

ATSC 03

Requirements Management

Information Systems Engineering & Advanced Topics in Software Construction

Prof. Armin B. Cremers, Prof. Rainer Manthey
Stefan Lüttringhaus-Kappel, Günter Kniesel, Tobias Rho,
Serge Shumilov, Daniel Speicher, Volker Steinhage



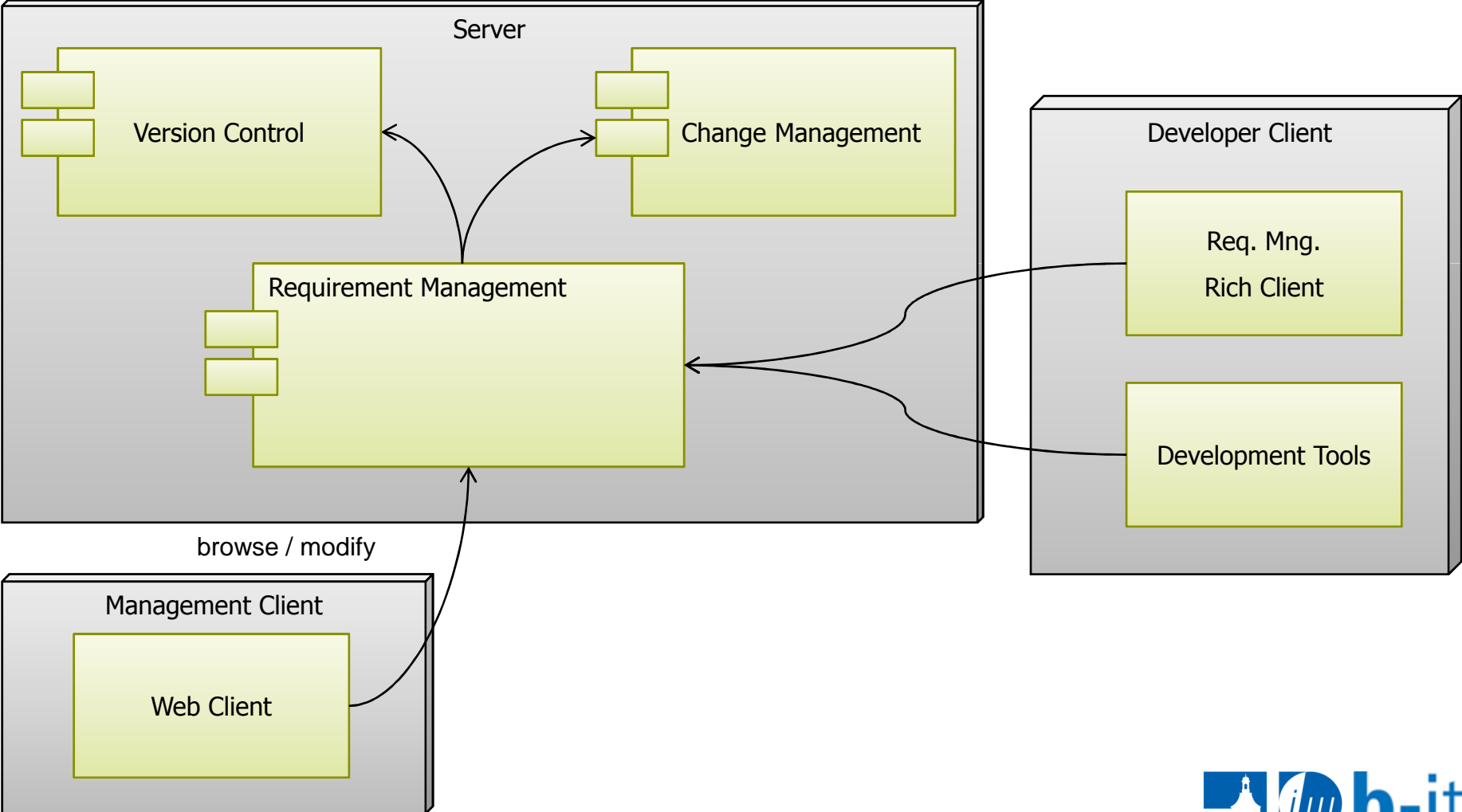
Overview

- ◆ What is Requirements Management?
- ◆ Traceability of Requirements
- ◆ Storing Requirements, Requirements Tools
- ◆ Requirement Change Management

What is Requirements Management?

