

Object-Oriented Software Engineering

Requirements Management

Radovan Cervenka

Requirement

- What is it?
 - a condition or capability to which the system must conform
- Examples
 - The QBS system shall, upon user request, display detailed customer information
 - Each user shall have a unique login and associated password
 - The QBS Banking system shall be user friendly
 - The QBS Banking system requires 32 MB of RAM and a Pentium Processor for maximum performance
 - Maintain Accounts
 - Generate Reports
 - After printing is complete, the system returns to ready
 - ...

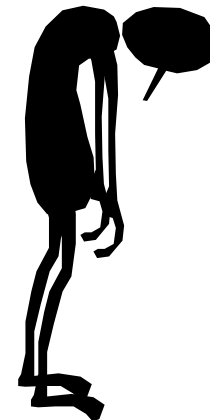
Requirements Management

- a systematic approach to eliciting, organizing, documenting and distribution of requirements, and
- a process that establishes and maintains agreement between the customer and the project team on the changing requirements of the system

- Requirements Workflow in RUP:
 - To come to an agreement with the customer and the users on what the system should do
 - To give system developers a better understanding of the requirements on the system
 - To delimit the system
 - To provide a basis for planning the technical contents of iterations
 - To define a user-interface for the system

Problems of RM

- Many sources and interested parties
- Not obvious and difficult to express in words
- Different levels of abstraction
- Unmanageable huge amount, if not controlled
- Needed to be managed by cross-functional groups of people
- Change
- Time-sensitivity



Results of RM Errors

- Poor understanding of the tasks
- An undisciplined approach to prioritizing, managing, and tracking requirements
- Lack of communication
- Schedule delays
- Cost overruns
- A product that only bears a slight resemblance to what users wanted
- A loss of competitiveness in the industry
- ...



Definition of RM

- Requirements model
- Requirements artifact types
- Stakeholder types and their roles
- Workflows
- Tools and their customization
- (Process and model quality criteria)



Requirement Types

→ a class of requirements

Examples

- stakeholder needs
- system features
- functionality (Use Cases)
- non-functional requirements (Supplementary Specifications)
 - usability, reliability, performance, supportability, design constraints, etc.
- implementation requirements
- interface requirements
- data requirements
- physical requirements
- Test Cases
- ...

