

Best Practices for Software Projects - Software Measurements

October 2004 - Pragmatic Software Newsletters

Most software projects fail to deliver on-time and on-budget. To reduce the risk of failure, project managers should implement measurements, allowing them to more accurately estimate projects and to enhance the quality of releases.

The key to efficient measurement is to first determine what goals you are trying to accomplish and what problems you are attacking. Many organizations waste time and money by measuring more things than are necessary. Before beginning a measurement strategy, determine the goals for your measurement. Here are some common reasons for not delivering on-time and on-budget:

- Software coding efforts are not properly estimated
- Testing efforts are not properly estimated
- Software quality is poor, therefore the testing duration is longer than need be
- Customer changes impact the software project, thereby extending the project dates

Attacking the Common Problems

Software Coding Efforts are Not Properly Estimated

This problem normally arises due to these issues:

- **Customer Requirements** - To properly estimate coding effort, you must create solid customer requirements. The requirements must contain adequate detail to allow the programmers to create detailed designs. From a measurement perspective, you should track the amount of time it takes to develop each customer requirement. Track both estimated and actual hours so that you can use this information to improve future estimates.
- **Detailed Designs** - It is impossible to estimate coding effort without creating a detailed design. The detailed design allows the developer to think through all the tasks that must be done to deliver each requirement. From a measurement perspective, you should track the amount of time it takes to develop each detailed design. Track both estimated and actual hours so that you can use this information to improve future estimates.

Testing Efforts are Not Properly Estimated

This problem normally arises due to these issues:

- **Test Plans** - Once the customer requirement and detailed design is created and estimated, the test leader should create a detailed test plan that estimates the testing effort. This is done by thinking through the test cases that will be created based on the requirement and design. From a measurement perspective, you should track the amount of time it takes to develop each test plan. Track both estimated and actual hours so that you can use this information to improve future estimates.

Software Quality is Poor

This problem normally arises due to these issues:

- **No Code Reviews** - If regular code reviews are not done, there is a much higher chance of delivering software with poor quality. For large projects, these problems are compounded over time, so it is best to do code reviews early and often (at least weekly). From a measurement perspective, you should track the amount of rework time required due to failed code reviews. This can aid you in planning for rework on future projects.
- **Failed Smoke Tests** - By running weekly smoke tests, you can shorten the testing phase as issues are caught early in the coding and testing cycle. From a measurement perspective, track

the number of test cases passed and failed during smoke tests, week by week. The goal is to reduce the number of failed test cases as the project progresses.

- **Defect Tracking** - As testing commences, track the number of open defects vs. total defects to help predict project release dates. Track the number of defects found during code reviews vs. test case execution. This will help track and improve estimation accuracy. Track the percentage of total defects before product release, as this will help assess product quality.

Customer Changes Impact the Software Project

This problem normally arises due to these issues:

- **Missing Change Control Processes** - As a project progresses, clients sometimes ask for features to be changed or for features to be added or removed. Before making any changes to the project, each request should be thoroughly investigated and a risk assessment should be done for each request. If changes are necessary and agreed upon by the client, project timelines are adjusted. From a measurement perspective, track the number of change requests, how many were approved vs. rejected, and the effort for estimating reviewing and assessing each change request. This information can be used in future projects to predict the number number of change requests that are approved and estimated as to build time into your projects to mitigate that risk.

Using Online Tools

The project software life cycle can be greatly improved by managing all phases of the software life cycle online. A web-based tool can sometimes be easiest to use, as it requires no client configuration. There are many solutions to choose from, consider [Defect Tracker](#) or [Software Planner](#), depending on your needs. What ever tool you select should have the ability to collect metrics (like number of defects over time, etc.) without much extra effort on your part. Both of the products mentioned have this capability:

<http://www.pragmaticsw.com/GuidedTours/Default.asp?FileName=Reports>

The tool you select should also allow you to create your own custom lists of information. This is especially important as to allow you to create your own screens to track measurements for each area of the system you wish to track measurements for. [Defect Tracker](#) and [Software Planner](#) both have a List Manager feature that allows you to create custom screens to track anything you like. You can use that to track your measurements:

<http://www.pragmaticsw.com/GuidedTours/Default.asp?FileName=ListManager>

Conclusion - Helpful Templates

As you can see, implementing **measurements** can greatly improve your software quality and can prepare your team for delivering on-time and on-budget. 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>
- **EggPlant (Automated testing by Redstone Software)** - <http://www.redstonesoftware.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

This document was created with Win2PDF available at <http://www.daneprairie.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.