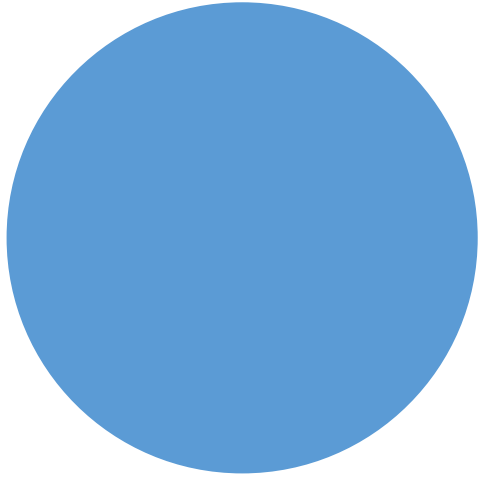


Group session

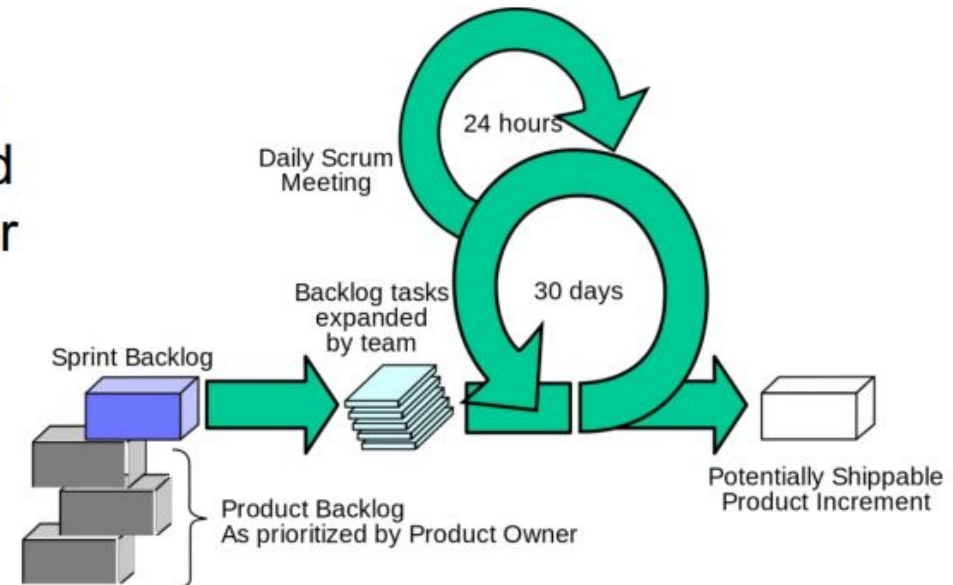
IN5140

Agile teams - roles,
practices, and
challenges

Recap

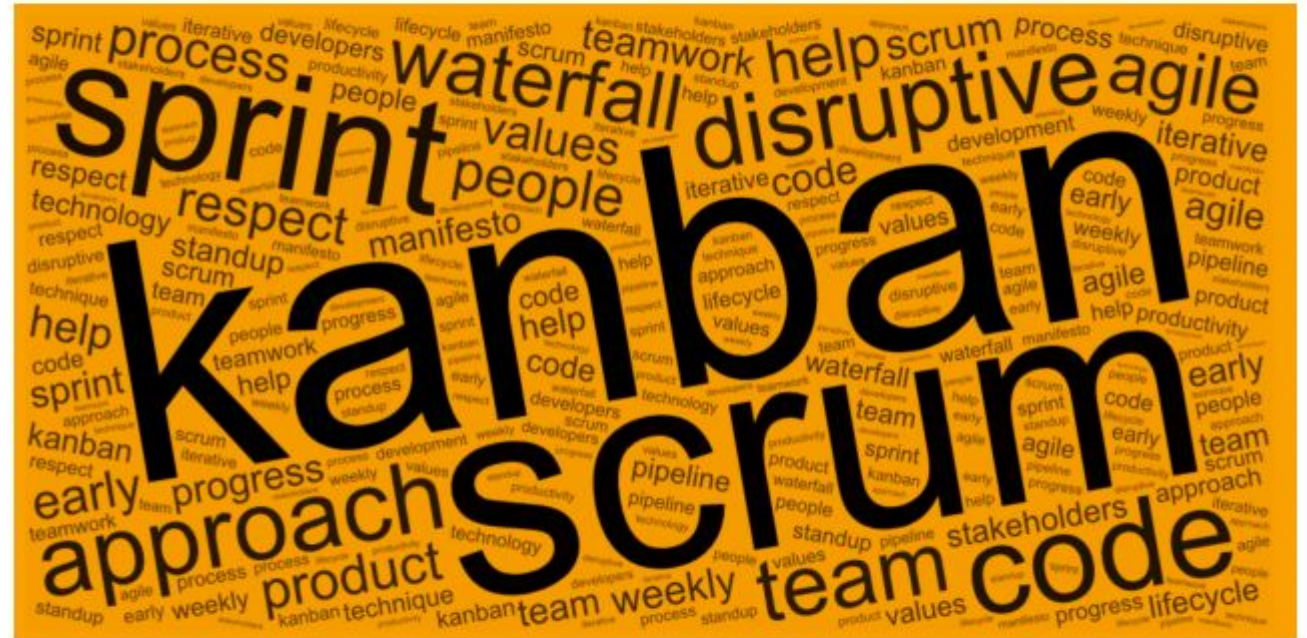
What is Agile software development?

