

IN5140

Smart processes and agile methods in Software Engineering

Group session 8



Agenda

First hour:

- Some information
- Brief repetition of lecture
- Quiz

Second hour:

- Discussion
- Previous Exam Questions

Feedback Deliverable 1

- Will be published in Devilry by Wednesday, this week
- If you have any comments/questions regarding your feedback, please contact the TAs

Questions?

Speak out loud, or type in chat



Repetition from lecture

Repetition

- What is agile software development?
- The 12 agile principles & The agile manifesto
- Compound and Feature teams
- Agile roles (6)
- Agile Organizational Practices
 - Sprint Planning
 - Stand-up meetings/daily meetings
 - Retrospective meetings
 - Sprint review and demo
- Challenges with agile teamwork : Coordination inter-team or between teams and why?

The 12 agile principles

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support their needs, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity--the art of maximizing the amount of work not done--is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Sprint planning

- A **planning** meeting for each upcoming sprint
 - Duration: 1 hour per week of the sprint, e.g., a two-hour meeting for a two-week sprint
- The team collectively agrees on a **sprint goal**
- Team members pick preferred tasks and the team collectively plans on how to divide resources
- How do we know something is finished?
 - “Definition of Done”

Stand-up meetings

- 15 minutes status meeting every day
- All team members participate
- Facilitated by Scrum master
- All team members answer these three questions:
 - 1) What did you do yesterday?
 - 2) What will you do today?
 - 3) Are there any impediments in your way?

Retrospective meetings (Scrum)

- «At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly» **#12**
- At the end of a period of working together, typically a sprint
- Scrum: Max 3 hours

A Typical Sprint Retrospective Model

What worked well?

What could be improved?

What will we commit to doing in the next Sprint?

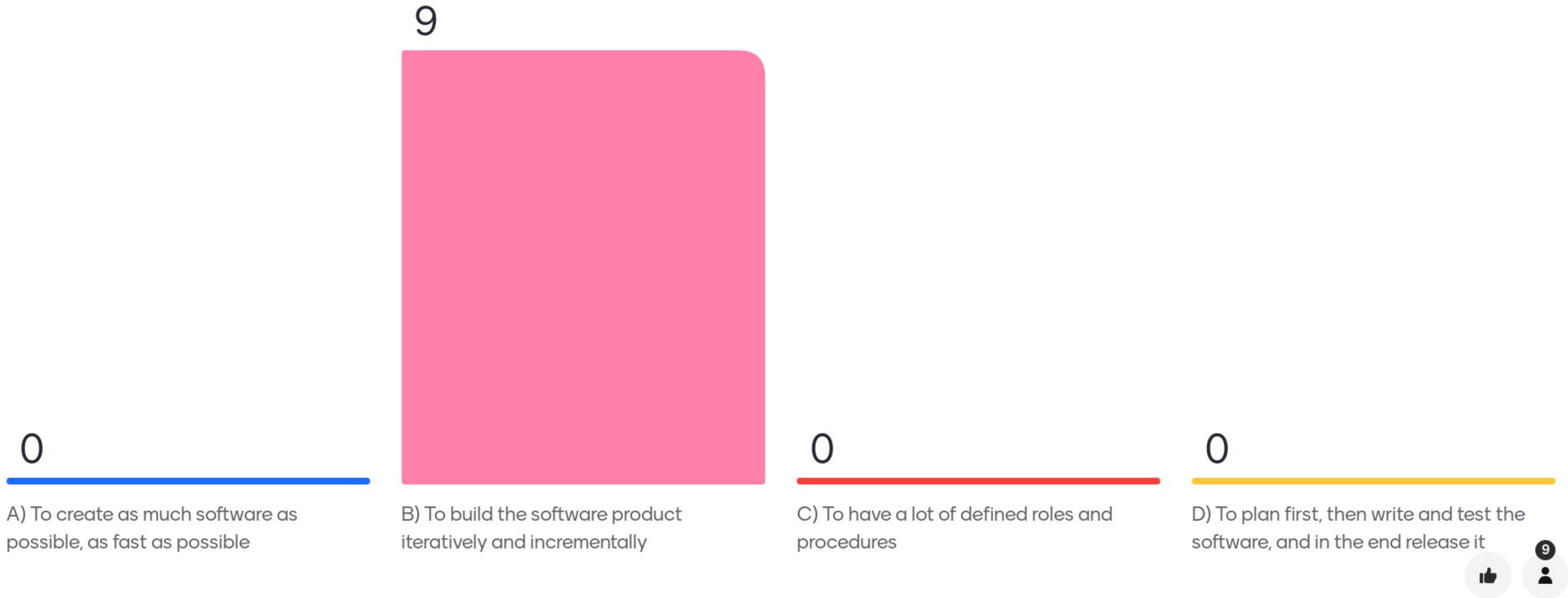
*Scrum Team members
make actionable
commitments*

Sprint Review and Demo

- Held at the end of sprints to inspect the Increment and adapt the Product Backlog if needed (i.e. put unfinished items back, add new items, etc.)
- Go through and evaluate backlog, and/or demonstrate increment
- Attendees: The team, Product Owner and invited key stakeholders
- Duration: Max four hours.
 - Too long?
 - My observations: 15-30 minutes demo keeps people interested

Quiz Time

What is a key defining feature of agile methods?



Question 1

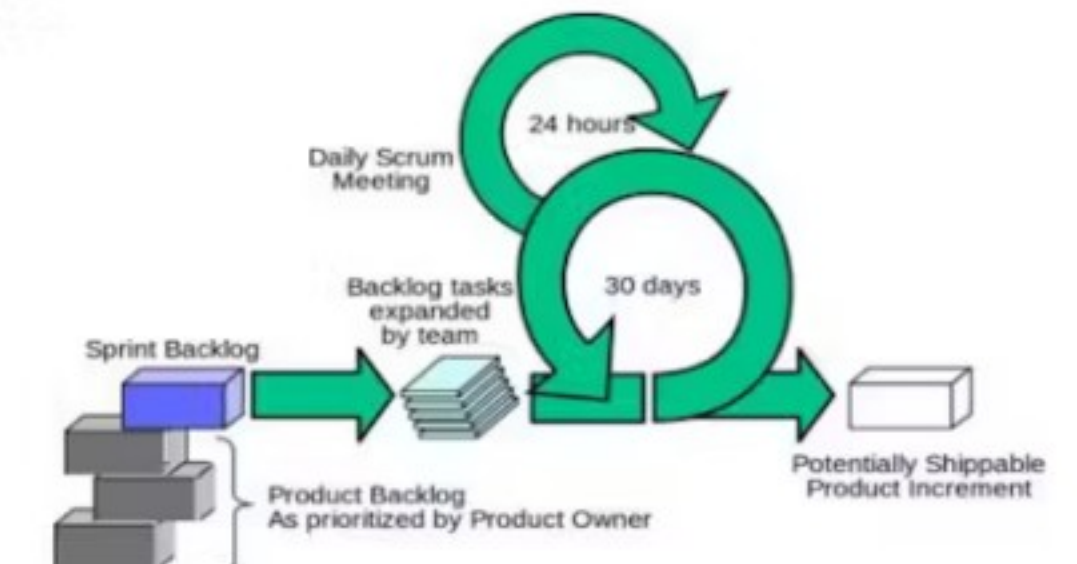
What is a key defining feature of agile methods? (1p)

- 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

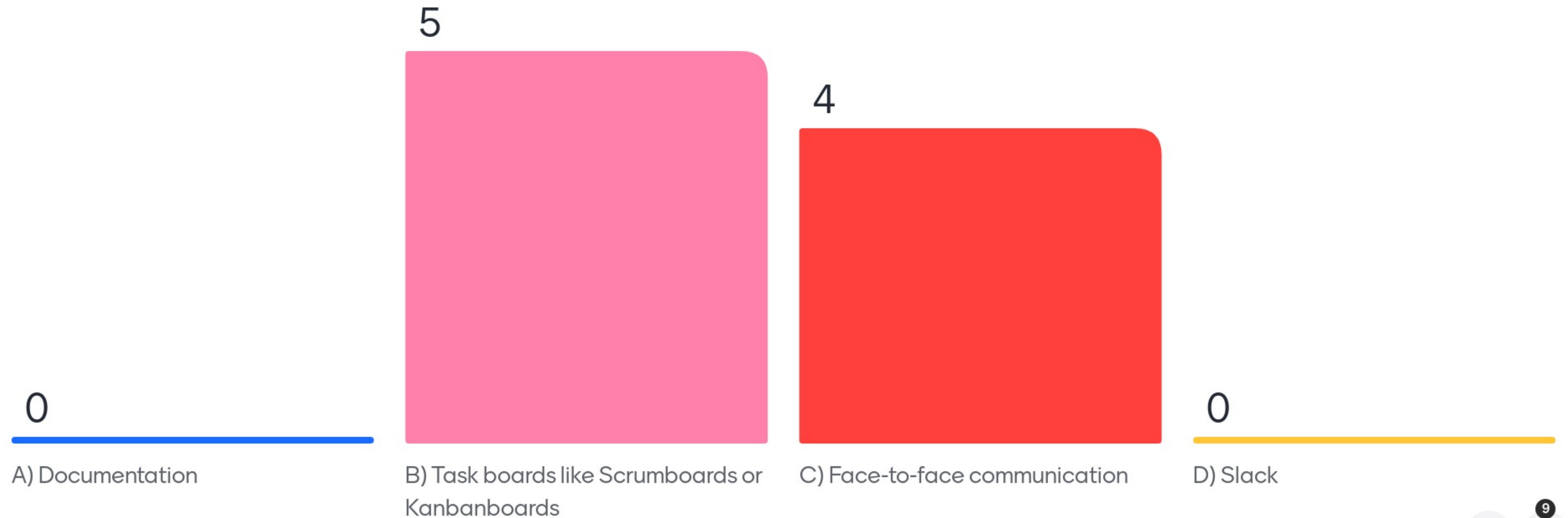
What is agile software development?

