

Software Development Lifecycle (SDLC) - Measuring Team Goals

May 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? This newsletter is the last in a series of newsletters, below are the topics that have been covered:

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.
http://www.PragmaticSW.com/newsletters/Newsletter_2006_01_SP.htm
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.
http://www.PragmaticSW.com/newsletters/Newsletter_2006_02_SP.htm
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.
http://www.PragmaticSW.com/newsletters/Newsletter_2006_03_SP.htm
4. **Support Management** - 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.
http://www.PragmaticSW.com/newsletters/Newsletter_2006_04_SP.htm
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.
http://www.PragmaticSW.com/newsletters/Newsletter_2006_05_SP.htm

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. Below are some best practices for goal setting and tracking:

1. **Define Departmental Goals** - Most software departments want to deliver their software projects on-time and on-budget, increase software quality, improve product features as to allow the product to be more marketable, and to provide a great experience for the clients that utilize their software. Knowing that you must define specific, measurable goals to meet these objectives. Below are some examples, notice that each goal is measurable and has a timeframe associated with it.

1. *To deliver 80% of our software projects on-time and on-budget by year ending 2006.*

2. *To reduce product support tickets to 100 per month and maintain no more than 50 open defects by year ending 2006.*

3. *To increase product revenue by 30% by year ending 2006 by improving our product features for marketability.*
4. *To ensure a 90% customer satisfaction rating from our clients by year ending 2006.*

2. **Goal # 1 (Deliver on-time and on-budget)** - In our example, the departmental goal of completing 80% the software projects on-time and on-budget will require that the project manager collect this information.

To do this, the project manager will keep track of each project that is started, and record the estimated and actual hours as the project progresses. No project will deliver on exactly the same number of hours that you estimate, this is not usually reasonable. So you must decide what criteria you will use to judge a project as being successful or unsuccessful. Most teams assume that if you deliver the project within 5% of the estimates, then the project is successful. The project manager may create a spreadsheet that tracks this, similar to this:

Identify Project Success Rate

Project	Estimate	Actual	Difference	Variance %	Successful?
Widgets Release 4.1	248	312	-64	-20%	No
Widgets Release 4.2	360	390	-30	-7.6%	No
Widgets Release 4.3	422	445	-33	-5%	Yes
Widgets Release 4.4	500	490	10 Under Budget		Yes
Widgets Release 4.5	400	420	-20	-4%	Yes
Widgets Release 4.6	330	300	30 Under Budget		Yes
Widgets Release 4.7	400	300	100 Under Budget		Yes
Widgets Release 4.8	500	425	75 Under Budget		Yes
Widgets Release 4.9	500	490	10 Under Budget		Yes
Widgets Release 5.0	1000	980	20 Under Budget		Yes

Summary: 80% of projects were delivered on-time and on-budget

Note: Instead of using a spreadsheet, [Software Planner](#) and similar tools allow the Project Manager to review project estimates vs. actuals, here are a couple of example reports:

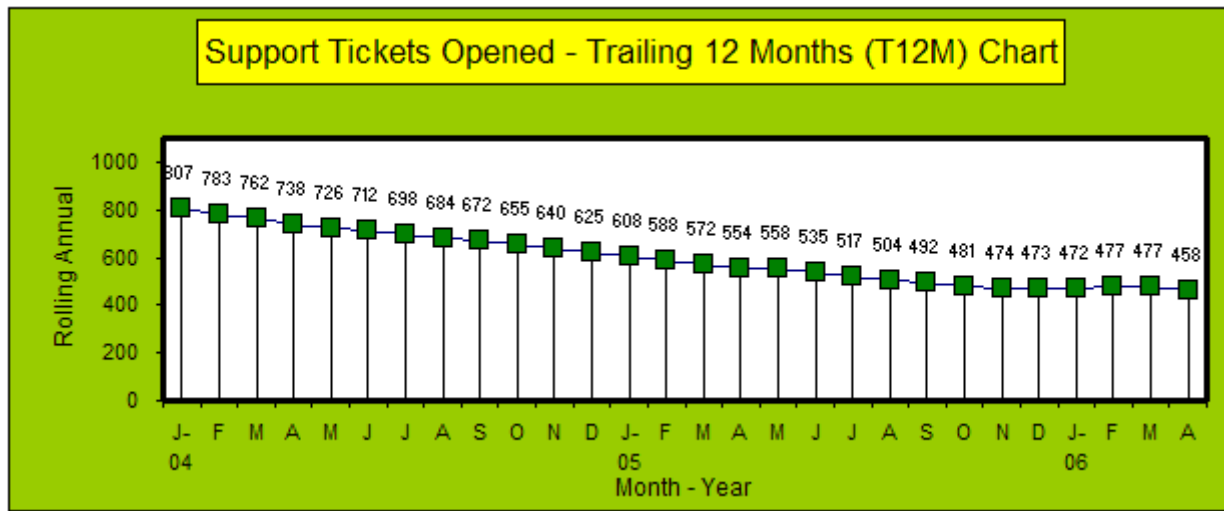
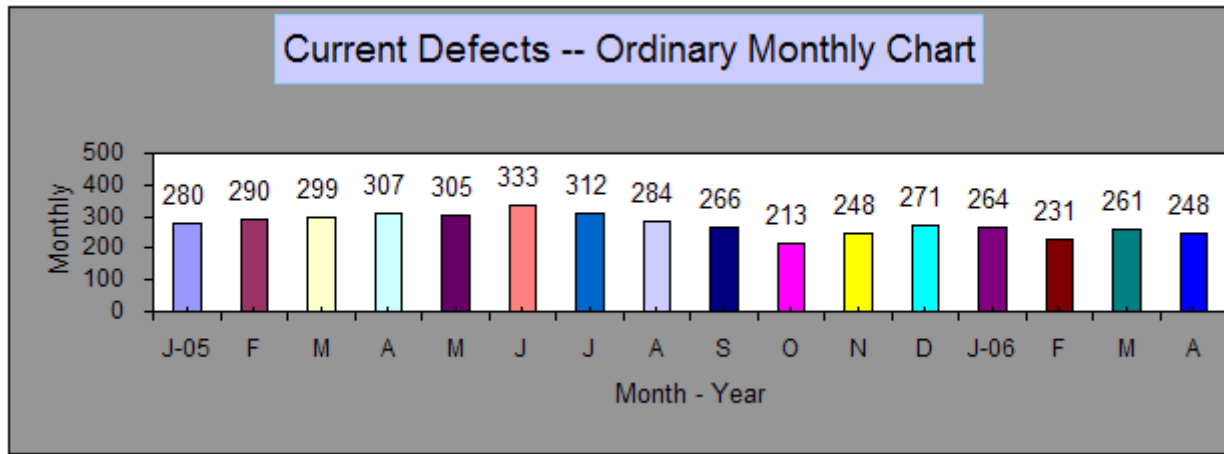
- **Project Variance Report:** http://www.PragmaticSW.com/newsletters/SP_ProjectTasksByProject.pdf
- **Project Variance by Assignee:** http://www.PragmaticSW.com/newsletters/SP_ProjectTasksByAssignee.pdf

