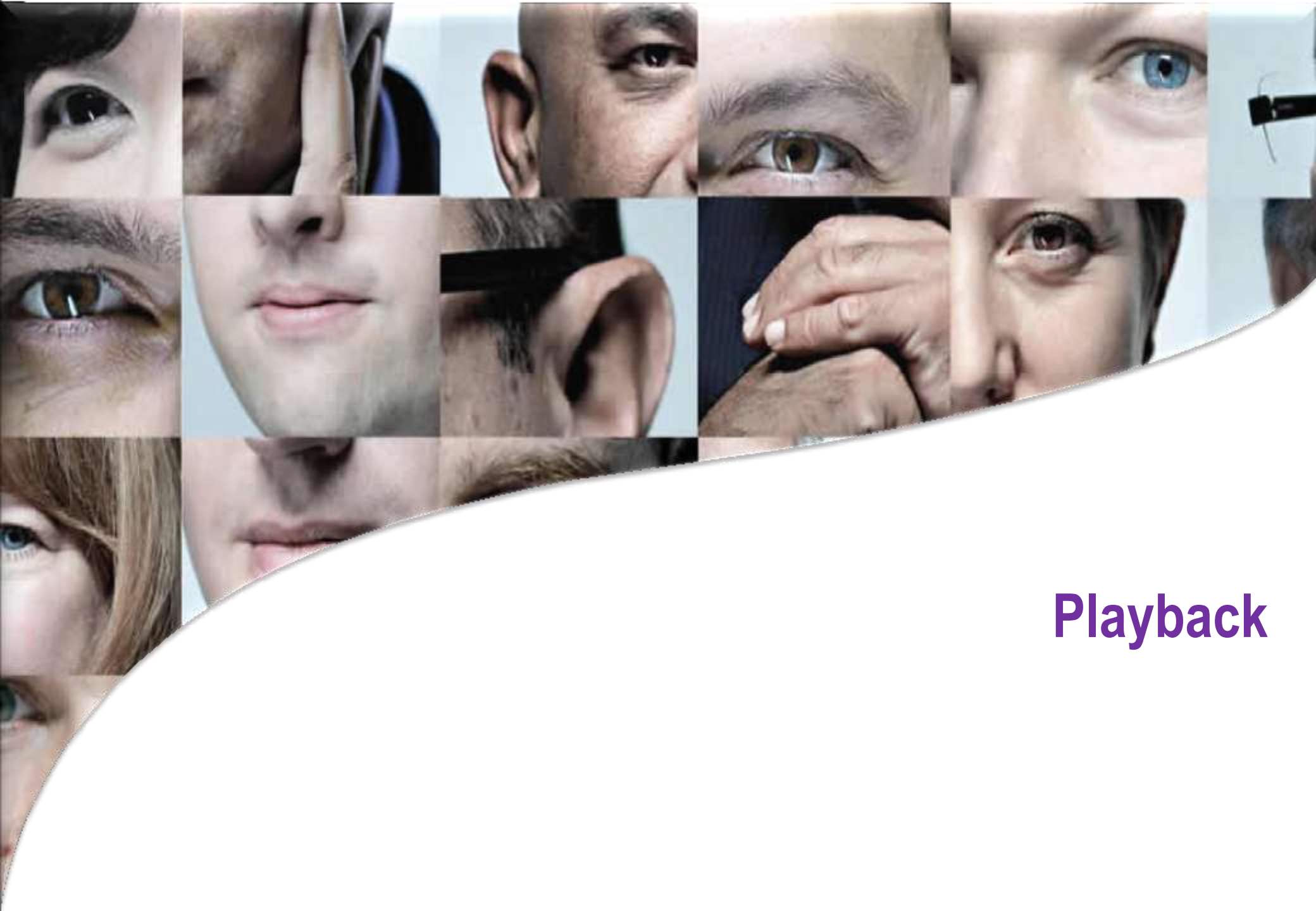


Agile Foundation and Introduction to Scrum

Day 2 – Scrum (Roles)

Divay Makkad
2017





Playback



Scrum



Scrum Framework & its Principles



What is Scrum?

