



LUND
UNIVERSITY

ETSN15 2024

Requirements Engineering

Lecture 5:

Prototyping [PROTO1&2] & Agile RE [AGRE]

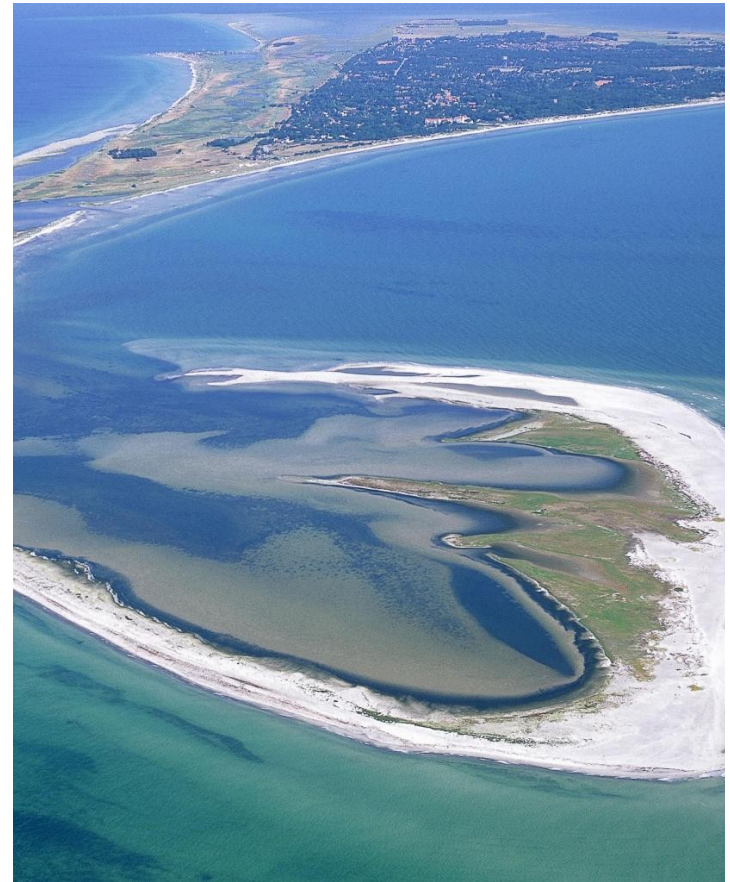
Elizabeth Bjarnason

Björn Regnell

<http://www.cs.lth.se/krav>

Requirements change

- Constantly
- Sometimes quickly!
- Why?
 - We learn
 - Changed needs & priorities by users & other stakeholders
 - Tech development
 - Competition
 - Time-to-market pressures



Underlying assumptions → agile RE

The Agile Manifesto, <http://agilemanifesto.org/>, 2001

Extensive documentation

- Costly
- Time consuming
- RE competence **REQUIRED**
- Dev - stakeholder interaction **NOT** required

Light-weight / agile RE



Cheaper - initially



Quicker - initially



RE competence **nice-to-have**



Dev – stakeholder interaction **REQUIRED**

“We don’t do requirements. We are agile.”

Wrong! Exactly all projects need & have requirements ==

ideas/decisions of what the product should do

In Agile projects, some requirements ***are*** documented

- as traditional requirements
- as user stories & acceptance criteria
- as backlog entries
- as test cases
- combo of “requirements” and other artefacts