3. **Goal #2 (Reduce Support Tickets and Defects)** - In our example, the departmental goal of reducing support tickets to 100 or less per month and maintaining 50 or less active defects will require that the technical lead collect this information.

To do this, record the number of support tickets that come in each day, as well as the number of open defects. Here is a spreadsheet to help with that:

http://www.PragmaticSW.com/Newsletters/SP_DefectMetrics.xls

From here, you can create graphs that can allow you to determine if you are meeting the goal:



Note: Software Planner has a [Daily Summary Report](http://www.SoftwarePlanner.com/Newsletters/SP_Daily_Summary_Report_viaEmail2.pdf) that is automatically emailed to you each day so that you do not have to manually collect these statistics each day. If you are not using Software Planner, determine if this feature is available from your defect tracking solution so that you ease the collection of statistics.
http://www.SoftwarePlanner.com/Newsletters/SP_Daily_Summary_Report_viaEmail2.pdf

- 4. **Goal #3 (Increase Product Marketability)** - In our example, the departmental goal of increasing product revenue by 30% will require that customer feature requests are tracked. It will also require that product revenue is tracked each month, comparing it to the prior year.

To do this, begin tracking customer feature requests. When doing product demos and customer surveys, keep track of each new request that clients ask for. Begin compiling a list, and make notations as to how often these requests come up. Then plow these new features into new releases of your software. Your spreadsheet may look similar to this:

Feature Requests

Feature Requested	# Times Requested
Print Preview Button	5
Microsoft Outlook Integration	25
Auditing	30
Email Alerts	5
Project Management Module	2

To ensure that your product revenue is increasing by 30%, you must track product sales each month, as compared to last year. This way you can determine if your product revenue is trending properly.

Product Revenue

Month	This Mo - 2006	This Mo - 2005	YTD - 2006	YTD - 2005	Pct Gain in 2006
Jan	500,000	400,000	500,000	400,000	20%
Feb	400,000	400,000	900,000	800,000	11%
Mar	800,000	500,000	1,700,000	1,300,000	23%
Apr	etc				

5. **Goal #4 (Increase Customer Satisfaction)** - In our example, the departmental goal of maintaining a customer satisfaction of 90% will require that surveys be administered and results tracked.

To do this, create email based surveys that ask your customer to rate their experience with your product. Allow them to choose a level of satisfaction with several aspects of your product. Also, ask them for feature requests (these can feedback into Goal #3). Keep the surveys brief, as to allow the client to quickly provide feedback. Below is an example:

Question	Your Answer (1-5) 1=Poor 2=Less than average 3=Average 4=Above average 5=Outstanding!
How would you rate the usability of the product?	
How would you rate the customer support of the product?	
How would you rate the flexibility of the product?	
How would you rate the overall satisfaction with the product?	
What new features would you like to see in the product?	

Customer Satisfaction Statistics

Client	Average Score	Pct of Satisfaction
Client A	3	50%
Client B	5	100%
Client C	4	80%
Client D	4.5	90%
Client E	4.5	90%

6. **Goal Tracking Best Practices** - When tracking goals, follow these best practices:

- Goals should be measurable and should have a timeframe.
- All team members that contribute to goals should be made aware of the goals, understand how to reach the goals, and understand their stake in it.
- Each team should meet weekly to record progress toward goals and to identify behaviors that must be changed to keep the team on track for the goals.
- Each department should meet monthly to record progress to the year-end goals and to identify behaviors that must be changed to keep the team on track for the goals.
- Once goals are achieved, all team members that contributed to the success of the goals should be rewarded in a timely manner.

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