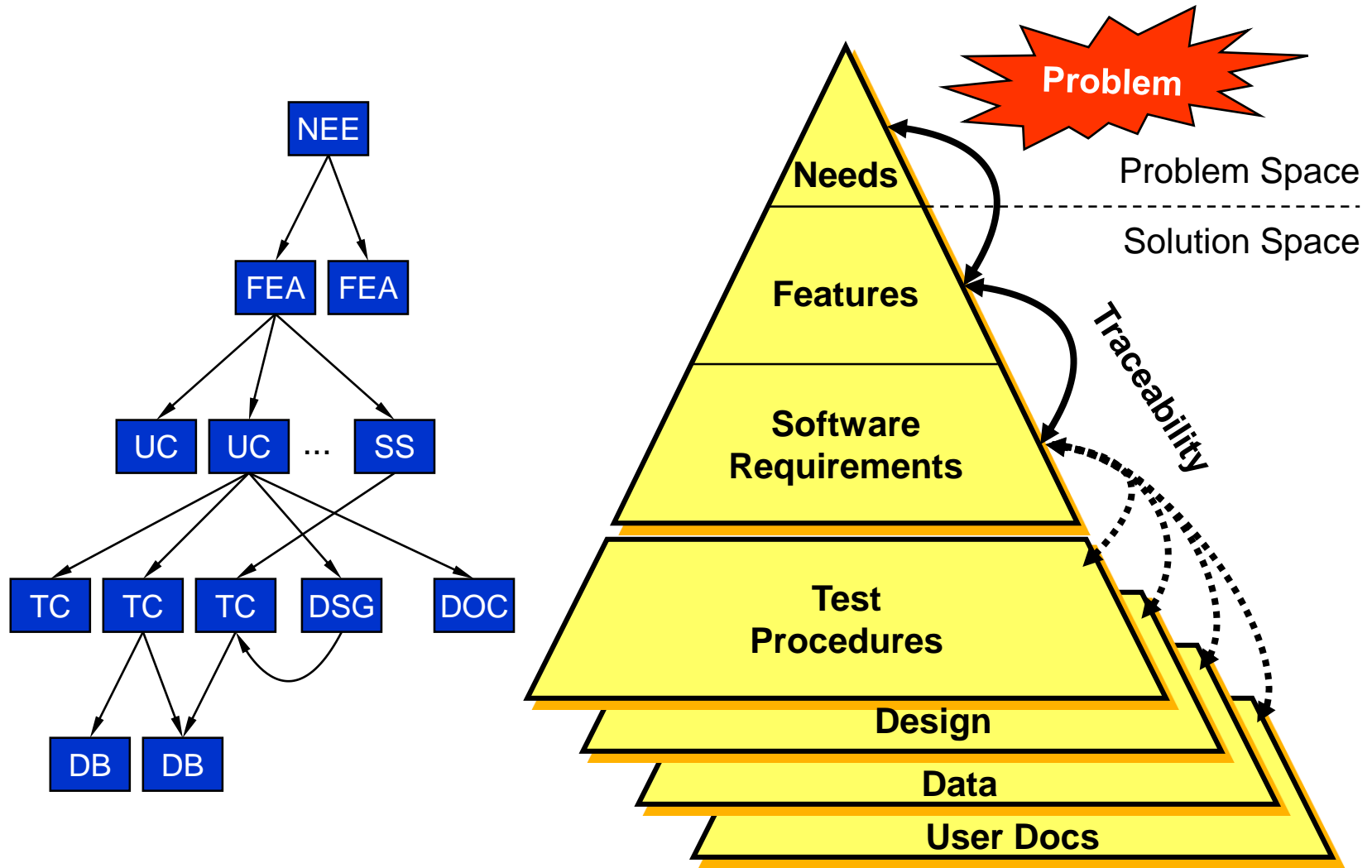
Requirement Attributes

- an information associated with a particular requirement providing a link between the requirement and other project elements, e.g. priorities, schedules, status, design elements, resources, costs, hazards, etc.
- each requirement type may have its specific attributes

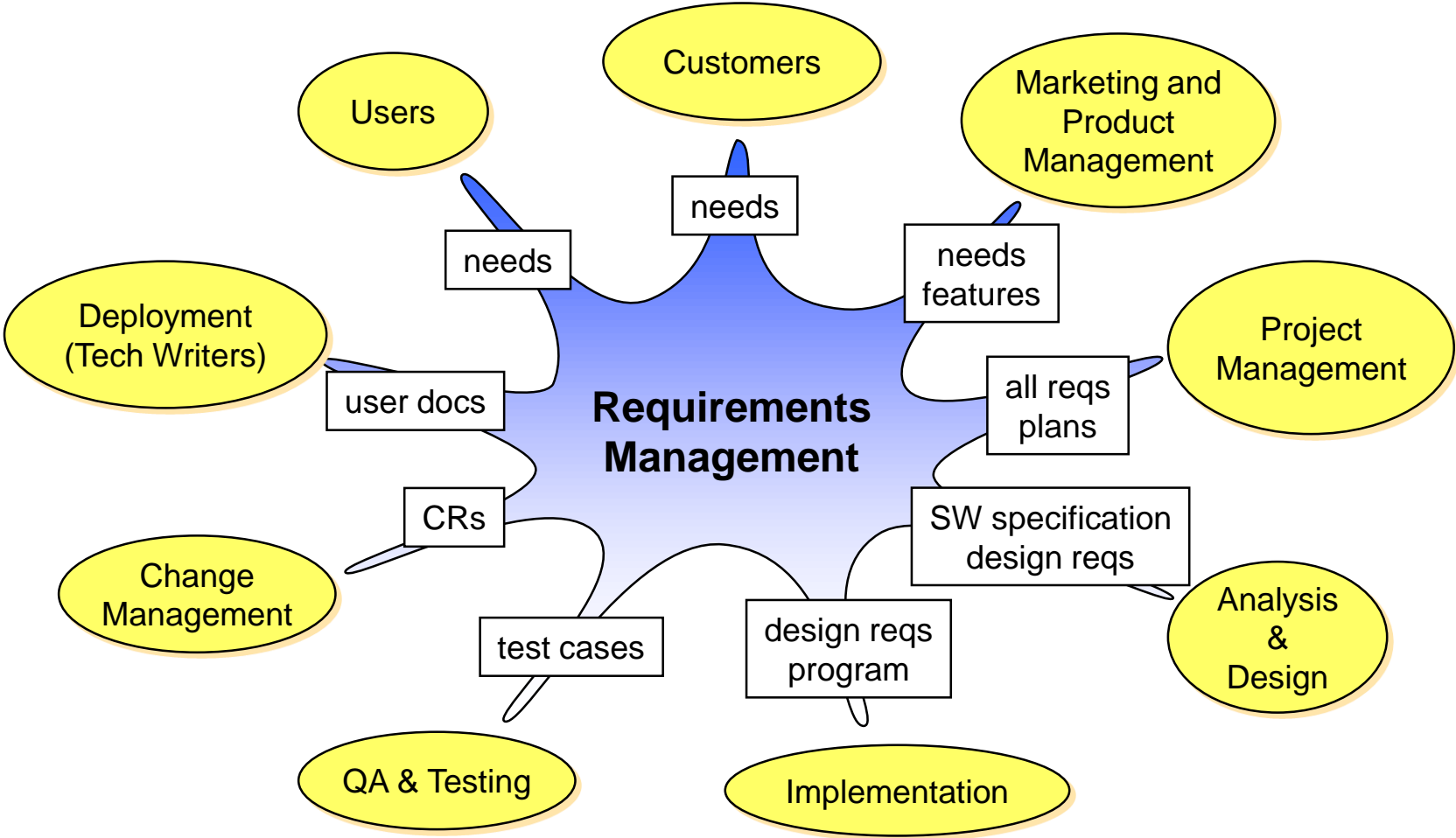
Features					
Id	Name	Priority	Difficulty	Risk	Stability
FEA 1	Save and restore user's configurations	Mid High	Mid Low	Low	High
FEA 2	Ability to cooperate with Xetra	Very High	Very High	High	Middle
FEA 2.1	Ability to send all kinds of orders to Xetra	High	Very High	High	Low

