MACIASZEK, L.A. (2007):

Requirements Analysis and System Design, 3rd ed.

Addison Wesley, Harlow England

ISBN 978-0-321-44036-5

Chapter 2
Requirements Determination

Topics

- From business processes to solution envisioning
- Functional and nonfunctional requirements
- Requirements elicitation
  - traditional methods and modern methods
- Requirements negotiation and validation
- Requirements management
- Requirements business model
  - system scope, business use case model, business glossary, business class model
- Requirements document

3. Requirements elicitation methods

- The least technical phase of system development
- Demands social, communication and managerial skills
- Delivers a (mostly narrative) definition of user requirements

Traditional methods of requirements elicitation

- Simple and cost effective
- When clear objectives and low project risks
- Include:
  - Interviewing customers and domain experts
  - Questionnaires
  - Observation
  - Study of documents and software systems

Interviewing customers and domain experts

- **With customers** – mostly use case reqs
- **With domain experts** – frequently a straight knowledge transfer
- **Structured** (formal) interview
  - Open-ended questions (unanticipated responses)
  - Close-ended questions (a list of possible responses known)
- **Unstructured** (informal) interview
- **Questions to be avoided**
  - **Opinionated** questions (do we have to do things the way we do them?)
  - **Biased** questions (you are not going to do this, are you?)
  - **Imposing** questions (you do things this way, don’t you?)
- **Summary report** of the interview should be sent to the interviewee within a day or two, along with a request for comments

Interview – kinds of questions

- about specific details
  - five Ws: what, who, when, where, and why.
- about vision for the future
- about alternative ideas
  - these may be questions to an interviewee and suggestions from the interviewer asking for opinions
- about minimally acceptable solution to the problem
  - good usable systems are simple systems
- about other sources of information
  - can discover important documents and other knowledge sources unknown before to the interviewer
- soliciting diagrams
  - drawn by an interviewee to explain business processes may prove invaluable for understanding the requirements

Questionnaires

- **In addition** to interviews
- A **passive** technique
  - advantage – time to consider the answers
  - disadvantage – no possibility to clarify questions and answers
  - who are the people which did not respond? how would they respond?
- **Close-ended questions**
  - **Multiple-choice** questions
    - additional comments may be allowed
  - **Rating** questions (e.g. strongly agree, agree, ...)
    - when seeking opinions
  - **Ranking** questions
    - ranked with numbers, percent values, etc.

Observation

- **In addition** to interviews (and possibly questionnaires)
- When the user cannot convey sufficient information and/or has only fragmented knowledge
- **Three forms**
  - **Passive**
    - no interruption or direct inmvolvement
    - video camera may be used
  - **Active**
  - **Explanatory**
    - explaining what is done when observed
- Should be carried for a **prolonged time**, at different times and at different workloads
- People tend to **behave differently** when watched
  - ‘**work to rule’** is a form of industrial action
  - **‘knowledge work’** is not susceptible to observation

Study of documents and software systems

- **Always** used, but may target portions of the system
- **Use case** requirements
  - Organizational documents
    - including procedures, policies, descriptions, plans, charts, internal and external correspondence
  - System forms and reports (if prior computer system exists)
    - record of change requests (defects and enhancements)
- **Domain knowledge** requirements
  - domain journals and reference books
  - ERPSs
  - using Internet searches

Modern methods of requirements elicitation

- Offer better insights at higher cost and effort
- When high project risks (unclear objectives, undocumented procedures, unstable requirements, eroded user expertise, inexperienced developers, insufficient user commitment, etc.)
- Include:
  - Prototyping
  - Brainstorming
  - Joint Application Development (JAD)
  - Rapid Application Development (RAD)

Prototyping

- '**Quick and dirty**' solution to obtain feedback
- Necessary in **complex** and **innovative** projects
- Two kinds:
  - **Throw-away** prototype
    - targets requirement determination
  - **Evolutionary** prototype
    - targets the speed of product delivery

Brainstorming

- to form new ideas or to find a solution to specific problem **by putting aside** judgment, criticism, social inhibitions and rules
- to reach consensus among stakeholders
- ‘cool’ analysis and decision making done afterwards
- The process:
  - prior to meeting, the **facilitator** provides **probortunity statement** (the problem/opportunity area)
    - generic trigger questions to challenge the participants (e.g. what features should the system support? what are the main ‘business objects’?
  - 12 to 20 participants **around a table**, feeling equal with the facilitator “one of the crowd”
  - answers to trigger questions **recorded and passed around** to stimulate more answers/ideas
  - answers/ideas get **anonymous**
  - answers/ideas are **discussed**
  - **voting** to prioritize answers/ideas

JAD

- Brainstorming-like technique capitalizing on **group dynamics**
  - Groups increase productivity, learn faster, make more educated judgments, eliminate more errors, take riskier decisions (this may be a negative though!), focus participants' attention to most important issues, integrate people, etc
- Frequently part of **facilities management** - when the operation/implementation of an organization's information systems are contracted out to a third party
- The **membership**
  - Leader (communication skills, not a stakeholder, knowledge of business domain)
  - Scribe (touch typing, software development knowledge, CASE tool skills)
  - Customers
    - Users
    - Managers
  - Developers

RAD

- Five techniques
  - **Evolutionary prototyping**
  - **CASE** tools
    - with code generation and round-trip engineering
  - Specialists with Advanced Tools (**SWAT**)
    - colocated with the users
  - **Interactive JAD**
    - scribe replaced by a SWAT team with collaborative CASE tools
  - **Timeboxing**
    - no ‘scope creep’, timeboxed project
- Problems
  - suitable for **smaller projects**; too risky for larger developments
    - inconsistent GUI
    - specialized solutions with little reuse potential
    - deficient docs
    - unsupportable solution likely

Review Quiz 2.3

1. What are the three forms of closed-ended questions in questionnaires?
2. What are the participants in a JAD session?
3. How is the RAD development team called?