



Carnegie Mellon
Software Engineering Institute

Pittsburgh, PA 152 13-3 890

Using Architecture-Centric Methods within Plan Driven and Agile Software Development Processes

Robert L. Nord
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213-3890 USA

Sponsored by the U.S. Department of Defense
© 2005 by Carnegie Mellon University

Version 1.0

page 1

[Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)
[31](#) [32](#)



Outline

The need for effectively integrating architecture-centric activities into the software system life cycle.

Architecture-centric methods: QAW, ADD, ATAM.

An integrated approach to software architecture design and analysis.

Plan driven approaches as exemplified by RUP.

Agile approaches as exemplified by XP.

[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)
[31](#) [32](#)



Addressing a Need

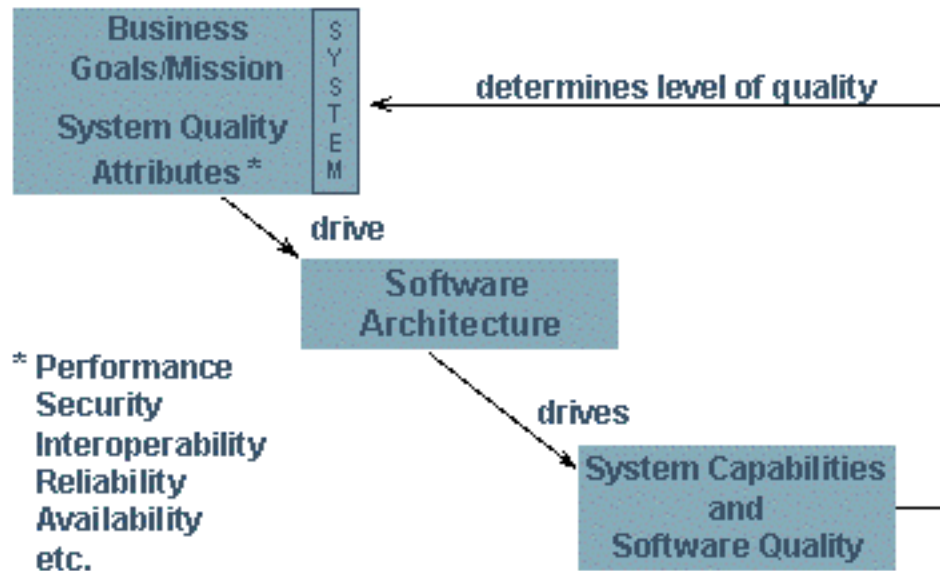
1. Software architecture is the bridge between mission/business goals and a software-intensive system.
2. Quality attribute requirements drive software architecture design.
3. Software architecture drives software development throughout the life cycle.

[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)
[31](#) [32](#)

System Qualities and Software Architecture



[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)
[31](#) [32](#)

QAW Overview

The Quality Attribute Workshop (QAW) is a facilitated method that engages system stakeholders early in the life cycle to discover the driving quality attribute requirements of a software-intensive system.



[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)
[31](#) [32](#)

ADD Overview

The Attribute-Driven Design (ADD) method is an approach to defining a software architecture by basing the design process on the quality attribute requirements of the system.





ATAM Overview

The Architecture Tradeoff Analysis Method[®] (ATAM[®]) is a method that helps a system's stakeholder community understand the consequences of architectural decisions with respect to the system's quality attribute requirements.



