Agile Workshop

Rahul Gujar

SPC SAFe® 4

CERTIFIED DEVOPS FOUNDATION® DEVOPS INSTITUTE

AgilePM Practitioner

APMG International™ AgilePM

CSM ScrumAlliance®

Project Management Institute PMP® Certified

DISCIPLINEDAGILE CONSORTIUM

Agenda

1. Agile Principles — 50
2. Scrum Framework — 20
3. Scrum Roles — 50
4. Scrum Artifacts — 100
5. Scrum Events — 100
6. Agile Estimation & Planning — 150
7. Agile in Action — 100
8. Agile Readiness — 50

What is Agile

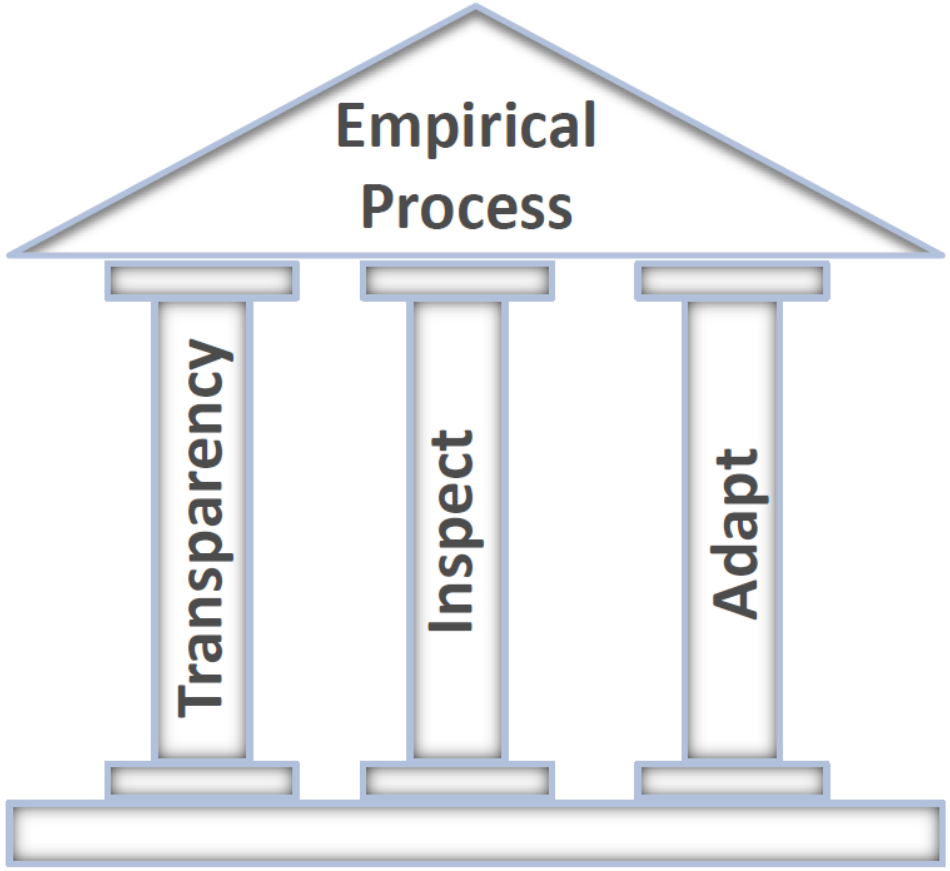
What is Agile

It’s a philosophy that uses organizational models based on

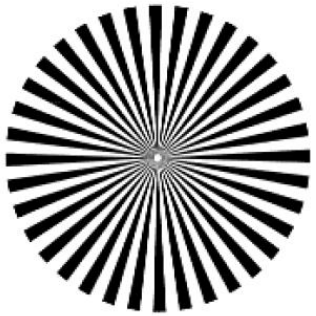
Exercise: LET'S PLAY A GAME

Passing the Ball

House of Principles



Values

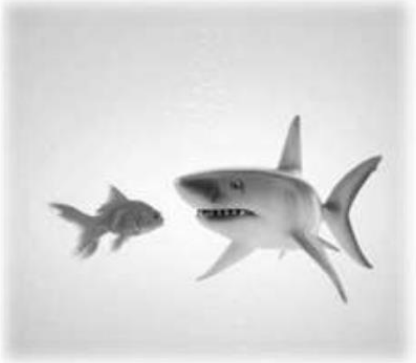


Focus

Respect

Commitment

Openness



Courage

Agile Manifesto

Image Source: http://udayanbanerjee.wordpress.com/category/agile

Agile Manifesto

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to 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

That is, while there is value in the items on the right, we value the items on the left more

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 for 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 they need, 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.

agilemanifesto.org/principles.html

Agile Principles

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.

agilemanifesto.org/principles.html

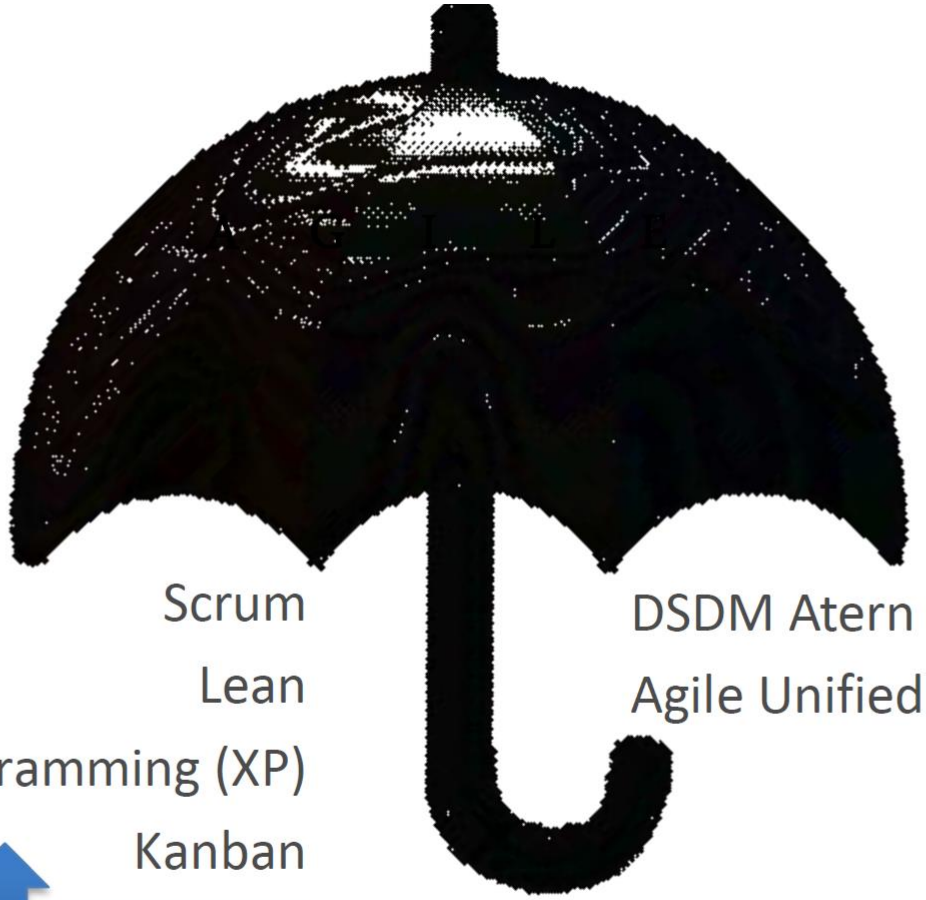
Exercise:

Discuss at each table on what would you do differently based on what you have learnt in Agile Principles.

Acceptance criteria:

- Share 1-2 ideas

Agile Methodologies



Scrum

Lean

Extreme Programming (XP)

Kanban

DSDM Atern

Agile Unified Process (AUP)

Lightweight Approaches

Fuller Approaches (but still agile)

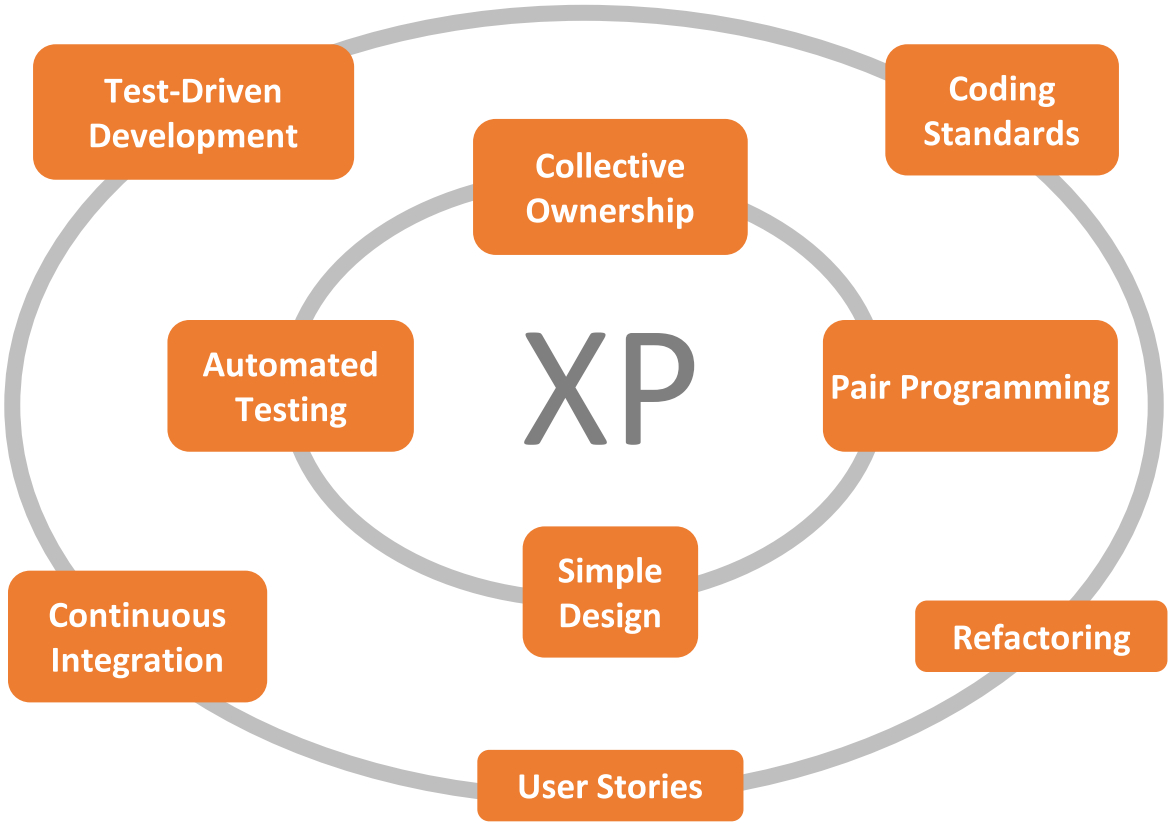
Extreme Programming (XP) in one slide

XP practices drive code quality to unprecedented levels.

Most high-performance teams use Scrum and XP together.

It is hard to get a Scrum with extreme velocity without XP engineering practices.

—Jeff Sutherland, co-creator of Scrum



Adapted from xprogramming.com

Why Do Companies Adopt Agile

- Accelerate Time to Market
- Manage Changing Priorities
- Increase Productivity
- Better Align IT/Business
- Enhance Software Quality
- Improve Project Visibility

WHO Uses Scrum

- ✓ Google
- ✓ Oracle
- ✓ Salesforce.com
- ✓ Toyota
- ✓ BBC
- ✓ Nokia
- ✓ Lockheed Martin
- ✓ Philips
- ✓ Siemens
- ✓ Capital One
- ✓ Time Warner
- ✓ Yahoo

Questions

Scrum Framework

Scrum Explained

"The… 'relay race' approach to product development…may conflict with the goals of maximum speed and flexibility. Instead a holistic or *'rugby' approach—where a team tries to go the distance as a unit, passing the ball back and forth—may better serve today's competitive requirements."* - Hirotaka Takeuchi and Ikujiro Nonaka, "The New New Product Development Game", Harvard Business Review, January 1986

In Scrum you work in **iterations** delivering **value-adding** results **incrementally**

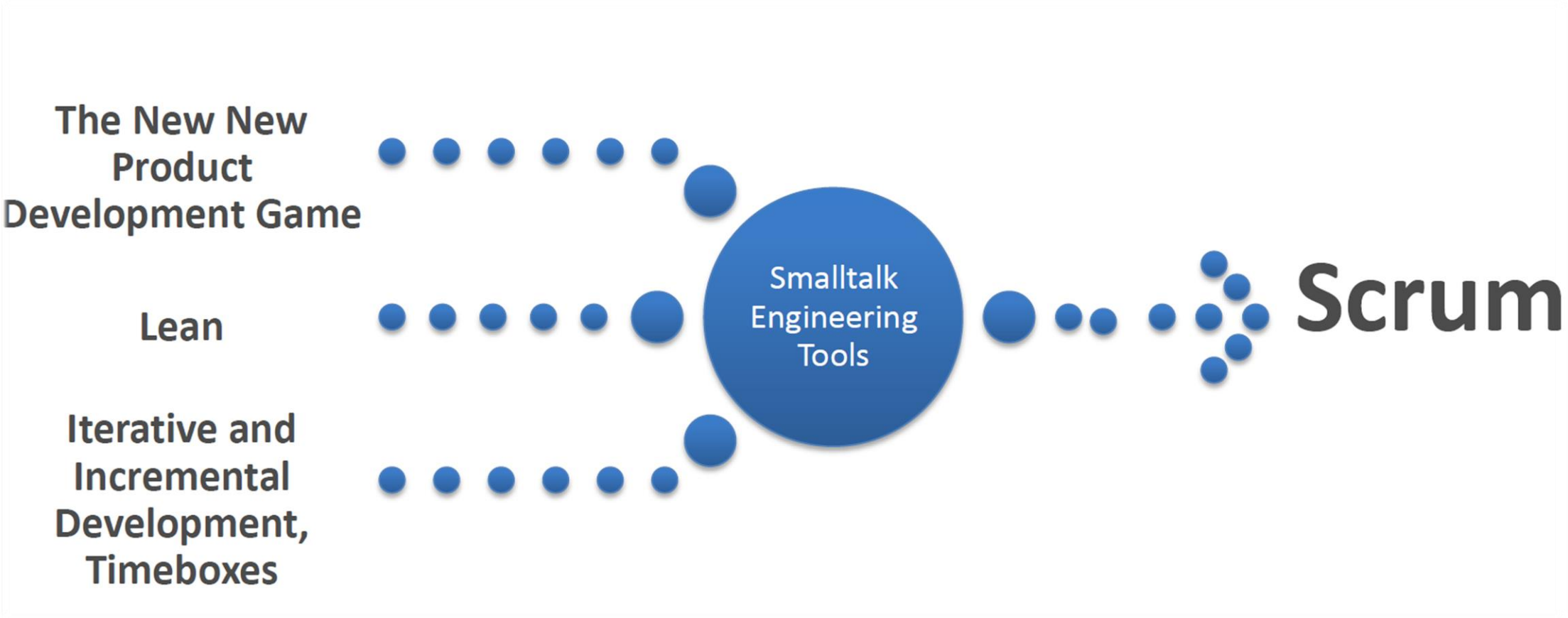
Scrum Definition

A **framework** within which people can address **complex adaptive problems** while **productively and creatively delivering** products of the **highest possible value**"

