Gamified Learning of Software Tool Functionality

Design and implementation of an interactive in-app learning interface for complex software

SONIA CAMACHO HERRERO
Abstract

Almost every application or platform that is developed nowadays includes a user onboarding experience. Onboarding is a process that starts the moment users enter an application and it aims to show the benefits of using the application, teach how the functionality works and motivate users to return. When the target application is considered a complex software, the teaching methodology needs to be carefully studied and designed. Through the example of a complex project management software, this master thesis aims to develop an in-app teaching interface that allows the users to understand the different functionalities and improve their work performance in a complex application. In this work, we developed an integrated learning platform as a result of methodical research-oriented design processes. The outcome of the research is a modular onboarding experience called the Learning Center. It includes a collection of courses based on video tutorials and interactive guided tours. Gamification elements such as progress bars and checklists are included as a way to engage users and encourage them to complete the courses. User tests showed an improvement in functionality understanding and a reduction in error rates. The Learning Center was considered useful and adequately approached. Future research includes making the appropriate learning material directly available from each software feature.

Sammanfattning

Gamified Learning of Software Tool Functionality

Sonia Camacho Herrero
KTH Royal Institute of Technology
Stockholm, Sweden
soniach@kth.se

ABSTRACT
Almost every application or platform that is developed nowadays includes a user onboarding experience. Onboarding is a process that starts the moment users enter an application and it aims to show the benefits of using the application, teach how the functionality works, and motivate users to return. When the target application is considered a complex software, the teaching methodology needs to be carefully studied and designed. Through the example of a complex project management software, this master thesis aims to develop an in-app teaching interface that allows the users to understand the different functionalities and improve their work performance in a complex application. In this work, we developed an integrated learning platform as a result of methodical research-oriented design processes. The outcome of the research is a modular onboarding experience called the Learning Center. It includes a collection of courses based on video tutorials and interactive guided tours. Gamification elements such as progress bars and checklists are included as a way to engage users and encourage them to complete the courses. User tests showed an improvement in functionality understanding and a reduction in error rates. The Learning Center was considered useful and adequately approached. Future research includes making the appropriate learning material directly available from each software feature.

ACM Classification Keywords
H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous

Author Keywords
In-app learning; onboarding; gamification; contextual-help; tutorials; software support.

INTRODUCTION
Complex software allows users to perform a variety of tasks. Nevertheless, greater functionality and support for complex tasks dramatically increases the challenge for novice users to understand and learn how to work in the application’s environment. How to teach users a software’s functionality depends on the characteristics of the tool, its most relevant learnability problems, and the type of users.

Traditionally, this aid has come in the form of user manuals and documentation. However, it has been observed that users are not usually keen on reading manuals [20, 17] or they don’t find them helpful enough in many contexts [24, 8]. As a result, different approaches have been developed in recent years. Contextual help, such as tooltips [7, 5], overlays [6] or tours [25, 6] are a recurrent approach in many applications. Video tutorials have also become a popular source of information [18], although they are normally accessed through external websites and not as part of the application itself. However, there is not a unique solution for teaching software functionality that works for all scenarios. A careful study needs to be done in each case in order to determine the best approach to implement.

This study investigates in-app teaching interfaces and onboarding methods applied to an advanced web-based tool for visualizations, measurements and planning of large software and hardware projects, called Feature & Product Tracker (FPT). This software was developed by Meepo AB (Stockholm, Sweden), which has collaborated actively in the present study. Nowadays, FPT is used by more than two thousand employees at the multinational telecommunication company Ericsson AB. The tool has grown organically over time and currently contains vast functionality, which provides powerful features for expert users, but simultaneously overwhelms novice users.

This project aims to investigate suitable methodologies to effectively teach novice users the functionality of such a complex software. Through a research-oriented design, learnability issues of the tool are identified in order to develop an interactive learning experience within the FPT software in charge of helping the users overcoming their challenges.

RELATED WORK
When referring to how to introduce new users into an app, the term user onboarding is commonly utilized. User onboarding centers on how the first contact users have with an application impacts their future interactions with it. In these first steps, users should recognize the value of the tool and the benefits of using it and understand how to make use of the available features in order to meet their needs. User onboarding is the experience designed to achieve these goals and is being a topic of high interest among the UX practitioner community (e.g., [1, 10]) due to its effect in boosting user retention and engagement, reducing support tickets, and increasing free trial-to-paid conversion rates.

