

AGILE SCRUM

Introduction to Scrum

Outline

- ▶ What is Scrum?
- ▶ Scrum History
- ▶ Three Pillars of Scrum
- ▶ Scrum Concepts
- ▶ Scrum Process Overview
- ▶ Real-Life Example

What is Scrum?

- ✓ **Iterative and incremental** agile software development framework
- ✓ Defines a flexible, holistic strategy to **work as team** to reach the goal
- ✓ Enables teams to **self-organize** by encouraging people collaboration
- ✓ Follows **Empirical Process control** to increase speed and flexibility
- ✓ Framework consists of **roles, artifacts, events and rules**
- ✓ Challenges assumptions of the "traditional, sequential approach"

History

- In **Rugby**, a [scrum](#) refers to the manner of restarting the game after a minor infraction.
- In 1986, **Hirotaka Takeuchi** and **Ikujiro Nonaka** described a new approach to product development that would increase **speed and flexibility**.
- They called this **rugby approach**, as the whole process is performed by **cross functional teams** "tries to go the distance, passing the ball back and forth"



1986 –Takeuchi & Nonaka coined Scrum product development in Harvard Business Review

1993 – Jeff Sutherland created Scrum process

2001 – Agile Manifesto

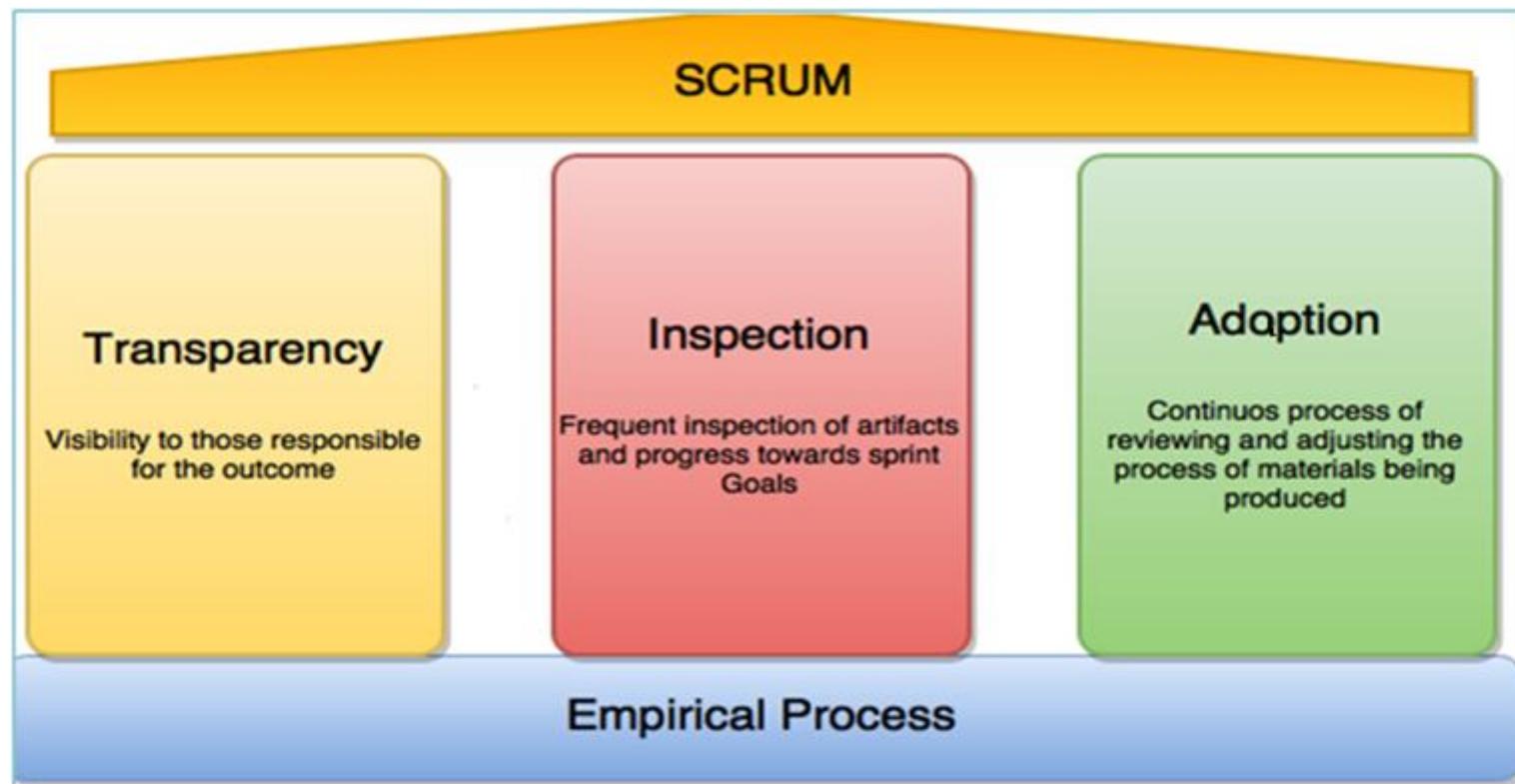
2002 – Scrum Alliance Created

2007 – Scaled Agile Framework (SAFe)



Three Pillars of Scrum

Scrum is founded on empirical process control theory, or empiricism. Empiricism asserts that knowledge comes from experience *and* making decisions based on what is known. Scrum employs an iterative, incremental approach to optimize predictability and control risk.



Scrum Concepts



Empirical Process Control

Scrum prescribes making decisions based on observation and experimentation rather than detailed upfront planning.



Self-Organization

Scrum believes that today's workers have much more knowledge to offer than just their technical expertise and that they deliver greater value when self-organized.



Collaboration

In Scrum product development is a shared value-creation process that needs all the stakeholders working and interacting together to deliver the greatest value.



Prioritization

Delivering the greatest value in the shortest amount of time requires prioritization and selection of what will be done from what needs to be done.



Time-boxing

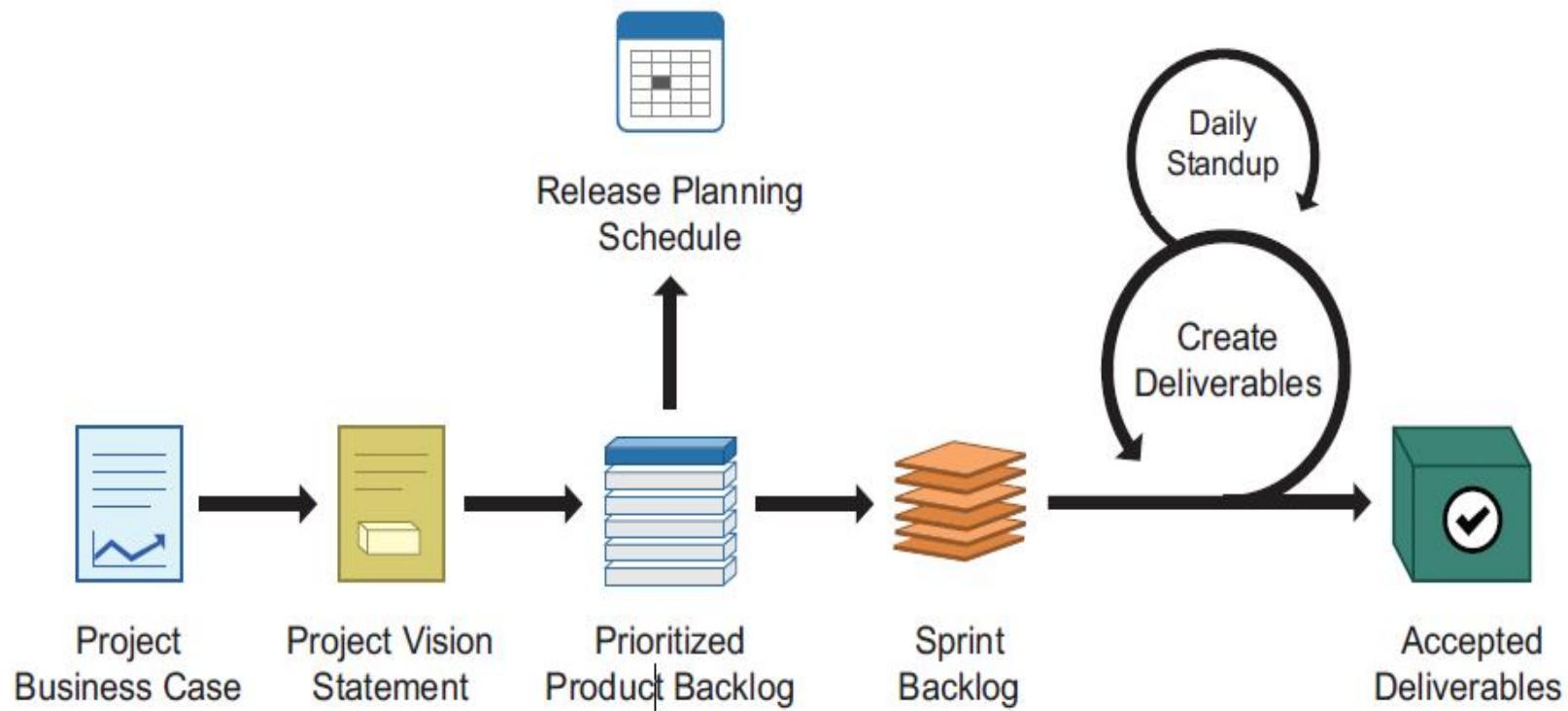
Time is treated as a **limiting constraint**, and time-boxing is used as the rhythm to which all stakeholders work and contribute.



