



IBM Software Group


# The Impact of Iterative Development on Configuration Management

Rational software



business on demand software


IBM Software Group | Rational software




## Introduction

Mark Roberts  
Technical Consultant  
Rational Software : IBM Software Group


[mark\\_roberts2@uk.ibm.com](mailto:mark_roberts2@uk.ibm.com)




IBM Software Group | Rational software 

## Agenda


- wThe Software Crisis
- wIterative Development Principles
- wCommon Pitfalls of Iterative Development
- wImplementing SCM Best Practice in an Iterative Process




IBM Software Group | Rational software 


## Agenda

- wThe Software Crisis
- wIterative Development Principles
- wCommon Pitfalls of Iterative Development
- wImplementing SCM Best Practice in an Iterative Process




IBM Software Group | Rational software 

## Software is everywhere ...




The image shows a grid of various objects and scenes, each representing a domain where software is used. The objects include a PDA, a laptop, a mobile phone, a car, a jet fighter, a satellite tower, and a human ear with a hearing aid. The background is a dark grid pattern.

IBM Software Group | Rational software 

## The Software Crisis (1995)

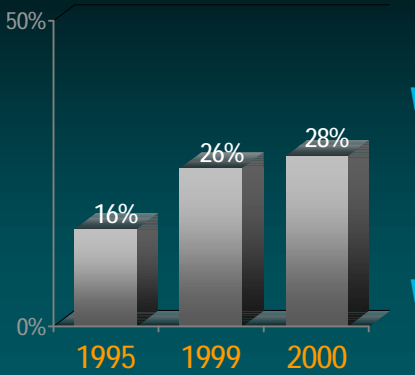
- w 1995 Standish Group study
  - § U.S. companies will spend \$81 billion for canceled software projects
  - § 31% will be canceled before they are completed
  - § Over 50% will cost more than twice the original estimate
  - § Only 16% will be delivered on time and within budget
- w Recommended practices
  - § Executive management support
  - § Clear requirements
  - § Proper planning
  - § User involvement

IBM Software Group | Rational software 

## Software Development: *Current State*

### Many Companies


#### Software Project Success



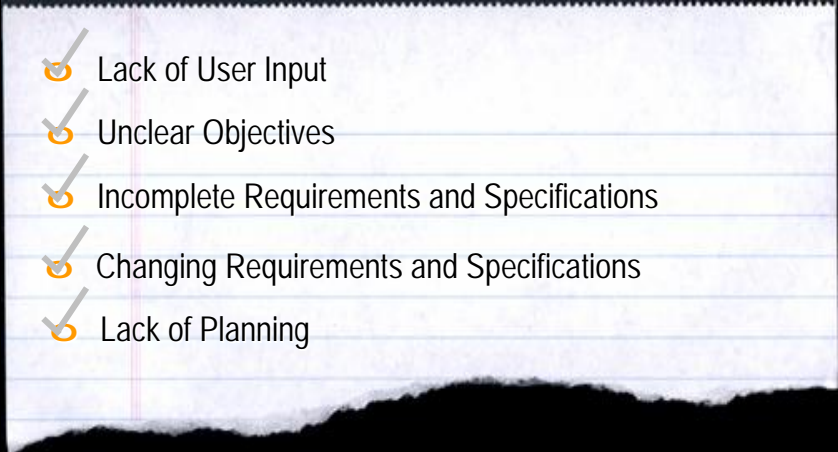
Year	Success Rate
1995	16%
1999	26%
2000	28%

Source: Standish Report

- w** Over budget
  - \$** Avg. cost overrun: 189%
- w** Delivered late
  - \$** Avg. schedule overrun: 222%
- w** Failed to meet expectations
  - \$** Avg. coverage: 61%


IBM Software Group | Rational software 

## Why Do Projects Fail?




- Lack of User Input
- Unclear Objectives
- Incomplete Requirements and Specifications
- Changing Requirements and Specifications
- Lack of Planning


Source: Standish Report

IBM Software Group | Rational software 

## The 80/20 Rule

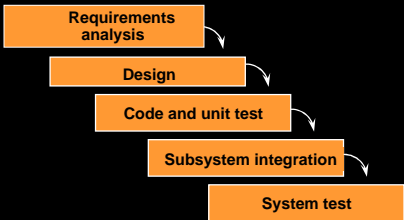
- w** 80% of the engineering is consumed by 20% of the requirements.
- w** 80% of the development cost is consumed by 20% of the components.
- w** 80% of the errors are caused by 20% of the components.
- w** 80% of development phase scrap and rework is caused by 20% of the errors.
- w** 80% of the resource consumption (execution time, disk space, memory) is consumed by 20% of the components.
- w** 80% of the engineering gets accomplished by 20% of the tools.
- w** 80% of the progress is made by 20% of the people.