Jeff Sutherland

Ken Schwaber

Influences of Scrum



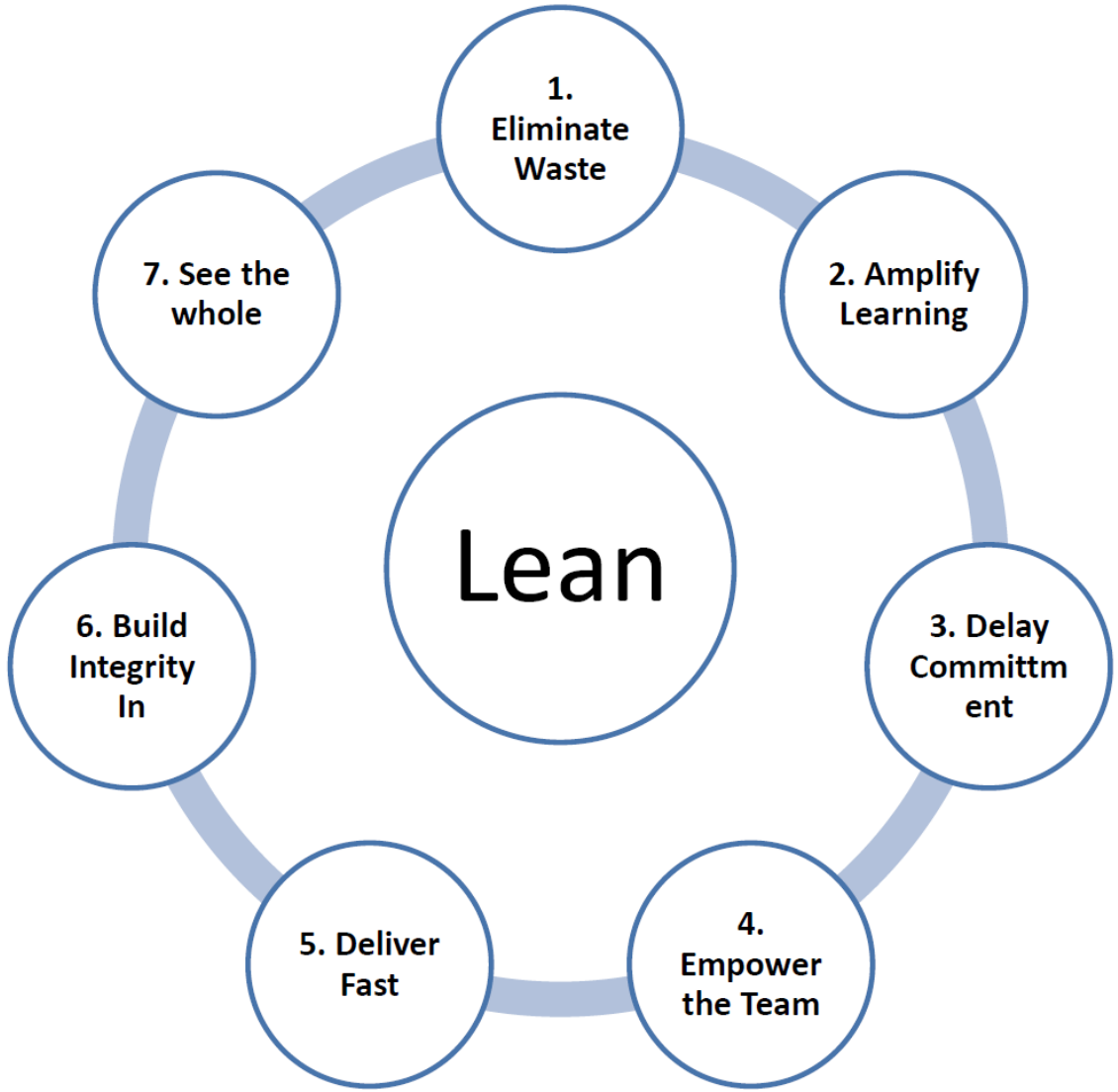
The New New Product Development Game

The New New Product Development Game

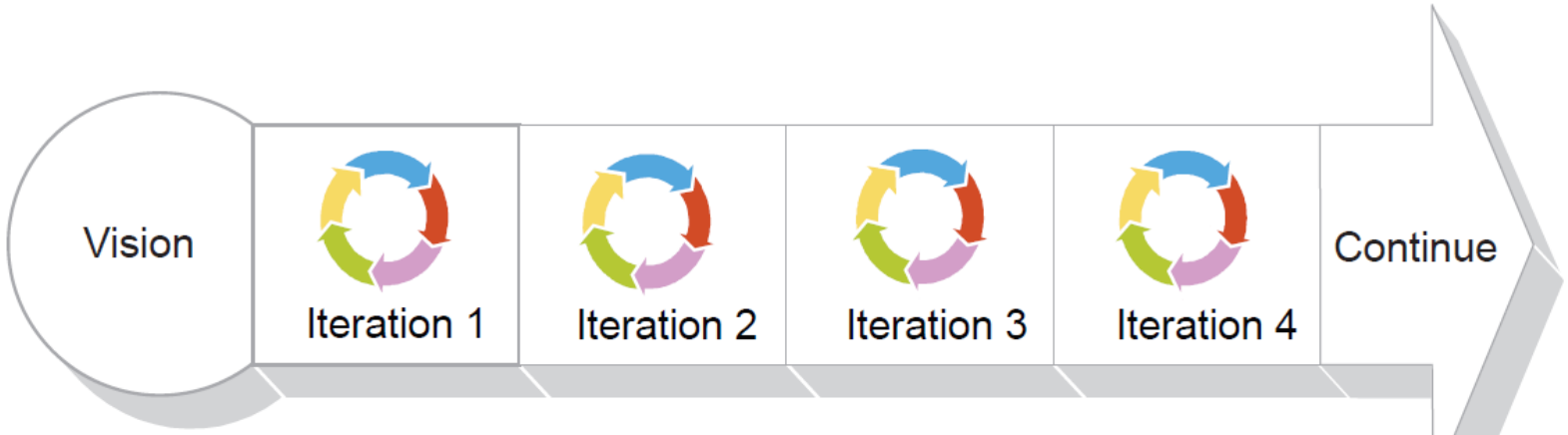
by Hirotaka Takeuchi and Ikujiro Nonaka

- Built-in instability
- Self-Organizing project teams
- Overlapping development phases
- Multilearning
- Subtle Control
- Organizational transfer of learning

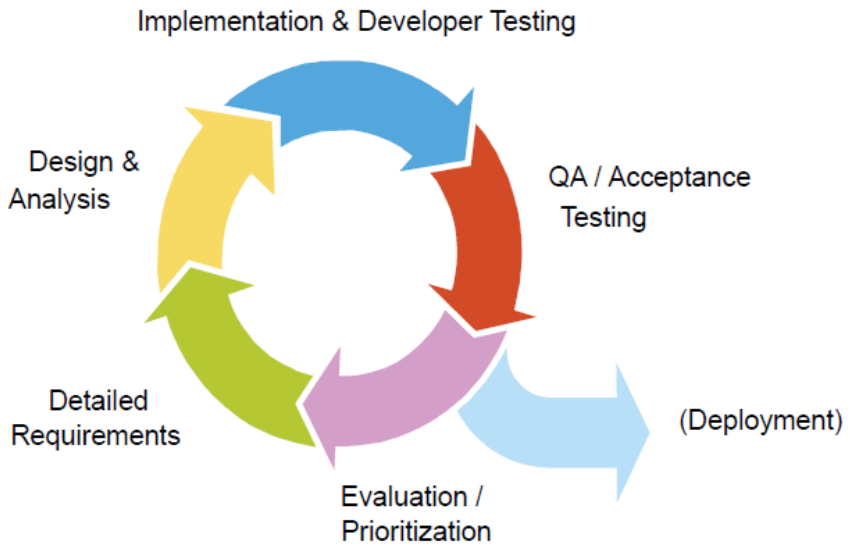
Scrum is based on Lean Principles



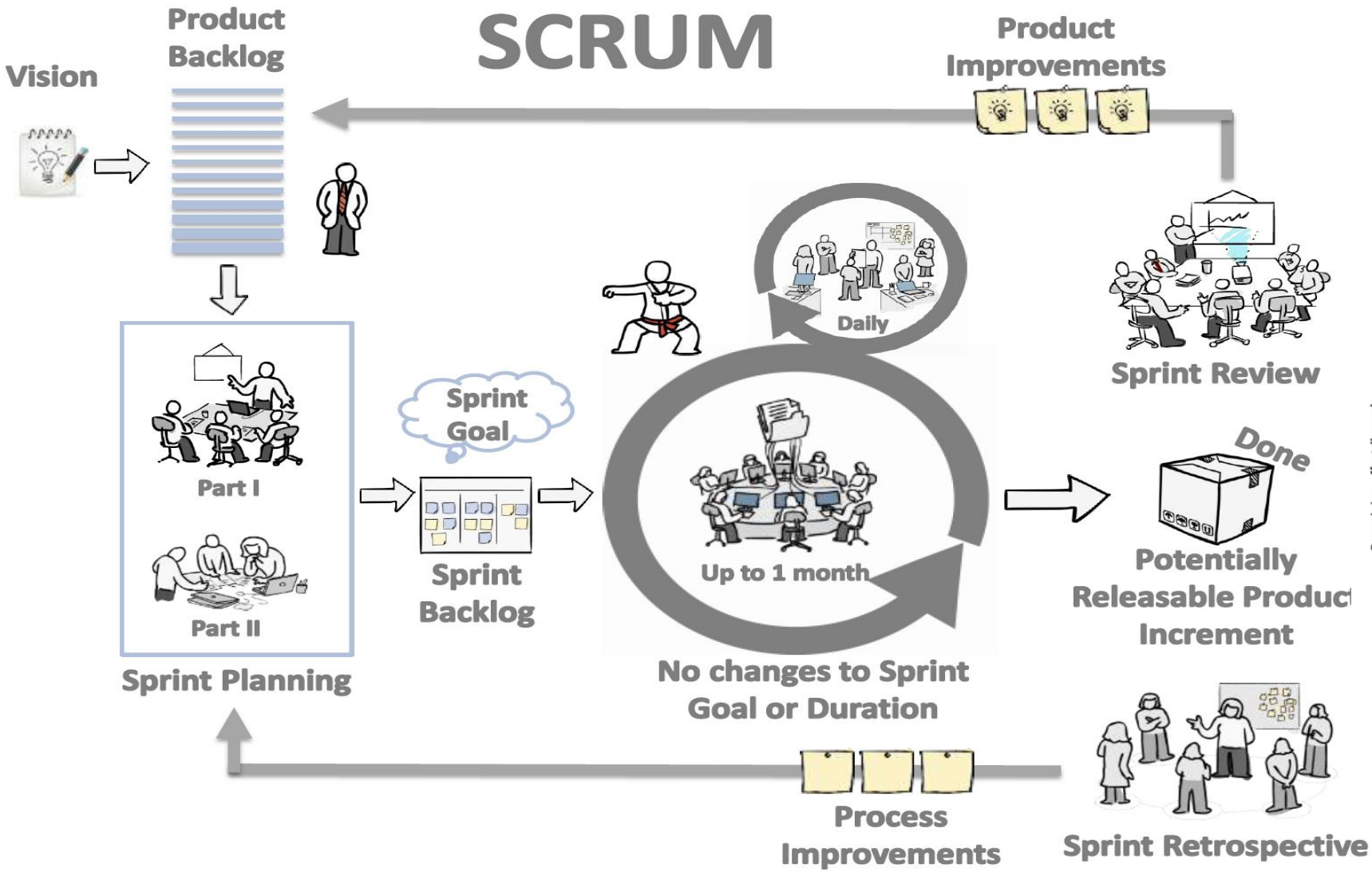
Its Incremental & Iterative

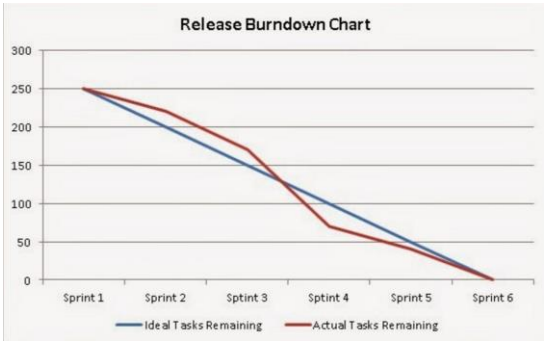


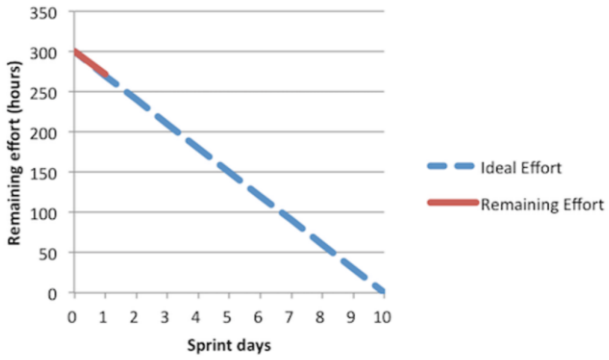
Iteration Detail



Scrum Framework







Scrum Adoption

- ✓ Google
- ✓ Oracle
- ✓ Salesforce.com
- ✓ Toyota
- ✓ BBC
- ✓ Nokia
- ✓ Lockheed Martin
- ✓ Philips
- ✓ Siemens
- ✓ Capital One
- ✓ Time Warner
- ✓ Yahoo

Questions

Scrum Roles

3 Roles in Scrum Team

Product Owner

Scrum Master

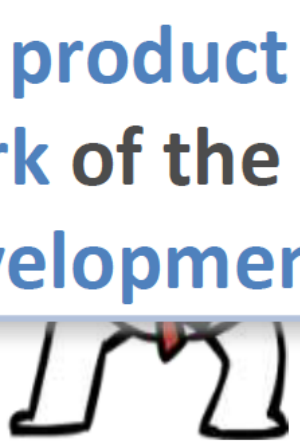
The Development Team

The SCRUM Team

Product Owner

responsible for maximizing the value of the product and the work of the Development Team

