



Agile Development with SCRUM

Thilina Halloluwa (PhD)

Based on, Rubin, K. S. (2012). Essential Scrum: A practical guide to the most popular Agile process. Addison-Wesley.

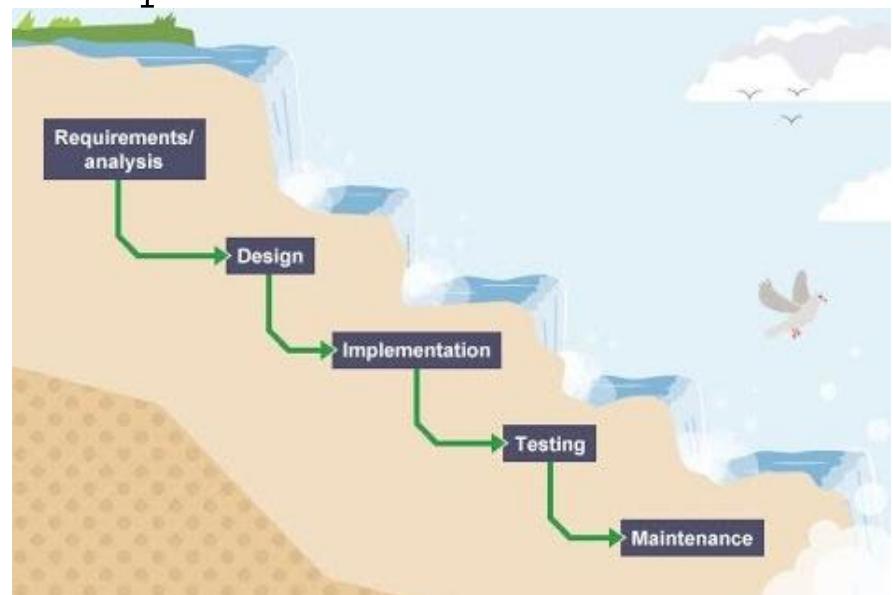
Introduction

- Are you happy with the results of our software development efforts?
- Do you think we deliver good customer value in a timely, economical, and quality manner?
 - Project failure rate is unacceptably high
 - Deliverables are late
 - Return on investment frequently falls short of expectations
 - Software quality is poor
 - Productivity is embarrassing
 - No one is accountable for outcomes

There must be a better way!!!

The Problem

- Waterfall life cycle
 - rigid process
 - preparing documents
 - Do not start any phase before the previous one was finished
 - Rarely go back to previous phases



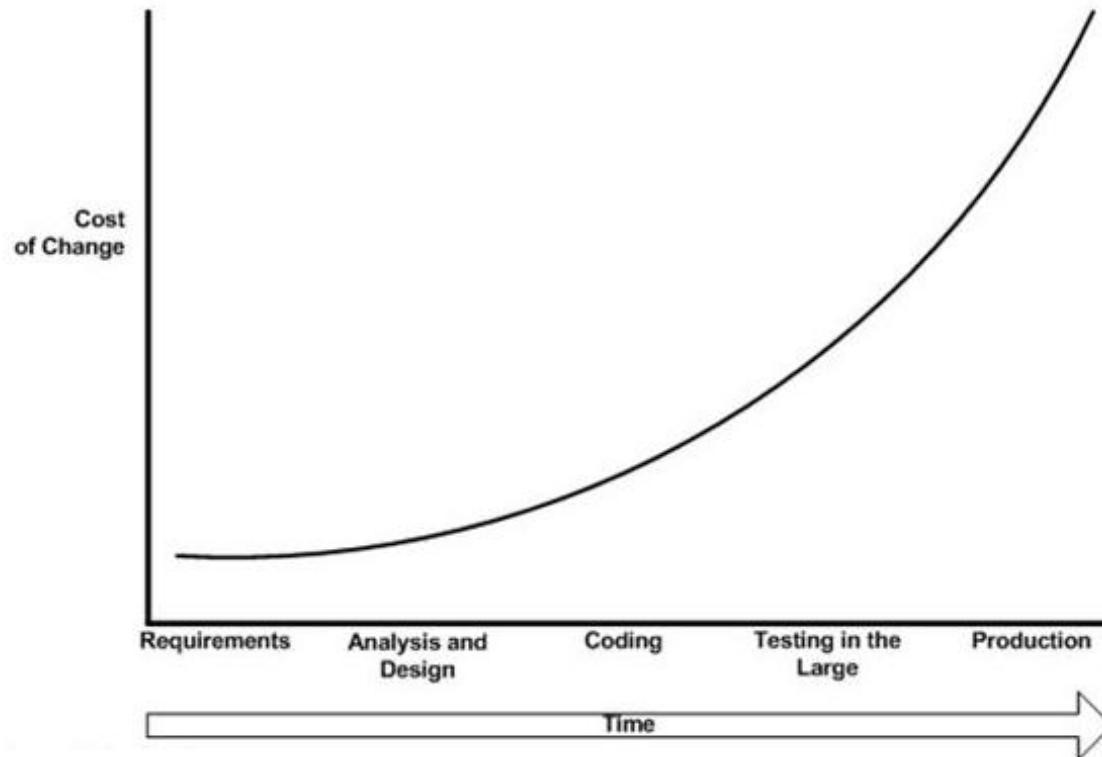
Worrying Statistics

- ❖ An average of 70% software development projects fail!!
- ❖ Reasons for FAILURE?
 - ❖ Not meeting the timelines
 - ❖ Costs overrun
 - ❖ Customers have NOT got what they asked ^



Cost of Change : Past

- Cost of change grows exponentially with time.





Change?

- ❖ Changes in Requirements
- ❖ Changes in Design, Implementation
- ❖ Changes in Technology
- ❖ Changes in Team
- ❖ Changes in users/client contacts

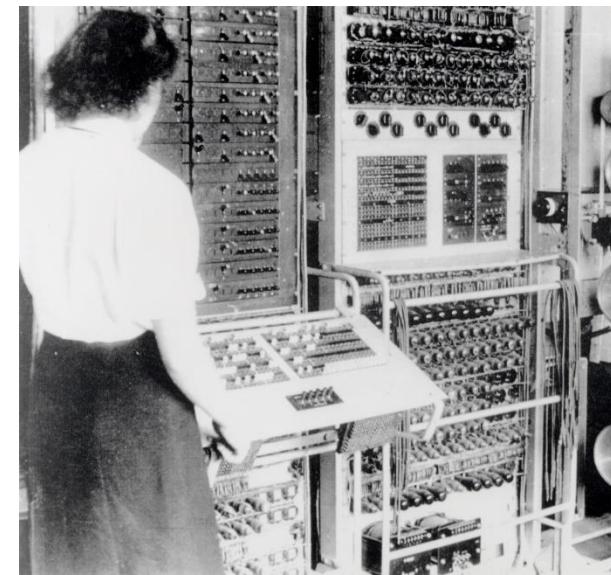
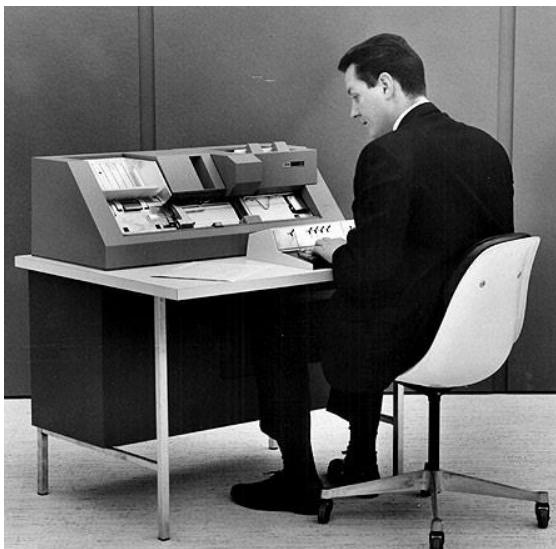
What should we do??

- Discover errors early – upfront planning



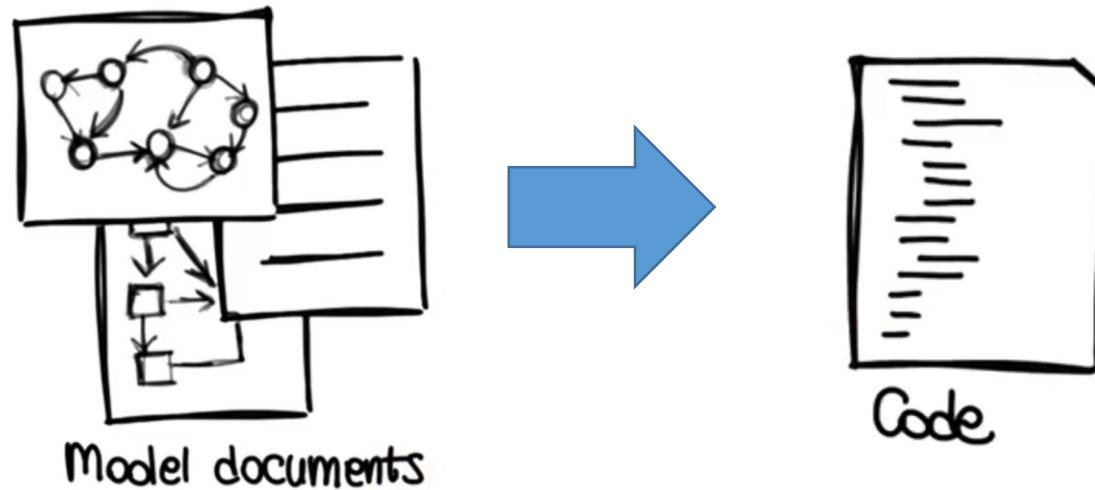
What should we do??

- Discover errors early – upfront planning
- Back in the day..
 - Punch cards
 - Cost on infrastructure
 - Slow and expensive computers
 - Developers did work manually



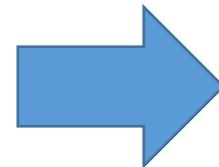
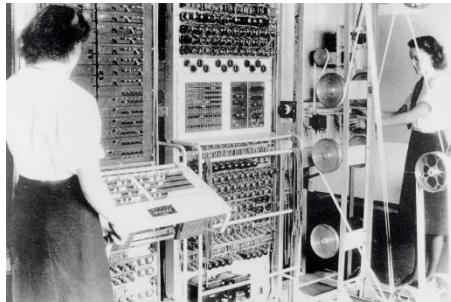
What should we do??

- Discover errors early – upfront planning
- Models (documents, drawings, paper prototypes) are cheaper to modify than code
 - willing to make large investments in upfront analysis and design models
- Only after this, we move to implementation



Present

- Cost on infrastructure
 - Cloud Computing
- Slow and expensive computers
 - Now even a mobile phone can do all the computations
- Developers did work manually
 - Now automation



cost of implementation is much less than it used to be

Cost of Change : Present

- There are a few interesting consequences
 - Upfront work becomes a liability
 - If there are ambiguities, better to delay

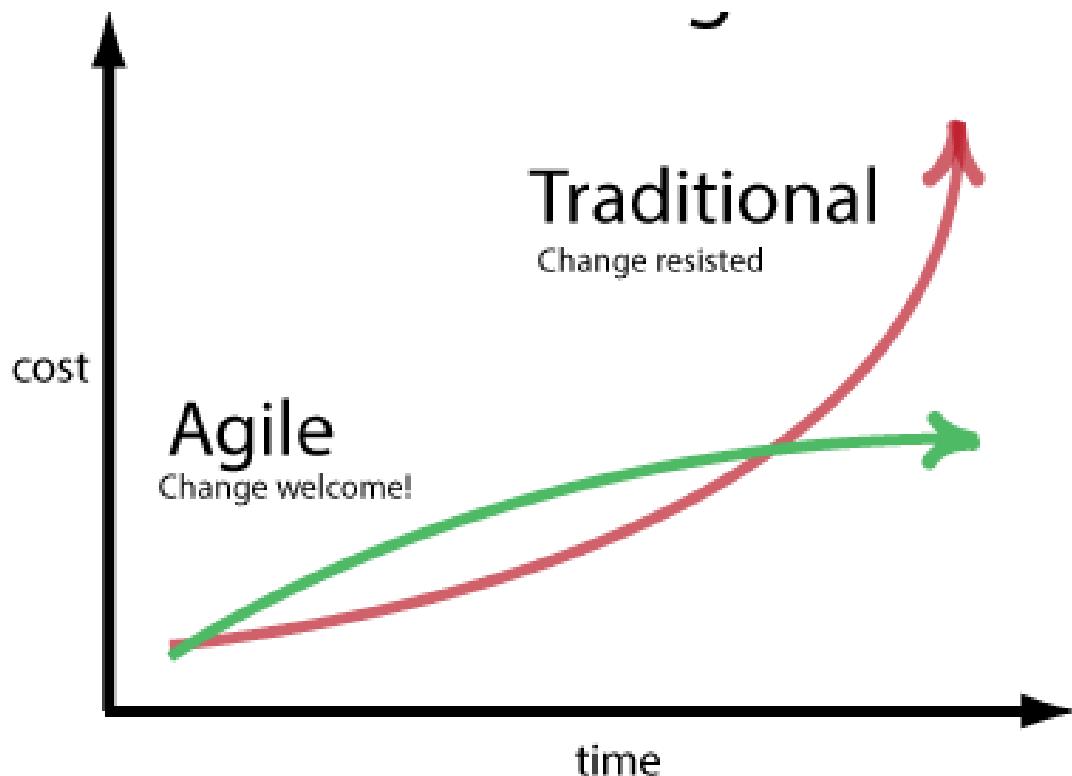
Birth of Agile

- February 2001
- 17 software developers
- Published the manifesto for agile software development



Cost of Change : Present

- Agile helps you to better deal with changes



As long as we use tools, practices and principles in the right way.



Agile Methods

- Aim at flat cost and a decrease in traditional overhead
 - to deliver working software quickly
 - to evolve it quickly based on **feedback**.
 - Expectation of requirements will change

Some Statistics



- Agile projects are 28% more successful than traditional projects.
- Almost 86% of 101,5 international surveyed software developers use Agile
- Almost 35% of projects aren't baselined at the planning stage.
- 27.4% of manufacturing companies rely solely on Agile.
 - 56.6% rely on the combination of methodologies.

(2018)

Agile Methodologies

- ❖ SCRUM – Jeff Sutherland, Ken Schwaber
- ❖ Xbreed - Mike Beedle
- ❖ DSDM - Arie van Bennekum
- ❖ XP – Kent Beck, Ward Cunningham, Ron Jeffries , Robert C. Martin
- ❖ Martin Fowler (Thoughtworks)
- ❖ FDD – Peter Coad, Jon Kern
- ❖ Testing - Brian Marick
- ❖ Adaptive Software development – Jim Highsmith
- ❖ Crystal Family – Alistair Cockburn
- ❖ Pragmatic programmers - Andrew Hunt , Dave Thomas



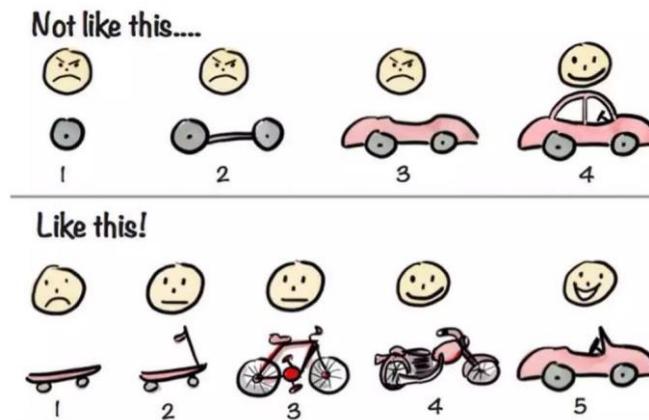
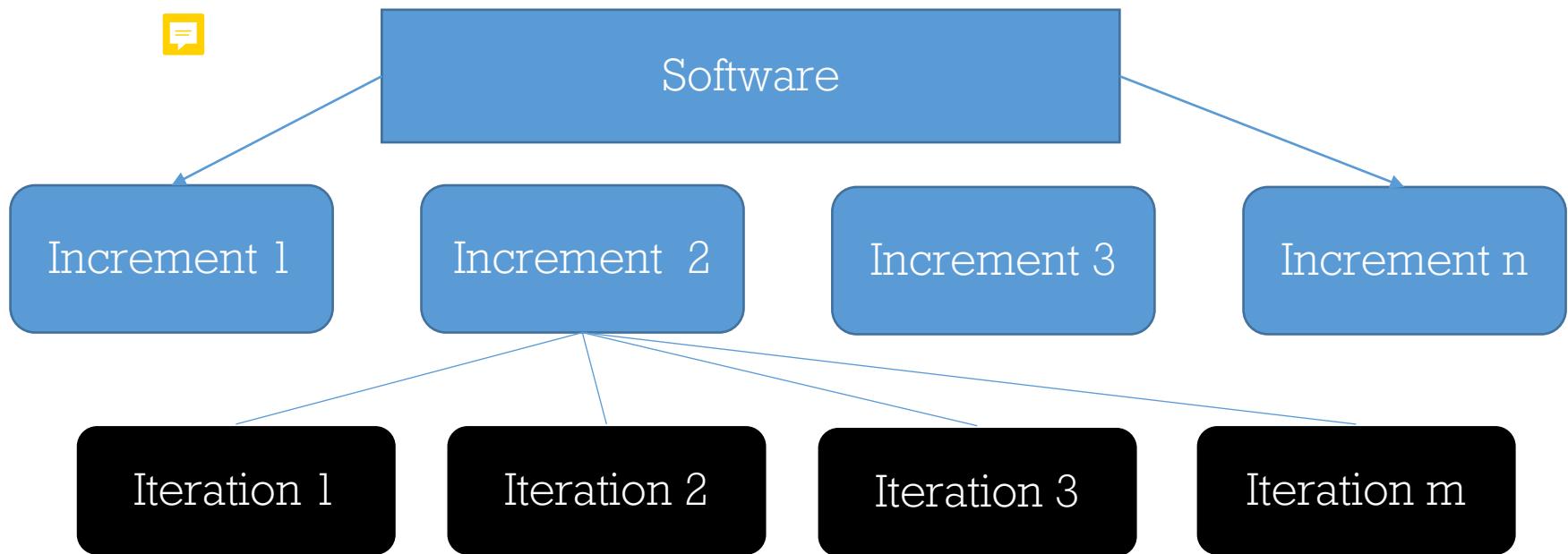
SCRUM

