.github.io/FLOSSUX/

There is the direct link btw

#1
The idea is good and the design is clean and easily searchable, so that's good :)
I can see devs using that kind of resource. I'm often looking into how to present specific things, so a resource that helps me decipher how to best organize a screen or a section of the app is always welcome
As you said, it's often hard for us devs to figure out higher level decisions about UX
I can look at the best practices for typography or something, but that won't help me organize a settings screen, for example
I'm taking that as an example, because it's something we're dealing with currently on the new Vue client for Jellyfin
So something going over the larger concepts of UI and UX design would definitely help
And I can see a lot of other projects which don't have access to a UX team referencing such a resource

Researcher
Right. I also plan to have links to some existing design resources that can also help to further these things.

Interestingly i noticed, just to take a small detour, that while i was writing and working on this you have received a number of what seems like pure design decisions in jellyfin-vue, in the discussions tab. They seem quite well received. What was your initial thought about that contribution type? Did that work for you, or did you feel that it was basically now up to you or someone else to make it real?
*contributions not decisions sorry

#1
We generally like that kind of submission, when done that way (If we don't know the user)
As in it's presented more as ideas and suggestions than as a request
We're very open to discussing proposals like these (We've discussed a bit with the person in the dev-client matrix room since then)

Researcher
Very cool to hear. This was very much what I was thinking about when initially interviewing you, and the findings very much show that what you are doing in jellyfin-vue is aligned with what needs to be done to further UX.

To elaborate a little on why i asked:

https://**Researcher**.github.io/FLOSSUX/keyfindings.html#democracy-and-consensus

Would you say you agree with the points in "Democracy and consensus" with those contributions in mind?

#1
I would agree with your assessment

There's a kind of sense that we spend a lot of time on code, so some of us can think it's "more important" (After all, we build the final product, that sort of makes sense)
It's a bit difficult to tackle if developers aren't open to changes, like on some projects
Even on jellyfin-vue, there are conflicts on which direction to take the UI in, often based on nothing more than gut feeling
Researcher
That is the challenge of UX. Often times it looks like this

https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcS1pNKqODXBtutWa6THr5SHHEe9W3OkX-vrMA&usqp=CAU
But discussions, iterations and debate is bound to solve some of that challenge.

#1
lol I can imagine

It's a bit easier if we know who we have in front of us, if that helps
Since most of the feedback we get is from regular users making demands, a legitimate designer chiming in might get lost in the noise if they're not clear about who they are and what they're trying to contribute

Researcher
Totally agree, great point and it seems like it fits right in the knowledge management section! Actually the "next" part of this was that, if you want to, to read rough (take all the time you want) and skim through the key findings overview (or the expanded stuff if you want) and create a discussion 🙂

This is totally on your own time and it doesn't have to be today. Just whenever you feel like it.
How do you feel about that?

#1
Sounds good 🙂 I'm gonna give it a read and come back to you :)

Researcher
So cool! Really, again many thanks for wanting to participate! And i'm thinking it can be literally anything. A point on something, rewording, a story, graphics, disagreeing. Whatever you think is valuable to point out 🙂

#1
You're welcome 🙂 And thank you for doing this

Researcher
It's my pleasure. I hope it gets just a little traction 🙂 I don't want this to be just another academic paper that sits behind some paywall database somewhere

#1
Well, it's bound to have at least some influence on how we do things on this project, since so many of us participated in the study :)

Researcher
That's great to hear! That is a nice meta-finding. The intervention into projects themselves are also a conversation starter.

#1
And I know there's a desire for better UX among the Jellyfin devs. It's just that the project is kind of an anarchy in how it's run, so it can be difficult to push things through....

Definitely

Researcher
Yeah ties abit into

https://**Researcher**.github.io/FLOSSUX/keyfindings.html#the-developer-playground

The notion that its a project that is done largely for fun on developers own terms, but suddenly overnight caters to 1000's of people. Creates a weird discrepancy between need for very organized work and running a community in your free time

#1
Exactly ^^
It's been a big pain point of mine recently :p

Researcher
I can imagine!

#1
I feel like we're at the point where some more organization is needed, but it's still kind of free for all with some far off invisible hand guidance from the core team.... 😛 No real written plans, no roadmap or anything
I think it's detrimental to the growth of the project, but that's my opinion :p

Researcher
Yeah. That's a tough spot. Jellyfin is also unique in the sense that you had insane growth. You really filled that vacuum in your niche between plex and emby as being the only true FLOSS project. The growth might be faster than the collective initiative. Hopefully it catches up a bit
Hopefully the -vue repo can focus on its own little roadmap for the time being

#1

Yeah, it's been a bit of a journey lol Since I joined about a year ago, we've had like 10 new team members come in, and several dozen unique contributors

That's the plan :)

10.11. Feedback #5, Jellyfin

Researcher, [04.02.21 13:18]
Hi **#5**
First of all thanks again for participating in the interview and helping my study with empirical findings. Here is a little "status update" regarding the project. As we talked about the deadline for the paper was the 4th of January. However, due to Covid and some other personal matters the deadline has been moved to the 6th of April. I am currently working on the thesis itself, but also on the design and "solution" if you will. I was wondering if you would be open to talking/chatting about some feedback for said solution within the next couple of weeks?

