

Summary session

IN5140

Practical information

- No materials permitted
- Platform: Inspira
- All topics covered in the course is relevant for the exam (including guest lectures)

Questions asked

- Will BPMN models be in the exams?
 - Yes
- How much details do we need to know? Every icon and symbol there is or just the most important ones?
 - Only the most important ones. Lanes, actors, events, activities

How to answer on the exam?

The student must be able to show that they **understand** the curriculum and are able to **apply** it to different cases and **discuss** the results

In general:

- Answers are mostly correct and concise, no irrelevant text
- Answers are well structured
- The student reasoned about the topics and gives rationale
- Content is related to the material in the lectures and slides

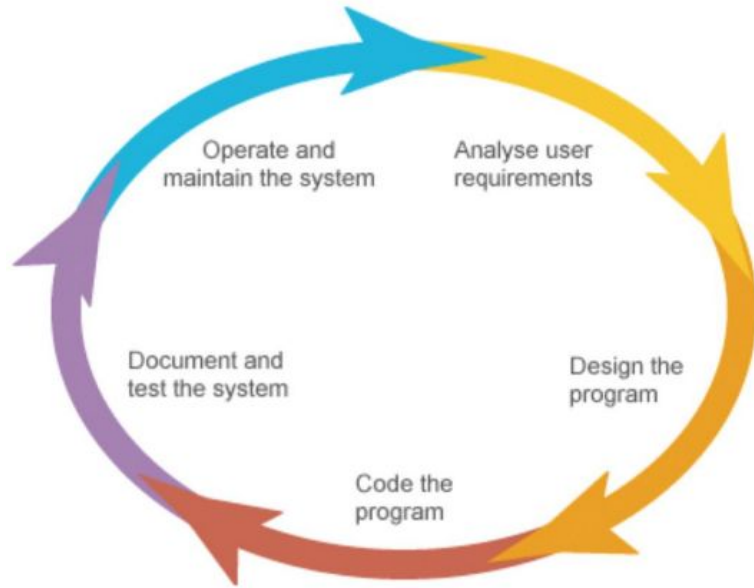
Summary

Slides and figures from group sessions and lectures

Software processes

Systematic approach to SPI

(Software Process Improvement)



- **PLAN** what you want to accomplish over a period of time and what you might do, or need to do, to get there
- **DO** what you planned to do
- **CHECK** the results of what you did to see if the objective was achieved
- **ACT** on the information – standardize or plan for further improvement

Types of maintenance

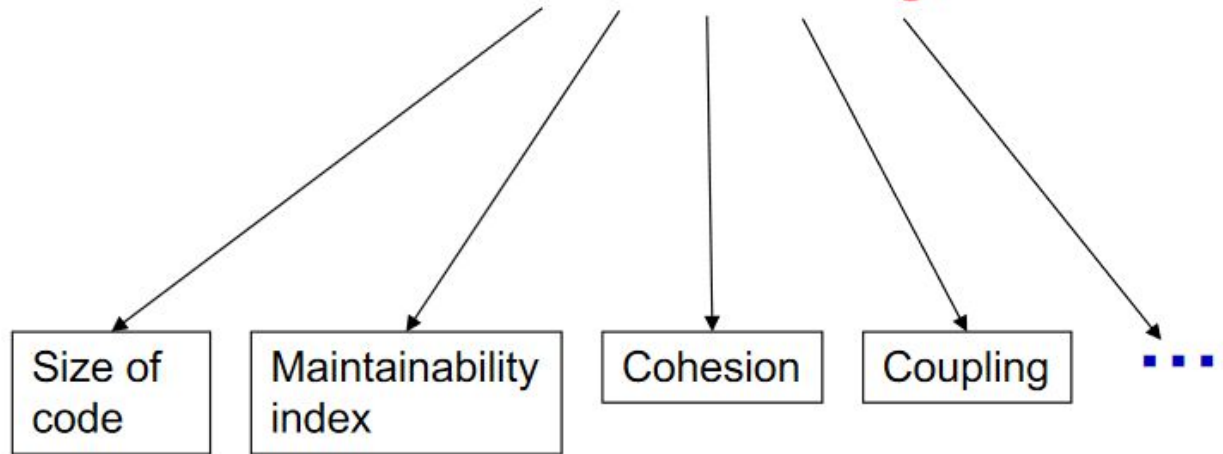
- Corrective maintenance - Fixing bugs
- Preventive maintenance - Improving structure (re-engineering refactoring)
- Adaptive maintenance - Adapt to changing environments
- Perfective maintenance - New functionality

Maintainability

Conceptual level

Maintainability

Operational
(measurable) level

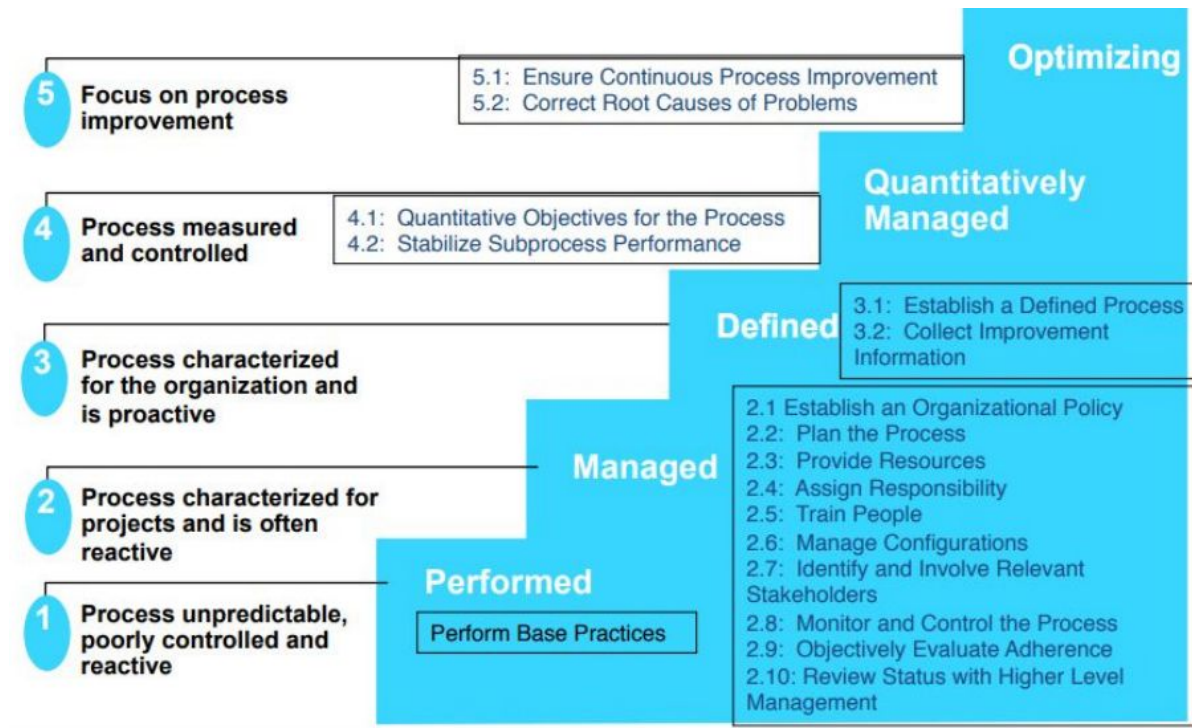


Formal and real process

- Formal process: An abstract representation of a process. The model describes the process from a certain perspective such as formal process is presenting what we say we do or **what we should do**. e.g. "Cooking recipe", "Delivery route".
- Real process: Activities that are carried out in a development project such as execution of processes that describes **what we actually do**.

