

Verteilte Web-basierte Systeme – SS 2006

Verteilte Web-basierte Systeme

Dr.-Ing. Martin Gaedke
Universität Karlsruhe (TH)

Verteilte Web-basierte Systeme – SS 2006

Part IV

Pre-Planning

Part 4 – Overview

Understanding The Problem

1. Introduction
2. Requirements Engineering
 1. Initiate Phase
 2. Elicit Phase
 3. Assess Phase
3. Managing Requirements & Change
4. Further Readings

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke

Verteilte Web-basierte Systeme – SS 2006

Chapter://1

Introduction

Part IV ► Chapter://1 ► Introduction

Introduction

- ☉ Web Engineering projects are likely to fail due to misunderstanding the problem – as a result of “wrong” requirements
- ☉ Standish Group’s CHAOS Reports:
 - ≡ 1994 and 1997 most significant contributors to project failure relate to **requirements**
- ☉ Computer Industry Daily / Sequent Computer Systems, Inc. study:
 - ≡ Most frequently named failure: “changing user requirements.”

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke

Part IV ► Chapter://1 ► Introduction

Requirements and the Web

Common problems:

- ☉ Requirements change
- ☉ Requirements mean different things to different people
- ☉ Requirements are not always obvious
- ☉ Requirements are difficult to express
- ☉ Requirements are related to one another
- ☉ Requirements are driven by different parties
- ☉ Requirements have different levels of detail
- ☉ Requirements are not equally easy to meet
- ☉ Requirements are difficult to gather
- ☉ And there is even more...

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke

Dealing with Requirements

- ⊗ Web applications differ significantly compared to other software applications
 - Data-intensive, different Performance aspects
 - Highly distributed and heterogeneous environments
 - High demands for security and privacy
 - High demands for accessibility and usability etc.
 - Aesthetic aspects
 - Follow trends
 - High demands for Legal aspects
 - Internet Web application: User / Audience unknown
- ⊗ Requirements **will** change *at least* if a Web application evolves

Dealing with Requirements II

- ⊗ **Defining and managing** requirements are a **major** success factor
 - Requirements mean different things to different people: Educate **all** project participants about this fact
 - Document every requirement
- ⊗ This relates to CHAOS success factor No.1 to improve user involvement
 - Who are the stakeholders?

Stakeholders

- ⊗ Stakeholder – an individual who serves as the primary source for some information that can affect the outcome of the project and/or who is affected by its outcome.
 - E.g. Customer, Employee, Marketing/Press, Administration, Customer Support, Content Creators, Domain Experts
 - Usually do not share a common understanding
- ⊗ Known Stakeholder
 - Intranet & Extranet applications → interview stakeholder
- ⊗ “Unknown” Stakeholder
 - Internet access → search for statistics
 - Later: user feedback, tracking and profiling for refining requirements

The Requirements Approach

- ⊗ *Idea* – Apply a systematic approach to define and manage requirements for a desired software product in the WWW



Verteilte Web-basierte Systeme – SS 2006

Chapter://2

Requirements
Engineering

What is a Requirement?

- ⊗ **Requirement – is**
 - 1) a **condition or capability** needed by a user to solve a problem or achieve an objective;
 - 2) a condition or capability that **must be met or possessed** by a *Web-based software product* or component to satisfy a contract, standard, specification, or other formally imposed documents;
 - 3) a **documented representation** of a condition or capability as in (1) or (2).

[Source: IEEE: IEEE Standard Glossary of Software Engineering Terminology, IEEE Standard 610.12-1990, IEEE, New York, 1983.]

