Design Science Methodology
MIKS

Winter 2016 – 2017

Prof. Dr. Roel Wieringa
0. Introduction
0.1 Goal of the course
Goal of the course

• Help you do your research projects (e.g. Master thesis)
  – Improve your capability to justify your solution
  – Help you structure your Master’s thesis

• Improves your problem-solving capability
  – But not a creativity course
Reality check

What kind of problems?

• Business Information Technology master thesis at the University of Twente:

• Computer Science master thesis at the University of Twente:

• Business Administration master thesis at the University of Twente:

• Master theses in human-media interaction
Two kinds of research problems

• (1) Design problems
  – Improve something, design something, how-to-do something
  – Problem, design of a treatment, validation of the treatment
  – Design cycle
  – Improvement is the goal, utility is the criterion
  – Knowledge is a side-effect
  – “Technical research problems”

• (2) Knowledge questions
  – Describe, explain, predict
  – Questions, research design, research execution, data, analysis
  – Empirical research cycle
  – Knowledge is the goal, truth is the criterion
  – Utility is a side-effect
Focus on justification

• This is not a creativity course
  – Not about how to be original

• The course is about how to **justify** and **report** your research results
  – Why would anyone use your design? There are many other designs.
  – Why would anyone believe your answers? Opinions are cheap.

• This also helps you to organize the project itself.
0.2 Organization of the course
Material

• Slides

Schedule

• Today
  – Course on design cycle
  – Questions and exercises during the day

• After today: Make outline the table of contents of your thesis

• 21\textsuperscript{st} February
  – Present your table of contents on a poster
  – Course on empirical research design
  – Finalize poster
Questions?
1 What is design science?
2.1 The subject of design science
• Design science is the **design** and **investigation** of artifacts in context
Reality check:
What is the artifact and what is the context?

- Business Information Technology master thesis at the University of Twente:
- Computer Science master thesis at the University of Twente:
- Business Administration master thesis at the University of Twente:
  - [http://essay.utwente.nl/view/programme/60644.html](http://essay.utwente.nl/view/programme/60644.html)
- Master theses in human-media interaction
Subject of design science

Artifact:
SW component/system, HW component/system, Organization, Business process, Service, Method, Technique, Conceptual structure, ...

Problem context:
SW components & systems, HW components & systems, Organizations, Business processes, Services, Methods, Techniques, Conceptual structures, People, Values, Desires, Fears, Goals, Norms, Budgets, ...

Interaction

Not designed by you or your colleagues

Something to be designed

Something to be influenced

MIKS 17 January 2017 © R.J. Wieringa
• Without a context, an artifact does nothing
What is designed and what is given

• The problem context is given to you
  – It is not designed by you
  – May be designed by others

• The (renewed) artifact is (re)designed by you
  – It is not given to you
  – An older version of the artifact may be given to you
Interaction should provide a service for the context

• The artifact interacts with the problem context ... in order to improve the context
• The interaction provides a service to the problem context
2.2 Research problems in design science
Research problems in design science

To design an artifact to improve a problem context

Problems & Artifacts to investigate
Knowledge, Design problems

To answer knowledge questions about the artifact in context

Artifact of a design problem = the artifact to be designed
Artifact of a knowledge question = the artifact about which we ask the knowledge question

---

**Design software to estimate Direction of Arrival of plane waves, to be used in satellite TV receivers in cars**

- Is the DoA estimation accurate enough in this context?
- Is it fast enough?

**Design a Multi-Agent Route Planning system to be used for aircraft taxi route planning**

- Is this routing algorithm deadlock-free on airports?
- How much delay does it produce?

**Design a data location regulation auditing method**

- Is the method usable and useful for consultants?
Heuristics

• Design problems

✓ Call for a change of the world
✓ Solution is design
✓ Many solutions
✓ Evaluated by utility
✓ Many degrees of utility
✓ What is useful depends on stakeholder goals

• Knowledge questions

✓ Ask for knowledge about the world
✓ Answer is a proposition
✓ One answer
✓ Evaluated by truth
✓ Many degrees of certainty about the answer
✓ What is considered “true” does not depend on stakeholder goals

http://www.factcheck.org
Reality check:
What is the artifact and what is the context?

• Business Information Technology master thesis at the University of Twente:
• Computer Science master thesis at the University of Twente:
• Business Administration master thesis at the University of Twente:
• Master theses in human-media interaction
Conclusions

• The title of your thesis is the shortest summary of your research project.
  – The best titles mention the artifact and the context.

• The top-level research problem of a thesis is either a design problem or a knowledge question
  – The motivation of the research may be both curiosity/fun, as well as utility
Exercise:
Ingredients for your thesis title

- What research problem(s) are you investigating?
  - Artifact and context
2.3 The social context of a design science project
The social context of design research

Social context design research project:
Location of stakeholders

Goals, budgets

Designs

Design science

Improvement design ➔ Answering knowledge questions

“Design a DoA estimation system to be used in cars”:
Stakeholders: Researchers, NXP (sponsor), component suppliers, car manufacturers, garages, car passengers

“Design an assurance method for cloud service provider data compliance”.
Stakeholders: KPMG (sponsor), KPMG consultants (end-users), researchers, CSPs, CPS clients.
2.4 The knowledge context of a design science project
The context of design research

Social context:
Location of stakeholders

Goals, budgets

Design science

Improvement design

Answering knowledge questions

Existing problem-solving knowledge, Old designs

New problem-solving knowledge, New designs

Existing answers to knowledge questions

New answers to knowledge questions

Knowledge context:
Mathematics, social science, natural science, design science, design specifications, useful facts, practical knowledge, common sense, other beliefs
Knowledge sources

• Scientific literature
  – Scientific, peer reviewed journals and conferences (math, natural science, social science, design sciences)

• Technical literature
  – Design specifications, manuals

• Professional literature
  – Non-peer reviewed professional magazines, trade press, marketing literature, white papers (useful facts and opinions, practical knowledge, common sense)

• Oral communication
  – Colleagues, supervisors, practitioners (useful facts and opinions, practical knowledge, common sense, other beliefs)
What about the Web?