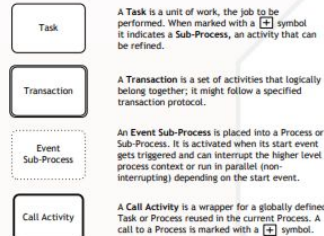
Software maturity levels

CMMI (Capability Maturity Model Integrated)



BPMN

Activities



Activity Markers

Markers indicate execution behavior of activities:



Task Types

Types specify the nature of the action to be performed:



Sequence Flow

defines the execution order of activities.

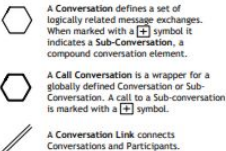
Default Flow

is the default branch to be chosen if all other conditions evaluate to false.

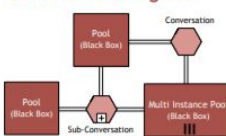
Conditional Flow

has a condition assigned that defines whether or not the flow is used.

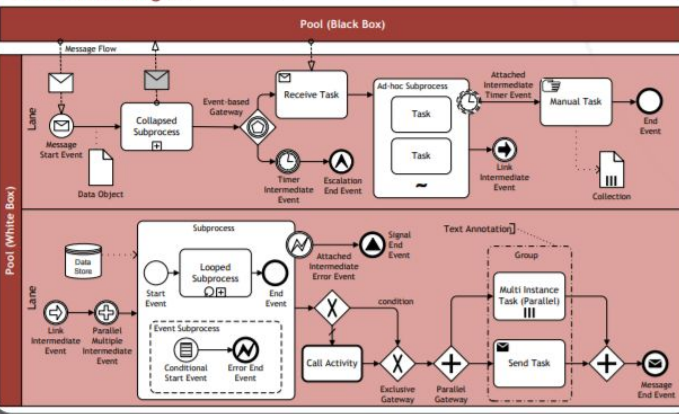
Conversations



Conversation Diagram



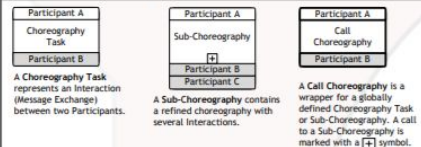
Collaboration Diagram



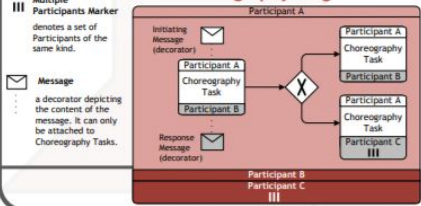
Swimlanes



Choreographies



Choreography Diagram



Events

	Start	Intermediate	End
Standard			
Event Sub-Process Interrupting			
Event Sub-Process Non-Interrupting			
Catching			
Boundary Interrupting			
Boundary Non-Interrupting			
Throwing			
Standard			

None: Untyped events, indicate start point, state changes or final states.

Message: Receiving and sending messages.

Timer: Cyclic timer events, points in time, time spans or timeouts.

Escalation: Escalating to an higher level of responsibility.

Conditional: Reacting to changed business conditions or integrating business rules.

Link: Off-page connectors. Two corresponding link events equal a sequence flow.

Error: Catching or throwing named errors.

Cancel: Reacting to cancelled transactions or triggering cancellation.

Compensation: Handling or triggering compensation.

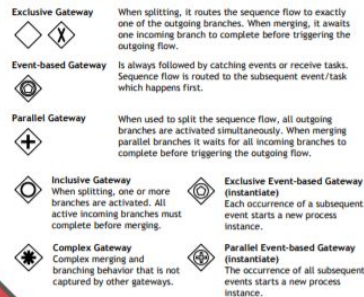
Signal: Signalling across different processes. A signal thrown can be caught multiple times.

Multiple: Catching one out of a set of events. Throwing all events defined.

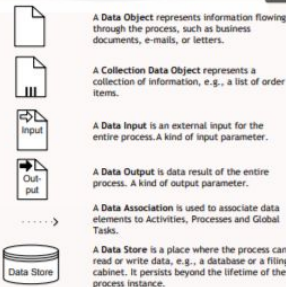
Parallel Multiple: Catching all out of a set of parallel events.

Terminate: Triggering the immediate termination of a process.

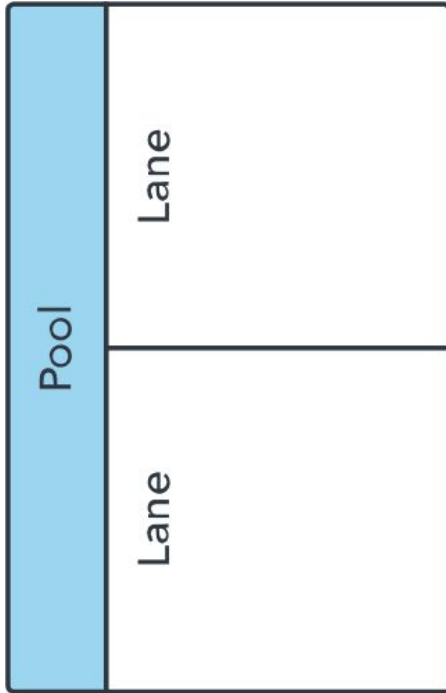
Gateways



Data



Pools and Swimlanes



- **A pool** represents major participants in a process. A different pool may be in a different company or department but still involved in the process.
- **Swimlanes** within a pool show the activities and flow for a certain role or participant, defining who is accountable for what parts of the process.

Events



Start



Intermediate



End

- A trigger that starts, modifies or completes a process.
- They are shown by circles containing other symbols based on event type.

Activities

- A particular activity or task performed by a person or system. It's shown by a rectangle with rounded corners.
- They can become more detailed with for example sub-processes.