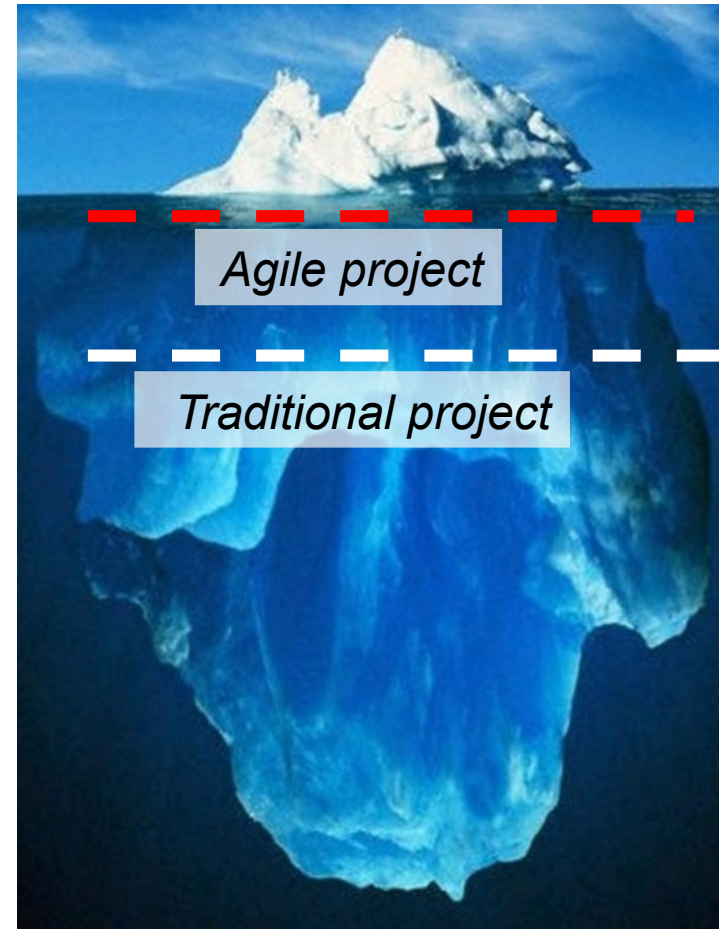
Many requirements are ***NOT*** documented (can be risky)

RE in Agile Projects

Practices [AGRE]

- *Iterative RE: Gradual detailing*
- *Work order*
 - *Extreme prioritization: Just-in-time*
 - *Constant planning*
- *Integrated RE:*
 - *Dev roles more involved in RE*
 - *Face-to-face communication*
 - *Reviews & tests*
 - ***Prototyping***
 - *Test-driven development*

Level of detail at dev start



The Practice of Prototyping [PROTO1 & 2]

Prototyping

use of a prototype to explore, communicate, and evaluate potential solutions

Prototype

early sample, model, or release, which simulates aspects of the final product and enables cost effective testing with real users

- Communicate
- Validate
- Elicit



requirements, goals, ideas, priorities, ...

PURPOSE, SCOPE, USE, STRATEGY

PURPOSE of Prototyping [PROTO1]



PURPOSE

Exploration & learning

Communication: sales, alignment

Incremental development

Quality improvement

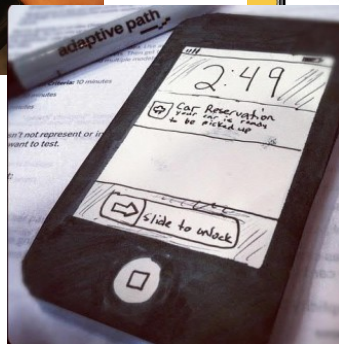
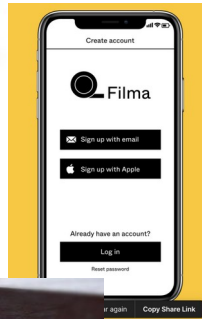
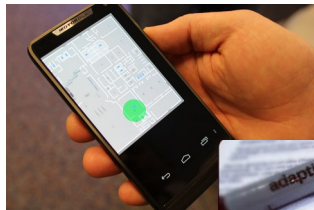
Validation & Testing

- Fit: Problem-solution, Product-market
- Technical feasibility
- Usability testing

Advice:

- Consider your **purposes** with prototyping and for each prototyping instance
- Select **scope, media, and use** of prototype to match purpose

SCOPE & MEDIA of Prototype [PROTO1]



SCOPE

Breadth of functionality

Functional refinement

Visual appearance

Interactive & haptic behaviour

Data realism

MEDIA

Sketch: paper or computer-based

Wireframe: paper or computer-based

Mock-up: paper or computer-based

Source-code software

Other: video, interview

Advice:

- Consider which **functional breadth** and **refinement**, **visual appearance** and **interactive behaviour** that is needed for your **purposes**
- Balance the **costs** of prototype building (affected by Scope and Media) against possible **benefits**

USE of Prototype [PROTO1]