While “onboarding” is the common term used in the business context, it is addressed in HCI research under the more general term “learnability” (e.g., [6, 2, 9]).

Although this topic has been extensively addressed, little has been done in the development of research guidelines for onboarding experience design. Related to this is the work of
Strahm et al. [23], who presented a research-informed method for designing and validating mobile onboarding experiences based on minimalist instructional principles [3] and existing practitioner guidance. As another example, Renz et al. focus on improving the onboarding user experience in MOOCs [19].

**Elements of user onboarding**

User onboarding should be an introduction to the app, showing its value to new users and helping them to understand how the different features work. Deciding which strategy is the most optimal depends on the characteristics of the app, the complexity of its features and the targeted users. In this research, some of the most relevant elements for onboarding have been identified and described with suggestions on when they are best used:

- **Welcome messages:** A welcome screen is usually shown the first time a user enters the app [1, 4, 19]. It introduces the user to the platform, often by following the welcome message with a few slides that highlight the value of the app and what the users will get from it.

- **Guided tours:** In order to show the users how the app works, a guided tour can be a good option to include. It can easily show the steps that are necessary to use a certain functionality and localize the features in context. Tooltips (figure 1) and overlays (figure 2) are proper guides for the purpose of indicating the location of functionality [6, 9, 14]. If an app includes features that are more complex to explain or difficult to understand, a video demonstration can be an effective alternative to include in the tour or make it available as contextual help [8].

- **Interactive tours** can also be designed with tooltips or media content, however they will not have a “next” button to continue with the tour [19]. Instead of that, they will ask the user to perform a certain task, providing hints on how to do it (see figure 3). This approach will allow the users to “learn by doing” and familiarize with specific features of the app.

- **Progress bars and checklists** are gamification elements that encourage users to complete the tutorials and explore the functionality on the app [1, 19].

**Evaluation methods**

Measuring and evaluating learnability is essential for the development and improvement of human-computer interaction, in general, and for user onboarding experiences, in particular. Depending on the goal of the tests, evaluation methods can be categorized as formative or summative [16].

**Formative evaluation** allows the researchers to improve the design of a system interface during its development by helping them promptly identify the learnability issues of the system.

**Summative evaluation** is intended to measure the overall quality of an interface, which can be used, for example, to compare between two competing systems or between different versions of the same product. In contrast to formative evaluations, summative evaluations typically occur at the end of a large development cycle and its main purpose is to measure the performance of an application compared to a control.

---

Figure 1. Guided tour with tooltips in a user onboarding for a MOOC platform [19].

Figure 2. Example of overlays from Snapseed app to show key features and behaviors [6].

Figure 3. Interactive step in a guided tour in Airtable 

[Airtable: https://airtable.com](https://airtable.com)
**Formative Learnability Evaluation Methodologies:**

The *think-aloud protocol* stands as one of the most common methods for testing usability and initial learnability [9, 16]. In this test, participants are asked to “think out loud” as they use the system. This method allows the researchers to understand the thoughts and the reasons behind the user’s actions and recognize possible misconceptions. Typically, participants will state their intentions, their plans to accomplish these goals, the steps in the process, their observations of the results of these steps, and a statement of whether the goal was achieved. If the goal is not achieved, participants will be encouraged to hypothesize on the causes of this unexpected outcome and to verbalize an experiment on how to succeed. The user will continue until the goal is achieved or aborted.

The *question-asking protocol* was proposed as an alternative to the think-aloud method with a more “ordinary” way for participants to verbalize their thoughts [13]. With this method, an expert of the system sits together with the participant and provides answers to any question the participant may have during the experiment. This way of communication is more natural than asking the subjects to describe their thoughts and might be more appropriate to use in some cases. This protocol is focused on how computer users learn to use a new interactive system and provides the researchers with insights about the user’s needs and perceptions about that system, instructional information that might be lacking, and specific problems in the interface that are a source of misconceptions.

Given that the question-asking protocol is only focused on initial learnability, it cannot be used for understanding how users find more efficient strategies for performing known tasks. For evaluating the extended learnability of a system, a new variation of the method was designed, turning the question-asking protocol into the *question-suggestion* protocol [9], which adds the possibility for the expert to provide advice to the user. Besides the outcomes of the previous protocol, this method allows the researchers to identify causes for suboptimal performance.

