Mentimeter

IN5140

Smart processes and agile methods in Software Engineering

Group session 4

Mentimeter

Agenda

First hour:

- Project Information
- Quick recap

Second hour:

- Previous Exam Questions

Project milestones throughout IN5140

What	Due date	% of total score
Deliverable 1 – Brochure	5th of Oct, 23:59	passed/not passed
Deliverable 2 – Retrospectives and BPMN on project	TBU	passed/not passed
Deliverable 3 – Improved BPMN and SPI suggestions on project	TBU	passed/not passed
Project Presentations	TBU	passed/not passed

- Submit deliverables here (PDF): https://devilry.ifi.uio.no/
- Feedback within 2-3 weeks

The Booklet

Contents:

- First page: What is the product? (text, images, quotes)
- **Prioritization list (ordered list of features)**
- Needed tech stacks
- Needed employees (number and roles)
- **Minimum 1 page per feature mockup**
- Gantt chart

Mentimeter

Things to remember

For Product Owners:

- Check the cross-group requirements document.
- You will receive more requirements from TAs underway.

For Groups:

- Get in contact with your group and PO if you haven't already.
- Remember to submit a page on Devilry detailing your group's participants and their contributions to the group and the booklet.

For Everyone:

- Reach out to the TAs if you have any issues.

Mentimeter

Questions?

Speak out loud, or type in chat

Mentimeter

Recap from the lecture

1

Mentimeter

Process Improvement, Lean, and Agile

- How are process improvement, Lean, and Agile connected?

Lean

- What is the purpose of Lean?
- How do the magic triangle and systems thinking explain this?
- Which Lean practice (e.g., root cause analysis, JIT, no storage, component production, continuous learning) do you think is most powerful? why?

Mentimeter

Lean : Value, Waste, and Flow

- How do we know if something is value or waste? What is the difference between value and waste?
- What is an example of “necessary but non-value added” work?
- What does “flow” mean in Lean?

Mentimeter

Agile

What are the 12 Agile principles?

- Which ones are organizational?
- Which ones are technical?

Mentimeter

Kanban vs Scrum

How do Kanban and scrum deal with WIP (Work in Progress)?

Mentimeter

Agile vs Lean

What do they have in common?

What are their differences?

Mentimeter

Communication

What are two effective ways to communicate with customers?

Mentimeter

Exercise

1

Mentimeter

Exercise 1

Discuss and describe the following overall lean principles:

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

1

Discuss and describe the following overall lean principles

Satisfying the customer is important because lean is designed around producing minimal waste, this means that understanding the customer is important to avoid wasteful work, costs/processes.

Satisfying the customer is about that the customer gets what he wants and reduce waste Flow is about continuous delivery and learning and reduce bottlenecks

Flow in lean is related to waste management, the goal is to achieve a continuous workflow without brakes, delays, pauses etc. This flow contributes to produce minimal waste.

Visualization is to see how you want the process to go

Flow is about ensuring that the work process is not disturbed by hindarances. If a problem occurs, it should be fixed right away. Similar to continous delivery in agile

Supporting changes in Lean means helping people adopt, maintain, and improve new processes by giving them leadership backing, training, involvement, and follow-up, so that improvements last.

Satisfying the customer: Only providing what the customer is willing to pay for Flow: continous delivery, no bottlenecks in assembly line Visualization: seeing the process Waste: customer no want

Visualization is important to understand the tasks at hand, it's visualised with graphs, kanban board etc. Was hand written before tech age.

3 17

Mentimeter

Discuss and describe the following overall lean principles

Avoiding waste is about avoiding unnecessary things and only things the costumer only wants

make a product that the customer wants to pay for movement of work in lean, minimize waste, delays visualize the process avoid uneccasery production accept change, dont stick religiously to the plan

A good flow in the work process is crucial for steady progress. Do not do too many tasks simultaneously.