IBM Software Group | Rational software 

## Conventional software process issues

**Conventional Process**



- w** Delays confirmation of critical risks
- w** Measures progress by the wrong measures
- w** Can't deploy sooner
- w** Usually major unplanned repeats of this process




IBM Software Group | Rational software **IBM**

## The Symptom of the problem : *Focus of effort*

**W** Conventional resource allocations (time and effort) :

Management	5%
Requirements	5%
Design	10%
Coding and unit test	30%
<b>Integration and test</b>	<b>40%</b>
Deployment	5%
<u>Environment</u>	<u>5%</u>
Total	100%

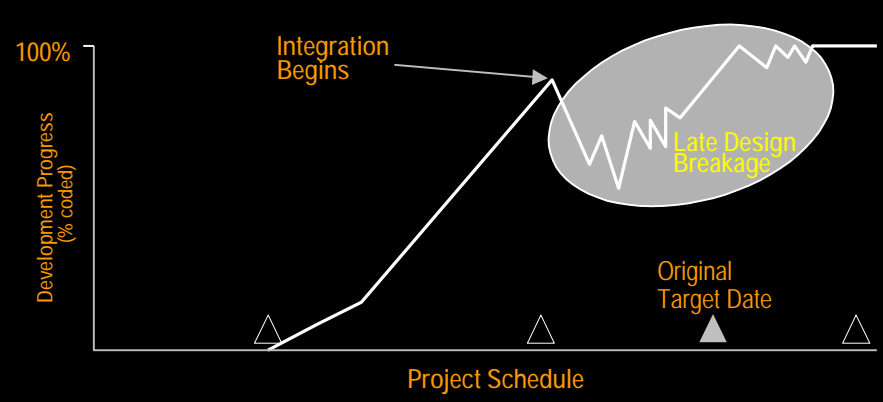
**W** A recurring result of conventional processes:  
**\$** Integration/test consumes over 40% of the life-cycle resources



IBM Software Group | Rational software **IBM**

## Conventional software process

Sequential activities:  
Requirements → Design → Code → Integration → Test



100%


Development Progress (% coded)


Integration Begins

Late Design Breakage

Original Target Date


Project Schedule




IBM Software Group | Rational software 

## Agenda

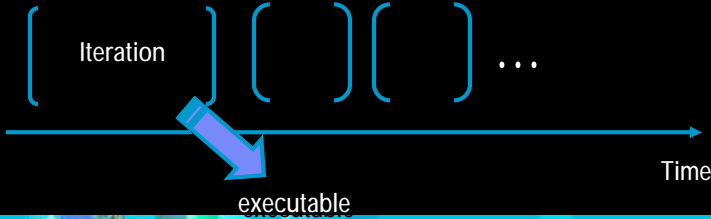
- w The Software Crisis
- w Iterative Development Principles
- w Common Pitfalls of Iterative Development
- w Implementing SCM Best Practice in an Iterative Process




IBM Software Group | Rational software 


## What is iterative development?

- w Iterative development is developing the project in a number of iterations
- w An iteration results in working, tested executable code
- w An iteration covers the normal development lifecycle, every iteration




executable




IBM Software Group | Rational software 


## Why iterative development?

- W An initial design will likely be flawed with respect to its key requirements
- W Late-phase discovery of design defects results in costly over-runs and/or project cancellation



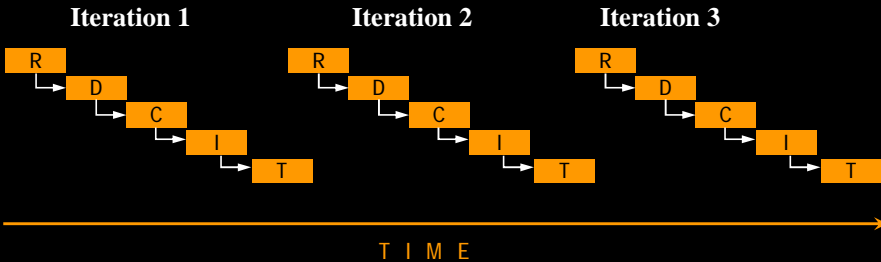
*The time and money spent implementing a faulty design are not recoverable*



