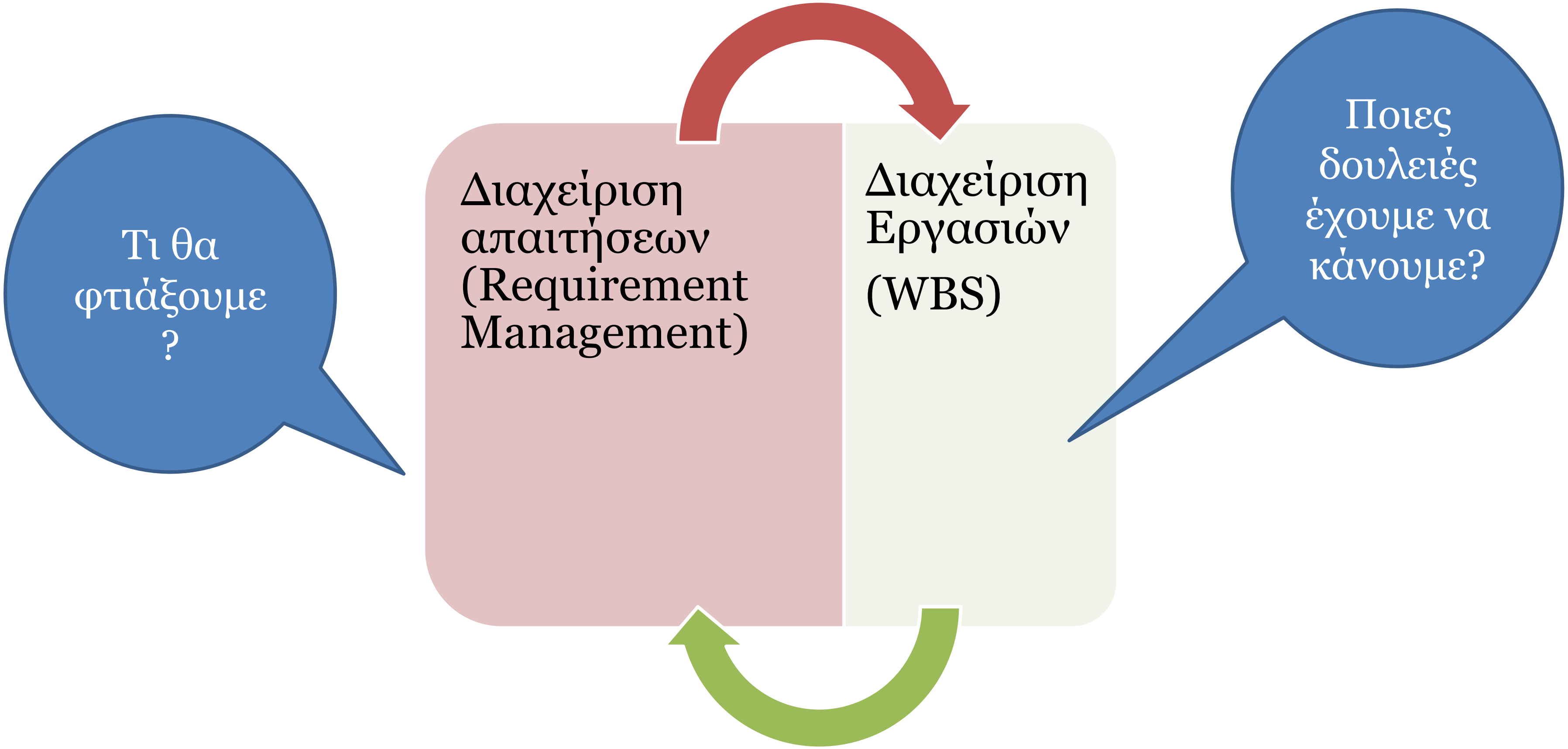
ΔΙΑΧΕΙΡΙΣΗ ΕΡΓΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

Πάνος Φιτσιλής
pfitsilis@gmail.com

Διαχείριση αντικείμενου εργασιών

Διαχείριση Αντικειμένου Εργασιών (Scope Management)



Τι θα φτιάξουμε ?

Διαχείριση απαιτήσεων (Requirement Management)

Διαχείριση Εργασιών

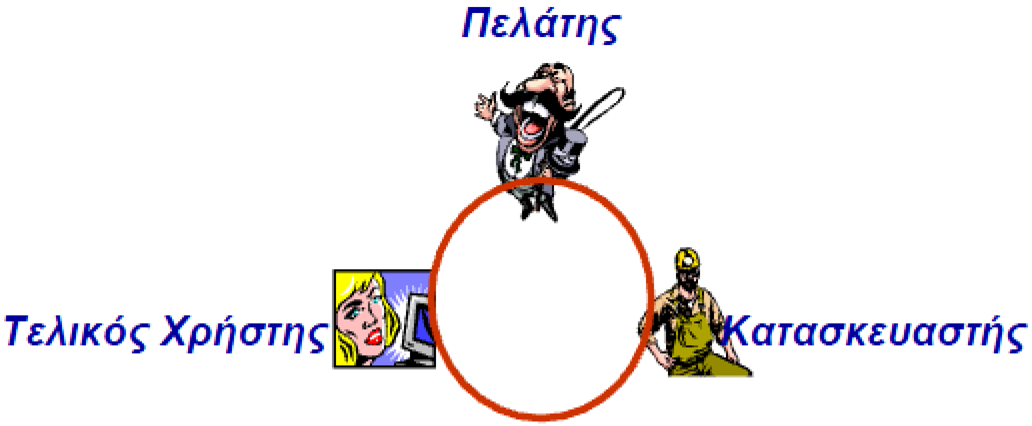
(WBS)

Ποιες δουλειές έχουμε να κάνουμε?

THE PROJECT REQUIREMENTS ARE FORMING IN MY MIND
NOW THEY'RE CHANGING... CHANGING... CHANGING... CHANGING... OKAY. NO, WAIT... CHANGING... CHANGING... DONE.
NATURALLY, I WON'T BE SHARING ANY OF THESE THOUGHTS WITH ENGINEERING.
I BUDGETED FOR SOME GOONS TO BEAT IT OUT OF YOU.
S. Adams
E-Mail: SCOTTADAMS@AOL.COM
©1994 United Feature Syndicate

The requirements problem

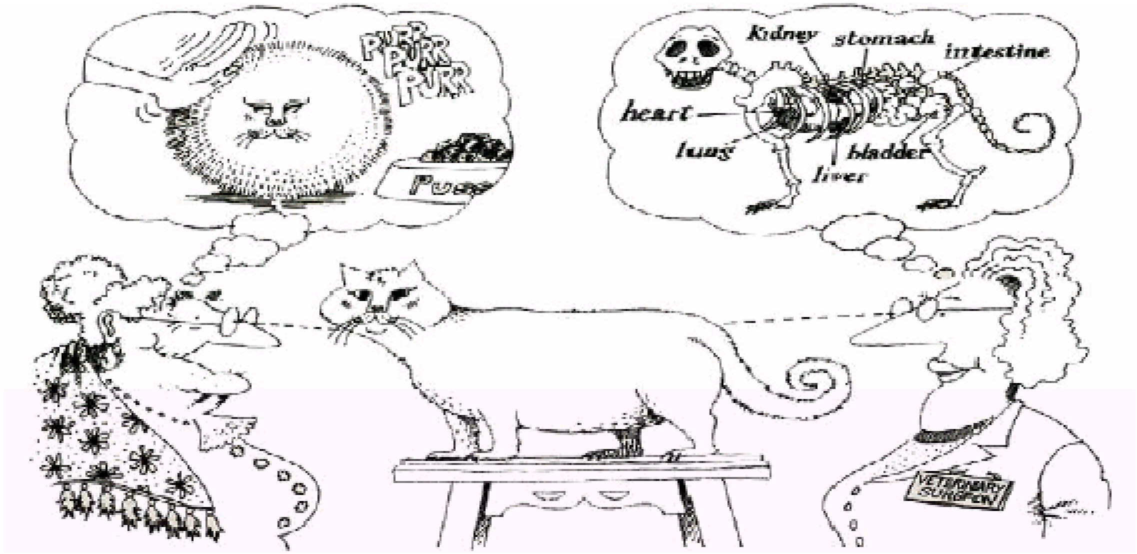
- The goal of software development is to develop quality software—on time and on budget—that meets customers real needs.
- Project success depends **on good requirements** management.
- Requirements errors are the most common type of systems development error and the most costly to fix.
- A few key skills can significantly reduce requirements errors and thus improve software quality.



Πελάτης

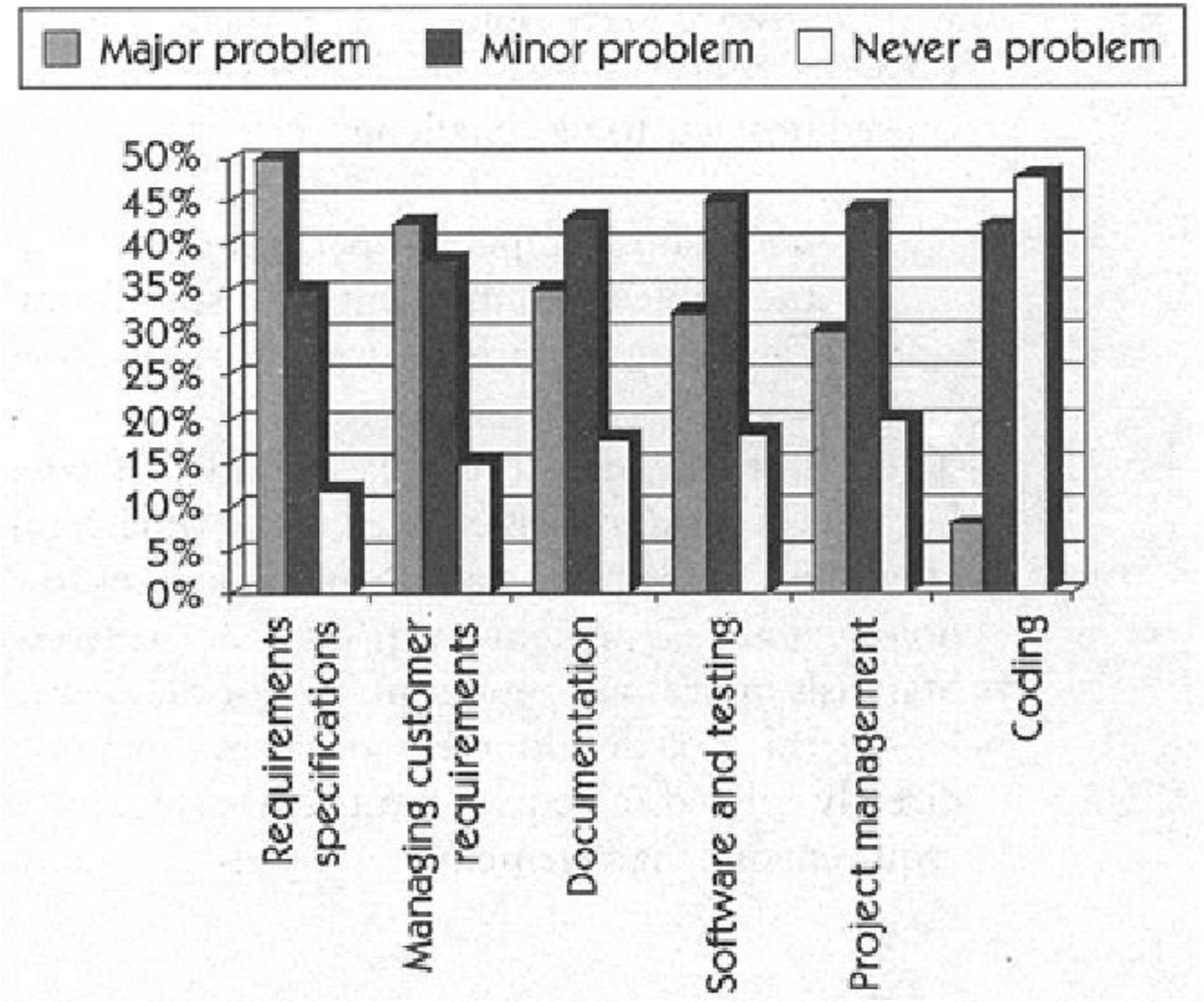
Τελικός Χρήστης

Κατασκευαστής

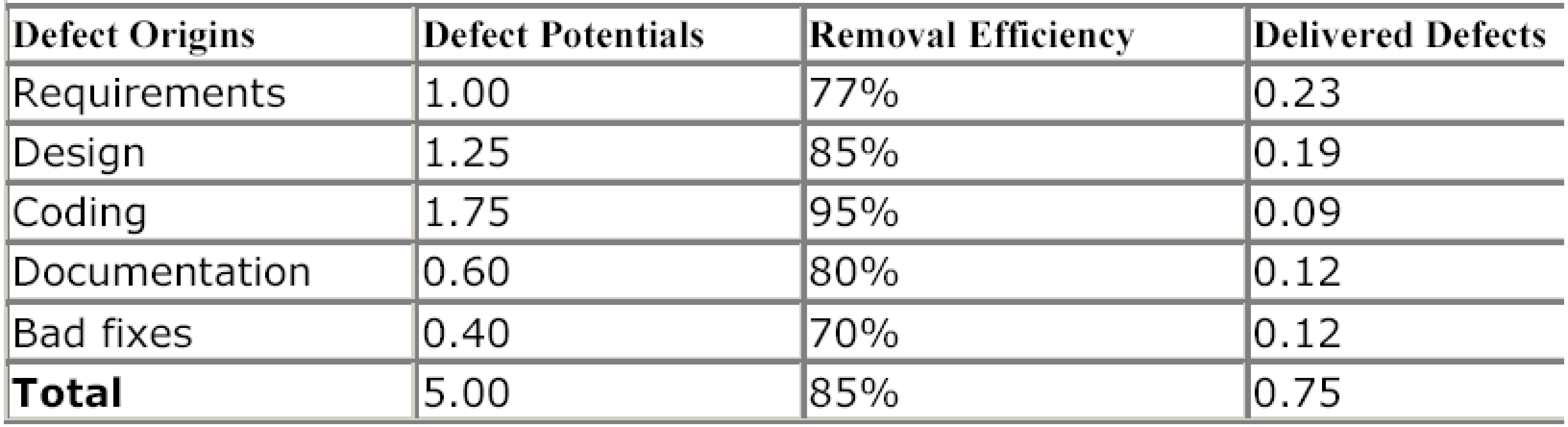


Major software development problems

European Software Process Improvement Training Initiative (ESPITI)

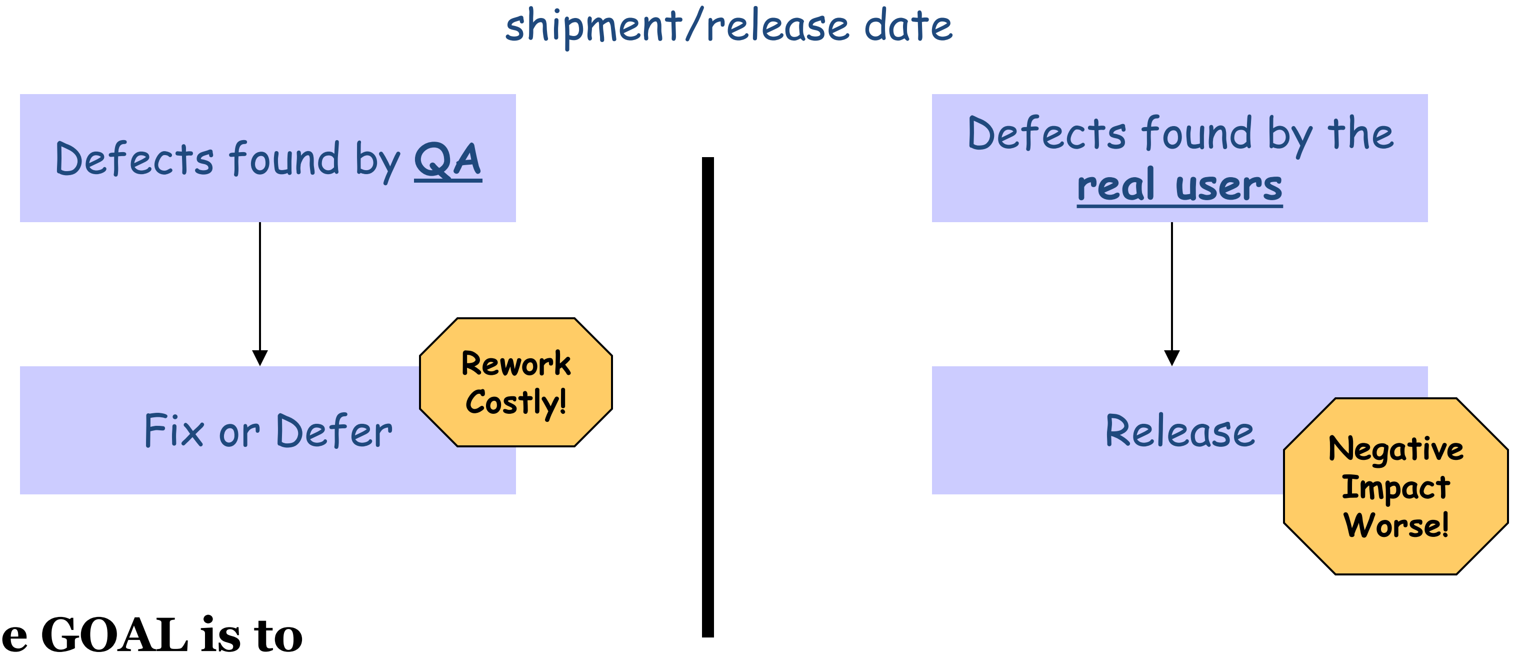


Defect summary(Capers Jones)



Defect Origins	Defect Potentials	Removal Efficiency	Delivered Defects
Requirements	1.00	77%	0.23
Design	1.25	85%	0.19
Coding	1.75	95%	0.09
Documentation	0.60	80%	0.12
Bad fixes	0.40	70%	0.12
Total	5.00	85%	0.75

- **Defects**



- **The GOAL is to**
  - Decrease the # of defects
  - Find as many defects as early as possible in work products while running scenarios before the product is shipped out of the door

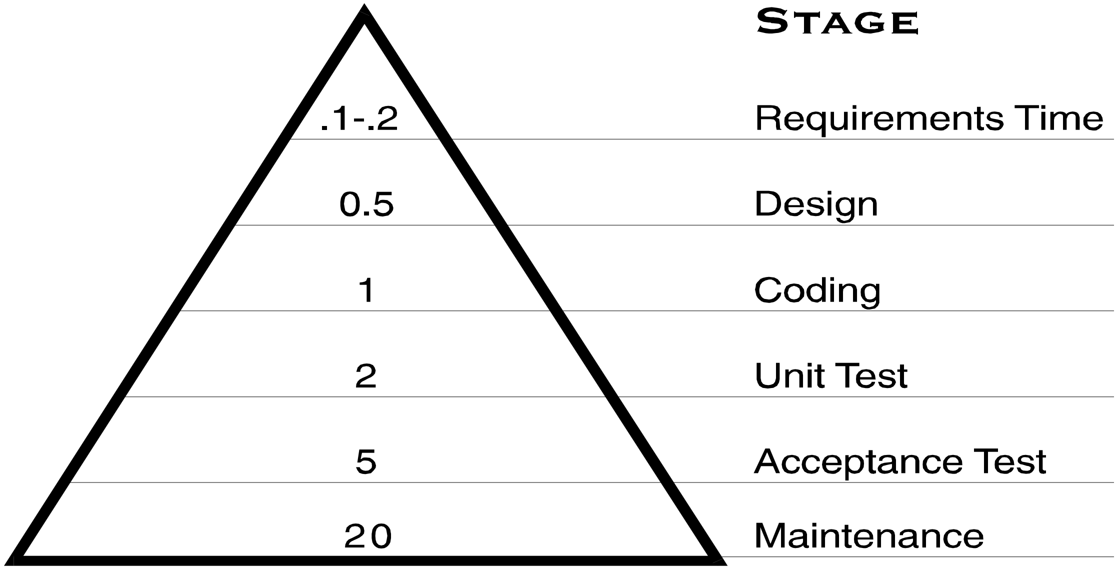
DEFECT ORIGINS

- **Defects breakdown:**
  - **Requirements management 62.5%**
  - Design 11.3%
  - Coding 10%
  - Environment 4.9%
  - Test data or test scripts or execution errors 11.3%
- **Requirement management defects breakdown:**
  - **Requirement Completeness 37.5%**
  - Requirement Presentation 34.7%
  - Requirement Changes 11.2%
  - Requirement Incorrect 8.7%
- **Requirement Completeness defects breakdown:**
  - Incomplete Requirements 73.4%
  - Missing requirements 11.2%
  - Overly generalized requirements 4.6%

-- “Defect Prevention Techniques for High Quality and Reduced Cycle Time”
http://www.iscn.ie/select_newspaper/measurement/motorola2.html

Impact of Requirements Problems?

As much as a 200:1 cost savings results from finding errors in the requirements stage versus finding errors in the maintenance stage of the software life-cycle.



Relative Cost to Repair	Stage
.1-.2	Requirements Time
0.5	Design
1	Coding
2	Unit Test
5	Acceptance Test
20	Maintenance

Barry Boehm-
‘76, 88

56% of all bugs can be traced to errors made during the requirements stage

Requirements management

- A requirement is a capability that the system must deliver.
  - A software capability needed by the user to solve a problem to achieve an objective
  - A software capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documentation
- Requirements management is a process of systematically eliciting, organizing, and documenting requirements for a complex system.
- Our problem is to understand users‘ problems in their culture and their language and to build systems that meet their needs.
- A feature is a service that the system provides to fulfill one or more stakeholder needs.
- A use case describes a sequence of actions, performed by a system, that yields a result of value to a user.

Why we need to manage requirements?

- Small software product has 1000 requirements
- Boeing 777 has to satisfy 300,000 requirements
- The questions are:
  - Which project team members are responsible for the wind speed requirement (#278), and which ones are allowed to modify it or delete it?
  - If requirement #278 is modified, what other requirements will be affected?
  - How can we be sure that someone has written the code in a software system to fulfill requirement #278, and which test cases in the overall test suite are intended to verify that the requirements have indeed been fulfilled?