• The Web is a communication channel, not a source of information

• Sources are more diverse
  – Scientific literature
  – Technical literature
  – Professional literature
  – On-line databases
  – Social networks

• Did the information survive
  – Empirical tests?
  – Critical judgment of peers?

• How is the channel managed?
• How does the source ensure quality of information?

• Fact check
• Logic check
Your research aims at theories

• Knowing the relevant properties of a particular artifact in a particular context is not enough
  – Theories should be general, so you can use them for prediction
  – Theories should explain, so that you understand why phenomena occur

• If the artifact prototype that you built disappears, what is the knowledge remains?
  – Tested, critiqued knowledge
Sciences of the middle range


generalization

universal generalization

existential generalization

basic sciences

physics, chemistry, parts of biology

special sciences (about the earth):
biology, psychology, sociology, ...

applied sciences:
astronomy, geology, meteorology, political sciences, management science, ...

design sciences:
software engineering, information systems,
computer sciences, electrical engineering,
mechanical engineering, ...

case research:
engineering, consultancy,
psychotherapy, health care,
management, politics, ...

idealized conditions realistic conditions conditions of practice

realism
• Useful idealizations in software engineering and information systems
  – All clocks are synchronized and correct
  – Synchronicity of response and stimulus
  – Unlimited memory (Turing machines)
  – Message arrival guarantees
  – Rational users
  – Organizations with a clearly defined structure
  – ...

• Conditions of practice
  – Incorrect input
  – Messages get lost
  – Timeouts are discovered too late
  – Clocks drift
  – Users do not behave according to expectations
  – ...
Scaling up

- We will never scale up to the upper right corner
- But try to get as far as possible

Stable regularities

Population

Samples

Single case

Idealized conditions

Realistic conditions

Conditions of practice

Robust mechanisms
Main points chapter 1
What is design science

• Design science is the **design** and **investigation** of **artifacts** in **context**
  – Research problems are **design problems** or **knowledge questions**
  – Artifacts **interact** with their context to deliver a **service**

• The **social context** of a design science project consists a.o. of stakeholders and their goals and budgets, laws, processes, norms, expectations, etc.

• The **knowledge context** consists of scientific knowledge, design specifications, useful facts, practical knowledge, common sense, etc. You aim to contribute scientific theories.
  – Sources and channels of information

• The design sciences are **middle-range sciences** aiming for partial generalizations about realistic conditions.
  – Need to scale up from idealized to practical conditions
  – Universal generalizations make unrealistic assumptions
Exercise:
Material for your elevator pitch

1. What design(s) will be delivered by your project?
   – What is new?

2. Who are the stakeholders of your project?
   – What are their goals?

3. What knowledge will be produced by your project?
   – What is new?
2. Research Goals and Research Questions
2.1 Research goals
External goals

Social context:
- Stakeholders,
- **Goals that are external to design research**
- Budgets,
- Application scenarios

Design research

- Design an artifact to improve a problem context
- Answer knowledge questions

Goals, budgets → Designs
Goal structure

**Social context**

External goals

To improve a problem context

**Contribution**

**Design research**

To (re)design an artifact

To answer knowledge questions

To (re)design a research instrument

**Motivation** of the research goal: friends, family, the government, sponsors, investors, etc. are interested in these.

A design **research goal** is the desired outcome of a research project, to which the research budget is allocated. Colleagues are interested in these.
Examples

Ucare

• **External goals:**
  – Reduce health care cost (government)
  – Reduce work pressure, increase quality of care (health personnel)
  – Increase quality of care, increase independence (elderly)

• **Design goals**
  – Design a mobile home care system for use by elderly that provides
    • Medicine dispensing
    • Blood pressure monitoring
    • Agenda
    • Remote medical advice
Two kinds of design research problems

- To achieve the design goal, we need to answer research questions.
  - Design problems
    - A.k.a. technical research questions
  - Knowledge questions
    - Analytical research questions: can be answered by analysis
    - Empirical research questions: must be answered by collecting data
2.2 Design problems
Template for design problems

• Improve <problem context>
• by <treating it with a (re)designed artifact>
• such that <artifact requirements>
• in order to <stakeholder goals>

• Improve my body / mind health
• by taking a medicine
• such that my headache disappears
• in order for me to get back to work
Template for design problems

- Improve *<problem context>*
  - by *<treating it with a (re)designed artifact>*
  - such that *<artifact requirements>*
  - in order to *<stakeholder goals>*

- *Improve my body / mind health*
  - *by taking a medicine*
  - *such that my headache disappears*
  - *in order for me to get back to work*

External: Problem context and stakeholder goals
Template for design problems

- Improve <problem context>
- by <treating it with a (re)designed artifact>
- such that <artifact requirements>
- in order to <stakeholder goals>

- Improve my body / mind health
  - by taking a medicine
  - such that my headache disappears
  - in order for me to get back to work

Design research problem: Artifact and its desired interactions
Template for design problems

- Improve <problem context>
- by <treating it with a (re)designed artifact>
- such that <artifact requirements>
- in order to <stakeholder goals>

- Improve my body / mind health
  - by taking a medicine
  - such that my headache disappears
  - in order for me to get back to work

  Particular problem

- Improve home care
  - By a mobile support device
  - That provides some services ...
  - So that cost are reduced etc.

