

# Web Engineering Requirement Engineering for Web Applications

Lecture II – October 5, 2009

Katharina Siorpaes

## Overview

- Introduction to requirements engineering
  - Principles
  - Adapting traditional requirements engineering to Web applications
  - Specifics in Web Engineering
- Elicitation & Negotiation
- Specification
- Validation and Management
- Example

- *Requirements Engineering (RE)* – the principles, methods, & tools for eliciting, describing, validating, and managing project goals and needs.
- Given the complexity of Web apps, RE is a critical initial stage, but often poorly executed.
- What are the consequences?
  - Inadequate software architectures
  - “Unforeseen” problems
    - Budget overruns
    - Production delays
    - “That’s not what I asked for”
  - Low user acceptance

- „Ermittlung der Anforderungen sowie die Verwaltung der Anforderungen im Entwicklungsprozess werden von 80% der Firmen als Probleme im Entwicklungsprozess genannt (*ESSI ESPITI Survey*)“
- „30% aller Projekte scheitern vorzeitig, 70% der verbleibenden Projekte erfüllen nicht die Kundenerwartungen. Problemursachen stehen in mehr als der Hälfte der Fälle in engem Zusammenhang mit den Anforderungen (*StandishGroup CHAOS Report*)“
- „Nur 16% der Web-Anwendungen decken die Bedürfnisse der Auftraggeber voll ab, 53% der ausgelieferten Systeme haben nicht den geforderten Funktionsumfang (*Cutter Consortium*)“

## What is a Requirement?



- A requirement describes a property to be met or a service to be provided by a system.
- IEEE 601.12 definition of *requirement*:
  1. Condition needed to solve a user's problem
  2. Condition to be met or possessed by the system to satisfy a formal agreement
  3. Documented representation of conditions as in 1 and 2
- Stakeholders: persons or organisations that are involved in the Web application and have direct influence on the requirements.

## Tasks of requirements



- Identify and negotiate requirements
- Description of requirements
- Validation of requirements
- Management of requirements

## Stakeholder



- Customer
- User
- Developer
- For Web apps extremely relevant:
  - Content providers (responsibles)
  - Domain experts
  - Usability experts
  - Responsibles for market and target group analysis
- Goals:
  - Requirements on a higher level of abstraction.
  - Means to define a **shared** vision.

## Examples stakeholder goals



- *Web app must be available on Sep. 1, 2010. (Customer)*
- *Web app must be able to handle 2500 concurrent users. (Customer, quality related)*
- *J2EE should be used as a development platform. (Developer)*
- *Data transactions must be secured. (User, quality-related)*
- *The user interface must allow having different layouts for different groups of customers. (Customer)*
- *An arbitrary user must be able to find the desired product within 3 minutes.*
- *Etc.*

## Why do we need Requirements?



- Bell & Thayer (1976) – Requirements don't define themselves.
- Boehm (1981) – Removal of mistakes post hoc is up to *200 times* more costly.
- The Standish Group (1994) – 30% of projects fail before completion & almost half do not meet customer requirements
  - Unclear objectives, unrealistic schedules & expectations, poor user participation

## Requirement elicitation



- Requirements not by simple questionnaires
- Rs are a result of a joint learn and consensus finding process
- Various methods:
  - Creativity techniques
  - Scenario based
  - Multi-criteria decision processes
  - Moderation techniques
  - Interviews
  - Document analysis

## Requirement description



- Requirement analysis document
- Various forms
  - Informal (e.g. Users stories aus extreme programming)
  - Semi-formal (e.g. Use cases)
  - Formal
- Decision depends on
  - Project risk
  - Stakeholders

## Validating requirements



- Validation (Did we specify the right thing?)
- Verification (Did we specify correctly?)
- Methods
  - Reviews
  - Inspections
  - Prototyping

## Requirements management



- Changes are natural.
- Requirements are not static but change.
- Requirements management includes
  - Adding new requirements
  - Changing
  - Management of inter-dependencies

## Good Requirements Specifications I



- Correct
  - Correspond to actual need
- Unambiguous
  - Can be interpreted only in one way
- Complete
  - Any external imposed requirement should be included
- Consistent
  - Conflicting requirements should be avoided