**Summative Learnability Evaluation Methodologies:**

Summative evaluation methods are mostly quantitative measurements and, in general, can be applied to measure both initial and extended learnability. From an extensive literature survey, Grossman et al. [9] provide the categorization of learnability metrics into seven groups. Here, only two of them are targeted as relevant for this context: task metrics and subjective metrics.

**Task metrics** are those based on task performance. In order to obtain them, a group of users is asked to perform a set of predefined tasks, while time and error data are collected. Some examples of metrics include task completion time, task completion rate within a certain time frame, and number of user errors [16].

**Subjective metrics** are those based on user feedback. A common metric of this category is the User Experience Questionnaire (UEQ) [15], which provides a fast and reliable method for measuring quantitatively the user experience of a product. It consists of 26 items in the form of semantic differ-entials corresponding to six categories/scales: attractiveness, perspicuity, efficiency, dependability, stimulation, and novelty. It is, however, a method for measuring all different aspects of usability, therefore, for the present purpose, some scales can be removed from the questionnaire, as suggested by the authors [21]. Thus, it is possible to shorten the questionnaire as long as the remaining scales have all their corresponding items. This method is typically used for comparing the user experience between two products or two versions of the same product [22].

**Research question**

This study aims to answer the following research question:

**Q. 1:** What impact does in-app teaching have on the learning, the performance, and the attitude of novice users of a complex project management application?

In order to answer this question, the learning material needs to be designed and implemented as part of the system. At this point, two additional questions emerge:

**Q. 2:** What are the most salient learning obstacles for new users of this software?

**Q. 3:** How can the in-app learning material be designed to overcome these challenges?

**METHOD**

The first analysis in the present study aimed to find the difficulties that new users face when they interact with the project management application (Q.2). Two evaluation methods were used for this purpose: A survey sent to the users through the application itself, and a round of user tests following the *question-suggestion* protocol.

**Initial survey**

A survey was designed and implemented in the system. It consisted of four free-text questions, one like-scale and a reduced version of the UEQ (see Appendix: Survey questions). The first ones were intended to extract information about how the software is being used, what level of understanding participants had about its features and which ones they were most interested in learning. The UEQ provided quantitative feedback about the general user experience. It was adapted to ask for items belonging to only three categories, out of the total of six categories in the full version. These chosen categories were attractiveness, perspicuity/ease of use and stimulation.

The survey was active for two weeks and had a 50% probability of popping up to users that logged into FPT. Users had the option of answering, skipping or opting out. At the end of the period, a total of 32 participants had answered the survey.

**Initial question-suggestion user tests**

User tests were carried out to get insights into what obstacles novice users find when they interact with the system. The question-suggestion protocol proposed by Grossman et al. [9] as a variation of Kato’s question-asking method [13] was chosen for this purpose.
Five Ericsson employees participated individually in the study. All were novice users of the software. The study was presented to them as a training session where they would be given some tasks to complete and would have the opportunity to ask any questions to an expert of the tool, in this case, one of the developers of the software. The screen and voice were recorded and written notes were taken during the sessions. The participants were encouraged to ask concrete questions about any specific problem they encountered, avoiding vague questions like “what should I do next?”. The expert was also allowed to make some suggestions when the participants were inefficiently performing the tasks.

In-app learning material design and implementation
The next step in the present study was to design and implement a teaching solution as part of the application. In order to do that, a study needed to be conducted to find suitable methodologies for the existing learning difficulties (Q.3). This study was conducted through an extensive literature review in the topics of learnability, user onboarding, in-app guidance, contextual help, tutorials, and gamification.

The characteristics of the methods collected were compared. The learning interface would be designed as a combination of the most appropriate elements identified. During the design process, wireframes and mockups were developed and discussed among members of the FPT development team through various iterations. Then, the final design was implemented in the FPT system.

Final evaluation
Once the learning material was included in the system, the final evaluation could be conducted. This evaluation was in charge of testing the new teaching interface and comparing the user’s understanding of the system and work performance after going through the learning material (Q.1). This was analyzed from a second round of users tests, with a structure similar to the ones conducted in the initial evaluation.