⊗ ATAM, and Architecture Tradeoff Analysis Method are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

© 2006 by Carnegie Mellon University

Version 1.0

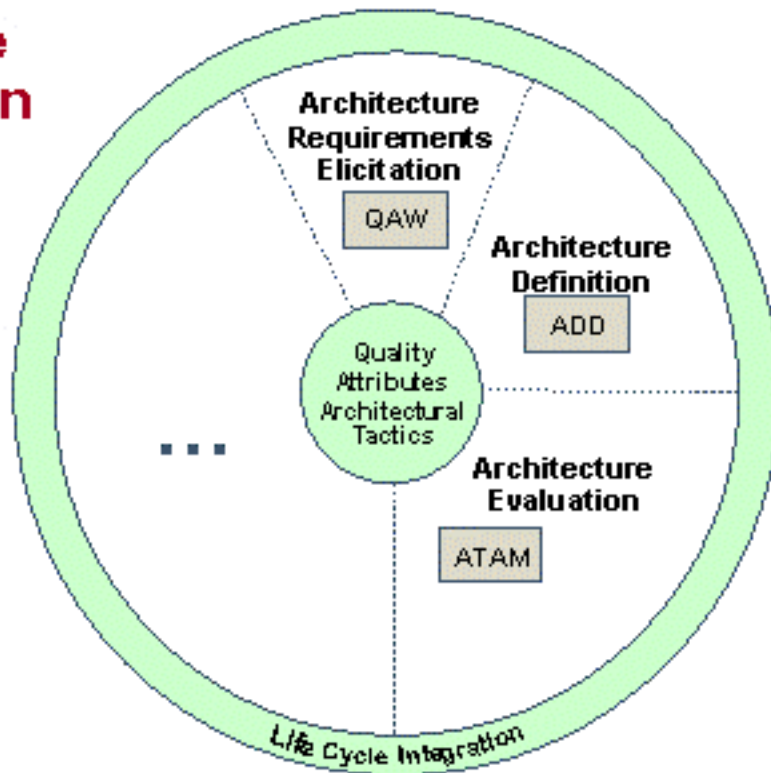
R. Word - page 18

[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)
[31](#) [32](#)

Life Cycle Integration

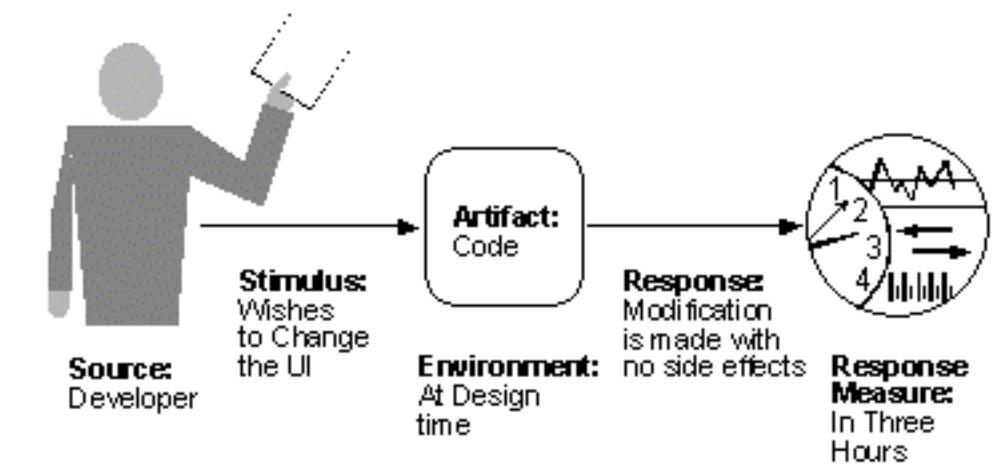


[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)
[31](#) [32](#)