Λειτουργικές και Μη-Λειτουργικές Απαιτησεις (ΛΑ & ΜΛΑ) (Functional and Non-Functional Requirements)

- Λειτουργικές Απαιτήσεις (ΛΑ) Functional Requirements (FR)
  - Περιγράφουν τι πρέπει να κάνει το σύστημα (π.χ. ως συναρτήσεις που λαμβάνουν είσοδο και δίδουν έξοδο)
- Μη-Λειτουργικές Απαιτήσεις (ΜΛΑ)
  - Περιγράφουν ιδιότητες του συστήματος που συνήθως εκφράζονται βάσει χαρακτηριστικών της μορφής:
    - Απόδοση (performance)
    - Χρηστικότητα (usability)
    - Ασφάλεια (security)
    - Νομιμότητα (legislative)
    - Ιδιωτικότητα (privacy)
  - Με άλλα λόγια: περιγράφουν το πώς (ή το πόσο καλά) το σύστημα θα υποστηρίξει τις λειτουργικές απαιτήσεις

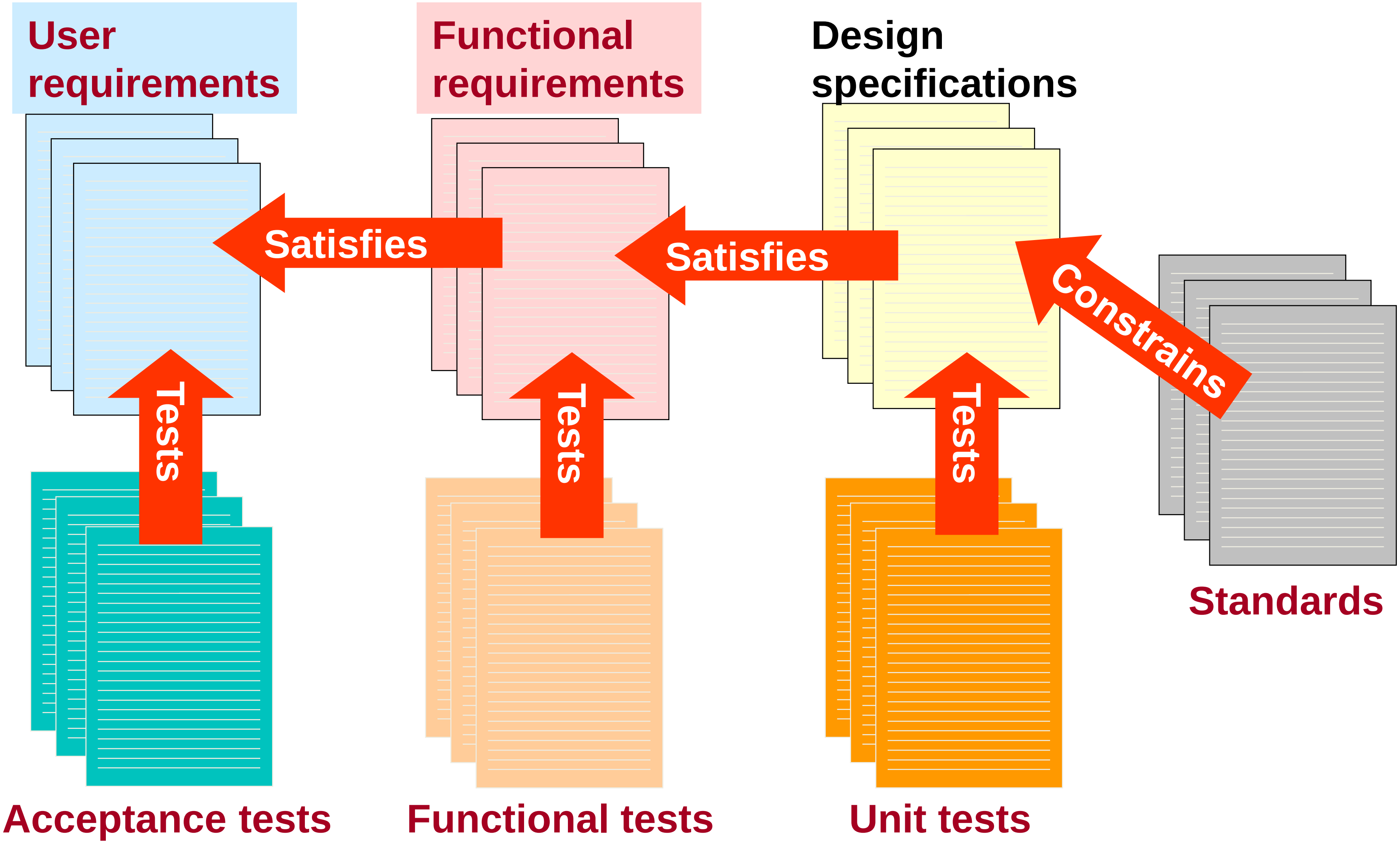
Ειναι ΛΑ η ΜΛΑ

- *Ο χρόνος απόκρισης του συστήματος δεν πρέπει να υπερβαίνει τα 3 δευτερόλεπτα*
- *Το σύστημα πρέπει να μπορεί να ενοποιηθεί με το υπάρχον*
- *Τα προσωπικά στοιχεία των πελατών πρέπει να προστατεύονται.*
- *Να αποθηκεύει τα στοιχεία των πελατών*
- *Να τυπώνει συγκεντρωτικές αναφορές*
- *Το σύστημα πρέπει να λειτουργεί αδιάλειπτα (όλο το χρόνο)*
- *Να τυπώνει αποδείξεις*
- *Μόνο οι διευθυντές πρέπει να έχουν πρόσβαση τους μισθούς*
- *Το σύστημα πρέπει συμμορφώνεται με τα πρότυπα της βιομηχανίας*
- *Το σύστημα πρέπει να μπορεί να υποστηρίξει πολλές φυσικές γλώσσες*

Ειναι ΛΑ η ΜΛΑ

- *Ο χρόνος απόκρισης του συστήματος δεν πρέπει να υπερβαίνει τα 3 δευτερόλεπτα*
- *Το σύστημα πρέπει να μπορεί να ενοποιηθεί με το υπάρχον*
- *Τα προσωπικά στοιχεία των πελατών πρέπει να προστατεύονται.*
- *Να αποθηκεύει τα στοιχεία των πελατών*
- *Να τυπώνει συγκεντρωτικές αναφορές*
- *Το σύστημα πρέπει να λειτουργεί αδιάλειπτα (όλο το χρόνο)*
- *Να τυπώνει αποδείξεις*
- *Μόνο οι διευθυντές πρέπει να έχουν πρόσβαση τους μισθούς*
- *Το σύστημα πρέπει συμμορφώνεται με τα πρότυπα της βιομηχανίας*
- *Το σύστημα πρέπει να μπορεί να υποστηρίξει πολλές φυσικές γλώσσες*

A Simple Project Model



User requirements

Functional requirements

Design specifications

Satisfies

Satisfies

Constrains

Tests

Tests

Tests

Standards

Acceptance tests

Functional tests

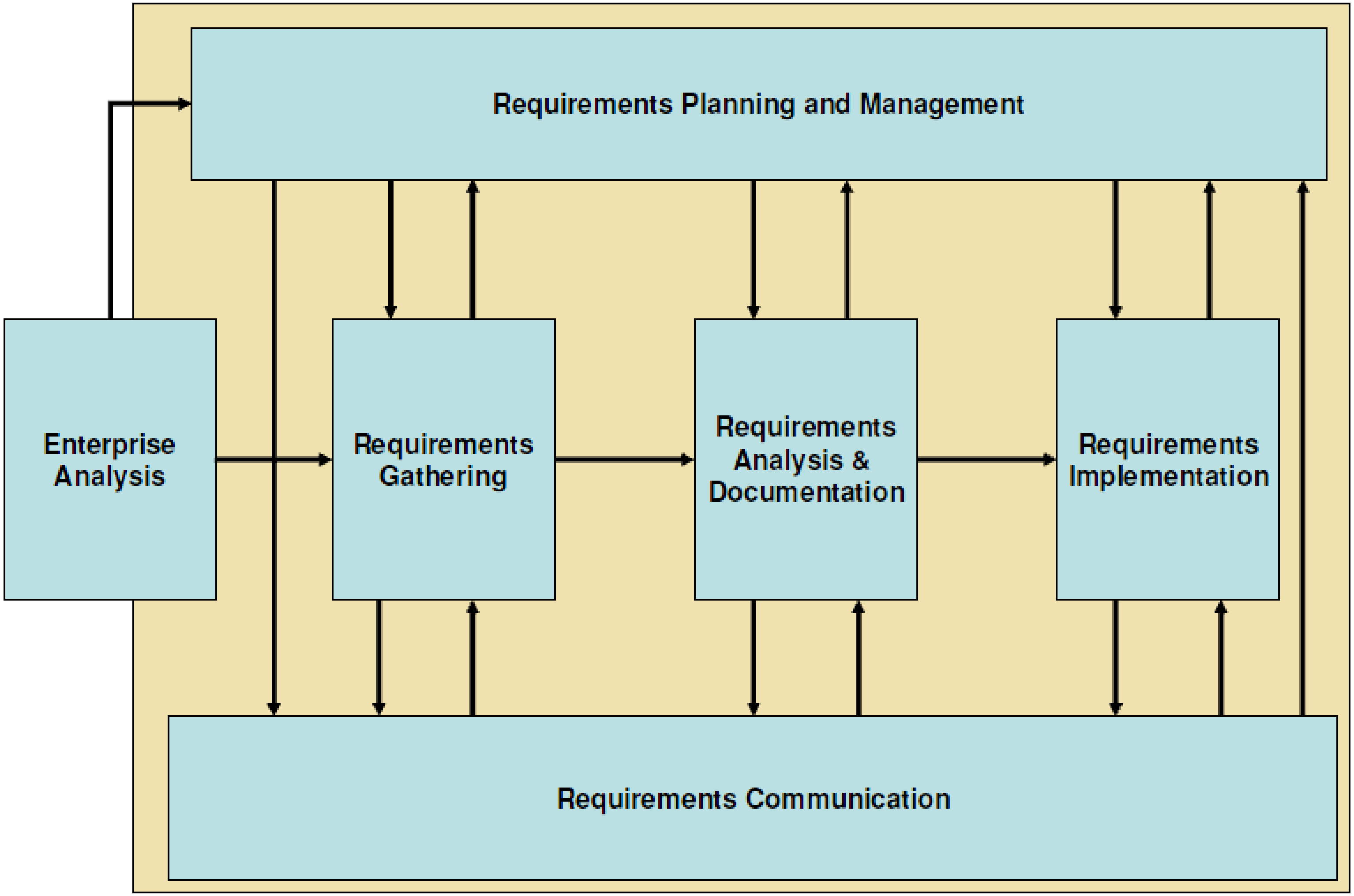
Unit tests

Requirements processes

- Enterprise Analysis
- Requirements Planning and Management
- Requirements Gathering
- Requirements Communication
- Requirements Analysis and Documentation
- Requirements Implementation

Guide to IIBABody of Knowledge, International
Institute of Business Analysis

Requirements processes


Requirements Planning and Management
Enterprise Analysis
Requirements Gathering
Requirements Analysis & Documentation
Requirements Implementation
Requirements Communication

Guide to IIBABody of Knowledge, International Institute of Business Analysis

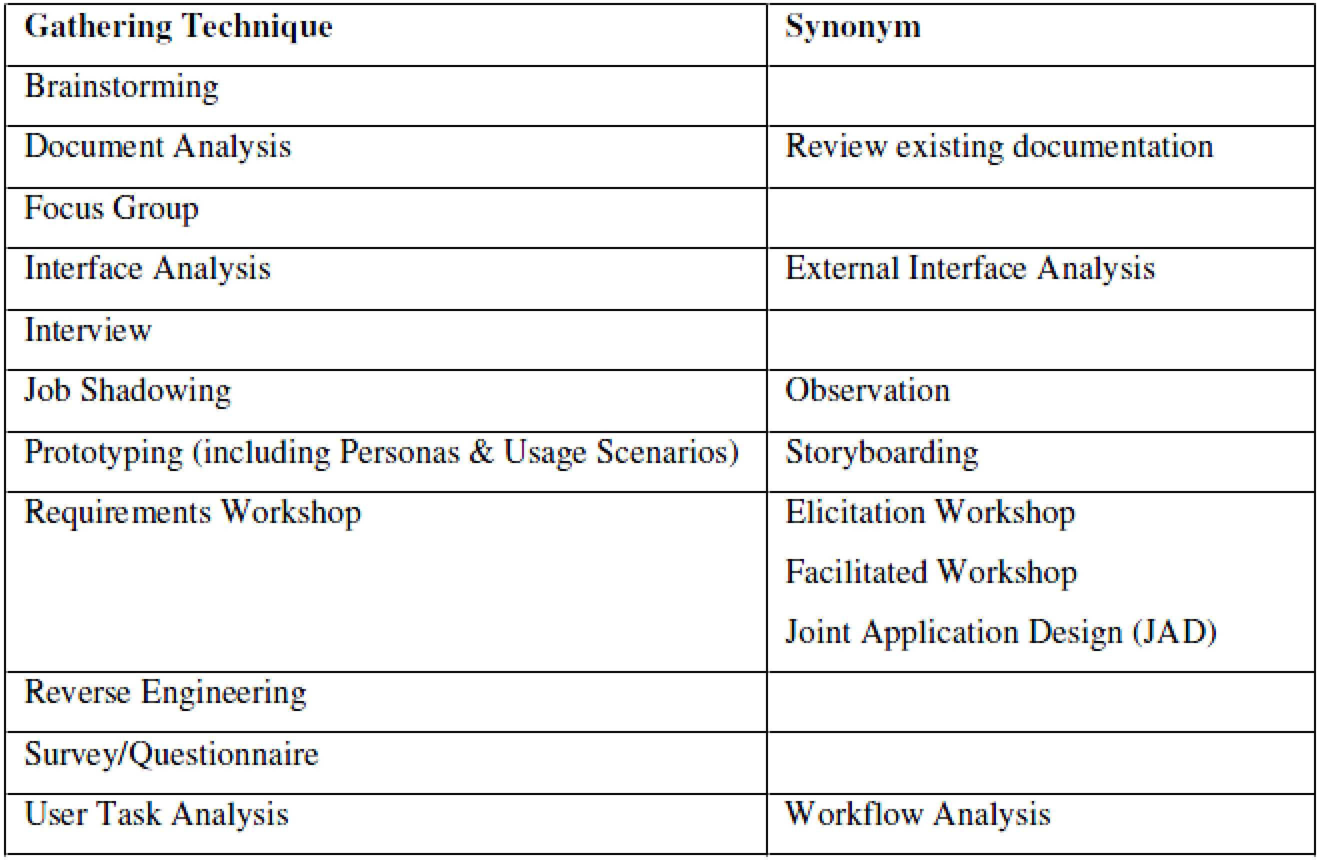
Enterprise Analysis

- Creating and maintaining the business architecture
- Conducting feasibility studies
- Identifying new business opportunities
- Scoping and defining new business opportunities
- Preparing the business case for new business opportunities
- Conducting the initial risk assessment for new business opportunities.

Requirements Planning and Management

- This is necessary to ensure:
  - the set of requirements activities undertaken are the most appropriate, given the unique circumstances of the project,
  - the requirements work effort is coordinated with the other work being done for the project,
  - the whole requirements team on a project has a common understanding of what activities they are undertaking,
  - business analysts are able to monitor and react to requirements challenges and slippage,
  - the tools, resources and requirements contributors are available as needed for the requirements activities,
  - and, changes are captured correctly and consistently.

Requirements Gathering

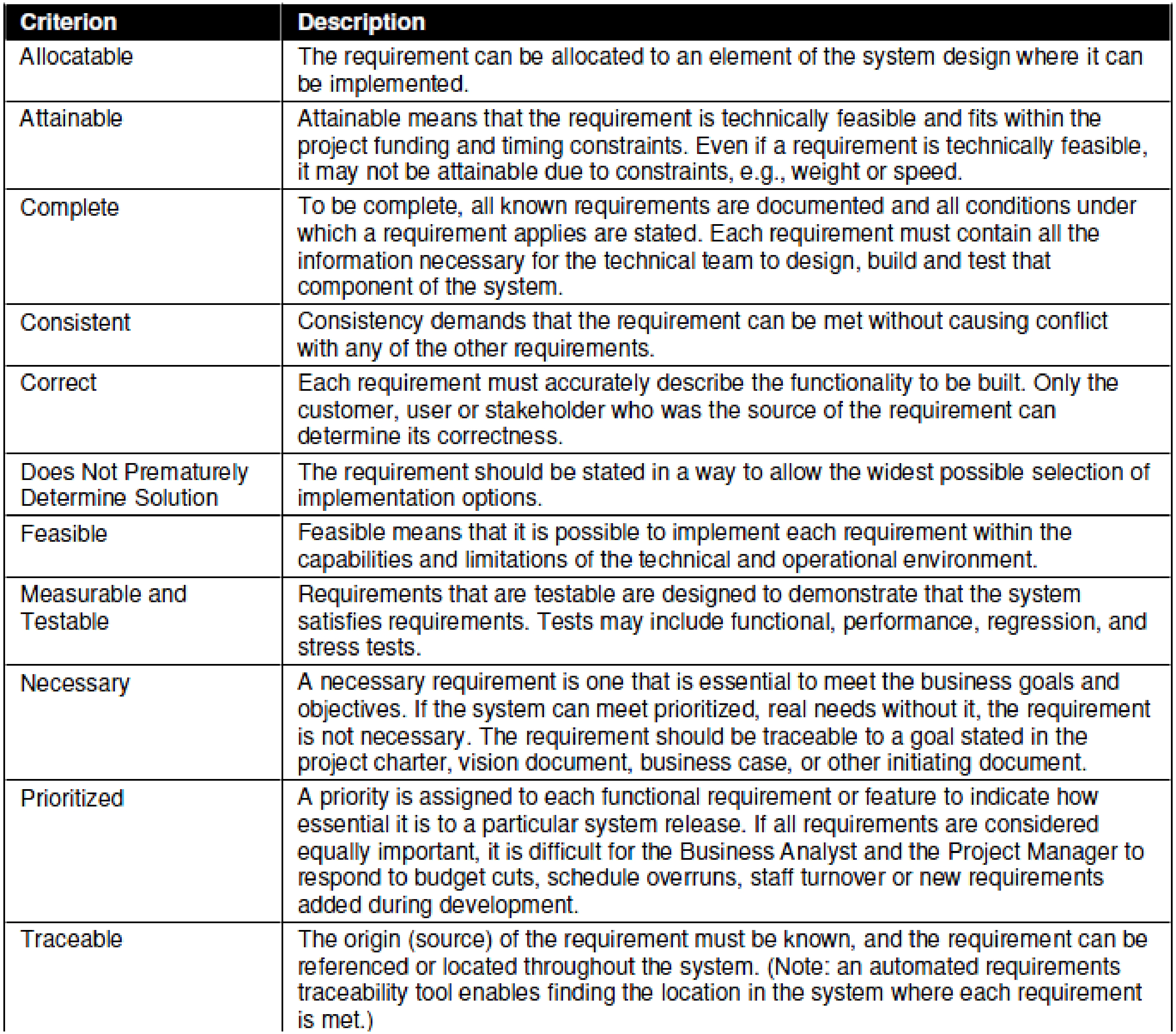


Gathering Technique	Synonym
Brainstorming	
Document Analysis	Review existing documentation
Focus Group	
Interface Analysis	External Interface Analysis
Interview	
Job Shadowing	Observation
Prototyping (including Personas & Usage Scenarios)	Storyboarding
Requirements Workshop	Elicitation Workshop Facilitated Workshop Joint Application Design (JAD)
Reverse Engineering	
Survey/Questionnaire	
User Task Analysis	Workflow Analysis

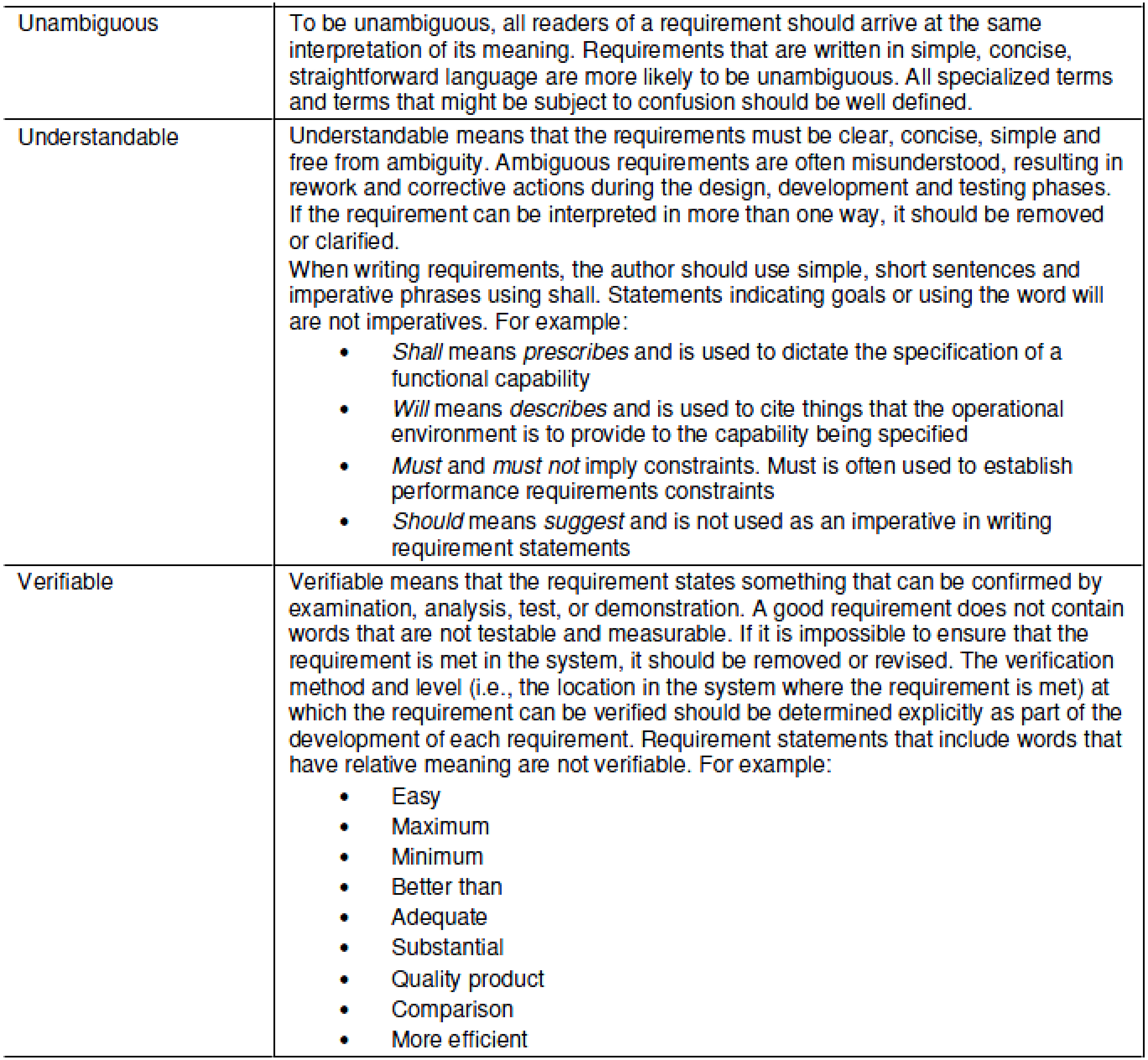
Requirements Analysis and Documentation

- Analyze functional requirements
- Analyze non functional requirements
- Process/flow models
  - Workflow models
  - Flow charts
- Usage models
  - Storyboards,
  - Use cases
- Requirements attributes
- Requirements traceability

Requirements quality

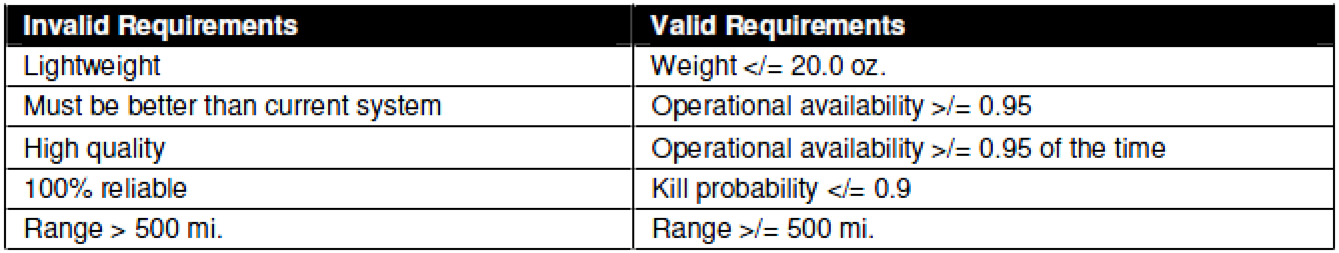


Criterion	Description
Allocatable	The requirement can be allocated to an element of the system design where it can be implemented.
Attainable	Attainable means that the requirement is technically feasible and fits within the project funding and timing constraints. Even if a requirement is technically feasible, it may not be attainable due to constraints, e.g., weight or speed.
Complete	To be complete, all known requirements are documented and all conditions under which a requirement applies are stated. Each requirement must contain all the information necessary for the technical team to design, build and test that component of the system.
Consistent	Consistency demands that the requirement can be met without causing conflict with any of the other requirements.
Correct	Each requirement must accurately describe the functionality to be built. Only the customer, user or stakeholder who was the source of the requirement can determine its correctness.
Does Not Prematurely Determine Solution	The requirement should be stated in a way to allow the widest possible selection of implementation options.
Feasible	Feasible means that it is possible to implement each requirement within the capabilities and limitations of the technical and operational environment.
Measurable and Testable	Requirements that are testable are designed to demonstrate that the system satisfies requirements. Tests may include functional, performance, regression, and stress tests.
Necessary	A necessary requirement is one that is essential to meet the business goals and objectives. If the system can meet prioritized, real needs without it, the requirement is not necessary. The requirement should be traceable to a goal stated in the project charter, vision document, business case, or other initiating document.
Prioritized	A priority is assigned to each functional requirement or feature to indicate how essential it is to a particular system release. If all requirements are considered equally important, it is difficult for the Business Analyst and the Project Manager to respond to budget cuts, schedule overruns, staff turnover or new requirements added during development.
Traceable	The origin (source) of the requirement must be known, and the requirement can be referenced or located throughout the system. (Note: an automated requirements traceability tool enables finding the location in the system where each requirement is met.)



Unambiguous	To be unambiguous, all readers of a requirement should arrive at the same interpretation of its meaning. Requirements that are written in simple, concise, straightforward language are more likely to be unambiguous. All specialized terms and terms that might be subject to confusion should be well defined.
Understandable	Understandable means that the requirements must be clear, concise, simple and free from ambiguity. Ambiguous requirements are often misunderstood, resulting in rework and corrective actions during the design, development and testing phases. If the requirement can be interpreted in more than one way, it should be removed or clarified. When writing requirements, the author should use simple, short sentences and imperative phrases using shall. Statements indicating goals or using the word will are not imperatives. For example: • *Shall* means *prescribes* and is used to dictate the specification of a functional capability • *Will* means *describes* and is used to cite things that the operational environment is to provide to the capability being specified • *Must* and *must not* imply constraints. Must is often used to establish performance requirements constraints • *Should* means *suggest* and is not used as an imperative in writing requirement statements
Verifiable	Verifiable means that the requirement states something that can be confirmed by examination, analysis, test, or demonstration. A good requirement does not contain words that are not testable and measurable. If it is impossible to ensure that the requirement is met in the system, it should be removed or revised. The verification method and level (i.e., the location in the system where the requirement is met) at which the requirement can be verified should be determined explicitly as part of the development of each requirement. Requirement statements that include words that have relative meaning are not verifiable. For example: • Easy • Maximum • Minimum • Better than • Adequate • Substantial • Quality product • Comparison • More efficient

Examples



Invalid Requirements	Valid Requirements
Lightweight	Weight </= 20.0 oz.
Must be better than current system	Operational availability >/= 0.95
High quality	Operational availability >/= 0.95 of the time
100% reliable	Kill probability </= 0.9
Range > 500 mi.	Range >/= 500 mi.