«**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 (sprint)
- An increment is a subset of requirements (e.g., backlog items, user stories) for the system
- Agile software development : creating small pieces of software at the time(incremental), repeating the process many times (iterative) in development sprints.



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



Question 2

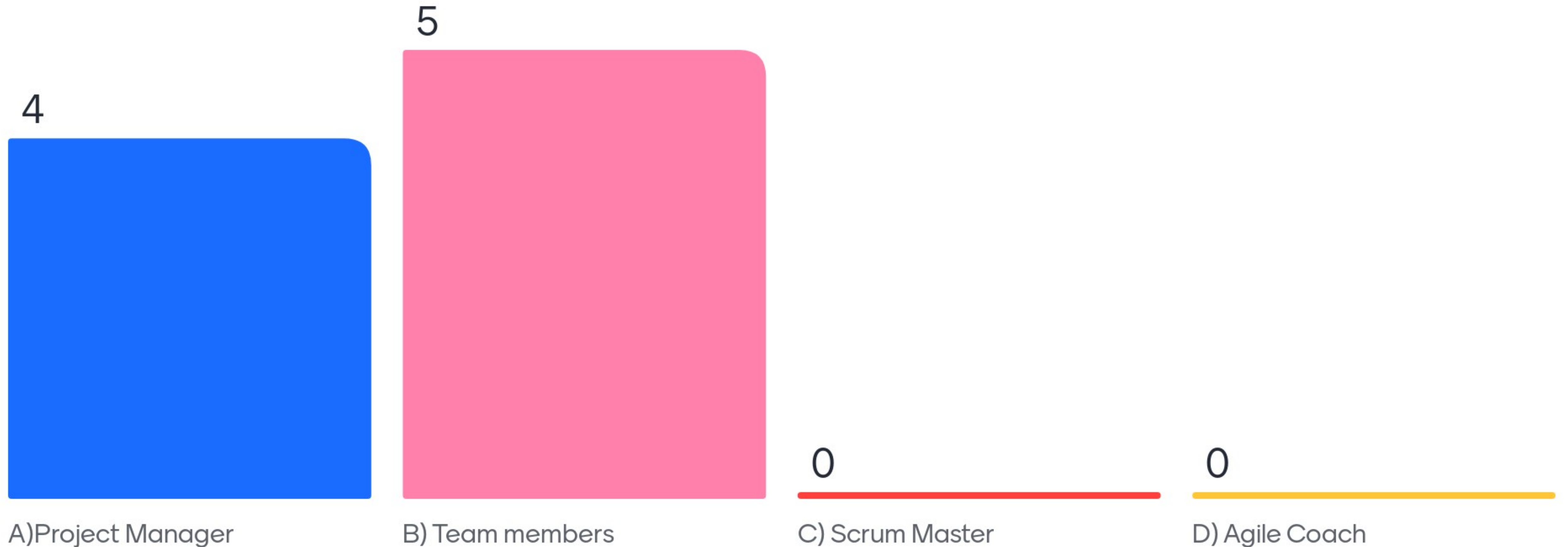
According to agile principles, what is the most efficient way to share information in a development team? (1p)

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

The 12 agile principles

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support their needs, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is **face-to-face conversation**.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity--the art of maximizing the amount of work not done--is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

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



Question 3

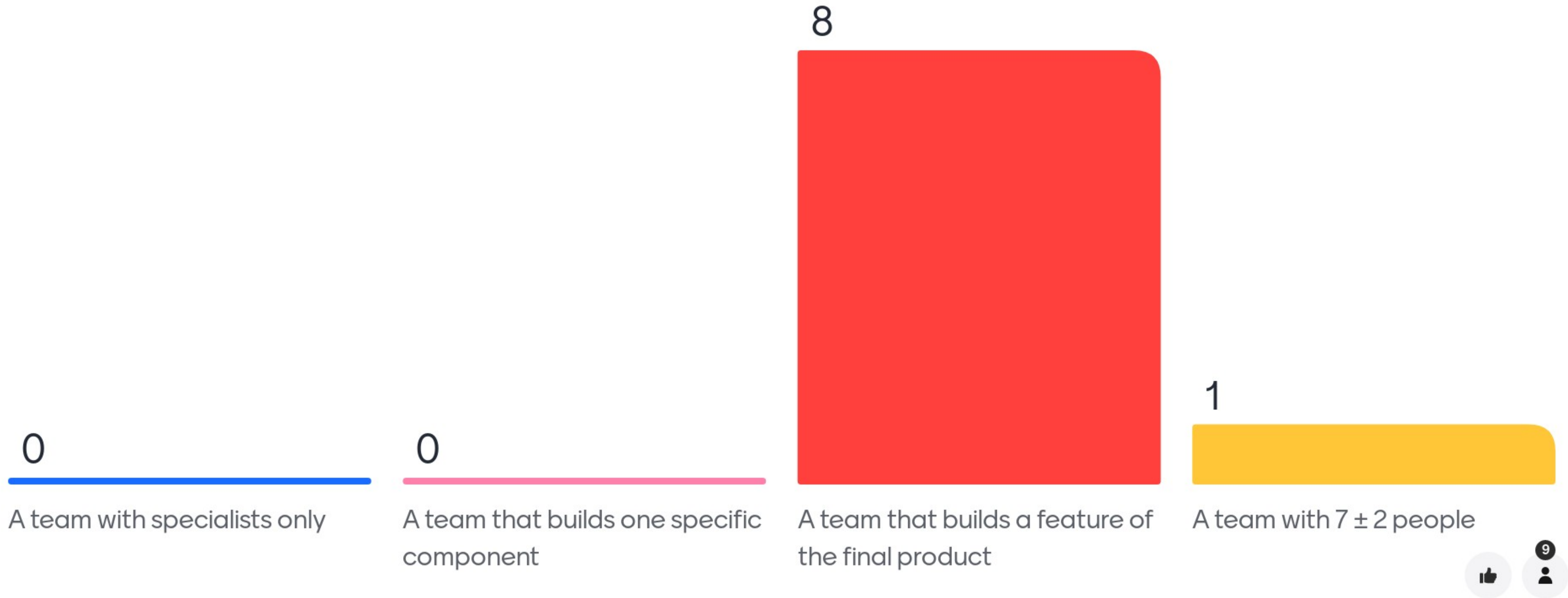
On a typical agile team, who are likely to have the best insight into task execution? (1p)

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

The 12 agile principles

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support their needs, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity--the art of maximizing the amount of work not done--is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

