



Initiation of the architecture development project

IT-architecture and user driven software design (BUITA)

SDD1

18. September 2018

Magnus R. P. Hansen



The architecture description



Architecture models

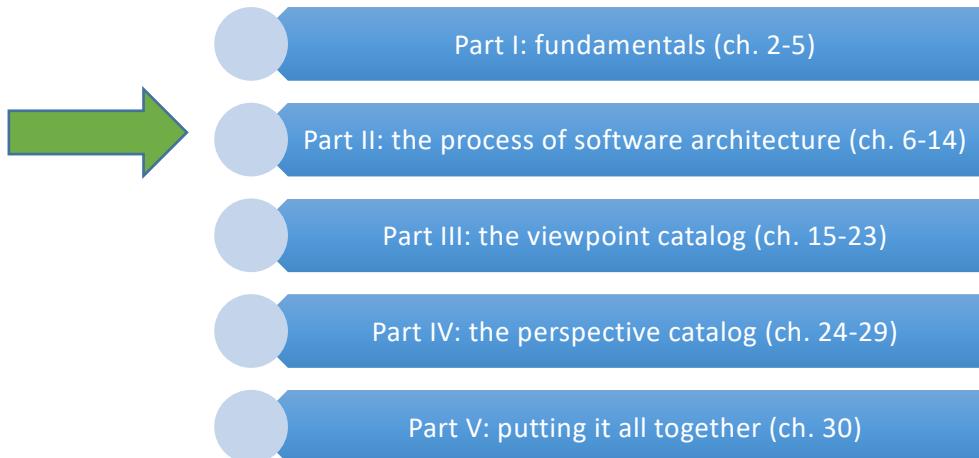


Scenarios (and prototyping)





Textbook: Software Systems Architecture



Magnus R. P. Hansen - BUITA5 - magnuha@ruc.dk

3



Supplementary readings:

A large blue circle icon is positioned to the left of the first reading.

Rudd, J., Stern, K., & Isensee, S. (1996). Low vs. High-fidelity prototyping debate. *Interactions*, 3(1), 76-85.

A large blue circle icon is positioned to the left of the second reading.

Tohidi, M., Buxton, W., Baecker, R., & Sellen, A. (2006). User sketches: A quick, inexpensive, and effective way to elicit more reflective user feedback.

Magnus R. P. Hansen - BUITA5 - magnuha@ruc.dk

4

Learning goals



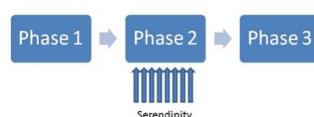
- Know the central concepts of scenarios: types, uses, capturing, and application.
- Know the central concepts of architectural models: qualitative, quantitative, and types of styles for addressing your audience
- Know the structure of the architectural description (AD)
- Be able to choose and use techniques for depicting a chosen scenario in the architecture definition process as a prototype mockup sketch

Magnus R. P. Hansen - BUITA5 - magnuha@ruc.dk

5

Main problems to ponder

- How do you focus on which phase you are in the process?
- How do you determine a proper level of abstraction (e.g. abstract model or fully dressed usage scenario?)
- How do you model an architecture when migrating when the existing architecture are non-tactile?
- What is the role of architectural products in relation to:
 - Communication
 - Feedback
 - Verification
 - Go no-go decisions



Magnus R. P. Hansen - BUITA5 - magnuha@ruc.dk

6

The architectural description

Rozanski & Woods

Chapter 13

Magnus R. P. Hansen - BUITA5 - magnuha@ruc.dk

7

Products of the AD



Content

1. Document control
2. Table of Content
3. Intro + management summary
4. Stakeholders
5. General Arch. Principles
6. Design Decisions
7. Central viewpoints
8. Views
9. Quality property summary
10. Important scenarios
11. Issues awaiting resolution

Questions to answer

1. How do you ensure currency?
2. How is the AD structured?
3. What is the purpose?
4. Who are involved
5. How will you achieve it technically?
6. What decisions, rational, and alternatives
7. What are the most relevant concerns?
8. What are the qualities of the concerns?
9. What are important insights so far?
10. How will the system behave?
11. What have you learned that needs solving later on?

Magnus R. P. Hansen - BUITA5 - magnuha@ruc.dk

8



Principles of how to judge your AD

Principle

1. Correctness
2. Sufficiency
3. Timeliness
4. Conciseness
5. Clarity
6. Currency
7. Precision

How you can ensure it by focusing on:

1. ...addressing needs and solutions
2. ... covering (all) relevant concerns
3. ...the most risky/most insightful aspects
4. ...*architectural significant* areas
5. ...making sure relevant stakeholders understand the AD
6. ...minimal AD kept up to date
7. ...abstraction, key principles, key views, diagrams, tables, remove all text in sentences

Magnus R. P. Hansen - BUITA5 - magnuha@ruc.dk

9



The architectural model

Rozanski & Woods

Chapter 12

Magnus R. P. Hansen - BUITA5 - magnuha@ruc.dk

10



The architecture model

- “...is an **abstract, simplified, or partial** representation of **some aspects** of an architecture, the purpose of which is to **communicate those aspects of the system to one or more stakeholders.**”

Understand

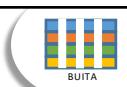
Communicate

Analyze

Organize

Magnus R. P. Hansen - BUITA5 - magnuha@ruc.dk

11



Strategies

- Match the complexity and detail of your architectural model to the interests and skill level of your audience
- Ensure that your audience is aware of any simplifications and approximations

Magnus R. P. Hansen - BUITA5 - magnuha@ruc.dk

12



Quantitative models

- “Make statements about the measurable properties of an architecture, such as performance, resilience, and capacity.”
- Focuses on **“System Quality”**
- Will often result in rough approximations (due to time restrictions)
- Usually in the form of an **equation**



Qualitative models

- “Illustrates the key structural or behavioral elements, features, or attributes of the architecture being modeled”.
- Focuses on **“System Behavior”**
- Choose a modeling language and stick to it
- Some models require different ‘languages’ if you want them to communicate with both technology stakeholders and business stakeholders
 - Eg. UML vs. BPMN
- (A domain model is a good beginning)



Sketches

- “A sketch is a deliberately informal graphical model, created in order to communicate the most important aspects of an architecture to a non-technical audience. It may combine elements of a number of modeling notations as well as pictures and icons.”
- Examples include “rich pictures” (see BUITA 9 – SDD2)
- Useful way to help stakeholders understand the essence of the AD