Gateways



Exclusive



Event based



Parallel



Inclusive



Exclusive
event based



Complex



Parallel
event based

- Decision point that can adjust the path based on conditions or events. They are shown as diamonds.
- Gateways can also be an empty diamond

Flow



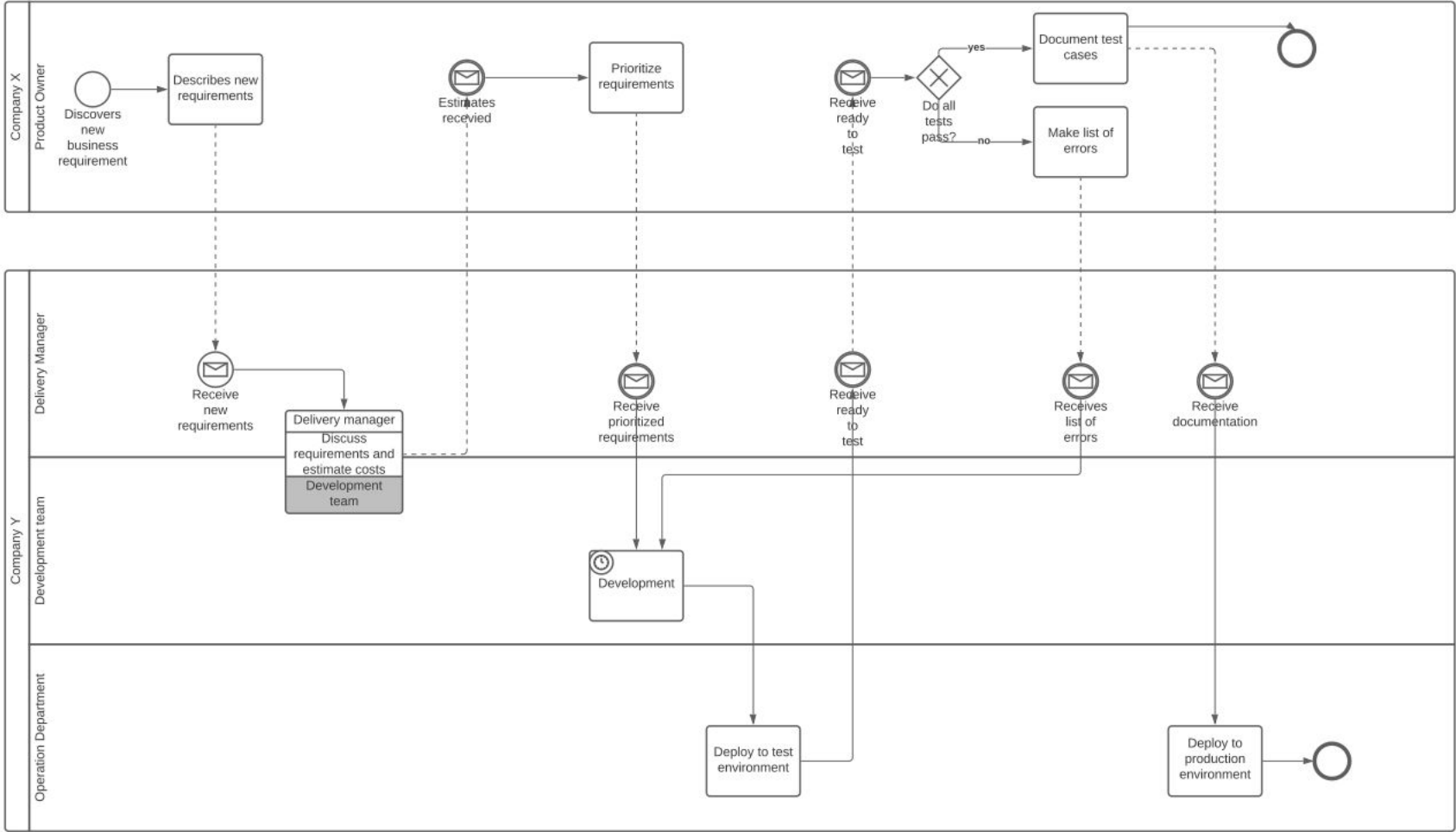
- **Sequence flow:** shows the order of activities to be performed



- **Message flow:** depicts messages that flow across “pools,” or organization boundaries such as departments. It shouldn’t connect events or activities within a pool.

How to engage BPMN

- **Skim through** the textual description
 - Try not to drown in the text
- Note potential **pools**
- Identify the **roles** of each pool
 - Swim lanes
 - What are their tasks
 - What are their dependencies
- Model the **tasks** with metadata
- Check that you are still answering the exercise
- Do not make the BPMN diagram too extensive and complicated



Empirical methods and measurements

Directly or indirectly measurement

- Directly: Measuring that exact variable.
- Indirectly: Measuring the variable through measuring something else.

Objective or subjective

- Objective:
 - Based on facts rather than feelings, opinions, prejudices or interpretations [Merriam-Webster].
- Subjective:
 - Related to the way people experience things in their own mind
 - Based on feelings or opinions rather than facts, modified or affected by personal views, experience or background [Merriam-Webster]

Quantitative or qualitative

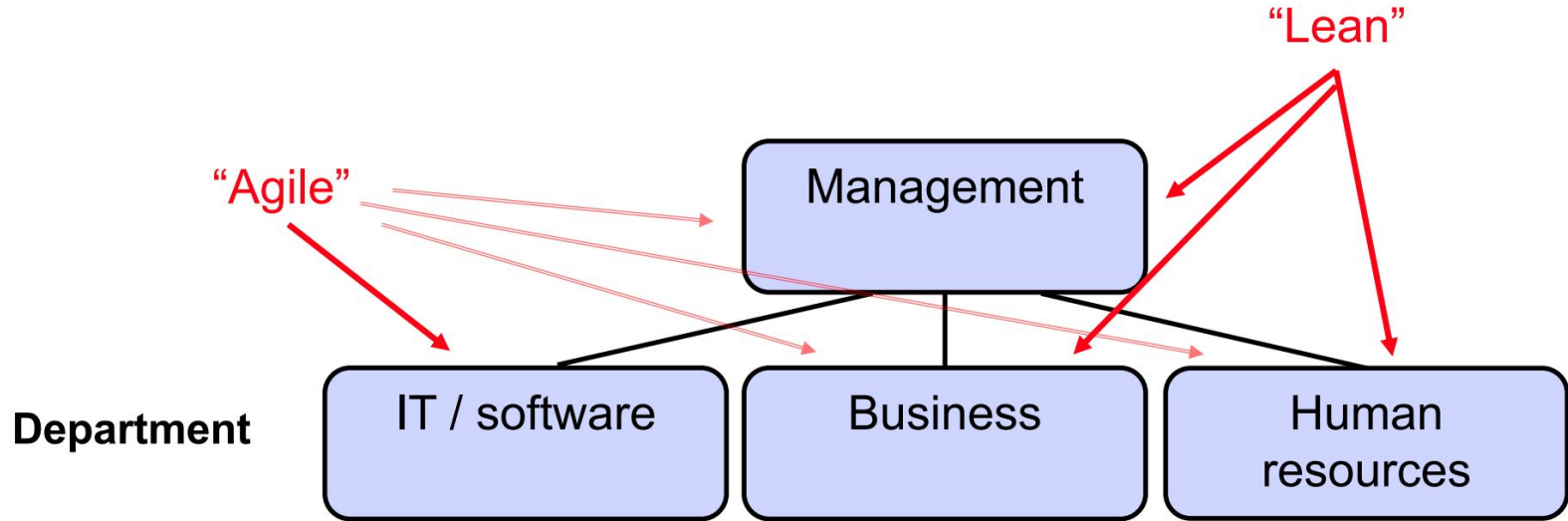
- Quantitative:
 - Data expresses quantity
 - Data expressed as numbers
 - Used in statistics
- Qualitative:
 - Data expresses quality in some sense
 - Data expressed as text, images, audio or video but not numbers
 - Not used in statistics

