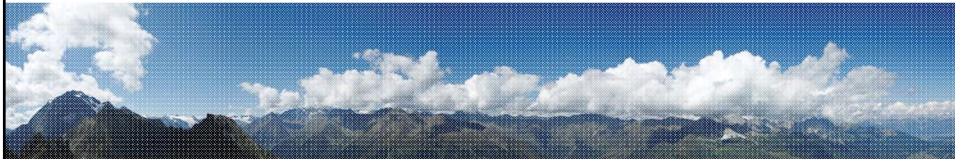


Web Engineering

Requirements Engineering for Web Applications



© Copyright 2013 Ioan Toma & Srdjan Komazec

What is the course structure?



#	Date	Title
1	5 th March	Web Engineering Introduction and Overview
2	12th March	Requirements Engineering for Web Applications
3	19 th March	Web Application Modeling
4	9 th April	Web Application Architectures
5	16 th April	Developing Applications with WebML
6	23 rd April	Testing and Usability of Web Applications
7	30 th April	Maintenance and Performance of Web Applications
8	7 th May	Web Technologies I
9	14 th May	Web Technologies II
10	21 st May	Web Application Development Process
11	28 th May	Project Management for Web Applications
12	4 th June	Web Application Security
13	11 th June	Mobile Application Development I
14	18 th June	Mobile Application Development II
14	25 th June	Final Exam

2

Overview



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

3



Why do we need Requirements Engineering?

INTRODUCTION

4

Introduction



- Requirements Engineering (RE)
 - the principles, methods and 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

5

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 a *requirement*:
 1. Condition needed to solve a problem of a user.
 2. Condition to be met or possessed by the system to satisfy a formal agreement (e.g., contract).
 3. Documented representation of conditions as in 1 and 2.
- Stakeholders: persons or organizations that are involved in the Web application and have direct influence on the requirements.

6

Tasks of requirements



- Requirements identification and negotiation
- Requirements description
- Requirements validation
- Requirements management

7

Requirement elicitation



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

8

Requirement description



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

9

Validating requirements



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

10

Requirements management



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

11

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.

12

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.*
- ...

13

Why do we need Requirements?



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

14

Good Requirements Specification I



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

15

Good Requirements Specification II



- **Requirements are 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
 - Adopts cross referencing
 - Requirements are clearly separated
- **Traceable**
 - Can be tracked from originating design documentation

16

Types of Requirements



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

17

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 adapts itself according to user or environment profile
- Transactional Requirements
 - How the application behaves internally

18

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

19

Principles for Requirements Engineering 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

20

Principles for Requirements Engineering 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

21

Principles for Requirements Engineering 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

22

Principles for Requirements Engineering IV



- 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

23

Specifics in Web Engineering



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

24

Specifics in Web Engineering



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

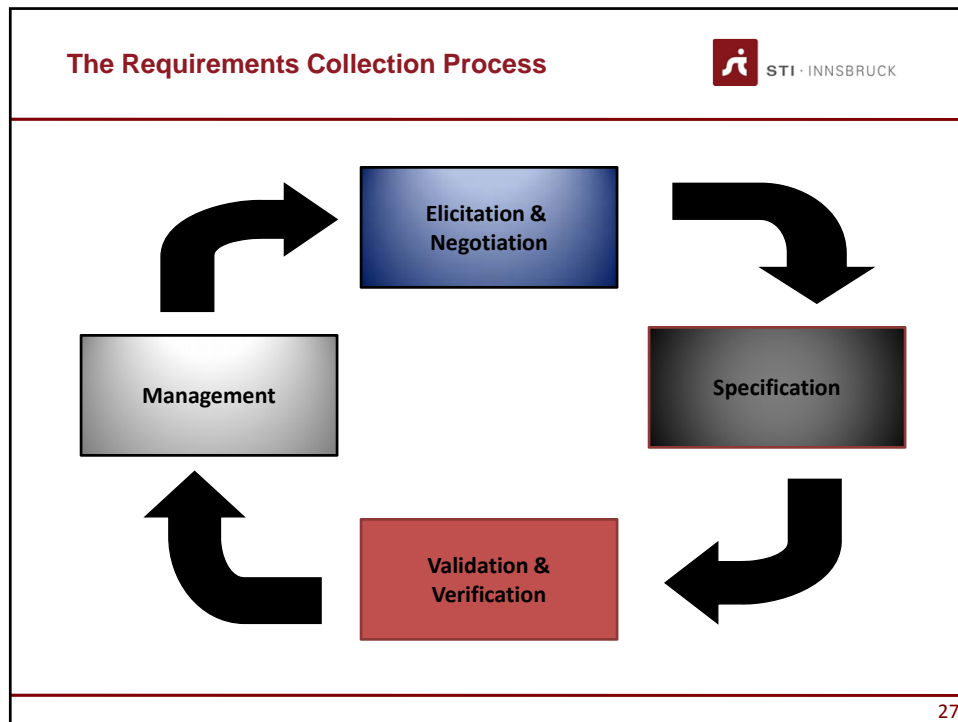
25

Adapting RE to Web Applications



- There isn't one single "right way" to do RE among the many methods, techniques, and tools 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?

