User Interface Prototyping – Concepts, Tools, and Experience

Dirk Bäumer
RWG
Räpplenstr. 17
70191 Stuttgart
daabd@rwg.stgt.sub.org

Walter R Bischofberger
UBS
Bahnhofstr. 45
Ch-8021 Zurich
bischi@ubilab.ubs.ch

Horst Lichter
ABB Corp. Research
Postfach 101332
D-69003 Heidelberg
lichter@decrc.abb.com

Heinz Züllighoven
University of Hamburg
Vogt-Kölnstr. 30
D-22527 Hamburg
zuelligh@informatik.uni-hamburg.de

Abstract

In recent years the development of highly interactive software systems with graphical user interfaces has become increasingly common. The acceptance of such a system depends to a large degree on the quality of its user interface. Prototyping is an excellent means for generating ideas about how a user interface can be designed, and it helps to evaluate the quality of a solution at an early stage.

In this paper we present the basic concepts behind user interface prototyping, a classification of tools supporting it and a case study of nine major industrial projects. Based on our analysis of these projects we present the following conclusions: Prototyping is used more consciously than in recent years. No project applied a traditional life-cycle approach, which is one of the reasons why most of them were successful. Prototypes are increasingly used as a vehicle for developing and demonstrating visions of innovative systems.[1]

1. Introduction

Prototyping is a development approach used to improve planning and execution of software projects by developing executable software systems (prototypes) for experimental purposes. It is very suitable for gaining experience in new application areas and for supporting incremental or evolutionary software development.

Many experience reports on prototyping have been published (e.g. [8, 9, 5, 12, 13]). They illustrate the impact of prototyping on software construction and the overall development process. Recently Gordon and Bieman have taken stock and presented a survey of published and still unpublished experience reports [7]. They have identified three kinds of experience reports: commercial, academic, and military. The reports were analyzed from a process and product view, resulting in conclusions about benefits and possible problems in areas such as design quality and end-user participation.

During the past few years the development of highly interactive software systems with graphical user interfaces has become increasingly common [14]. The acceptance of such systems depends to a large degree on the quality of their user interface. Prototyping is an excellent means for generating ideas about how a user interface can be designed and it helps to evaluate the quality of a solution at an early stage. This is the reason why user interface prototyping is applied in an increasing number of projects.

In this paper we present a case study on nine major industrial projects where the main focus was on user interface prototyping and where different tools were used to build different kinds of user interface prototypes. It starts with a brief introduction of our prototyping and tool related terminology. A short tabular overview of the investigated projects is followed by an analysis of the application of prototyping approaches and tools. The appendix presents more detailed overviews of the projects. In-depth descriptions of the latter can be found in the technical report on which this paper is based [1].

2. User interface prototyping

2.1 Approaches to user interface prototyping

For classifying approaches to prototyping, Floyd's three "E" model is widely accepted and used in Europe [6]. Differences exist in the interpretation of the "E"s: exploratory, experimental and evolutionary. We distinguish two different approaches: the *process* view which concentrates on the development process and its goals [3] and the *product* view which concentrates on the results of the process [2]. In the following we discuss only the process view. An extensive discussion of the terminology can be found in [1].

- *Exploratory Prototyping* serves to clarify the requirements and potential solutions. It results in discussions of what should be achieved by a task and how it can be supported with IT. Results are

1 published in *Procs. of the 18th International Conference on Software Engineering (ICSE)* Berlin, 25-30 March 1996

usually *presentation* prototypes and *functional* prototypes.

- *Experimental Prototyping* focuses on the technical realization of selected requirements. It results in experience about the suitability and feasibility of a particular design/implementation. Results are usually functional prototypes and *breadboards*.
- *Evolutionary Prototyping* is a continuous process for adapting an application system to rapidly changing organizational constraints. It is not merely used in the context of a single project. Although in the process of evolutionary prototyping all kinds of prototypes may be built, the construction of *pilot systems* is of particular importance.

2.2 Classification of user interface prototypes

Besides classifying the different approaches to prototyping it is also important to classify the different kinds of prototypes that can be built.