Magnus R. P. Hansen - BUITA5 - magnuha@ruc.dk

15



Example A) quantitative modeling

$$\text{serverDataLoad} = \text{transactionThroughput} \times \text{transactionLoad}$$

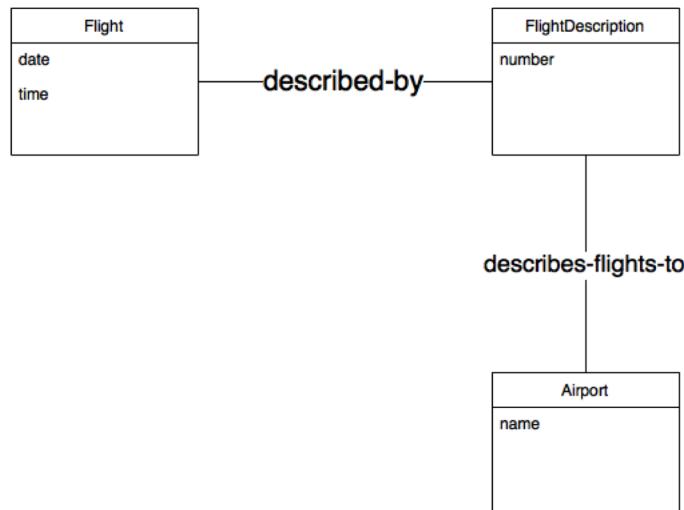
Magnus R. P. Hansen - BUITA5 - magnuha@ruc.dk

16



Example B) qualitative modeling

Relational model



Magnus R. P. Hansen - BUITA5 - magnuha@ruc.dk

17



Context diagrams...

- Describe the context of your architectural scope
- The scope (and thus the diagram) will change as you engage with more stakeholders
- Are a central diagram for the context viewpoint
- Usually occurs on 3 levels of abstraction

Magnus R. P. Hansen - BUITA5 - magnuha@ruc.dk

18



Abstraction level 0

- Black box of all internal processes and elements
- Often the whole organisational context is depicted here
- Only central stakeholders are depicted
- Abstract external elements are depicted to denote the inside/outside relationship and context



Abstraction level 1

- Central elements and their relation to the outside elements and stakeholders are depicted
- This is the main way to identify the scope of the architecture – the specific elements used in the AD



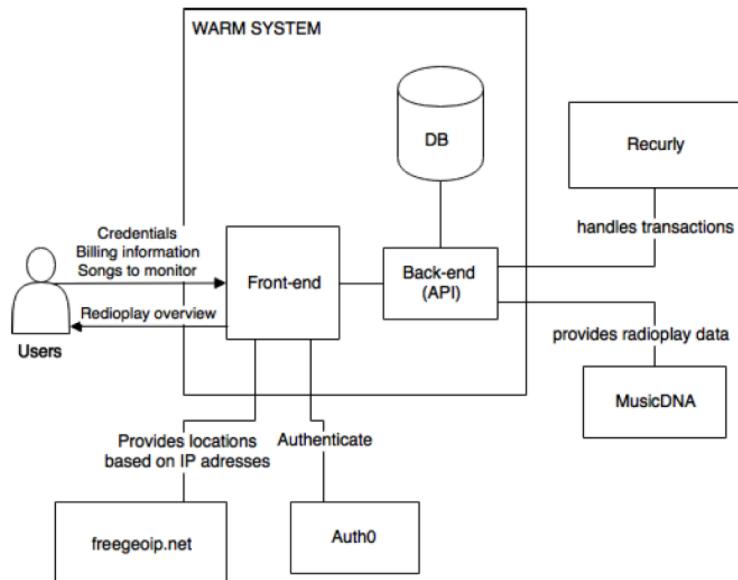
Abstraction level 2

- Detailed sub-elements are depicted:
 - Frameworks, functional calls
- Will often focus on a specific area of the architectural scope
- Will often be relegated to a "component" diagram and/or other process diagrams such as dataflow or system sequence diagrams
 - (thus be moved to the "Functional" viewpoint)

Magnus R. P. Hansen - BUITA5 - magnuha@ruc.dk

21

Context diagram of a music service



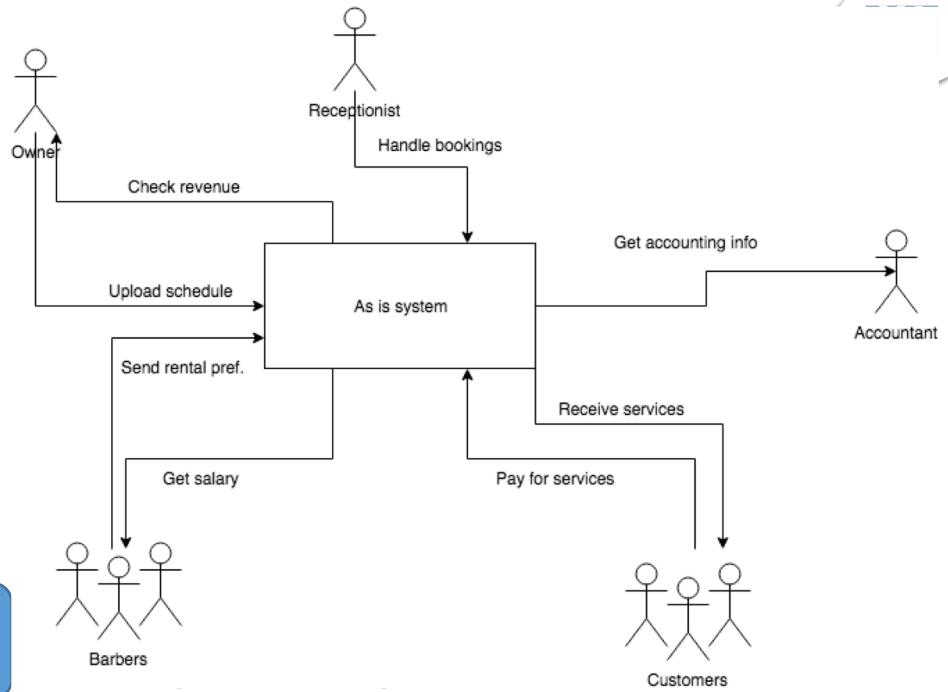
What abstraction level would you think this is?