Satisfying the customer - making something the customer will pay for Flow - fix problems as they occur to ensure a steady stream of development Visualisation - models to see how the process should go

Avoiding waste

1. What the customer actually needs and gives them value 2. About improving the processes and delivering when needed 3. idk 4. Don't make things the customer doesn't need 5. Always aim to improve

Avoiding waste: only work on what has been requested, remove unnecessary meetings

A good flow means features moves smoothly from idea to working software without sitting in queues, waiting on approvals, or getting stuck between diifferent teams or processes

Involves maximizing value to the custumer through high quality and JIT Flow involves ensuring that the entire process/workplace goes smootly as a interuption can delay all other processes. Out of spac

3 17

Mentimeter

Discuss and describe the following overall lean principles

Supporting changes is about always learn and improve

Satisfying the customer = customer focused developing with just in time focus, Flow = good work flow, Visualization = kanban board for easier tracking, Avoiding waste = no unnecessarily work

Base the work on customer needs not the employees Lean flow focus on elements needed to deliver value to customer Identify and remove waste task with no value Learn from mistakes or of experience

Avoiding waste means to only produce value for the customer.

Avoiding waste - do not spend time and resources on things that are not of value for the customer Supporting changes - aim to improve processes by enabling changes

Satisying the customer: , we cater to what the customer need Flow: unsure Visualisation unsure Avoiding waste: make sure all the resources spent give value Supporting change: be adaptable

Avoiding waste is about reducing processes that requires resources and does not give value to the customer

Supporting changes is important for the core principle of continous learning.

3

17

Mentimeter

Discuss and describe the following overall lean principles

deliver value, smooth development, visible process development to improve problem solving, non value activities save time, adapt quickly to encourage change

Giving the customer what they ask for. Working at sustainable pace without delays Make the work visible so it is easier to understand Eliminate unnecessary stuff that does not add value

- Avoiding waste
- Supporting changes

3

17

Exercise 1

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

Mentimeter

Exercise 1

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

1

Exercise 1

Visualization:

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

Exercise 1

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.

Mentimeter

Exercise 1

Supporting change:

- Welcome changes
- Accommodate customers' need
- Competitiveness

Exercise 2

- a) What does the Just-in-Time principle mean?
- b) What are the advantages of following this principle? Are there any risks?

What does Just In Time (JIT) principle mean?

Producing something when there is a demand.

Make decisions as late as possible

Dont make something before you need it

Make/do something as late as possible

Only develop things that the customer wants

JIT mean that something should be produced when the cystomer requests it/when it is needed

Supply on demand instead of producing before selling to avoid storage expenses.

It's about delivering What is needed when it's needed. The rate of production should match the rate of demand

14

What does Just In Time (JIT) principle mean?

It means that you do something only when it is needed

To make changes as late as possible and that you do what is actually demanded

JIT means to not build anything until someone needs it. Let the customer demand "pull" work through the system instead of pushing to produce as much as possible

Deliver value when needed

JIT is when you create/make something the moment it is ordered this way you reduce temporary storage and secure high quality goods.

Reduce waste

1.Reduces waste 2. Can increase time before customers get their product (e.g car), customers might choose immediately available options

Mentimeter

JIT: Make decisions as late as possible, that is, not before they are demanded

Any advantages and risks come with JIT?

Delays can make the customer unsatisfied since it takes more time than expected.

Advantages reduce waste because you only do what is demanded

Time and resource efficient

Advantage: little/no storage cost

high quality creation, but a lot of requests might pose problems to this concept as there will be too big of a demand?

not stockpiling product that doesnt get used, risking demand beeing higher than what is possible to produce

Risk: Longer supply time

Reduces waste, but may lead to delays

1

13

Any advantages and risks come with JIT?

Reduces waste Not having enough time to deliver what is requested

Risk is delays and costumer not happy with product

Advantage: reduce waste, straightforward Risk: unexpected problems occur

Reduces unnecessary costs. Could lead to not enough being produced if demand suddenly spikes

Pro: Reduce waste. Con: customer doesnt know what they want. So you just wait and thats waste

avoid building features that become unnecessary when requirements change. Waiting too late to make decisions can leave insufficient time for proper implementation when the need finally arises

Less risk of producing waste but is not a proactive way of developing

Bad planning and bad requirements so they don't now exactly what to do

Mentimeter

Advantages with JIT:

- → Avoid waste (i.e. produce only what you need)
- → Focus only on the current task
- → Contributes to flow

Mentimeter

Potensial risks with JIT:

- → Unforeseen influxes can back up order flows
- → Planning required
- → Rushes can cause unsatisfactory results

Mentimeter

Fishbone diagram

Also called Ishikawa diagram...