- *Presentation Prototypes* are built to illustrate how an application may solve given requirements. As they are often used as part of the project proposal, they are strongly focused on the user interface.
- *Functional Prototypes* implement strategically important parts of both the user interface and the functionality of a planned application.
- *Breadboards* serve to investigate technical aspects such as system architecture or functionality of a planned application. They are built to investigate certain aspects of special risk. They are not intended to be evaluated by end users.
- *Pilot systems* are very mature prototypes which can be practically applied.

In this paper we frequently use the term "user interface prototype" for a prototype that serves to clarify user inter-face aspects. Its classification depends on how and to what degree its functionality has been implemented. User inter-face prototypes range from presentation prototypes that can be complete mock-ups to fully functional pilot systems.

3 Classification of user interface prototyping tools

While analyzing the projects, we have identified four categories of tools that were used to build user interface prototypes. To provide a well defined terminology, we briefly define these categories.

HyperCard-like tools

HyperCard is a tool providing an interactive environ-ment for developing simple information systems with graphical user interfaces consisting of cards, stacks of cards, links, and event handling scripts. The combination of links and scripts makes HyperCard a powerful proto-typing tool. Links can be used to quickly connect a set of drawn user interface states into a mock-up application while "real" functionality can be implemented with scripts.

The success of HyperCard resulted in the development of clones on various platforms. For this reason we talk about HyperCard-like tools in the rest of this paper.

Interface builders

These are tools that serve to define user interfaces on a high abstraction level either textually or with a graphical editor. They support the creation and laying out of user interface elements and the specification of the reaction on events. Only interface builders that provide a graphical editor are of interest for prototyping purposes.

4th generation systems

A 4th generation system (4GS) is a complete development environment for information systems. A 4GS usually provide tools for graphically designing data models and user interfaces, an integrated interpretive scripting language, and various other tools such as report generators, program editors and debuggers. They are ideal for prototyping of information systems because fully functional prototypes can be built very quickly.

Object-oriented application frameworks

Object-oriented application frameworks are class libraries that comprise an abstract, reusable design for interactive document centered applications as well as concrete implementations of the functionality that is common to such applications.

Application frameworks make it possible to develop user interfaces based on complex direct manipulation in a short time. They are suited for prototyping of user inter-faces that can not be composed of standard components.

An application framework provides not only user interface components but also the overall system architecture. This decreases the risk of making major architectural mistakes during prototyping and makes it easier to incrementally evolve a prototype into the target system.

4 Analyzed projects

In this section we present a brief overview of the analyzed projects in the form of two tables. Table 1 introduces for each project an acronym and a short description. Table 2 shows what kind of prototypes have been built, what prototyping approaches have been applied, and what kind of tools were used. More detailed information can be found in the appendix.

project name	project overview
Customer Advice System (CAS)	A bank software provider develops a new customer advice system. The major goal is to obtain a user interface which enables a customer advisor to carry out complex client specific tasks.
Ticket Vending Machine (TVM)	A public transport provider plans to introduce a new generation of ticket vending machines. It asks several companies to bid for the contract. Because of the importance of the quality of the user interface a user interface prototype has to be submitted as part of the bidding.
GUI for Debugger (GD)	A software house intends to adapt standard UNIX development tools to its platform. This includes the development of a graphical user interface for a command line debugger.
Multimedia Sales Support System (MSS)	A large car company wants to find out if it makes sense to support its sales force with a multimedia sales support system. Such a system has to provide a customer with written and spoken text, two and three dimensional images, and movies about the actual products.
Project Calculation and Transaction System (PCT)	A company that is specialized on building large steel processing plants wants to improve its development process with a new project calculation and control system. The software engineering department of a university gets a contract to develop such a system.
Account Representative Support System (ARS)	A large bank wants to improve the quality of work of their customer support agents with a new generation of software systems. A first application is built for the support of account representatives.
SWIFT Message Editor (SME)	A bank software provider investigates if a new way to handle inter-bank messaging (SWIFT) would be accepted by its customers. The main area of concern is if the actual clients are willing to use an interactive tool for defining their message streams.
Function Editor for Technical Systems (FET)	A research department develops a system to improve the quality of mechatronic systems (systems consisting of mechanical and electronic parts). One component of this system is an application for interactively specifying, simulating, and analyzing mechatronic systems.
Swaps-Manager (SM)	The research department of a large bank develops a prototype which permits swaps traders to define, simulate and, analyze complex deals while they are trading on the phone.

