IBM Software Group

Rational. software

What You Didn't Know About RUP

Ivar Jacobson

IBM Rational

ihjacobs@us.ibm.com

Jaczone AB

ivar@jaczone.com

e business on demand software

Agenda

- What you may know about RUP
- What you should know about RUP
- RUP is futurized
- RUP for outsourcing
- Summary

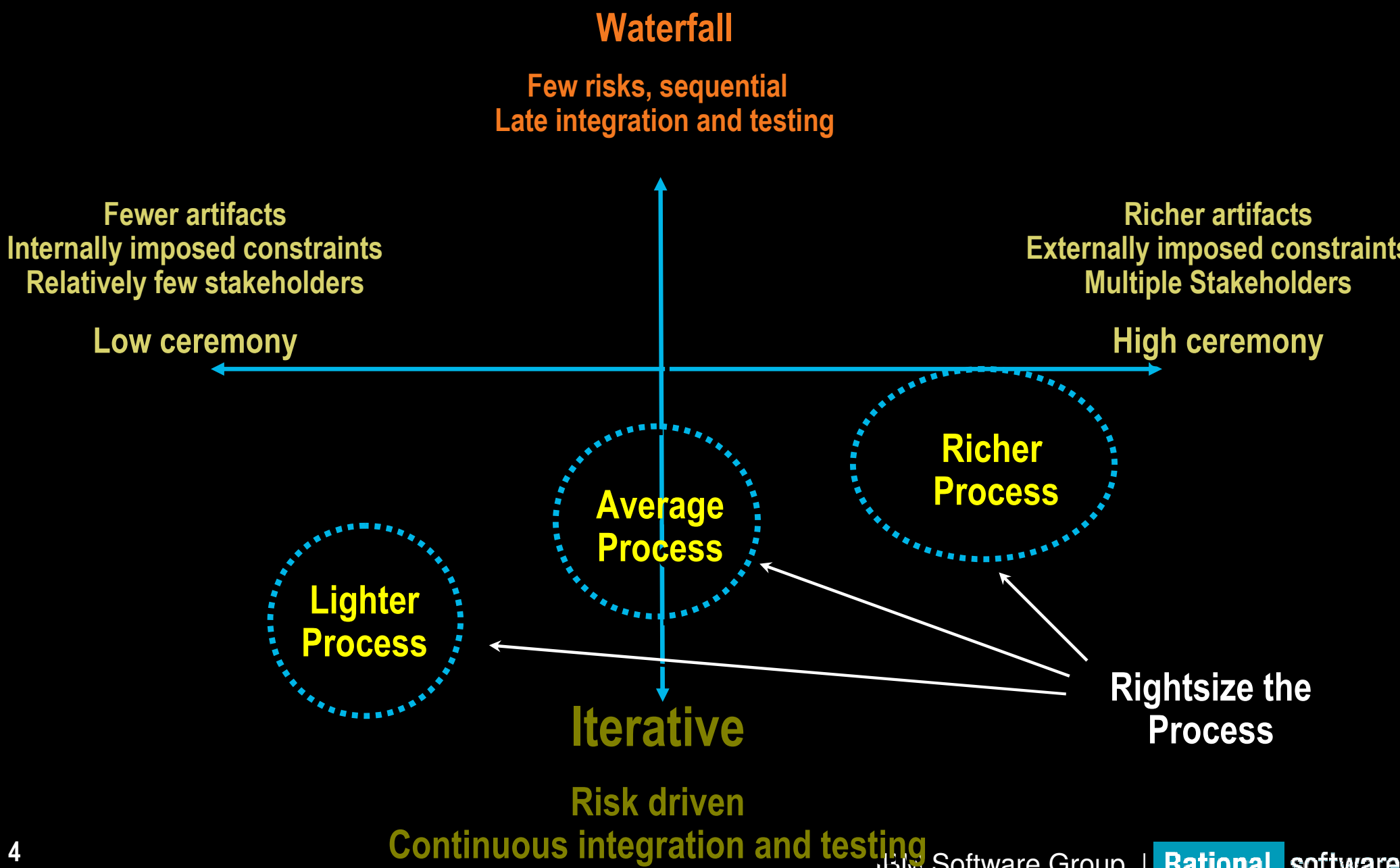
Best Practices of Modern CBD

Use component architecture and

- Develop iteratively (result-based instead of activity-based management)
- Develop models with UML – MDD
- Make architecture first
- Drive requirements from business mo
- Manage requirements with use cases
- Verify quality from the beginning
- Control changes
- Design for and with reuse
- Use result-oriented process with integral t

This is Rational Unified Process

Use Result-Oriented Process



Agenda

- What you may know about RUP
- What you should know about RUP
- RUP is futurized
- RUP for outsourcing
- Summary

Software development has never been as complex

- You need to know about
  - Programming languages and environments
  - Systemware and middleware – J2EE, .NET, database management systems, Websphere,
  - Packaged solutions, webservices, legacy systems
  - Business modeling, requirements, analysis, design, coding, test
  - Workflow spec languages, UML, XML,..
  - Configuration management, project management
  - Process and process configuration
  - All kinds of tools
  - Etc.
- Where do you learn all this? ☺

From books? Here some Rational books

Authors: Ivar Jacobson | Grady Booch | Jim Rumbaugh | Walker Royce
Philippe Kruchten | Dean Leffingwell | Agneta Jacobson
Nasser Kettani | Magnus Christerson | Maria Ericsson I Brian White
Terry Quatrani | Jim Conallen | Gunnar Overgaard | Murray Cantor