Basic RM Database Attributes

- Requirement Catalogue Numbers
- Requirement Text
- Comments
- Creation Date
- Last Modified Date
- Priority
- Status

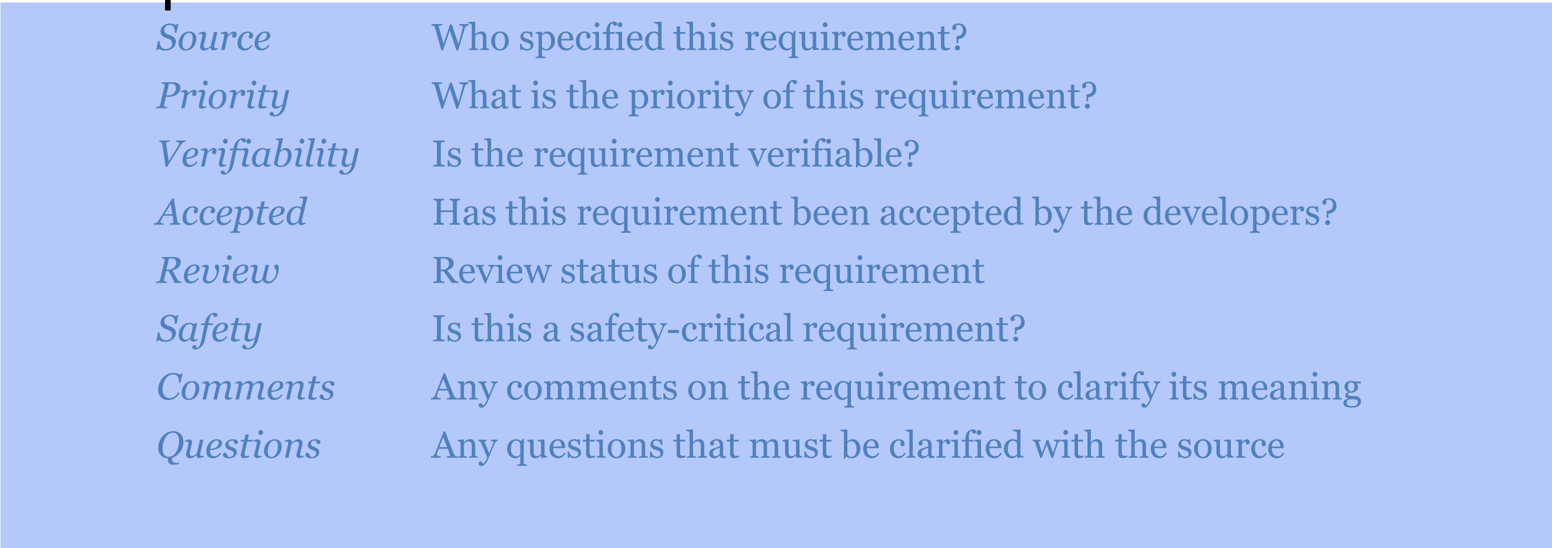
- Revision Identification
- Test Plan Identification
- Test Methodology
- Design Derived (yes or no)

Example Database-Generated Build Schedule Report

SSDD ID	CSCI	SSDD Function	Bl	In	SSDD Text
SS1000-a-	CPM	DISTRIBUTE_DATA	1	2	The ACMS shall use Milstar systems time on the ACMS to ACMS interface as specified in the ACMS to ACMS EXTIRS-B1-U-XXXX.
SS1000-b-	CPM	DISTRIBUTE_DATA	1	2	The ACMS shall use UTC on the ACMS to ACMS interface as specified in the ACMS to ACMS EXTIRS-B1-U-XXXX.
SS1160a	CPM	DISTRIBUTE_DATA	1	2	The ACMS shall interface to other ACMSs as specified in the ACMS to ACMS EXTIRS-B1-U-XXXX.
SS1160b	SOE	MANAGE_DATABASE	1	2	The ACMS shall interface to other ACMSs as specified in the ACMS to ACMS EXTIRS-B1-U-XXXX.
SS1160c	SOE	PROVIDE_SOE_SERVICES	1	2	The ACMS shall interface to other ACMSs as specified in the ACMS to ACMS EXTIRS-B1-U-XXXX.
SS2000-b-	CPM	DISTRIBUTE_DATA	1	1	The ACMS shall exchange communication planning data with other ACMSs as specified in the ACMS to ACMS EXTIRS-B1-U-XXXX.
SS2000-b-	CPM	DISTRIBUTE_DATA	1	2	The ACMS shall exchange communication planning data with other ACMSs as specified in the ACMS to ACMS EXTIRS-B1-U-XXXX.
SS2000-b-	CPM	DISTRIBUTE_DATA	1	3	The ACMS shall exchange communication planning data with other ACMSs as specified in the ACMS to ACMS EXTIRS-B1-U-XXXX.

DOORS tool -What is an Attribute?

- Attributes are additional defined characteristics of a requirement; they provide essential information in addition to requirement text



Source	Who specified this requirement?
Priority	What is the priority of this requirement?
Verifiability	Is the requirement verifiable?
Accepted	Has this requirement been accepted by the developers?
Review	Review status of this requirement
Safety	Is this a safety-critical requirement?
Comments	Any comments on the requirement to clarify its meaning
Questions	Any questions that must be clarified with the source

- You can define attributes that will support your process and make your database more productive for you

Writing Good Requirements

- Writing what people think they need is easy, *but mostly off the mark*
- The hard work is:
  - Knowing what to write about
  - Determining appropriate values
  - Achieving balance between:
    - » Ensuring specificity (explicit conditions) and avoiding ambiguity
    - » Ensuring completeness and sufficiency (adequate detail to fulfill needs) without redundancy

Software Project Survival Guide, *by Steve McConnell*

Writing Good Requirements (2)

- Understand who your audience is
  - Who is the user of the requirement?
  - Who is the recipient of the functionality?
  - Who verifies requirement?
  - Who validates requirement?
- Know the language and the business sector (e.g., aerospace, forces afloat);
  - Create primitive requirement statements; focus on attributes
  - Move up to simple complete sentences: single ideas per sentence are easier to verify and test
  - Decompose more complex sentences into simple single sentences
  - Elaborate with follow-on comments
- Rigid compliance with a format
- Develop and review “Use Cases” as ways of describing system behavior under various conditions
- Plan requirements activities and work products
- Manage changes to requirements

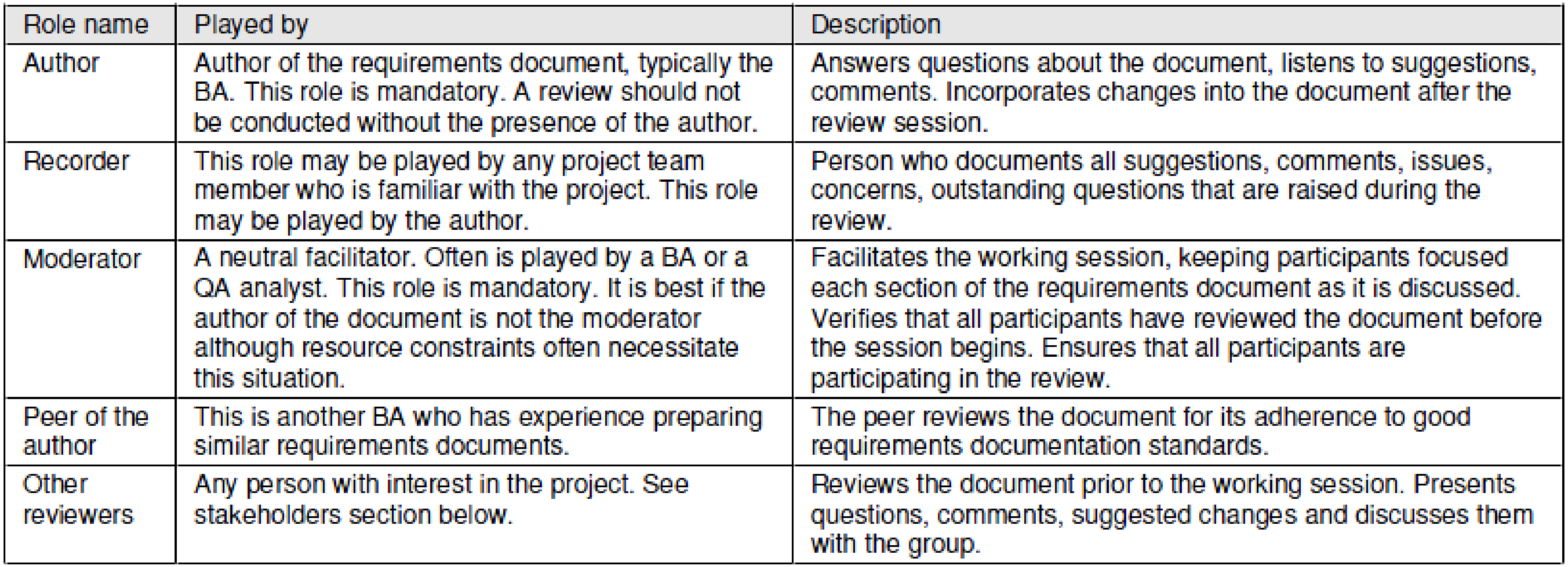
Requirements Communication

- Determine the appropriate requirements format
- Create a requirements package
- Conduct requirements presentation
- Conduct a formal requirements review
- Obtain consensus and signoff on the requirements

Purpose of review

- The purpose of the review should be clearly stated and may encompass any of the following:
  - completeness of requirements (all requirements have been captured)
  - removal of superfluous requirements
  - clarity of requirements (removal of ambiguity)
  - correctness of requirements (the requirement reflects the business need or business rule)
  - scope (the requirement fits within the stated scope of the project)
  - conformance to project/organisational quality standards
  - feasibility of requirements
  - prioritization of requirements

Roles in review

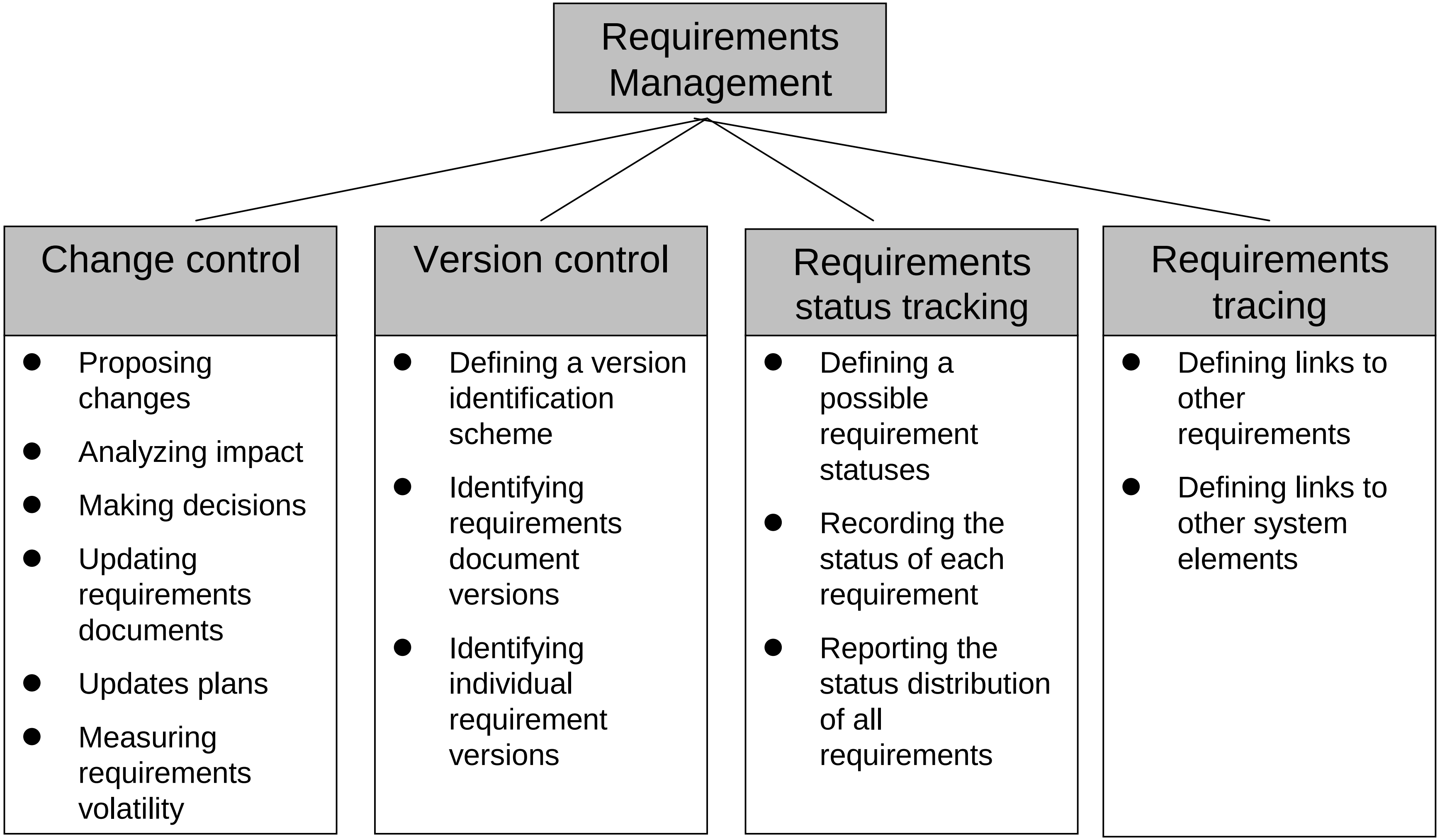


Role name	Played by	Description
Author	Author of the requirements document, typically the BA. This role is mandatory. A review should not be conducted without the presence of the author.	Answers questions about the document, listens to suggestions, comments. Incorporates changes into the document after the review session.
Recorder	This role may be played by any project team member who is familiar with the project. This role may be played by the author.	Person who documents all suggestions, comments, issues, concerns, outstanding questions that are raised during the review.
Moderator	A neutral facilitator. Often is played by a BA or a QA analyst. This role is mandatory. It is best if the author of the document is not the moderator although resource constraints often necessitate this situation.	Facilitates the working session, keeping participants focused each section of the requirements document as it is discussed. Verifies that all participants have reviewed the document before the session begins. Ensures that all participants are participating in the review.
Peer of the author	This is another BA who has experience preparing similar requirements documents.	The peer reviews the document for its adherence to good requirements documentation standards.
Other reviewers	Any person with interest in the project. See stakeholders section below.	Reviews the document prior to the working session. Presents questions, comments, suggested changes and discusses them with the group.

Requirements Implementation

- Develop alternate solutions
- Evaluate technology options
- Facilitate the selection of a solution
- Ensure the usability of the solution
- Support the Quality Assurance process
- Support the implementation of the solution
- Communicate the solution impacts

Requirements Management SUPPORT Activities



Requirements Management

Change control

- Proposing changes
- Analyzing impact
- Making decisions
- Updating requirements documents
- Updates plans
- Measuring requirements volatility

Version control

- Defining a version identification scheme
- Identifying requirements document versions
- Identifying individual requirement versions

Requirements status tracking

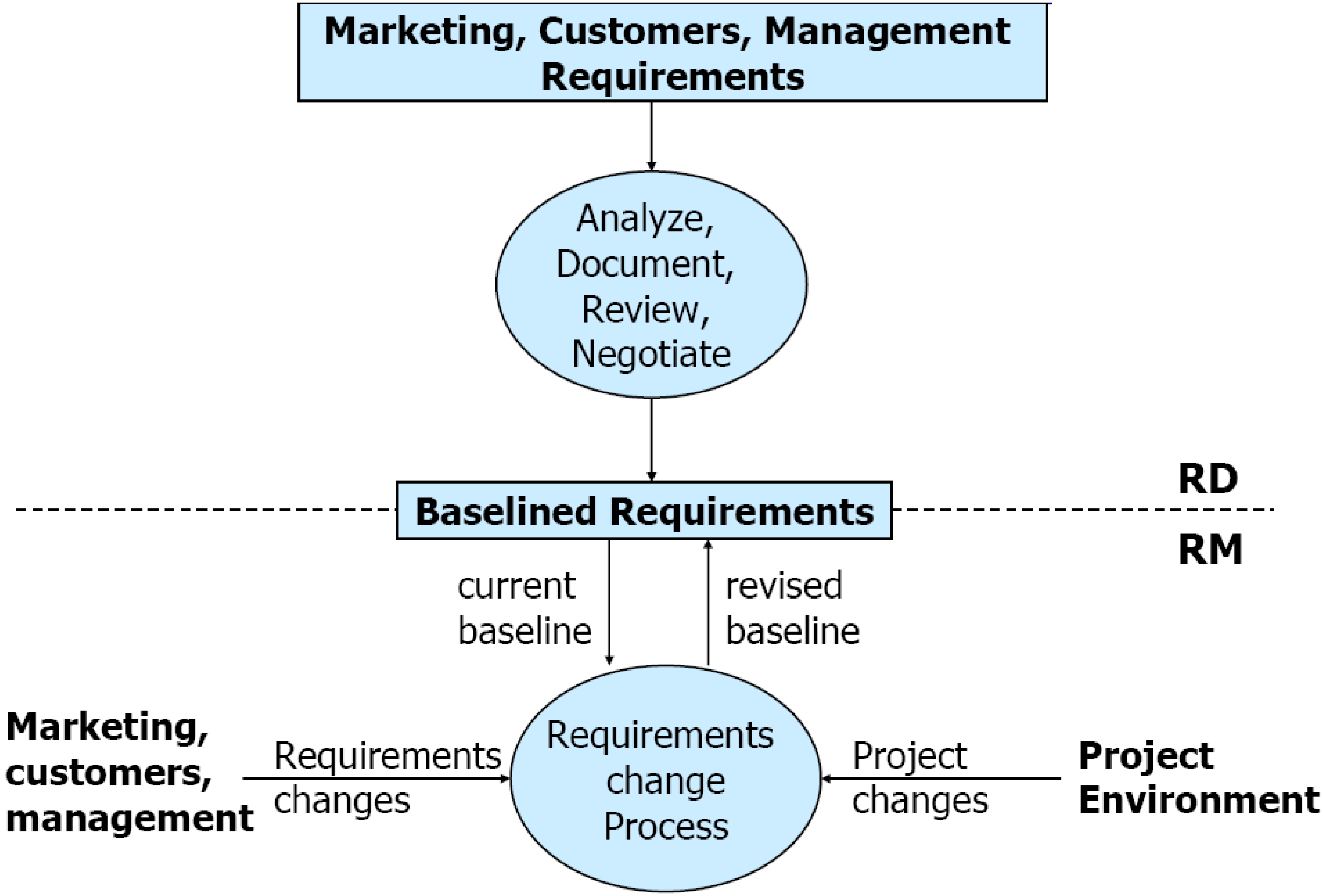
- Defining a possible requirement statuses
- Recording the status of each requirement
- Reporting the status distribution of all requirements

Requirements tracing

- Defining links to other requirements
- Defining links to other system elements

Source: Wiegers, 1999

Requirements Development (RD) and Management (RM)


Marketing, Customers, Management Requirements
Analyze, Document, Review, Negotiate
RD
Baselined Requirements
RM
current baseline
revised baseline
Marketing, customers, management
Requirements changes
Requirements change Process
Project changes
Project Environment

Source: Wiegers, 1999

From Management to Tools

- Changes lead to a need for management
- There is no management without:
  - Traceability
  - Baselines enabling comparisons
- From a practical point of view, there is no traceability or management without appropriate tools

In theory, practice and theory are similar…
But in practice they are different ☺

Requirements Change Factors (1)

- Requirements errors, conflicts, and inconsistencies
  - May be detected at any phase (when requirements are analyzed, specified, validated, or implemented)
- Evolving customer/user knowledge of the system
  - When the requirements are developed, customers/users simultaneously develop a better understanding of what they really need
- Technical, schedule, or cost problems
  - Difficult to plan and know everything in advance
  - We may have to revisit the list of requirements and adapt it to the current situation

Requirements Change Factors (2)

- Changing customer priorities, new needs
  - May be caused by a change in the system environment (technological, business, political…), i.e., the context
  - Business and strategic goals may change
  - May be caused by the arrival of a new competitor
  - Laws and regulations may change
  - Collaborating systems may change
  - May also be caused by technology changes in the enterprise (migration to a new operating system, DBMS…)
  - May be caused by organizational changes (organizational structure, business processes, employees…)

Requirements Volatility

- Requirements continuously change
  - While the requirements are being elicited, analyzed, specified, and validated and after the system has gone into service
- Some requirements are usually more subject to change than others
  - Stable requirements are concerned with the essence of a system and its application domain
    - Derived from the client's principal business activities or the domain model
    - They change more slowly than volatile requirements
    - E.g., a hospital will always have doctors, nurses, patients…
  - Volatile requirements are specific to the instantiation of the system in a particular environment for a particular customer at a particular time
    - E.g., in a hospital, we can think of requirements related to the policies of the government health system

Types of Volatile Requirements

- Mutable requirements
  - These are requirements which change because of changes to the environment in which the system is operating
- Emergent requirements
  - These are requirements which cannot be completely defined when the system is specified but which emerge as the system is designed and implemented
- Consequential requirements
  - These are requirements which are based on assumptions about how the system will be used
  - Once the system is in place, some of these assumptions will be wrong
- Compatibility requirements
  - These are requirements which depend on other equipment, technology, or processes

Expectations of Requirements Management (1)

- Identification of individual requirements
- Traceability from highest level requirements to implementation
  - Established via links through a requirements database
  - Links between requirements and design models, tests, code…
  - Coverage and consistency analysis
  - What are the traceability policies? What types of links? From where? To where?
- Impact assessments of proposed changes
  - Analysis tools let you see which other requirements (and other linked artifacts) will be affected by a change

Expectations of Requirements Management (2)

- Controlled access to current project information
  - A shared database ensures that all users are working with current data (consistency, parallel access)
  - A central repository allows all users to see the information that they need to see (visibility)
- Change control
  - Change proposal system implements controlled process for managing change
  - How do we collect, document, and address changes?
- Deployment of required tool support
  - To help manage requirements change

Requirements Statuses

- Help manage the requirement lifecycle
  - Their number and nature depend on the process in place
- Example of a set of statuses:
  - Proposed: by some stakeholder
  - Approved: part of baseline, committed to implement
  - Rejected: after evaluation
  - Implemented: designed and implemented
  - Verified: Relevant tests have passed
  - Deleted: Removed from list
- RM includes amongst its tasks the tracking of the status of all requirements during the project

Version Control

- Another essential aspect of requirements management
  - Every version of a requirement needs to be uniquely identified
  - The last version of a requirement must be available to all team members
  - Changes need to be documented and clearly communicated
  - A version identifier must be updated with every change to the requirement
- Requirements documents should include
  - A revision history: changes, dates, by whom, why…
  - Standard markers for revisions (e.g., strikethrough or underlined text, coloring, line markers…)
- Version control tool may be used
  - To store and manage the revision history
  - To store justifications (to add, modify, delete, reject a requirement)

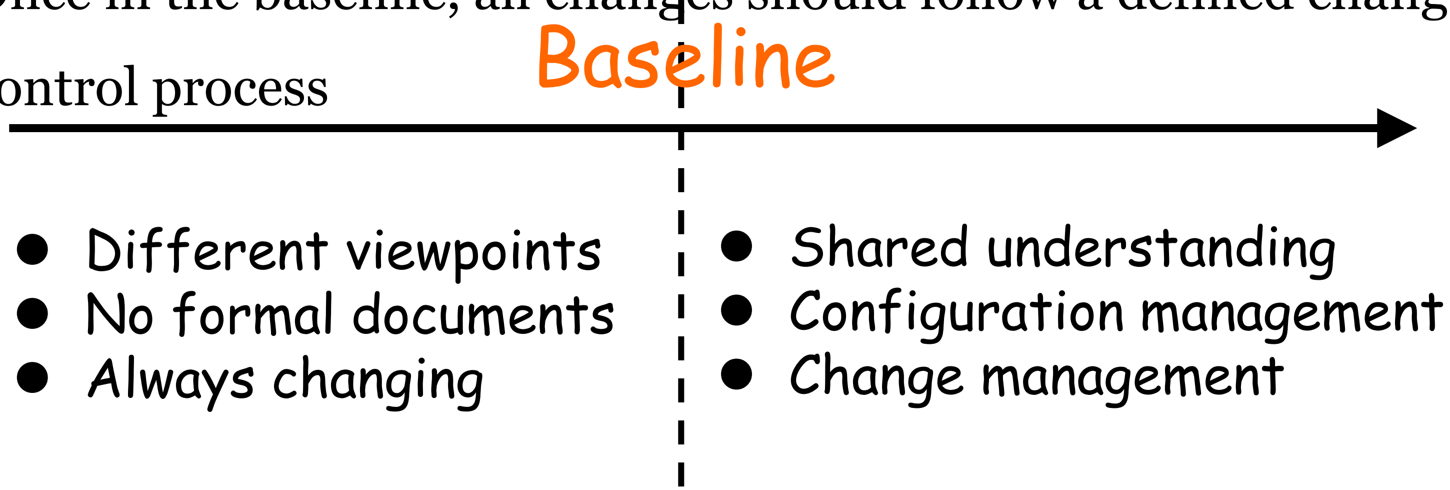
Baseline

- Non-modifiable (read-only) version of a document
  - Describes a moment in time
  - May include multiple documents at the same time
- Enables document comparison and management
- Comes with a change history for the document
  - Information on objects, attributes, and links created, deleted, or edited since the creation of the baseline
  - Often also contains information on user sessions (when the document was opened, by whom…)
- Requires access control

- It is advisable to establish a baseline for a new document that is imported into the document management system
  - In order not to lose any changes

Baseline for Requirements

- Represents the set of functional and non-functional requirements that the development team has committed to implement in a specific release
- Before going into the baseline, the requirements should be reviewed and approved by stakeholders
- Once in the baseline, all changes should follow a defined change control process