Part IV ► Chapter://2 ► Requirements Engineering

Types of Requirements

- ⊗ **Functional requirements**
 - Fundamental subject matter of the system, i.e. something the product must be able to do
 - Measured by concrete means
 - E.g. Data values, class diagrams, algorithms
- ⊗ **Non-functional requirements**
 - Behavioral properties that the specified functions must have, e.g. performance, usability, security etc.
 - Measured by specific means (the “nice-looking and easy to use problem”)
 - E.g. Performance: resource response time < 2 sec with less than 100 users

Note: A lot of different notations and terms exist, try to understand the core idea

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke

Part IV ► Chapter://2 ► Requirements Engineering

Requirements Engineering

- ⊗ **Requirements Engineering (RE) –**
 - is a **recurring process** of the systematic, disciplined, and quantifiable application of approaches to define and manage the purpose and the external behavior of a proposed software product (here: in the WWW), and, which continues throughout the whole life cycle of the product
- ⊗ RE includes *elicitation, analysis, specification, verification, and management* activities
 - Traceability is therefore a **MUST**
 - Cf. IEEE Std. 830 SRS and IEEE 828 Configuration Management Plans
- ⊗ Requirements development process:

Based on "Software Requirements" 2nd Edition by K.E. Wiegers

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke

Part IV ► Chapter://2 ► Requirements Engineering

Performing RE

- ⊗ **Requirements Analyst Role (RA)**
 - Responsible for “clear communication” – bridging the gap between vague stakeholder notions and **clear specifications**
- ⊗ Some skills required
 - Interviewing skills,
 - Listening skills,
 - Facilitation skills,
 - Observational skills,
 - Writing skills

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke

Part IV ► Chapter://2 ► Requirements Engineering

Goal of RE

- ⊗ The goal of requirements engineering is the production of a good **software requirements specification (SRS)** and the disciplined management of its evolution
- ⊗ Aspects of a good Requirement *Statements* and **Specifications**:

Modifiable	Consistent	<i>Feasible</i>	<i>Necessary</i>	<i>Verifiable</i>
Traceable	Complete	<i>Precise</i>	<i>Prioritized</i>	
Versioning		<i>Correct</i>	<i>Unambiguous</i>	

Usable during operations and maintenance

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke

Part IV ► Chapter://2 ► Requirements Engineering

Levels of Requirements

- ⊗ **Business Requirements**
 - High-level objectives of the organization or customer
- ⊗ **User Requirements**
 - Tasks that users must be able to perform using the new product
- ⊗ **Operational Requirements**
 - Tasks that operations staff must be able to perform using the new product
- ⊗ **Environmental Requirements**
 - Aspects of the technology available and to be applied as well as the project’s ecosystem

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke

Part IV ► Chapter://2 ► Requirements Engineering


The Requirements Big Picture

© 2004 Dr. Martin Gocher, University of Bamberg

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke

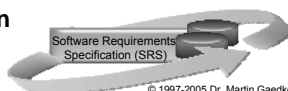
Part IV ► Chapter://2 ► Requirements Engineering

RE & Project Life-Cycle



Project evolves (Change Management)

1. **Prepare for RE activities**
 - Team Mind-Set, Glossary etc.
2. **Initiate Phase**
 - Business Requirements under change control
3. **Elicit Phase**
 - Gathering information – Refine Requirements
4. **Assess Phase**
 - Transition from gathering to analyzing
5. **Continuous evolution**
 - **Requirements & Change Management**



© 1997-2005 Dr. Martin Gaedke

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke

Part IV ► Chapter://2 ► Requirements Engineering

Prepare for RE Activities

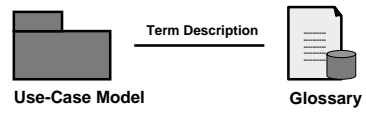
- ☉ Structure project team adequate
 - Roles, e.g. requirements analyst (RA), quality assurance, standards-specialist, domain-expert, CCB
- ☉ Project team must be aware of RE
 - Common vocabulary
 - Procedures for proposing, reviewing, and resolving changes
 - Status reports
- ☉ Prepare to elicit
 - Observe your customer/users
 - Train for interviewing (Journalism School?)

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke

Part IV ► Chapter://2 ► Requirements Engineering

Prepare: Glossary

- ☉ Defines important and common terms of a domain
- ☉ Used by analysts and developer to support same language
- ☉ System analyst or use case specifier responsible for use case glossary



Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke

Verteilte Web-basierte Systeme – SS 2006

Section://1

Initiate Phase

Part IV ► Chapter://2 ► Requirements Engineering: Initiate Phase

Objective

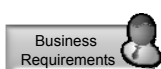
- ☉ **Initiation Phase** – Initiation is the process of formally authorizing a new project
 - Links project to the performing organization
 - Formal document and output of this phase: **Project Charter** (issued by manager)
- ☉ Usually formal initiation after completion of further assessments
 - Functional Specification
 - Feasibility study
 - Etc.