Five new users of the system (different from the previous participants) took part in these tests. They started by interacting with the learning material while thinking aloud about their impressions. After that, the same tasks as in the previous evaluation were given to the users, who were allowed to check any doubt by turning back to the learning material or asking the expert (question-suggestion protocol).

The study compared the users’ performance in both conditions (before and after including the learning material in the system). Since the number of participants was relatively small, only a qualitative analysis was performed. The variables to compare included the number of errors committed by the users in each task, the number and the type of questions asked to the expert, and the level of understanding of the task they were performing.

RESULTS
Initial survey
An initial survey (see Appendix: Survey questions) was sent out before any onboarding or interactive learning interface was developed. Only a text-based “quick guide” existed previously in the home page of the application.

The survey was shown randomly, during a two-week period, to any user that logged in FPT. A total of 32 users answered it, from beginners to advanced users, covering a great range of roles regarding how they worked with the application.

The first part of the survey consisted on a reduced version of the User Experience Questionnaire (UEQ). Figure 4 shows the mean value per item in the answers received. And figure 5 shows the distribution of the answers. While all the average results were positive, some of them were relatively low. That was the case of difficult to learn/easy to learn, boring/exciting and demotivating/motivating. The fact that the software is not seen as very easy to learn inspires the present study. Motivation and engagement are other aspects that boosting.

The rest of the survey contained free-text and likert-scale questions. They provided actionable feedback regarding which features of the application where most difficult to understand (like filtering, team planning, and working efficiently with the backlog item list). Some users pointed out general problems such as the difficulty to “keep finding new features all the time” or the little hands-on experience in the tool because he/she “more or less only use filtered searches, done by someone else”. And they also wrote some important suggestions, such as the creation of “a place to share best practices”.

An important finding from this survey was that even users that had been working with the tool for more than two years didn’t feel confident in knowing and understanding all its functionality. Answering to whether they would get more out of the system if they had some training, 52% of the participants
agree and a 24% said that “maybe”, leaving only another 24% to disagree.

Initial user tests
Five novice users of FPT participated in the first round of user tests. They assisted individually to meetings that were prepared not only as user tests but also as training sessions. An expert of the software, one of the developers, sat next to the subject acting as a personal tutor that would answer their questions and explain unclear features.

The session started with an introduction of the purpose of the test and the instructions to correctly follow the question-suggestion protocol. The participants gave their consent to the researchers to take notes about their behavior and record the screen and their voice. The data was treated anonymously. A total of 17 tasks were prepared for the tests, spread into different views such as Kanban boards or Gantt charts, thus covering a broad range of functionality in the tool.

The participants were given one task at a time. They were encouraged to ask the expert any question, avoiding vague questions unless they were truly confused about how to proceed. The expert was also encouraged to suggest specific actions or functionality when the participant’s choices were not the most efficient ones.

After the completion of the tasks, the subject had the opportunity to ask any other question about the application that had not been covered by the given tasks. Finally, the expert would show additional functionality in different views of the software depending on the role and interests of each user.

An analysis of the type of questions asked by the participants, their common errors, and the suggestions given by the expert resulted in a collection of learnability issues. These problems were classified following the same categorization as Grossman et al. [9]. Three out of their five categories were identified in different features of FPT:

- **Awareness of functionality.** An common issue in this category was that users didn’t know that they could save their current settings and filters or send a view with specific items filtered to a colleague.

- **Locating functionality.** This was the case of some specific settings that were located in the top right corner of the window. Most of the users didn’t look at that part of the screen.

- **Understanding functionality.** One of the tasks in the user test proved to be remarkably difficult. Only one out of the five participants was able to complete it without the help of the expert. The reason of this problem was that the users did not understand how a specific feature worked. The issue was related to the introduction of a new set of filters in one of the views. Despite having the same structure, each group of filters worked in a different way. One would change the displayed color of the items, while the other would show or hide teams from the list. Understanding this functionality was not straightforward.

**Design and implementation of the Learning Center**

The collection of learnability issues that users found during the tests was the basis for designing the new in-app teaching of FPT, coupled with a literature review to find guidelines for solving each type of problem.

The first requirement that the learning material had to fulfill was to be accessible to everyone in the application. From the initial survey, it was proven that even users that had been working with FPT for years still had things to learn and discover. Therefore, the idea of designing the learning material only as part of an onboarding experience was discarded. Instead, the decision turned into the development of a Learning Center, a collection of courses that would teach users the different features of FPT and would be part of the application itself.