Nominal, ordinal, interval or ratio scale measurement

- Nominal: Divides the set of objects into categories, with no particular ordering among them. E.g. labeling, classification, defect type.
- Ordinal: Divides the set of entities into categories that are ordered. E.g. ranking, difficulty, failure severity, complexity.
- Interval: Comparing the differences between values is meaningful. E.g. temperature, start and end date of activities.
- Ratio scale: There is a meaningful “zero” value, and ratios between values are meaningful. E.g. Length, weight, lines of code, number of errors.

Lean

Lean versus Agile

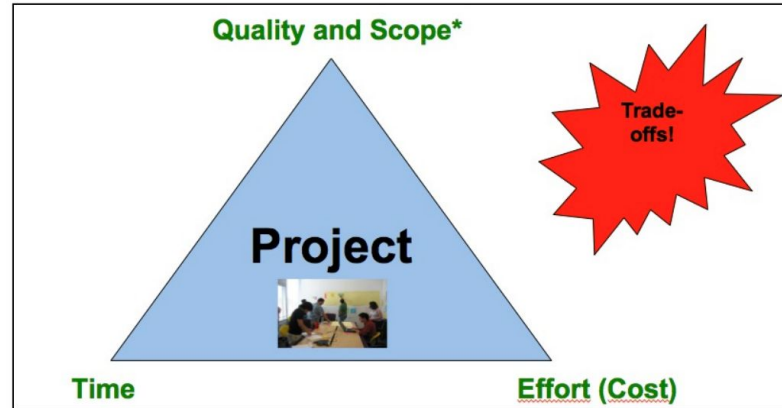


- Some agile principles come from Lean (Lean predated agile), but Lean is not an “agile method” (as indicated by the structure of Ch. Ch. 9 in the book), but “rather a philosophy made of a set of general observations”, p. 135

Purpose of Lean

- All parts of the organization should optimize their processes to achieve a common goal: **To create value for the customer**
- Value = what the customer wants and is willing to pay for

Customer value →



Project/Magic/Iron triangle

Lean production

- “The Japanese school”, primarily Toyota
 - If failure discovered, stop the assembly line and find the reason for the failure, instead of only collecting and fixing the failures (**root cause analysis** and **root cause fix**)
 - “**Just-in-time**” (JIT) principle: don’t produce before somebody demands it
 - Tempo in production determined by **pull** from customer or next element in the production chain rather than push internally to produce as much as possible
 - **Removal of temporary storage**
 - **Component-based production** (same chassis, bumper, etc. on different car models). Production geared towards the customer, components can be outsourced to third-party vendors
 - **Continuous learning** and improvement (Kaizen)
- Result: few failures and fast production
- The most productive factories spent least resources on management and administration (“lean management”)



Lean principles (some of them)

- Satisfying the customer
- Flow
- Visualization
- Avoiding waste
- Supporting changes

Satisfying the customer

- Create value for the customer (what the customer is willing to pay for)
- Produce what the customer **actually** want
- Produce high quality and valuable software

Flow

- How items or people in a process move from the first step to the last
- Move as quick as possible, but without any risk to quality and customer satisfaction
- Optimized and planned, stable and with minimal waste

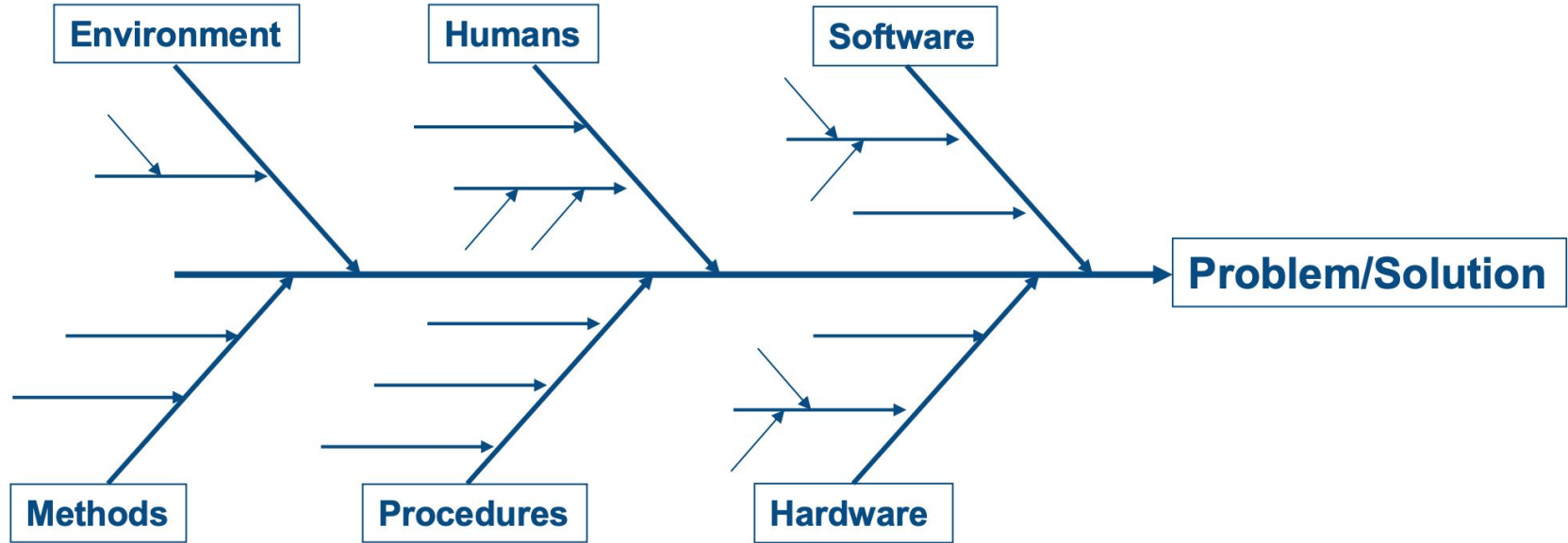
Visualization

- Gives an overview which is helpful for:
 - common understanding for all stakeholders
 - planning
- Examples: Obeya room, Kanban board

Waste

- Everything that requires resources and does not give value to the customer (time, work effort, space, money, equipment)
- Waste increase costs, not value
- Minimal waste ensures flow
- Unnecessary features, excessive documentation, partly done work, duplication of data, etc.