Table 1: The investigated projects in overview

		CAS	TVM	GD	MSS	PCT	ARS	SME	FET	SM
prototypes built	presentation	•		•	•	•	•	•		
	functional		•		•	•		•		•
	breadboard	•	•	•	•				•	
	pilot system					•	•	•	•	•
prototyping approach used	exploratory	•	•		•		•			
	experimental	•	•	•	•		•			
	evolutionary					•	•	•	•	•
tools used	HyperCard	•	•				•			
	interface builder	•		•	•					
	4GS					•	•			
	framework						•	•	•	•

Table 2: Prototypes built, prototyping approaches used, and tools used

5 Analysis of the application of prototyping

After this short overview of the investigated projects we summarize the findings relevant for prototyping. We analyzed the projects based on the following three questions:

- What were the reasons for building prototypes?
- What was the overall development strategy that led to the construction of prototypes?
- What is the relation between prototypes and target systems?

5.1 Goals for building prototypes

The investigated projects clearly show that prototyping is well suited to develop and communicate a vision of the future system among the members of the development team (cf. Table 3). Frequently the end users are not integrated into these prototyping cycles if they are not an integral part of the team. End users are usually consulted once a coherent vision has been built.

Similarly, prototypes help to increase the probability that IT and customer management make a decision favored by the project team. Usually, it is not important that these prototypes model the domain specific and technical aspects in great detail. It is important that they sketch the intended solution and make it easily communicable.

It is no surprise that many projects focus on testing and measuring the quality of the look and feel of applications. Surprisingly, however, only few of them obtained help from a user interface expert or from a graphics designer.

In more and more projects the analysis of the application area is considered an integral part of the development process. It is interesting that this domain specific knowledge was not acquired with the help of external experts. There was also no separate information analysis phase at the beginning of the projects. We deduce from these observations that prototyping is a valid means for knowledge transfer between developers and end users. Furthermore the observations support our thesis that both domain specific and technical knowledge have to be available in a project team. This is strongly encouraged by taking a prototyping approach.

In many projects, prototypes were built to answer technical questions. Technical questions can arise during the entire development process and they are seldom answered by team members only. Experts are frequently consulted for specific areas such as networking, databases or hardware.

In summary the following trends were observed:

- Prototypes are built to develop visions for domain specific and technical solutions.
- They influence decision making in ways not possible with written reports.
- The importance of usability in the overall quality of an application has been recognized. Nonetheless, specialists in this area are not incorporated into the teams.
- Both domain specific and technological knowledge are needed in a development team. Prototypes are an excellent means to acquire both kinds of knowledge and to evaluate it together with experts. The use of prototyping for this purpose is at odds with classical life-cycle strategies.

	CAS	TVM	GD	MSS	PCT	ARS	SME	FET	SM
generating visions	+	+	-	+	?	+	(+)	(+)	+
supporting decision making	+	+	+	-	?	?	+	+	+
evaluating look & feel	+	+	+	(+)	+	+	+	?	+
supporting analysis of domain	+	+	-	-	+	+	-	+	(+)
showing technical feasibility	+	?	+	+	?	+	(+)	+	+

Table 3: Goals for building prototypes (+ explicit project goal, (+) goal of secondary importance, - no explicit goal, ? not mentioned)

	CAS	TVM	GD	MSS	PCT	ARS	SME	FET	SM
user-developer communication	+	+	(+)	+	+	+	+	+	(+)
interdisciplinary team	+	+	-	(+)	-	+	(+)	-	+
evolutionary system development	+	+	(+)	-	+	(+)	(+)	(+)	(+)
evaluation of tools	+	-	-	+	-	(+)	(+)	(+)	(+)

Table 4: Strategies of the prototyping process (+ explicit project goal, (+) goal of secondary importance, - no explicit goal, ? not mentioned)

5.2 Development strategies and the prototyping process

The study clearly shows that in most projects the importance of communication between developers and end users has been recognized. This communication works only if suitable support is available (cf. Table 4). Prototypes have proven to provide suitable support. This finding clearly contradicts the idea underlying conventional project strategies. Here, management tries to minimize the communication overhead and tries therefore to prevent communication between developers and end users except at the beginning of the life-cycle. It is no surprise that the projects we investigated were not organized in a conventional way.

