

Introduction to Spiral Development

May 18, 2012

Peter Hantos
Software Acquisition and Process Department
Software Engineering Subdivision

Prepared for:

Space and Missile Systems Center
Air Force Space Command
483 N. Aviation Blvd.
El Segundo, CA 90245-2808

Authorized by: Senior Vice President, Engineering and Technology Group

Approved for public release; distribution unlimited.

20120613107

Introduction to Spiral Development

May 18, 2012

Peter Hantos
Software Acquisition and Process Department
Software Engineering Subdivision

Prepared for:
Space and Missile Systems Center
Air Force Space Command
483 N. Aviation Blvd.
El Segundo, CA 90245-2808

Authorized by: Senior Vice President, Engineering and Technology Group

Approved for public release; distribution unlimited.

Introduction to Spiral Development

Approved by:



Leslie J. Holloway, Department Director
Software Acquisition and Process
Department
Computers and Software Division
Engineering and Technology Group

© The Aerospace Corporation, 2012.

All trademarks, service marks, and trade names are the property of their respective owners.

Introduction to Spiral Development

Dr. Peter Hantos
Software Acquisition and Process Department
Software Engineering Subdivision
The Aerospace Corporation

Outline

- Definitions
- The Original Spiral Model of Software Development
- Invariant Characteristics of the Spiral Model
- Basic Spiral Concepts
- The Spiral as a Process Model Generator
- Modeling Concurrency Using UML® Activity Diagrams
- Various Depictions of the Spiral Development Model
- Invariants 1-6
- Example Hierarchy of System and Software Life Cycles
- The Final Word...
- Acronyms
- References
- Backup



Definitions

- **Definition of Iteration**
 - *A procedure in which repetition of a sequence of operations yields results successively closer to a desired result*
- **Iterative Development**
 - *Involves repetition - **iterative**, **spiral**, **cyclical** are synonyms*
 - *Iterative development involves **learning***
 - Create – Review – Change (Improve) on the basis of feedback
 - *Iteration is **planned revision***
 - Work units (scope of iteration) determined by engineering objectives
 - *Note that work units of iterations do not necessarily provide additional capability or functionality; the objective might be experimentation or performance enhancement*
 - *Iteration in development is a **risk mitigation mechanism***
 - to deal with uniqueness, complexity and technology uncertainties

Spiral development **is** iterative development, with additional constraints



Definitions-2

- **Build**
 - A software system “build” is defined as a version of the software system that delivers a specified subset of the requirements that the completed software system will meet
 - To run a simple program, we only have to compile and link it; the process is straightforward, the created build is small
 - A typical, large-scale project involves dozens to even thousands of components and libraries, requiring a more complex build process to create an executable image that can be run on a computer
- **Release**
 - The noun “release” refers to a subset of the end product
 - A software system release is instantiated through the delivery of a build
- **Increment**
 - “Increment” is the difference (delta) between two subsequent releases

Increment is a conceptual term that in software is instantiated through a tangible object, the “build”



