

# A Roadmap for Using Agile Development in a Traditional System



*Phoenix*

**Barbara Streiffert  
Thomas Starbird**

**Jet Propulsion Laboratory,  
California Institute of Technology  
Pasadena, CA**

---

**[Barbara.Streiffert@jpl.nasa.gov](mailto:Barbara.Streiffert@jpl.nasa.gov)  
[Tom.Starbird@jpl.nasa.gov](mailto:Tom.Starbird@jpl.nasa.gov)**

*Mars Science Laboratory (MSL)*

**SpaceOps 2006 Rome, Italy June 19 – 21, 2006**



# Agenda

- **Introduction**
- **Agile Development**
- **Waterfall Development**
- **Ensemble Agile Development**
- **NASA and JPL Requirements**
- **Software Management Plan**
- **Ensemble Software Development Process**
- **Conclusion**



# Introduction

## Ensemble Development Group

- Produces activity planning software for in-situ spacecraft
- Built on Eclipse Rich Client Platform (open source development and runtime software)
- Funded by multiple sources including the Mars Technology Program
- Incorporated the use of Agile Development

## Next Generation Uplink Planning System

- Researches the Activity Planning and Sequencing Subsystem for Mars Science Laboratory (APSS)
- APSS includes Ensemble, Activity Modeling, Constraint Checking, Command Editing and Sequencing tools plus other uplink generation utilities.
- Funded by the Mars Technology Program
- Integrates all of the tools for APSS



# Agile Development

## Agile Development Methodology

- **Goals**
  - **Produce Usable Software in Short Time Increments**
  - **Accommodates changes or reordering of implementation**
  - **Reduce Risk**
- **Characteristics**
  - **Customer is key part of the process**
  - **Automatic tests are created prior to implementation**
  - **Generally implementation is one to four weeks duration**
  - **Functional capabilities are divided into small chunks (called stories)**
  - **Frequent build process includes regression testing**



# Waterfall Development

## Waterfall Development Methodology

- **Goals**

- **Linear approach to requirements, design, implementation and test**
- **To produce usable software at end of single cycle**
- **To be a predictive approach for scheduling**

- **Characteristics**

- **Customer is part of requirements and acceptance testing**
- **Testing stage is performed at the end of implementation**
- **Milestones are set with deliverables due at a milestone**
- **Changes to requirements/implementation are tracked as change requests**
- **Failure reports are scheduled for future deliveries based on severity.**



# Agile Development at JPL

## Ensemble Team Approach to Agile Development:

- **Process:**
  - **Determine requirement/issue/functional capability (typically can be implemented in 1 week or less)**
  - **Discuss in weekly design team meeting**
  - **Document design in Confluence (a wiki – server software used to create and modify web page content)**
  - **Assign tasks for item 1 in weekly planning meeting**
  - **Enter tasks into JIRA (an issue tracking system)**
  - **Software pairs design automatic tests**
  - **Software pairs fix any defects**
  - **Software pairs integrate requirement**
  - **Software is rebuilt**
  - **Repeat weekly**



# Agile Development at JPL

## Ensemble Team Approach to Agile Development: Confluence Sample

Dashboard > Ensemble Development > Ensemble Software Development Process > A Roadmap for Using Agile Development in a Traditional Environment

Welcome Thomas

Ensemble Development ☆ ✉

**A Roadmap for Using Agile Development in a Traditional Environment**

Added by [Thomas Starbird](#), last edited by [Thomas Starbird](#) on May 15, 2006 ([view change](#))

Labels: (None) [EDIT](#)

This is a draft of a paper being submitted to SpaceOps 2006

**A Roadmap for Using Agile Development in a Traditional Environment**

Barbara Streiffert\* and Thomas Starbird\*\*  
Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, 91109  
Sven Grenander\*\*\*  
Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, 91109

**Abstract**

One of the newer classes of software engineering techniques is called "Agile Development". In Agile Development software engineers take small implementation steps and, in some cases, they program in pairs. In addition, they develop automatic tests prior to implementing their small functional piece. Agile Development focuses on rapid turnaround, incremental planning, customer involvement and continuous integration. Agile Development is not the traditional waterfall method or even a rapid prototyping method (although this methodology is closer to Agile Development). At Jet Propulsion Laboratory (JPL) a few groups have begun Agile Development software implementations. The difficulty with this approach becomes apparent when Agile Development is used in an organization that has specific criteria and requirements handed down for how software development is to be performed. The work at the JPL is performed for the National Aeronautics and Space Agency (NASA). Both organizations have specific requirements, rules and processes for developing software. This paper will discuss the some of the initial uses of the Agile Development methodology, the spread of this method and the current status of the successful incorporation into the current JPL development policies.