26



STI · INNSBRUCK

How to interact with Stakeholders

ELICITATION & NEGOTIATION

28

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?

29

Techniques for Elicitation & Negotiation



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

30

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.

31

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.

32

How to “formalize” received inputs

SPECIFICATION

33

Specification – Traditional RE

- 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.

34

Specification – RE for Web Apps



- 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.

35



VALIDATION AND MANAGEMENT

36

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

37

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

38

Management



- Several tools are available to support requirements management (also Open Source)
 - INCOSE Requirements Management Tools Survey
 - <http://www.incose.org/ProductsPubs/products/rmsurvey.aspx>
- Tool support is crucial for big projects
- Enable
 - Traceability
 - Modifiability
 - Verifiability

39



Taken from WebML Acer Usecase: News management system

EXAMPLE

40

Requirement analysis

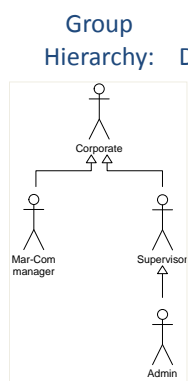


- Revision and formalization of the collected requirements, producing in output a set of semi-formal specifications, typically in terms of:
 1. Group specification
 2. Use-case specification
 3. Data dictionary specification
 4. Site view specification
 5. Style guidelines specification
 6. Acceptance tests specification

Group specification



- Clustering of users into groups (formally described)



Group Description:

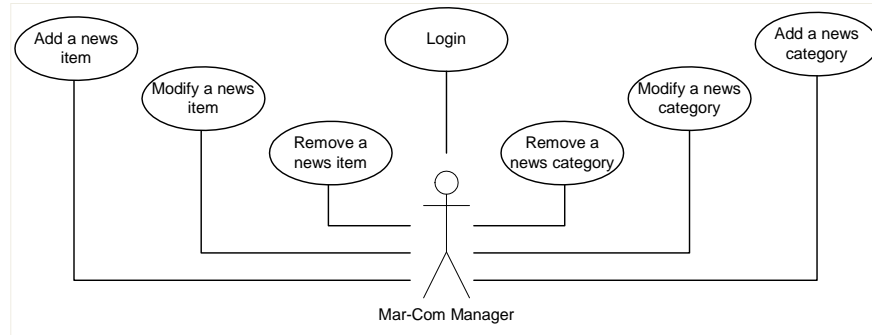
Group name:	Mar-Com Manager
Description:	marketing and communication personnel inserting, modifying, and deleting mkt materials.
Profile data:	First name, last name, email, office address. Profile data are provided explicitly by the user.
Super-group:	Corporate.
Sub-groups:	None.
Relevant use cases:	"Login", "Add a news item", "Modify a news item", "Delete a news item", "Add a news category", "Modify a news category", "Delete a news category", "Modify profile data".
Objects - read mode:	Product and Product News.
Objects - content mgmt mode:	Product News.