Increments and Builds

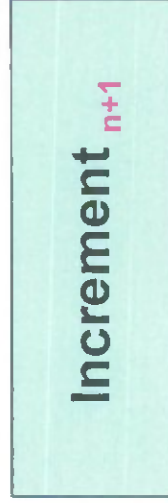
Requirements Subset $n-1$



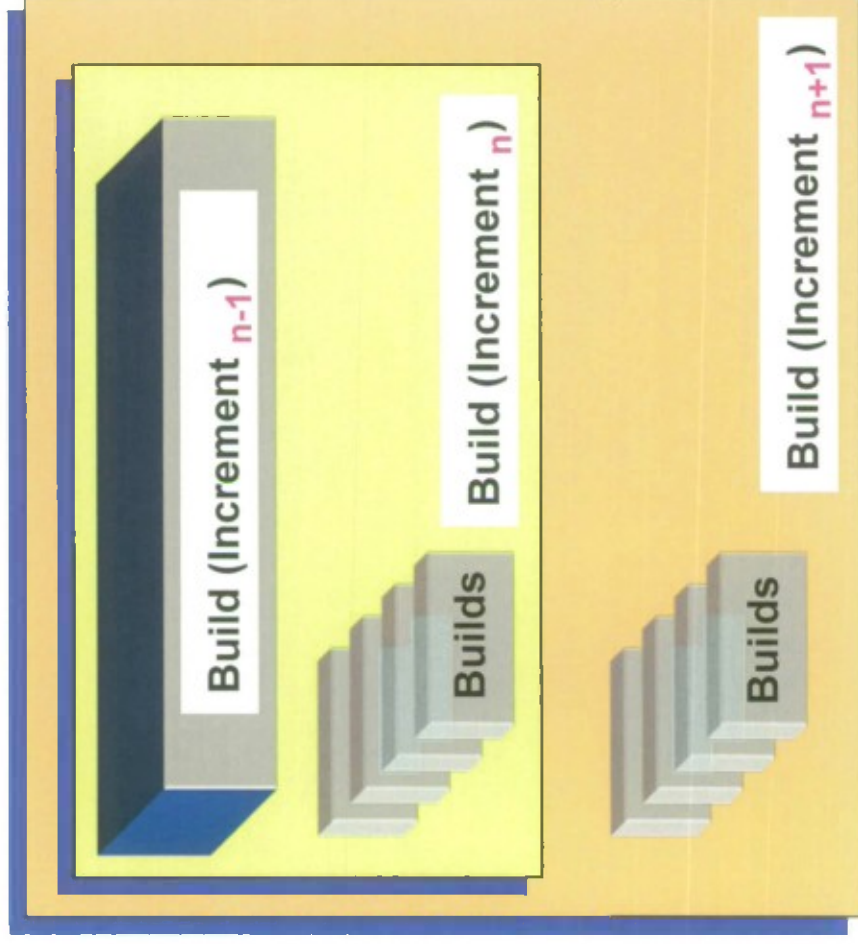
Requirements Subset n



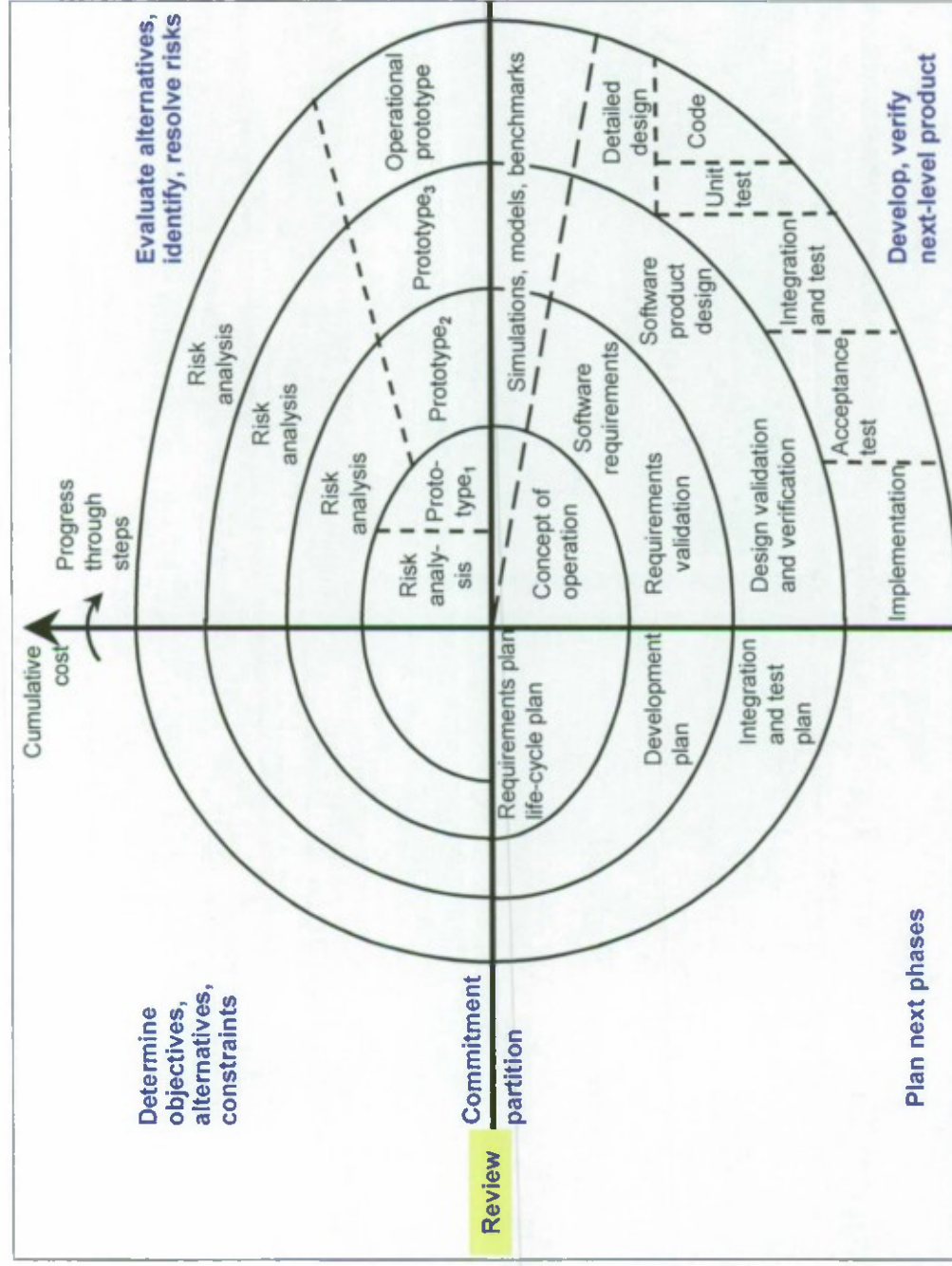
Requirements Subset $n+1$



...



The Original Spiral Model of Software Development*



*Source [Boehm 88]

