Abstract

In this paper we discuss the use and development of prototypes during the requirements phase of the software development lifecycle, and contrast this with the more traditional use of prototyping during the development phase. We show how this approach has been applied in the development of a broadband email client, and we argue that the prototypes developed have significantly increased the quality of the final product.

Keywords: Software Engineering, Requirements, Prototyping

Introduction

In traditional software engineering/user interface design methodologies requirements are generated as part of the analysis phase, and these are fed to the developers who may produce prototypes as part of the design phase of the software lifecycle[6]. We have taken a different approach in this paper, where we describe the use of prototyping as part of the development and refinement of requirements during the analysis phase.

Beginning prototyping during the analysis phase is an effective way of defining and validating functional and user requirements, features and information architecture. Prototypes can be built concurrently as functional requirements and user requirements are gathered, collected and iterated. Prototyping conceptual designs during the analysis phase can give the following benefits:

1. Provide early feedback from team members and users on a prototype before features and requirements are finalized; this is done through group brainstorming sessions, user studies, market research and design reviews;
2. Allow product management, project management and engineering an understanding of the scope of the product to plan a feature roadmap and project schedule;
3. Allow developers to spot potential feature problems and usability problems before development begins; and
4. Allow exploration of interface concepts, information architecture, interface design, and features through an iterative design process.
The use of prototypes during the requirements phase was used during the development of the first version of an email client called @Home Mailpoint. The first version of the software was developed from April 2000 to November 2001. The email client was initially developed to provide @Home broadband subscribers with:

1. A broadband communications application that integrated broadband content (greeting cards, invitations, photos) by utilizing Excite @Home web properties and creating a robust email user experience through the use of rich text messaging, personalization and use of rich media to send and receive email messages;
2. Reduce customer care support and configuration costs for @Home broadband subscribers utilizing third party email clients like Outlook Express or Netscape Messenger; and
3. Offer @Home subscriber’s remote access to their email via a web-based interface.

**Prototyping**

Prototypes are mockups of the physical representation of an interface. Prototypes are used to communicate product functionality. There are different types of prototypes[5,7]:

- **Passive** – a prototype that sits there and does nothing. An example is a pencil drawing.
- **Active** – A model that functions either as a simulated or partial implementation, an active prototype can act as the real thing to a certain degree
- **High Fidelity** – A prototype that resembles the interface it depicts.
- **Low Fidelity** – A prototype that is often conceptual and quickly iterative. A low fidelity prototype is often pencil and paper or a whiteboard and post its.
- **Horizontal** – A superficial representation of an interface system that contains most of the interface, but nothing behind the screen. It will exhibit minimal activity.
- **Vertical** – A narrow slice through the system that exhibits most or all of the complete interaction and simulates the behavior of the application.
- **Demonstration** – A prototype that is used for marketing purposes. It often appears as a fully interactive and finished product but usually consists of a highly polished and graphic interface. There is little to nothing running behind the scenes (usually referred to as “smoke and mirrors”.)
- **Wire frame** – A passive high fidelity prototype that contains screen elements and layout for a screen, but doesn’t contain polished graphic elements. This is usually black and white.
Prototypes can be developed for different reasons:

1. Exploration of architecture and functionality;
2. Communication of software functionality to customers, product managers and designers; and
3. Proof of concept or test of feasibility.

Prototyping is part of an iterative design process. A prototype should never be considered the final product, especially if it is highly polished and graphically visual. It is often mistaken for the final product, especially if it closely resembles the user interface of a finished product. Customers, clients and other often mistake demonstrations for actual products. Engineers and designers can find it easier to prototype than code an application because they are only prototyping the application behavior and interaction. Prototypes are quicker and cheaper to iterate instead of programming code changes.

**Product Information**

An individual in Excite product management initially envisioned the @Home Mailpoint client. This individual envisioned four objectives to drive the project:

1. Development of a broadband email client to replace third party email clients and reduce customer support costs;
2. Creation of broadband communications application that offered @Home subscribers a robust email user experience and integrated content from the Excite Network;
3. Offer remote access to @Home subscribers via a web based interface; and
4. Re-architect the backend server capability to support an email client and remote email access, and reduce storage costs.

During the requirements phase, UI design was concerned with the development of the broadband email client and communications applications, and participated in the brainstorming sessions around the backend re-architecture discussions.