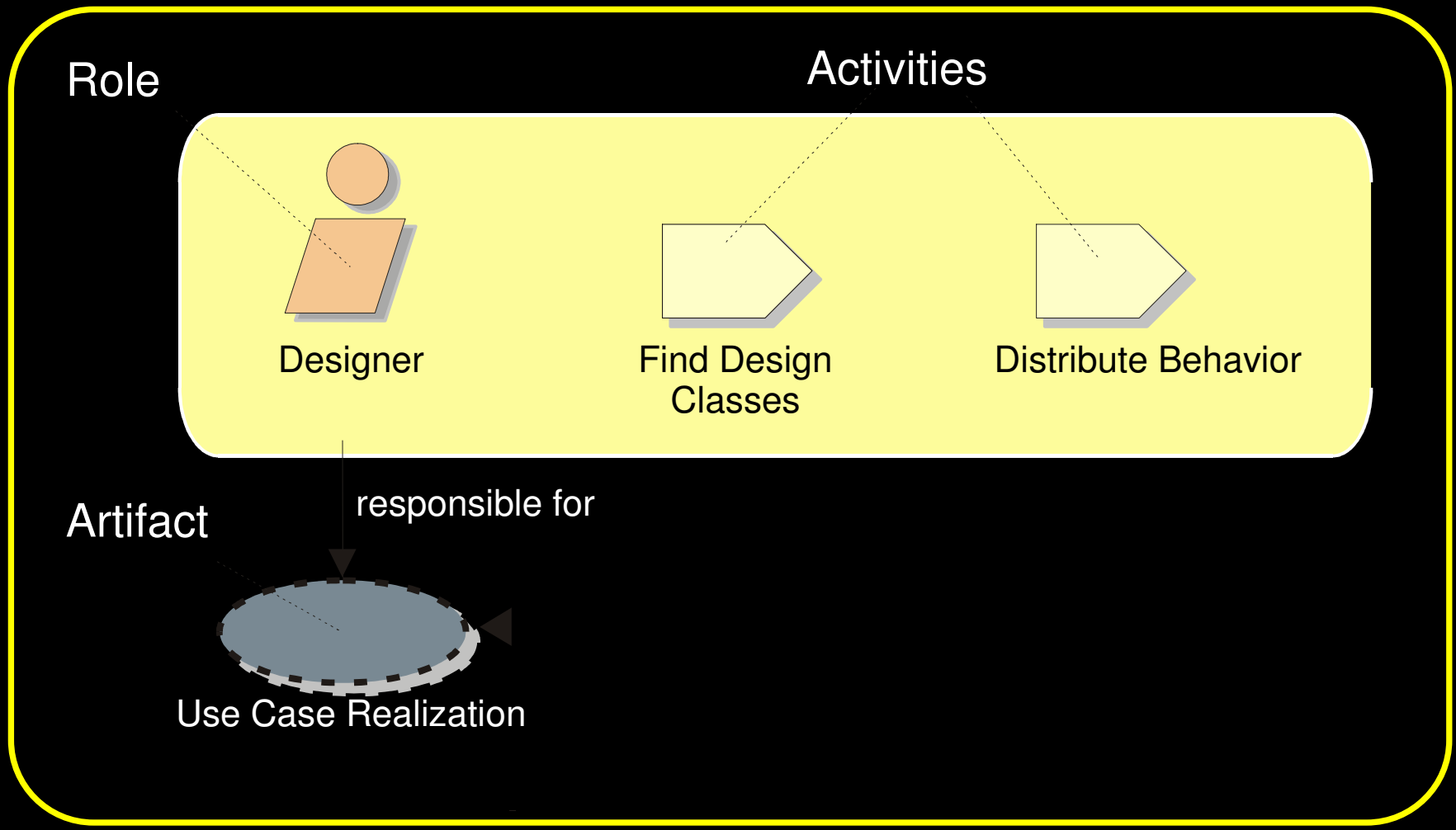
You need 30 books or so

- Books written by different authors
- Books written with different terminology and approach
  - Overlapping, conflicting
- Books with 30 different focus
- Nobody can read 30 books
- No single book has it all

RUP is a Huge Knowledgebase

- RUP is a huge knowledge base
  - Covering most lifecycle activities from business modeling to deployment
  - Covering cross-lifecycle activities – project mgmt, configuration mgmt, development environment
  - Nobody reads it all
  - But it is there ready to access

RUP is Process Engineered



Why is RUP engineered?

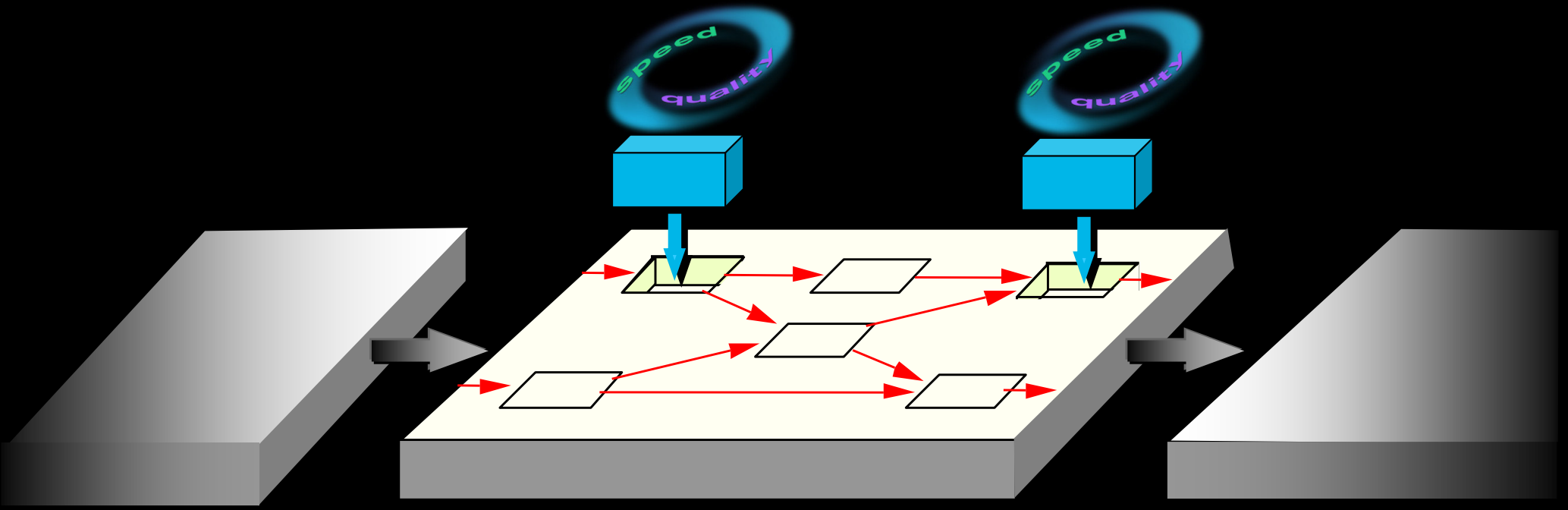
To get to the depth and breadth required to really provide substantial knowledge we need a team to develop the process in collaboration