## Good Requirements Specifications II



- Ranked for importance and/or stability
  - Requirements are not equally important
  - Requirements are not equally stable
- Verifiable
  - It's possible to use a cost-effective process to check it
- Modifiable
  - Can be restructured quickly
  - Adopt cross reference
  - Requirements are clearly separated
- Traceable
  - Can be tracked from originating design documentation

## Types of Requirements



- Many taxonomies exist to describe requirements, but most divide them into two groups:
  - Functional – describes the capability's purpose
    - e.g., the ability to transfer money between user accounts
  - Non-functional – describes the capability's properties
    - e.g., the Home Page must load within 5 seconds on a dial-up connection



## Types of requirements



- Functionality
- Content
- Quality
- System environment
- User interface
- Evolution requirements
- Project management

## Functional Requirement Types



- Data Requirements
  - How information is stored and managed
- Interface Requirements
  - How the user is going to interact with the application
- Navigational Requirements
  - How the user is going to navigate through the application
- Personalization Requirements
  - How the application adapt itself according to user or environment profile
- Transactional Requirements
  - How the application behave internally

## Non-Functional Requirement Types



- Content
- Quality
  - Functionality, Usability, Portability, Scalability
  - Reliability, Efficiency, Security, Maintainability
- System Environment
- User Interface
  - Self-explanatory & intuitive
  - Usage-centered design
- Evolution
- Project Constraints

## Principles for RE I



- Understanding the system context
  - Web apps are always a component of a larger entity
  - Why do we need the system?
  - How will people use it?
- Involving the stakeholders
  - Get all groups involved.
  - Balance – one group's gain should not come at the expense of another.
  - Repeat the process of identifying, understanding and negotiating.

## Principles for RE II



- Iteratively define requirements
  - Requirements need to be consistent with other system aspects (UI, content, test cases)
  - Start with key requirements at a high level; these will serve as the basis for:
    - Feasible architectures
    - Key system use cases
    - Initial plans for the project
  - As the project progresses, requirements can become more concrete.

## Principles for RE III



- Focusing on the System Architecture
  - The “solution space” – existing technologies & legacy systems – defines the “problem space.”
  - The architecture must be considered in the elicitation stage.
  - Refine requirements and architecture iteratively with increasing level of detail.

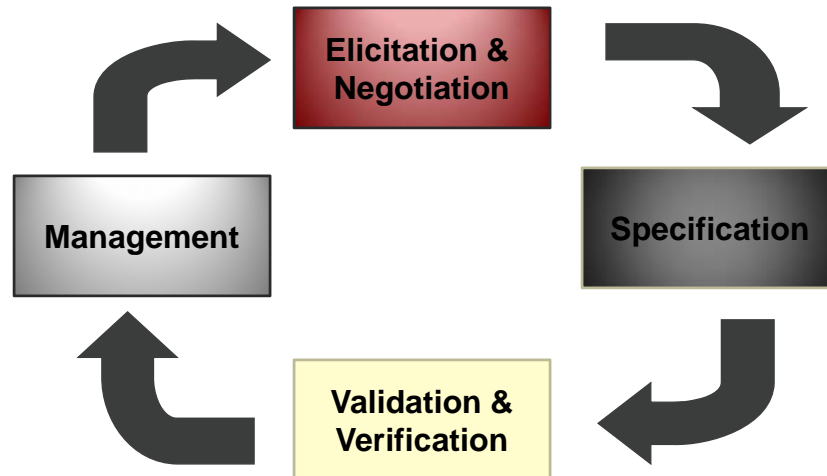
- Risk Orientation
  - Risk management is at the heart of the analysis process.
  - What are the typical risks?
    - Integration issues w/ legacy systems
    - Expected vs. actual system quality
    - Inexperience of developers
  - How to mitigate risks?
    - Prototyping (avoid IKIWISI)
    - Show changes to customer iteratively
    - Integrate existing systems sooner than later

- Is RE for the Web really that different than RE for conventional software?
- Top distinguishing characteristics
  - 1) Multidisciplinary teams: experts from various areas (multimedia, authors, software architects, usability experts, database experts, domain experts, ...)
  - 2) Stakeholders unknown: e.g. future users. Challenge is to find suitable representatives.
  - 3) Rapidly changing requirements & constraints: dynamics of Web (technology, devices, etc.)
  - 4) Unpredictable operational environment: Web changing constantly, usage is hard to predict, some factors are not under the control of the team.
  - 5) No manual for the user interface: I know it when I see it.
  - 6) Content Management: provision and maintenance of content.