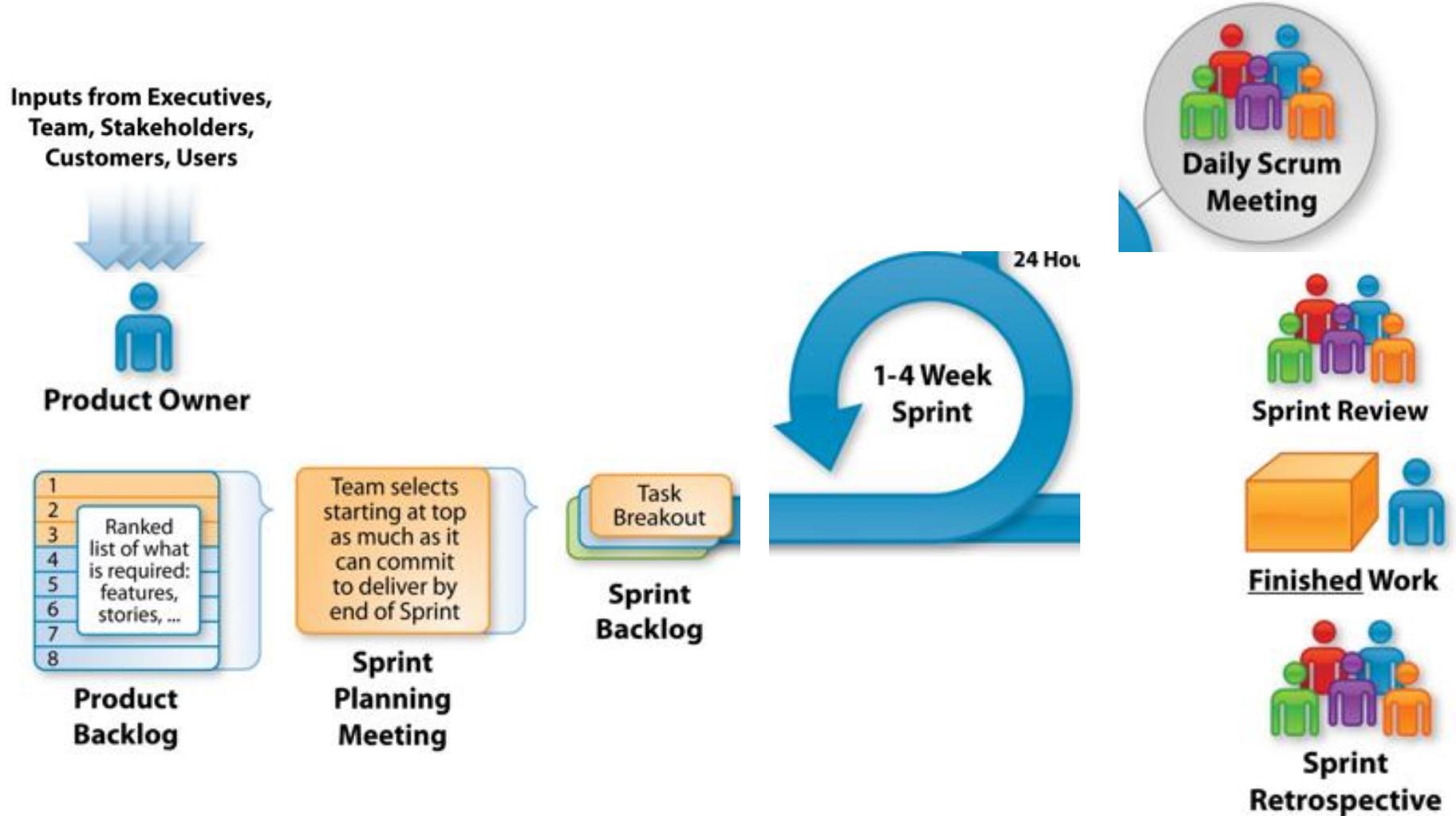
Iterative Development

The customer may not be always able to define very concrete requirements. The iterative model is more **flexible** in ensuring that any change requested by the customer can be included as part of the project.

Scrum Process Overview



Scrum Process Overview contd...

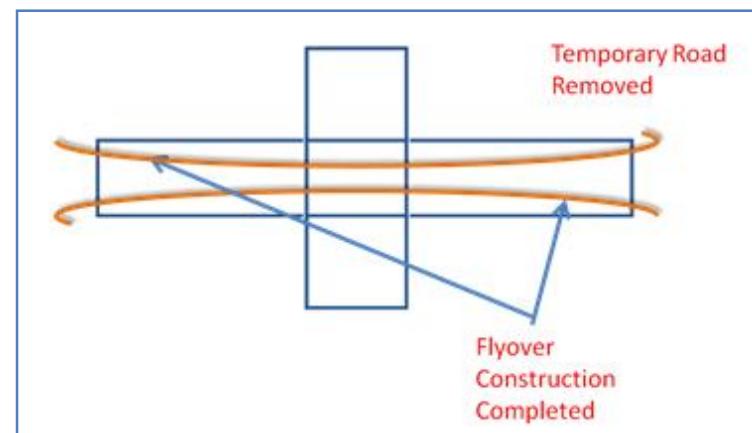
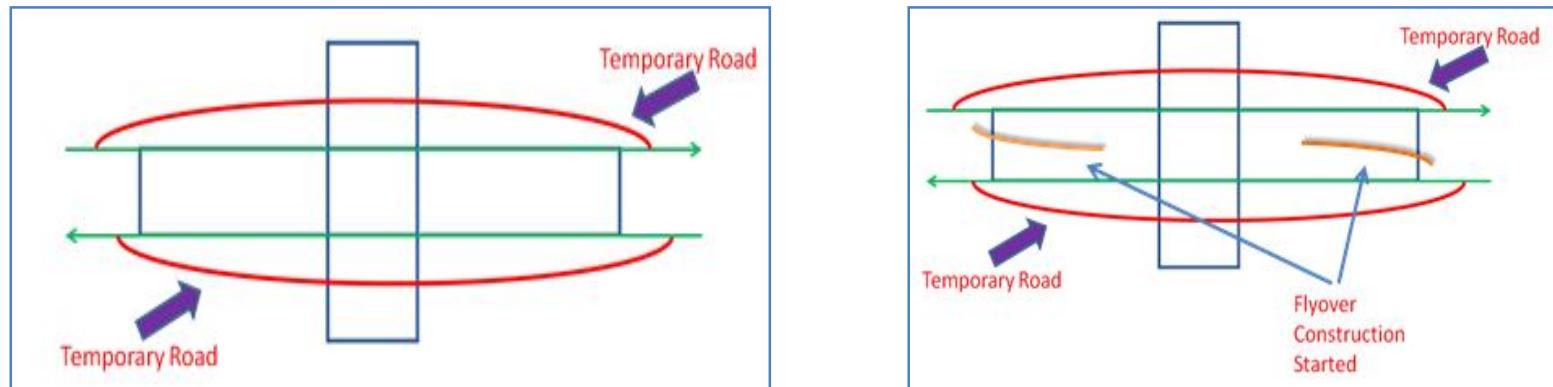


The key is to fail early, change course, and continue along the path to success.