Best, Daniel

#5, [08.02.21 22:23]
Hello Daniel and sorry for the delayed reply.

Yes, of course, for whatever you need! I'm with exams (I end them next week) but feel free to ping me any question you have here as you did last time, and I will answer it as soon as possible.

Hope you're recovering well and your work is going good

Best, **#5**

Researcher, [09.02.21 08:13]
No worries :) how does tomorrow, Wednesday the 10th Sound? It doesn't take more than 15 - 30 minutes and we can just do text. 🙂

#5, [09.02.21 14:37]
Yes, of course!

Researcher, [09.02.21 15:28]
Great! I will write to you then :)

Researcher, [10.02.21 14:30]
Im working on the questions and methodology for later. Is tonight fine? And we are in the same time zone I believe right (gmt +1)?

#5, [10.02.21 15:29]
Yes

Researcher, [10.02.21 15:41]
👍

Researcher, [10.02.21 17:54]
I'm just about ready :) What about you?

#5, [10.02.21 17:54]
Go ahead whenever you want

Researcher, [10.02.21 17:59]
Okay. To start off, just to explain where I am right now with the project.

From the interviews i have summarized a lot of information and the general most prominent take-away seems to be that in order for FLOSS projects to have stronger UX decisions being made, the most straightforward approach is simply to make more inquiries into projects (being more designers joining) and secondly, creating more documentation and discussion to support FLOSS projects with thinking about these decisions.

So..... How does one do that? I have tried to create a piece of documentation that is very much still a prototype (so excuse any spelling and missing formatting). Here it is:

https://github.com/dani763f/FLOSSUX

The idea is that the findings from my study will fuel some discussions in FLOSS projects (i'm going to approach some more projects i think) and these discussions will be logged and become part of the documentation piece:

So think of a fully open-source UX documentation that goes into the high-level ideas and considerations that might help a project become more organized and oriented towards action when it comes to design where anyone can contribute a discussion that will make it into the documentation. When i say high-level documentation, I mean trying to instill a mindset of thinking about questions like "what motivates me as a developer?", "Who are our users?", "What are our project goals in terms of design". Not specifically how many pixels wide something has to be or which font to use et cetera.

So first task. Go ahead and read the home page and let me know what your initial gut feeling and take-aways are. :)

#5, [10.02.21 18:04]
I already saw all of it because [redacted] sent the link to our rooms :)

#5, [10.02.21 18:04]
Really liked your work

#5, [10.02.21 18:04]
Let me do another review of it though :)

Researcher, [10.02.21 18:06]

Ah okay :) Yeah take your time, and don't hold back. Let me know if something doesn't make sense or something!

#5, [10.02.21 18:19]
About the home page, I agree with all the points you mention. I have some stuff to add in point number 1 through: Discussions and guidelines of projects should be understandable at most levels of what an end-user is. Most of the people who are used to working on open source software have a notion of what a code of conduct and ideas on how a project might work, as most of the open source projects share most, if not all, the code of conduct rules and way of contributing. However, people who don't necessarily work on open source software frequently, or it's their first time, might not be aware of the procedures and way of working with tools like git.

The documentation must be accessible to these kinds of people. However, it's also people's responsibility to have the willingness to take a look at those before contributing if they're not used to the way of working in the project. As we discussed in the last interview, most of the people that are not aware of this and need this documentation are people who usually come, do one thing, and go, and they usually don't pay attention to those documents, as their goal is clear: get whatever they want fixed upstream and wait for the next version.

People who are inside a project team and know each other are usually aware of the way of working and most decisions and this is also a reason why I think this kind of documentation is usually neglected.

#5, [10.02.21 18:20]
As for good documentation examples, design-wise of course the best is to look to the SDKs or documentation that the designer provided: material.io if you want your project to follow Material Design guidelines or Apple's guidelines if you want to follow Apple ones

#5, [10.02.21 18:22]
As for "project goals" documentation, the best-in-class I ever found is the Telegram's one for it's support initiative: https://tsf.telegram.org/ and https://tsf.telegram.org/manuals

#5, [10.02.21 18:22]
Those show the clear goals, without roundabouts and extra unnecessary stuff, of why Telegram exists, why Telegram relies on volunteers for support and why they believe it's good and how to proceed and the way of working inside it

Researcher, [10.02.21 18:26]
Nice links thanks! You already are mentioning some really good points! Which leads me to my next question... Could you, over the next couple of days(take as much time as you need) and try out the procedure? Find a point or two on the website that you have a new angle on? It can be a little example of how you think about a concept or how your project specifically dealt with something. You might also disagree with something if that's the case. This is very much a design-through-use experiment and as such it's nice to get the ball rolling.