USE of prototype

Reviewers: internal, FFF (family-foes-friends), external

Prototype interaction: yes, no (demo)

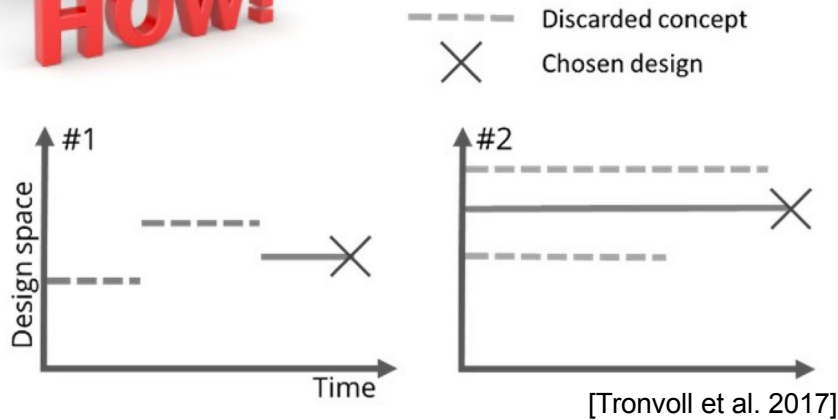
Review approach: scenario-based, free

Usage environment

Advice:

- Select **reviewers** that represent **stakeholders** and **user categories** that can provide feedback needed for chosen **purposes**
- Design **review approach** and **interaction** to align with **purpose** and focus of prototyping
- Select **usage environment** to match **purpose**

Exploration STRATEGY [PROTO1]



Iteration focus: Business, product, feature, optimisation
Iteration size

Single vs Parallel exploration

Advice

- Consider the size of potential solution space and select suitable type of **exploration** and **iteration size**
- In early stages, consider **parallel exploration**, when more certain, switch to **single exploration**
- Match **prototype scope**, **media** and **use** to the iteration focus, and align with purpose

Prototyping Aspects Model Summary [PROTO1]

PURPOSE of Prototyping

- Exploration & learning
- Communication: sales, alignment
- Incremental development
- Quality improvement
- Validation & Testing
 - problem-solution / product-market fit,*
 - technical feasibility, usability testing*

SCOPE of Prototype

- Breadth of functionality
- Functional refinement
- Visual appearance
- Interactive & haptic behaviour
- Data realism

Prototype MEDIA

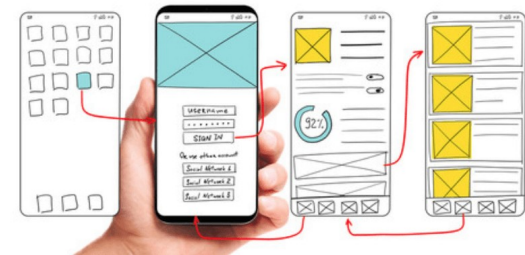
- Sketch: paper or computer-based
- Wireframe: paper or computer-based
- Mock-up: paper or computer-based
- Source-code software
- Other: video, interview

USE of prototype

- Reviewers: internal, FFF, external
- Prototype interaction: yes, no (demo)
- Review approach: scenario-based free
- Usage environment

Exploration STRATEGY

- Single vs parallel exploration
- Iteration focus: Business, product, feature, optimisation
- Iteration size



[This Photo](#) by Unknown Author is licensed under [CC BY-ND](#)

Prototyping in SW Startups

[PROTO2]



- **Natural part of product development**
- **Early stages**
 - Sketches & mock-ups
 - **Internal** use
 - Elicit, test & communicate ideas & requirements
- **Later stages**
 - Interactive mock-ups and early product versions
 - **External** use with customers and sponsors
 - Communicate & validate product proposals
 - Elicit further ideas & requirements

Sketches were not immediately mentioned when asked about prototyping. But several startups use simpler drawings and visualization in the early stages of product development to test and communicate their ideas.

Prototyping in Startups [PROTO2]



Implicitly required for funding.
Want to show that...

- Customers willing to pay
- Market & business viability
- Trust & Confidence in startup's ability

→ **Startups prefer** prototypes with a broad & refined scope
→ increases cost

Need to consider Cost-Benefit!