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

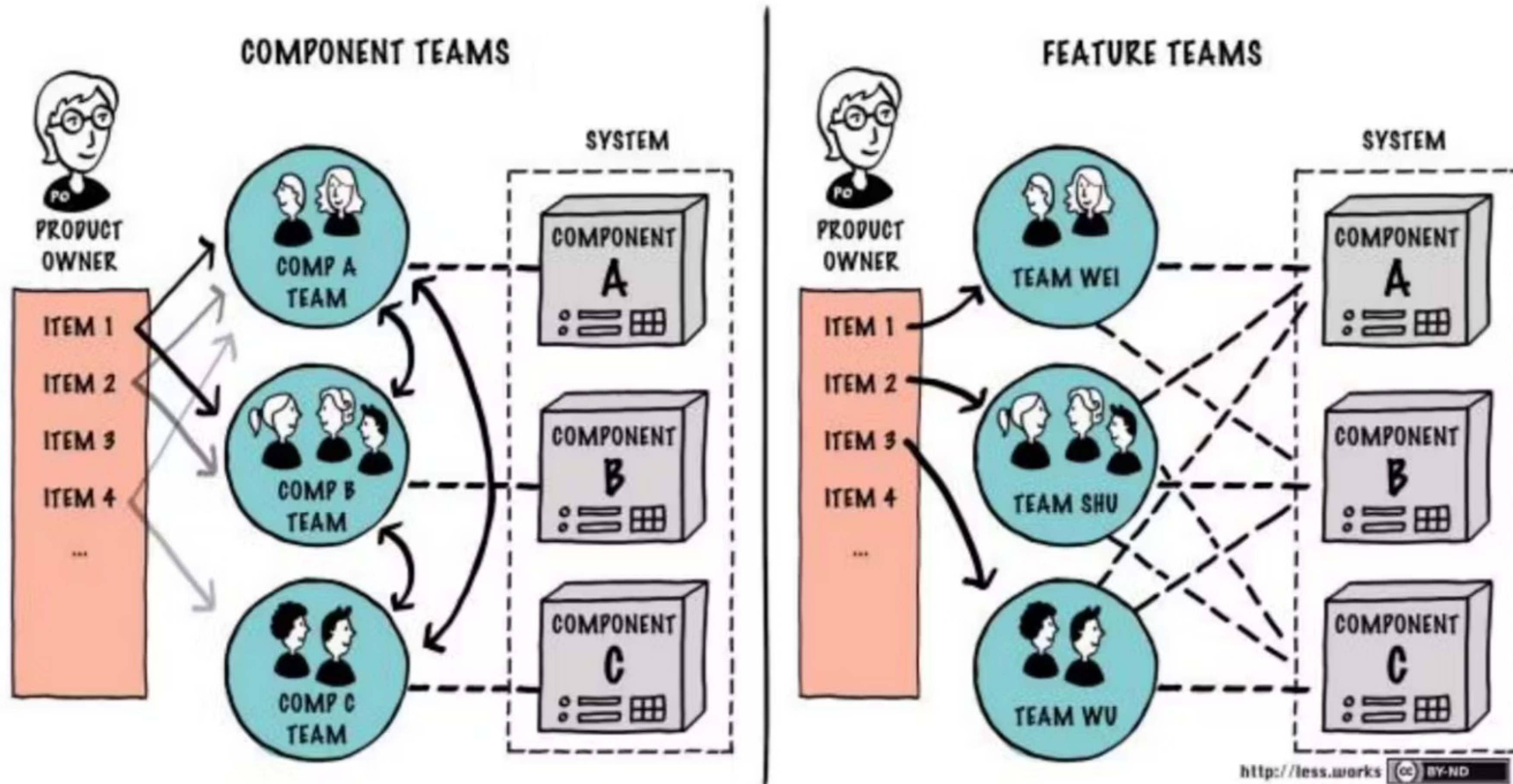


Question 4

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

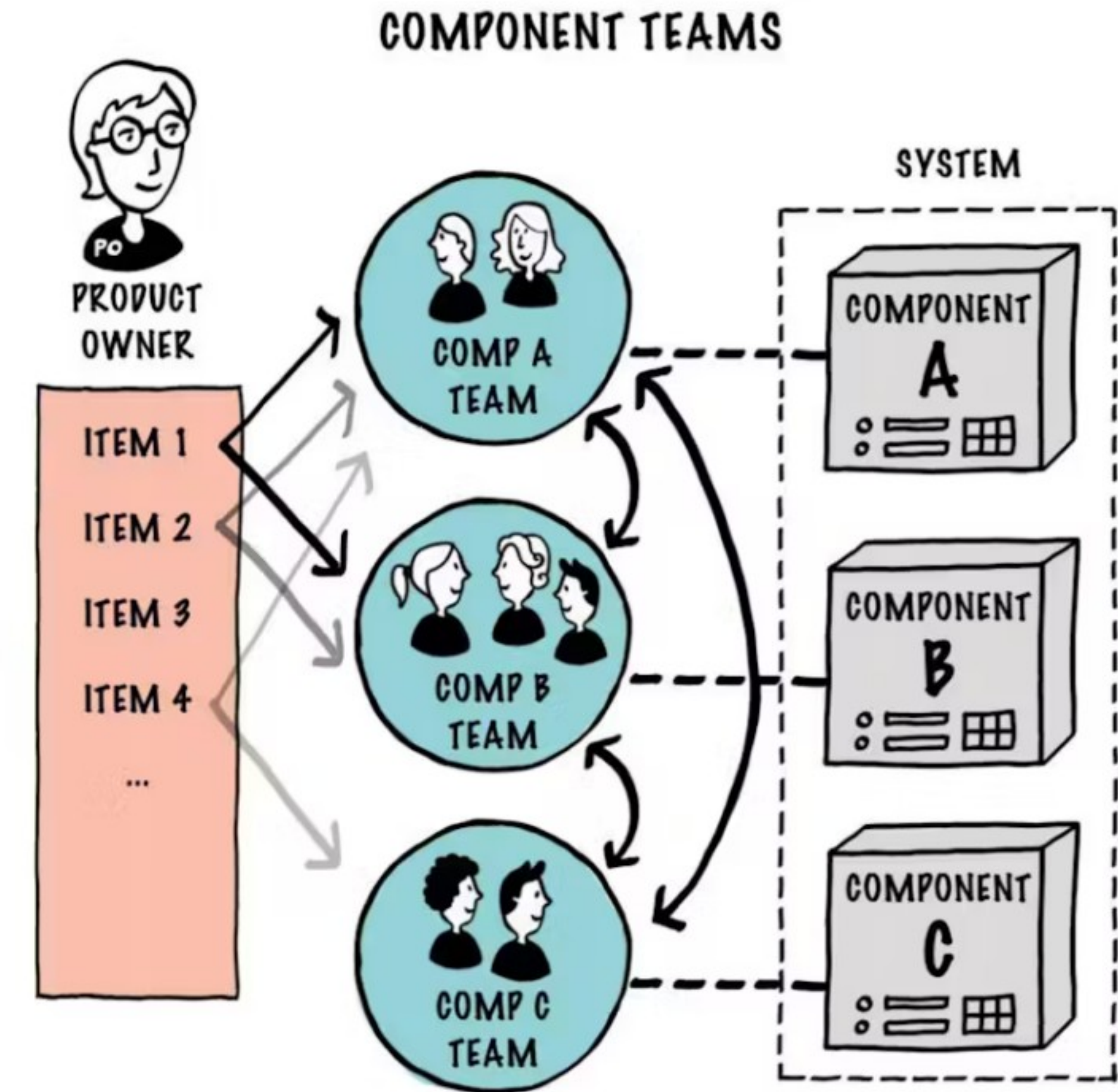
- A) A team with specialists only
- B) A team that builds one specific component
- C) *A team that builds a feature of the final product***
- D) A team with 7 ± 2 people

Two types of software development teams



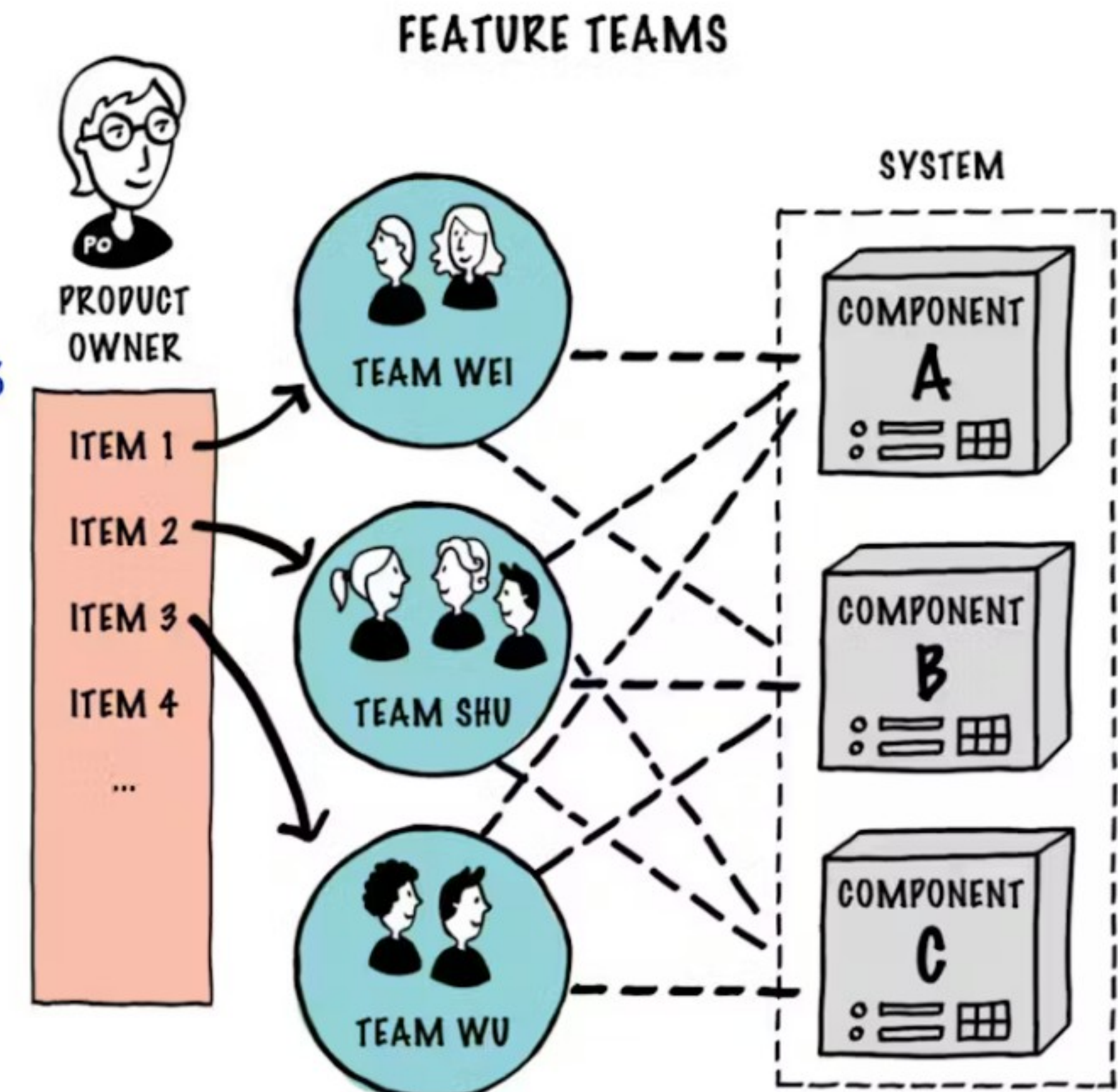
Component (specialized) teams

- More **traditional** approach
- Focus one component of the overall product or development process
- **Clearly defined roles**
 - e.g., Backend team, Front-end team, test team
- Understand their own component well, but may lack overall system understanding



Feature (cross-functional) Teams

- **Modern** way of organizing teams
- Build features of the final product
- Team members **not committed to one specialists** (i.e., full-stack)
- preferably co-located
- Focus on value delivery
- Common in agile teams (e.g. Scrum teams)
- Typically, 7 ± 2 people



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

7



A) Architectures, requirements and designs

0

B) Documentation and code reviews

0

C) Product Managers and Product Owners

1



D) Coordination problems and bottlenecks

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

The 12 agile principles

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support their needs, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity--the art of maximizing the amount of work not done--is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Question 6

