Carnegie Mellon
Software Engineering Institute
Pittsburgh, PA 15213-3890

A Comparison of Requirements Specification Methods from a Software Architecture Perspective

Len Bass
John Bergey
Paul Clements
Paulo Merson
Ipek Ozkaya
Raghvinder Sangwan

Sponsored by the U.S. Department of Defense

Carnegie Mellon
Software Engineering Institute

Motivation

The fact is

quality attributes shape architectural approaches

We have observed

architecturally significant requirements are **not routinely specified** in a manner that makes them **useful** to an architect

We ask

how well do the existing requirement specification methods **assist specifying** quality attribute requirements for an architect's use?

Our eventual goal is

to give guidance for **transforming** higher management level **business analysis** to **architecturally significant requirements**

Carnegie Mellon
Software Engineering Institute

Outline

Motivation

Evaluation criteria

Review of selected methods

- Natural language requirements: “shall” and “will”
- Use case analysis
- Quality attribute workshop (QAW)
- Global analysis
- O’Brien’s approach after Fergus O’Brien

Comparison and Conclusions

Carnegie Mellon
Software Engineering Institute

Evaluation Criteria

1. **Quality attribute expressiveness**
   - Does the expression form allow for the specification of any quality attribute and context, discouraging too vague requirements?
2. **Ease of organizing quality attribute requirements**
   - How are easy searching for and organization based on a variety of different criteria facilitated?
3. **Traceability**
   - Is it possible to track what business goal a requirement supports and how the requirement is satisfied by the architecture?
4. **Checking for completeness and consistency**
   - What support is there for fine and coarse grained expression?
   - What support is there for checking for consistency?

Carnegie Mellon
Software Engineering Institute

Evaluation Criteria 2

5. **Support for testing**
   - How well suited are the requirements generated by the method for the testing process?
6. **Tooling**
   - What kind of tool support exists for the method?
7. **Support for variability**
   - How are eliciting and expressing variability requirements for a collection of systems supported?
8. **Skill level necessary to carry out the method**
   - What special skills should those carrying out the method possess?
9. **Support for prioritizing requirements**
   - Is there support for prioritizing requirements in respect to different attributes?

Carnegie Mellon
Software Engineering Institute

Outline

Motivation

Evaluation criteria

Review of methods

- **Natural language requirements: “shall” and “will”**
- **Use case analysis**
- **Quality attribute workshop (QAW)**
- **Global analysis**
- **O’Brien’s approach after Fergus O’Brien**

Comparison and Conclusions

Carnegie Mellon
Software Engineering Institute

Methods: Shall and will

- Natural language specification with a general form

<entity> *shall* (or *will*) <textual description of specific requirement>

  - *Shall* traditionally indicates the requirement is mandatory
  - *Will* is used to express a declaration of purpose
- Often results in a disparate set of requirements that correspond to a collection of “point” requirements
- Used mostly in the U.S. DoD, but commercial organizations use the method as well
  - following organization specific processes
- Both great power and problems
  - due to natural language-based specification

Methods: Shall and will 2

	Shall and will	Use case analysis	QAW	Global Analysis	O'Brien's approach
QA Expressiveness	Highly expressive				
Ease of organization	Easy to organize				
Traceability	Easy to trace				
Completeness and consistency	Difficult to check				
Support for testing	Low support				
Tooling	Tools available				
Variability	Moderate support				
Skill requirements	High skill level				
Support for prioritization	Low support				

Carnegie Mellon
Software Engineering Institute

Methods: Use Case Analysis

- Primarily looks at organizing how external entities interact with the system, hence tends to give higher priority to eliciting functional requirements of a system
- Focuses on defining the system boundary using actors and their goals
  - Find the system actors
  - Find the use cases
  - Define the sequences of actions
  - Identify scenarios
  - Structure the use cases
  - Refactor
- Quality attributes may appear in the resulting use cases and scenarios, but are traditionally captured in a supplemental specification

Methods: Use Case Analysis 2

	Shall and will	Use case analysis	QAW	Global Analysis	O'Brien's approach
QA Expressiveness	Highly expressive	Low expressiveness			
Ease of organization	Easy to organize	Hard to organize			
Traceability	Easy to trace	Of medium difficulty			
Completeness and consistency	Difficult to check	Difficult to check			
Support for testing	Low support	Low support			
Tooling	Tools available	Tools available			
Variability	Moderate support	Little support			
Skill requirements	High skill level	High skill level			
Support for prioritization	Low support	Low support			