Definition from rugby football:

A scrum is a way to restart the game after an interruption, where the forwards of each side come together in a tight formation and struggle to gain possession of the ball when it is tossed in among them





History of Scrum

- 1995:
 - Analysis of common software development processes → not suitable for empirical, unpredictable and non-repeatable processes
 - Design of a new method: Scrum by Jeff Sutherland & Ken Schwaber
 - Enhancement of Scrum by Mike Beedle & combination of Scrum with Extreme Programming
- 1996:
Introduction of Scrum at OOPSLA conference
- 2001:
Publication “Agile Software Development with Scrum” by Ken Schwaber & Mike Beedle



History of Scrum

- Jeff Sutherland
 - First person to apply concepts of Scrum to software development
 - A variation of Sashimi
 - Japanese designed an all at once approach after their bad experiences with Waterfall
- The initial use of the word "Scrum" was used in 1987 to describe time-boxed, self-organizing, teams in product development
- Jeff Sutherland and Ken Schwaber
 - collaborated to define the process through 1995. In 1996 wrote the seminal article for Scrum Software Development process
 - jointly used and improved Scrum at a variety of software development organizations from 1996 until now



Scrum – An Agile Process

- SCRUM is an agile, lightweight process for managing and controlling software and product development in rapidly changing environments.
- Iterative, incremental process
- Team-based approach
- developing systems/ products with rapidly changing requirements
- Controls the chaos of conflicting interest and needs
- Improve communication and maximize cooperation
- Protecting the team from disruptions and impediments
- A way to maximize productivity

Scrum Framework





Scrum Principles

- 1. Empirical Process Control** - This principle emphasizes the core philosophy of Scrum based on the three main ideas of transparency, inspection, and adaptation.
- 2. Self-organization** - This principle focuses on today's workers, who deliver significantly greater value when self-organized and this results in better team buy-in and shared ownership; and an innovative and creative environment which is more conducive for growth.
- 3. Collaboration** - This principle focuses on the three core dimensions related to collaborative work: awareness, articulation, and appropriation. It also advocates project management as a shared value-creation process with teams working and interacting together to deliver the greatest value.



Scrum Principles

4. **Value Based Prioritization** - This principle highlights the focus of Scrum to deliver maximum business value, from beginning early in the project and continuing throughout.
5. **Time-boxing** - This principle describes how time is considered a limiting constraint in Scrum, and used to help effectively manage project planning and execution. Time-boxed elements in Scrum include Sprints, Daily Standup Meetings, Sprint Planning Meetings, and Sprint Review Meetings.
6. **Iterative Development** - This principle defines iterative development and emphasizes how to better manage changes and build products that satisfy customer needs. It also delineates the Product Owner's and organization's responsibilities related to iterative development.



The Scrum Master



Scrum Master

- Acts as a Servant Leader (Facilitator) for the team
- Represents management to the project
- A specific role with well articulated responsibilities
- Responsible for enacting scrum values and practices
- Remove impediments
- Shields the team from external distractions



The Servant-Leader Role



Servant Leader

- Behind the name is a role, and a pattern of duties and responsibilities.
- The name "Scrum Master" represents a pattern known as *Servant Leadership*.
- A Servant Leader facilitates a team not by telling them what to do, but by removing impediments that get in their way and coaching them in agile best practices.



Servant Leader

What does a Servant Leader do?

- Shields the team from diversions and distractions
- Facilitates planning sessions
- Facilitates reviews and retrospectives
- Coaches the team in agile best practices
- Helps the team to collaborate better
- Advocates the team's position
- Anticipates impediments for the team
- Finds ways to remove team impediments
- Makes sure daily standups happen, and are conducted properly
- Encourages transparency and associated metrics
- Understands and explains the team's progress to interested stakeholders
- Arbitrates between team members when necessary



Servant Leader

What a Servant Leader Doesn't do?

- Direct the team by telling them what to do. Instead, he or she should assist the team to self-organize and do what it takes to expedite their progress
- A Scrum Master should simply ensure that the standup happens and is conducted properly.
- Estimate the team's work. If an agile team are using estimates in their planning, they must be responsible for them. The Scrum Master will arbitrate if needed.
- Go off his or her own sweet way. A good Scrum Master will always maintain an awareness of where the team is in relation to its goals.



