JAXA's IV&V Activity and Value Concept

JAXA IV&V team
Hiroki Umeda, Tsutomu Matsumoto, Naoko Okubo, Masa Katahira
okubo.naoko@jaxa.jp
2013/09/10
Contents

1: Background
   • Characteristics of JAXA’s IV&V Program
   • Change of IV&V NEEDS

2: Vision of NEW IV&V Program in JAXA
   • Overview of NEW IV&V Value Concept
   • Issues for NEW IV&V Value Concept

3: Detail of issues
   • Visualization
   • Optimization
     - IV&V Reference model

4: Conclusion
Brief history of IV&V activities in JAXA

Start of IV&V (over decades ago)

IV&V Program become widespread → NEEDS are changed

Apply to various projects

NOW
Characteristics of JAXA’s IV&V Program

- Beneficiaries pay IV&V costs (Projects have funds).
- IV&V is finished when a result is reported to a project team.
Examples of Project’s Request

- The Result of the Questionnaire to five projects, which used IV&V program in 2012

Q1: How do you use IV&V in your SW development pass?

- Criterion to decide to hold/pass Review
- Info. for Review
- Confirmation of V&V
- Guarantee of SW quality
- other

Q2: Is there a room for improvement for IV&V Finding lists?

- Yes, there is
- No, it’s enough

KEYWORDS

1. Confidence
   - Can move on to next milestone or not?

2. Guarantee
   - Outcome SW

3. Improvement
   - Show the results effectively (ex. Show the risk for operation)
Clarification of IV&V NEEDS

- NEEDS 1: Clear accountability for “Confidence”
- NEEDS 2: “Guarantee” the SW quality as a whole
- NEEDS 3: Show traceability between SW defects on orbit and operational risks.

**BEFORE**

- Figure out significant problems of software development.
- Understand verification attributes and scope of IV&V.
- Finish up with merely identifying the problems.

**NOW**

- Confidence
  - Gain a future advice and judgment stuff for development which can be learned from software defects.

- Guarantee
  - Know how much IV&V contributed to the high-reliable software together with V&V.

- Improvement
  - Understand how software problems influence on operations. Traceability between software defects and risk for operations is required.
Quality Visualization

Quality Improvement of Development Products

Identification of Technical Failure

Outcomes of IV&V activities

Prediction of Risks

Removing Problems

Visualization

Project Managers

Project Team members

Clear Accountability for confidence

SW Quality Improvement

Understand defects and its effect easily

Optimization

Use IV&V knowledge across projects.

Set verification attributes from a system-building/operational viewpoint.

Establish verification techniques different from developers.

• Store information of identified problems.

• Define verification attributes after identifying important functions and scenes in missions or operations.

• Improve IV&V Manual, make use of Formal Method, etc.

Analysis of Characteristics of the Target Products

Storage of Cost-benefit Data per Attribute/Method

2013/9/10

Project Milestone

Requirements Phase

Design Phase

Coding Phase

Test Phase

Expectations for IV&V Activities

Project Team members

IV&V Team

Overview of New IV&V Value Concept

2013/9/10
Issues of New IV&V Value Concept

- Visualization: How to present the result effectively to Project?
- Optimization: How to find defects efficiently?

**IV&V Techniques**
- Techniques to fix a scope (Software Scope)
- Techniques to fix criteria (Attribute)
- Techniques to find defects

**Base of Competitive IV&V**
- Collect data including finding approaches.
- Detect logical inconsistency.
- Optimization

**IV&V Value**
- Prediction
- Removal

**Project Milestone**
- System Requirement Review (SRR)
- System Definition Review (SDR)
- Preliminary Design Review (PDR)
- Critical Design Review (CDR)
- Development Completion Review

**Conduct IV&V**
- Visualization
- Optimization

2013/9/10
Detail of Optimization

- Effective way to collect and use Data (Left part of Fig.)
- Suitable method to find defects (Right part of Fig.)