Baseline

- Different viewpoints
- No formal documents
- Always changing

- Shared understanding
- Configuration management
- Change management

Baseline Usage

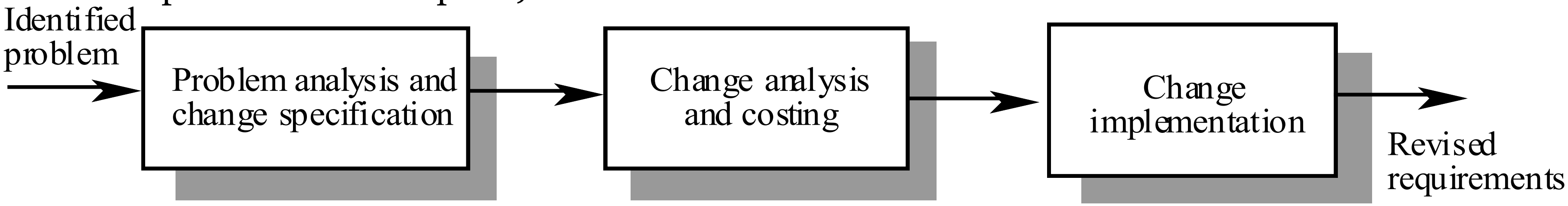
- Baselines may be
  - Created
    - Complete image of requirements state at a given time
  - Deleted
  - Visualized
    - Possibility to go back
  - Compared
    - To see changes since a certain time
  - Copied
  - Signed
    - For authorization, contract

Change Management

- Concerned with the procedures, processes, and standards which are used to manage changes to a system requirements
- Change management policies may cover
  - The change request process and the information required to process each change request
  - The process used to analyse the impact and costs of change and the associated traceability information
  - The membership of the body that formally considers change requests
  - Software support (if any) for the change control process
- A change request may have a status as well as requirements
  - E.g., proposed, rejected, accepted, included…

Change Management Process

- Some requirements problem is identified
  - Could come from an analysis of the requirements, new customer needs, or operational problems with the system
  - The requirements are analysed using problem information and requirements changes are proposed
- The proposed changes are analysed
  - How many requirements (and, if necessary, system components) are affected? Roughly how much would cost, in both time and money?
- The change is implemented
  - A set of amendments to the requirements document or a new document version is produced (of course this should be validated with whatever normal quality checking procedures are in place)


Identified problem
Problem analysis and change specification
Change analysis and costing
Change implementation
Revised requirements

Change Request Form

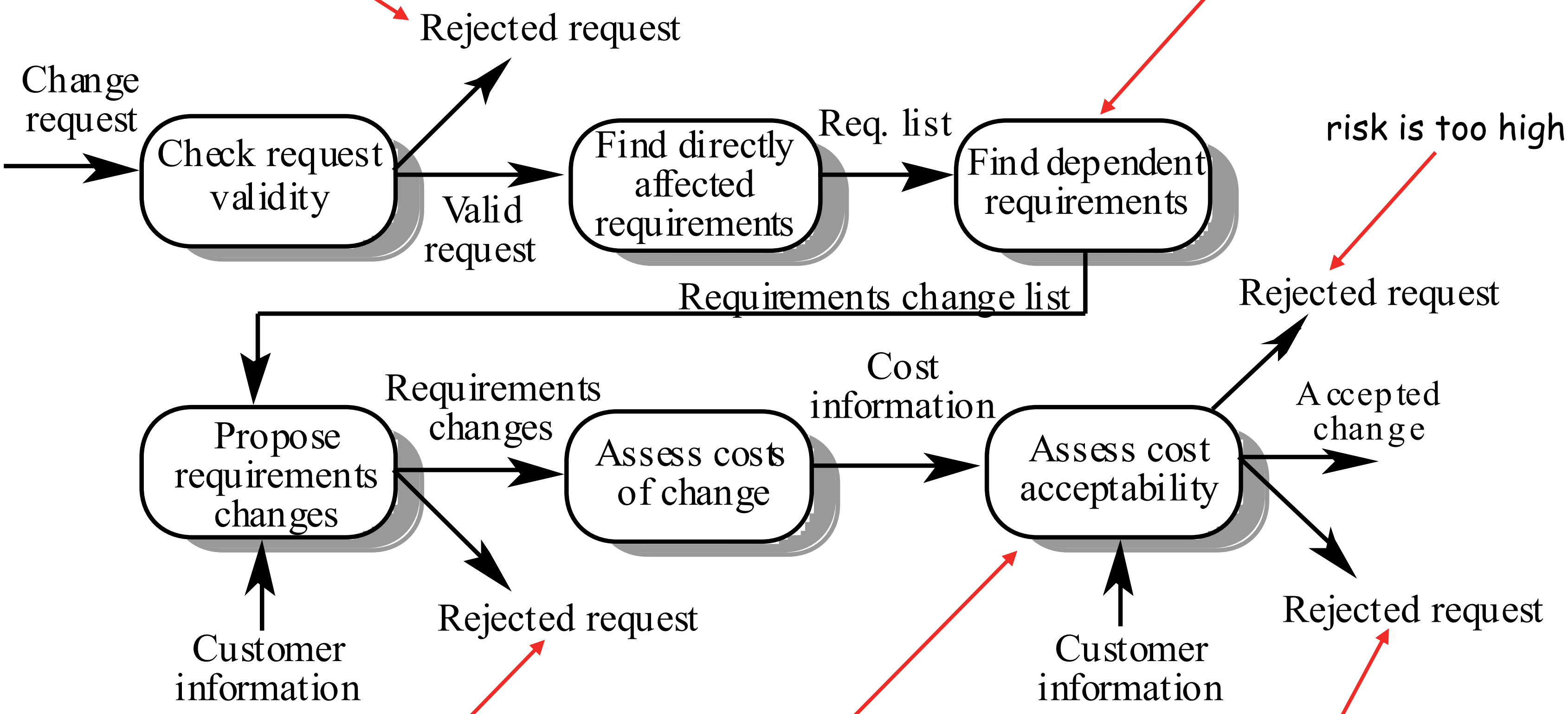
- Proposed changes are usually recorded on a change request form which is then passed to all of the people involved in the analysis of the change
- Change request forms may include
  - Date, Customer, Requester, Product including version
  - Description of change request including rationale
  - Fields to document the change analysis
  - Signature fields
  - Status
  - Comments

Change Analysis and Costing – Example

customers may misunderstand requirements and their context and suggest unnecessary changes

with the help of traceability information

risk is too high



negotiations with customers

consequential changes may be unacceptable to user/customer

cost/time required for the implementation of change is too high/long

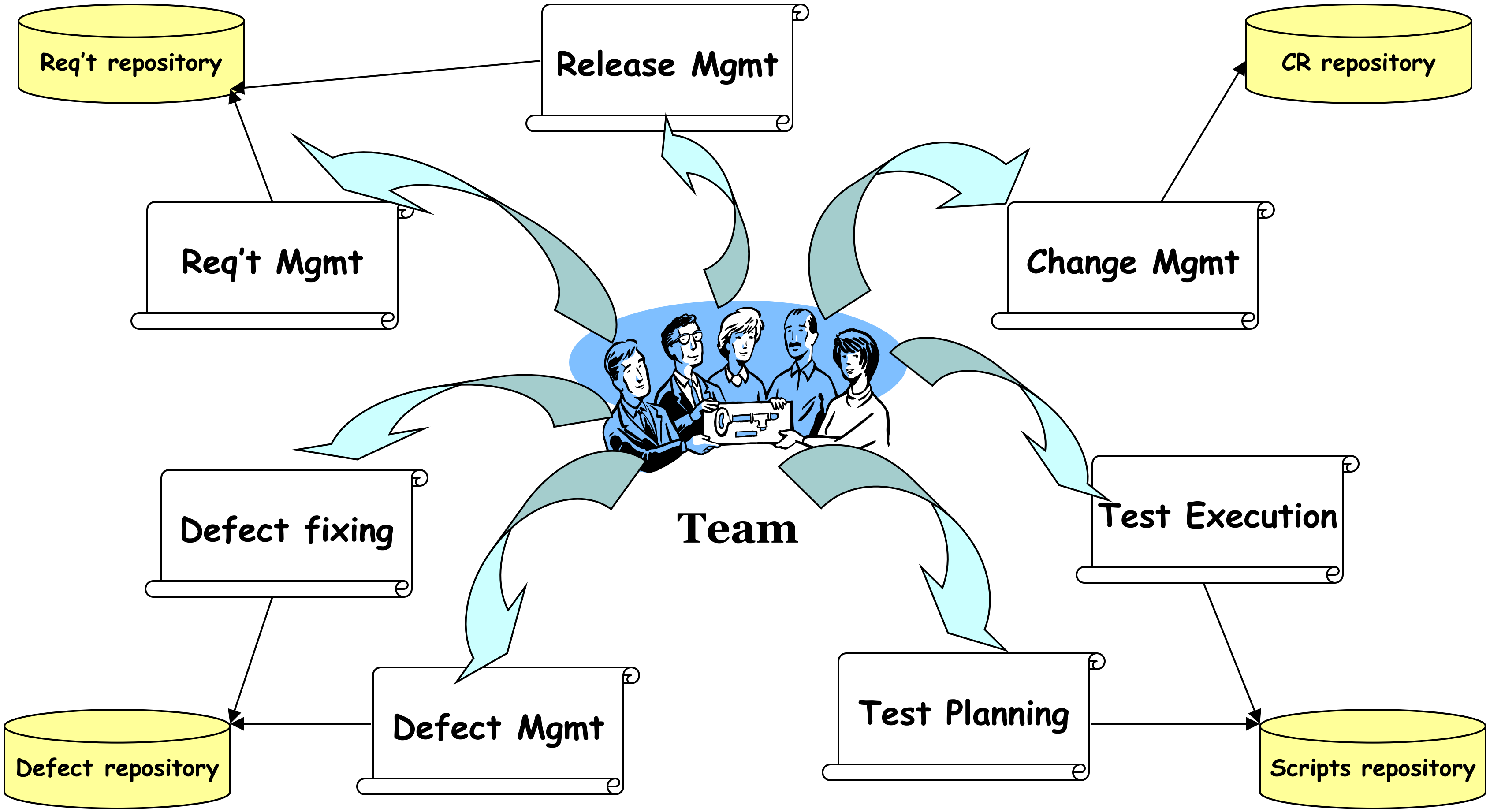
Source of figure: Kotonya and Sommerville

Different Management Aspects

- Change Management
    - How does a customer submit change requests?
    - How is this request being monitored, prioritized, and implemented?
- Configuration Management
    - Versioning, labelling, and tracking code and other components during the development cycle of software
- Release Management
    - Defines how and when different hardware and software will be made available together as a product

Tool Support for Change Management

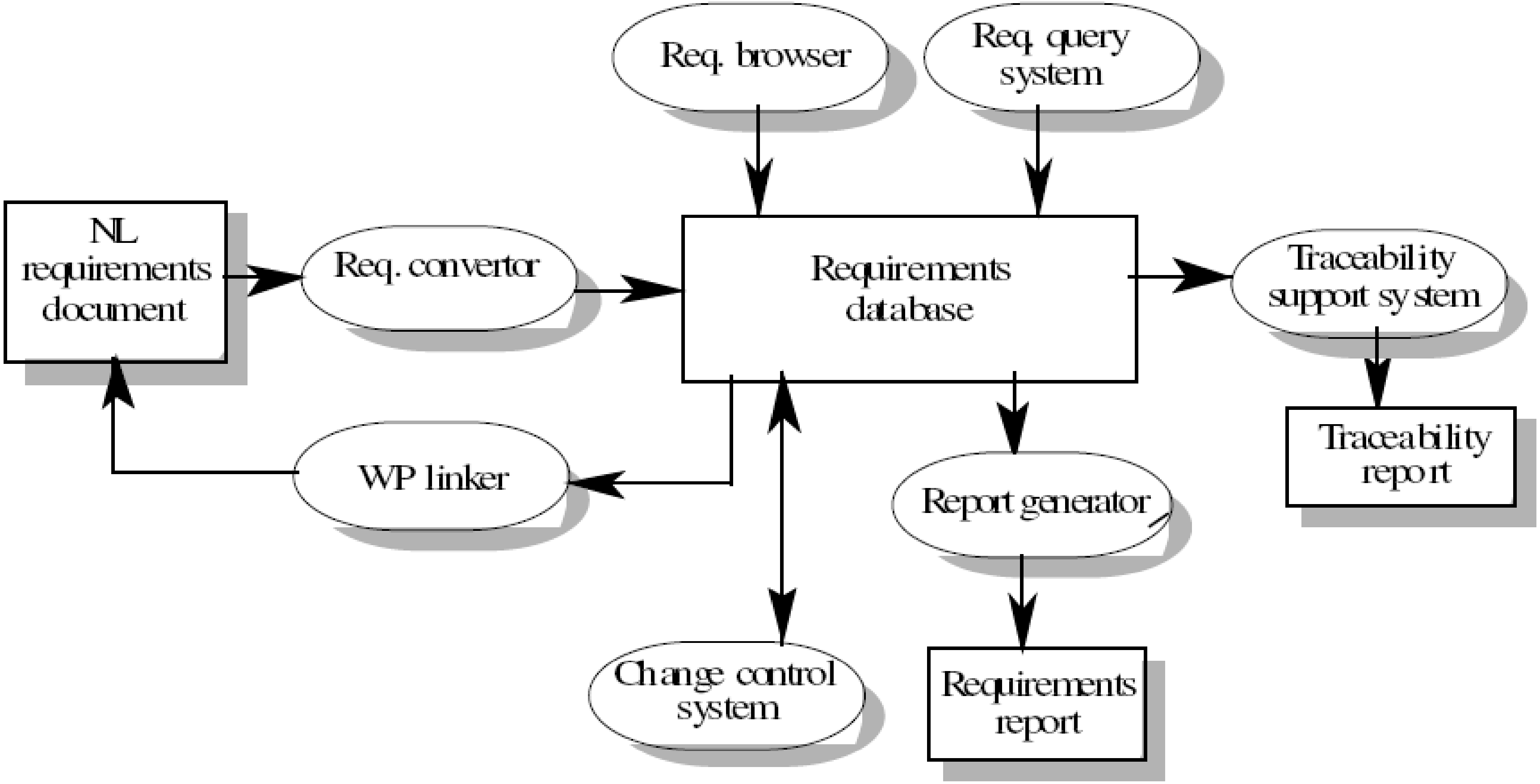
- May be provided through requirements management tools or through configuration management tools
- Tool facilities may include
  - Electronic change request forms which are filled in by different participants in the process
  - A database to store and manage requests
  - A change model which may be instantiated so that people responsible for one stage of the process know who is responsible for the next process activity
  - Electronic transfer of forms between people with different responsibilities and electronic mail notification when activities have been completed
  - Electronic signatures
  - Discussion forums
  - In some cases, direct links to a requirements database


Req't repository
Release Mgmt
CR repository
Req't Mgmt
Change Mgmt
Defect fixing
Team
Test Execution
Defect Mgmt
Test Planning
Defect repository
Scripts repository

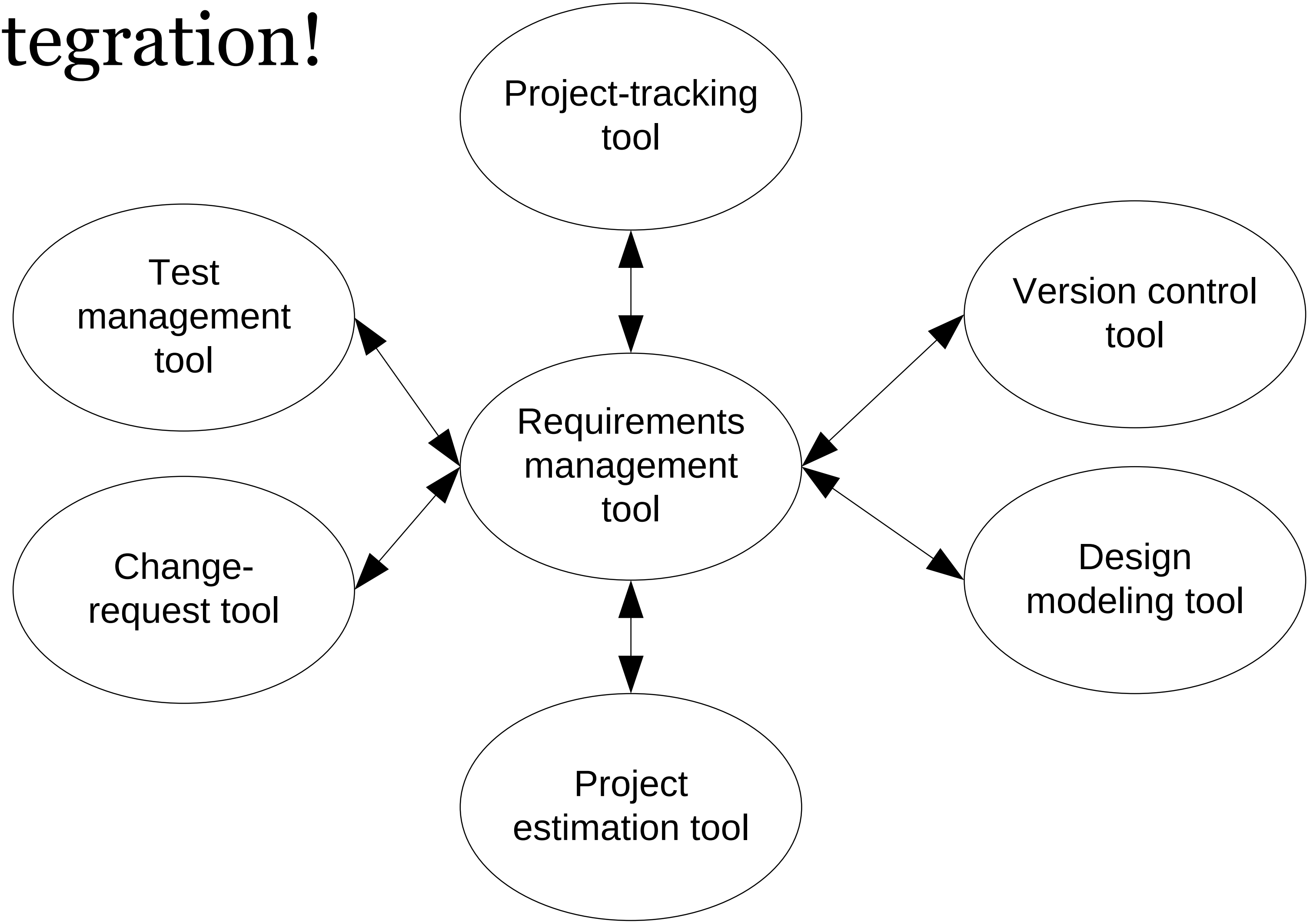
What Kind of Tool Do We Need?

- Different companies will use different tools, which may or may not be tailored to the requirements management task
  - Word processor (Microsoft Word with templates…)
  - Spreadsheet (Microsoft Excel…)
  - Industrial-strength, commercial RM tools
    - IBM/Telelogic DOORS, IBM Requisite Pro, Borland CaliberRM, Visual Paradigm
  - Open source RM tools
    - OSRMT: http://sourceforge.net/projects/osrmt
  - Bug tracking tools (free or not)
    - Bugzilla…
  - Collaboration tools (free or not)
    - TWiki…

RM Tool Architecture – Example



Requirements Management Implies Integration!



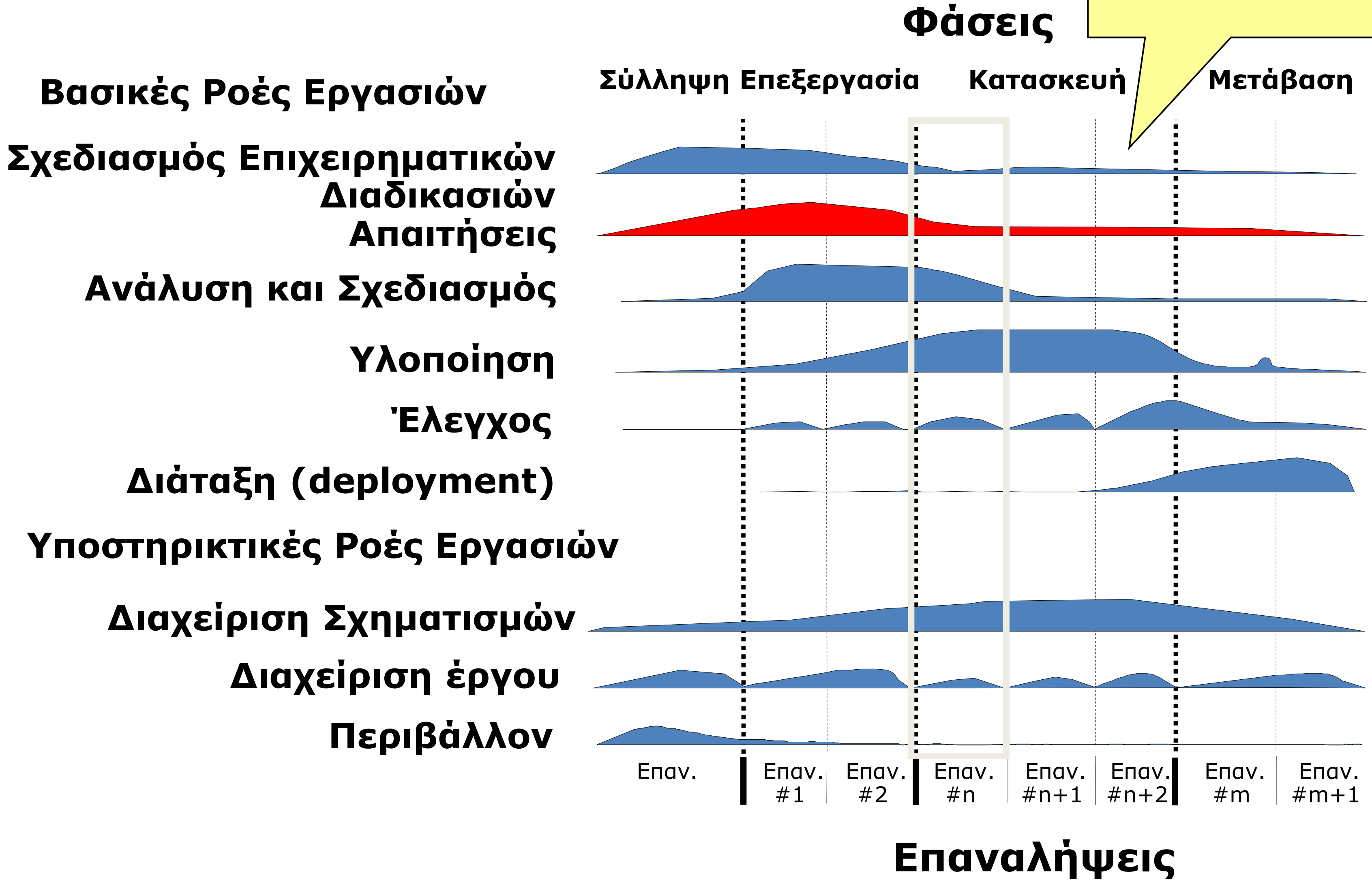
Workshop – Scenario #1

- **The project:**
  The project MyHomeDVD is to deliver a simple web-based movie library management application for the family members who can browse their own records including the movies borrowed and the due dates for the borrowed movies.
- **The project status:**
  The coding phase is completed and the integration test is done. Based on the pre-defined test cases, 2 defects are found. After a thorough investigation, 1 of the 2 defects is not valid – the defect is caused by a missing requirement (e.g. forgot the Logoff button). After negotiating with the key stakeholders, it is agreed that it is critical to implement the missing requirement. As a result, CR101 is raised.
- **The question:**
  If you take over the project right at this moment, what are those key things that you would like to do?
- **Your action items:**
  Please discuss this with your group to come up with a summary. Then each group presents the summary to the audience.
- **Hints:**
  Focus on requirement change management, think of the requirement repository, affected requirements, test plan and test cases.

