

Managing Software Portfolios: A Comparative Study

Sebastian Draxler, Adrian Jung, Gunnar Stevens

University of Siegen, Information Systems and New Media,
Hölderlinstrasse 3, 57068, Siegen, Germany
{sebastian.draxler, adrian.jung, gunnar.stevens}@uni-siegen.de

Abstract. Software applications that can be changed, modified and extended are nowadays pretty mainstream. But only few researchers focused on the role of the users social network for actual modifying practices and hurdles. Therefore this paper, studies in a comparative manner, how users modify software applications by using markets of existing components. We examine two popular applications: the universal tool platform Eclipse as an example for work applications and the game of World of Warcraft as an example for leisure applications. Despite the difference of the contexts, we found common patterns in collaborative actions within the social networks, that lead us to discuss the role of sharing and support for modification awareness for end users.

Keywords: EUD, Collaborative Tailoring, Software Ecosystems, Awareness

1 Introduction

For a long time, research on managing software portfolios primarily focused on the appropriation of single applications. At a time, when applications had a clear border and when the software market was very limited this was well-suited. Examples are the work of Mackay [1] or Gantt and Nardi [2] who empirically investigated into tailoring efforts. One remarkable result of both studies was, how much collaboration in form of artifact or knowledge sharing they had observed. But since then the basic conditions have changed. Today it is often tried to establish so called *software ecosystems*. They consist of an open, extensible software platform that attracts different manufacturers and hobbyists, creating small-scale components, which can be individually assembled by end users [3]. Software ecosystems are an interesting topic to look upon, when it comes to study end user development. They empower the end user to choose or add functionality to their software by orchestrating pre-existing modifications. Compared to the situation of the 1990s, software ecosystems and the involved stakeholders are globally networked. We believe that this changes the users opportunities for modifying applications, since under these circumstances local networks of users (e.g. a company) collaboratively makes use of software ecosystems. Our first goal is therefore to understand: “*How and based on what information do people modify their personal software installations?*” To answer this question, we followed a similar approach as Mackay [1], which can be described as a set of empirical field studies, consisting of observations and interviews. Since we are not expecting an universal

answer, we chose two quite different software ecosystems to investigate into, hoping for contrasting results. First we analyzed how professional software developers modify their Eclipse installations during their day to day work. This was followed by a second series of investigations, that examined how players of the online role playing game World of Warcraft (WoW) modify their game clients during leisure time.

2 Two field studies on managing software portfolios

Our research process is loosely oriented on Mackay's [1] studies. For each study (Eclipse and WoW) we started exploring the relevant literature on the software, organizations and communities that are related to the case. This was followed by partially-structured interviews and in the case of Eclipse, on-site observations. The work is still carried out as open ended qualitative study. The material presented here was transcribed and the transcripts analyzed using coding mechanisms of the Grounded Theory [4] approach. While this is not a full grounded theory (so far we rely on In-Vivo codes), the approach has still proven very helpful to carefully analyze the material, without subsuming our observations under pre-defined categories from literature.

For the Eclipse case we cooperated with six software companies with 10 to 250 employees. At each company, we conducted at least two semi-structured interviews of at least one hour (altogether, we conducted 17 interviews. Additionally, we visited two of the smaller companies over a period of 3-5 days for on-site observation. For the WoW study, we interviewed a small WoW guild that is constituted by 8 to 10 active members of various game experience and with a different educational background. Whereby a guild is an in-game association of player characters formed to make the accomplishment of group-related tasks easier. We conducted at least two semi-structured interviews of 15 to 60 minutes with each player. Additionally, we recorded the changes in their addon configuration over a period of one month.

2.1 Customizing the Eclipse IDE (Study 1)

Eclipse is a multi-language integrated development environment (IDE) and an extensible software ecosystem. It began as a toolbox for the Java programming language at IBM. It was designed to integrate future tools under one roof, using a plug-in mechanism. The platform was made freely available, open source and steered by a non-profit foundation to attract other companies and other contributors.

Eclipse provides all of its functionality on top of a core runtime system and can be extended by using additional third-party plug-ins. The Eclipse core and most additional plug-ins come free of charge and are released under the terms of an open source license. An Eclipse plug-in is constituted by an XML description and Java code that supplies the functionality. There are about 900 tools available that consist of thousands of *plug-ins* called components. Tools are either installed using the included install and update mechanism or just copied to a folder manually.