Use Cases						
Id	Name	Goal	Scenario	Trigger	Precond.	...
UC 1	Save Trading Overview Configurations	User action	...	
UC 2	Submit Container Window Status	User action	...	
UC 50	Submit Unrestricted Order to Xetra	...	1 Open Trading Window 2 Drag TI into FOE line 3 Press Submit button	User action	...	
UC 51	Submit FOK Order to Xetra	High	...	
UC 84	Get Xetra Trade	Xetra GW	...	

Requirements Traceability



RM Connectivity



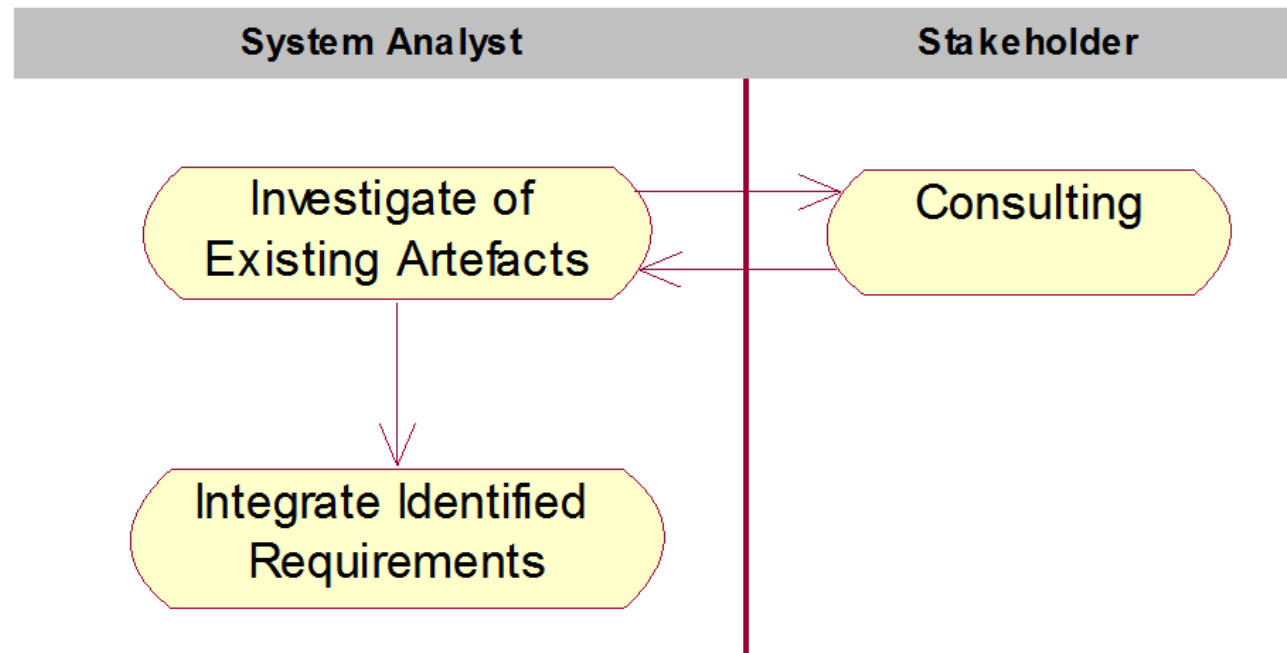
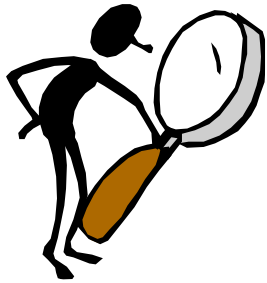
RM Skills

- Problem Analysis
 - understand business problems, target initial stakeholder needs, and propose high-level solutions
- Understanding Stakeholder Needs
 - determine project stakeholders and elicit their needs resulting in a “wish list”
- Defining the System
 - translate needs into meaningful description of the system to be built, early prototyping and high-level analysis
- Managing the Scope of the Project
 - managing requirement attributes, such as priority, effort, risk, ...
- Refining the System Definition
 - detail UCs, supplementary specifications, and GUI
- Managing Changing Requirements
 - determining traceabilities, establishing a baseline, maintain history and version control; decision process