Quality Attribute Scenarios

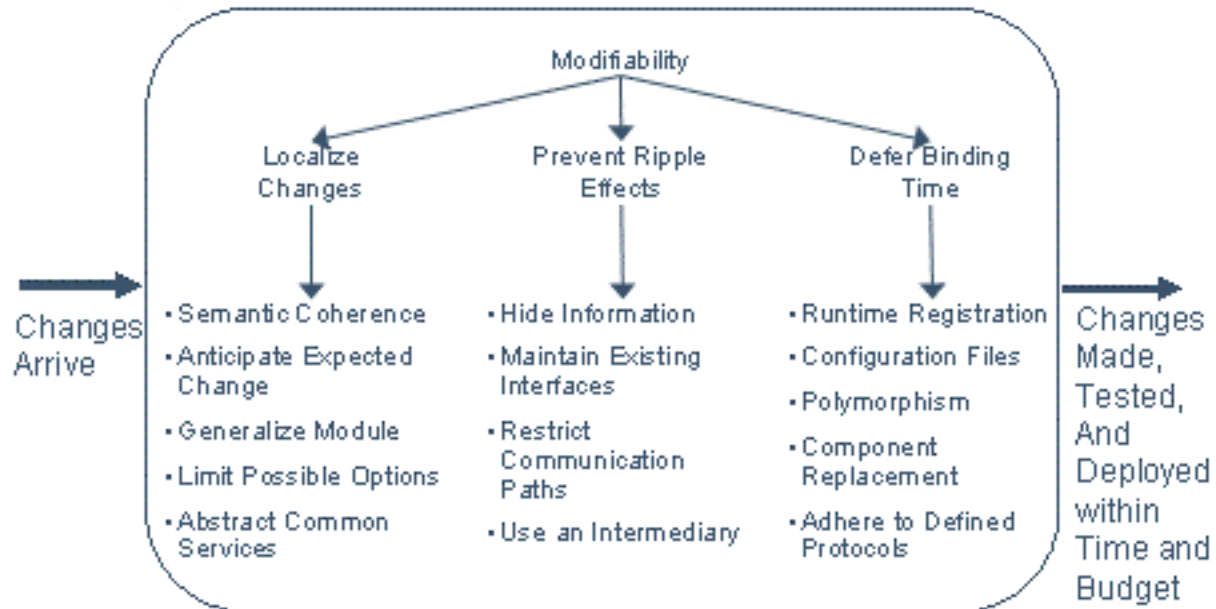


[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)
[31](#) [32](#)

Architectural Tactics



[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)
[31](#) [32](#)



Methods and Quality Attributes

The architecture-centric methods:

- are explicitly focused on quality attributes
- directly link to business and mission goals
- explicitly involve system stakeholders
- are grounded in state-of-the-art quality attribute models and reasoning frameworks
- are documented for practitioner consumption

[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)
[31](#) [32](#)

Methods and Life-Cycle Activities

Life-Cycle Activities	QAW	ADD	ATAM
Business needs	In	In	In
Requirements	In; Out	In	In; Out
Design and analysis		Out	In; Out

[Previous](#) [Next](#)[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)
[31](#) [32](#)



Life-Cycle and Architecture-Centric Activities

Life-Cycle Activity	Architecture-Centric Activity
Business Needs and Constraints	Create a documented set of <i>business goals</i> using a business presentation template.
Requirements	Elicit and document six-part <i>scenarios</i> using general scenarios, utility trees, and scenario brainstorming.
Design and Analysis	Design the architecture using <i>architectural tactics</i> . Analyze the architecture with respect to <i>architectural drivers</i> .