  General problem
2.3 Knowledge questions
Kinds of empirical knowledge questions

• Empirical knowledge questions may be
  – descriptive or explanatory,
  – open or closed,
  – effect-related or requirement-related
Knowledge questions

• **Descriptive questions:**
  – What happened?
  – When?
  – Where?
  – What components were involved?
  – Who was involved?
  – etc.

• **Explanatory questions:**
  – Why?
    1. What has *caused* the phenomena?
    2. Which *mechanisms* produced the phenomena?
    3. For what *reasons* did people do this?
Example

• **Descriptive question:** What is the performance of the Ucare system?
  – Accuracy of output
  – Reliability of communication infrastructure
  – Usability of interfaces
  – Etc. etc.

• **Explanatory question:** Why does Ucare have this performance?
  1. **Cause:** data entrance at 03:00 causes the data to be lost
  2. **Mechanism:** because the hospital database server is down for mainenance at night and there is no fallback retention mechanism
  3. **Reasons:** Users feel free to enter data any time they are awake, and they are awake at 03:00.
Prediction problems

• There are no predictive knowledge questions
  – We cannot know the future
  – Descriptive and explanatory questions are about the present and the past

• But there are prediction problems
  – How will the program behave when given this input?
  – How would users behave when the program is changed?

• To solve a prediction problem, we need a general theory that tells us what happens
Second classification of knowledge questions

• **Open questions (exploration):**
  – No hypothesis about the answers.
    • *What is the execution time?*

• **Closed questions (testing):**
  – Specific, testable hypotheses as possible answers.
    • *Is execution time less than 1 second?*
      – *Hypothesis: the execution time is less than 1 second.*
Third classification: Design research questions

• **Effect question**: Context X Artifact → Which Effects?
  – **Trade-off question**: Context X *Alternative artifact* → Effects?
  – **Sensitivity question**: *Other context* X artifact → Effects?

• **Requirements satisfaction question**: Do these Effects satisfy requirements sufficiently?
Example

• **Open descriptive effect questions: What is the performance of the Ucare system?**
  – Accuracy of output
  – Reliability of communication infrastructure
  – Usability of interfaces
  – Etc. etc.

• **Open descriptive trade-off questions**
  – What happens to the performance if we change the design?

• **Open descriptive sensitivity questions:**
  – What happens if it is used by other elderly, in other homes?

• **Open explanatory questions:**
  – Why does Ucare have this performance?

• **Open descriptive requirements satisfaction questions:**
  – Does this satisfy our requirements?
Main points chapter 2
Research goals & questions

• A design science project has goals that range from designing an instrument (lowest level) to contribution to external stakeholder goals (highest level).

• Design problems have the form
  – Improve <problem context> by <treating it with a (re)designed artifact> such that <artifact requirements> in order to <stakeholder goals>

• Knowledge questions may be analytical or empirical.
  – Empirical knowledge questions may be
    • descriptive or explanatory,
    • open or closed,
    • effect-related or requirement-related

• To answer prediction problems, we need general theories
Questions about chapter 2?
Exercise: your top-level design problem

• What is/are your top-level design problem(s), using our template?
  – Improve <problem context>
  – by <treating it with a (re)designed artifact>
  – such that <artifact requirements>
  – in order to <stakeholder goals>

• For a knowledge-oriented thesis, think of a top-level design problem that motivates your knowledge question
Research questions

• Research questions form a hierarchy
  – Some questions are knowledge questions, others are design problems
  – All are subproblems of the top-level research problem

• Business Information Technology master thesis at the University of Twente: [http://essay.utwente.nl/view/programme/60025.html](http://essay.utwente.nl/view/programme/60025.html)
• Computer Science master thesis at the University of Twente: [http://essay.utwente.nl/view/programme/60300.html](http://essay.utwente.nl/view/programme/60300.html)
• Business Administration master thesis at the University of Twente: [http://essay.utwente.nl/view/programme/60644.html](http://essay.utwente.nl/view/programme/60644.html)
Exercise:

your research questions

• Formulate the subproblems of your top-level research problem
3 The design cycle
Activities in design science

Improvement design

Engineering cycle

Problems to be investigated, artifacts to be investigated

Answering knowledge questions

Knowledge

Research cycle
3.1 The design and engineering cycles
Engineering cycle

Treatment implementation

- Stakeholders? Goals?
- Conceptual problem framework?
- Phenomena? Causes, mechanisms, reasons?
- Effects? Positive/negative goal contribution?
- Specify requirements!
- Requirements contribute to goals?
- Available treatments?
- Design new ones!

Implementation evaluation = Problem investigation

Treatment validation

- Context & Artifact → Effects?
- Effects satisfy Requirements?
- Trade-offs for different artifacts?
- Sensitivity for different Contexts?

Treatment design
Treatment

• We avoid the word “solution”.
  – Every solution is imperfect
  – ... and introduces new problems
Specification and design

• Treatments are designed, and the design is specified

• **Designing** is deciding what to do
• **Specifying** is documenting that decision

• Contrast with the terminology in software engineering
  – Word games with “what” and “how”.

MIKS 17 January 2017 © R.J. Wieringa

67
What is implementation?

• Depends on who you talk to
  – For a software engineer, this is writing and debugging a program until it works.
  – For a mechanical engineer, this is assembling the physical machine until it works
  – For the manager, this is introducing the machine in the organization until it works
  – For a marketeer, this is selling the system
Implementation

• **Implementation** = introducing an artifact in the intended problem context
  – What this means depends on what your problem was
  – *For a software engineer: To construct software*
  – *For a mechanical engineer: To construct physical machine*
  – *For the manager: To change an organization*
  – *For a marketeer: To sell a product*

• In this course, our problems are real-world problems
  – Implementation = transfer to the problem context
  – = technology transfer to the real world
Design cycle

Real-world implementation evaluation =

Real-world problem investigation

• Stakeholders? Goals?
• Conceptual problem framework?
• Phenomena? Causes, mechanisms, reasons?
• Effects? Positive/negative goal contribution?