Scrum Master Routine

Start of the day

- Update and review the current Burndown chart for the current Sprint, for each Team of interest
 - If a Team is running behind schedule, investigate, and try to help bring it back on track.
- Review the Sprint Backlog Items and their associated tasks
 - Check to see if any tasks have missing information
 - Look for inconsistencies
- Follow up with appropriate Team members, to ensure they supply missing information, and correct inconsistencies

During the Day

- Identify any issues that are impeding progress
- Assist Team members to resolve issues
- Facilitate the Daily Scrum meeting
- Review new User Stories, Technical Stories, and Defects added to the Product Backlog

End of Day

- Same as Start of Day: Review status, look for missing or inconsistent information



The Product Owner



Product Owner

- Define the features of the product
- Decide on release date and content
- Be responsible for the profitability of the product (ROI)
- Prioritize features according to market value
- Adjust features and priority every iteration, as needed
- Accept or reject work results.



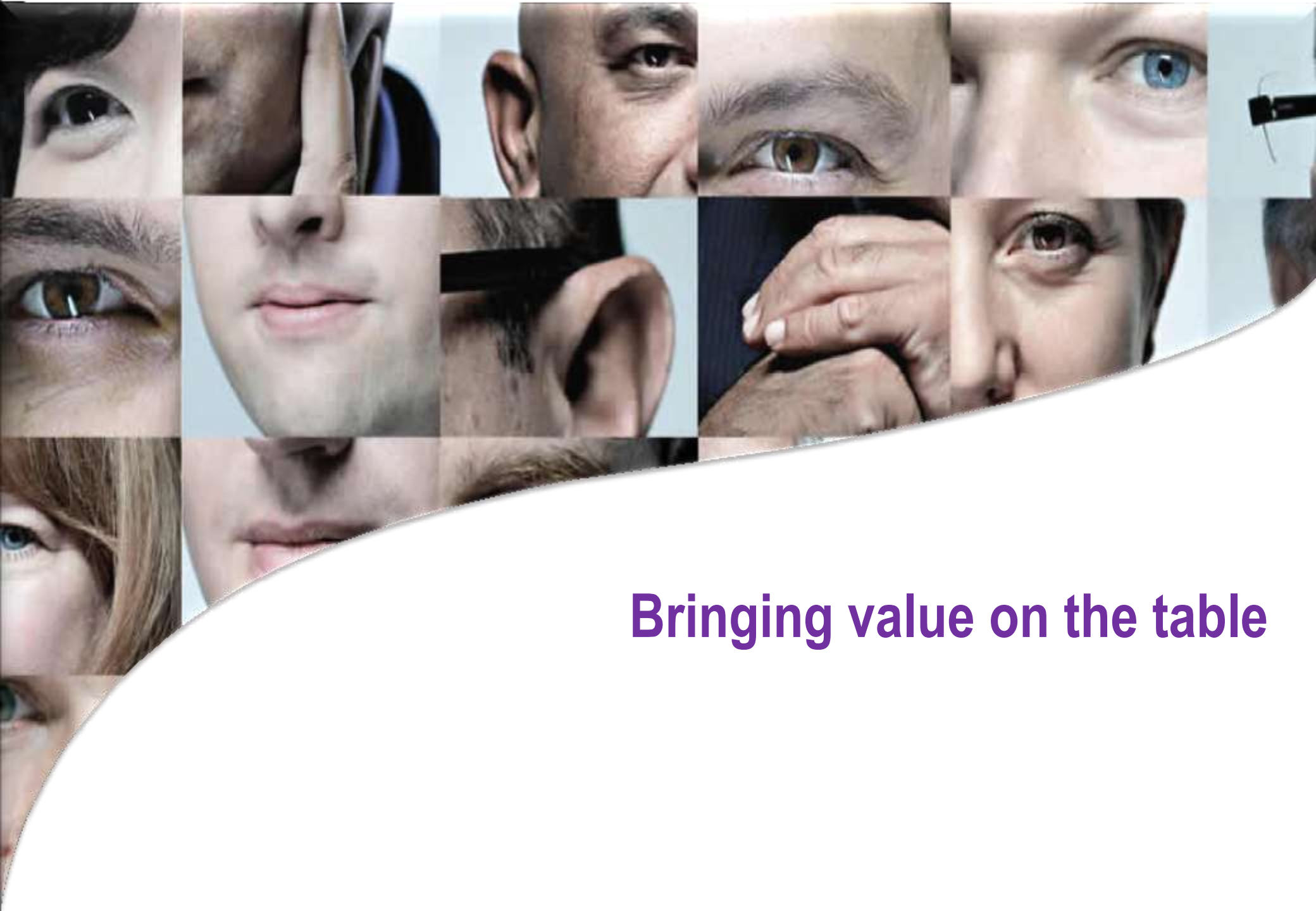
Product Owner Routine

Start of the day

- Look at the Burndown chart for this Sprint of your project.