Product Owner



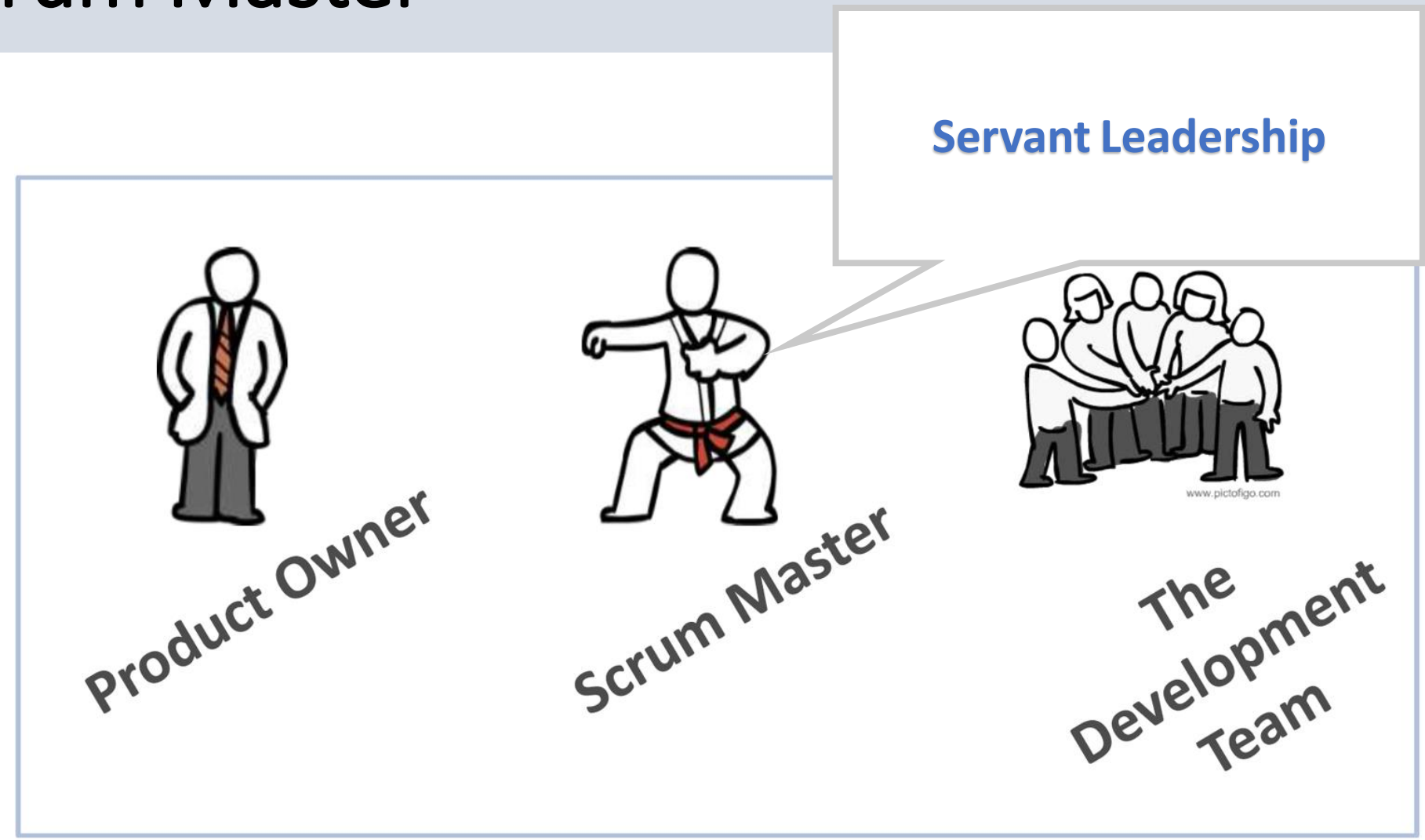
Scrum Master

The Development Team

Product Owner Responsibilities

- Owns Product Backlog
- Defines Product Vision
- Prioritizes features in PB based on risks & dependencies
- Clarifies Requirements
- Stakeholder Consultation & Communication
- Accepts or Rejects Value delivered by SDT
- Defines what goes in Releases and prepares Release Plan
- Demonstrates the Product to Stakeholders

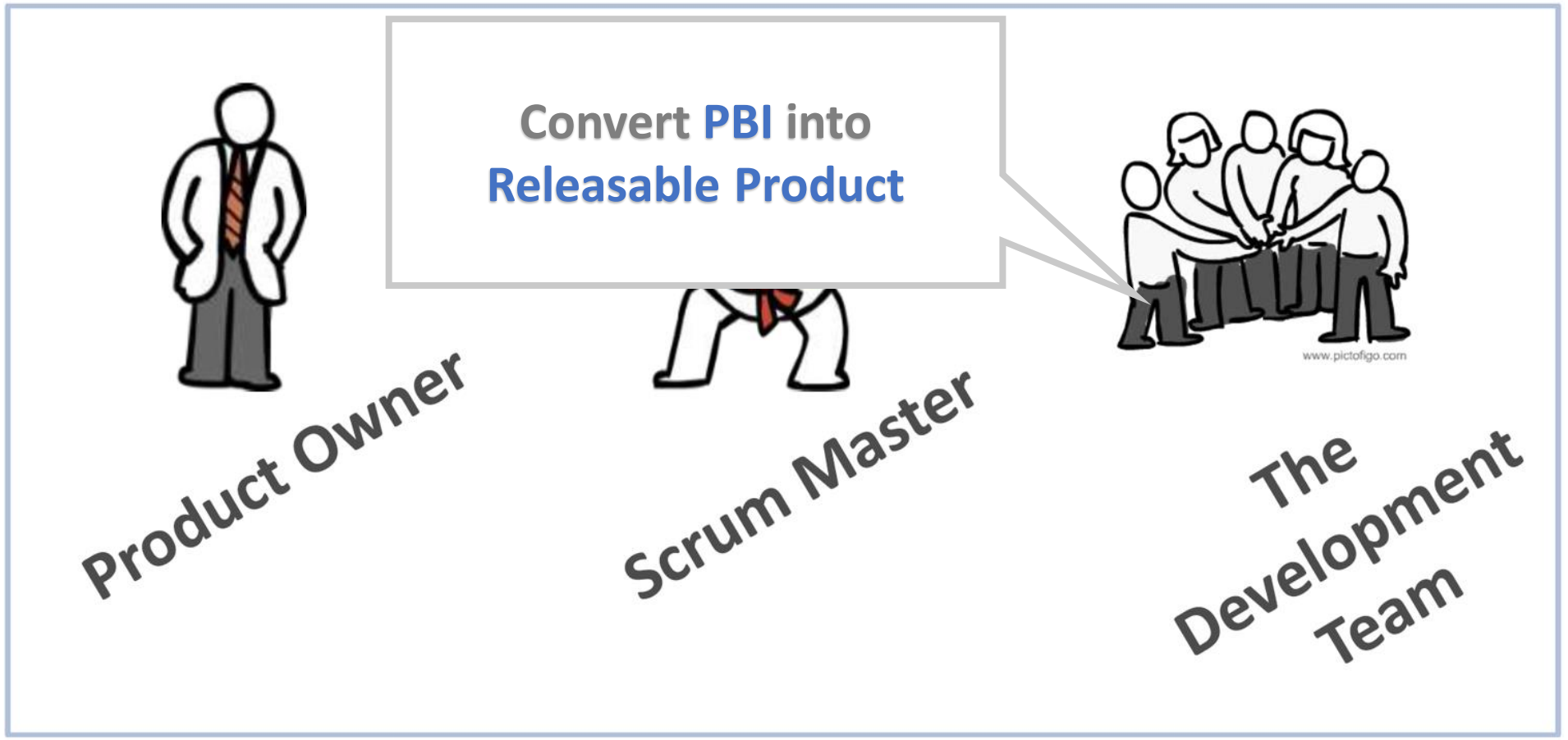
Scrum Master



Scrum Master Responsibilities

- **Towards Dev Team**
  - Train & Mentor team and align to common goals
  - Protects Team from outside interferences
  - Resolves team impediments
  - Scrum Expert
  - Cause changes that would improve productivity of Dev Team
  - Facilitate Ceremonies
  - Understand product vision and help team work towards achieving it.
- **Towards Product Owner**
  - Helping PO do prioritization using various techniques
  - Assist in creating clear and concise product backlog items
  - Understand long term product planning
- **Towards Organization**
  - Be the Change Agent in Agile Adoption
  - Work with other Scrum Masters and improve on shared knowledge
  - Help others in the community to understand and enact scrum

Scrum Development Team (SDT)



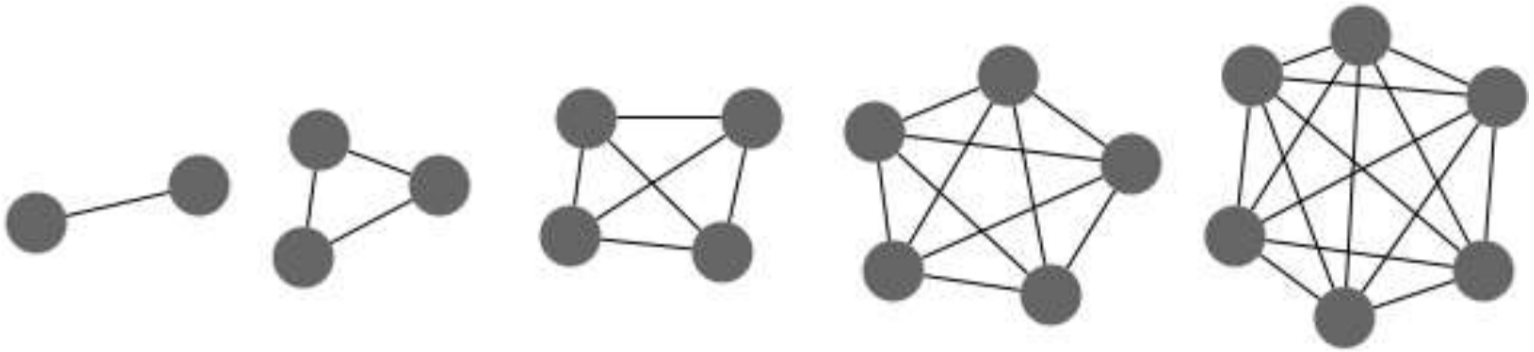
Development Team

The Development Team is :

- A cross-functional team
- Self-organizing
- Empowered and autonomous
- Co-located
- Collectively responsible for converting product backlog into product increment

Development Team Size

The communication channels increase exponentially



1 3 6 10 15

Most commonly used development team size is 7 ± 2

Development Team Responsibilities

- Produce Potentially releasable Product Increments
- Convert Product Backlog into product increment every sprint
- Actively participate in Sprint Retrospectives for process improvements
- Demonstrate work done.
- Create Sprint Backlog
- Commit to Sprint Goals
- Create Just enough design for sprint goal
- Update artifacts on regular basis
- Inspect & Adapt frequently
- Delivery value on quality

- **Towards Product Owner**
  - Helping PO in Grooming Product Backlog
  - Understand Product Vision and internalize it

Exercise: Scrum Roles

- Each Table receives a set of responsibility cards
- Identify which role that card belongs to and stick them to the appropriate role.

Acceptance criteria:

- Responsibilities identified for each role

Questions

Scrum Artifacts

Product Backlog

Product Backlog

An ordered list of everything that might be needed in the product and is the single source of requirements for any changes to be made to the product.

Product Backl

Sprint Backl

Product Im ement

Product Backlog

An ordered and emerging list of user needs plus anything else that is required to fulfill the Product Vision.

Each item in the product is called *Product Backlog Item [PBI]*

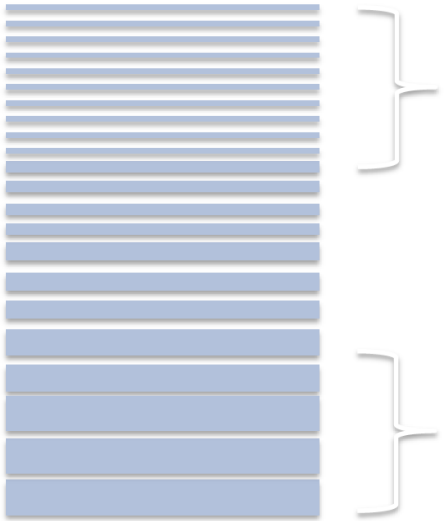
Product Backlog Should Be..

Detailed enough

Emergent

Estimated

Prioritized



Fine grained with details and ready to be pulled in next sprint.

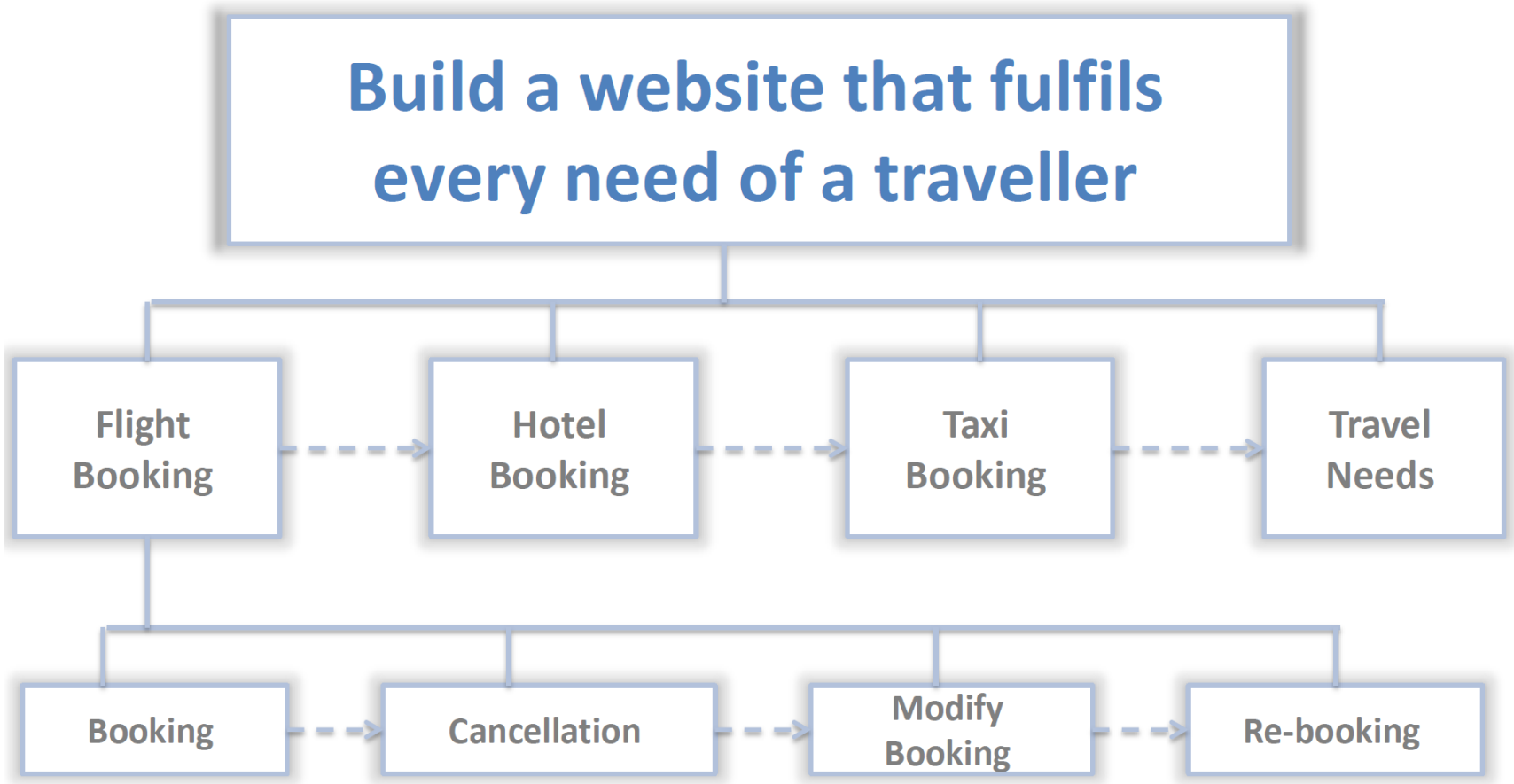
Coarse-grained requirements

Why should it be emergent?