- «**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



There are many different approaches to agile, all with similar underlying ideas, values and principles

- Agile methodologies
- Agile methods
- Agile processes
- Agile practices
- Agile principles
- Agile teams
- Agile organizations
- Agile transformations
- **AGILE MINDSET**



The agile manifesto

As software developers, we value

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

While there is value in the items on the right, we value the items on the left more.

Why is **the team** so important in agile?

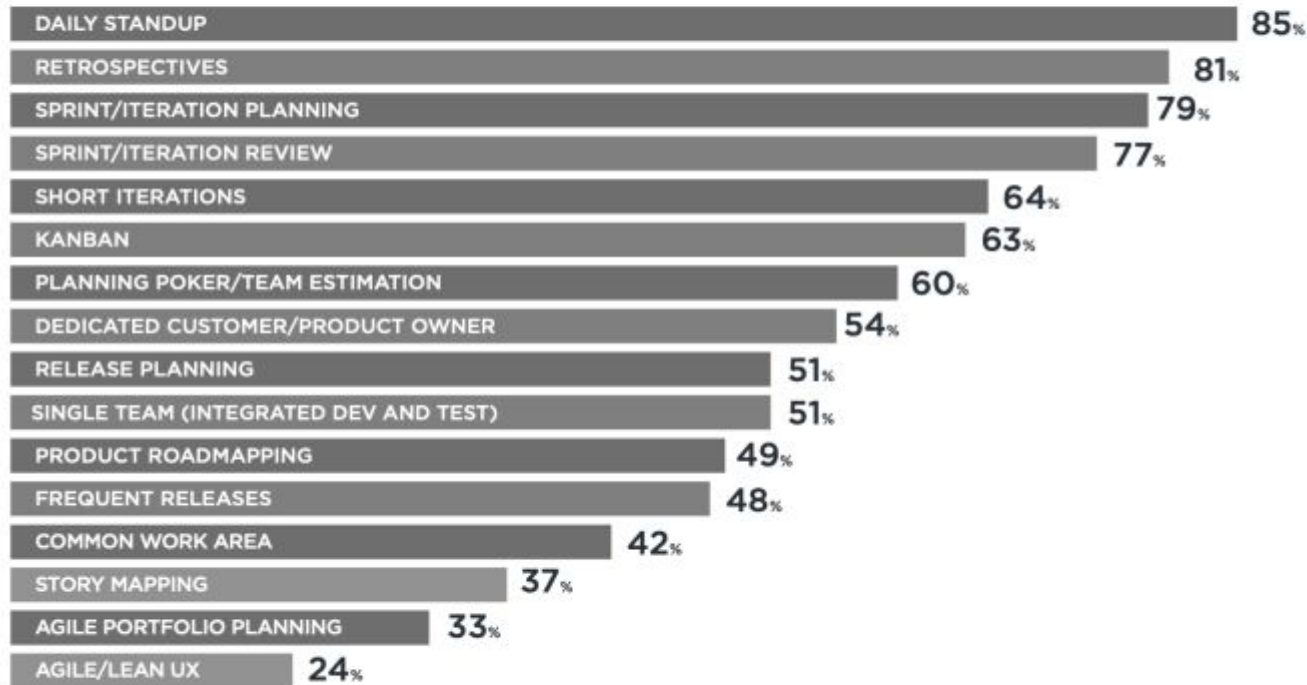
Most software systems and products are developed by **groups of people** who need to **interact to deliver the product**