Magnus R. P. Hansen - BUITA5 - magnuha@ruc.dk

22

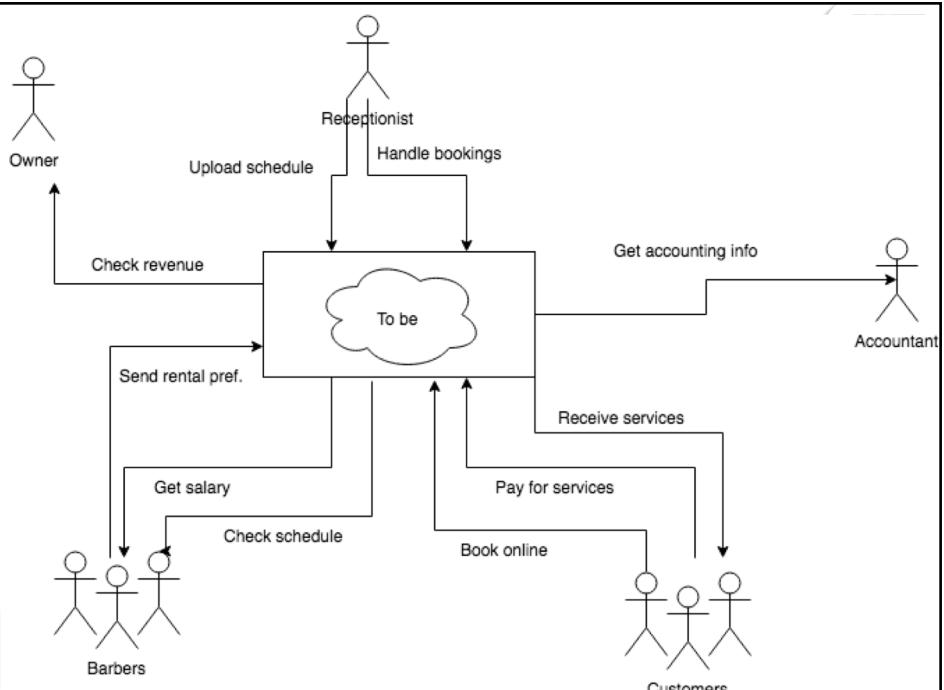
Context diagram of the barber shop

Abstraction level 0

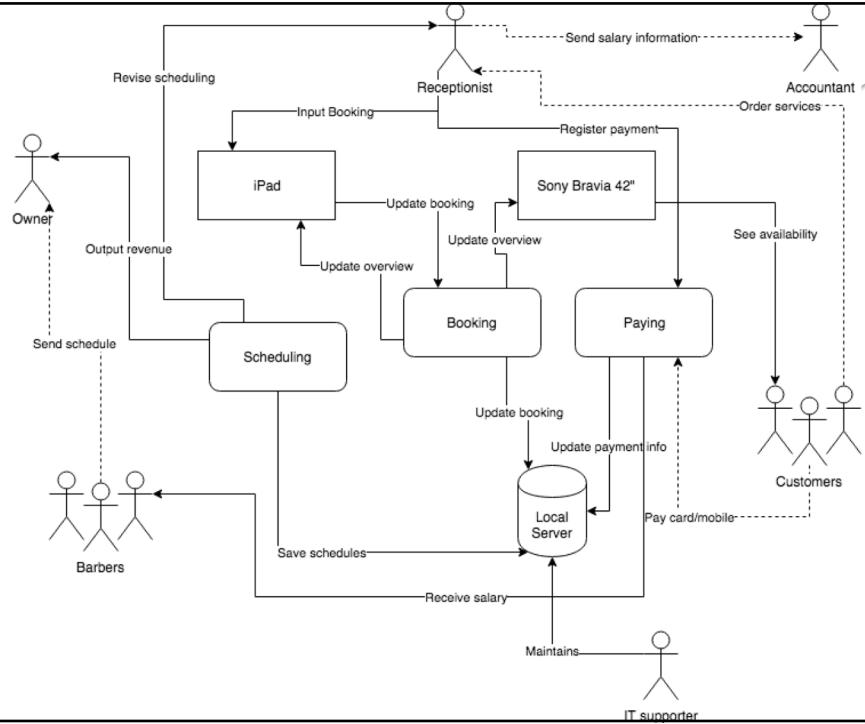


Context diagram of the barber shop – to be

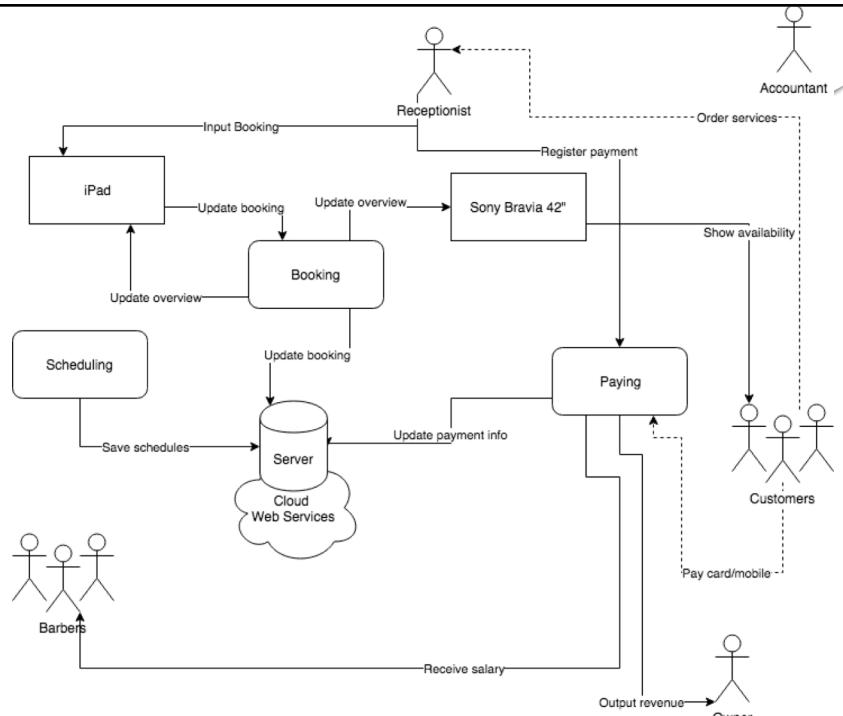
Abstraction level 0



Context diagram of the barber shop – as is



Context diagram of the barber shop – incomplete to be



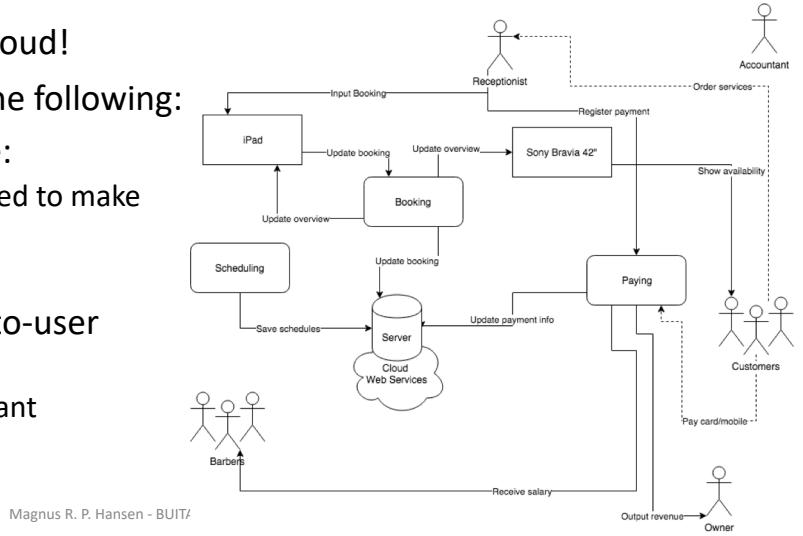


Make a "to-be" context diagram of the barber shop – abstraction level 1 (!)

The system is now in the cloud!

The scope and context is the following:

1. Online booking module:
 - a) What 'module' is required to make this happen?
 - b) Who will be using it?
2. Trim down direct user-to-user communication (---) :
 - a) No files sent to accountant
 - b) No files sent to owner



Guidelines 1

Proposal

1. Model purposefully
2. Address an audience
3. Abstract carefully and precisely
4. Focus your efforts according to your risks
5. Choose descriptive names

Example

1. What is the purpose of the model?
2. Who is the audience, what do they know?
3. Include only significant concepts
4. Only include the important scenarios (risk + consequence)
5. Names become sticky, choose a neutral one or one with shared meaning



Guidelines 2

Proposal

- 6. Define your terms
- 7. Aim for simplicity
- 8. Use a defined notation
- 9. Beware of implied semantics
- 10. Validate models
- 11. Keep models alive

Examples

- 6. Define each element and how to use it
- 7. Find a balance between effective understanding and banality
- 8. Choose a notation that is not based on your own intuition
- 9. What is **not** being explicit in the model?
- 10. Use the stakeholders of the real world
- 11. Keep changing and revising the models as you learn more.

Magnus R. P. Hansen - BUITA5 - magnuha@ruc.dk

30

Scenarios

Rozanski & Woods

Chapter 10

Magnus R. P. Hansen - BUITA5 - magnuha@ruc.dk

