MONTHLY NEWSLETTER   PRAGMATIC S·O·F·T·W·A·R·E

Apr 2009 - Pragmatic Software Newsletters

The Seven Habits of Highly Effective Scrum Masters

Published in 1989, *The Seven Habits of Highly Effective People*, written by Stephen R. Covey has helped millions establish great habits for achieving true interdependent effectiveness in their life and their jobs.   This article discusses the 7 habits, framing the habits for **highly effective scrum masters**. Below are the 7 habits:

1. Be Proactive
2. Begin with the End in Mind
3. Put First Things First
4. Think Win/Win
5. Seek First to Understand, Then to be Understood
6. Synergize
7. Sharpen the Saw

Habit 1 - Be Proactive
A Scrum Master's goal in any software project is to empower team members to get things done and to remove impediments. Below are some ideas for being proactive on software projects:

1. **Run Daily Scrum Meetings** - Each day, your team should hold a Daily Scrum Meeting. Normally, this daily meeting is set for 15 minutes and each team member is asked 3 questions:

   "What did you do yesterday?"
   "What will you do today?"
   "Are there any roadblocks or anything impeding your progress?"

   15 minutes is not always enough time to have a good dialog and a meaningful meeting, some days may take 30 to 45 minutes. To speed this up, consider having each team member post a summary of what they did yesterday and a summary of what they plan to do tomorrow in a daily discussion forum that is automatically distributed to all team members. This allows you to spend the Daily Scrum meeting talking about roadblocks, design decisions, and impediments to progress.

   References:
   Discussion Forums - http://www.pragmaticsw.com/GuidedTours/Default.asp?FileName=DiscussionBoard

2. **Best Foot Forward** - A key way to reduce cost and time overruns is to prevent them during design. Some Agile evangelists might argue that doing user stories (which normally do not contain detailed requirements, prototypes or designs) is the best approach. Although our team has been using Agile for years, we found that user stories are not detailed enough, as they tend to cause too much rework and have a tendency to reduce the accuracy of estimates. Agile is designed to be flexible and tweaked for your specific needs so by deploying a more structured requirements gathering, we enjoy the flexibility of Agile and the reduction of rework, providing us with the best of both worlds.

   References:
   Requirements - http://www.pragmaticsw.com/GuidedTours/Default.asp?

FileName=FunctionalSpecifications
Detailed Design - http://www.pragmaticsw.com/Templates.asp

3. **Review Metrics Daily** - It is important to ensure that the Agile sprint will complete on time. You can ensure this by having team members enter their time daily and reviewing burndown and velocity charts.

   References:
   Burn Down Chart - http://www.pragmaticsw.com/WhitePaper_SP_Metrics.pdf

Habit 2 - Begin with the End in Mind

Your end goal for the sprint should be to deliver high quality software that meets the goals of the sprint. Before coding begins, you should make a list of success criteria that you judge the project on. For example, your success criteria may be that the software produces specific results, has no known defects (or a small number of low severity defects), is reusable, is maintainable, is well documented, is easy to use, etc. By defining the success criteria up front, you can objectively evaluate whether the project met the criteria or not. Solicit help from all team members (product owner, scrum master, developer, tester, etc.) when defining the success criteria. By getting a team perspective of the success criteria, you will have better and more measurable criteria and you will get much better buy-in from the team. Below are some tips for ensuring your meet your success criteria at the end:

- **Identify success criteria** - Make sure your success criteria is published and agreed upon by the team members.
- **Review success criteria** - At least weekly (in one of your daily team meetings), review the success criteria. This can include reviewing your defect statistics, test case run history, etc. to allow you to determine if you are progressing towards the success criteria as the project continues.
- **Retrospective** - Once your project is complete, do a "post mortem" or "retrospective" to determine if you met your success criteria.

   References:
   Retrospectives -
   http://www.softwareplanner.com/Newsletters/newsletter_2008_08_SP.htm

Habit 3 - Put First Things First

Prioritizing work effort is critical. You must apply effort to the most important things first, followed by less important things. Work the higher priority items first, then the lower priority items if time allows.