Scrum - Introduction

- What is Scrum?
 - An Agile methodology
 - It is an iterative incremental process for software development
 - An approach that controls the chaos of changing requirements 
 - A team-based approach to development
 - The whole team travels the distance as a unit (Rugby)

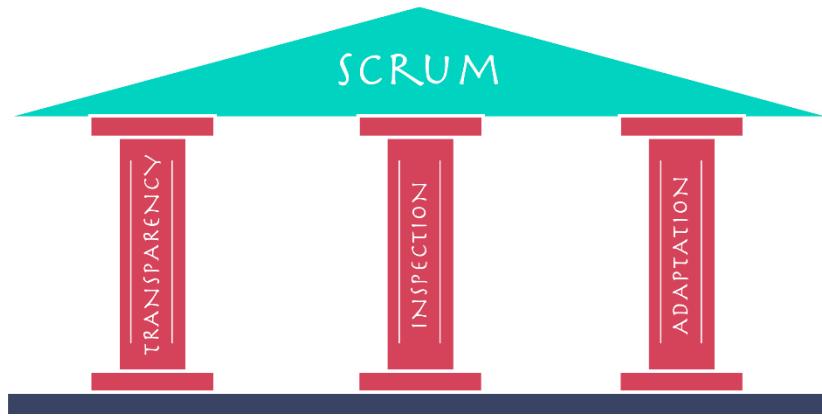


Iterative & Incremental



SCRUM Pillars

- SCRUM is based on 3 pillars (TIA)
- Any aspect of SCRUM has to be inspected
- You can only inspect if its visible.
- All aspects has to be transparent to be visible
- You can only adapt something if you can inspect it



These 3 pillars support each other

Definition of Done

- One way of SCRUM maintain transparency is to explicitly define “DONE”
- This is understood by the SCRUM team and the external stakeholders at an early stage



Epics

EPIC	USER STORY	ACCEPTANCE CRITERIA
As an Acquisition Gateway User , I need to access the Acquisition ordering platform behind a secure login so that I can purchase products.	As an Acquisition Gateway User , I need to select an Auction product in the Acquisition ordering platform so that I can bid on it.	<p>Ensure the Acquisition Gateway User is able to:</p> <ul style="list-style-type: none">• log in to Acquisition Gateway• navigate to the Auction page• able to select a product(s) to bid on
	As an Acquisition Gateway User , I need to review my previous bids in the Acquisition ordering platform so that I can remove expired bids.	<p>Ensure the Acquisition Gateway User is able to:</p> <ul style="list-style-type: none">• log in to Acquisition Gateway• navigate to a page to review items previously bid upon• select one, or multiple, expired bids• remove expired bids

User Story

Story Title

User Story 1

As a(stakeholder)
I want to (task),
So That (desired result)

Acceptance Criteria

Measurable results, what defines "done"?
And I know I am done when

User ID

Importance

Estimate

Type

- Payment
- Report/view
- Search
- Manage data
- Workflow



EXAMPLE

US: Forgot password

As an unauthorized user,
I want to have an opportunity to
recover my password using my email,
so that I can have a new password for
my account.

Story ID:	Story Title:
User Story:	<p>Importance:</p> <p>Estimate:</p> <p>Type:</p> <ul style="list-style-type: none"> <input type="checkbox"/> Search <input type="checkbox"/> Workflow <input type="checkbox"/> Manage Data <input type="checkbox"/> Payment <input type="checkbox"/> Report/View
<p>As a: <role></p> <p>I want: <some goal></p> <p>So that: <some reason></p>	<p>Priority:</p> <p>Size:</p> <p>As a:</p> <p>I want to:(what)</p> <p>So that: (why)</p>
Acceptance Criteria	<p>And I know I am done when:</p>



More on Sprint Planning



www.BostonAgileTraining.com

Sprint Planning 1: Sprint Backlog

USER STORY

As a

I want to

So that

I KNOW I AM DONE WHEN....

-
-
-

SIZE

--

TEAM

- Design
- Front-End
- Back-End

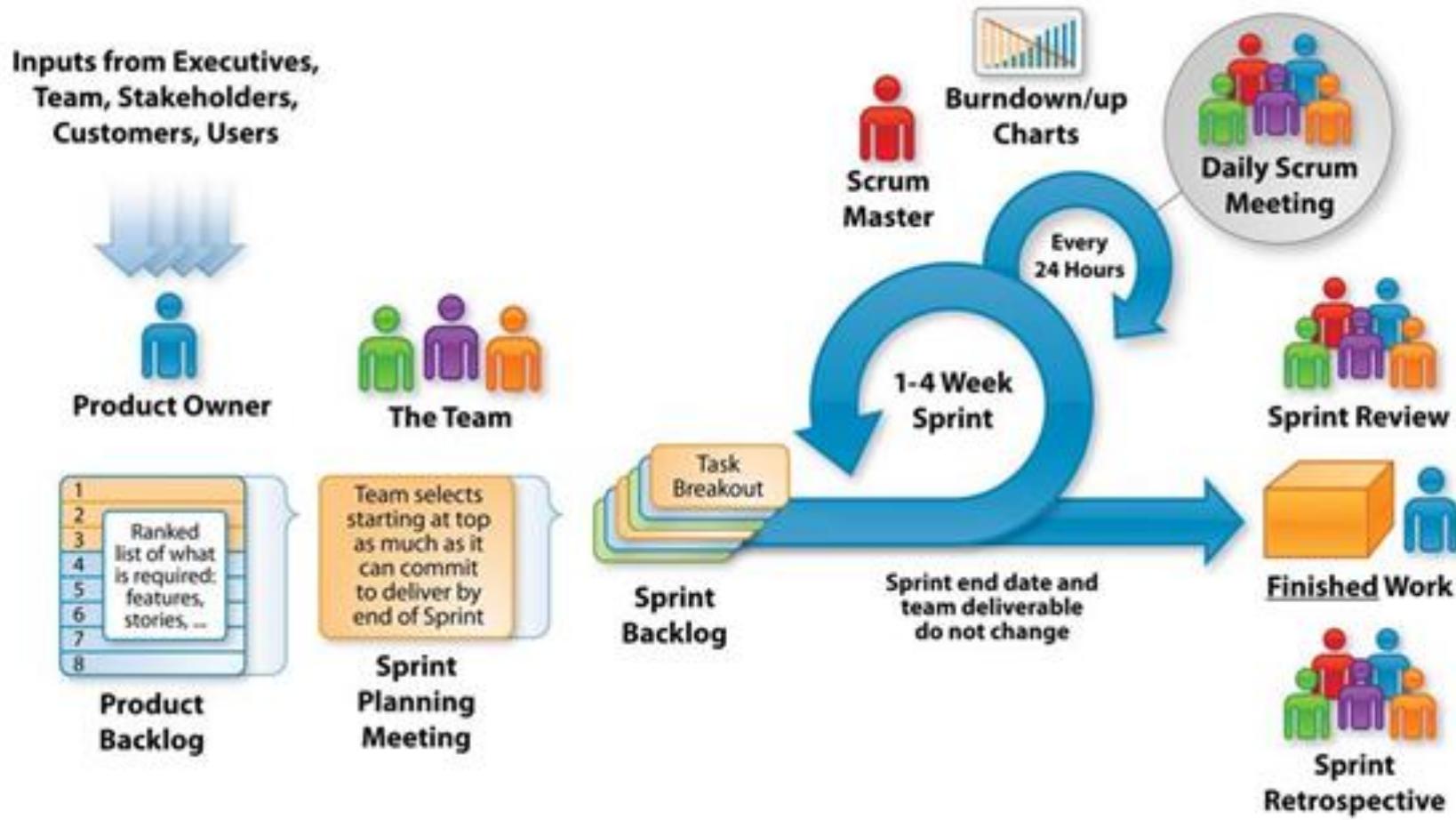
Initiatives

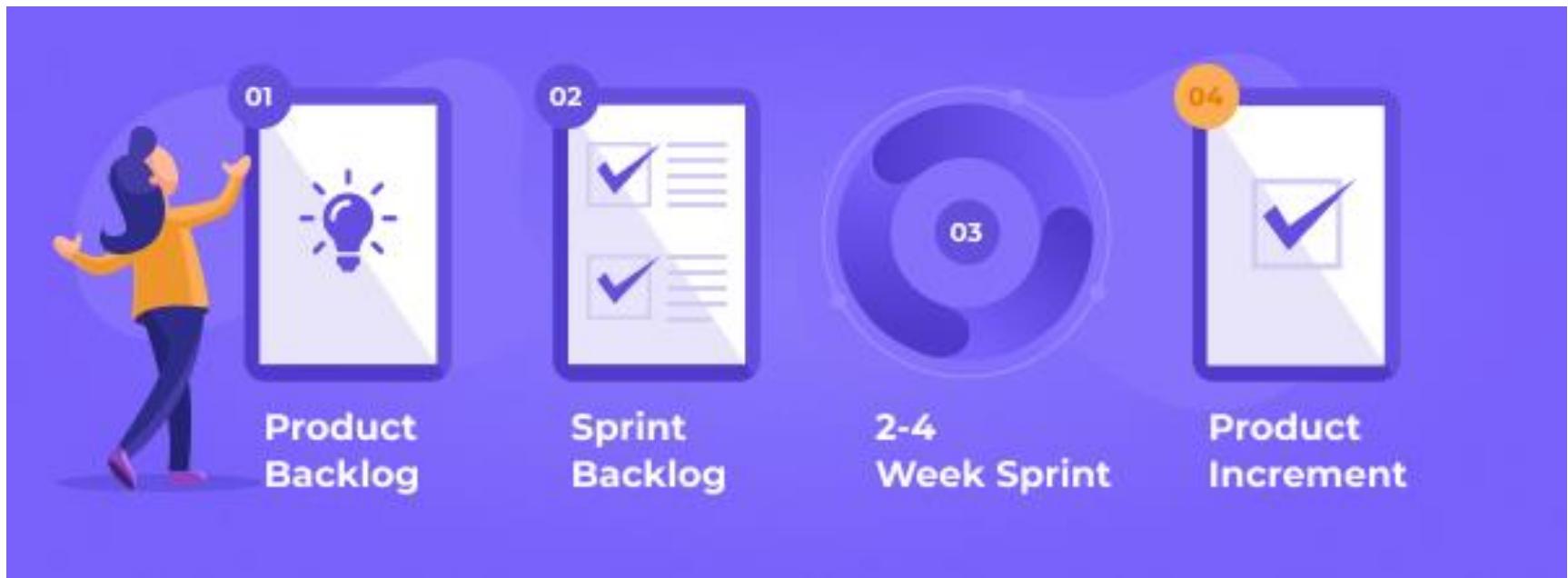


Initiatives vs Epics vs Stories



Scrum Framework





SCRUM RULES

- ❖ Cross-functional Teams
 - ❖ 7 – 9 members (excluding PO & SM)
- ❖ Time-boxes
- ❖ Feedback and learning through frequent delivery
- ❖ Predictability
- ❖ Collective and Adaptive planning to build the right product
- ❖ Productivity & Quality
 - ❖ Via Reduction of waste
 - ❖ Through team and time-boxed focus



SCRUM ROLES

SCRUM ROLES

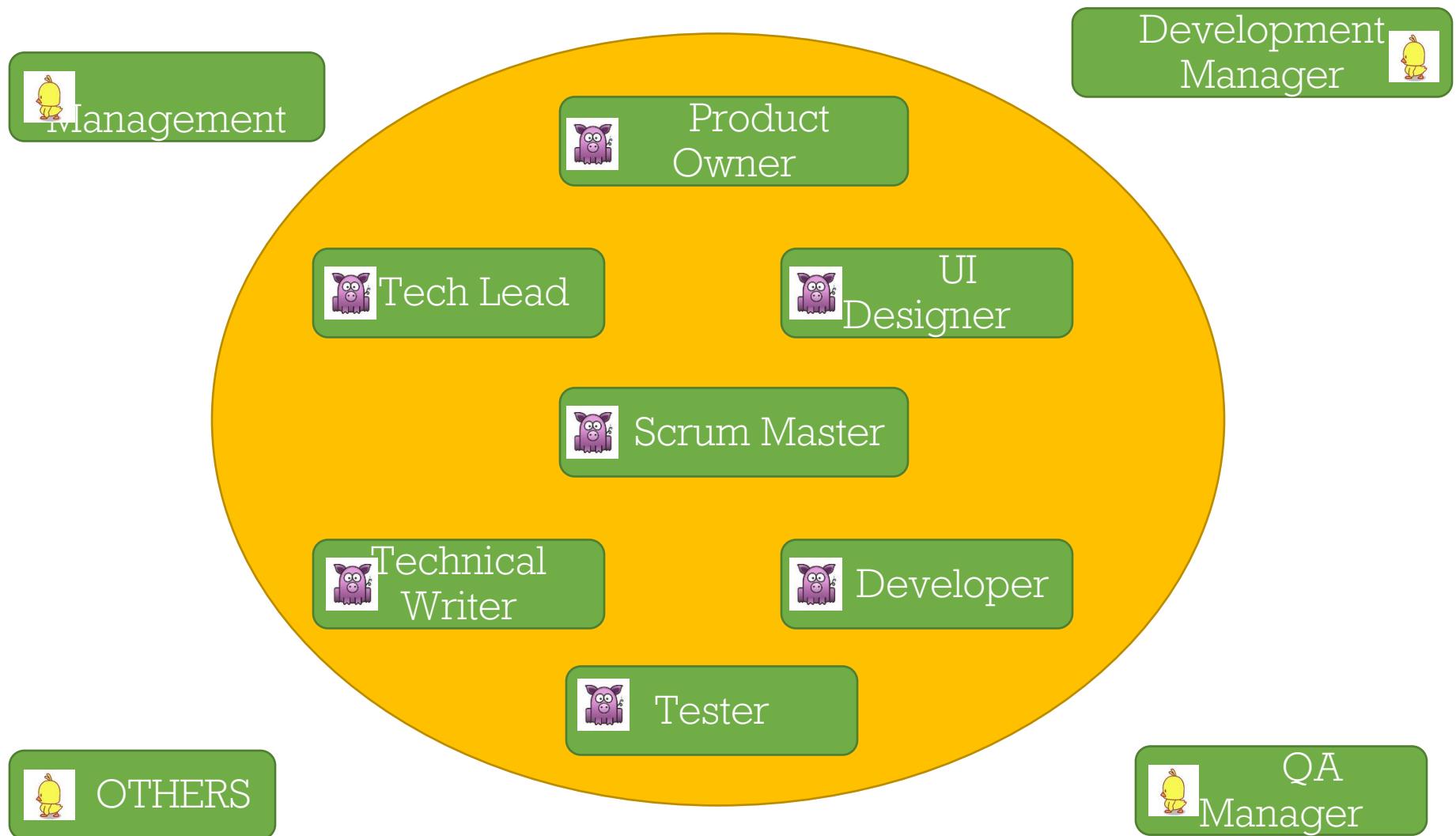


The Chicken and the Pig

- A Pig and a Chicken are walking down the road. The Chicken says,
- "*Hey Pig, I was thinking we should open a restaurant!*".
- Pig replies,
 - "*Hm, maybe, what would we call it?*".
- The Chicken responds, "*How about 'ham-n-eggs'?*".
- The Pig thinks for a moment and says,

No thanks. I'd be committed, but you'd only be involved!