- The process team needs to understand the process
- We want to
  - Understand it as well
  - Configure the process
  - Specialize it
  - Extend it – all the time
- We want to understand what kind of tools we need to efficiently develop software

RUP is a Configurable Process

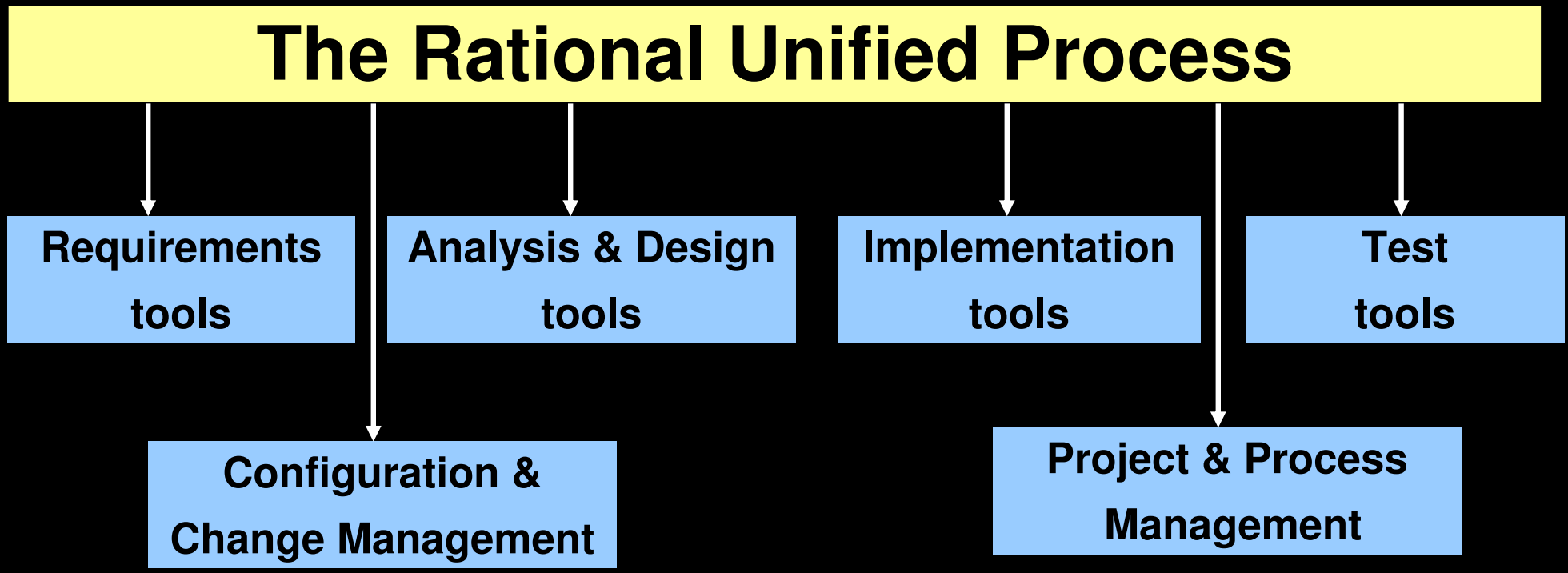
It is a Process Framework

- Configurable to the needs and constraints of the organization
- Our technique for modeling process has evolved into an OMG standard -- SPEM



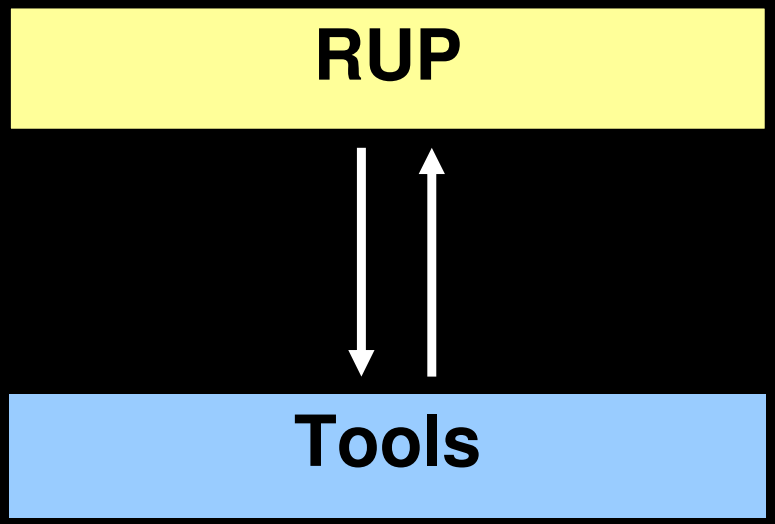
RUP is Tool Supported

- RUP is a specification for tool support
  - It defines the activities that need tool support
  - It defines all artifacts used and produced



Process and Tools evolve together

- RUP and supporting tools will evolve integral to one another
  - A process of model Y2000 is just an academic exercise if tools are not developed concurrently
  - It will be deeply integrated with tools from many sources



Agenda

- What you may know about RUP
- What you should know about RUP
- RUP is futurized
- RUP for outsourcing
- Summary