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke

Part IV ► Chapter://2 ► Requirements Engineering: Initiate Phase

Developing Solution Concept

- ☉ Identify business opportunity
 - The first customer meeting
 - Interview with key personnel, senior management (CEO, CTO, CIO)
- ☉ Gather business requirements
 - Use Cases and interview techniques
 - Focus on business improving processes, boundaries, external relationships, key business process stakeholder



Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke

Use Cases

- ☉ **Use Case (UC)** – A (possibly ordered) set of actions, including variants, that a system performs that yields an observable result of value to a particular actor
 - ≡ UC complete from the outside actor's view
 - ≡ Bulleted form, written structured:

Natural Language, nothing else – just that

Note: Be aware of the many different definitions of similar or related terms

Use Cases Realizations & Co.

- ☉ **Use Case Realizations:** Different ways to carry out a use case
- ☉ **Use Case Scenario:** Single path through a use case
 - ≡ A story telling us how something (usually a business process) is done, i.e. a particular combination of conditions within that use case
- ☉ **Use Case Diagram:** puts UCs and Actors into a graphical context
- ☉ Very effective with Senior Management

Drafting the Vision/Scope

- ☉ Analyze and consider business impact
 - ≡ Review (current and future) business processes and opportunities
 - ≡ Identify Stakeholder
 - ≡ Typical categories: common, specific, competitive, etc.
- ☉ For Internet-Sites
 - ≡ Create market requirements document (MRD)
 - ≡ MRD describes requirements of market segments, e.g. audience characteristics / profiles

Drafting the Vision

- ☉ Develop a first Vision statement
 - ≡ Strategic plan – long-term view on solution
 - ≡ Focus on business requirements
 - ≡ Aligns all stakeholder in common direction
 - ≡ Cf. For-Who-The-Is-That-Unlike-Our Product example
- ☉ Major Features
 - ≡ Labels and Names for the major capabilities of the product
- ☉ Assumptions, Dependencies, Legal Issues
 - ≡ Describe assumptions etc. mentioned by stakeholders

Example: Vision eConcierge

- ☉ **FOR** guests of the hotel
- ☉ **WHO** need assistance in enhancing their stay by choosing restaurants or cultural events
- ☉ **THE** eConcierge Service (eCS)
- ☉ **IS** a portal
- ☉ **THAT** will provide an overview of events, activities and selected restaurants (partners)
- ☉ **UNLIKE** the current black-board approach
- ☉ **OUR PRODUCT** will provide ubiquitous assistance from the early beginning of the ordering process as well as during the stay by allowing to access the system with different devices

Based on "Software Requirements" 2nd Edition by K.E. Wiegers

Drafting the Scope

- ☉ Define a first Scope statement
 - ≡ Decomposition of Vision into "Business Feature Sets"
 - ≡ Sum of high-level deliverables and services to be provided
 - ≡ Manageable chunks
- ☉ Scope of Initial Release (Version 1.0)
 - ≡ Focus only on those features that will provide the most impact
- ☉ Limitations and Exclusion List
 - ≡ List what might get in and what is definitely out of scope (core rules for the CCB)

Part IV ► Chapter://2 ► Requirements Engineering: Initiate Phase

Product Scope

- ☉ **Statement of Requirements (SOR)**
 - SOR will be the document against which change control will be exercised
 - Sometimes called Functional Requirements Document (FDS) - set of statement of requirement
- ☉ Prepare for iterative approach (multi-versioning)
 - **Baseline** current version
 - Baselined → Put under change control

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke 31

Part IV ► Chapter://2 ► Requirements Engineering: Initiate Phase

Project Scope

- ☉ Work that must be done to deliver a product with the specified features and functions
- ☉ **Aka Statement of Work (SOW)**
 - Narrative description of products or services to be supplied under **contract**
 - Sufficient detail required to allow team to determine if capable of providing the item