Optional Video on Prototyping in practice from 2022:
<https://canvas.education.lu.se/courses/28212/pages/15a-prototyping-2022>



Hampus Jakobsson

Angel investor in > 100 companies

Now: Pale Blue Dot, \$100m climate-tech fund

Previous: LTH, built & scaled 2 startups

Paper [AGRE]

*Agile Requirements Engineering Practices:
An Empirical Study*

by Balasubramaniam Ramesh and Lan Cao

In: IEEE Software, pp. 60-67, January/February 2008

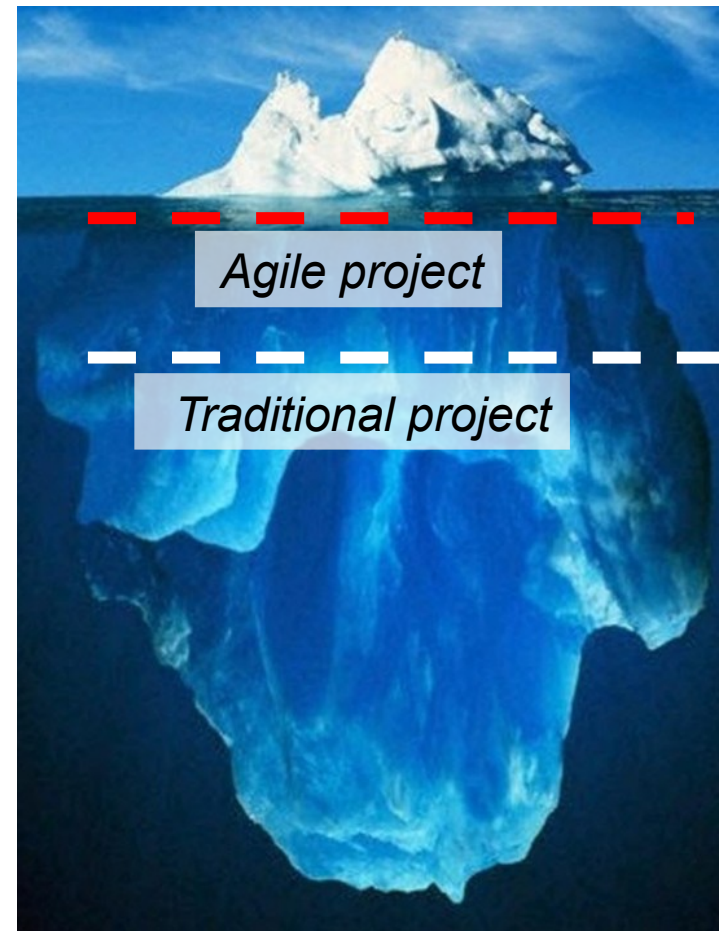


RE in Agile Projects [AGRE]

Practices

- *Iterative RE: Gradual detailing*
- *Work order*
 - *Extreme prioritization: Just-in-time*
 - *Constant planning*
- *Integrated RE:*
 - *Dev roles more involved in RE*
 - *Face-to-face communication*
 - *Reviews & tests*
 - *Prototyping*
 - *Test-driven development*

Level of detail at dev start



Agile RE practices in 16 companies

Adoption level	Practice						
	Face-to-face communication	Iterative RE	Extreme prioritization	Constant planning	Prototyping	Test-driven development	Reviews & tests
High	8	9	10	8	8	5	11
Medium	8	5	6	6	3	1	4
Low	0	2	0	2	0	0	1
None	0	0	0	0	5	10	0