During the Day

- Assist Team members to resolve issues
- Review the implementation of each Sprint Backlog Item as early as possible
- Write User Stories to address future requirements
- Write Epics to address future functional requirements too big to fit in a User Story
- Write Defects for any bugs you find
- Attend the Daily Scrum meeting if PO and the Team believe this is a productive thing to do.



Bringing value on the table



Understanding and Conveying Value

- Product Owner is responsible for the profitability of the product (ROI)
- The Product Owner is the owner of Product Backlog
- Every work item must be present in the Product Backlog in the form of a User Story, Spike or defect.
- Each User Story in the Product Backlog should have a corresponding business value assigned.
- Typically assign (L,M,H) Low, Medium, High
- PO prioritizes Backlog items by highest value



Understanding and Conveying Value

Creating Requirements for Scrum

- Scrum does not define just what form requirements are to take, but simply says that they are gathered into the Product Backlog referred to generically as “Product Backlog Items,” or “PBIs”
- Most Scrum projects borrow the “XP” (Extreme Programming) practice of describing a feature request as a “User Story,” some use the older concept of a “Use Case.”
- A User Story describes a desired feature (functional requirement) in narrative form. E.g.

Name: As a User I want to enter new contact into address book, so that I can contact the person later by postal or electronic mail

Description: User enters standard contact information (first and last name, two street address lines, city, state, zip / postal code, country, etc.) into contact-entry screen. He clicks “Save” to keep the data, and “Cancel” to discard data and return to previous screen.

How to test: Tester enters and saves the data, finds the name in the address book, and clicks on it. He sees a read-only view of the contact-entry screen, with all data previously entered.



User Story, Spike and Defect

Elements of a user story in Previous example are

- Name / Title
- Description
- How to Test / Acceptance Criteria

Additionally you can add

- Story Points / Priority / Risk / Tags
- Wireframes / Screenshots

Spikes / Technical User Stories

- A spike is a short throwaway research / POC code related to a story (discernible in PB)
- Written by Team members, added to Product Backlog after informing PO.

Defect

- A Defect, or bug report, is a description of a failure of the product to behave in the expected fashion.
- Defects are stored in a bug-tracking system, which may or may not be physically the same system used to store the Product Backlog. (Recommended to add to PB).



The Team Member



Team Member

- Team is cross-functional and consists of 3-9 people
- There are no set project roles within the team
- Team defines tasks and assignments
- Team is self-organizing and self-managing
- Maintains the Sprint Backlog
- Conducts the Sprint Review



Team Member Routine

Start of the Day

- Look at the Burndown chart for this Sprint
- Review the tasks you'll be working on today.

During the Day

- When you start or finish a task, update its status immediately
- Work on your tasks. If you need help, get it right away! Don't wait.
- Collaborate! Talk to anyone you need to talk to in order to get the work done.
- Attend the Daily Scrum meeting.
- Assist Product Owner with requirements development
- Write Technical Stories to address future, non-user-facing requirements
- Write Defects for any bugs you find
- Demonstrate each Sprint Backlog Item to the Product Owner as early as possible.

End of the Day

- Update the status of any tasks that need to be updated.
- Look at the Burndown chart to see how the Team is doing on this Sprint.



