Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

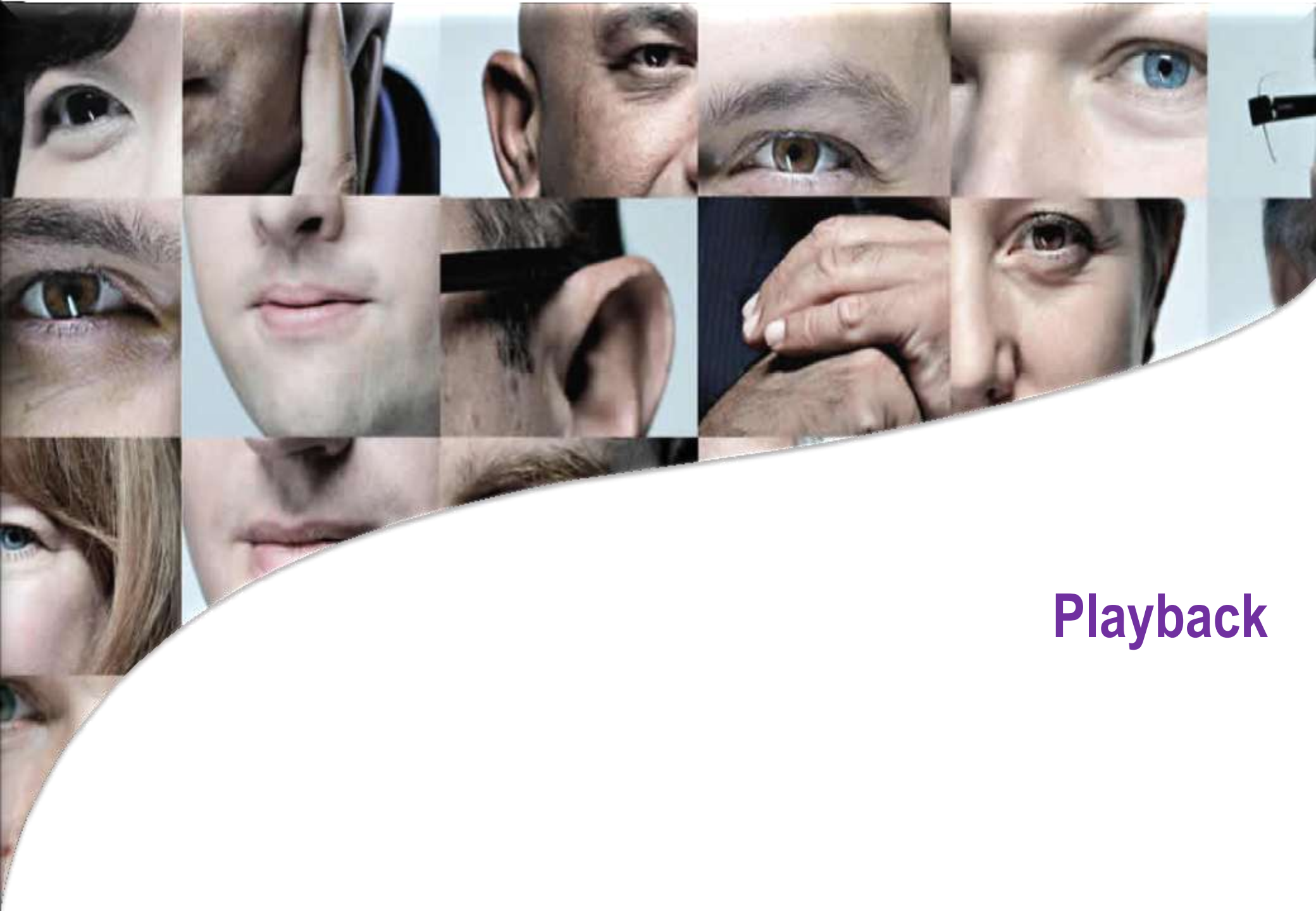
People matter, results count.

Agile Foundation and Introduction to Scrum

Day 4 – Scrum (Ceremonies)