- A technique for root cause analysis
- Indicates relationship between effects and their possible causes

If programmers are working slower than expected, what could be the possible causes? “Which causes do you think is the most critical to solve?

They don’t now what they are supposed to do

motivation

Confusion about task(s)

Demotivation

Burnout

Too much work in progress

Bad planning and requirements

Poor planning

14

Mentimeter

If programmers are working slower than expected, what could be the possible causes? “Which causes do you think is the most critical to solve?

- Don’t know what or how to do it
- Task switching, too much multitasking reduces efficiency.
- Working on multiple projects at the same time, context switching?
- Too many/few tasks
- Unclear requirements
- Unsustainable pace
- Too big task
- Too many tasks at the same time

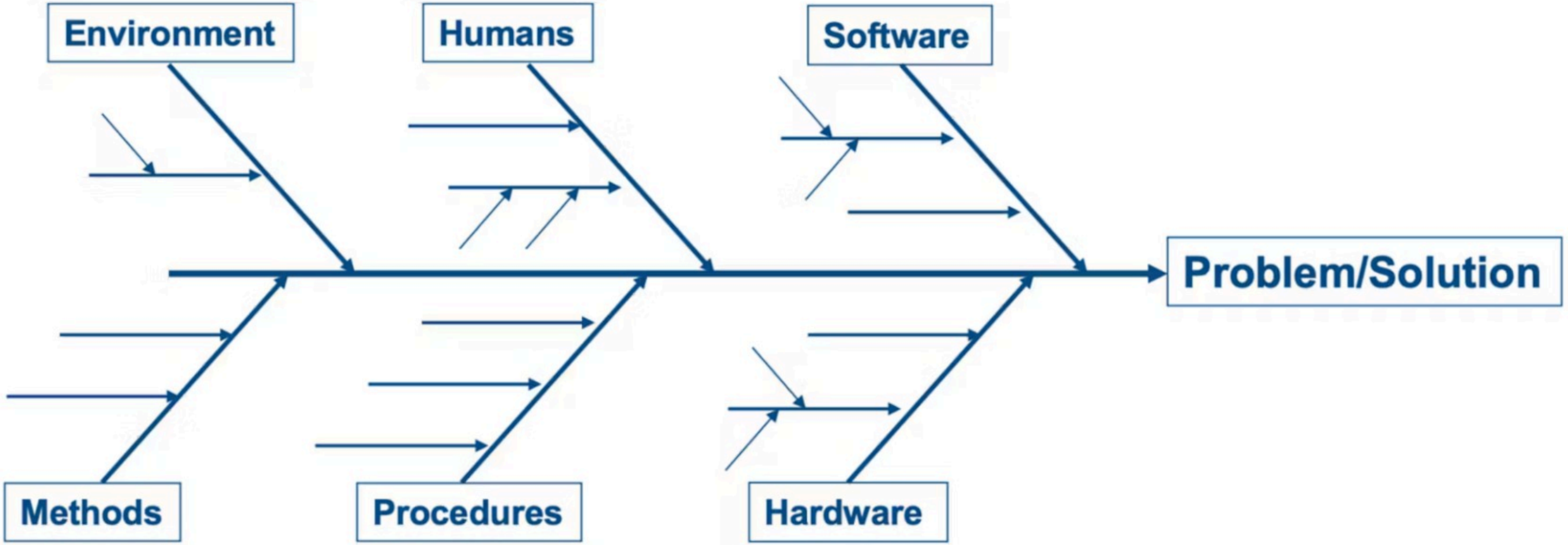
14

If programmers are working slower than expected, what could be the possible causes? “Which causes do you think is the most critical to solve?

