RE

Enterprise challenges

- speed
  - changes quickly, handle changes quickly
- productivity
  - profit
- knowledge
  - marker knowledge
  - customer knowledge
  - product value

Goal

Produce high-quality software, on time and on budget that fully meets the customers' real needs

Success factors

- user involvement
- clear requirements

Challenges

- requirements gathering
- documentation
- management

Costs of faults

- increasing during the SDLC (reqs., design, coding, testing, maintenance)

Project

on time (schedule), all scope, within budget
not sure that the customers also be happy

What is Soft. Req.?

Is a documented representation of a condition or capability

- needed by the user: to solve a problem
- satisfy a contract, standard, specification

Stakeholder

A person or organization that has an (direct or indirect) influence on the requirements of the system

- customer, supplier, user, PM, sponsor, tester, experts

Contract

- binding document
- agreed by the customer and the supplier
- includes the technical and organizational requirements, cost and schedule
- informal expectations

Requirement Engineering

An approach: specify and manage requirements

Requirement Engineer

- Business Analyst
- analyzes the organization or business domain
- documents, processes, systems, integrations
- may overlap: PM or Project Consultant
- often requires marketing and financial roles

Requirement levels and types

- Business level
  - Business requirements
- Stakeholder level
  - Stakeholder requirements
  - Business roles
  - Quality
- Product level
  - Constraints
  - Functional requirements
  - Non-functional requirements
  - Data requirement
  - Requirement specification
  - External interface requirements

Business requirements

- same reason, stakeholder agreement
- defines stakeholder and product requirements
- must align with the context
- requirement must help achieving business objectives

Stakeholder requirements

- main services of the system

- natural language or diagrams
- readable to everybody
- serves business objective to targeted users

Product requirements

- services and constraints of the system (in detail)
- useful for design and development
- precise and covers all cases
- structure

Transition requirements

- what needs to be done to translate to the solution
- data conversion, migration
- user access and security rights
- user acceptance test
- user preparation
- pilot test
- organization changes
- policy changes

Statement of need

- stakeholder requirements <-validate-> acceptance test
- system requirements <-verify-> system test
- sub-system requirements <-qualifying-> sub-system test
- component requirements <-qualifying-> component test

Questions for RE

- Why? What?

Types of product requirements

Functional requirements

- services what should provide
- results, behaviors
- categories: functional, data, behavioral requirements

Non-functional requirements

- quality concerns
- performance, availability, scalability, portability
- often related to functional requirements
- measurable -> test/verify/check
- speed, size, ease of use, reliability, robustness, portability
- conflict is possible -> priorities are needed

Constraint requirements

- constraints of services and functionalities
- limits between functions and quality

Domain requirements

- domain of the system
- functional or not functional

Product quality

- portability
- maintainability
- usability
- functionality
- security
- compatibility
- stability

Kano model for Quality requirements

- connection between satisfaction/dissatisfaction and requirements met/not met
- basic quality/expected quality/exciting quality

Requirement activities

Requirement inception

- starting the process
- business case, scope
- business need, market, idea

Requirement development

- eliciting
- analyzing
  - with negotiations
  - ensuring the correct understanding
  - reflect need of the stakeholders
- documenting
  - record in an accessible medium
  - communicate with the stakeholders
- validating
  - result by criteria

Requirement management

- establish and maintain agreement
- orthogonal

- structured, prepared, maintained
- maintain persistent availability over all product lifecycle

Glossary

- avoid conflict of different interpretations
- synonyms, homonyms, abbreviations, acronyms, context specific terms
- commonly accessible, centralized, stakeholder agreement, consistent, maintained

Traditional (plan-driven) -> agile (adaptive)

- individuals and interactions <- processed tools
- working product <- comprehensive documentation
- customer collaboration <- contract negotiation
- responding to change <- follow plan

Plan-driven documents

- market and business level: market requirement document
- product level: product requirement document
- build: functional specification document

Adaptive

- market and business level: product vision, portfolio backlog
- product level: product backlog
- build: actual code

Product vision

- statement of desired future state
- target customer, customer needs, product attributes
- target time frame in budget
- comparison with other products

Backlogs

- ordered list of works

Portfolio backlog

- very large items (epics)
- works across multiple project teams
- reviewed frequently
- prioritized
- scheduling
- implementation
- release