31



Main task of software architecture

- Meeting the needs of stakeholders
- "Performing certain tasks while exhibiting specific properties important to stakeholders"
- Tradeoffs between competing needs
- Keep focused, don't lose sight: root the AD in *reality*
- *Scenarios*



An architectural scenario

- "... is a well-defined description of an interaction between an external entity and the system. It defines the event that triggers the scenario, the interaction initiated by the external entity, and the response required of the system."
- Three points of a scenario:
 - Trigger
 - Interaction performed by actor
 - Response of the system



Using scenarios in order to:

- Providing input to the AD
- Define and validate system scope
- Evaluate the architecture
- Communicate with stakeholders
- Find missing requirements
- Drive the test process



Two types of scenarios

- Functional scenarios: external events; stimuli and feedback
- System Quality: System reacting internally to the environment



Examples of scenarios

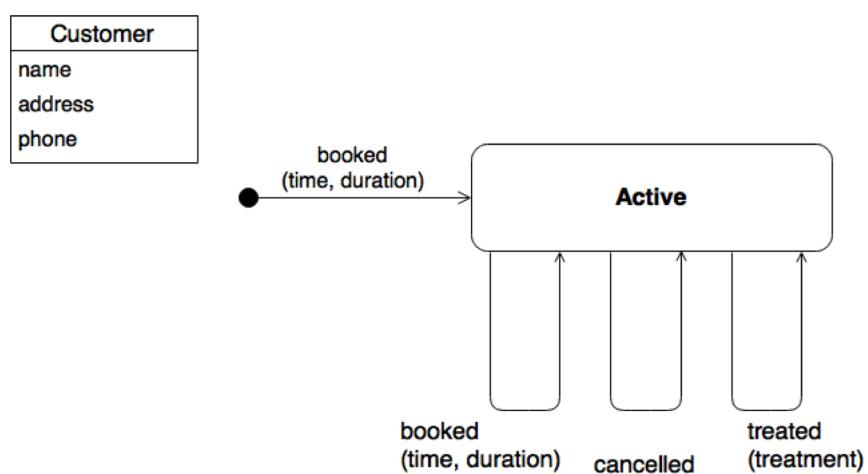
- 1. A particular set of interactions from the user to respond to
 - 2. Processing automatically at months end
 - 3. A peak load situation
 - 4. A demand of an external regulator
 - 5. Responses to failure
- 1. "Netbank" (online banking) → Show current account balances and recent trans.
 - 2. Payroll system → Calculate sick days, time spent at work, vacation and perform transaction to workers
 - 3. Elite Dangerous using AWS → On more than 10% package loss, scale up amount of virtual servers
 - 4. Changes to exam regulation → Change values in database according to exam changes
 - 5. Minklinik.dk, opret bruger via Safari → Check user browser model and use corresponding css and javascript standards

Magnus R. P. Hansen - BUITA5 - magnuha@ruc.dk

36



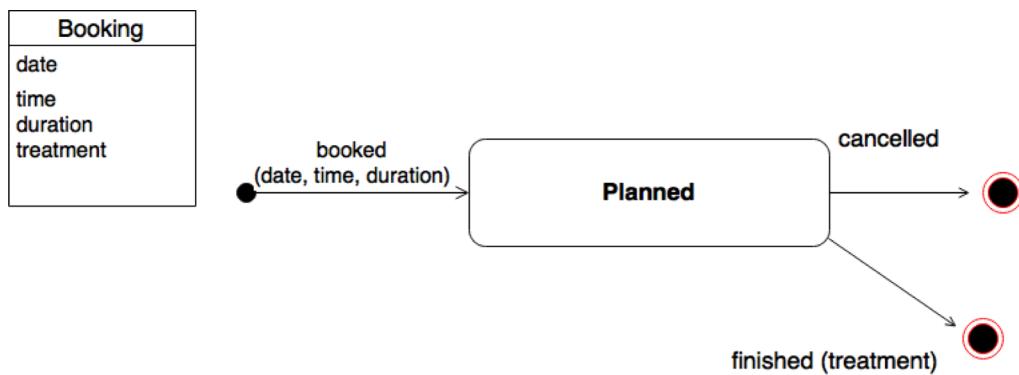
State diagram for the barbershop – customer class



