

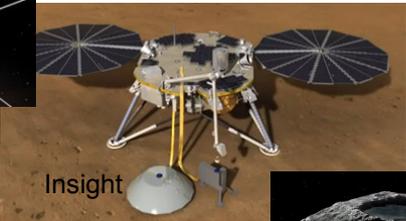
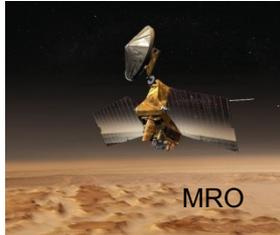
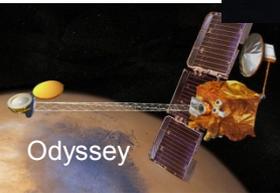


Using Modern Methodologies with Maintenance Software

Barbara Streiffert, Laurie Francis, Benjamin Smith

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA.

Barbara.Streiffert@jpl.nasa.gov
Laurie.K.Francis@jpl.nasa.gov
Benjamin.D.Smith@jpl.nasa.gov



Multi-Mission Planning & Sequencing Users



Agenda

- Background Mission Planning and Sequencing
- Mission Planning and Sequencing (MPS) Software
- Scrum Methodology
 - Scrum Methodology Tracking
 - Scrum Sample Burndown Chart
- MPS Scrum
- MPS Software Categories
- MPS Scrum Initial Characteristics
 - MPS Scrum Initial Task Board
 - MPS Initial Burndown Chart
- MPS Scrum Changes
 - MPS Scrum Burndown Charts
 - MPS Scrum WIKI Page
- MPS Scrum Current Process
 - MPS Scrum Greenhopper
- Conclusion
- MPS Scrum Standup Room



Background

- Mission Planning and Sequencing (MPS) provides Multi-Mission Software to JPL and other NASA Missions
- MPS provides and maintains 19 software applications
- The applications are NASA Class B (mission critical) and Class C (mission important)
- All of the software is in maintenance
- Defect correction and updates to support new mission capabilities are implemented and tested
- The software simulates the state of the spacecraft, translates and packages commands and provides utility services for editing and processing commands



Scrum Methodology

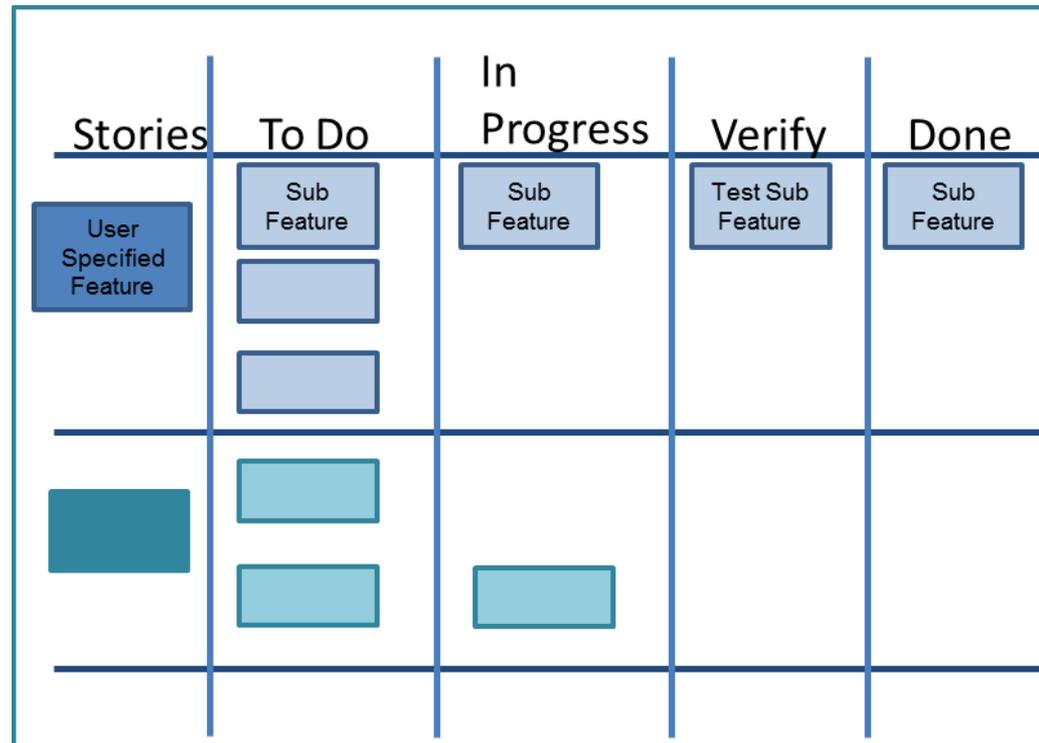
- One of the Agile Development Methodologies
- Incremental and Iterative Development
- Implementation performed in bursts called Sprints
- Sprints defined by the team and time-boxed
- Three roles: Scrum Master, Product Owner, Team
 - Scrum Master is the facilitator who makes sure everything moves smoothly
 - Product Owner sets the priorities for the tasks and represents the users
 - Teams pick tasks that can be performed in a Sprint based on priority
- Unselected tasks put on backlog
- Backlog tasks retrieved if team member finishes tasks early
- Three types of meetings: Planning, Standups, Retrospective
 - Planning Meeting: tasks selected by team
 - Standups: Detail what each team member did the day before, what they plan to do today and any blockage to the task. (Not a status meeting)
 - Retrospective: Details what items should be kept, what items should be changed and what items should be worked
- Demonstration at end of Sprint
- Typically used for new development