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke 32

Part IV ► Chapter://2 ► Requirements Engineering: Initiate Phase

Vision and Scope

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke 33

Part IV ► Chapter://2 ► Requirements Engineering: Initiate Phase

Prepare Project Initiation

- ☉ Prepare Project Initiation Documents
- ☉ Prepare risk assessment document
- ☉ Check for readiness: Personnel and training needs, possibly expert judgment
- ☉ First assignments

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke 34

Part IV ► Chapter://2 ► Requirements Engineering: Initiate Phase

Finalizing Initiate Phase

- ☉ Analyze and discuss benefits and first draft documents
 - Vision and Scope Verification by stakeholder
 - Note: ALL Requirements gathered are high-level requirements and **will be further refined or even changed**
- ☉ **Initiated: Commitment to begin the next phase**
 - Project Team – Customer: Memorandum of Agreement (*Not a contract*)
 - Project Team – Management: Project Charter

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke 35

Part IV ► Chapter://2 ► Requirements Engineering: Initiate Phase

Initiate Phase – Summary

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke 36

Verteilte Web-basierte Systeme – SS 2006

Section://2

Elicit Phase

Part IV ▶ Chapter://2 ▶ Requirements Engineering: Elicit Phase

Introduction

- ☉ Goal:
 - Refining the business requirements
 - Better understanding the problem
 - Enhancing Vision/Scope Specification by refining the scope
- ☉ Refine Requirements for the solution
 - Business Requirements (**why**)
 - Functional Requirements (**what**)
 - Non- Functional Requirements (**how**)

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke 36

Part IV ▶ Chapter://2 ▶ Requirements Engineering: Elicit Phase

Performing Elicitation

- ☉ Identify further stakeholder & Find stakeholder representative (SR)
 - SR aka Product Champion
 - Elicit stakeholder requests
 - Categorize information as well as SRs
- ☉ Ongoing Process → change control

© 1997-2005 Dr. Martin Gaedke

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke 39

Part IV ▶ Chapter://2 ▶ Requirements Engineering: Elicit Phase

Better Understanding

- ☉ Gather information in a holistic manner
 - Stakeholder (Personnel and Training Needs)
 - User profiles and audiences
 - Organizational structure (Both current and projected)
 - Market / Industry position
 - Organizational political climate
 - Business reach or scope
 - Current and future regulatory requirements
 - Product boundaries, constraints
- ☉ Requires finding stakeholder representatives
 - The question to solve: What would X need to do?
 - Be aware of implicit users

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke 40

Part IV ▶ Chapter://2 ▶ Requirements Engineering: Elicit Phase

Refining Scope

Business Requirements

- ☉ Refining high-level business requirements
 - Rewriting Use Cases
 - Focusing on Use Case Scenario
 - Rules of thumb: Describe workflow not just purpose, all possible processes within a business use case, only those inside the business, only relevant ones
- ☉ Transforming – Graphical Notation
 - Use Case Diagram: puts UCs and Actors into context

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke 41

Part IV ▶ Chapter://2 ▶ Requirements Engineering: Elicit Phase

Use Case Diagram

Business Requirements

Notation: UML

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke 42

Part IV ► Chapter://2 ► Requirements Engineering: Elicit Phase

Techniques For Gathering

- Standard approaches:
 - interviewing, shadowing, focus groups / group interview, brainstorming sessions, surveys, user instruction, prototyping, statistics
- Working with Subject Matter Expert (SME)
 - Class Responsibility Collaborator Modeling

Class Name (e.g. Person, Place, Thing, Event)	
Responsibility (what Class knows and does)	Collaborators (classes to work with)

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke

Part IV ► Chapter://2 ► Requirements Engineering: Elicit Phase

Mind-Maps

Structuring the results of Brainstorming and Interviewing sessions

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke

Part IV ► Chapter://2 ► Requirements Engineering: Elicit Phase

Categories of Information

- Based on existing Business Requirements
- Identify and classify Requirements with focus on
 - Business
 - Focus on **Data and Processes (DATA)**
 - User
 - Focus on **User-Interface Experience (UIX)**
 - Operations
 - Focus on the **System Management and Operation (SMO)**
 - System
 - Focus on the **Process and Communication Aspects (DSA)**
- And enhance the accuracy of Vision/Scope and SRS documents