Ownership and Commitment



Ownership and Commitment

- Scrum encourages and expects Self Organization among teams.
- Scrum advocates collective ownership (We not I)
- We win together, we fail together.
- We commit to sprint goal and the goal is for the whole team.
- Completing a user story is just a step towards achieving sprint goal.
- Transparency enhances Commitment and Ownership



Cross-functional, Self-organizing Team

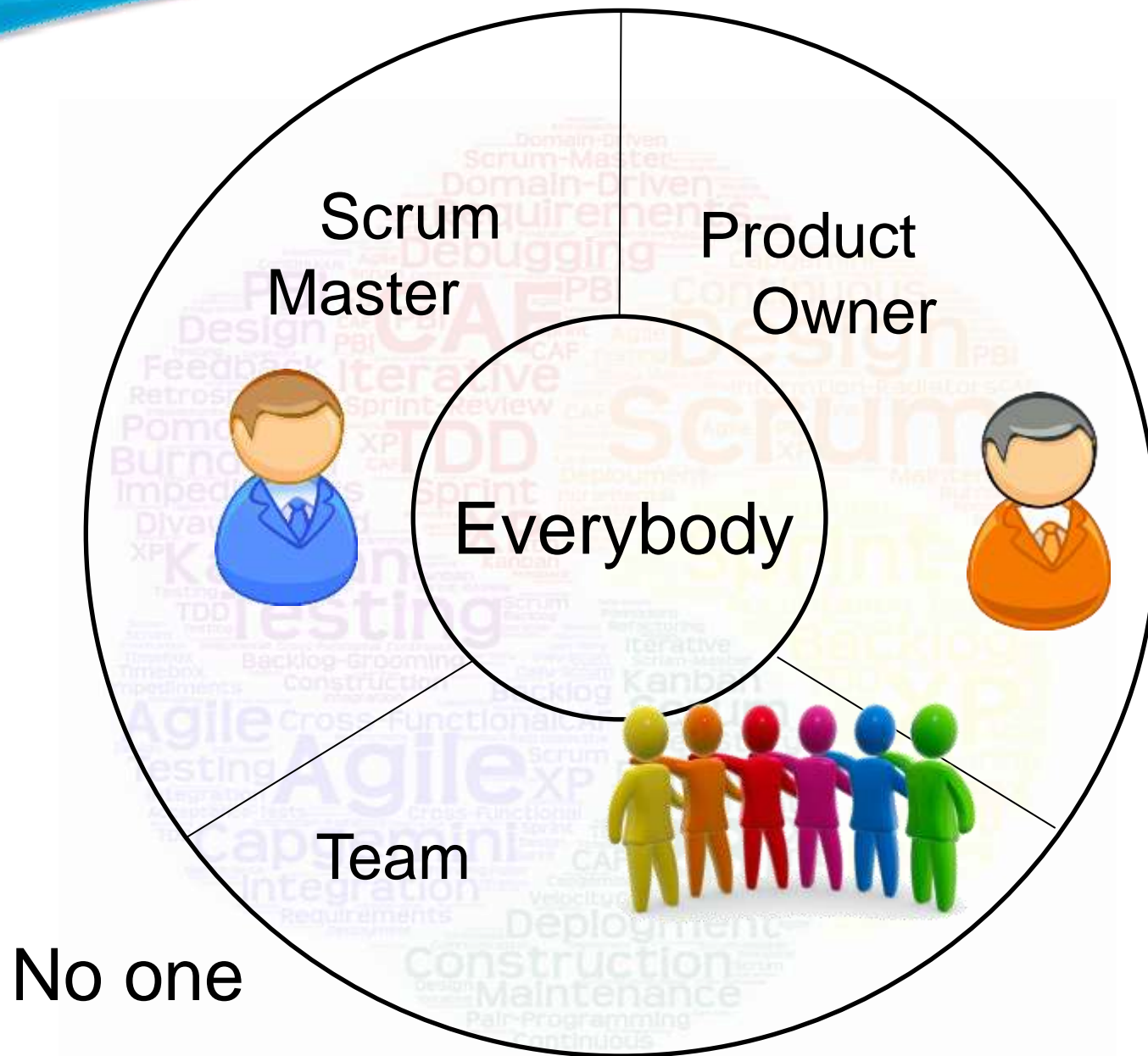


Cross-functional, Self-organizing Team

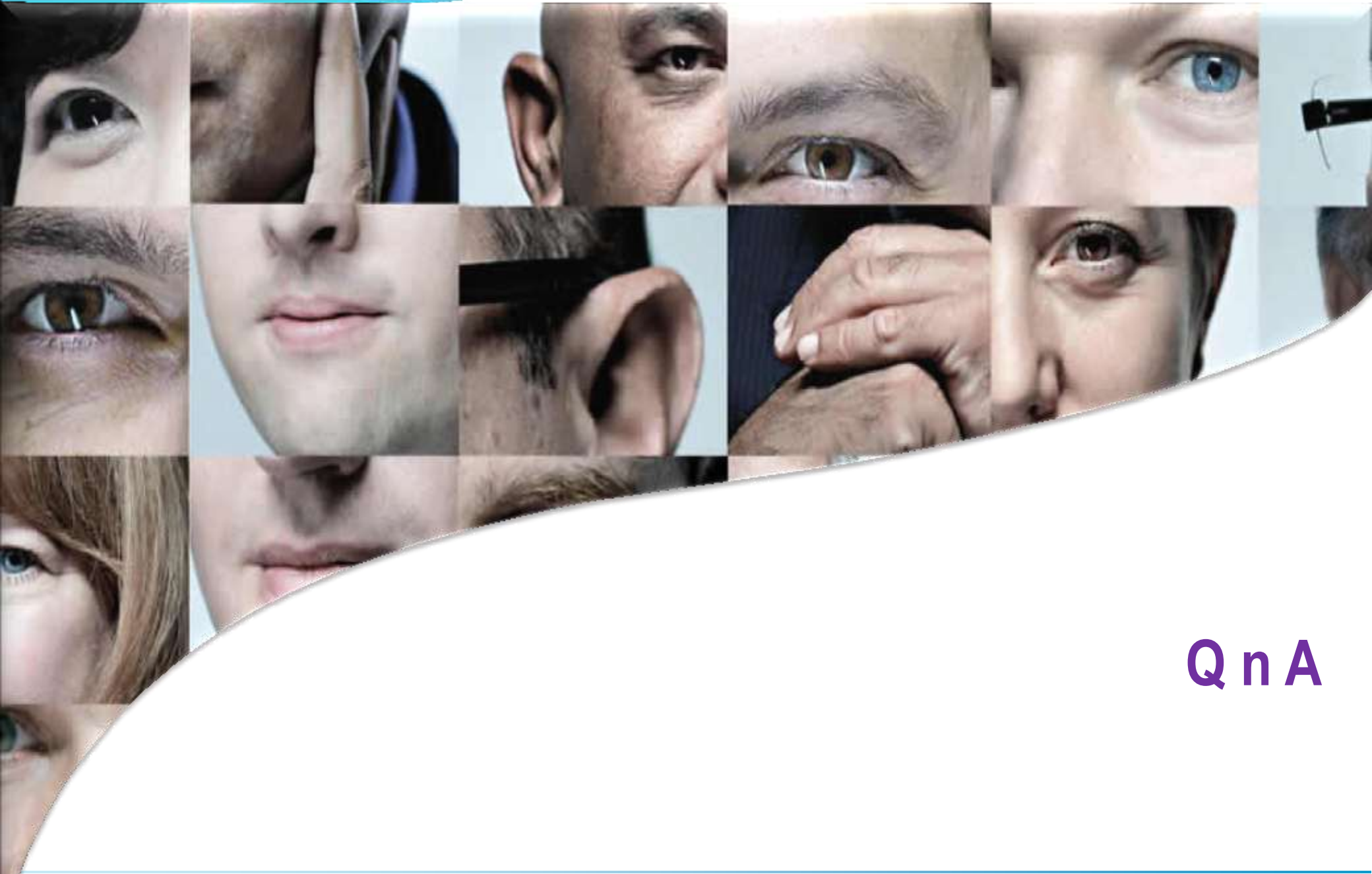
- Scrum Teams are self-organizing and cross-functional.
- Self-organizing teams choose how best to accomplish their work, rather than being directed by others outside the team.
- Cross-functional teams have all competencies needed to accomplish the work without depending on others not part of the team.
- Since the Team is responsible for producing the product, it must also have the authority to make decisions about how to perform the work.
- Team members decide how to break work into tasks, and how to allocate tasks to individuals, throughout the Sprint.
- The team model in Scrum is designed to optimize flexibility, creativity & productivity.



The Herculean Donut Poker Style



No one



Q n A

Thank you!