Real-Life Example of Agile Scrum

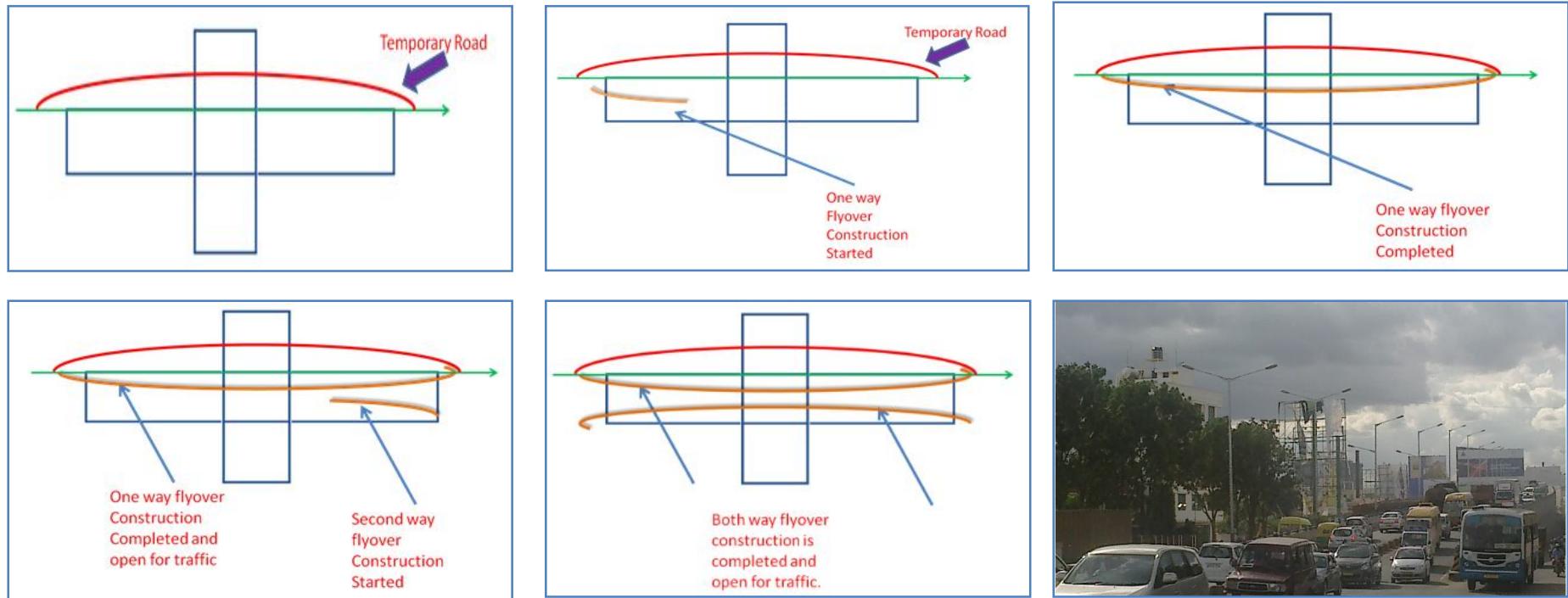
- Construction of a flyover on an extremely busy intersection in Bengaluru
- Takes 18 months of construction for this magnitude
- High traffic inflow (during peak hours) - managed with two-side temporary roads till completion

Traditional Construction Approach:



Real-Life Example of Agile Scrum contd...

Incremental Construction Approach:



- End user (commuter) got to use the flyover in half the time (9 months instead of 18)
- With a divider on the first way flyover, used for two-way traffic – reduced traffic at junction
- Flyover completed with one temporary road – reduced cost
- Second way flyover constructed without extreme pressure, as one flyover is operational

Scrum (Deep Dive)

Outline

- ▶ Scrum Roles
- ▶ Scrum Artifacts
- ▶ Scrum Events
- ▶ User Stories
- ▶ Estimation
- ▶ Planning
- ▶ Review and Tracking

Scrum Roles



Product owner

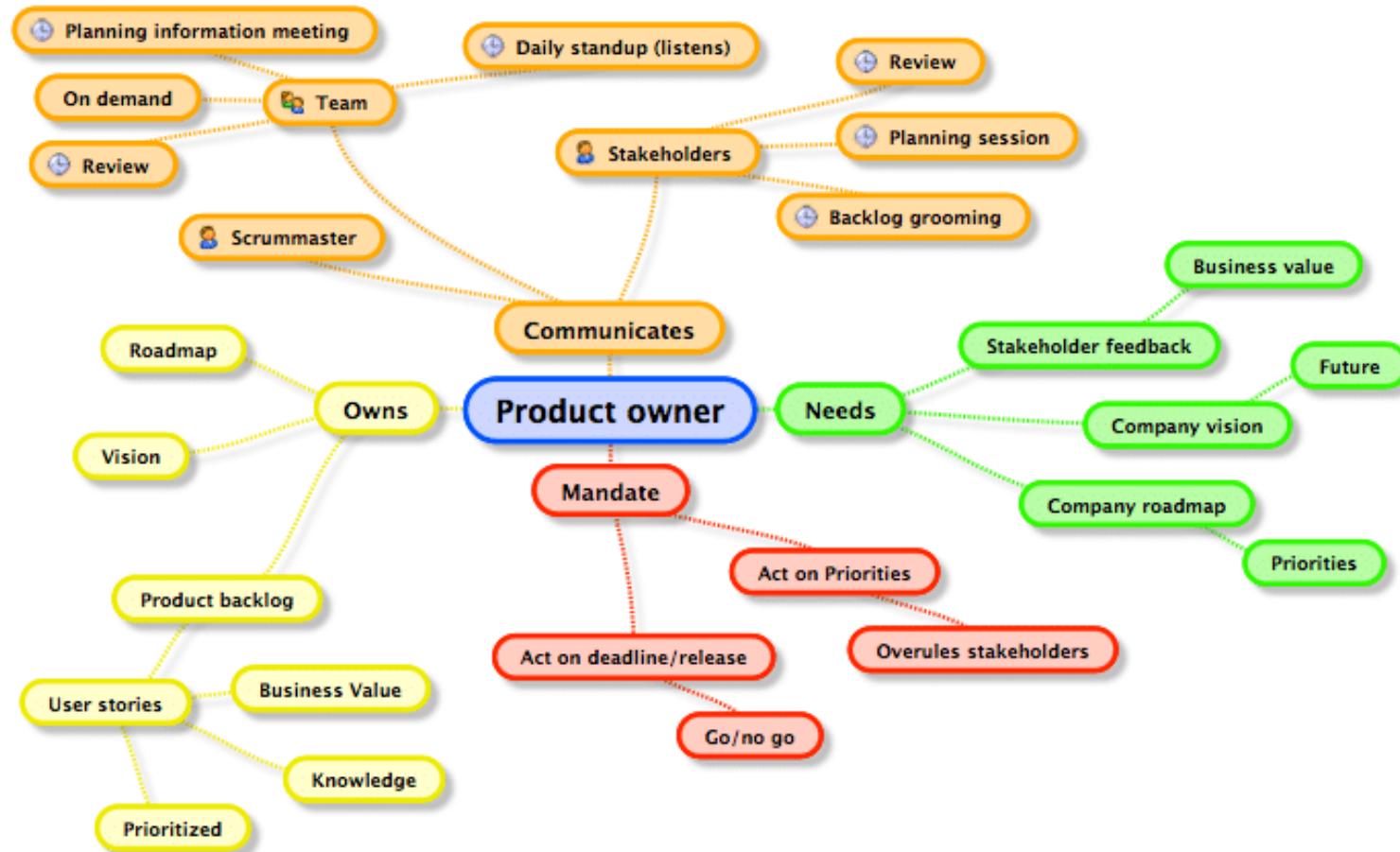


Product owner contd...



- Responsible for assessing viability and ensuring delivery of the product
- Decides on the project vision, release dates and is the voice of the customer
- Prioritize items in the Product Backlog according to business value
- Ensures transparency and clarity on the Prioritized Product Backlog items
- Provides acceptance criteria and inspects deliverable to validate them

Product owner contd...