Ideal Scrum Team





Product Owner

Product Owner

- ❖ Share the product vision/goals with the team
- ❖ Identify the requirements
- ❖ Prioritize the requirement
- ❖ Be the “GO TO” person for requirement clarifications
- ❖ Provide feedback
- ❖ Negotiate with stakeholders
- ❖ Has the most authority on what should be built

Product Owner

- The product owner must understand the needs and priorities of the stakeholders to act as their representative.
- Product owner acts as
 - product manager
 - business analyst
 - part tester

Product Owner



Product owner

Defines what will be built

User story 1
Requirement 1
Requirement 2
Enhancement 1

User story 2
Defect 1

Epic 1
Epic 2

Product backlog

Product Owner- Participate in Planning



- Portfolio planning
 - Position the product correctly in the portfolio backlog
 - Determine when to start and end product development
- Product planning
 - Works with the stakeholders to envision the product
- Release planning
 - Define the content of the next release
- Sprint planning
 - Define a sprint goal

Product Owner- Groom the Product Backlog

- Creating and refining
- Estimating
- Prioritizing

While he/she is not doing these personally , he /she is responsible for making sure that the grooming activities take place

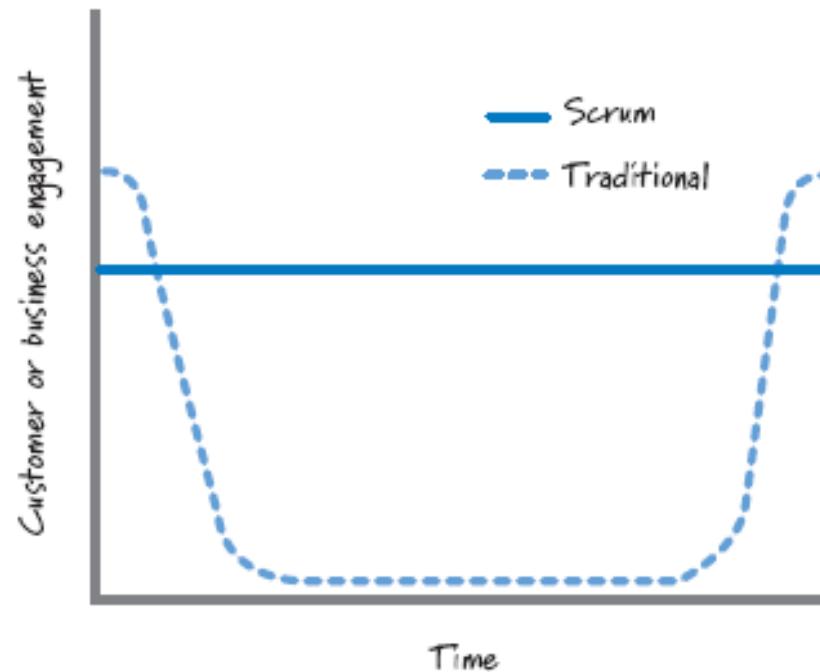
Product Owner- Define Acceptance Criteria and Verify That They Are Met

- Defining the acceptance criteria for each product backlog item.
- The product owner may also write acceptance tests corresponding to the acceptance criteria

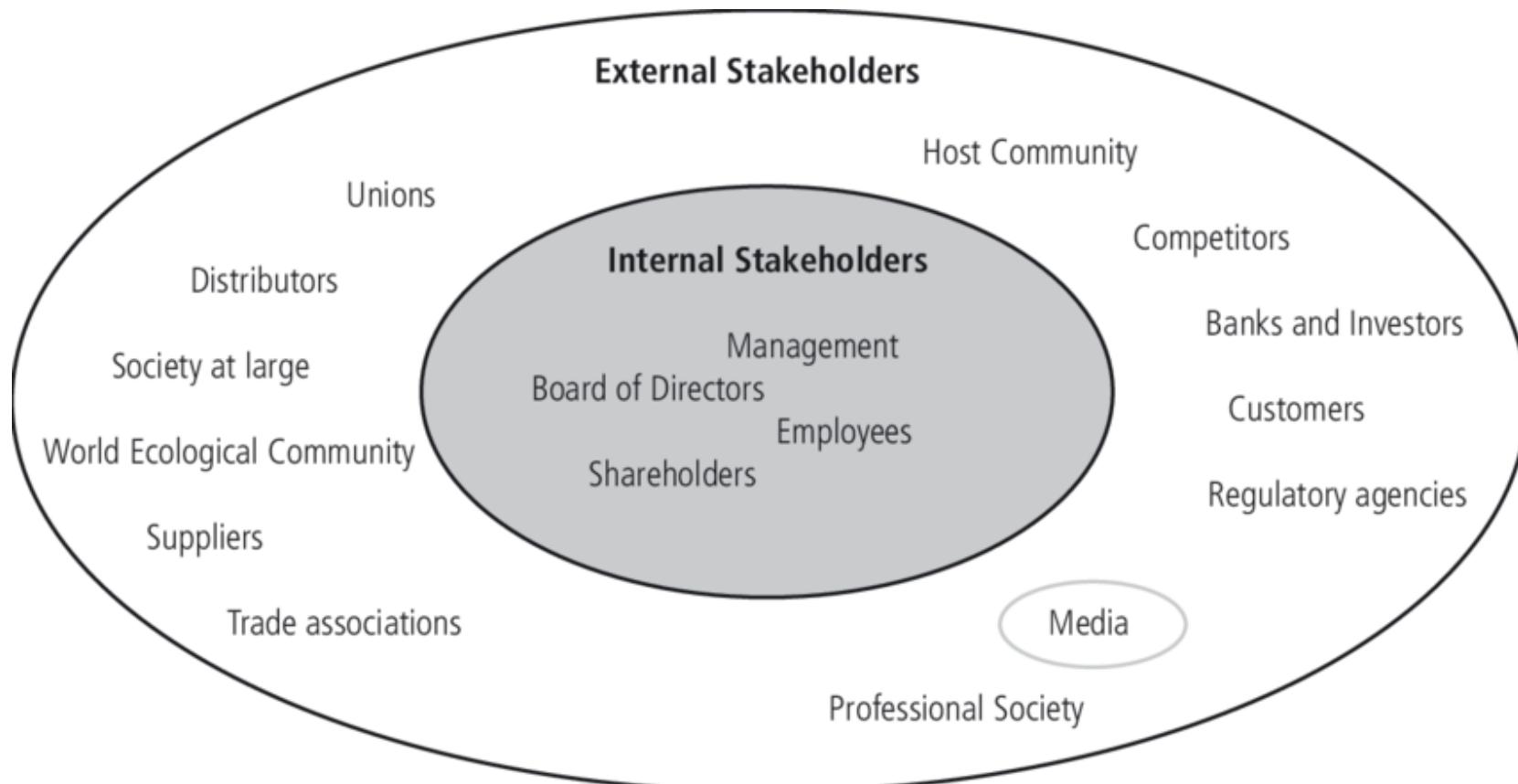
The product owner must ensure that acceptance tests are run prior to the review

Product Owner- Collaborate with the Development Team

- Must closely collaborate with the development team on a frequent basis.
- An engaged, committed, everyday role



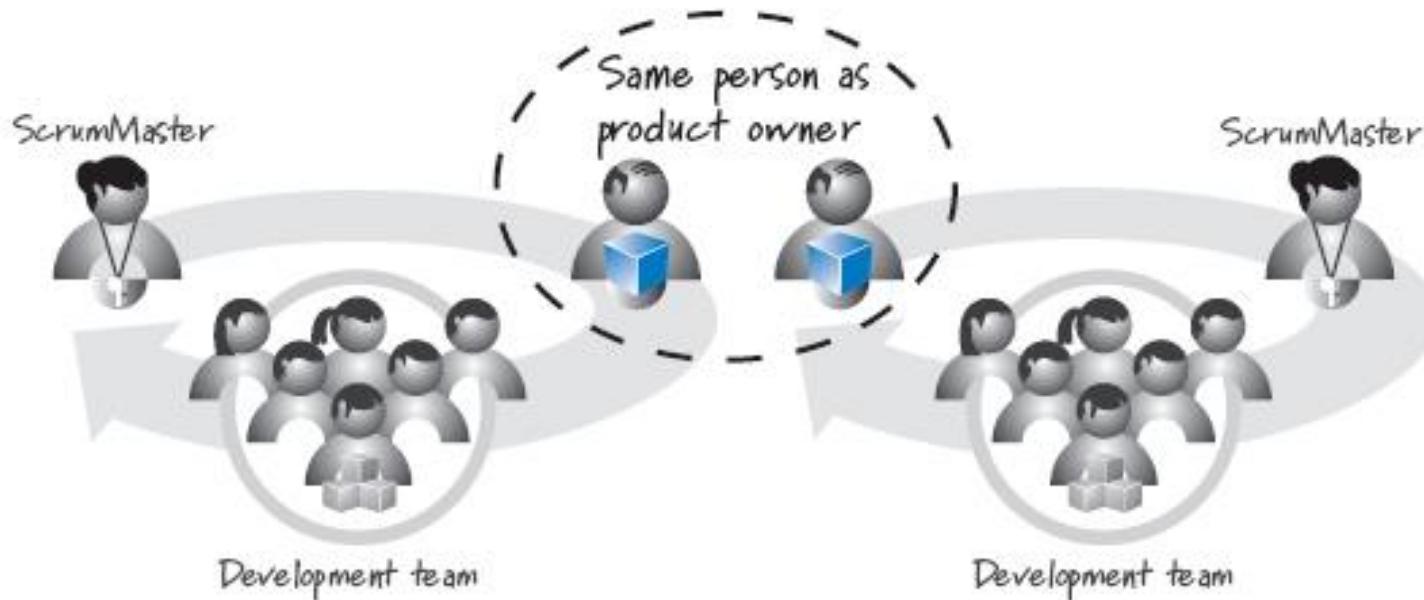
Product Owner- Collaborate with the Stakeholders



Must work closely with the entire stakeholder community

Product Owner–Multiple Roles

- product owner and a member of the development team
- product owner and the Scrum Master



Product Owner–Product Owner Team

- Every Scrum team **MUST** have a single person who is identified as the product owner
- Should we allow a team of people to perform the product owner role?
- Why a “product owner team” ?
 - Product Owner Proxy (HW)
 - Chief Product Owner (HW)
 - Product line owner (HW)
 - Feature product owner (HW)

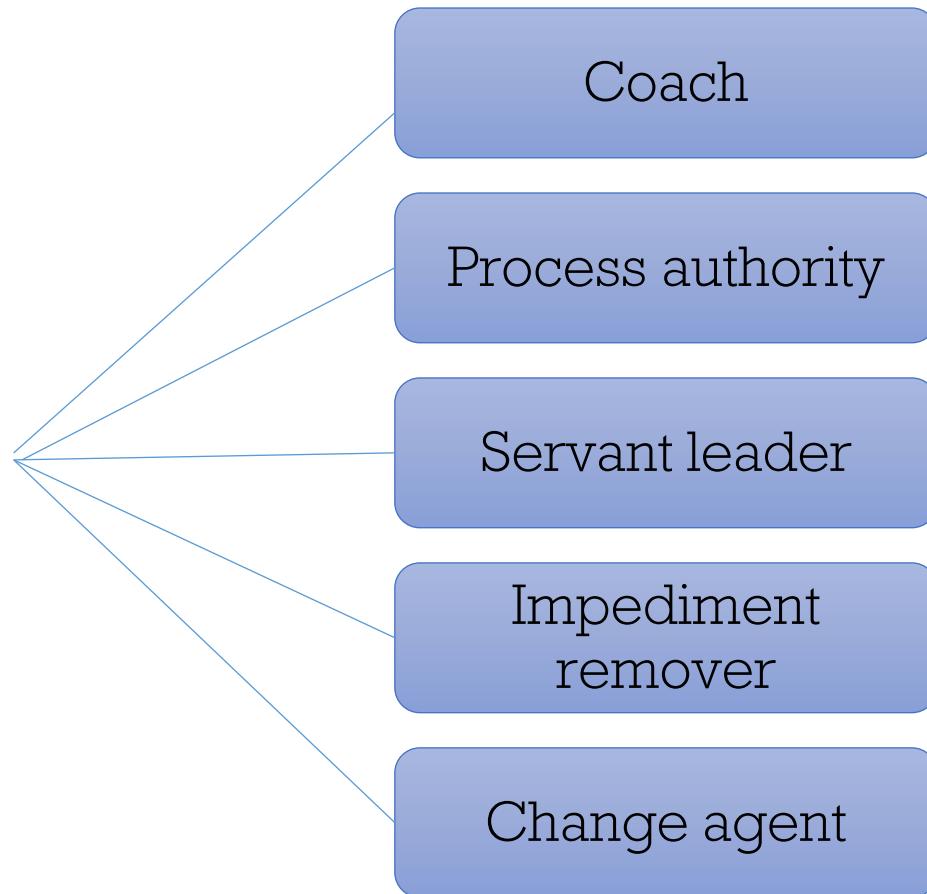


Scrum Master

SCRUM master

- ❖ Be a facilitator
 - ❖ focused on helping everyone understand and embrace the Scrum values, principles, and practices
- ❖ Remove obstacles faced by the team
- ❖ Assist the team in achieving the iteration goals
- ❖ Coach the team on SCRUM principles
 - ❖ Acts as a coach to both the development team and the product owner

Scrum Master- Responsibilities



Scrum Master- Coach

- ❖ Coaches both the development team and the product owner
 - ❖ Can remove barriers between the roles and enable the product owner to directly drive development
- ❖ Observes how the team is using Scrum and does anything possible to help it get to the next level of performance

“I’m not here to solve your problems for you; instead, I’m here to help you solve your own problems.”

Scrum Master -Servant Leader

❖ The Scrum Master is a servant to the Scrum team

Asks ,

- What can I do today to help you and the team be more effective?

Instead ,

- What are you going to do for me today ?

Scrum Master -Process Authority

- ❖ Ensure that the Scrum team adheres to the Scrum values, principles, etc.
- ❖ Not the same type of authority that a project manager would have ,
 - ❖ The Scrum Master doesn't hire and fire .
 - ❖ Cannot dictate to the team what tasks to do or how to do them.
 - ❖ Not responsible for making sure the work gets done

Scrum Master- Interference Shield

- ❖ Protects the development team from outside interference
 - ❖ remain focused on delivering business value every sprint
- ❖ Acts as an interceptor

Scrum Master- Impediment Remover

- ❖ Responsible for removing impediments / obstructions that inhibit the team's productivity.

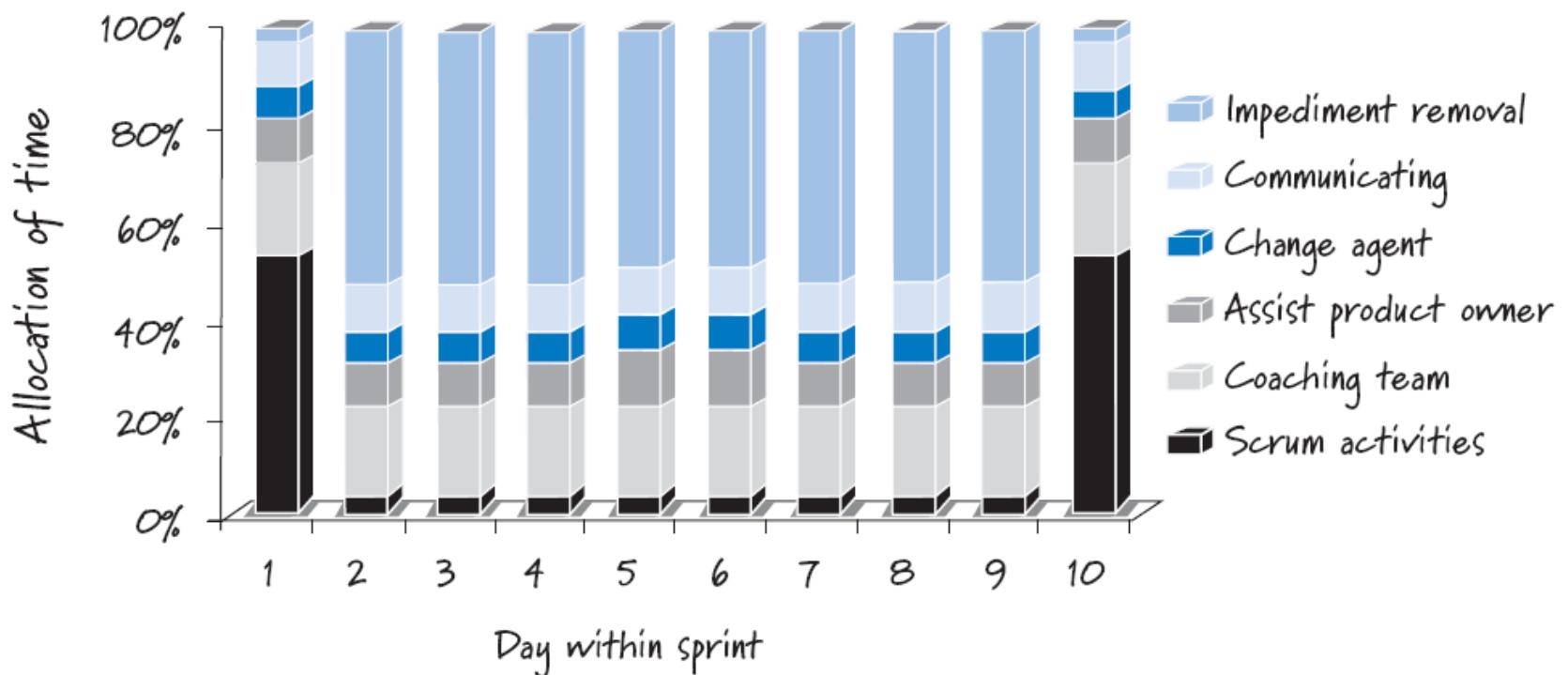
Eg: A Scrum team that was consistently fails to meet its sprint goals

impediment can be an unstable production server that the team using during testing.

Scrum Master - Change Agent

- ❖ The scrum master not only help change or fix technical issues but change minds too
- ❖ Helps others understand the need for change, the impacts of Scrum outside of the Scrum team, and the broad-reaching benefits Scrum can help achieve.

Scrum Master- Spending the day



Scrum Master - Characteristics/Skills

- ❖ Knowledgeable
- ❖ Questioning
- ❖ Patient
- ❖ Collaborative
- ❖ Protective
- ❖ Transparent

Scrum Master- Combined with Other Roles

- ❖ If talented enough , one can work both as development team member and Scrum master
 - ❖ could suffer from a conflict of interest
- ❖ Can be the scrum master for more than one team
- ❖ Same person serve as both scrum master and product owner is highly discouraged.