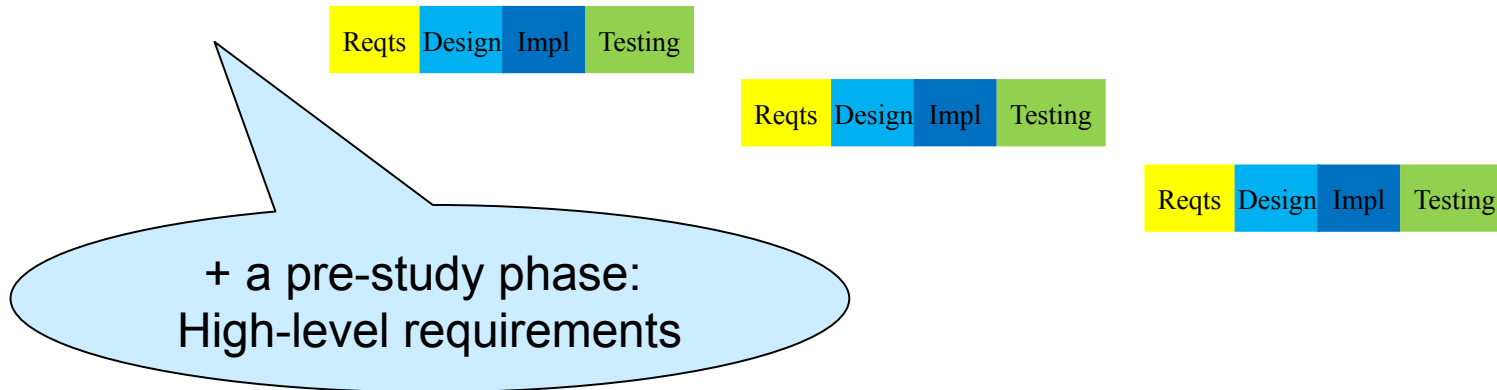
Organization pseudonym	Industry and products
Enco	Energy and communications. Offers forecasting tools.
HealthCo	Healthcare and utilities. Offers an online service to help customers select health insurance and utility services.
Venture	Across industries. Helps brick-and-mortar companies develop a Web presence.
Entertain	Film and television industry. Offers high-tech indexing and search tools online.
HuCap	Administration. Carries out human-resource administration for other companies online.
TravelAssist	Transport and tourist industry. Offers online services.
ManageRisk	Across several industries. Offers insurance online.
Transport	Transportation and logistics industry. Offers services online.

Transport	Transportation and logistics industry. Offers services online.
ServeIT	Consulting and services. We studied the part of the firm that offers consulting services for business-to-business communication.
HealthInfo	Healthcare information systems. Offers information systems solutions to hospitals, physicians' offices, and home healthcare providers.
SecurityInfo	Security software. Offers software for Internet security.
AgileConsult	Software consulting. Offers consulting services on agile software development.
EbizCo	Packaged software development. Offers e-business connections and transactions.
FinCo	Online financial-transaction support. Offers online payments.
NetCo	Network software consulting. Offers services on developing network systems and architectures.
BankSoft	Banking information systems. Offers software that handles financial transactions.