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke

Part IV ► Chapter://2 ► Requirements Engineering: Elicit Phase

Business

Business Requirements

- Example from Business Requirements
 - "I need to order rooms" vs. "Only selected restaurants are allowed to add to the account"
 - This is: What vs. Who/How
- What: Functional Requirement
 - User Requirement: Customer
- Who/How: Non-Functional Requirement
 - Business Rules: Selected restaurants

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke

Part IV ► Chapter://2 ► Requirements Engineering: Elicit Phase

Business

Business Requirements

- Prioritize projected business processes
 - Will determine order and construction of planning and development process
 - Identify internal/external dependencies
- Business data (Data/Content)
 - Describe entities/objects, refine glossary
- Data flow (Process and Interaction)
 - Specify the flow of data from a business perspective
 - E.g. DeMarco & Yourdon, Gane-Sarson model or UML analysis model
 - This will be important input for DSA

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke

Part IV ► Chapter://2 ► Requirements Engineering: Elicit Phase

UI Experience

User Requirements

- (User Interface) Design
 - Design is manifested in appearance of the Web solution
 - Visual ornamental characteristics embodied in, or applied to, chunks of information as well as their composition
- Relates to dimensions:
 - Presentation – Deals with appearance of the Web application, i.e. layout, color, fonts, links, etc.
 - Navigation – Application of the hypermedia paradigm to access information or perform tasks
 - Dialogue – Relates to interacting (including manipulating) the information space
- User-Centered Design puts these three dimensions into context with the overall tasks the user has to perform to fulfill his/her business goals

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke

Part IV ► Chapter://2 ► Requirements Engineering: Elicit Phase

UIX: Experience

User Requirements

- Focus on Overall Feel of the Site
 - Who will use it? → audiences/user context
 - Where is it used? → location context
 - How is it used? → task/job context
 - Which devices? → technical context (UA restrictions)
- Cultural aspects of using the application
 - I18N, L10N, G11N
 - Identify localization requirements
 - Globalization requirements

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke

Part IV ► Chapter://2 ► Requirements Engineering: Elicit Phase

UIX: Accessibility

User Requirements

- Accessibility Requirements
 - <http://www.w3.org/WAI> Web Accessibility Initiative
 - <http://www.w3.org/TR/WCAG> Web Content Accessibility Guidelines
 - Web Accessibility Design
- User-Agent/Device specific Requirements
 - E.g. the use of specific handheld devices is required

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke

Part IV ► Chapter://2 ► Requirements Engineering: Elicit Phase

UIX: Navigation

User Requirements

- Relationship-Navigation Analysis
 - Michael Bieber et al., NJIT, NJ, USA
- Intensify hypermedia mindset of stakeholders and teams
 - Promote a hypermedia mindset
- Process RNA steps (simplified)
 - Element of interest analysis
 - Relationship analysis
 - Navigation analysis

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke

Part IV ► Chapter://2 ► Requirements Engineering: Elicit Phase

UIX: RNA's Taxonomy

User Requirements

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke

Part IV ► Chapter://2 ► Requirements Engineering: Elicit Phase

Operations

Operational Requirements

- Identify Requirements with focus on operation of the solution, e.g.

Efficiency	Availability	Security	Scalability	Testability
Maintainability	Reliability	Interoperability	Deployment	
Robustness	Reusability	Policies	Integrity	

- These often relate to **non-functional** requirements, rules or quality attributes
 - Help in specification of SLA, OLA

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke

Part IV ► Chapter://2 ► Requirements Engineering: Elicit Phase

Technology

Environmental Requirements

- Integration and Performance Requirements
 - What software, data, existing systems (**legacy**) will be accessed
 - How well or how rapidly a system must perform
- Solution elements of the distributed system
 - Hardware:** Computer, firewall, router, switch, network etc.
 - Operation:** Operating system, network topology / VLAN etc.
 - Hosting:** Web server, messaging server, database server, etc.
 - Core Services:** Web services, mail, UDDI, IP/STS, location srv
 - Application:** Application composition, integration and federation
- Analyze the Impact of solution elements on the IT environment and vice versa
 - E.g. Bandwidth, Cache, (required) response time, connections
- Focus on the **Distributed System Aspects (DSA)**

