Software Development Lifecycle (SDLC) - Project Task Measurement

February 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. More...

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.

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. Below are some best practices for measuring individual tasks.

1. **Develop Your Project Plan** - Once a project begins, you must develop a list of tasks, estimate the tasks (in work hours), assign each task to a team member, and level the resources (so that each person is not overloaded with work). How do you do this?

   First, you must detail each of the requirements and develop a list of tasks for the delivery of each requirement (if you need help developing requirements, click here). With a good set of *requirements, the project manager can work with the project team (programmers, testers, etc) to develop a list of tasks that must be completed for each requirement, along with the estimated effort of each task. Once this is done, record your list of tasks, assign them to team members, and track their progress. As tasks are completed, record the number of actual hours it took to complete each task, as to allow you to determine the correctness your estimation process. You can use this information on future projects, to create a buffer for improving the estimation 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 tasks and assignment information. As your team members work on items, each day you should make note of actual hours and costs thus far, and percentage complete. As items are finished, update the spreadsheet with Actual Hours, Actual Costs, and Completion Date.
Example: ProjectTasks.xls

Advanced Approach - A better approach is to utilize an SDLC tool that allows tracking of project tasks, assignment of the tasks to team members, tracking of hours and costs, and allows the team members to update their percentage complete. For ease of update, the software should be web based, so that it can be accessed from any location. To do this, you can use Software Planner, Microsoft Project or some other project management tool. The disadvantage of using windows-based tools (like Microsoft Project) is that they are not web based, so individual team members can not easily update their hours, costs and percentage complete; the project manager is forced to update that information. Software Planner (and other web based tools), empower the people doing the work to update this information, and has email alerts that alert the Project Manager as items are updated.
Example: Software Planner Task Management Importing into Software Planner from Microsoft Project

2. **Identify Slipping Items** - The key to delivering on-time and on-budget is keeping a keen eye on each project task. If one task is not completed on time, it could affect the entire project plan, forcing the plan to be over budget forcing a late delivery. How do you do prevent this?

   You must review your project tasks every day, track the number of hours worked on each task, determine the number of hours remaining to complete each item (or percentage complete), and you must take action if specific tasks are slipping. This may mean adding resources, having team members work additional hours, or allowing other team members to pitch in to aid the slipping task.

   Basic Approach - If you do not have an SDLC tool, update the project plan spreadsheet each day. Speak with each team member daily to obtain the number of hours they have worked on each task and to obtain their percentage complete. Have them assure you that they will be done on the stated due date, or alert you that their tasks are slipping.

   Advanced Approach - A better approach is to utilize an SDLC tool that automatically alerts you of items that are slipping, and allows team members to update their hours and percentage complete as they work on items. To be proactive, you should be able to receive an automatic email nightly that identifies all the items that are coming due and the ones that are current overdue (and who they are assigned to). Also, when managing your project tasks, it is great if the overdue tasks turn Red, so that you can easily spot the ones that are overdue. Below are some example reports that come bundled with Software Planner, but any good SDLC tool should provide these types of reports.

   Examples:
   A) Overdue Items Report (via Email nightly)
   B) See overdue items in RED
   C) Project Tasks by Assignee Report
   D) Project Task Trending Report

E) Project Task Durations Report (Actual Hours logged)

3. **Post Mortem** - As tasks are completed, it is important to determine what tasks were completed on time and what tasks were not. By recording this information, you can begin improving your estimating process. This also allows you to identify team members that continuously over promise and under deliver, so that you can take corrective action. How do you do this?

   The key to this is to have each person record their actual hours worked on each task daily (while it is still fresh in their mind). Also, as tasks are completed, record the date each task was completed, so that you can compare that to the date it was due. By using this information, you can roll this information up for all team members on the project to determine how far off your estimates were. You may find that overall, your estimates were 10% overstated, or 20% understated. This information can be used on subsequent projects to build in buffer that would allow you to come closer to completing projects on-time and on-budget, and to identify how individual team members performed.

   Basic Approach - If you do not have an SDLC tool, update the project plan spreadsheet each day. Speak with each team member daily to obtain the number of hours they have worked on each task and to obtain their percentage complete.
   Example: ProjectTasks_Final.xls

   Advanced Approach - A better approach is to utilize an SDLC tool that empower each individual to enter their hours, costs, and percentage complete each day. It is also helpful to have a report that automatically calculates variances to target. Below are variance reports that comes bundled with Software Planner, but any good SDLC tool should provide this type of report.

   Examples:
   A) Project Tasks Variances by Project - Shows variances by project
   B) Project Tasks Variances by Assignee - Shows variances by team member
   C) Project Tasks Variances by Owner - Shows variances by project manager (owner)

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