Traditional Development Process



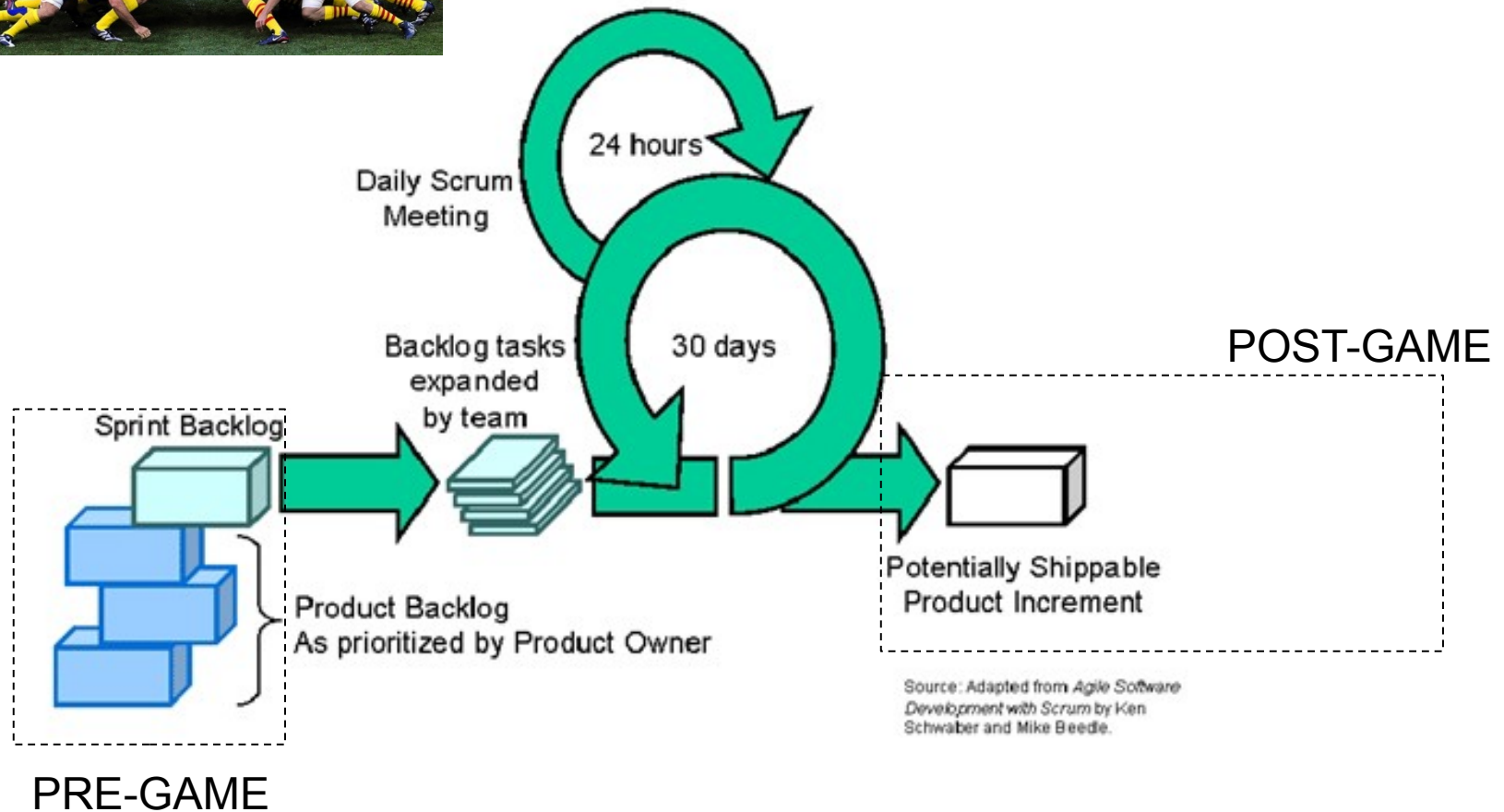
Agile Development Process – **Integrated RE**



- Same activities, different sizing and timing
 - Different principles and management approach
 - Different people detailing requirements
 - Different documentation formats



Scrum sprints - Time boxed iterations



- Requirements **INTEGRATED** in backlog, test cases, design docs etc

Continuous Feedback & Transparency

Business, Management and Development roles involved in

- Sprint planning meeting
- Daily stand-up meetings
- End-of-sprint demo
- Sprint retrospective meetings



User story & Acceptance Criteria

User story:

As a passenger, I can cancel a flight reservation

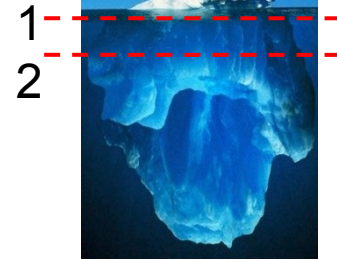
Acceptance criteria / test cases

- Verify that a premium member can cancel the same day without a fee
- Verify that a non-premium member is charged 10% for a same-day cancellation
- Verify that an email confirmation is sent
- Verify that the hotel is notified of any cancellation

Cohn, Mike. *User stories applied: For agile software development*. Addison-Wesley Professional, 2004.

