

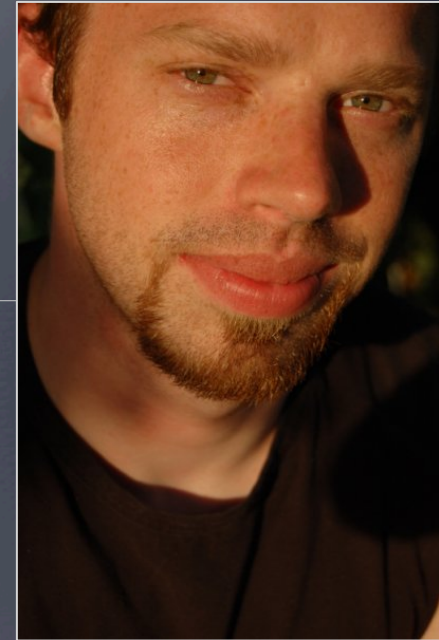
Workshop 1

Requirements Specification, Natural Language Requirements and User Stories.

Emil Börjesson, 2011
Emil.Borjesson@Chalmers.se

Who am I?

- Bachelor in Computer Science
- Master in Software Engineering
- PhD. Student in Software Engineering
 - Academic PhD. Student working close to industry
 - Focus area at the moment: Verification and Validation
- Contact:
 - Emil.Borjesson@Chalmers.se
 - Jupiter building, 4th floor



Recap from yesterday

- What is a Requirement?
 - What is Requirements Engineering?
-

Requirements?

Customer Contact

Project Proposal

Requirements Elicitation

Requirements Specification

Design

Implementation

ETC

High-level
description of the
software
development
process.

Overview

Implementation

SW Design

Further
development

Requirements

Maintenance

Business goals

Testing

And more!

How hard can it be? ;)



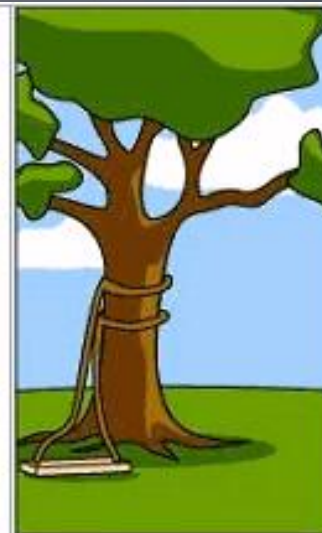
How the customer explained it



How the project leader understood it



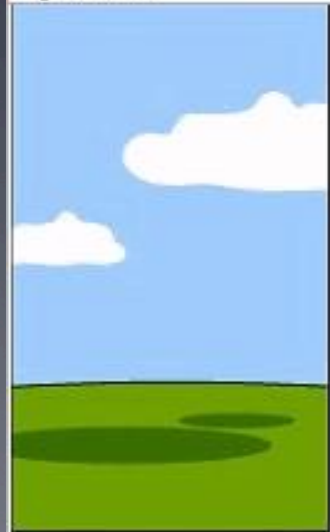
How the analyst designed it



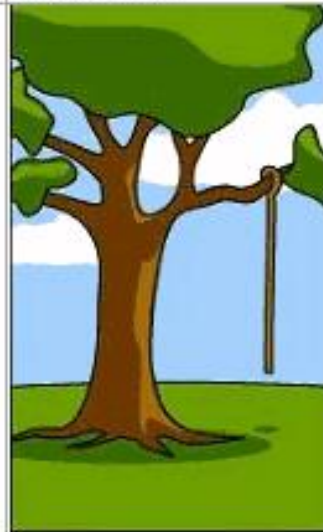
How the programmer wrote it



How the sales executive described it



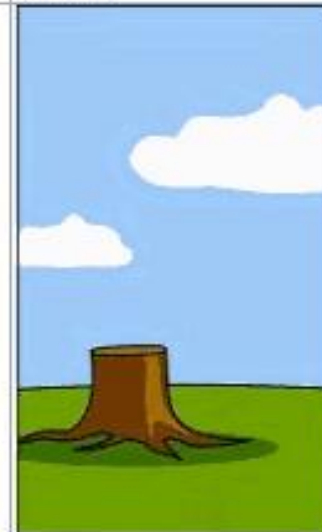
How the project was documented



What operations installed



How the customer was billed



How the helpdesk supported it



What the customer really needed

How hard can it be? ;)

- As it turns out... VERY!
 - Requirements are subject to a lot of problems
 - **Ambiguity** – Requirements can and will be interpreted differently by different people depending on their roles etc.
 - **Different taxonomies** – Different industrial domains, companies, even roles at a company have different understanding of different requirements.
 - **Work practices and processes** – Different processes (i.e. Agile vs Waterfall) handle Requirements engineering in different ways.
 - **Requirements change** – Customers don't know what they want, technology changes, competitors steal your idea, markets change, etc.
 - **AND MORE!!!**
-

How can we mitigate these problems?