Interdisciplinary teams are an important way to establish continuous communication. Such teams consist of developers and experts in the application area. This does not necessarily imply that the latter have to work full-time for the project but they have to be fully accepted and integrated in the team. Application domain experts participate regularly when it is important to evaluate design decisions, prototypes, and development documents such as scenarios.

In most of the investigated projects an evolutionary development strategy was applied. This is a strategy that does not imply a sequence of life-cycle phases as defined, for example, in [4]. In an evolutionary strategy phases are replaced by iterative processes where different prototypes are designed, implemented, and evaluated depending on the critical domain specific or technical decisions that have to be taken. Decisions about whether to continue or abandon a project are usually taken at the end of the evaluation of a prototype. A very important step in many projects was the deployment of pilot systems to end users.

The quality of prototyping tools differs widely and the set of available tools is changing constantly. Tools such as HyperCard have lost appeal since we carried out our study, while there is a new generation of prototyping tools such as VisualAge and VisualBasic.

The general insecurity about which tool should be chosen for building which kind of prototype was also observed in the investigated projects. It was even a side goal in several projects to evaluate the quality of development tools during prototyping. This makes only sense for the development of throw-away prototypes where tools can be easily switched.

In summary the following trends were observed:

- Prototypes are an important means of communication between developers and end users. The importance of this kind of communication is increasing. From an organizational point of view, communication is facilitated by interdisciplinary teams.
- Traditional life-cycle approaches are being replaced by evolutionary strategies in projects focused on building user-friendly systems. Prototyping is today an established part of these evolutionary strategies.
- The tool market is still difficult to survey and changes quickly. For this reason the evaluation of tools has to be planned as a part of an evolutionary development strategy in developing innovative applications.

5.3 From prototypes to target system

Several of the investigated projects did not have a target system as a major project goal. Major goals were the acquisition of information about feasibility, market interest or experimental experience (cf. Table 5). We discussed this trend earlier in [3] but at this time we found our examples mostly in an academic environment. It seems that the understanding has grown that prototypes are excellent sources for innovative ideas, even in industry (cf. [11]). We even found projects with results that were so compelling that it was decided to develop a target system although this was not planned in advance.

There is no clear tendency for (parts of) prototypes to be reused for building the target system. What can be stated is that reuse makes only sense if the parts to be reused are technically sound. In the investigated projects this held mostly for user interfaces that were developed with a graphical interface builder and for information systems developed with a 4GS. Presentation prototypes were usually planned as throw aways. The reason for this is mostly strategic. It is important to explain to end users and their management from the very beginning that a presentation prototype only shows a vision. Such a prototype is implemented as quickly and cheaply as possible and is therefore likely to be

	CAS	TVM	GD	MSS	PCT	ARS	SME	FET	SM
target system planned	+	-	+	-	+	-	(-)	(-)	(-)
target system built	+	+	+	-	+	-	+	-	-
reuse of building blocks	-	-	+	-	+	-	+	(+)	+
seperate teams	-	+	*	(+)	-	-	*	-	-

Table 4: Relation between prototype and traget system (+ explicitly yes, (+) probably yes, (-) probably no, - explicitly no, * yes, but transfer planned)

thrown away.

Not many teams in large traditional companies are able to apply prototyping because developers lack the required technical or methodological skills. This results frequently in a separation between the prototyping team and the team that engineers and maintains the target system. Our investigation shows that this can result in various problems. Some organizations have recognized the inherent danger of such an approach and they take precautions, if they have to separate teams at all. The most important point is to ensure the know-how transfer between the teams. In several of the investigated projects this was achieved by letting some of the developers of the target system participate in the prototyping process for a limited time.