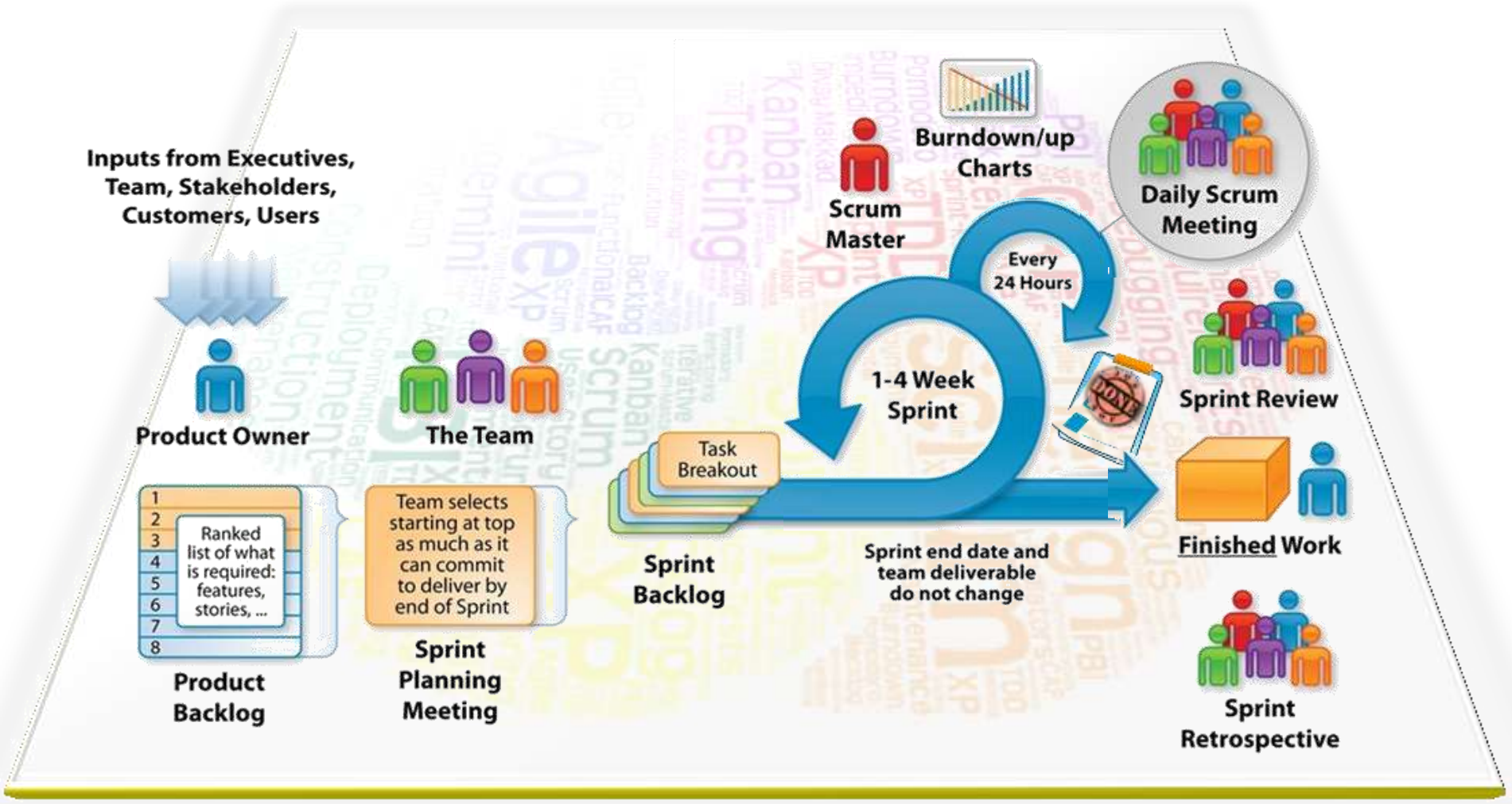
Divay Makkad

2017



Playback

Scrum Framework



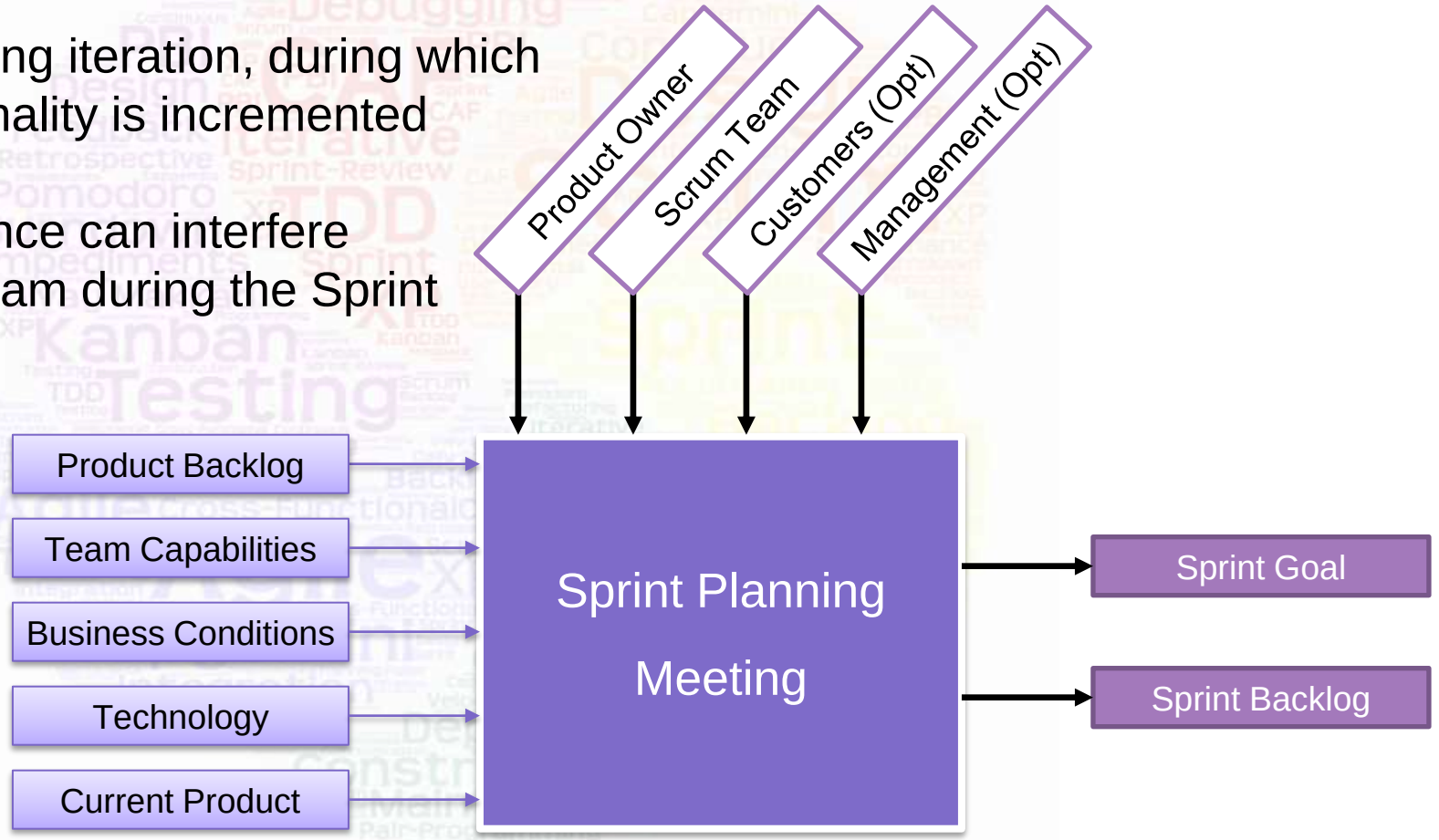
Sprint Planning Meeting

Sprint Planning Meeting

- Parameters
  - Beginning of the Sprint
  - 4 Hours for 2 week Sprint, 8 Hours for 4 week Sprint
- Participants
  - Development Team (Mandatory)
  - Scrum Master
  - Product Owner (Mandatory)
- Outcome:
  - Sprint Goal
  - Sprint Backlog

Sprint Planning Meeting

- Each Sprint begins with the Scrum Planning
- Up to a 4 week long iteration, during which a product functionality is incremented
- No outside influence can interfere with the Scrum team during the Sprint



Sprint Planning Meeting

- First activity of a new Sprint is the Sprint Planning and can be seen as two parts:
  - Part One - What will be delivered in this Sprint?"
  - Part Two - How the work needed to deliver will be achieved?"
- **Time, duration: begin of Sprint, each part up to max 4h**, depending on the length of the Sprint
- **Participants: Product Owner(s), Team, Scrum Master**
- Sprint Planning Meeting is time-boxed to max.
  8h (4h each part) for a max four week Sprint.
  For shorter Sprints this event is proportionately shorter.

Sprint duration [weeks]	Sprint Planning Meeting maximal duration [hours]
1	2
2	4
3	6
4	8

Sprint Planning Meeting

- 1st Part:
  - Grooming Product Backlog
  - Determining the Sprint Goal.
  - Estimating story size
  - May also be done as a separate meeting: (Product) Backlog Grooming Meeting
  - Participants: Product Owner, Scrum Master, Development Team

- 2nd Part:
  - Participants: Scrum Master, Development Team
  - Creating Sprint Backlog
  - Breaking Stories into tasks

When the Product Owner and the rest of the Scrum Team are in different locations, it is recommended to conduct the first half through video conferencing and for the second half, to allow the team to do the planning by itself, but the Product Owner should be available on the phone in case clarifications are needed.

Meeting Flow

- Check pending items from previous sprint (Any unfinished item should either go back into the product backlog or brought to current sprint if PO wants it to be done first.)
- Team Capacity information is gathered . (Team availability for the next Sprint, excluding holidays, trainings etc.)
- Product Owner presents ordered Product Backlog and shares a sprint goal for the team.
- Development Team tries to forecast the functionality to develop during the Sprint. Past velocity of development is being used to estimate possible capacity of items to choose from ordered Product Backlog
- Only the Development Team can choose the final number of selected items and estimate, what they can achieve in the upcoming Sprint.
- The Product Owner helps to make trade-offs if needed.

Meeting Flow

- During estimation, each story is discussed through the 3 Cs model (Cards, Conversation and Confirmation) and then estimated using an estimation technique e.g. Planning poker, T-Shirt Sizing
- Once all stories are estimated, the selected stories are then broken down further into granular tasks
- The team self organizes around the work and convey who would work on which story as per their core competency. Ultimately, whole team is responsible for completing all sprint stories. If not all items are assigned or planned, it will happen later - just-in-time.
- In formal settings, Product Owner can be asked to sign off the sprint plan.

Choosing a Sprint Goal

- The Sprint Goal is crafted – it's an objective that will be met during the Sprint. It guides the Team on why it is building the Increment
- For example the Goal can be: Allow user to manage their profile data.
- The Sprint Goal can be some kind of milestone to achieve (on the Release Plan)
- Realistic goals should be chosen

S - specific, significant, stretching

M - measurable, meaningful, motivational

A - agreed upon, attainable, achievable, acceptable, action-oriented

R - realistic, relevant, reasonable, rewarding, results-oriented

T - time-based, time-bound, timely, tangible, trackable

Daily Standup Meeting

Daily Standup Meeting

- Parameters
  - Daily
  - 15-minutes
  - Stand-up
  - Not for problem solving
- Participants
  - Development Team (Mandatory)
  - Scrum Master
  - Product Owner (Optional)
- Three questions:
  - What did you do yesterday
  - What will you do today?
  - What obstacles are in your way?

Daily Standup Meeting

- Is NOT a problem solving session
- Is NOT a way to collect information about WHO is behind the schedule
- Is a meeting in which team members make commitments to each other and to the Scrum Master
- Is a good way for a Scrum Master to track the progress of the Team
- Moderated by the Scrum Master. If there is a technical issue which warrants discussion, it's a good practice to just keep the issue aside in Parking Lot and take it up after Daily Scrum Meeting

Daily Standup Meeting

- Why daily?
  - "How does a project get to be a year late?"
    - "One day at a time." Fred Brooks, The Mythical Man-Month.
- Can Scrum meetings be replaced by emailed status reports? **NO**
  - Entire team sees the whole picture every day
  - Sense of ownership to do what you say you'll do

Sprint Review Meeting

Sprint Review Meeting

- Parameters
  - End of the Sprint
  - Up to 2 Hours for 2 week Sprint and 4 Hours for 4 week Sprint
- Participants
  - Development Team
  - Scrum Master
  - Product Owner
  - Any other business user, end user, stakeholders
- Outcome:
  - Acceptance of stories by Product Owner

Sprint Review Meeting

- Development Team presents its results, verifying what has been achieved,
- Meeting is open to all people – especially PO, Team and stakeholders (client),
- Scrum Master in consultation with Product Owners invites right stakeholders
- Product Owner identifies and confirms what has been “Done” and what not.
- Development Team discusses what went well, what problems they ran into and how they were solved (Brief user story development experience, should not be confused with Retrospective.)
- All team members should know and understand the Definition of Done(DoD).
- Don’t spend not to much time in preparation of Sprint Demo.
- Feedback from Sprint Review can lead to: revision of Backlog items, new features (if needed) not supplied Backlog items, changes in priorities, changes in Team structure, Release Sprint or end of project.