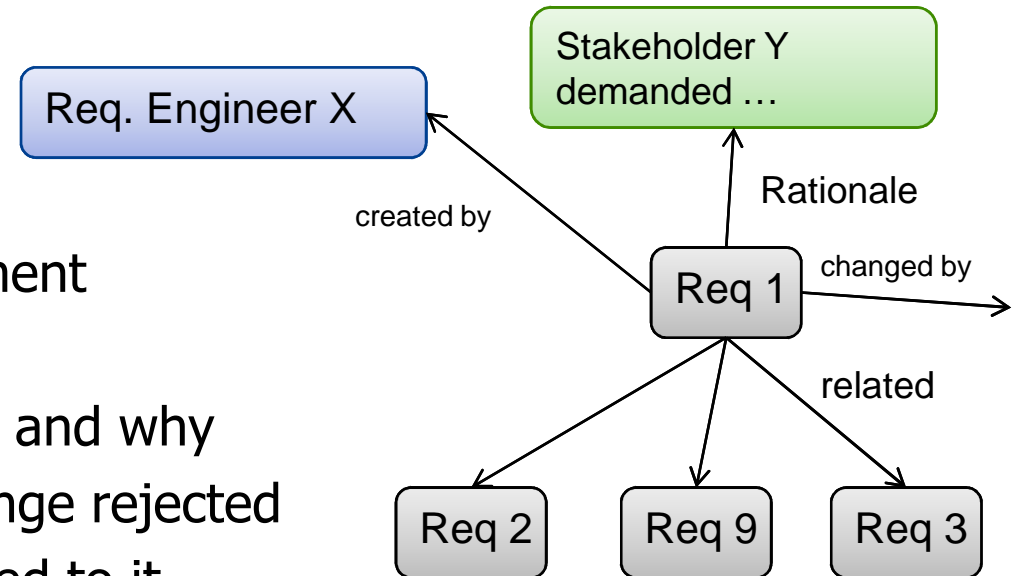
- ◆ The process of managing the changes of requirements
- ◆ Hard Problem because of continuous changes during development process
- ◆ Managing
 - ◆ the relationships between requirements
 - ◆ priorities between requirements
 - ◆ the dependencies between different documents (traceability)
 - ◆ requirements document
 - ◆ specification
 - ◆ and other documents produced in the systems engineering process
 - ◆ Versioning
 - ◆ Refinement
 - ◆ Baselining
 - ◆ changes to agreed requirements
 - ◆ Branches

Requirements Repository



Traceability

- ◆ Requirements cannot be managed effectively without traceability
 - ◆ Interrelations
 - ◆ Rationale
- ◆ Traceable means knowing
 - ◆ Who suggested the requirement
 - ◆ Why the requirement exists
 - ◆ Who changed a requirement and why
 - ◆ Why was a requirement change rejected
 - ◆ What requirements are related to it
 - ◆ Attributes for the relation



