

Requirements Engineering

Mark van den Brand



College komende weken

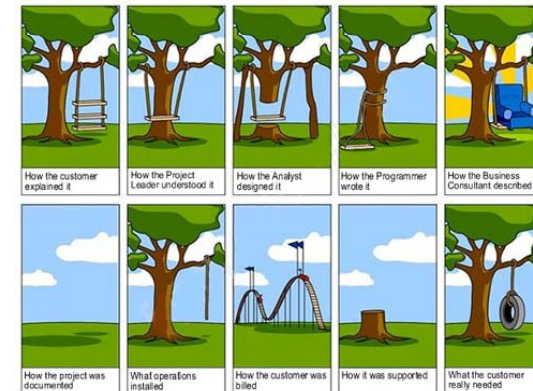
- Geen college op vrijdag 18 februari
- Maar op dinsdag 15 februari, 7+8 uur, Matrix 1.41

- Maandag 21 februari en vrijdag 25 februari:
 - Gast docenten
 - Requirementselicatie en interviewing technieken

Questions

- Waarom zit prototyping tussen waterval en agile?
- Wat is het proces model van het Software Engineering Project?
- Levert het agile proces model onderhoudbare software op? Geef een aantal motiverende redenen.

Requirements Engineering



Domain Analysis

- The process by which a software engineer learns about the domain to better understand the problem:
 - The *domain* is the general field of business or technology in which the clients will use the software
 - A *domain expert* is a person who has a deep knowledge of the domain
- Benefits of performing domain analysis:
 - Faster development
 - Better system
 - Anticipation of extensions

Domain analysis

Domain analysis document:

- Introduction
- Glossary
- General knowledge about the domain
- Customers and users
- The environment
- Tasks and procedures currently performed
- Competing software
- Similarities to other domains

Domain analysis

Example document:

<http://www.site.uottawa.ca/~laganier/seg3700/cemdomain.htm>

Domain analysis

A. Introduction
This document describes background information that has been gathered about events in organizations and how they are handled. This information is to be used to guide the development of software to automate the process of informing people.

B. Glossary

- **Event:** A meeting, a social occasion or an activity involving a significant number of employees. Several categories of events have been identified.
 - **Open event:** An event that starts at a precise instant but with no predetermined duration. Meetings and celebrations often fall into this category.
 - **Fixed event:** An event that starts at a precise instant and with a predetermined duration. Course lectures and seminars are examples of this kind of event.
 - **Day events:** An event associated with a particular day without precise start and end times. Birthday, thematic journey are such events.
 - **Recurrent event:** An event that occurs repeatedly on some regular schedule (for example daily, weekly or monthly). The event normally has a starting date and an ending date. Courses and social activities are often recurrent events.
 - **Composite event:** An event composed of several sub-events. For example, a training activity can be composed of a registration period (fixed event), a series of seminars (recurrent events), and a final evening celebration (open event).

C. General knowledge about the domain

- Most events occur during working days.
- Events are generally associated with a location (where the event is to be held).
- The name of a contact person is often associated with an event. That person is the one that organize the event or that can give complementary information about the event.
- Group of interest are often created to target more precisely peoples that might be interested by a certain event.
- Outdated events are of little interest.
- Each event has a title, a location and the name of a contact person associated with it.
- Events may be seen by anyone within the organization, but there should be some control over posting events to reduce the risk of duplicate postings and other chaos.

D. Clients and users

Potential clients are medium or large companies whose staff use computers to perform many kinds of daily work. Others impacted by the system will include:

- Employees at all levels have an interest in events and are potential users of event manager software. These employees range from computer novices to sophisticated programmers; however they all have a computer on their desk, have access to a web browser. They have been exposed to and accept new technology but are subject to tight time constraints and have little time to learn and customize new software tools.
- A system administrator normally manages the computer environment.
- Technicians typically install software that must be available to all users.

E. The environment

The actors all have a computer on their desks, it is most common for this to be MS-Windows based, but a significant minority of potential clients use other platforms.