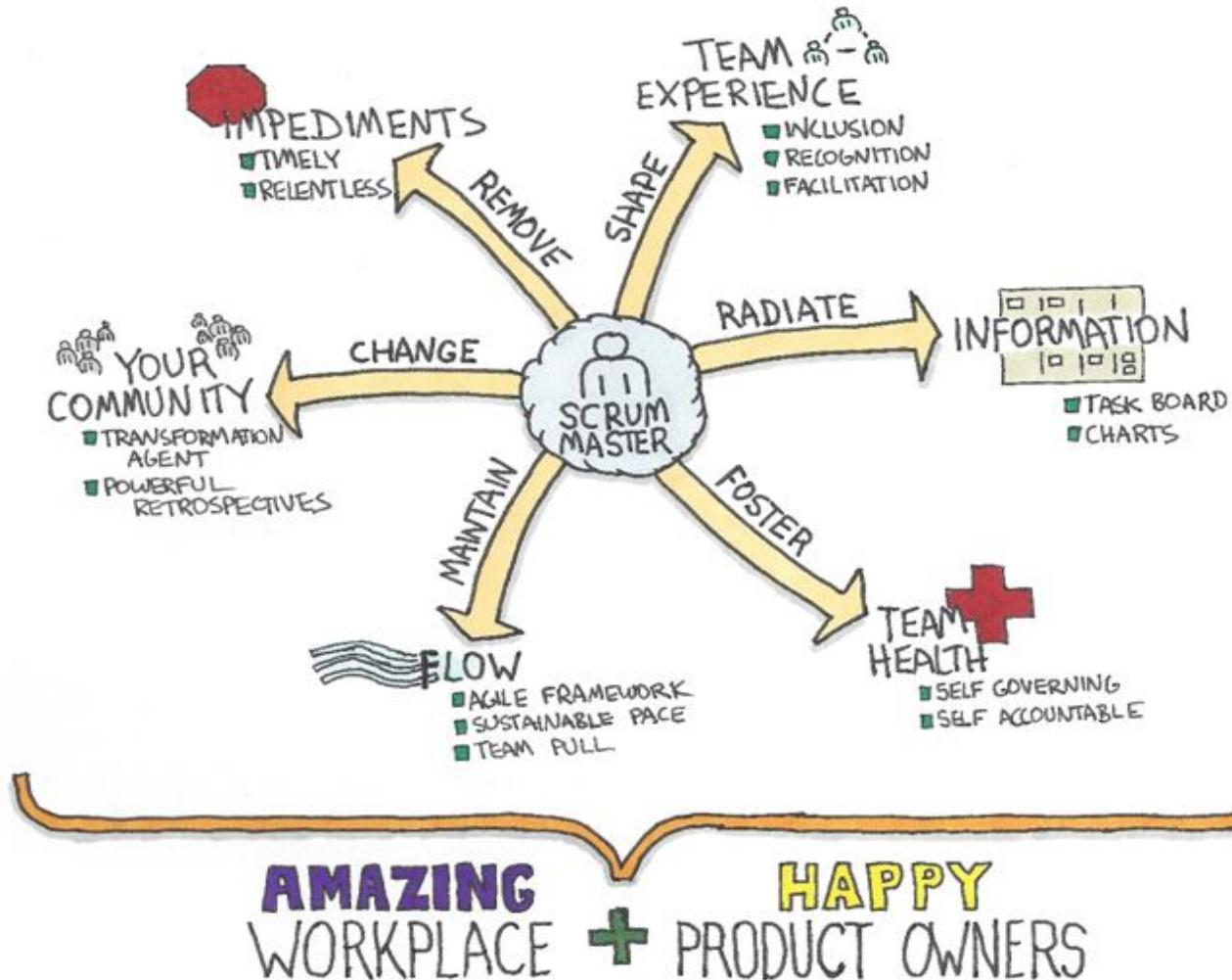


Scrum Master



A servant leader would never ask, “So, what are you going to do for me today?” Instead, a servant leader asks, “So, what can I do today to help you and the team be more effective?”

Scrum Master contd..



The team



The team contd..

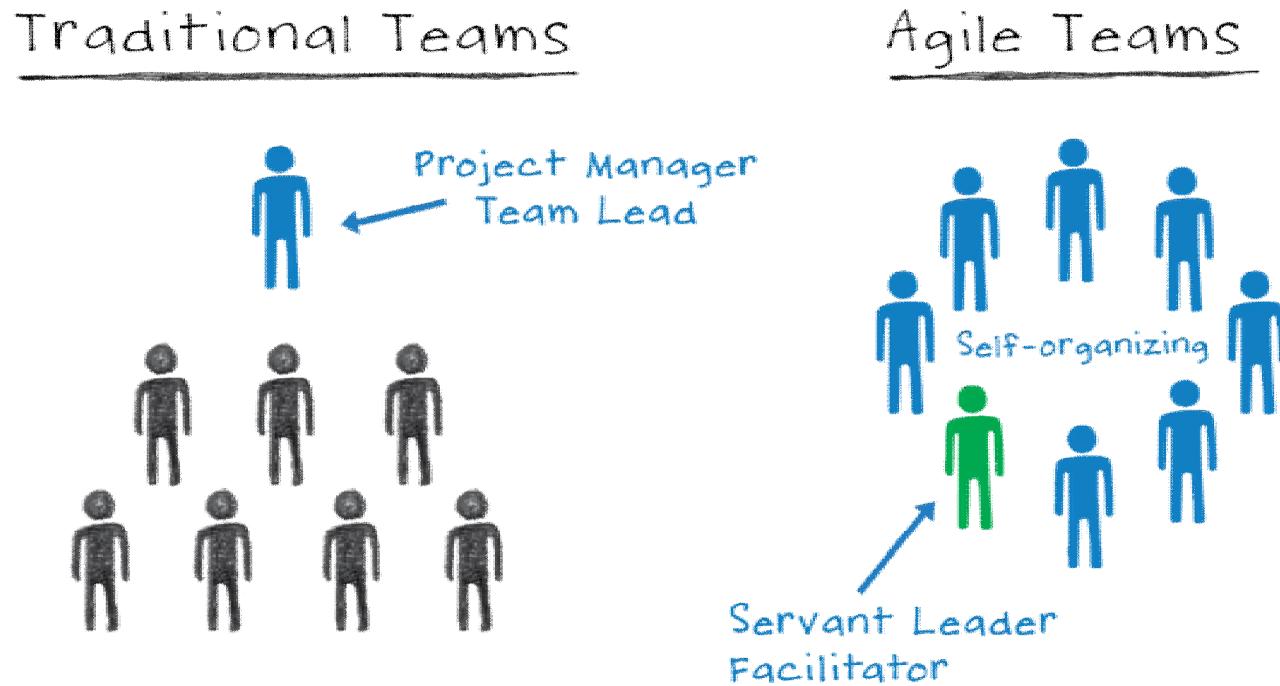


- Typically a small team of 6-10 members with no further sub-division of teams
- Cross-functional and self-organizing and enjoys complete autonomy during a Sprint
- Members are generalists across domains and specialists in at least one area
- Responsibility of the work lies with the whole team
 - Responsible for Estimation and Work Break-down
 - Participation in below ceremonies
 - Product Backlog Grooming
 - Release Planning
 - Sprint Planning
 - Daily Scrum Meeting
 - Sprint Review
 - Sprint Retrospective
 - Release Retrospective



The team contd...

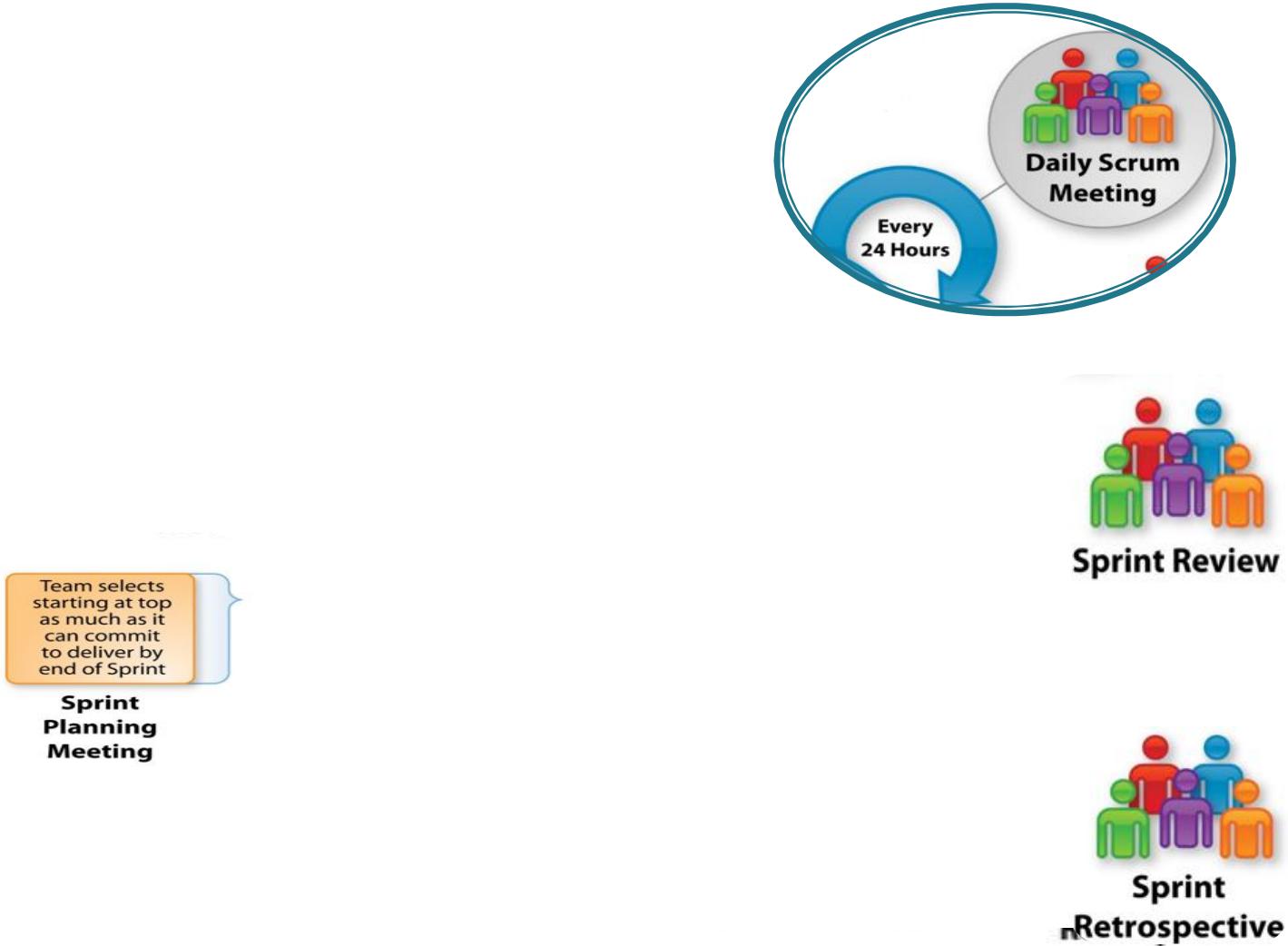
- ▶ Teams are self-organised and self disciplined.
- ▶ Team is responsible for completing their commitment in a sprint.
- ▶ Responsible to have a potentially shippable functionality at the end of a sprint.