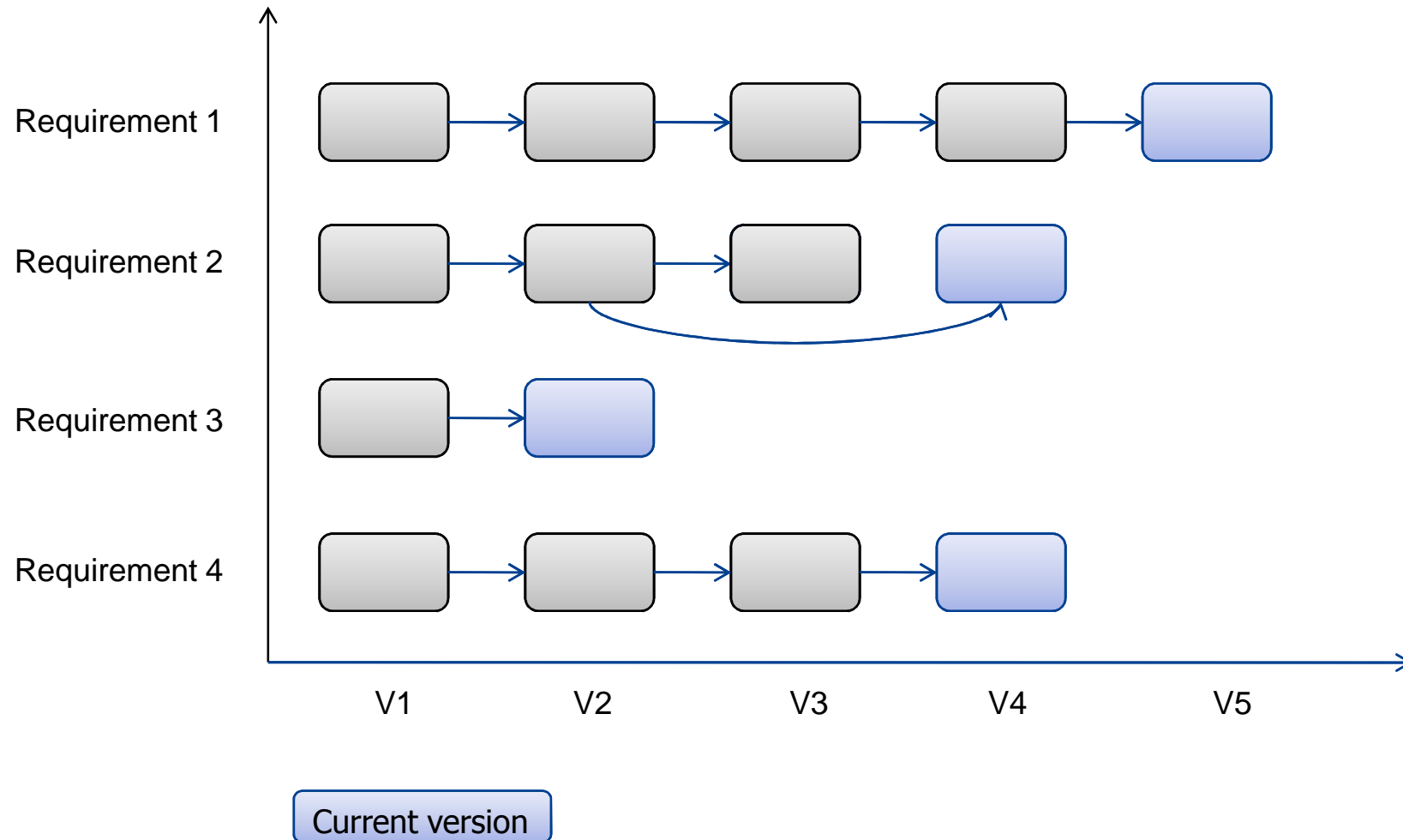
Additional types of traceability

- ◆ Requirements-architecture traceability
 - ◆ sub-systems where these requirements are implemented
 - ◆ important when sub-systems are developed by different sub-contractors
- ◆ Requirements-design traceability
 - ◆ specific hardware or software components in the system
- ◆ Requirements-interface traceability
 - ◆ interfaces of external systems

Versioning

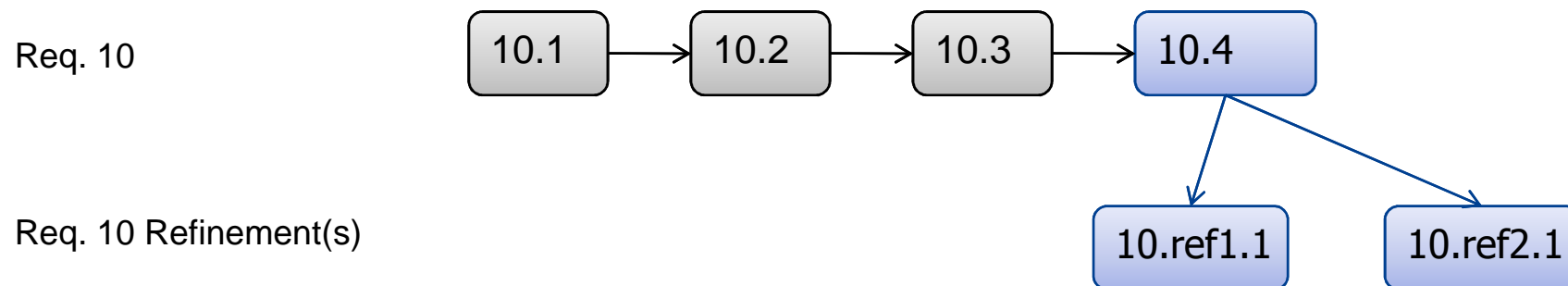
- ◆ Replace an old version of a requirement with a new version
- ◆ Old version
 - ◆ Invalidated
 - ◆ Kept in the history
- ◆ Typically a linear chain or a tree in the case of branching
- ◆ Tracing of
 - ◆ Who changed the requirement
 - ◆ When
 - ◆ Why