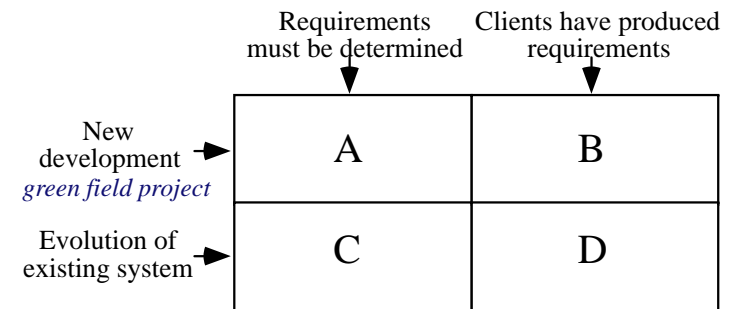
A wide variety of software is installed on these computers, with each actor having a unique configuration. Some software may be installed on every computer using a site-wide license.

F. Tasks and procedures currently performed

- Informing others about events: The organizer of the event (or someone who has heard about it) compiles information concerning an event and posts it so members of the organization can see it.
 - One approach is to post the event on a physical bulletin board at a place that most users pass each day. In a large organization, this method is too unreliable due to the sheer volume of events.
 - Another approach is to send email and paper leaflets to all members of the staff without knowing exactly who will be interested. In some cases there are mailing lists to make event notification more selective; however mailing lists are of little interest unless they are about an event.
 - In either approach there is typically no central listing of available events and the information about an event is not presented in a standard fashion.

It was hoped that other methods to post and organize events to ensure the information is current. However this can lead to duplication and inconsistency.

Starting Point for Software Projects

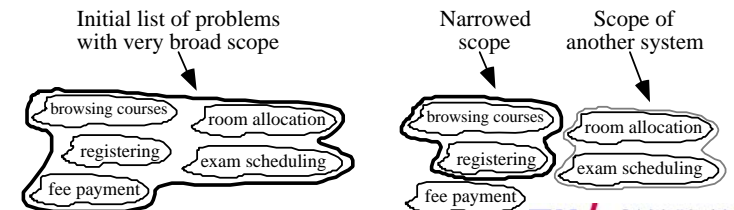


Defining Problem and Scope

- A problem can be expressed as:
 - A *difficulty* the users or customers are facing,
 - Or as an *opportunity* that will result in some benefit such as improved productivity or sales.
- The solution to the problem normally will entail developing software
- A good problem statement is short and succinct

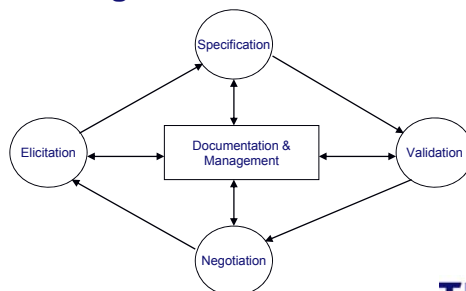
Defining the Scope

- Narrow the *scope* by defining a more precise problem
- List all the things you might imagine the system doing
 - Exclude some of these things if too broad
 - Determine high-level goals if too narrow
- Example: A university registration system



Processes in requirements engineering

- Requirements elicitation
- Requirements specification
- Requirements validation and verification
- Requirements negotiation



What is a Requirement ?

- It is a statement describing either
 - 1) an aspect of what the proposed system must do,
 - or 2) a constraint on the system's development.
- In either case it must contribute in some way towards adequately solving the customer's problem;
- the set of requirements as a whole represents a negotiated agreement among the stakeholders.
- A collection of requirements is a *requirements document*.

Types of Requirements

- **Functional requirements**
 - Describe *what* the system should do
- **Quality requirements**
 - *Constraints* on the design to meet specified levels of quality
- **Platform requirements**
 - *Constraints* on the environment and technology of the system
- **Process requirements**
 - *Constraints* on the project plan and development methods

Functional Requirements

- What *inputs* the system should accept
- What *outputs* the system should produce
- What data the system should *store* that other systems might use
- What *computations* the system should perform
- The *timing and synchronization* of the above

Quality Requirements

- **All must be verifiable**
- **Examples: Constraints on**
 - Response time
 - Throughput
 - Resource usage
 - Reliability
 - Availability
 - Recovery from failure
 - Allowances for maintainability and enhancement
 - Allowances for reusability