An Integrated Toolset for Agile Systems Engineering Requirements Analysis

Phyllis Marbach

19 May 2011

This document does not contain technical data within the definition contained in the International Traffic in Arms Regulations (ITAR) and the Export Administration Regulations (EAR), as such is releasable by any means to any person whether in the U.S. or abroad. The Export Compliance log number for this document is Export Approval # RBE3973-NT (assigned IAW PRO-4527, PRO 3439).
<table>
<thead>
<tr>
<th>1. REPORT DATE</th>
<th>2. REPORT TYPE</th>
<th>3. DATES COVERED</th>
</tr>
</thead>
<tbody>
<tr>
<td>19 MAY 2011</td>
<td></td>
<td>00-00-2011 to 00-00-2011</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4. TITLE AND SUBTITLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>An Integrated Toolset for Agile Systems Engineering Requirements Analysis</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>5a. CONTRACT NUMBER</th>
<th>5b. GRANT NUMBER</th>
<th>5c. PROGRAM ELEMENT NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>5d. PROJECT NUMBER</th>
<th>5e. TASK NUMBER</th>
<th>5f. WORK UNIT NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>6. AUTHOR(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boeing Defense, Space &amp; Security, PO Box 516, St. Louis, MO, 63166</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>8. PERFORMING ORGANIZATION REPORT NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>10. SPONSOR/MONITOR’S ACRONYM(S)</th>
<th>11. SPONSOR/MONITOR’S REPORT NUMBER(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>12. DISTRIBUTION/AVAILABILITY STATEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approved for public release; distribution unlimited</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>13. SUPPLEMENTARY NOTES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Presented at the 23rd Systems and Software Technology Conference (SSTC), 16-19 May 2011, Salt Lake City, UT. Sponsored in part by the USAF. U.S. Government or Federal Rights License</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>14. ABSTRACT</th>
</tr>
</thead>
<tbody>
<tr>
<td>An Integrated Toolset for Agile Systems Engineering Requirements Analysis introduces an iterative approach to analyzing requirements and the integrated toolset that enables this analysis. The requirements analysis may occur in parallel and one iteration ahead of software development or it may occur after software exists to produce needed artifacts as in the case to be presented. An agile process for systems engineering to analyze requirements and provide to the agile software team will be introduced. The commercial-off-the-shelf tools that allow collaboration, standard formatting, logging unique identification numbering, linking to defined tests, and formal document production, configuration control and actual program use will be presented. This end-to-end integrated toolset allows the sync-up of the requirements to test descriptions not only at each formal release, but daily as the iterations Copyright ? 2010 Boeing. All rights reserved. 28 requirements evolve during the iterations. This results in planned deliverable products at the end of each iteration to internal and external customers.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>15. SUBJECT TERMS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>16. SECURITY CLASSIFICATION OF:</th>
<th>17. LIMITATION OF ABSTRACT</th>
<th>18. NUMBER OF PAGES</th>
<th>19a. NAME OF RESPONSIBLE PERSON</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. REPORT unclassified</td>
<td>Same as Report (SAR)</td>
<td>28</td>
<td></td>
</tr>
<tr>
<td>b. ABSTRACT unclassified</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c. THIS PAGE unclassified</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Standard Form 298 (Rev. 8-98)  Prescribed by ANSI Std Z39-18
**Scrum** is an **iterative, incremental** methodology for project management often seen in **agile software development**, a type of **software engineering**.

![Scrum Process](http://en.wikipedia.org/wiki/File:Scrum_process.svg)
Introduction to Agile Systems Engineering

For Software Development Requirements Analysis

Boeing Defense Space & Security | Lean-Agile Software

Agile Systems Engineering Activities

- **Requirements Iterations**
  - Identify, gather, define, and develop requirements (in story format)
  - Stories selected and estimated by team based on importance and need

- **Managed Iterations**
  - Controlled and Managed

- **Prioritized**

Agile Software Development Activities

- **Development Iterations**
  - Stories
  - Stories broken into tasks and estimated

- **Sprint Backlog**
  - Prioritized Stories broken into tasks and estimated
  - Stories selected and estimated by team based on importance and need

- **Product Backlog**
  - Controlled and Managed

- **Sprint**
  - Design
  - Code
  - Test
  - Acceptance testing
  - Functional system integration
  - Unit integration

Richard Carlson, Phillip Matzuc; A Viable Systems Engineering Approach, SSTC 2010

Copyright © 2010 Boeing. All rights reserved.
Agile Sys Engrg Requirements Analysis
Parallel and One Iteration Ahead of Software Development

Boeing Defense Space & Security | Lean-Agile Software

Richard Carlson, Phillip Matzuc; A Viable Systems Engineering Approach, SSTC 2010
Analysis of Existing Code