Requirement management in Unified Process

Η συνολικη εικονα

Σε κάθε φάση εκτελείς όλα τα workflow



Φάσεις

Σύλληψη **Επεξεργασία** **Κατασκευή** **Μετάβαση**

Βασικές Ροές Εργασιών

Σχεδιασμός Επιχειρηματικών Διαδικασιών

Απαιτήσεις

Ανάλυση και Σχεδιασμός

Υλοποίηση

Έλεγχος

Διάταξη (deployment)

Υποστηρικτικές Ροές Εργασιών

Διαχείριση Σχηματισμών

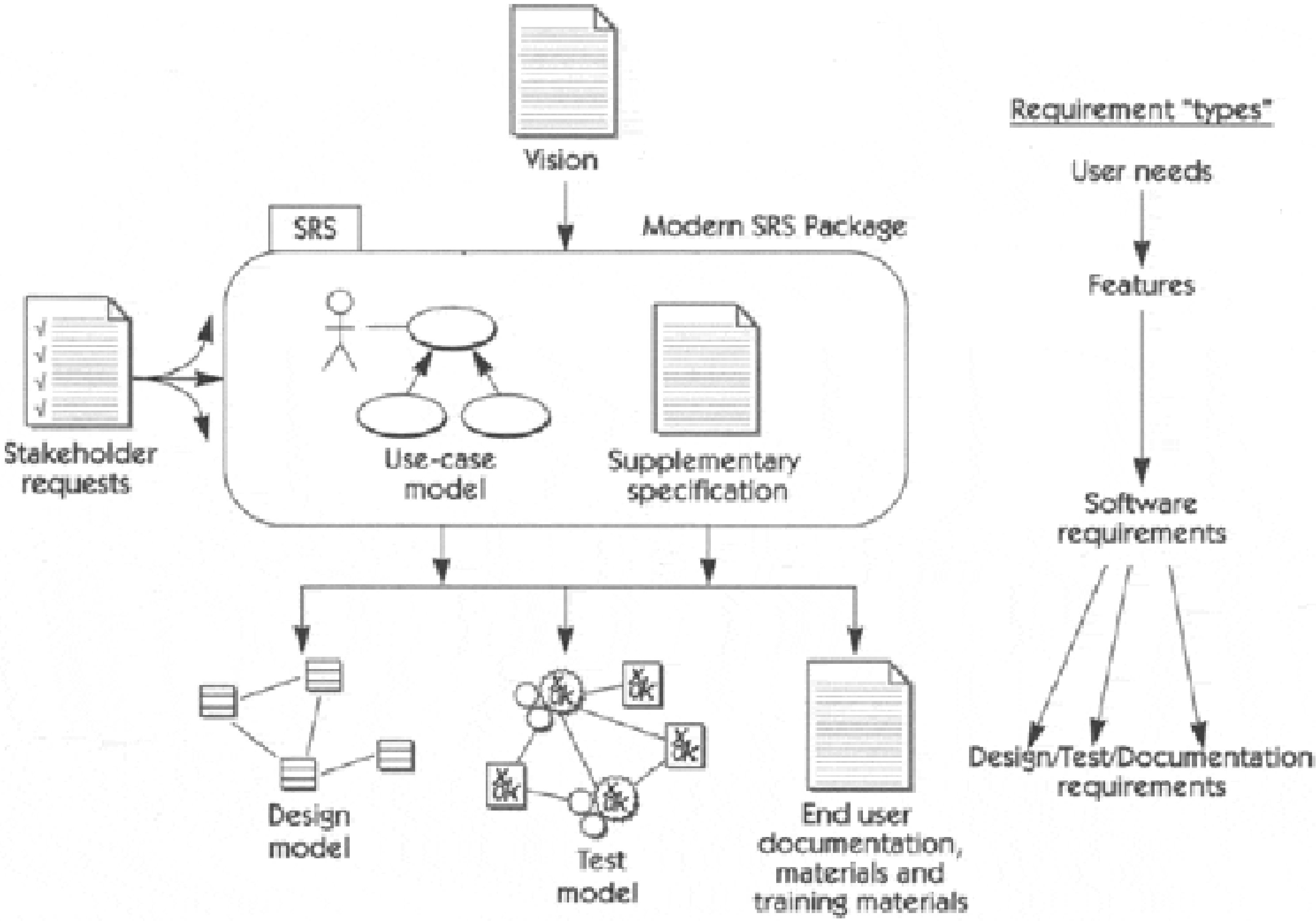
Διαχείριση έργου

Περιβάλλον

Επαν. | Επαν. #1 | Επαν. #2 | Επαν. #n | Επαν. #n+1 | Επαν. #n+2 | Επαν. #m | Επαν. #m+1

Επαναλήψεις

Requirements management in UP



Requirements management in UP

- ***Stakeholder requests***, the collection of any type of requests, including formal change requests, needs, or other input from any stakeholders, during the life cycle of the project, that might affect the product requirements
- The ***Vision document***, which summarizes the overall vision of the system under consideration: main characteristics, major features, key stakeholder needs, and key services provided
- The ***use-case model***, the organized set of use cases that constitute the bulk of the requirements
- The ***supplementary specification***, which captures any requirements that cannot be tied directly to any specific use case, in particular, many of the nonfunctional requirements and design constraints

Other requirements artifacts

- ***Requirements attributes***, a repository of information containing requirements-related information that is used to track requirements status and to maintain traceability to other project elements
- **Use case storyboards**, systematically derived from the essential use cases involving human actors to model the user interface and to elaborate some of the usability requirements
- **User interface prototypes**, developed to get feedback from the various stakeholders
- A **project's glossary**, which captures and defines the terms used in the project domain

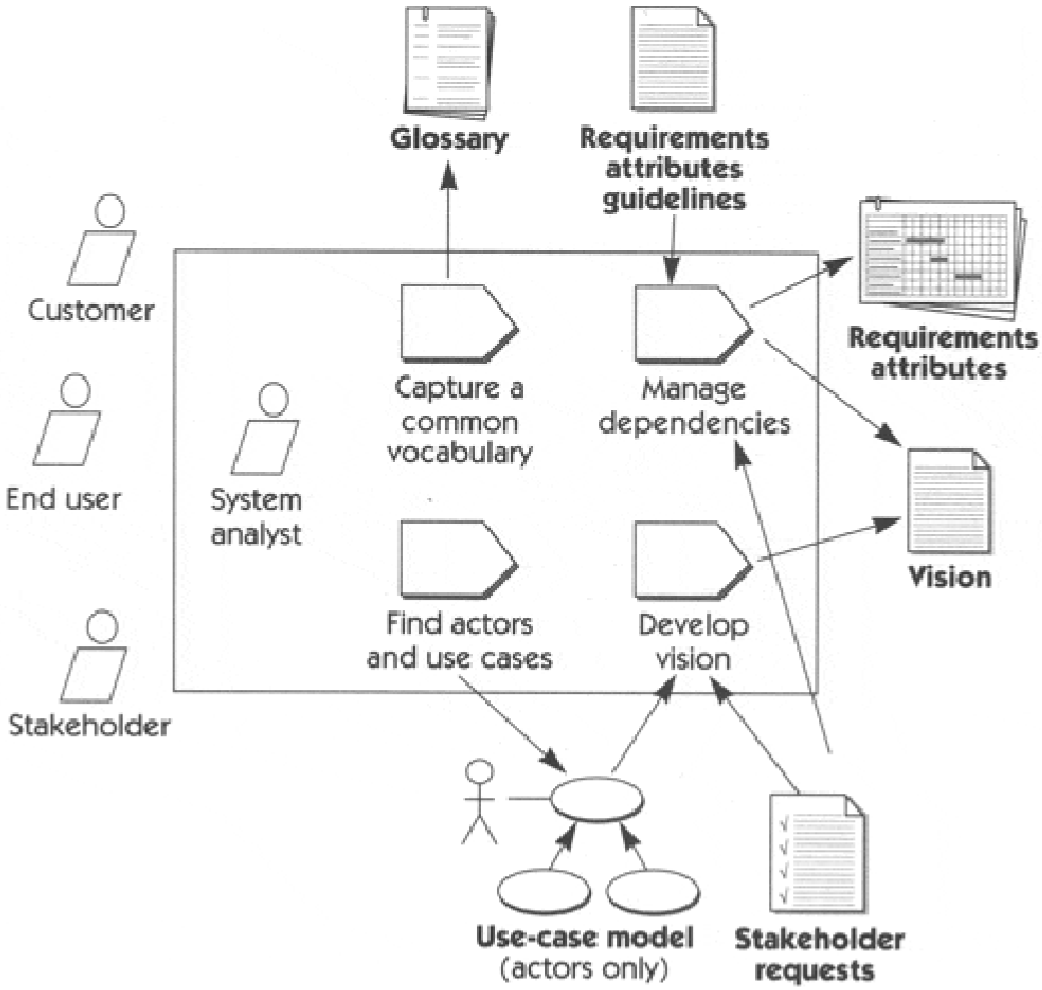
Workers

- ***Stakeholder, customer, end user****,* or whoever within the development organization represents the role of anyone providing input to the requirements process (it is often the marketing manager who plays this role in some companies)
- ***System analyst****,* who leads and coordinates requirements elicitation and use-case modeling by outlining the system's functionality and delimiting the system: for example, establishing what actors and use cases exist and how they interact, along with nonfunctional requirements and design constraints
- ***Use case specifier,*** who details the specification of a part of the system's functionality by describing the requirements aspect of one or several use cases
- ***User interface (UI) designer,*** who develops use case storyboards and UI prototypes and involves other stakeholders in their evaluation
- ***Requirements reviewer*** (a role usually played by several team members), who plans and conducts the formal review of the use-case model and other requirements specified in the supplementary specifications

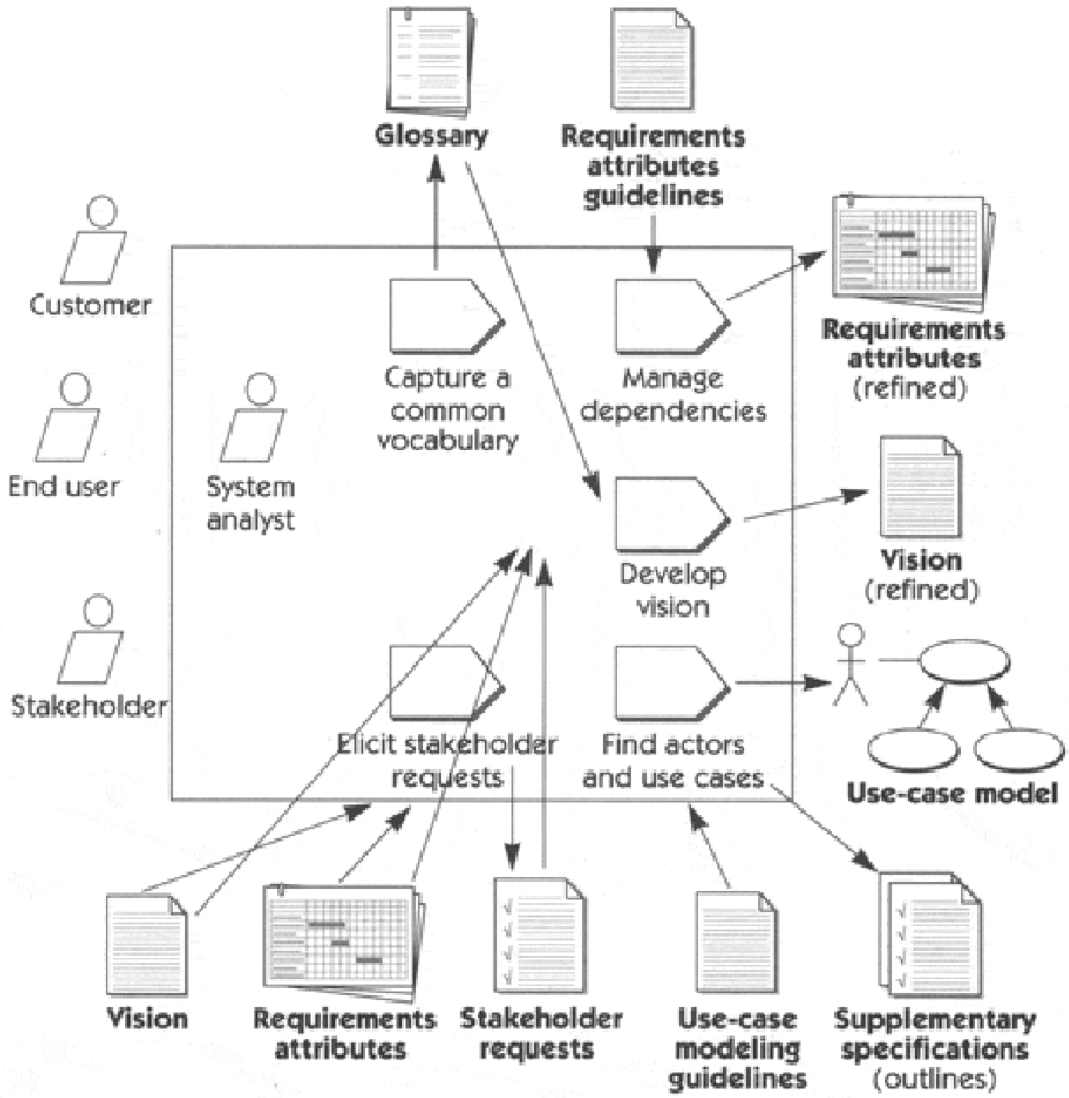
Requirements workflow details

- Analyze the problem
- Understand Stakeholders' Needs
- Define the System
- Manage the Scope of the System
- Refine the System Definition
- Manage Changing Requirements

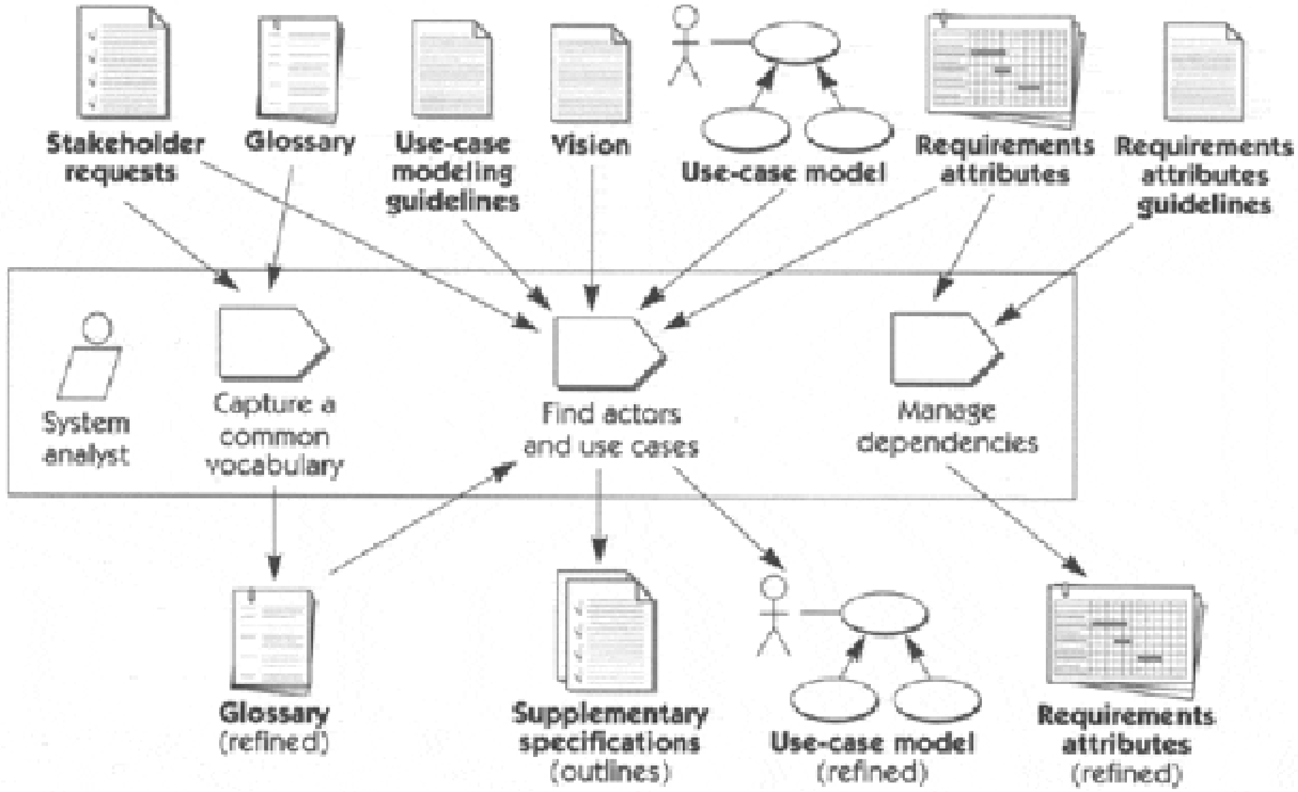
Analyze the Problem



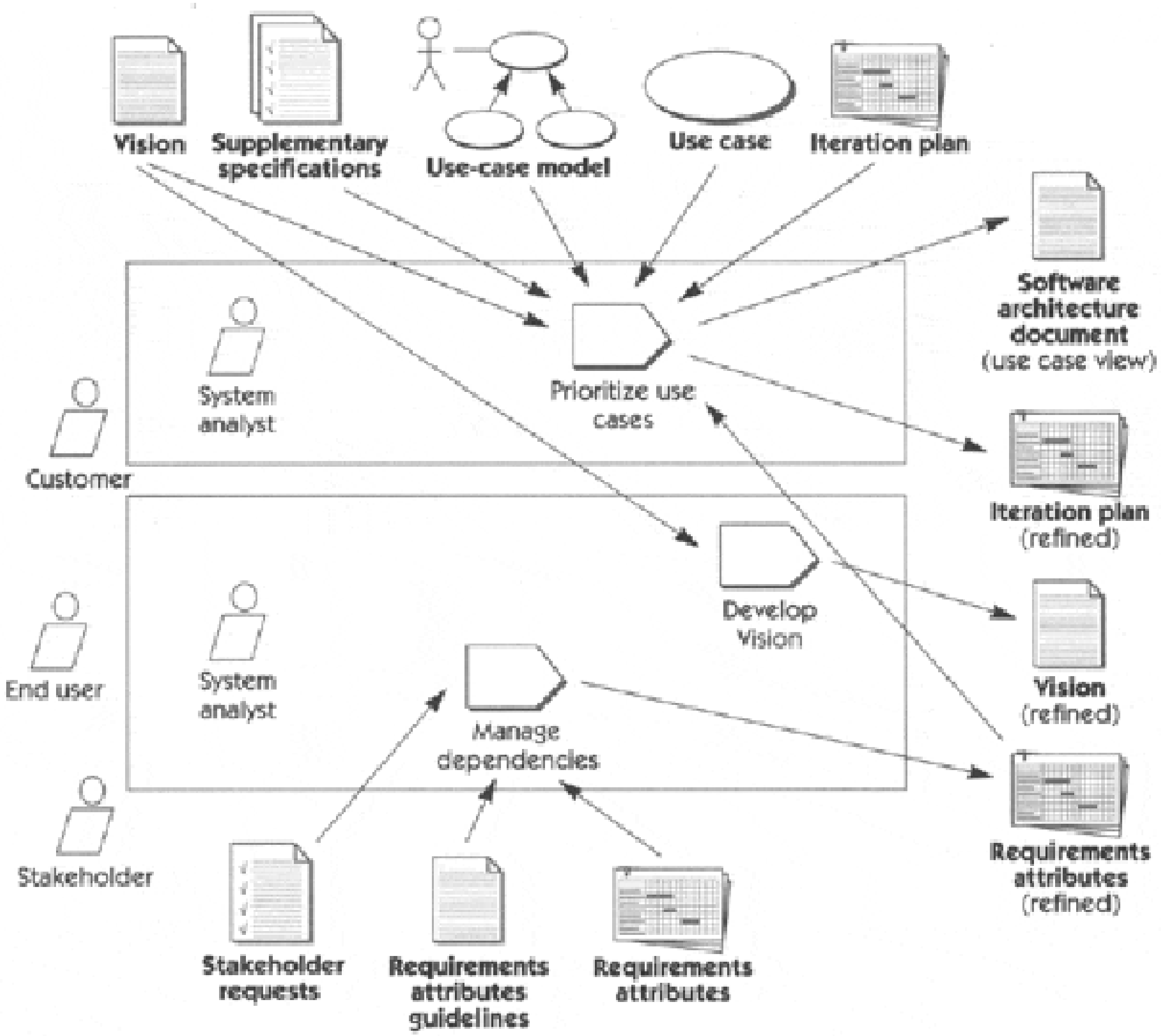
Understand Stakeholders' Needs


Glossary
Requirements attributes guidelines
Customer
End user
System analyst
Stakeholder
Capture a common vocabulary
Manage dependencies
Requirements attributes (refined)
Develop vision
Vision (refined)
Elicit stakeholder requests
Find actors and use cases
Use-case model
Vision
Requirements attributes
Stakeholder requests
Use-case modeling guidelines
Supplementary specifications (outlines)

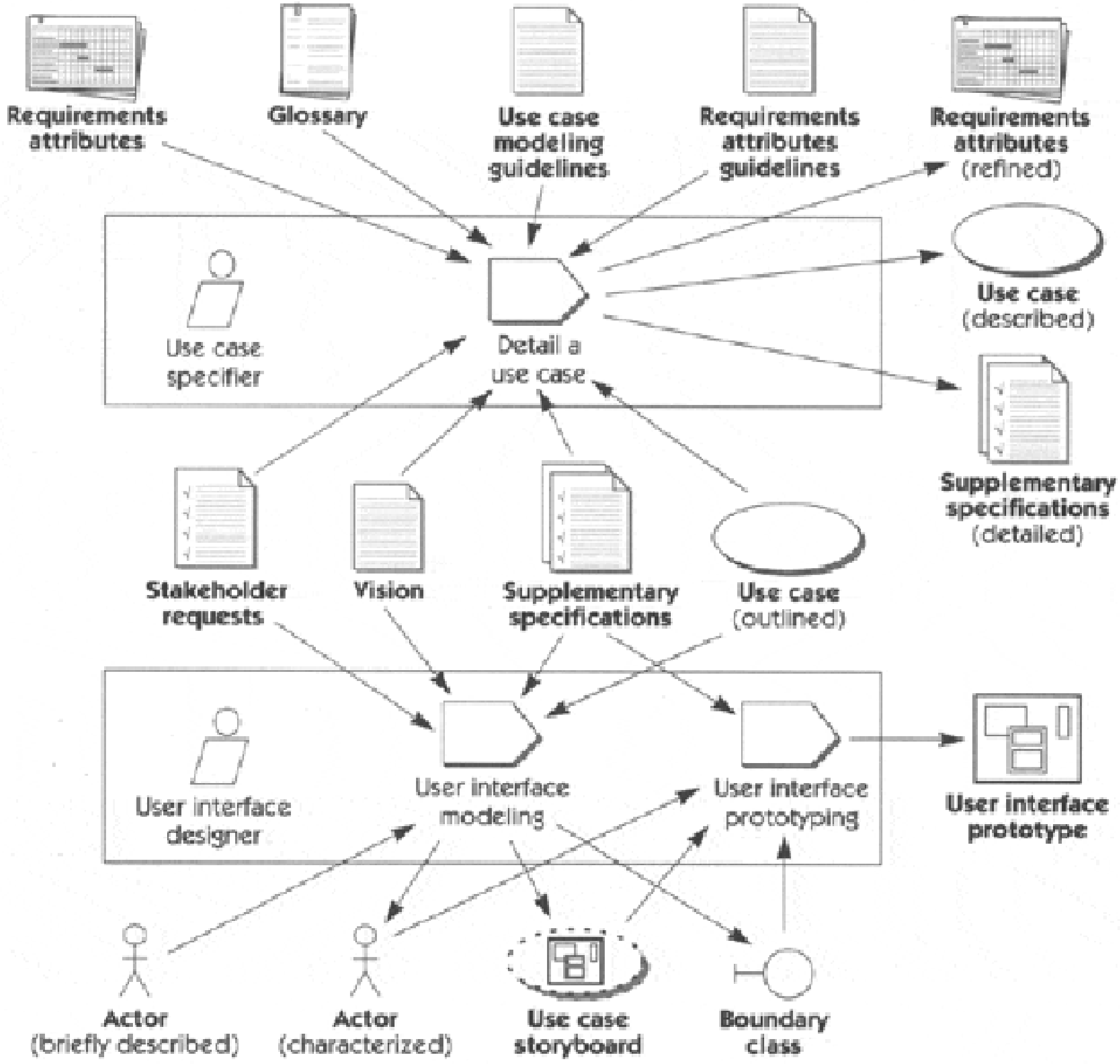
Define the System


Stakeholder requests
Glossary
Use-case modeling guidelines
Vision
Use-case model
Requirements attributes
Requirements attributes guidelines
System analyst
Capture a common vocabulary
Find actors and use cases
Manage dependencies
Glossary (refined)
Supplementary specifications (outlines)
Use-case model (refined)
Requirements attributes (refined)