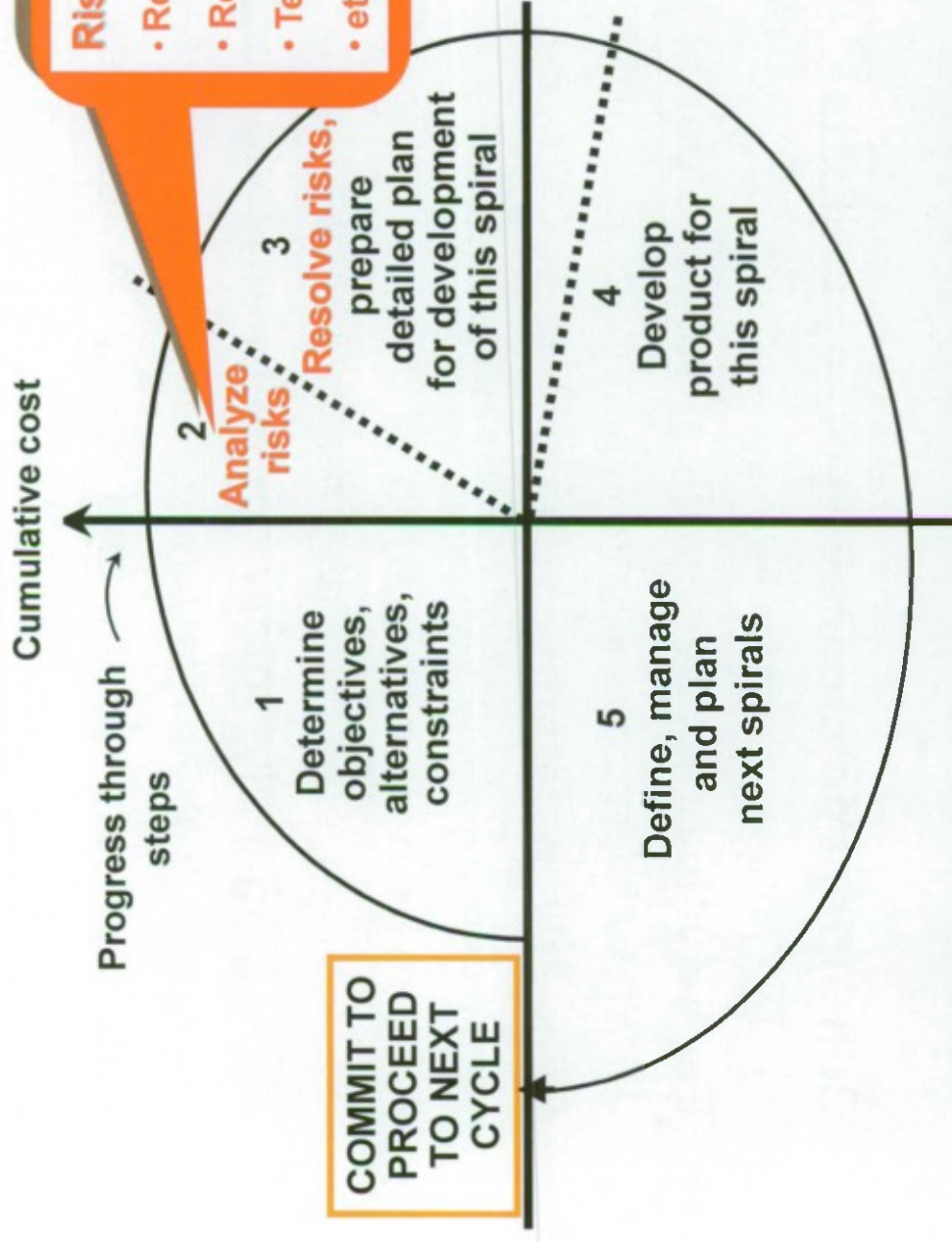
Invariant Characteristics of the Spiral Model*

- **Concurrent determination of key artifacts**
 - *The process is artifact-driven, and not document-driven*
- **Each cycle considers critical stakeholder objectives**
 - *Stakeholder commitment is obtained on all alternatives*
- **Risk-driven determination of level of effort within cycles**
 - *Avoids overkill or belated risk resolution*
- **Risk-driven determination of degree of detail for artifacts**
 - *Avoids overkill or belated risk resolution*
- **Managing stakeholder commitments via Anchor Points**
 - *Brings in an architecture-centric management view*
- **Emphasis on system and life cycle activities and artifacts**
 - *Rather than only software and initial development*

* Source [Boehm00]



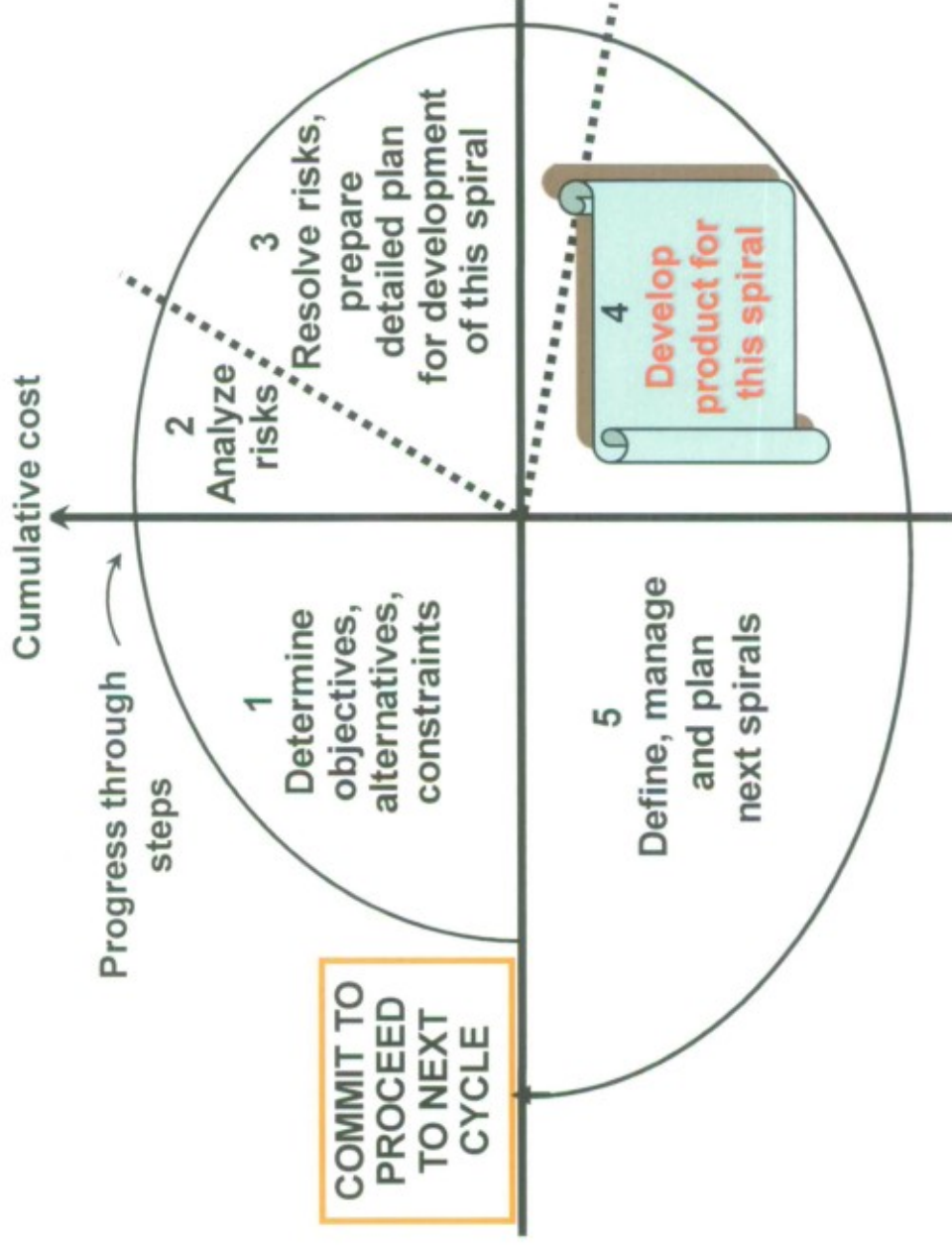
Basic Spiral Concepts



Requirements are defined, updated, or elaborated separately for every successive spiral