- Requirements elicitation
- Requirements specification
- Multi-aspect and Multi-level specification
- Different methods



Requirements Specification?

“The deliberate documentation of requirements to a degree that makes the associated risks tolerable”

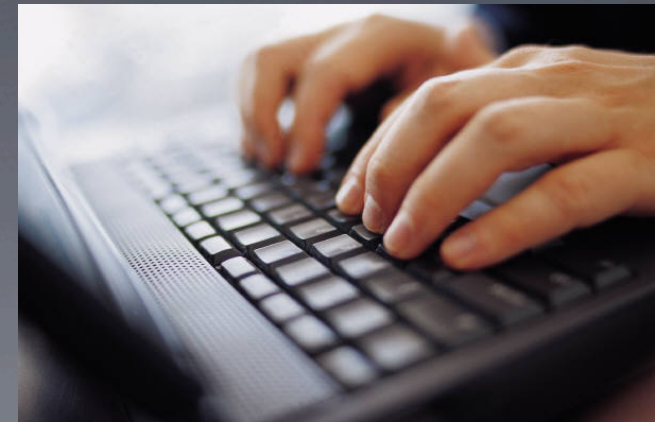
I.e. write down requirements in a specific way to avoid problems later during design, development, testing, etc.

Why?

- Requirements are still ambiguous & open-ended after elicitation, which leads to:
 - Developers make decisions/assumptions later, which leads to:
 - User <-> Dev difference: User not satisfied
 - Dev <-> Dev difference: Inconsistent system
 - Overall: Higher cost!
 - BUT:
 - Goal is ideal PRODUCT not ideal Req Doc!
 - Thus: Just enough Req Spec to reduce Risks!
-

Purpose of Req. Specification?

- Communication device between all parties
- Customers, Marketing, Sales, Finance, Management, Devs, Testers
- Drives design and choices
- Drives testing
- Drives project management
- Basis for evolution / releases



Specification Techniques

Word doc,
Excel doc,
DB/ Req tool

Text

Scenario,
Use Case,
Storyboard

Interaction/
Sequence-based

State
transition
diagram,
UML state
diagram

State-based

Decision
tables/
trees

Decision-based

Planguage,
Volere,
Probabilistic,
Quality
Pattern

Quality
Requirements

UI-
Standards,
Text,
Prototype,
Sketches

User Interface

Z, CSP,
VDM,
Property-
based

Formal

How do you document the requirements?

Industrial Practice	Percentage Used
Office (Word, Excel, Visio)	23.8%
None	15.3%
Requisite Pro	10.2%
Quality Center	9.6%
Don't know	5.1%
Focal Point / DOORS	4.0%
Caliber	3.4%
Customer-Specific	3.4%
RSA	3.4%
Clear Case	3.4%
Req Test	3.4%
Rest/Other(max 2 mentions per tool)	18.6%

So much to choose from!?!

- Stakeholders must understand the requirements (Natural Language)
- Models where NatLang has risks:
 - Complex interactions/sequences/states/decisions
 - Interfaces
 - BUT not “One model to rule them all!”
- Quality requirements:
- Quantify
- Capture in structured English or PLanguage



Natural Language Requirements

- What it sounds like, requirements documented in natural language sentences.
 - I.e. The button should be red.
 - Structured and unstructured
 - Documented in Word, excel, powerpoint...
 - Easy for the customer to read
 - Ambiguous!
-

Aspects to think about!

- Use complete sentences! Use correct grammar & spelling!
- Keep sentences short
- Use Active Voice
- User Terms Consistently
- State requirements in a consistent fashion
 - ex: “The [actor] shall [action verb] [observable result]”
 - “The door management system shall display all users that have exited the building in the past 48 hours”
- Avoid Vague Terms. Avoid Comparative Words.
- RFC 211



RFC 2119?