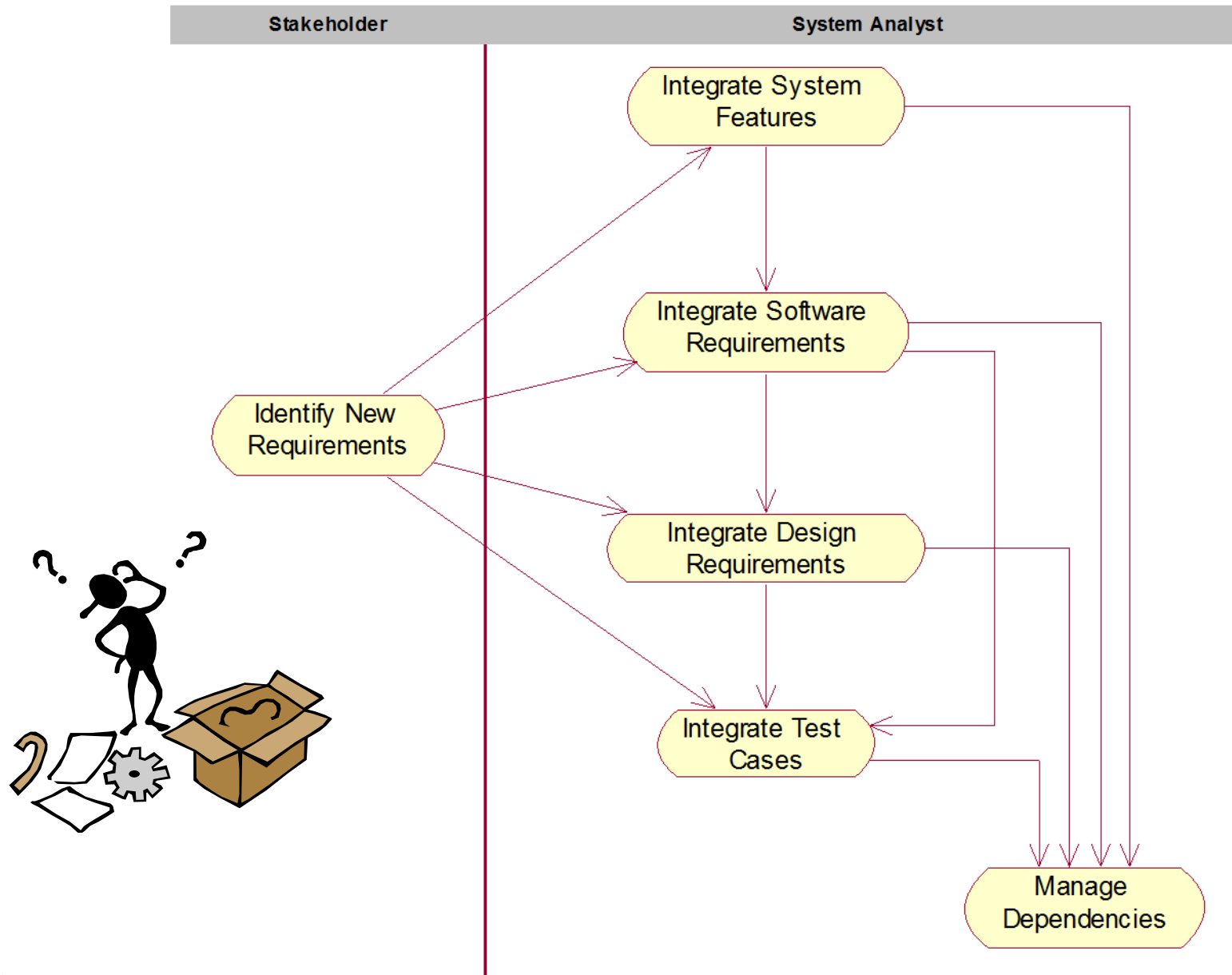
RM Workflows (Reqs. Lifecycle)

- Initial Requirements Identification
 - building of initial requirements model from existing artifacts and know-how
- New Requirements
 - identifying of new requirements and their integration into the requirements model
- Requirements Refining
 - detailing of use case model and test cases
- Requirements Change
 - changing of requirements, assessment of their impacts and consequent processing
- Requirements Review
 - official requirements reviewing
- Control of Requirements Adherence
 - checking the consistency of requirements, artifacts (A/D documents and models, test cases, user's documentation, etc.) and implementation