The Spiral as a Process Model Generator

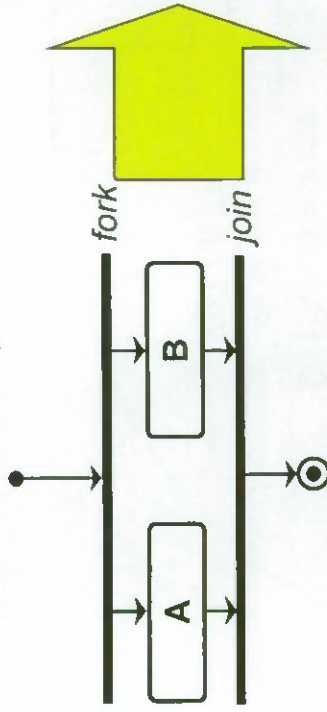


Note that the model doesn't really specify "how" the product for this spiral will be developed



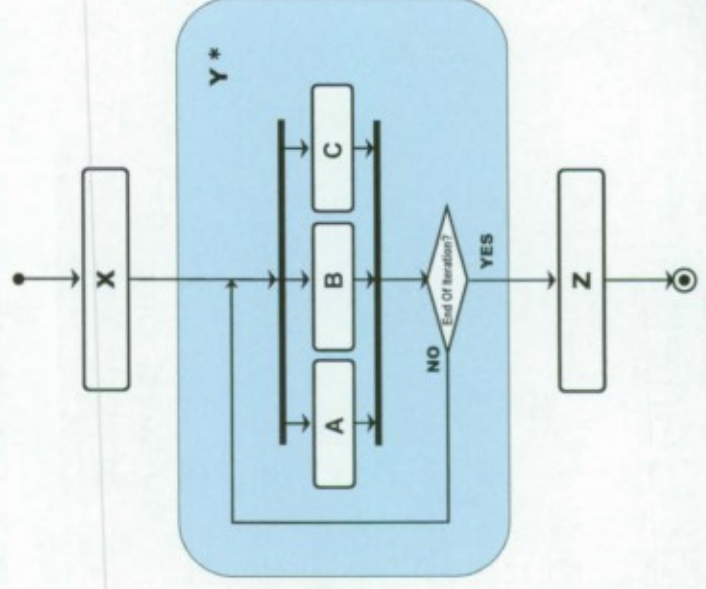
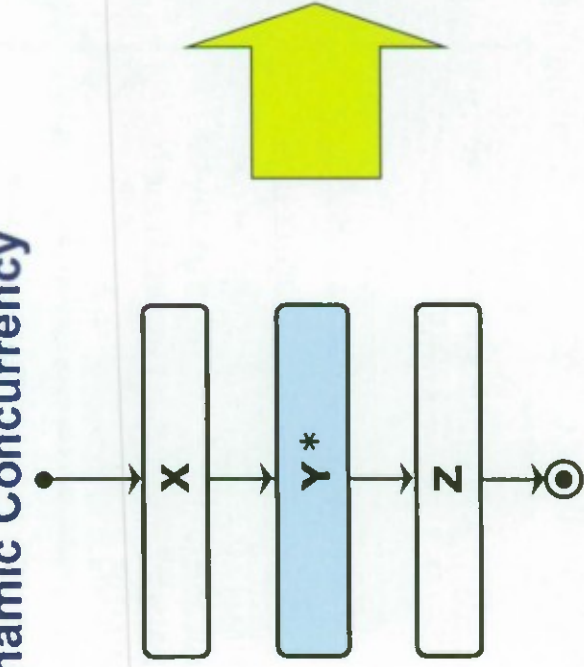
Modeling Concurrency Using UML Activity Diagrams

Concurrency



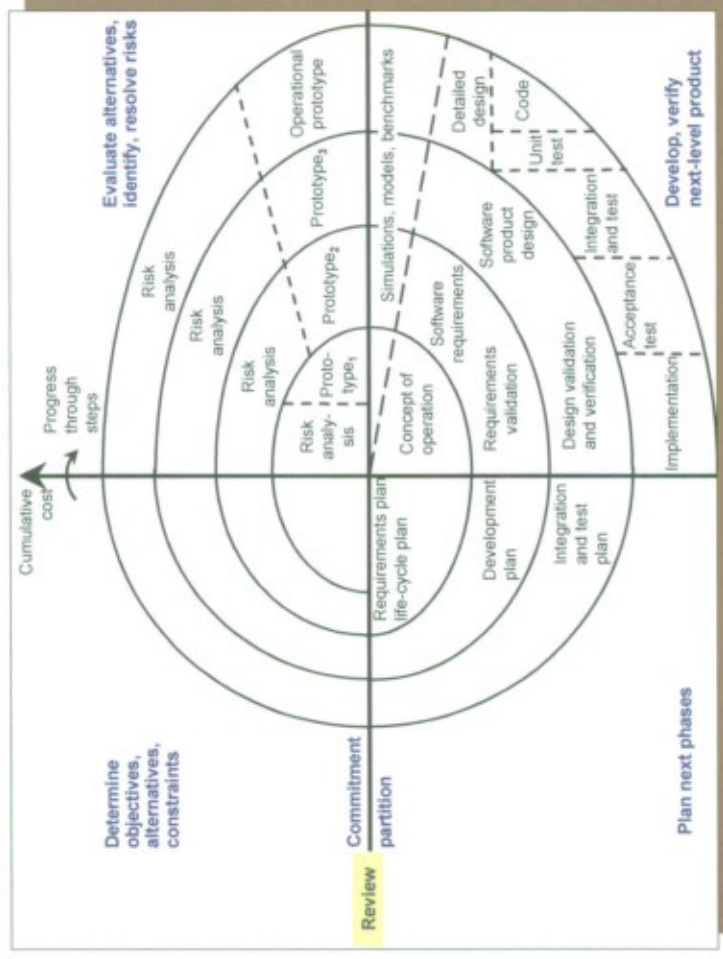
- Task execution sequence is irrelevant
 - A, B
 - B, A
 - A & B simultaneously
- i.e., execution is interleaved