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

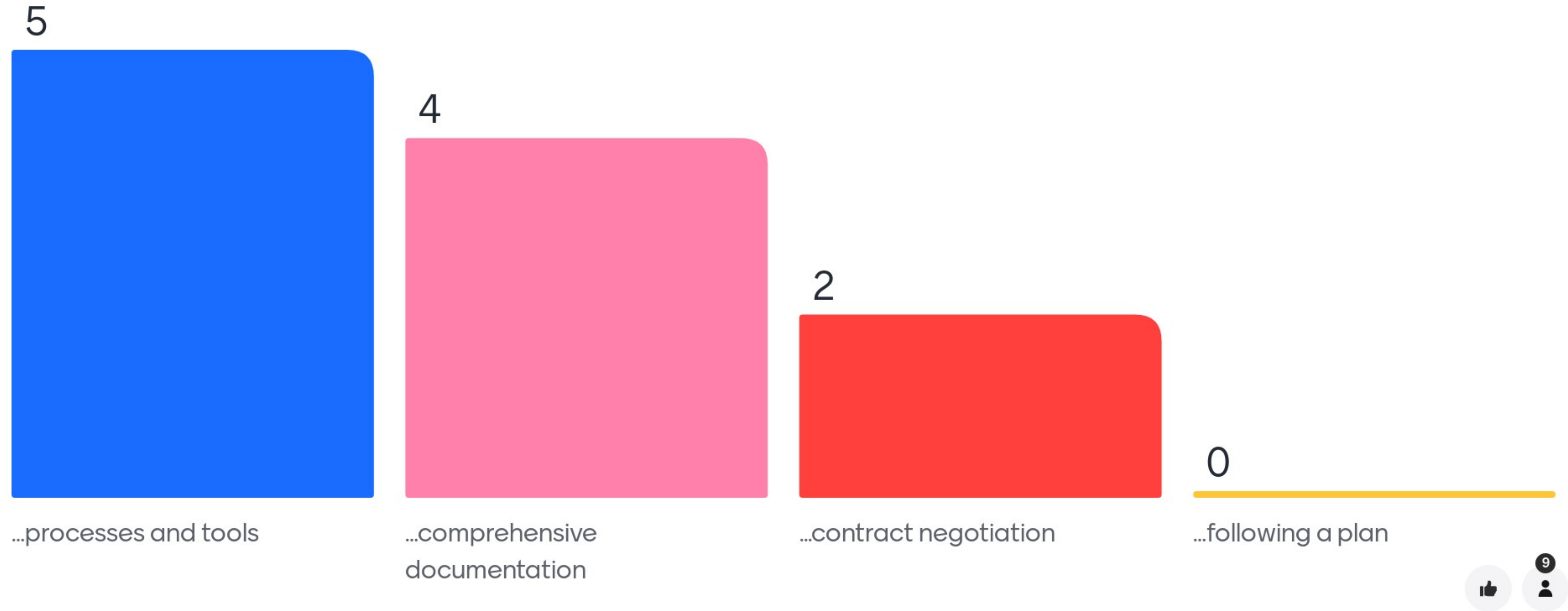
Individuals and interactions over...	...comprehensive documentation
Responding to change over...	...contract negotiation
Customer collaboration over...	...processes and tools
Working software over...	...following a plan

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

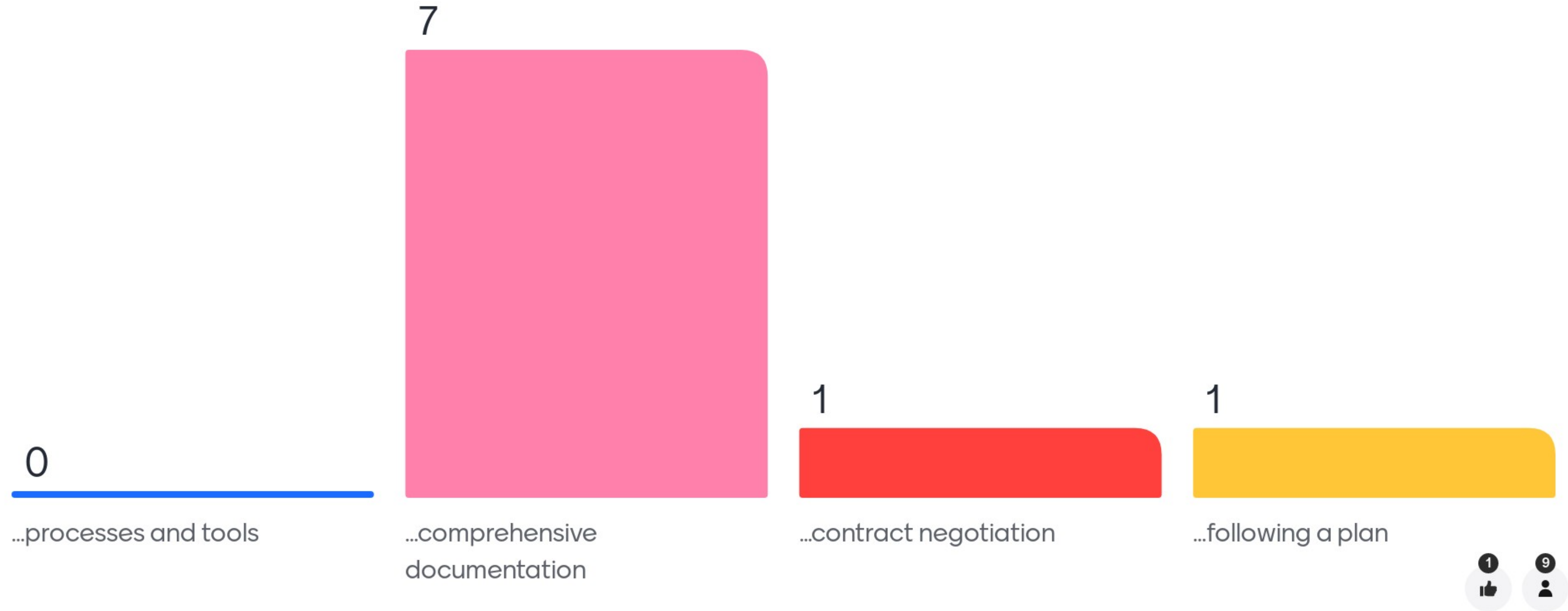


Pair the following sentences so they fit..

"Individuals and interactions over..."

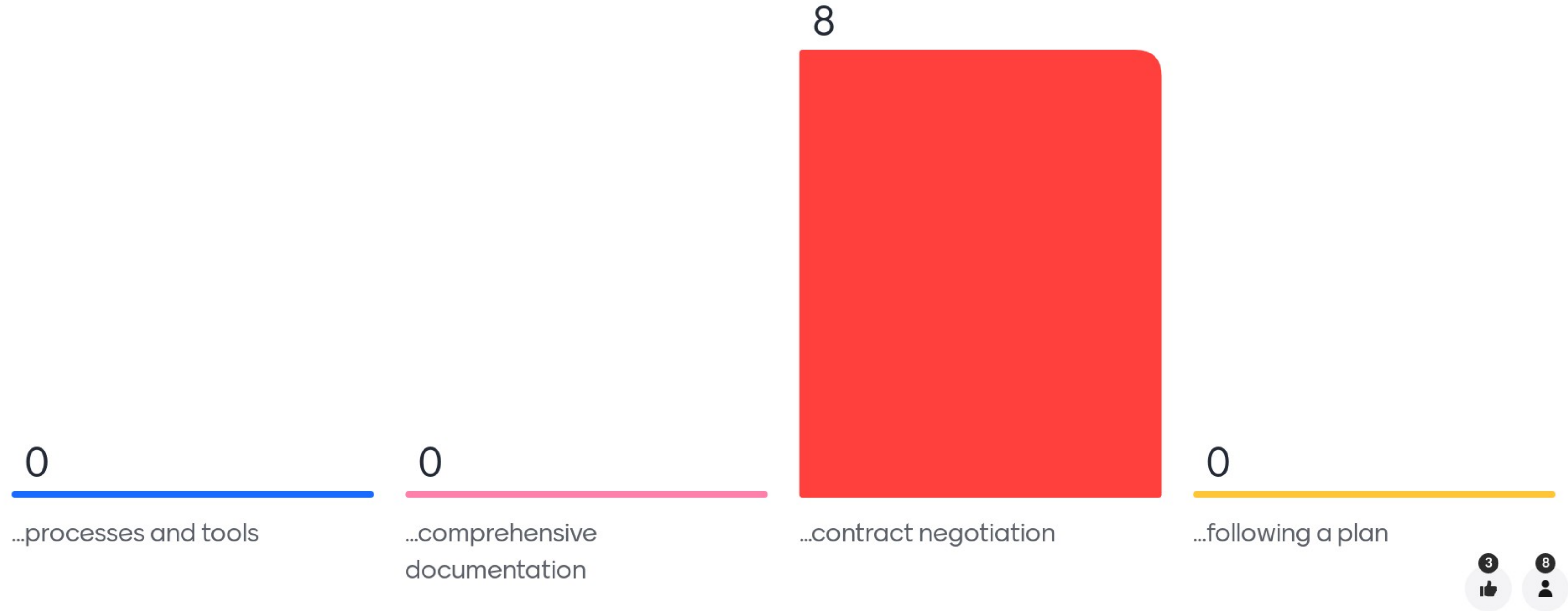


Pair the following sentences so they fit..
"Working software over..."