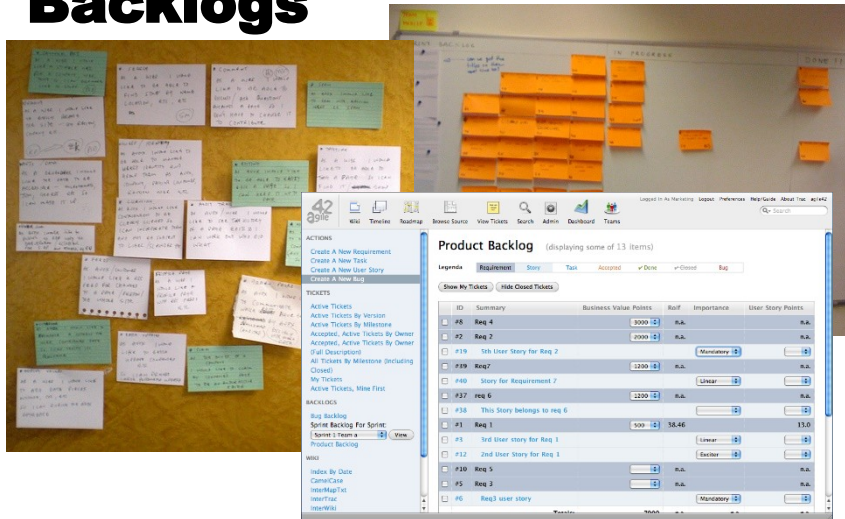


Specification of user stories



1. Product Owner/Customer defines & prioritizes Epics/User stories in **product backlog**
2. Team defines details for each user story in **sprint backlog**
 1. Tasks
 2. Acceptance criteria & test cases

Backlogs



Story cards

Test Cases as Requirements in Agile practice

Benefits	Challenges
Elicitation and validation	
EB1 Cross-functional communication	EC1 Good Customer-Developer relationship
EB2 Align goals & perspectives between roles	EC2 Active customer involvement
EB3 Address barrier of specifying solutions	EC3 Sufficient technical and RE competence
EB4 Creativity supported by high-level of requirements	EC4 Complex requirements, e.g. quality requirements
Verification	
VB1 Supports regression testing	VC1 Varying (biased) results for manual tests
VB2 Increased requirements quality	VC2 Ensuring correct requirements info to test
VB3 Test coverage / RET alignment	VC3 Quality requirements
Tracing	
TB1 Implicit Requirements - test case tracing	TC1 Tool integration
Managing changes	
MB1 Communication of changes	MC1 Locating impacted requirements
MB2 Requirement are kept updated	MC2 Missing requirement context
MB3 Maintaining RET alignment	MC3 Multiple products in one product line
MB4 Detecting impact of changes	
Customer agreement/contractual	
CB1 Facilitate resolving conflicting views	CC1 Use-case related structuring
CB2 Support certification of compliance	

Table 7 in ATCR.pdf (optional paper in zip not included in exam)]

Bjarnason, Unterkalmsteiner, Borg, & Engström (2016). *A multi-case study of agile requirements engineering and the use of test cases as requirements*. Information and Software Technology, 77, 61-79.



Face-to-face communication

Direct communication between customer and development

- Techniques
 - User Stories == high-level requirements spec
 - Complemented by other artifacts, e.g. "backlog"
- Prerequisites
 - Active involvement of (knowledgeable) customers

Customers can steer project

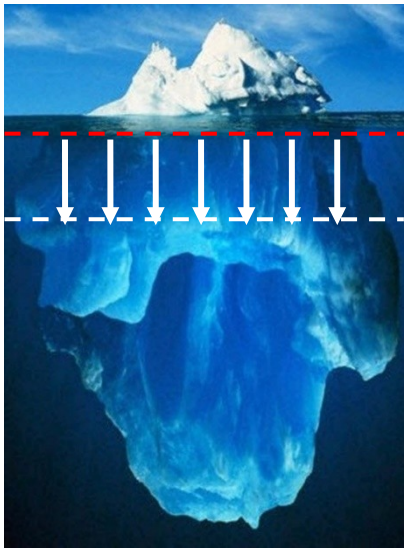
Avoids time-consuming documentation

Risk of **inadequate requirements**

On-site customer rep is challenging

Handling **more than one customer**

Relies on trust rather than agreed requirements



Iterative RE

Requirements **emerge** during development based on **initial high-level requirements**

- Techniques
 - Requirements analysis and detailing for each development cycle
 - Requirements intertwined with design