**Project Milestone**
- System Requirement Review (SRR)
- System Definition Review (SDR)
- Preliminary Design Review (PDR)
- Critical Design Review (CDR)
- Development Completion Review

**IV&V Value**
- Prediction
- Removal

**Conduct IV&V**

**IV&V Techniques**
- Techniques to determine a scope (Software Scope)
- Techniques to determine criteria (Attribute)

**IV&V Reference Model**
- Method Optimization
- Detect logical inconsistency.
- Collect data including finding approaches.

**Optimization**
- Data
- IV&V Reference Model
Attempt of IV&V Reference model

IV&V Reference Model:
Normalized Model/Method to clarify defects and risks
1: Stable IV&V Quality
2: Comparable with other SW, or projects
3: Improvement of persuasion

Issues for defects/risks clarification

Before

How to describe defects/risks depends on individual IV&V engineers.

Hard to ...
✓ understand promptly
✓ compare with others
✓ convince the projects

IV&V Reference Model

Describes the defect and risk using reference data

Defect (what/where)
Reasons of the defect (why)
Feature, and trend of SW

Reference Data
Factors of IV&V Reference Model

Reason of the Defects (Includes the interpretation)

Reason of the Defects

- Person: Experience of development, etc.
- Organization: Org. of development, Org. of management
- Process: Change of specification

Feature & trend (Feature which can use to check the similar defects)

- System feature: HW, Algorithm, Exception handling, etc.
- Project feature: Development term, Development cost, Organization, Change of specification
- SW Development feature: Experiment of developer, Complexity of developing process, Method of development, Complexity of Design
- Design Feature: Un-Synchronize, Design, cannot test

Classification Model for Defects

Defects
- Happen at Request
- Happen at Design
- Happen at Code
- Happen at Test

Bug
- Happen at SW
- Happen at System

Process
- Change of specification

System feature
- HW
- Algorithm
- Exception handling

Project feature
- Development term
- Development cost
- Organization
- Change of specification

SW Feature
- Re-use
- Environment for the Development
- Environment for the Testing

SW Type
- Type: Navigation and Control, Data Handling, Mission Sensor, Ground system, etc.
### Attempt of Defect Classification Model

<table>
<thead>
<tr>
<th>No.</th>
<th>Types of Defects</th>
<th>Details</th>
<th>Error</th>
<th>lack</th>
<th>Unclear-ness</th>
<th>Inconsist-ency</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Functionality Definition</td>
<td>Errors related to definition of system processing. Lack of definition of the subject function etc.</td>
<td>○</td>
<td>○</td>
<td>–</td>
<td>–</td>
<td>• Lack of functionality</td>
</tr>
<tr>
<td>2</td>
<td>Data Definition</td>
<td>Errors related to data definition. Inconsistent common commands, lack of definition of the subject data, unclear specification about debug commands, etc.</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>• Lack of data definition</td>
</tr>
<tr>
<td>3</td>
<td>Processing Logic</td>
<td>Errors related to processing logic/order. Wrong order of processing main/sub tasks, wrong setting with no loop, etc.</td>
<td>○</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Mode Transition</td>
<td>Errors related to system mode transition. Wrong space where the mode transit to, missing definition of transition conditions, etc.</td>
<td>○</td>
<td>○</td>
<td>–</td>
<td>–</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Exception Handling</td>
<td>Errors related to exception or error handling in system malfunction. Unclear behaviors when receiving unintended data, missing definition of error handling, etc.</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>–</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Timing (Synchronization/Exclusive Processing)</td>
<td>Errors related to timing between functions or between hardware and software. Inconsistency of sending and receiving timing etc.</td>
<td>○</td>
<td>–</td>
<td>–</td>
<td>○</td>
<td></td>
</tr>
</tbody>
</table>
# Attempt of Defect Classification Model