Treatment design

• Specify requirements!
• Requirements contribute to goals?
• Available treatments?
• Design new ones!

Treatment validation

• Context & Artifact → Effects?
• Effects satisfy Requirements?
• Trade-offs for different artifacts?
• Sensitivity for different Contexts?

Real-world treatment implementation:
Technology transfer
This is a very special engineering cycle. Later we will call this the empirical cycle. It is performed to answer empirical knowledge questions.
Validation versus evaluation

• To **validate** a design for stakeholders is to justify that it would contribute to their goals **before** transfer to practice
  – Predicted effects?
  – Satisfaction of requirements?
  – (Requirements contribute to goals?)

• To **evaluate** an implementation is to investigate whether an implementation has contributed to to stakeholder goals **after** transfer to practice
  – Stakeholders, goals?
  – Effects?
  – Contribution?
What is the difference?

• **Implementation valuation** research studies real-world implementations with respect to actual stakeholder goals
  – Real-world research

• **Treatment validation** research uses a validation model to predict effects
  – Simulation
What kind of project do you have?

• Some projects do implementation evaluation
  — *E.g. investigate how UML is used in practice*
  — *Investigate traffic flow on internet*
  — *Investigate why our project effort estimations are always so wrong*

• Many projects design and validate treatments
  — *E.g. improve malware detection methods to get higher accuracy*
  — *Explore the use of social networks to communicate with our customers*

This determines the kind of research questions that you can ask.
3.2 Design and engineering processes
• The design and engineering cycles are **rational reconstructions** of design and engineering
  – Rational reconstruction of mathematical proofs
  – Of empirical research
  – Of administrative processes

• The design and engineering **processes** execute tasks in different orders
  – Resources (time, money, people) must be managed
  – Deliverables must be scheduled, deadlines must be met
Concurrent engineering

- Development may be organized concurrently with successive versions of the artifact
Systems engineering

• Cycles of systems engineering
  – High level goals, high level requirements
  – Iterative refinement until
  – Low-level approved interfaces, low-level implemented specs.

• Shown on next slide
• Iteratively reduce uncertainty about the problem
• Once the goals are clear enough, reduce risk of choosing the wrong treatment
Two kinds of design decisions

- Adding components
- Adding information about a component
- Architectural decomposition

Magic square
- A development process is a path through the square
- Commutative
Engineering management

• Management is the art of achieving results by the work of others.
  – Acquiring resources
  – Organizing them
  – Planning work
  – Managing risks
  – Motivating people
  – Evaluating outcomes

Systems engineering is a particular way to plan work & manage risks
Main points chapter 3
The design cycle

• The engineering cycle is a rational decision cycle:
  – Problem/evaluation: Look where you are and what you want to do;
  – Design possible treatments;
  – Validate treatments without executing them;
  – Choose one and implement it;
  – Evaluation/problem: Look where you are now and what you now want to do.

• The design cycle is the preparation for action:
  – Problem-design-validation.

• The cycles can be organized in many different ways.
  – All of them must allow you to justify your choices afterwards.
  – The engineering cycle allows you to justify your actions (validation) and to learn from their effects (evaluation)
Questions about chapter 3?
Exercise (design-driven thesis) your table of contents

• Make a poster with the outline of the table of contents of your thesis, following this pattern:
  1. Introduction: Societal improvement problem, stakeholders and their goals, current designs, gap with improvement needs.
  2. Research problem: top-level design problem; decomposition into subproblems and knowledge questions
  3. Research methodology
  4. State of the art: existing designs
  5. Requirements for a new design; motivation in terms of stakeholder goals; evaluation of current designs against the requirements
  6. New design
  7. Validation of new design: prototypes, simulations, field experiments, etc.
  8. (More designs and validations)
  9. Conclusions, recommendations, and further work
Exercise (knowledge-driven thesis): your table of contents

- Make a poster with the outline of the table of contents of your thesis, following this pattern:
  1. Introduction: Societal improvement problem, stakeholders and their goals, current knowledge, gap with desired knowledge.
  2. Research problem: Top-level knowledge question; decomposition into sub-questions
  3. State of the knowledge: existing knowledge
  4. Research methodology
  5. Study: observational study, experimental, case-based, sample-based, etc.
  6. (More studies)
  7. Conclusions, recommendations, and further work
4. Stakeholder and Goal Analysis
Engineering cycle

! = Action
? = Knowledge question

**Treatment implementation**

- Stakeholders? Goals?
- Conceptual problem framework?
- Phenomena? Causes, mechanisms, reasons?
- Effects? Positive/negative goal contribution?

**Implementation evaluation = Problem investigation**

- Specify requirements!
- Requirements contribute to goals?
- Available treatments?
- Design new ones!

**Treatment validation**

- Context & Artifact → Effects?
- Effects satisfy Requirements?
- Trade-offs for different artifacts?
- Sensitivity for different Contexts?
4.1 Stakeholders
Stakeholders

• A stakeholder of a problem is a biological or legal person affected by treating a problem.
  – People, organizations, job roles, contractual roles, etc.

• Typical stakeholders of a design research project
  – Researchers, sponsors, developers, users, etc.
  – They have an interest in the outcome.

• Typical stakeholders of a development project
  – Designers, programmers, testers, users etc.