Development Team

Development Team

- ❖ Traditional software development approaches define various job types
 - ❖ Architect
 - ❖ Programmer
 - ❖ Tester
 - ❖ database administrator etc...
- ❖ Scrum defines the role of development team, which is a cross-functional collection of these types of people

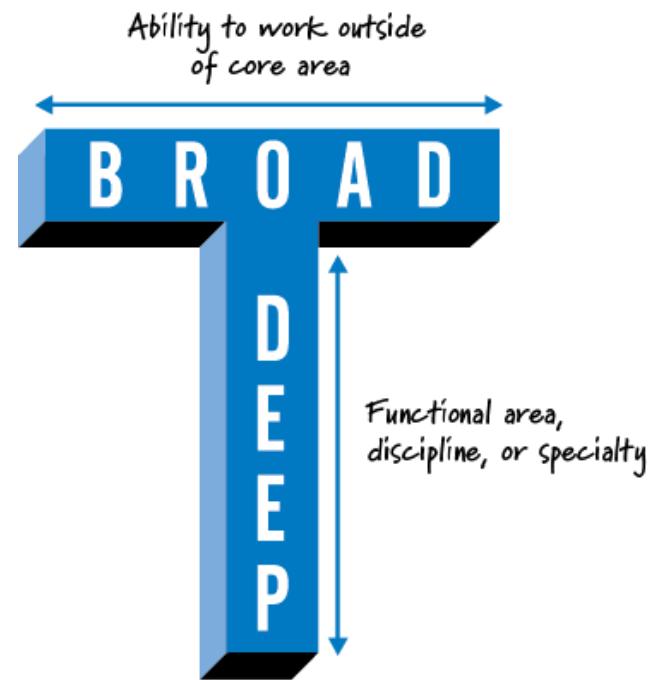
The development team must do all of the work to produce one or more vertical slices of working product functionality

Development Team: Responsibilities

- ❖ Perform Sprint Execution
- ❖ Inspect and Adapt Each Day
- ❖ Groom the Product Backlog
- ❖ Groom the Product Backlog
- ❖ Inspect and Adapt the Product and Process

Development Team-Characteristics

- ❖ Self-organizing
- ❖ T-shaped skills
- ❖ Cross-functionally diverse and sufficient
- ❖ Long-lived
- ❖ High-bandwidth communications
- ❖ Musketeer attitude
- ❖ Right-sized
- ❖ Transparent communication
- ❖ Works at sustainable pace
- ❖ Focused and committed





Agile Development with SCRUM – Part 2

Thilina Halloluwa (PhD)

Based on, Rubin, K. S. (2012). Essential Scrum: A practical guide to the most popular Agile process. Addison-Wesley.

Last Week

Why Agile

What is SCRUM

SCRUM pillars of success

SCRUM Framework

SCRUM Rules

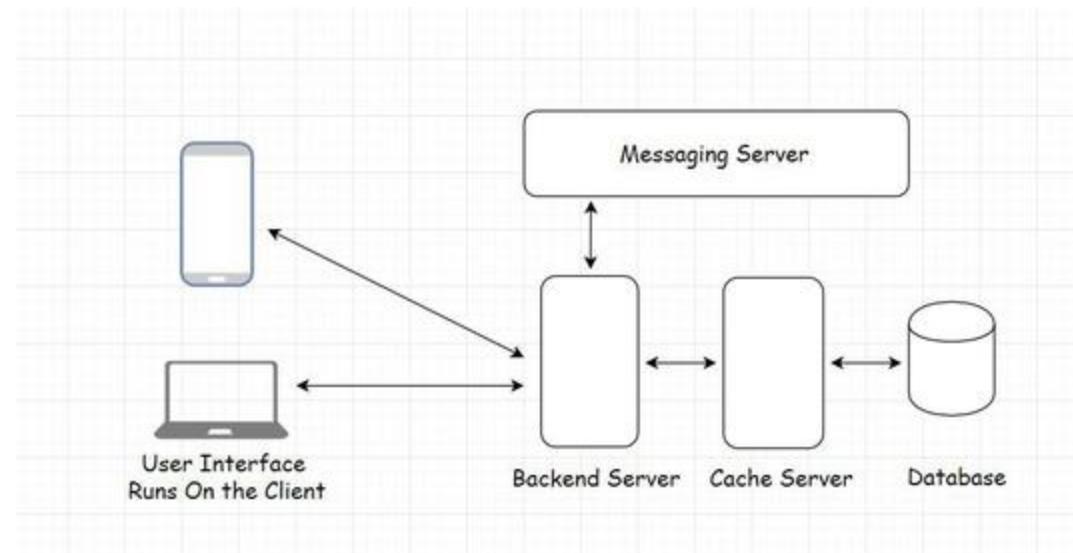
SCRUM Roles



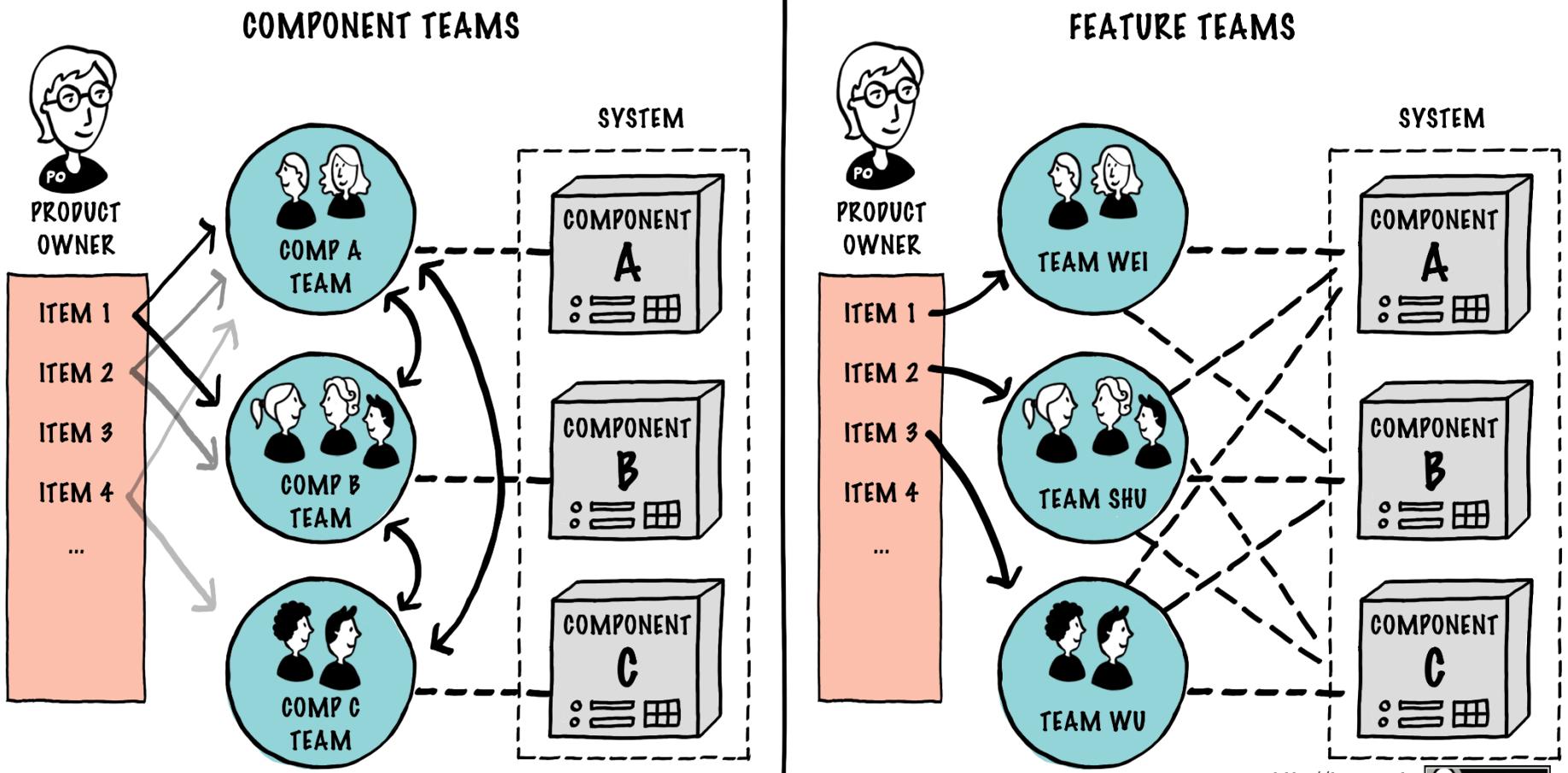
TEAM STRUCTURES

SCRUM Team Structures

- Follow two common ways to organize themselves
 - Feature Teams
 - Component Teams
- Feature Vs Component

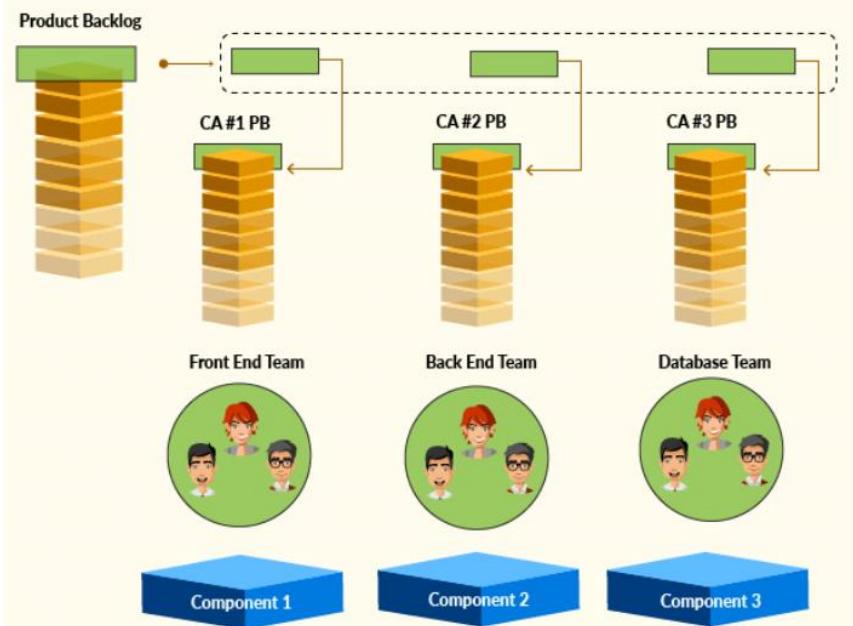


Feature Vs Component Teams



Scrum favors feature teams. Unfortunately, many organizations prefer component teams

Feature Vs Component Teams



Component Team



Feature Team

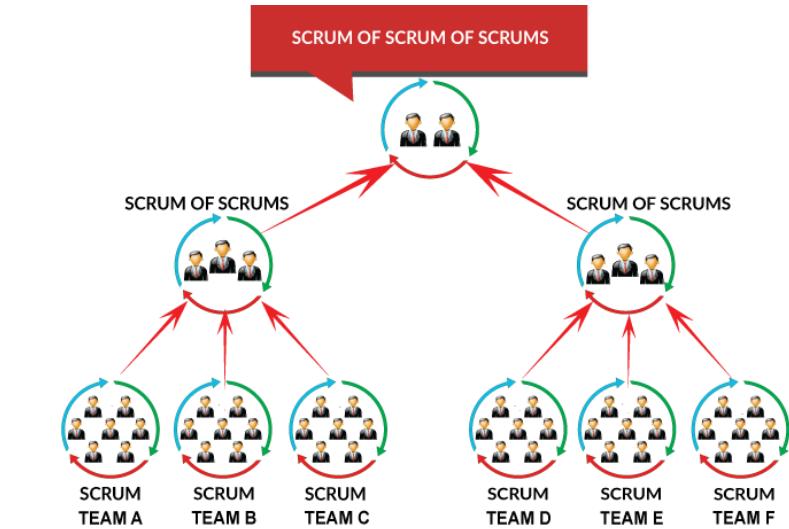
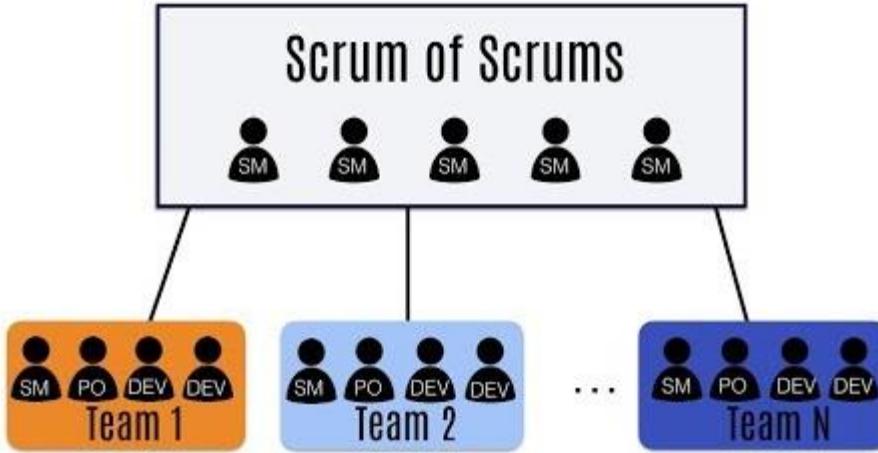
Multiple-Team Coordination

- ❖ Scrum scales by having multiple right-sized Scrum teams
- ❖ How to coordinate those teams ?
 - ❖ scrum of scrums
 - ❖ release train.



Team Coordination : Scrum of Scrums

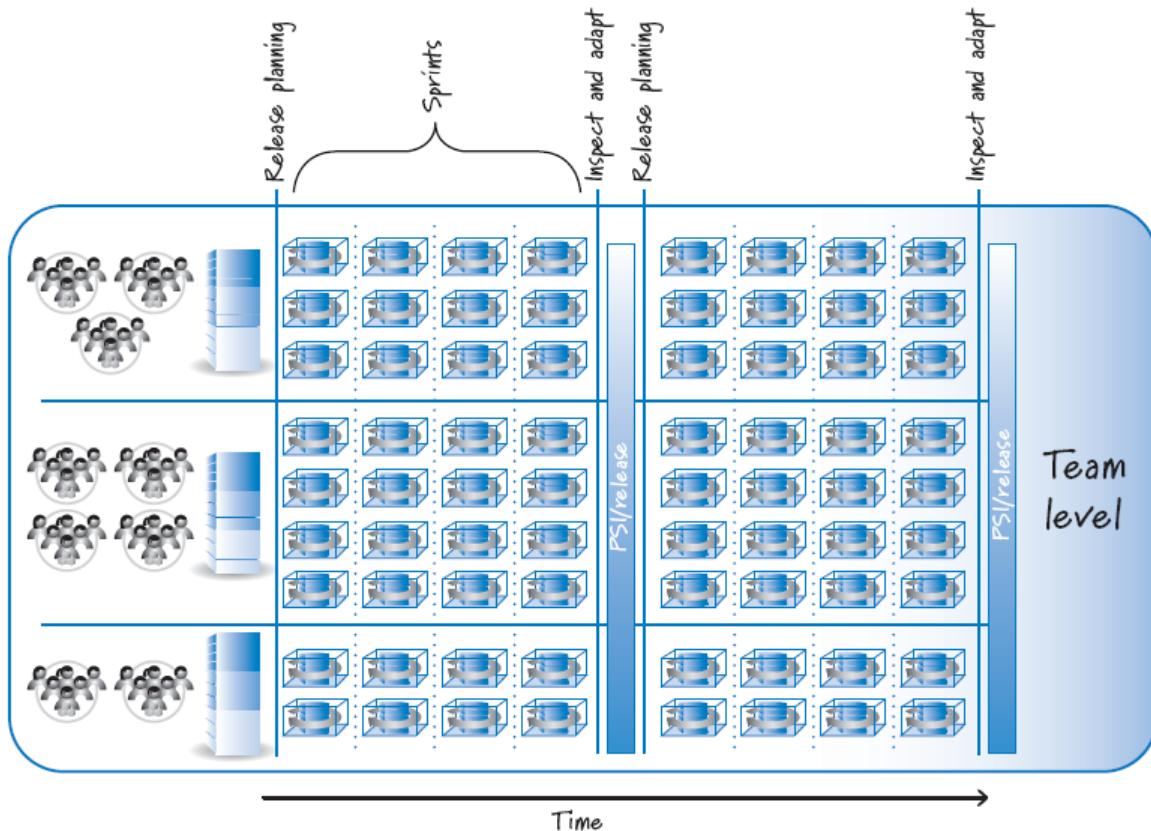
- ❖ Allows multiple teams to coordinate their inter-team work
- ❖ Individual members of the various development teams participate
 - ❖ development team determines based on who can best speak to the inter-team dependency issues