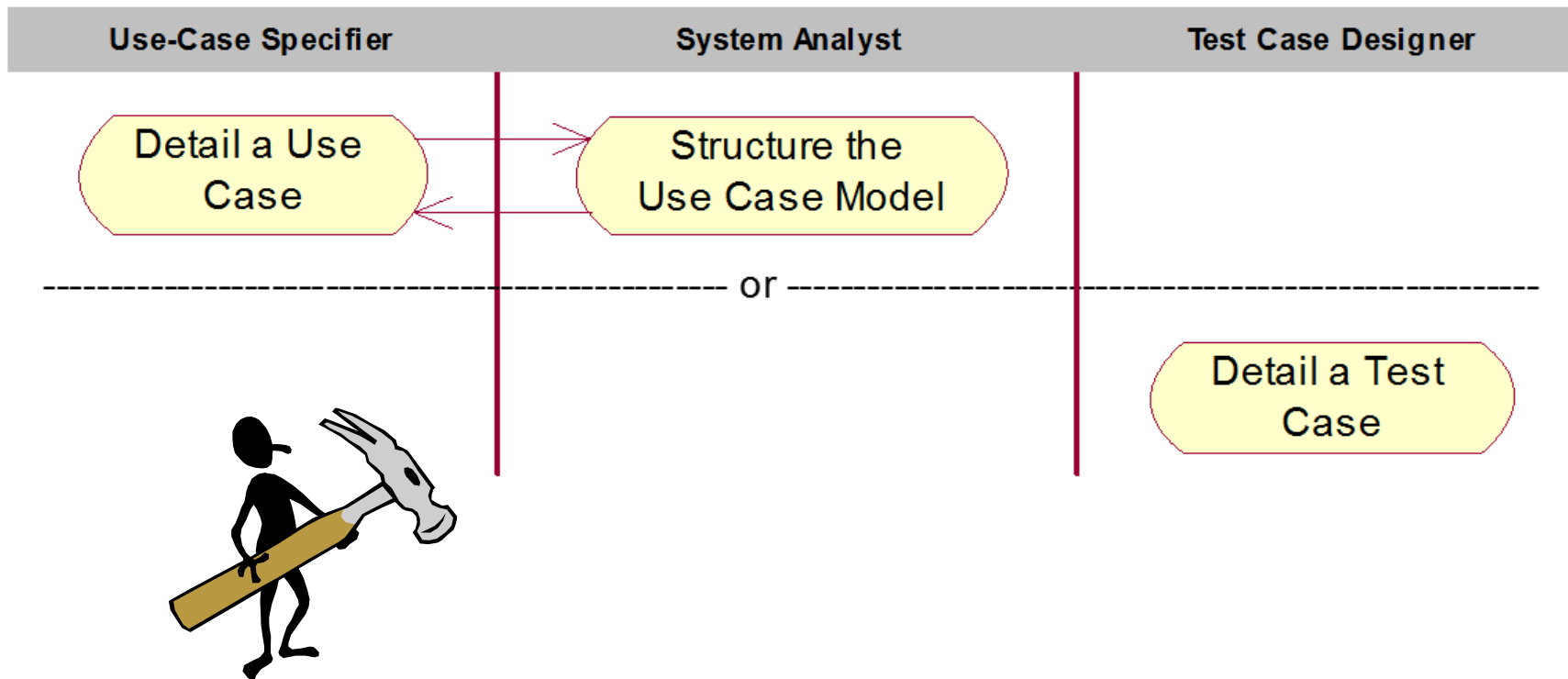
Initial Requirements Identification



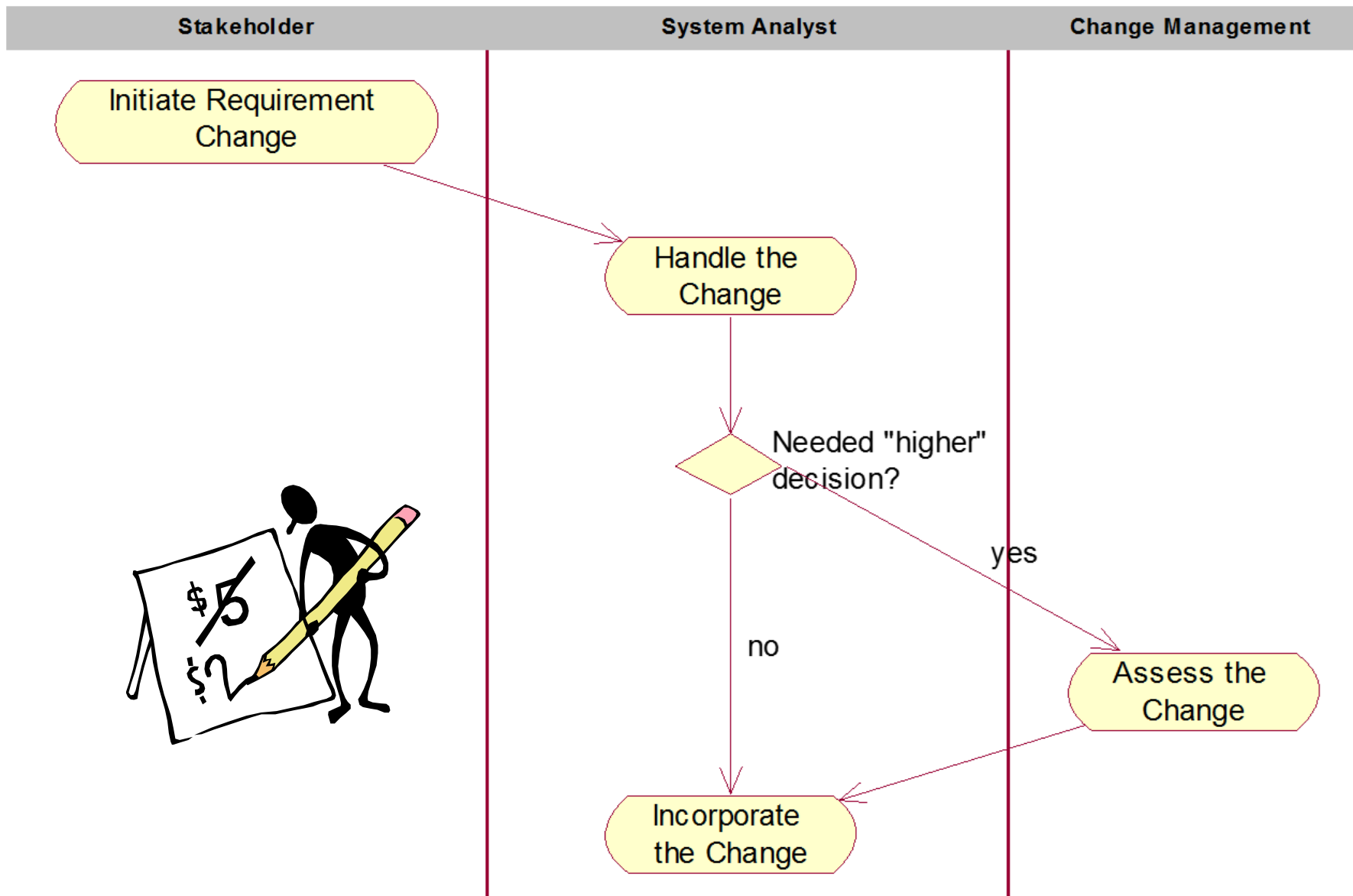