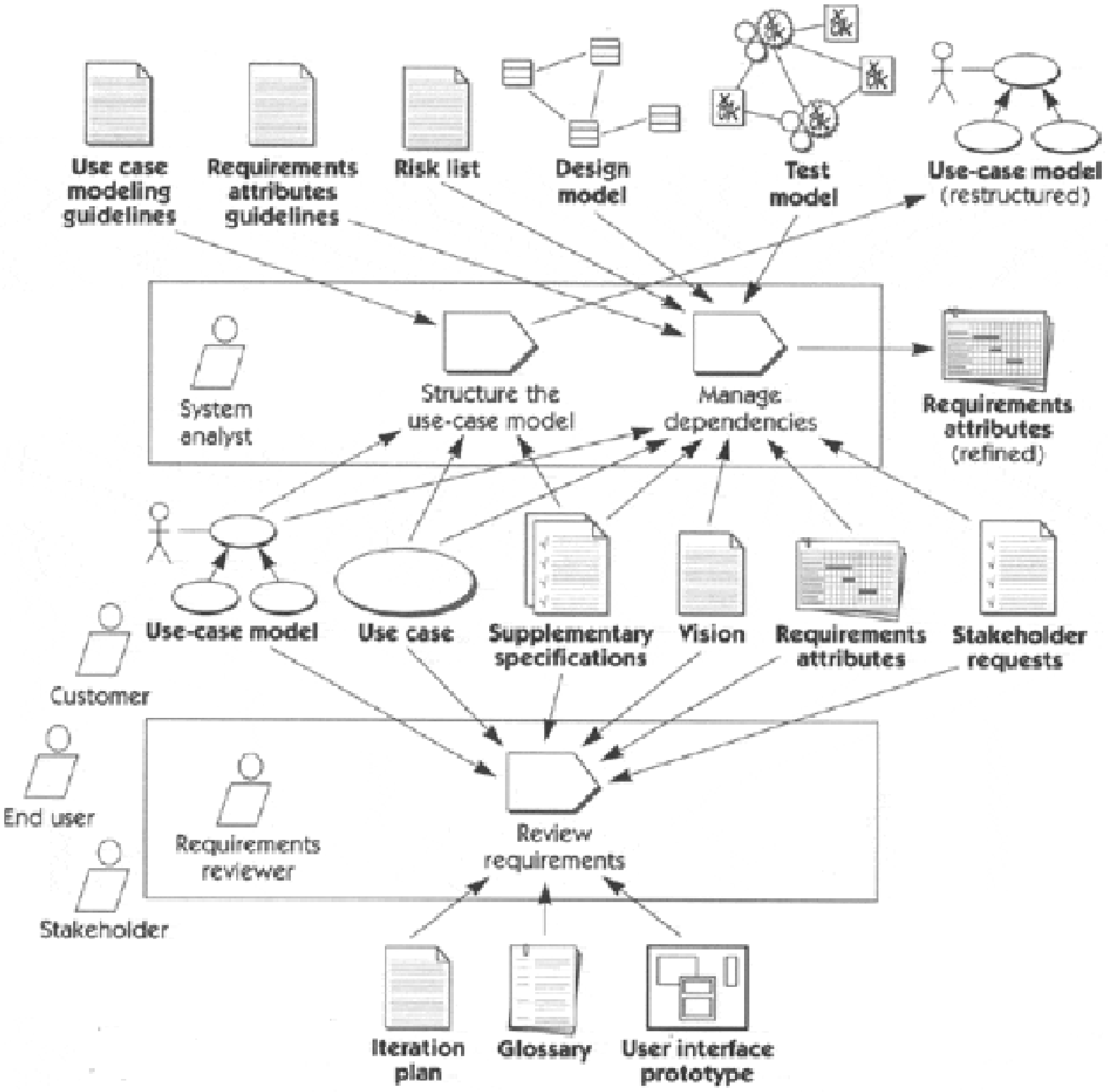
Manage the Scope of the System



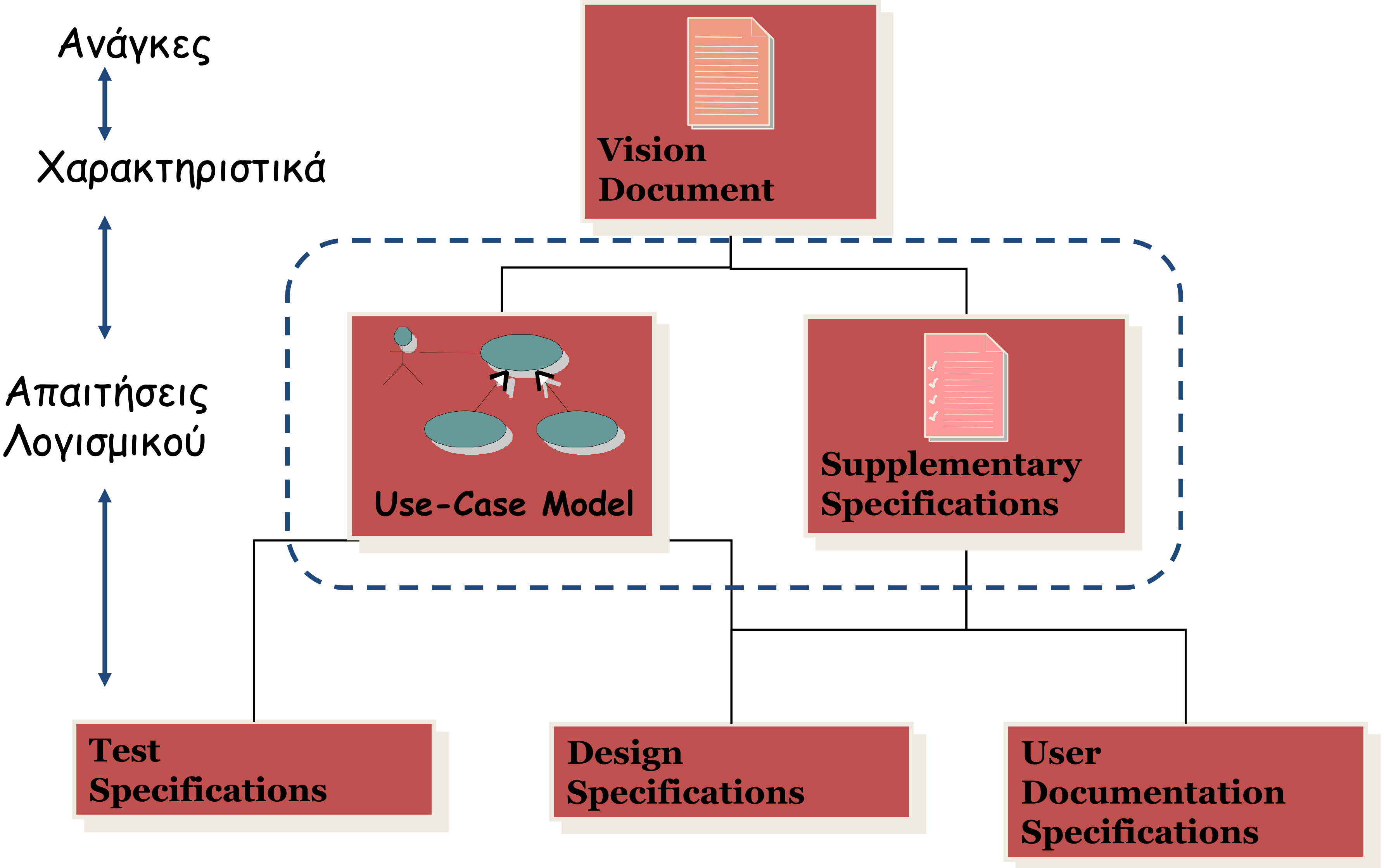
Refine the System Definition



Manage Changing Requirements



Η Δομή Καταγραφής Απαιτήσεων



Οι περιπτώσεις χρήσης

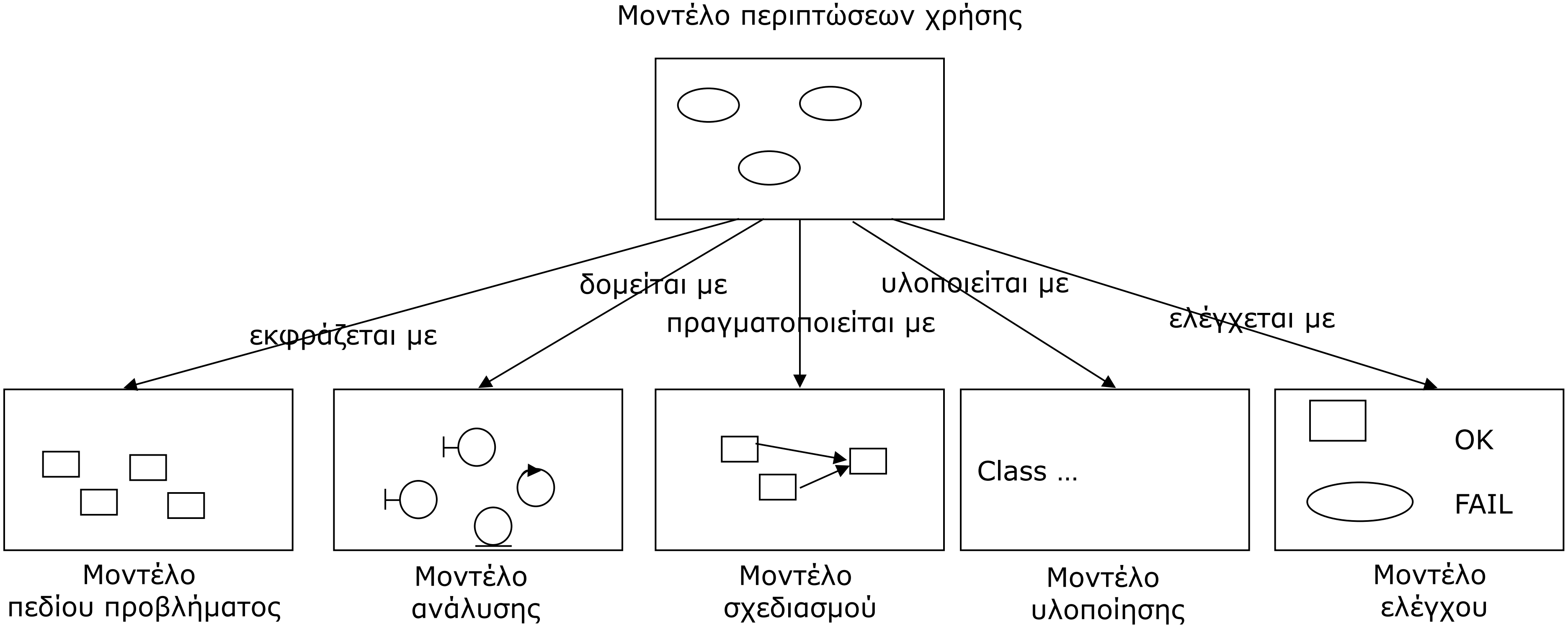
Οι περιπτώσεις χρήσης (use case)

- Προσδιορίζουν την συμπεριφορά του συστήματος. Το τι πρέπει να κάνει το σύστημα.
- Είναι ένα σύνολο διαδοχικών ενεργειών (που μπορεί να περιλαμβάνει εναλλακτικές ενέργειες) που οδηγεί σε ένα χρήσιμο αποτέλεσμα για έναν χειριστή του συστήματος.
- Περιγράφει τις λειτουργικές απαιτήσεις ενός συστήματος
- Το μοντέλο των περιπτώσεων χρήσης περιγράφεται με
  - Τα διαγράμματα περιπτώσεων χρήσης και
  - Τη λεκτική περιγραφή των περιπτώσεων χρήσης

Που χρησιμοποιούνται οι περιπτώσεις χρήσης (use case)

- Οι περιπτώσεις χρήσης χρησιμοποιούνται
  - για την περιγραφή επιχειρηματικών διαδικασιών (business processes)
  - για την περιγραφή των προδιαγραφών (specifications) του συστήματος
  - για τον προσδιορισμό των απαιτήσεων(requirements)
  - για την τεκμηρίωση της λειτουργικότητας του συστήματος
  - ως σημείο αναφοράς για τα άλλα μοντέλα που απαιτούνται για την ανάπτυξη του συστήματος

Το μοντέλο των περιπτώσεων χρήσης



Μοντέλο περιπτώσεων χρήσης

εκφράζεται με

δομείται με

πραγματοποιείται με

υλοποιείται με

ελέγχεται με

Μοντέλο πεδίου προβλήματος

Μοντέλο ανάλυσης

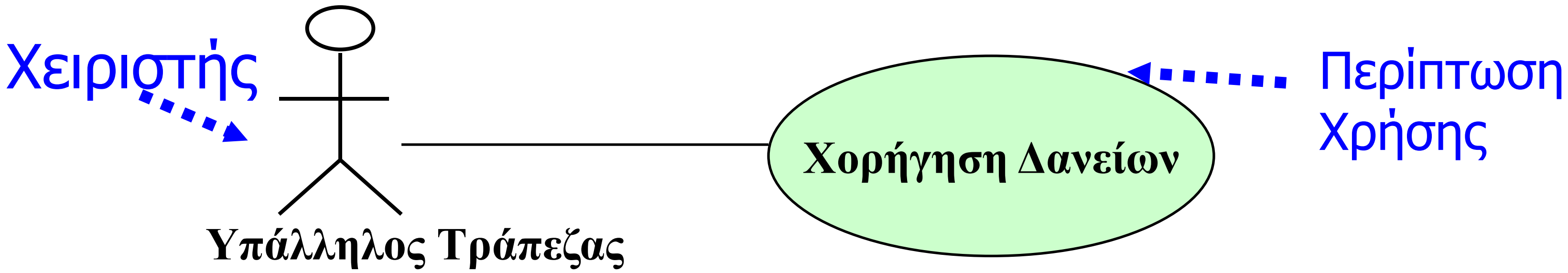
Μοντέλο σχεδιασμού

Μοντέλο υλοποίησης

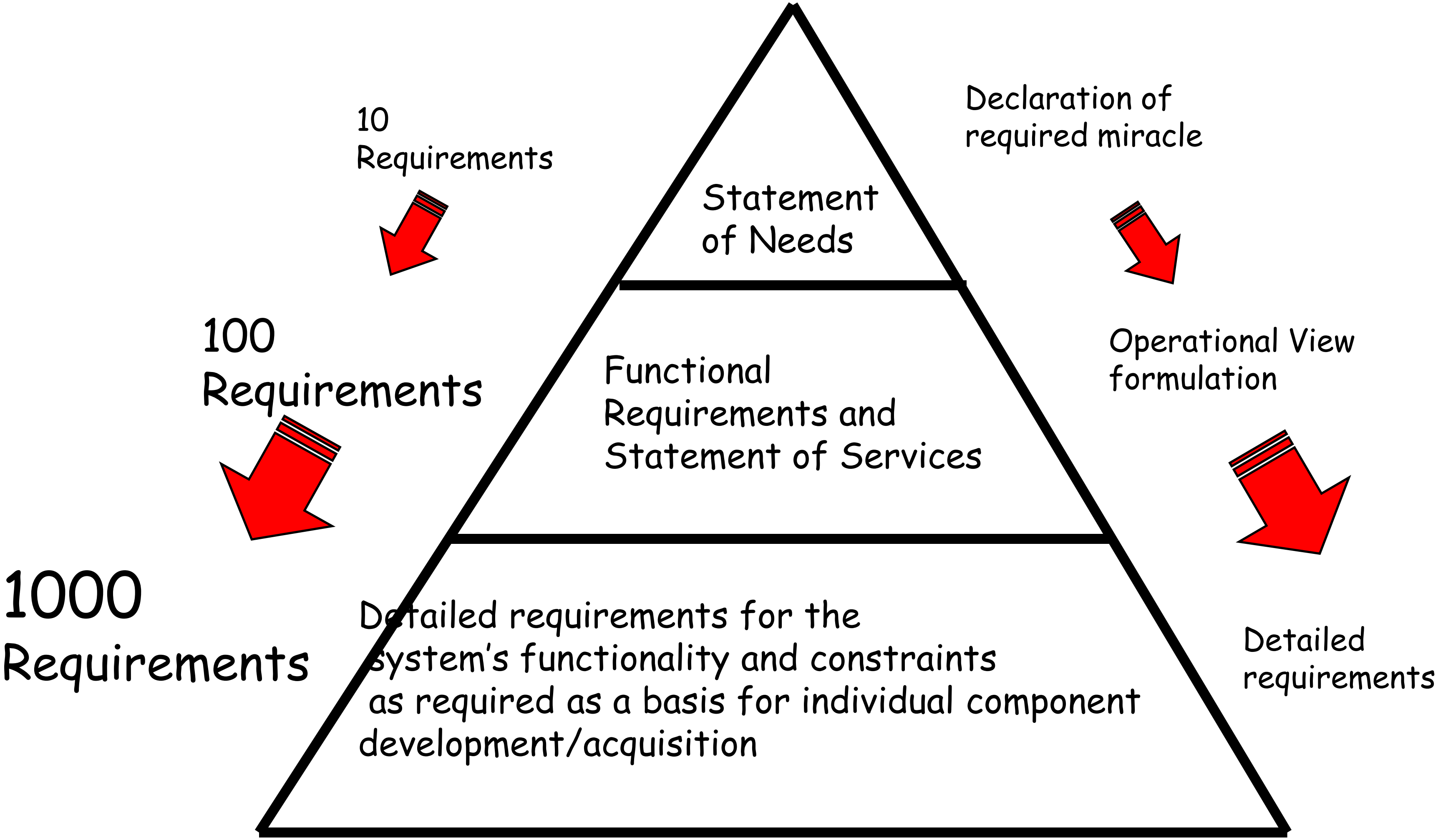
Μοντέλο ελέγχου

Βασικά στοιχειά των διαγραμμάτων περιπτώσεων χρήσης

- Δύο είναι τα βασικά στοιχεία του διαγράμματος περιπτώσεων χρήσης
  - Οι χειριστές και
  - Οι περιπτώσεις χρήσης



Requirements Explosion



10 Requirements

Statement of Needs

Declaration of required miracle

100 Requirements

Functional Requirements and Statement of Services

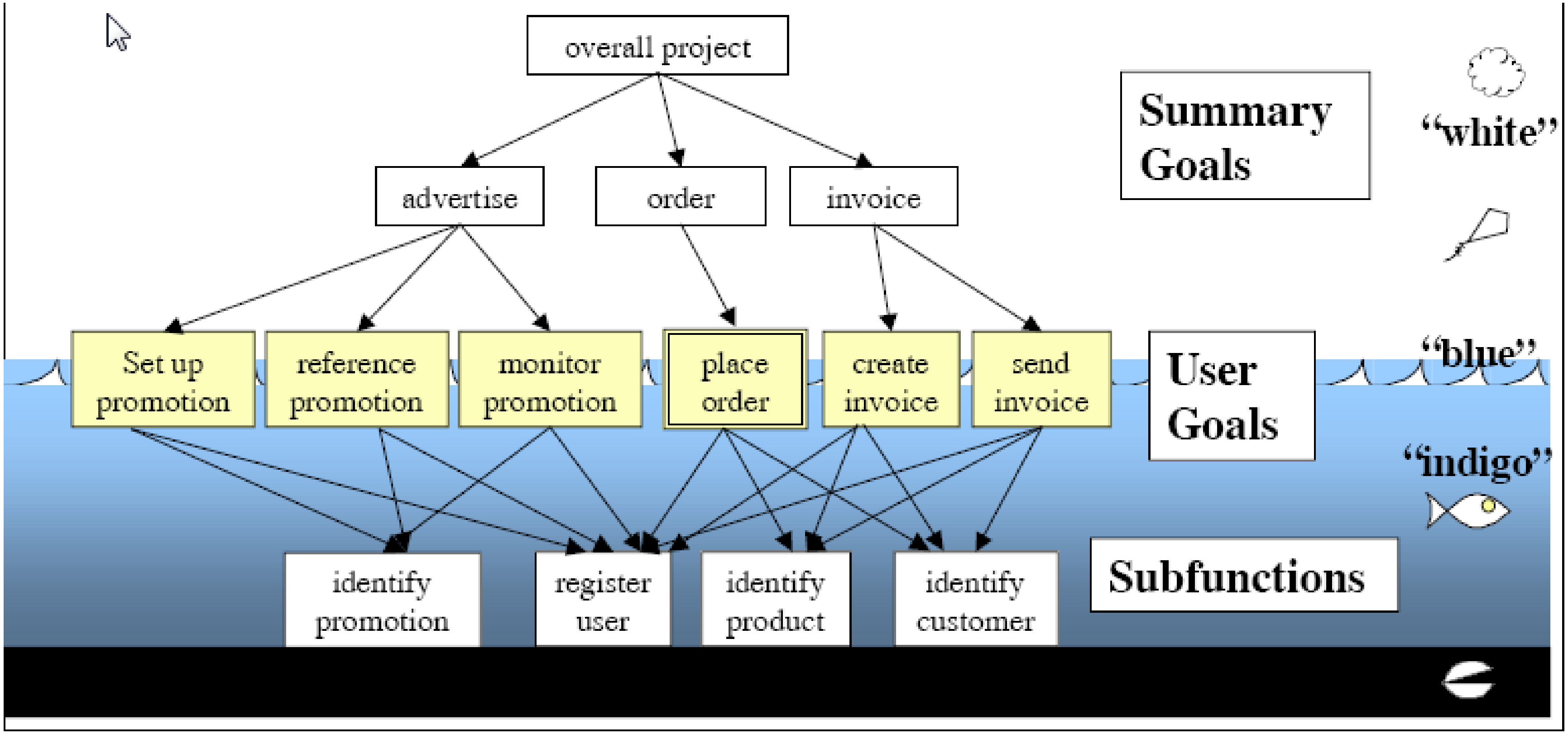
Operational View formulation

1000 Requirements

Detailed requirements for the system's functionality and constraints as required as a basis for individual component development/acquisition

Detailed requirements

Τα επίπεδα της λεπτομέρειας



Από το βιβλίο του A.Cockburn, Writing Effective Use Cases

Work break down structure

WBS Definition

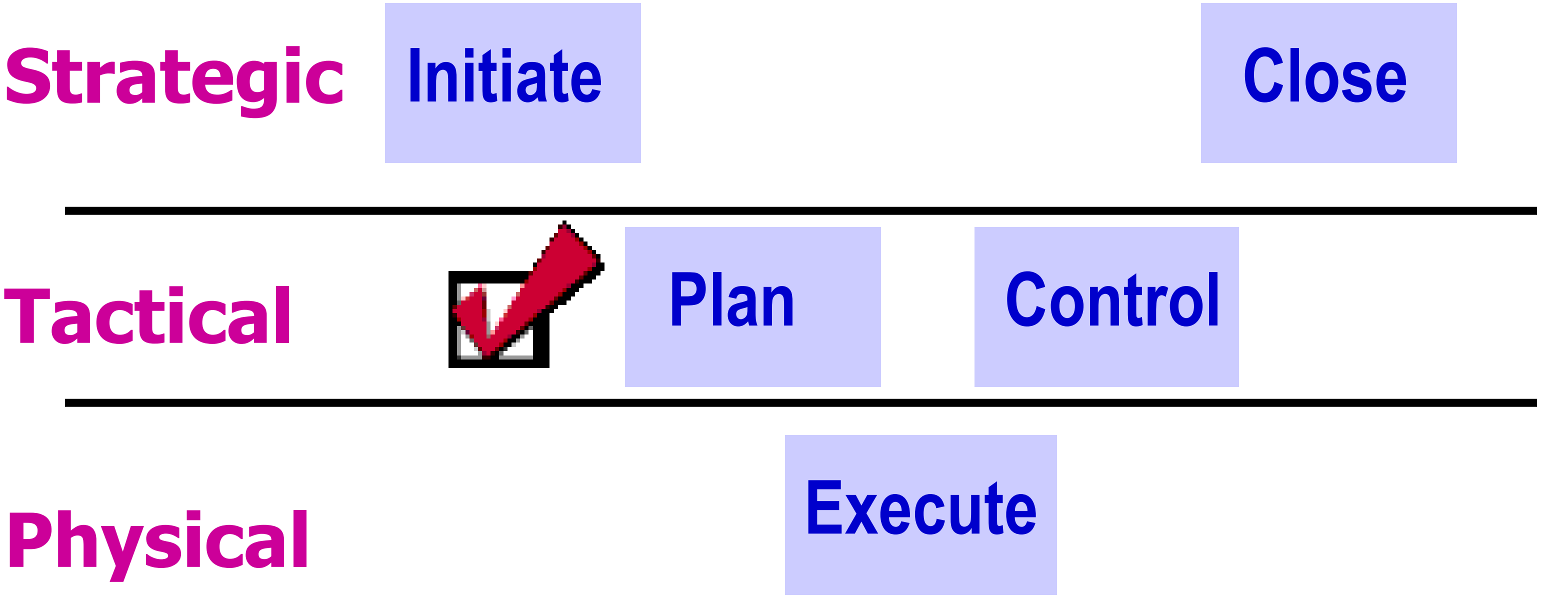
- A deliverable-oriented grouping of project elements that organizes and defines the total scope of the project work.

- Work not in the WBS is not in scope of the project.

- Each descending level represents an increasingly detailed description of the project elements.

- Often used to develop or confirm a common understanding of project scope.

Where the WBS Fits



Strategic

Initiate

Close

Tactical

✓

Plan

Control

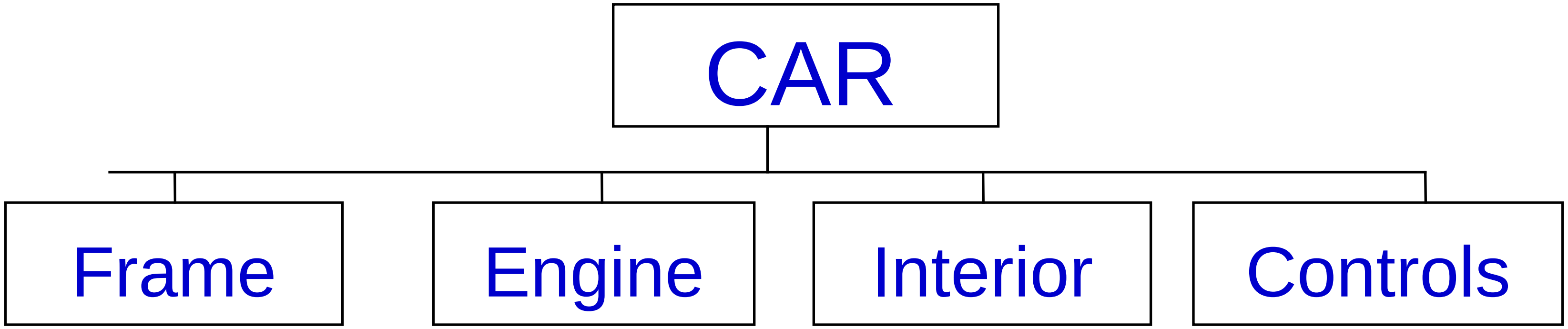
Physical

Execute

Breakdown Structures

WHAT?	HOW?	WHO?
PBS	WBS	OBS
Product Breakdown Structure	Work Breakdown Structure	Organizational Breakdown Structure

Example - PBS/What?



