Tailoring and Combining the CMMI-ACQ and Quality Models to Improve the Military’s Requirements Process

SEPG 2008
Tampa Florida
Tampa Convention Center
March 17-20

Ira Monarch, Dennis Goldenson, Peter Capell and Jim Wessel

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

January 2008
Improving the Military’s Requirements Process

The requirements process in general

- A **top-5** issue in all systems/software engineering studies.
- **Requirements stakeholders’ different value propositions** a primary reason

=> The military’s approach to requirements development in policy & practice

- **Sponsorship** for requirements **shifts** throughout the JCIDS process
- **Users** are joint developers of requirements and can play the role of sponsor
- JCIDS KPPs – partial fit with other **quality models & usability** marginalized
- Yet **usability** can be at issue in requirements development & testing

Content analysis of CMMI-ACQ re JCIDS, quality models & problem reports

- Content Analysis **Techniques**
- **Findings**: re Customers & Users, Organization, Usability & Other Quality Attributes
- **Building a Concept Space for Defining Quality Attributes in Practical Contexts**

Using **Meta-Modeling** to Tailor, Combine, Apply & Improve Policy & Models
The Military’s Approach to Requirements: JCIDS

**JCIDS is the Joint Capabilities Integrated Development System**

**JCIDS a Joint Chiefs** approach to requirements development to

- “stop paying for the same things twice” or even multiple times:

- **Negotiate & reconcile differences in value proposition** among:
  - Combatant Commanders
  - Science and Technology Representatives
  - Combat Developers
  - Material Developers
  - Testers and Evaluators

- focus on **Key Performance Parameters** (KPPs)
  - Overlap with other **quality attribute** models
  - KPPs supported by Key System Attributes (KSAs)
  - Other Attributes of importance in particular contexts
Quality Attribute Coverage Discussed in JCIDS

Examples of Quality Attributes in JCIDS are:

- **Survivability** KPPs like **speed**, **maneuverability**, **detectability**, and **countermeasures**, reducing likelihood of being engaged by hostile fire.

- **Operational Suitability including Sustainment** KPPs such as Materiel **Availability** and a supporting KSAs, Materiel **Reliability, Maintainability, supportability, safety**.

- **Net-Ready KPPs** like **interoperability** that are to be used in Information Support Plans to identify support required from outside a program.

- **KPPs covering characteristics of the future force**: being knowledge empowered, networked, expeditionary, **Adaptable/tailorable (Adaptability, Changeability, Modifiability)**, enduring/persistent, **Accuracy**, **lethality**, and precise, fast, resilient/agile (**Efficiency/Performance**).

- **Information Assurance KPPs (Security)** that protect availability, integrity, authentication, confidentiality, and non-repudiation.

Attributes outside or marginal to KPPs, KSAs & value determiners, e.g.,

- **Usability**, or what JCIDS calls Human Systems Integration (HSI).

*Named in other quality models – ones not named are military specific*
## ISO/IEC 9126 – Software Product Quality

<table>
<thead>
<tr>
<th>Quality Characteristics</th>
<th>Sub-characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Functionality</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Suitability</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Accuracy</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Interoperability</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Security</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Compliance</strong></td>
</tr>
<tr>
<td>• Reliability</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Maturity</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Fault tolerance</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Recoverability</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Compliance</strong></td>
</tr>
<tr>
<td>• Usability</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Understandability</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Learnability</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Operability</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Comp</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Attractiveness</strong></td>
</tr>
<tr>
<td>• Efficiency</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Time behavior</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Resource utilization</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Compliance</strong></td>
</tr>
<tr>
<td>• Maintainability</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Analyzeability</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Changeability</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Stability</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Testability</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Compliance</strong></td>
</tr>
<tr>
<td>• Portability</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Adaptability</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Installability</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Co-existence</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Replaceability</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Comp</strong></td>
</tr>
</tbody>
</table>
# Quality Attribute Scenarios: For SW Architecture*