Carnegie Mellon
Software Engineering Institute

Methods: QAW

- Advocates that
  - Concrete quality attribute requirements can be described in the form of quality attribute scenarios
  - Stakeholders are the best carriers of the different perspectives manifested in quality attribute requirements

- Results in prioritized set of quality attribute requirements that are candidates for architectural drivers

1. QAW presentation and introductions
2. Business and mission presentation
3. Architectural plan presentation
4. Identification of architectural drivers
5. Scenario brainstorming
6. Scenario consolidation
7. Scenario prioritization
8. Scenario refinement

Carnegie Mellon
Software Engineering Institute

Methods: QAW 2

	Shall and will	Use case analysis	QAW	Global Analysis	O'Brien's approach
QA Expressiveness	Highly expressive	Low expressiveness	**Highly expressive**		
Ease of organization	Easy to organize	Hard to organize	**Of medium difficulty**		
Traceability	Easy to trace	Of medium difficulty	**Of medium difficulty**		
Completeness and consistency	Difficult to check	Difficult to check	**Difficult to check**		
Support for testing	Low support	Low support	**High support**		
Tooling	Tools available	Tools available	**No tools available**		
Variability	Moderate support	Little support	**High support**		
Skill requirements	High skill level	High skill level	**High skill level**		
Support for prioritization	Low support	Low support	**Moderate support**		

Carnegie Mellon
Software Engineering Institute

Methods: Global Analysis

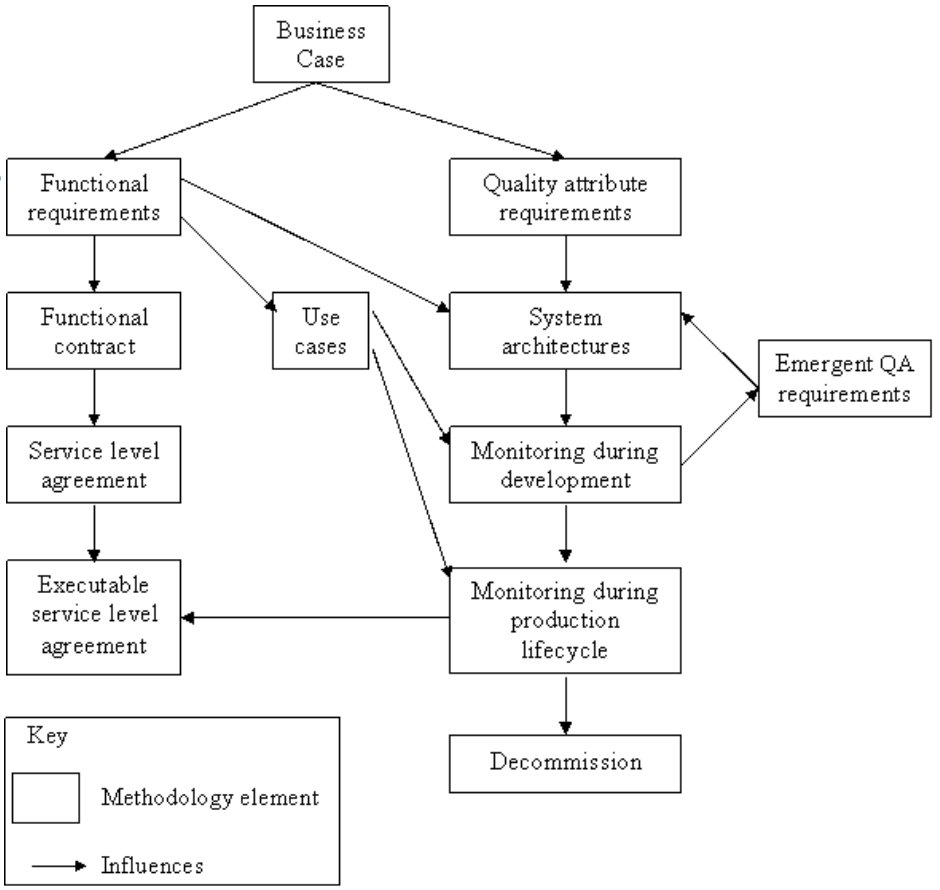
- Focuses on identifying and analyzing the factors that have global influence on the architecture
- Provides guidance for a classification scheme for factors
  - Technological; e.g. software technology, architecture technology, standards
  - Organizational; e.g. development schedule, budget
  - Product factors; e.g. functional features, user interface, performance
- Meant to complement risk and requirements analysis, which can be performed using other techniques
  - Starts before architectural views are defined and continues through out the development process
- Results in issue cards with a list of influencing factors and a discussion of strategies to address the issue