Disagreememt

Mentimeter

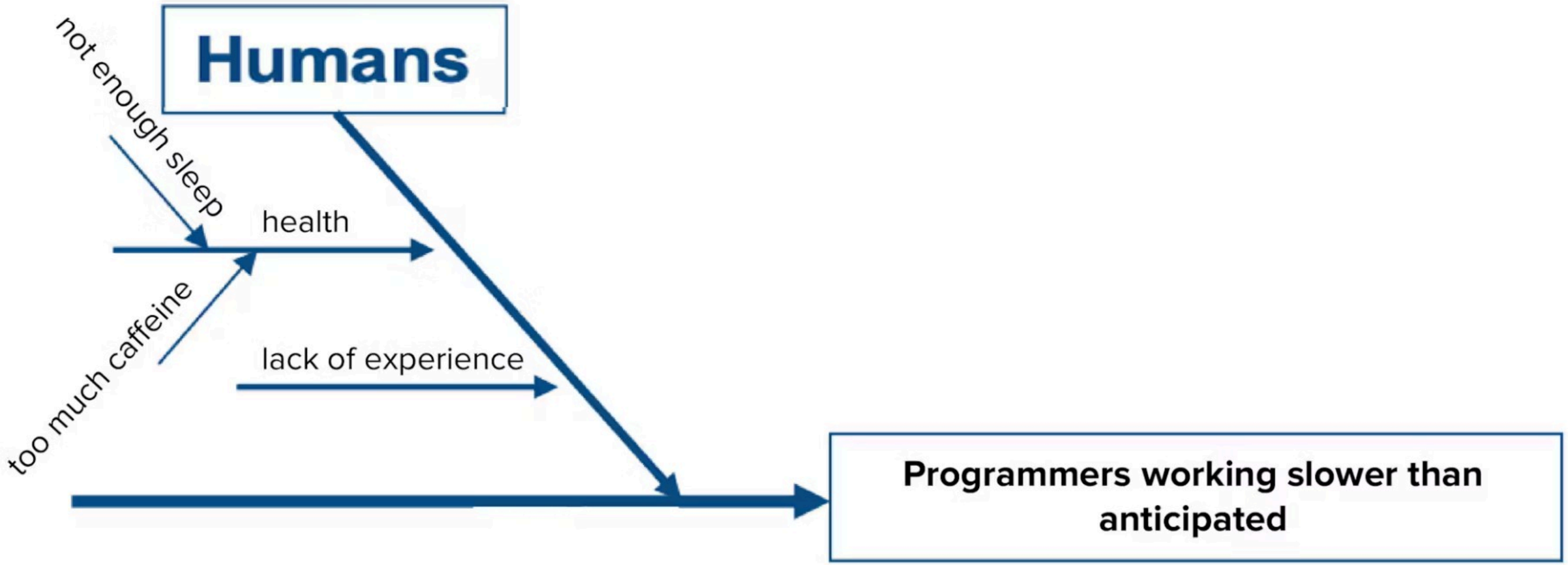
Fishbone diagram



1

Mentimeter

Fishbone diagram



Mentimeter

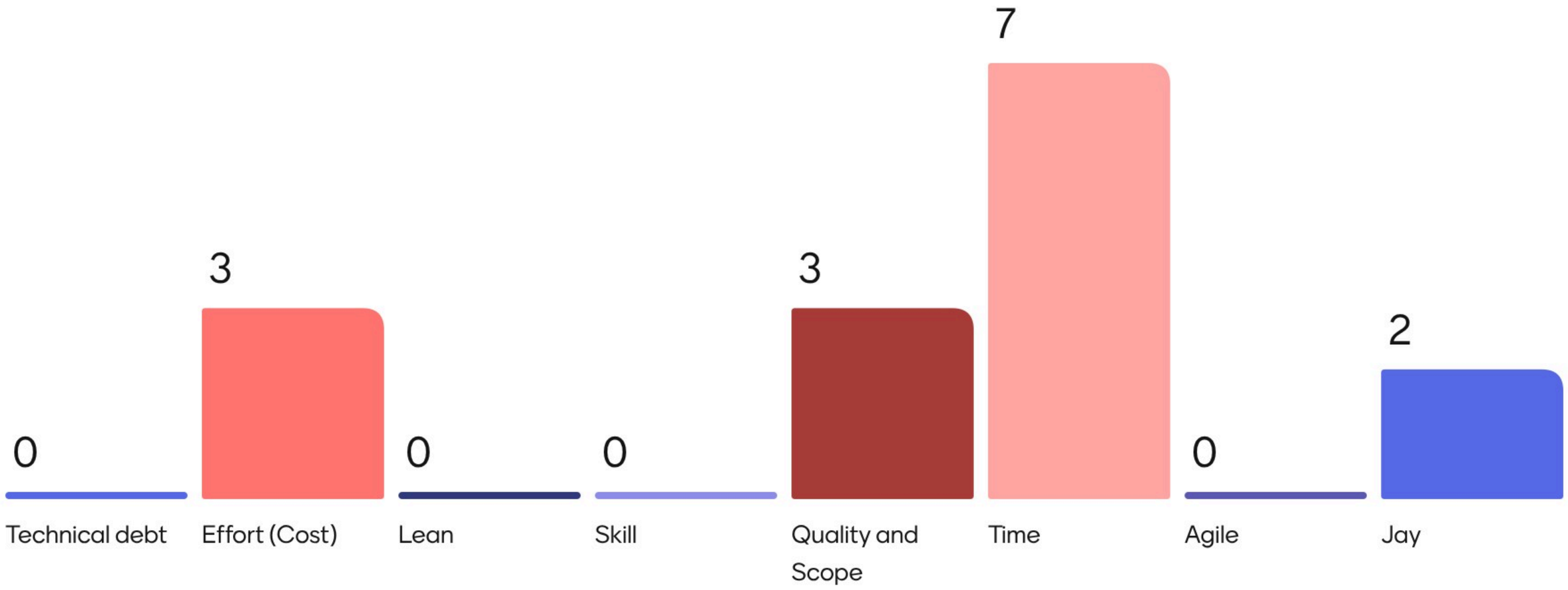
15 minute break

Mentimeter

Previous Exam Questions