- 6) Content Management: provision and maintenance of content. Quality factors:
  - Correctness
  - Detail
  - Objectivity
  - Relevance
  - Up to dateness
  - Completeness
  - Consistency
- 7) Lack of experience with technology
  - Technologies are new
  - New tools, techniques
  - Wrong estimates

- There isn't one single "right way" to do RE among the many methods, techniques, tools, etc. available.
- For your Web application project, ask the following questions:
  - What are the critical requirements?
  - How should requirements be documented?
  - What tools should be used, if any?

## The Requirements Collection Process



How to interact with Stakeholders

## ELICITATION & NEGOTIATION

## Elicitation & Negotiation



- Identify and involve (if possible) the *stakeholders*
  - Those that directly influence the requirements
  - Customers, users, developers
- What are their expectations?
  - May be misaligned or in conflict.
  - May be too narrowly focused or unrealistic.
- Why is the web application being developed in the first place?

## Techniques for Elicitation & Negotiation



- Interviewing
- Joint Application Design
- Brainstorming
- Concept Mapping
- Storyboard
- Use Case Modeling
- Questionnaires
- Terminology Comparison

## Challenges with Stakeholders



- McConnell (1996)
  - Users don't know what they want.
  - Lack of commitment.
  - Ever-expanding requirements.
  - Communication delays.
  - Users don't take part in reviews.
  - Users don't understand the technology.
  - Users don't understand the process.

## Challenges with Developers



- Users and engineers/developers speak different “languages”.
- The tendency to “shoe-horn” the requirements into an existing model
  - Saves time for developers, but results may not meet user's needs.
- Engineers & developers are also asked to do RE, but sometimes lack negotiating skills and domain knowledge.



	Präzision	Leichtigkeit der Prüfung	Aufwand	Eignung für Nicht-Experten	Skalierbarkeit
Stories			****	****	**
Anforderungslisten		*	***	****	***
Formatierte Spezifikationen	**	***	**	***	***
Formale Spezifikationen	****	****			

How to “formalize” received inputs

# SPECIFICATION

- 4 main categories of notation
  - Stories – Plain-language scenarios; understandable to non-technical persons.
  - Itemized Requirements – Plain-language lists of requirements
  - Formatted Requirements – Accurately-defined, but allow for plain-language descriptions
    - Ex. Use case scenarios in UML
  - Formal Specifications – Expressed in formal syntax & semantics; rarely used in Web applications.

- So, what's best for a Web development project?
  - Formatted requirements (i.e. use cases) and stories are heavily used.
  - Low to medium accuracy is suitable for Web apps; formal specifications very rarely required.
  - Keep effort for eliciting and managing requirements low.
  - Scalability is (most likely) important.

# VALIDATION AND MANAGEMENT

## Validation

- This step is essential to verify that requirements specification corresponds to user's needs and customer's requirements
- Iterative feedback from stakeholders is essential
  - Is the requirement feasible?
  - Do the results meet stakeholders' expectations?
- We will discuss testing in greater detail later

## Validation Techniques



- Review or walk-through
  - Reading and correcting the requirements definition documentation and models
- Audit
  - Partial check of the results presented in the review documentation
- Traceability Matrix
  - Comparison of the application objectives with the requirements of the system
- Prototyping for Validation
  - Implement a partial set of functional requirements but provide a global vision of the user interface

## Management



- Several tools are available to support Requirements management (also Open Source)
  - <http://www.paper-review.com/tools/rms/read.php>
- Tool support is crucial for big project
- Enable
  - Traceability
  - Modifiability
  - Verifiability