Magnus R. P. Hansen - BUITA5 - magnuha@ruc.dk

37

State diagram for the barbershop – Booking class



Magnus R. P. Hansen - BUITA5 - magnuha@ruc.dk

38

Identifying scenarios



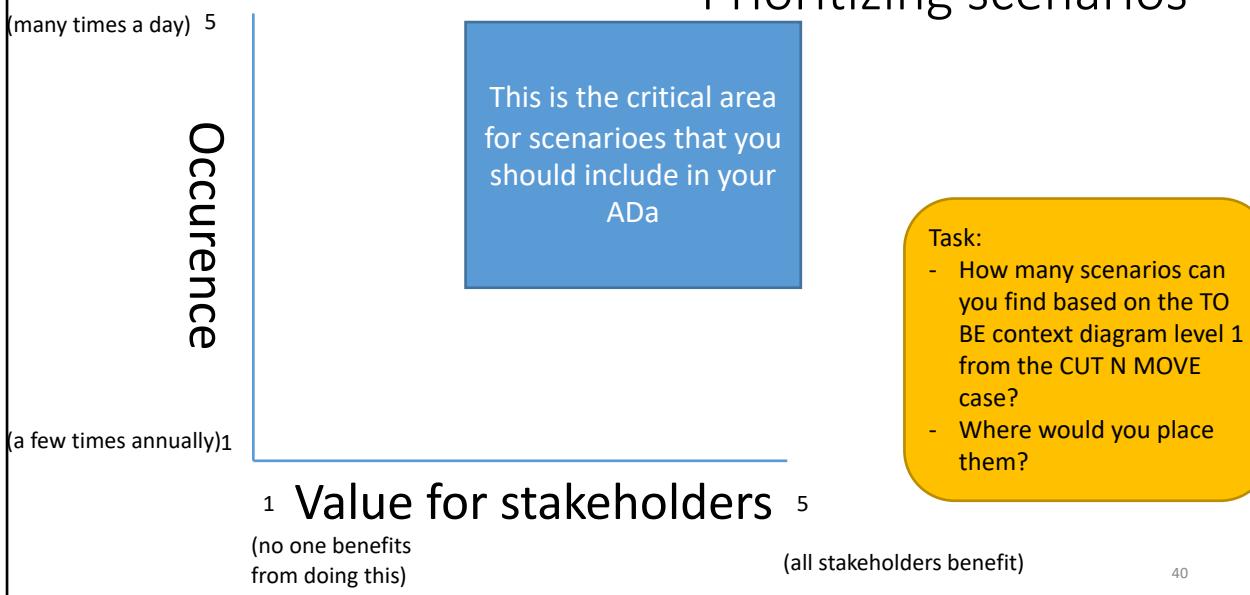
- Will originate from
 1. Requirements
 2. Stakeholders
 3. Experience
- 1. Functional requirements to a functional scenario vs. behaviours of the system
- 2. Workshops, observation, interviews
- 3. From similar systems or environments

Magnus R. P. Hansen - BUITA5 - magnuha@ruc.dk

39



Prioritizing scenarios



40



Levels of comprehensiveness

- Brief
- Fully dressed
- Casual

Single paragraph of the main success scenarios

Use case name
Scope
Level
Primary actor
Stakeholders and interests
Preconditions
Success Guarantee
Main Success scenario
Extensions
Special Requirements

(taken from Larman, C. (2006). *Applying UML and patterns, chapter 6.*)

Magnus R. P. Hansen - BUITA5 - magnuha@ruc.dk

41



Casual scenarios capture 6 (!) pieces of information

Functional

1. Overview
2. Stakeholders and interests
3. System state
4. System environment
5. External stimulus
6. Required system response

System Quality

1. Overview
2. Stakeholders and interests
3. System state
4. System environment
5. Environment changes
6. Required system behavior

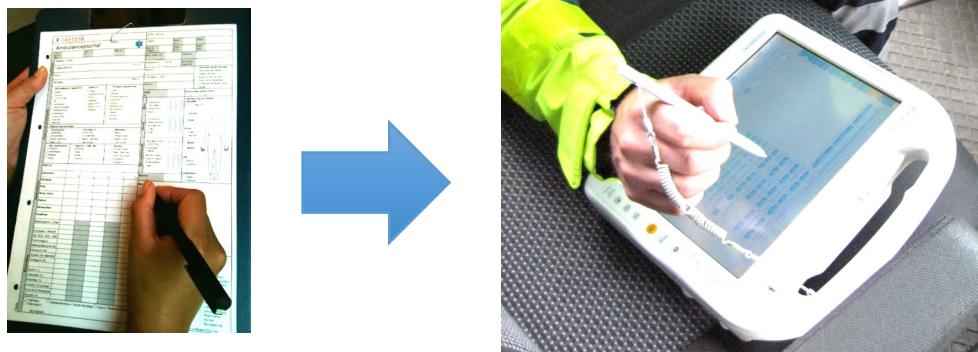


Example of a casual scenario

From the real life

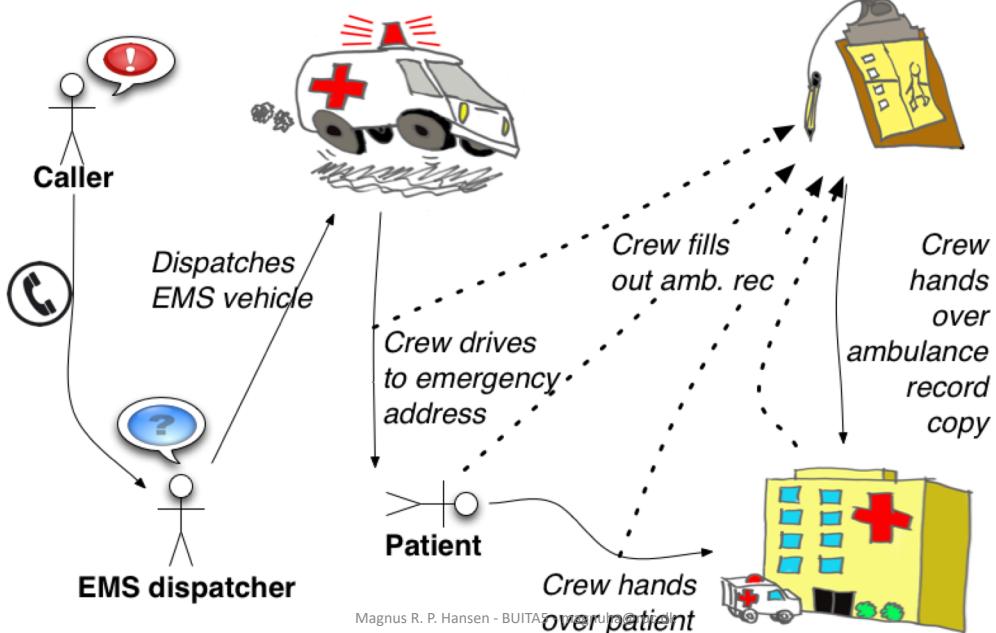


A new electronic ambulance record architecture



Magnus R. P. Hansen - BUITA5 - magnuha@ruc.dk

44



45



Overview – scenario name

- Inputting basic patient data into ambulance record



Magnus R. P. Hansen - BUITAS - magnuha@ruc.dk

47



Stakeholders and interests

- Ambulance paramedic (primary actor)
- Ambulance assistant (supporting)
- Patient (supporting)