IBM Software Group | Rational software 

## Iterative development characteristics


- W Resolves major risks before making large investments
- W Enables early user feedback
- W Makes testing and integration continuous
- W Focuses on short-term, objective milestones
- W Makes possible deployment of partial implementations

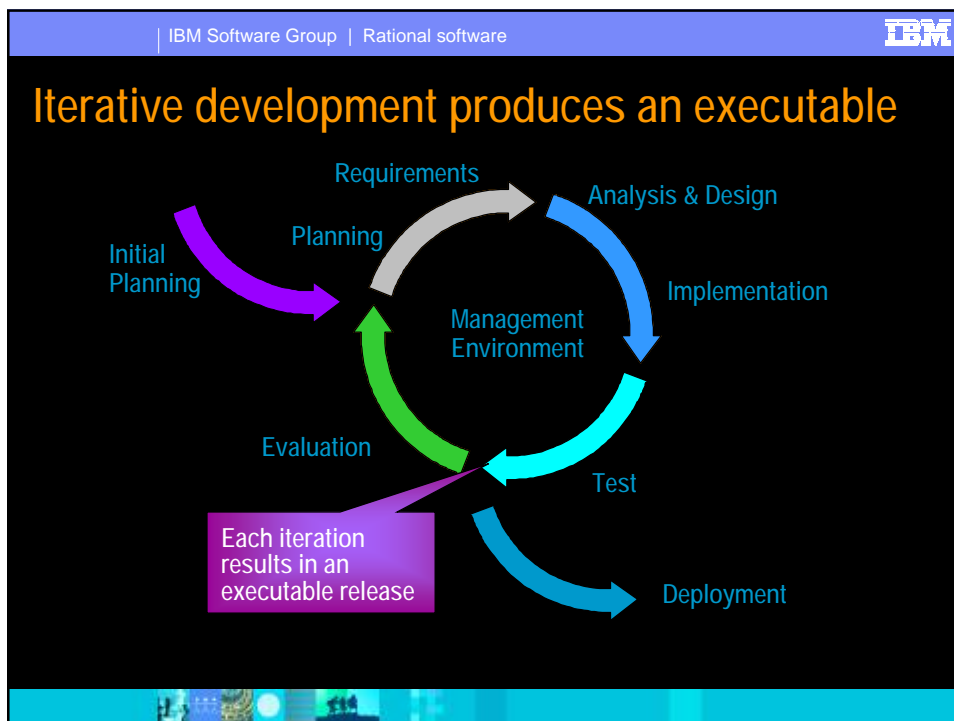
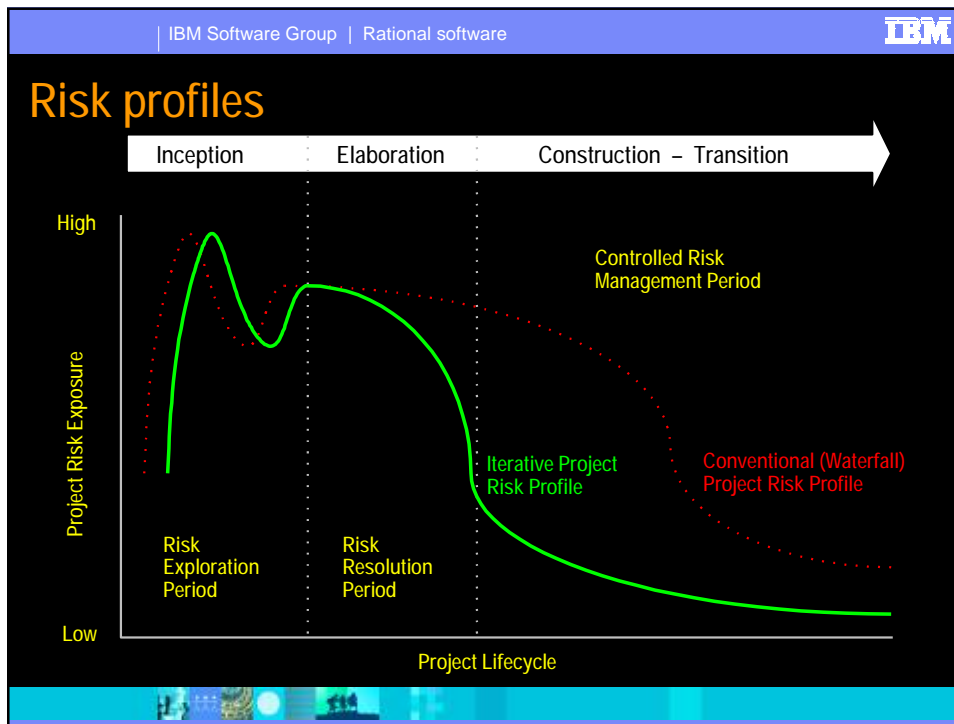



