# Customer Best Practices - Production Phase

October 2005 - Pragmatic Software Newsletters

## Customer Best Practices - Production Phase

According to the Standish Group, the primary contributor for project success (projects delivered on-time and on-budget) is great customer involvement. Knowing that, it makes sense to define best practices for dealing with customers. In the coming months, we will look at best practices for customer involvement that deal with all phases of the software life cycle:

1. **Planning** - Getting the customers involved in the planning process is critical. Gathering, analyzing and evaluating requirements is key to ensuring that your project meets the needs of your customer. *More Information...*

2. **Design** - Good designs are key to ensuring that changes the client requests require minimal changes to your framework. *More Information...*

3. **Iterative Coding and Testing** - Customers play a major role in Quality Assurance and User Acceptance testing prior to production releases. *More Information...*

4. **Production** - Having good release management and support ticket management procedures in place will ensure quality releases and great customer support as issues arise.

5. **Post Mortem** - Upon completion of projects, it is of paramount importance to involve your customers in the project review process. Post Mortem reviews can aid in discovering and document common issues that arose during the project and allows for others to better plan for those risks in upcoming projects.

## Coding and Testing Phase

Once your software is in production, customer interaction is just as important as it was during the development phase. Below are some best practices to aid project management in the Production phase:

- **Source Control** - If your source code was not being managed during development, it is imperative that it be managed during production. A good source control system provides a central repository for source code, provides a historical record of changes made to source over time, and provides a way for multiple developers to work on the same code at the same time, merging their code together at a later time. There are many products that provide this: Microsoft Visual Source Safe Perforce Vault

- **Quality Assurance** - Once a product is in production, it is imperative to set up a Quality Assurance (QA) server for testing changes prior to moving them to production. The QA server should closely match the production environment, as to make it easy for migration. All items tested in QA will not affect your production server, thereby allowing you to find any defects before they make it to production.

- **Change Orders** - Once a product is in production, new enhancements and change orders should be documented, analyzed and planned for. Approved change orders will go back into the full development lifecycle, beginning at requirements gathering and ending in production.

- **Support Ticket Management** - Once production begins, it is critical to have a good support

ticket system. This system should allow your user based to report issues and check the status of them online, at any time. Software Planner (http://www.softwareplanner.com) offers a support manager add-on that allows your users to post support tickets. Once posted, it emails a copy of the ticket to the support team and a copy goes to the user that reported the issue. The user can check the status of the ticket at anytime and the support team can work on the ticket externally (from the support ticket email interface) or via Software Planner. *More Information...*

## Helpful Templates

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
- **Defect Tracker** - http://www.DefectTracker.com
- **Remoteus (Remote Desktop Sharing)** - http://www.PragmaticSW.com/Remoteus.asp

### 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.
1745 Shea Center Drive
Suite 400
Highlands Ranch, CO 80129

Phone: 720.344.4846
Fax: 720.344.4847
Web site: http://www.PragmaticSW.com
E-mail: info@PragmaticSW.com