- MUST = REQUIRED = SHALL
 - Absolute requirement of a specification
 - MUST NOT / SHALL NOT: Absolute prohibition
 - SHOULD = RECOMMENDED
 - May exist valid reasons to ignore in particular circumstances, but the full implications must be understood
 - SHOULD NOT / NOT RECOMMENDED
 - MAY = OPTIONAL
 - item is truly optional
-

Natlang requirement examples

- The quit button may be big.
 - The logoff button should be red and big.
 - The logoff button should be red, located in the top left corner and be big.
 - The logoff button must be red, located in the top left corner of the UI and be 25% bigger than all other buttons.
 - The logoff button should be red, located in the top left corner of the UI and be 500x500 pixels.
-

Structured NatLang

- ID number
 - Title
 - Preconditions
 - Postconditions
 - Rationale
 - Description
 - Relations
-

Structured NatLang Examples

Idnum: 5

Title: Save document

Preconditions: A document exists with unsaved data.

Postconditions: A new file/old file has been created/overwritten with unsaved data.

Rationale: Users want to save their work for the future.

Description: By clicking the save icon or selecting "save" from menu "Archive" the user can save the document.

Relations: None

Idnum: 6

Title: Close application

Preconditions: The application is running.

Postconditions: The application closes and data is saved.

Rationale: Users want to quit the application

Description: By clicking on the close button or selecting "Quit" from menu archive the user can terminate the application. If unsaved data exists in the document in the application the user is prompted with a save window.

Relations: 5

Use Cases

- Scenario based
 - Captures customer interaction and needs
 - Different types
 - Structured
 - UML support
-

Standard UML Use Case

Enter Building

Description: A user enters the building

Precondition: The person is a user in the system

Postcondition: Person has entered the building

User Intention

1. User swipes magnetic card

4. User enters the code

8. User opens door & enters building

System Response

2. Verifies that card is valid

3. Asks for user code

5. Verifies that code is valid for swiped card.

6. Opens door

7. Sound buzzer

9. Logs entry of user

Use cases, another notation

Start computer

Description: User starts computer

Precondition: Computer is turned off

Postcondition: Computer is turned on

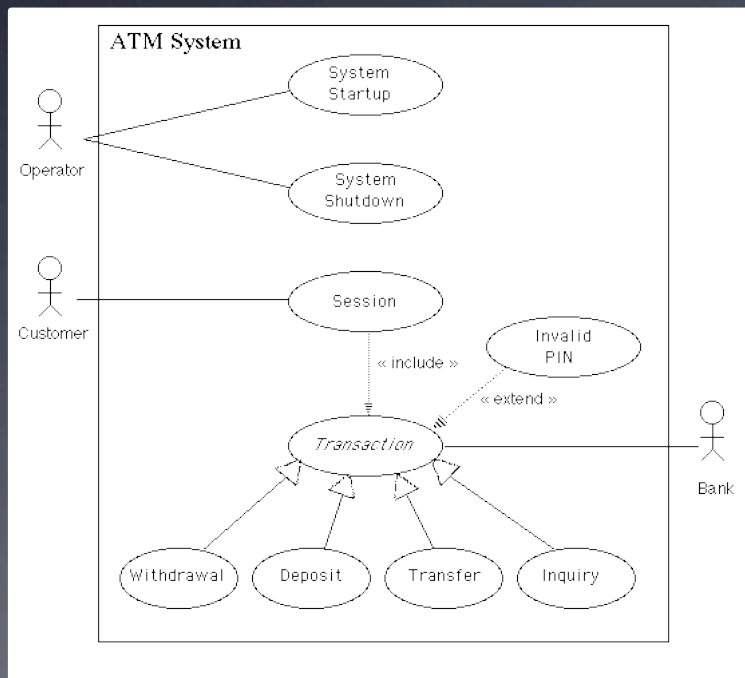
Main Scenarion:

1. User pushes "ON" button
2. Computer loads operating system (OS = Windows)
3. OS checks for updates
4. OS finds no updates
5. OS shows login screen to user
6. User enters password
7. OS checks password
8. Etc.