**I. Introduction**

Agile Development is a new software methodology that is gaining acceptance throughout the software community. Most Agile



# Agile Development at JPL

## Ensemble Team Approach to Agile Development: JIRA Sample

**JIRA** Ensemble JIRA  
HOME BROWSE PROJECT FIND ISSUES CREATE NEW ISSUE

All Projects : **Maestro** (Key: MAE)

Project Lead: [Jeff Norris](#)  
URL: <http://qong.jpl.nasa.gov:8080/>  
Description:  
Main development effort for Maestro.

[Create a new issue in project Maestro](#)  
 [Release Notes](#)

Select: [Open Issues](#) [Road Map](#) [Change Log](#)

### Road Map

[View personal road map](#)  
 Scope: next 3 versions | [all versions](#)

**Iteration 82** ( 26/May/06 | [Release Notes](#) )  
5/22 - 5/26 Progress: 0 of 2 issues have been resolved

- [MAE-1186](#) UNRESOLVED [WA/TDA for M&A task](#) ↑
- [MAE-1155](#) UNRESOLVED [data product view refactor](#) ↑

**Iteration 81** ( 19/May/06 | [Release Notes](#) )  
5/15 - 5/19 Progress: 0 of 24 issues have been resolved

- [MAE-1150](#) UNRESOLVED [ATHLETE telemetry: JMS-PVM converter](#) ↑
- [MAE-1178](#) UNRESOLVED [Athlete moldump log products](#) ↑
- [MAE-1176](#) UNRESOLVED [Athsim Sequence upload](#) ↑
- [MAE-1177](#) UNRESOLVED [Athsim command dictionary](#) ↑
- [MAE-1181](#) UNRESOLVED [CEV: ISS data server](#) ↑
- [MAE-1159](#) UNRESOLVED [Choice should also have a display value](#) ↑
- [MAE-1175](#) UNRESOLVED [Data Type for Mini-TES missing](#) ↑



# Agile Development at JPL

## Ensemble Team Approach to Agile Development: Cruise Control Sample

[Turn autorefresh on](#)

Project	Last build result	Last build time	Last successful build time	Last label
<a href="#">MERDataProductCatalogPopulator</a>	passed	05/15/2006 04:45:14	05/15/2006 04:45:14	build.666
<a href="#">MSLFeature</a>	passed	05/15/2006 04:22:13	05/15/2006 04:22:13	build.1196
<a href="#">MaestroMER</a>	passed	05/15/2006 04:01:13	05/15/2006 04:01:13	build.859
<a href="#">Moonrise</a>	passed	05/11/2005 17:34:10	05/11/2005 17:34:10	build.133
<a href="#">PDSToJPEG</a>	passed	05/15/2006 04:18:32	05/15/2006 04:18:32	build.601
<a href="#">Phoenix</a>	passed	05/15/2006 05:01:28	05/15/2006 05:01:28	build.593
<a href="#">RoverWare</a>	passed	05/15/2006 04:51:43	05/15/2006 04:51:43	build.669
<a href="#">SASPaH</a>	passed	05/15/2006 04:36:35	05/15/2006 04:36:35	build.313
<b>Total</b>	<b>8</b>			
Passed	8	100%		

listing generated at 13:14



# NASA and JPL Requirements

## NASA Requirements:

- **NASA Procedural Requirements (NPR's)**
  - **NASA 7150.2 governs software development at NASA centers**

## JPL Requirements:

- **Software Development Requirements (SDR)**
  - **SDR is JPL's response to NASA's 7150.2**
  - **SDR contains requirements that correspond to applicable 7150.2 requirements.**
  - **All software development at JPL is to be SDR compliant.**
  - **Software classifications are defined in the SDR.**
    - **A = Human Rated**
    - **B = Mission Critical**
    - **C = Contributes to Mission Objectives but not Critical**
    - **D = No Impact to Mission Primary or Secondary Objectives**
  - **SDR requires Software Management Plan (SMP)**



# Software Management Plan

## **SMP Includes Requirements on:**

- **Roles & Responsibilities**
- **Reviews**
- **Design**
- **Development**
- **Test**
- **Delivery**
- **Documentation**
- **Budget Data (including Work Breakdown Structure – WBS)**
- **Software Acquisition**
- **Software Safety**
- **And Adherence to JPL Software Processes**



# Ensemble Software Development Process

**JPL established a group called Software Quality Improvement (SQI).**

- **Goals:**
  - **Help projects to write SMP's and Development Processes**
  - **Ensure compliance with the JPL SDR**
  - **Ensure compliance with Capability Maturity Model Integrated (CMMI)**