Mentimeter

[2022]What are the three dimentions of the Magic Triangle? Select Three

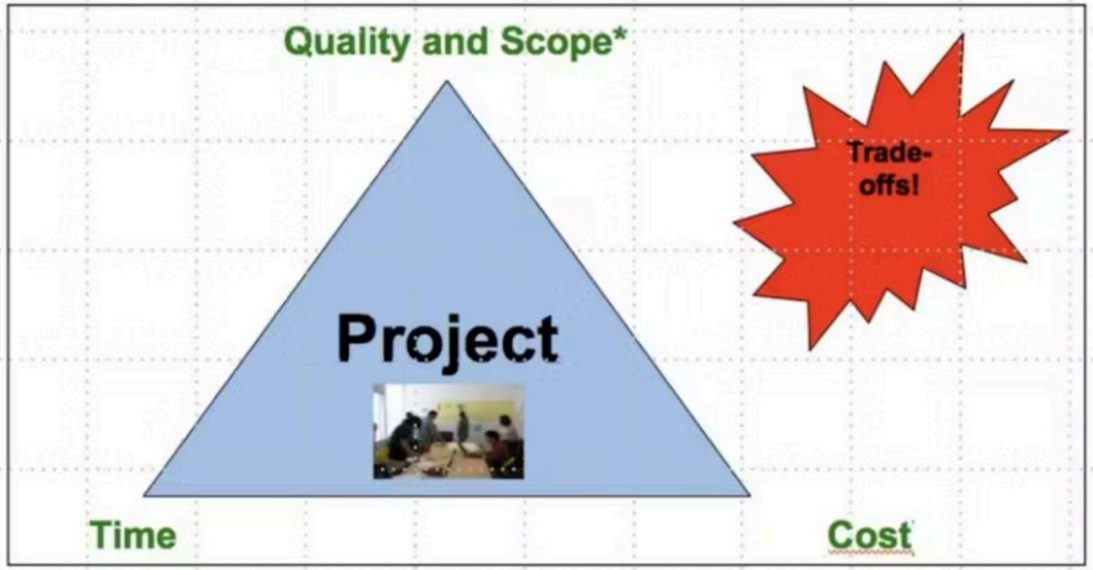


4

15

Mentimeter

[2022]
What are the three dimensions of the Magic Triangle?