• Typical stakeholders of a software product
  – See next slides
The organization may be a company, government organization, department, project, etc.
Checklist by role (Ian Alexander http://www.scenarioplus.org.uk/papers/papers.htm > A taxonomy of stakeholders)

System under Development
- Normal operator (end user)
- Operational support
- Maintenance operator

Immediate context
- Functional beneficiary (client)
- Roles responsible for interfacing systems

Wider context
- Political beneficiary (who gains status)
- Financial beneficiary
- Negative stakeholder (who is/perceives to be hurt by the product)
- Threat agent (who wants to hurt the product)
- Regulator

Involved in development
- Champion/Sponsor
- Developer
- Consultant
- Purchaser (customer)
- Suppliers of components

None of these lists is complete
Examples of stakeholders

• **PISA:** Design a system to help individuals to maintain their privacy on the internet at a desired level
  – Free lancer
  – Teleworker
  – Home banker
  – Concerned parent

• **Ucare:** Design a system that provides health care support for elderly people at home
  – Medicine taking
  – Blood pressure monitoring
  – Agenda
  – Remote advice

• We omit researcher goals henceforth
4.2 Desires
Stakeholder awareness and commitment

Not aware: Some possibility that stakeholders are not aware of

Aware, not committed: Indifferences, Desires, Fears
Not interested enough to commit resources (money, time)

Aware & Committed: Goals
Resources committed to act for a goal

An event pushes the possibility into awareness

• Possibility to receive satellite TV in car
• Possibility to reduce taxiing time

Stakeholder makes resources (time, money) available

• We could upgrade car DVD player to TV
• We could optimize taxi routes dynamically
• Invest in car satellite TV
• Develop a prototype multi-agent route planning system
• A **goal** of a stakeholder is a desire to the realization of which the stakeholder has committed resources (time, money)
  – People want a lot but they have only a few goals
  – Some goals are imposed
Anything can be the object of desire, fear or indifference

Desires, fears and indifference are mental states:
- They can be **directed upon** anything, whether real or imaginary
- Every mental state is **about** something
- They can even be about desire, fear or indifference
**Problem context**

- SW components & systems,
- HW components & systems,
- People,
- Organizations,
- Business processes,
- Services,
- Methods, Techniques,
- Conceptual structures,
- Values, Desires, Fears,
- Indifferences, Goals, Norms,
- Resources, ...
Examples of problem contexts

• **Ucare**: Design a system that provides health care support for elderly people at home.
  – Context: Patient’s home
    • Patient and their physical and technical context, budget, desires, norms and values
    • Friends and their budget, desires, norms and values
    • Family and their budget, desires, norms and values
    • Home care nurses and their budget, desires, norms and values
    • Remote medical personnel and their budget, desires, norms and values
    • The law
    • Ethical constraints
4.3 Desires and conflicts
The multitude of desires

• Any one stakeholder may have infinitely many potential desires, fears and indifferences

• Many desires of one or more stakeholders may conflict
Conflicting desires

• **Logical conflict:**
  – Analysis of the descriptions of the desires shows that both descriptions have opposite meaning; they are logically inconsistent.
  – *Spend your money and keep it*

• **Physical conflict:**
  – Realization of one desire makes realization of the other physically impossible.
  – *Eat more and stay the same weight*
  – *Add TV to a car and reduce weight without changing anything else*
  – Stakeholder lives in a phantasy world
• **Technical conflict:**
  – There is currently no technology to realize both desires in the same artifact.
  – *Secure and user-friendly system*
  – New technology may remove the conflict

• **Economic conflict:**
  – Desires exceed the budget

• **Legal conflict:**
  – Desires contradict the law

• **Moral conflict:**
  – Desires contradict moral norms
Examples of conflicting desires

• Ucare: Design a system that provides health care support for elderly people at home
  – Technical conflict: Artifact should be simple to use, but is fragile & advanced technology.
  – Economic conflict: Artifact should be cheap, but is expensive
  – Value conflict: patient likes Skyping more than the advice functions

• Conflicts give us relevant design goals.
Discussing questions 4 of ch 2 and 1 of ch 3

• ..\Q&A\Questions and Assignments.pdf
Main points chapter 4
Stakeholder and goal analysis

• **A stakeholder** of a problem is a biological or legal person affected by treating a problem
  – Positively or negatively affected
  – There are checklists of possible stakeholders

• **A goal** of a stakeholder is a *desire* to the realization of which the stakeholder has *committed* resources (time, money)
  – Desires are many, goals are few

• Desires may **conflict** with each other
  – Therefore, goals of one or more stakeholders may conflict too.
  – Logical, physical, technical, economic, legal, moral conflict
Exercise

- Make a list of stakeholders of your thesis project.
- What are the goals of each stakeholder?
5 Implementation Evaluation and Problem Investigation
Engineering cycle

! = Action
? = Knowledge question

Treatment implementation

Implementation evaluation = Problem investigation

- Stakeholders? Goals?
- Conceptual problem framework?
- Phenomena? Causes, mechanisms, reasons?
- Effects? Positive/negative goal contribution?

Treatment validation

- Context & Artifact → Effects?
- Effects satisfy Requirements?
- Trade-offs for different artifacts?
- Sensitivity for different Contexts?

Treatment design

- Specify requirements!
- Requirements contribute to goals?
- Available treatments?
- Design new ones!
5.1 Research goals
Two alternative top-level goals of real-world research

- **Implementation evaluation** is the investigation of the effects of a treatment implementation after the improvement has been implemented.

- **Problem investigation** is the investigation of the problem context before an improvement is undertaken.

- There is always a current implementation of *something*!
  - So the research questions are the same, only the goals are different.
Examples

• Implementation evaluation
  – Investigate the use of the UML in companies in Brazil. Our goal is to find out the extent of usage.
  – Investigate the sources of phishing messages received by our organization. Our goal is to find out how bad it is.