© 2006 by Carnegie Mellon University

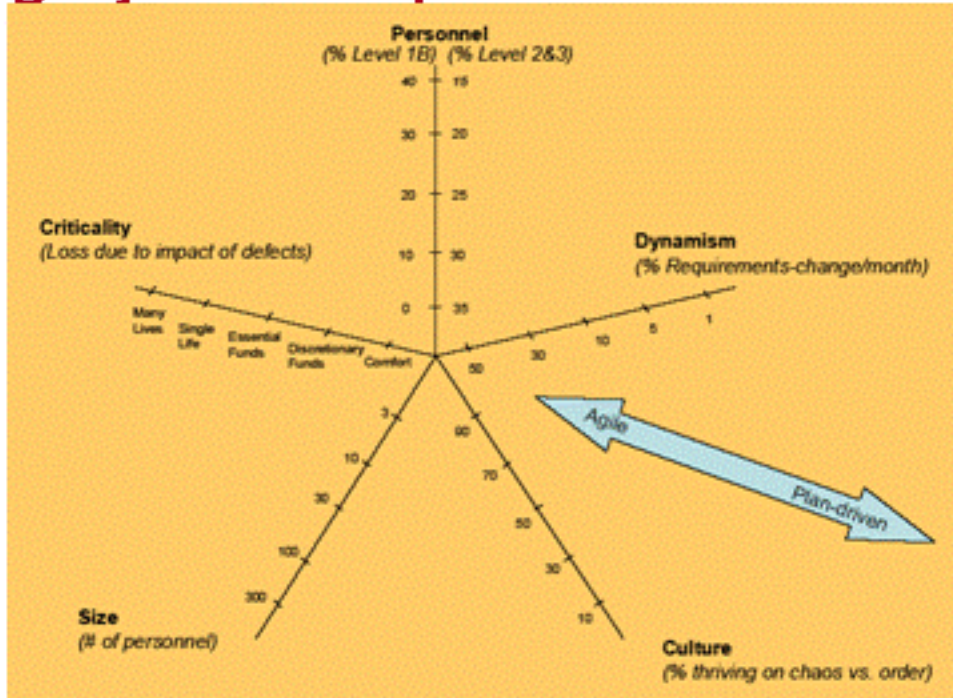
Version 1.0

R. Word - page 26

[Previous](#) [Next](#)
[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)
[31](#) [32](#)

Agility and Discipline



Boehm, B. and Turner, R. *Balancing Agility and Discipline: A Guide for the Perplexed*, Addison-Wesley, 2004.

© 2006 by Carnegie Mellon University

Version 1.0

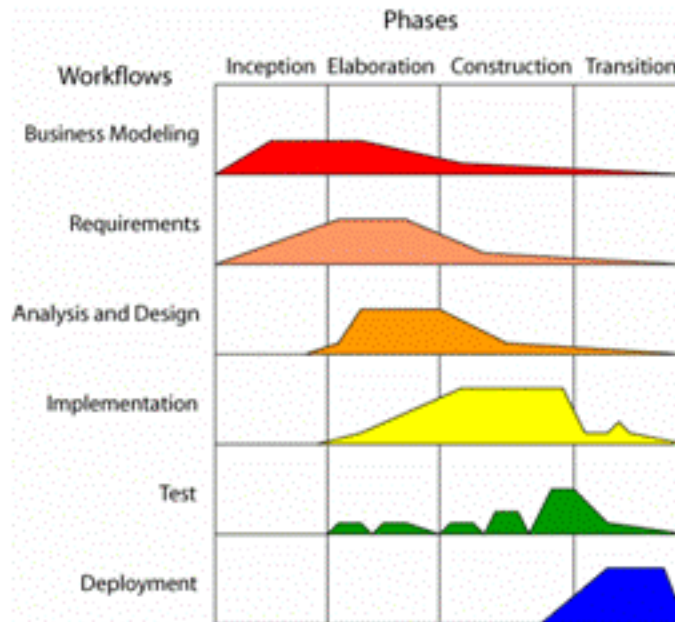
R. Word - page 27

[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)
[31](#) [32](#)