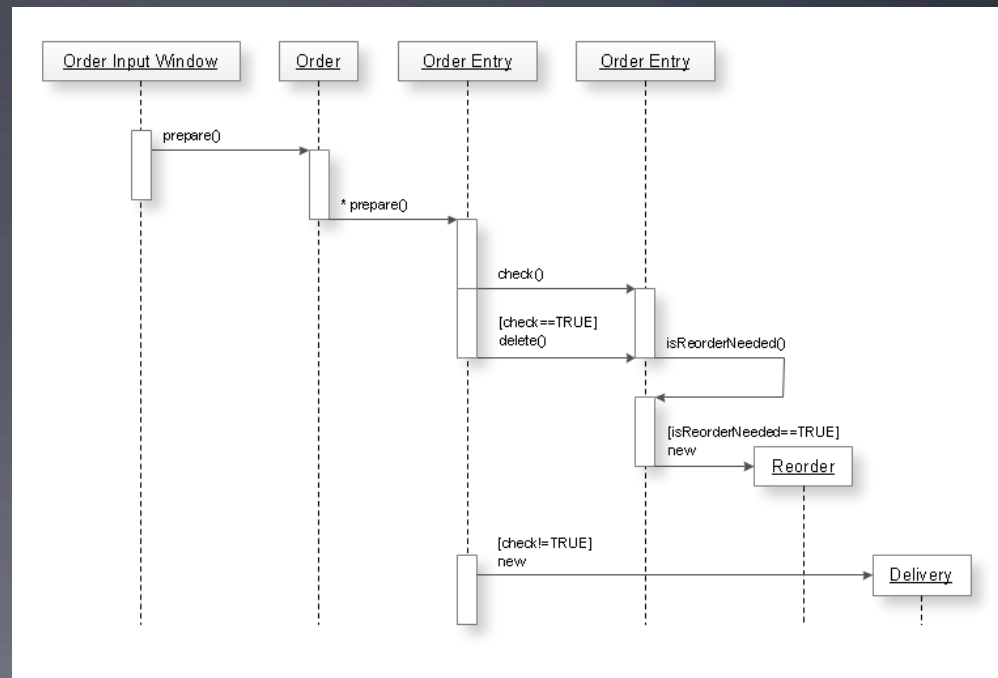
Alternative Scenario

- 2B. Computer fails to load operating system
 - 2C. Computer shows bluescreen of death
 - 2D. Computer is not turned on
-

Graphical Use Cases



Use case



Sequence Diagram

User Stories

- Written with customer on small notes (Agile practice)
 - Should be short and descriptive
 - Not finite once written, i.e. customer can change his/her mind
 - User stories can follow the templates:
As a <role>, I want <goal/desire> so that <benefit>
 - Or simplified as:
As a <role>, I want <goal/desire>
-

User Stories Examples

- Customer DB search
 - As a user, I want to search for my customers by their first and last name.
 - Drone Navigation
 - As an autonomous drone, I want to control my onboard navigation system.
 - Closing application
 - Upon closing the application, the user is prompted to save (When Anything has changed in data since the last save!)
-