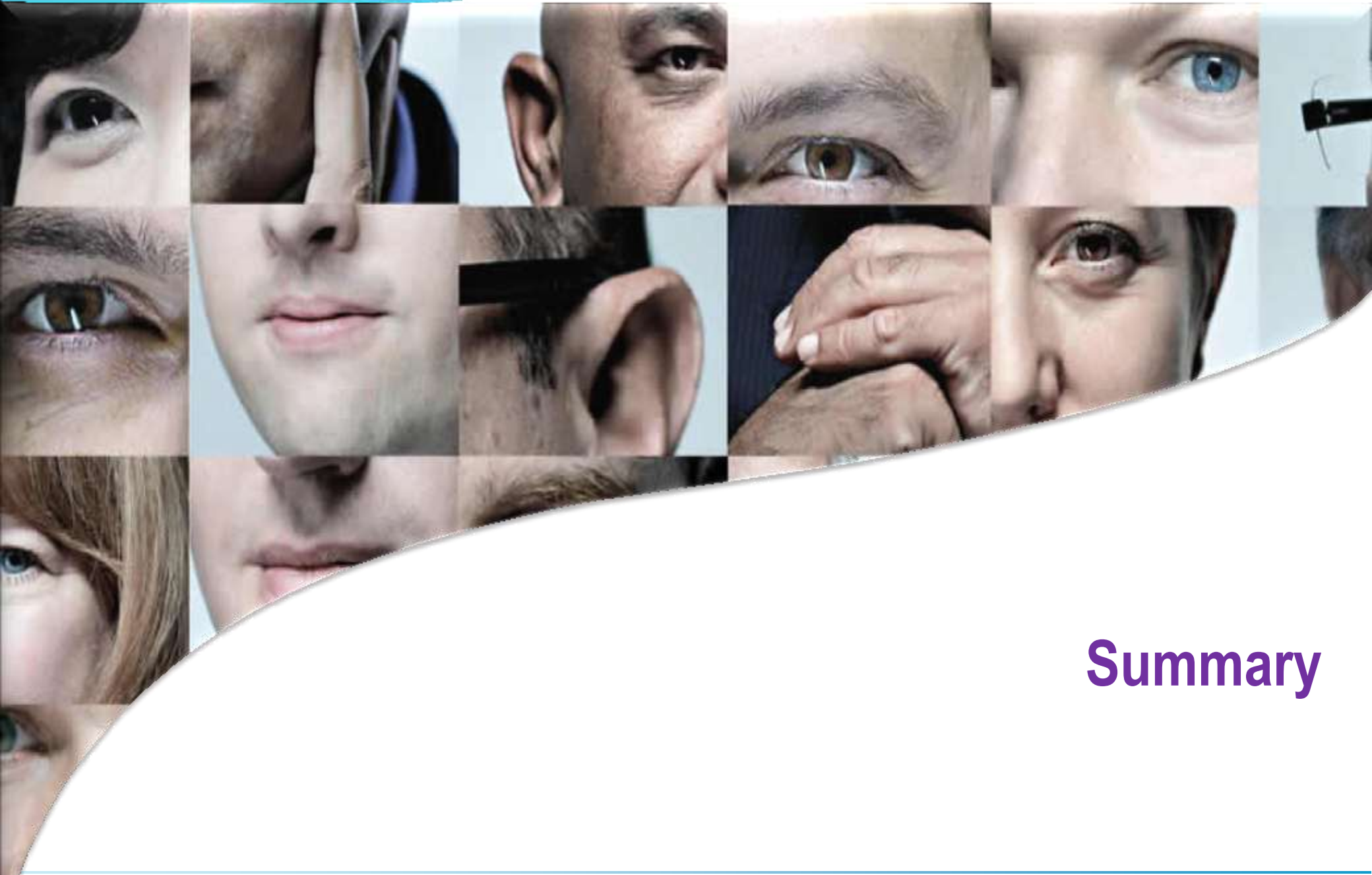
Sprint Retrospective Meeting

Sprint Retrospective Meeting

- Parameters
  - End of the Sprint
  - Up to 2 Hours
- Participants
  - Development Team
  - Scrum Master
  - Product Owner
- Outcome:
  - Action Plan
  - Agreements on improved ways of working

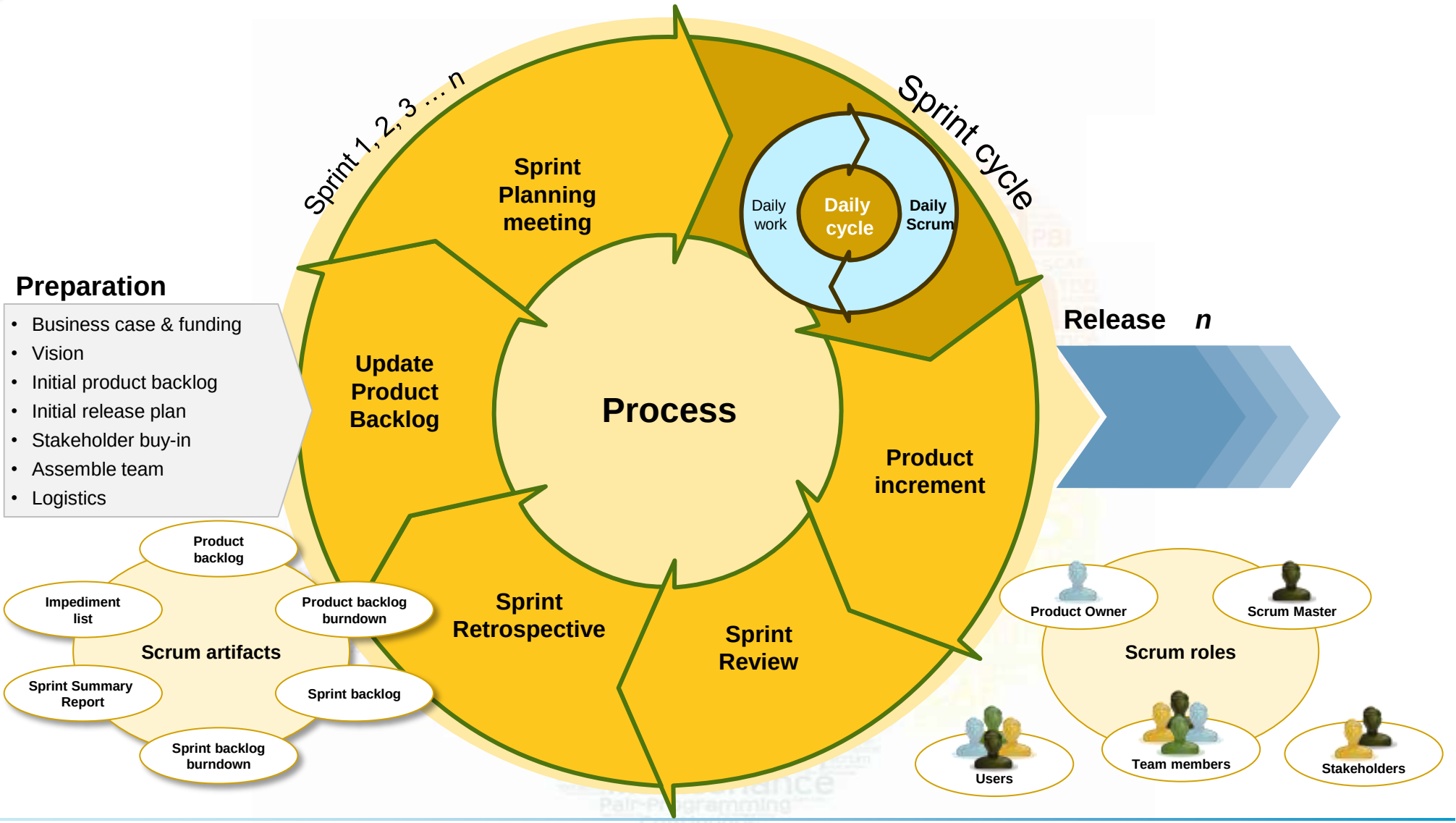
Sprint Retrospective Meeting

- Scrum Team has total discretion whether to allow the Product Owner to participate.
- The Product Owner presence should not influence exchange of opinions
- Don’t forget to review last retrospective
- Collect facts (most important events in the last Sprint)
- Inspect if the Sprint went well with regards to people, relationships, process and tools
- Identify and prioritize what went well and what should be improved
- Take decisions and create plan of improvements for the next Sprint
- Sprint Retrospective is a dedicated time to inspect and adapt
- Try to activate and engage all team members
- For interactive participation, try to facilitate the meeting using different retrospective activities.