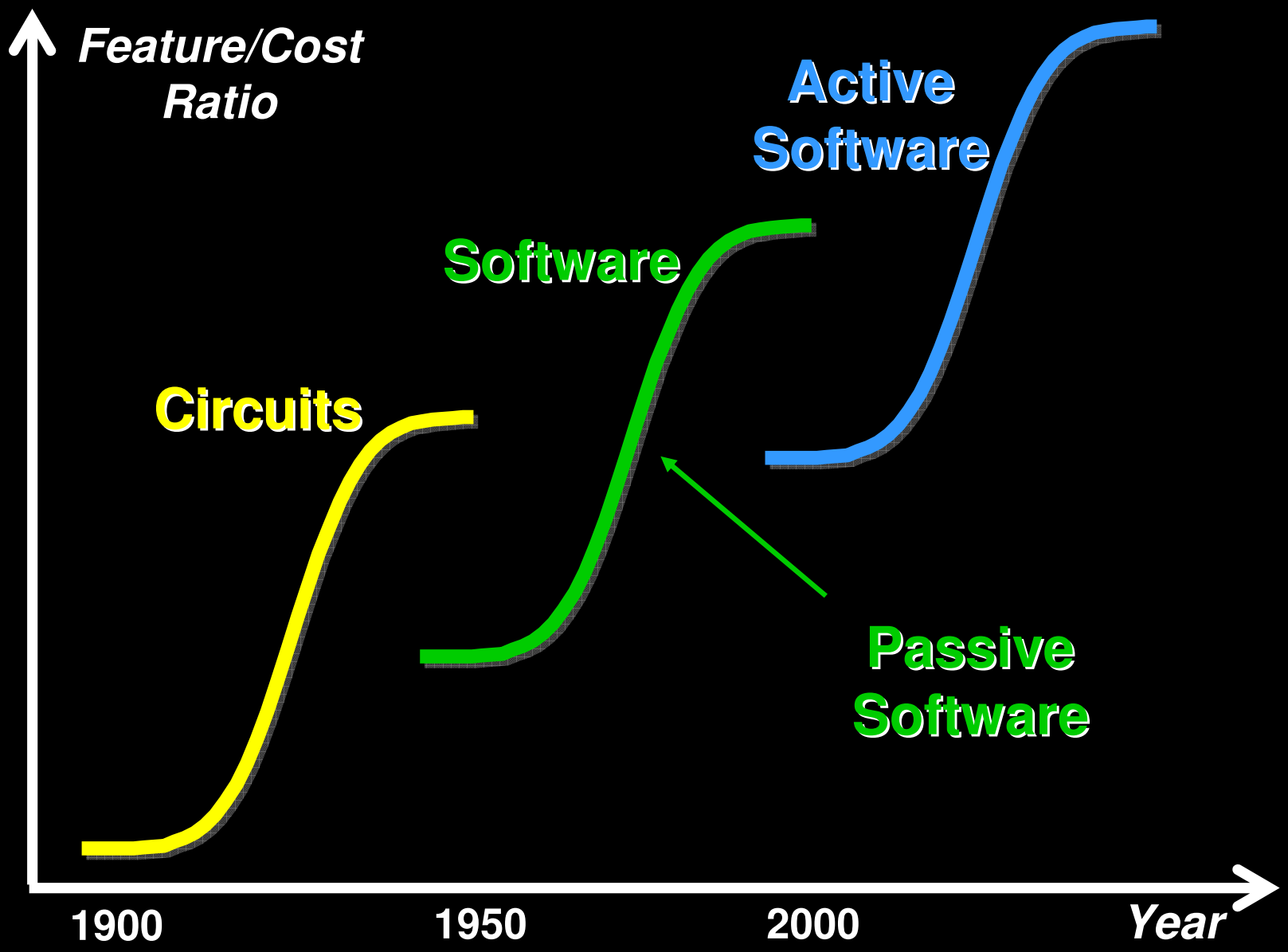
RUP is Futurized

In many ways, my personal top favorites are:

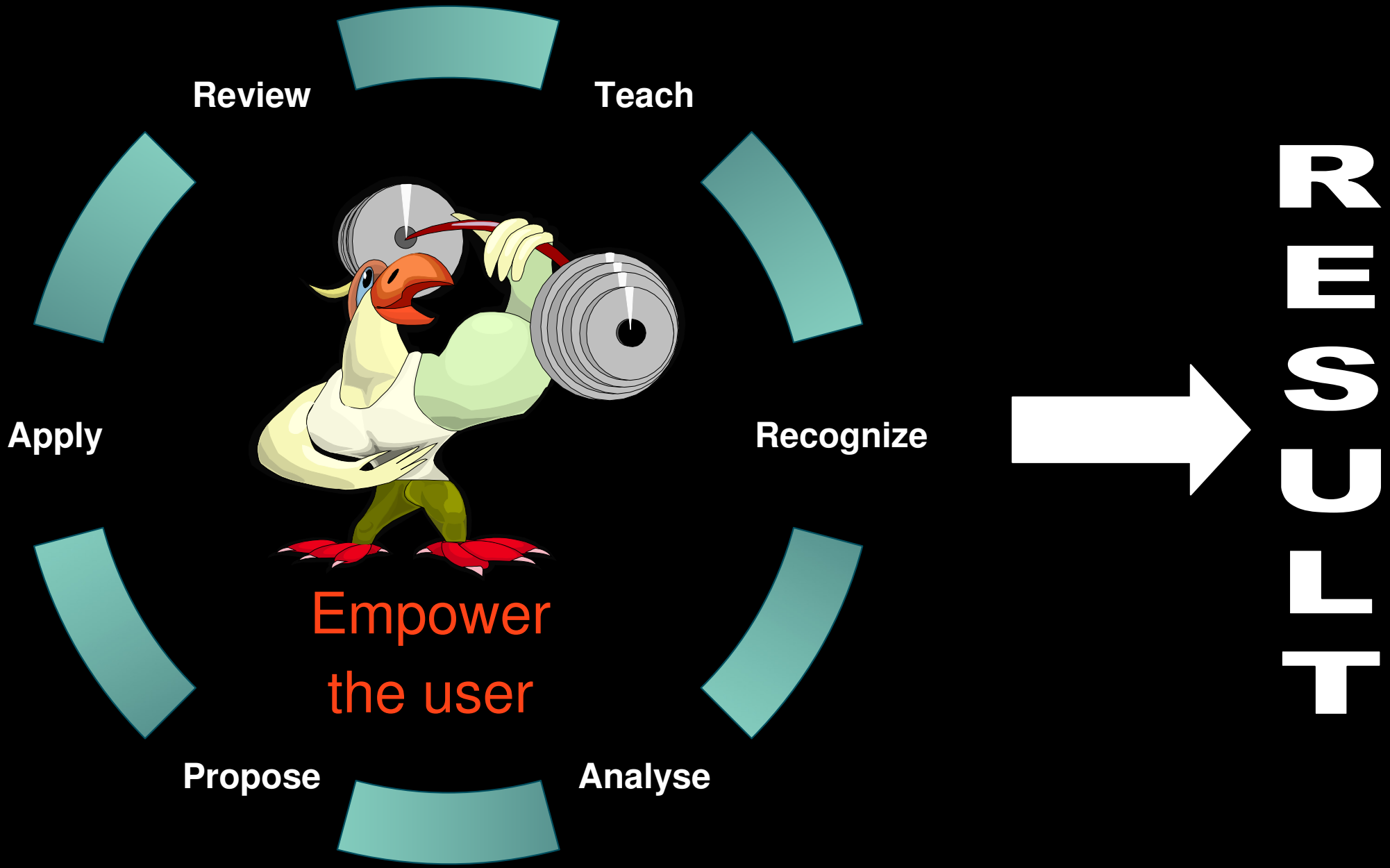
1. Making Software Active (instead of Passive)
2. Making Software Process Active
3. Building Truly Extensible Systems with Aspects