In summary the following trends were observed:

- The benefits that arise from applying prototyping to acquire information about feasibility, market interest or to gain experimental experience have been recognized. Some of the investigated projects did not even have a target system as a major goal.
- The reuse of prototypes for the development of a target system can only be recommended if the development tools produce prototypes with a clean system architecture. Many presentation prototypes are planned as throw aways for this reason.
- There is a strong trend for one team to carry out the entire development cycle. Due to lack of know-how many organizations are still dealing with different teams for prototyping and the development of target systems. This problem is at least clearly recognized in many companies.

6. Analysis of the application of tools

In general, tools were used as intended by their developers. The only exception was HyperCard which was used mostly to implement mock-up prototypes. Many projects have made a sub-optimal use of available tools. The reasons vary: first of all, there seems to be a lack of knowledge about the tool market. In addition, there is a well-known reluctance to use new tools. Finally, there are projects where customers make an unjustified but mandatory selection of development tools as part of a contract.

Another aspect we observed is that projects have to be aware of the dichotomy, or even contradiction, between sound software architectures and systems which have been developed "surface down". We believe that it is hardly possible to develop long-lasting and flexible software by starting at the user interface, as those system parts, not related to the user interface, will lack substance.

The consequence of this finding is that user interfaces should not be derived or generated from the kernel, such as the data model, as this approach ignores the potential of an innovative interface and the important issues of usability and adequate handling.

Domain specific kernels and interactive interface parts should be developed complementarily. This is backed by a current tendency in the tools area: more and more integrated development environments (not CASE tools!) support the construction of both the user interface and the nucleus of an application. But beware of the management illusion that tools are the magic wand for every software engineering problem. Development tools are only useful if employed within a sensible methodological framework.

7 Conclusions

Prototyping is now used more consciously than in recent years. As an illustration, the reader should note that none of the investigated projects have followed a traditional life cycle or waterfall approach. In all these projects, prototyping has been part of a deliberate evolutionary strategy, on the operative level, if not by decision of the senior management. In addition, prototyping was well-planned and not taken as an excuse for shipping half-baked systems to customers.

This conscious approach to prototyping seems to be the key to the high percentage of successful projects in this study. Note, that we did not preselect the most promising projects for this study. In a related case study conducted 5 years ago [10] the authors found a majority of projects that took a combination of life-cycle and prototyping approaches and failed.

A newly emerging trend, we observed, is to use prototypes as a vehicle for developing and demonstrating visions of innovative systems. This source for innovation can be tapped not only for individual software projects but also for various kinds of marketing research and field studies.

References

[1] Bäumer D., et. al.: Prototyping von Benutzungsoberflächen (in German), UBILAB Technical Report 94.9.2., Union Bank of Switzerland, UBILAB, 8021 Zurich

[2] Bischofberger W.R., Pomberger G. (1992): Prototyping-Oriented Software Development–Concepts and Tools, Springer-Verlag.

[3] Budde R., Kautz K., Kuhlenkamp K., Züllighoven H. (1992): Prototyping– An Approach to Evolutionary System Development., Springer-Verlag.

[4] Boehm, B.W. (1976): Software Engineering, IEEE Transactions on Computers, Vol. 25, No. 12, Dec. 1976

[5] Carey, J.M., Currey, J.D (1989): *The Prototyping Conundrum. Datamation,* June 1, 1989, pp. 29-33.

[6] Floyd C. (1984): A Systematic Look at Prototyping, In: Budde R. et. al. (eds.), Approaches to Prototyping, Springer-Verlag.

[7] S.S. Gordon, J.M Bieman: *Rapid Prototyping: Lessons Learned*, IEEE Software, January 1995, pp 85 -96.

[8] Gordon, S., Bieman,J. (1991): Rapid Prototyping and Software Quality: Lessons From Industry. Technical Report CS-91-113, Department of Computer Science, Colorado State University.

[9] Goma, H. (1986): *Prototypes - Keep Them or Throw Them Away?*. In: Lipp, M.E. (ed.): Prototyping - State of the Art Report. Pergamon Infotech Ltd, Maidenhead, 14:4, pp 41-54.