Summary

Summary



Key Performance Indicators

Key Performance Indicators

1. **Actual Stories Completed vs. Committed Stories** – the team's ability to understand and predict its capabilities. To measure, compare the number of stories committed to in sprint planning with the stories identified as completed in the sprint review.

2. **Technical Debt Management** – the known problems and issues delivered at the end of the sprint. It is usually measured by the number of bugs, but may also include deliverables such as training material, user documentation and delivery media.

3. **Team Velocity** – the consistency of the team's estimates from sprint to sprint. Calculate by comparing story points completed in the current sprint with points completed in the previous sprint; aim for +/- 10 percent.

Key Performance Indicators

4. **Quality Delivered to Customers** – Are we building the product the customer needs? Does every sprint provide value to customers and become a potentially releasable piece of the product? It's not necessarily a product ready to release but rather a work in progress, designed to solicit customer comments, opinions and suggestions. This can best be measured by surveying the customers and stakeholders.

5. **Team Enthusiasm** – a major component for a successful scrum team. If teammates aren't enthusiastic, no process or methodology will help. Measuring enthusiasm can be done by observing various sprint meetings or, the most straightforward approach, simply asking team members "Do you feel happy?" and "How motivated do you feel?"

Key Performance Indicators

6. **Retrospective Process Improvement** – the scrum team's ability to revise its development process to make it more effective and enjoyable for the next sprint. This can be measured using the count of retrospective items identified, the retrospective items the team committed to addressing and the items resolved by the end of the sprint.

7. **Communication** – how well the team, product owner, scrum master, customers and stakeholders are conducting open and honest communications. Through observing and listening you will get indications and clues about how well everyone is communicating.

Key Performance Indicators

8. **Team's Adherence to Scrum Rules and Engineering Practices** – Although scrum doesn't prescribe engineering practices—unlike XP—most companies define several of their own for their projects. You want to ensure that the scrum team follows the rules your company defines. This can be measured by counting the infractions that occur during each sprint.

9. **Team's Understanding of Sprint Scope and Goal** – a subjective measure of how well the customer, product team and development team understand and focus on the sprint stories and goal. The goal is usually aligned with the intended customer value to be delivered and is defined in the acceptance criteria of the stories. This is best determined through day-to-day contact and interaction with the team and customer feedback.

ScrumBut – Good or Bad?

ScrumBut – Good or Bad?

- ScrumButs are reasons why teams can't take full advantage of Scrum to solve their problems and realize the full benefits of product development using Scrum.

- Every Scrum role, rule, and timebox is designed to provide the desired benefits and address predictable recurring problems.

- ScrumButs mean that Scrum has exposed a dysfunction that is contributing to the problem, but is too hard to fix.

- A ScrumBut retains the problem while modifying Scrum to make it invisible so that the dysfunction is no longer a thorn in the side of the team.

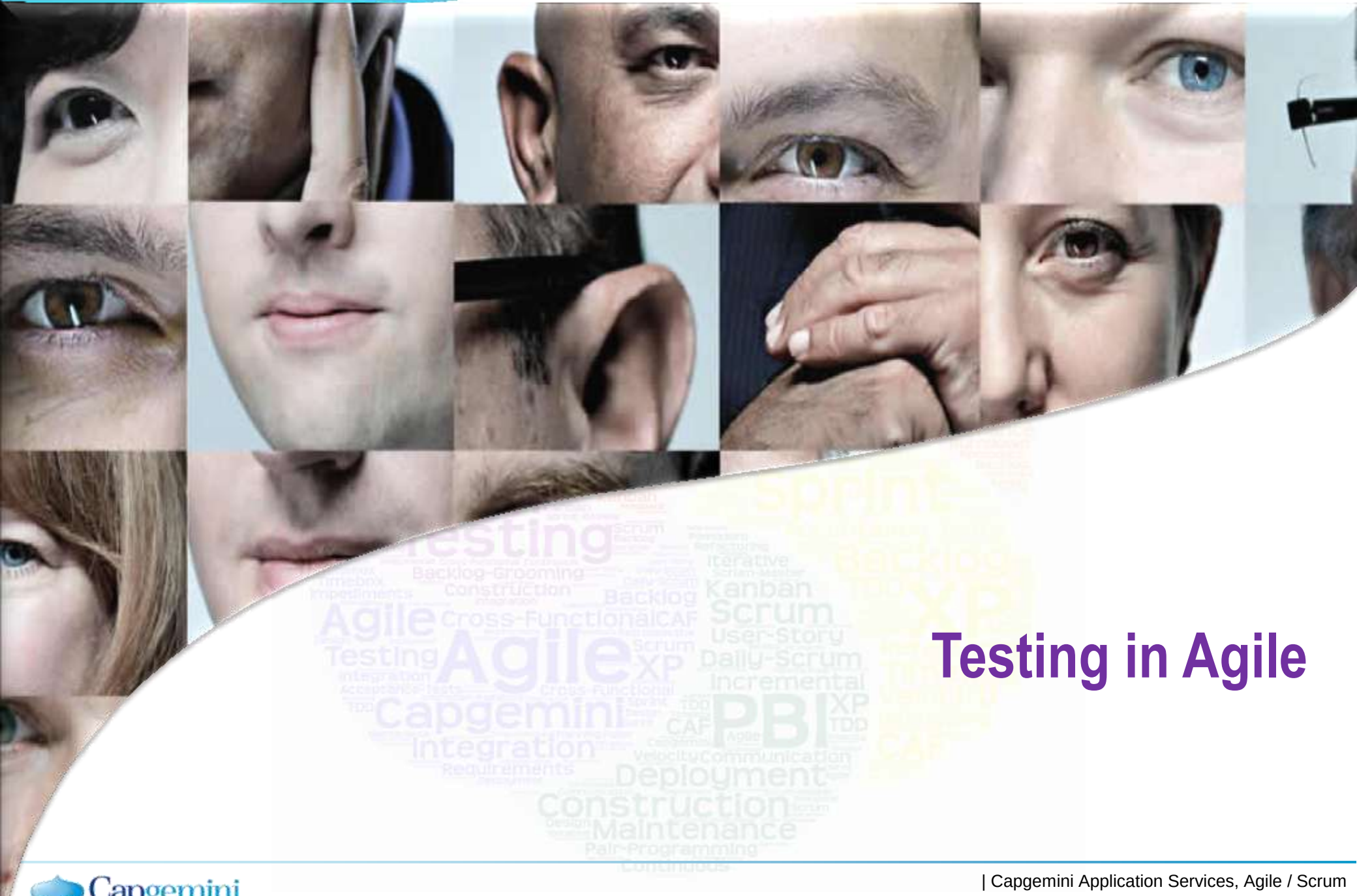
ScrumBut – Good or Bad?

A ScrumBut has a particular syntax: (**ScrumBut**)(**Reason**)(**Workaround**) e.g.**:**

- "(We use Scrum, but) (having a Daily Scrum every day is too much overhead,) (so we only have one per week.)"
- "(We use Scrum, but) (Retrospectives are a waste of time,) (so we don't do them.)"
- "(We use Scrum, but) (we can't build a piece of functionality in a month,) (so our Sprints are 6 weeks long.)"
- "(We use Scrum, but) (sometimes our managers give us special tasks,) (so we don't always have time to meet our definition of done.)“

Sometimes organizations make short term changes to Scrum to give them time to correct deficiencies.