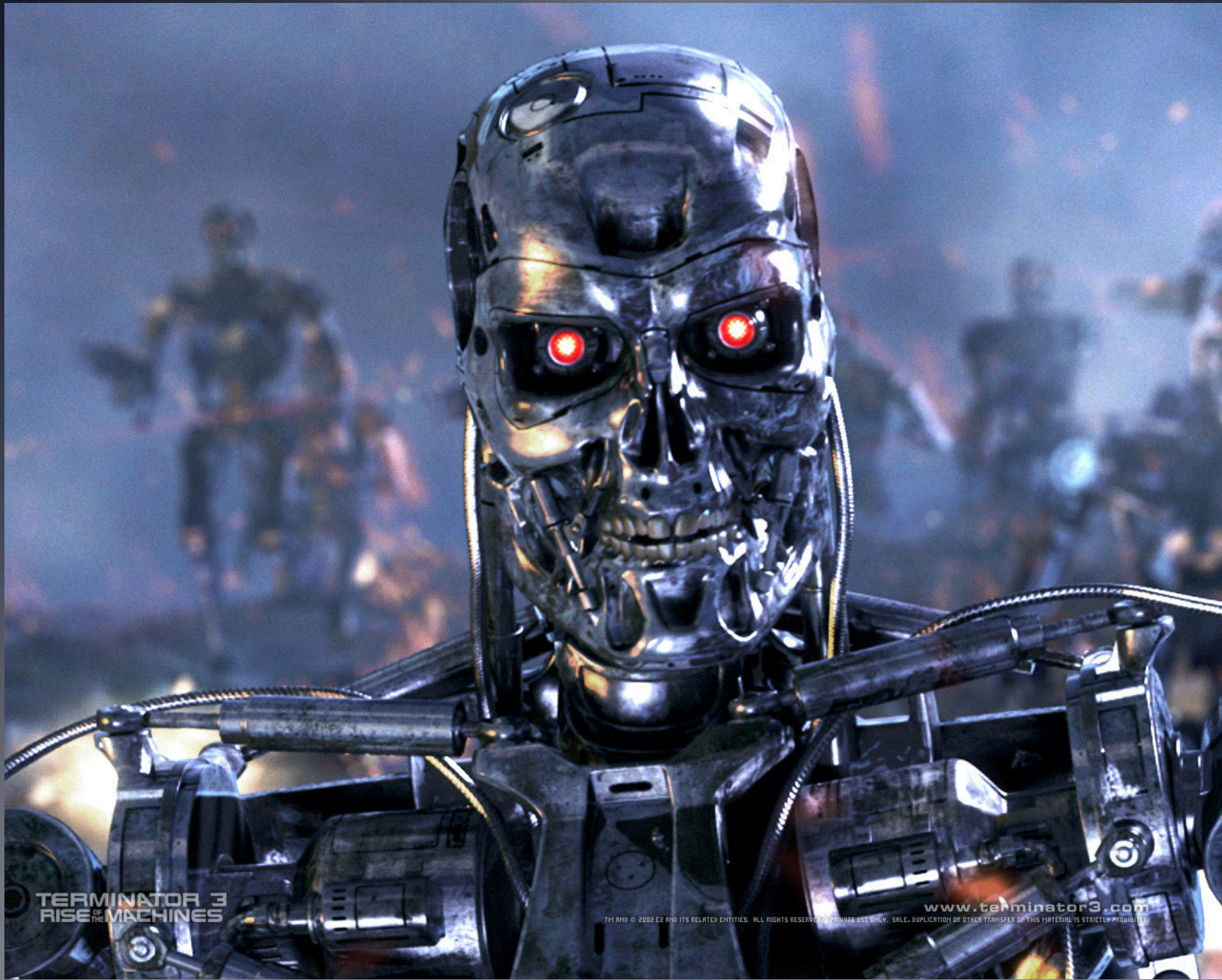
User story card



As role, I want feature, so that value.

Assignment

- You are a Requirements Engineer working at the company Cyberdyne Systems that just got a new defense contract to develop a new system with the project name Cyberdyne Systems T-101, following the success of the T1 series. Your assignment is to create software requirements for the system.
 - The T-101 is a cybernetic organism, a robot called a Terminator. Those of you not familiar with the movies, portraying Arnold Schwarzenegger, can read about them on the internet, search for “Terminator”. In this assignment we are however only interested in the Software requirements of the system!
-



TERMINATOR 3
RISE OF THE MACHINES

www.terminator3.com

TM AND © 2003 C2 AND ITS RELATED ENTITIES. ALL RIGHTS RESERVED. PRIVATE USE ONLY. SALE, DUPLICATION OR OTHER TRANSFER OF THIS MATERIAL IS STRICTLY PROHIBITED.

Assignment Specification

- Write **15 natural language** Software requirements for the T-101.
 - Write **5 use cases** for the T-101, using standard UML use case notation.
 - Write **3 user stories** for the T-101.
 - Overlap among Natlang, US and UC **are** permitted.
 - Only Software requirements! No hardware.
 - Maximum number of pages: 6
 - Deadline: **5/9 at 18.00!** Submission through the Fire system (Link on the website)
 - The assignment is Individual!
 - Submissions should only include your anonymous code. **NO NAMES!**
-

FIRE system

- Link: <https://fire.cs.chalmers.se:8067/cgi/Fire-reqeng>
 - (Also on course homepage)
 - Create an Account with the same e-mail you used to get your anonymous code!
 - Read carefully when you are submitting your solution!
 - Missing the deadline because you failed to follow the instructions in FIRE is not a valid excuse!
 - Neither is:
 - My dog ate my computer.
 - I was too stoned to do it maaaaan!
 - I don't like assignments, but my split personality does. Unfortunately he was sick this weekend.
 - Etc.
-

Reminder!

- Tomorrow there will be a guest lecture with people from Saab Air Traffic Management (ATM)!
- Your chance to get some insight into how industry works with Requirements Engineering but also Software Engineering in general.
- Write 5 questions each regarding Software Engineering at Saab and 5 questions about Requirements Engineering at Saab.



Next time

- Requirements documentation (SRS's)
 - Quality attributes of requirements and groups of requirements
-

Questions?