Program backlog

- works from the portfolio backlogs
- approved for implementation
- ready to be decomposed, estimated and scheduled

Product backlog

- prioritized features
- short descriptions
- release backlog: subset of product backlog
- iteration backlog: current work

Requirement inception

Goal: gain better understanding of the problem before development

- identify why the system is necessary
- identify stakeholders and their needs
- determine scope and solution boundary
- results: product vision and project scope

Steps:

- gain agreement on the problem (product vision)
    - document problem, seek agreement
    - ask the stakeholder to write statement in an agreed form
- understand the roots (problem behind problem)
    - pareto diagram
        - estimate relative impact of root causes
        - some not be worth fixing
- identify the requirement sources
    - stakeholder: directly or indirectly
    - documents: standards, legal documents, organization specification documents
    - systems in operation: legacy systems, competing systems, interfaces
    - identify stakeholders:
        - stakeholders are individuals, groups, organizations who are actively involved in the project, affected by its outcome or able to influence the outcome
- define the solution system and context boundaries
    - typical aspects: stakeholders, documents, standards, other system interactions
    - basis of the systematic elicitation
    - system context: system's relevant part
    - interface of the system: inputs, outputs
    - the boundary must be defined as precise
    - documents: use case diagrams, data flow diagrams, class diagrams
- identify the constraints to be imposed on the solution
    - restrictions on the solution space
    - limitations on abilities
    - usually non-functional requirements
    - sources: economics, politics, technology, system, environment, schedule and process
- define the scope
    - the scope comprises those aspect that cannot be changed during system dev
    - scope describes the aspects belonging to the environment and provides functionality related constraints for the system to be developed
    - product scope: features measured against predefined requirements, conditions, capabilities
    - project scope: work needed to deliver the product, measured against project management plan
    - new requirements evaluated according to the scope

Stakeholder responsibility

- making decisions
- supplies requirements
- prioritizes requirements
- communicates changes

Requirement Engineer

- responsible to understand the need of the users and stakeholders
- capabilities: analytic thinking, empathy, communication skills, conflict resolution skills, self-confidence, persuasiveness
- speaks the language of the stakeholders
- familiar with the domain
- create requirement documentation and maintain it
- can present ideas
- ensure that the system satisfies the demands of the stakeholders
- techniques
    - brainstorming
    - data modelling
    - analyzing decisions
    - data flow diagram
    - KPIs, metrics
    - non-functional requirements
    - problem tracking
    - use cases

Requirement elicitation

Process of discovering the product level requirements

- Communication with customers, system users

Techniques

- extract information
- distinct between conscious, unconscious and subconscious requirements
- level of detail
- time, budget, stakeholder availability
- experience
- survey: interview, questioning
- creativity techniques: brainstorming, 6-3-5
- document-centric techniques: perspective-based reading
- observation techniques: process observation
- support techniques: mind mapping, prototypes, workshop

Guidelines

- reuse requirements
- look for domain constraints
- prototype poorly understood
- record sources
- identify and consult stakeholders

- multiple viewpoint check
- use correct techniques

Analysis and Negotiation

Process of studying and analyzing the needs to arrive at a definition

- involves refining the requirements (to ensure the stakeholder understandability)
- includes requirement decomposition, prototype building, evaluation and negotiating

Features of the system

A feature is a service what fulfills one or more stakeholders' needs

- simple descriptions
- the features and system requirements will drive its design and implementation
- needs -> features -> system requirements (solution domain)

Managing complexity

- small and manageable
- comprehensive and complete
- decision can be made for each feature
  - for later release
  - implement immediately
  - reject entirely
  - investment further
- 50 the preferred feature number for a new or existing system

Quality criteria for features/requirements

- agreed, ranked, unambiguous, valid and up-to-date, correct, consistent, verifiable, realizable, traceable, complete, understandability

Feature/requirement interactions

- discover interactions and conflict
- prove that there are not any conflicts

Documenting requirements

- specifications
- collection of requirements
- basis for development
- complex
- accessible
- meets the quality factors

Perspectives of requirements

Static-structural (data)