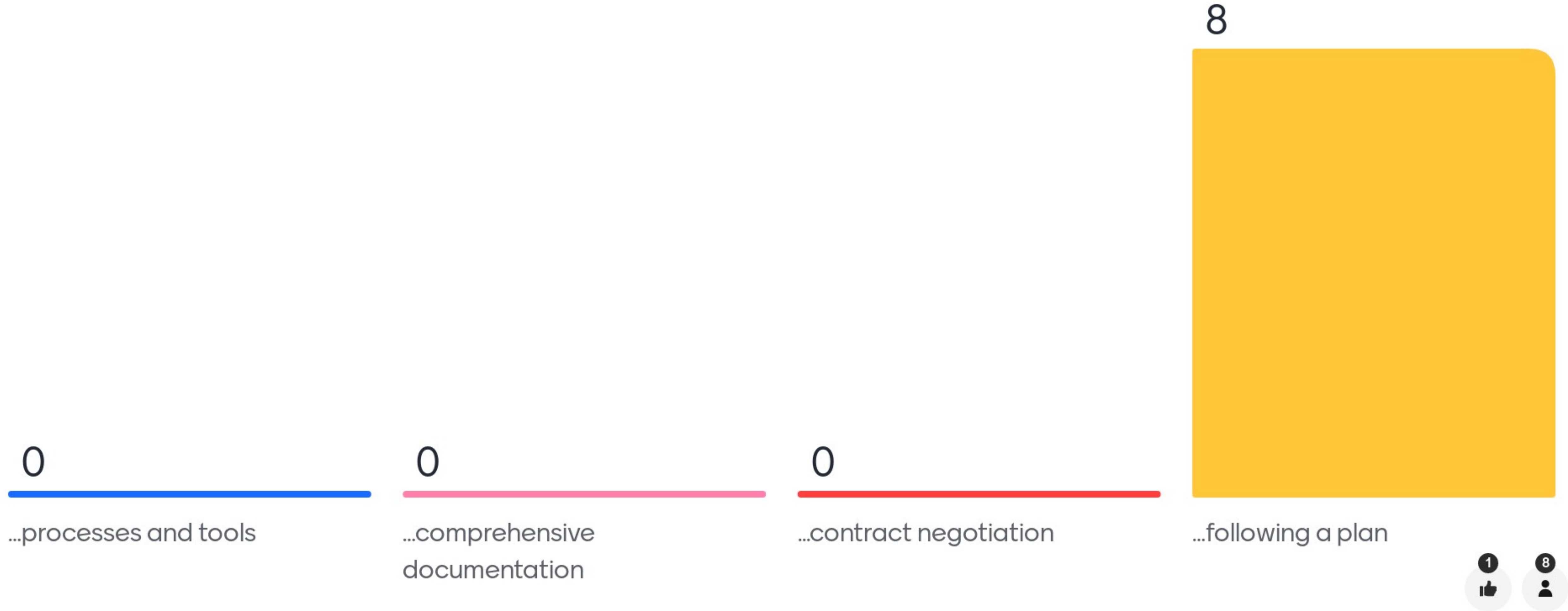
Pair the following sentences so they fit..

"Customer collaboration over..."



Pair the following sentences so they fit..

"Responding to change"



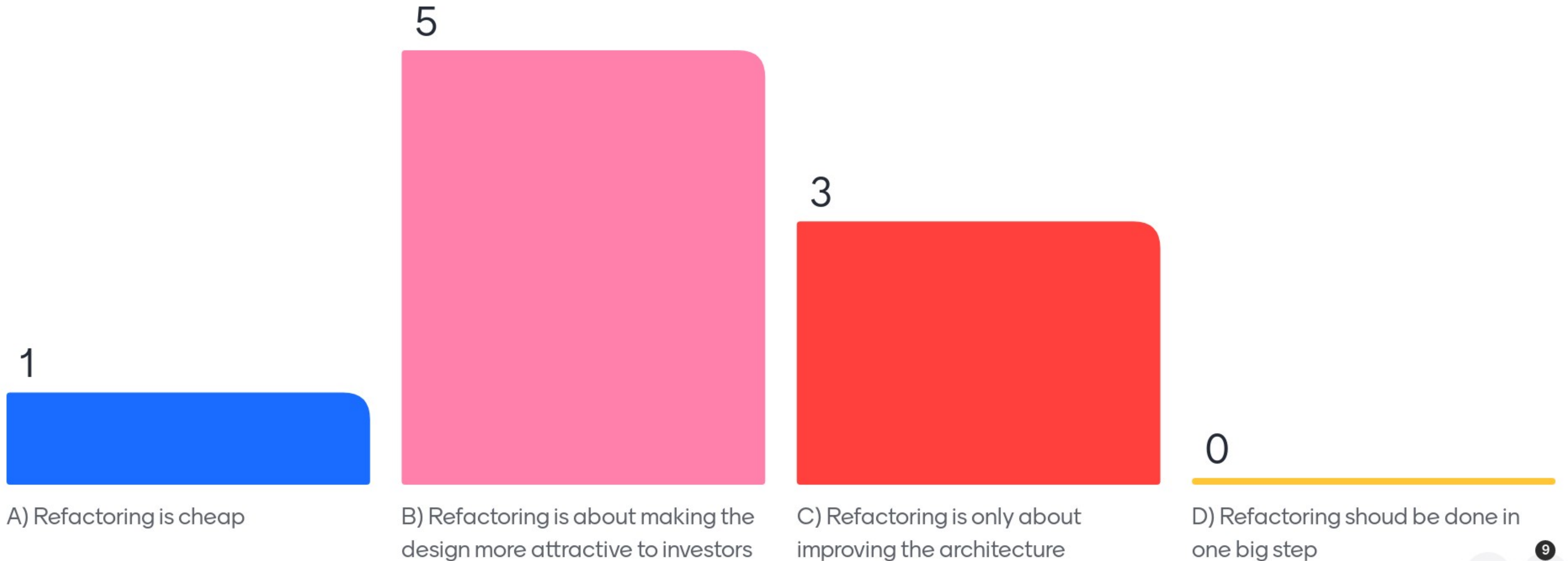
Question 6

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 🤖

Which statement about refactoring is true? (1p)



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

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)

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

0

A) Celebrate!

0

B) Work slower to send signals to management

9

C) Reflect on their own practices

0

D) Perform code refactoring

1



9



Question 8

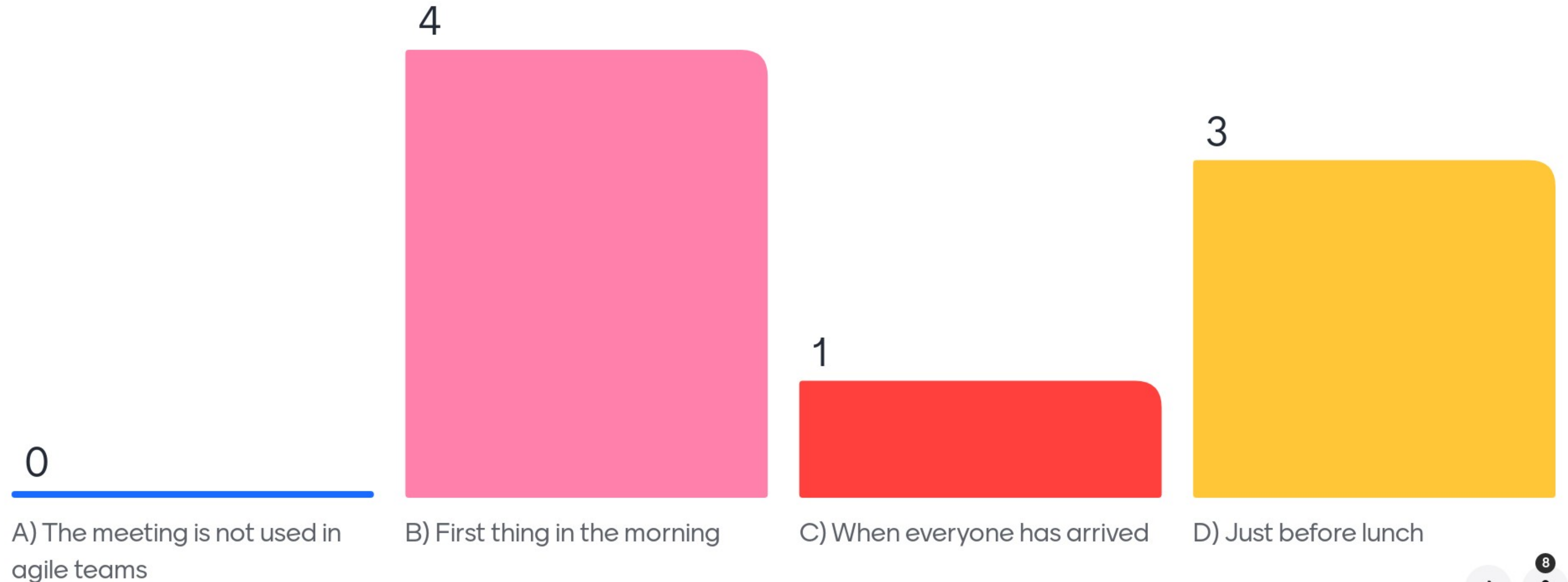
According to agile principle #12, at regular intervals, the team should do what? (1p)

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

The 12 agile principles

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support their needs, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity--the art of maximizing the amount of work not done--is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

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



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***

Organizational practices: Daily meetings

What it is: A short status meeting where team members share what they work on.
Also known as the daily stand-up meeting or daily scrum.