CAR

Frame

Engine

Interior

Controls

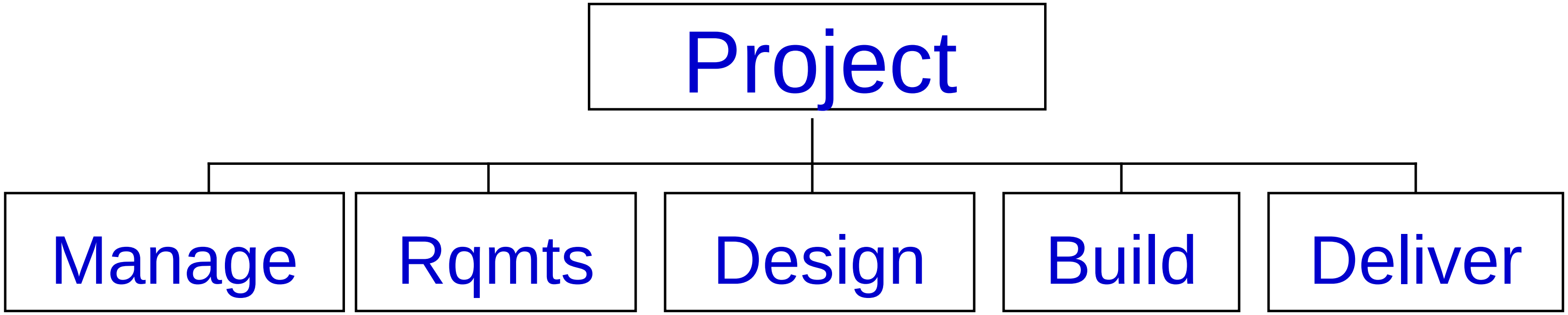
Fenders
Roof
Hoods
Doors
Windows
•
•

Carburetor
Cylinders
Piston
Cam
•
•
•

Front
Seats
Back
Seats
Radio
•

Steering
Fuel
Odometers
Wipers
Lights
•
•

Example - WBS/How?



Project

Manage

Rqmts

Design

Build

Deliver

Plan
Execute
Control
Close
•
•

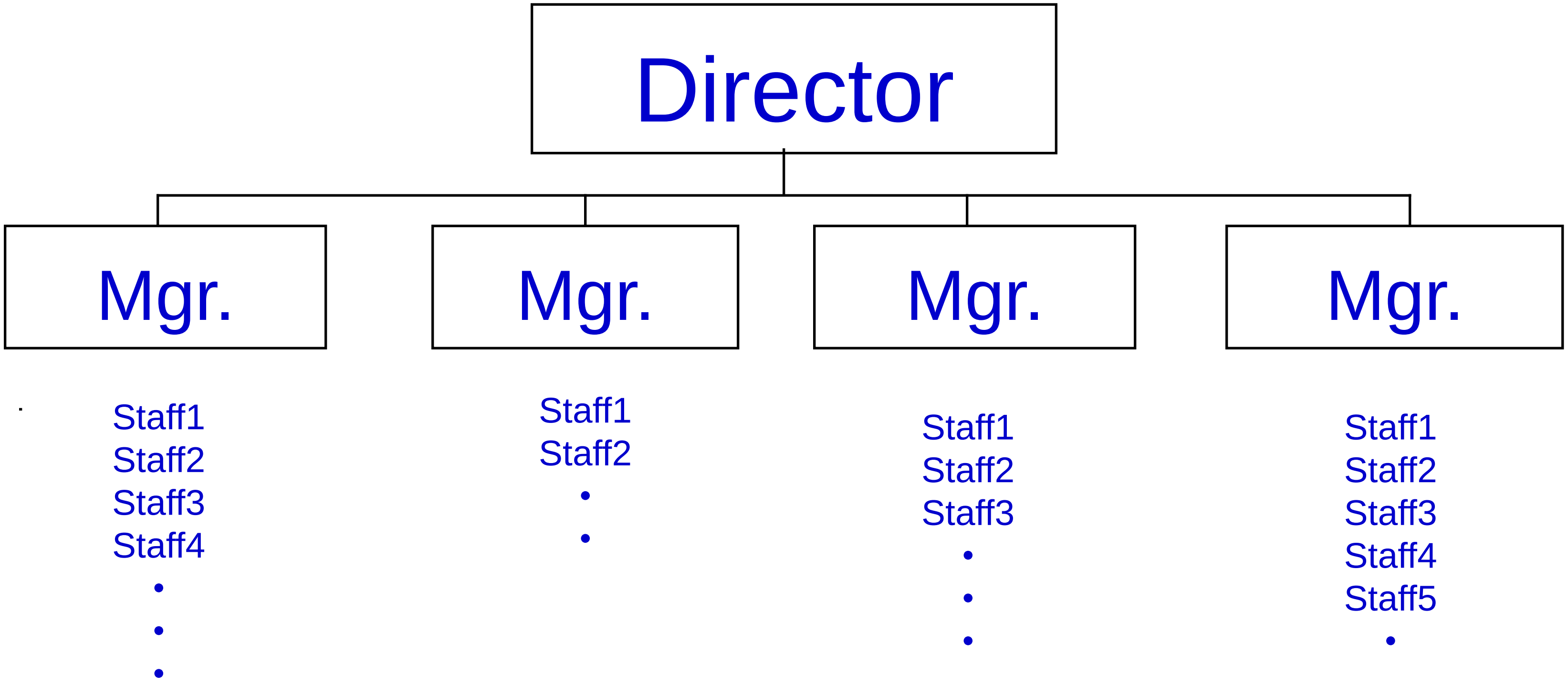
Assess
Determine
Analyze
Propose
•
•

Engineer
Specify
•
•
•

Construct
Test
Validate
Integrate
•
•
•

Install
Train
•
•
•

Example - OBS/Who?



Director

Mgr.

Staff1
Staff2
Staff3
Staff4
•
•
•

Mgr.

Staff1
Staff2
•
•

Mgr.

Staff1
Staff2
Staff3
•
•
•

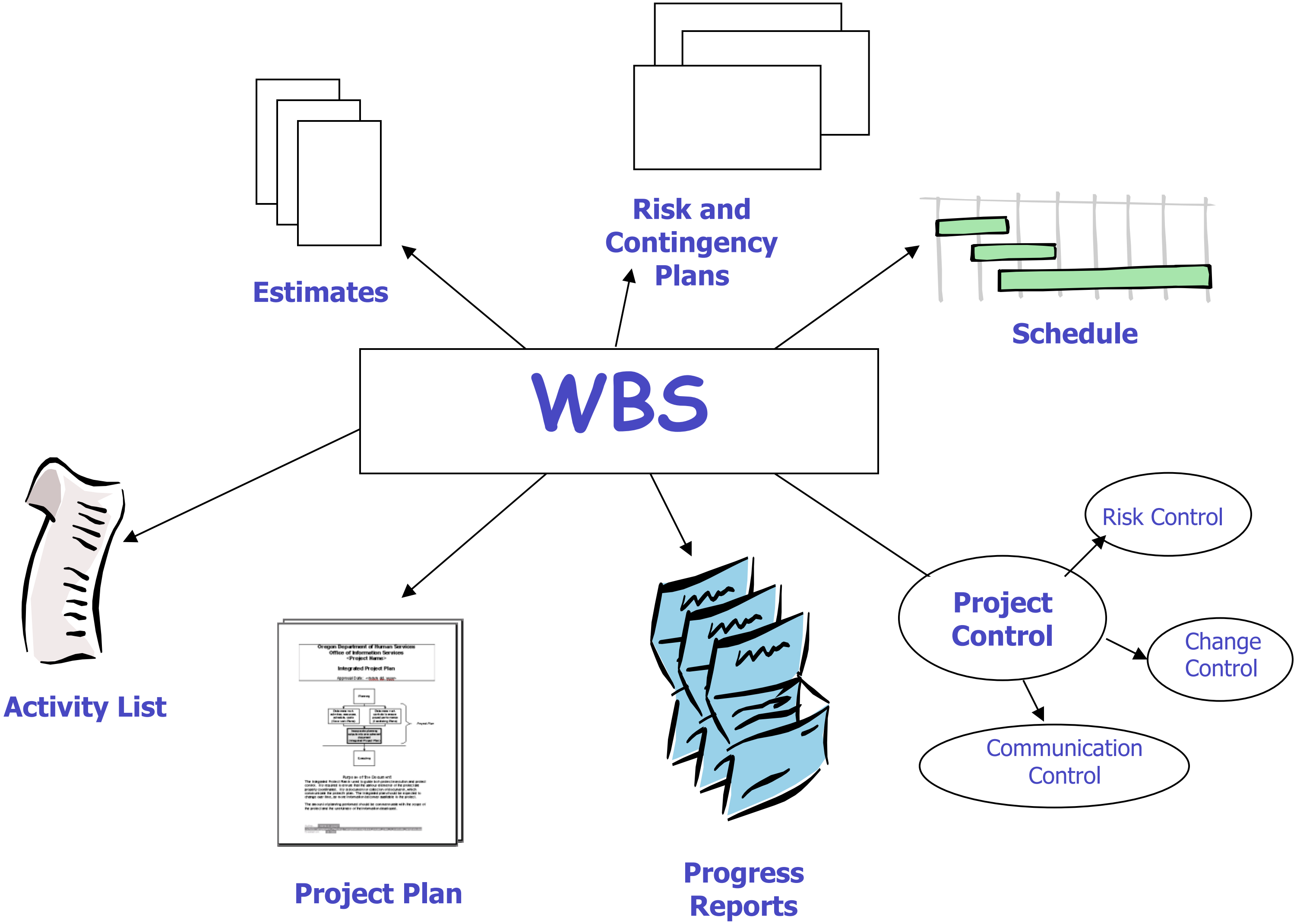
Mgr.

Staff1
Staff2
Staff3
Staff4
Staff5
•

How I should break down the project?

- By geographically separated areas for product or activities
- By major chronological time periods
- By structural, process, system, or device components
- By “intermediate” deliverables required in the production of the “end” deliverables
- By separate areas of responsibility, departments, or functional areas

Benefits of the WBS



WBS

Estimates

Risk and Contingency Plans

Schedule

Activity List

Project Plan

Progress Reports

Project Control

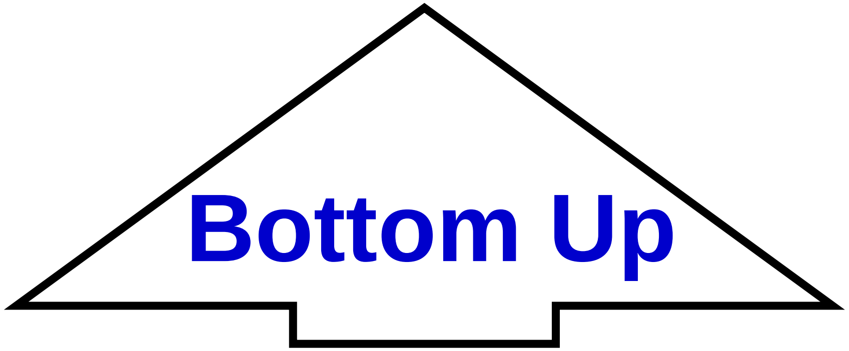
Risk Control

Change Control

Communication Control

Common Approaches

Brainstorming all work to be done and then grouping into a hierarchy.

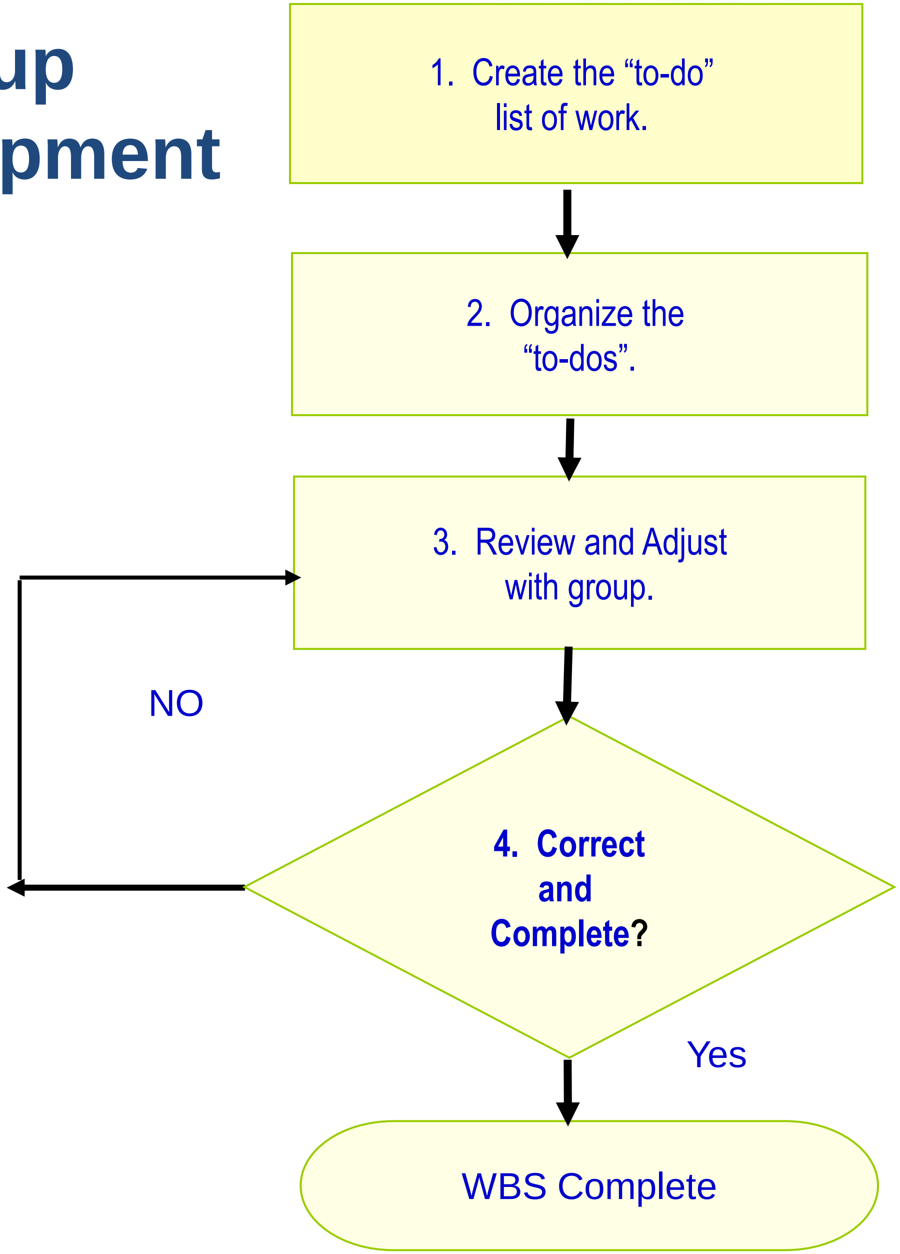


Bottom Up

Top Down

Using a general-to-specific structure to progressively detail the work.

Bottom up WBS Development



1. Create the "to-do" list of work.

2. Organize the "to-dos".

3. Review and Adjust with group.

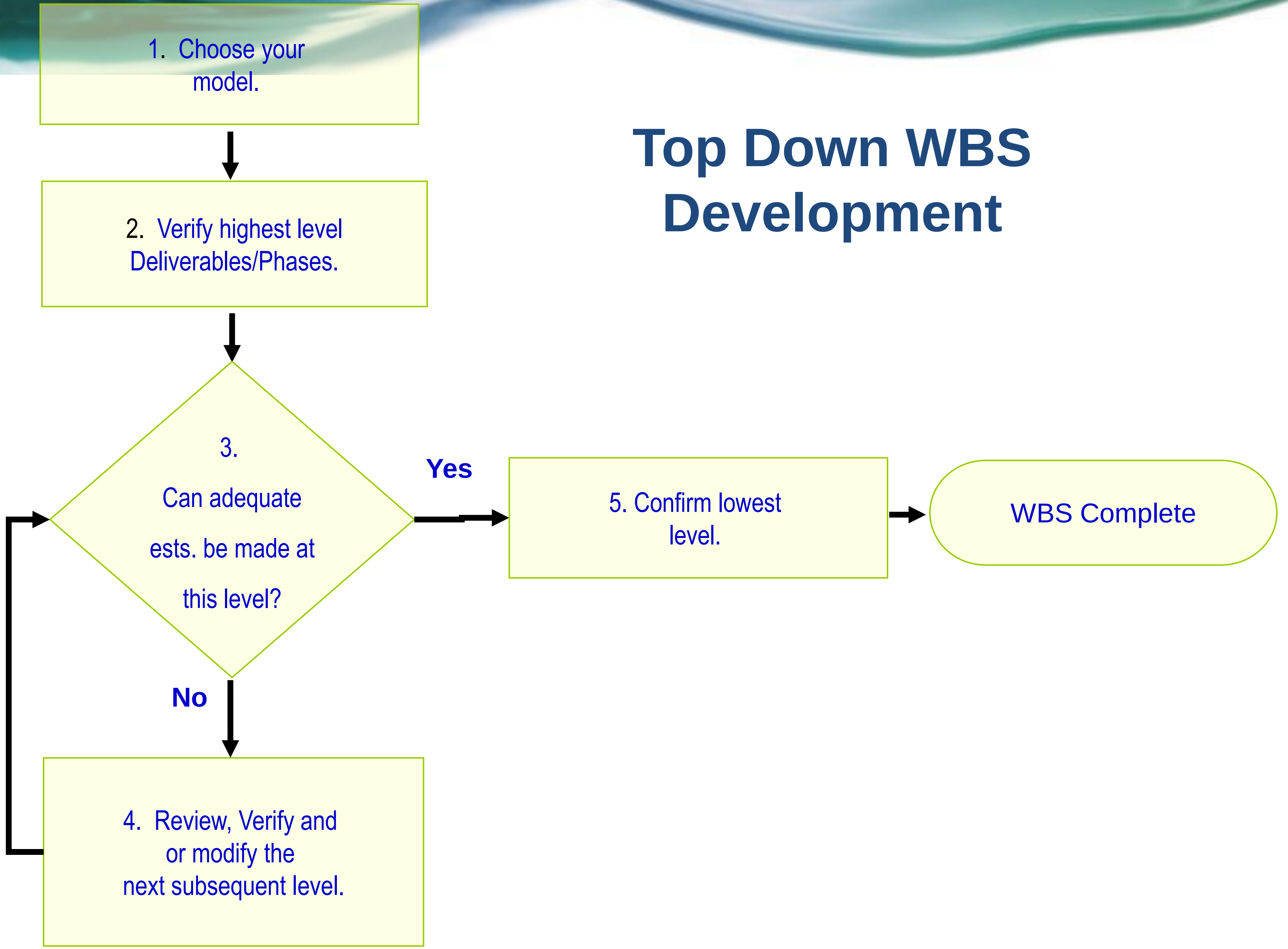
4. Correct and Complete?

NO

Yes

WBS Complete

Top Down WBS Development



1. Choose your model.

2. Verify highest level Deliverables/Phases.

3. Can adequate ests. be made at this level?

Yes

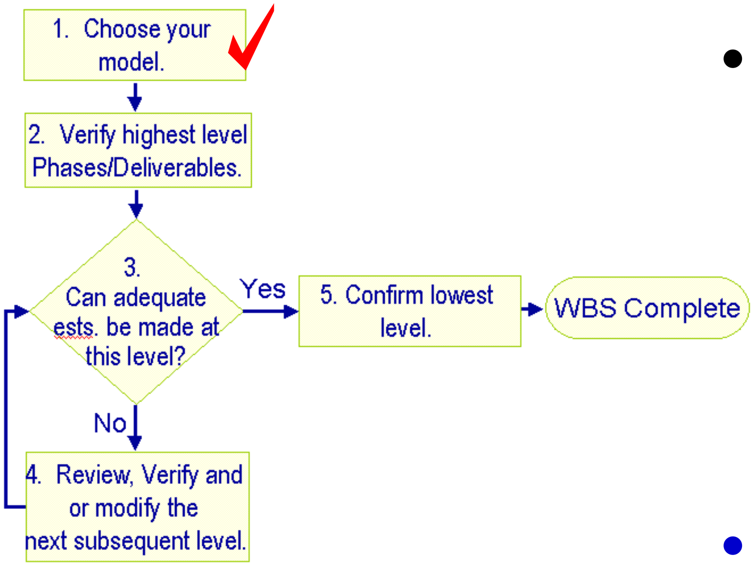
No

4. Review, Verify and or modify the next subsequent level.

5. Confirm lowest level.

WBS Complete

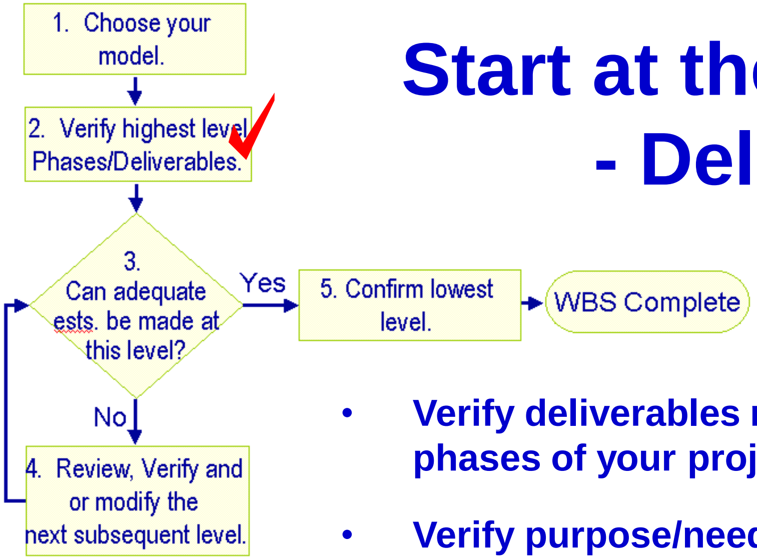
1. Choose your model


1. Choose your model.
✓
2. Verify highest level Phases/Deliverables.
3. Can adequate ests. be made at this level?
Yes
No
5. Confirm lowest level.
WBS Complete
4. Review, Verify and or modify the next subsequent level.

- **Review various:**
  - **life cycle models,**
  - **similar project's WBS, or**
  - **life cycle templates.**
- **Choose a model closest to your specific project.**

Top Down

2. Verify highest level phases/deliverables

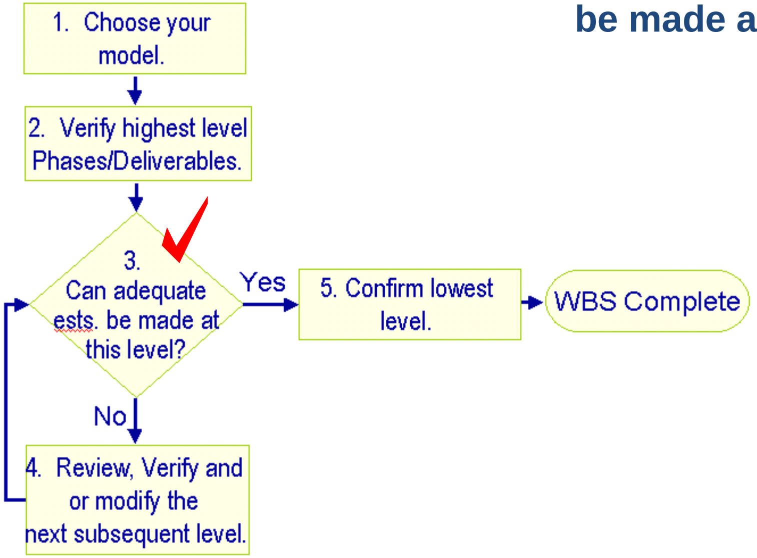

1. Choose your model.
2. Verify highest level Phases/Deliverables. ✓
3. Can adequate ests. be made at this level?
Yes
No
4. Review, Verify and or modify the next subsequent level.
5. Confirm lowest level.
WBS Complete

Start at the top of a model - Deliverables

- Verify deliverables represent the major phases of your project,
- Verify purpose/need of each major deliverable or phase,
- Determine if a previous project completed a major deliverable, e.g. Feasibility.
- Choose to: eliminate or modify deliverable after review of previous completed work

Top Down

3. Can adequate estimates be made at this level?


1. Choose your model.
2. Verify highest level Phases/Deliverables.
3. Can adequate ests. be made at this level?
Yes
No
4. Review, Verify and or modify the next subsequent level.
5. Confirm lowest level.
WBS Complete

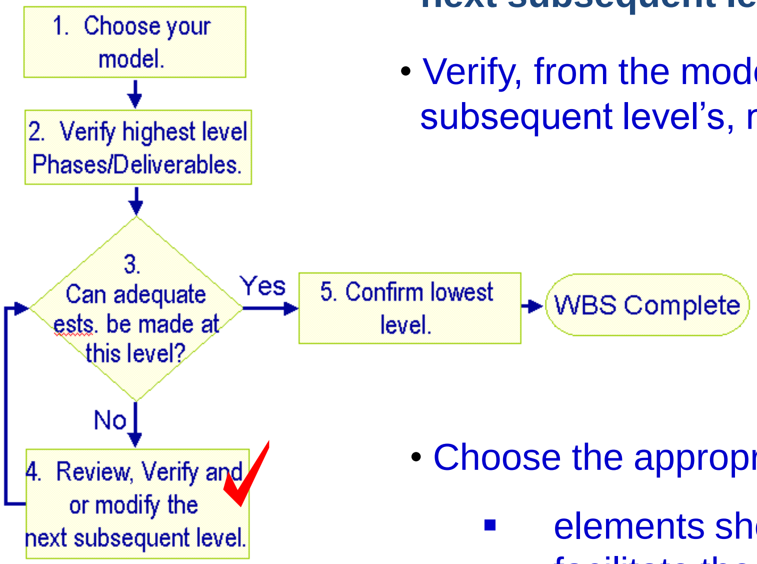
"Yes" Decisions Guidelines

- Can it be completed within a 2 – 3 week period?
- Adequate may change over the course of the project.
- Estimating a major work package that will be produced 6 – 12 months out may not be possible.

Note This step's question means - different levels of decomposition are appropriate for each*

of the major deliverables/phases.

Top Down


1. Choose your model.
2. Verify highest level Phases/Deliverables.
3. Can adequate ests. be made at this level?
Yes
No
5. Confirm lowest level.
WBS Complete
4. Review, Verify and or modify the next subsequent level.