Habit 4 - Think Win/Win

When dealing with projects, you want to foster a win/win relationship between your team and the client. The alternative to that is:

- **Win/Loss** - In this scenario, your team wins but the client loses. This can cause loss of future business with the client.
- **Lost/Win** - In this scenario, your team loses but the client wins. This can cause team burnout, financial distress and other issues.
- **Win/Win** - This is the scenario you want to foster. In this scenario, both your team and the client wins. How is it done? Normally this centers around the project management pyramid (Features, Time, Cost). To foster a win/win relationship, one of those variables must be flexible. Once your team and client agree to that, it is much easier to make

objective decisions about how to plan the project. Here are examples:

1. **More Features / Less Time** - The flexible variable is cost, so your client agrees to absorb more costs (you can hire more people).
2. **Cost Savings / Less Time** - The flexible variable is features, so less features will be delivered but costs and time will be less.
3. **More Features / More Costs** - The flexible variable is time, allowing you to extend the timeline.

Habit 5 - Seek First to Understand, Then to be Understood

Many of us have a bad habit of blocking out a conversation and not listening because we so desperately want our opinion to be heard.   Every team member (product owner, scrum master, developer, tester, etc.) has different experiences, different perspectives and motivations. Before you can solve any problem, it is important to first listen intently and diligently to fully understand the problem.   Once you feel you have all the facts, solicit ideas for multiple solutions. Having several options can provide better discussions and allows team members to tweak initial solutions into solutions that are more far reaching and solve the problem in a more direct way.   If you disagree with an approach, don't attack the person that offered the approach. Instead, explain based on your past experiences why you think there might be a better approach.

Habit 6 - Synergize

Team collaboration is the key to a synergized team.   A synergized team is made up of divergent team members that have different strengths, different backgrounds and different perspectives. Encourage these differences but provide your team with tools that allow you maximize their effectiveness. Highly collaborative teams communicate with each other by sharing their calendars, posting their statuses into discussion forums so that everyone is aware of what the other is doing and accomplishing. These teams keep track of all tasks they work on each day, the number of hours worked, the number of hours remaining and variances to plan. They also share documents that illustrate best practices and produce white papers that teach others what they have learned.

References:
Tools - http://www.pragmaticsw.com/GuidedTours/Default.asp?FileName=OverviewSoftwarePlanner
Calendar Sharing - http://www.pragmaticsw.com/GuidedTours/Default.asp?FileName=Calendar
Discussion Forums - http://www.pragmaticsw.com/GuidedTours/Default.asp?FileName=DiscussionBoard
Task Tracking - http://www.pragmaticsw.com/GuidedTours/Default.asp?FileName=ProjectPlans
Document Sharing - http://www.pragmaticsw.com/GuidedTours/Default.asp?FileName=SharedDocuments

Habit 7 - Sharpen the Saw

Productive scrum masters see the need to continue honing their skills and love learning new techniques, best practices and approaches. Below are some links that might be of interest to you:

- **Agile Overview** - http://www.pragmaticsw.com/Newsletters/newsletter_2008_02_SP.htm
- **Team Composition** - http://www.pragmaticsw.com/Newsletters/newsletter_2008_03_SP.htm

- **Understanding Scrum Rules** -
  http://www.pragmaticsw.com/Newsletters/newsletter_2008_04_SP.htm
- **Scrum Kickoff and Product Backlog** -
  http://www.pragmaticsw.com/Newsletters/newsletter_2008_05_SP.htm
- **The 30 Day Sprint and Daily Scrum Meeting** -
  http://www.pragmaticsw.com/Newsletters/newsletter_2008_06_SP.htm
- **Agile Reporting and Metrics** -
  http://www.pragmaticsw.com/Newsletters/newsletter_2008_07_SP.htm
- **Retrospectives** -
  http://www.pragmaticsw.com/Newsletters/newsletter_2008_08_SP.htm
- **Tailoring Scrum to your Needs** -
  http://www.pragmaticsw.com/Newsletters/newsletter_2008_09_SP.htm

Helpful Resources

Below are some helpful resources and templates to aid you in developing software solutions:

- **Software Planner** - http://www.SoftwarePlanner.com
- **Automated Testing Resources** - http://www.Star-QA.com
- **Pragmatic Agile Development** -http://www.pragmaticsw.com/PADOverview.pdf
- **Agile Training** - http://www.PragmaticSW.com/Services.asp
- **Software Development /QA Templates** -
  http://www.pragmaticsw.com/Templates.asp

About the Author

Steve Miller is the President of Pragmatic Software (http://www.PragmaticSW.com). With over 24 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.asp.