Agile principle #11:

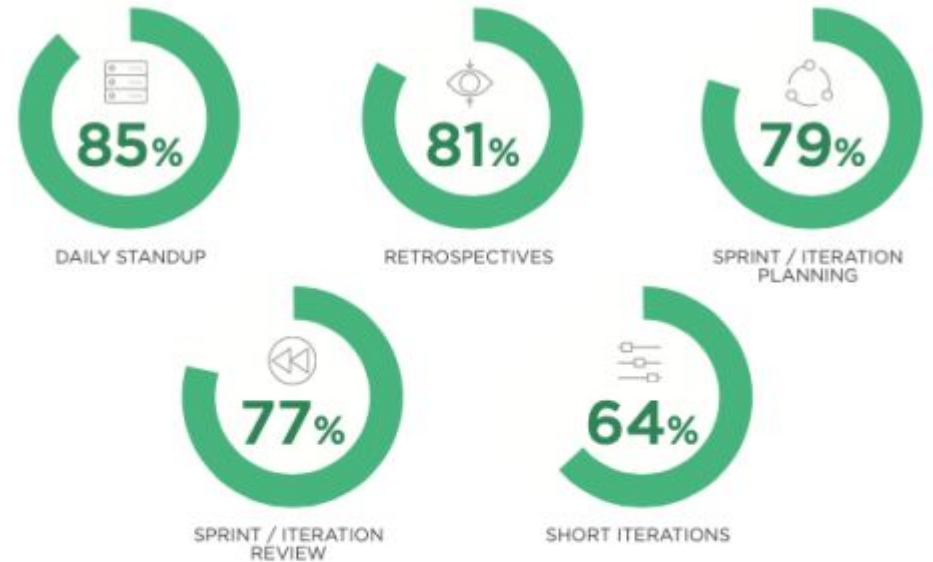
«*The best architectures, requirements and designs emerge from **self-organizing teams***»



The most popular agile organizational practices



TOP 5 AGILE TECHNIQUES



Some of the most popular technical development practices

Organizational practices

1. Daily meetings
2. Development sprints
3. Retrospective meetings
4. Documentation writing

Technical practices (chapter 7)

5. Continuous integration
6. Pair programming
7. Test-driven development
8. Refactoring

Technical practices: Continuous integration/deployment (CI/CD)

What it is: Rather than waiting until the end of the sprint/project before putting “everything” together, continuous integration refers to frequently committing changes, assembling and testing code. Continuous deployment involves releasing the code into production.

How it should be done:

- Principle no 3: “**Deliver working software frequently**, from a couple of weeks to a couple of months, with a preference to the shorter timescale.”
- Integration cycles has increasingly gone shorter
 - Microsoft’s daily builds
 - XP: “integrate changes after no more than a couple of hours”

In practice:

- CI is generally considered to increase productivity and efficiency
- But can also bring technical and organizational challenges such as increased complexity in the development environment
- CI is associated with increased confidence in the software quality, sometimes even a false sense of confidence (Soares et al., 2022)

Technical practices: Pair programming

What it is: When two developers work together in real-time at the same workstation, ideally co-located, with one person writing the code (driver) and the other reviewing (navigator).

How it should be done:

- Pair programming is a co-located practice
 - “write *all programs* with two people sitting at one machine” (Beck, 2003)
 - “cover your mouth when you cough”(!)



- Changing roles and changing pairs is encouraged

In practice

- Pair-programming works well in WFH and hybrid work (Smite et al., 2021, Tkalich et al., 2023)
- Many developers prefer to work with the same people most of the time (Tkalich et al., 2023)
- Do not use it as a mentoring activity!
 - Many developers prefer a partner who have complementary skills to their own, (Begel & Nagappan, 2008)

Technical practices: Test-Driven Development (TDD)

What it is: An approach to software development where tests are always written before the actual code. TDD can be considered a software development method in itself.

How it should be done:

1. Quickly add a test
2. Run all tests and see the new one fail.
3. Make a little change.
4. Run all tests and see them all succeed
5. Refactor to remove duplication (Beck, 2003)

In practice

- Very few development projects apply TDD in this original form
- The test-first principle has become widespread
- Practical challenges with TDD include
 - Difficulties with adopting the mindset
 - Lack of knowledge and experience with this way of working
 - Tests are written to be passed rather than actively finding issues (Staegemann et al., 2022)

Technical practices: Refactoring

What it is: The process of restructuring code with the goal of improving the internal quality without changing the code's functionality.