The product backlog starts with a vision statement

Build a website that fulfils every need of a traveller

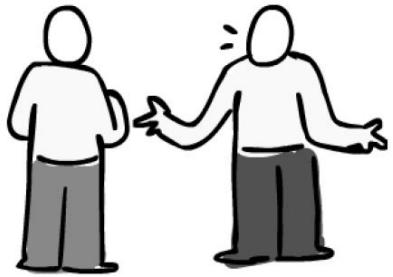
Details get added over a period of time



Inputs to Product Backlog

Ideas as the working software is used

Feedback from the stakeholders and customers



Conversations

Grooming Sessions

Output is PBI

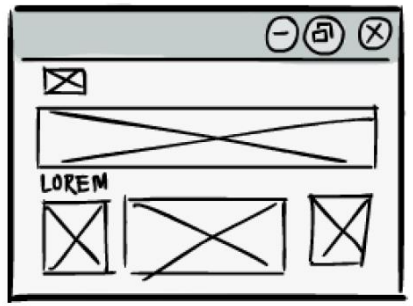
Frequent flyer can re-book the last journey

Each PBI is a

- ❑ User's need
- ❑ Feature description
- ❑ Planning item
- ❑ Token for conversation
- ❑ Mechanism for deferring the conversation

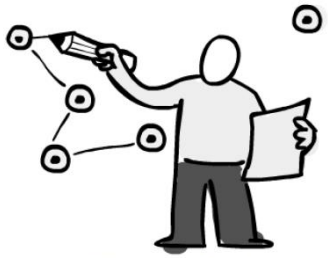
What get added to PBI as they emerge?

Acceptance Criteria



Mockups

Additional Documents



Algorithms

Data points

PBI cut across layers

The architecture evolves secondary to the value created by early regular releases of working software

User Story (PBI)

User stories

- Containers for user or Customer value
- Written using the following template:

As a <user role> I can <activity> so that <business value>

- **User role** is the description of the person doing the action
- **Activity** is what they can do with the system
- **Business value** is why they want to do the activity

As a driver, I can limit the amount of money before I fuel so that I can control my expenditure

As a driver, I can get a receipt after fueling so that I can expense the purchase

As the Finance Department, we can print receipts only for drivers who request them so that we can save on paper

(Roles can be people, devices, or systems)

User Story guidelines — The 3 Cs

Card

Written on a card or in the tool and may annotate with notes

As a spouse,
I want a clean garage so that I can park my car and not trip on my way to
the door

Conversation

The details are in a conversation with the Product Owner

What about the bikes?

Oh yeah, hang the bikes

Confirmation

Acceptance criteria confirm the story correctness

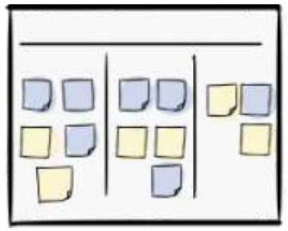
- Tools have been put away
- Items on the floor have been returned to the proper shelf
- Bikes have been hung

Source: 3 Cs coined by Ron Jeffries

Sprint Backlog

Sprint Backlog

Product Backlog



Sprint Backlog

Product Increment

set of Product Backlog items selected for the **Sprint** plus a **plan** for **delivering the product Increment** and **realizing the Sprint Goal**

Sprint Backlog

Feature	Tasks	Estimate	1	2	3	4	5
Custom Actions	Create custom actions tab with the list of actions performed on the job with the status	2					
Custom Actions	Create stdout and stderr viewer in custom actions tab	10					
Custom Actions	Create dynamic Input Dialog with input parameters con with app associated v selected job	4					
Submission Form	Basic data model for j submssion form						
Submission Form	service to get applica definition [M2]						

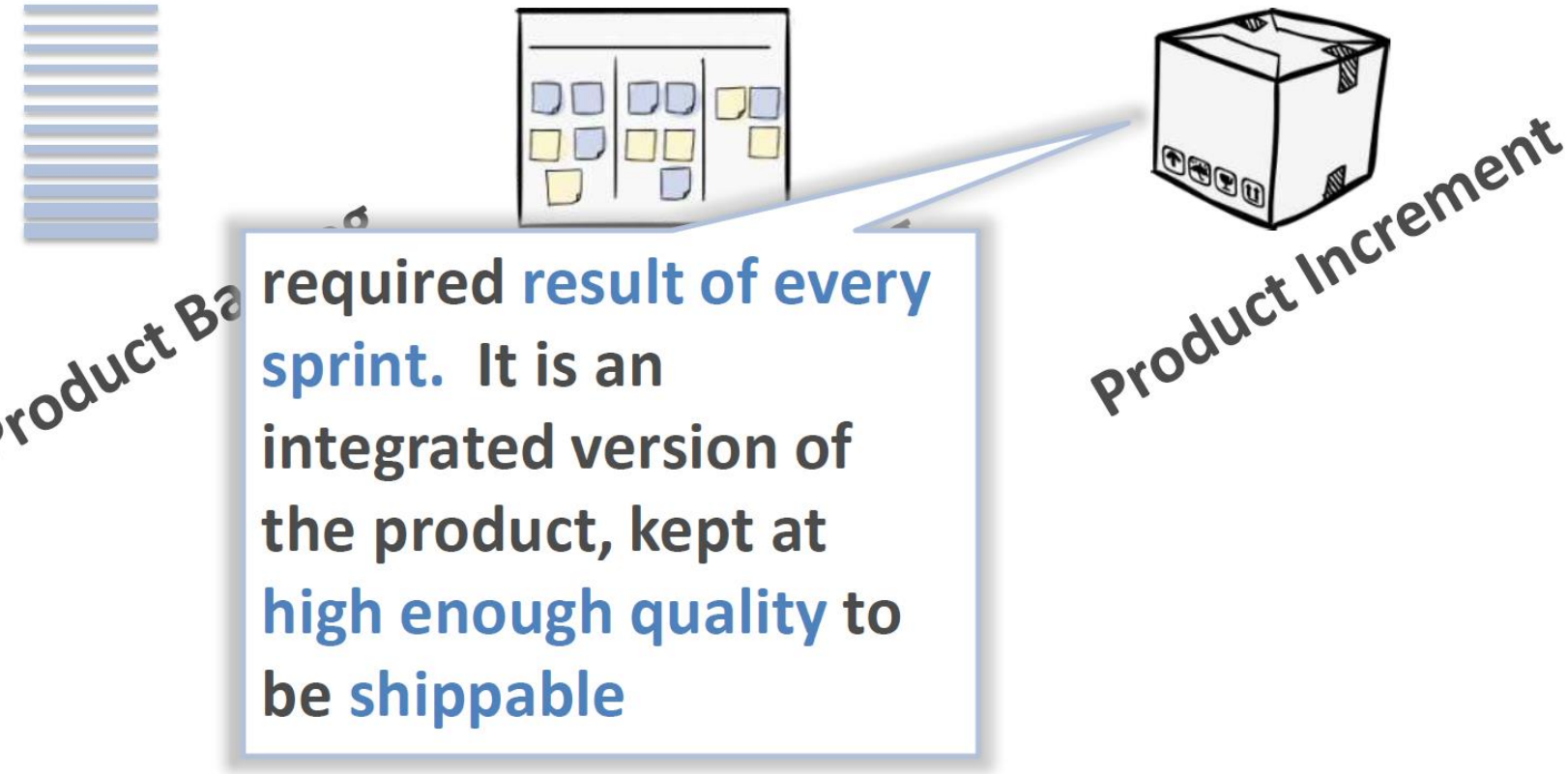
The Product Backlog items selected for this Sprint plus the plan for delivering them is called the Sprint Backlog

Sprint Backlog

- ❑ **Decomposition** of PBI into Tasks
- ❑ **Tasks** required for each PBI are **estimated in Hrs**
- ❑ Team members **sign up for PBI & Task**
- ❑ Any team member can **add/delete/change** sprint backlog

Product Increment

Product Increment



Product Increment

- SDT build product increment which is **working functionality** of the product

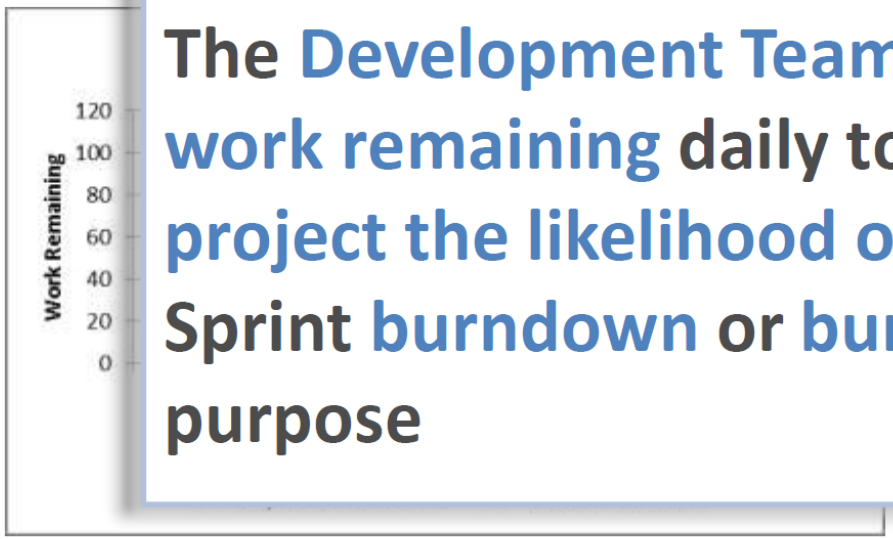
- Each Increment is "**potentially**" shippable

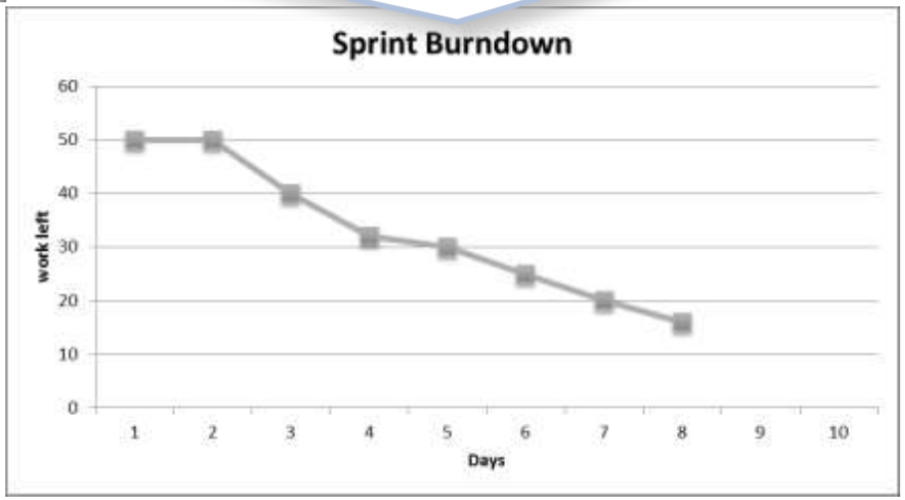
- Each Increment is –
  - Coded as per practices
  - Thoroughly tested
  - Meets **Definition of Done**

Burndown Charts

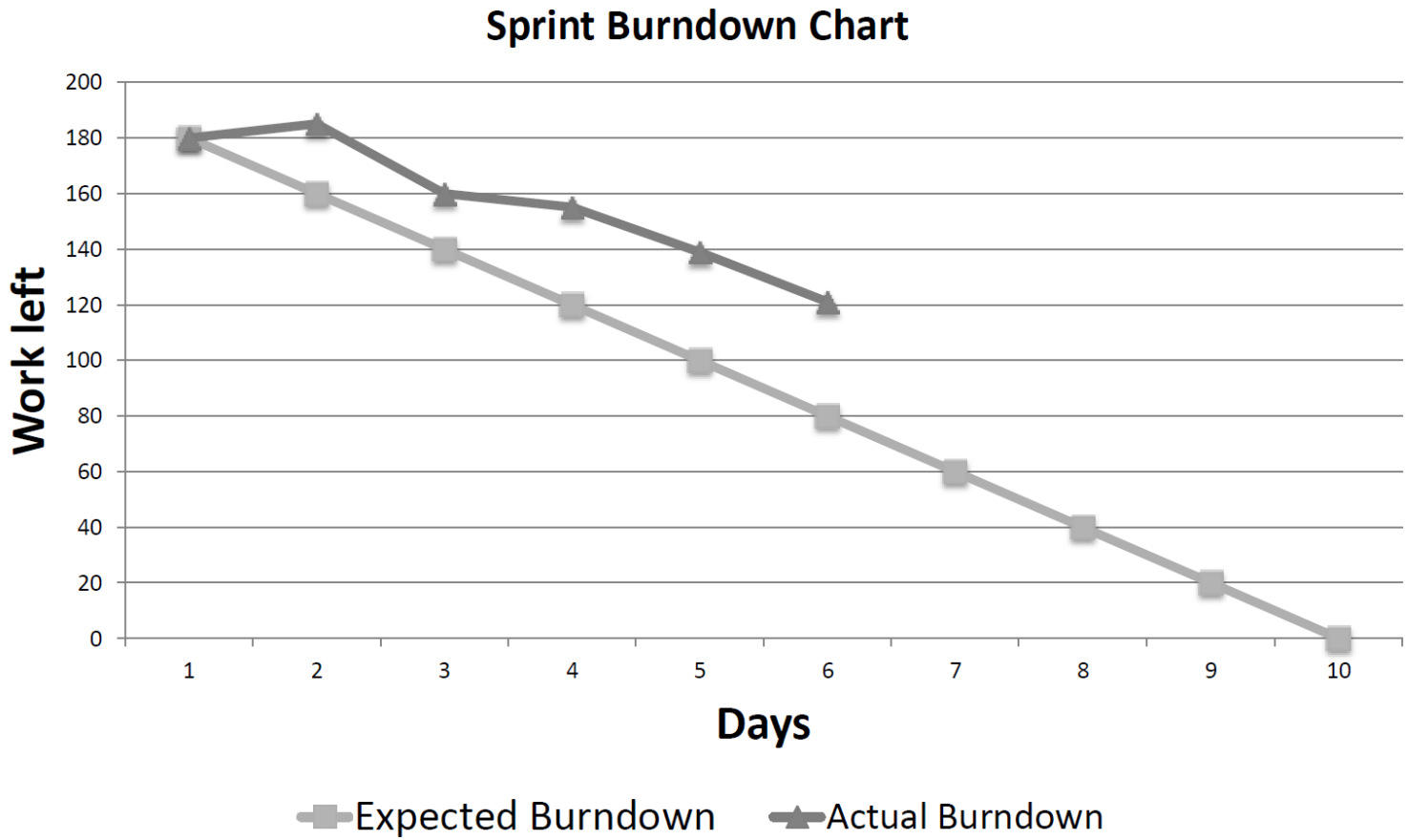
Sprint Burndown Chart

The Development Team compares the amount of work remaining daily to the time they have to project the likelihood of achieving the Sprint Goal. Sprint burndown or burnup can be used for this purpose

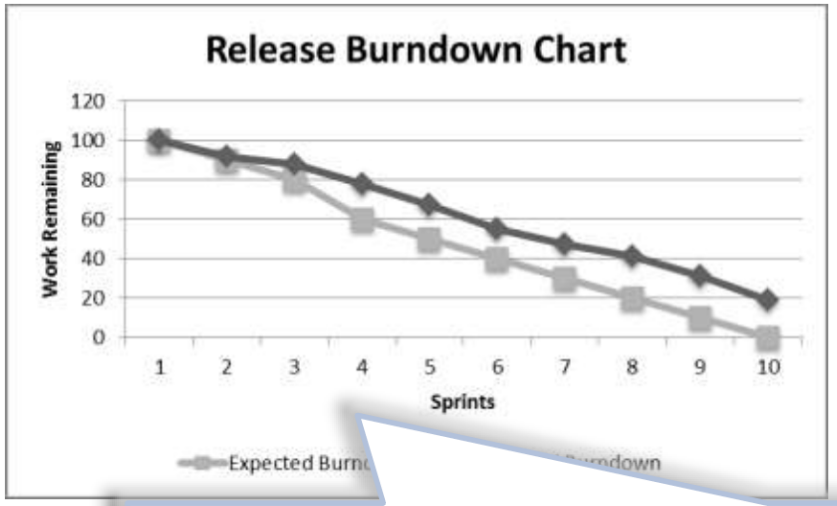




Burndown chart based on tasks



Release Burndown Chart



The Product Owner looks at the amount of work remaining at the end of every Sprint and the trend to assess progress toward completing projected work by the desired time for the goal. Release burndown or burnup can be used for this purpose

Questions

Scrum Events

Sprint Planning

Sprint Planning Meeting

Sprint Planning

the team meets with the product owner to choose a set of work to deliver during a sprint

Sprint Review

Sprint Retrospective

Sprint Planning Overview

DEEP Product Backlog

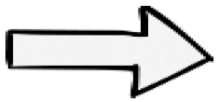
Latest Product Increment

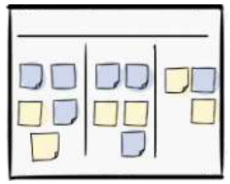
Projected Team Capacity

Past Performance

Part I: "What are we doing in this sprint"

Part II: "How are we going to achieve the Sprint Goal"





Sprint Backlog

Sprint Planning : 8 hours for one month sprint

Part 1 – The “What”

1. The product owner presents the ordered PBIs to the Development Team.
2. The Development team Pulls and discuss PBIs, ask clarifying questions and understand the acceptance criteria. Select the PBI for Sprint.
3. Continue 2 until Sprint backlog is full
4. The Scrum team crafts the sprint goal.