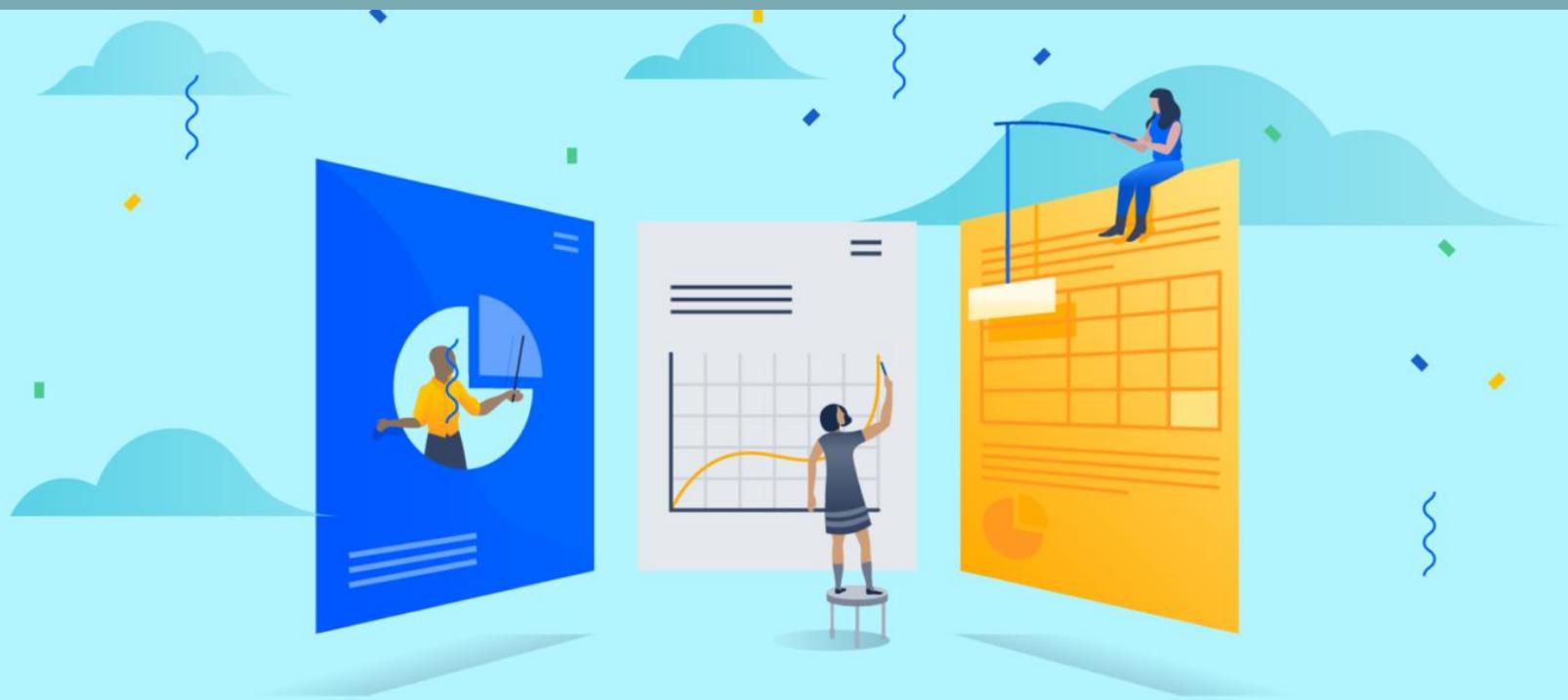
Team Coordination : Release Train

- ❖ Cross-team synchronization.
- ❖ Focuses on fast, flexible flow at the level of a larger product.
- ❖ The train metaphor is used .
 - ❖ Features will “leave the station.”
 - ❖ Need to get their “cargo” onto the train at the appointed time.
 - ❖ The release train always “departs” on time and “waits” for no one
 - ❖ if a team misses the train, there will be another train departing at a known time in the future

Team Coordination : Release Train



- Nine teams clustered into three feature areas
- Each team within a feature area performs its own sprint
- Using a technique like scrum of scrums, all the teams within a feature area coordinate their work



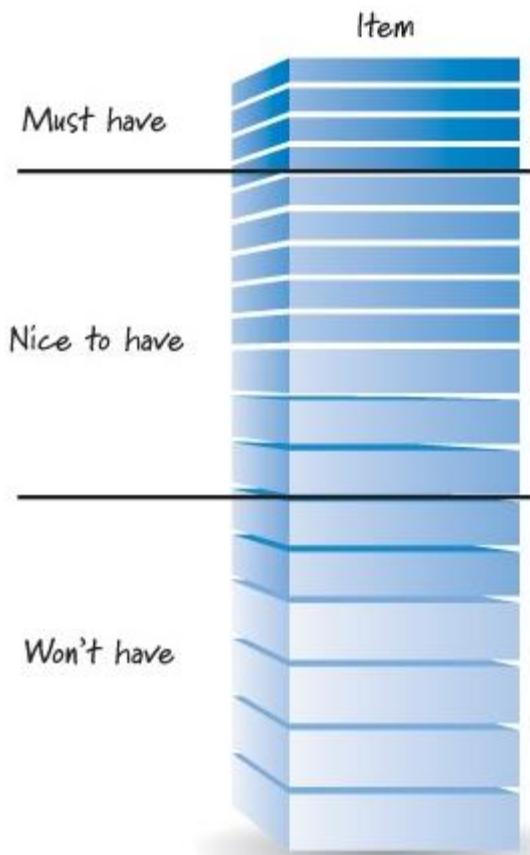
SCRUM Artefacts

Artefacts



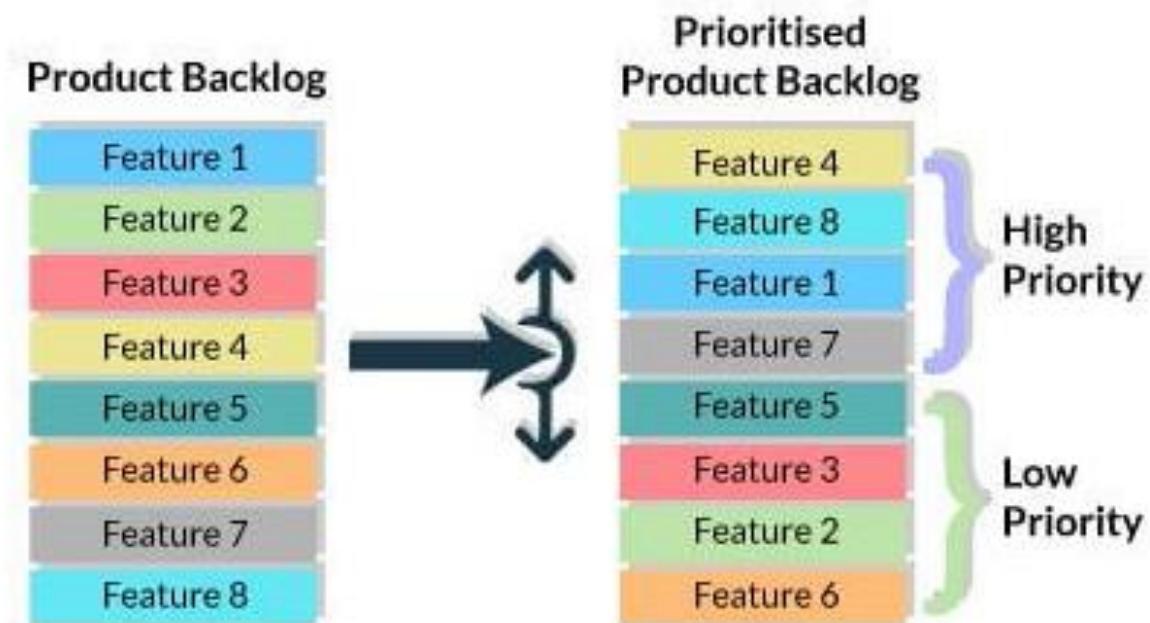
Product Backlog

- A list of
 - the new features
 - changes to existing features
 - bug fixes
 - infrastructure changes
- product owner has the responsibility
 - prioritizing product backlog items,
 - deciding which product backlog items should be removed
 - Facilitating product backlog Grooming.

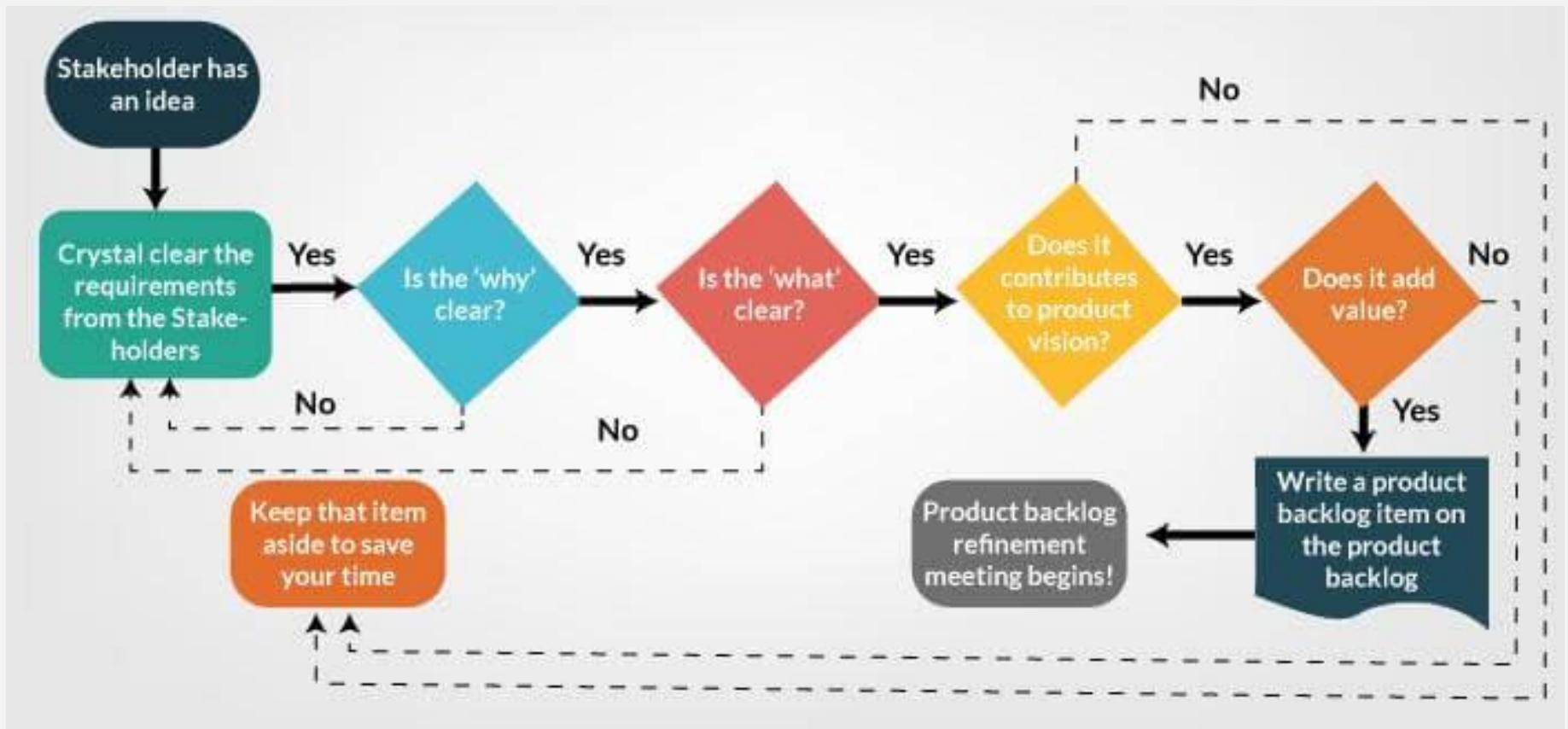


Product Backlog Grooming

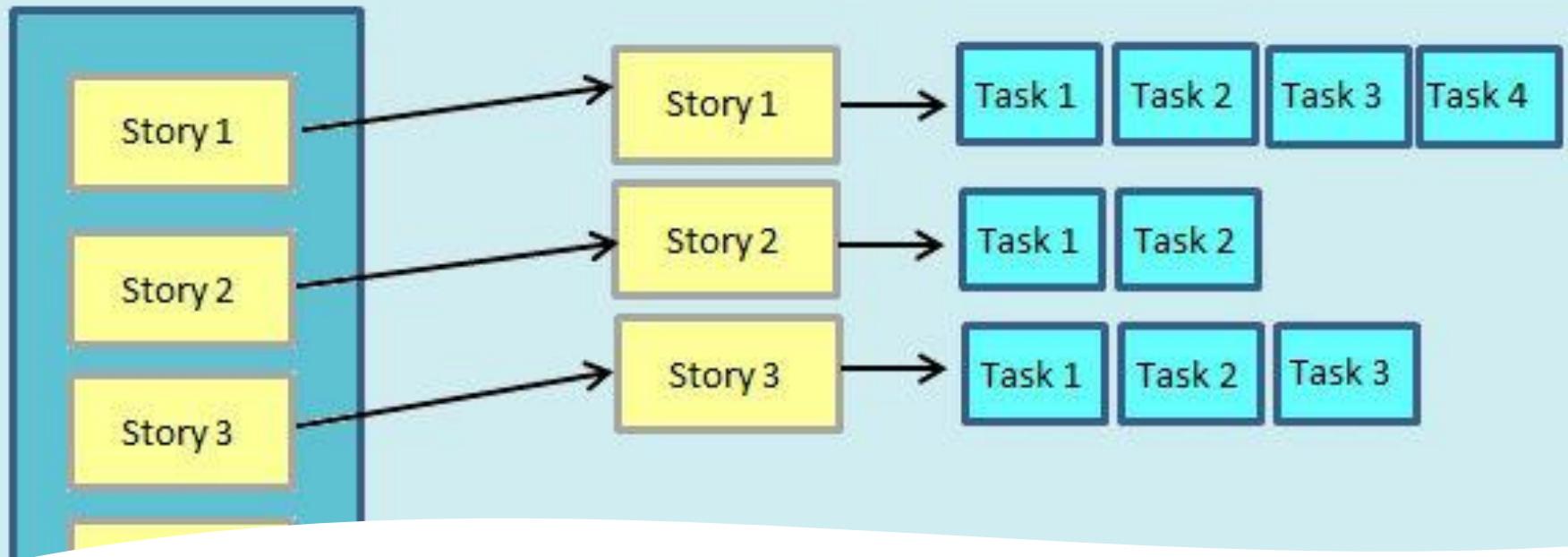
- Product Backlog Refinement (PBR)
- Why Groom?
 - To reduce the time of Sprint Planning



Product Backlog Grooming meeting



Product Backlog



Sprint Backlog

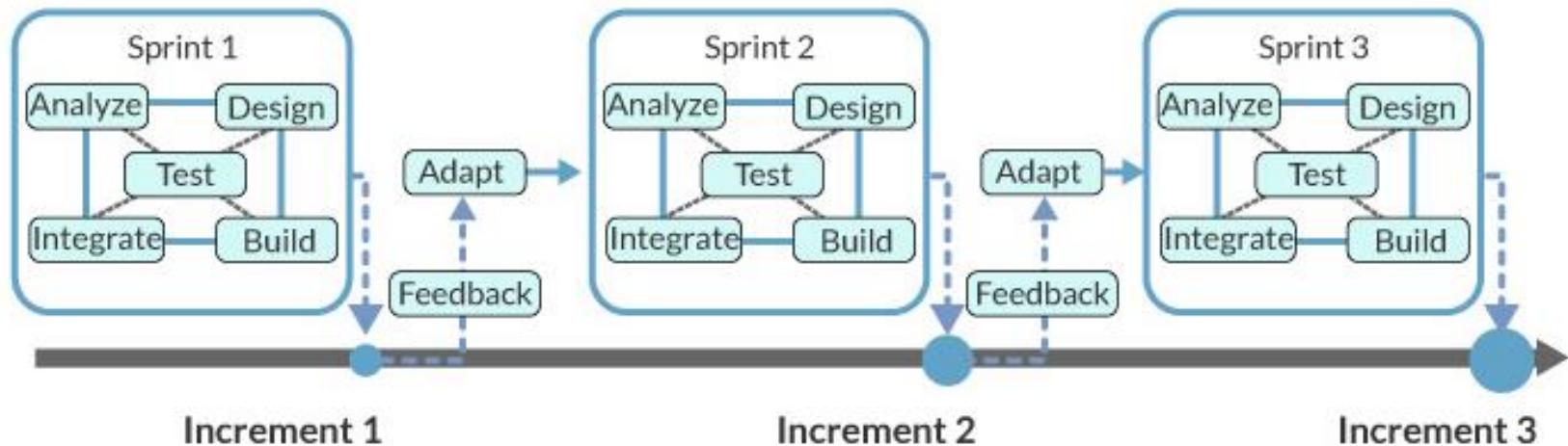
- List of tasks to be completed by the end of the sprint.
- Picked from the Product Backlog based on the priorities set by the Product Owner

Selecting items to sprint backlog

- ❖ Selection can be done in several ways
 - ❖ If a sprint goal is available
 - ❖ select product backlog items that align with that goal
 - ❖ If a sprint goal is unavailable
 - ❖ select items from the top of the product backlog
- ❖ Start-only-what-you-can-finish rule (HW)

Product Increment

- The most important final Artifact in the Scrum framework is the actual Product Increment.
- This is a potential shippable product





SCRUM Planning

Traditional Projects vs SCRUM Projects

- ❖ Traditional projects : Creates a detailed plan up front before development work begin.
 - ❖ get it right at the beginning so that rest can follow in an orderly fashion
- ❖ SCRUM : Does not believe it can be done in the very beginning
 - ❖ Can't produce all of the planning artifacts up front
 - ❖ Up-front and just-in-time planning

SCRUM Planning Principles

- ❖ Plan to learn fast and pivot when necessary
 - ❖ Doing something
 - ❖ Learning fast
 - ❖ Changing directions if needed

Multilevel Planning

- ❖ Plan at multiple levels of detail and at multiple times throughout development.
 - ❖ Different levels of planning
 - ❖ Portfolio
 - ❖ Product
 - ❖ Release
 - ❖ Sprint
 - ❖ Daily
- SCRUM formally define only these two

Portfolio Planning

- ❖ Determine which products to work on, in what order, and for how long.