Fishbone diagram



Large-Scale Agile

Main challenges

Main challenges/issues when scaling up Agile?

Challenge type

Change resistance 16 (38%)

General resistance to change
Skepticism towards the new way of working
Top down mandate creates resistance
Management unwilling to change

Lack of investment 13 (31%)

Lack of coaching
Lack of training
Too high workload
Old commitments kept
Challenges in rearranging physical spaces

Agile difficult to implement 20 (48%)

Misunderstanding agile concepts
Lack of guidance from literature
Agile customized poorly
Reverting to the old way of working
Excessive enthusiasm

Coordination challenges in multi-team environment 13 (31%)

Interfacing between teams difficult
Autonomous team model challenging
Global distribution challenges
Achieving technical consistency

Different approaches emerge in a multi-team environment 9 (21%)

Interpretation of agile differs between teams
Using old and new approaches side by side

Hierarchical management and organizational boundaries 14 (33%)

Middle managers' role in agile unclear
Management in waterfall mode
Keeping the old bureaucracy
Internal silos kept

Requirements engineering challenges 16 (38%)

High-level requirements management largely missing in agile
Requirement refinement challenging
Creating and estimating user stories hard
Gap between long and short term planning

Quality assurance challenges 6 (14%)

Accommodating non-functional testing
Lack of automated testing
Requirements ambiguity affects QA

Integrating non-development functions 18 (43%)

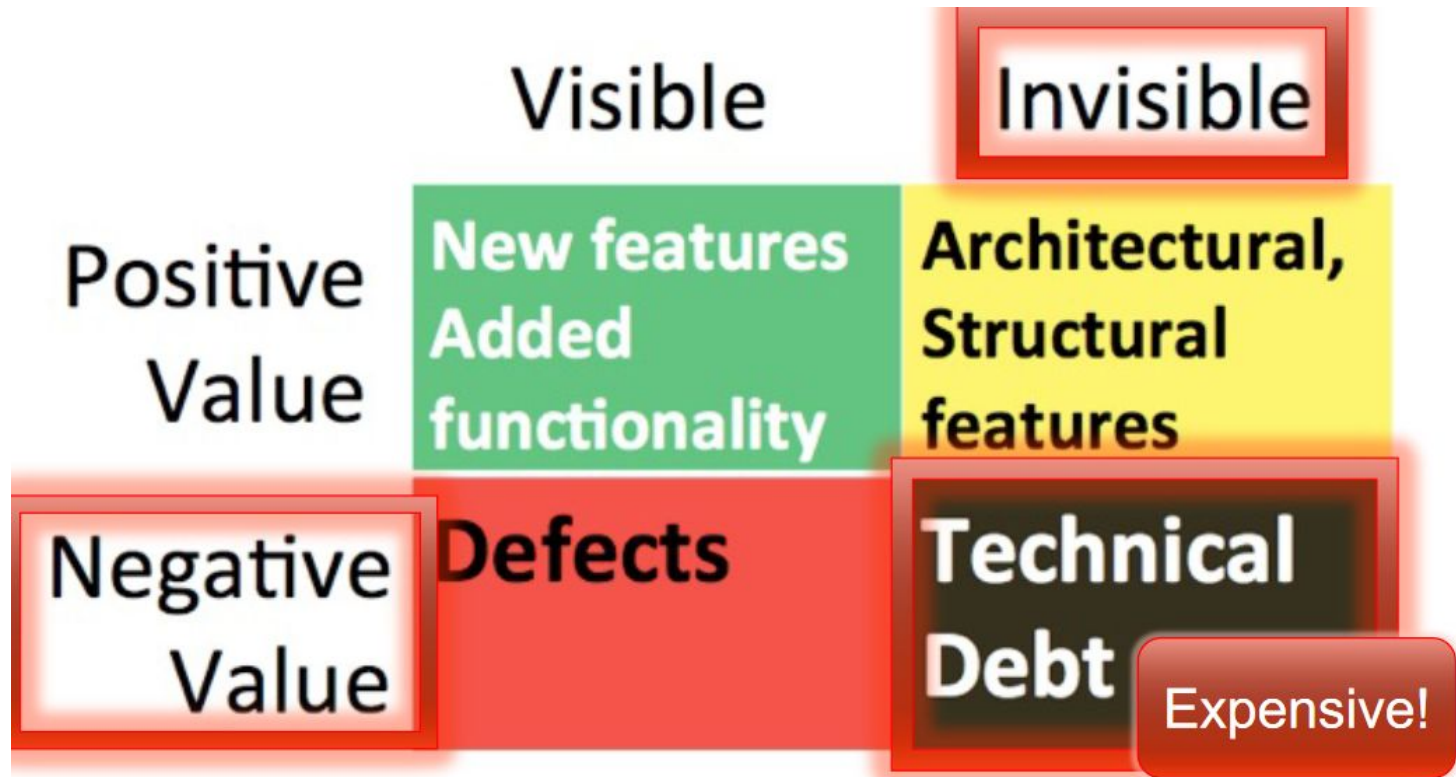
Other functions unwilling to change
Challenges in adjusting to incremental delivery pace
Challenges in adjusting product launch activities
Rewarding model not teamwork centric

INTEF

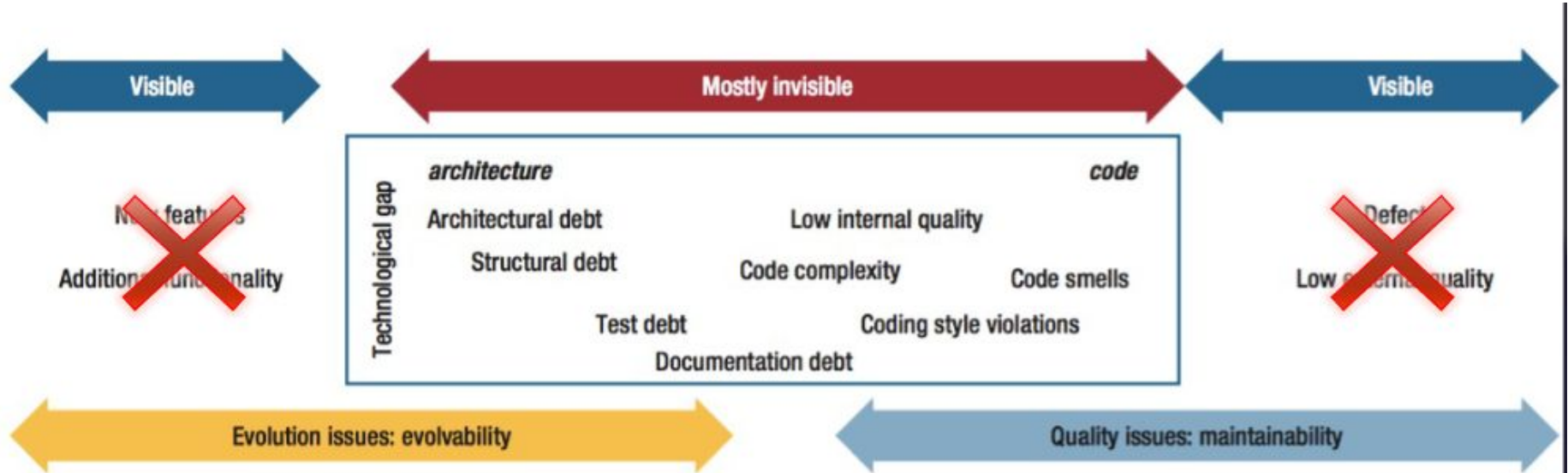
(Dikert et al., 2016)