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke

Part IV ▶ Chapter://2 ▶ Requirements Engineering: Elicit Phase

Updating Vision/Scope

- ⊗ **Update Vision/Scope Specification**
 - Update product scope
 - Draft project scope
 - Possibly: Gain customer agreement
 - Start Assess Phase

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke 55

Verteilte Web-basierte Systeme – SS 2006

Section://3

Assess Phase

Part IV ▶ Chapter://2 ▶ Requirements Engineering: Assess Phase


Assessing Requirements

- ⊗ ...is about understanding and organizing requirements
 - In the general software engineering sense
 - With a dedicated focus on the product dimensions of distributed Web-based systems
- ⊗ Review of functional requirements
 - Focus on product dimensions and features
- ⊗ Dealing with non-functional requirements
 - Business rules
 - Quality attributes

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke 57

Part IV ▶ Chapter://2 ▶ Requirements Engineering: Assess Phase

Product Dimensions


Functional Requirements 

- ⊗ Review and (if possible) Classify Functional Requirements with focus on:
 - **Data**
 - **User Interface Experience**
 - **Presentation: Layout & Design**
 - **User Interaction**
 - **Navigation**
 - **Distributed System Architecture**
 - **Process (check Business Processes)**
 - **Communication**

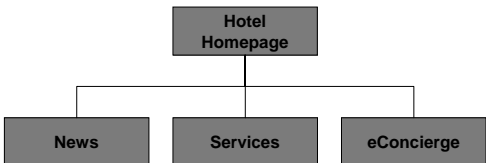
Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke 58

Part IV ▶ Chapter://2 ▶ Requirements Engineering: Assess Phase

Prototyping

Functional Requirements 

- ⊗ Proof-of-concept
 - Vertical Prototype
 - For some parts: From interface to process components, like the real-system will do
- ⊗ Mock-ups, Sample-Screenshots
 - Horizontal Prototype
 - Initial navigation / presentation concept




```

graph TD
    A[Hotel Homepage] --- B[News]
    A --- C[Services]
    A --- D[eConcierge]
            
```

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke 59

Part IV ▶ Chapter://2 ▶ Requirements Engineering: Assess Phase

Prototyping

Functional Requirements 

- ⊗ Prototyping is for Risk Reduction
- ⊗ Further approaches
 - Throwaway prototypes
 - Paperware
 - Storyboards
 - Illustrations

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke 60

Part IV ► Chapter://2 ► Requirements Engineering: Assess Phase

Features

Functional Requirements

- ☉ **Feature** – Functionality that the solution must deliver to be complete
 - Based on the functional and non-functional requirements of business and user requirements
 - Describe: benefit, increase customer or user satisfaction
 - Provides a name for a part of the scope
- ☉ **Feature Sets**
 - Set of features that belong together to support a certain need
 - May evolve and developed in several scopes

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke 61

Part IV ► Chapter://2 ► Requirements Engineering: Assess Phase

Business Rules

Non-Functional Requirements

- ☉ Approach to non-functional requirements: **Business Rules Statements**
 - Define or constrain aspects
 - Assert business structure
 - Control or influence the behavior
 - Define parameter for System Management and Operation (SMO), e.g. for SLA
- ☉ Business Rules Taxonomy
 - **Facts** (Invariants) – e.g. all pages with terms of use
 - **Constraints** – MUST, SHOULD etc. rules
 - **Action Enablers** (Trigger) – if x then y
 - **Computations** – e.g. Orders > 100 EUR → item price decr. 5%
 - **Inferences** – If Web Service does not respond within 2 sec. then the services is out of order

Cf. Software Requirements, K. Wiegers
Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke 62