Iteration 1      Iteration 2      Iteration 3

R   D   C   I   T      R   D   C   I   T      R   D   C   I   T

TIME







IBM Software Group | Rational software 

### Iterative vs Conventional : *Resource allocation*


	Conv.	Iterative	Change
Management	5%	10%	+5
Requirements	5%	10 %	+5
Design	10%	15 %	+5
Coding and unit test	30%	25 %	-5
Integration and test	40%	25 %	-15
Deployment	5%	5 %	0
Environment	5%	10 %	+5
<u>Total</u>	<u>100%</u>	<u>100 %</u>	




IBM Software Group | Rational software 

### Agenda


- w The Software Crisis
- w Iterative Development Principles
- w Common Pitfalls of Iterative Development
- w Implementing SCM Best Practice in an Iterative Process




IBM Software Group | Rational software 

## Common Pitfalls of Iterative Development


- w** Perfect is the enemy of good ...
  - §** Focus on delivering the requirements, don't embellish
  - §** Organise components within CM repositories
  - §** Baseline code at the end of an iteration
  - §** Expect many iteration baselines prior to code completion
  - §** Lock completed components
  - §** Ensure file changes are activity based
  - §** Manage and prioritise activities
  - §** Consider the global impact of change




IBM Software Group | Rational software 

## Common Pitfalls of Iterative Development


- w** Let's Start ...
- w** Set clear goals for development
  - §** Work to specific high risk requirements
  - §** Validate work by testing early
- w** Continually validate product with stakeholders
  - §** Ensure stakeholders delegate problem resolution
  - §** Don't accept delegation of problem definition




IBM Software Group | Rational software 

## Common Pitfalls of Iterative Development


- wVictim of your own success
  - § Accept change will happen
  - § Manage all change requests aggressively
  - § Avoid requirements creep
  - § Avoid priority creep
- wAvoid snatching defeat from the jaws of victory




IBM Software Group | Rational software 

## Agenda


- wThe Software Crisis
- wIterative Development Principles
- wCommon Pitfalls of Iterative Development
- wImplementing SCM Best Practice in an Iterative Process




IBM Software Group | Rational software 

## Best Practices of SCM - 1


- w** *Identify and Store Artifacts in Secure Repository*
  - §** *All project artefacts*
  - §** *(source code, plans, test scripts, results, deliverables)*
- w** *Organise component repositories earlier in project*
  - §** *Plan for flexibility, expect change*
- w** *Demand project directory structure is defined*
  - §** *Development tools may suffer without this*
- w** *Consider reuse early and identify key assets*
- w** *Get development tools in place early*



IBM Software Group | Rational software 

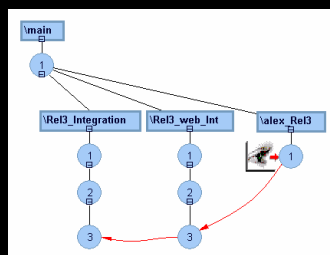
## Best Practices of SCM - 2

- w** *Control and Audit Changes to Artifacts*
  - §** *Who changed what where and when ?*
- w** *Iterative Approach*
  - §** *Change control board to assess changes against requirements*
  - §** *Assess architectural risk and project stability risks*
  - §** *Clearly record the iteration in which any work was done*
  - §** *Audit change to look for weakness and training opportunities*
  - §** *Improve planning ability – What can be achieved in an iteration*




## Best Practices of SCM - 3

- w** *Maintain Stable and Consistent Workspaces*
  - § *Balance information sharing with isolated workspaces*
- w** *Workspace management capability is important*
  - § *Empower developers to manage their environment*
- w** *Instantly switch between configurations*
  - § *Developers latest development stream*
  - § *Current integration stream*
  - § *Last iteration baseline*




## Best Practices of SCM - 4


- w** *Support Concurrent Changes*
  - § *Parallel development*
- w** *Avoid parallel iterations*
  - § *Getting commitment to complete prior iterations is difficult*
  - § *Problems get pushed to next iteration rather than solved*
  - § *No time for review, or assessment of new requirements*
  - § *Use parallel development within iterations NOT for iterations*
  - § *Define the parallel development structure*
  - § *Use additional branch annotation to identify iteration*