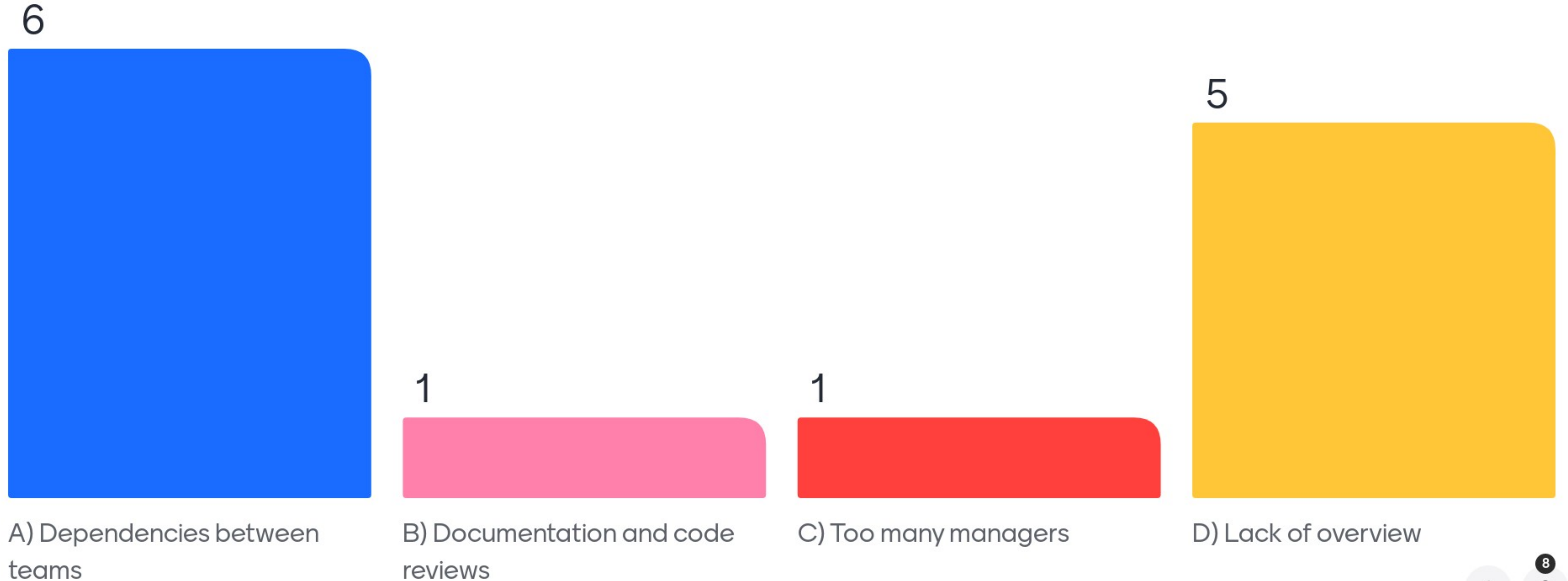
How it should be done:

- The meeting is to be held every morning
- Face-to-face
- 15 minutes only
- Everyone must be standing up
- No problem-solving or discussion
- Everyone answer “the three questions” only

In practice:

- Teams should (and do) adapt the daily meeting to meet their specific needs
 - Time zone differences
 - Flexible work schedules and work locations
 - Team size (a large team may need more time)
- For example, having the meeting just before lunch and focus only on the questions “what will I do” and “what impediments” may be more efficient (Stray et al. 2016).

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



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***

Previous Exam Questions

(2024)

2.2 Finish the sentences

Finish the sentences below regarding agile, so that they are correct.

Each correct sentence gives 1 point. 1 bonus point if all are correct.

Sentence 1 - The agile manifesto

As software developers, we value individuals and interactions over (contract negotiation, comprehensive documentation, following a plan, processes and tools).

Finish the sentences. "As software developers, we value individuals. and interactions over <?> "

processes and tools

process and tools

process and tools

processes and tools

processes and tools

process and tools

The agile manifesto reflects this mindset

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

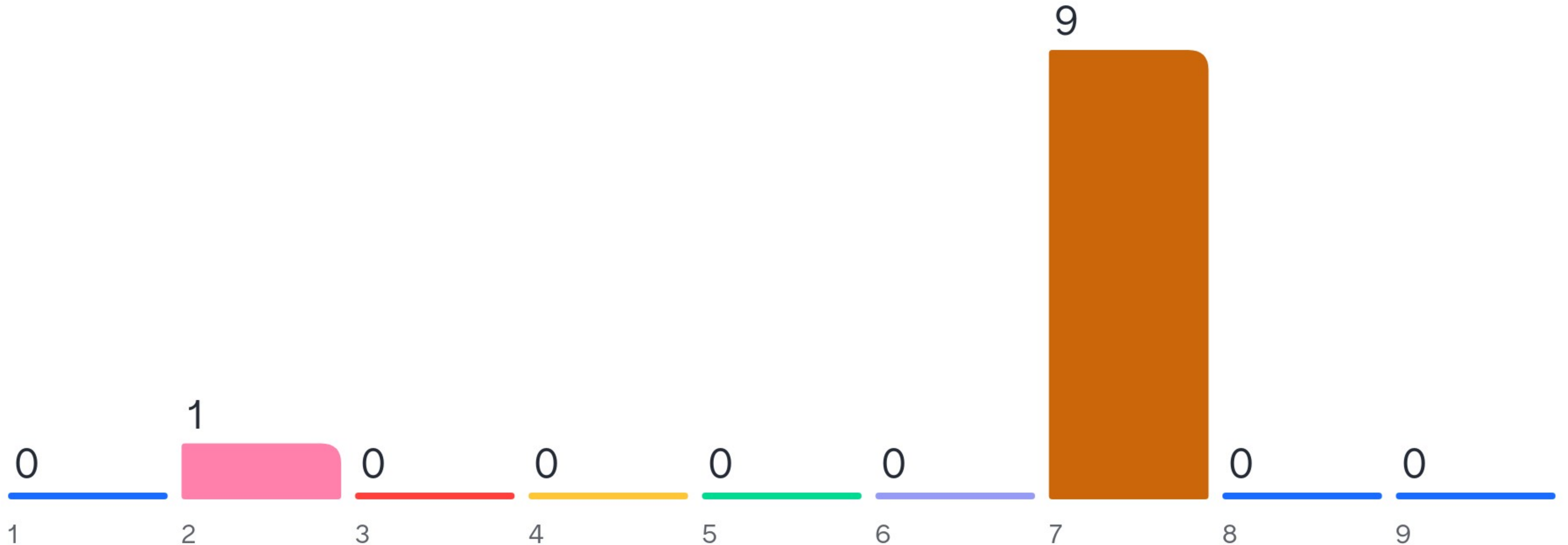
While there is value in the items on the right, we value the items on the left more.

(2024)

Sentence 2 - Cross functional team size

A cross-functional team (feature team) typically consists of (1, 2, 3, 4, 5, 6, 7, 8, 9)+/- 2 team members.

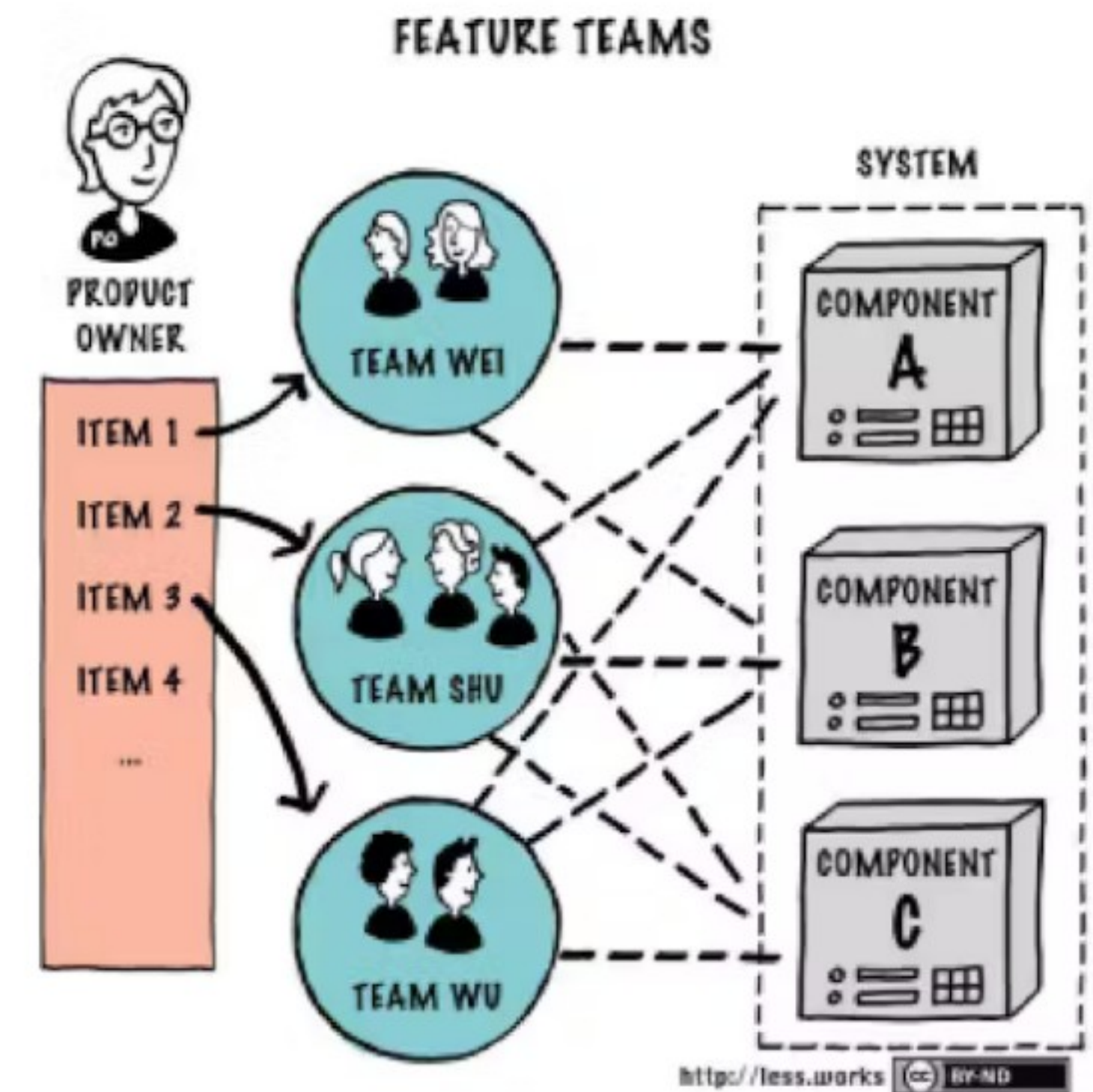
A cross-functional team (feature team)
typically consists of "?" +/- 2 team members



Cross-functional teams (feature teams)