Dynamic Concurrency

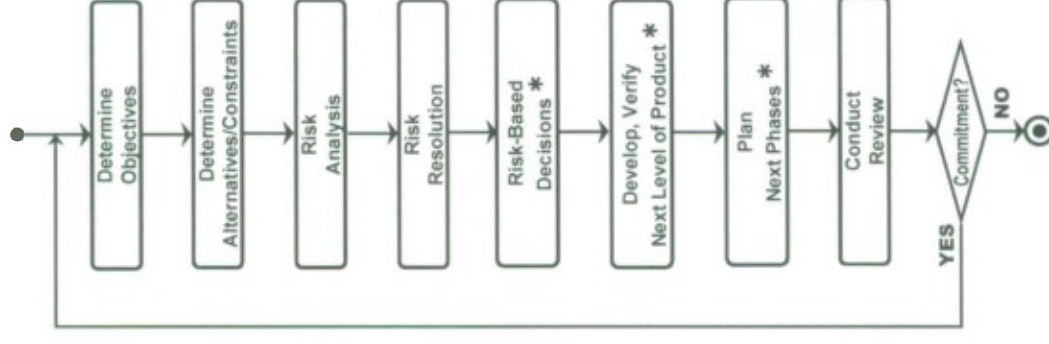


Various Depictions of the Spiral Development Model

Boehm 1988



UML Activity Diagram

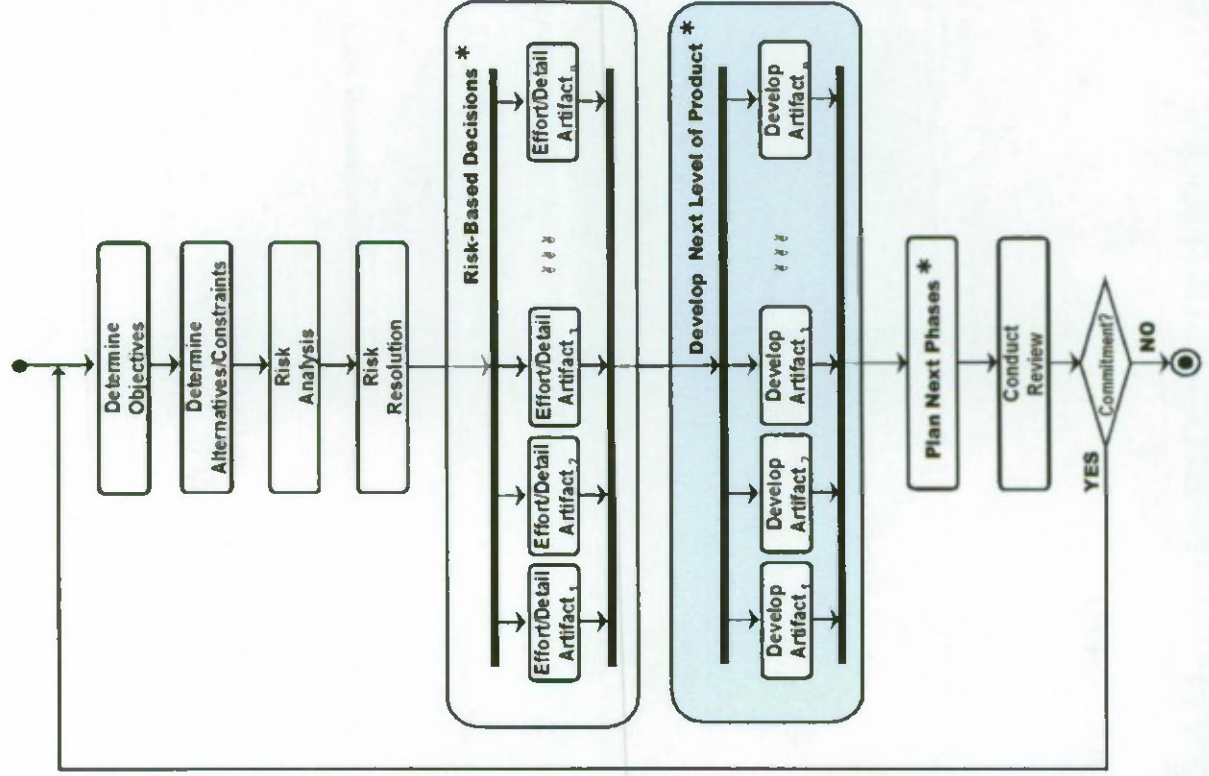


Missing: The "uncoiled" spiral

Boehm 2000

The Spiral Development Model is a risk-driven process model generator for guiding multi-stakeholder concurrent engineering of software-intensive systems. Its distinguishing features include a cyclic approach for incrementally growing a system's degree of definition and implementation, and a set of anchor point milestones for ensuring feasibility of the incremental definitions and implementations.