Rational Unified Process (RUP)¹



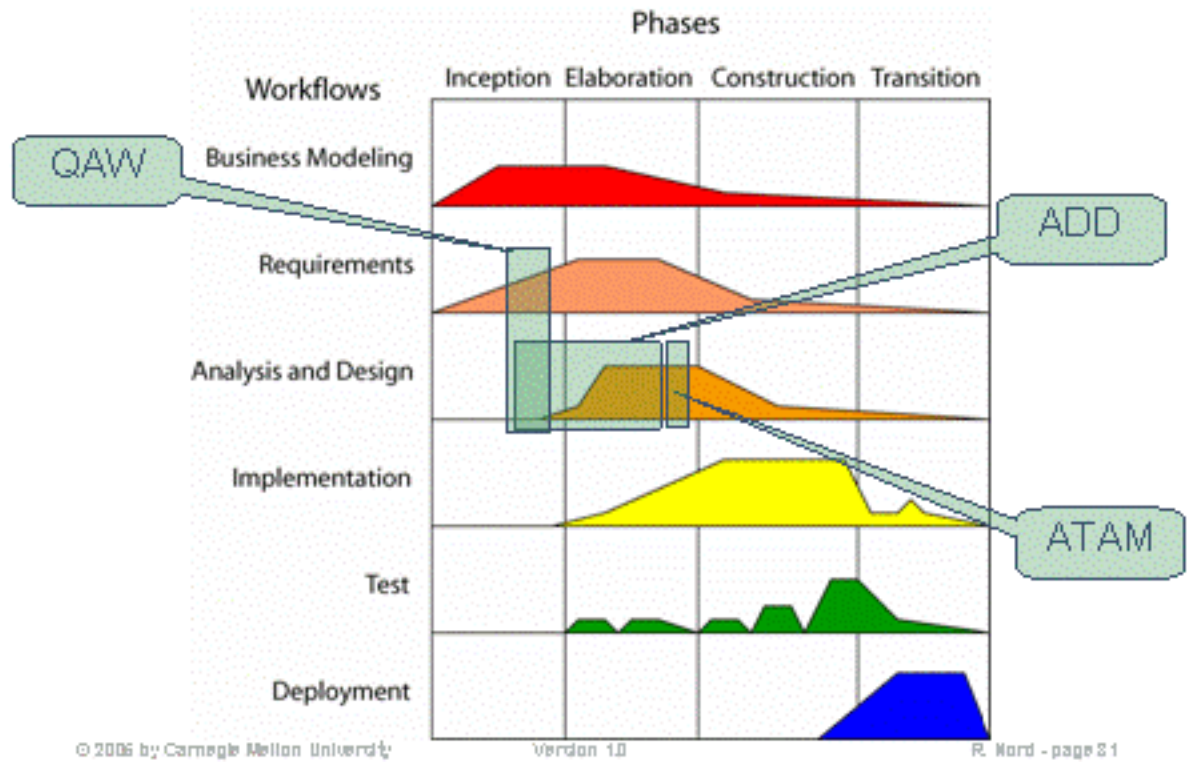
1. Krotzsch, P. *The Rational Unified Process: An Introduction*, 2nd ed. Boston, MA: Addison-Wesley, 2000.

[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)
[31](#) [32](#)

RUP and the Architecture-Centric Methods



[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)
[31](#) [32](#)



Architecture-Centric Methods as RUP Activities

Method	Role	Discipline	Workflow Detail	Artifacts Affected
QAW	Systems analyst	Reqs	Understand Stakeholder Needs	Business case Supplementary specifications
ADD	Software architect	Analysis & Design	Define a Candidate Architecture Perform Architectural Synthesis	Software architecture document (SAD)
ATAM	Technical reviewer	Analysis & Design	Refine the Architecture	Review record SAD

© 2006 by Carnegie Mellon University

Version 1.0

R. Ward - page 22

[Previous](#) [Next](#)
[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)
[31](#) [32](#)



The QAW as a RUP Activity

Elicit Quality Attribute Scenarios Using the QAW	
Purpose: Engage system stakeholders early in the life cycle to discover the driving quality attribute requirements of a software-intensive system.	
Role: Systems analyst [Analysis team]	
Frequency: As required, typically once per iteration in the Inception Phase and once in the Elaboration Phase.	
Input Artifacts:	Resulting Artifacts:
<ul style="list-style-type: none"> • business case [business drivers] • vision document [architectural plan] 	<ul style="list-style-type: none"> • business case [business goals] • supplementary specifications [scenarios]
Workflow Details:	
<ul style="list-style-type: none"> • Requirements <ul style="list-style-type: none"> • Understand Stakeholder Needs 	

[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)
[31](#) [32](#)