IBM Software Group | Rational software 

## Best Practices of SCM - 5


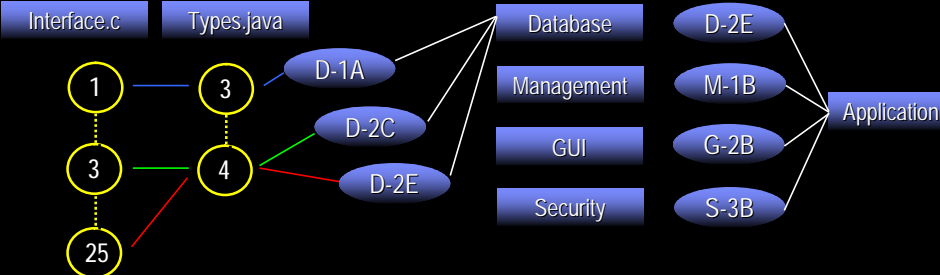
- w** Integrate Early and Often
  - §** Drive an iterative process with frequent integration
- w** Plan the iterative approach
  - §** Stay within iteration timeframe – do not allow slippage
  - §** Complete parallel development integration within the iteration
- w** Integrate and test
  - §** Complete component integration prior to iteration completion
  - §** Regression test all tests from previous iterations




IBM Software Group | Rational software 

## Component Baseline : Component Integration

- w** Baseline the components to identify a collection of files
  - §** 3 Baselines applied to the Database component
- w** Integrate components to create application
  - §** Select appropriate baselines of the components
  - §** Use baseline 'quality' or 'lifecycle' annotation to ensure selection is right





IBM Software Group | Rational software 

## Best Practices of SCM - 6

- w** *Organize Artifacts into Versioned Components*
  - §** *Reduces complexity, promote sharing and reuse*
- w** *Catalogue reusable components*
  - §** *Identify status within iterative lifecycle*
- w** *Ensure 'reuse clients' understand iterative approach*
  - §** *Component baseline does not identify component completion*
  - §** *Define a baseline naming convention*


0.1 0.2 0.3 | 1.0 | 1.01  
Pre release iterations | First release | Enhanced or fixed release




IBM Software Group | Rational software 

## Best Practices of SCM - 7


- w** *Record and Track Requests for Change*
  - §** *Categorise change drivers for workflow*
- w** *Risk driven processes*
  - §** *Assess risk – use impact analysis techniques*
  - §** *Assess business impact to customer*
  - §** *Avoid requirement creep*
- w** *Manage Change*
  - §** *Select the appropriate iteration to make the changes*
  - §** *Assess all outstanding changes at iteration boundary*




IBM Software Group | Rational software 

## Best Practices of SCM - 8


- w** *Organize and Integrate Using Activities*
  - §** *Activities drive change*
- w** Scenario 1 - During the last iteration ...
  - §** ... 789 Files have been modified between
- w** Scenario 2 - During the last iteration ...
  - §** 50 Activities have been completed
  - §** 40 defects fixed
  - §** 10 requirements implemented




IBM Software Group | Rational software 

## Best Practices of SCM - 9


- w** *Create Baselines at Project Milestones*
  - §** *Identify versions of assets at specific points*
- w** Define baselines
  - §** Iteration completion
- w** Baselines are more than blanket labels
  - §** Store baseline deliverables and build audit trails
  - §** Enable regression testing of the iteration
- w** Satisfy customer reproducibility requirements



IBM Software Group | Rational software 

## Best Practices of SCM – 10

- w**Ensure Reproducibility of Software Builds
  - §** Audit builds for recording and comparing
- w**Evolve a build structure
  - §** Create an expandable build environment
  - §** Provide version management of build files and key deliverables
  - §** Plan for extensive regression testing



IBM Software Group | Rational software 

## Iterative Development : Underpins Best Practice

<b>Best Practices</b> <i>Process Made Practical</i>	<b>w</b> All process impacts CM
Develop Iteratively	<b>§</b> Define a process and communicate !
Manage Requirements	<b>w</b> CM impacts all development
Use Component Architectures	<b>§</b> Minimize developer intrusion
Model Visually (UML)	<b>§</b> Gather useful information, not data
Continuously Verify Quality	
Manage Change	

- w**Iterative Development drives all process
  - §** Implement an iterative process and other best practice follows
  - §** More management, less code and integration, reduced risk !!



