

A Lightweight Development Process for Implementing Business Functions on the Web

Abstract

From a software engineering perspective, moving a business software application to the web presents interesting challenges. These include frequent changes to requirements, changes to the underlying business function as a consequence of moving to the web, tight time and cost constraints, and producing quality user interfaces within the restrictions of the web environment. This paper describes a lightweight, two-phase process that integrates advantages from incremental development, throwaway prototyping, and waterfall process models to address these challenges. A successful development experience using this process is summarized.

1 Introduction

Businesses of all kinds are clamoring to migrate key functions to the web. As contract software developers, we have seen much of this development effort occurring in a haphazard and rushed manner, with little room left for a formal process. This lack of management makes it difficult to plan the project, estimate costs, schedule resources, monitor and report progress, and review quality. A process must be found which can meet the unique challenges of developing quality software for the web in a timely manner. In this paper, we define a lightweight development process which does so and we describe our experience using it. We also generalize this experience for similar development projects.

2 Challenges of Developing a Web-based Application Process

One serious challenge concerns the difficulty of pinning down the requirements early. This is caused by several factors. At the start, the users may be unprepared or unwilling to abandon old methods in favor of new web-based processes. They may lack a vision of just how much the web-based application (WBA) can do for them. Consequently, as the user discovers what can be done, he or she adds requirements piecemeal throughout the entire software life cycle. This can lead to dramatic and frequent changes which ripple through every work product. *A process for WBA development must adapt to changing requirements.*

A related challenge is that the business function itself evolves as a consequence of migrating it to the web. As the user sees what can be done, he or she also begins to rethink the underlying business model. This also changes the developing WBA's functional requirements throughout the entire life cycle. One industry example is when revenue once generated from product sales is replaced by revenue from advertisements on a web page offering the products for free. *A process for WBA development must allow the user and the underlying business function to evolve.*

Another challenge is that due to the time and cost constraints of typical WBA development, formal methods are viewed as obstructive and are discarded in favor of a haphazard and rushed approach, with all the problems that implies. *Early stages of the process need to be unobtrusive; documentation and management should be kept to the minimum necessary to preserve accountability.*

Another challenge is that, contrary to popular opinion, user interface development on the web is difficult. The users of any web application are potentially very diverse in experience, age, language, training, and more. The user interface of the web application will likely undergo many revisions throughout the development process. *The process should produce a complete user interface concept, if not a working product, as early as possible.*

A final challenge is that the clients providing the requirements, the WBA developers and the WBA users may be geographically scattered, making frequent meetings impractical. *The process should succeed despite the lack of face-to-face interaction between any of the primary players in the project.*

These challenges are not comprehensive, nor are they unique to web development. However, we have often encountered them, and in our experience they seem to occur frequently and together in web-based applications.

3 Process Description

We propose a lightweight development process that addresses each of the challenges described above. This process is based on our recent and successful experience with implementing a grant application system for the Federal Highway Administration (FHWA), National Scenic Byways (NSB) Program. We provide an overview of the process below and describe our experience in Section 4.

Our process is similar to the evolutionary prototyping model which consists of cascading phases of interdependent waterfall models. Our process, however, differs from the evolutionary prototyping process model in several ways. For example, it involves just two phases instead of an undefined number. The first follows a modified incremental development model with an emphasis on rapid prototyping. The second follows a lean waterfall model with an emphasis on quality. See Figure 1. Our process also deals with poorly understood requirements early in the development life cycle; whereas, the evolutionary prototyping process is more likely to start off with well understood requirements [2].

Phase 1 is based on an incremental process model and therefore involves the iteration of core development activities. However, it focuses on just five: requirements definition, systems analysis, design, coding and testing. Here, the requirements definition consists of one-on-one conversations with the users (both clients and customers) and produces an outline of desired features. The systems analysis focuses on understanding the key entities and their relationships, interactions and behavior. It refines a model describing the existing business functions and their reincarnation on the web. Although any reasonable conceptual model language can be used, we prefer Object-oriented Systems Analysis [3] and Unified Modeling Language [1] for their expressiveness and solid foundations. The design includes high-level organization, user interface, and database issues, and it refines module specifications, screen sketches, and database schema. The coding involves programming, graphic artwork, and writing, and it produces incremental versions of the system with all the necessary icons, backgrounds, decorations, and on-line documentation. In some cases, particularly during early iterations, the coding is for prototypes that are intended to be throwaway and