How to design the courses was the next step to study. The followed approach was to look for methods that could solve the types of learnability problems identified in the initial user tests.

Two different options were chosen to be used in the sections of the courses depending on the feature explained. On one hand, the courses would include multimedia content in the form of short video tutorials. They would be used to explain with visual demonstrations how certain complex functionality works. On the other hand, some parts of the courses would include guided tours. This choice was made to overcome the challenge of making the users aware of the existence of certain functionality and where to find it. These guided tours would combine passive and active steps, which would invite the users to interact with the tool. This method would offer a “learn by doing” approach.

Another suggestion that was considered during the design process of this Learning Center was that it should be engaging so that users would go through all its content. For that, different
gamification elements were analyzed. The proposed design included progress bars and a checklist indicating which courses were finished or in progress.

Several sketches, wireframes and mockups were developed and shared with members of the FPT development team in order to discuss and define the design of the interface.

The Learning Center was implemented inside the FPT system. It was built in Angular 7, with a back-end in .NET and a database in SQL to save the user progress in the courses.

Accessible from the Home menu in the top bar of the application, its main view shows the total user progress in the courses and a collection of clickable cards to access each of them (see figure 11 in Appendix: Learning Center). A static panel located on the left shows the index of courses together with an icon representing the progress in each of them in the form of a checklist (see figure 6). By clicking on any of the titles or icons the user will also navigate to the corresponding course.

Each course is focused on a specific view, such as the Kanban Board or the Backlog Item List, and is divided into short sections explaining the different functionality that the view offers (see example in figure 12 in Appendix: Learning Center). A section can be constituted by a short text and a video tutorial or a button to start a guided tour. When taking a guided tour, the user will be redirected to the corresponding view and will go through a series of short steps explaining the task flow of a certain feature. Some of the steps will only require the user to click on “next” to continue with the following step, while others will require to perform an action, like clicking on an element in the interface (see figure 8). The user will also have the option to skip the tour at any time.

Besides the implementation of the Learning Center interface and the user’s progress database, the course content had to be created. This material was prepared in collaboration with developers of the FPT software and consisted of short description texts for each section of the courses, task flows and instructions for the steps of the guided tours, and two-minute-long video tutorials. The videos were recorded in English, with a presenter introducing the view or feature and showing its functionality by demonstrations, visualizing the screen and the cursor interactions (see figure 9). A total of five courses were created, which included three guided tours and seven videos distributed among several sections. In order to mark a section as completed, the user was required to watch the entire video or finish the guided tour.

Finally, an additional feature was developed outside from the Learning Center: a notification badge for highlighting new views or functionality added in the latest version of the application. Including this element in the software was an idea that arose from the survey feedback and other conversations with stakeholders of the tool. In general, it was not easy to stay up to date with the features that were included in each new version of the application. Release notes are always shown the first time a user loads an updated version, and they contain a summary of all the new features added, the changes and the bug fixes. However, as discussed in the introduction, people don’t usually read. For that reason, a complement to the
release notes had to be designed. The proposed solution was to show a notification badge next to the new feature developed. That visual hint would indicate users what feature was added and where to find it, and would disappear once they interact with it.

The first use of this badge was for nothing other than the Learning Center (see figure 10). It was designed to be visible in the navigation bar and indicate where the new feature was located, so that the user could easily find the access to it.

**Final evaluation**

Once the Learning Center was implemented and included in FPT, the final evaluation could be conducted. Five new users of FPT participated in this second round of user tests. The difference compared to the initial round was that this time the sessions lasted one hour and had two phases. In the first half of each meeting, the participants went through the courses, watched the videos and took the guided tours. During that process, they gave feedback regarding their impressions on the courses, their level of understanding about the features explained and their comments about the methods used for teaching them.

In the second phase of the user tests, the experimenter gave the participants the same tasks as the ones prepared for the initial evaluation. Following the same protocol as previously, an expert of the software was sitting next to the participant answering their questions and suggesting actions to improve their performance.