Good customer relationship

Clearer and understandable requirements

due to direct customer interaction

Accurate cost and scheduling of project

Neglect of **quality requirements**

Lack of documentation beyond dev team



Extreme Prioritization & Constant Planning

Aim to deliver **most valuable features first**

Responsive to changes in customer demands

- Techniques
 - ◆ Work on most valuable features first
 - ◆ Continuously revise prioritisation & planning (for each iteration)
 - ◆ Constant feedback from customer

Customer provides **business prio**
Re-prioritization supported by dev process
Early validation **minimizes** need & cost for
major changes

Other criteria suffer, e.g. quality
Instability in dev work
Inadequate architecture and
increased costs
Refactoring requires time and experience



Prototyping & Reviews & Acc Test

Communicate through prototypes and frequent review meetings
Involves customers, developers and testers
Requirements **validation** and **refinement** through feedback

- Techniques
 - End-of-sprint sign-off meeting

Efficient **validation**
Assess **project status**
Trust: Customer, Mgmt
Early **problem identification**

Risks with **evolving prototypes in production**
Unrealistic expectations regarding leadtime
Weak **formal validation, consistency checks**
Dev of acc tests **require access to customers**

Test-Driven Development

Developers **create test before writing new code**

Tests specify expected behaviour of code

Tests **capture complete requirements**
Traces to production code facility **reqts**
changes

Requires **competence in testing,**
requirements understanding and
customer collaboration

Most organizations fail to implement this practice

Summary of Benefits & Challenges of Agile RE

Practices	Benefits	Challenges
Face-to-face communication	<ul style="list-style-type: none">• Customers can steer the project• No time-consuming documentation	<ul style="list-style-type: none">• If no intensive interaction, then bad reqts.• On-site customer representation is difficult
Iterative RE	<ul style="list-style-type: none">• Better relationship with the customer• More understandable reqts	<ul style="list-style-type: none">• Cost & Schedule Estimation• Lack of documentation• Neglect of non-functional requirements
Extreme prioritization	<ul style="list-style-type: none">• Customers provide business reasons• Opportunities for reprioritization.	<ul style="list-style-type: none">• Business value not enough• May lead to instability
Constant planning	<ul style="list-style-type: none">• Minimizes the need for major changes• Cost of addressing a change decreases	<ul style="list-style-type: none">• Early architecture becomes inadequate• Refactoring isn't always obvious
Prototyping	<ul style="list-style-type: none">• Help communicate with customers to validate and refine requirements	<ul style="list-style-type: none">• Risky to deploy prototypes into production• Create unrealistic expectations
Test-driven development	<ul style="list-style-type: none">• Gives traceability that make changes easier	<ul style="list-style-type: none">• Developers unused to test before coding• Requires a thorough understanding of reqts and extensive collaboration between the developer and the customer
Reviews & acceptance tests	<ul style="list-style-type: none">• Help to know if project is on target• Increase customer trust and confidence• Identify problems early• Obtain management support	<ul style="list-style-type: none">• No formal model or verification of reqts• Consistency checking or formal inspections seldom occur.• Difficult if lacking customer access

Pros & Cons of Agile Development

Strengths

- quickly delivers working increments
- avoids unnecessary overhead
- short communication paths
- feedback from early stages used in developing latter stages

Weaknesses

- weak long-term and overall perspective
- weak / missing documentation
- weaker specialist competence
- less structure/guidance for weaker engineers

To do ...

- Read [AGRE], [PROTO1 & 2], [MDRE], [INTDEP], [RP] [OSSRE]
- **Attend Lecture 6:** Product mgmt., Rel plan , MDRE, OSSRE
- **Attend Exercise 3** on Functional requirements [Lauesen chapter 3-5]
- **Hand in Release R1**
- **Book meeting** with supervisor
- **Prepare Lab2** (next week) = Quality requirements (QR) & Release Planning (RP)
 - ◆ Please note: **Preparations** include **reading + working**, take significantly **more time** than for lab1, and is based on attending lectures and exercises
- *Next week*
 - Attend one lecture L7 on Quality Requirements [QUPER, Lau:6-7]:
 - Watch detailed QUPER-video before or after the lecture:
<https://cs.lth.se/krav/quality-requirements/>
 - Attend Exercise 4 where you work on QR in your project
 - Attend Lab 2 (bring preparations)