**Process**

The @Home Mailpoint team followed a user centered design methodology to develop the product. During the requirements phases the following was accomplished:
1. **User Profiling** – Demographic data pertaining to @Home subscribers was collected. Data regarding email users and rich media users; profiles were created that represented and reflected the demographics of the @Home subscribers. It was decided to profile families, as there is more than one user in a typical @Home subscriber home.

2. **Task Analysis** – A user study was conducted regarding how the targeted audience utilized email application. Specifically targeted were users of third party applications (Outlook Express), competitive products (AOL) and webmail products (Yahoo Mail, MSN Hotmail). A report was produced of the findings of the user study. We refined the user profiles as an output of the task analysis.

![Figure #1: Users’s Model of the Email Platforms](image)

a) **User Goals and Tasks:** User goals and task were defined to provide design and development qualitative objectives. No quantitative objectives measures were defined. The initial user goals focused on the broadband nature of the email application[2,4]:

<table>
<thead>
<tr>
<th>Goal</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communications with family and friends</td>
<td>Maintain and increase the level of communication between family members and friends through the use of messaging</td>
</tr>
<tr>
<td>Convenience</td>
<td>Allow users convenient access to new and recent email anywhere.</td>
</tr>
<tr>
<td>Efficiency &amp; Productivity</td>
<td>Experience a high level of efficiency and productivity through the use of messaging</td>
</tr>
<tr>
<td>Ease of Use</td>
<td>Experience an easy to use product that allows the user to do any task associated with email.</td>
</tr>
</tbody>
</table>
b) **User Scenarios** - User scenarios were created to help design and define the information architecture and interaction design for the @Home Mailpoint product and define the back end re-architecture.

Concurrently, product management researched functional requirements. Engineering researched technical requirements and backend considerations. Cross-functional brainstorming sessions consisting of product management, engineering and UI design were conducted. The brainstorming sessions involved discussions of the user profiles, task flow and high-level exploration of email functionality for a broadband communications application and re-architecture of the @Home mail backend. Ample use was made of sketchpads, post its and whiteboards.

Once the user profiles, task analysis, user flows, user goals, user tasks and scenarios were completed, we created a first draft of a User Experience Requirements Document (UERD). Product management had not yet created a Marketing Requirements Document (MRD) detailing functional requirements. The team proceeded to explore conceptualization of the email product and began prototyping without the functional requirements defined. Because the focus of the initial product was a broadband communications application with integration of Excite network content, the scope of the requirements was rather extensive. It was decided that starting prototyping and brainstorming around concepts would help in the process of defining the functional requirements for the product. This way product management, engineering and UI design would be able to view visual concepts of the product and functionality. Also, the email product was a cross-company, cross-departmental initiative and involved input from several divisions of the company to determine functionality, priorities and scheduling. The prototypes would function as visual representations of product concepts the team was discussing. The team conducted five rounds of prototyping during the course of the requirements phase.

![Wire Frame Prototypes from the third and fourth rounds of prototyping](image)
Prototyping

The first round of prototyping consisted of drawing interface elements and functionality using sketchpads, post its and whiteboards. Product management and UI design collaborated together in participatory design sessions. We established the initial information architecture based on inputs from the user studies and user requirements, and refined the architecture through discussions. The paper prototypes focused on functionality definition and flow of the application. The prototypes were used in brainstorming sessions to discuss and define functionality for the @Home Mailpoint product.

The second round of prototypes consisted of black and white wire frames constructed in Adobe Illustrator. The wire frames detailed conceptual designs for an email client and focused on navigational organization and integration of Excite Network content into the broadband email application. The email team utilized the wire frames to facilitate further discussion around the nature of a broadband communications application.

The team iterated conceptualization and further explored themes of what constituted a broadband email application for the third round of prototyping. We continued utilizing Adobe Illustrator to create black and white wire frames. We focused on creating a broadband email application utilizing content from Excite Network properties (Blue Mountain Arts, Webshots, etc.), and explored the types of tasks a consumer would want to use for asynchronous communication utilizing an always-on broadband connection. The third set of prototypes were passive and horizontal in nature, since they consisted of approximately over 40 screens of a robust communications application. We further refined the navigational structure to reflect use of interface elements according to Macintosh and Windows guidelines. Product management, engineering and design continued meeting on a weekly basis reviewing the wire frame prototypes. The prototype allowed product management to view the scope of the initial vision before defining final functional requirements for the project.