Boeing Defense Space & Security | Lean-Agile Software

- **Have:**
  - Code
  - Test Bed
  - User Interface
  - User Procedures

- **Lack:**
  - Requirements documentation
  - Architecture & Design Diagrams
  - Trace Matrix of Tests to Requirements
  - Software Test Descriptions
  - Hazard Analysis
Analysis with Integrated Toolset

Boeing Defense Space & Security | Lean-Agile Software

Collaboration Tool

1. Use Cases
2. Functional Requirements
3. Hazard Analysis
4. Software Test Descriptions

Web Pages

Data Mgmt Tool

- Trace
- STD
- SRS

Deliverable Quality Docs

Peer Review

Requirements

ID #-------
ID #-------
ID #-------
ID #-------

Test

STD

STD

STD

No: update, peer review, rerun test

Norms

Yes, into tool.

link to reqmts

STDDone?

?No: update, peer review, rerun test
Getting Started

- Code
- Domain experts – not always available
- Existing documentation in program repositories – charts, operator procedures
- Determine Next Steps
Epics and Backlog Items

30 Epics were created from the User Interface Features, examples:
- Power On
- Start Up Feature
- Shutdown Feature
- Operate Component
- Operate Another Component

Product Owner prioritized the most important ones

Each epic has 5 significant backlog items (took 3 Iterations to reach these 5):
- Functional Analysis
- Requirements
- Hazard Analysis
- Draft Test Procedure
- Finalize Test Procedure
### Created templates

- Goal is to identify tasks that take 16 hour max
- Include what “Done” means in the template

#### Backlog Item Templates

<table>
<thead>
<tr>
<th>Title</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Story Template - OLD</td>
<td>B-01017</td>
</tr>
<tr>
<td>Task Template - OLD</td>
<td>B-01018</td>
</tr>
<tr>
<td>Update Documentation or Work Products Template</td>
<td>B-01243</td>
</tr>
<tr>
<td>Functional Analysis Template</td>
<td>B-01225</td>
</tr>
<tr>
<td>Research and Document Functionality</td>
<td>TK-02479</td>
</tr>
<tr>
<td>Requirements Template</td>
<td>B-01216</td>
</tr>
<tr>
<td>Generate Functional Requirements</td>
<td>TK-02420</td>
</tr>
<tr>
<td>Peer Review Requirements</td>
<td>TK-02547</td>
</tr>
<tr>
<td>Update and Post Requirements</td>
<td>TK-02469</td>
</tr>
<tr>
<td>Hazard Analysis Template</td>
<td>B-01219</td>
</tr>
<tr>
<td>Identify and Analyze Potential Hazards</td>
<td>TK-02546</td>
</tr>
<tr>
<td>Peer Review Hazard Analysis</td>
<td>TK-02547</td>
</tr>
<tr>
<td>Update Hazard Analysis</td>
<td>TK-02548</td>
</tr>
<tr>
<td>Draft Test Procedures Template</td>
<td>B-01252</td>
</tr>
<tr>
<td>Generate Draft Test Procedures</td>
<td>TK-02543</td>
</tr>
<tr>
<td>Peer Review Draft Test Procedures</td>
<td>TK-02544</td>
</tr>
<tr>
<td>Update and Post Draft Test Procedures</td>
<td>TK-02545</td>
</tr>
<tr>
<td>Finalize Test Procedure Template</td>
<td>B-01251</td>
</tr>
<tr>
<td>Run Test Procedures</td>
<td>TK-02539</td>
</tr>
<tr>
<td>Update and Post Finalized Test Procedures</td>
<td>TK-02540</td>
</tr>
</tbody>
</table>
Manage the Backlog

Boeing Defense Space & Security | Lean-Agile Software

- **Application Lifecycle Management (ALM) Tools**
  - IBM – Rational Team Concert
  - MKS, Inc.
  - Atlassian - JIRA with GreenHopper
  - CollabNet
  - HP
  - Micro Focus
  - Microsoft
  - Rally Software Development
  - Serena Software
  - VersionOne

Integrated End-to-End Toolset

Boeing Defense Space & Security | Lean-Agile Software

Collaboration
Standard formatting

Data Management
Configuration Control

Standard formatting
Logging
Unique identification numbering
Linking to defined tests
Formal document production

Collaboration Tool

1. Use Cases
2. Functional Requirements
3. Hazard Analysis
4. Software Test Descriptions

Web Pages

Deliverable Quality Docs

Trace

STD

SRS

Reqmts Mgmt Tool

Requirements
ID #-------
ID #-------
ID #-------

3b

Yes, into tool, link to reqmts

Test

STD

Reqmts

No: update, peer review, rerun test

Copyright © 2010 Boeing. All rights reserved.
Collaboration tool should be:
- Easy to access
- Easy to use
- Easy to comment
- Easy to change

