RAPID PROTOTYPING ON THE EXAMPLE OF SOFTWARE DEVELOPMENT IN AUTOMOTIVE INDUSTRY

The Importance of their Provision for Software Projects at the Correct Time

Andreas Holzinger, Olivia Waclik, Frank Kappe
Institute of Information Systems and Computer Media, Graz University of Technology
Infeldgasse 16, A-8010 Graz, Austria

Stephan Lenhart, Gerbert Orasche
AVL List GmbH Graz, Hans-List-Platz 1, A-8020 Graz, Austria

Bernhard Peischl
Softnet Austria, Institute of Software Technology, Graz University of Technology, Infeldgasse 16, A-8010 Graz, Austria

Keywords: Software prototyping, Usability, User experience, User-centered design, Prototyping.

Abstract: Software prototyping is a powerful method for the identification of usability problems at the very beginning of software development. This paper deals with the development of a prototype used for usability testing and presenting it to stakeholders at the correct time. A low-fidelity (lo-fi) prototype was created for a software product in the automotive industry, however the usability test was shifted to conduct it with the latest build of the software application. This paper emphasizes on the effectiveness of prototypes together with usability studies. It gives an overview about the experiences with usability testing on a high-fidelity (hi-fi) prototype late in the software development process. The main conclusion is that we assume that solving the usability findings of a hi-fi prototype is more difficult and expensive than using results from a lo-fi prototype earlier. In future, we will conduct a lo-fi prototype usability study to confirm this assumption.

1 INTRODUCTION

AVL List GmbH, an Austrian based enterprise in the automotive industry, has developed AVL InMotion, which is a sophisticated three-dimensional software solution for maneuver and event based testing at the test bed. It supports key business objectives such as hybridization and electrification of power trains. AVL integrated this standalone software into their existing business software suite. In a first step, the standalone application has been reviewed based on usability heuristics in order to find out how the application can fit into the existing software suite. For this purpose a lo-fi prototype (mock-up) has been developed (Brown & Holzinger, 2008), (Holzinger & Brown, 2008) in order to verify the graphical user interface (GUI) design proposals and to identify the end user needs, along with the weaknesses and strengths of the existing application

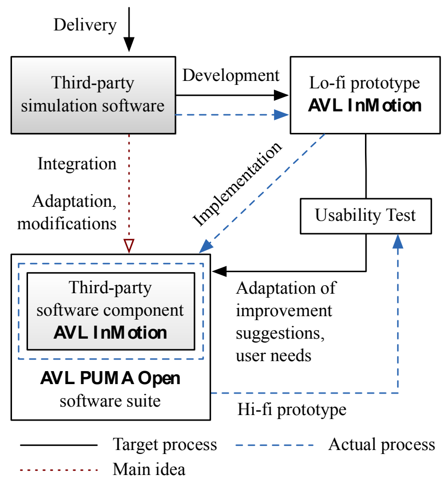

Figure 1: Usability test - target vs. actual process.

Holzinger A., Waclick O., Kappe F., Lenhart S., Orasche G. and Peischl B..
RAPID PROTOTYPING ON THE EXAMPLE OF SOFTWARE DEVELOPMENT IN AUTOMOTIVE INDUSTRY - The Importance of their Provision for Software Projects at the Correct Time.
DOI: 10.5220/0003524200570061
In *Proceedings of the International Conference on e-Business* (ICE-B-2011), pages 57-61
ISBN: 978-989-8425-70-6

(Figure 1). This paper emphasizes the importance of prototyping combined with usability testing before the implementation of a software product takes place and to figure out the best moment in which to present results to the decision makers.

2 BACKGROUND

After gathering the requirements for a software application, a concept fulfilling these requirements has to be developed. This concept has to be implemented into a lo-fi prototype, which shows the possible layout and structure of the system and supports developers in getting an idea of the interaction and design of the software product. It is known that the exploration of different low fidelity prototypes foster creativity (Fonseca et al., 2009).

Moreover, software prototyping is an effective tool used to reduce risks and point out requirements that should be considered (Hsia, Davis & Kung, 1993). It is an effective method to evaluate early designs of the user interface (UI) and to get early feedback from potential end users. Prototypes are especially used to create an authentic UI for design and evaluation in the early stages of a software project (Holzinger, 2004).