Methods: Global Analysis 2

	Shall and will	Use case analysis	QAW	Global Analysis	O'Brien's approach
QA Expressiveness	Highly expressive	Low expressiveness	Highly expressive	**Moderate expressiveness**	
Ease of organization	Easy to organize	Hard to organize	Of medium difficulty	**Easy to organize**	
Traceability	Easy to trace	Of medium difficulty	Of medium difficulty	**Of medium difficulty**	
Completeness and consistency	Difficult to check	Difficult to check	Difficult to check	**Difficult to check**	
Support for testing	Low support	Low support	High support	**Moderate support**	
Tooling	Tools available	Tools available	No tools available	**No tools available**	
Variability	Moderate support	Little support	High support	**Little support**	
Skill requirements	High skill level	High skill level	High skill level	**High skill level**	
Support for prioritization	Low support	Low support	Moderate support	**Moderate support**	

Methods: O'Brien Method

- Provides guidance for linking architectural decisions to measurable quality attributes that flow from explicitly capturing business goals
- Aims to capture deriving quality attributes from the business case
  - e.g. longevity: product lifetime
- Advocates defining and monitoring for *visibility,*
  - measure of design error for observable outcome



Adapted from:
O'Brien, F. *The Engineering of Software Quality*, Pearson SprintPrint, 2004

Methods: O'Brien Method 2

	Shall and will	Use case analysis	QAW	Global Analysis	O'Brien's approach
QA Expressiveness	Highly expressive	Low expressiveness	Highly expressive	Moderate expressiveness	**Low expressiveness**
Ease of organization	Easy to organize	Hard to organize	Of medium difficulty	Easy to organize	**Of medium difficulty**
Traceability	Easy to trace	Of medium difficulty	Of medium difficulty	Of medium difficulty	**Easy to trace**
Completeness and consistency	Difficult to check	Difficult to check	Difficult to check	Difficult to check	**Some help in checking**
Support for testing	Low support	Low support	High support	Moderate support	**High support**
Tooling	Tools available	Tools available	No tools available	No tools available	**No tools available**
Variability	Moderate support	Little support	High support	Little support	**No support**
Skill requirements	High skill level	High skill level	High skill level	High skill level	**High skill level**
Support for prioritization	Low support	Low support	Moderate support	Moderate support	**Moderate support**

Carnegie Mellon
Software Engineering Institute

Outline

Motivation

Evaluation criteria

Review of selected methods

- Natural language requirements: “shall” and “will”
- Use case analysis
- Quality attribute workshop (QAW)
- Global analysis
- O’Brien’s approach after Fergus O’Brien

Comparison and Conclusions

Comparison and Conclusions

	Shall and will	Use case analysis	QAW	Global Analysis	O'Brien's approach
QA Expressiveness	Highly expressive	Low expressiveness	Highly expressive	Moderate expressiveness	Low expressiveness
Ease of organization	Easy to organize	Hard to organize	Of medium difficulty	Easy to organize	Of medium difficulty
Traceability	Easy to trace	Of medium difficulty	Of medium difficulty	Of medium difficulty	Easy to trace
Completeness and consistency	Difficult to check	Difficult to check	Difficult to check	Difficult to check	Some help in checking
Support for testing	Low support	Low support	High support	Moderate support	High support
Tooling	Tools available	Tools available	No tools available	No tools available	No tools available
Variability	Moderate support	Little support	High support	Little support	No support
Skill requirements	High skill level	High skill level	High skill level	High skill level	High skill level
Support for prioritization	Low support	Low support	Moderate support	Moderate support	Moderate support