Versioning Example



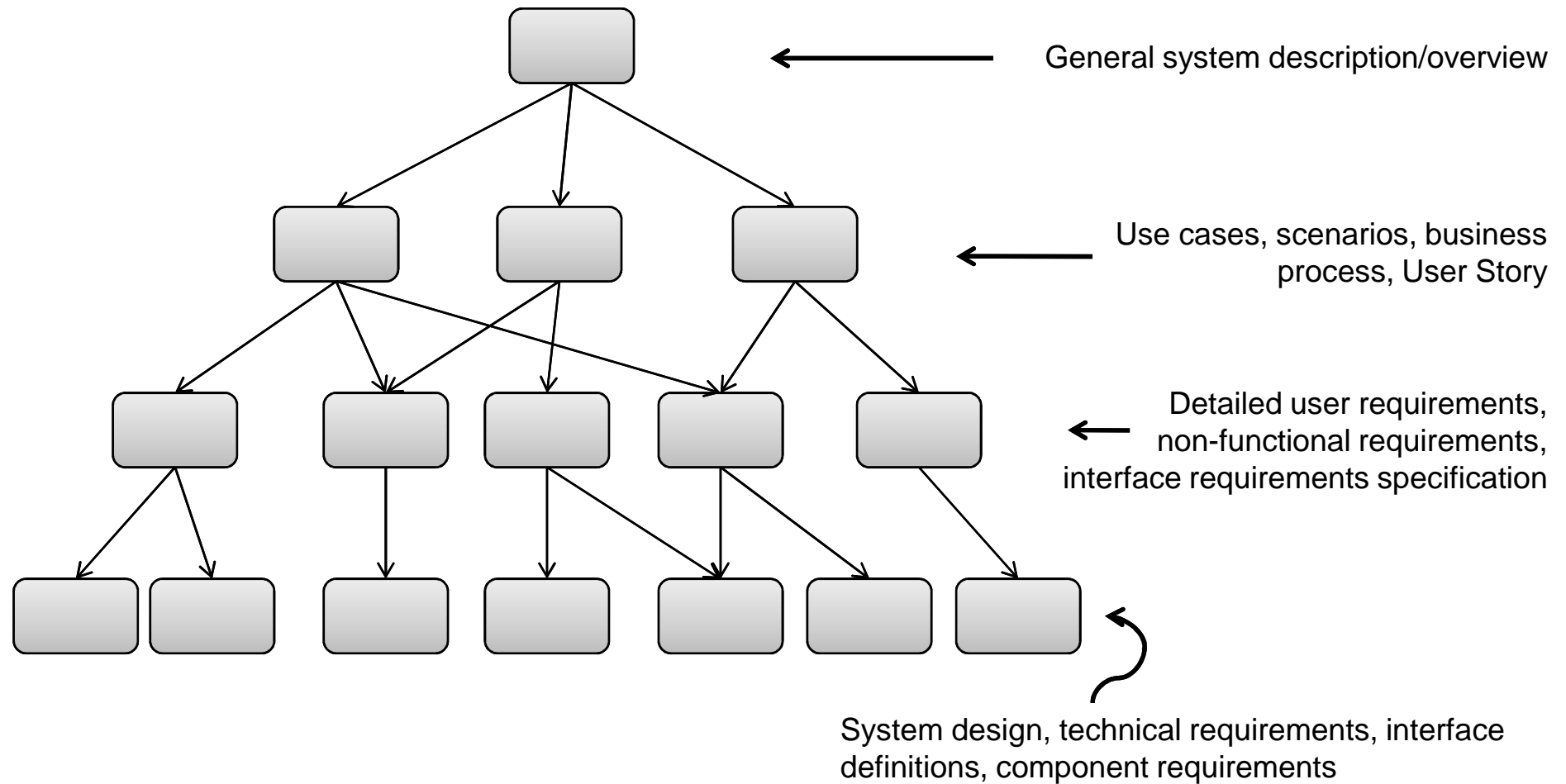
Refinement

- ◆ Methodical difference to Versioning
 - ◆ Original requirement and refined requirement are both valid at the same time
 - ◆ Refined version has lower abstraction level
- ◆ Different refinement levels for different stakeholders
- ◆ Typically a tree structure
 - ◆ Only possible if the perspective is kept (domain model)



Refinement

Example Change of Perspective



Versioning vs. Refinement

“in order to” Rule

- ◆ Simple check if a new version of a requirement is a new version or a refinement
- ◆ Is it possible to connect two requirements with “in order to”?
 - ◆ Yes -> refinement
 - ◆ No -> new version
- ◆ Example 1
 - ◆ R1: The system should allow the managers to add new data.
 - ◆ R2: The system should allow the managers to add new employee data.
 - ◆ “In order to add new data the system should allow to add new employee data” (!?)
- ◆ Example 2
 - ◆ R3: The system should allow a guest to type in registration data.
 - ◆ R4: The system should allow a guest to type in his/her name, address and bank account data.
 - ◆ “In order to type in registration data the system should allow the input of name, address and bank account data.”
 - ◆ Example for a replacement of a term by its components

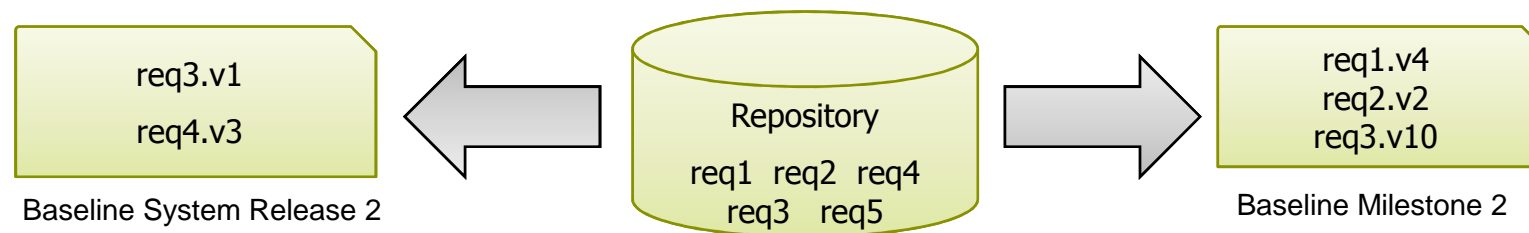
Versioning vs. Refinement

General Comments

- ◆ Refinement is the more complex step
 - ◆ In tools and practice
- ◆ Often only versioning is used
- ◆ But, without refinements
 - ◆ later references to/refinements of the original requirement are not possible
 - ◆ grouping in specification abstraction levels is not possible
- ◆ Using refinements where versioning is suitable leads to inconsistency
- ◆ Applying “in order to” helps to find the right type