Project/Magic/Iron triangle

3

Mentimeter

[2017]Which types of waste (as defined in Lean) are you likely enter in Scrum?

waste of time

Time

Unwanted features due to bad backlog prioritization

Waste of time

Waiting time?

configuring work environments (needed waste). meetings and syncs are also a type of waste as it does not produce anything for the costumer, but it is needed to create a good product

Can equired changes and bad planning so can be much waste on doing unnecessary things and make changes

Too much documentation or unnecessary meetings

11

Mentimeter

[2017]Which types of waste (as defined in Lean) are you likely enter in Scrum?

Unnecessary meetings can waste time

Non-value Added Work, for example infrastructure tasks

Too many meetings, in some Scrum teams daily standups every day is unnecessary.

11

Mentimeter

[2017] Which types of waste (as defined in Lean) are you likely to encounter in Scrum?

- Wishful thinking
  - Focusing on features and functionality which potentially can be useful in the future for customer → can be regarded as non-added value, and therefore waste.
- Overengineering
  - Investing too much time into architecture and design of software product in the beginning is wasteful
- Unfinished work at the end of a sprint
  - Can cause team members to be idle
- Too much planning and documentation for a product which does not exist yet is also considered to be waste.

1

[2020]

Viking Unicorn AS is experiencing several challenges in their software development process. Using principles and techniques from the lecture, propose an approach to improve Viking Unicorn AS's current process.

Your answer should 1)suggest at least one learn technique that could be applied. Explain how this technique would help solve at least two of the challenges above.

Rigid Architecture – Architecture is decided upfront and cannot easily be adapted.

Architect–Developer Conflict – Lack of trust and poor collaboration between architects and developers.

Changing Requirements – New requirements from the customer come continuously, but teams struggle to integrate them.

Late Testing – Testing happens only before release, leading to many late-discovered issues.

Technical Debt – Bugs and quality problems accumulate because there is no time to fix them.

Low Productivity – Teams spend more time fixing bugs than building new features.

Mentimeter

Suggest one lean technique that could be applied and how this technique would solve one of the addressed challenges

Backlog so that everyone know what the requirements are and know what hey will do

Only start developing a new feature once a current feature is tested (WIP limit)

Root cause analysis can help with low productivity since bugs often have underlying reasons

Low productivity can be solved by talking face-to-face to be able to speed up decision making

Don't have to many requirements

Changing requirements can be solved by JIT and continuous learning. This will cause the team to keep up with customer needs

JIT would solve changing requirements because you dont make decisions before the customer actually demands it and as late as possible

Root cause analysis to find the reasons for failure instead of only fixing the failure itself

1

11

Mentimeter

Suggest one lean technique that could be applied and how this technique would solve one of the addressed challenges

JIT

JIT - only focus on the tasks that are needed - makes the architects happy

Embracing change, making changes that defines a scope dialed in to the easiest solution for the customer.

Continuous learning that the flow is good and everyone is learning and improving

chi

on

ts -

Late Testing – Testing happens

Component Production, focus on properly finishing and testing one component before going on to the next.

an

ms

the company should try to do a root cause analysis on their issues, this will slow development a little, but the time spent analysing the cause to the low work ethic m.m will also give continuous lear

1

11

The purpose of Lean is to optimize the process in all parts of an organization to achieve a common goal. It is known for eliminating waste and can overall increase productivity and morale. Based on this, it could be a fitting approach for MediC AS.

Avoiding waste:
Find a balance between upfront and emergent design. Investing too much time into architecture and design of software products upfront can be wasteful.