• Problem investigation
  – Investigate the causes why our effort estimations are usually wrong. Our goal is to find improvement opportunities.
  – Investigate coordination problems in global software engineering projects. Our goal is to reduce these problems.
Research questions for implementation evaluation & problem investigation

• Effect questions
  – Descriptive: What effects does the implemented artifact have?
  – Explanatory: Why do these effects arise? (causes, mechanisms, reasons)

• Goal contribution questions
  – Evaluative: Do they contribute to/detract from stakeholder goals? To which extent?
  – Explanatory: why does this happen? (causes, mechanisms, reasons)
5.2 Theories
Scientific theories

• A scientific theory is a belief about patterns in phenomena that has
  – been validated against experience
  – survived criticism by critical peers

• Examples
  – *Theory of classical mechanics*
  – *Theory of evolution*
  – *Theory of cognitive dissonance*

• Non-examples
  – *Theory that the gods were astronauts*
  – *Conspiracy theories about who killed president Kennedy*
  – *The belief that my thoughts are monitored by aliens*
Problem theories

• Scientific theory of a problem
  – beliefs about problem patterns that have been validated against experience and survived critical analysis by peers

• Ucare project: Design a system that provides health care support for elderly people at home.

• Problem theory:
  – People stay home till a higher age than previously
  – Travelling to health care centers is unpleasant
  – Health care personnel is expensive and is overburdened
  – Health care budgets grow at unsustainable rate
  – ...
Satellite TV reception system for a car, contains an antenna array. Problem to be solved by a software system: recognize direction of arrival of plane waves.

Problem theory:

- Definitions of concepts: Plane waves, wave length, bandwidth, etc.

- Generalization about the problem: \( \varphi = 2\pi \frac{d}{\lambda} \sin \theta \)
5.3 Research Methods
The goal of empirical research is to develop, test, refine change, or otherwise update scientific theories.
# Kinds of empirical research methods

<table>
<thead>
<tr>
<th></th>
<th>Experimental study (treatment)</th>
<th>Observational study (no treatment)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sample-based:</strong> investigate samples drawn from a population, look at averages and variation, infer population parameters</td>
<td>• Statistical difference-making experiment</td>
<td>Survey</td>
</tr>
</tbody>
</table>
| **Case-based:** investigate cases one by one, observe case architecture and at interaction mechanisms among components | • Expert opinion  
• Mechanism experiments  
• Technical action research | Observational case study |

- The methods in **bold** are useful for Problem research
The empirical research setup

You

The instruments that you need to provide input to the OoS and to collect data

The laboratory simulations or field cases that you want to study

All problems similar to the one you want to treat
Survey research

- **Surveys** of instances of the problem (large sample)
  - *Survey of the use of role-based access control in large companies*
  - *Survey of the use of agile development methods in small and medium-sized companies*

- Useful to describe statistical regularities (descriptive statistics, mean, variance, correlations) in classes of problems.
Observational case studies

- **Observational case study** of instances of an implementation or problem:
  - *Case study of problems with effort estimation of project managers in one company*
  - *Field study of the behavior of elderly at home*

- Useful to describe implementations and problems in detail, and understand the mechanics and reasons behind their effects.
In a **single-case mechanism experiment**, we test a social or technical system

- *Observing elderly at home*
- *Penetration-testing the security of existing systems*

Useful to describe the behavior of implemented technology, and to understand this in terms of underlying mechanisms
In statistical difference-making experiments, we investigate whether in a sample, a difference in an independent variable X makes a statistical difference to a dependent variable Y.

- Apply several input scenarios to a company network and compare average behavior in scenarios with and without these inputs
- Treatment group/control group experiment with software engineers to test their comprehension of UML diagrams
Main points chapter 5
Implementation evaluation & problem investigation

• Implementation evaluation and problem investigation have different research goals but the same research questions.
  – Who are the stakeholders? What are their goals?
  – What conceptual framework shall we use to describe the phenomena?
  – What are the phenomena? Their causes, mechanisms, reasons?
  – What if we do nothing? How good/bad wrt goals?

• Useful research methods are
  – surveys,
  – observational case studies,
  – single-case mechanism experiments and
  – statistical difference-making experiments
Assignment chapter 5

• Drenthen (2014) - *Towards continuous delivery in system integration projects*
  – Artifact is a continuous delivery method using an automated test tool.
  – Context is the delivery of identity solutions by Everett.

• Schoutsen (2012) - *Fraud detection within Medicaid*
  – Artifact: data warehouse
  – Context: fraud detection within Medicaid

• Van der Graaf (2012) - *EPR in Dutch hospitals-a decade of changes*
  – Artifact: EPRs
  – Context: Dutch hospitals

• [Page 15 in Q&A](#)
Exercise

• What concepts do you need to describe your problem domain?
• What problematic phenomena are happening in the problem domain? Why is this happening? (Causes, reasons, and mechanisms behind these phenomena)
• What happens if nothing changes? How does this contribute (positively or negatively) to the stakeholder goals?
Discuss these questions

- Chapter 4 2(c)
- Chapter 5 questions 6, 7
6. Requirements Specification
### Engineering cycle

! = Action  
? = Knowledge question

#### Treatment implementation

- Stakeholders? Goals?  
- Conceptual problem framework?  
- Phenomena? Causes, mechanisms, reasons?  
- Effects? Positive/negative goal contribution?

#### Implementation evaluation = Problem investigation

- Specify requirements!  
- Requirements contribute to goals?

#### Treatment validation

- Context & Artifact → Effects?  
- Effects satisfy Requirements?  
- Trade-offs for different artifacts?  
- Sensitivity for different Contexts?