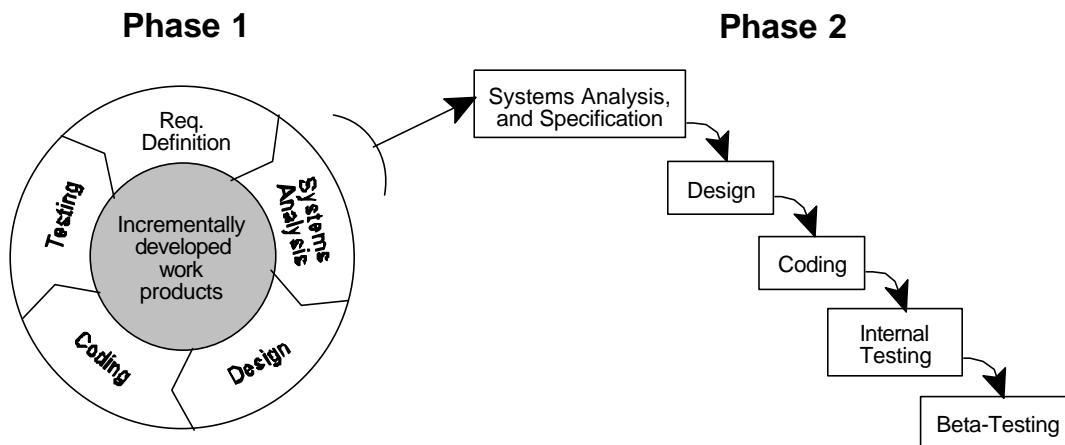


Figure 1 Overview of development process

redone in later iterations. The testing focuses on both the validation and verification of the software system and is done at unit, integration, and system levels. During early iterations the programmers test their own work. In later iterations, team members test each other's work, and in the final iterations, the selected users participate. In all cases, the testing activities produce feedback for the next iteration.

The activities and work products of the first phase include only those which are essential to capture the core ideas, communicate with the customer, manage the project, and produce a usable product. Several common activities, such as system specification and detail design, are explicitly mentioned in the process model. Although these activities can still be performed when appropriate, they are de-emphasized and folded into other activities. For example, some specification for defining high-level modules occurs during design and some detailed design occurs as part of coding.

The first phase differs from the traditional incremental model in several ways. First, it is as lightweight and informal as possible and yet still produces a usable product. Second, the requirements during the first phase are more fluid than would otherwise be desired or allowed. Third, it integrates the notion of throwaway prototypes for high-risk features which are either poorly understood, have a large impact on the rest of the system, or are of particular importance to the client.

Phase 2 is based on a waterfall model and therefore consists of well-defined steps that correspond closely to development activities. Work products from one step become inputs for the next. Unlike the first phase, progress is measured in terms of completion of development activities, instead of the completion of features.

The first step of Phase 2 is system analysis and specification. This step will use the entire set of work products and experiences from the first phase as a basis for re-analyzing the system and coming up with a concise and accurate specification. The second step is design which produces module definitions, screen sketches, and database schema. The coding step implements the entire system. The testing step includes unit, integration, and alpha testing. The last step is beta testing which involves testing with a selected set of users. Like other waterfall processes, Phase 2 allows for feedback loops to correct problems and respond to issues raised during testing.

Like Phase 1, Phase 2 is kept as lightweight as possible. Some steps are intentionally excluded or folded into other steps. For example, requirements definition doesn't exist since the entire first phase serves that purpose. Also, system analysis and specification are combined and produce a single work product.

It is important to note that both phases are critical to the overall success of the development process. The first is necessary because the process must be adaptable as all involved parties react to change. Also, if development stopped after the first phase or continued on in an incremental fashion, the product might never reach its full potential. One reason is because it was designed before the client made the all important shift to a web-based system.

4 Our Experience with the Process

The FHWA-NSB required a cost- and time-effective means of gathering and reviewing submitted discretionary grant applications. At that time, the submissions were paper-based and entered by hand into a database which then generated the necessary reports. This method took many man-months. We used the two-phase process above to develop a web-based application to automate the entire discretionary grant application and award process.

Initially, our client, the FHWA, was reluctant to abandon the old method and switch to the new one. Also, many of our experienced users (associated with the state byway programs) resisted the change. In addition, the FHWA was revising the older method even as we worked.

Our client was in Washington, D.C., we were in Utah, and our ultimate users were scattered throughout the U.S. Funding allowed for just one face-to-face meeting, so phone calls, faxes, and the Internet were the primary means of communication.

Time constraints were severe, especially for 1998. The FHWA and the state byway agencies required delivery of a working product just seven weeks after assigning us the project. This was so that they could meet deadlines imposed by the legislature.

Phase 1

Our objectives for Phase 1 were:

1. produce a working product in time for that year's discretionary grant deadline;
2. enable byway agencies to submit correct and complete applications
3. enable state and federal agencies to review them
4. help the client and customers move from a paper-based system to a web-based system.

Phase 1 began the first week in June 1998. Systems analysis, database design, and user interface design took 2-3 weeks. Implementation began in earnest June 26th with the first client review occurring July 6th. The actual release was July 24th, one week before the legislative deadline. For unrelated reasons, the deadline was moved to Aug. 7th, and during those three weeks, 38 different users submitted 83 grant applications online.