Part IV ► Chapter://2 ► Requirements Engineering: Assess Phase

Quality Attributes

Non-Functional Requirements

- ☉ Labeled Collection of quality related findings
 - Efficiency, Availability, Scalability, Integrity etc.
 - Focus on the most important ones
 - Usability, security, performance, reliability and reusability are major issues!
- ☉ Example:
 - PERF-1: Homepage rendered < 1 sec with following browsers on the following hardware
 - PERF-2: Page download < 2 sec over DSL
 - USAB-1: Concierge and trained staff shall be able to submit all eConcierge forms
 - USAB-2: Restaurant pages must be WAI-compliant

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke 63

Part IV ► Chapter://2 ► Requirements Engineering: Assess Phase

Requirements Prioritization

Feature / Feature Sets	SR1 Benefit	SR1 Penalty	SR2 Benefit	SR2 Penalty	...	Total Value
F1. Order a room	4	7	5	1	...	21
F2. See list of selected restaurants	3	1	7	3	...	40
F3. Add restaurant costs to account	2	5	9	1	...	21
F4.					

- ☉ Requirements Prioritization Matrix a living document
 - Different complex solutions exists
 - E.g. Quality Function Deployment (QFD) or Total Quality Management (TQM) approaches
- ☉ Prioritize by VALUE, RISK, READINESS

© 1997-2005 Dr. Martin Gaedke
SR: stakeholder representative
Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke 64

Part IV ► Chapter://2 ► Requirements Engineering: Assess Phase

Documenting and SRS

SRS

- ☉ Documenting Requirements
 - SRS - many people will rely on it
 - Maintainable and ready for change
- ☉ Dealing with change
 - Label Requirements: Sequence number, hierarchical numbering or textual tags
 - Prepare for the unknown (To Be Determined / TBD-Process)
- ☉ Dictionary helpful to use terms in SRS consistently (use existing Glossary)

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke 65

Part IV ► Chapter://2 ► Requirements Engineering: Assess Phase

Prepare for Planning


SRS

- ☉ Check feasibility of project direction / vision
- ☉ Define tasks, schedules, deadlines, WBS, reports etc.
- ☉ Prepare risk assessment document
- ☉ Check for readiness: Personnel and training needs, possibly expert judgment
- ☉ First assignments

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke 66

Part IV ▶ Chapter://2 ▶ Requirements Engineering: Assess Phase

Manage Scope



- ☉ Check scope every Major Milestone
- ☉ Activities
 - Assign values to requirement attributes
 - Plan further progress with project and development management
 - Focus on highest risk requirements
 - Major refinement necessary?

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke 67

Part IV ▶ Chapter://2 ▶ Requirements Engineering: Assess Phase

Vision/Scope Specification

- M ☉ **Vision/Scope Specification Complete**
 - Inspect with stake holder and gain customer agreement
 - Vision/Scope document baselined
 - Initial SRS developed and baselined
 - Start Planning
- ☉ Study: Most successful projects spent 10-30% of total project resources until this milestone
- ☉ From problem description to plan
 - Transform customer into developer language

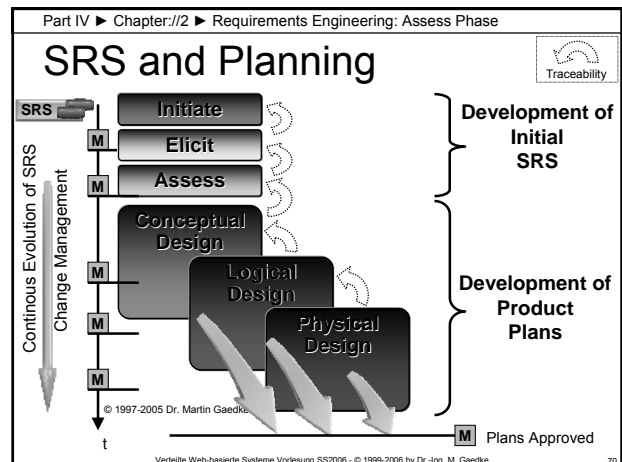
Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke 68

Part IV ▶ Chapter://2 ▶ Requirements Engineering: Assess Phase

Beyond SRS

- ☉ From SRS to Plans, Designs, Codes etc.
- ☉ Shift
 - Shift from problem definition to solution design
 - **Shift from customer language to developer language**
- ☉ SRS guides process
 - Provides a framework
 - Source for plan, schedule and build solution
 - Serves as contract between team and customer