Team started with a collaborative tool
- Mediawiki, open source
- TWiki™, open source, collaboration platform
- Confluence
- SharePoint
- Socialtext
Collaboration – Home Page

- Introduction about the analysis underway
- Link to a list of functional threads: links have the work products themselves
- Links to references used
- Links to test environment information
- Links to templates for work products with instructions
  - Collaboration Tool Templates
  - Functional Descriptions
  - Requirements/Use Cases/
  - Hazard Analysis/Risk Mitigation
  - Test Procedures/Test Cases/Test Descriptions
Example

- Collaboration tool exports content to a Word Document
- Word Document is parsed into DOORS
- All feature reqmts in DOORS create final SRS
- Released documents are baselined in the Data Mgmt Tool Repository that provides Configuration Mgmt control

- Description of Functionality
  - Overview
  - Functional Decomposition
  - Use Case Development
    - Phase 1 level

- Requirements
  - Use Case Development
    - Phase 1 (operator/functional) level
  - Functional Requirements
  - Requirements Documents
    - SRS Document in TWiki
    - SRS Document in DOORS
    - SRS Document in PIMS

- Test Procedures
  - Existing Test Procedures
  - FQT Team Test Case/Test Procedure Development
    - Test Cases
    - Test Procedure Document
    - Expected Test Results
    - Test Procedures to Requirements Trace
    - Software Test Description (STD)

- Test Results

- Hazard Analysis/Risk Mitigation
  - Hazards/Mitigation
Capabilities include:

- Draft folders/repository
- Peer Review records
- Action Item creating/tracking/closure
- Release folders/repository
- Calendar
- Meeting notification
- Distribution Lists and access control to records
- Configuration Management work flow and approvals
- Collaboration across companies, subcontractors, customers

Examples (to name a few)

- Master Data Management Tool: Microsoft, Data Foundations, Kalido,
- Business to Business Tools, Amalto Technologies, Entreon Corp.,
During each iteration:
- Software Requirement Specification is created Feature by Feature rather than all at once.
- Software Test Descriptions are created as each feature is analyzed
- Hazard Analysis is performed one feature at a time.

At each release:
- More features are complete within the SRS,
- More STDs are complete and
- More Hazards Analysis are complete.
Each backlog item included conducting peer reviews of the content.

The peer review was the acceptance criteria before work could be posted into DOORS or claimed done and be included into the demonstration of the iteration.

One team member responsible for the entire backlog item of tasks but other team members might be actually performing the tasks.

Conducting peer reviews, as soon as possible, instead of waiting until the week before the demonstration, helped get findings removed and more work ready for demonstration.

Peer Review records were kept in a data management tool where the data included:

- Artifact in review
- Peer review date,
- Personnel reviewing,
- Time spent reviewing,
- Findings discovered and removed
Define “Done” – Includes Logging

Boeing Defense Space & Security | Lean-Agile Software

- Work was not complete until content was posted into the Requirements Management Tool
- INCOSE site has 34 listed as of 3/18/2011: http://www.incose.org/ProductsPubs/products/rmsurvey.aspx
- Two in use at Boeing:
  - IBM Rational DOORS
  - IBM Rational RequisitePro
- Unique record Identification numbers are automatically assigned
- Developed the software test descriptions
- Ran these in the Test Lab to verify complete
- Found some common repeatable test steps
- Created these as common test descriptions that could be called from other procedures reducing work and making future test procedure development faster.
- Linking the STDs to the requirements in the requirements management tool began the Trace Matrix
The format for the Software Requirement Specification followed company standards and was populated into the requirements management tool.

As each release occurred the document produced met format standards.

The format of the Software Test Descriptions followed company standards and was posted into the collaboration tool so all documents started with the standard format.

STDs were posted into requirements management tool and content linked to requirements to enable the production of the Trace Matrix eventually.
Each day the requirements in the Requirements Management Tool was exported into an HTML file onto a web page so that those not familiar with or licensed for the Requirements Management Tool could see the latest, most complete list.

Each month the requirements and test descriptions in the Requirements Management Tool could be exported into word files to be posted as drafts into the Data Management Repository so those more comfortable working with documents could see the latest demonstrated list.

Each quarter the requirements and test descriptions in the Requirements Management Tool were exported into word files to be posted as releases into the Data Management Repository.
Configuration Control for SRS

Boeing Defense Space & Security | Lean-Agile Software

- **Sequence of events:**
  - Develop draft in collaboration tool
  - Perform peer review using the data management tool
  - Populate requirements management tool
  - Create the final SRS word document from the requirements management tool and post baseline into the data management tool.
  - Changes to baselined content approved in a Change Board
  - Approved changes added to Product Backlog based on priority
**Sequence of events:**

- Develop draft in collaboration tool
- Perform peer review using the data management tool
- Run the test in the test lab, redline the STD
- Update the collaboration tool with redlines
- Update the requirements as needed
- Perform peer review with updates/redlines
- Run the updated test in the test lab
Configuration Control for STD (Cont.)