Scrum Artifacts



Scrum Events



Scrum Phases and Processes

Initiate	Plan & Estimate	Implement	Review & Retrospect	Release
Create Project Vision	Create User Stories	Create Deliverables	Convene Scrum of Scrums	Ship Deliverables
Identify Scrum Master & Stakeholder(s)	Approve, Estimate and Commit User Stories	Conduct Daily Standup	Demonstrate and Validate Sprint	Retrospect Project
Form Scrum Team	Create Tasks	Groom Prioritized product Backlog	Retrospect Sprint	
Develop Epics(s)	Estimate Tasks			
Create Prioritized Product Backlog	Create Sprint Backlog			
Conduct Release Planning				

Role Play: Mock Product Development

In this role-play, we will develop an Online Sales Channel for a fictional company.

We will go through a simulated but complete Scrum project, practicing all meetings prescribed by Scrum and using the related artifacts as we learn one by one.

Each step of the role-play will use the output/results of preceding steps.

For practical purposes, for this role-play we will focus solely on the Web pages needed for an Online Sales Channel and assume that any necessary infrastructure (databases, server-interfaces, etc.), back-office functionality, human, and other resources already exist.

Since we can't develop real Web pages during a two-day class, we will develop them on paper using a technique called "Paper Prototyping".

Product Vision Meeting



Stakeholders such as Program Product Owner, Program Scrum Master, Chief Product Owner, and sponsors meet in a Project Vision Meeting to identify the Product Owner.

Role Play: Business Case

ValueFoods is a 10-year-old, nationwide grocery chain with about 100 outlets. Lately, the management team at ValueFoods has observed that their customers are leading a fast-paced lifestyle and do not travel long distances to do their grocery shopping. They also feel that because grocery shopping is not a highly involved process (and because ValueFoods always provides high-quality products), the best way to increase market share would be to deliver groceries to the customer's residence.

In this regard, a representative has approached your team on behalf of ValueFoods to create a website for customers to prepare their online delivery order and make payments.

The vision and generic requirements are brainstormed and created by team members based on above case.

Product Vision Statement

- First and foremost step in a scrum project
- Defines the purpose and goals of the product
- Proposed by the Product Owner and accepted by the team, it is a one sentence aim for the product

For (target customer)

Who (statement of the need of opportunity)

The (product name) **is a** (product category)

That (key benefit, compelling reason to buy)

Unlike (primary competitive alternative)

Our product (statement of primary differentiation)

Product Vision Statement Sample

For a mid-sized startup who need a customer relationship management solution, the SalesForce.com platform is a cloud-based service that provides sales tracking, lead generation, and sales representative support features that improve customer relationships

Unlike Oracle CRM, Salesforce.com provides very capable services at a moderate cost

Role Play: Create Product Vision Statement

Create product vision statement for given business case and elicit generic requirements to build product.

Team members to act as business stakeholders and provide inputs to product owner to create vision statement, review and agree the project vision statement.

For grocery shoppers

Who wants to shop groceries online

The eValueFoods is a online grocery store

That provides product catalogue, shopping cart, order placement, payment gateway, order tracking and returns features to facilitate online grocery shopping

Unlike BigBasket and Grofers

Our Product provides free delivery for all orders with real-time tracking of delivery

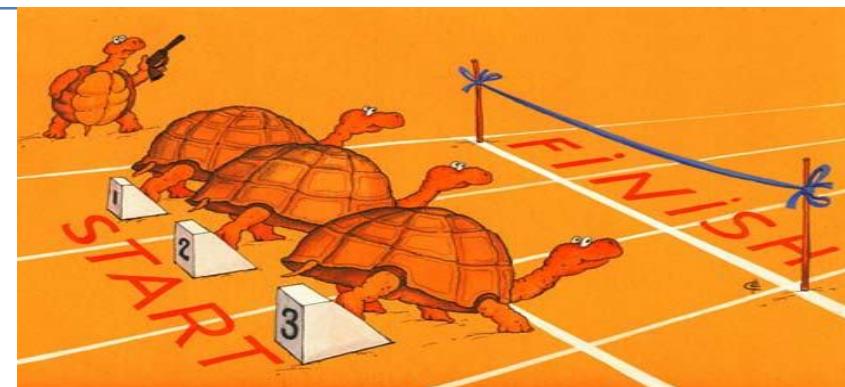
Sprint

Sprint is a time-box of one month or less during which a “Done”, useable, and potentially releasable product Increment is created

- Basic unit of development in scrum. Also known as iteration.
- Consistent Duration throughout a development cycle
- Starts immediately after the conclusion of the previous sprint

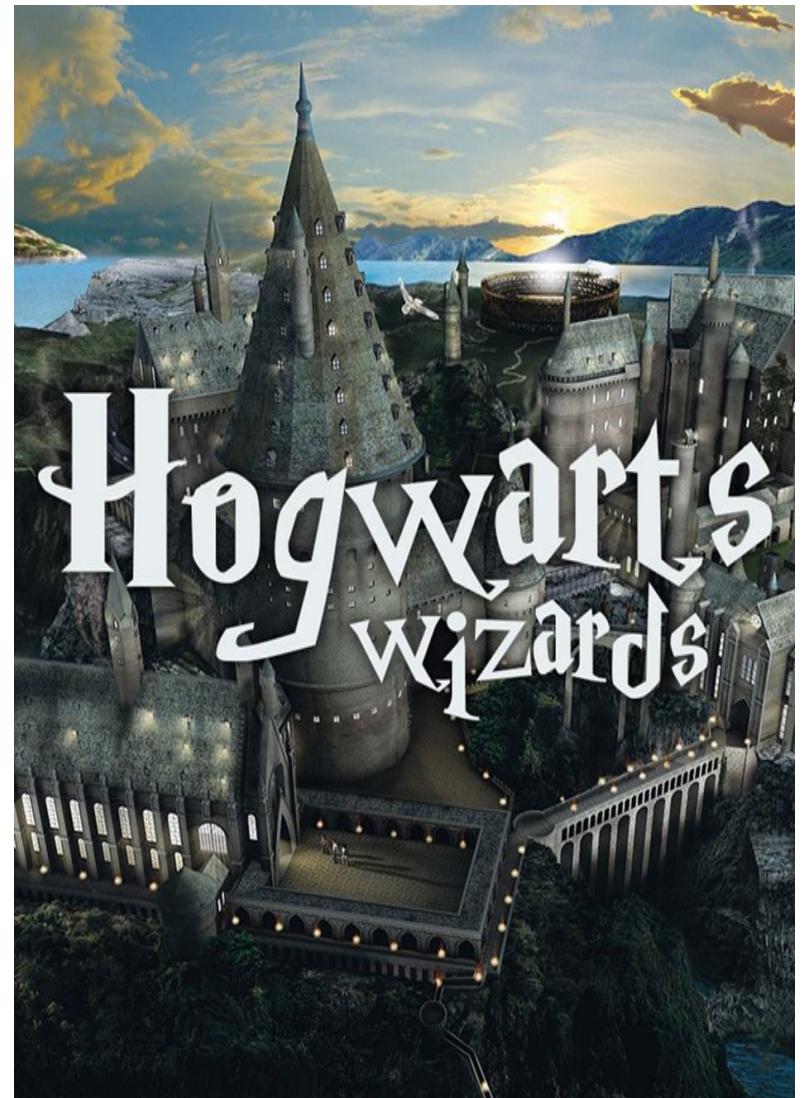
During Sprint:

- No changes are made that would endanger the Sprint Goal
- Quality goals do not decrease
- Scope may be clarified and re-negotiated between the Product Owner and Development Team as more is learned

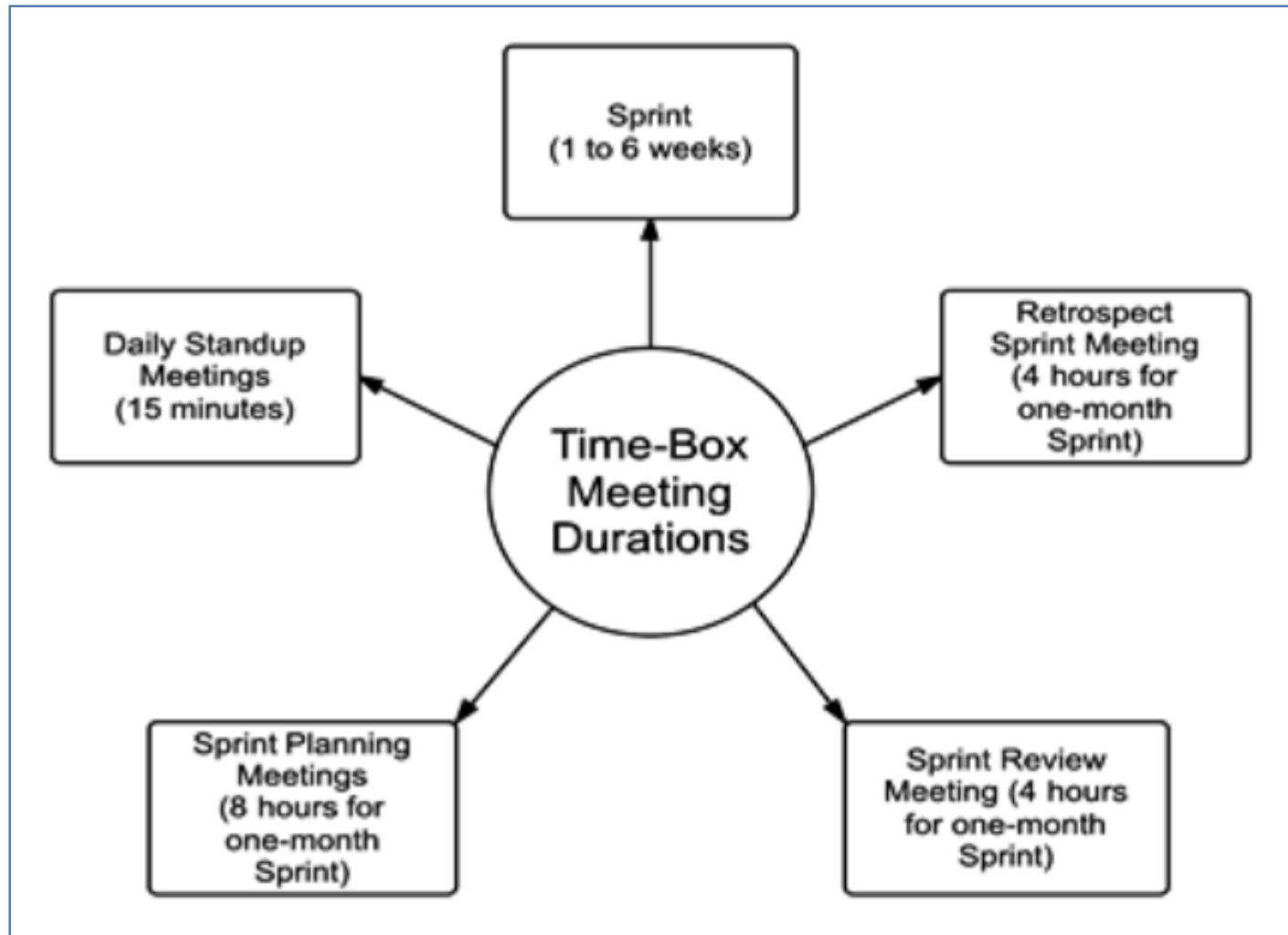


Sprint Zero Activities

- ▶ Identify and Name the team
- ▶ Choose characters/alias
- ▶ Product Backlog creation
- ▶ Architecture vision and roadmap
- ▶ Definition of Done
- ▶ Decision on scrum ceremony times



TimeBoxing

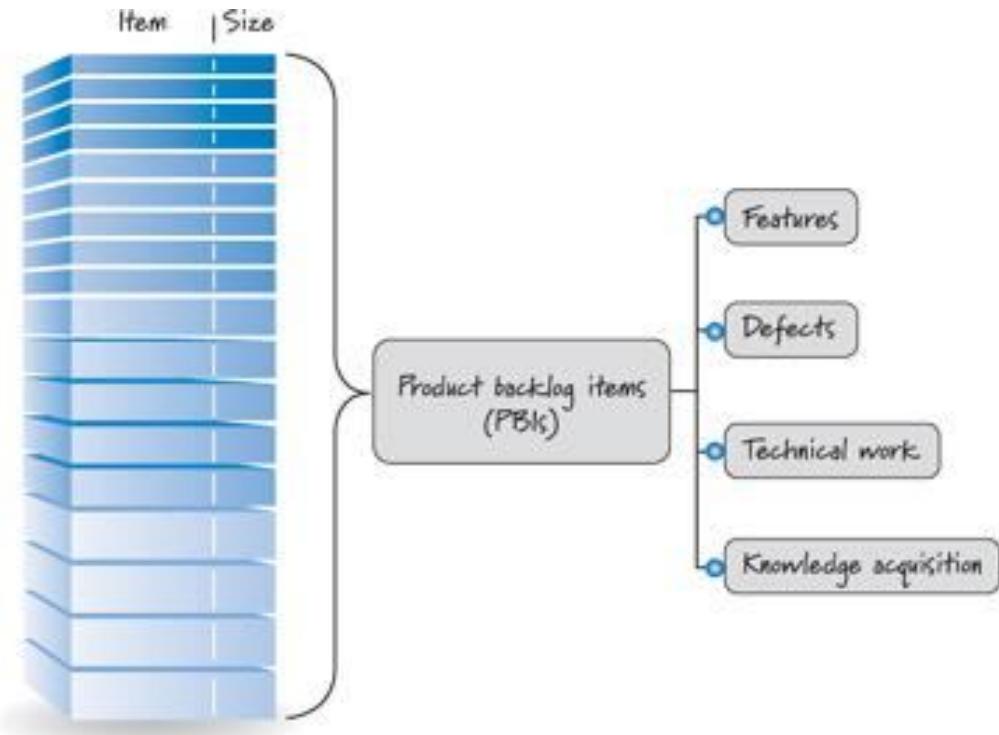
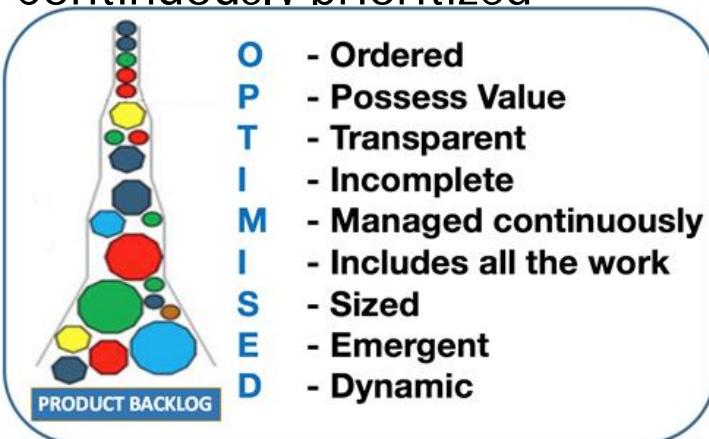


Product Backlog



Product backlog contd..

- Ordered list of all work remaining or a project
- Represents the customer's product needs and wants
- Each item will have Description, Estimate, Business Value and Priority
- Managed by the product owner
- Continuously prioritized



Writing product backlog items

- ▶ Scrum doesn't have any recommendations on this.
- ▶ User stories have become a very popular way to write PBIs in Scrum
- ▶ Not mandatory – in scrum you can write your Requirement any way you want – but user stories are highly recommended.

Backlog item
Allow a guest to make a reservation
As a guest, I want to cancel a reservation.
As a guest, I want to change the dates of a reservation.
As a hotel employee, I can run RevPAR reports (revenue-per-available-room)
Improve exception handling
...
...