Taken from WebML Acer Usecase: News management system

## EXAMPLE

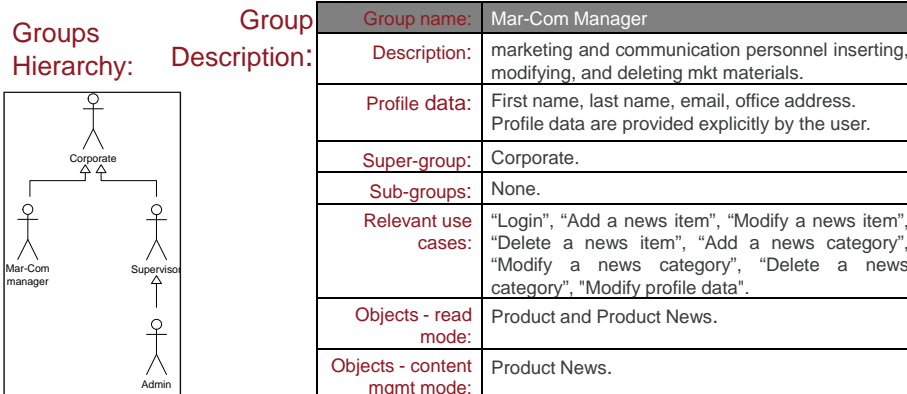
## Requirement analysis

- Revision and formalization of the collected requirements, producing in output a set of semi-formal specifications, typically in terms of:
  - I. Group specification
  - II. Use-case specification
  - III. Data dictionary specification
  - IV. Site view specification
  - V. Style guidelines specification
  - VI. Acceptance tests specification

## I. Group specification



- Clustering of users into groups (formally described)

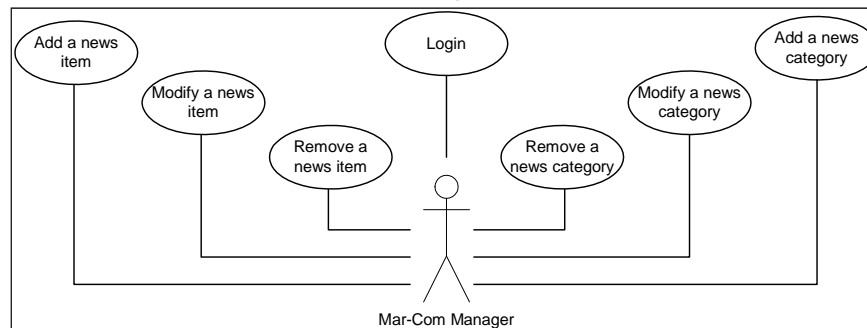


## II. Use-case specification I



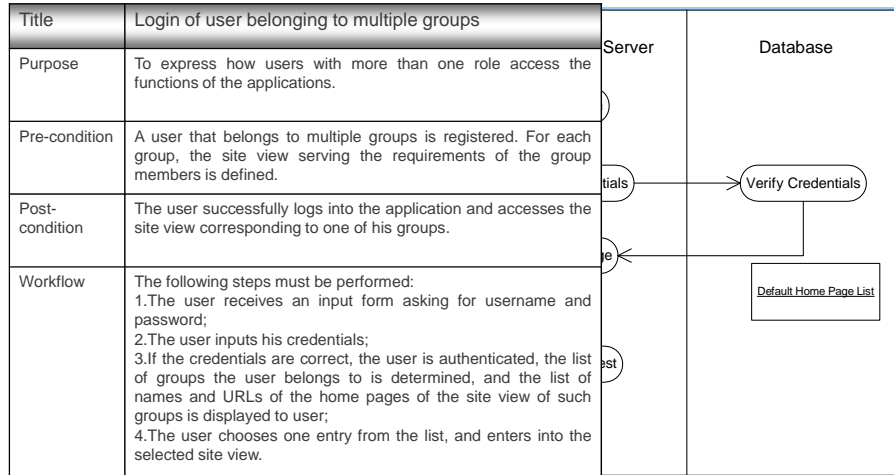
- Formal description of units of interaction with the application by users of a given group (e.g., thru tables or UML diagrams)

1. Use cases list for a user (use case diagram)



## II. Use-case specification

### 2. Single use case specification (table or activity diagram)



## III. Data dictionary specification

- List of the main information objects identified during data requirements collection
- Each entry can be specified by:

- Name
- Synonyms
- Description
- Sample instances
- Properties
- Relationships
- Components
- Super-concept
- Sub-concepts

- *NewsItem*
- *Piece of news*
- *A corporate or product piece of news*
- *TravelMate 610 launched, 20th June 01*
- *Title, Body, Image, Date, ...*
- *NewsToProduct*
- *None*
- *None*
- *Highlighted news*