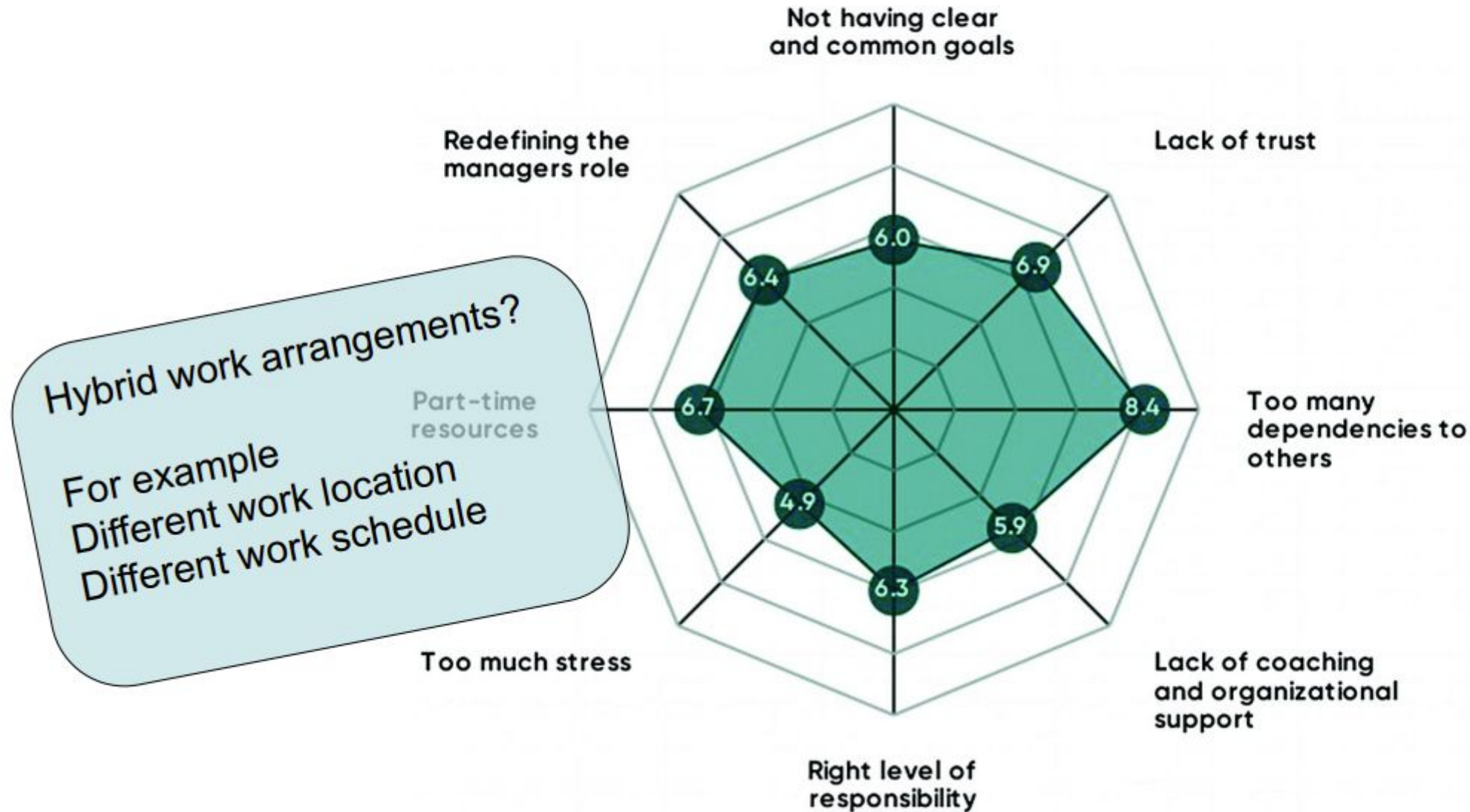
How it should be done:

- Refactoring is only about improving the quality of the architecture
- Refactoring should “make the design simpler”
 - No duplication of code
 - As few classes and methods as possible
- Refactoring should be done in small, incremental steps

In practice:

- Over time, technical debt is likely accumulate, and a “Big refactoring” may be needed to fix it
- The cost of managing technical debt, including refactoring, in large software organizations is estimated to be, on average, 25% of the whole development time (Martini et al., 2018)
- Refactoring is increasingly being automated (Baquais & Alshayeb, 2020)

4. Challenges with agile teamwork





QUIZ TIME!