Changing the World
From Passive to Active Software

Automation in Perspective



Our Approach: Actions in Context



2: Making Software Process Active

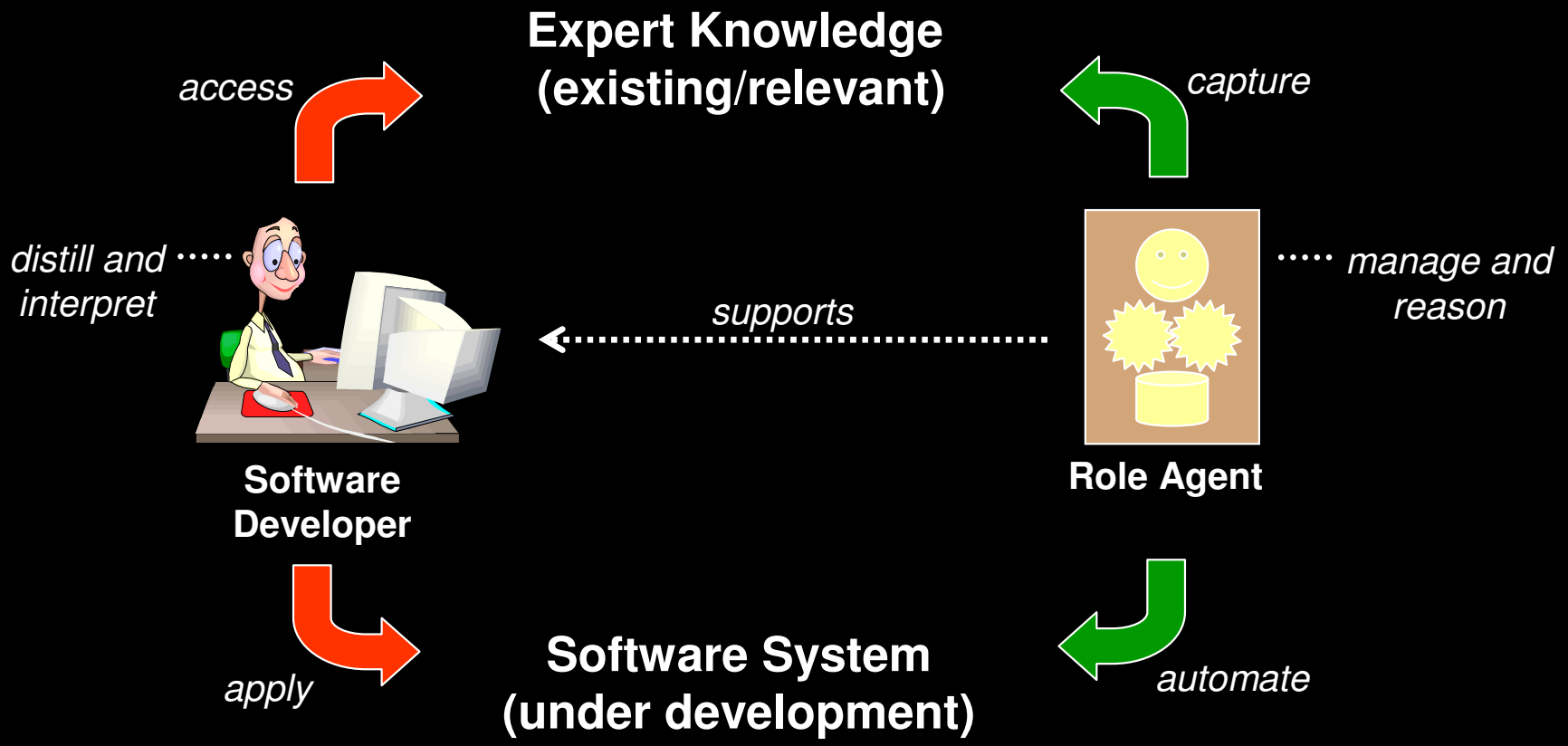
Software development has never been as hard as today – we need to SIMPLIFY

- Minimal training – learn as you go
- Make process invisible – yet very present
- Make it personal, make it light
  - Without sacrificing quality
- Give them context-dependent, concrete advices
- Make people collaborate
- Let them focus on creative tasks

Empower the developer

A process engine in the hands of each developer and the whole team.

Agents and Software Developers



Today

- Consider a new feature for a base program
  - Identify a location in the base program where you want to insert the new behavior
  - Code the new behavior and insert it at the location in the base program
- The base program has been changed. The new feature was invasive to the base.

How would AOP help?

AOP allows us to make the base program oblivious of the extensions (≈ aspects)

- AOP allows us to weave back extensions before execution
- UML has already support for extensions in use case modeling
  - we need to extend UML to allow extensions between arbitrary design and implementation elements
- AOP allows us to keep use cases separate all the way