Product Planning

- ❖ Product vision
 - ❖ provides a clear description of the value stakeholders get value.
- ❖ High-Level Product Backlog
- ❖ Product Roadmap

For people worldwide who are interested in Scrum, the new Scrum Alliance website will be their trusted source of Scrum knowledge. It will be feature and content rich and will be their first stop on the Internet for learning more about Scrum or to collaborate on Scrum topics of interest

The outputs of product-level planning became inputs to portfolio planning

Release Planning

- ❖ Making scope, date, and budget trade-offs for incremental deliveries.
- ❖ To get an idea of what you can deliver by a fixed date -create and estimate a sufficient number of product backlog items
 - ❖ Draw a line through the product backlog
 - ❖ All of the items above the line are planned for the release
- ❖ Time dimension

Plan just in time and just enough

Release Planning

- ❖ Helps answer
 - ❖ When will we be done?
 - ❖ Which features can I get by the end of the year?
 - ❖ How much will this cost?
- ❖ Longer term planning
- ❖ Every organization must identify the proper sequence for releasing features
 - ❖ Different ways of releasing
 - ❖ Release after multiple sprints
 - ❖ Release every sprint
 - ❖ Release every feature
- ❖ Also known as Longer-term planning or Milestone driven planning

Release planning involves collaboration between the stakeholders and the full Scrum team

Sprint Planning

- ❖ Determine the specific product backlog items that the Scrum team will work on in the next sprint

Daily Planning

- ❖ The most detailed level of planning
- ❖ Team members get together and each person takes turns stating what he/she got done since the last daily scrum and the plan for today

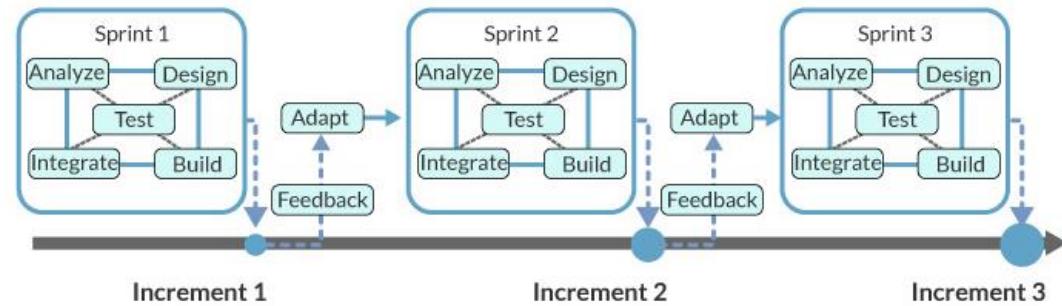
“Today I am going to work on the stored procedure task, and I should have that done by lunch. Whoever is going to work on the business logic task, please keep in mind that the business logic task is on the critical path for getting our work done this sprint and you should be ready to work on it right after lunch.”



Sprint Planning

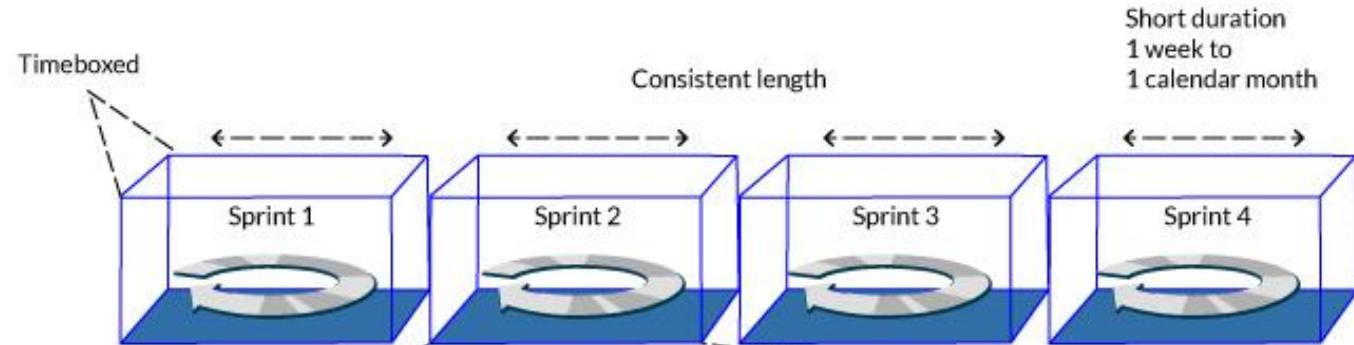
Introduction

- ❖ Product backlog represents weeks/months of work
- ❖ Sprint planning is used to determine the most important subset of product backlog items to build in the next sprint.
- ❖ A recurring, just-in-time activity that takes place at the beginning of each sprint
- ❖ The combination of product backlog items and the plan forms the sprint backlog.

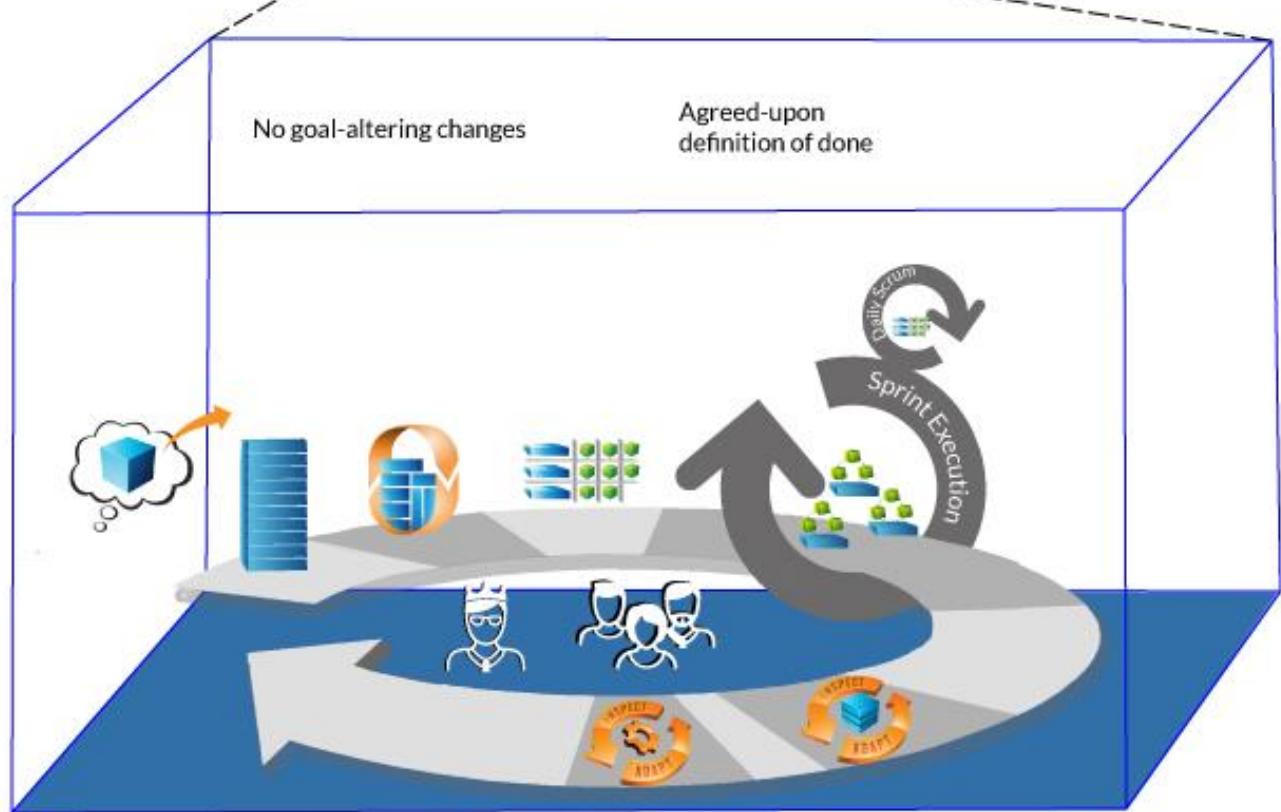


Sprint

- A **Sprint** is the heart of Scrum
- Time-box of 2-4 weeks.
- Sprints should be of consistent length.
- Goals should not be altered



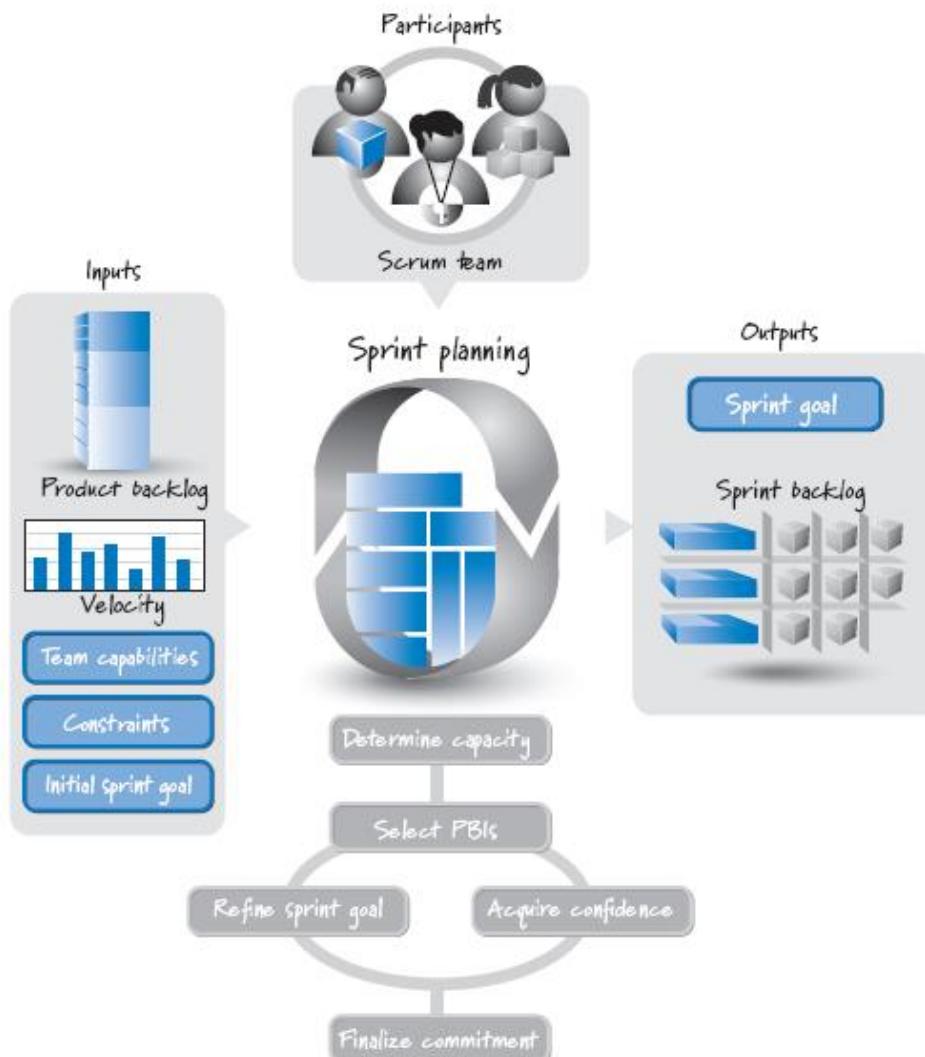
SCRUM is iterative and incremental through Sprints



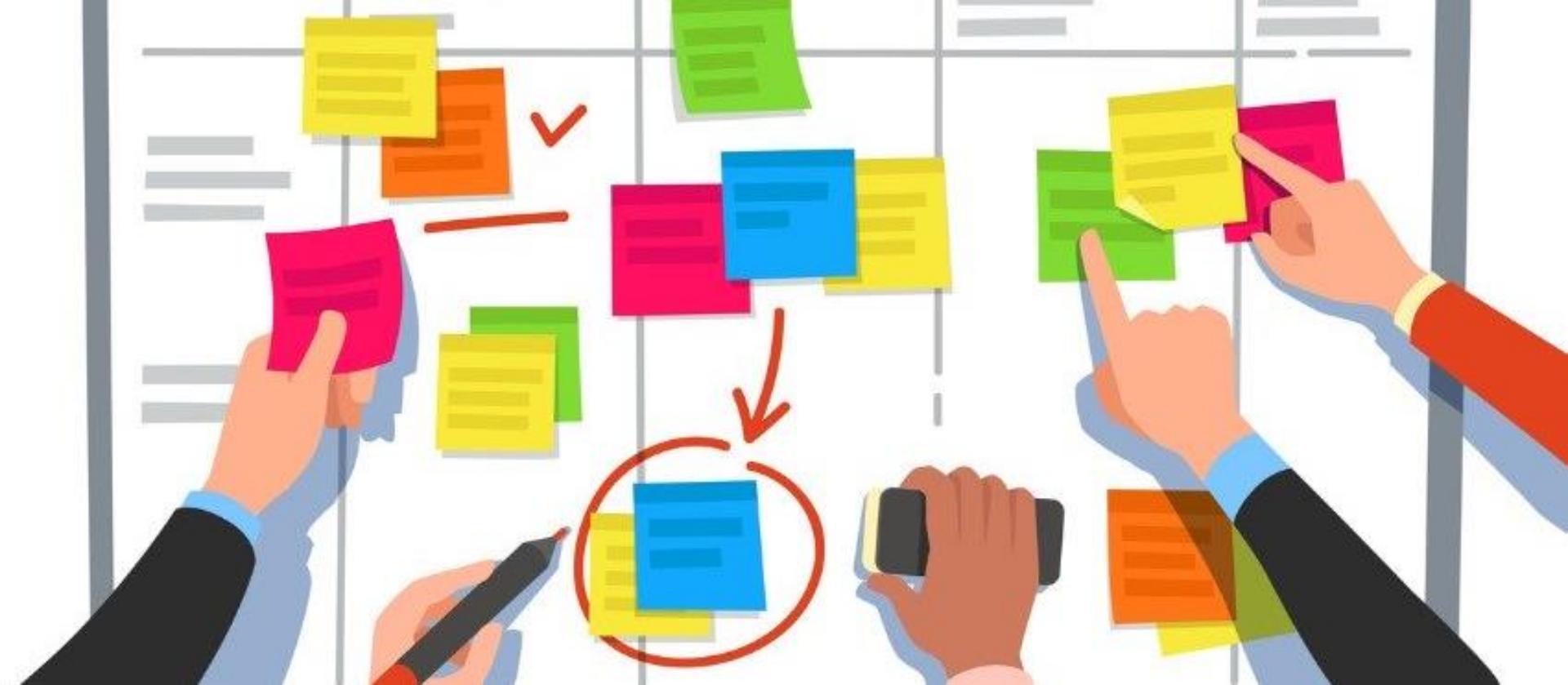
Sprint Planning - Participants

- ❖ The full Scrum team collaborates
- ❖ The product owner
 - ❖ Shares the initial sprint goal
 - ❖ presents the prioritized product backlog
 - ❖ Answer questions
- ❖ The development team
 - ❖ determine what can be delivered after the sprint
- ❖ The Scrum Master
 - ❖ Acts as the Scrum team coach
 - ❖ Observes the planning activity
 - ❖ Asks questions and facilitate the meeting

Sprint Planning - Process



- ❖ Inputs
 - ❖ Product backlog
 - ❖ Team velocity/ Capacity
 - ❖ Constraints
 - ❖ Team capabilities
 - ❖ Initial sprint goal



Sprint Planning Methods

- Velocity-based Sprint Planning
 - Capacity-based Sprint Planning

Story Points

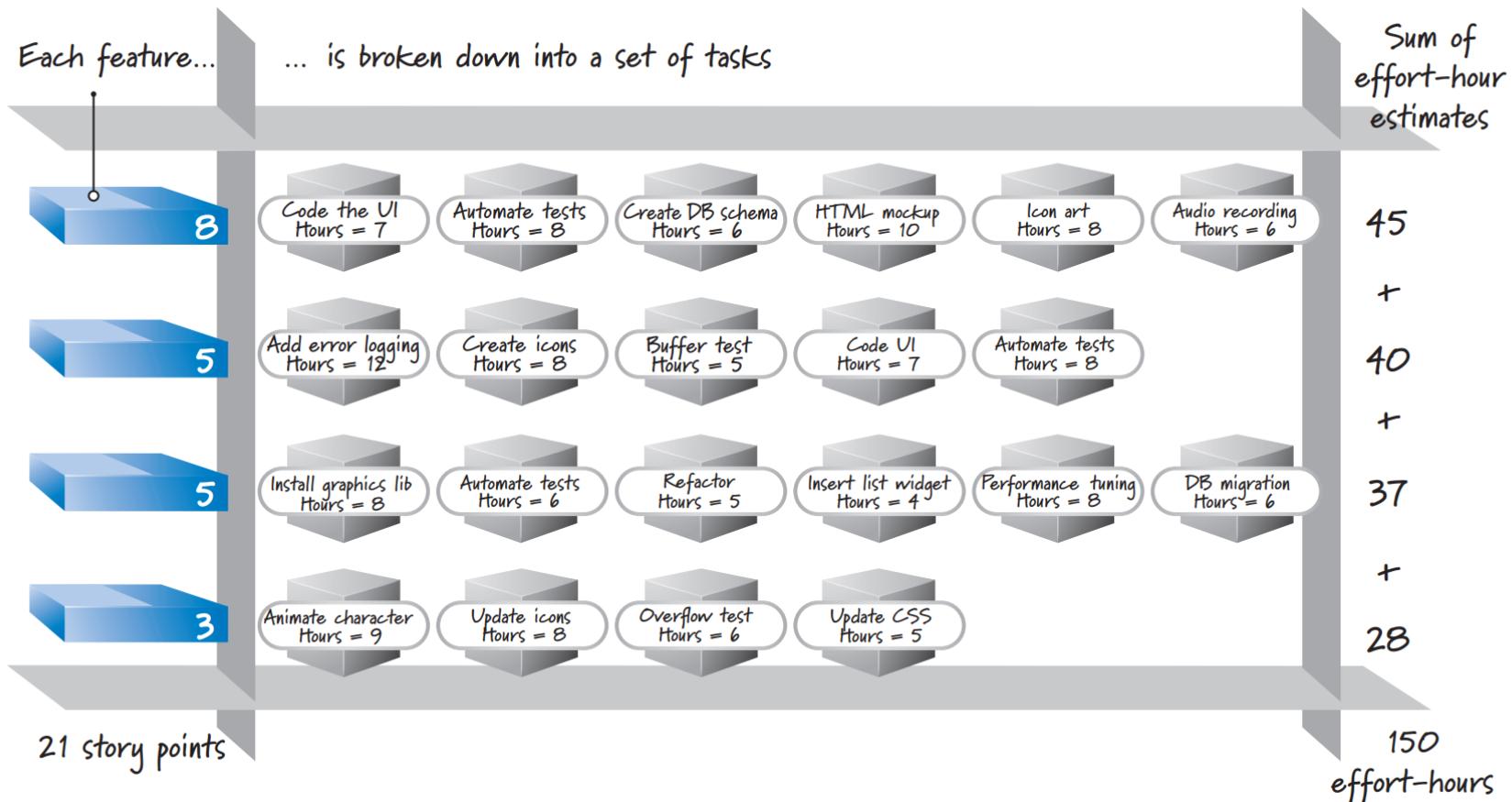
- Story points
 - A high-level value you assign to a user story



- ❖ Influenced by several factors, mainly complexity and physical size
 - ❖ The end result is small, but the effort is high.
 - ❖ A story might be physically quite big but not complex.