**Ensemble SE, NGUPS SE and SQI met to develop an acceptable Agile Development Process.**

**The Ensemble Software Development Process has been created**



# Ensemble Software Development Process

Jet Propulsion Laboratory

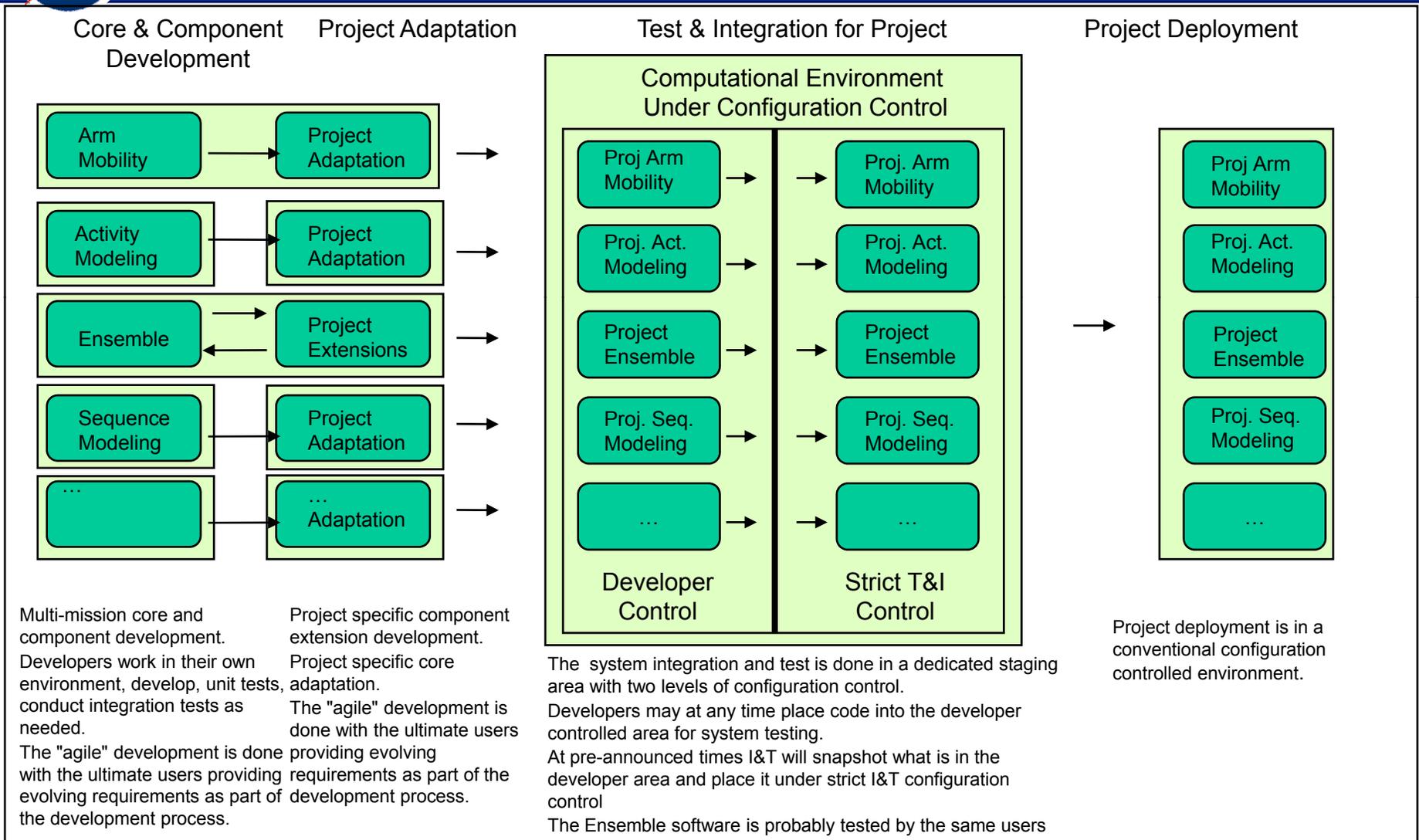
## **The Ensemble Software Development Process:**

### **Some Modifications to the Ensemble Agile Development Process in the Document are:**

- 1. In addition, to the weekly design and planning meetings high level planning that creates long-term agreement on release dates and general contents at a high level of functionality are addressed.**
- 2. Reviews are specified.**
- 3. Testing and deliveries are staged with tighter controls imposed as the process continues. The next slide shows the progression.**
- 4. Other elements from the SDR are addressed as well including budgetary and safety ones.**



# Ensemble Delivery Stages



Multi-mission core and component development. Developers work in their own environment, develop, unit tests, conduct integration tests as needed. The "agile" development is done providing evolving with the ultimate users providing requirements as part of the development process.

Project specific component extension development. Project specific core adaptation. The "agile" development is done with the ultimate users

The system integration and test is done in a dedicated staging area with two levels of configuration control. Developers may at any time place code into the developer controlled area for system testing. At pre-announced times I&T will snapshot what is in the developer area and place it under strict I&T configuration control. The Ensemble software is probably tested by the same users

Project deployment is in a conventional configuration controlled environment.



## Conclusion

**The Ensemble Software Development Process has the support of SQL and currently only needs some minor modifications to be compliant.**

**At JPL the Agile Development Process will soon be a reality in a traditional system (with a few compromises)!**