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke 69



Verteilte Web-basierte Systeme – SS 2006

Chapter://3

Managing Requirements & Change

Part IV ▶ Chapter://3 ▶ Managing Requirements & Change

Requirements meet Change

- ☉ Requirements will change
 - Focus on: **Requirements Management**
- ☉ Causes for Change
 - Incident – Event deviates from expected behavior
 - Respond to new/changed business requirements
 - Introduce new and updated components and services
 - A Web-based system must be treated like a garden
- ☉ Key concepts to handle
 - **Incident Management**
 - **Baselining**, “Freeze late”, etc.

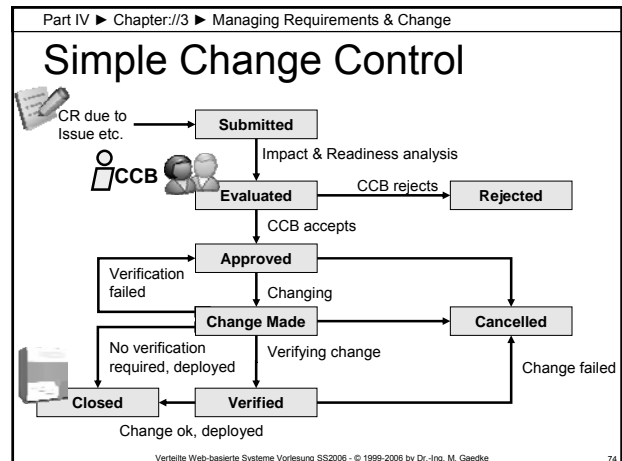
Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke 71

Part IV ► Chapter://3 ► Managing Requirements & Change

Managing Change

- ☉ **Project's Change Management**
 - ≡ No requirement, feature, function, component etc. added or changed without approval
 - ≡ Responsible Change Control Board (CCB)
 - ≡ Cf. Standard IEEE 828
 - ≡ Essential to hinder creeping user requirement, gold plating
 - ≡ **Should be applied from the early beginning of the project for all assets!** (not exclusively related to RE)
- ☉ **Change Request (CR)** – A description of a potential improvement to the Web Application, often identified by the users

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke 74



Part IV ► Chapter://3 ► Managing Requirements & Change

Change Management

- ☉ Remember:
 - ≡ Change Management is not a phase of Requirements Engineering
 - ≡ Change Management is the starting point for your project and continues from then on

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke 75

Verteilte Web-basierte Systeme – SS 2006

Chapter://4

Further Readings

Part IV ► Chapter://4 ► Further Readings

Literature

- ☉ Institute of Electrical and Electronics Engineers. *IEEE Standard Glossary of Software Engineering Terminology*, IEEE Standard 610.12-1990, IEEE, New York, 1983.
- ☉ S. Ward and P. Kroll. *Building Web Solutions with the Rational Unified Process: Unifying the Creative Design Process and the Software Engineering Process*, Rational Software Whitepaper, 1999.
- ☉ R. Oberg, L. Probasco, M. Ericsson. *Applying Requirements Management with Use Cases*, Rational Software Whitepaper, 2000.
- ☉ Chapter 5: Thomas A. Powell, *Web Site Engineering*, Prentice Hall PTR
- ☉ Chapter 2-6: S. W. Ambler, *Process Patterns – Building Large-Scale Systems Using Object Technology*, Cambridge University Press

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke 77

Part IV ► Chapter://4 ► Further Readings

Literature

- ☉ Karl E. Wiegers, *Software Requirements*, Microsoft Press, 2003
- ☉ Suzanne and James Robertson, *Mastering the Requirements Process*, Addison-Wesley, 1999
- ☉ Chapter 8.2: David Lowe and Wendy Hall, *Hypermedia and the Web – an Engineering Approach*, John Wiley & Sons
- ☉ Chapter 3: Ian Sommerville, *Software Engineering*, Addison-Wesley

=====

Further information available at **Lecture Web Site**

=====

Verteilte Web-basierte Systeme Vorlesung SS2006 - © 1999-2006 by Dr.-Ing. M. Gaedke 78