The ADD Method as a RUP Activity

Design the Software Architecture using the ADD Method	
Purpose: Define software architectures basing the design process on the quality attribute requirements.	
Role: Software Architect	
Frequency: Optionally in inception. First elaboration iteration. Later iterations if substantial changes to the architecture need to be explored.	
Input Artifacts: <ul style="list-style-type: none"> • Vision [constraints] • Arch. Proof-of-Concept [constraints] • Use-Case Model [functional reqts] • Suppl. Specs [quality attribute reqts] 	Resulting Artifacts: <ul style="list-style-type: none"> • Software Architecture Document [decomposition - module, concurrency, and deployment views]
Workflow Details: <ul style="list-style-type: none"> • Analysis and Design <ul style="list-style-type: none"> • Define a Candidate Architecture • Perform Architectural Synthesis 	

© 2006 by Carnegie Mellon University

Version 1.0

R. Ward - page 57

[Previous](#) [Next](#)
[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)
[31](#) [32](#)



The ATAM as a RUP Activity

Evaluate the Software Architecture Using the ATAM	
Purpose: Assess the consequences of architectural decisions in light of quality attribute requirements and business goals.	
Role: Technical reviewer [Evaluation team]	
Frequency: Occurs at least once per iteration, especially during Elaboration.	
Input Artifacts: <ul style="list-style-type: none"> • business case, vision [business drivers] • software architecture document [architectural documentation] • supplementary specifications [scenarios] 	Resulting Artifacts: <ul style="list-style-type: none"> • review record [supplemented with risk themes and impact they have on achieving the business goals] • software architecture document [annotated with sensitivity points and tradeoffs]
Workflow Details: <ul style="list-style-type: none"> • Analysis and Design <ul style="list-style-type: none"> • Refine the Architecture 	

© 2006 by Carnegie Mellon University

Version 1.0

R. Ward - page 88

[Previous](#) [Next](#)
[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)
[31](#) [32](#)



Agile Approaches (Agile Manifesto)

"We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- individuals and interactions over processes and tools
- working software over comprehensive documentation
- customer collaboration over contract negotiation
- responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more."

Kent Beck
Mike Beedle
Arie van Bennekum
Alistair Cockburn
Ward Cunningham
Martin Fowler

James Grenning
Jim Highsmith
Andrew Hunt
Ron Jeffries
Jon Kern
Brian Marick

Robert C. Martin
Steve Mellor
Ken Schwaber
Jeff Sutherland
Dave Thomas

© 2001, the above authors

This declaration may be freely copied in any form, but only in its entirety through this notice.

© 2006 by Carnegie Mellon University

Version 1.0

R. Ward - page 41

[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)
[31](#) [32](#)



Agile Development

High-level life cycles for agile software development contain the following phases:

- Initial Requirements Up Front (IRUF)
 - Initial Architecture Phase
 - Construction Phase
 - iterates through small increments
 - Deployment Phase
 - Production Phase
- } one iteration per release

The phases apply to software development during which a new product or a new release of an existing product is delivered.

[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)
[31](#) [32](#)

Agile Development

High-level lifecycle for agile software development contains the following phases:

- Initial Requirements Up Front (IRUF)
- Initial Architecture Phase
- Construction Phase
 - iterates through small increments
- Deployment Phase
- Production Phase

QAW

ADD

one iteration
per release

ATAM

The phases apply to a software development where a new product or a new release of an existing product is delivered.

[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)
[31](#) [32](#)

Architecture-Centric Methods and XP Values

XP Values	Architecture-Centric Methods Added Value
Communication	Stakeholder concerns regarding quality attribute requirements are captured and communicated to developers so that they influence design.
Simplicity	Just enough architecting. Triage: utility trees and prioritization focus efforts.
Feedback	Early feedback for understanding technical tradeoffs, risks, and return on investment of architectural decisions.
Courage	Risks are exposed early in the life cycle and give developers justification for investing resources to mitigate them.