Part 2: “The How”

Discuss how to achieve the Sprint goal and deliver the product increment

1. Discuss the rough architecture
2. Make design decisions
3. Identify tasks the team needs to do

The team may renegotiate the Sprint Backlog items with Product Owner

Rough Design (EDUF)

- ❑ Identify components and interfaces
- ❑ Identify dependencies
- ❑ Identify the external integration points
- ❑ Identify Data to be used and data sources
- ❑ Discuss Architectural patterns to be applied
- ❑ Discuss Testing strategy

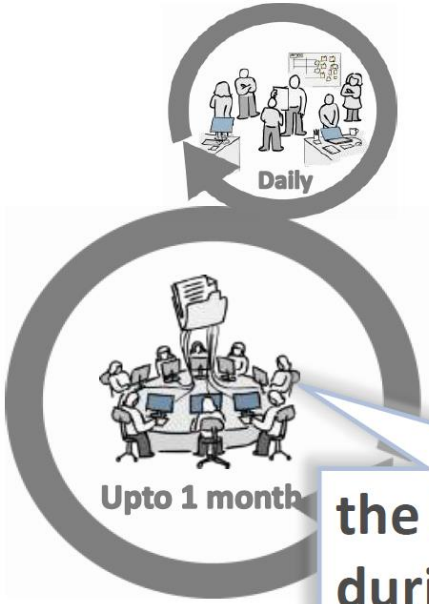
Commit to Sprint Goal

- An objective that will be met within the Sprint through the implementation of the Product Backlog
- It provides a guidance to the Development team on why its building the increment
- may be a milestone in the larger purpose of the product roadmap

Sprint

Sprint

Sprint Planning



Sprint Review

the heart of the scrum, during which a "Done", useable, and potentially releasable product Increment is created.

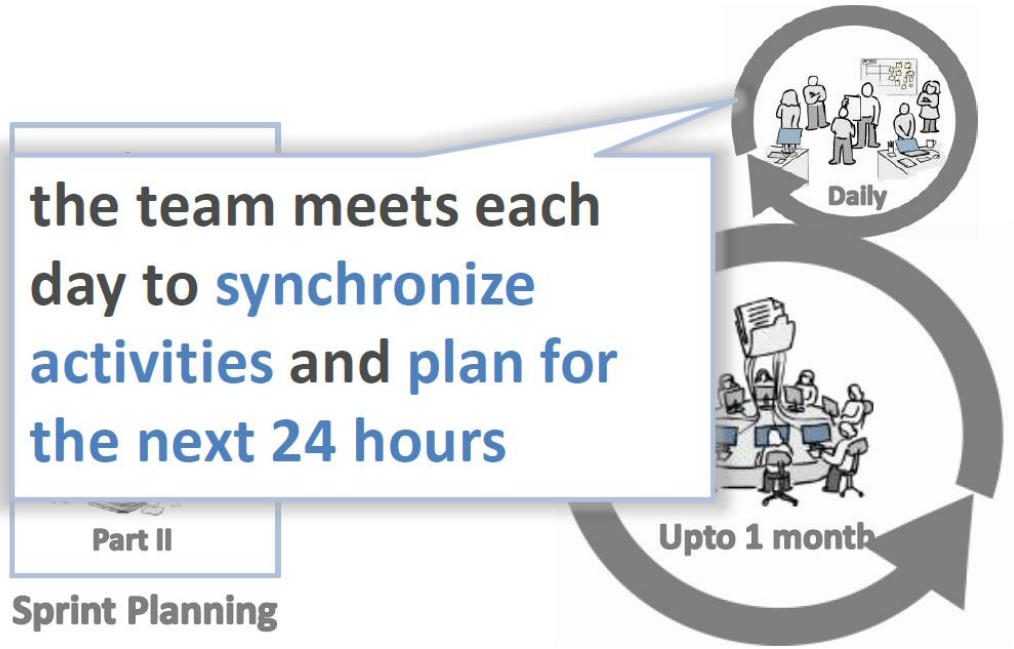
2 to 4 Weeks

Sprint

- It's a timebox of one month or less and is a Deming cycle
- Its an iteration in which the team produces a shippable product increment
- There will be no change in the Sprint duration or Sprint Goal

Daily Standups

Daily Stand-up Meeting

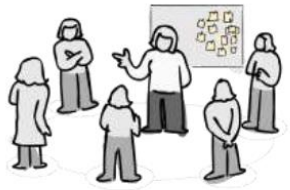



Sprint Review

Q1 - What did I do Yesterday?

Q2 – What will be done today

Q3 – What are the issues / hindrances still bother?


Sprint Retrospective

15 Minutes

Daily Stand-up Meeting

Duration: 15 minutes

Attendees:

- Scrum Master (Facilitates)
- Development team
- Product Owner

Purpose:

For Development team to synchronize the activities and create plan for next 24 hours

Inspect and adapt

DSMs shouldn’t be ..

Sprint Review

Sprint Review

Sprint Planning

the scrum team demonstrates to the stakeholders what it has completed during the sprint

Upto 1 month

Sprint Review

Sprint Retrospective

Sprint Review Meeting

Duration: 4 Hour for One-month Sprint

Attendees:

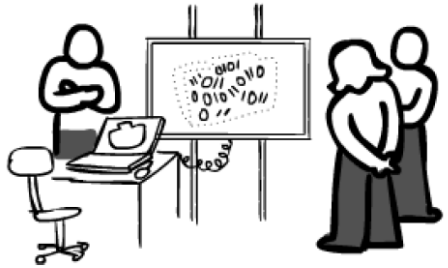
- Scrum Master (facilitates)
- Product Owner,
- Development team
- Stakeholders

Purpose:

Inspect the product increment and adapt the product backlog

Sprint Review

Focus on Work done. PO identifies the work done and what has not been done.



Development team discusses how the last Sprint went and demonstrate the working product increment.

PO discusses Product Backlog as it stands and project the likely completion dates based on the progress to date.

Sprint Retrospective

Sprint Retrospective

Sprint Plann

Sprint Review

the scrum team inspects itself and create a plan for improvements to be enacted during next sprint.

Sprint Retrospective

Sprint Retrospective

Duration: 3 hours for one month sprint

Attendees:

- Scrum Master (facilitates)
- Product Owner,
- Development team

Purpose:

- ❑ Inspect how last sprint went with regards to people, relationship, process, and tools
- ❑ Identify and order the major items that went well and potential improvements

Steps in Retrospectives

I. Setting the Stage

II. Gather Data

III. Generate Insights

IV. Decide What to Do

V. Close the Retrospective

Gather Data

- ❑ **Review available data:** Burndown charts, Velocity Trend etc..
- ❑ **Create a common view**
- ❑ **Discuss what happened during last Sprint**
  - Mood Timeline
  - Plus/Delta
  - Four Square
  - Brainstorming etc.

Retrospective Focus

Focus on		Focus off
Inquiry	rather than	Processes and tools
Dialogue	rather than	Debate
Conversation	rather than	Argument
Understanding	rather than	Defending

Generate Insights

- Generate insights on what could have caused the plus/delta
  - 5 whys
  - Fishbone diagram
  - Brainstorming + Silent Grouping

Decide what to do

- ❑ Discuss solutions and SMART goals
- ❑ Create backlog of action items
- ❑ Prioritize the backlog based on:
  - "What are we best positioned to try next?"
  - "What do we really want to try (or sustain)?"
- ❑ Make sure each backlog item has volunteer owner.

Closure

- ❑ Summarize the results
- ❑ Agree on action items
- ❑ Appreciate the results achieved from earlier retrospectives

Events are Time-boxed

Every event in Scrum including the Sprint is Timeboxed.

This helps the team in maintaining the sustainable pace.

Sprint = Max of 30 days (Ideal of 2 weeks)

Sprint Planning = 8 Hrs for 1month Sprint

Daily Stand-up = 15min

Sprint Review = 4 Hrs for 1 month Sprint

Sprint Retro = 3 Hrs for 1month Sprint

Backlog Grooming

Product Backlog Grooming

- It is the act of adding detail, estimates, and order to PBI

- On going process of collaboration between PO and the Development Team

- It's a part-time activity during the sprint and how and when is decided by the Scrum Team

Consideration for Grooming

Customer feedback

Interaction with the stakeholders

Conversations

Grooming Meeting

What happens in Grooming

- Enrich PBI with new details
- Split large PBIs
- Write new PBIs
- Estimate PBIs.
- Re-prioritize PBIs as needed.

Acceptance Criteria

- Acceptance criteria provide the details of the Story from a testing point of view
- Acceptance criteria are created by the Team and the PO

As a driver, I can limit the amount of money before I fuel so that I can control my expenditure.

Acceptance criteria

1. The fueling process stops automatically on the exact value
2. I can stop fueling before the limit has been reached and will only be charged for the amount fueled

As a driver, I can get a receipt after fueling so that I can expense the purchase.

Acceptance criteria

1. Receipt includes: Amount fueled, Amount Paid, Tax, Vehicle number, Date, Time

Questions

Agile Estimation & Planning

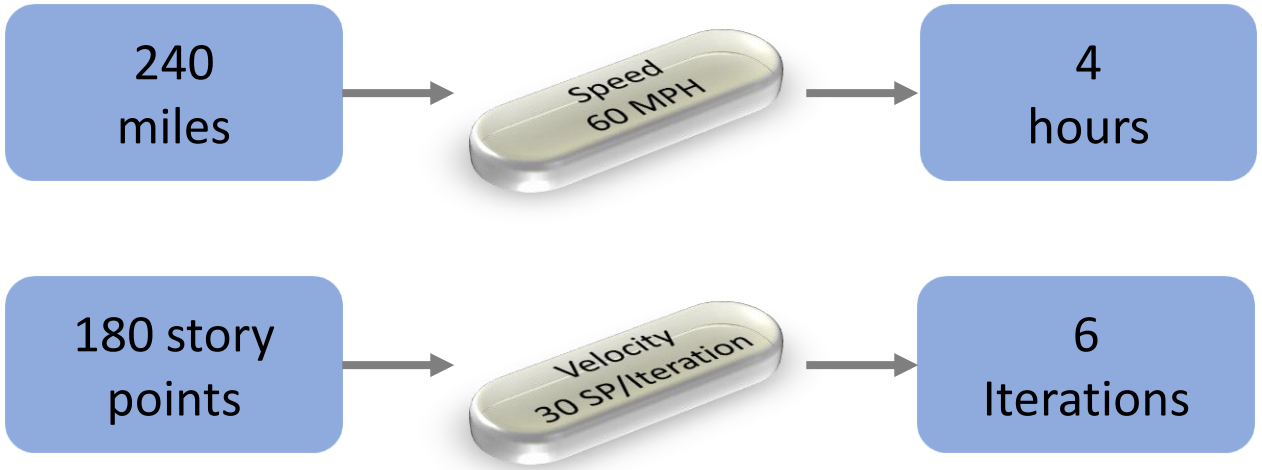
Agile Estimation

Scrum doesn't prescribe any estimation techniques. Instead it promotes empiricism.

© Scott Adams, Inc./Dist. by UFS, Inc.

Using size to estimate duration

Examples



Establish velocity by looking at the average output of the last Iterations.

So How do I estimate?

Use whatever technique your team is comfortable with. Scrum provides more opportunities to inspect and adapt

Planning Poker

- The whole scrum team participates but estimates are given by the development team
- Everyone estimates overall size of the item (not just part of the work)

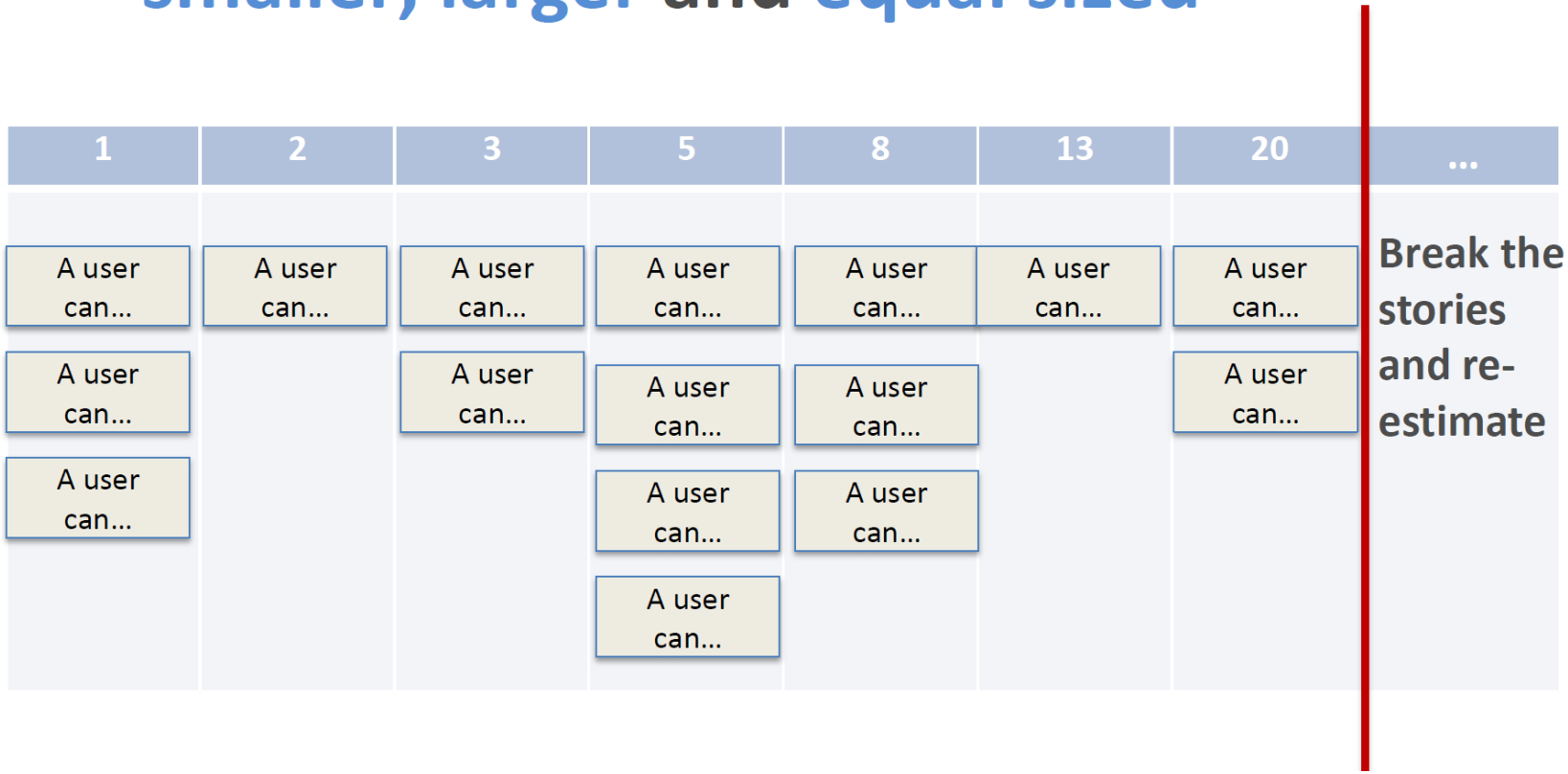
* Planning Poker is not part of core scrum but a practice widely used by Scrum teams.