For example, "done" may not initially include regression and performance testing because it will take several months to develop automated testing. For these months, transparency is compromised, but restored as quickly as possible.



Testing in Agile

Testing in Agile

1. Tester is a part of each team – as a role in the team, or as a dedicated member specialized in testing.
2. Testing is a part of each task/story – preparing a test should be a normal task in your Sprint. Definition of “Done” must include "tested”.
3. Focus on automation possibilities
   - Test-minded developers concentrate on designing better code – easier to be tested. Writing automation scripts is simple for them, because they have appropriate knowledge and skills.
   - Use test automation on each level of testing pyramid (unit tests, acceptance, regression, GUI tests, etc.).
   - Letting lots of test to be executed automatically saves time for important, really exploratory manual testing.
4. Agile-minded testers know how to apply agile principles in order to provide a better product. They have also wider view on business perspective than programmers, who usually concentrate on the current story.

Testing in Agile

1. Set up testing environment:
   - Set up continuous integration as soon as possible in your project – configure your builds and tests, so that they are verified at least daily. Project environment provides a proper technical solution for that. Appropriate bug tracking systems are also available.
   - Be able to run special test builds (e.g. with more detailed logging)
   - Create and use database schema for testing with realistic data, also very extreme (super large, or empty). There should be separate schemas for testing, integration, development.
2. Assure that your technical environment and tests design provide feedback in short cycle.
3. Prepare a communication plan/channels with client to be able to give quick satisfaction-feedback after each demo

Testing in Agile

Agile vs. Waterfall testing

- In opposite to waterfall, testing Agile Testing is integrated throughout the lifecycle
- Each feature is fully tested as it's developed rather than at the end of all the development.
- Testers know the examples given by customers. In this way, they can write test which help developers better understand the current use case (story).

Ideal world: You don't need additional acceptance test phases since you release production-ready version with each Sprint.

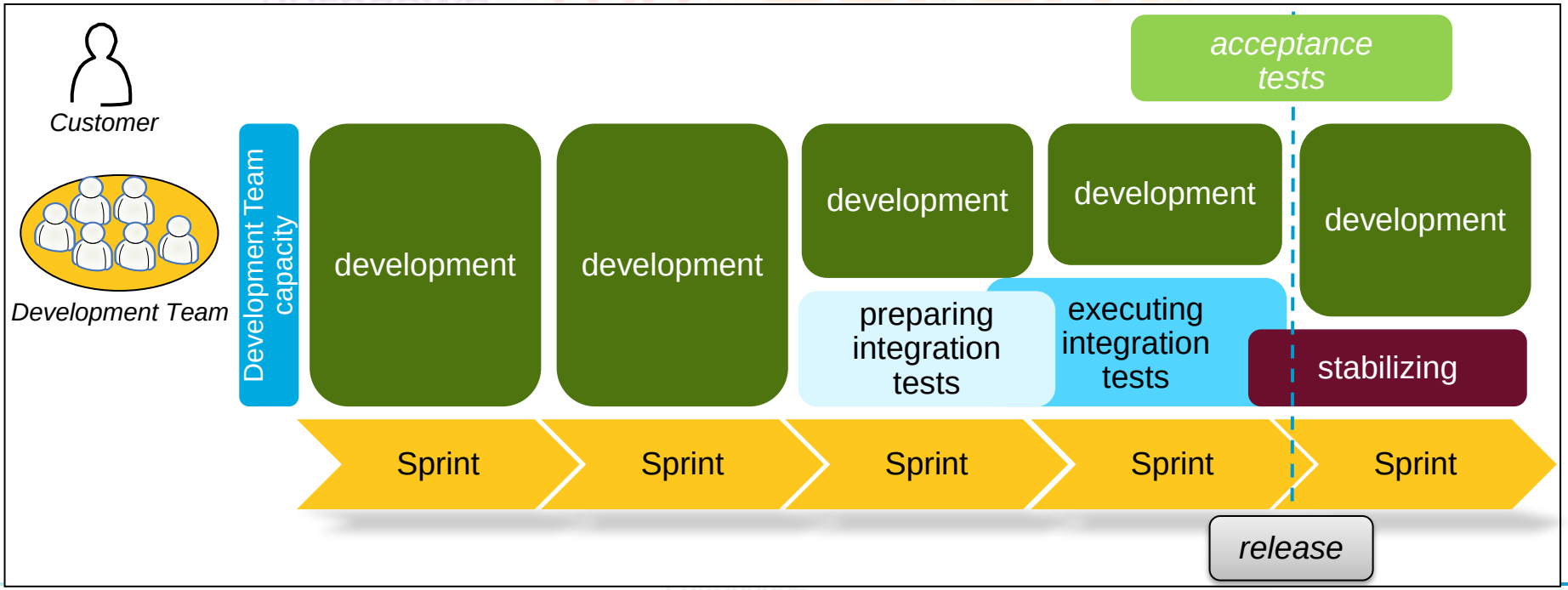
Reality: After the Sprint release, end-users report some bugs. How to deal with them?

Solution 1: Add non time-boxed release period between Sprints dedicated only to testing, debugging and fixing, until the product is really ready (drawback: deregulated Sprint pace)

Solution 2: Release at the end of each Sprint. While planning the next Sprint dedicate some time for bugfixing tasks at the beginning of the Sprint. Assure, that the Sprint is long enough so that there is also much time for normal tasks. Optionally larger bugs can be added to the product backlog and prioritized.

Testing in Agile

- Integration tests can be prepared as part of development or separately.
- Integration tests must be executed before software is released.
- Capacity reserved for Dev is reduced when preparing and executing integration tests and for stabilizing
- Found bugs can be fixed immediately (in time planned for test execution) or planned as Spring Backlog artifacts and fixed during stabilization.
- Parallel to the Team, Customer can execute acceptance tests and report bugs that are planned as Sprint Backlog artifacts and fixed in time planned for stabilization.