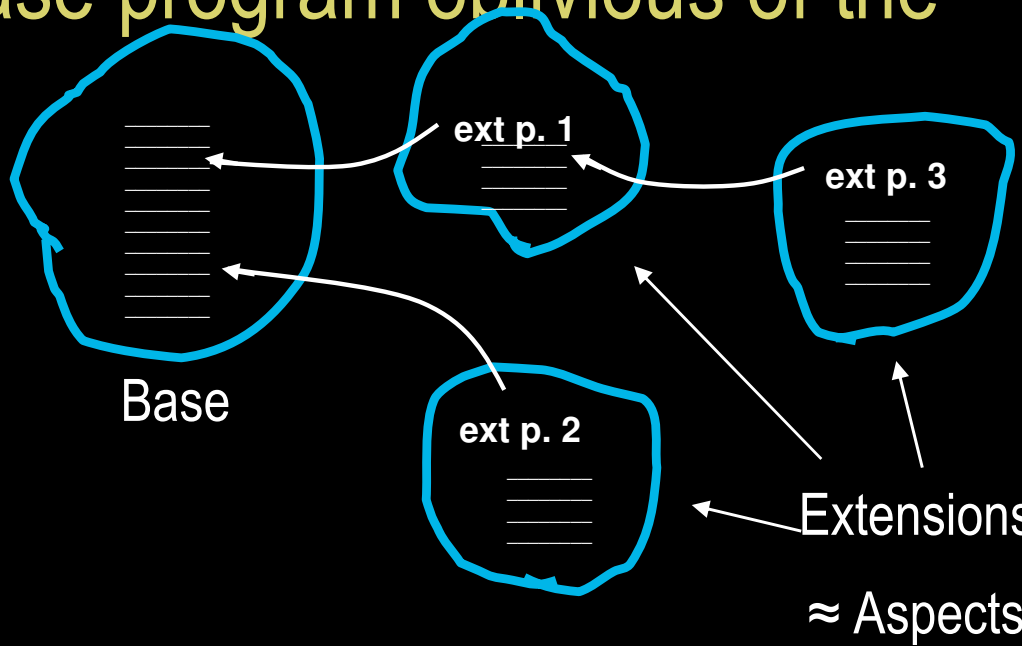


Figure from: Language Support for Changeable Large Real Time Systems, Ivar Jacobson, 1986

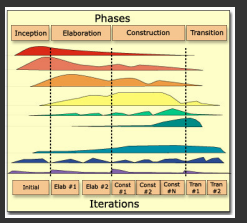
Agenda

- What you may know about RUP
- What you should know about RUP
- RUP is futurized
- RUP for outsourcing
- Summary

RUP is Designed For Outsourcing

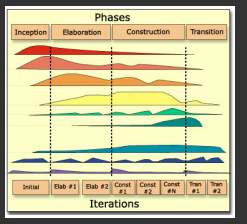
- When you outsource projects
  - You want outsourcee to deliver a quality system
    - Clear requirements
    - Sound architecture

    RUP defines these artfacts

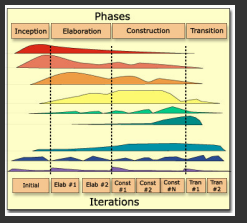
  - You want outsourcee to run project effectively
    - They know what to do
    - They know what to report

    RUP defines roles and activities

  - You want to track progress
    - Well defined milestones
    - Checkpoints and criteria