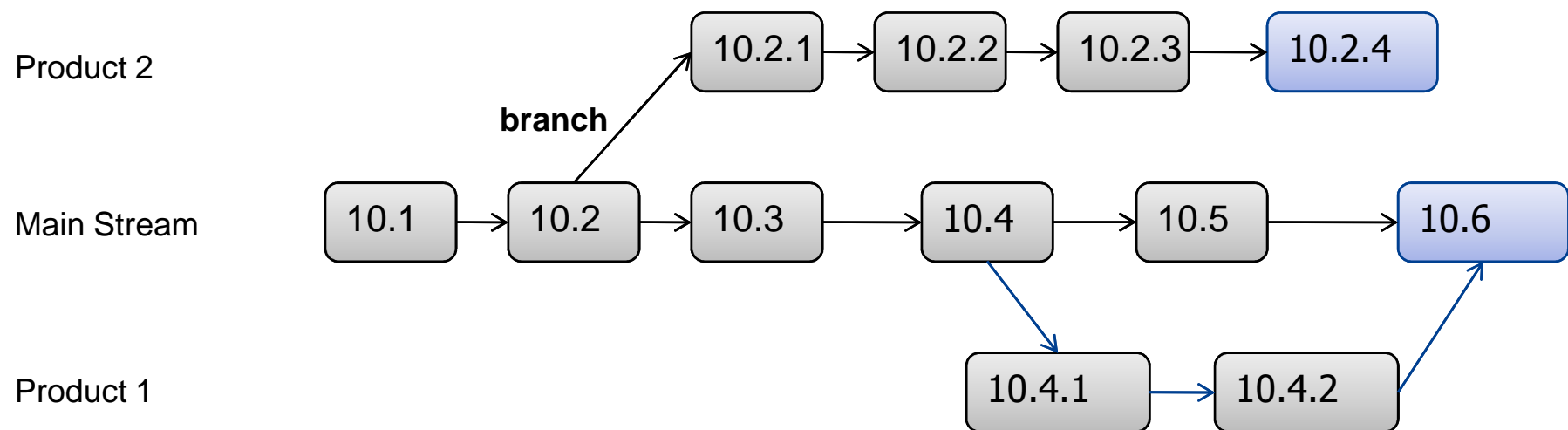
Baseline

- ◆ Consistent set of requirements at a certain point in time
- ◆ Aligned with
 - ◆ Completion of the specification (or defined parts of it)
 - ◆ Milestones
 - ◆ System releases
- ◆ Most times only contain parts of the specification
 - ◆ Inconsistent or incomplete parts are typically removed
 - ◆ If the baseline was created for a certain stakeholder
- ◆ Base for
 - ◆ discussions / implementation for ongoing projects
 - ◆ Rollbacks / Reproductions if further development went into a wrong direction



Branching

- ◆ Copy of the requirement set with (small) differences
- ◆ May be merged later into the main branch (trunk)
- ◆ Often used in *main-stream* development



Visualizing Relationships Traceability tables

- ◆ Show relationships between requirements or between requirements and design components
- ◆ Requirements are listed along the horizontal and vertical axes and relationships between requirements are marked in the table cells

The screenshot displays the Rational RequisitePro interface. On the left is a tree view of the project structure, including folders for Coverage Analysis, Design, Features and Vision, Glossary, Stakeholder Requests, Supplementary Requirements, and Use Cases. The 'Use Cases' folder is expanded to show 'Bid on Item', 'Register', and 'Sell Item' with their respective sub-use cases and flows.