The first phase provided important feedback regarding how the Learning Center could be improved to offer more clearance to the users. One of the most relevant issues was related to the mix between passive and interactive steps in a tour. The participants got used to click “next” on the first steps of the guided tour and when an interactive step appeared, they were confused about why there was not a “next” option. After explaining that they had to perform the action described in the text of that step in order to proceed with the tour, they all understood it and most of them admitted that they had read it too fast that they didn’t realize the text was asking the user to perform an action. One of the participants suggested to write the required action in bold to highlight it or even to change the color of the text box, so that the user could notice that it was a different type of step.

Regarding the videos, some issues were also reported by the participants. One of them suggested to include subtitles, since “English was not [her] mother tongue” and could not understand everything. Two of them also mentioned that it was difficult to be focused on what the person in the tutorial was explaining at the same time as finding where they were pointing at. The cursor in the video demonstrations was moving too fast sometimes that the participants were distracted from the explanation trying to locate the position of the cursor.

The second phase of the user tests showed the improvement in the users’ understanding of the features compared to the results from the initial tests. Even though the participants kept asking questions to the expert and committed a few errors while trying to complete the tasks, the majority of the issues that were present on the initial tests had disappeared in this round of tests. The most relevant observations are described below.

Even though sometimes participants did not remember if they had seen a certain task flow in a video or a guided tour, they recalled the features. They were aware of their existence and were most of the times able to quickly locate them.

An example of this is that all the participants checked the specific settings located in the top right corner of the screen to see if they provided the functionality asked in certain tasks of the test. In some cases, the participants didn’t remember exactly which features were available under these settings, but all of them checked the element to find it out. Whereas in the initial tests, some participants weren’t even aware of the existence of that element and didn’t look at that part of the screen until the expert pointed at it.

Some participants still committed a few errors when they were asked to perform certain tasks. However, significant improvements were observed. That was the case, among others, of a particular feature that none of the participants in the initial evaluation understood without the help of the expert. In contrast to that, three out of the five participants on the final evaluation completed the task making use of that feature without any help.

**DISCUSSION**

The evaluation showed that the Learning Center helped novice users have a better understanding of the software and provided
excellent feedback on how it could be improved in the next design iteration.

Even though some of the features that were implemented couldn’t be tested, they were discussed in posterior meetings with the stakeholders. That was the case of the notification badge for new features. Feedback for this implementation was positive. However, additional tests would be beneficial in order to evaluate how the visibility of the new features has improved and whether the badges could be perceived as something intrusive.

Other features were suggested after analyzing the user feedback but they remain as future work. The most relevant proposal was to show a link to the videos and guided tours from the corresponding views and features. That is, besides having all the learning material available in the Learning Center, an info button could be placed next to the real features. This way the users could find the information exactly when they needed it, instead of having to check if that feature was covered in any of the courses in the Learning Center.

Another interesting feature that was proposed by some participants and discussed with stakeholders was the creation of a platform where users could share “best practices” about the tool functionality. This cooperative approach would prompt novice users to adopt more efficient behaviors suggested by users with more expertise. This platform could provide a bidirectional communication. On one hand, it would allow users to share pieces of advise, practical examples of certain functionality or any material that could help their colleagues. On the other hand, it could also allow users to publish their own questions about the tool. Furthermore, additional courses could be included in the Learning Center in relation to the most common questions asked through this platform.

Regarding the methods utilized in the evaluation of this interactive learning interface, some points can be discussed. Evaluating interactive learning interfaces usually involves measuring knowledge retention and, for that, a longer testing period is required [12, 11]. As future research, it is suggested to divide the user test into two sessions separated at least one week from each other. The second session would measure how much the user was able to retain from the courses that were taken the previous week.

With a longer evaluation period available, a second survey could have also been sent to the users. This survey would have provided a quantitative comparison of the user experience and learnability of the tool after the addition of the Learning Center, by means of the UEQ and other specific questions.

Finally, with an even longer period of time (two or three months for both the initial and the final evaluation) software usage analytics could have been measured. This would have proved whether new users were quicker to adopt a large span of features with the help of the learning material, and whether they were more engaged with the tool. This could have been done by tracking analytic metrics and comparing between two conditions: before and after the implementation of the learning interface, or between users that have taken the courses and those who have not. This metrics would include the user activity (are users logging into FPT more often?), the feature views (are users adopting a larger span of features?) and the session time (are the users spending more time in FPT?).