The characteristics of prototypes range from lo-fi to hi-fi prototypes. Lo-fi prototypes are created quick and simple and are relatively cheap in development. Both users and stakeholders (decision makers) are able to get a first idea of how the system will look (Casaday & Rainis, 1995), (Beyer & Holtzblatt, 1999).

Prototypes that are developed with design software (GUI Builder, cf. with (Spinellis, 2002) or GUI Designers, cf. with (Gunderloy, 2005)) provide GUIs with a slightly higher fidelity and are closer to the final design. Furthermore, HTML, Flash or Silverlight can be used for the creation of such design concepts. Especially the latter is well suited for 3D simulation software (Maiti, 2009).

In contrast, hi-fi prototypes are already very close to the target system. These prototypes contain functions that are integrated into the software application in order to enable users to get a more realistic impression of the software and to make it easier for them to make decisions on the software design (Rettig, 1994).

3 MATERIALS AND METHODS

3.1 Prototype Development

3.1.1 The Target System

The target system is represented by the software component AVL InMotion that supports hardware-in-the-loop (HIL) tests of hybrid vehicles at early stages of the development process. The application simulates virtual routes and maneuver based testing at the test bed, as shown in Figure 2.

Figure 2: Maneuver and event based testing via AVL InMotion.

3.1.2 Customization and Design

The integration of an external component into an existing, complex software system usually comes with problems. The reasons for this can be different, for example, incompatible system architectures and frameworks, or missing interfaces that prevent the communication between two applications. However, the problems need not be primarily technical in nature. There can be communication problems between developers or different goals pursued by stakeholders (Larsson, Crnkovic & Ekdahl, 2004).

"The target is to integrate components into a product and to ensure that the product works appropriately so that it can be delivered to customers" (Larsson et al., 2004).

In most cases, it is impossible to use a third-party component without making adaptations to suit the particular needs. (Szyperski 1997) discusses the problems and implications regarding the specification of functionality and quality attributes as well as the challenges in the integration of third party components. There is also a need for AVL List GmbH to modify the purchased software application for the integration into their proprietary existing software system, especially considering that the corporate design of the system must be consistent.
(Holzinger, 2005) emphasizes that a modification and reengineering of the UI can be more expensive

and complex to set up in its implementation. This can be avoided by considering user-centered design (UCD) at the very beginning of software development.

3.1.3 Prototype of AVL InMotion

The primary objective for developing a prototype for the application AVL InMotion was the creation of a user friendly GUI. In this context, the documented requirements had to be analyzed to determine the required parameters, which were necessary in order to create a model for the design of the application. The acquired, externally developed software application provided many individual windows as well as confusing menu entries, so that the goal was to create a less complex UI for AVL InMotion by considering the existing interaction design of the AVL software suite and by following the AVL corporate design guidelines. It was challenging to cover the essential and most important functions of the standalone application as well as to inspect each window in order to create the new GUI for the software component AVL InMotion, which should present all key parameters and collect all functions in one common frame.

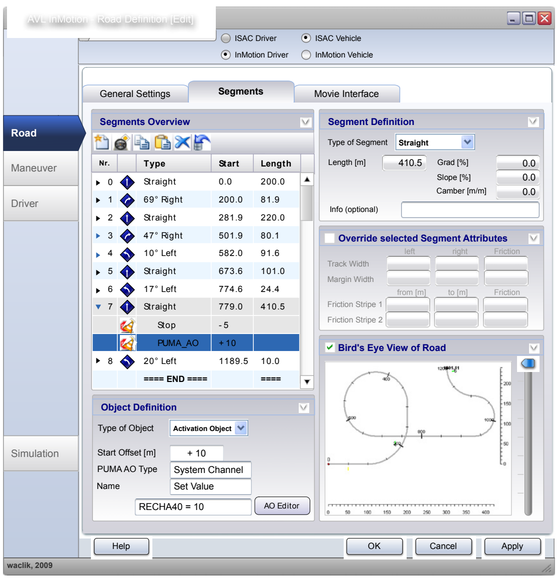

Figure 3: Screenshot of the lo-fi prototype.