The main window shows a traceability table with the following structure:

Relationships: - all displayed	FEAT1: Auction site users (the Sellers and Buyers) should be able to create accounts with the...	FEAT2: The Seller should be able to create an auction by describing an item or service s/he wants to sell and...	FEAT3: Any user should be able to browse a catalog of open auctions, read auctioned item or...	FEAT4: The Seller should be able to reject bids (for whatever reason that may be) or accept a winning...	FEAT5: The Auction Administrators should be able to close auctions without accepting any bids...	FEAT6: The Buyer should be able to enter a bid on an item. The Buyer should be able to cancel his/her...	FEAT7: The system should support system security and in particular: J2EE authentication users	FEAT8: The Auction Delight should be deployed on a RI-compliant J2EE deployment environment	FEAT9: The application should be developed using the Rational product suite and should follow the...	FEAT10: The system does not have specific performance requirements beyond those commonly...
UC1: Bid on Item A buyer can place a bid on an item.						↗	↗			
UC1.1: Brief Description When browsing an item currently available via auction, a Buyer may opt to place a bid on the item. Once accepted the entered bid becomes the...	↗									
UC1.2: Basic Flow Basic Flow										
UC1.2.1: BID This use case starts when the system presents options available to the Buyer. These options are "bid on item" and "view bid history"...			↗							
UC1.2.2: ENTER AMOUNT The Buyer enters the bid amount. The system validates the bid amount. The entered bid must be greater than the current (i.e., greatest) bid by...										
UC1.2.3: BUYER CONFIRMS The system provides information that tells the legal obligations of placing a bid. The Buyer confirms that the bid should be placed.						↗	↗			
UC1.2.4: POST BID The system posts the bid for the auction. The entered bid becomes the current (i.e., greatest) bid.		↗								
UC1.2.5: SEND EMAIL The system sends an e-mail confirmation to the Buyer, including the bid amount for the auction item, as well as when the auction will...									↗	

IBM Rational Requisite Pro



Traceability policies

- ◆ Define what and how traceability information should be maintained
- ◆ Traceability policies may include
 - ◆ **Kind** of information which should be maintained
 - ◆ **Techniques**, such as traceability matrices, which should be used for maintaining traceability
 - ◆ **When** the traceability information should be collected during the requirements engineering and system development processes
 - ◆ **Roles** of the people, such as the traceability manager, who are responsible for maintaining the traceability information

Factors influencing traceability policies

- ◆ Number of requirements
- ◆ Estimated system lifetime
- ◆ Project team size and composition
- ◆ Type of system
 - ◆ Real-time or safety-critical systems need more comprehensive traceability policies than non-critical systems
- ◆ Specific customer requirements
 - ◆ Customers may demand that specific traceability information is delivered as part of the system documentation

Overview

- ◆ What is Requirements Management?
- ◆ Traceability of Requirements
- ◆ **Storing Requirements, Requirements Tools**
- ◆ **Requirement Change Management**

Storing Requirements

Requirements Documents

Tool Support

Storing requirements

- ◆ Requirements have to be stored in such a way that they can be easily
 - ◆ accessed
 - ◆ changed
 - ◆ linked (with other requirements)
 - ◆ described (in text as well as in graphics...)
 - ◆ enhanced (by adding external information)
- ◆ Possible storage techniques are
 - ◆ One or more word/spreadsheet files
 - ◆ requirements are stored in the requirements document
 - ◆ Requirements database / Information Systems

Storing requirements: Word/Spreadsheet processor documents

- ◆ Advantages
 - ◆ Easy to construct, maintain, cheap
 - ◆ Requirements may be accessed by anyone with the right word processor
 - ◆ Requirements can be described informal, unstructured...
 - ◆ It is easy to produce the final requirements document
 - ◆ Link requirements via hyperlinks
 - ◆ Version Control via dedicated VCS, e.g. SVN
- ◆ Disadvantages
 - ◆ Navigation between documents fragile (changing directories / file names)
 - ◆ Search facilities are limited
 - ◆ Parallel editing commonly not available
 - ◆ Distributed editing needs further support (e.g. VPN)
 - ◆ Not possible to link requirements with requirements changes
 - ◆ Not possible to have version control on individual requirements (only whole document)
 - ◆ No built-in dependency analysis

Storing requirements: Requirements database

- ◆ Each requirement is represented as one or more database entities
- ◆ Database query language is used to access requirements
- ◆ Advantages
 - ◆ Good query and navigation facilities
 - ◆ Support for change and version management
 - ◆ Versioning of single requirements
- ◆ Disadvantages
 - ◆ Readers may not have the software/skills to access the requirements database
 - ◆ Flexibility can be reduced by limited input forms
 - ◆ Higher costs

Combination: Word Processor + Database

- ◆ Often used combination, e.g. Rational Requisite Pro
- ◆ Requirements are written in a word processor file
- ◆ Database gets updates via add-ins / plug-ins
- ◆ Links to other requirements are transparently resolved by the central RM server

Requirements DB: Choice factors

- ◆ The statement of requirements
 - ◆ text, graphics, photos
 - ◆ → external linked storage or multimedia database
- ◆ The number of requirements
- ◆ Teamwork, team distribution and computer support
 - ◆ Distributed team of requirements engineers
 - ◆ → Remote, multi-site access
 - ◆ → Web-Browser interface
- ◆ Change management
- ◆ Traceability support: uni- or bi-directional
- ◆ Prioritize requirements by sorting and filtering