How to go about Planning Poker

- ❑ Create a common understanding of the User Story.
- ❑ Choose a reference card.
- ❑ Estimate size in relation to reference – but don't tell yet
- ❑ Show your cards at the same time
- ❑ Discuss differences
- ❑ Repeat estimation until consensus
- ❑ Estimated user story becomes new reference

Triangulation

Find three stories as reference: smaller, larger and equal sized



Lets Play

Giraffe

Horse

Crocodile

Chicken

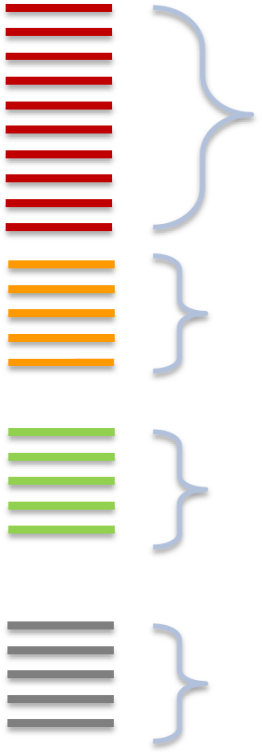
Gorilla

Hyena

Elephant

Estimate using Planning Poker

MuSCoW Prioritization (DSDM)



Must Have: Minimum Usable SubseT

Should Have: Expected in this release

Could Have: Possibly have them in this release

Won't Have: Out of Scope for this timeframe

- Requirements that cannot be de-scoped without causing the project to fail
- Requirements that can be de-scoped as a last resort to keep the project on track
- Requirements that can be de-scoped without causing significant problems

Exercise: Create Sprint Backlog for 2 Sprints

CAR POOL WEBSITE

F1: User can **SELECT** a RIDE

F2: User and **BOOK** a RIDE

F3: User can **POST** a RIDE

F4: User can **CANCEL** a RIDE

Acceptance criteria:

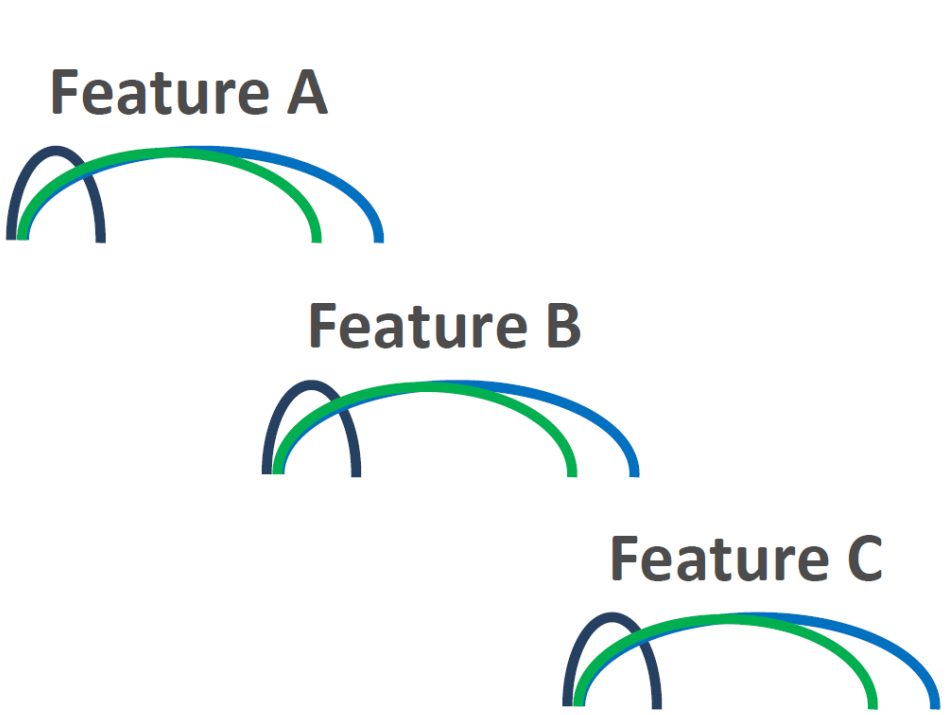
- **All User Stories Prioritized & Estimated**
- **Create Product Backlog for 1 or 2 Feature**

Questions

Scrum in Action

Foster Agile Practices

SDLC Activities during Sprint



Analyze and Design

Implement

Test

..but continuously and all the time (not as a mini-waterfall)

Sprint

Regularly update Artifacts

- Ensure that you update the sprint backlog and Sprint Burndown as you want to know where you stand before starting the DSM.

- The focus of DSM should only be “Are we there yet?” and “What we need to get there?”. Discuss the details of “How” outside the DSM.

Define the DoD

Definition of Done[DoD]

- A shared understanding within the Scrum Team on when a PBI is considered as “Done”.
- A checklist of valuable activities required to produce releasable product increment.

- ✓ Fully implemented
- ✓ Tested
- ✓ All Acceptance Criteria fulfilled.
- ✓ No known issues etc.

Adverse effects of undone work

The technical debt increases and pulls down your velocity.

You will end up with hardening sprints.

Sprint / Iteration Metrics

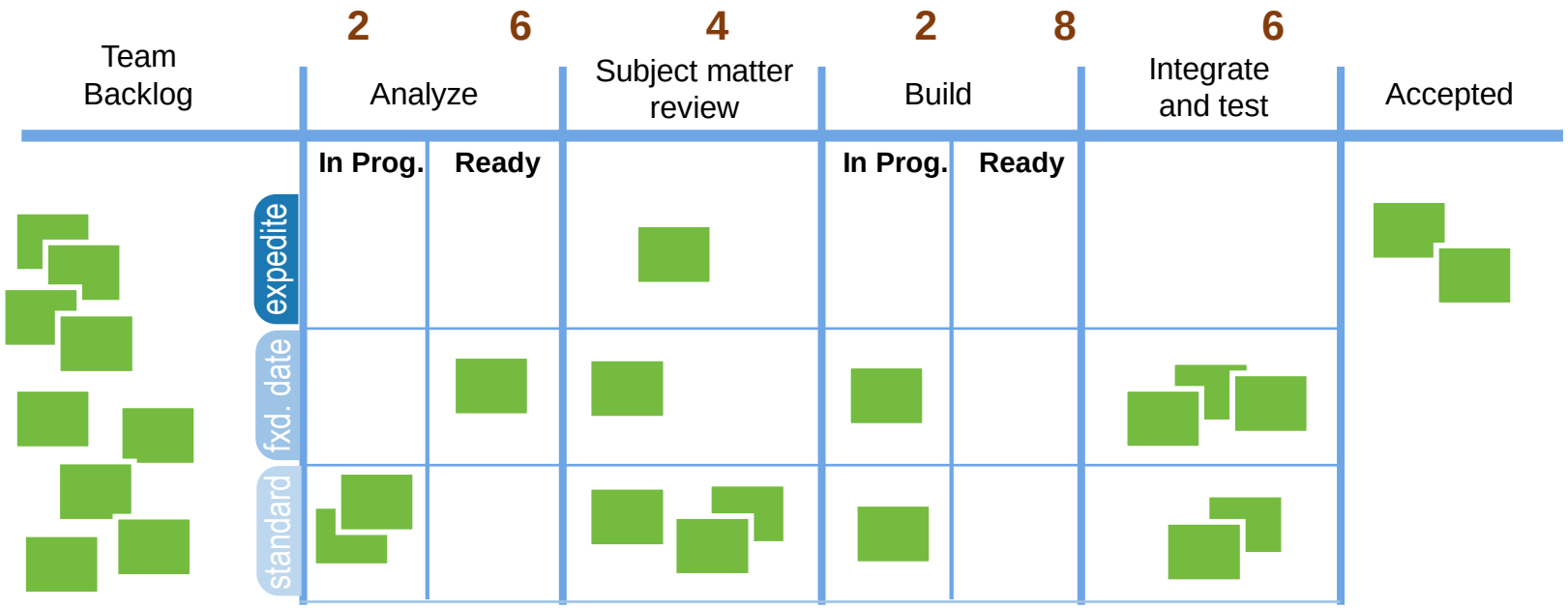
Functionality	Iteration 1	Iteration 2
# Stories (loaded at beginning of Iteration)		
# accepted Stories (defined, built, tested, and accepted)		
% accepted		
# not accepted (not achieved within the Iteration)		
# pushed to next Iteration (rescheduled in next Iteration)		
# not accepted: deferred to later date		
# not accepted: deleted from backlog		
# added (during Iteration; should typically be 0)		

Quality and test automation
% SC with test available/test automated
Defect count at start of Iteration
Defect count at end of Iteration
new test cases
new test cases automated
new manual test cases
Total automated tests
Total manual tests
% tests automated
Unit test coverage percentage

Build your Kanban board

Understanding Kanban

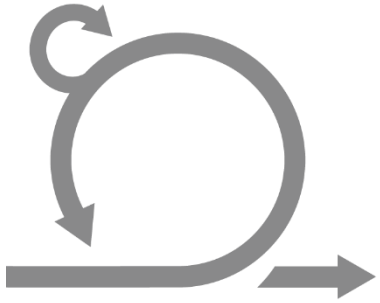
Kanban is a pull system that visualizes work flow and uses WIP limits at different work flow steps to facilitate the flow.

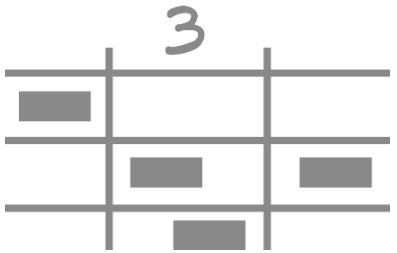


- Kanban relies on empirical data for continuous improvement
- Helps avoid overload by pulling work only when there is excess capacity at a step
- Helps the team reflect their unique work flow
- Makes process policies and decisions visible to the entire team

Applicability

Kanban perfectly extends Scrum by providing granular pull mechanisms that drive more effective Iteration execution.



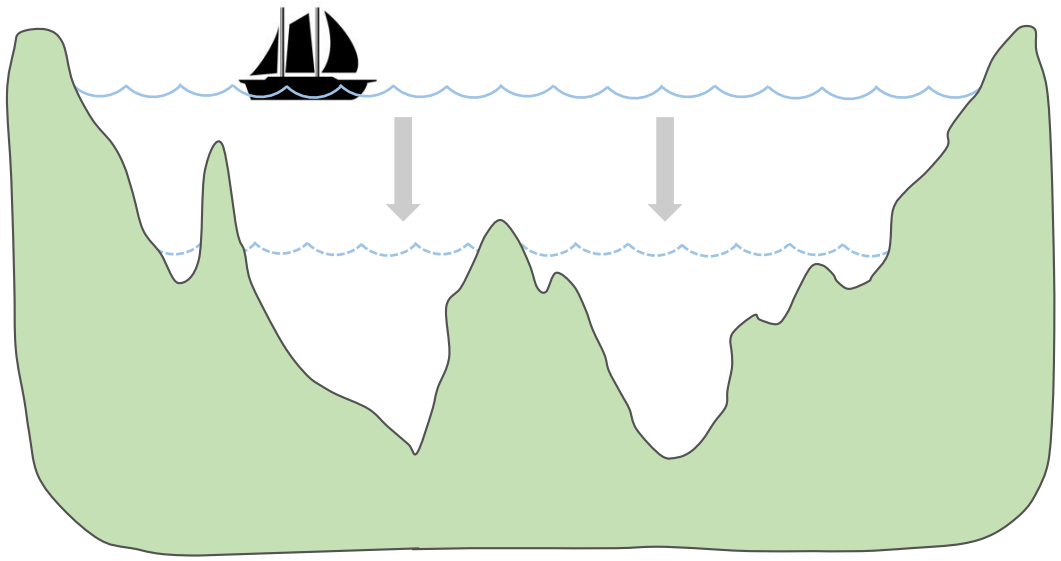


- Kanban “connects” capacity-based planning in Scrum with a throughput-based approach
- It helps improve Iteration outcomes
- It allows better visibility into the progress of work based on the team-specific work flow

LIMITING WIP

Work in process (WIP) and flow

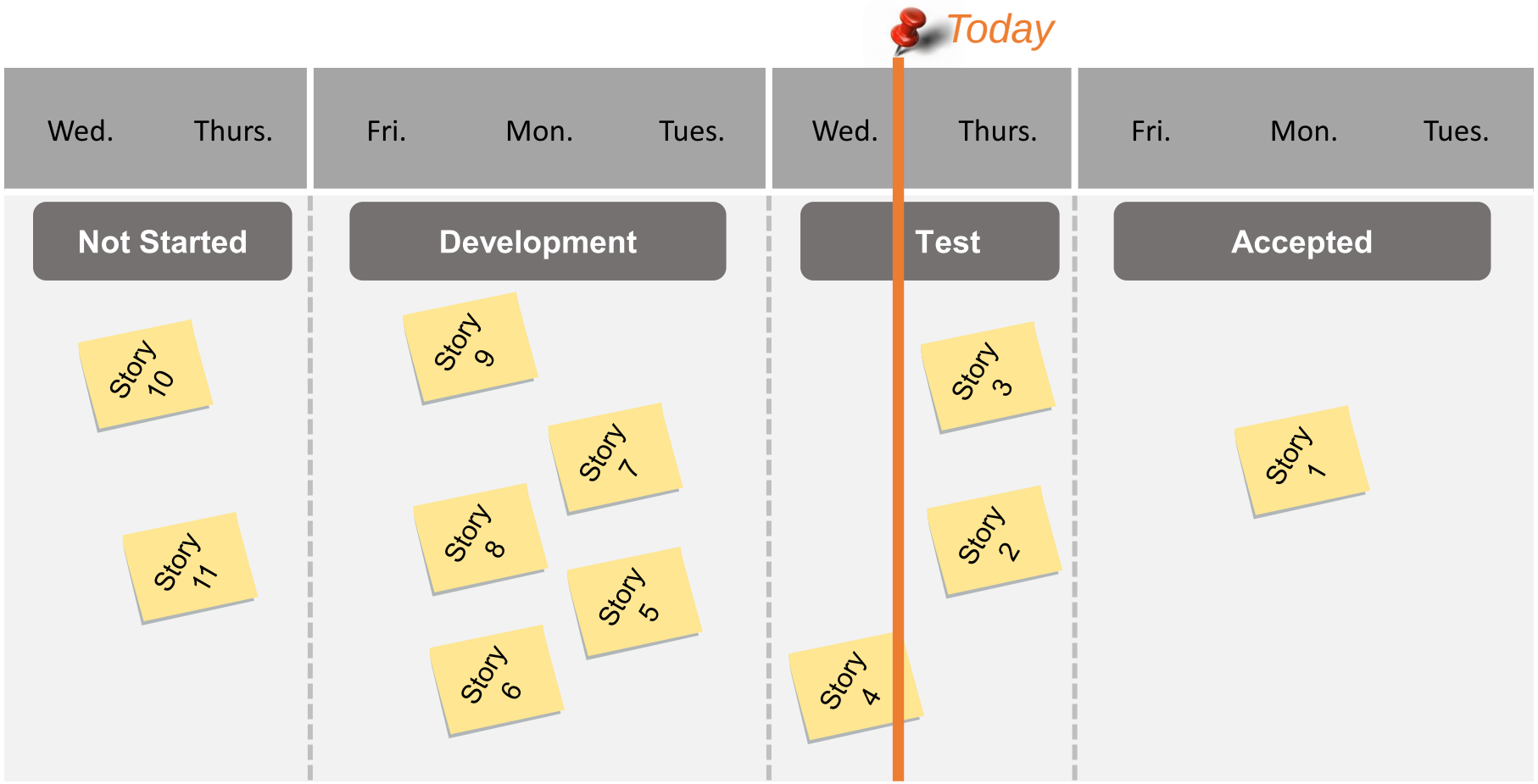
When there's too much WIP, there's no visibility into the bottlenecks and the system is usually highly inefficient.