Backend Re-architecture

During the four successive iterations of the prototyping exercise, the backend re-architecture technical requirement emerged. A business decision indicated that the @Home mail architecture would change from a POP to an IMAP protocol and the architecture would allow aging of messages to allow costs saving on storage space and hardware. To change the architecture of the email backend would require a considerable amount of engineering resources and time.
Implementing IMAP on the backend architecture allowed cost saving which would enable development of a web based remote access product for @Home subscribers in addition to the client interface. There currently is no web based remote access product for @Home subscribers. To access email remotely when they are off the @Home Network, @Home subscribers must use a dial up number and configure a POP client to receive the email messages. An @Home subscriber would call @Home Customer Care to request a special remote account and then configure the POP client. In addition, if an @Home subscriber did not possess a dial up ISP account, and went through @Home for dial up access, they would pay an extra fee.

During the initial contextual inquiry into how people used email in the home, users were familiar and comfortable using a web interface to access email messages. After reviewing all technical and functional requirements product management decided to narrow the scope of the release to:

1. Deliver an email client application with basic messaging functionality to users;
2. Offer remote access via web interface that allowed users to access email anywhere;
3. Re-architect the backend server configuration and move from a POP to IMAP protocol, and age messages off the server after \( x \) days.

The prototypes were intensely useful during the backend server re-architecture discussions because:

1. They provided a discussion for what was important to the user versus what was important for the business.
2. The prototypes provided project management with an estimate of the features, resources and schedule of the first release. Once preliminary schedules were completed, engineering, design and product management decided that the scope of the project exceeded the bandwidth of resources and priorities. Decisions were made to limit the functionality of the email product vision to a basic email. The features for a broadband email application were scheduled for future versions. This was relatively easy to do since we already had an overview of the features from the prototype.
3. Viewing the prototypes allowed product management and UI design to reexamine user requirements and functional requirements. After conducting a research survey pertaining to remote access and message aging, it was determined that the objective for the first release would be remote access with message aging on the server to allow for cost savings.
Product management completed an initial draft of the MRD. The user requirements and functional requirements were refined and iterated so they reflected the new product objectives goals.

**Continuing Prototyping**

The next round of prototyping focused on creating a basic email application. Major design concerns were:

1. Should we present messaging aging to the user;
2. How would we present message aging to the user; and
3. How would we convey the benefits of the remote access solution to the user.

In the initial contextual inquiry, it was noted that novice to intermediate users of email didn’t grasp the concept of mail saved locally versus mail saved on a server. In using a competitor product that ages mail, it was noted that a) users didn’t know mail was being aged and deleted; and b) users didn’t know that they could save mail locally to a filing cabinet because the default preference was turned off to save mail. The feeling was that the company imposed its priorities of aging mail on the user in a transparent fashion that exhibited bad usability; the competitor email product didn’t display any information regarding its aging policy to its end users except in its help system.

UI design and product management decided that it would be wise to present aging to users, because we would be changing their existing mail storage policies, but also enhancing their user experience through the combined solution of the email client and remote access product. We decided to represent the aging as through the positive theme of remote access, as the survey revealed the overwhelming customer support for remote access of mail via a web mail interface.

In the fourth round of prototyping a separate visual design team came in to design conceptual pieces around the email client product and how to present aging. The visual design team worked with the user interface designer iteratively and produced several designs. The designers utilized Adobe Photoshop and created high-fidelity passive prototypes. By the end of a three-week period, the visual design team utilized a high fidelity active demonstration of their concept piece to the whole team. The final visual design concept demonstration utilized some interesting metaphors, visual paradigms, but the concept design did not meet the user requirements and functional requirements. After discussion between UI design and product management, it was decided to return to a similar navigation and layout from the round 4 prototypes (left to right layout). The visual design team returned to the drawing board and created Photoshop prototypes focusing on left hand navigation for the email client.
During this time, it was decided to begin work on the web mail interface, as all time and resources had been devoted to the client interface. The UI designer focused on the web mail interface and designed three prototypes. The initial prototype utilized existing design guidelines for Excite web products, but it was determined that the guidelines were inadequate for application design, since they were meant for content design. The second prototype utilized a tabular navigation found on some Excite applications, but there were no guidelines pertaining to the tabular navigation. The third prototype utilized the visual elements found on the client and focused on similarity and consistency of interface between the client and web mail interfaces. Product management and UI design decided the best course of action was to use the third prototype design as the proposed web mail interface due to schedule and resource constraints, and lack of available guidelines pertaining to web application development at Excite@Home.