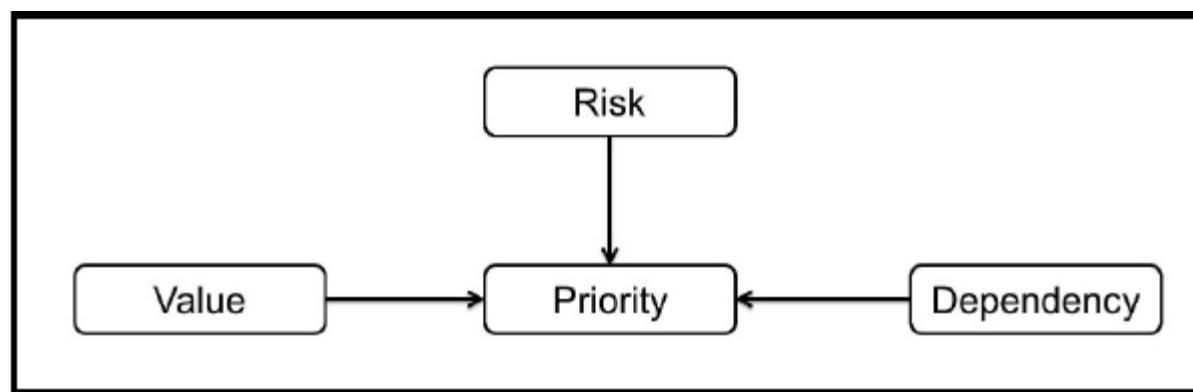
Role Play: Create Initial Product Backlog

With your team, create an initial Product Backlog (containing high-level user stories, also called “Epics”) for the given Product Vision. Some Epics have already been defined for your reference. Create additional Product Backlog items (another 7–10).

Item #	Epic Description	Priority	Estimate
1	Login feature		
2	Access to payment gateways		
3	Generic browsing and shopping cart facility		

PBI – Prioritization Techniques

- ▶ MoSCoW Prioritization
- ▶ Monopoly Money
- ▶ 100-point method
- ▶ Kano Analysis
- ▶ Value based Prioritization



PBI – Prioritization using MoSCoW technique



Sample:

	<i>Requirement</i>	<i>MoSCoW</i>
A	Users can log onto the web site.	Must
B	Users should be able to avail of a "Forgotten Password" utility .	Should
C	Users can change account details.	Must
D	A user can send an email to the system requesting a change to the account page.	Could
E	When a user clicks on a phone number on the web page a call is made automatically from their desk phone to that number.	Won't

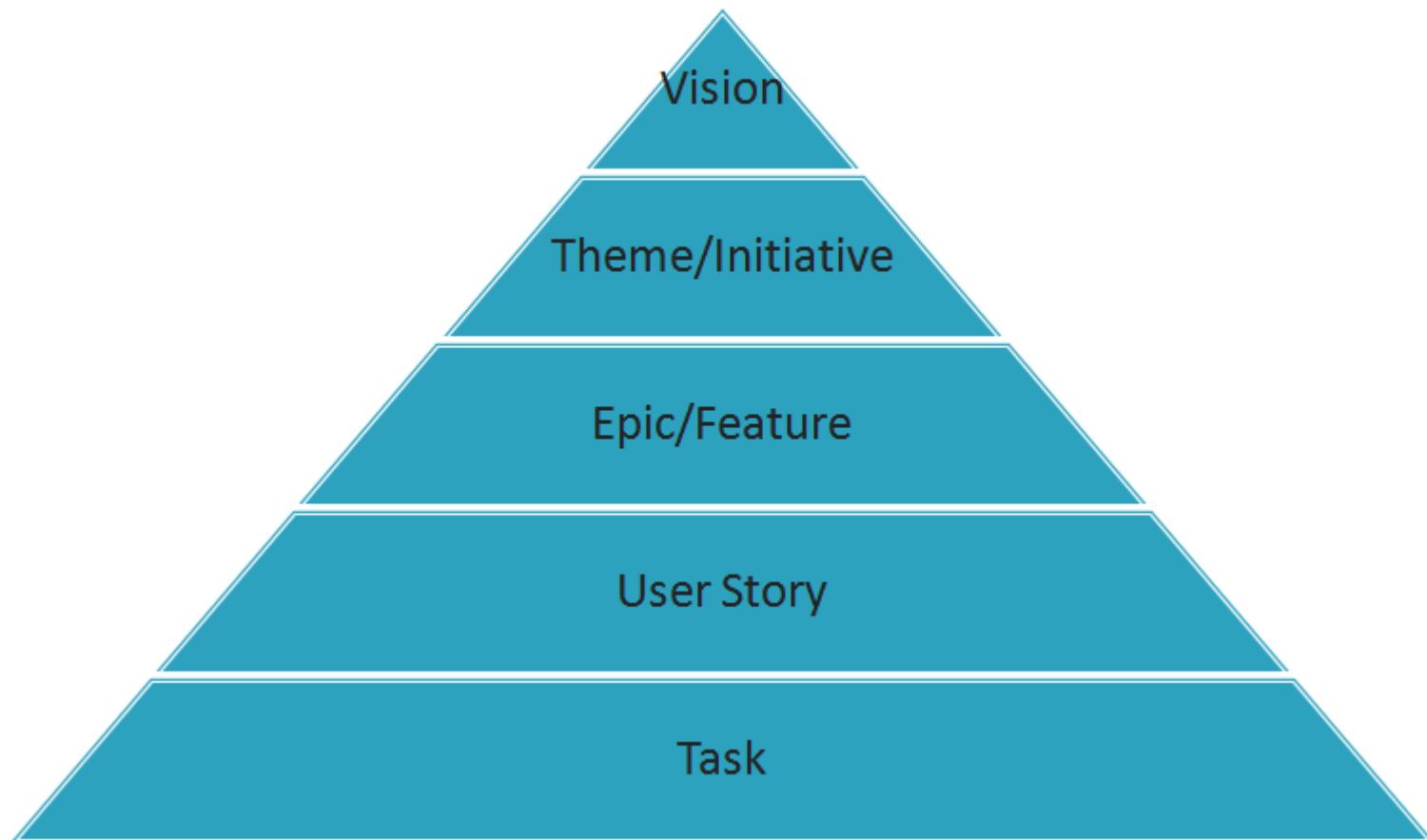
Role Play: Prioritize the Initial Product Backlog

The Product Owner from each team will prioritize the Product Backlog items in consultation with business stakeholders as per business need and market demand applying MoSCoW prioritization technique.

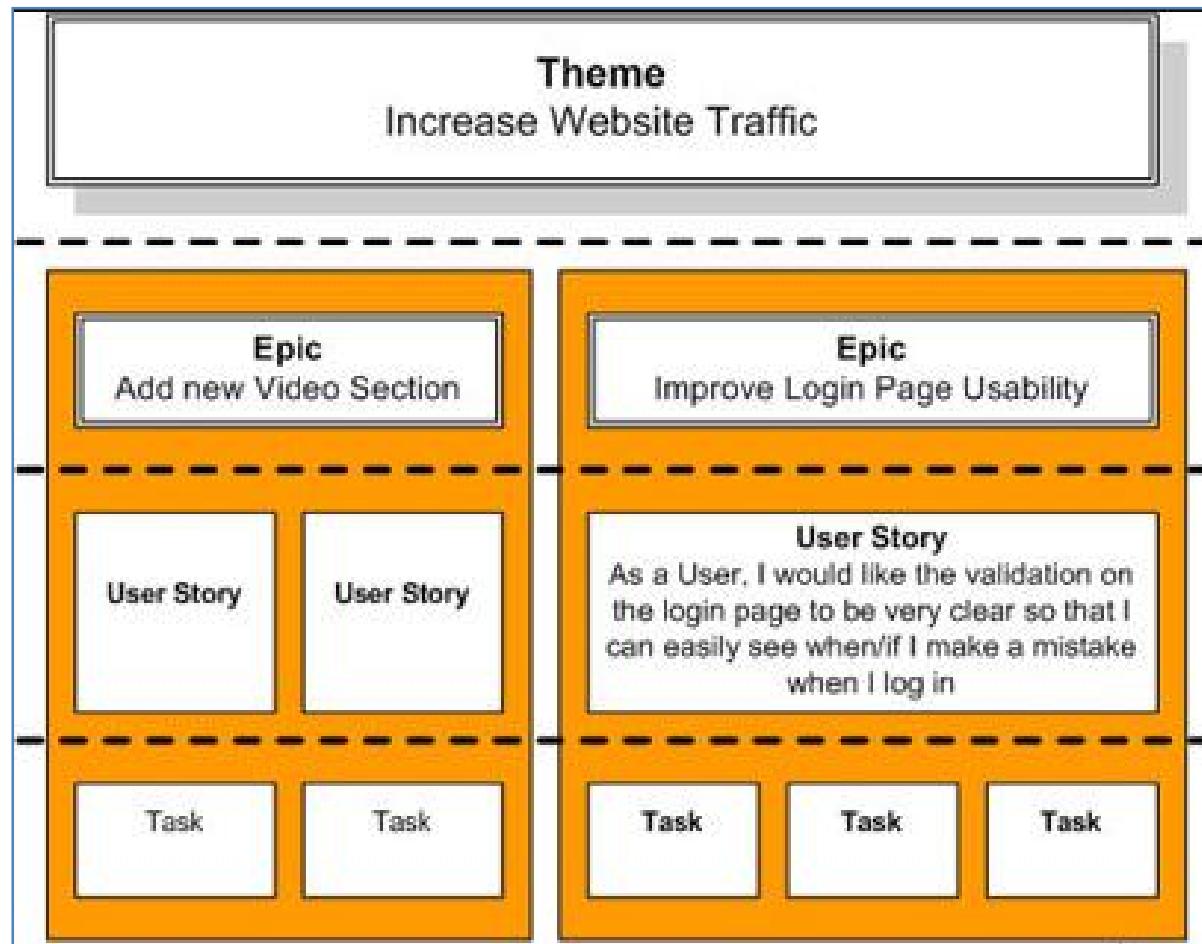
The Product Owner walk through the Product Backlog to scrum team members and re-prioritize if required