[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)
[31](#) [32](#)



Architecture-Centric Methods and XP Activities

Method	What Happens	XP Practices and Artifacts Affected
QAW	Understand stakeholder concerns	Planning game, on site customer, test driven development, user stories
ADD	Define a course-grain architecture	Planning game, metaphor, simple design, architectural spike, refactoring
ATAM	Evaluate the architecture	Planning game, on site customer, refactoring, release plan

[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)
[31](#) [32](#)



QAW and XP Practices

XP Practices	Value Added Through the QAW
Planning Game	User stories supplemented with six-part quality attribute scenarios. Scenario prioritization and refinement guide selection of user stories for each iteration.
On-Site Customer	Additional stakeholders during a one-day workshop.
Test-Driven Development	Scenarios can be used later to evaluate the design and provide input for analysis during testing.

[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)
[31](#) [32](#)



ADD and XP Practices

XP Practices	Value Added Through ADD
Planning Game	Building a utility tree to identify architectural drivers is useful in choosing user stories.
Metaphor	Step-by-step approach to defining the architecture using module decomposition, concurrency, and deployment views.
Simple Design	Course-grained architecture that ensures the design meets its quality attribute requirements and mitigates any associated risks.
Refactoring	Refactoring, which is driven by quality attribute needs (make it faster, make it more secure, etc.), is aided by the application of architectural tactics.

[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)
[31](#) [32](#)



ATAM and XP Practices

XP Practices	Value Added Through the ATAM
Planning Game	User stories supplemented with six-part quality attribute scenarios. Scenario prioritization and refinement guide selection of user stories for each iteration.
On-Site Customer	Additional stakeholders during an evaluation workshop.
Refactoring	Artifacts (e.g., sensitivity points, tradeoffs) necessary for understanding the design before refactoring.

[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)
[31](#) [32](#)



Conclusions

The benefit of including the SEI methods is to address quality attributes in an explicit, methodical, engineering-principled way.

We believe that quality attribute requirements drive the software architecture and that architecture-centric activities (with an explicit focus on quality attributes) drive the software system's life cycle.

[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)
[31](#) [32](#)



Source of material

Kazman, R.; Nord, R.L.; Klein, M. *A Life Cycle View of Architecture Analysis and Design Methods* (CMU/SEI-2003-TN-026). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2003.

<<http://www.sei.cmu.edu/publications/documents/03reports/03tr026.html>>.

Kazman, R.; Kruchten, P.; Nord, R.L.; Tomayko, J.E. *Integrating Software Architecture-Centric Methods into the Rational Unified Process* (CMU/SEI-2004-TR-011), 2004.

<<http://www.sei.cmu.edu/publications/documents/04reports/04tr011.html>>.

Nord, R.L.; Tomayko, J.E.; Wojcik, R. *Integrating Software-Architecture-Centric Methods into Extreme Programming (XP)* (CMU/SEI-2004-TN-036), 2004.

<<http://www.sei.cmu.edu/publications/documents/04reports/04tn036.html>>.

[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)
[31](#) [32](#)



For Additional Information

Linda Northrop
Director
Product Line Systems Program
Telephone: 412-268-7638
Email: lmn@sei.cmu.edu

Hal Stevens
Business Development
Product Line Systems Program
Telephone: 412-265-8207
Email: hfs@sei.cmu.edu

U.S. Mail:
Software Engineering Institute
Carnegie Mellon University
4500 Fifth Avenue
Pittsburgh, PA 15213-3890
Fax: 412-268-5758

Robert L. Nord
Product Line Systems Program
Telephone: 412-268-1705
Email: rn@sei.cmu.edu

World Wide Web:
<http://www.sei.cmu.edu/architecture>

[Previous](#) [Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)
31 [32](#)



Acronyms

ADD	Attribute-Driven Design
ATAM	Architecture Tradeoff Analysis Method
QAW	Quality Attribute Workshop
RUP	Rational Unified Process
SEI	Software Engineering Institute
XP	Extreme Programming

[Previous](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)
[31](#) [32](#)