Magnus R. P. Hansen - BUITAS - magnuha@ruc.dk

48



System state

- User is already logged in with user credentials
- New record already open with empty data fields
- Ambulance arrival time sent from Ambulance radio
- Power saving on (screen black)

Magnus R. P. Hansen - BUITA5 - magnuha@ruc.dk

49



System environment

- Bluetooth connected enabled to LP15 (LifePak15, defibrillator) IF needed
- GPRS data available
- Users located outside or inside ambulance



Magnus R. P. Hansen - BUITA5 - magnuha@ruc.dk

50



External stimulus

1. User turns on the handheld console
 - 1a. presses wake-up button on the handheld console
 - 1b. Taps screen
2. User inputs the "Patient arrival time"
 - 2a. User presses "Current time" button
 - 2b. Manually by selected time from dropdown
3. User inputs Basic patient information
 - 3a. User swipes national medical card (NMC) on magnetic stripe reader
 - 3b. User inputs DOB, Civil Registration Number, name

Magnus R. P. Hansen - BUITA5 - magnuha@ruc.dk

51



Required system response



- Data input is visible during and after inputting
- Data is validated during input
- Autosave data to central server
- Autosave status shown
- GPRS status shown
- LP15 connection status shown by recording vital parameters

Magnus R. P. Hansen - BUITA5 - magnuha@ruc.dk

52



What kind of scenario was this?

- Functional or system quality scenarios?
- Why? Why not?

Magnus R. P. Hansen - BUITA5 - magnuha@ruc.dk

53

Sequence diagrams using UML

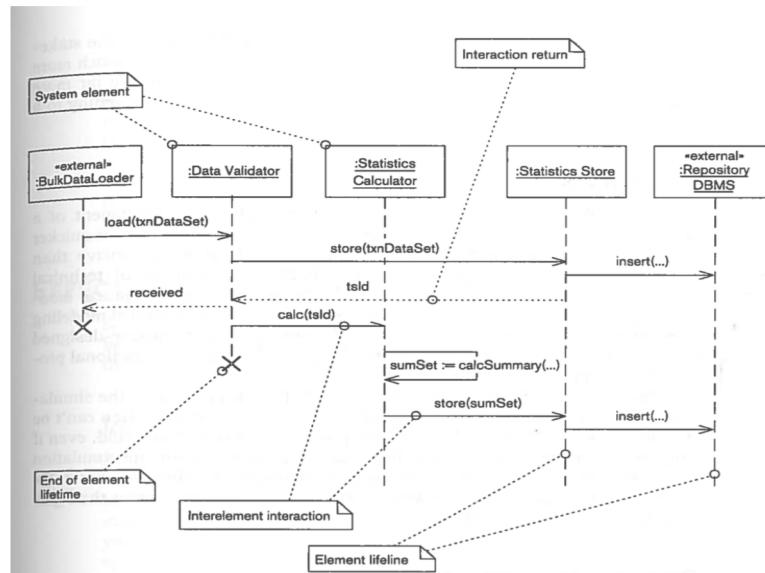
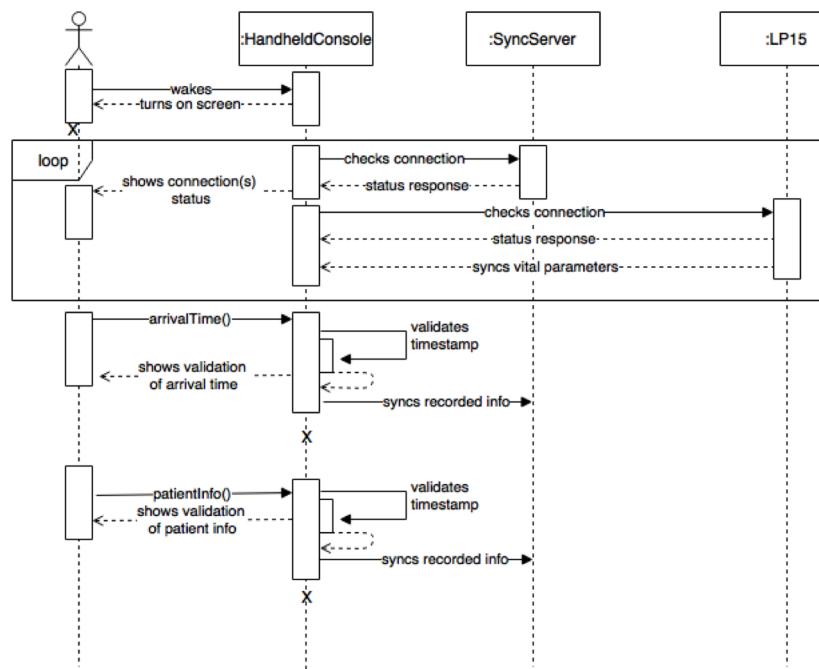


FIGURE 10-1 UML SEQUENCE DIAGRAM FOR THE INCREMENTAL STATISTICS UPDATE SCENARIO

Magnus R. P. Hansen - BUITA5 - magnuha@ruc.dk

54

System sequence diagram



55

Group task

- Make a "casual usage scenario"
- Make a "system sequence diagram" (if you have time)
- Choose a success and a failure scenario
- Discuss and reflect on flaws, pros & cons of making, using and evaluating scenarios
- Draw on the guidelines for quality on the following slides

Based on the CUT N MOVE case and the identified scenarios



Quality of scenarios



- Credible
- Valuable
- Specific
- Precise
- Comprehensible

Magnus R. P. Hansen - BUITA5 - magnuha@ruc.dk

58



The Usefulness of scenarios

- Walkthroughs
- Simulations
- Prototype implementation testing
- Full-scale live testing

Magnus R. P. Hansen - BUITA5 - magnuha@ruc.dk

59



Final words on scenarios

- Identify a **focused** set of scenarios'
- Use **distinct** scenarios
- Use scenarios **early**
- Include the use of **system quality scenarios**
- Include the of **failure scenarios**
- Involve **stakeholders** closely

Magnus R. P. Hansen - BUITA5 - magnuha@ruc.dk

60



Checklist

- Wide enough range of system quality scenarios?
- Wide enough range of failure and exception scenarios?
- Prioritized scenarios on consequences and risk?
- Kept the number of scenarios small, 15-20?
- Reviewed and agreed on required responses and behaviours with appropriate stakeholders?
- Included your own valued scenarios?
- Catalogued and named?
- Addressing gaps or mistakes from requirements identified from scenarios?
- Revised the architectural design appropriately?