- I/O data
- static data dependency of external system services

Behavioral

- reactions/events
- conditions and effects

Functional

- data processing and flows
- execution order also documented

Models and relationships

- data perspective: domain objects - class diagram
- behavioral perspective: system state - state diagram
- functional perspective: system functionalities - use case diagram, sequence diagram, activity diagram

Documenting methods

- natural language
  - pros
    - no stakeholders must learn
    - well suited to the perspective
  - cons
    - may be ambiguous
    - perspective can be mixed up
- conceptional model
  - modelling languages
  - compact, must understand
  - specific knowledge
- hybrid documentation

System Requirements Specification

- communicate requirement to the other
- simplified reuse, incorporating
- find content quickly
- baseline of change control

Good documentation

- traceable, clear, correct, verifiable, consistent, complete, unambiguous, modifiable, extendable
- about what, not how
- requirements have unique identification
- rules for requirements: short, simple, direct (one sentence / requirement)
- limited vocabulary
- avoid incompletely specified conditions
- write clearly and explicitly
- avoid complexity
- avoid requirement mixing
- avoid passive voice, negative specifications
- avoid speculation, wishfulness

Requirement attributes

- identifier, name, description
- creation, update, author, stakeholders
- version no.
- status, comments, cross references acceptance
- responsible, release no.

Requirement validation

- check and certify the requirements
- analysis and negotiations
- validate requirements and representation
- inputs:
  - draft document
  - various standards
  - expected quality
  - business knowledge
  - deadlines
- outputs
  - list of problems
  - agreed actions
- difficulty
  - comparing draft
  - requirements are correctly represented, documented and they are clear enough

Quality aspects of validation

- content
  - all relevant requirements have been elicited and documented
  - completeness, traceability, consistency, necessity, correctness
- documentation
  - by guidelines
  - conformity, format, structure, understandable, rules
- agreement
  - by stakeholders
  - conflicts are resolved
- requirements agreed by all stakeholders
  - also, after changes

Principles of validation

- involve stakeholders (internal, external, independence)
- separate the identification and correction of errors
- validation from different views
- construction of development artifacts
- repeated validation (long project, reused requirements, unknown domain)

Validation techniques

- reviews
  - peer-review (commenting)

    - walk-through
    - formal inspection
    - perspective-based reading
- prototyping
- model validation
- checklist

Requirement management

- eliciting, organizing, documenting
- maintain requirements
- handle changing requirements
    - problem: changing without any impact assessment
    - problem: unmatched/outdated requirements cause confusion and unnecessary rework

Management activities

- manage changes to agreed requirements
- manage changes to baseline
- keep project synced with requirements
- control version of documents
- manage relation between requirements
- manage dependencies of requirements
- track statuses

Expectation of management

- identification of requirements
- traceability from requirements to implementation
    - links between requirements
    - links between requirements and design models, tests, code
    - coverage and consistency analysis
- impact assessments of proposed changes
- controlled access
- change controller
    - proposal system
- tool support

Metrics and views

- requirement status and plan
- external interface status vs plan
- selective views, conclused views

Traceability

- effective management support
- traceable if you can discover who suggested, why exists, what requirements are related and how relates to other system information
- benefits
    - accountability, verifiability, impact analysis, change control, process monitoring, improved soft quality, maintenance, reuse, risk reduction

- difficulties
    - various stakeholders require different info
    - huge amount of info must be tracked
    - manual link creations is very demanding
    - specialized tools needed
- relations
    - between requirements, their sources and post artifacts
    - links between dependent requirements
- types
    - source, rational, requirement, architecture, design, interface, feature, test code

Baselines

- basis for release planning
- read-only document
- may include other documents
- enabled document comparison and management
- change history for the document
- requires access control
- can be used for estimations
- may be: created, deleted, visualized, compared, copied, signed

Versioning

- each requirement has only one version and connect with other on exact version

Change management

- how customers submit change
- how requests being monitored, prioritized and implemented
- configuration aspects
    - versioning, labeling, code tracking
- release aspects
    - defines how and when hardware and software will be made available together as a product
- change requests
    - status (proposed, rejected, accepted, included)
- impact analysis, cost estimations
- software support

Change request tasks

- classify
- evaluate
- decide about acceptance or rejection
- prioritize
- define change or new requirements
- estimate effort
- validate