CASE tools for requirements management

- ◆ Large number of requirements management tools
 - ◆ Telelogic AB (IBM) – DOORS
 - ◆ IBM Rational – Requisite Pro
 - ◆ OSRMT (Open Source Requirements Management Tool)
 - ◆ RM Tool Surveys
 - ◆ <http://www.volere.co.uk/tools.htm>
 - ◆ <http://www.paper-review.com/tools/rms/>
- ◆ Other CASE tools may be adopted for requirements engineering
 - ◆ Groupware (e.g. Lotus Notes)
 - ◆ Issue Tracking System (JIRA/Confluence, Trac)

Examples

IBM Rational Requisite Pro

- ◆ Customizable, filterable requirement attributes
- ◆ Integrates with Microsoft Word
- ◆ creation and comparison of requirement baselines
- ◆ Cross referencing the requirements to each other – traceability
- ◆ Change impact analysis with audit trails
- ◆ Formatting and presenting the requirements to others
- ◆ Web-access for distributed teams
- ◆ Predefined templates for common processes (e.g. RUP)
- ◆ Integrated with other Rational Tools
 - ◆ ClearCase: Version Control
 - ◆ ClearQuest: Change Management
 - ◆ Rational Architect: Model-driven Development

Examples

OSRMT

- ◆ *Open Source Requirements Management Tool (OSRMT)*
- ◆ free, full-features RMT
- ◆ Database-centric
- ◆ Java Rich Client + Web-access
- ◆ Configurable requirement description
- ◆ Bidirectional tracing
- ◆ Audit support
- ◆ Change management module with custom views
- ◆ Arbitrary documents attachable to an artifact
- ◆ Issues
 - ◆ Misses elaborated templates
 - ◆ Stability problems
 - ◆ Not maintained anymore

Requirement Change Management

Change Factors

Change Management

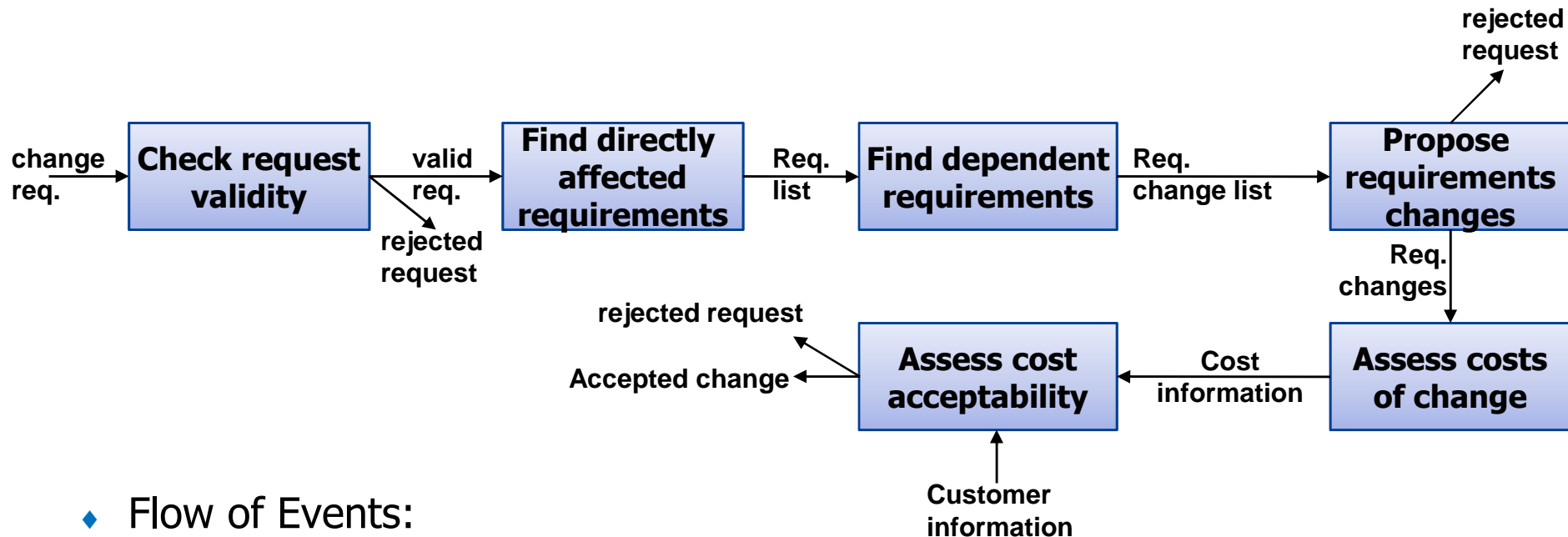
Requirements Change Factors

- ◆ Requirements errors, conflicts and inconsistencies
- ◆ Evolving customer/end-user knowledge
- ◆ Technical, schedule or cost problems
- ◆ Environmental changes
 - ◆ System requirements change to maintain compatibility with new environment
- ◆ Changing customer priorities
 - ◆ The emergence of new competitors, staff changes, etc.
 - ◆ Changes in structure and processes in the target organization
 - ◆ New laws, regulations
- ◆ New technology (technology push)