Software developers as we interviewed and observed have to fulfill quite different tasks, from documenting requirements, modeling, coding, testing, debugging to talking to customers. For many of these tasks special tools are either needed or at least are a great help. Furthermore the resulting artifacts are often shared among other developers who work on the same or similar tasks. Eclipse allows to be extended by additional plug-ins that could support the task in question. Most of the observed and interviewed users were capable of creating such plug-ins on their own. But due to the amount of plug-ins that already exist on the global Eclipse ecosystem and the time and effort necessary to create a new plug-in, they chose to first search for existing alternatives that might fit their requirements.

One of the key findings of the study, when the need for a new tool arose, suitable recommendations regarding tool selection, installation, and configuration were sought out from co-workers who also found themselves in similar working contexts. We especially could observe this in environments where software developers organized themselves in agile teams. People did trust in their co-workers advice much more than in recommendations found on the Internet or in magazines. Within the observed companies, several related strategies were established or put into practice by accident.

If it was obvious that someone should be told about what plug-ins to install, e.g. if a new person joined a project team, either the plug-in names or even the whole set of artifacts were passed to that person. In case of a problem, people went to colleagues and just asked for advice which plug-in to pick or how to proceed if problems occurred. Unfortunately quite often it was not clear who could be an experienced colleague for a certain topic. As some Eclipse users were constantly trying to stay informed on plug-in related topics, they sometimes stumbled upon interesting news that could also be relevant for their colleagues. This was an interesting source for innovation for their colleagues. But as they did not consider chatting about new tools as a central part of their job, it was often unclear if news were important enough to be shared. On several occasions Eclipse users sat together at one machine to discuss a problem or how to proceed with a project or task. While doing so, the colleague did discover new icons in the Eclipse toolbar and so the topic moved to new plug-ins. The result was an exchange on new interesting plug-ins.

Overall we observed that Eclipse users tended to ensure personal information exchange on which plug-ins are interesting, how to install them and which problems can occur. On the other hand it was often unclear if general plug-in related news, tips, experiences should be spread and through which channels. Therefore this was often triggered by accident.

2.2 Customizing World of Warcraft (Study 2)

With more than 12 million subscribers the massive multiplayer online role-playing game World of Warcraft (WoW) is currently the world's largest game of this kind. WoW was developed by Blizzard Entertainment and released in November 2004. Three expansions for the game have been released since then, in addition updates of the game client are regularly released.

WoW is an interesting example for EUD in games, because it is possible for players to create their own addons for the game client that is used to play. Addons are

constituted by describing metadata, XML documents describing the user interface and the functionality, which is written in the scrip language LUA. The development of addons is officially encouraged by providing the user with access to certain game API functions. There are currently over 5900 user-created addons [5] for the player to choose from. Addons are installed by downloading a code package from the internet and then placing it in a specific addon folder in the WoW installation.

The game is designed in a way that people work together in groups to accomplish complex tasks in the game world. Players organize themselves in guilds in order to simplify group building and represent social networks. Although the game is designed in a way that players do not necessarily need addons, they play an important role because they can enhance the players or groups performance. Addon functionality can range from displaying additional information that is helpful to the player to automating certain tasks or reorganizing the built-in chat function. Not every player benefits from certain addons as the usefulness of an addon depends on the role that the player seeks to fulfill in the game.

Throughout the whole interview study, every person had modified his/her game client using addons, although this was not a selection criterion. Most of the collaborative innovation process happens via in game chat and voice chat. All interviewees use an external tool called TeamSpeak to coordinate group-related tasks and having general discussions. TeamSpeak works similar to a Skype or a telephone conference where the players connect to a persistent server in order to talk to each other. One of the key findings of our study was that most of the addons are installed, based on recommendations from other guild members. These recommendations are seen as more or equally important as recommendations on the addon-related websites. The experienced players try to stay up-to-date by informing themselves on various WoW addon sites about updates for their current addons or new addons that could enhance their playing experience. As part of the study, we discovered several non-formal modifications practices.

If players want to accomplish certain tasks or face problems concerning certain game elements, this was often discussed with guild members, as helping other players is quite common. As part of this, players often received recommendations on what addons could simplify the completion of this task. If a problem with an addon or the game client arose, the other guild members were always asked first. Unfortunately, often it was not clear which guild member could be an expert for a certain addon or problem. The continuous use of voice chat benefits the virtual collaboration in a way that it can create virtual over-the-shoulder learning situations. For example in one specific case a player mentioned certain statistics about his character and another player asked where he could find this statistic. The first player then realized that this statistic was generated by an addon and recommended it. All interviewees use a third party tool, called Curse Client (see [5]) to install new and update existing addons. It was developed to help players in installing new addons and managing their current addon configuration, by providing them with an easy-to-use interface representation of a rich addon database. Information about new and interesting addons is spread verbally to players which are suspected to have an interest in these. New players usually get recommendations on certain addons they should install. These recommendations are usually given to them before they engage a difficult task with other guild members.