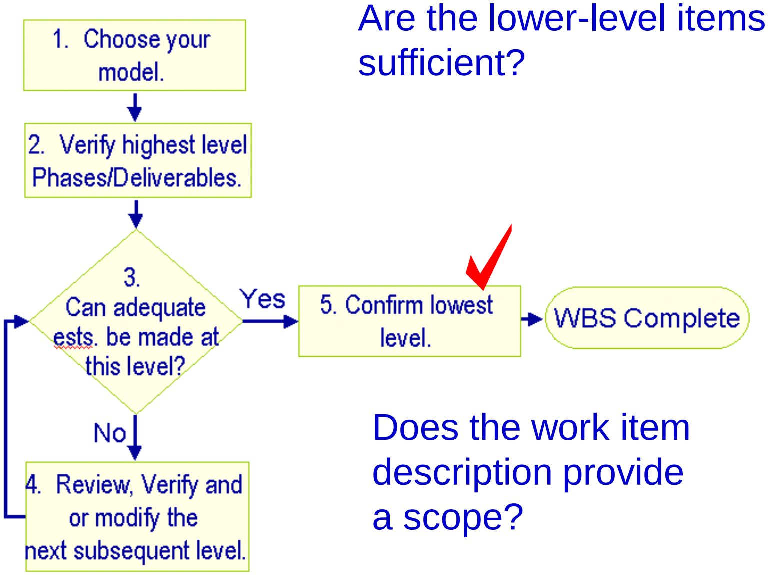
4. Review, verify and or modify the next subsequent level.

- Verify, from the model, the next subsequent level's, more specific work detail.

- Choose the appropriate work elements.
  - elements should be described in tangible, verifiable results in order to facilitate the project progress.
- Repeat step 3 for each work element that you have chosen necessary for the project.

5. Confirm lowest level

Top Down


1. Choose your model.
2. Verify highest level Phases/Deliverables.
3. Can adequate ests. be made at this level?
Yes
No
4. Review, Verify and or modify the next subsequent level.
5. Confirm lowest level.
WBS Complete

Are the lower-level items both necessary and sufficient?

If no, combine items, add to, delete, redefine.

Does the work item description provide a scope?

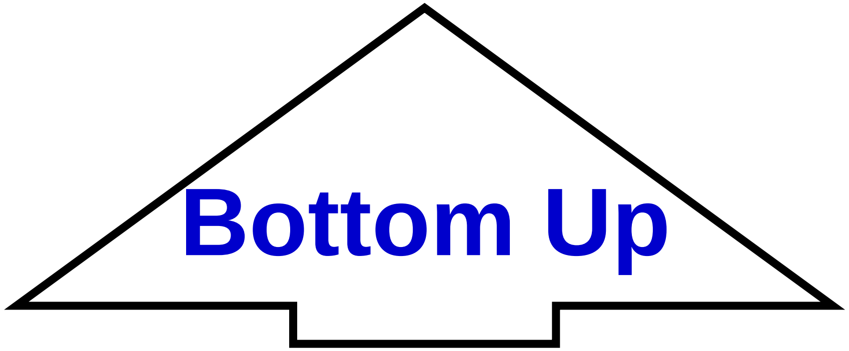
If no, revise or expand the descriptions

Can the item be scheduled? Budgeted? Assigned to a specific organizational unit (e.g., department, team, or person)?

If no, the item must be modified,split, redefined.

Lesson Learned

- Easy to start.
- No terminology issues.
- Higher participation.
- What do we do with this?


Bottom Up

Top Down

- Requires more up front discussion.
- Terminology & structure can get in the way.
- Decreases participation.
- Slower to start.

What’s Next?

Further decomposition into a **WBS Dictionary**

- Briefly describe each item
- Reference by number
- List associated activities
- List milestones
- List other information needed to facilitate work

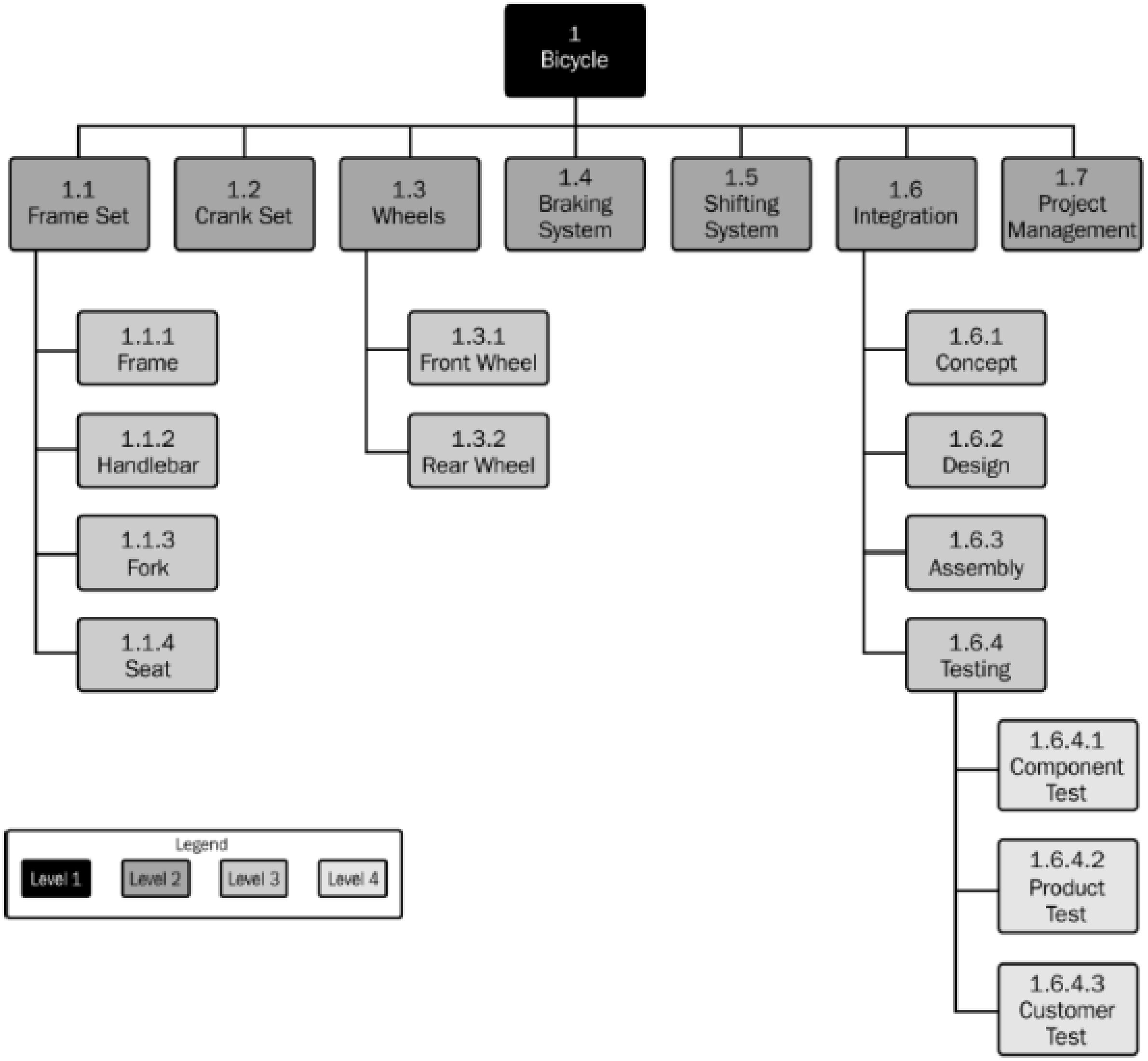
Closing

The greatest benefit of developing a work breakdown structure is providing a common understanding of all the work required to deliver the product or service.

http://www.oregon.gov/DHS/admin/pmo/publications/tools_and_techniques.shtml

WBS Summary

- Defines the hierarchy of deliverables
- Supports the definition of all work required to implement deliverables
- Graphical representation of project scope
- Framework for all deliverables
- Framework for schedule and cost calculations
- Facilitates assignment of resources
- Facilitates reporting
- Provides a framework for project evaluation



1 Bicycle

1.1 Frame Set

1.1.1 Frame

1.1.2 Handlebar

1.1.3 Fork

1.1.4 Seat

1.2 Crank Set

1.3 Wheels

1.3.1 Front Wheel

1.3.2 Rear Wheel

1.4 Braking System

1.5 Shifting System

1.6 Integration

1.6.1 Concept

1.6.2 Design

1.6.3 Assembly

1.6.4 Testing

1.6.4.1 Component Test

1.6.4.2 Product Test

1.6.4.3 Customer Test

1.7 Project Management

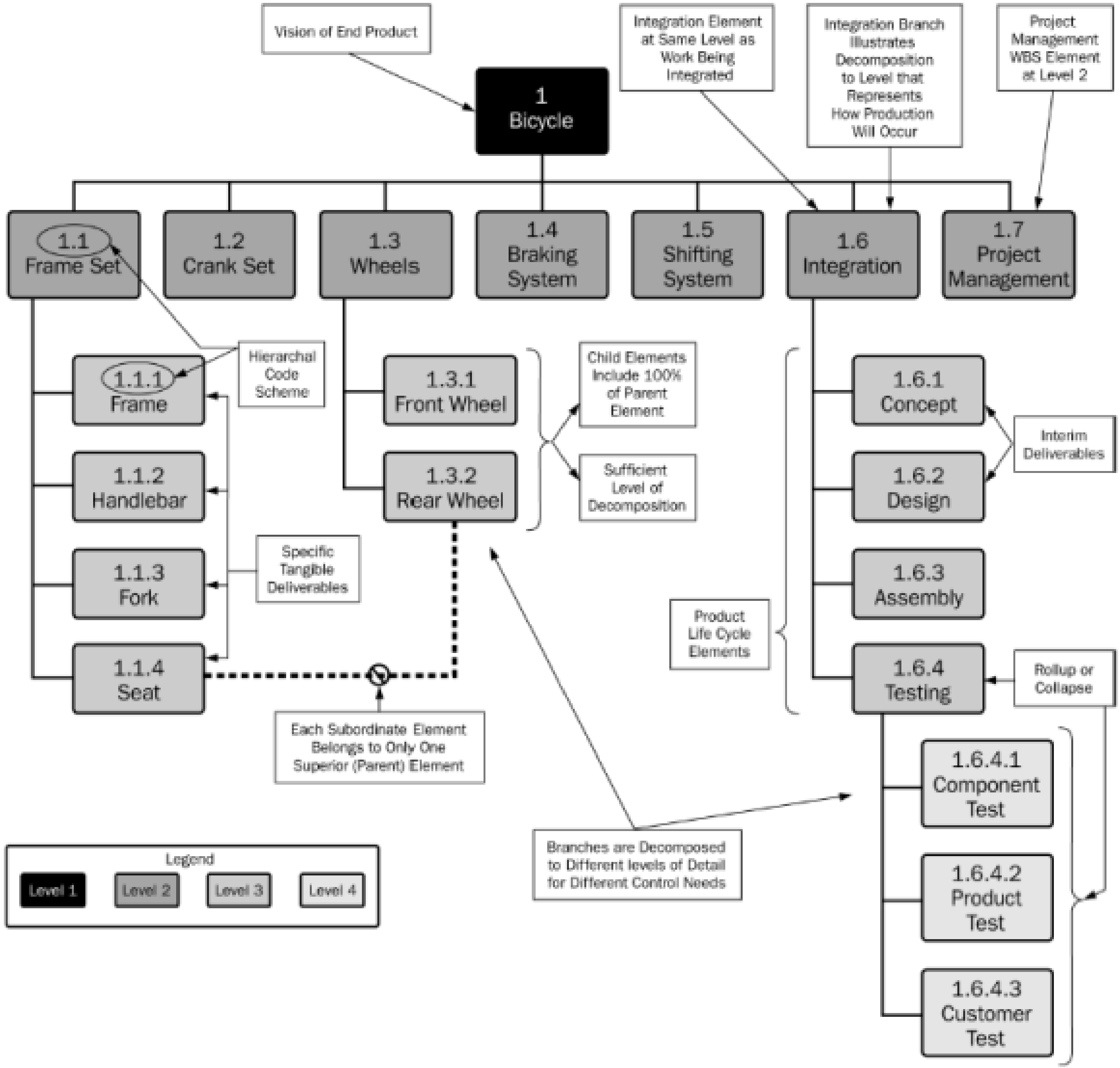
Legend

Level 1

Level 2

Level 3

Level 4



Vision of End Product

1 Bicycle

Integration Element at Same Level as Work Being Integrated

Integration Branch Illustrates Decomposition to Level that Represents How Production Will Occur

Project Management WBS Element at Level 2

1.1 Frame Set

1.2 Crank Set

1.3 Wheels

1.4 Braking System

1.5 Shifting System

1.6 Integration

1.7 Project Management

Hierarchal Code Scheme

1.1.1 Frame

1.1.2 Handlebar

1.1.3 Fork

1.1.4 Seat

Specific Tangible Deliverables

1.3.1 Front Wheel

1.3.2 Rear Wheel

Child Elements Include 100% of Parent Element

Sufficient Level of Decomposition

Each Subordinate Element Belongs to Only One Superior (Parent) Element

Branches are Decomposed to Different levels of Detail for Different Control Needs

1.6.1 Concept

1.6.2 Design

1.6.3 Assembly

1.6.4 Testing

Interim Deliverables

Product Life Cycle Elements

Rollup or Collapse

1.6.4.1 Component Test

1.6.4.2 Product Test

1.6.4.3 Customer Test

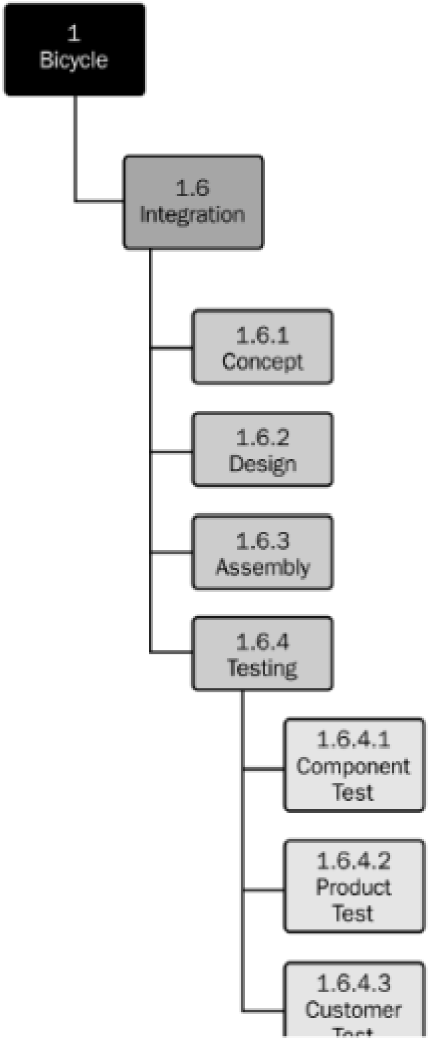
Legend

Level 1

Level 2

Level 3

Level 4



1
Bicycle

1.6
Integration

1.6.1
Concept

1.6.2
Design

1.6.3
Assembly

1.6.4
Testing

1.6.4.1
Component
Test

1.6.4.2
Product
Test

1.6.4.3
Customer
Test

1 Bicycle

1.6 Integration

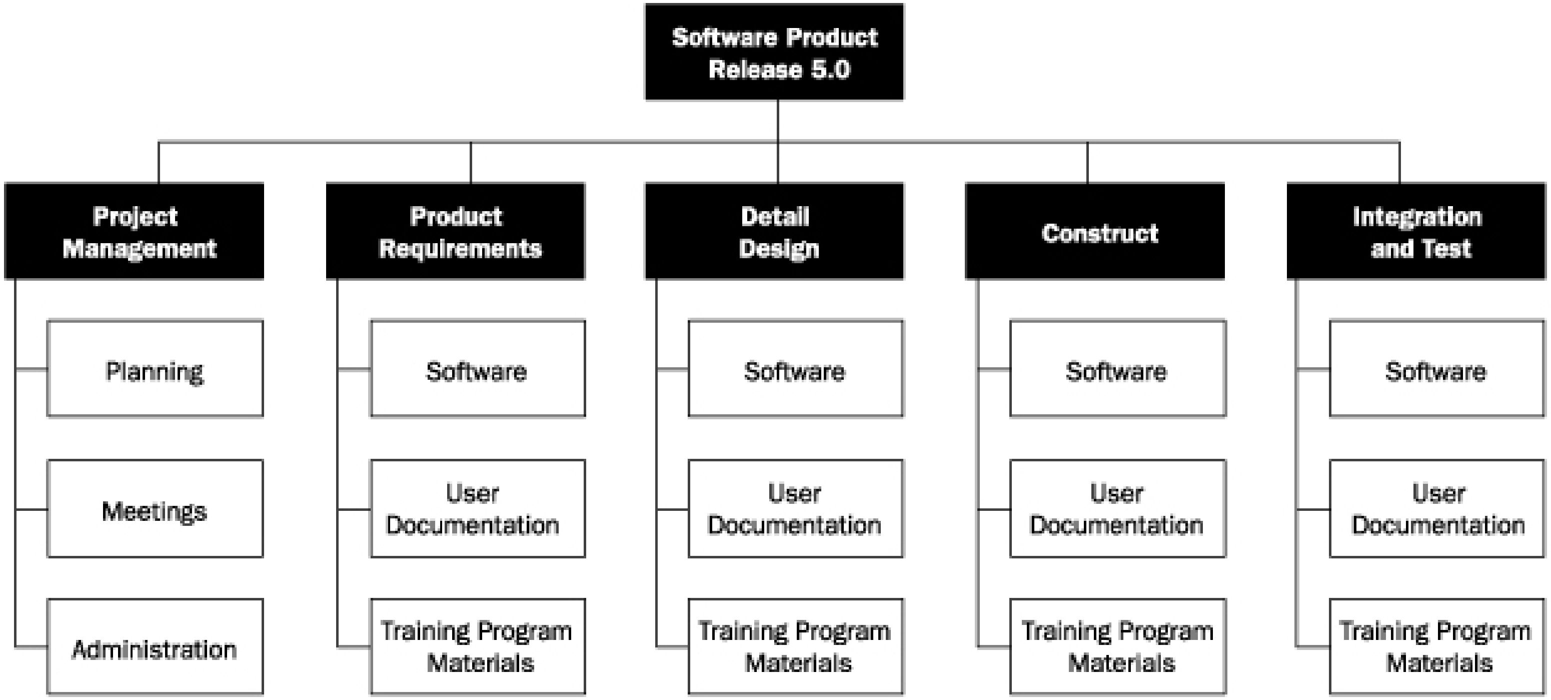
- 1.6.1 Concept
- 1.6.2 Design
- 1.6.3 Assembly
- 1.6.4 Testing
    - 1.6.4.1 Component Test
    - 1.6.4.2 Product Test
    - 1.6.4.3 Customer Test

An exercise

- Create a WBS for creating “brownies”

Brownies WBS

1. Prepreparation
    - 1.1 Read recipe
    - 1.2 Check ingredients at home
    - 1.3 Make shopping list
    - 1.4 Go on shopping trip
2. Preparation
    - 2.1 Gather tools
    - 2.2 Gather ingredients
    - 2.3 Mix ingredients
        - 2.3.1 Melt butter and chocolate in microwave
        - 2.3.2 Mix other ingredients in bowl
    - 2.4 Grease pan
    - 2.5 Preheat oven
    - 2.6 Pour batter in pan
3. Baking Process
    - 3.1 Bake in oven
    - 3.2 Test with toothpick
    - 3.3 Remove from oven
    - 3.4 Turn off oven
4. Cleanup Process
    - 4.1 Wash dishes
    - 4.2 Cool brownies



Software Product Release 5.0

Project Management
- Planning
- Meetings
- Administration

Product Requirements
- Software
- User Documentation
- Training Program Materials

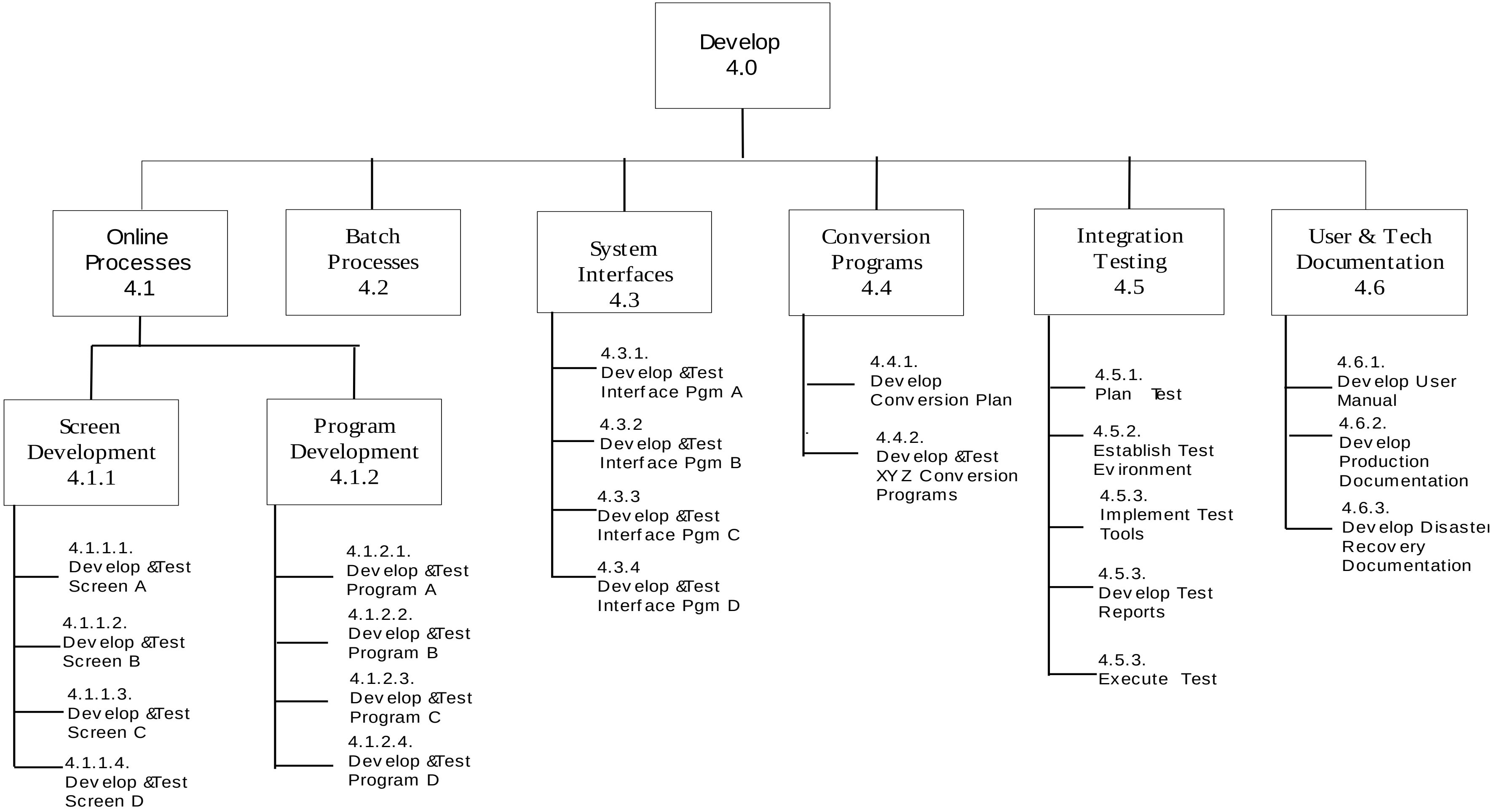
Detail Design
- Software
- User Documentation
- Training Program Materials

Construct
- Software
- User Documentation
- Training Program Materials

Integration and Test
- Software
- User Documentation
- Training Program Materials

The WBS is illustrative only. It is not intended to represent the full project scope of any specific project, nor to imply that this is the only way to organize a WBS on this type of project.

DETAILED WBS EXAMPLE
CUSTOM SYSTEM DEVELOPMENT



Develop
4.0

- Online Processes 4.1
  - Screen Development 4.1.1
    - 4.1.1.1. Develop &Test Screen A
    - 4.1.1.2. Develop &Test Screen B
    - 4.1.1.3. Develop &Test Screen C
    - 4.1.1.4. Develop &Test Screen D
  - Program Development 4.1.2
    - 4.1.2.1. Develop &Test Program A
    - 4.1.2.2. Develop &Test Program B
    - 4.1.2.3. Develop &Test Program C
    - 4.1.2.4. Develop &Test Program D
- Batch Processes 4.2
- System Interfaces 4.3
  - 4.3.1. Develop &Test Interface Pgm A
  - 4.3.2 Develop &Test Interface Pgm B
  - 4.3.3 Develop &Test Interface Pgm C
  - 4.3.4 Develop &Test Interface Pgm D
- Conversion Programs 4.4
  - 4.4.1. Develop Conversion Plan
  - 4.4.2. Develop &Test XYZ Conversion Programs
- Integration Testing 4.5
  - 4.5.1. Plan Test
  - 4.5.2. Establish Test Evironment
  - 4.5.3. Implement Test Tools
  - 4.5.3. Develop Test Reports
  - 4.5.3. Execute Test
- User & Tech Documentation 4.6
  - 4.6.1. Develop User Manual
  - 4.6.2. Develop Production Documentation
  - 4.6.3. Develop Disaste Recovery Documentation

Other tools

- Scope Matrix – Requirements inventory
- Deliverable Tracking Matrix