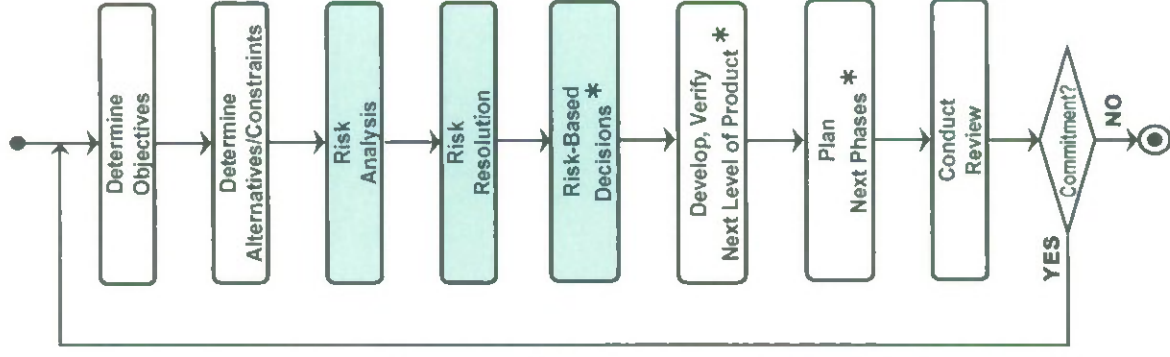
Invariant 1: Concurrent Determination of Key Artifacts



- Concurrent Engineering
 - Refers to the concurrent development of **artifacts**, not WBS elements
 - Typical artifacts:
 - Requirements
 - Plans and schedules
 - Estimates
 - Design documents
 - Code
 - Test plans
 - Test cases
 - User documentation
 - Etc.
 - Effort/Detail determination for artifacts is a risk-based decision
 - Concurrency is **dynamic**
 - A certain level of iteration is needed amongst the concurrent activities



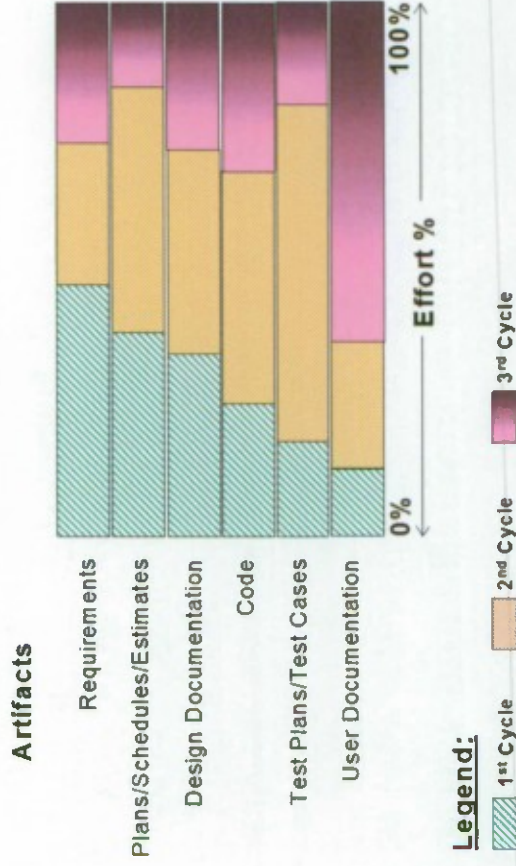
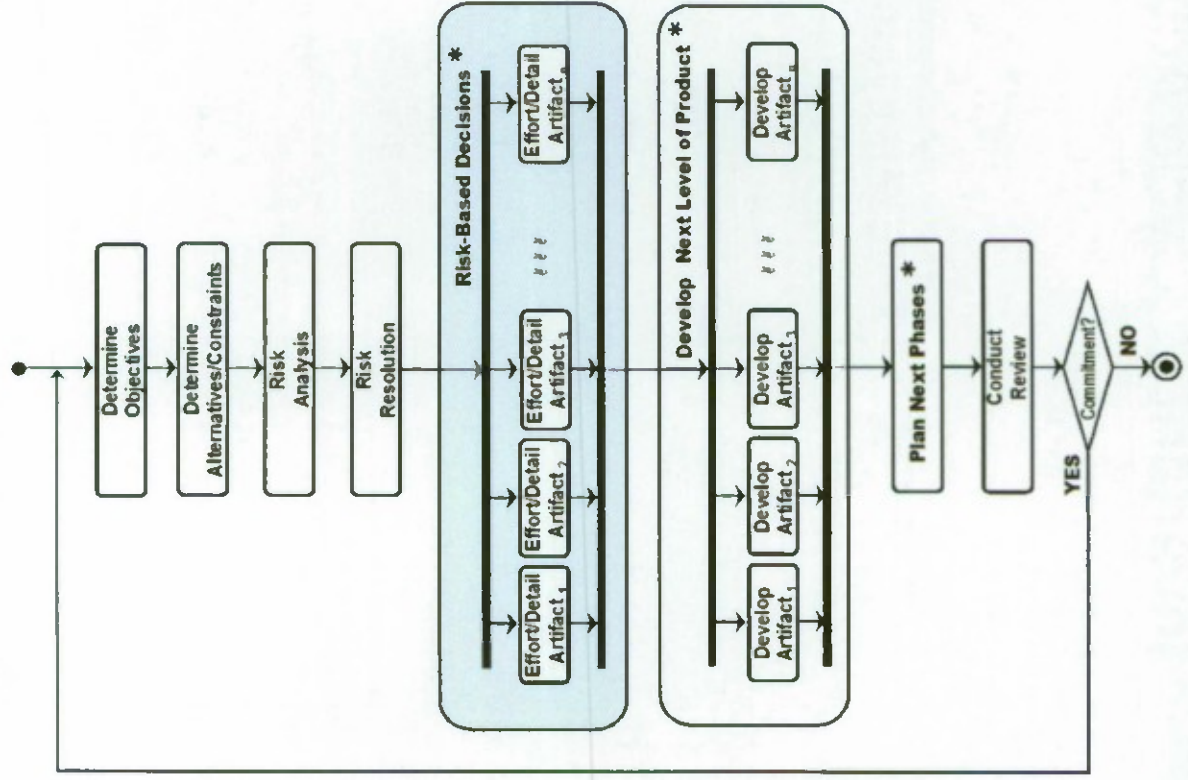
Invariant 2: Cycle Activities