Comparison and Conclusions 2

	Shall and will	Use case analysis	QAW	Global Analysis	O'Brien's approach
QA Expressiveness	Highly expressive	Low expressiveness	Highly expressive	Moderate expressiveness	Low expressiveness
Ease of organization	Easy to organize	Hard to organize	Of medium difficulty	Easy to organize	Of medium difficulty
Traceability	Easy to trace	Of medium difficulty	Of medium difficulty	Of medium difficulty	Easy to trace
Completeness and consistency	Difficult to check	Difficult to check	Difficult to check	Difficult to check	Some help in checking
Support for testing	Low support	Low support	High support	Moderate support	High support
Tooling	Tools available	Tools available	No tools available	No tools available	No tools available
Variability	Moderate support	Little support	High support	Little support	No support
Skill requirements	High skill level	High skill level	High skill level	High skill level	High skill level
Support for prioritization	Low support	Low support	Moderate support	Moderate support	Moderate support

Comparison and Conclusions 3

	Shall and will	Use case analysis	QAW	Global Analysis	O'Brien's approach
QA Expressiveness	Highly expressive	Low expressiveness	Highly expressive	Moderate expressiveness	Low expressiveness
Ease of organization	Easy to organize	Hard to organize	Of medium difficulty	Easy to organize	Of medium difficulty
Traceability	Easy to trace	Of medium difficulty	Of medium difficulty	Of medium difficulty	Easy to trace
Completeness and consistency	Difficult to check	Difficult to check	Difficult to check	Difficult to check	Some help in checking
Support for testing	Low support	Low support	High support	Moderate support	High support
Tooling	Tools available	Tools available	No tools available	No tools available	No tools available
Variability	Moderate support	Little support	High support	Little support	No support
Skill requirements	High skill level	High skill level	High skill level	High skill level	High skill level
Support for prioritization	Low support	Low support	Moderate support	Moderate support	Moderate support

Carnegie Mellon
Software Engineering Institute

Comparison and Conclusions 4

- Each method has its own strengths and weaknesses
  - O'Brien's approach is the only method that explicitly starts with business goals for extracting architecture significant requirements, but the method is mostly focused on the process
  - Shall and will approach is very expressive, but its potential is not utilized in practice
  - There is wide application of use case analysis, but the method does not provide enough guidance for quality attribute elicitation and specification
  - There is not enough information about effectiveness of global analysis and QAW in practice

Carnegie Mellon
Software Engineering Institute

Next steps

We have started collecting *business stories* both in the context of DoD and commercial organizations; e.g.:

Company X makes all of its profits from selling embedded hardware. X sells management systems for its various hardware lines for very little money and in fact looses money on them. Based on the current trends, X has determined that the time it can make a profit on the embedded devices is limited. These devices are becoming commodities. In order to survive X has must turn its management system business into a profit center. The strategies they developed are:

- Reduce R&D costs on their management systems
- Expand the market for their management systems

Currently their management systems are separated into 8 product lines that support 45 different hardware lines and geographic markets. X has determined that they can reduce their development costs by developing a single system for all of these needs.

X has also determined that there are a number of ways to expand their market.

- Develop a system that Value added resellers (3rd party vendors) are able to sell for hardware systems developed by other vendors
- Develop a system that will allow for features not currently available for particular vertical markets
- Develop a system appropriate for emerging markets (e.g. China)

Carnegie Mellon
Software Engineering Institute

More information

Shall & Will *IEEE-SA Standards Board Recommended Practice for Software Requirements Specifications*, Software Engineering Standards Committee of the IEEE Computer Society, June 25, 1998

Use Case Analysis Armour, F.; & Miller G. *Advanced Use Case Modeling*, Boston, MA: Addison-Wesley, 2000

QAW Barbacci, M. R.; Ellison, R.; Lattanze, A. J.; Stafford, J. A. ; Weinstock, C. B. ; Wood, B. G. *Quality Attribute Workshops, 3rd ed.* CMU/SEI-2003-TR-016

O'Brien Approach O'Brien, F. *The Engineering of Software Quality*, Pearson SprintPrint, 2004

Global Analysis Hofmeister, C.; Nord, R.; & Soni, D. *Applied Software Architecture*. Boston, MA: Addison-Wesley, 2000

Paulish, D. *Architecture-Centric Software Project Management*. Boston, MA: Addison-Wesley, 2001