New Requirements



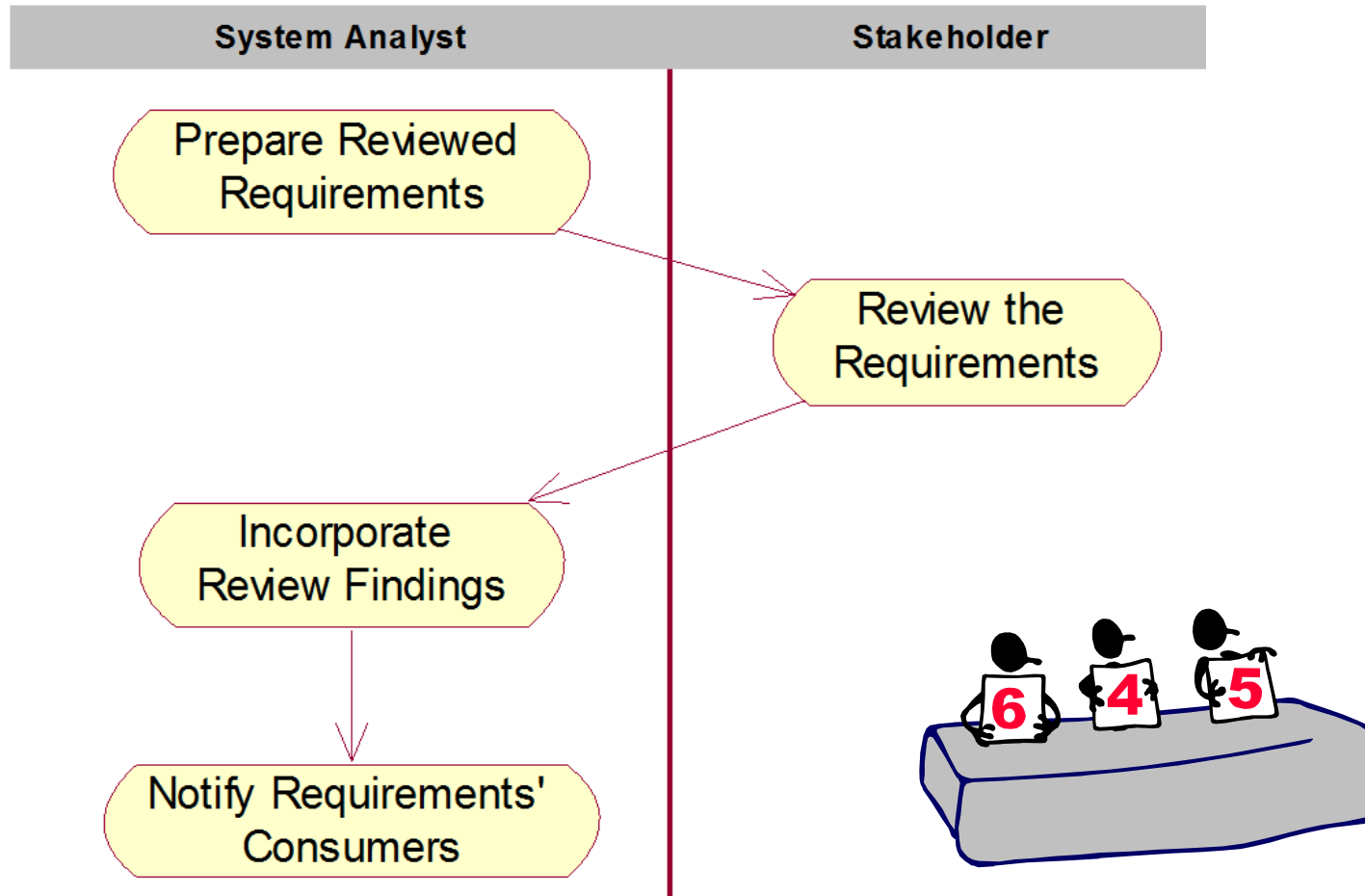
Requirements Refining