- **Each cycle considers:**
 - Critical stakeholder objectives and constraints
 - Product and process alternatives
 - Risk identification and resolution
 - Stakeholder review
 - Commitment to proceed
- **Caveat: role of risk management is misunderstood**
 - The common view is that risk management is a continuous activity
 - This notion implies that risk management is practiced concurrently with development
 - However, **risk-based planning** (risk analysis, risk resolution, and miscellaneous risk-based decisions) must precede the development of the “Next Level of Product” and can not be done concurrently



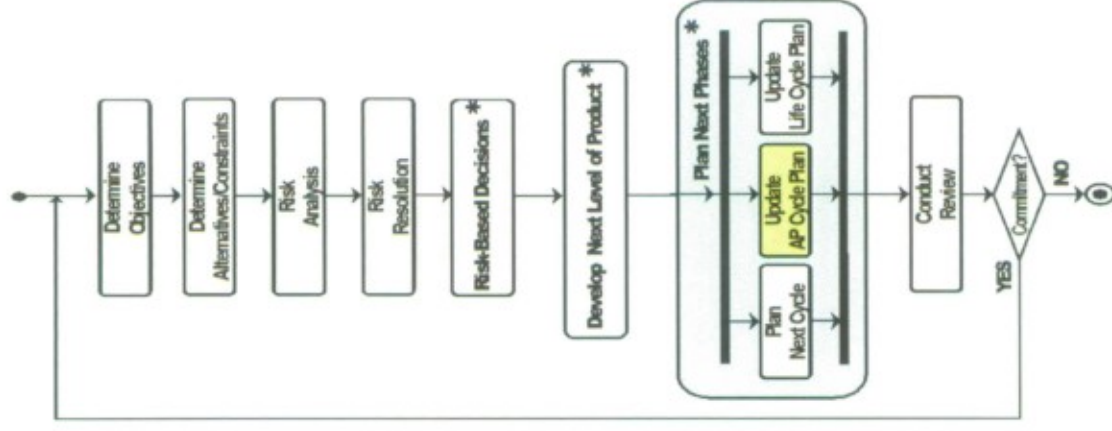
Invariants 3 & 4: Level of Effort and Degree of Detail Determination



- **Invariant 3:**
 - **Level of effort** is driven by risk considerations
- **Invariant 4:**
 - **Degree of detail** of artifacts is driven by risk considerations



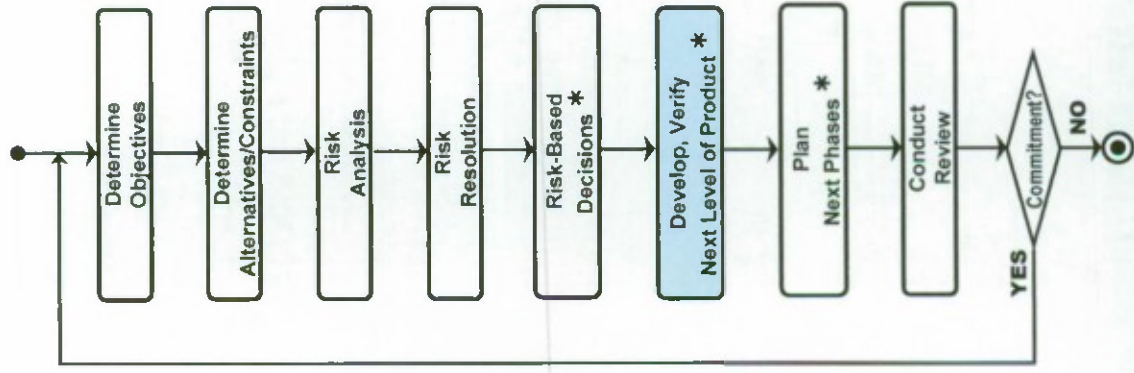
Invariant 5: Anchor Point Milestones



- **Stakeholder life cycle commitments are managed via anchor point milestones**
 - Note that *Anchor Points (AP)*, or intermediate milestones, were formally introduced only in 1996*
 - Nevertheless, update of the global life cycle plan was always part of the model
 - Anchor points represent a bridge between short term cycle objectives and long-term life cycle objectives
 - **AP is a risk-driven, incremental approach to ensure the achievement of the project's global life cycle objectives**
 - The three planning activities are done concurrently
 - The concurrency is dynamic, because iteration is needed across the plans