**Technical-, social- and process
debt**

Technical debt

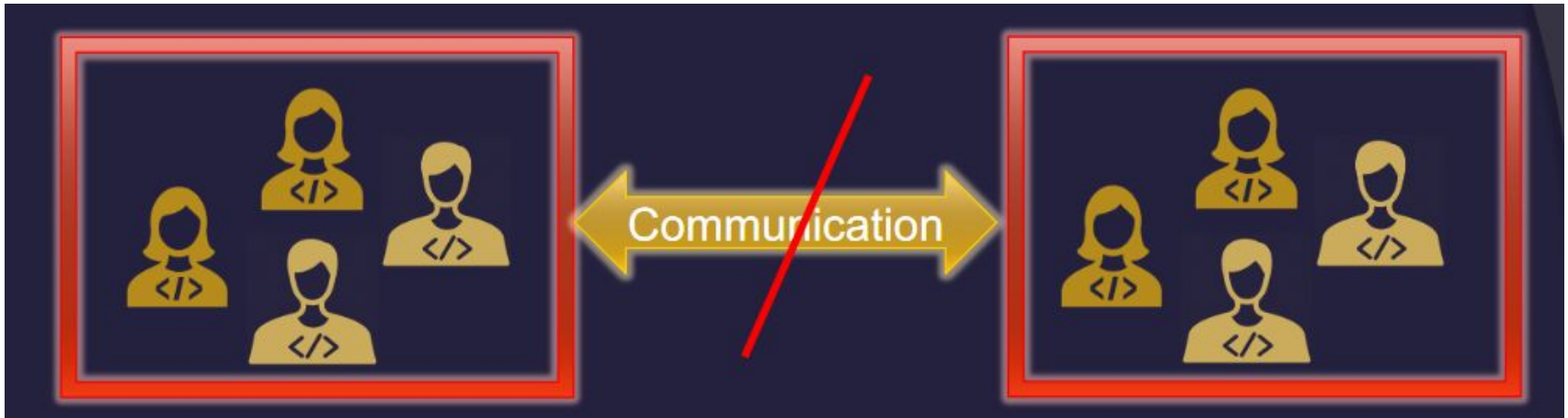


Technical debt



Social debt

- “The presence of sub-optimality in the development community, which causes a negative effect”



Process debt

- Sub-optimal process design
- Divergence from optimal formal process
- Deficiencies in the infrastructure that might be beneficial in the short term
- Negative long term effects

How do we fix technical debt?

Proactive approaches:

- Education
- Culture
- Organization
- Process
- Guidelines
- Visualization

Continuous approaches:

- Semi-automatic identifications
- Code reviews
- Retrospectives
- Technical leadership
- Dedicated refactoring sprints

Reactive approaches:

- Impact map
- Roadmap evaluation
- Resources to remove TD
- TD information used in planning and budget

Quality Assurance

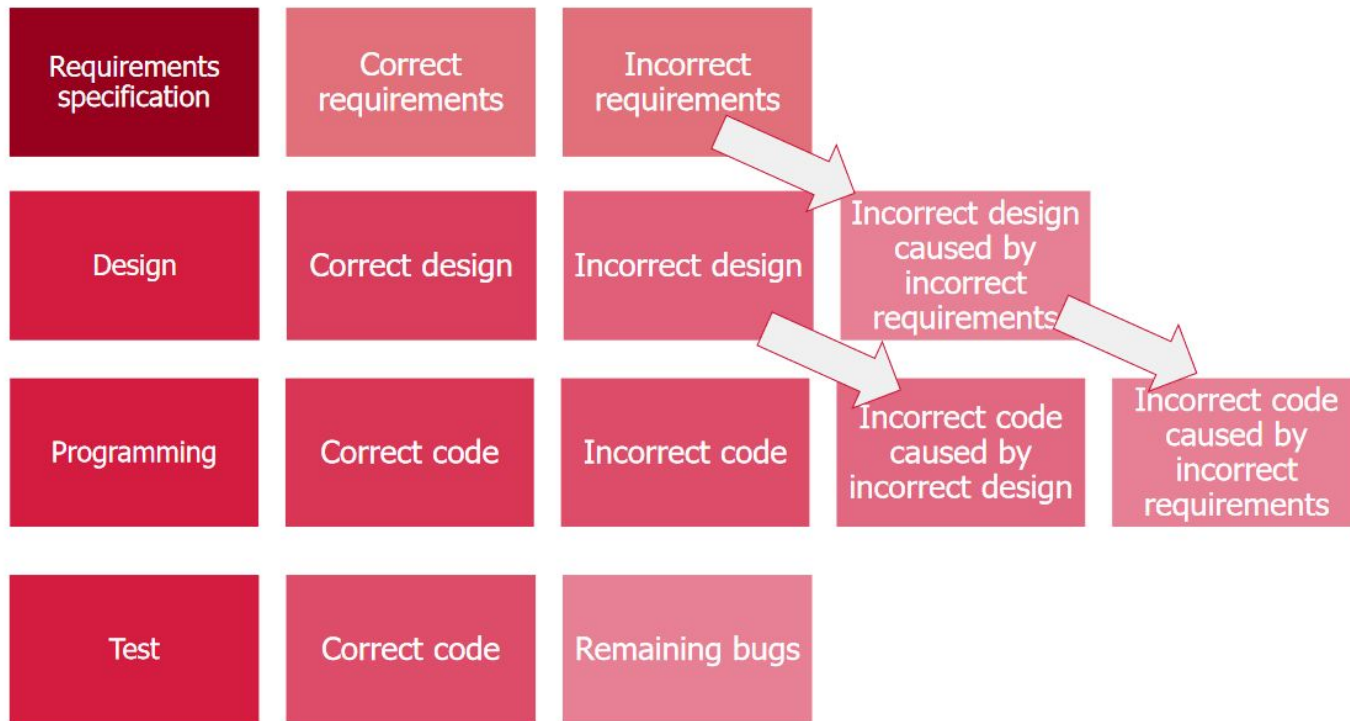
Quality

Could for example be:

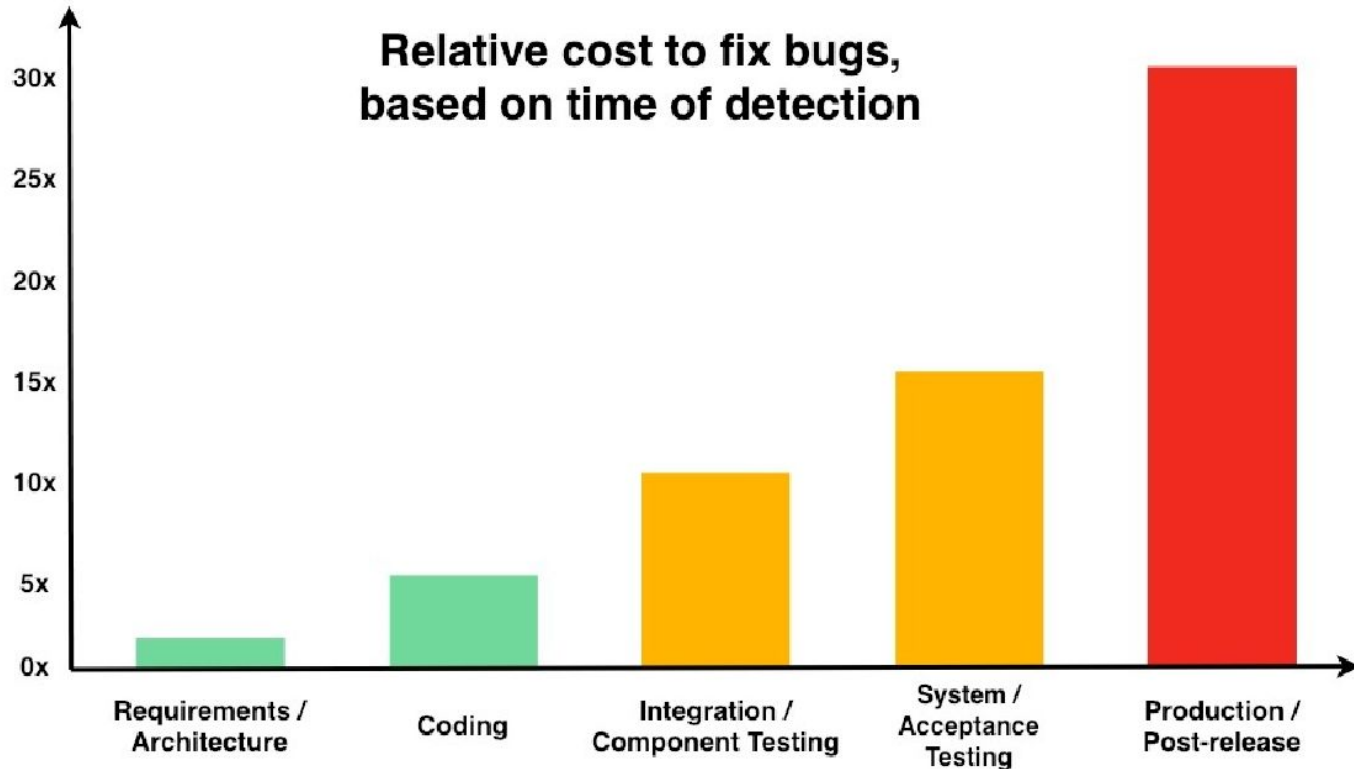
- Correctness
- Maintainability
- Reliability
- Robustness



When do bugs occur?



Cost of bugs



DevOps



Agile practices

Agile

Agile is a structured and iterative approach to project management and product development.

It recognizes the volatility of product development, and provides a methodology for self-organizing teams to respond to change without going off the rails.

Today, agile is hardly a competitive advantage. No one has the luxury to develop a product for years or even months in a black box. This means it's more important than ever to get it right.

Kanban

Kanban is all about visualizing your work, limiting work in progress, and maximizing efficiency (or flow).

Kanban teams focus on reducing the time a project takes (or user story) from start to finish.

They do this by using a kanban board and continuously improving their flow of work.

Scrum

Scrum teams commit to completing an increment of work, which is potentially shippable, through set intervals called sprints.

Their goal is to create learning loops to quickly gather and integrate customer feedback.

Scrum teams adopt specific roles, create special artifacts, and hold regular ceremonies to keep things moving forward.

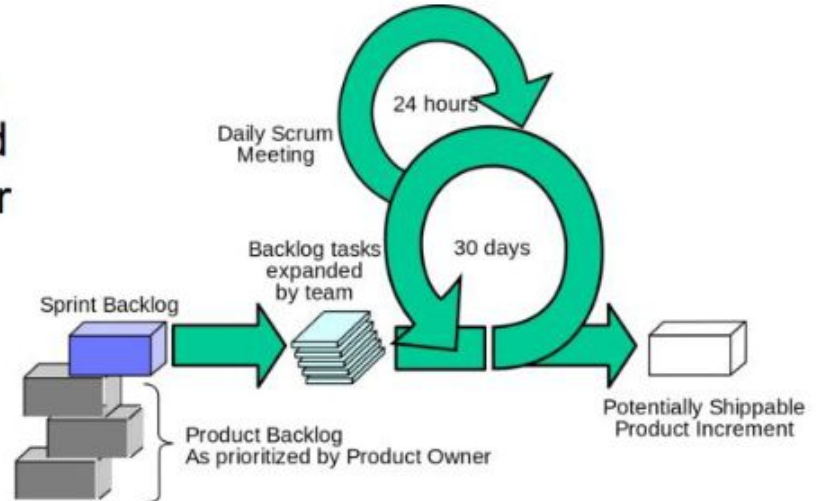
Difference

	Scrum	Kanban
Origin	Software development	Lean manufacturing
Ideology	Learn through experiences, self-organize and prioritize, and reflect on wins and losses to continuously improve.	Use visuals to improve work-in-progress
Cadence	Regular, fixed-length sprints (i.e. two weeks)	Continuous flow
Practices	Sprint planning, sprint, daily scrum, sprint review, sprint retrospective	Visualize the flow of work, limit work-in-progress, manage flow, incorporate feedback loops

Agile development

What is Agile software development?