The ideal of agile teams

- Build *features of the final product*
- Team members not committed to one specialist area (i.e., they are full-stack)
- Ideally, any developer can work on any task
- Should work together for a longer time period (preferably co-located)
- Typically, 7 ± 2 people

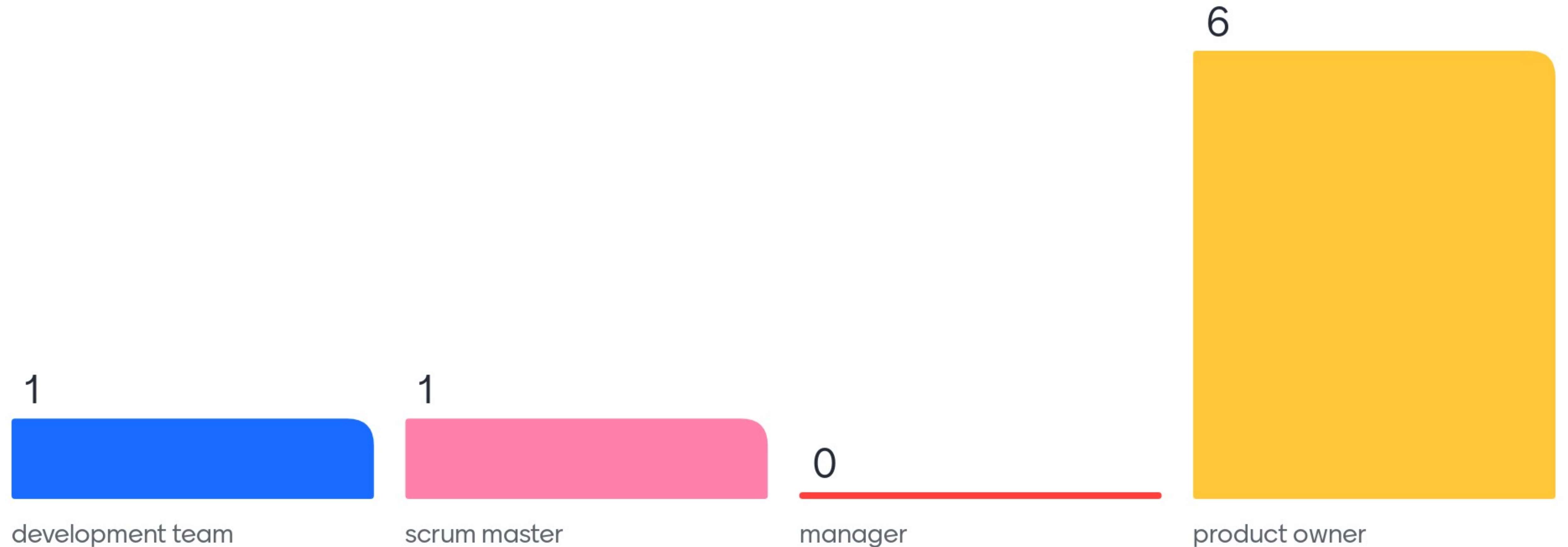


(2024)

Sentence 3 - Roles

The (development team, scrum master, manager, product owner) is often responsible for coordination and prioritization of features in the backlog.

the " " is often responsible for coodniation and priortization of features in the backlog



Agile roles: The Product Owner

The core tasks of the role:

- Define and maintain the product backlog (the list of features to be developed)
 - What to be made, not how to make it. The “how” is decided by the team.
- Involved at the start and end of the sprint
 - Select user stories from the backlog
 - Evaluate the results of the sprint



Product Owners do a lot more!

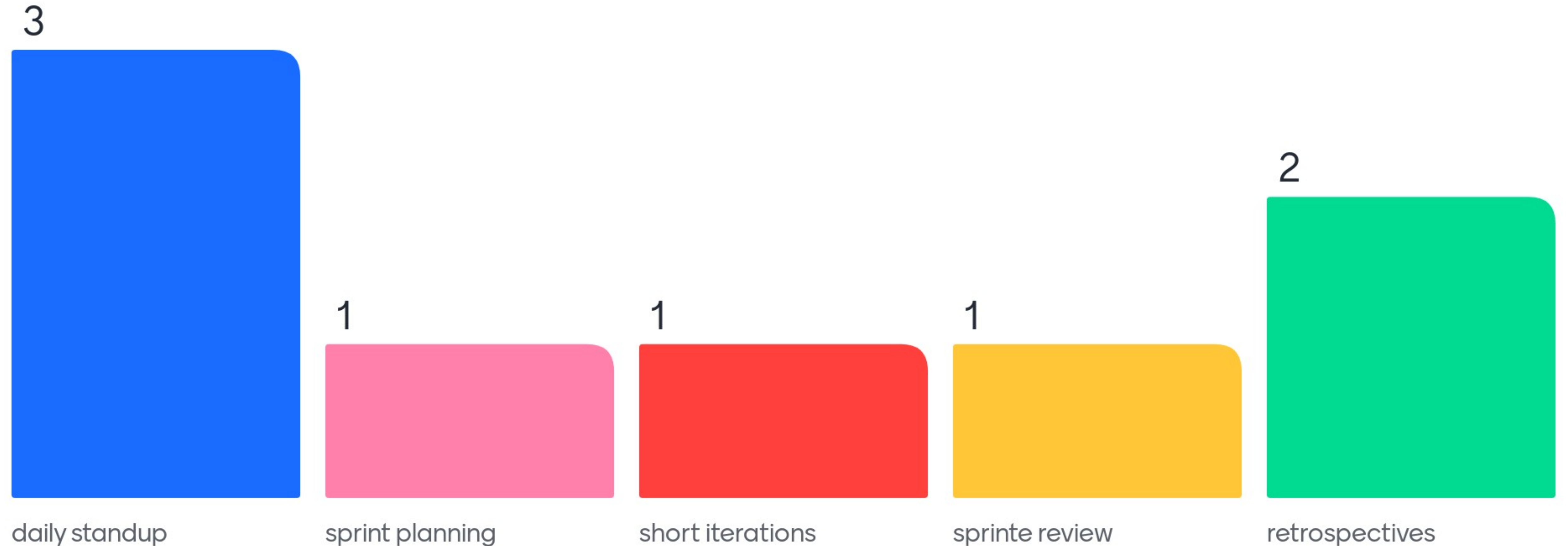
- In large-scale projects, they are often responsible for coordination and prioritization of features across teams (Berntzen, Moe & Stray, 2019)
- They must handle conflicting business needs, groom and prioritize requirements, act as release master, risk assessor, governor, technical architect... and more! (Bass & Haxby, 2019)

(2024)

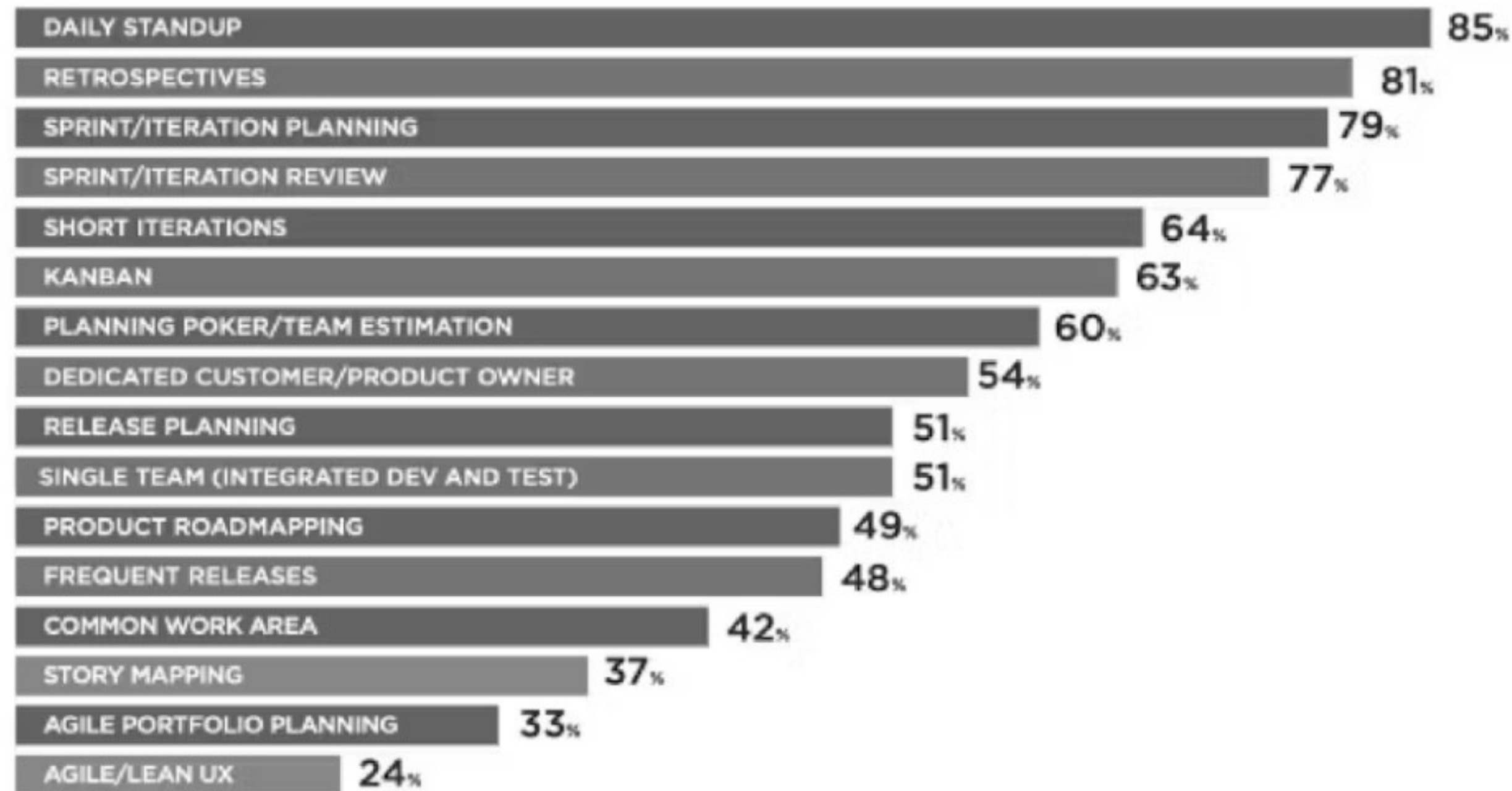
Sentence 4 - Agile practices

According to **14th Annual State of Agile Report** (2020), the most popular agile practice is (daily standup, sprint planning, short iterations, sprint review, retrospectives).