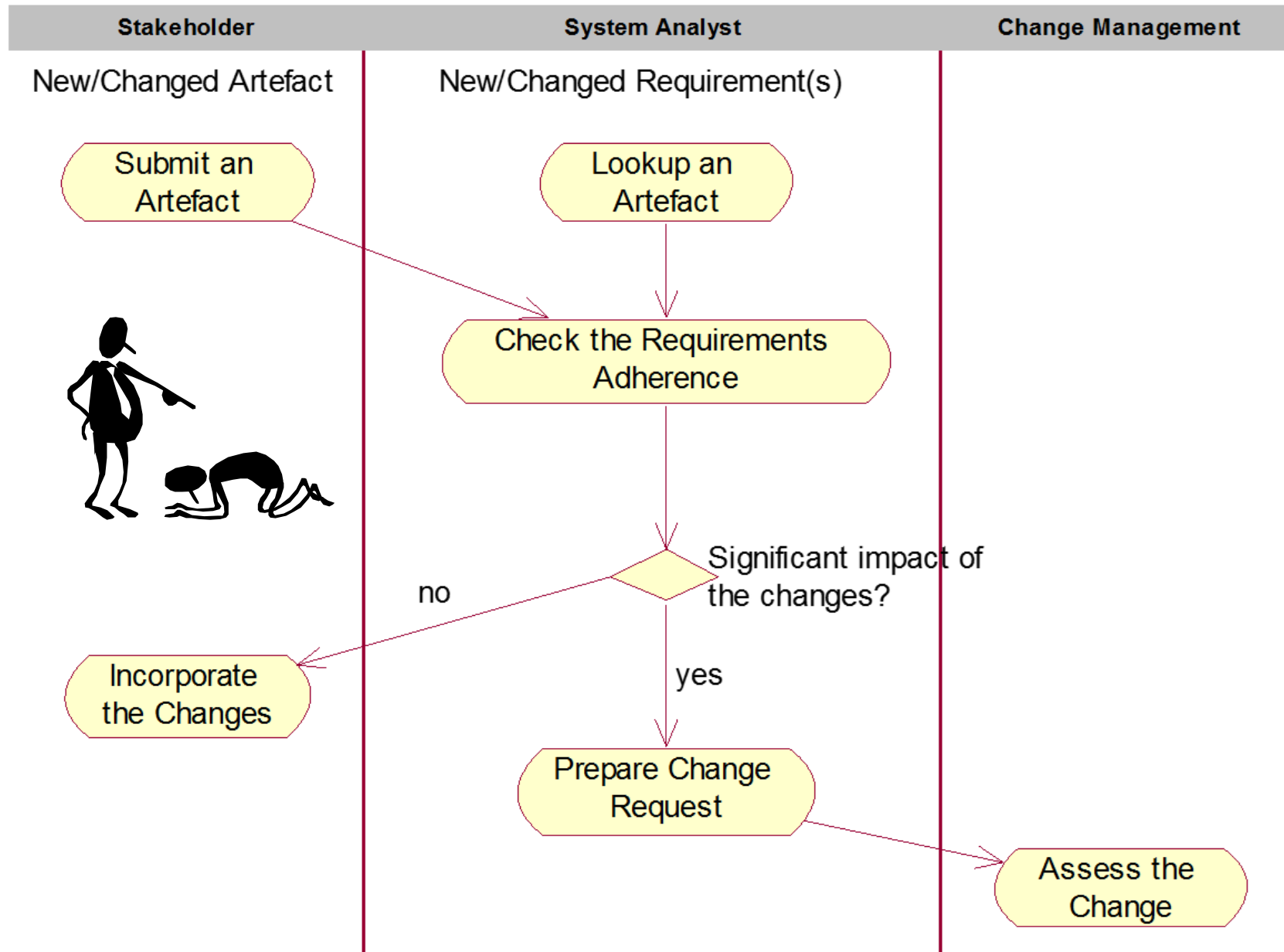
Requirements Change



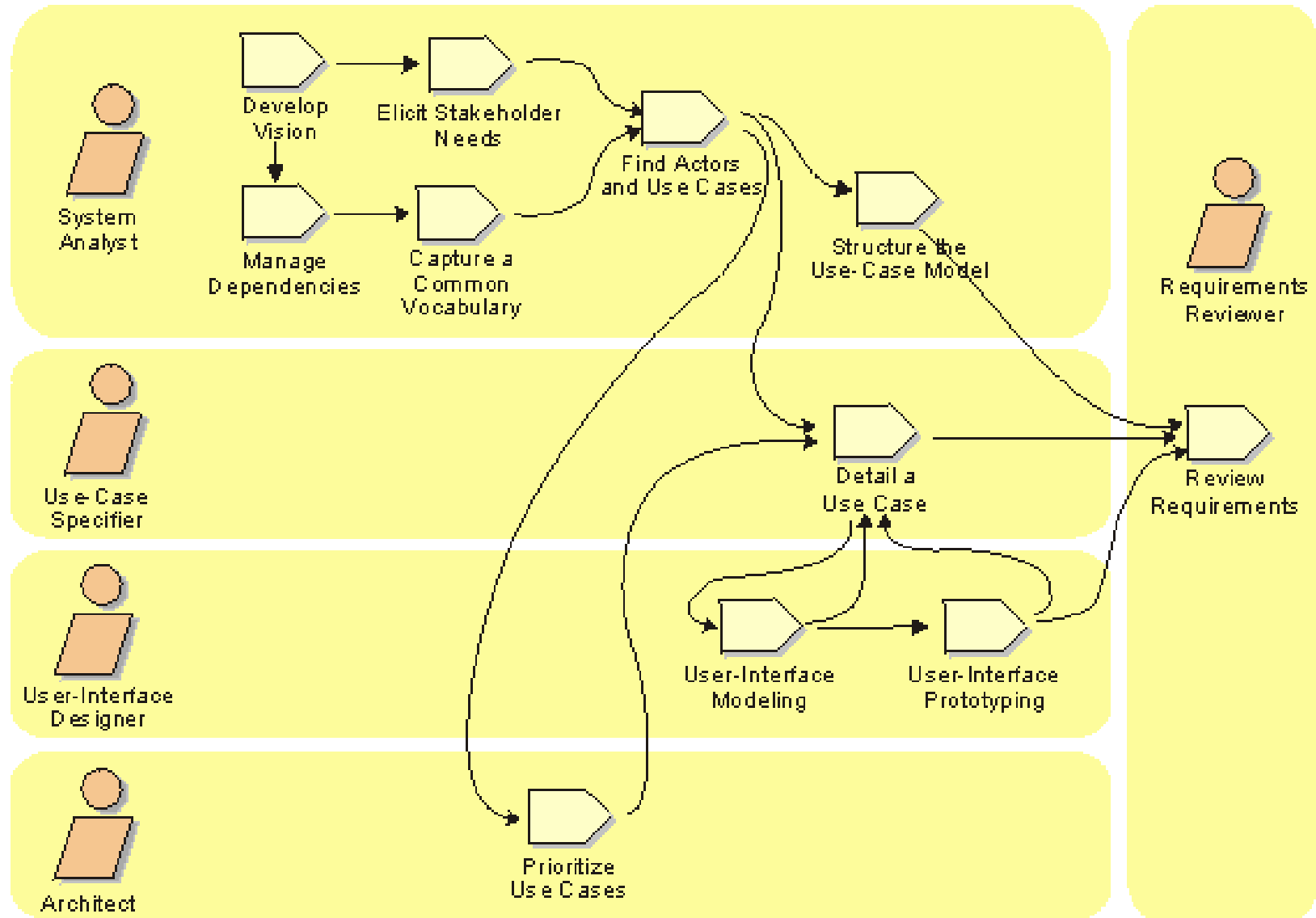
Requirements Review



Control of Requirements Adherence



RUP: Requirements Workflow



RUP: Requirements Artifacts