The final lo-fi prototype of AVL InMotion (see Figure 3) was developed with HTML, in combination with a GUI-Builder tool.

This represents a quick method to prepare a presentation of the first scenarios of the planned software design.

Typically, these mockups are called "click dummies", which represent a prototype that looks like the final software, but does not provide specified functionality. The idea is to achieve a perception test of the user interface by providing little interaction opportunities for the test users. By using this prototype, the test users were able to navigate through the different main menus and submenus represented by tabs and they were able to interact with the main features provided by AVL InMotion.

3.2 Experimental Settings

The goal of this usability test was to evaluate the ease of use and simplicity of the software component AVL InMotion and to get feedback about the look and feel of the prototype. In addition the planned usability test should verify the UI design proposals. All findings and possible recommendations for usability improvements should directly influence the development of the application AVL InMotion afterwards.

In practice this approach could not be followed as supposed in advance. As the prototype was presented to responsible stakeholders, they immediately started implementing the proposed UI design without testing it with representative users before. Instead of using the lo-fi prototype for the usability study, the fully implemented software system of AVL InMotion was finally used for testing.

3.2.1 Participants

During the testing phase, it seemed that five test participants were sufficient to gain meaningful results. All participants were AVL List employees who had no experience with using the integrated software component AVL InMotion, but already had experience with parts of the existing software suite. It turned out that these participants and their experiences perfectly matched the ideal representative user profile.

3.2.2 Test Procedure

Each test participant was asked to complete a questionnaire before and after the test, in order to acquire the necessary background knowledge for the further course of the analysis.

After completing all the formalities, the test participants had to solve 19 tasks in a feasible period of time. The interaction of the test participants with the UI was observed, and at the same time the test participants were asked to express all their thoughts and actions during the tasks aloud. According to this it should be mentioned that thinking aloud (THA) is probably the most common technique in testing usability (Holzinger, 2006). The advantage of this method is the documentation and traceability of individual steps via the verbalization of the thoughts of the test participants and so problems can be systematically located and identified.

For further evaluation of the task completion, the time the test participants needed to finish each task was logged. For rating each specified scenario the so-called "Time-to-Task" had to be analyzed. Figure 4 shows the test environment of the usability study.

Figure 4: A test person is using the application within the usability lab.

4 RESULTS

A closer look is given at the number of main issues that were found by the participants who tested the software component AVL InMotion during the usability test.

4.1 Identified Problems

The identified problems were grouped into manageable categories and they were prioritized according to their severity. The ranking of the severity (Levi & Conrad, 1996) is listed in Table 1. Overall, 35 problems were identified, whereby the weighting of the severities is as follows: Four problems were assigned to severity level 4, eleven problems to severity level 3, nine problems to severity level 2 and eleven problems to severity level 1. No problem was classified as level 0.

4.2 Recommendations

A list of recommendations was prepared on the basis of the results provided by the usability test and its identified problems. In total, 16 recommendations were identified and their categorization was as follows: [R1] use of contextual menu, [R2] meaning of labels, [R3] parameterizing of segments, [R4] redesign of drop-down lists, [R5] behavior of input fields, [R6] composition of driver input mask, [R7] modification of the sequence of segments/ maneuvers, [R8] adding new segments/ maneuvers, [R9] structure and layout of the entire dialog, [R10] functionality and layout of toolbars, [R11] bird's eye view of road, [R12] functionality of the bird's eye view of road, [R13] maneuver settings, [R14] three-dimensional movie, [R15] simulation of the 3D movie in PUMA and [R16] online editing mode of test run. For each of the previously identified categories, a problem description and a possible solution were presented to AVL.

Table 1: Severity rating scale for usability problems.

Severity	Description
0	I don't agree that this is a problem at all.
1	Cosmetic problem only - need not be fixed unless extra time is available on project.
2	Minor usability problem - fixing this should be given low priority.
3	Major usability problem - important to fix, so should be given high priority.
4	Usability catastrophe - imperative to fix this before product can be released.