2.3 Discussion

A common phenomenon that we expected was that people tend to employ pre-existing modifications rather than developing their own, hence the fact that certain people in both studies had the skills to develop them. The interviewees in both studies argued, that using pre-existing modifications saves them a lot of time. What we did not expect were the similarities at the practice level as well as the reasons behind certain actions.

But by comparing the two studies, we found more similarities. Despite the differences of the contexts (work vs. play) and heterogeneous user groups (very skilled vs. varying) the sharing behavior was very similar. In both cases people relied on the recommendations of their friends or co-workers more or as much as they relied on recommendations made by external people or websites. We classified these as follows:

#1 asking, because of a problem

People actively asked their friends or co-workers about their software configuration and which modifications they use. This happened mostly if a problem or a new and unknown task appeared. In both studies we traced this back to the belief that the co-workers or friends better understand each others context and therefore are a more reliable source of information.

#2 asking, triggered by accident

In several cases, people either accidentally observed or discussed the use of an modification unknown to others. This triggered a need for awareness of what the colleague or friend is using and resulted often in a discussions on modifications.

#3 actively spreading the news

If people were actively informing themselves or by accident picking up news on modifications, they tried to spread this information further to (potentially) interested people in their near social network.

#4 actively introducing new workers/players

Sometimes a new person joins the game of WoW or a colleague joins a certain development project or even the company. In this case people did introduce the “junior” not only to the work/game, but also recommended certain modifications.

If we mirror these categories back to the several observations and interviews, they lead back to a lack of awareness within social networks as colleagues or guilds. Even more, there is a constant and latent need for this kind modification or EUD awareness. But since there was no support, it took certain points of interaction like a breakdown, beneficial accident or a complex task to bring the topic of modifications to the center of attention and create this sort of awareness.

3 Related work

Mackay [1] as well as Gantt and Nardi [2] empirically investigated into the collaborative effects of tailoring, as sharing knowledge and artifacts. Our work is similarly structured but takes current developments into account. More recent work on collaborative tailoring and the related topic of software appropriation was

especially done by Pipek and Kahler [6]. They did describe different shared scenarios (concerning usage, artifacts or infrastructure) that also lead to a need for awareness. While their work discusses this topic marginally, we wanted to focus more deeply on the role of awareness in collaborative EUD processes.

While existing research efforts investigated into organizations or closed groups as target for their research, we focus on groups, that act as social networks and are embedded in software ecosystems [3]. Therefore we can address appropriation efforts in environments where large numbers plug-ins/addons from 3rd parties already exist. This results in a different view on collaborative tailoring efforts and awareness.

3 Conclusion

Our research shows that people may trust in particular recommendations of local peers, regarding modifications as plug-ins or addons. Both, Eclipse and WoW users facilitate third party tools, to orchestrate and maintain their sets of plug-ins or addons. These tools all share the same basic features. Curse for WoW, as well as the Eclipse Marketplace or Yoxos for Eclipse represent central repositories of components to modify the application. These tools keep track of what a user installs, they help keeping addons or plug-ins up to date and ensure an installation that is not broken after modifying. On the other hand, they miss to reflect the collaborative nature of modifying software that creates a demand for addon awareness and recommendations.

In small local or remote groups as the ones we examined, we found plenty of incidents where breakdowns, accidents or planned intervention functioned as a trigger that revealed a need for awareness. This resulted in discussions and recommendations. This was just “the tip of the iceberg”, as we observed a constant and latent need for modification awareness in groups. And as there already exist tools to support some of the users EUD efforts, future research should suggest and evaluate possibilities to support *modification awareness*, as it could improve the reach of EUD.

References

1. Mackay, W.: Patterns of sharing customizable software. Proc. of the 1990 ACM CSCW, pp. 209-221, Los Angeles, California, United States (1990).
2. Gantt, M., Nardi, B.A.: Gardeners and gurus: patterns of cooperation among CAD users. Human factors in computing systems. Monterey, California (1992).
3. Bosch, J.: From software product lines to software ecosystems. 13th International Software Product Line Conference. CMU, San Francisco, California (2009).
4. Glaser, B.G., Strauss, A.L.: The Discovery of Grounded Theory: Strategies for Qualitative Research. Aldine Pub (1999).
5. Curse.com: World of Warcraft Addons - WoW Addons - Curse.com, <http://wow.curse.com/downloads/wow-addons/default.aspx>. (2011/03/14)
6. Pipek, V., Kahler, H.: Supporting Collaborative Tailoring. End User Development. Springer (2006).