## IV. Site Map specification



- **IN:** list of user groups, list of use cases, data dictionary
- **OUT:** list of needed **site maps**, specified by:
  - Name
  - Description
  - Target User Groups
  - Implemented use cases
  - Site view map: a table illustrating the different areas that compose the site view. Each **area** is specified by:
    - Area Name
    - Area Description
    - Accessed/Managed Objects
    - Priority level

Site View	News Content Management		
Description	Includes the pages through which the Mar-Com Managers will access content management functions, for inserting or updating content about news categories and news items.		
User Groups	Mar-Com Managers		
Use Cases	"Login", "Add a news category", "Edit a news category", "Remove a news category", "Add a news item", "Edit a news item", "Remove a news item".		
Site View Map			
Area Name	Area Description	Objects	Priority
News Content Management	In the default page, the user accesses the list of countries for which he is content manager and selects a country to administer. In the News Category page, the user accesses the list of news categories for the selected country. Here, the user can perform content management functions over news categories, according to the use cases "Add a news category", "Edit a news category", "Remove a news category". Otherwise, he can select one category, and access the list of the available news items in the selected category. In the News page, the user can perform content management functions over a selected news item according to the use cases "Add a news item", "Edit a news item", "Remove a news item".	NewsCategory NewsItem	High



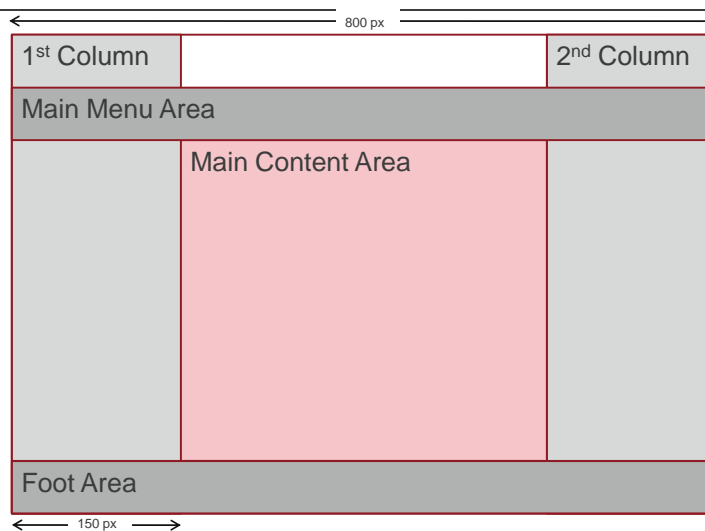
## V. Style guidelines specification



Rules for the presentation of pages:

- Specification of standard page grids: rows, columns and cells arrangement
- Content positioning specification: banners, logo, menus positioning
- Graphical guidelines: rules for graphic items like fonts, colors, borders and margins
- Device-specific and browser-specific guidelines
  
- Example: **Mock-ups**: sample representations of a few typical application pages (for a specific device and rendition language)

## V. Style guidelines specification



That's almost all for day...

## WRAP-UP

### Things to keep in mind (or summary)

- Know your Audience & Objectives
  - Balancing stakeholder interests
  - Focus on high-level requirements first.
- Elicitation & Negotiation is a learning and an iterative process
- RE requires flexibility
  - Iterative changes should be expected
  - Be sure stakeholders understand this!
- Clear documentation is critical

## Bibliography



- Mandatory reading
  - Web Engineering
    - Chapter 2
- Suggested
  - IEEE Recommended Practice for Software Requirements Specifications, IEEE Std 830-1998
  - M.J. Escalona and N. Koch, *Requirements Engineering for Web Applications - A Comparative Study*, JWE Vol.2, N. 3
    - <http://www.rintonpress.com/xjwe1/jwe-2-3/193-212.pdf>

## Assignment



- Requirement analysis for a small Web application: a library management system
- A small library (research group, max 60 users)
- Junior and senior researchers, admin staff, etc.
- Functionality:
  - Browse all books
  - See which books are borrowed (and by who)
  - Reserve a book (and borrow it)
- Form: can be informal; e.g. Some Powerpoint slides
- Due: 2 weeks
- 8 points

## Questions?