CONCLUSION
An interactive learning interface has been designed and developed as a way to onboard new users into a complex software, as well as to teach more experienced users some advanced functionality they didn’t know before. A pre-study with an online survey and a round of user tests was conducted in order to collect information about the obstacles and challenges novice users found when working with the tool, a project management software called Feature & Product Tracker. These problems were classified into different types of learnability issues to facilitate the task of finding solutions to every kind of issue. From a literature review, suitable teaching methods were identified. Several iterations in the design process with feedback from the software development team led to the final structure for the learning interface.

The Learning Center is the proposed solution to overcome the challenges of this complex software. It consists on a collection of courses based on video tutorials and interactive guided tours that show from basic to more advanced functionality of FPT. Gamification elements such as progress bars and checklists are included in the interface to show the users their progress in the courses and to encourage them to complete all the sections.

A second round of user tests showed the improvements in work performance and understanding of the system by novice users compared to the initial evaluation. User feedback was collected and analyzed in order to propose changes in the design of the Learning Center. Guided tours, for instance, should have a clear visual distinction when they require an action to continue in opposition to a passive step where the user can click on a “next” button. Video tutorials could be improved by a better highlighting of the cursor position and the addition of speech pauses while showing the interaction with the interface, so that the user can focus on one thing at a time. They should be kept short, ensuring that the users are not overwhelmed by getting too much information at a time.

A link to the videos and guided tours from the software features and not only from the Learning Center page is proposed as future work. This would allow users to find the information in the moment they need it, while they are working with a specific view or functionality they are not experienced with.

The results from this project have provided insights into the user onboarding and learning process in complex software tools. Although this study is centered in a particular application (the FPT software), the followed methodology, the design choices and the final results can be relevant as a reference for future research and implementation of onboarding experiences by both the practitioner and research community.

ACKNOWLEDGMENTS
I would like to thank my supervisor at KTH, Mario Romero, for his guidance and advice during this research. I would also like to express my gratitude to the all the team at Meepo, who gave me the opportunity to collaborate with them in this
inspiring project; in special, to Sebastian Geidenstam, Anders Hård and Ah Zau Marang for their invaluable help, and to Sandra Barkestam and Christian Sterngren for the recording and editing of the video tutorials. I am also very grateful to the 42 anonymous subjects from Ericsson, whose participation and feedback made this study possible. Finally, I wish to present my special thanks to Marcos and to my family, for their unconditional support and encouragement throughout this thesis work and all my studies.

REFERENCES


APPENDIX

LEARNING CENTER

Figure 11. Course progress in the Learning Center.

Figure 12. Example of a course in the Learning Center.
SURVEY QUESTIONS
1. With the help of the word-pairs, please enter what you consider the most appropriate description for FPT. Please click on your choice in every line!

<table>
<thead>
<tr>
<th>annoying</th>
<th>enjoyable</th>
</tr>
</thead>
<tbody>
<tr>
<td>not understandable</td>
<td>understandable</td>
</tr>
<tr>
<td>easy to learn</td>
<td>difficult to learn</td>
</tr>
<tr>
<td>valuable</td>
<td>inferior</td>
</tr>
<tr>
<td>boring</td>
<td>exciting</td>
</tr>
<tr>
<td>not interesting</td>
<td>interesting</td>
</tr>
<tr>
<td>good</td>
<td>bad</td>
</tr>
<tr>
<td>complicated</td>
<td>easy</td>
</tr>
<tr>
<td>unlikable</td>
<td>pleasing</td>
</tr>
<tr>
<td>unpleasant</td>
<td>pleasant</td>
</tr>
<tr>
<td>motivating</td>
<td>demotivating</td>
</tr>
<tr>
<td>clear</td>
<td>confusing</td>
</tr>
<tr>
<td>attractive</td>
<td>unattractive</td>
</tr>
<tr>
<td>friendly</td>
<td>unfriendly</td>
</tr>
</tbody>
</table>

2. What work-tasks (if any) does FPT help you with?

3. How much do you agree with the following statement?: "I feel confident with my skills in using all the features that FPT offers for my needs."

   Strongly disagree  O O O O O O O  Strongly agree

4. Can you name any FPT features which you think would help you but that you would need to know more about in order to use them?

5. Do you feel that you would get more out of the tool if you had some training in it?

6. Any additional comments or feedback?