Scrum Methodology Tracking



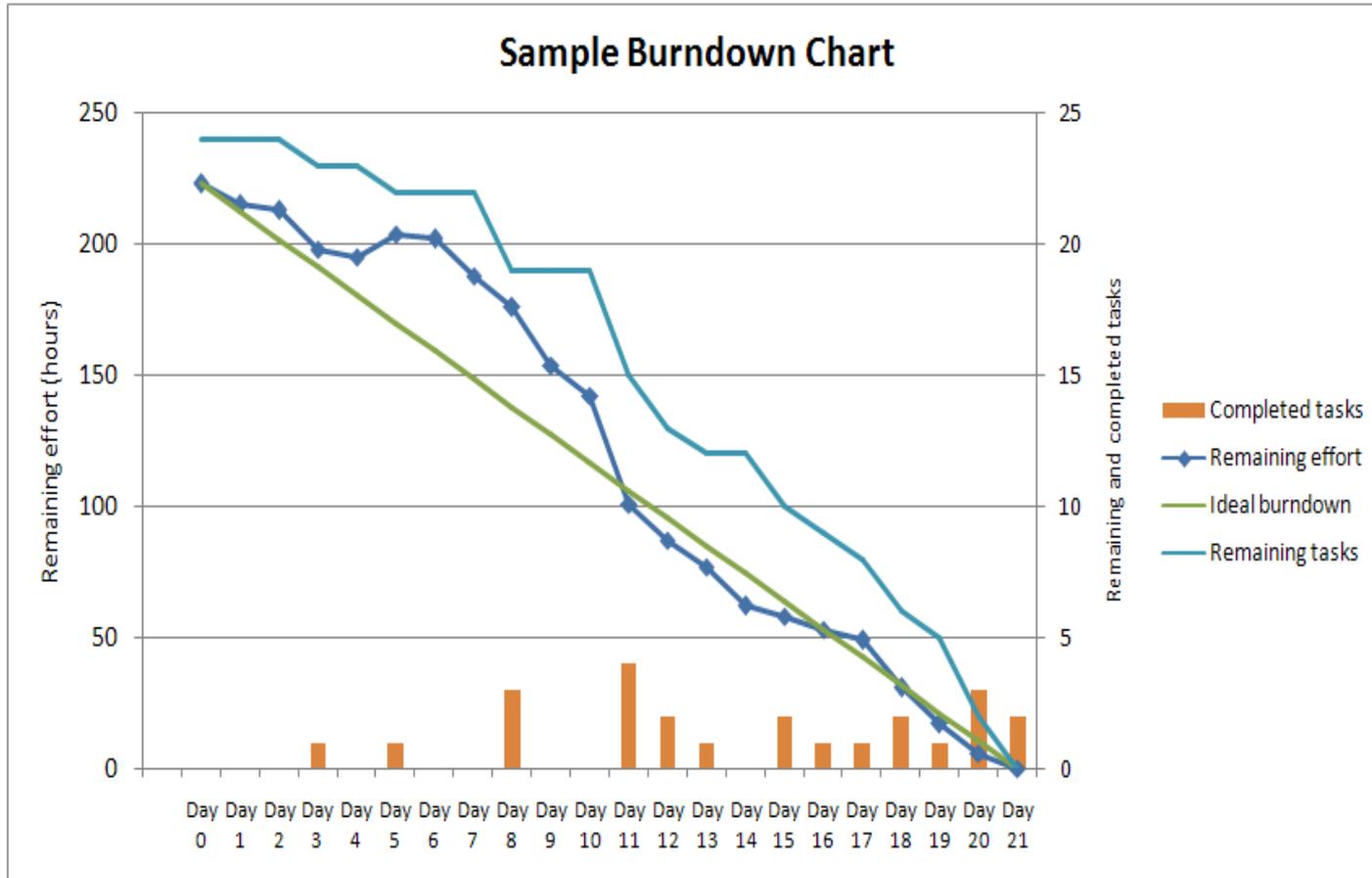
- Stories (issues) created for tasks
- Tasks kept on 3X5 cards
- Stories/Tasks given points for complexity
- Burndown charts kept to track task completion rate
- Charts can be kept based on issues, hours, or points



