Quality starts by defining your goals!
1. Why do we need Requirements Engineering?
2. The Objectiver approach
3. Deeper inside the Objectiver approach
4. References
5. Demo
Requirements Engineering – Why?

Project completed on time, on budget, and with all functions originally specified

**success**
16%

Project completed but... over budget, late, and with fewer functions than initially specified

**challenged**
53%

Project cancelled before completion or never implemented

**failure**
31%

Source: The Standish Group. Sample size was 365 companies and 8380 software projects.
http://www.standishgroup.com/sample_research/chaos_1994_1.php
Top 10 Reasons for Project Failures

- Lack of User Input 13%
- Incomplete Reqs & Specs 12%
- Changing Reqs & Specs 11%
- Lack of Executive Support 8%
- Technology Incompetence 7%
- Lack of Resources 6%
- Unrealistic Expectations 6%
- Unclear Objectives 5%
- Unrealistic Time Frames 4%
- New Technology 3%
1. Why do we need Requirements Engineering?
2. The Objectiver approach
3. Deeper inside the Objectiver approach
4. References
5. Demo
Objectiver goal-driven approach to requirements engineering

Start by studying the problem and its environment rather than by specifying the solution straight away.

Link the project requirements to the business strategy.

Establish traceability from business processes to business goals.

Find out how changes to business goals shall impact your processes and project requirements.
A strong scientific background

University of Louvain (Belgium):
- research

Non-profit applied research organization:
- formal tools (FAUST)

Private company:
- Consulting services
- Objectiver vendor

Market
1. Why do we need Requirements Engineering?
2. The Objectiver approach
3. Deeper inside the Objectiver approach
4. References
5. Demo
Goal Model

1. Identify all the goals pursued by the project stakeholders and all involved people
2. Link them to higher-level strategic goals
3. Define how goals can be achieved

The use of a highly-graphic notation facilitates:
- Talking about project requirements and goals
- Quick identification of concept interrelationships
- Validation of the model
Goal Model

Explains WHY

Access limited to authorized users
Privacy protected while service being used
Customer automatically signed out after 10 minutes of inactivity

Explains HOW

High-level of security
Account information secured
Strong system security level

requirements

Java on both client and server sides
Highest level of encryption used (128-bit SSL) for transmissions

online mail messages and questions are secured

refines

goal
Expectations

Include in your model what is to be expected from the environment (system context)

- Access limited to authorized users *
- Public key cryptography used for authentication
- Access to user's account controlled by a front-end Java applet
- Password kept secret by customer

(expectation)
Alternatives

When alternatives are identified, further interviews and analysis shall be conducted to decide which solution shall be preferred over the other one.
Some goals may conflict with each other under certain circumstances.

Solutions:
1. Introduce new goals/requirements that shall prevent the conditions leading to the conflict from occurring
2. Opt for an alternate solution (if we can’t live with this conflict in the system)
Back to our alternative

The conflict is used to guide our choices between existing alternatives
Obstacles

1. Identify the obstacles that could prevent reaching any goal
2. Identify new goals/requirements that shall resolve or alleviate these obstacles

![Diagram showing obstacle, resolve, and obstruct relationships]
1. Refine generic obstacles as more specific obstacles
2. Resolve the specific obstacles individually
1. Identify agents
2. Assign them responsibility for the expectations and requirements identified in the goal model

Agent: a human, device or system component
- **system agent**: part of the system being modeled
- **environmental agent**: part of the system environment

Requirement: a low-level goal placed under the responsibility of a unique system agent

Expectation: assigned to an environmental agent. As such expectations cannot be enforced by the system.
Responsibilities for requirements

Assign requirements to system agents

- Highest level of encryption used (128-bit SSL) for transmissions
- Customer automatically signed out after 10 minutes of inactivity
- Access to user's account controlled by a front-end Java applet
- Public key cryptography used for authentication

is responsible for

system agent

requirement
Assign expectations to environmental agents

Responsibilities for expectations

Password kept secret by customer

environmental agent

User <<Abstract>>
Model the domain objects, including entities, agents.
Operation Model

- Model business or system processes
- Get traceability from operations (processes) to requirements

**Operation** = behavior that a system agent has to adopt to meet a requirement (that he’s responsible of)

**Events** trigger operations
Operation Model

- Event: User connection
  - Cause: Security module
  - Performance
  - Input: Verify credentials
    - Output: Credentials verified
      - Access denied
        - Cause: Display error message and return to login page
      - Access granted
        - Cause: Display menu
  - Input: User login details
    - Output: User access history
  - Input: Log request
    - Output: User login details

- Operation: Request login details

- Agent: Objectiver
Traceability to requirements

Link operations to requirements for traceability

Access to user’s account controlled by a front-end Java applet

operationalizes

is responsible for
1. Why do we need Requirements Engineering?
2. The Objectiver approach
3. Deeper inside the Objectiver approach
4. References
5. Demo
Objectiver, the Req Eng tool

Diagram editor

Property editor

Text editor
**Connection of a user to the system**

*Details* (username and password), the system shall verify his credentials *(Verify credentials)* before granting or denying access to the system.
More than a simple drawing tool

• Elicit and specify requirements in a **systematic** way

• Produce structured, motivated, easy to understand requirements documents

• Facilitate **communication**
• **Calls for tender** managed more easily

• Provide **traceability** from processes to business goals

• Highly **integrated views** on the model

• Powerful **querying** tool (for analysis, validation & verification)
A power tool to engineer your business and technical requirements

Get your free trial version now!

www.objectiver.com