Software Development Lifecycle (SDLC) - Defect and Test Case Measurement

January 2006 - Pragmatic Software Newsletters

Measuring the Software Development Lifecycle

Employing a solid software development lifecycle (SDLC) methodology can drastically increase your ability to deliver software projects on-time and on-budget. Once a solid SDLC methodology is in place, how do you know how efficient it is and how well it is performing? In the coming months, we will look at best practices for measuring the key indicators of the SDLC, and equip you with the tools to improve your SDLC. Below are the topics to be covered in the coming months:

1. **Defect and Test Case Measurement** - Defect and Test Case Measurement is a pre-production activity that allows teams to determine the quality of their software development, and indicates when the software is ready to be released to production.

2. **Project Task Measurement** - Project Task measurement allows your team to determine how well individual tasks were estimated, how well they were defined, and whether items are completed on-time and on-budget.

3. **Overall Project Measurement** - It is important to measure overall project success by determining if the project was estimated properly, risks were identified and mitigated, requirements were correctly identified and documented, and if the project was delivered on-time and on-budget. From this, we learn to provide better estimates, collect better requirements, and do better risk management.

4. **Support Ticket Measurement** - Support Ticket management is a post-production activity that allows teams to determine the quality of the software release, the quality of User Guides and other documentation, and provides insight as to how well the software was architected and implemented.

5. **Measuring Team Goals** - For technical teams to flourish, team goals must be established and measured. Constant evaluation of the goals, and progress towards them, is critical to ensuring that team goals contribute to departmental goals.

6. **Measuring Departmental Goals** - Establishing and measuring departmental goals allow your company to grow, allow your department to identify it's contribution to company growth and fosters and environment where team members thrive.

Defect and Test Case Measurement

Defect and Test Case Measurement is a pre-production activity that allows teams to determine the quality of their software development, and indicates when the software is ready to be released to production. Below are some best practices for measuring software quality.

1. **Test Case to Requirements Traceability** - Prior to developing your test cases, you must ensure that you have an adequate number of test cases for each requirement (functional specifications). This ensures that you have adequate testing for each customer requirement. How do you do this?

   Make a list of every test case you have defined, and link each test case back to the requirement. Then meet with your project team (project manager, developers and testers) to review the test cases. This meeting will spawn new test cases you may not have thought of and will allow everyone visibility into your test process.

   Basic Approach - If you do not have a software development lifecycle tool, a low-cost and simple approach to this is to create a spreadsheet that contains a list of your functional specifications (requirements) and lists each test case that has been created for each requirement. **Example:** TestCase_Traceability.xls

   Advanced Approach - A better approach is to utilize an SDLC tool that allows tracking of functional

specifications (customer requirements), and allows creating of test cases that are linked to the functional specification. The software should provide a traceability report that shows how the test cases are linked to each requirement. Below are some example reports that come bundled with Software Planner, but any good SDLC tool should provide these types of reports.
Example: Software Planner_TestCase_Traceability Report.pdf Tracking Requirements Tracking Test Cases

2. **Test Case Trending** - As the testing cycle begins, each test case should be executed and marked as **Passed** or **Failed**. As your testing cycle progresses, you should have a clear indication as to whether your software quality is increasing or decreasing. How do you do this?

   *You can determine this by the number of test cases that are moving from **failed** to **passed** status over time. Prior to testing commencing, you should have all your test cases defined and **awaiting run**. Once testing begins, your team will execute each test case and mark them **passed** or **failed**. In the early stages of the testing cycle, you may have a large number of **failed** test cases, defects will be generated and resolved by the development team. As time progresses, you should re-test all the **failed** test cases until they all have **passed** (or an acceptable number have **passed** based on your production migration procedures).*

   Basic Approach - If you do not have an SDLC tool, a low-cost and simple approach to this is to create a spreadsheet that contains a daily journal of the number of test cases each day, and how many have passed and failed, with this information graphed so that it is easy to spot trends. **Example:** TestCase_Trending.xls

   Advanced Approach - A better approach is to utilize an SDLC tool that automatically creates trending reports. If the software can send you a daily or weekly email with metrics, all the better. Below are some example reports that come bundled with Software Planner, but any good SDLC tool should provide these types of reports.
   Example: Software Planner_TestCase_Trending Report.pdf Software Planner Daily Summary Report (via Email)

3. **Defect Trending** - As the testing cycle proceeds, defects will be found. It is important to track the number of defects over time, so that you can determine how buggy a specific release is, and if the release is improving over time. How do you do this?

   You can determine this by measuring the number of defects and the number that are being resolved and closed over time. As your testing cycle proceeds, the number of outstanding defects (active, open, etc.) should decrease, and the number of resolved and closed defects should increase. It is wise to have a pre-set policy that dictates when a release is production-ready. For example, you may have a policy that indicates that once you have no severity 1 or 2 defects, you can move the software to production. Here are some common severities you might use:
   Severity 1 - Crashes the system
   Severity 2 - Major defect with no workaround
   Severity 3 - Major defect but with workaround
   Severity 4 - Trivial defect

   Basic Approach - If you do not have an SDLC tool, a low-cost and simple approach to this is to create a spreadsheet that contains a daily journal of the number of defects each day, by status, with this information graphed so that it is easy to spot trends. **Example:** Defect_Trending.xls Defects_Advanced_Trending.xls

   Advanced Approach - A better approach is to utilize an SDLC tool that automatically creates trending reports. If the software can send you a daily or weekly email with metrics, all the better. Below are some example reports that come bundled with Software Planner, but any good SDLC tool should provide these types of reports. For easier data entry, it would be great if your SDLC tool allows you to automatically generate defects test cases are marked **Failed**. Software Planner does have this feature.
   Example: Software Planner_Defect_Trending Report.pdf Software Planner Daily Summary Report (via Email)

4. **Other Reporting** - In addition to basic metrics, here are some other reports that can provide valuable information as you manage the software development lifecycle. Below are some example reports that come bundled with Software Planner, but any good SDLC tool should provide these types of reports.

   Defect Reports:
   A) Defects by Assignee
   B) Defects by Project
   C) Defects by Status and Priority
   D) Defect Detail
   E) Daily Summary Report

   Test Case Reports:
   A) Test Cases by Assignee
   B) Test Cases by Project
   C) Test Cases by Status
   D) Test Cases Detail

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.
383 Inverness Parkway
Suite 280
Englewood, CO 80112

Phone: 303.768.7480
Fax: 303.768.7481
Web site: http://www.PragmaticSW.com
E-mail: info@PragmaticSW.com