Does that sound okay?

Researcher, [10.02.21 18:30]
The key findings overview is what i'm thinking about specifically

Researcher, [19.02.21 10:31]
Hey **#5**! Did you get a chance to create a discussion on Github or are you still reading through the stuff? 😁

10.12. Feedback #6, Jellyfin

Researcher
Hi [redacted]
First of all thanks again for participating in the interview and helping my study with empirical findings. Here is a little "status update" regarding the project. As we talked about the deadline for the paper was the 4th of January. However, due to Covid and some other personal matters the deadline has been moved to the 6th of April. I am currently working on the thesis itself, but also on the design and "solution" if you will. I was wondering if you would be open to talking/chatting about some feedback for said solution within the next couple of weeks?

#6
Yeah sure, I can talk about it over the next few weeks. I'm currently at uni, so I'm free 90% of the time, just let me know when you'd like to talk

Researcher
Great! How does Tuesday the 9th sound? At say 13:00 your time?

Researcher
Today is not the best for me, and i'm assuming uni is keeping you busy :) When do you have time during the week?

Researcher
Hey again Cameron

Assuming you are still busy. Long story short i would like to invite you to take a look, and create a discussion on my page https://github.com/**Researcher**/FLOSSUX

The project is about community discussions on UX in FLOSS projects and the initial findings are based on the interviews I had with you guys. A discussion can be about pretty much everything, so if something resonates with you feel free to add on about an experience you had or how you tackle a specific problem.

Thanks again!

Researcher
Slight link change

https://github.com/**Researcher**/FLOSS-UX

10.13. Feedback #7, Jellyfin

Researcher
Hey Joshua 🙂 Does today still fit?

#7
hey
yep
i'm free now

Researcher
Great 🙂 Had time to create a discussion yet?

#7
oh crap no i completely forgot about that 🤦‍♂️

Researcher
haha no worries 😛 We can just discuss now instead
Ready in 5 minutes, just need to get my stuff in order here

#7
alright, no rush 🙂
i'm just drinking my coffee and reading the news haha

Researcher
damn that sounds good ☕

Researcher
Do you prefer voice or text?

#7
text might be easier for me today 🙂
if you don't mind it

Researcher
Sure thing :)
Just means i don't have to transcribe so fine with me!

#7
haha awesome

Researcher
So... I already "spoiled" it a little bit but with regards to my github repo. But the TLDR version would be this.

A high-level UX documentation piece, through discussions contributed by FLOSS communities. participants, designers or whoever will educate and inspire projects to look at their own process and ways of working with design.

And by high-level i mean not dealing with minutia UI details such as pixels-this margins-that, but rather "Why is talking to our users important?"

That type of questions

#7
ok

Researcher
Through the findings it seems quite clear that, for instance, in jellyfin-vue that most of those details are abstracted away a bit by using vue and by extension vuetify.
First a little detour actually.

https://github.com/jellyfin/jellyfin-vue/discussions/749

Have you seen the design contributions over at jellyfin-vue recently?

#7
hmm, actually no lol
vue has sort of become its own little world within the project. I check in from time to time and see what has been happening, but for the most part it's self-contained

Researcher
right, i gathered as much. Just to summarise, since i interviewed you guys the first time some actual "pure" design contributions have come through. As you can see they take the form of mock ups and discussions. Very much related to what I was asking about the first time around. Skimming through these, what is your take on that type of contribution?

#7
I think they're extremely valuable. Though it has caused some friction, I like that many people seem interested in the design aspects and are willing to voice opinions about it.

Researcher
Cool. Many of the findings that drove my idea for a "solution" seem to suggest that discussion is a large part of what will solve design issues in FLOSS projects at least. So its nice to see it happening naturally. Would you attribute this contribution maybe also to the fact that Github discussions has been enabled since then?