A useful Lean technique is **"Just-in-Time"** which essentially means "Do not produce it before somebody demands it". The technique is useful for reducing waste, since only what the customer actually wants is produced. I propose this technique since the probability of an architectural decision to be wrong decreases over time. This is because more information is available and more experience is gathered. The architecture should be made just in time, if not; delay it. This could help MediC AS reduce their social debt involving developers and architects. The architecture should then be more flexible to accommodate new requirements and issues found by developers.

Supporting change:

- Do shorter release cycles – you will have faster feedback. Therefore, you will have time to fix things based on the feedback and not build on top of them until it is too late and the feedback is not that helpful.
- Continuous Integration – will help the team to scale up as described above while monitoring the changes.
- Develop unit and integration tests – developers will be sure that their new features are not breaking anything covered by those tests in the system. This way they will have better quality of the code.

Flow:
Improve the inter-team communication – introduce Slack as a tool that will help the communication between teams as well as between members of the same team. This way everything will be more transparent and honest. This will also help the Social Debt that has been accumulated in MediC.

Satisfying the customer:

- Instead of planning too much in the beginning, include the customer as much as possible in the process. So that you will get more information.
- Deliver more frequently and smaller increments - so that you get feedback and adjust the future decisions based on that. This way the architects will be adapting to a more balanced design and not do everything upfront. Because some requirements might not be needed anymore or are not understood correctly. The more information you get the better the decisions that each team will make will be.

Visualization:
Tracking the development progress – introduce Jira to keep track of the:
Visions/Objectives, Plans and milestones, Tasks, Metrics, Timing or technical problems

Mentimeter

[2025]Describe similarities between Lean and Agile

both are topics in this course

Both are customer first and supports change and improvement

Both focuses on satisfying the customers needs

Lean and Agile both focus on working smarter. Some agile ideas come from lean, lean is a philosphy, not an agile method. Both aim for process improvement

Both want to give value to the costumer, always improve and learn.

The customer at the centre, self organizing teams, continous learning and kaizen, flow

Continiously deployment. Do sprints where you work on a feature. Customer focused

Both focus on processes and time-management

[2025]Describe similarities between Lean and Agile

both emphasise continuous development and learning in teams, the customer is the main point in both methodologies

both focus on continuous learning, customer first, maximizing the value of the customer

Continuous learning, customer value, sustainable development, simplicity

Both have strong teams and choose distribution of tasks Both focus on bringing value to the customers Both focus on continius learning

Both want to be efficient and you do what you need

Lean and agile are both focused on the customer needs. They both focus on working on a maintainable pace. They both reflect to improve future processes.

Both Lean and Agile focus on distributing value to the customer and conitnuious learning. While Lean is more of a philosophy for the company, agile is a team specific method.

Both are recuding waste and work closer to the customera and are adaptable to changes during the development process

Mentimeter

[2025]Describe similarities between Lean and Agile

Satisfied :)

14

[2025]
Describe similarities between Lean and Agile development.

Lean & Agile : both aim to deliver value to the customer quickly and efficiently, but they approach to this goal from slightly different angles.

Aspect	Lean	Agile	Common Ground
Customer Value	Understand needs, deliver value	Understand needs, deliver value	Customer-centered
Continuous Improvement	Kaizen (culture of improvement)	Retrospectives, iterative development	Ongoing learning & adaptation
Efficiency & Waste	Eliminate all forms of waste	Short iterations, daily stand-ups	Streamlined development, efficiency
Teamwork & Collaboration	Close communication, self-organization, shared responsibility	Same emphasis on collaboration and teamwork	Strong teamwork & shared responsibility
Flexibility & Adaptability	Just-in-Time, flexibility under uncertainty	Respond to changing requirements	Embrace change
Complementary Role	Framework for process optimization	Iterative & incremental practices	Hybrid approach → suits team needs & context

Mentimeter

Lean and Agile are complementary.

Lean : framework for optimizing the entire software development process

Agile : specific practices and principles for iterative and incremental development

Many teams use hybrid approach for their specific needs and context

Mentimeter

Feedback

Mentimeter

Thank you for today!

Mentimeter

Next Week:

Empirical research methods in Software Engineering