5 LESSONS LEARNED

It is important to provide the developed prototype at the correct time; in principal prototyping is an attractive approach in gaining a quick and cost-efficient impression of the possible design of the user interface. However as the preliminary version of the UI of AVL InMotion was created and demonstrated to AVL, the implementation and technical development followed immediately and it was decided to conduct the usability test with the latest build of the software application - without any evaluation of its UI design based on various usability criteria. After the completion of the usability test, several issues were identified concerning the usage and design of AVL InMotion. This led to the conclusion that it is not always beneficial to present a prototype to responsible parties without including a user perspective. Although solutions have to be developed in time and within budget constraints, it is far more complex and takes much more effort to implement the usability issues too late in the software development process when the development of the software applications is nearly finished.

6 CONCLUSIONS

This usability study represents an essential input for all further activities of the AVL InMotion development. It shows the current weaknesses and strengths of the application and even of the companies' software development process.

It shows that it is essential to integrate usability engineering at a very early stage of development to ensure cost-effectiveness and to even have the possibility to integrate most findings into the final product. This will of course involve a change from an engineering-centered corporate culture to a more user-centered one, but will further enhance the long-term growth-perspective and customer satisfaction of a product.

The responsible development departments were extremely pleased with the results and plan to integrate the recommendations into the next product version. Increased user efficiency and reduced development costs will ensure the long-term success of AVL InMotion.

ACKNOWLEDGEMENTS

Special thanks to the profound support of the software engineering team of AVL.

REFERENCES

Beyer, H. & Holtzblatt, K. (1999) Contextual design. *interactions,* 6, 1, 32-42.

Brown, S. & Holzinger, A. (2008) Low cost prototyping: Part 1, or how to produce better ideas faster by getting user reactions early and often. In: Abuelmaatti, O. & England, D. (Eds.) *Proceedings of HCI 2008.* Liverpool: John Moores University (UK), British Computer Society.

Casaday, G. & Rainis, C. (1995) Models, prototypes, and evaluations for HCI design: making the structured approach practical. *Conference companion on Human factors in computing systems*. Denver, Colorado, United States, ACM.

Fonseca, M. J., Jorge, J. A., Gomes, M. R., Goncalves, D. & Vala, M. (2009) Conceptual Design and Prototyping to Explore Creativity. In: Kotze, P., Wong, W., Jorge, J., Dix, A. & Silva, P. A. (Eds.) *Creativity and Hci: From Experience to Design in Education*. New York, Springer, 203-217.

Gunderloy, M. (2005) *Developer to Designer: GUI Design for the Busy Developer.* New York, Wiley.

Holzinger, A. (2004) Application of Rapid Prototyping to the User Interface Development for a Virtual Medical Campus. *IEEE Software,* 21, 1, 92-99.

Holzinger, A. (2005) Usability engineering methods for software developers. *Communications of the ACM,* 48, 1, 71-74.

Holzinger, A. (2006) Thinking-aloud - eine Königsmethode im Usability Engineering. *OCG Journal,* 31, 1, 4-5.

Holzinger, A. & Brown, S. (2008) Low cost prototyping: Part 2, or how to apply the thinking-aloud method efficiently. In: Abuelmaatti, O. & England, D. (Eds.) *Proceedings of HCI 2008.* Liverpool: John Moores University (UK), British Computer Society.

Hsia, P., Davis, A. & Kung, D. (1993) Status Report - Requirements Engineering. *IEEE Software,* 10, 6, 75-79.

Larsson, S., Crnkovic, I. & Ekdahl, F. (2004). *On the Expected Synergies between Component-Based Software Engineering and Best Practices in Product Integration*. EUROMICRO Conference, 430-436.

Maiti, A. (2009) Flight simulation in 3D Web Map Services using Silverlight (TM) 3 to Reduce Simulation Costs in Pilot Training Institutes. In: Ao, S. I., Douglas, C., Grundfest, W. S. & Burgstone, J. (Eds.) *World Congress on Engineering and Computer Science*. Hong Kong, Int Assoc Engineers-Iaeng, 591-594.

Rettig, M. (1994) Prototyping for tiny fingers. *Communications of the ACM,* 37, 4, 21-27.

Spinellis, D. (2002) Unix tools as visual programming components in a GUI-builder environment. *Software-Practice & Experience,* 32, 1, 57-71.