Use-case specification I



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

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

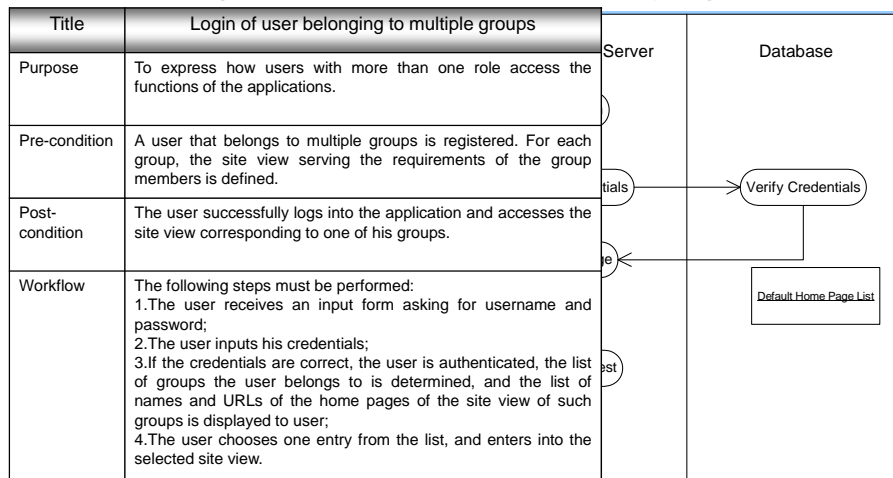


43

Use-case specification II



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



44

Data dictionary specification



- List of the main information objects identified during data requirements collection
- Each entry can be specified by:
 - Name
 - *NewsItem*
 - Synonyms
 - *Piece of news*
 - Description
 - *A corporate or product piece of news*
 - Sample instances
 - *TravelMate 610 launched, 20th June 01*
 - Properties
 - *Title, Body, Image, Date, ...*
 - Relationships
 - *NewsToProduct*
 - Components
 - *None*
 - Super-concept
 - *None*
 - Sub-concepts
 - *Highlighted news*

45

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

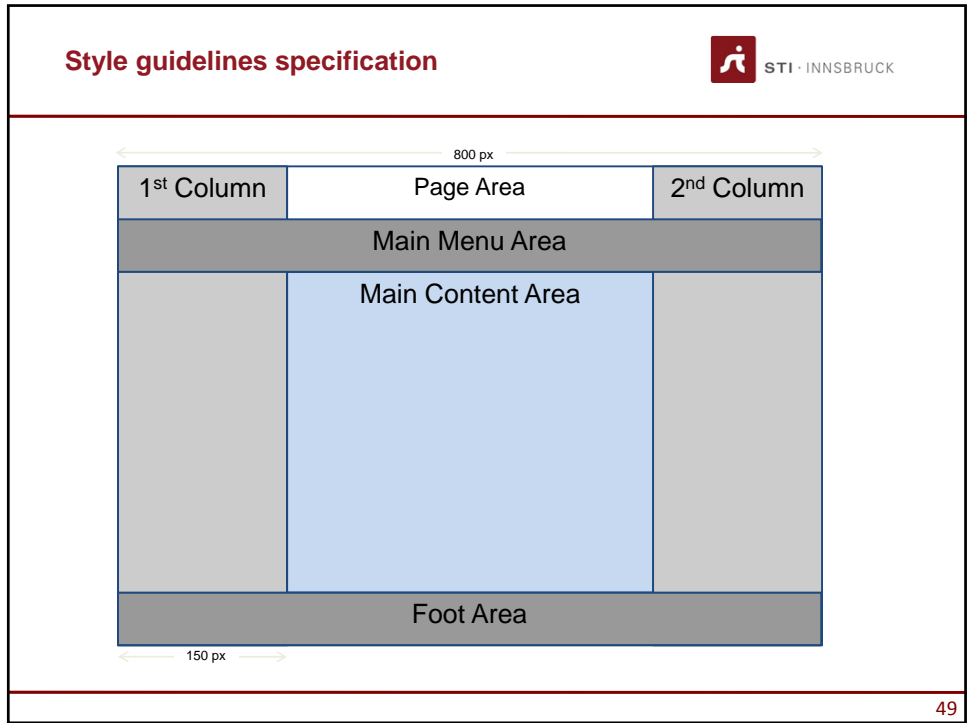
46

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

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 presentation language)



STI · INNSBRUCK

That's almost all for day...

WRAP-UP

50

Things to keep in mind



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

51

Bibliography



- Mandatory reading
 - Kappel, G., Proll, B. Reich, S. & Retschitzegger, W. (2006). *Web Engineering*, Wiley & Sons. **2nd Chapter**
 - M.J. Escalona and N. Koch, *Requirements Engineering for Web Applications - A Comparative Study*, JWE Vol.2, N. 3
- Suggested
 - IEEE Recommended Practice for Software Requirements Specifications
 - IEEE Std 830-1998
- Wiki and Web references
 - Requirements Management http://en.wikipedia.org/wiki/Requirements_management

52

Next Lecture



#	Date	Title
1	5 th March	Web Engineering Introduction and Overview
2	12 th March	Requirements Engineering for Web Applications
3	19th March	Web Application Modeling
4	9 th April	Web Application Architectures
5	16 th April	Developing Applications with WebML
6	23 rd April	Testing and Usability of Web Applications
7	30 th April	Maintenance and Performance of Web Applications
8	7 th May	Web Technologies I
9	14 th May	Web Technologies II
10	21 st May	Web Application Development Process
11	28 th May	Project Management for Web Applications
12	4 th June	Web Application Security
13	11 th June	Mobile Application Development I
14	18 th June	Mobile Application Development II
14	25 th June	Final Exam

53

Questions?



54