Standup Board Layout
Sample:



Scrum Sample Burn Down Chart



Burndown Chart in Hours



MPS Scrum

- Differences between MPS and other Scrum Teams
 - MPS has 19 applications instead of 1
 - MPS applications are in maintenance instead of new development
 - Software engineers are responsible for multiple applications instead of being responsible portions of a single application
 - Applications are implemented in multiple languages instead of one
 - Software engineers interruptible at any time based on missions' needs
- Learning Scrum Tenets
 - MPS Manager, System Engineer and Test Engineer attended Scrum Master course
 - Scrum instructor made suggestions
 - Scrum is flexible and should be tailored to fit the application
 - Task Software Engineers at a lower percentage to account for interruptions and keep the sprint from being interrupted
 - Keep the scrum roles and the basic meetings (planning, standups and retrospective)
 - Keep the concept of the sprint



MPS Software Categories



Variety of MPS Applications

Categories:	Components:
Editing Viewing Utilities	Automatic Sequence Checker Client/Server, MPS Editor Event Viewer, Uplink Summary Event Sorter, Command Translation Communication
Planning	Science, Engineering and Navigation Planning
Command Simulation/ Execution	Command Simulation Ground Sequence Engine Execution Flight Sequence Execution Engine
Command Packaging	Command Translation Command Packaging Sequence Compiler