<table>
<thead>
<tr>
<th>Source of Stimulus</th>
<th>Stimulus</th>
<th>Artifact</th>
<th>Environment</th>
<th>Response</th>
<th>Response Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Availability</strong> (Reliability)</td>
<td>External to System</td>
<td>Unanticipated Message</td>
<td>Process</td>
<td>Normal Operation</td>
<td>Inform Operator</td>
</tr>
<tr>
<td><strong>Modifiability</strong> (Maintainability)</td>
<td>Developer</td>
<td>Change UI</td>
<td>Code</td>
<td>Design Time</td>
<td>Modification &amp; No side effects</td>
</tr>
<tr>
<td><strong>Performance</strong> (Time behavior)</td>
<td>Users</td>
<td>Initiate Transactions</td>
<td>System</td>
<td>Normal operations</td>
<td>Transactions processed</td>
</tr>
<tr>
<td><strong>Security</strong> (Security)</td>
<td>Correctly identified individual</td>
<td>Tries to modify information</td>
<td>Data within system</td>
<td>Normal operation</td>
<td>System maintains audit trail</td>
</tr>
<tr>
<td><strong>Testability</strong> (Testability)</td>
<td>Unit Tester</td>
<td>Performs Unit Test</td>
<td>System component</td>
<td>Component completion</td>
<td>Component behavior observed</td>
</tr>
<tr>
<td><strong>Usability</strong> (Usability)</td>
<td>Users</td>
<td>Minimize impact of errors</td>
<td>System</td>
<td>Runtime</td>
<td>Cancel current operation</td>
</tr>
</tbody>
</table>

* SEI SW architecture quality attribute – ISO/IEC 9126-1 terminology in parenthesis – sometimes as sub-characteristics. ISO covers additional sub-characteristics that SEI SW architecture does not.
Usability in Requirements Development & Testing

Analysis of requirements documents & problem reports and interviewing at several military sites identified problems downstream that could be mitigated by documented considerations upstream.

One of the patterns identified suggests **usability** is not being considered, articulated & quantified during requirements development **because**

- functional criteria are better understood
- usefulness of a system is perceived as so important that users will learn how to use it in spite of usability issues
- operability issues can be fixed as they arise or deferred according to urgency.

**Other reasons:** a requirements developer prior to testing would have to

- know all the situations where a usability scenario applies
- formulate scenarios, at levels below the sub-characteristics of ISO 9126, re
  - potential data entry error and feedback
  - display of misleading alerts and warnings
  - understandability of display & operability of a PDA reusing software from a desktop computer
  - a maintainability feature (e.g., a stack dump) causing a disruption that could interfere with usability.
Focusing Content Analysis 1: Issues Combining JCIDS Policy, Quality Models and Practice

Combatants, their representatives, acquirers, maintainers and testers can be *full-fledged participants in requirements development*.

— All must know how to participate in design reasoning
— All must collaborate in the evaluation and test of Systems of Systems.

Requirements processes can *shift responsibility* for specification of

- user/customer requirements (operational capabilities)
- contractual requirements (acquisition’s translation for developer understanding)
- system requirements (developer’s/sustainer’s translation of contractual requirements)
- system of systems requirements.

*Quality attributes can be specified early*, starting with *usability*.
Focusing Content Analysis 2: *Disconnects between Practice, Policy, Quality Models & the CMMI-ACQ*

Analysis guided by disconnects between:

- **On the one hand,** issues with respect to JCIDS & current military practice
  - multiple organizations have to share processes
  - users and customers can be JCIDS sponsors
  - quality attributes in the form of KPPs are essential

- **On the other hand,** elements of CMMI-ACQ, i.e.:
  - Single stable acquisition organizations are responsible for both customer and contractual requirements
  - Customers and end users are the sources for requirements
    - acquirers take full responsibility for specifying requirements
  - Quality attributes are factors to consider when expressing requirements, but are not essential.
What is Content Analysis?

Content Analysis: a well-understood methodology

- to study documented communication
- using systematic, replicable techniques
- compressing many words of text into a few categories via explicit rules of coding.*
- that predicted bombing of London by the Germans by analyzing Goebbels’ speeches

* Authors consulted are Berelson, 1952; Krippendorff, 1980; and Weber, 1990
Content Analysis Techniques* 1

Automated text analysis

• Tools identify recurring concepts & themes
• Employs computational algorithms using Bayesian conditional probabilities
• Similar to factor analysis

Semantic classification, inference & validation

• Initially by analysts using the text analysis tool who:
  — classify automatically generated themes semantically
  — infer Quality Attributes (or other conceptual content)
• Iterative corroboration & enhancement with domain experts
  — Fully engaged to identify their own most problematic areas.

* A number of tools have been used, CAIR (developed at the SEI), Semio Text, TextSmart, Lexquest Mine and Leximancer. This is not an exhaustive list and SEI does not rank them in any way.
Content Analysis Techniques 2

Progression from Text to Concept to Theme *

- **Text blocks (usually several sentences) are where concepts co-occur**
- **Concepts** are *synonym lists* of strongly related co-occurring *terms*
- **Themes** are *collections* of co-occurring *concepts*
  - more strongly related to each other than to concepts in other themes
  - automatically named by the concept most strongly related to other concepts in the theme.