Change management

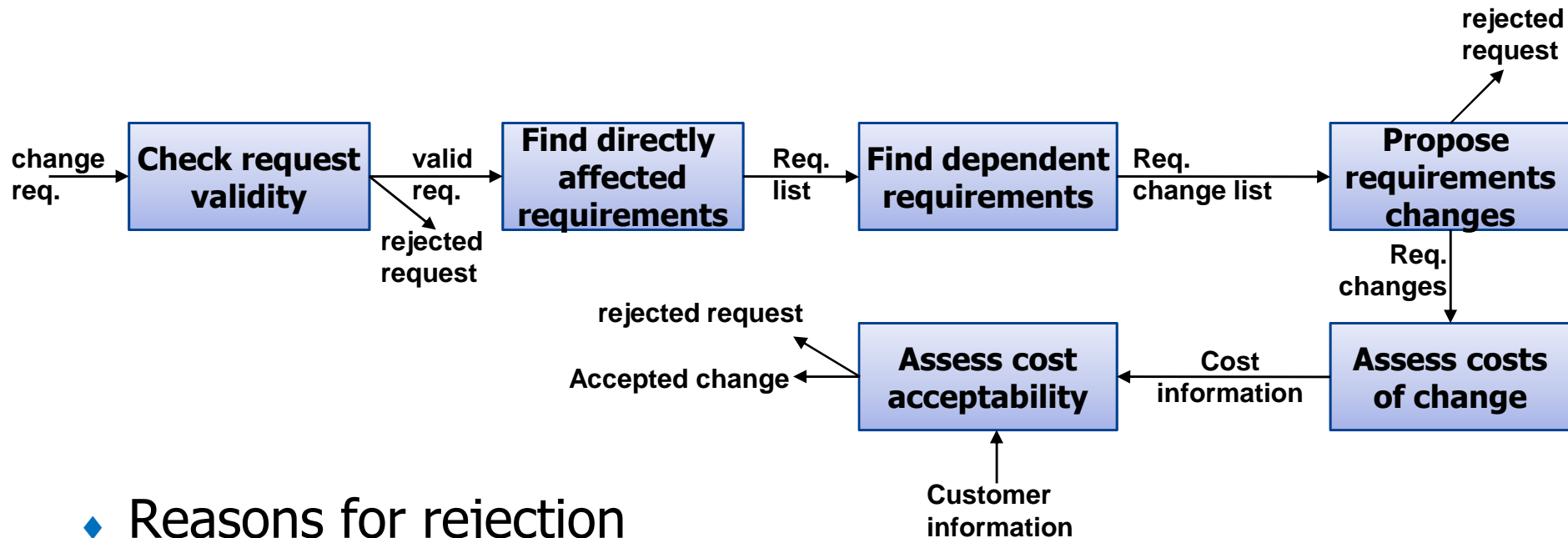
- ◆ Concerned with procedures, processes and standards which are used to manage changes to system requirements
- ◆ Change management policies may cover:
 - ◆ Change request **process**
 - ◆ Analyze the impact and
 - ◆ Costs of change
 - ◆ Traceability information
 - ◆ Change Request Board
 - ◆ Software support (if any) for the change control process

Change management: Change analysis and costing 1/2



- ◆ Flow of Events:
 - ◆ Check for validity
 - ◆ Customers can misunderstand requirements and suggest unnecessary changes
 - ◆ Requirements directly affected by the change are discovered
 - ◆ Dependent requirements are discovered (using Traceability Information)
 - ◆ Propose actual changes (consultation with customer)
 - ◆ Estimating costs of making the changes
 - ◆ Negotiations with customers
 - ◆ Are the costs acceptable?

Change management: Change analysis and costing 2/2



◆ Reasons for rejection

- ◆ Change request is invalid: customer has misunderstood some requirements, proposed change isn't necessary
- ◆ Too many dependent requirements: consequential changes are unacceptable to the user
- ◆ Costs are too high or take too much time

Tool support for change management

- ◆ Provided through requirements management tools, or
- ◆ by integration of specialized components (e.g. Rational ClearQuest)
- ◆ Tool facilities may include
 - ◆ Electronic change request forms
 - ◆ A database to store and manage these forms
 - ◆ Group support:
 - ◆ Electronic transfer of forms between people with different responsibilities
 - ◆ Electronic mail notification
 - ◆ Document Management Support:
 - ◆ Direct links to a requirements database
 - ◆ Affected requirements
 - ◆ Automatic Updates

Summary

- ◆ Requirements change is inevitable
- ◆ Unique identification of requirements is needed
- ◆ Traceability
- ◆ Storage of requirements: Text document, database
- ◆ Change management