Best Practices for Software Projects - Prototyping

February 2004 - Pragmatic Software Newsletters

The most critical part of software development and project management is to ensure that you first agree with your client on the requirements and the scope of the project. If this is not done well, your project has very little chance of succeeding. Since everything in the software lifecycle is driven from requirements, all follow-on tasks to the requirements gathering will be flawed if the requirements are not correct.

A technique for improving the requirements gathering process is prototyping. With prototyping, your team works with the client to collect their requirements, normally done during JAD (Joint Application Development) sessions. As requirements are collected, your technical team should develop prototypes of the screens, reports, processes and the associated workflow. Once developed, the JAD participants can tangibly see that you have (or have not) captured their requirements accurately. Discussions by JAD participants will refine the prototypes until you have reached agreement on the exact requirements.

Below are the advantages to prototyping:

- provides early feedback on the requirements when the cost of making changes is low
- provides buy-in from client and JAD members to ensure that everyone agrees to the requirements
- provides visible progress, improving team moral and sense of accomplishment
- reduces project timeline because requirements are solidified up front, allowing programmers to architect a better solution that enhances code reuse
- lowers defect rates because of clear requirement definition
- quickens the ability for new team members to get up-to-speed on the project, as everything is well defined

The best way to demonstrate prototyping is from a real-life example. Recently, we implemented a time sheet system for a client that was delivered on-time and on-budget. The names of the client and sensitive information have been removed. Here is how we did it:

1. **JAD Sessions** - Initially, we held JAD sessions with the client. Because this was a fairly small project, the JAD participants were the client's project manager, and 2 of the end-users. From our team, we had the project manager and the software architect attend the meeting. The project manager was responsible for setting up and running the meetings as well as putting all the documentation together. The software architect was responsible for putting together the prototypes and aiding the project manager in putting together the documentation. The JAD session consisted of 3 meetings that spanned over one week.

2. **JAD Session 1** - Prior to meeting, the client sent us some rough requirements that consisted of time sheet layouts for salaried vs. hourly employees and a short description of how each person would be using the time sheet system. Our project manager called the client project manager and discussed the requirements in detail and talked about workflow. Then the project manager and software architect worked together to create a first draft of the requirements without any prototypes. Here is a copy of the initial requirements document we put together and discussed in **JAD Session 1**:

   http://www.pragmaticsw.com/newsletters/JADVIL0039a.pdf

   Notice that this document does not have any prototypes, it just explains the time sheet system in general terms. In **JAD Session 1**, we reviewed the requirements with the client to ensure we had captured their requirements and workflow accurately. At the conclusion of the meeting, another follow-on meeting was scheduled for 2 days later and we promised a prototype.

3. **JAD Session 2** - Prior to the meeting, our project manager worked with the software architect to develop a prototype based on the feedback from **JAD Session 1**. The software architect used Microsoft FrontPage to create the prototype (but could just have easily used Adobe Photoshop or Dreamweaver). Here is a copy of the requirements document we put together and discussed in JAD Session 2:

http://www.pragmaticsw.com/newsletters/JADVIL0039b.pdf

Notice that this is the same document used in JAD Session 1, but contains screen shots and other information about the prototype. It discusses the behaviors of each function, etc. In JAD Session 2, we reviewed the prototype with the client to ensure we had captured their requirements and workflow accurately. At the conclusion of the meeting, a final follow-on meeting was scheduled for 2 days later and we promised an estimate and a workflow prototype.

4. **JAD Session 3** - Prior to the meeting, our project manager worked with the software architect to develop a workflow prototype based on the feedback from **JAD Session 2**. The software architect used Macromedia's RoboDemo to create the workflow prototype and also put together a detailed deisgn so that he could estimate the total effort. Here is a copy of the requirements document we put together and discussed in **JAD Session 3**:

   http://www.pragmaticsw.com/newsletters/JADVIL0039c.pdf

   Notice that this is the same document used in **JAD Session 2**, but contain links to a workflow prototype, shows screen shots of the prototype and also contains a Detailed Design. This allowed us to provide the client with a fixed price estimate for doing the project. At the conclusion of **JAD Session 3**, the client gave us final approval to move forward.

As you can see, creating prototypes can dramatically increase your chances of delivering on-time and on-budget. As development began, we used Software Planner (http://www.softwareplanner.com) to track project deliverables and defects found during the testing process.

Below are some helpful templates to aid you in developing software solutions on-time and on-budget:

- **Project Management Guidelines -** http://www.PragmaticSW.com/Pragmatic/Templates/ProjectMgtGuidelines.rtf
- **Functional Specifications -** http://www.PragmaticSW.com/Pragmatic/Templates/FunctionalSpec.rtf
- **Architectural Overview -** http://www.PragmaticSW.com/Pragmatic/Templates/ArchitectureOverview.rtf
- **Detailed Design** - http://www.PragmaticSW.com/Pragmatic/Templates/DetailedDesign.rtf
- **Strategic Planning Document** - http://www.PragmaticSW.com/Pragmatic/Templates/StrategicPlanning.rtf
- **Test Design** - http://www.PragmaticSW.com/Pragmatic/Templates/TestDesign.rtf
- **Risk Assessment** - http://www.PragmaticSW.com/Pragmatic/Templates/Risk%20Assessment.rtf
- **Weekly Status** - http://www.PragmaticSW.com/Pragmatic/Templates/WeeklyStatusRpt.rtf
- **User Acceptance Test Release Report** - http://www.PragmaticSW.com/Pragmatic/Templates/UATRelease.rtf
- **Post Mortem Report** - http://www.PragmaticSW.com/Pragmatic/Templates/PostMortem.rtf
- **All Templates** - http://www.PragmaticSW.com/Templates.htm
- **Prior Newsletters** - http://www.PragmaticSW.com/Newsletters.htm
- **Software Planner** - http://www.SoftwarePlanner.com

About the Author

Steve Miller is the President of **Pragmatic Software** (http://www.pragmaticsw.com). With over 20 years of experience, Steve has extensive knowledge in project management, software architecture and test design. Steve publishes a monthly newsletter for companies that design and develop software. You can read other newsletters at http://www.PragmaticSW.com/Newsletters.htm. Steve's email is steve.miller@pragmaticsw.com.

Pragmatic Software Co., Inc.
9085 E. Mineral Circle
Suite 340
Englewood, CO 80112

Phone: 303.471.8355
Fax: 303.346.1749
Web site: http://www.pragmaticsw.com
E-mail: info@pragmaticsw.com