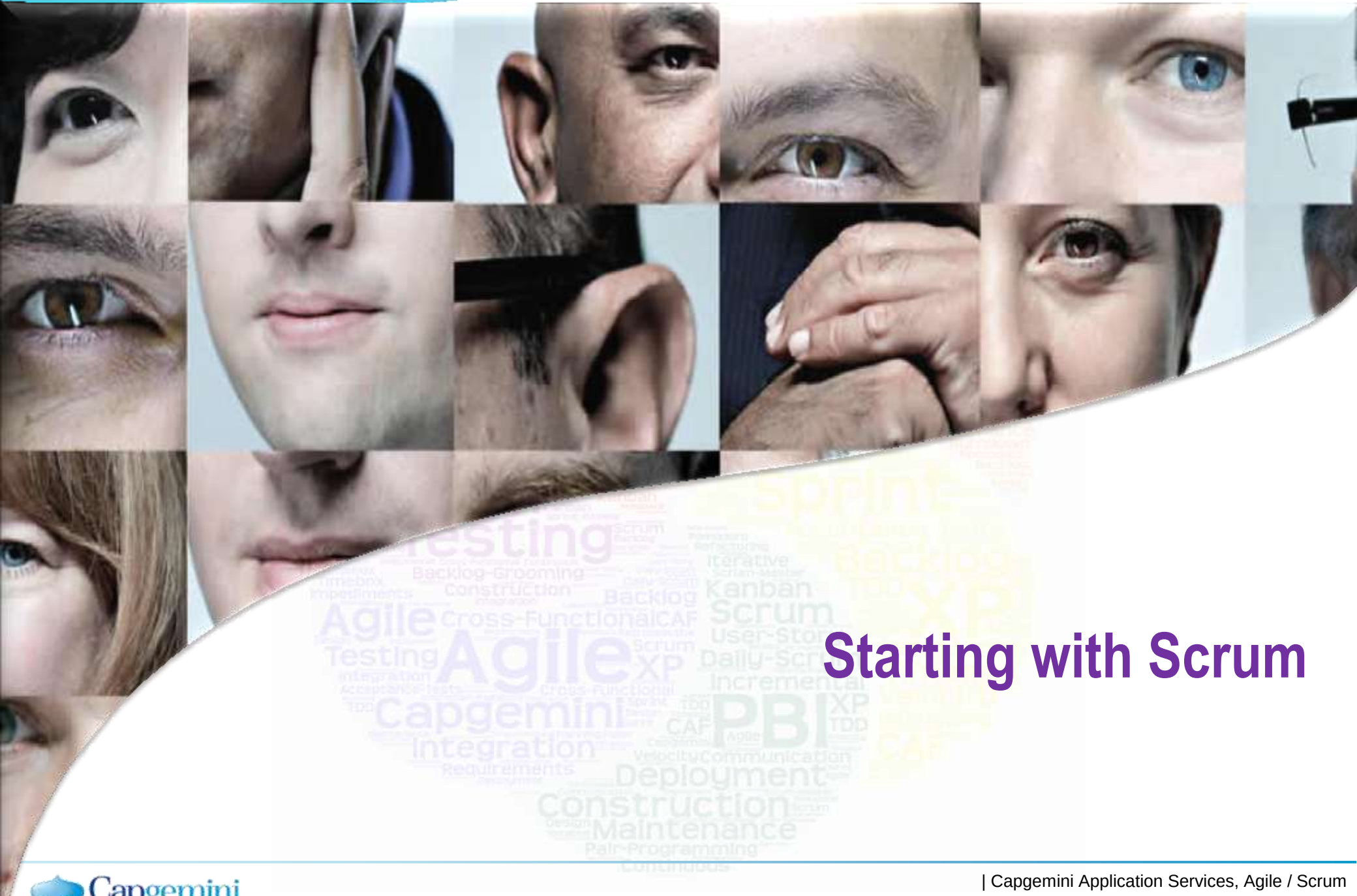
Testing in Agile

Don’t focus only on new stuff.

Remember that when the code is finished, you are probably still far away from production release. Don’t push too much with implementing new stories, when the old almost-ready things are not well tested or fixed.

Be careful with technical debt.

- Avoid using only quick fixes and hacks to have the problem solved, because it leads to have more complicated (and “ugly”) code hard to maintain.
- Choosing the easiest solution now means that you are taking a technical debt that sooner or later will have to be paid off.
- Try to assure time (budget) for **“refactoring Sprints”.**
- If it is not possible, try to find a person in your team that can fully engage only in refactoring from time to time.
- Don’t be afraid, that the changes would spoil the current system state – with good test coverage and short automation feedback cycle, your system is well protected and every possible negative change is immediately noticed (and can be fixed).



Starting with Scrum

Starting with Scrum

- Identify all parts that will be involved in the project
- Consider following aspects:
  - business (functional) requirements
  - company's IT-architecture
  - system operation and maintenance requirements
  - organizational and financial issues

- Decide if the team should be distributed/co-located
- Select people for the Scrum Roles: Product Owner, Scrum Master, Development Team
- Define the Product Owner, who will take responsibility for the whole project
- Ensure that the team has sufficient know-how, train the team members if needed or arrange for an agile coach
- Ensure proper mix of skills (also soft skills) in the team

- Product Owner(s) gathers ideas for the functionality desired in the product
- A list with short description of each functionality is created
- The list is sorted by the priority of the features

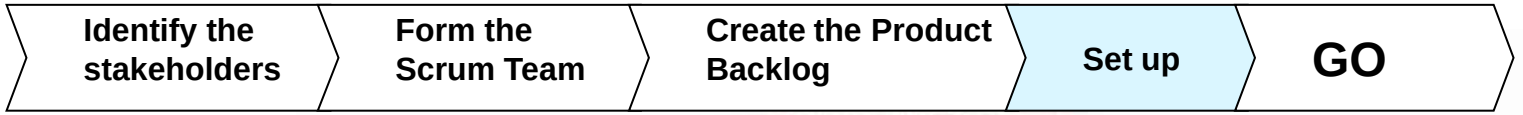
- Organize room
- Arrange virtual meeting place
- Arrange development and testing environment
- Create a time plan
- Decide general rules

(More on Next Slide)

- Inspect and adapt:
  - Let the team self-organize their work but
  - Inspect the results of the team work
  - Identify and solve or escalate the problems

With every stage the involvement of the Scrum Team members in the preparation steps rises

Starting with Scrum



If the team sits together

- Organize common room for the team, book (video-)rooms for the meetings in advance

Distributed teams: Arrange a virtual meeting place

- Arrange telephone conference environment (headsets!)
- Arrange videoconference environment (test the quality!)
- Arrange desktop sharing tool

Arrange development and testing environment

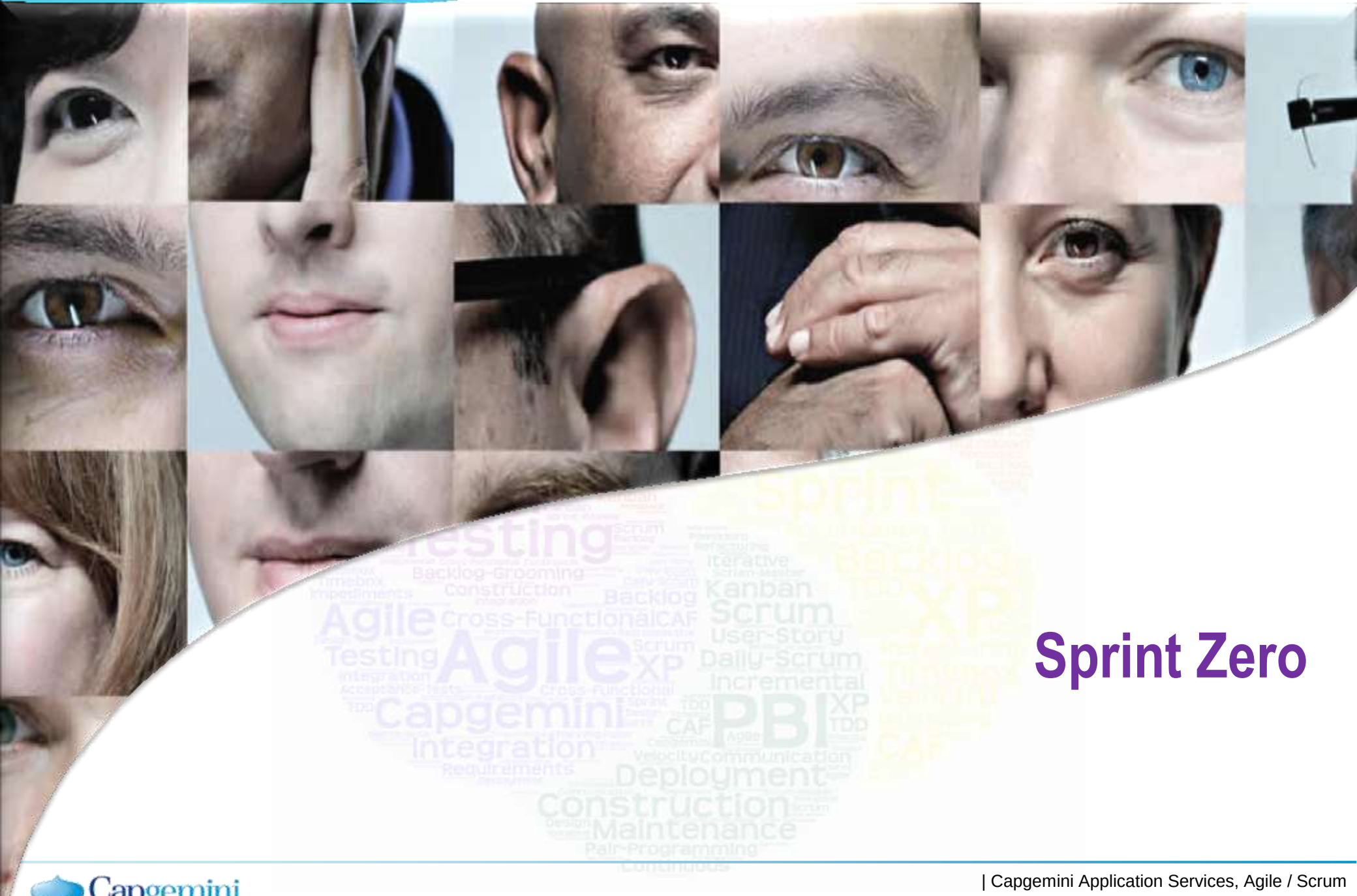
- In which location? How can people from other locations access the environment?
- Tools for automation: continuous integration, Testing, Code quality, continuous delivery

Create a time plan

- Decide the kick-off date
- Decide the Sprint length (optionally shorter cycles at the beginning)
- Plan the journey for distributed teams: the first Sprint or even two should be performed together

Decide the general rules in the project, e.g. …

- Who from the Product Owner's Team will take part in daily meetings?
- How will the Scrum meetings be held – telephone? Video? No need for remote meetings?