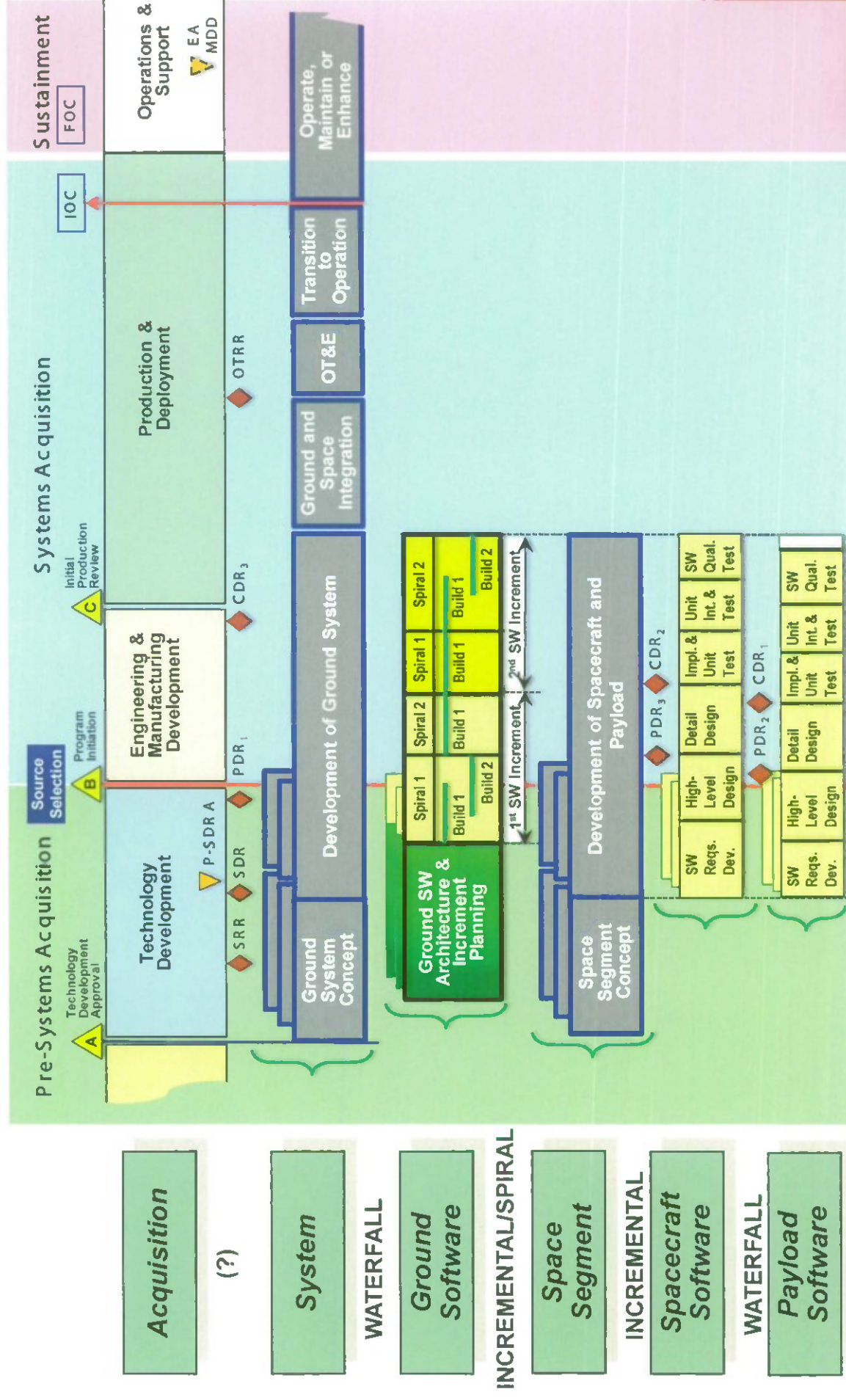
Invariant 6: Emphasis on System and Life Cycle Activities



- Emphasis on system and life cycle activities and artifacts rather than software and initial development artifacts
 - This is a restatement of the positioning of the Spiral as a **process model generator** rather than a specific process model
 - “Don’t sweat the small stuff”
 - Note that neither the **“What”** nor the **“How”** is specified in the “Develop, Verify Next Level of Product” activity box
 - The determination of specific processes can be also part of the “Risk-Based Decisions” activity cluster



Example Hierarchy of System and Software Life Cycles



Caveat: Diagram is only conceptual; phase durations are not to scale

The Expert's Voice...

“Iterative development is not a magic wand that when waved, solves all possible problems and difficulties in software development. Projects are not easier to set up, to plan, or to control just because they are iterative. The project manager will actually have a more challenging task, especially during his or her first iterative project, and most certainly during the early iterations of that project, when risks are high and early failure is possible.”

~~~ Philippe Kruchten\*

---

\*At the time of this quote, Kruchten was a well-respected software thought-leader, an IBM/Rational Fellow. The quote is from his article in 2000 [Kruchten 00]



# More Expert Voices – Or What My Dentist and Martin Fowler Have in Common...



- Sign in my dentist's office:  
***"Brush only those teeth you wish to keep..."***
- **Martin Fowler** on iterative development:  
***" You should use iterative development only on projects that you want to succeed."***  
(UML Distilled Second Edition, Addison-Wesley, 2000, pp 37)

---

\* Fowler is also a known and well-respected software consultant and author



# Acronyms

|                |                                   |
|----------------|-----------------------------------|
| <b>AP</b>      | Anchor Point                      |
| <b>CDR</b>     | Critical Design Review            |
| <b>DOD</b>     | Department of Defense             |
| <b>EA</b>      | Evolutionary Acquisition          |
| <b>FOC</b>     | Final Operational Capability      |
| <b>IOC</b>     | Initial Operational Capability    |
| <b>MDD</b>     | Materiel Development Decision     |
| <b>OMG</b>     | The Object Management Group       |
| <b>OTRR</b>    | Operational Test Readiness Review |
| <b>PDR</b>     | Preliminary Design Review         |
| <b>P-SDR A</b> | Post SDR Assessment               |
| <b>SDR</b>     | System Design Review              |
| <b>SRR</b>     | System Requirements Review        |
| <b>UML</b>     | Unified Modeling Language         |



## References

- [Boehm 00]** Boehm, B.W. (editor with Hansen, W. J.,) *Spiral Development: Experience, Principles, and Refinements*, Spiral Development Workshop, February 9, 2000, CMU/SEI-2000-SR-008
- [Boehm 88]** Boehm, B.W., *A Spiral Model of Software Development and Enhancement*, IEEE Computer, May 1988
- [Boehm 96]** Boehm, B.W., *Anchoring the Software Process*, IEEE Software, July 1996
- [Kruchten 00]** Kruchten, P., *From Waterfall to Iterative Development: A Challenging Transition for Project Managers*, The Rational Edge, December 2000



# Use of Trademarks, Service Marks, and Trade Names

Use of any trademarks in this material is not intended in any way to infringe on the rights of the trademark holder. All trademarks, service marks, and trade names are the property of their respective owners.