We completed Phase 1 within the allotted seven weeks. During that time, we iterated through the development activity shown in Figure 1 several dozen times. Each iteration added a feature or two to the system. Early iterations focused on analysis and database design, but also included the design, implementation, and testing of exploratory prototypes. Later iterations focused on user interface design and information flow. Final iterations focused on polishing the look-and-feel, terminology, and online help. The requirements were frozen for the final iterations so as to converge on a public release. This release was in time for that year's grant submission deadline, and contained sufficient features to enable the relevant agencies to submit and review grant applications.

We tested the WBA with the team and client. After the release, our users discovered several minor errors and one major error (a data field overflow with subsequent loss of data.) Most errors, including the major one, were fixed within 24 hours if not while the customer was online and talking to us. The correction of a few cosmetic problems and navigational inconveniences were intentionally postponed to the second phase.

During the first phase, frequent review sessions with the client were essential. These review sessions had several benefits. They provided valuable feedback on the emerging software system. They kept us focused and on track. Finally, they helped the client envision how the web-based system could not only augment but supplant the paper-based system. This set the stage for the second phase.

Phase 2

Our objectives for Phase 2 were:

1. produce a working product in time for this year's discretionary grant deadline;
2. enable byway agencies to submit correct and complete applications
3. enable state and federal agencies to review them
4. help the client and customers complete the move from a paper-based system to a web-based system.

Phase 2 began Nov. 18 and 19th, 1998 with a two-day face-to-face meeting with our client. This meeting was both a post-mortem on Phase 1 and a heads-down design session for Phase 2. Database redesign took two days, user interface design just one day and the design was completed by Nov. 31st. A beta release was completed Dec. 18th, which met and exceeded the capabilities of Phase 1's release. Beta testing was Dec. 28-31st, and the product was ready to release Jan. 4th this year. It lacked only help and sample grant application information which was to be supplied by the FHWA. The final product could not be officially released to the public until the FHWA announced this year's discretionary grants fund, which finally occurred Feb. 23rd, 1999. While no applications have been submitted at this writing (Feb. 25th), 12 customers have created online accounts and are currently exploring the system. They are in no hurry; this year, the legislative deadline is June 30th.

Our five beta testers submitted 46 minor suggestions, six major suggestions and found 16 minor and moderate errors. The major suggestions required changing part of the database scheme, changing user interface behavior or adding functionality. Since the release, three minor errors (a missing image, a missing explanation, and an out-of-date explanation) have been found and fixed.

Although its software development was as time-constrained as it was in the first phase (seven weeks, counting holidays), this phase went much more smoothly. Schedules were kept, milestones met or anticipated, and the preliminary reaction from client and users has been quite positive.¹

The amount of time spent in Phase 2 on requirements, analysis, specification and design was drastically reduced. In effect, Phase 1 embodied the requirements definition and systems analysis. The experience we gained during the hectic weeks of Phase 1 paid off in Phase 2. We gained a better understanding of the discretionary grant business function and developed a more comprehensive WBA in equivalent time. Our client switched from resisting a completely web-based application to enthusiastically supporting it. Even our customers, having been exposed to the Phase 1 product, were better prepared for the Phase 2 product.

6 Summary

The two-phase process described in this paper successfully addresses the challenges of web-based application development. The first phase plays a key role in helping the development team understand the requirements and adapt to frequent changes. It also helps the client and customers grasp the potential of this new medium and assist with unleashing its full potential. The second phase builds on the knowledge gained from the first phase and produces a product that has a better user interface, more extensible, and maintainable. Both phases are as lightweight as possible without compromising a managers ability to plan a project and evaluate its progress.

A key aspect of this process is that it works well even when the involved parties are geographically separate. Such situations have become commonplace with web-based application development.

The experience presented in this paper provides evidence that this two-phase process can work. We plan to apply the same process to additional projects in the coming year. We expect that these projects will share the same challenges as the NSB grant application project.

¹Readers interested in this site can find it at <http://www.byways.org/grants2>. We recommend that readers who wish to test the site do so on our development site, <http://ubn.cs.usu.edu:8080/grants2>. The sites are functionally identical, but are on different machines and use different databases. The scenic byways community will appreciate the courtesy.

References

- [1] G. Booch, J. Rumbaugh and I. Jacobson, *The Unified Modeling Language User Guide*. Reading, MA: Addison-Wesley Publishing Company, 1999.
- [2] A. M. Davis, E. H. Bersoff, and E. R. Comer, "A Strategy for Comparing Alternative Software Development Life Cycle Models," *IEEE Transactions on Software Engineering*, vol. 14, no. 10, pp. 1453-1461, Oct. 1988.
- [3] D. W. Embley, B. D. Kurtz, S. N. Woodfield, *Object-Oriented Systems Analysis: A Model-Driven Approach*. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1992.
- [4] Federal Highway Administration and Utah State University (1999), *National Scenic Byways Online, Grants Online*. [World Wide Web application]. URL <http://www.byways.org/grants2>