Note: Color coding corresponds to slide on page 4



MPS Scrum Initial Characteristics

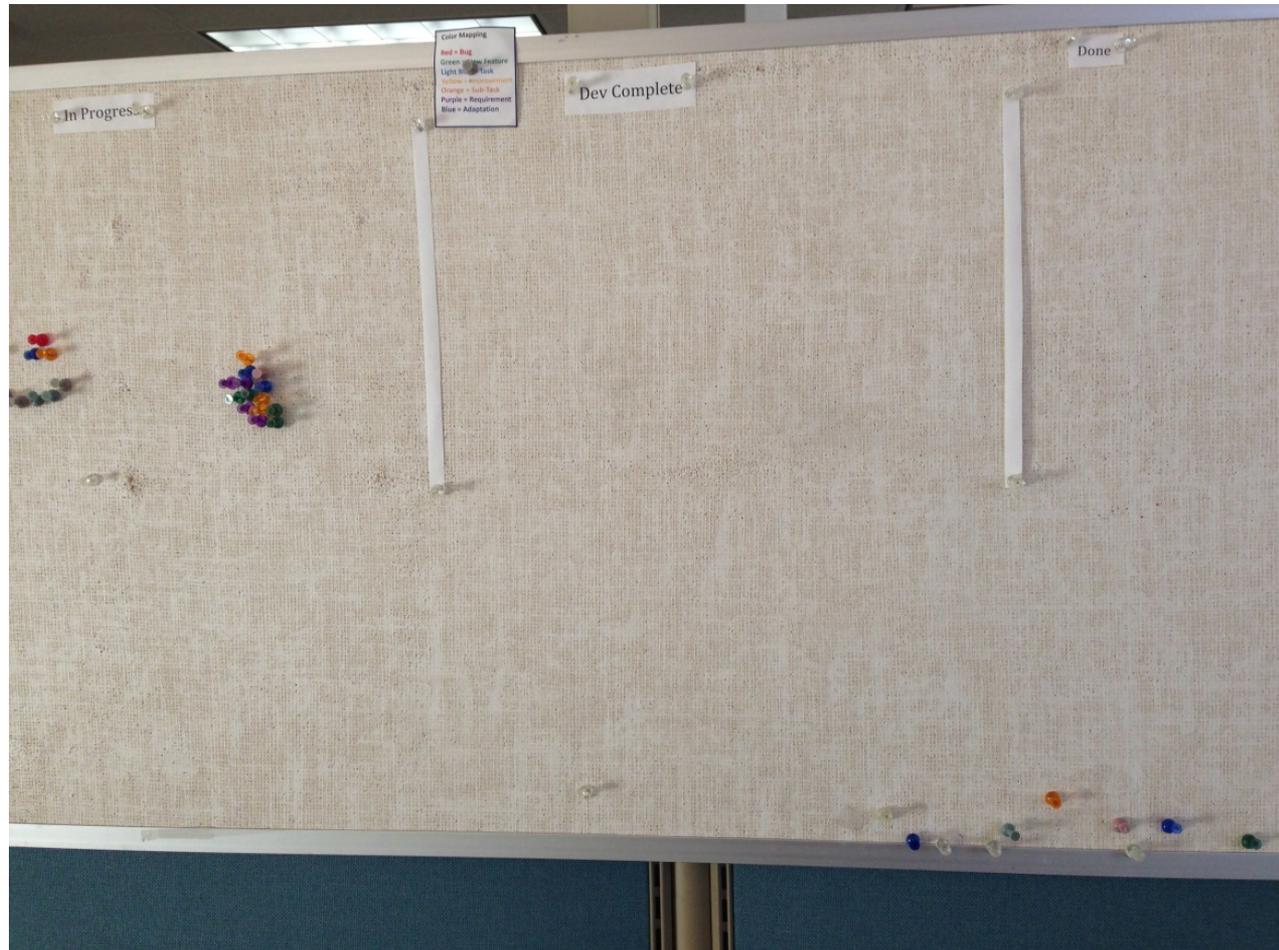


- MPS Core System Engineer became the Product Owner
- Test Engineer became the Scrum Master
- 3x5 Cards listed the Tasks on a Cork Board
- Columns on the Cork Board were “In Progress”, “Dev Complete” and “Done”
- Issues were kept in Jira (an issue tracking tool by Atlassian)
- At the planning meeting all the Jira issues were displayed and tasks were selected by team members based on priority
- Burndown chart was kept in Greenhopper (a Scrum tracking tool by Atlassian)
- Sprints were two weeks
- Standups were every day except Fridays (due to the 9/80 schedule)
- Team members were scheduled at 50% so that they would be available if a mission required support and the Sprint wouldn't be interrupted.
- If a team member finished the assigned tasks prior to the end of the sprint, then a task from the backlog would be added.