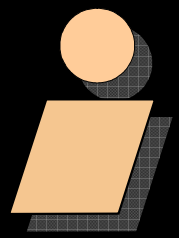
    RUP defines phases and checkpoints

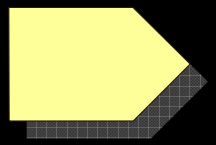
RUP Defines Roles and Activities

- RUP has well defined roles and activities
- Roles can be fulfilled by anyone

Outsourcer

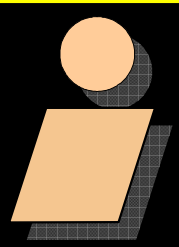


Analyst

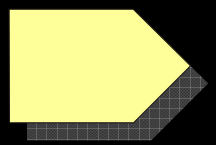


Find Actors and Use Cases

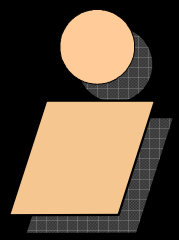
Outsourcee



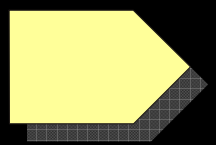
Architect



Architecture Analysis



Tester



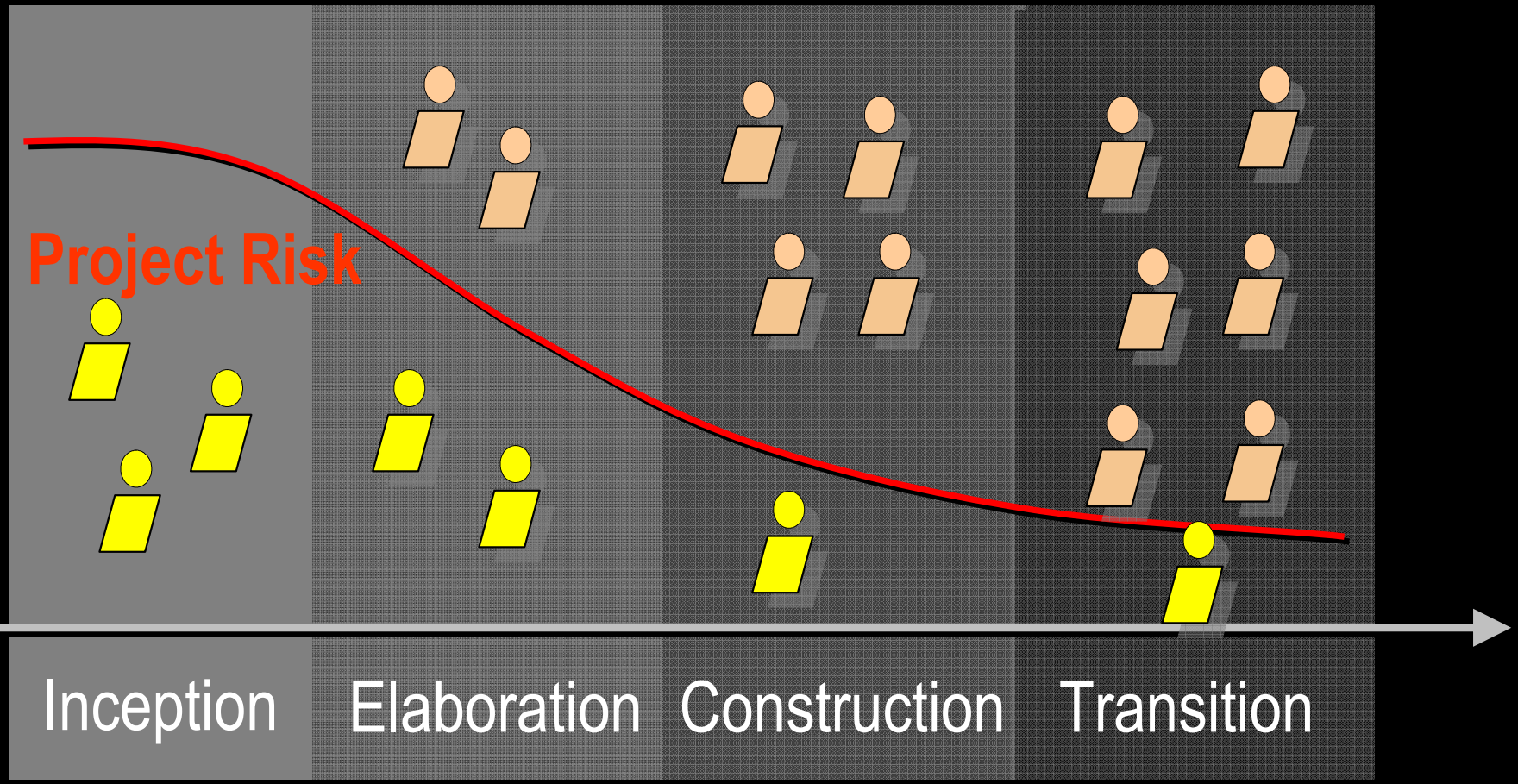
Execute Test

Simply map roles and activities to individuals or organizations

Mapping depends on outsourcee competence

RUP Helps You Balance Resources

- Resources shift over the project
- As project risk decreases, and as outsourcee acquires more know how, they can do more



You Still Need to Track the Progress Deligently

- Even if you have your own developers
  - They can run wild if left unchecked
- Problem is more apparent with outsourcee in a different geography, culture
  - You must check the work regularly
  - Artifacts grow over time, more things to review
  - Repeated errors
- You need tools to help
  - Not just IDEs, or Modeling Tools
  - But tool to ensure process is adhered
  - WayPointer

What Is WayPointer?

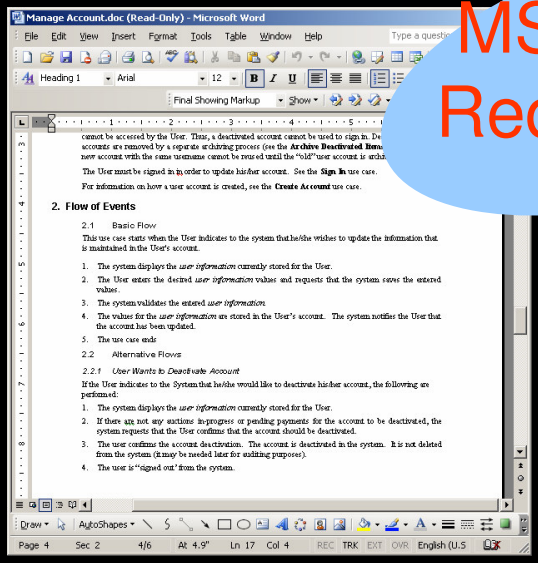
Active Guidance
to help you draft the initial artifacts

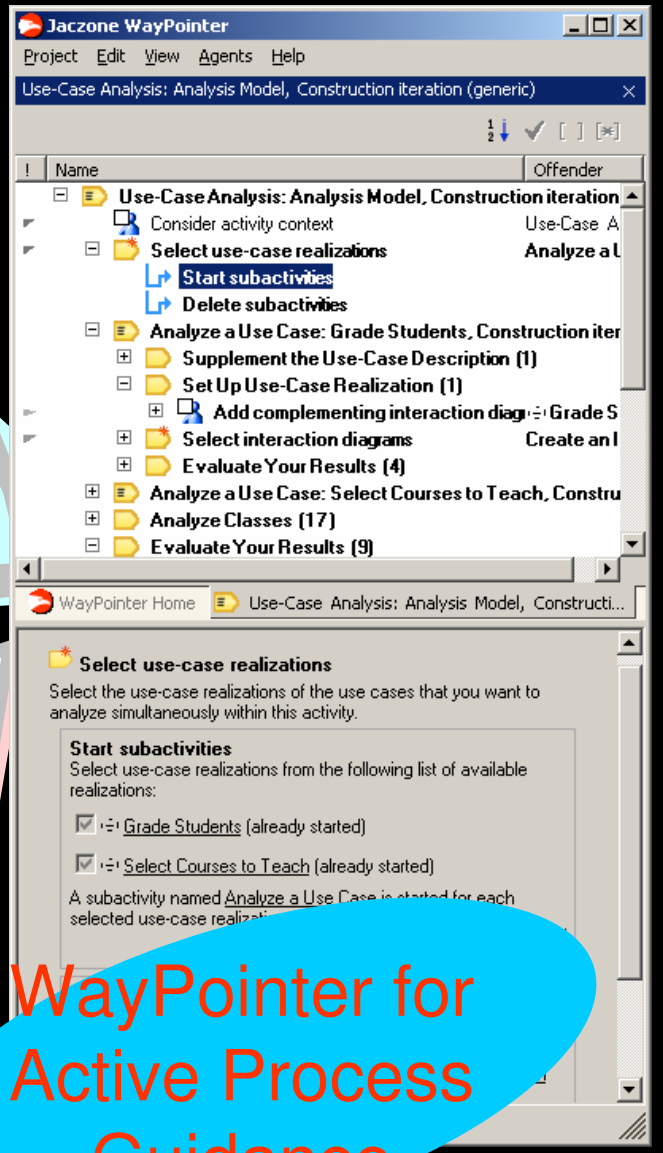
Active Observation
to you selectively focus on risk areas and conduct rigorous check

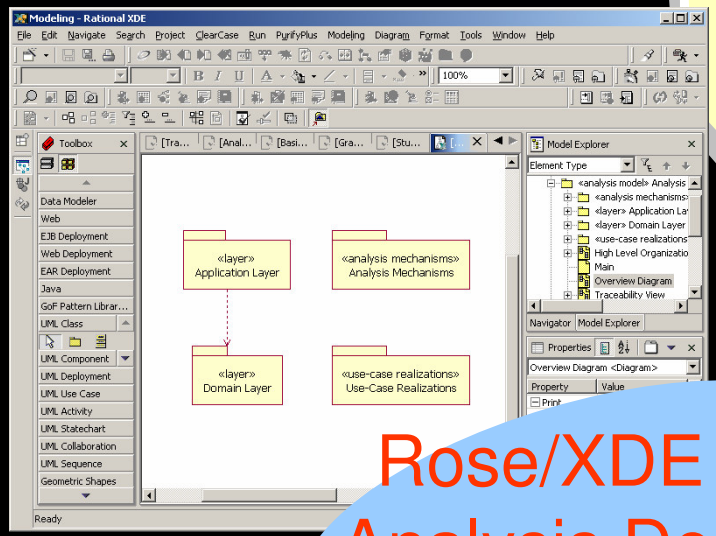
Active Facilitation
to do the mundane tasks for you