Boeing Defense Space & Security | Lean-Agile Software

- **Sequence of events:**
  - Finalize the STD
  - Populate the STD into the requirements management tool
  - Link the test descriptions to the requirements that are verified
  - Create the final STD word document from the requirements management tool and post baseline into the data management tool.
  - Changes to baselined content approved in a Change Board
  - Approved changes added to Product Backlog based on priority
## Agile Practices Drive LEAN Disciplines

<table>
<thead>
<tr>
<th>LEAN Disciplines</th>
<th>Agile Requirements Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Establish Clear Priorities</td>
<td>1. Product backlog is always prioritized; Team works on highest priority items first</td>
</tr>
<tr>
<td>2. Eliminate Bad Multitasking – Focus and Finish</td>
<td>2. Team is shielded from interruptions that cause bad multitasking</td>
</tr>
<tr>
<td>3. Limit the Release of Work in Process (WIP) to Deliver Earlier</td>
<td>3. Tasks are pulled from the iteration backlog one at a time to limit individual WIP</td>
</tr>
<tr>
<td>4. Prepare! Start → Finish</td>
<td>4. Requirements are not selected from the product backlog until everything needed is available</td>
</tr>
<tr>
<td>5. Use Checklists to Prevent Defects and Traveled Risk</td>
<td>5. Checklists and guides are used to prevent costly rework</td>
</tr>
<tr>
<td>6. Face into and Resolve Issues Quickly</td>
<td>6. Daily stand-up meetings force issues and risks to be identified and resolved quickly</td>
</tr>
<tr>
<td>7. Drive Daily Execution</td>
<td>7. Daily stand-up meetings drive team-based execution</td>
</tr>
</tbody>
</table>
# Acronyms and Abbreviations

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Docs</td>
<td>Documents</td>
</tr>
<tr>
<td>DOORS</td>
<td>Dynamic Object-Oriented Requirements System</td>
</tr>
<tr>
<td>EAR</td>
<td>Export Administration Regulations</td>
</tr>
<tr>
<td>FQT</td>
<td>Functional Qualification Test</td>
</tr>
<tr>
<td>h</td>
<td>Hours</td>
</tr>
<tr>
<td>HTML</td>
<td>Hypertext Markup Language</td>
</tr>
<tr>
<td>ID</td>
<td>Identification</td>
</tr>
<tr>
<td>INCOSE</td>
<td>International Council On Systems Engineering</td>
</tr>
<tr>
<td>IAW</td>
<td>In Accordance With</td>
</tr>
<tr>
<td>ITAR</td>
<td>International Traffic in Arms Regulation</td>
</tr>
<tr>
<td>Mgmt</td>
<td>Management</td>
</tr>
<tr>
<td>PRO</td>
<td>Boeing Procedure</td>
</tr>
<tr>
<td>Reqmts</td>
<td>Requirements</td>
</tr>
<tr>
<td>SRS</td>
<td>Software Requirement Specification</td>
</tr>
<tr>
<td>STD</td>
<td>Software Test Description</td>
</tr>
<tr>
<td>Sys Engrg</td>
<td>System Engineering</td>
</tr>
</tbody>
</table>
Phyllis R. Marbach is a Senior Software Manager in Boeing's Defense Space and Security (BDS). Marbach has over 32 years experience in aerospace programs including Satellites, chemical lasers, the International Space Station, and various propulsion systems. Currently she is a team lead with the Lean-Agile Software Services (LASS) for the BDS LASS Coaching Team, a Boeing Agile Software Process (BASP) Trainer and an active BASP Coach working with Unmanned Air Systems, Radio, and research programs. phyllis.r.marbach@boeing.com

Marbach holds a BS in Chemistry and Applied Mathematics from Centre College of Kentucky and an MS degree in Engineering from UCLA.
Abstract

An Integrated Toolset for Agile Systems Engineering

Requirements Analysis introduces an iterative approach to analyzing requirements and the integrated toolset that enables this analysis. The requirements analysis may occur in parallel and one iteration ahead of software development or it may occur after software exists to produce needed artifacts as in the case to be presented. An agile process for systems engineering to analyze requirements and provide to the agile software team will be introduced. The commercial-off-the-shelf tools that allow collaboration, standard formatting, logging, unique identification numbering, linking to defined tests, and formal document production, configuration control and actual program use will be presented. This end-to-end integrated toolset allows the sync-up of the requirements to test descriptions not only at each formal release, but daily as the requirements evolve during the iterations. This results in planned deliverable products at the end of each iteration to internal and external customers.