MPS Scrum Initial Task Board

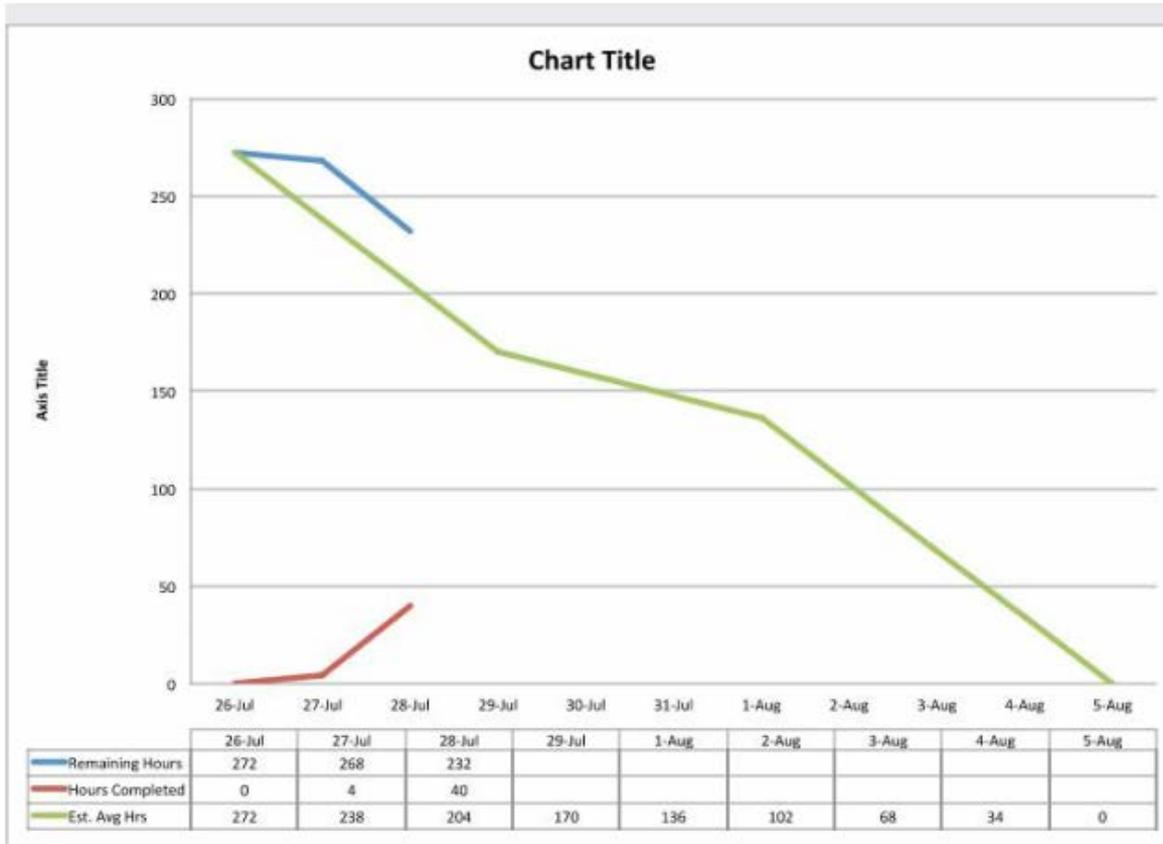
Task Board with Columns of In Progress, Dev Complete and Done





MPS Scrum Initial Burndown Chart

Burndown Chart in Hours





MPS Scrum Changes

One tenet of Scrum is to hold a retrospective at the end of the sprint. At the retrospective the following questions are asked:

1. What aspects of the sprint should be kept?
2. What aspects of the sprint should be changed?
3. What aspects of the sprint should be worked on?

Over time the following were changed:

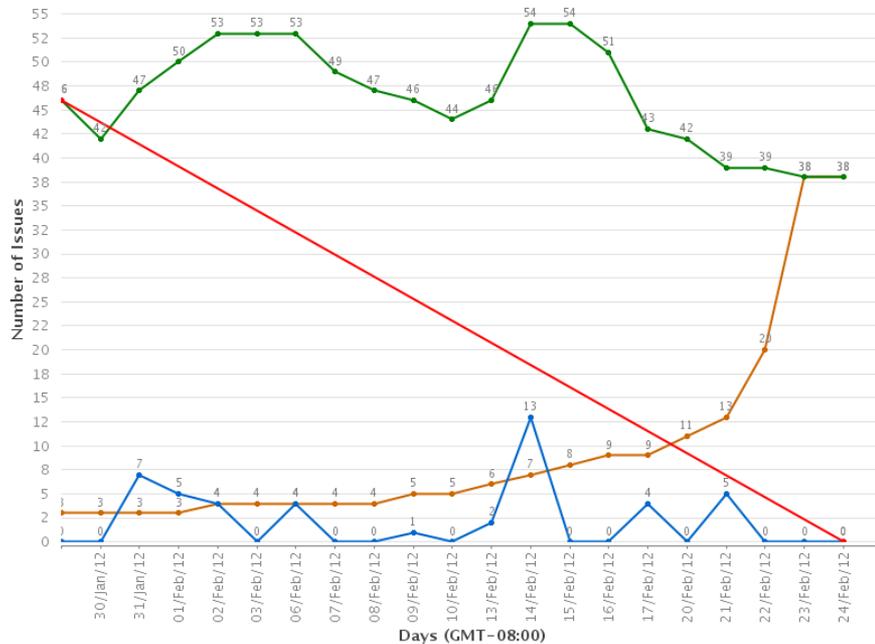
- Sprints were changed to one month
- The column headings were changed to “To Do”, “In Progress” and “Done”
- Greenhopper was used for standups and the cork board retired
- All coding tasks are required to have a “verify” task with it. (Note: the verify task is to build the automatic test to test the coding issue)
- Testing cycles would have sprints, but during those cycles standups would be on Mondays and Wednesdays
- Because of the volatile nature of needing to support missions, it was decided that tasks could be added to the sprint. However, a task of equal value was to be removed from the sprint. (Note: This aspect was not always successful)
- A set of WIKI pages keeps track of each sprint with the dates of the sprint, the issues involved, the final burndown chart and the retrospective notes



MPS Scrum Burndown Charts

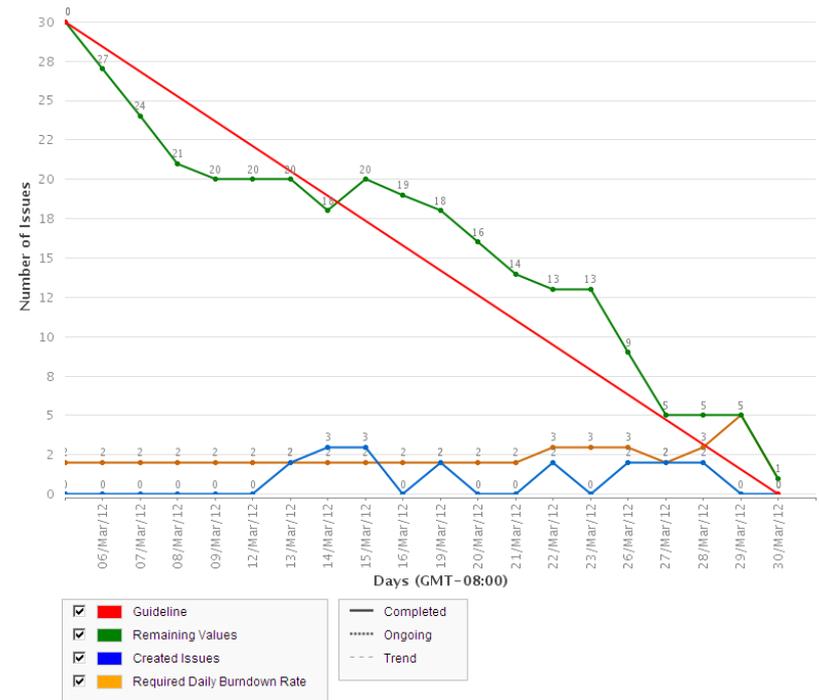
Unsuccessful Adding of Tasks:

Final Sprint Burndown



The blue line indicates task additions
 The green line indicates issues completion
 The red line indicates the ideal completion
 The orange line indicates the needed completion rate

Successful Adding of Tasks:





MPS Scrum WIKI Page

Scrum WIKI Documentation Includes:
Sprint Dates, Issues, Burndown Chart and Retrospective Notes