By the end of the fifth round of prototyping, the user requirements and functional requirements were almost finalized but the team decided to get user feedback before finalizing all requirements. We had determined the functional and user requirements after 5 rounds of prototyping, and determined a course of direction for the Mailpoint email client and web mail products. We usability tested the product a total of three times, from August 2000 to October 2000 using high fidelity active prototypes.
The usability feedback proved valuable, and it was decided that the way remote access and aging were presented to the end user was correct, though the design was refined with the feedback from each round of testing.

**Problems during Prototyping**

During the prototype process to major problems occurred:

1. **Multiple design organization existed at Excite.** There were overlapping roles and different process methodologies and questions of ownership for design decision. This led to confusion and conflicts regarding decisions during the requirements phase and prototyping. Questions were raised regarding who was responsible for feature decisions, project management, process, roles and design objectives. This prolonged the requirements phase and prototyping as time was spent to define roles and clarify responsibilities.
for the project. At a later date, the three design groups was consolidated into one organization so the design process could run efficiently and decisions could be made quickly.

2. **Product management scoped down features of the product due to budgetary and schedule constraints without consulting UI design.** The decision to cut features occurred after the fifth round of prototyping. It occurred without product management consulting UI design to access how the cutting of features would affect the user experience and product usability. Two key features cut were the status bar and offline capability. At a later point in the development phase, software engineering provided work around solutions to the status bar and offline capability. The solutions did not adequately meet the needs of the users, and contain usability problems. The two features require further analysis, definition and prototyping for the next release of the product.

3. **Engineering didn’t understand the value of prototyping.** During the course of the prototyping, product management and UI design found the prototyping extremely useful for discussions, and gathering feedback from team members. However, engineering raised issues about the value of prototyping, and felt that product management and design spent too much time iterating the prototypes and holding brainstorming sessions. It wasn’t until later, in the development phase of the product life cycle, that one of the engineers commented that time was being saved and engineers were only working 8-10 hours days instead of 12-16 hours days, due to the degree that the initial requirements were defined and then inputted into a User Interface Specification during the definition phase.

**What was learned**

Prototyping proved valuable to the email product team during the course of the requirements phase. In the beginning of the project a product manager had the unique vision of developing a broadband email application that would enable remote access for @Home subscribers and save @Home money on customer support and hardware costs. He knew what types of features the product should contain but didn’t know how the product would integrate with content, or behave as an application. The prototyping during the course of the requirements phase allowed all members of the email team (engineering, product management, design) to participate in the creative process and design discussions, and provide insight and definition into the @Home Mailpoint product, and allowed the team to achieve the following.
Gather Feedback

The prototypes allowed the email team to view and visualize the broadband email application vision and provide feedback on navigation, functionality and backend server re-architecture. Feedbacks from the email team allowed the design team to quickly and iteratively produce and refine several rounds of prototypes. The email team was rather large, consisting of 3 separate engineering teams, including web engineering, client software engineering, and network engineering. There were three different product managers, one customer support representative, one project manager, and three separate design teams at the beginning of the project. The email team held twice weekly design and brainstorming sessions led by the UI designer. These meetings provided a place to gather feedback on the prototypes, discuss and define requirements and brainstorm around the prototypes.

Aid in Discussions

The prototypes functioned as valuable aids in discussions, particularly in the backend server re-architecture and message aging policy discussions. The team initially knew that it was re-architecting the backend server configuration and implementing message aging, but did not know how both would affect the user experience. The initial contextual inquiry provided insight into how users dealt with aging on a competitor product. Outputs from the user study into the prototyping helped define a) the functionality of the backend re-architecture, b) the choice of an IMAP protocol rather than a POP protocol to support remote access and enable users access to new and current messages on the mail server, and to enable message aging on the server backend.

Also the product development was a cross-organizational and cross-company initiative and involved discussions with several groups who owned pieces we would integrate into the email application. The prototypes allowed the email team to produce early concept designs to discuss with other people from other parts of the company about the feasibility of the functionality and integration of their product with the Mailpoint product.

Aid Feature Definition and Schedule

During the course of the prototyping, engineering explored various development platforms to use to develop the Mailpoint product. The prototypes enabled engineering to evaluate what types of features they would be implementing and to determine the best course of action in choosing a technology platform. Based on the prototypes, engineering and product management produced and iterated schedules and feature lists several times during the course of the requirements phase based on the visual representation of features from the prototypes. Engineering
and product management were able to produce schedules, narrow the scope of the features and schedule, and balance user needs and business goals.