Quiz

Divide into teams. You can keep the teams in the next exercise.

When you are finished, discuss answers

Question 1

What is a key defining feature of agile methods? (1p)

- A) To create as much software as possible, as fast as possible
- B) To build the software product iteratively and incrementally
- C) To have a lot of defined roles and procedures
- D) To plan first, then write and test the software, and in the end release it

Question 2

According to the agile principles, what is the most efficient way to share information in a development team? (1p)

- A) Documentation
- B) Task boards like Scrumboards or Kanbanboards
- C) Face-to-face communication
- D) Slack

Question 3

On a typical agile team, who are likely to have the best insight into task execution?
(1p)

- A) Project Manager
- B) Team members
- C) Scrum Master
- D) Agile Coach

Question 4

What is a cross-functional development team? (1p)

- A) A team with specialists only
- B) A team that builds one specific component
- C) A team that builds a feature of the final product
- D) A team with 7 ± 2 people

Question 5

According to agile principle #11, *the best what or who* emerges from self-organizing teams? (1p)

- A) Architectures, requirements and designs
- B) Documentation and code reviews
- C) Product Managers and Product Owners
- D) Coordination problems and bottlenecks

Question 6

Pair the following sentences so they fit
(1 p per correct pair!)

Individuals and interactions over...	...comprehensive documentation
Responding to change over...	...contract negotiation
Customer collaboration over...	...processes and tools
Working software over...	...following a plan

Bonus question (1p): What did you just pair?

Question 7

Which statement about refactoring is true? (1p)

- A) Refactoring is cheap
- B) Refactoring is about making the design more attractive to investors
- C) Refactoring is only about improving the architecture
- D) Refactoring should be done in one big step

Question 8

According to agile principle #12, at regular intervals, the team should do what? (1p)

- A) Celebrate!
- B) Work slower to send signals to management
- C) Reflect on their own practices
- D) Perform code refactoring

Question 9

When should the stand-up meeting ideally be held? (1p)

- A) The meeting is not used in agile teams
- B) First thing in the morning
- C) When everyone has arrived
- D) Just before lunch

Question 10

Which of these are typical challenges in large-scale agile? (1p per correct)

- A) Dependencies between teams
- B) Documentation and code reviews
- C) Too many managers
- D) Lack of overview



ANSWERS

Question 1

What is a key defining feature of agile methods? (1p)

- A) To create as much software as possible, as fast as possible
- B) To build the software product iteratively and incrementally***
- C) To have a lot of defined roles and procedures
- D) To plan first, then write and test the software, and in the end release it

Question 2

According to agile principles, what is the most efficient way to share information in a development team? (1p)

- A) Documentation
- B) Task boards like Scrumboards or Kanbanboards
- C) *Face-to-face communication***
- D) Slack

Question 3

On a typical agile team, who are likely to have the best insight into task execution?
(1p)

- A) Project Manager
- B) Team members***
- C) Scrum Master
- D) Agile Coach

Question 4

What is a cross-functional development team? (1p)

- A) A team with specialists only
- B) A team that builds one specific component
- C) *A team that builds a feature of the final product***
- D) A team with 7 ± 2 people

Question 5

According to agile principle #11, the best what or who emerges from self-organizing teams? (1p)

- A) Architectures, requirements and designs***
- B) Documentation and code reviews
- C) Product Managers and Product Owners
- D) Coordination problems and bottlenecks

Question 6

Pair the following sentences so they fit
(1 p per correct pair!)