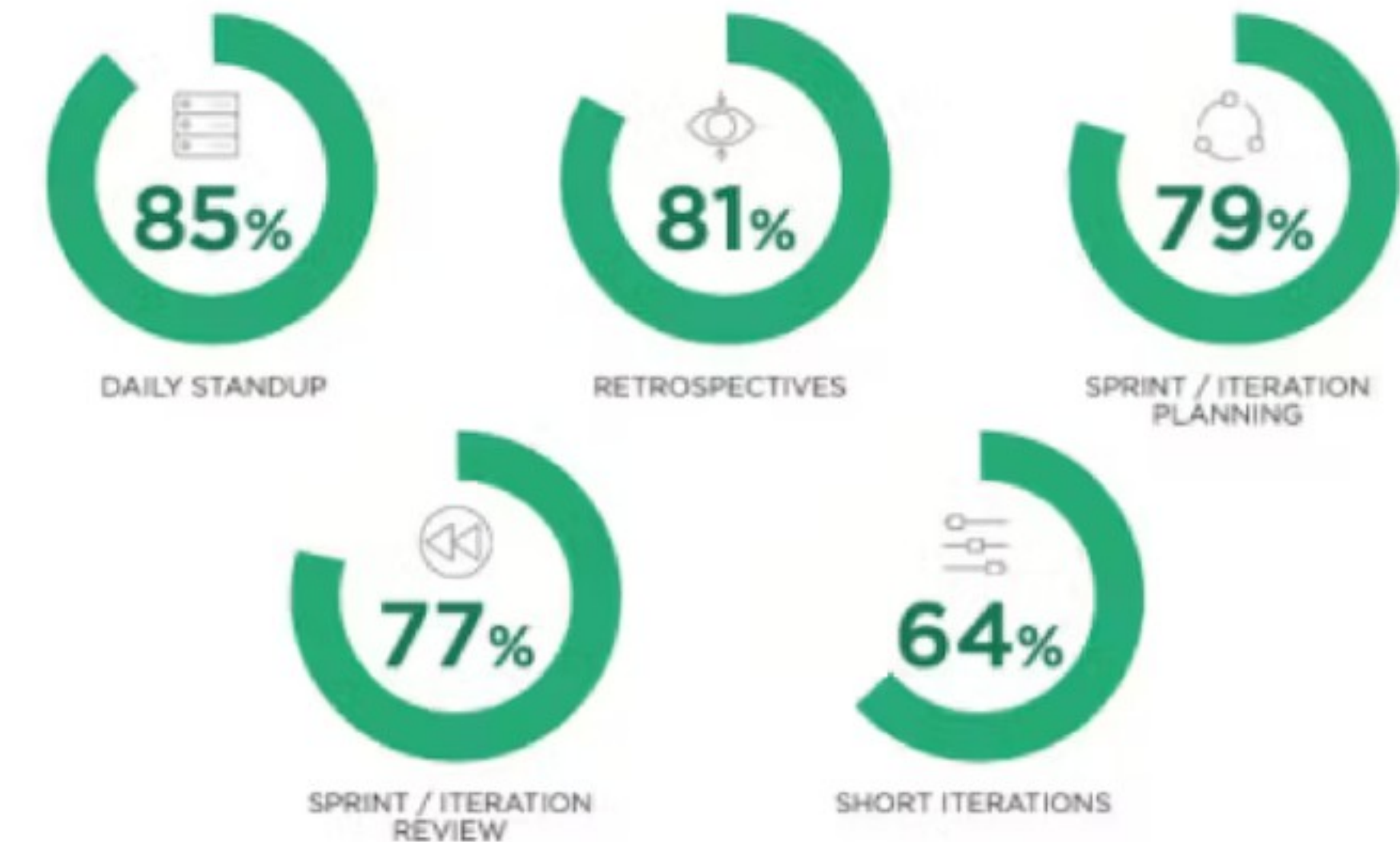
According to 14th annual state of agile report, the most popular agile practice is..



The most popular agile organizational practices



TOP 5 AGILE TECHNIQUES



(2024)

3.6 **Agile roles**

Briefly describe **five** Agile roles and what their responsibilities are:

Briefly describe "five" Agile roles and what their responsibilities are

Product owner: Leading the project and setting its objectives?

PO - owner of the product development team - develops the product
Scrum master - overview of the whole process
customer - the one the product are made for
product manager - keep the spirit up

Team members: task
excutive Manager: PO: manage backlog
Customer: make or modify requirements
Scrum master: hold daily stand up

Manager: assign tasks
PO: maintain product backlog
members: contribute to meetings and process
scrum master: the leader of scrum
customer: specific requirements and needs, communicating with PO

Team members: develop architecture through self organizing
Project manager: overview
Product owner: prioritize backlog
Developer:
Designer: Stakeholders:

Product owner - prioritiiize backlog
members - fully committed to the project
observers - involved, but from the sideline
manager - establish a good working environment

team member = develop things
product owner = backlog boss
project manager = manage customer = the one that wants something

Manager: Establish a good work environment.
Organize

Agile roles (chapter 5)

The self-organizing team

Manager

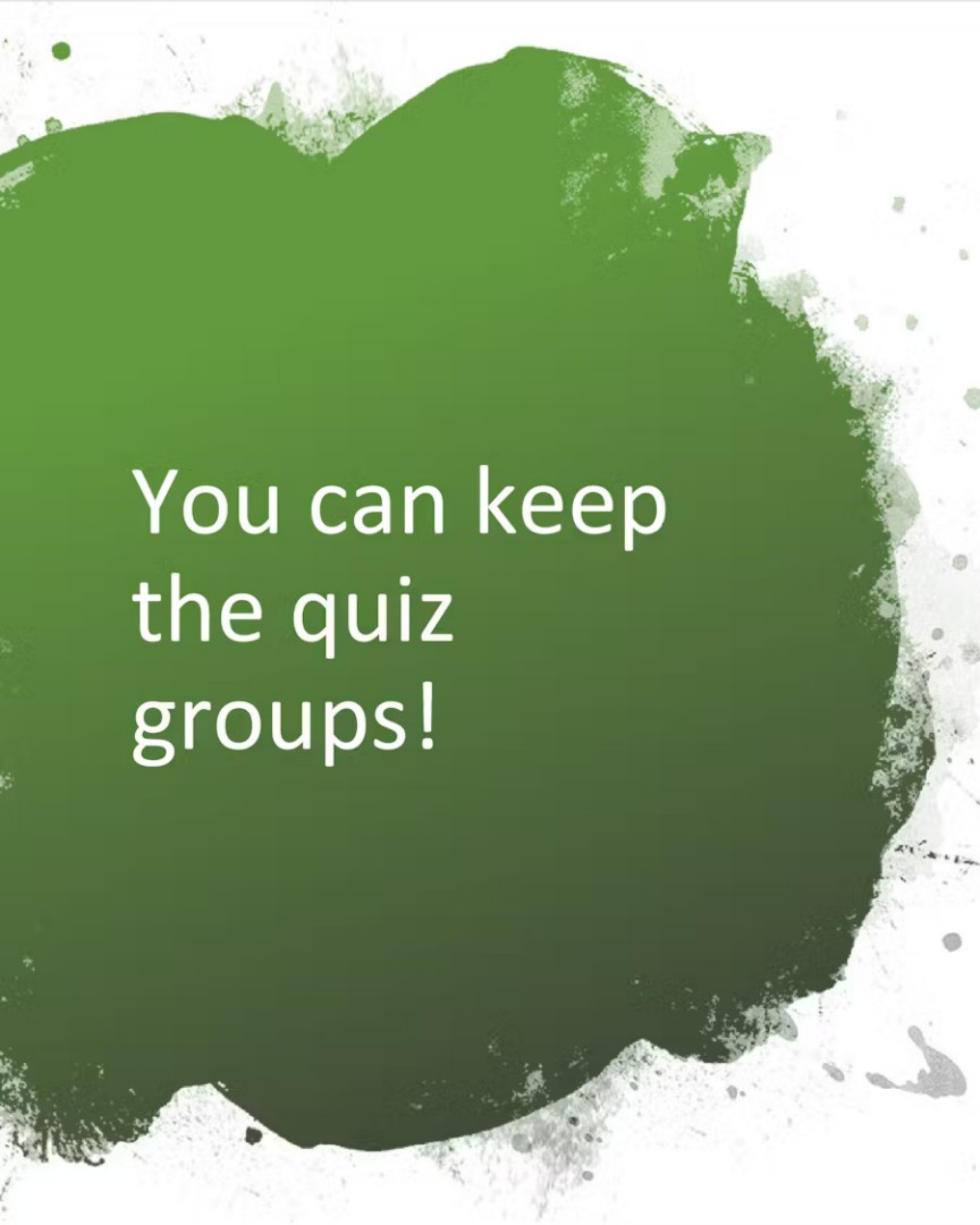
Product Owner

Members and observers

The customer

Agile coach/
Scrum Master

Discussion



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.

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.

You can keep
the quiz
groups!

Next Week

(Managing Technical- Social- and Process Debt)