#7
I definitely think so. It was very awkward to have these sort of discussions before, as they didn't really fit in cleanly with any of the existing discussion places (e.g. GitHub Issues, or fast-moving busy Matrix chats). Early in the project we pushed a bit to get a separate forum

going for these sort of discussions, but that fizzled out - having a dedicated place on GitHub, where everyone already is anyways, makes it much easier to track and manage them without feeling like there's "another place we have to look"

Researcher
Yeah i agree, and as we also discussed initially, the barrier of entry is really low. Now a designer, with no knowledge of git or code can make visually heavy contributions that are logged and available as reference for the whole project. Its been a little surreal seeing these things happening in parallel with my own writing as its basically all my conclusions that are just manifesting like magic or something 😀

#7
haha indeed! it's a very positive change. we had a lot of good discussions lost in the flow of Matrix rooms so it's good to get them somewhere permanent

Researcher
-vue serves as a really nice blueprint on how to do things right i my eyes, having a low barrier of entry, creative open discussions and iterations back and forth on what needs to improve.
https://**Researcher**.github.io/FLOSSUX/

So.. Here is the page I made. Very much still a prototype. Just skimming through the home page does the process seem clear and concise?

#7
yup
i've also been reading through it as we've been chatting, i like it

Researcher
Nice, thanks 🙂 How do you feel about the "high level" concepts? Did you expect something more down to earth or practical?

#7
I think it makes a lot of sense, and is about what I was expecting. I agree that too often things get bogged down in the tiny minutiae and that distracts from the real goal
how many pixels something is, or exact colour pallets, or such, get in the way of "how can we make this better for a first-time user", "how can we make it easy to navigate", bigger Q's like that

Researcher
Some of the concepts also are reflections on why FLOSS participants are motivated. Or say, the perception of design as a field. With this in mind, I hope that projects can also look inwards and stay critical of their own efforts and maybe understand their own biases towards different types of contributors and such. What are your thoughts on that? As i recall you didn't really have that many "process" discussions and such or?

#7
I think that's an important part. Without getting into details, the "friction" I mentioned earlier was actually revolving a lot around design, and how to handle differences of opinion. I think keeping critical of our efforts is thus really important, or it's easy to fall into a trap of only valuing the implementers over those who want to design. Working towards processes and specifically some firm decision-making guidelines (i.e. a project constitution) has been a major focus over this past week especially, and for a bit before

Researcher
Very interesting. So an actual constitution, in the sense of some authoritative mission statement type thing?

#7
correct!
modeled a lot off the Debian constitution
to sort of provide a written, authoritative document on a lot of the official relationships between various people in the project, define the terms, set out "who is allowed to overrule who's decisions",etc.

Researcher
Very interesting! Seems like a very good initiative. Process considerations also become more needed with scale i can gather from that

#7
yup indeed, that was sort of the impetus for it, feeling like we've gotten so big of a project that it's really impossible for any one person to keep track of it all, and wanting to really formalize how decisions are made. very often there would be contradictory decisions made, or one person (read: me) making a decision one day, literally completely forgetting what and why, then contradicting myself later
haha

Researcher
I can imagine the backend portion of jellyfin use the discussions tab for totally different things then. Has that been used as extensively in terms of discussing decisions then? Or what is it being used for there would you say?
*discussing

#7
That is a good question. to my knowledge it isn't being used all that much by the backend team, but it seems users have found it and are asking questions there
at this point the backend team seems to have a few people who are all doing their own things, generally unrelated to each other
which has some downsides, and some upsides

Researcher
aha i see

My design is very much a "design-after-design" sort of problem in that its affordances or its use is very much defined by the people who use it, and thus the findings its can inspire is dependent on actual use of the prototype. Are you still up for creating a discussion on the Github repo? It can be like a story, an experience or some point you feel needs further explaining. You might also disagree with something! The name of the game is discussion so that is sort of self-explanatory 😀

#7
yea, i'm definitely game
i'll have to think a bit about what to post haha
and re-read it a few times

Researcher
Nice! Yeah sure, the key findings overview is the condensed version. Then each of those findings are expanded upon in the study findings section. It can also be a meta-discussion surrounding the page itself or its structure 🙂 Its a free-for-all really

#7
alright!

Researcher
and feel free to share it if you want! You are now officially a UX designer, congrats 😀

#7
yay 😀
i've been working on a web frontend for my own project, so feels good to be official 🤣

Researcher
Nice! Yeah, i mean according to the design books i have read everything is designed in some way, shape or form. By that logic we are all designers insert mind-blown gif here
A last little question before i leave you too it

Have you ever heard of https://opensourcedesign.net/

#7
haha indeed
I have seen that site before, just in passing

Researcher
Yeah okay, seems like a huge resource but not being used for much in actual projects at least for what i can see.. hmm

Oh well I will leave you too it :) Just let me know if you have any questions or something! And again, huge thanks for all your time!

#7

always happy to!
if you ever have an follow up questions, feel free to ask here - doing it in chat as opposed to a more formalized interview setup is definitely more my forte haha
i'm gonna give that site a read-through too

Researcher
Will do! Thanks!