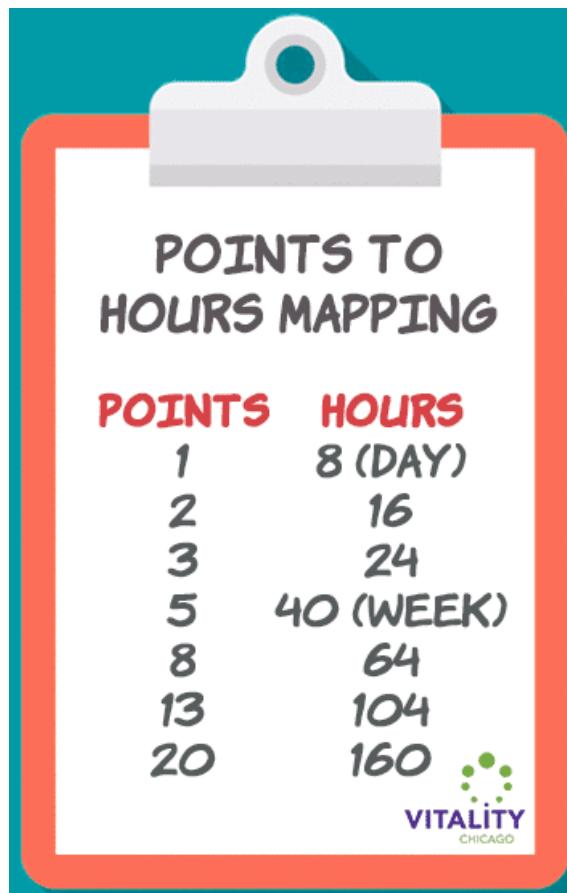
Effort Hours

- Effort hours represent the actual effort in man hours needed to complete all the tasks involved in a user story



Effort Hours

- Story point to effort hours



A clipboard icon with a teal header and a white notepad. The notepad has a red border and contains a table titled "POINTS TO HOURS MAPPING".

POINTS	HOURS
1	8 (DAY)
2	16
3	24
5	40 (WEEK)
8	64
13	104
20	160

VITALITY
CHICAGO

Velocity-based Planning

- The amount of work finished in each sprint
 - Calculated by adding all the story points given to each user story or Product Backlog Item (PBI)
- Why?
 - To help in sprint planning

Velocity-based Planning

- Main steps
 - Calculate the team's average velocity (from last 3 Sprints)
 - Select the items from the product backlog equal to the average velocity
 - Verify whether the tasks associated with the selected user stories are appropriate for the particular sprint
 - Estimate the tasks and check if the total work is consistent with past sprints

Velocity-based Planning

Team Ignition

Sprint 1



Sprint 2



Sprint 3



Sprint 4



28

\sum

13

36

\sum

28

\sum

30

Velocity: 30.5

Capacity-based Planning

- Also called Commitment-based Sprint planning
- Focus on team's available capacity (in hours) for the sprint
- Best because;
 - The capacity of the teams may vary from one sprint to another
 - Story points are coarse-grained where as hours are fine-grained

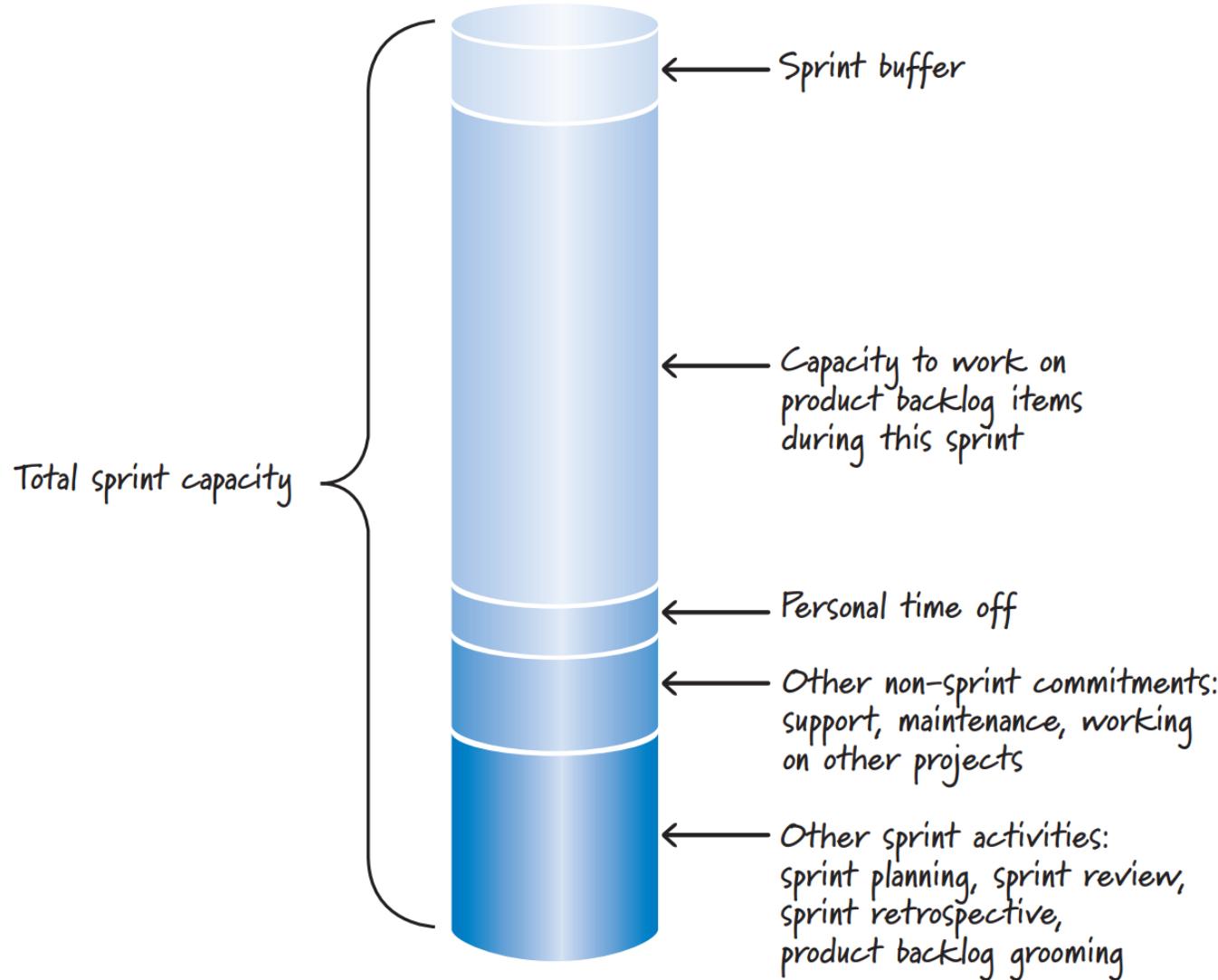
Capacity based Planning - Calculation

- How to know the team's capacity?
- Consider a team consisting of 6 people working for 8 hours per day for 3 weeks sprint (15 days).

$$\begin{aligned}\text{Team's Capacity} &= \text{number of team members} * \text{time in hours} * \text{days} \\ &= 6 * 8 * 15 \\ &= 720 \text{ hours}\end{aligned}$$

- Burn out the team
- Make them rush to reach the end goal
- Hamper the quality.

Capacity based Planning - Calculation



Capacity based Planning - Calculation

- Focus Factor
 - Capability of the teams to stay focused on the sprint goals without any impediments
 - lies in the range of 0.6 - 0.8.

Real capacity = total capacity * focus factor

Real Capacity = $720 * 0.6 = 432$ hours

Capacity based Planning - Calculation

- Team can use 432 hours
- Selects stories from the backlog
- divide those stories into tasks
 - Estimate tasks per hour
- Pick up the stories until the time-length does not exceed more than 432 hours



SPRINT Execution

Introduction

- ❖ The work the Scrum team performs to meet the sprint goal
- ❖ Accounts for the majority of time during a sprint.
 - ❖ begins after sprint planning
 - ❖ ends when the sprint review starts

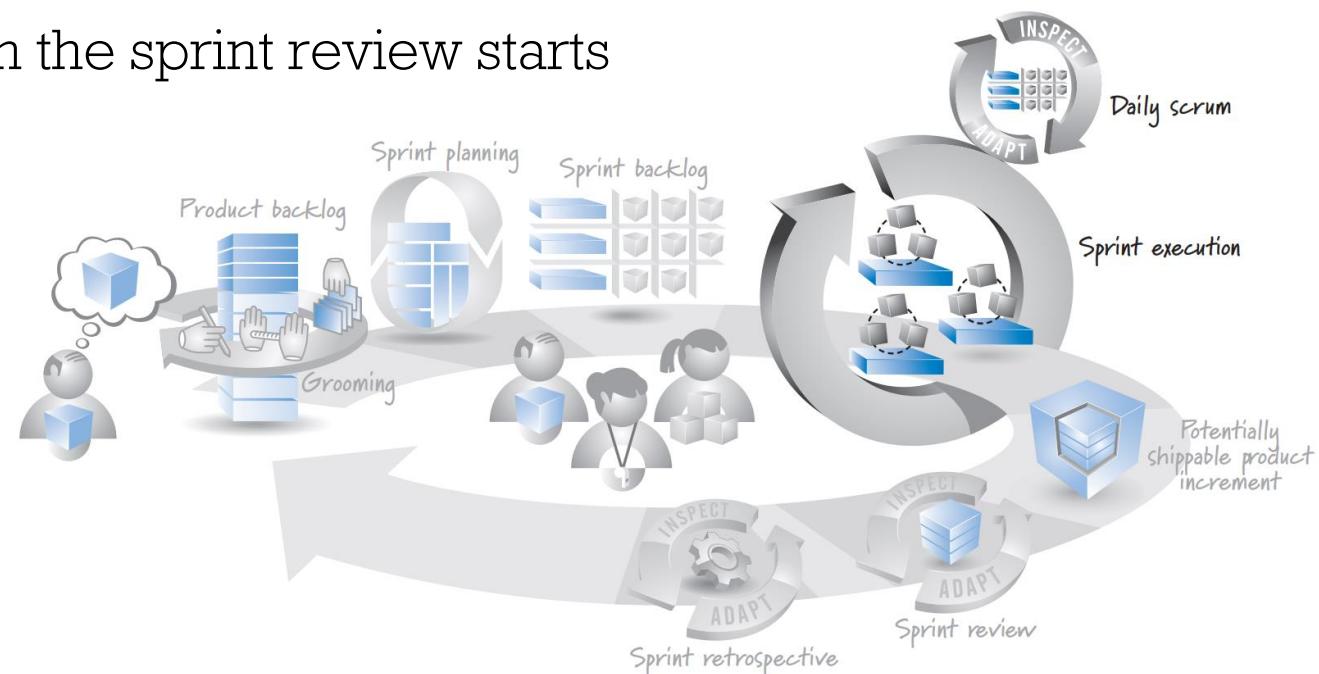


FIGURE 10.1

Flow Management

- ❖ The team is responsible for managing the flow of work during sprint execution
 - ❖ Decide on
 - ❖ Parallel work and swarming ?
 - ❖ Which work to start ?
 - ❖ How to organize task work ?
 - ❖ What work needs to be done ?
 - ❖ who does the work ?

Daily Scrum

- ❖ Daily inspect-and-adapt activity to help the team achieve faster, more flexible flow toward the solution
- ❖ 15-minute, time boxed activity that takes place once every 24 hours
- ❖ The goal is to give the team an idea of what is happening so that they can understand ,
 - ❖ How much to work on
 - ❖ Which items to start working on
 - ❖ How to best organize the work among the team member

Communicating

- ❖ Since the team size is small , you don't need complex charts or diagrams to communicate the progress
- ❖ The commonly used techniques are
 - ❖ Task Board
 - ❖ Burndown and/or Burnup charts

Communicating -Task Board

- ❖ The task board shows the evolving state of the sprint backlog over time

Story	To Do	In Progress	Done
Story A		Task	Task
Story B	Task	Task	Task
Story C		Task	Task

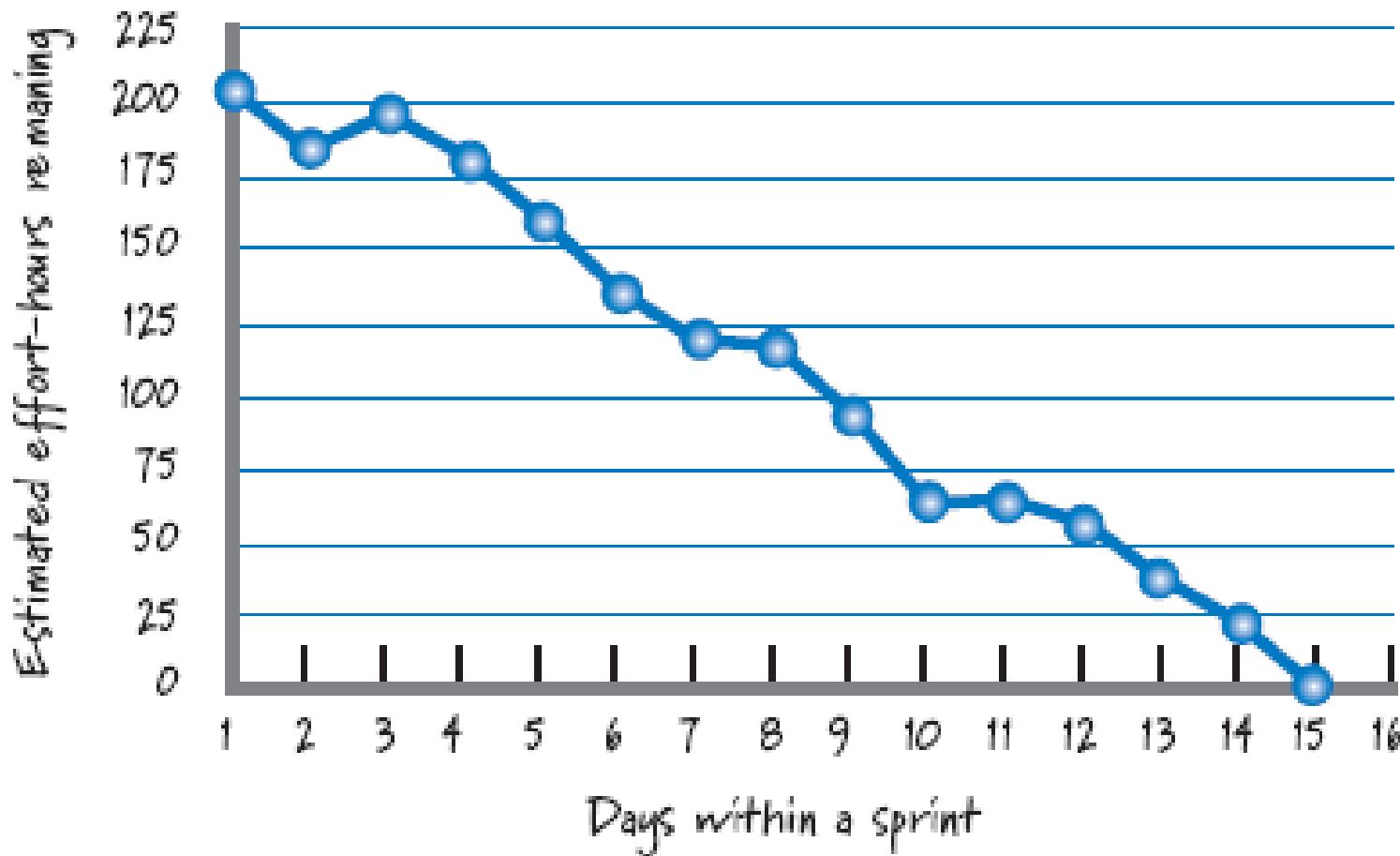
Communicating - Sprint Burndown Chart

- During sprint execution team members update the estimate of how much effort remains for each uncompleted task.