Individuals and interactions over...	...processes and tools
Working software over...	...comprehensive documentation
Customer collaboration over...	...contract negotiation
Responding to change	...following a plan

Bonus question (1p): What did you just pair?
The agile manifesto 😊

Question 7

Which statement about refactoring is true? (1p)

- A) Refactoring is cheap
- B) Refactoring is about making the design more attractive to investors
- C) Refactoring is only about improving the architecture**
- D) Refactoring should be done in one big step

Question 8

According to agile principle #12, at regular intervals, the team should do what? (1p)

- A) Celebrate!
- B) Work slower to send signals to management
- C) *Reflect on their own practices***
- D) Perform code refactoring

Question 9

When should the stand-up meeting ideally be held? (1p)

- A) The meeting is not used in agile teams
- B) First thing in the morning
- C) When everyone has arrived
- D) Just before lunch***

Question 10

Which of these are typical challenges in large-scale agile? (1p per correct)

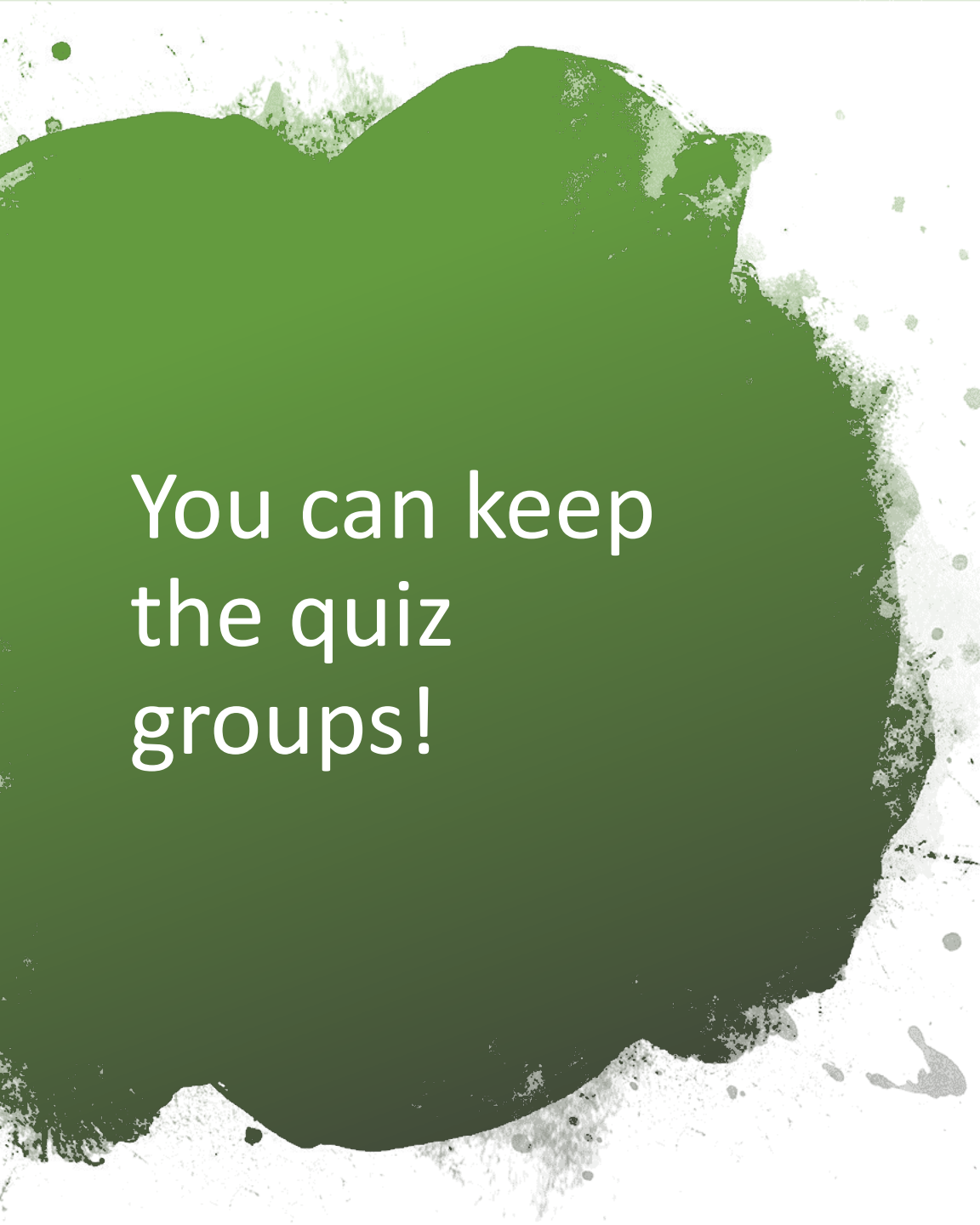
- A) Dependencies between teams***
- B) Documentation and code reviews
- C) Too many managers
- D) Lack of overview***