Dashboard > MPS > ... > SEQ Core Scrum > Sprint 2014-S1

Browse > Barbara Streiffert > Search

Component versioning

- SEQ Core Scrum
 - Process for Adding Sprint Pages
 - Sprint 32-B21
 - Sprint 32-B22
 - Sprint 32-B23
 - Sprint 32-T1
 - Sprint 32-T2
 - Sprint 32-T3
 - Sprint 32-T4
 - Sprint 2011-S1
 - Sprint 2011-S2
 - Sprint 2011-S3
 - Sprint 2012-S1
 - Sprint 2012-S2
 - Sprint 2012-S3
 - Sprint 2012-S4
 - Sprint 2012-S5
 - Sprint 2012-S6
 - Sprint 2012-S7
 - Sprint 2012-S8
 - Sprint 2012-S9
 - Sprint 2012-S10
 - Sprint 2012-S11
 - Sprint 2012-S12
 - Sprint 2013-S1
 - Sprint 2013-S2
 - Sprint 2013-S3
 - Sprint 2013-S4
 - Sprint 2013-S5
 - Sprint 2013-S6
 - Sprint 2013-S7
 - Sprint 2013-S8
 - Sprint 2013-S9
 - Sprint 2013-S10
 - Sprint 2013-S11
 - Sprint 2013-S12
 - Sprint 2014-S1

JPL Sprint 2014-S1 Edit Add Tools

Added by Barbara Streiffert, last edited by Barbara Streiffert on Jan 07, 2014 (view change)

Sprint Duration

January 6, 2013 - February 3, 2013

Final Sprint Burndown

Sprint Issue/Task List

[Sprint 2014-S1 Task List](#)

Retrospective Notes -

*Items to Keep:

*Items to Change:

*Items to Work:

[Add Labels](#)

Child Pages (1) Hide Child Pages | View in Hierarchy | Add Child Page

[Sprint 2014-S1 Task List](#)

[Add Comment](#)



MPS Scrum Current Process

The Current MPS Scrum Process has the following elements:

- The meetings: Planning, Standups and Retrospective
- The demonstrations are performed after multiple modifications have been made so that they are visible
- The roles are Product Owner, Scrum Master and Team
- Each Sprint is one month long
- Standups are held during test cycles, but are only on Mondays and Wednesdays
- Tasks may be added to the sprint due to mission needs, but with a discussion on whether all the tasks can still be completed or if a task must be removed.
- Jira and Greenhopper are used for tracking the sprint and the tasks in the sprint
- A Sprint WIKI is created for each sprint
- Coding tasks have at least one test task that builds the automatic test associated with it.
- The retrospective meeting for the current sprint occurs immediately prior to the planning meeting for the next sprint
- Stories are created for the issues. The story may contain multiple tasks or it may be a bug issue.
- Team members are scheduled at 50% of their time to allow for interruptions by mission needs.



MPS Scrum Greenhopper

MPS Planning Board for a Sprint

MPS Task Board by Team Member

The screenshot shows the JIRA Planning Board for a sprint. The board is titled 'MPS Core - On the fly' and is in 'Agile' mode. It displays a Kanban-style view with columns for 'To Do', 'In Progress', and 'Done'. Issues are grouped into columns of four. A detailed view of an issue is shown on the right, including fields like 'Component', 'Original Estimate', 'Time Spent', and 'Status'.

The screenshot shows the JIRA Task Board by Team Member. The board is titled 'MPS Core - Default' and is in 'Agile' mode. It displays a Kanban-style view with columns for 'To Do', 'In Progress', and 'Done'. Issues are grouped into columns of four. A detailed view of an issue is shown on the right, including fields like 'Component', 'Original Estimate', 'Time Spent', and 'Status'.

Planning board displays all issues for a given sprint in groups of four and displays the details of the sprint such as start and end date

Task board displays individual tasks for one team member for a given sprint



Conclusion

Even though MPS doesn't fit those who typically use Scrum, its flexibility has allowed MPS to use it. MPS uses the basics of Scrum, but has tailored Scrum to meet the MPS needs. Scrum has been successful and helped in the following ways:

1. The estimations of the team on what can be done are better
2. The communication and understanding of the tasks among the team members is better
3. The communication and understanding of the tasks with the system engineer and the test engineers is better
4. Scheduling the work has been easier
5. Adding test tasks has allowed the component automatic tests to be built during development and not at the end of development.
6. Identifying problems earlier in the cycle so that they can be fixed earlier is improved.
7. Allowing customers to know the status of the capabilities to be delivered and to take advantage of early releases in the form of engineering/evaluation versions is improved.

The MPS team is incredibly professional and productive. Scrum continues to be an experiment – so far a successful one.



MPS Scrum Standup Room

MPS Product Owner in Standup Room