Magnus R. P. Hansen - BUITA5 - magnuha@ruc.dk

61

Prototyping

Product, process and fidelity

Rudd et al., 1996

Tohidi et al., 2006

Magnus R. P. Hansen - BUITA5 - magnuha@ruc.dk

62



Prototyping in general

- High-fidelity vs low-fidelity
 - Completeness
 - Realism
 - Reusability
- To accomplish being an effective tool
- Both a product and a process

Magnus R. P. Hansen - BUITA5 - magnuha@ruc.dk

63



Low-fidelity prototype

- Trading off accuracy for speed
- Limited functionality
- Limited interaction
- Limited detail
- Level of fidelity = how it appears to the person viewing it
- Purpose
 - Communicate
 - Educate
 - Inform
- Shows design direction
- NOT navigation or interaction
- "See, not touch"

Magnus R. P. Hansen - BUITA5 - magnuha@ruc.dk

64



Low-fidelity prototyping

- A rapid development technique
- Often storyboarding as scenario tools
- A learning approach taken by a role (consultant, developer etc)
- Requires someone to be skilled at operating the prototype
- Good early on in the design process
- The product is used as a common language for articulating requirements
- Creates close relations with users
- No technical skills required

Magnus R. P. Hansen - BUITA5 - magnuha@ruc.dk

65



Low-fidelity prototyping

Advantages

- Lower development cost.
- Evaluate multiple design concepts.
- Useful communication device.
- Address screen layout issues.
- Useful for identifying market requirements.
- Proof-of-concept.

Disadvantages

- Creates gaps in expectations
- Overlooking important design decisions
- Not good for summative evaluation
- Limited error checking.
- Poor detailed specification to code to.
- Facilitator-driven. Asymmetric relations
- Limited utility after requirements established.
- Limited usefulness for usability tests.
- Navigational and flow limitations.

Magnus R. P. Hansen - BUITA5 - magnuha@ruc.dk

66



High-fidelity prototypes

- Fully interactive
- Represent core functionality
- Trading off speed for accuracy
- Vertical vs horizontal

Magnus R. P. Hansen - BUITA5 - magnuha@ruc.dk

67

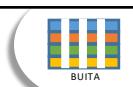


High-fidelity prototyping

- Users are involved themselves
- Requires a surveyer, not a presenter

Magnus R. P. Hansen - BUITA5 - magnuha@ruc.dk

68



High-fidelity prototyping

Advantages

- Complete functionality.
- Fully interactive.
- User-driven.
- Clearly defines navigational scheme.
- Use for exploration and test.
- Look and feel of final product.
- Serves as a living specification.
- Marketing and sales tool.

Disadvantages

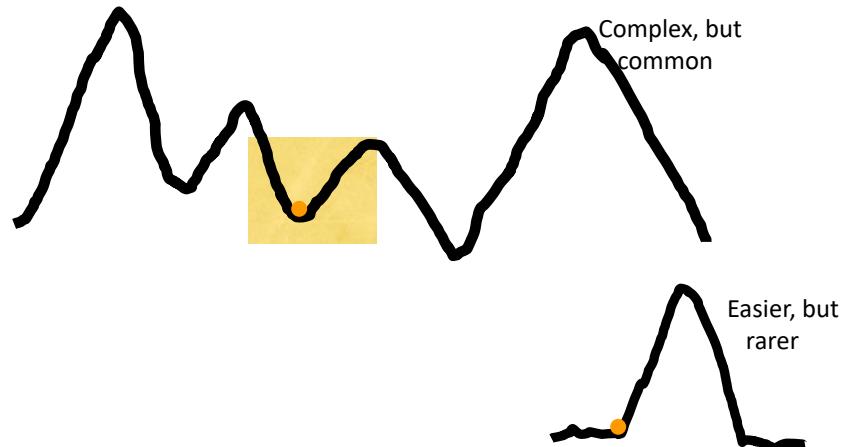
- More expensive to develop
- Time-consuming to create.
- Inefficient for proof-of-concept designs.
- Not effective for requirements gathering.
- Expensive to change

Magnus R. P. Hansen - BUITA5 - magnuha@ruc.dk

69



Exploring design spaces



Magnus R. P. Hansen - BUITA5 - magnuha@ruc.dk

70

Sketching

"Designers do not draw sketches to externally represent ideas that are already consolidated in their minds. Rather, they draw sketches to try out ideas, usually vague and uncertain ones. By examining the externalizations, designers can spot problems they may not have anticipated. More than that, they can see new features and relations among elements that they have drawn, ones not intended in the original sketch. These unintended discoveries promote new ideas and refine current ones."

Quote in Tohidi et al. (2006)

Magnus R. P. Hansen - BUITA5 - magnuha@ruc.dk

71

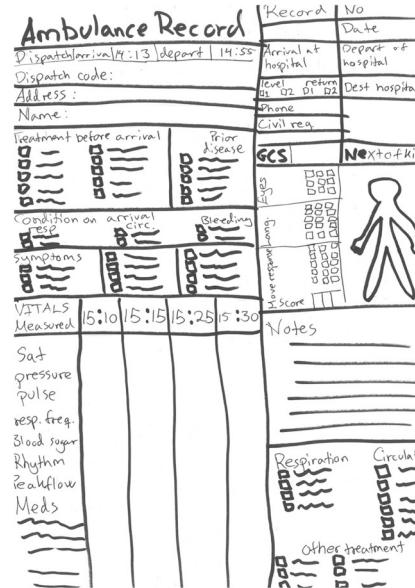
Sketching

- Rough ideas quickly (2 mins)
- Interaction with ideas
- Increases redesign and reflection
- Decreases criticism and reaction
- Fluid action, creation and redesign when sketching

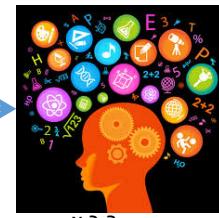
(Tohidi et al., 2006)

Magnus R. P. Hansen - BUITA5 - magnuha@ruc.dk

72



Sketching 2



x 2-3



Different possibilities!