<table>
<thead>
<tr>
<th>No.</th>
<th>Types of Defects</th>
<th>Details</th>
<th>Error</th>
<th>lack</th>
<th>Unclear-ness</th>
<th>Inconsist-ency</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Boundary Value</td>
<td>Errors related to boundary values in data. Validity of a scope of memory clearance, etc.</td>
<td>○</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Constant Value/Initial Value</td>
<td>Errors related of constant values and initial values of variables. Mismatch of constant values between specifications, missing initial values of a program, etc.</td>
<td>○</td>
<td>—</td>
<td>—</td>
<td>○</td>
<td>*Error in input value of data</td>
</tr>
<tr>
<td>9</td>
<td>Ambiguous Description</td>
<td>Errors of description in a subject specification which are not related to a system itself. Clerical errors, wrong page/revision numbers, etc.</td>
<td>○</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Integrity with Hardware</td>
<td>Errors related not just to software but to integrity with onboard hardware. Mismatch of specifications between software and hardware, lack of control when powers become excessive, etc.</td>
<td>○</td>
<td>○</td>
<td>—</td>
<td>○</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Inconsistency between Specifications</td>
<td>Errors related to consistency between a subject document and an upper specification or reference. Items defined in the upper specification are not correctly represented in the subject document, definition is missed in the upper one, etc.</td>
<td>○</td>
<td>○</td>
<td>—</td>
<td>○</td>
<td></td>
</tr>
</tbody>
</table>
Conclusion

• Change of IV&V NEEDS
• Proposed New IV&V Value Concept
• Issues for New IV&V Value Concept
  - Visualization
  - Optimization (Attempt of IV&V Reference Model)

• Future work
  - Goal of IV&V is to improve operability (usability) by checking consistency between SW design and operation.
    → Analysis with an operational view and define outputs for them
  - Make IV&V training course. Also, define required skills needed as IV&V engineer.
<table>
<thead>
<tr>
<th>No.</th>
<th>Types of Defects</th>
<th>Details</th>
<th>Error</th>
<th>lack</th>
<th>Unclear-ness</th>
<th>Inconsist-ency</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Functionality Definition</td>
<td>Errors related to definition of system processing. Lack of definition of the subject function etc.</td>
<td>○</td>
<td>○</td>
<td>—</td>
<td>—</td>
<td>• Lack of functionality</td>
</tr>
<tr>
<td>2</td>
<td>Data Definition</td>
<td>Errors related to data definition. Inconsistent common commands, lack of definition of the subject data, unclear specification about debug commands, etc.</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>• Lack of data definition</td>
</tr>
<tr>
<td>3</td>
<td>Processing Logic</td>
<td>Errors related to processing logic/order. Wrong order of processing main/sub tasks, wrong setting with no loop, etc.</td>
<td>○</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Mode Transition</td>
<td>Errors related to system mode transition. Wrong space where the mode transit to, missing definition of transition conditions, etc.</td>
<td>○</td>
<td>○</td>
<td>—</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Exception Handling</td>
<td>Errors related to exception or error handling in system malfunction. Unclear behaviors when receiving unintended data, missing definition of error handling, etc.</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Timing (Synchronization/Exclusive Processing)</td>
<td>Errors related to timing between functions or between hardware and software. Inconsistency of sending and receiving timing etc.</td>
<td>○</td>
<td>—</td>
<td>—</td>
<td>○</td>
<td></td>
</tr>
<tr>
<td>No.</td>
<td>Types of Defects</td>
<td>Details</td>
<td>Error</td>
<td>Lack</td>
<td>Unclear-ness</td>
<td>Inconsistentency</td>
<td>Examples</td>
</tr>
<tr>
<td>-----</td>
<td>--------------------------</td>
<td>-------------------------------------------------------------------------</td>
<td>-------</td>
<td>------</td>
<td>--------------</td>
<td>-----------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>7</td>
<td>Boundary Value</td>
<td>Errors related to boundary values in data. Validity of a scope of memory clearance, etc.</td>
<td>○</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Constant Value/Initial Value</td>
<td>Errors related of constant values and initial values of variables. Mismatch of constant values between specifications, missing initial values of a program, etc.</td>
<td>○</td>
<td>—</td>
<td>—</td>
<td>○</td>
<td>• Error in input value of data</td>
</tr>
<tr>
<td>9</td>
<td>Ambiguous Description</td>
<td>Errors of description in a subject specification which are not related to a system itself. Clerical errors, wrong page/revision numbers, etc.</td>
<td>○</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Integrity with Hardware</td>
<td>Errors related not just to software but to integrity with onboard hardware. Mismatch of specifications between software and hardware, lack of control when powers become excessive, etc.</td>
<td>○</td>
<td>○</td>
<td>—</td>
<td>○</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Inconsistency between Specifications</td>
<td>Errors related to consistency between a subject document and an upper specification or reference. Items defined in the upper specification are not correctly represented in the subject document, definition is missed in the upper one, etc.</td>
<td>○</td>
<td>○</td>
<td>—</td>
<td>○</td>
<td></td>
</tr>
</tbody>
</table>
IV&V Activities