[10] Kieback, A. et al. (1992): *Assessing Prototyping Projects*, Proc. of the HICSS 25, Koloa Hawai, pp 564 -567.

[11] Lichter H., Schneider-Hufschmidt M., Züllighoven H. (1994): *Prototyping in Industrial Software Projects - Bridging the Gap between Theory and Practice*, IEEE Trans. on SE, vol 20, no 11, pp 825 - 832.

[12] Sobol, M.G., Kagan, A. (1989): *Which Systems Analysts are more Likely to Prototype?,* Journal of Information System Management, Summer 1989, pp. 36-43.

[13] Tillmann, George (1989): *Prototyping for the Right Results.* Datamation, April 1, 1989, pp 42-45.

[14] B.A. Myers, M.B. Rosson: Survey on User Interface Programming, Proceedings of the SIGCHI´92, pp 195-202.

Appendix: Project descriptions

A.1 Customer Advice System (CAS)

Task

A software house of a consortium of banks intends to develop a family of applications supporting different kinds of customer support agents. These applications have to provide excellent user interfaces in order to handle complex customer related activities. The software house was very experienced in developing traditional screen form based mainframe applications but had no experiences in GUI based client/server applications. After a conventional approach failed it was decided to restart the project, to apply object technology, and to switch to GUI based PC applications as front ends. To support the development team external consultants were hired.

Prototyping

First a prototype was developed to build a shared vision between the developers and their management. This prototype was implemented by the consultants in HyperCard. The prototype which shows only a small subset of the overall user interface was evaluated by the developers and their management but not by the end users. HyperCard was applied because mock-up prototypes could be developed in a very short time. Later on, CASE/PM was used to build breadboards and to gain experience with the OS/2 Presentation Manager. Because CASE/PM was not optimally suited for this task the Smalltalk based PARTS environment was used to develop further prototype generations. It was never intended to reuse parts of the prototype code for the target system, which was developed from scratch in C++. The presentation and functional prototypes had only one purpose, to improve information exchange and mutual understanding between developers, management, and end users.

Procedure

In this project, emphasis was laid on the thorough analysis of the application area, the integration of the end users, and the development of components easing the construction of a homogeneous application family. The development of presentation and functional prototypes was an important part of the overall strategy. The application of prototyping improved the project management and planning considerably throughout the whole development cycle.

A.2 Ticket Vending Machine (TVM)

Task

A public transport provider plans to introduce a new generation of ticket vending machines. Because of the importance of the quality of the user interface, a user interface prototype has to be submitted as part of the bidding. A vending machine producer wants to participate in the bidding but does not have the know-how required to implement a software prototype. For this reason, it subcontracts the development of the prototype to a university department experienced in constructing highly interactive user interface prototypes.

Prototyping

Prototyping was applied for several reasons. The transport provider used the prototypes for selecting the "best" supplier. The developers pursued two major goals: generation of new ideas about how a ticket vending machine can be designed, and practical evaluation of the usability and end user acceptance of several functional prototypes.

To gain ideas about how a vending machine could look like, several presentation prototypes were built in parallel by different specialists such as graphics designers, software developers, and human factor specialists. From these prototypes a system vision was synthesized. Based on this vision two functional prototypes were developed. One of them provided a very flexible, stateless user interface while the other tried to closely monitor and guide a ticket buyer. SuperCard, a HyperCard-like tool, was selected to implement the functional prototypes because of the great flexibility it provides for user interface design.

Procedure

The project can be divided into two phases. In the first phase, the application area was analyzed and possible solutions were evaluated. This phase served also to evaluate several prototyping tools. Furthermore, elementary building blocks for ticket vending machines were constructed with SuperCard. These building blocks were used later on to speed up prototype development. Because the mock-up prototypes of the first phase could not be used for intensive testing two functional prototypes were developed. They were evaluated in large field studies where ticket buyers in train stations used and rated them. No employees of the ticket vending machine producer participated in the prototyping effort. For this reason no know-how transfer occurred.