At the surface however, it all looks smooth and promising.

Visualize and limit WIP

One team's Big Visible Information Radiator (BVIR)



How is this team doing? How do you know?

Build quality in

Built-in Quality

"You can't scale crappy code" (or hardware, or anything else).

Building quality in:

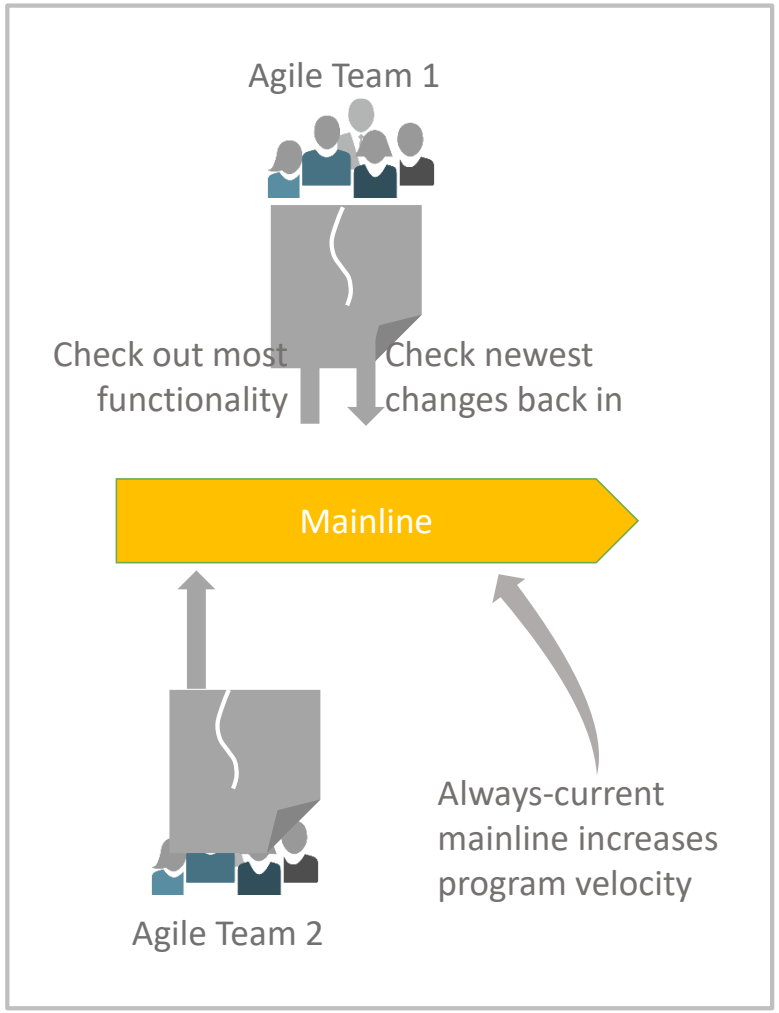
- Ensures that every increment of the Solution reflects quality standards
- Is required for high, sustainable development velocity
- Software quality practices (most inspired by XP) include Continuous Integration, Test-First, refactoring, pair work, collective ownership, and more

Built-in Quality

Continuous system integration

Teams continuously integrate assets (leaving as little as possible to the System Team).

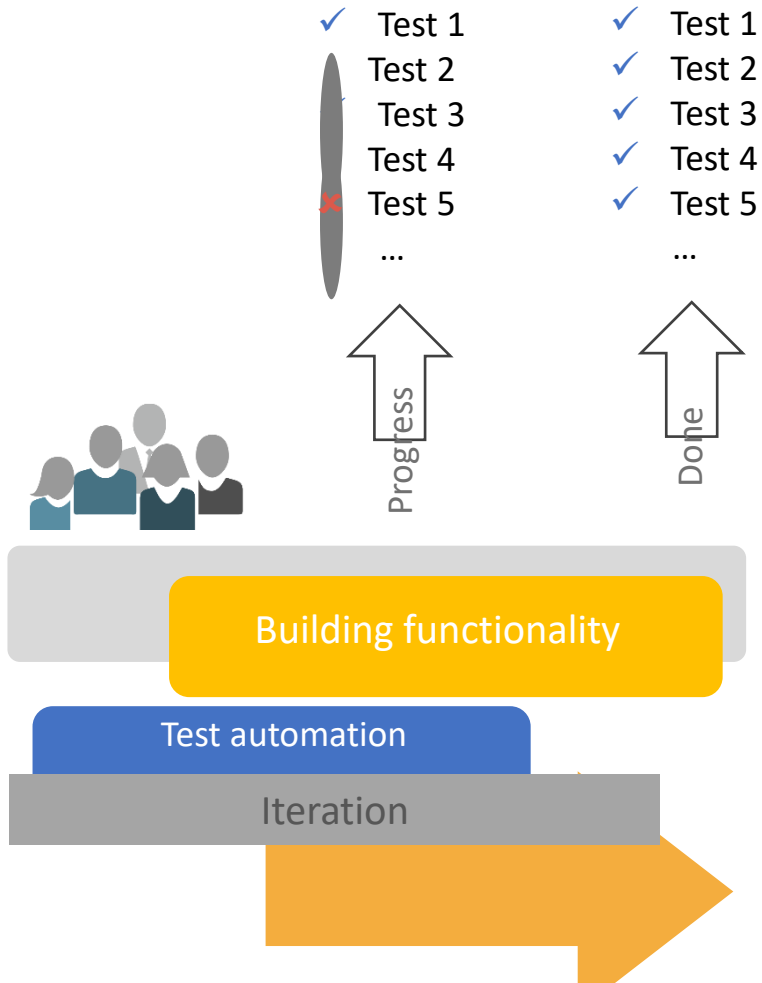
- Integrate every vertical slice of a User Story
- Avoid physical branching for software
- Frequently integrate hardware branches
- Use development by intention in case of inter-team dependencies
    - Define interfaces and integrate first, then add functionality



Test first: Test Driven Development

Otherwise velocity is bottlenecked, quality is speculative, and scaling is impossible.

- Automated tests are implemented in the same Iteration as the functionality
- The team that builds functionality also automates the tests
- Create an isolated automated test environment
- Actively maintain test data under version control
- Passing vs. not-yet-passing and broken automated tests are the *real* Iteration progress indicator



Explore anti-patterns

Many anti-patterns can be traced to the PO role

Underperforming in the Product Owner role can lead to dysfunction on the team

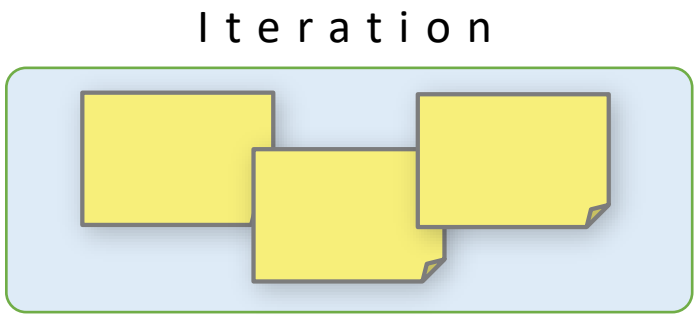
Key responsibilities of the Product Owner:

- Facilitate Team Backlog refinement
- Prepare for and participate in Iteration Planning
- Elaborate Stories and Enablers “just in time”
- Address team questions; be the “voice of the Customer”
- Accept Stories
- Participate in Team Demo and retrospective
- Coordinate with other Product Owners to manage dependencies

Big Stories are a frequent source of anti-patterns

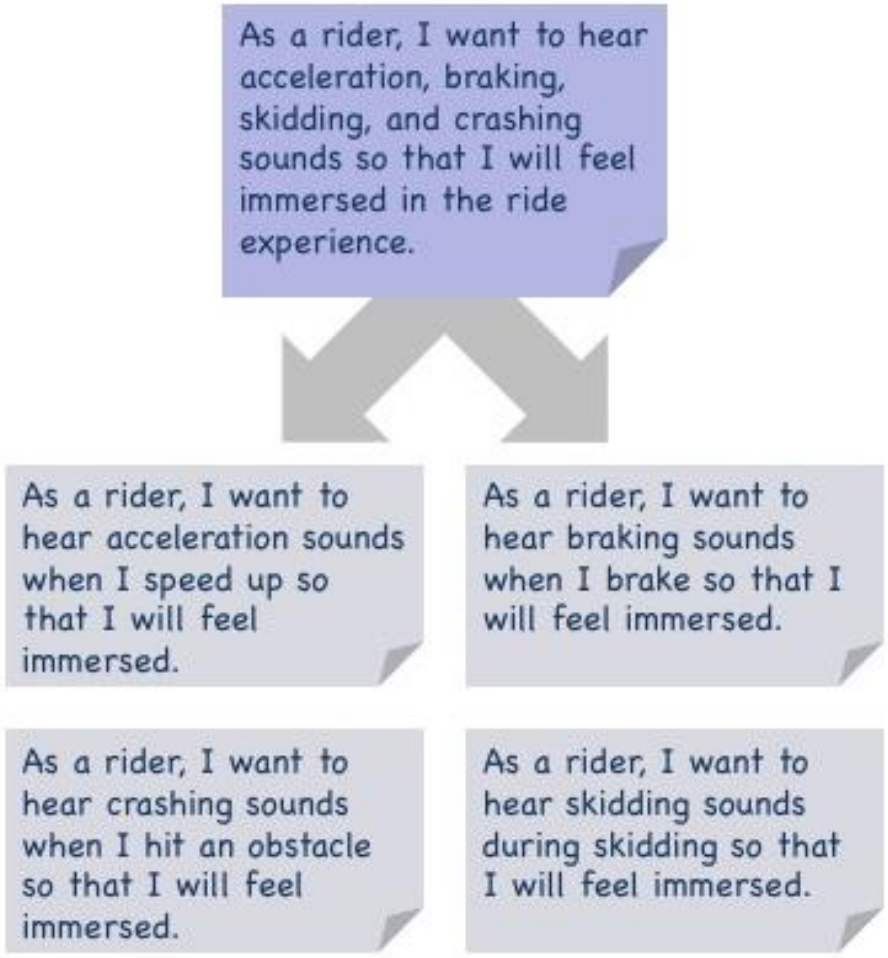
A team that can't iterate isn't able to inspect and adapt.

- Big Stories do not support team iteration
- Smaller Stories allow for faster, more reliable implementation
- Splitting bigger Stories into smaller ones is an essential skill for every Agile Team



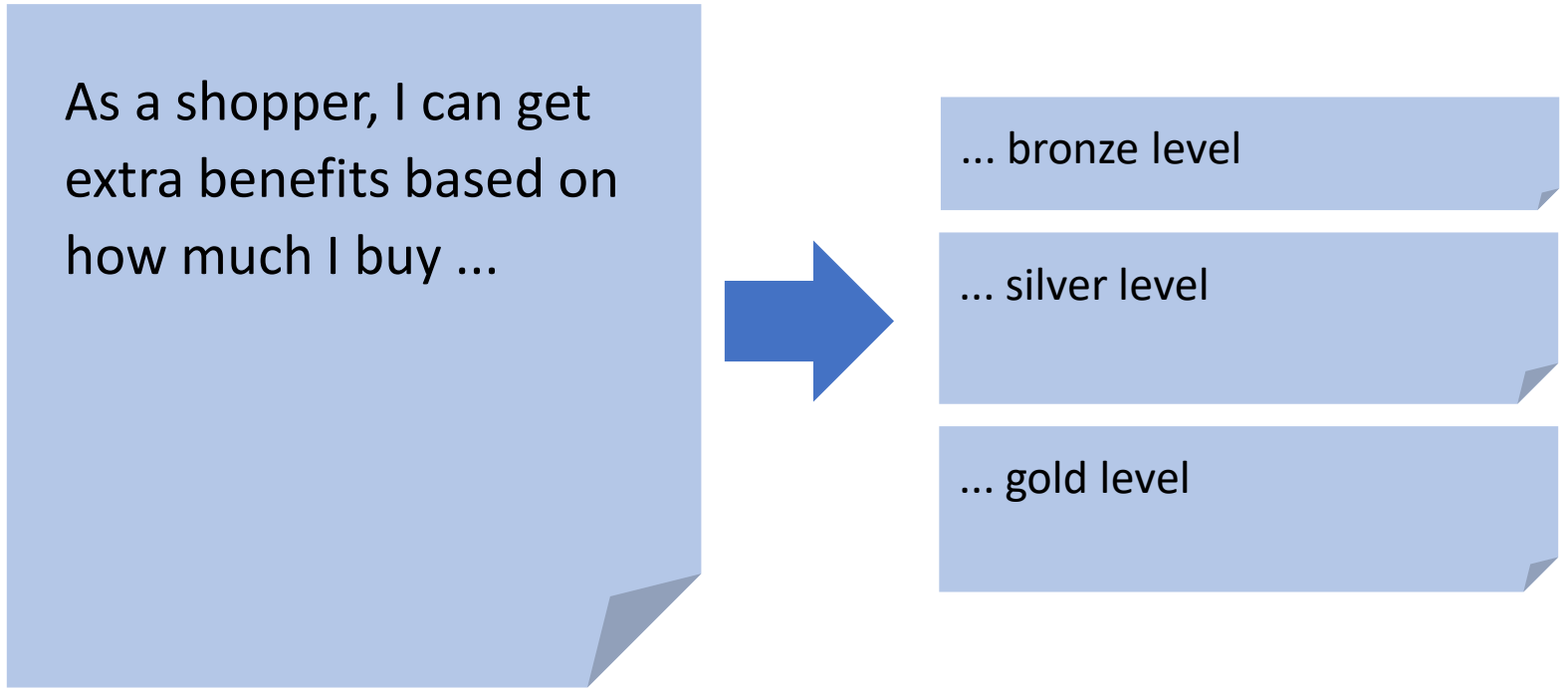
Ways to split a Story

- By business rule variations
- By use case scenario
- Simple/complex



Split by business rule variations

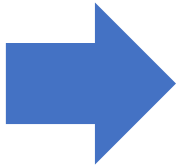
Business rule variations often provide a straightforward splitting scheme.