Tasks	D1	D2	D3	D4	D5	D6	D7	D8	D9	...	D15
Task 1	8	4	4	2							
Task 2	12	8	16	14	9	6	2				
Task 3	5	5	3	3	1						
Task 4	7	7	7	5	10	6	3	1			
Task 5	3	3	3	3	3	3	3				
Task 6	14	14	14	14	14	14	14	8	4		
Task 7						8	6	4	2		
Tasks 8-30	151	139	143	134	118	99	89	101	84		0
Total	200	180	190	175	155	130	115	113	90		0

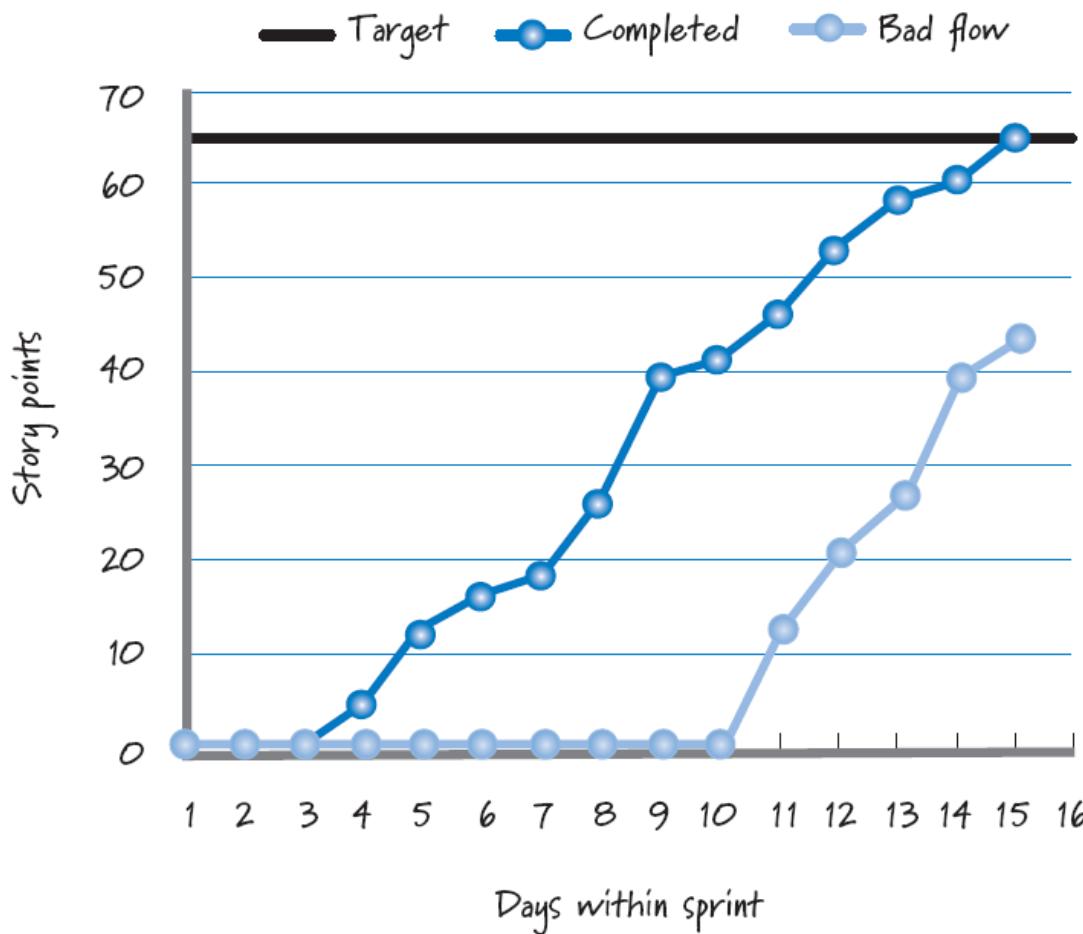
- Plot the row labeled “Total” in a graph – Sprint burndown chart

Communicating - Sprint Burndown Chart



Communicating - Sprint Burnup Chart

- A way to visualize progress through a



the work can be represented in either effort-hours or in story points . Here we have used effort hours .



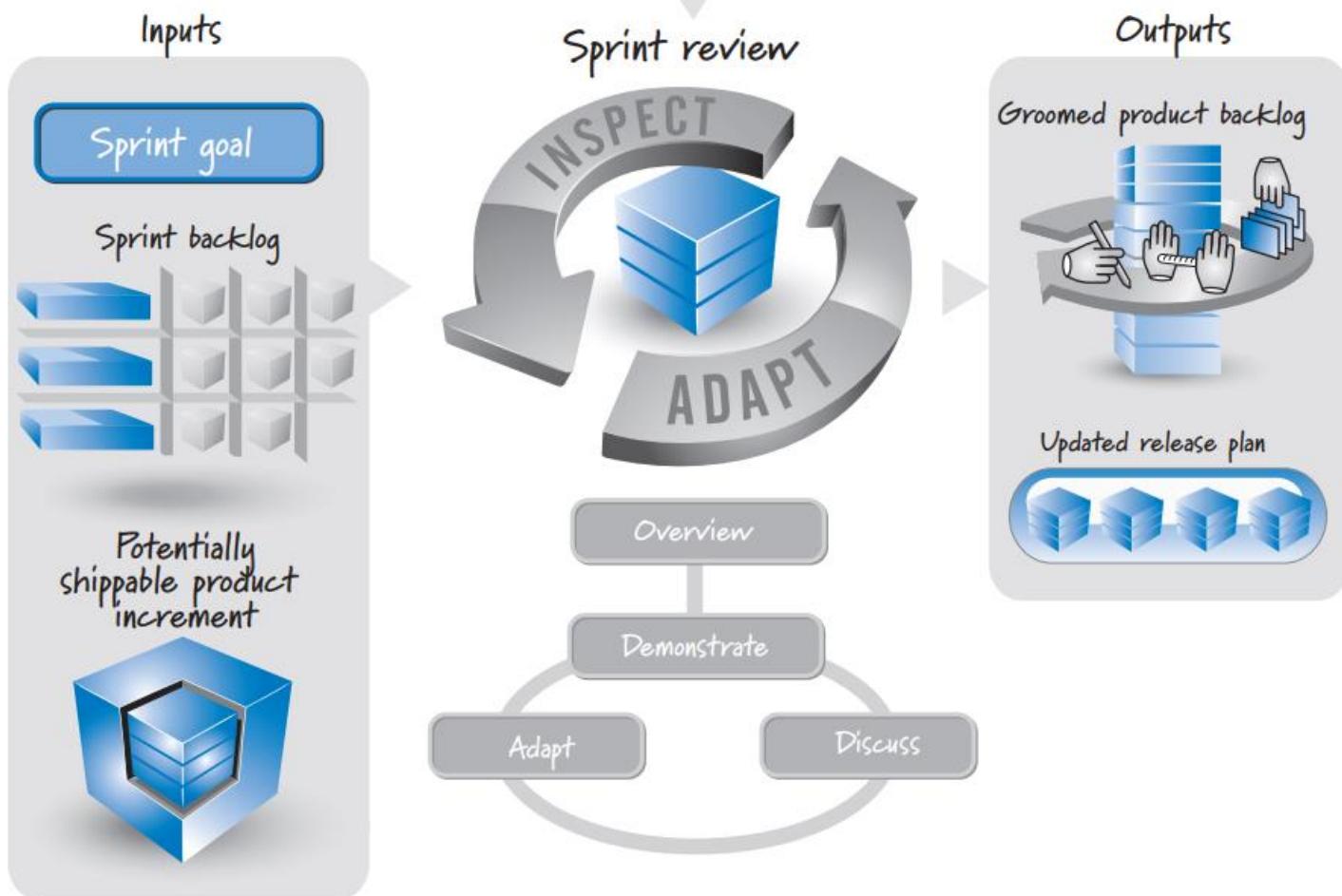
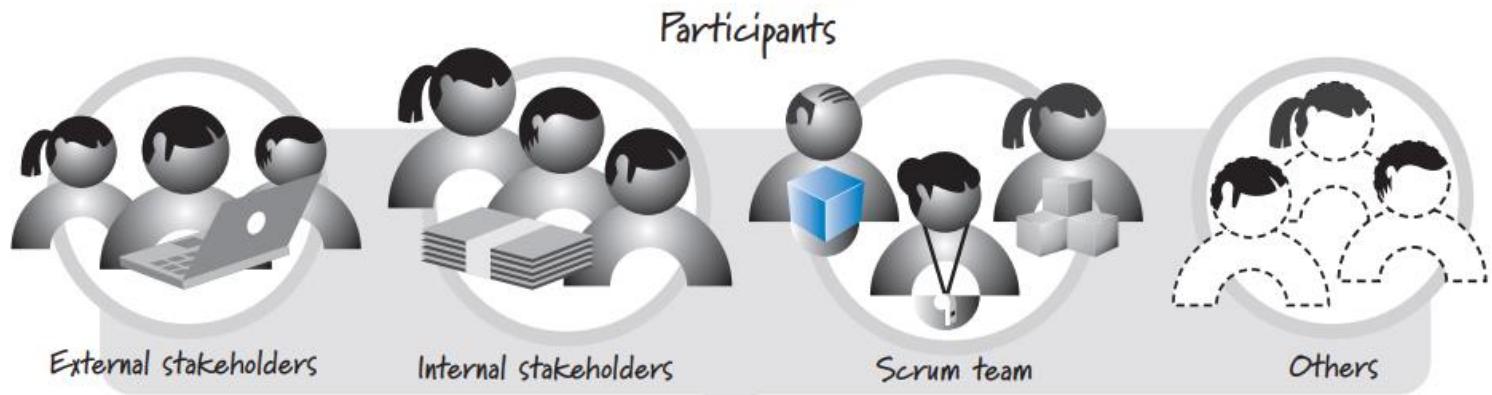
SPRINT Review

Introduction

- ❖ Near the end of the sprint, the team conducts two important inspect-and-adapt activities:
 1. The sprint review
 - focuses on the product itself.
 2. The sprint retrospective
 - looks at the process used
- ❖ During sprint planning we plan the work.
- ❖ During sprint execution we do the work.
- ❖ During sprint review we inspect the result
 - ❖ occurs near the end of each sprint cycle

Approach

- ❖ The outputs of the sprint review are a groomed product backlog and an updated release plan.
- ❖ Usually in the review ,
 - ❖ Provides a summary of what has and has not been accomplished
 - ❖ A demonstration of the increment
 - ❖ Discuss the current state of the product, and adapting the future product direction.

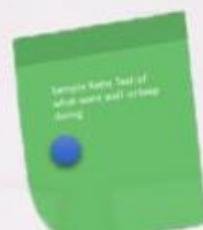


Sprint Review Issues (HW)

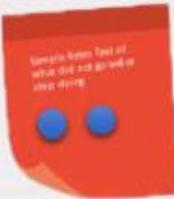
- Sign-offs
- Sporadic Attendance
- Large Development Efforts

Team Agile Digest Sprint X Retrospective

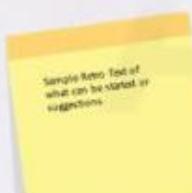
What went well



What did not go well



Any suggestions



Action Item

SPRINT Retrospective

SPRINT RETROSPECTIVE

- ❖ The sprint retrospective is one of the most important and least appreciated practices in the Scrum framework.
- ❖ A sprint retrospective can be as simple as the Scrum team members coming together to discuss questions such as
 - ❖ What worked well this sprint that we want to continue doing?
 - ❖ What didn't work well this sprint that we should stop doing?
 - ❖ What should we start doing or improve?
- ❖ The full scrum team takes part in this.

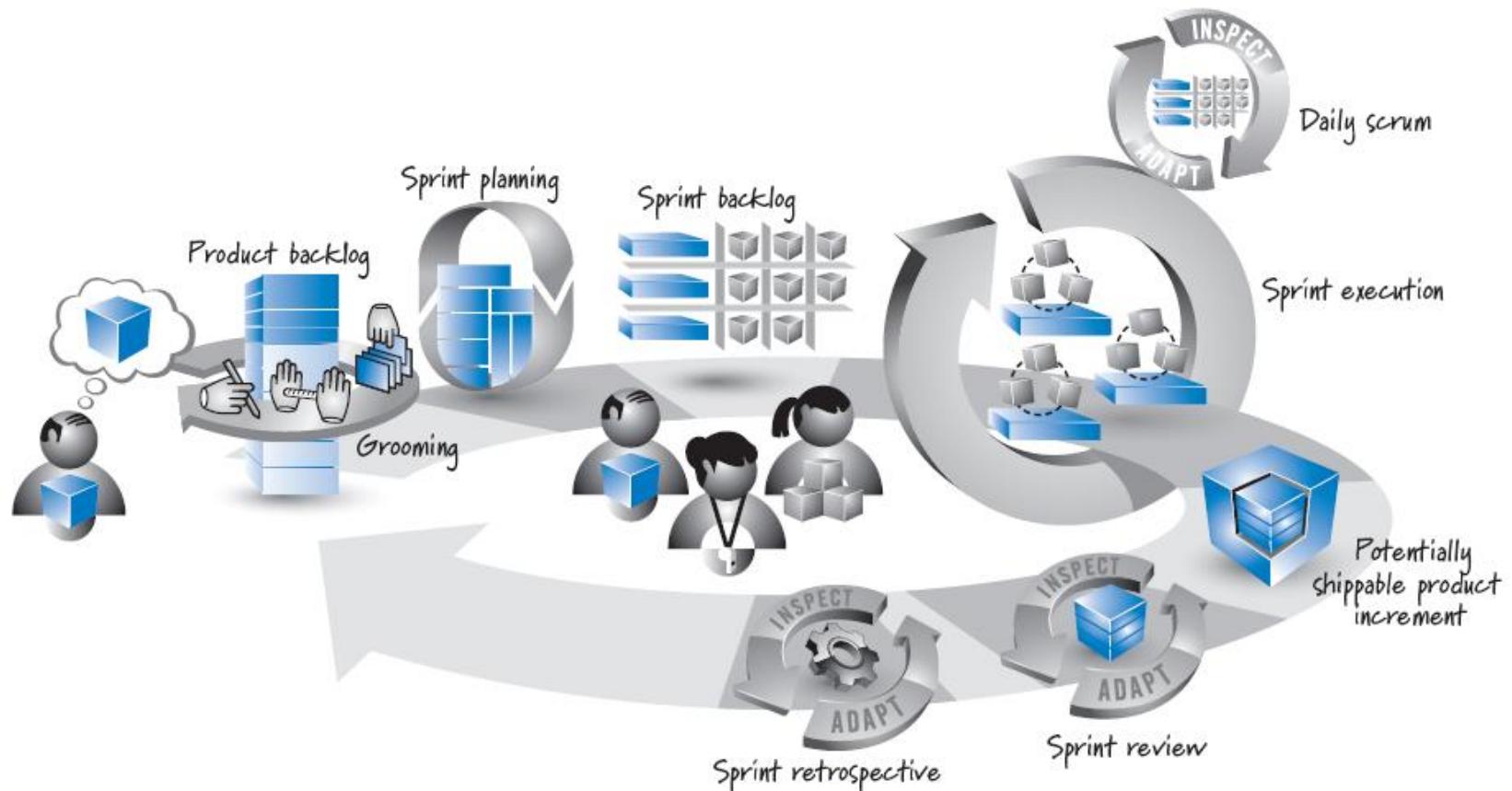
Approach

- ❖ The outputs of the sprint retrospective include
 - ❖ a set of concrete improvement actions that the team has agreed to perform in the next sprint
 - ❖ a backlog of insights collected during the current retrospective that the team will not address in the upcoming sprint but might choose to address in the future
 - ❖ improved camaraderie
- ❖ One approach is to
 - ❖ set the atmosphere
 - ❖ create a shared context among the participants
 - ❖ identify insights that can lead to improvements determine concrete
 - ❖ improvement actions to take during the next sprint
 - ❖ close the retrospective

Sprint Retrospective Issues

- ❖ Not doing the retrospective or low attendance
- ❖ All fluff, no stuff
- ❖ Ignoring the elephant in the room
- ❖ Poor facilitator
- ❖ Depressing and energy draining
- ❖ Blame game
- ❖ Complaint session
- ❖ Replaces ad hoc process improvement
- ❖ Too ambitious
- ❖ No follow-through

Summary



The background of the slide features a repeating pattern of colorful hand silhouettes, rendered in various colors like red, green, blue, yellow, orange, and purple, all pointing upwards towards the center of the frame.

Thank You

tch@ucsc.cmb.ac.lk

Summary

- Agile stresses four key issues:
 1. Importance of self-organizing teams that have control over the work they perform
 2. Communicate and collaborate between team members and between practitioners and their customers
 3. A recognition that change represents an opportunity
 4. Emphasis of rapid delivery of software that satisfies the customers