Themes containing concepts are represented spatially as Venn diagrams

- concepts labeling dots are in themes represented as circles
- dots can be linked by lines whose brightness represents frequency of co-occurrence
- dots can appear in the overlap of two (or more) circles
- circle size based on distribution of concepts included in the circle
  - brightness represents interconnectedness of concepts in the circle

* The following describes the approach used with Leximancer.
Tailoring and Combining the CMMI-ACQ and Quality Models to Improve the Military's Requirements Process,
SEPG | 2008
© 2008 Carnegie Mellon University

CMMI-ACQ Thematic Structure
The most frequent CMMI-ACQ concepts are listed at the left.

The absolute count is the number of text blocks where a given concept occurs.

The relative count is the percentage of text blocks where it occurs.

Not surprisingly for a process model, conceptual traces of **process** are found in all of the CMMI-ACQ text blocks.

**Project** and **organization** are the next most significant thematic concepts.

These are followed by **product** and then **supplier** all of which are important to the points that follow.

All are in the top 10% of concepts appearing in concept maps that follow.
The concept map shows the **top 10%** of the most frequent and connected concepts.

Like all CMM models, **process**, **product** and **project** all come under the purview of a **single organization**.

The model does not cover **changing sponsorship** and **multiple organizational** perspectives needed for **requirements** practice to be in accord with JCIDS.

With respect to **requirements**, **organization** is most frequently focused on **agreement** with the **supplier** – not **customer** or **users**.
The **customer** concept appears when the top 36 percent of concepts is shown. **Customer** appears only as a concept in the overlap between the **supplier** and **product** themes and is relatively frequently linked with **Requirements**.

At this point neither **users** nor **validation** yet appear as a basis for validating **Requirements**.
**Users** appears only as a concept in the **product** theme when 72% of concepts shown;

**Validation** appears only as a concept in the **interfaces** theme when 67% are shown.

Validation is frequently coupled to **product** (61% of text blocks it co-occurs in), **requirements** (30%) and **supplier** (22%), less with **customer** (10%) and even less with **users** (4%).

**Customer** and **users** are in a secondary position in the map with respect to **supplier**.
Customer and Users are not Themes

Unlike the supplier and the (acquisition) organization, neither customer nor users are concepts that are also themes.

- CMMI-ACQ could be articulated in such a way that customer and/or users would be just as well connected to other concepts as is supplier… thereby becoming a theme or themes
- The acquisition organization should give equal consideration to both.

But the CMMI-ACQ is already 400+ pages and this would make it bigger.

- As in the case of dealing with shifting sponsorship and multiple organizations, perhaps another Model or Guide can cover this.
- No one model can satisfy all perspectives.

A meta-model may be needed showing where gaps in one model may be covered by another.
The coverage of “quality attributes” in the model is quite minimal – attribute appears at 81%:

- Defect measures are cited as examples of quality attributes in Quantitative Project Management, but “quality attribute” has a different meaning in this context than in standard quality models.

- A characterization of quality attribute is expressed as a factor to consider when formulating customer requirements, but is not repeated again.

It may be worthwhile for the model to cross-reference quality models as a bridge between product and process quality.
Applying Content Analysis for Improvement of Requirements Development

By identifying misalignments in

• policy
• requirements specifications
• problem and field reports

**content analysis may facilitate identification & resolution of problems**

— to support early consideration, articulation and operationalization of usability and other quality attributes

Content analysis may provide sufficient basis to sub-categorize *usability* several levels down where it might be characterized in scenarios for a given context of use.

The range of acceptable or desired responses, whether

• Operational
• System
• Software
• System of Systems

can be quantified and used as a basis for tradeoff analysis and prioritization.
Applying Content Analysis in Improvement of Requirements Development

Content analysis can map out a conceptual space that can be used to formulate new quality attributes specifically applicable to a context of use

- Providing a richer bases for validation of architectures and products
  - whether at the operational, systems, subsystems or systems of systems levels.

Formulation of such context specific quality attributes might benefit from additional formalization and computational support enabling

- cross-referencing to higher level quality and process models
- indexing to requirements specifications and problem reports.
Meta-Modeling to Tailor & Combine Models for Use in Practice

By identifying misalignments between

• Policy and Practice on the one hand

and

• Process and Quality Models on the other,

content analysis can facilitate tailoring and combining process and quality models that cover each other’s gaps in different contexts of use.

This is tantamount to creating a meta-model that cross-references

• quality models like those described in ISO 9126 for software architectures
• process models like CMMI-ACQ
• guides like the current draft of the System of Systems System Engineering Guide.
Thank you for your attention!

For further information, please contact:

Ira Monarch
iam@sei.cmu.edu
1.412.268.7070

Dennis Goldenson
dg@sei.cmu.edu
1.412.268.8506