A.3 GUI for a UNIX-Debugger (GD)

Task

A large software house intends to adapt standard UNIX development tools to its platform. This includes the development of a graphical user interface for an existing command line debugger. Because the development department did not have the required experience for building graphical user interfaces, the project was subcontracted to a team of specialists.

Prototyping

In the first phase, a user interface prototype without any functionality behind it was developed and evaluated with the clients. It was implemented with the SNI DialogBuilder and it served to investigate alternative user interfaces. Several breadboards were developed after developers and clients considered the user interface prototype satisfactory. These breadboards served to investigate alternative communication and integration mechanisms. Based on the experience gained with the breadboards a functional prototype was developed which was then evolved into the target system. The functional prototype was evaluated by the quality assurance team of the client.

Procedure

The project was planned as a prototyping project from the beginning, although this was not intended by the client. The client wanted a conventional development project with a fixed price. For this reason, conventional milestone documents had to be delivered. The gap between the developers of the prototype and the developers of the target system was continually reduced during the project. This was achieved by having developers of the client participate in the evolutionary development process. This measure resulted in a successful know-how transfer in respect to the development of graphical user interfaces as well as the concrete realization of the debugger front end.

A.4 Multimedia Sales Support System (MSS)

Task

A large car company wants to find out if it makes sense to support its sales force with a multimedia sales support system. For this reason it contracts a software house to develop a prototype. This system has to provide a prospective car buyer with written and spoken text, two and three dimensional images, and movies about its actual products. The system has to be complementary to existing brochures and technical descriptions currently distributed during sales talks. The system should be able to gather special requirements of customers and prepare a sales contract based on this information. The system is planned as a prototype to help investigate the feasibility and commercial attractivity of such a product.

Prototyping

Prototyping was applied for two reasons. First, to find out how well a multimedia system can support sales staff. Second, to gain experience in developing multimedia systems.

Two prototypes were built, a presentation prototype which provided a better understanding of the basic requirements, and a functional prototype which was incrementally extended with support for different media. Both prototypes were built with SX/Tools.

Procedure

The first step was to develop a vision of how the planned system could look and work. The result of this step was a presentation prototype and a study that was evaluated by the car company. Based on the resulting feedback a functional prototype was developed in five months. This prototype comprised the complete user interface but only parts of the functionality. User-friendliness was not explicitly considered but the prototype was evaluated by a video production specialist who provided useful input. The final functional prototype was presented by the car company at several exhibitions.

A.5 Project Calculation and Transaction System (PCT)

Task

A company specializing in building large steel processing plants wants to improve its development process with a new project calculation and control system. The software engineering department of a university gets a contract to develop such a system for a network of personal computers.

Prototyping

Prototyping was applied in the context of an evolutionary development strategy. The target system was developed incrementally in close cooperation between developers, clients, and their management.

Two kinds of prototypes were developed. First, a presentation prototype helped the developers to learn about the requirements and demonstrated the technical possibilities. In a second step a series of functional prototypes was followed by a pilot system which was finally evolved into the target system. All prototypes and the target system were implemented in the 4th Dimension 4GS.

Procedure

The client started the project by writing a 40 page requirements specification including a scenario. Based on this specification a first prototype was developed in three weeks. During the evaluation of this prototype it became obvious that the requirements specification had been partially misunderstood and lacked many important details.

Based on the feedback, a series of functional prototypes, which were regularly evaluated by developers and end users, were implemented. Every few months formal project reviews were carried out that served to specify the most important steps for the next development cycles together with the management of the end users. Subsequently the prototype was deployed as pilot system. The pilot system continued to evolve until the target system was finished. At this time it was agreed that after one year of application, a technical and functional redesign would be carried out.

A.6 Account Representative Support System (ARS)

Task

A large bank wants to improve the quality of work of their customer support agents with a new generation of software systems. Until now the customer support agents have to use up to four different terminals and still do not get all the support they need. The support of account representatives is chosen as the first application area.

Prototyping

Prototyping was applied to improve the communication between developers and end users. Emphasis was laid on short feedback cycles to find innovative solutions. An evolutionary development approach was envisioned from the beginning.