Set an attribute of verification (Integrity, Completeness, etc.) for each development phase based on IV&V Manual.

<table>
<thead>
<tr>
<th>Techniques</th>
<th>Requirement</th>
<th>Design</th>
<th>Coding</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>System/Safety Analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Check List</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Model Analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Model Checking</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Static Code Analysis, Structure Analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Independent Testing</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Case 1**
- Target: Mode Transition in Software Requirements
- Attribute: Verification of Completeness
- Method: Model Checking

**An Example of Identified Problems**
Mode Transition could stop because some conditions executing safety mode transfer function were not sufficient.

**Case 1**
- Target: Asynchronous Processing (Interrupt) in Source Code
- Attribute: Verification of Integrity
- Method: Static Analysis

**An Example of Identified Problems**
Some processing could conflict with others when doing interrupt mask in Source Code, because some function calls were missing.
IV&V Value (1)

**Project Milestone**
- System Requirement Review (SRR)
- System Definition Review (SDR)
- Preliminary Design Review (PDR)
- Critical Design Review (CDR)
- Development Completion Review

**IV&V Value**
- Prediction
- Removal

**Conduct IV&V**

**IV&V Techniques**
- Techniques to fix a scope (Software Scope)
- Techniques to fix criteria (Attribute)

**Data**
- Collect data including finding approaches.
- Detect logical inconsistency.

**Method Optimization**

**Viewpoints of System/Operation**
- V&V (System Testing)
- V&V (Software Testing)
- IV&V (Independent Verification)
- Horizontal Development of Bugs
- Common Factors across Projects

**Position of IV&V (Verification)**
- Specific Factors per Project
- Viewpoint of Software
### 3 Issues to Achieve New IV&V Value Concept

<table>
<thead>
<tr>
<th>Keywords</th>
<th>Policy</th>
<th>Supplement</th>
</tr>
</thead>
</table>
| **Visualization** | • Explanation of cost-effectiveness  
• Risk-based analysis | • Prepare convincing explanations and indicators for projects (developers) and end customers.  
→ A new usage of outcomes of IV&V.  
Improve stakeholder satisfaction.  
• Appeal and transfer of techniques to other industries.  
Provide the activity, techniques, data and training materials. |
| **Differentiation** | • Reinforcement of unique viewpoints  
• Utilization of actual data | • Focus not on the number of methods but on their quality.  
• Apply methods to collect data.  
• Collected data of defects, bugs and findings of each software with background information such as factors are available for IV&V. |
| **Automation** | • Standardization of methods and processes  
• Utilization of tools | • Processes and outputs are standardized.  
→ Verification can be done by more than one engineer. |