Discussion
question

Menti 78 79 35





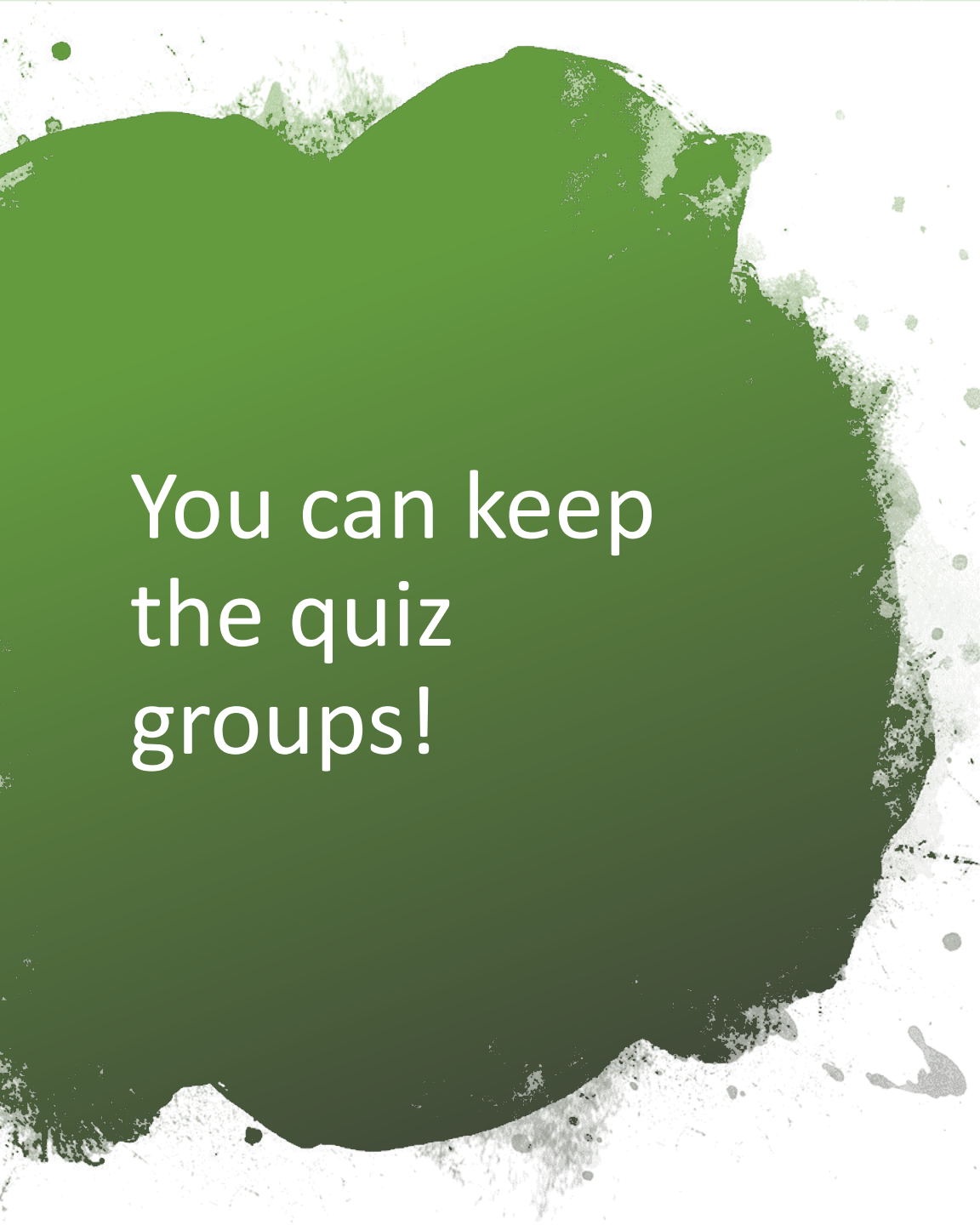
You can keep
the quiz
groups!

Consider your teamwork in your group
project in IN5140

Discuss the following questions:

- 1) What challenges might you face
during the teamwork?
- 2) How will you meet these challenges?

Share some thoughts with the other
groups.



You can keep
the quiz
groups!

Consider your teamwork in your group project in IN5140

Discuss the following questions:

- 3) Have you considered to use any agile practices or techniques?
- 4) Which practices from this week's lecture and chapters 6 and 7 in Meyer could be relevant for your group?

Share some thoughts with the other groups.