#### Treatment design

- Available treatments?  
- Design new ones!
6.1 Requirements
• **Requirements** are desired properties of the treatment
  – Stakeholder goals are what the stakeholder wants to achieve
  – Requirements are what the developer must achieve
    • Special kind of goal

• Sometimes, **constraints** on the internal composition of the artifact are distinguished from requirements on the externally observable properties of an artifact.
  – *E.g. a constraint to reuse some components*
• Requirements cannot be just “elicited” from stakeholders
  – We do not know what we want
• Research projects may have very vague requirements
  – See if you can do this (existence proof)
  – See if you can do this better (e.g. better execution time)
6.2 Contribution arguments
**Assumptions, requirements, goals**

**Assumptions C about the context**
- Should satisfy

**External stakeholder goals G**
- Should contribute to

**Artifact requirements R**
- Should satisfy

**Problem context**

**Interaction X**

**Artifact**

**Contribution argument**

- \((\text{Context assumptions C}) \text{ AND } (\text{Requirements R}) \implies (\text{contribution to stakeholder goal G})\)
Example

• **Ucare contribution argument**
  – *(assumptions about patient behavior & desires, IT infrastructure of home for the elderly, national communication infrastructure, third-party services)* AND
  – *(requirements on mobile health care support technology)* IMPLY
  – *(reduce health care cost, improved health service)*

• **We need to evaluate systems after transfer to practice to see if this argument is correct!**
6.3 Kinds of requirements
Classifications of requirements

- By stakeholder (Who wants it? Whose goals are served by it?)
- By priority (How strong is the desire?)
- By urgency (How soon must it be available?)
- By aspect (What is the requirement about? Which property?)
Requirements by aspect (ISO 9126)

• A **function** is a terminating part of the interaction that provides a service to some stakeholder

• **Quality properties** (a.k.a. “nonfunctional properties”)
  – Utility (“suitability”)
  – Accuracy
  – Interoperability
  – Security
  – Compliance
  – Reliability
  – Usability
  – Efficiency (time or space)
  – Maintainability
  – Portability

  • These are properties of functions
  • They usually have global implications for artifact components and architecture
Example

- **Ucare**
  - **Functions**
    - Medicine dispensing
    - Blood pressure monitoring
    - Agenda
    - Remote medical advice
  - **Quality:**
    - Usable by elderly and medical personnel
    - Reliable
    - Safe
    - Cheap

Classify this:
- By stakeholder
- By priority
- By urgency
6.3 Indicators and norms
Operationalization

• Some properties cannot be measured directly
  – *Usability, maintainability, security, ...*

• **Operationalize** them:
  – Define them in terms of one or more indicators that *can* be measured

• An **indicator** is a variable that can be measured
  – In software engineering, often called a *metric.*
Some examples of indicators

- **Utility indicator**: Opinion of stakeholder about utility
- **Accuracy indicator**: domain dependent, e.g. spatial resolution
- **Interoperability indicator**: effort to realize interface with a system
- **Security indicators**: availability, compliance to standards
- **Compliance indicator**: expert opinion about compliance
- **Reliability indicators**: mean time between failure, time to recover
- **Usability indicators**: effort to learn, effort to use
- **Efficiency (time or space) indicators**: execution time, disk usage
- **Maintainability indicators**: effort to find bugs, effort to repair, effort to test
- **Portability indicators**: effort to adapt to new environment, effort to install, conformance to standards

Norms

• Once we have defined indicators (“metrics”), we can operationalize requirements by means of norms

• A norm is a desired range of values of an indicator
  – Average effort to learn (indicator) is less that 30 minutes (norm)
  – Accuracy (indicator) is better than 1 degree (norm)
  – Function F (indicator) must be present (norm)
    • When it is time to dispense a medicine, the dispenser sends an alert to the ipad
    • If dispensing button is pushed, the dispenser releases medicine according to protocol defined for the patient
Informally stated requirements may be operationalized into a set of indicator/norm pairs.
Main points chapter 6
Requirements specification

• **Requirements** are desired properties of a treatment for which there is a stakeholder budget

• Must be motivated by **contribution argument**
  – (context assumptions) X (artifact requirements) contribute to (Stakeholder goals)

• Requirements can be classified according to **stakeholder goal, priority, urgency**

• **Functional requirements** are desired functions

• **Nonfunctional requirements** (quality properties)
  – Accuracy, efficiency, security, reliability, usability, ...

• Requirements may have to be operationalized
  – **Indicator** is measurable variable: measurable property
  – **Norm** is desired range of values of an indicator: measurable requirement
Exercise

• What are the requirements for a solution to your design problem?
• Classify the requirements
  – By stakeholder
  – By priority
  – By urgency
  – By aspect
7 Treatment Validation
Engineering cycle

Treatment implementation

Implementation evaluation = Problem investigation

- Stakeholders? Goals?
- Conceptual problem framework?
- Phenomena?
  - Causes, mechanisms, reasons?
  - Effects?
  - Positive/negative goal contribution?

Treatment validation

- Context & Artifact → Effects? Why?
- Trade-offs for different artifacts? Why?
- Sensitivity for different contexts? Why?
- Effects satisfy Requirements? Why?

Treatment design

- Specify requirements!
- Requirements contribute to goals?
- Available treatments?
- Design new ones!
7.1 The validation research goal
• Validation research questions are the same as implementation evaluation questions
  – But the goal is to validate new technology,
  – Not to evaluate implemented technology

• We find the validation research questions by analyzing treatment requirements (next slide)
Engineering cycle

Treatment implementation

Implementation evaluation = Problem investigation

- Stakeholders? Goals?
- Conceptual problem framework?
- Phenomena?
  - Causes, mechanisms, reasons?
  - Effects?
  - Positive/negative goal contribution?