Several prototypes were built. Exploratory prototypes were implemented with Intermedia and SuperCard. Exploratory prototypes and breadboards were built with the ET++ application framework. An evolutionary prototype that evolved into a pilot system and then into the target system was developed with Windows/4GL.

Procedure

The project started with an extensive analysis phase during which the developers learned about the working environment and needs of the account representatives. The next step was to establish a project team consisting of end users, developers, and external consultants. During the requirements definition process several prototypes were built and it was decided to use Windows/4GL as an evolutionary development environment. During this time the vision of the target system changed from hypertext toward a database application. Once the team was consolidated and the development environment was chosen the project proceeded smoothly with the incremental development of a series of prototypes, pilot systems, and the final product.

A.7 SWIFT Message Editor (SME)

Task

A bank software provider investigates whether a new way to handle inter bank messaging (SWIFT) would be accepted by its customers. The main area of concern is whether the actual clients will accept the use of an interactive tool for defining their message streams.

Prototyping

Prototyping was applied as part of a market study for an improved version of an existing software product. Prototypes were used to demonstrate the new product idea and to test the reaction of the client banks.

First, a presentation prototype was developed in Smalltalk on a PC under OS/2 to investigate different user interface variants. Afterwards, a functional Smalltalk prototype was implemented and used for field tests.

Procedure

During a short analysis and design process the development team collected information about the application area, made their first steps in the area of object technology, and built a first presentation prototype. The functionality of the prototype was determined by an existing collection of scenarios.

After an internal evaluation of the prototype by developers and marketing specialists, the presentation prototype was evolved into a functional prototype. The latter was then evaluated with potential customers and it was decided to develop a product. Technology transfer was achieved by transferring a part of the prototyping team into the product development team.

A.8 Function Editor for Technical Systems (FET)

Task

A research department develops a system to improve the quality of mechatronic systems. One component of this system is an application for interactively specifying, simulating, and analyzing mechatronic systems.

Prototyping

The application of prototyping was explicitly planned for several reasons. There was no experience with quality insurance for mechatronic systems within the development team. It was not possible to specify the planned system in detail. It was intended to gain further experience with a Smalltalk development environment.

Four prototypes were built during the development process. The first was a horizontal breadboard which served to investigate the approaches to interactive modeling of mechatronic systems. The second prototype had the same functionality but was completely reengineered. The third prototype evolved from the second prototype and covered all functionality asked for by the end users. The fourth prototype was a pilot system with the same functionality but with a clean re-engineered architecture.

Procedure

The first prototype served as a vehicle that permitted the developers to learn as much as possible about the application area. Based on this prototype a requirements definition was written that defined the functionality of the second prototype. This prototype was used to trigger a discussion about the possible advantages of software support for quality insurance. A wide range of people from potential users to managers participated in these discussions. The result of this process was a set of further requirements that were fulfilled by the third prototype. The latter was developed with the intention to deploy it as pilot system. The prototype was then reengineered to obtain a maintainable system with a clean system architecture.

A.9 Swaps-Manager (SM)

Task

The research department of a large bank develops a prototype of an application which permits swaps traders to define, simulate and analyze complex deals while they are trading on the phone. The project is started to prove what can be achieved with available technology. It is not planned to develop a target system. The major goal of the project is the evolutionary development of a system of exceptional usability.

Prototyping

It was obvious that the goals could only be reached by applying an evolutionary, prototyping-oriented development strategy. The evolutionary prototype was implemented with the ET++ application framework. The first functional prototype was implemented to verify that a user interface could be developed that allows the handling of complex deals while talking on the phone. It supported only the standard cases necessary for realistic field testing. The prototype was then evolved into the pilot system.

Procedure

The development of the first prototype consisted of two main activities. The analysis of the domain specific and mathematical foundation which was carried out in close cooperation with the end users, and the implementation and evolution of the functional prototype. This was a cyclical process during which the developers evolved their vision which was regularly evaluated by the traders. To guarantee short development and evaluation cycles the development was even partially carried out in the trading room.

The step from the functional prototype to the pilot system involved the definition and implementation of the detailed mathematical models employed for calculating the deals. This activity was carried out by the developers and financial analysts. The end users were not involved in this process.