**Aid in Defining User and Functional Requirements**

The initial project of designing a broadband communications application was extremely large in scope from both a feature perspective and a time perspective. The initial user research, profiling, and task analysis helped define the initial user requirements and functional requirements, but the scope of the project was extremely large and many people were involved in the decision process. The prototypes functioned as a visual representation to communicate the intent and scope of the project vision to team members.

Extremely helpful in the prototyping process were the paper prototypes and wire frames. Team members participated in Round 1 of the post it and sketchpad prototypes, which helped identify the initial functionality for a broadband email application.

**Aid in Defining Concepts**

While email has existed for 30 years, the idea of a broadband email application is relatively new. When the email team started this project, we sought to define new application areas that had not been created and explore new ways for end users to asynchronously communicate using an always on broadband connection. Many exciting and new concepts came out of the exploration of features, user needs and concepts during the prototyping phase. Though many of these features and concepts were cut for the initial 1.0 version, they have help provide understanding and strategy for future versions of the email product, and also integration of mail with other forms of communications, particularly instant messaging. As the team started the next version of the Mailpoint product, the team revisited the email prototypes and started defining an email strategy for future versions of the Mailpoint product.

**Conclusion**

In the requirements phase of the development cycle, prototyping helps define functional and user requirements, gather feedback from customers and team members, and functions as a visual representation of concepts to communicate the intent of the design to people.

In order to effectively use prototyping during the requirements phase, we must consider the following:
• **Communicate value of defining user requirements and functional requirements during requirements phase to team members and upper management.** The requirements phase is an important phase in the software development cycle because most planning is done during this phase for the design and development phases. If the requirements are not adequately defined, the schedule time and budget will increase during the design, definition and development phases.

• **Designate team members responsible for prototyping and track iterative prototype design.** If you have a large team, designate team members responsible for prototyping. If there are multiple team members, make sure they talk to each other regarding the types of prototypes they are designing. Also, if prototyping multiple iterations, track the iterative prototype designs and changes in the design with each round of prototyping. Document design decisions so during the definition and design phases, the team can refer to these decisions.

• **Define types of prototypes to be designed and built and purpose of prototype.** There are different types of prototypes (active, passive, high fidelity, low fidelity, horizontal, vertical, demonstration, wire frame). Prototypes can be constructed for different reasons: designing new features, usability testing, testing platform, etc. There may be instances where one type of prototyping may be better for certain phases, or save time or money (like paper prototyping).

• **Hold regular brainstorming sessions and design review and involve all members of the team in the design and prototyping process.** Make sure all appropriate stakeholders are involved in the brainstorming and design review sessions so they can provide input and be aware of the requirements analysis and prototyping.

• **Define and refine functional and user requirements with iterative prototypes.** If the scope of a product appears too large for the schedule, narrow the scope of the functional and user requirements and set realistic development goals. Especially during the requirements phase, the defining of functional requirements and user requirements is an iterative process, just like prototyping.

• **Usability test prototypes in the requirements phase.** The earlier prototypes are tested, the earlier the team can identify potential usability problems. You are also incorporating user feedback into your product and following a user centered design process by usability testing prototypes. Also set qualitative and quantita-
tive measures for testing to track iterative prototype improvements when testing during the requirements phase.

- **Don’t let a prototype or product demonstration be mistaken for an actual product.** Sometimes during the course of development people can mistake a prototype for an actual product, especially if it is an interactive aesthetically pleasing demonstration. A prototype or demonstration should never be mistaken for the actual product, especially if built in the requirements phase. The team still needs to perform the actual definition and design work done in the design, definition and development phases. The interface should be specified in a User Interface Specification. Usability testing of interfaces should continue in the design, definition and development phases to gather user feedback regarding usability problems.

- **Don’t let a prototype function as a specification.** When development of a product is done extremely quickly, prototypes often function as specifications, especially if they are displaying active behavior of the application. If time permits, specify interfaces so the intent of the design can be communicated to the development team and the team can track design decisions. Organizations will find that they save time and money, and avoid frustration in the development phase by specifying products and documenting the design.

Our experience has shown the value of developing and using prototypes during the analysis phase of the software development lifecycle. We have show how this approach has been used in the development of a commercial mail client, the problems encountered, but also the significant increase in the quality and functionality of the final product that was delivered.

**References**