Focus!

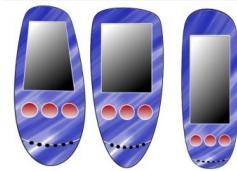
Magnus R. P. Hansen - BUITA5 - magnuha@ruc.dk

73



An example:

Sketching alternatives



- Sketching an idea is a way of getting it out of your head, thereby making room for new ideas
- The first two or three sketches reflect standard solutions. It is only after having made them you start to be creative
- The sketching of alternatives is meant to explore *different* possibilities, i.e. differences in structure, metaphor... (vs. refining the initial design)
- Alternative designs provide for focused evaluation

Magnus R. P. Hansen - BUITA5 - magnuha@ruc.dk

74



User sketches

How it may be done:

- Provide users with pen and paper, and ask:
- “Please sketch your ideal [thermostat] design in the space provided”
- In the studies, no time limit was imposed. Users spent an average of 3.9 minutes (!)

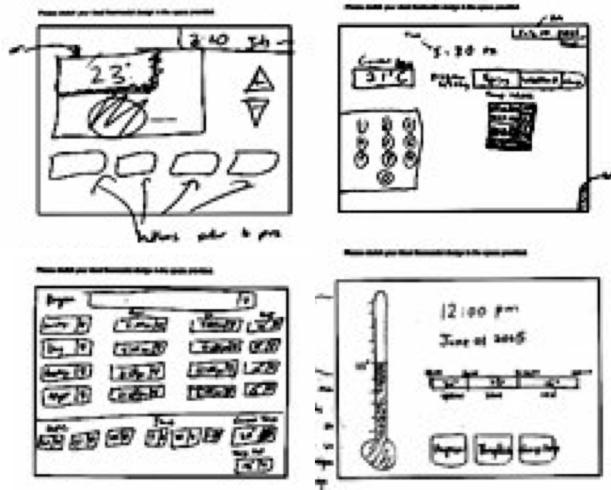
(Tohidi et al., 2006)

Magnus R. P. Hansen - BUITA5 - magnuha@ruc.dk

75



Examples



Magnus R. P. Hansen - BUITA5 - magnuha@ruc.dk (Tohidi et al., 2006)

76

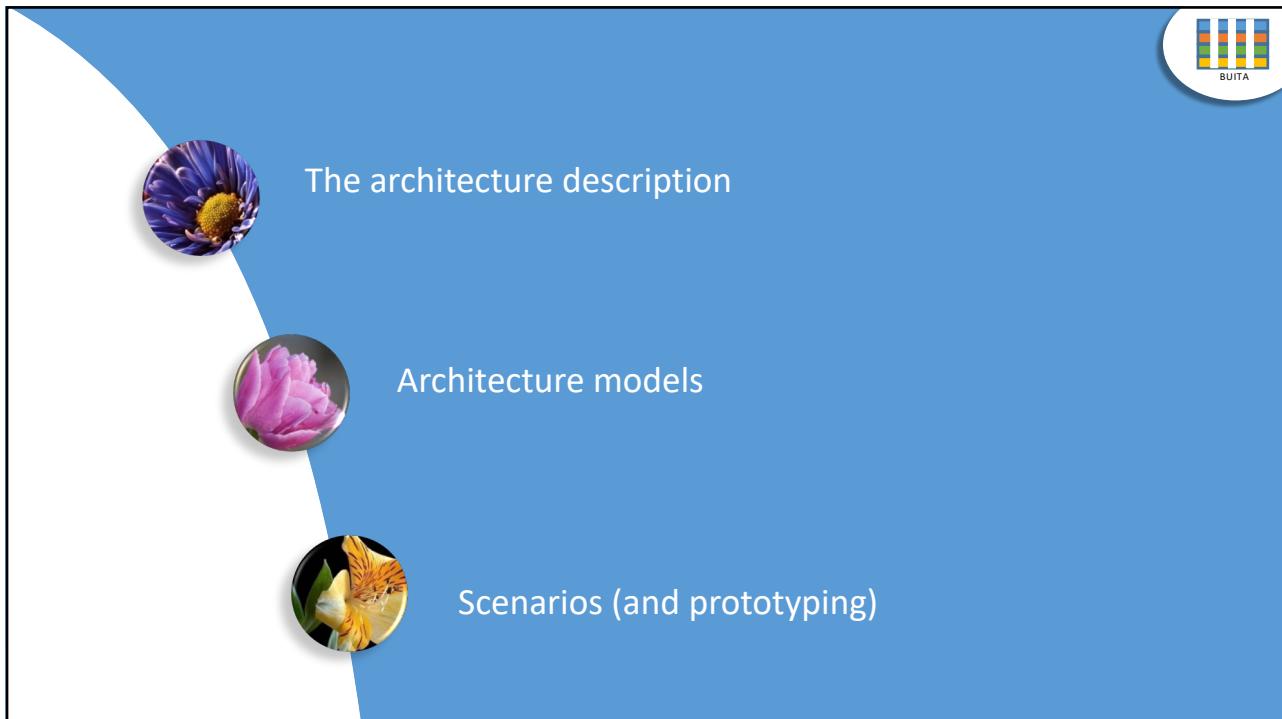


Group task

- Groups
 - Sketch 2 different user interface sketches
 - Use 4 minutes per sketch
 - Discuss pros and cons of the design themes
 - Use 10 minutes
 - Make 1 final 'hybrid' sketch
 - Use 3 minutes from this sketch
 - Present your sketch
 - Use 1 minute per person
- LPT: don't know what to sketch?
→ Just start drawing lines or circles or squares. Inspiration comes from the hand and eyes, not the brain
 - Don't be shy, keep your sketches rather than throw them out
 - It is **not** dangerous doing sketches

Based on the CUT N MOVE case and the identified scenarios

Magnus R. P. Hansen - BUITA5 - magnuha@ruc.dk



Putting it all together

- Context diagram is dependent on your stakeholders
- Scenarios are dependent on your context diagram and stakeholders' first cut concerns
- Choice of model is equally dependent on your stakeholders

Phase 1 → Phase 2 → Phase 3

Serendipity

Magnus R. P. Hansen - BUITA5 - magnuha@ruc.dk

79