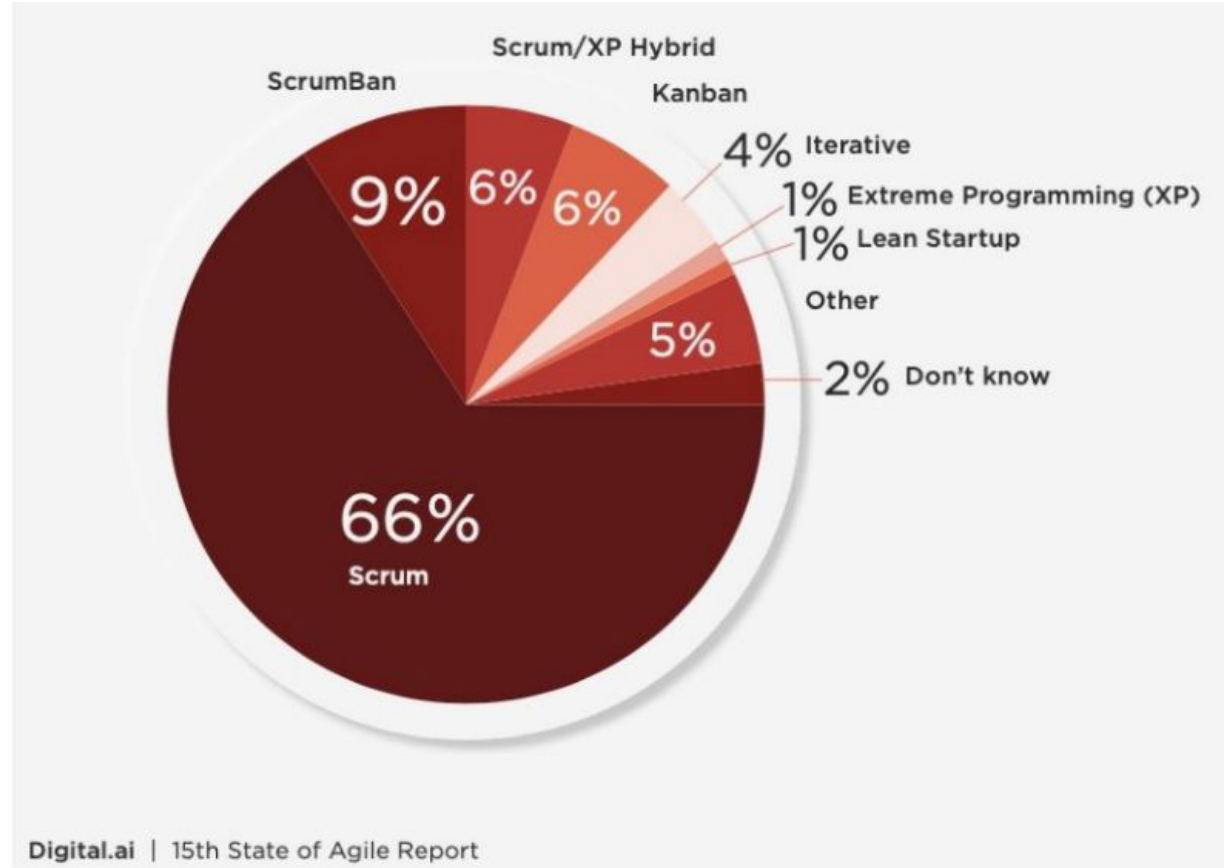
- «**Agile**» refers to a set of **iterative** approaches to software delivery that builds software **incrementally** from the start of the project, instead of trying to deliver it all at once near the end.
 - An **iteration** is a cycle in the development
 - Popularly known as a sprint
 - An **increment** is a subset of requirements for the system
 - For example, backlog items, user stories



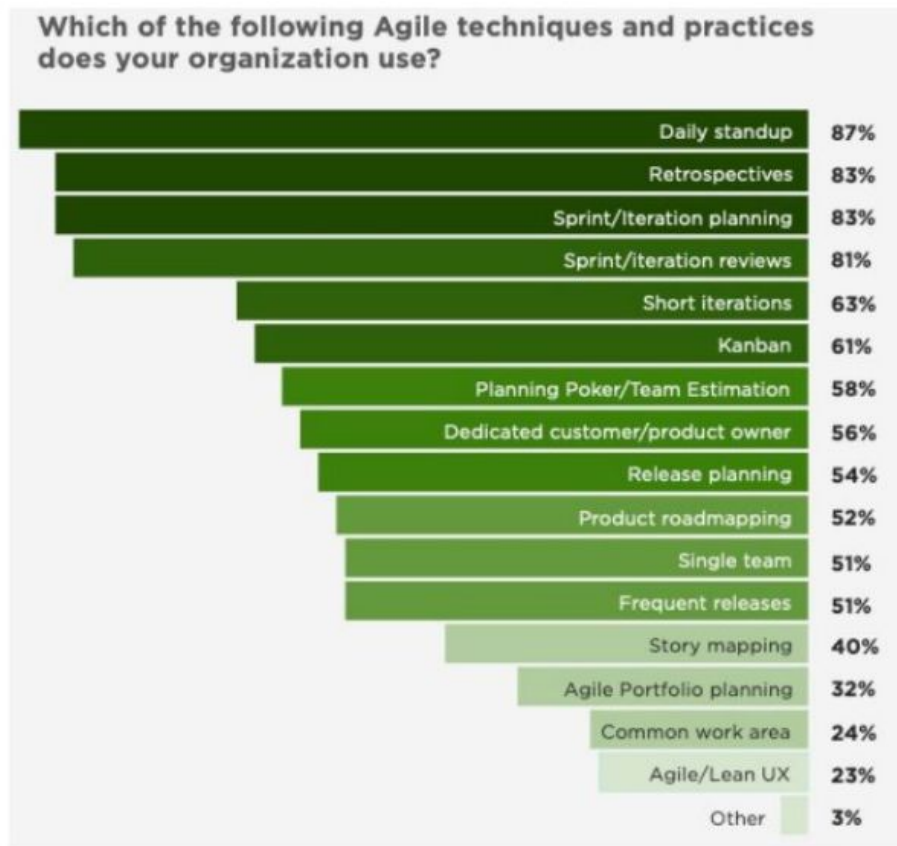
Agile methodologies

Which agile methodology do you think most teams use?

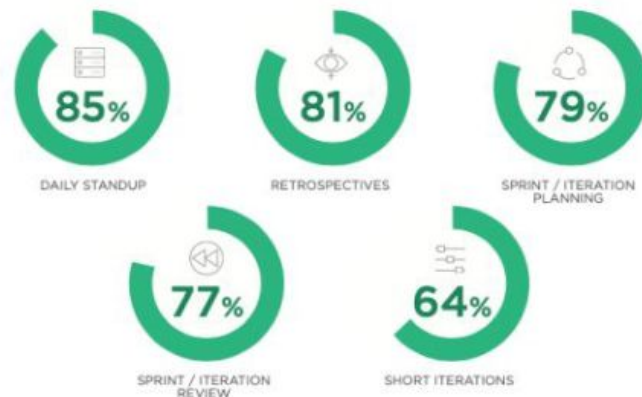
Results from a large international survey
→



Popular agile practices



TOP 5 AGILE TECHNIQUES



Digital.ai, "Annual State of Agile Report"

<https://digital.ai/resource-center/analyst-reports/state-of-agile-report/>

Questions?

Good luck on the exam! :)