Sprint Zero

Sprint Zero

The idea of Sprint Zero has been used & abused. Some reasons for doing Sprint Zero:

- The team needs to be assembled in Sprint Zero.
- Organization and logistics need to be set up in Sprint Zero.
- Consider planning, product backlog setup, and design.

Introducing a long Sprint Zero at the start and a long validation (Hardening Sprint) at the end helps keep the love for Waterfall alive. In addition, it hinders the effectiveness of Scrum.

Scrum believes that every sprint should deliver potentially usable value.

Sprint Zero

A working definition of Sprint Zero:

- Sprint Zero should be used to **create basic skeleton and plumbing** for the project so that future sprints can add incremental value efficiently. It may involve some research spikes.

- **Minimal design up front** is done in Sprint Zero so that **emergent design** is possible in future sprints. This includes creating a flexible framework so that refactoring is easy.

- For minimal design up front, the team picks **very few critical stories** and completes them. Delivering them includes putting the skeleton/framework in place, but it still delivers value.

Myths & Pitfalls

Myth #1: Scrum is Waterfall, only better.

Myth #2: Scrum means that there is no documentation

Myth #3: Scrum projects are completely unmanaged.

Myths & Pitfalls

1. **Agile Transition Pitfalls**
   - Not Educating Stakeholders Outside of the Core Team
   - Fear of Failure
   - Assuming That Agile = Faster

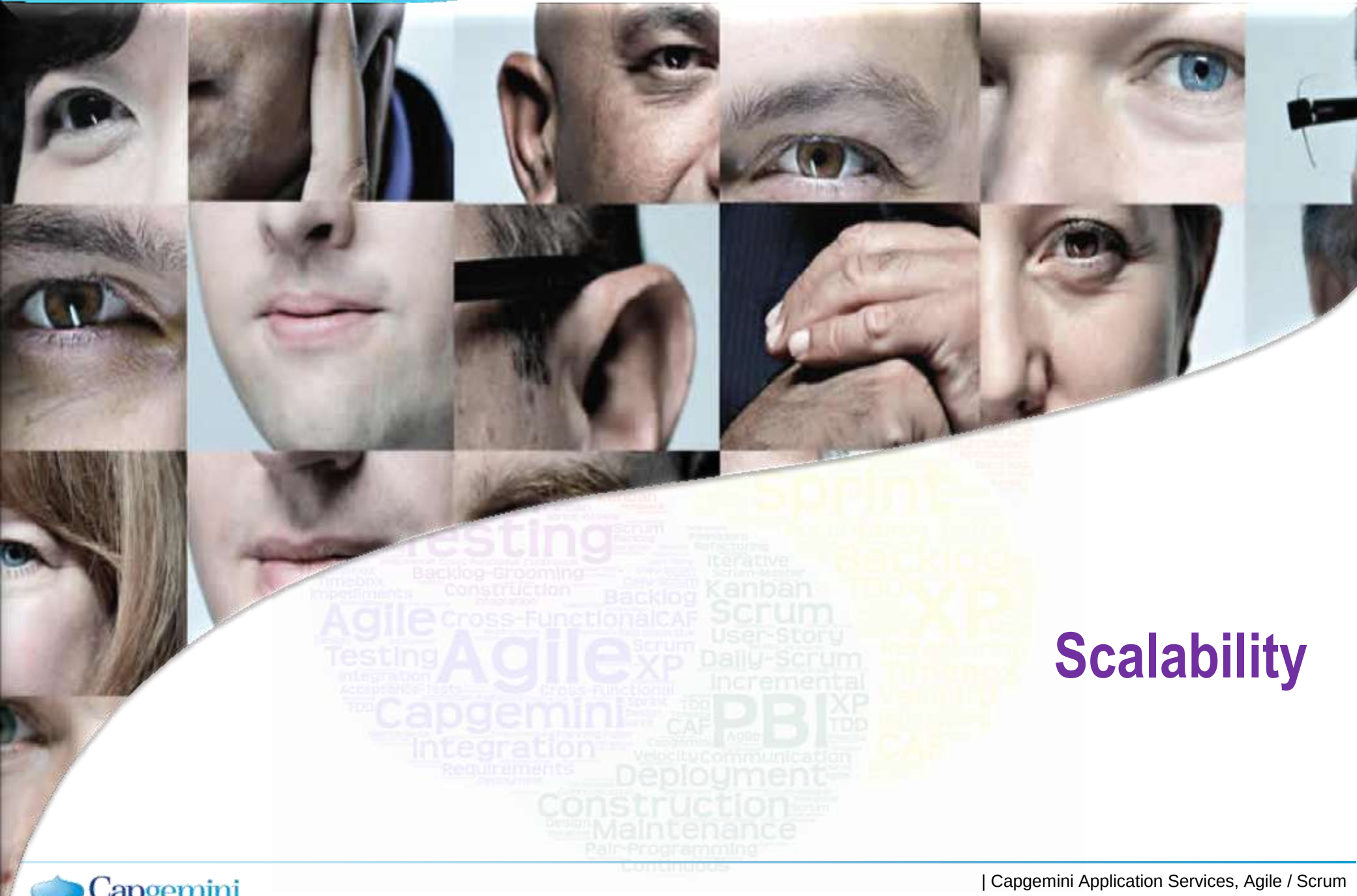
2. **Estimating / Design pitfalls**
   - Planning Every Iteration for the Project In Advance
   - Estimate Anchoring
   - Overcommitting Velocity
   - Trying to Estimate Velocity Perfectly
   - Planning for Longer Iterations so “More” Work Can Get Done
   - Forgetting Artifacts

3. **UAT and Release Pitfalls**
   - UAT Scripts are Developed at the End of the Last Sprint
   - Giving up on Test Coverage Because of Imposed Deadlines, Scopes, and Resources
   - Product Owner is not Engaged in Reviewing All Test Scripts