WayPointer and Your Environment

MS-Word for Requirements







Rose/XDE for Analysis Design

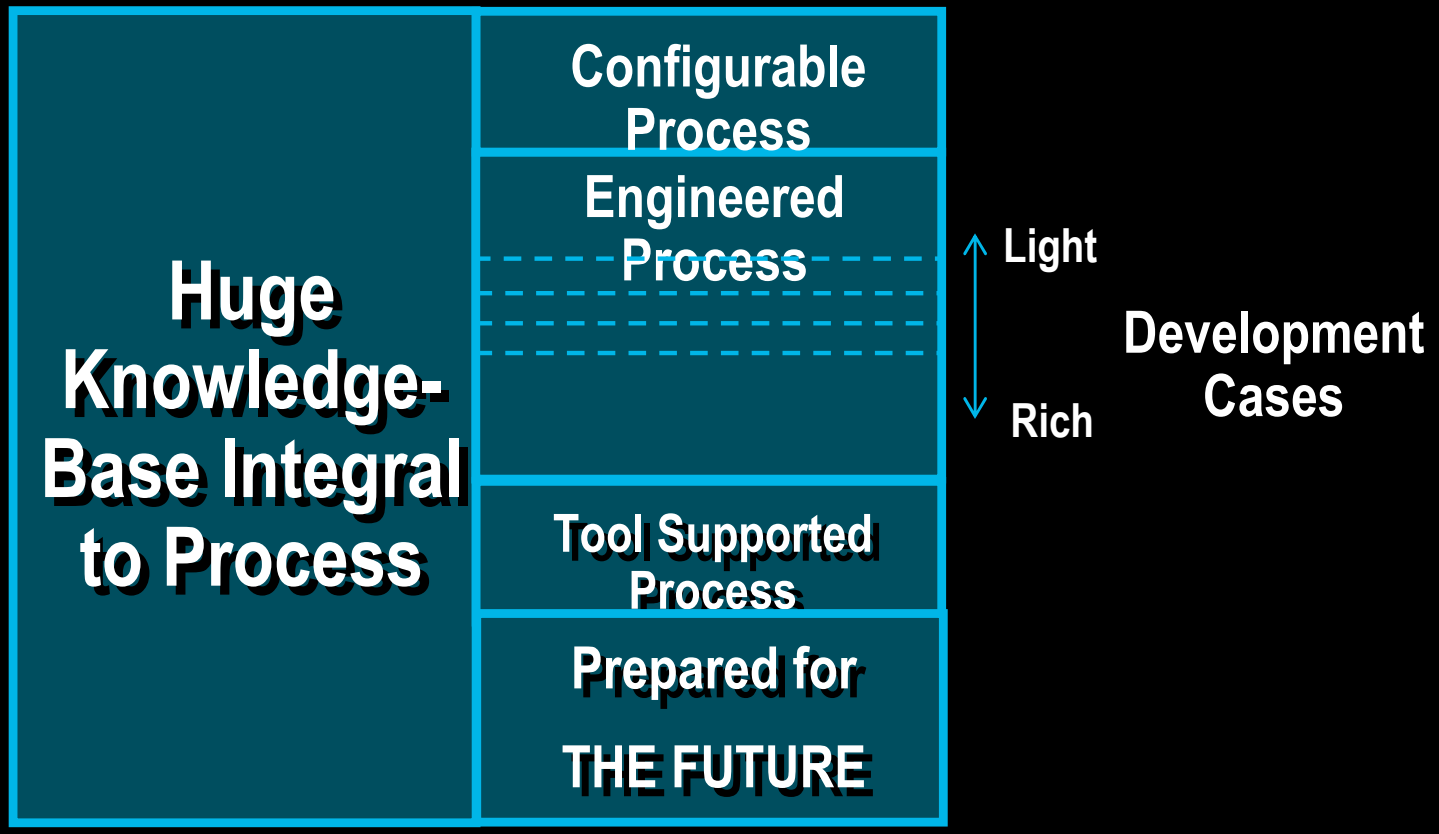
WayPointer for Active Process Guidance

Agenda

- Modern Software Development
- What you may know about RUP
- What you should know about RUP
- RUP for outsourcing
- RUP is futurized
- Summary

Summary

- RUP is many things



- A Technique For Outsourcing – it needs a tool
- WayPointer is such a tool

References

- Making Software Process Execute
  - Call for Expert Systems, Ivar Jacobson & Stefan Bylund, Application Development Trends, June 2002, http://www.adtmag.com/article.asp?id=3680
  - A Multi-Agent System Assisiting Software Developers, Ivar Jacobson & Stefan Bylund, See www.jaczone.com/papers
- Building Extensible Systems
  - Language Support for Changeable Large Real Time Systems, Ivar Jacobson, Proceedings of OOPSLA'86, pp 377-384, Sep 1986
  - Use Cases and Aspects – Working Together, Ivar Jacobson, soon to be published

Other Readings by Ivar Jacobson

- Object-Oriented Software Development--A Use Case Driven Approach (Addison Wesley)
  Jacobson et al, Addison Wesley Longman (1992)
- The Object Advantage: Business Process Reengineering with Objects (Addison Wesley)
  Jacobson et al, Addison Wesley Longman (1994)
- Software Reuse: Architecture, Process and Organization for Business Success (Addison Wesley)
  Ivar Jacobson, Martin Griss & Patrik Jonsson, Addison Wesley Longman (1997)
- Unified Software Development Process
  Jacobson, Booch, Rumbaugh, Addison Wesley Longman (1999)
- The Road to the Unified Software Development Process
  Ivar Jacobson, Stefan Bylund, Cambridge University Press, 2000

Other Unified Process Books

- *The Rational Unified Process - An Introduction*
  Philippe Kruchten (Addison Wesley)
- *Software Project Management - A Unified Framework*
  Walker Royce (Addison Wesley)
- *Visual Modeling with Rational Rose and UML*
  Terry Quatrani (Addison Wesley).

Success Stories

These stories as well as customer videos are available on Rational.com

- See: http://programs.rational.com/success/