Split by use case scenarios

If use cases are used to represent complex interaction, the Story can be split via the individual scenarios.

As a user, I can enroll in the energy savings program through a retail distributor ...

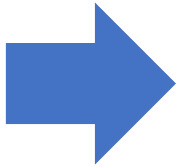


- Use Case/Story #1 (happy path): Notify utility that consumer has equipment
- Use Case/Story #2: Utility provisions equipment and data, notifies consumer
- Use Case/Story #3 (alternate scenario): Handle data validation errors

Split by simple/complex

Simplify! What's the simplest version that can possibly work?

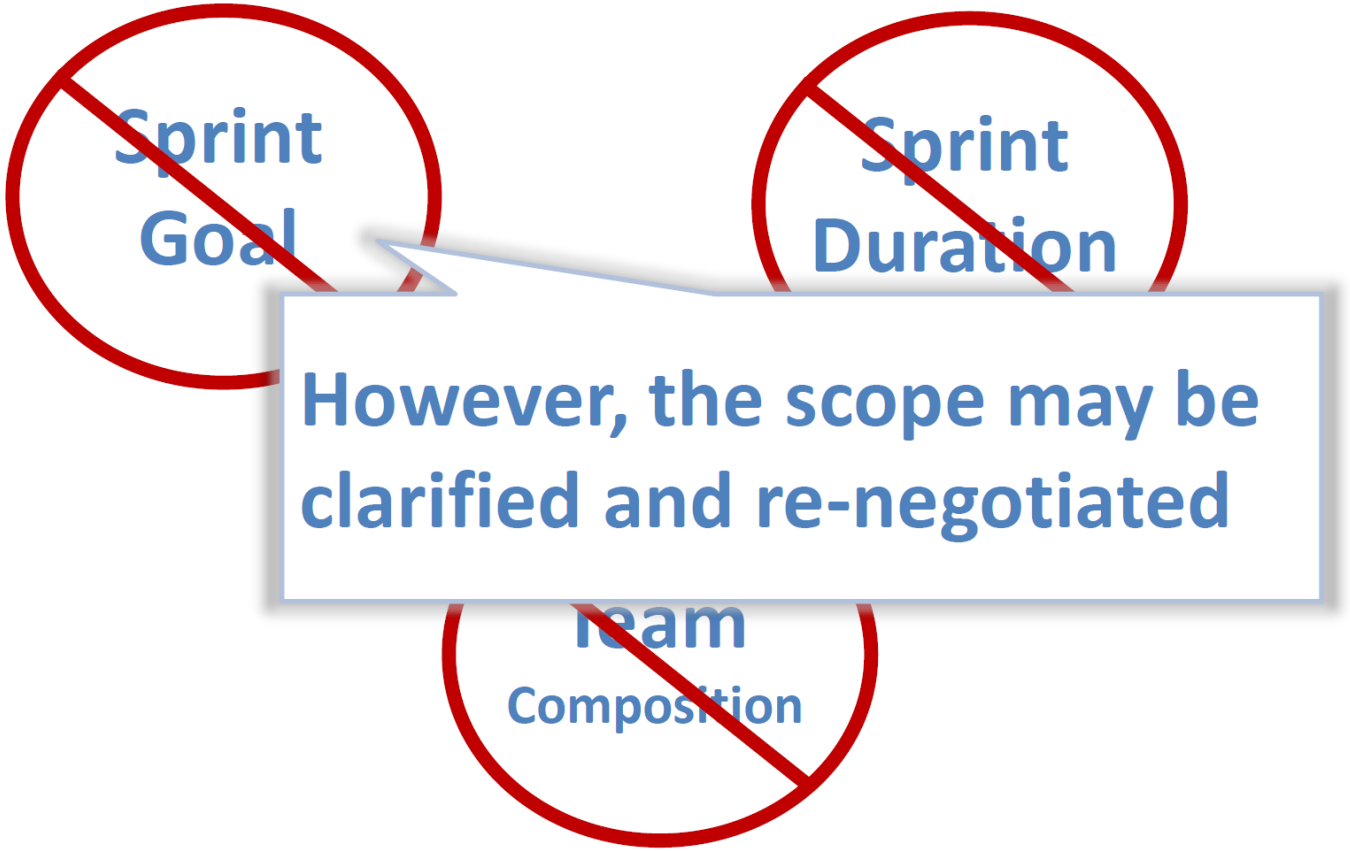
As a user, I basically want a fixed price, but I also want to be notified of critical peak pricing events ...



... respond to the time and the duration of the critical peak pricing event

... respond to emergency events

During Sprint



Exercise: Anti-patterns

- Which of the anti-patterns on the previous slides resonate most strongly?
- What other anti-patterns are you aware of?

Questions

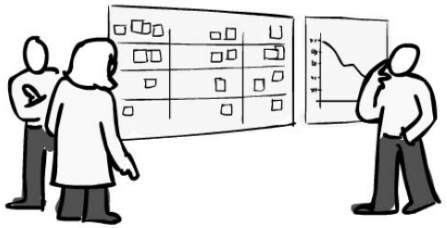
Scrum Adoption

Get everyone on same page

Train your organization on Scrum.

Make sure that everybody understands the Scrum the right way.

Identify Teams

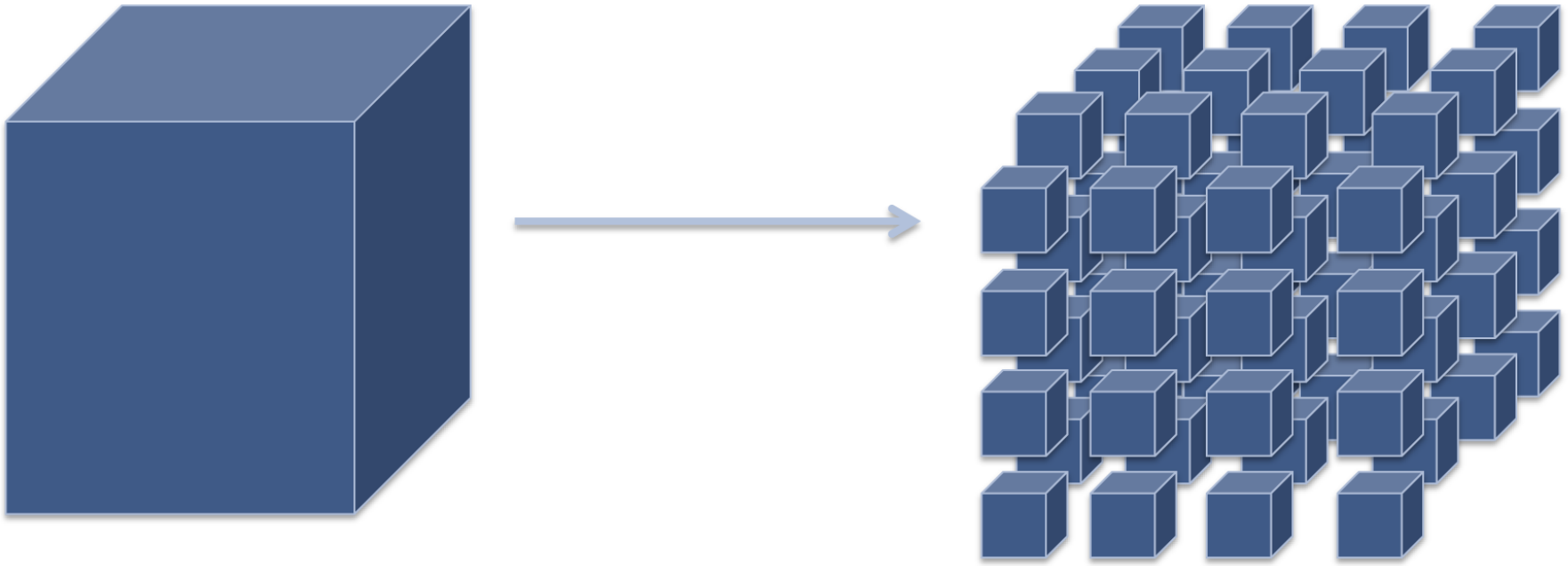


Self-Organizing

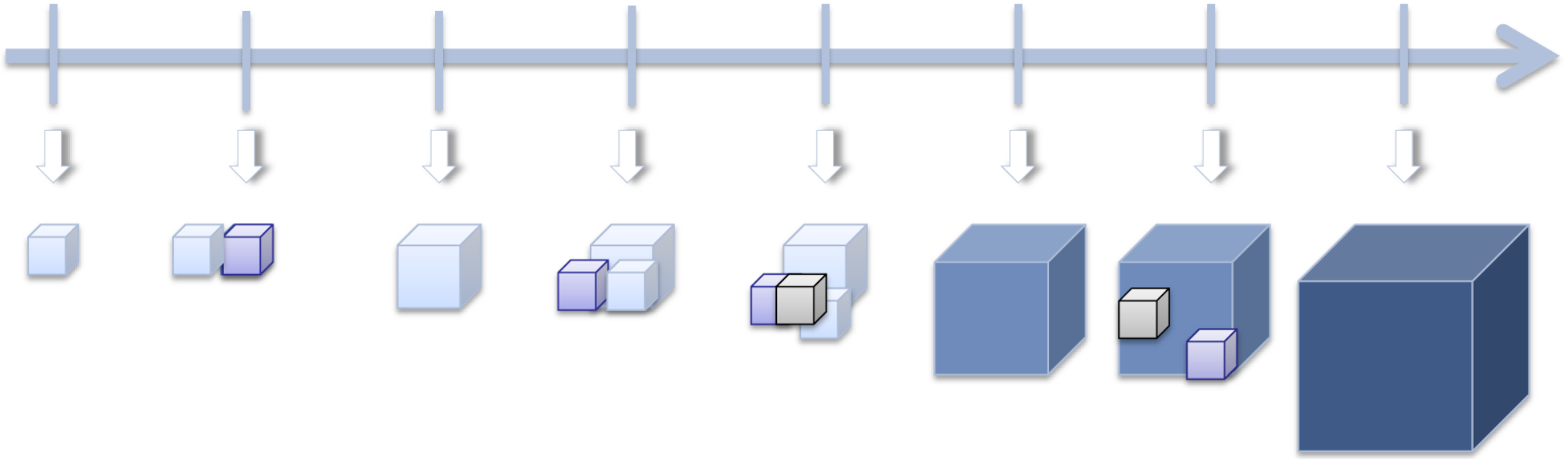
Team of Motivated Individuals

Cross-functional

Define deliverables of product



Release Planning



Setup the work environment

Same building, same floor and next to each other

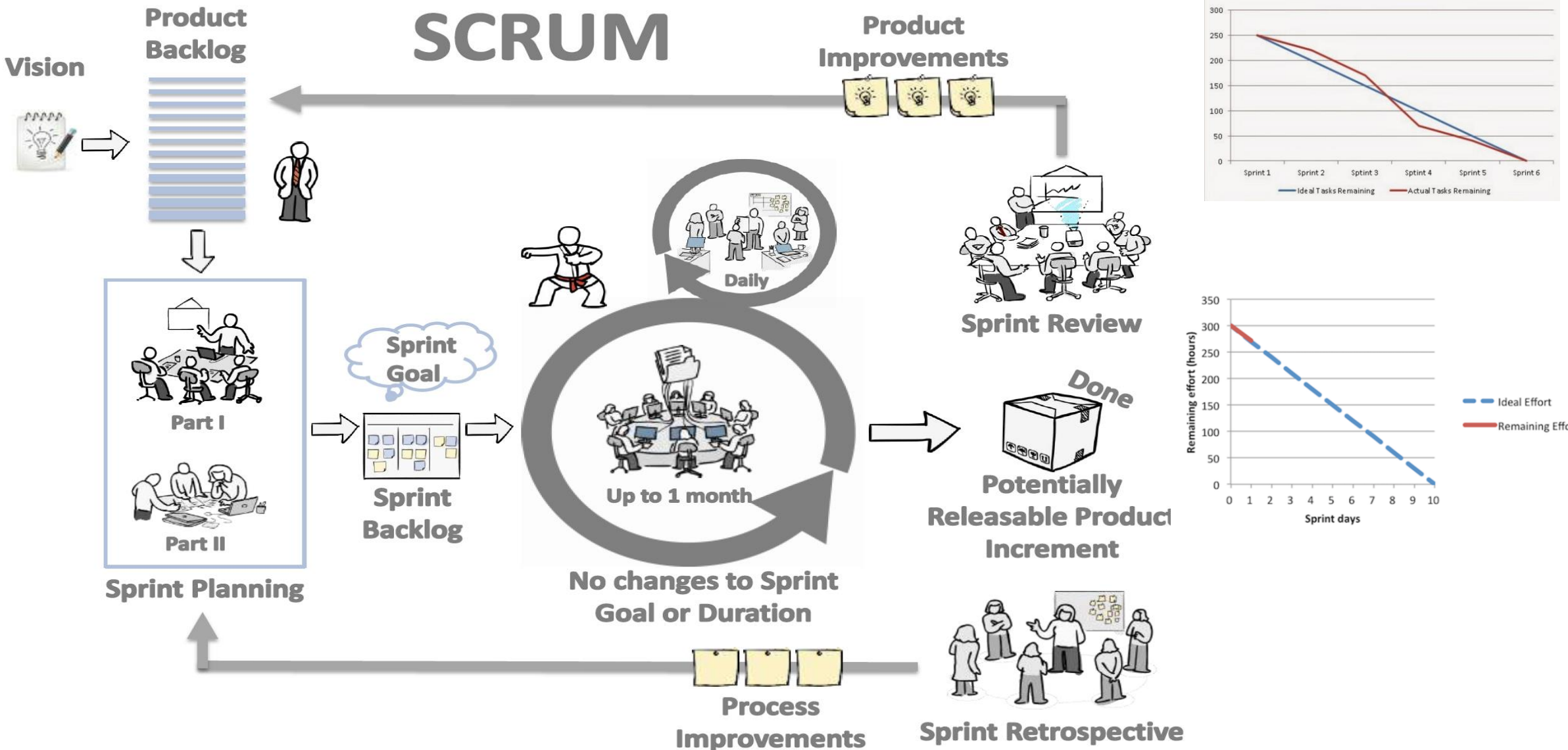
Invest on high quality video conference for exceptional cases of distributed teams

Ensure readiness to start

DoR

- ❑ Business Value clearly articulated
- ❑ Dependencies identified
- ❑ Infra Available
- ❑ Acceptance Criteria Understood
- ❑ Performance Criteria (If Any) defined and testable
- ❑ Estimation done
- ❑ No critical open questions
- ❑ Test data collected and available

Start the Process..



Reward Teams

STOP rewarding
individual contributions

START rewarding **team**
contributions

References

- Agile movement defined at http://agilemanifesto.org9
- https://www.mountaingoatsoftware.com/agile
- https://www.scrumalliance.org/
- https://www.**agile**business.org/what-is-**dsdm**

- **Downloads:**
  - **Scrum & JIRA Resources: https://tinyurl.com/jyt9p3y**

Questions

rgujar@gmail.com

+91-808 708 7574

https://www.linkedin.com/in/rgujar