Myths & Pitfalls

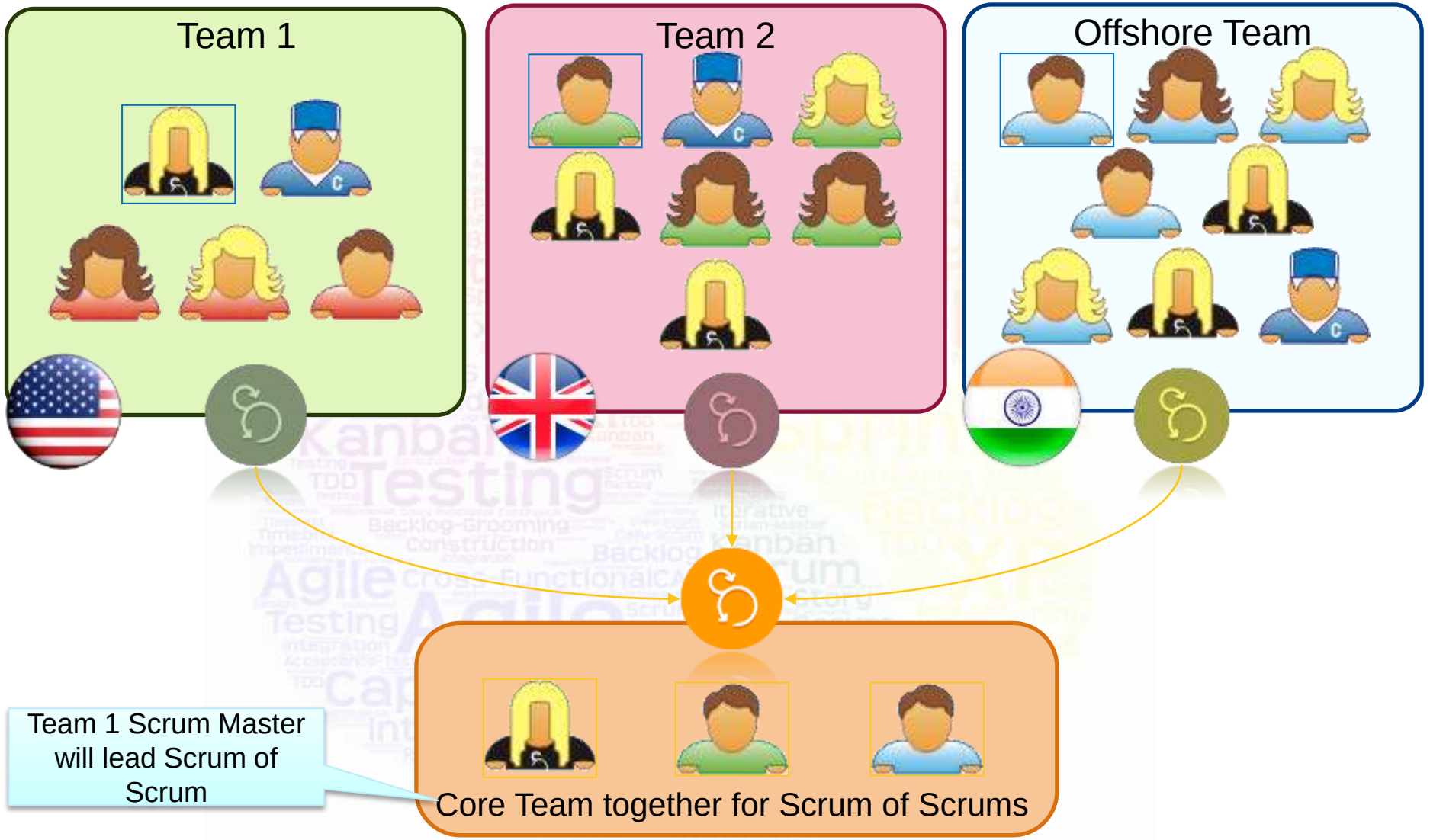
4. Development Pitfalls

- Seeing the Tree but Not the Forest
- Completing Sprints While Leaving Behind Technical Debt
- Problem Solving in the Daily Scrum
- Scrum Master as a Contributor
- Assigning Tasks
- Product Owner Mismatch
  - Highly available
  - Knowledgeable
  - Empowered
- Adding Stretch Goals to Sprints
- Individual Heroics to “Save” the Sprint
- Development Team Organizes Product Backlog
- Product Owner Specifies Solutions
- Not Allocating Enough Time for the Demo
- Using Metrics to Measure Productivity
- “Duck-Taping” the Demo
- Giving up on Quality Because of Imposed Deadlines, Scopes, and Resources



Scalability

Scalability



Scalability

Scrum of Scrums Does :

- Meet once a day after all involved sub teams have conducted their Daily Standup
- Mirror the team level Daily Standup.15 minutes or less with a potential 15minute follow up conversation (often called the parking lot) to flesh out impediments and/or possible solutions. Those not involved may get back to work.
- Remove impediments for individual teams if possible. If not, they raise them to the next Scrum of Scrums and so on until the impediments are removed or are raised to the Enterprise Action Team (EAT).
- Works with the EAT to help coordinate best practices across the entire enterprise.

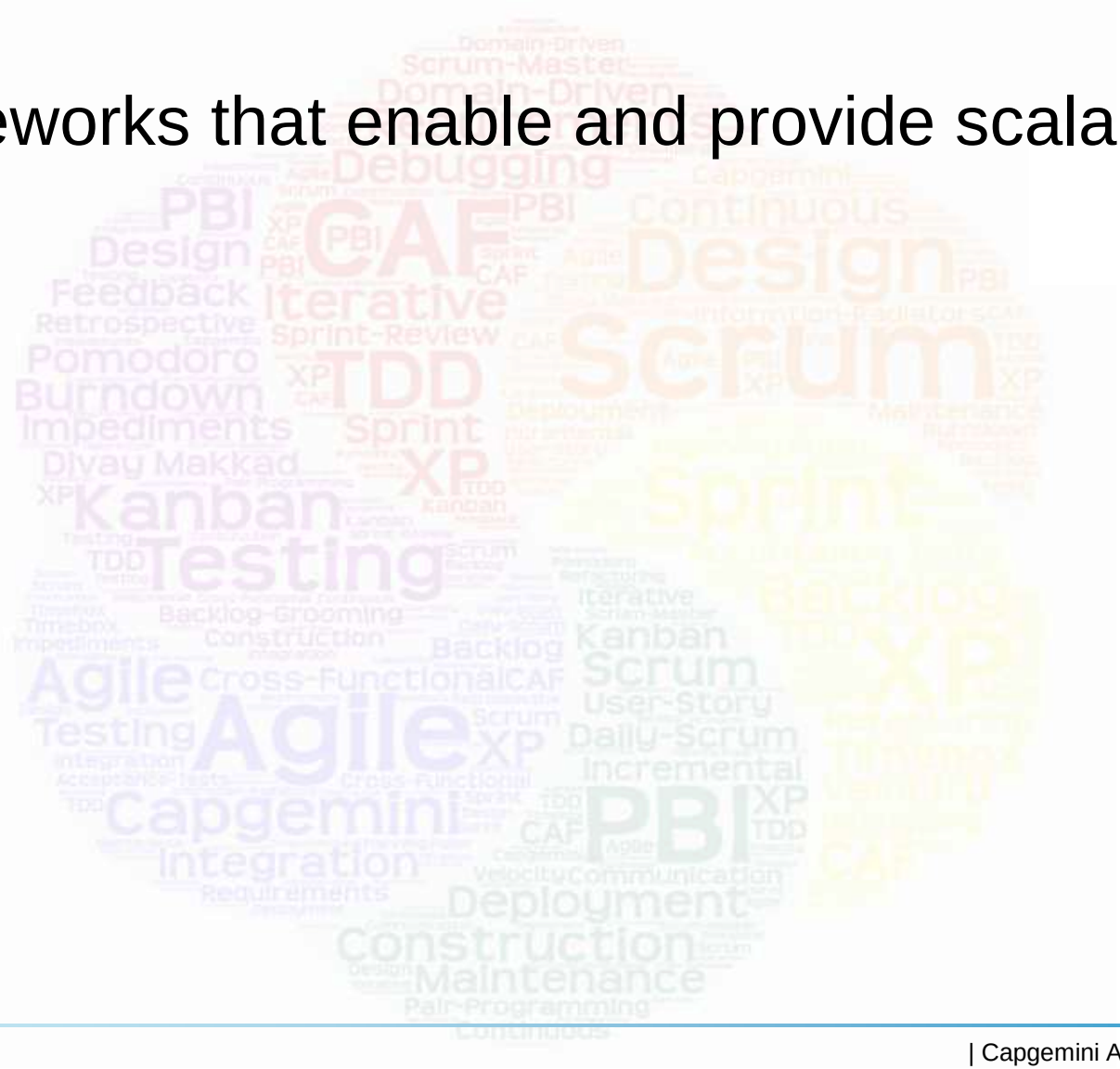
Scrum of Scrums Does not:

- Develop backlog. However, may add stories to backlogs.
- Decompose Backlog
- Coordinate Epics
- Plan releases
- Manage releases

Scalability

Scrum Frameworks that enable and provide scalability

- LeSS
- SAFe
- Nexus
- DAD



Resources

Scrum video:
http://www.youtube.com/watch?v=vmGMpME_phg

Scrum resources:
http://www.scrumalliance.org/resources

Scrum user guide:
http://www.scrum.org/storage/scrumguides/Scrum%20Guide.pdf

Key Scrum Concepts:
http://www.scrumalliance.org/pages/what_is_scrum

More videos:
http://www.infoq.com/presentations/Agile-Management-Google-Jeff-Sutherland

http://video.google.com/videoplay?docid=-7230144396191025011#

The Ballpoint Game

The objective of the game is to pass as many balls as possible through the team in 2 minutes. There are relatively few rules, but they must be adhered to. Here they are:

1. If you've played earlier, please participate silently so you don't spoil it for others.
2. You are one big team – you cannot change your team size.
3. Every team member must touch each ball for it to count.
4. As each ball is passed between team members, it must have air time, i.e. It must not be passed directly from hand to hand.
5. You cannot pass the ball to the person immediately to your left or right.
6. If you drop a ball, you cannot pick it up.
7. There will be a penalty (points deducted) if you break any of the rules.
8. Every ball must end where it started. For each ball that does, team scores 1 point.

Q n A

Thank you!