Treatment validation

- Context & Artifact $\rightarrow$ Effects? Why?
- Trade-offs for different artifacts? Why?
- Sensitivity for different Contexts? Why?
- Effects satisfy Requirements? Why?

Treatment design

- Specify requirements!
- Requirements contribute to goals?
- Available treatments?
- Design new ones!
• **Ucare requirements**
  – **Functions**
    • Medicine dispensing
    • Blood pressure monitoring
    • Agenda
    • Remote medical advice
  – Usable by elderly and medical personnel
  – Reliable
  – Safe
  – Cheap

**Validation research questions**

• Does it work?
  – Functions
    • Does it perform the medicine dispensing functions?
    • Does it perform the blood pressure monitoring functions?
    • Etc.
  – Is it usable by elderly and medical personnel?
  – Is it reliable?
  – Is it safe?
  – Is it cheap?

• What if we change the design?
• What if we vary the context?

• To get answerable research questions, we need to operationalize the requirements!
7.2 Validation models
The fundamental problem of validation

• We investigate the artifact outside its natural implementation context

• The artifact has not been implemented yet.
  – It has not been transferred to the real-world problem context yet

• So we study it in the lab

• Or we do a pilot study in the real world

These are more or less realistic models of a real-world implementation
Validation models

**Model of the artifact**

**Model of problem context** (systems, stakeholders)

 Representation

**Artifact**

**Problem context** (systems, stakeholders)

Fig 7.2
What is a model?

• An **analogic model** is an entity that represents entities of interest, called its **targets**,  
• in such a way that questions about the target can be answered by studying the model.  
• Examples
Example validation models

• A software prototype interacting with a simulated environment
• A class of students using a new software engineering method in a project that simulates a real-world project
• A researcher using an experimental method to solve a real-world problem
• Ucare
  – Nurses imagining how the system would function
  – Elderly using a prototype in their home
Similarity

• How reliable is the generalization from the validation models to the real-world implementations?

• Positive analogy: Properties known to be similar
  – Should support transfer of conclusions about the model to conclusions about the target

• Negative analogy: Properties known to be different
  – Should block the transfer of some conclusions
7.3 Design theories
Design theories

• Design theory = a belief that there is a pattern in the interaction between the artifact and the context, tested by experiment, critically analyzed by peers

• Design theory of the Ucare system, developed based on field tests:
  – The system helps elderly take their medicine, but not necessarily on time
  – Elderly may not use the Ucare functions but love to use the Skype function of the ipad
  – To provide reliable service, service providers must align the details of their interfaces as well as their maintenance procedures
7.4 Research methods
Prior beliefs:
- Theories
- Specifications
- Experiences
- Lessons learned

Knowledge questions

Empirical research

Posterior beliefs:
Updated
- Theories,
- Specifications,
- Etc.
# Kinds of empirical research methods

<table>
<thead>
<tr>
<th>Sample-based: investigate samples drawn from a population, look at averages and variation, infer population parameters</th>
<th>Experimental study (treatment)</th>
<th>Observational study (no treatment)</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Statistical difference-making experiment</td>
<td>Survey</td>
<td></td>
</tr>
</tbody>
</table>

| Case-based: investigate cases one by one, observe case architecture and at interaction mechanisms among components | Experimental opinion, Mechanism experiments, Technical action research | Observational case study |

- The methods in **bold** are useful for validation research
Expert opinion
Expert opinion

- Researcher asks experts about perceived usability and utility of new artifact in the contexts that they know first-hand.
  - *Expert opinion of nurses about U-Care functionality*
- Purpose is to weed out unrealistic ideas.
Single-case mechanism experiments

(a.k.a. simulations)
In a **single-case mechanism experiment**, we test a social or technical artifact

- *Testing a software prototype of Ucare using your colleagues*
- *Testing it with volunteers in a home for the elderly*

**Useful to validate new technology**
Technical action research
Technical Action Research

- In Technical Action research, we test a social or technical artifact by using it for a real-world problem
  - *Experimental use of a new enterprise architecture method in a consultancy with a real-world client*
- Useful to validate new technology
Statistical difference-making experiments
In statistical difference-making experiments, we investigate whether in a sample, a difference in an independent variable $X$ makes a statistical difference to a dependent variable $Y$.

– Compare a new software engineering technique with an existing one in an experiment with two groups of students

– Compare a new algorithm with an existing one by exposing them to a set of contexts to which they are randomly allocated
7.4 Scaling up
Scaling up

Fig. 7.3

Stable regularities

Population

Samples

Single case

Statistical difference-making experiments

Single-case mechanism experiments

Expert opinion, Technical action research

Idealized conditions

Realistic conditions

Conditions of practice

Robust mechanisms
Main points chapter 7
Treatment validation

• Validation is a prediction problem
  – What would be the effect of artifact in context?
  – Trade-offs in design of artifact?
  – Sensitivity to changes in context?
  – Satisfaction of requirements?

• Use validation models to build a design theory of A x C;

• Then use design theory to do predictions

• Research methods
  – Expert opinion
  – Single-case mechanism experiments
  – Statistical difference-making experiments
  – Technical action research

• Scale up from idealized to practical conditions
Exercise

• What artifact needs to be designed to treat your design problem?
• What are the validation research questions for this artifact?
  – Effect, trade-off, sensitivity, requirements satisfaction questions
• How will you investigate these questions?
  – Assume that you have enough time and money to do all research needed
Assignment for 21 February

• Make a poster for your research project
  – The context, the problem to be solved
  – Your research goal
  – Top-level design problem (following the template)
  – Subproblems and knowledge questions

• Table of contents