Item #	Epic Description	Priority	Estimate
1	Login feature	Could	
2	Access to payment gateways	Should	
3	Generic browsing and shopping cart facility	Must	

Themes, Epics and User Stories



Themes, Epics and User Stories



3 Cs of an user story

1. Card
2. Conversation
3. Confirmation

“Writing software that fully meets its specification is like walking on water.
For each, the former is easy if the later is frozen and near impossible if fluid.”

User stories

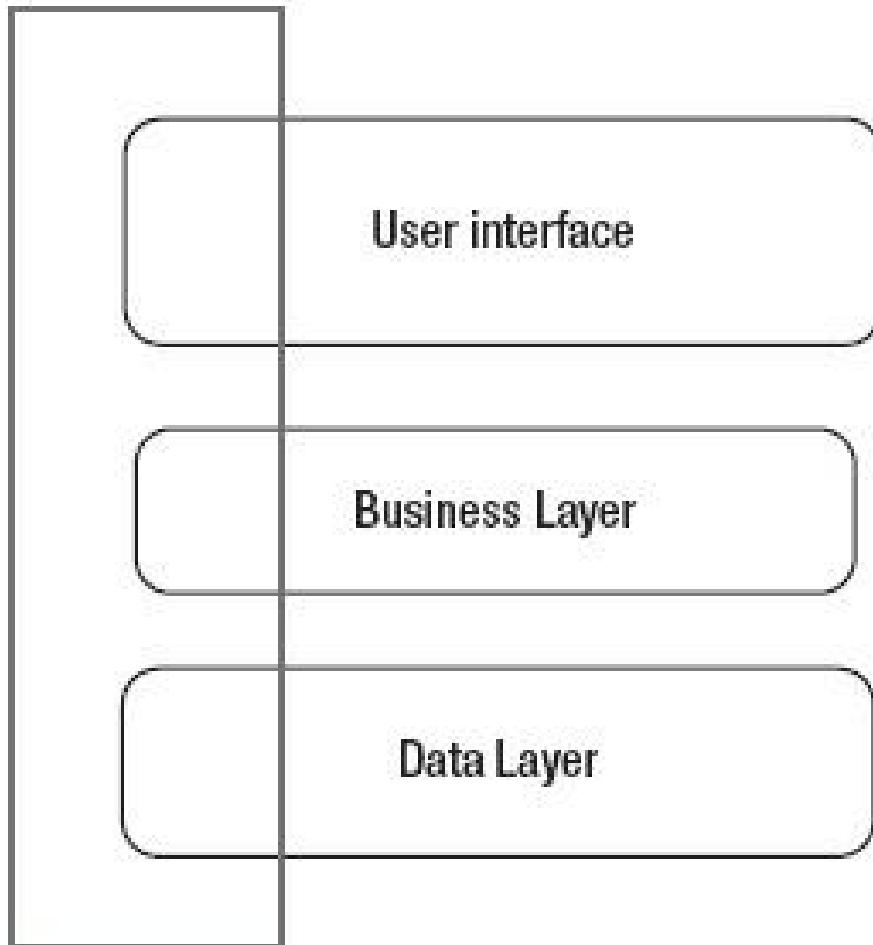
- ▶ Any feature, enhancement the customer wants.
- ▶ Bugs are not user stories.
- ▶ User stories are normally implemented in a single sprint.
- ▶ The following template is used to describe an user story

**As a <type of user>, I
want <some goal> so
that <some reason>**

*As a Game Player,
I want my Rocket to move back
and forth when I press left and
right arrows
so that I can avoid asteroids*

User stories

Vertical Slice



INVEST
Independent
Negotiable
Valuable
Estimable
Ssized appropriately
Testable

User story examples

- ▶ A user can post his/her resume
- ▶ A user can search for job postings
- ▶ A company can post job openings

- **The software will be written in C#**
- **The database connection will make use of connection pool**

User stories

- ▶ *As a* registered user, *I want* to post my resume *so that* companies can search my resume and contact me for job openings they have
- ▶ *As a* registered user, *I want* to search for job postings *so that* I can apply for the jobs which matches my skillsets.
- ▶ *As a* registered company user, *I want* to post job openings *so that* the users of the systems can search for the openings and apply for the same.

2nd C: Conversation

- ▶ The card does not include all the detailed information needed by the Dev Team
 - User Stories are designed to force a conversation between the Dev Team and the PO
 - We want to make it impossible for the Dev Team to start a User Story without a detailed discussion with the PO
- ▶ The conversation begins at the start of the project
 - The Dev Team and PO talk through the entire Product Backlog of User Stories at the project kick-off.
- ▶ The conversation continues Sprint-by-Sprint
- ▶ Together the PO, SM, and Dev Team decide how best to document the conversations – it can be as formal or informal as needed



3rd C: Confirmation

- ▶ During the conversations, the PO and Dev Team identify the “confirmations” for each User Story
- ▶ These are typically high-level acceptance criteria
- ▶ The Dev Team uses these as a guide to development, and as a way of confirming that the requirements have been met

Acceptance Criteria

- Clarification of the story
- Created by the product owner with the help of the customer

User story: As a conference attendee, I want to be able to register online, so I can register quickly and cut down on paperwork

Acceptance Criteria

- A user cannot submit a form without completing all the mandatory fields
- Information from the form is stored in the registrations database
- Spam protection is working
- Payment can be accepted as a credit card
- An acknowledgment email is sent to the user after submitting the form.

Definition of Done

- Is a checklist of valuable activities required to produce software
- The team defines definition of done for each iteration, feature, requirement task, etc.
- Typically
 - Software working?
 - Unit tests written?
 - Code review done?
 - Functional tests written and passed?
 - Non-functional tests written and passed?
 - User documentation done and reviewed?
 - Design documentation updated and reviewed?
 - Release notes updated?

Definition of Done vs Acceptance Criteria

Definition of Done	Acceptance Criteria (condition of satisfaction for User Story)
It serves the purpose of making unambiguous understanding of what all is needed before any product backlog item can be declared complete.	It serves the purpose of clarifying business requirements / conditions which must be met to satisfy the user for given requirement.
Definition of Done uniformly applied to all product backlog items.	It applies to specific product backlog item since it clarifies one item
Development team owns Definition of Done and it is understood and agreed by complete scrum team.	Product Owner owns Acceptance criteria, and development team understands them
Definition of Done does not change frequently, it is not expected to change during the sprint	Acceptance criteria are negotiable between product owner and development team.
Meeting Definition of Done ensures one meets the acceptance criteria	Just meeting Acceptance Criteria may not necessarily mean that Definition of Done is also met.

Role Play: Create User Stories

The Product Owner from each team will create user stories for the top priority epics available in the Product Backlog in collaboration with team members and update into Product Backlog.

Item #	Epic Description	User Story Description	Priority	Estimate
1	Login feature	As a registered user I want to login into the system so that I can view and shop grocery items	Could	
2	Access to payment gateways	As a registered user I want to access to payment gateways so that I can make payments online	Should	
3	Generic browsing and shopping cart facility	As a public user I want browse and add items to shopping cart so that I can place order online	Must	

Release planning

- ▶ Create a plan to deliver an increment to the product
- ▶ Product Vision, Roadmap
- ▶ Review previous releases
- ▶ Release name/theme
- ▶ Velocity / Determine Sizing Values
- ▶ Release Schedule
- ▶ Map stories to iterations/sprints
- ▶ Determine no of sprints

Role Play: Release Planning Meeting

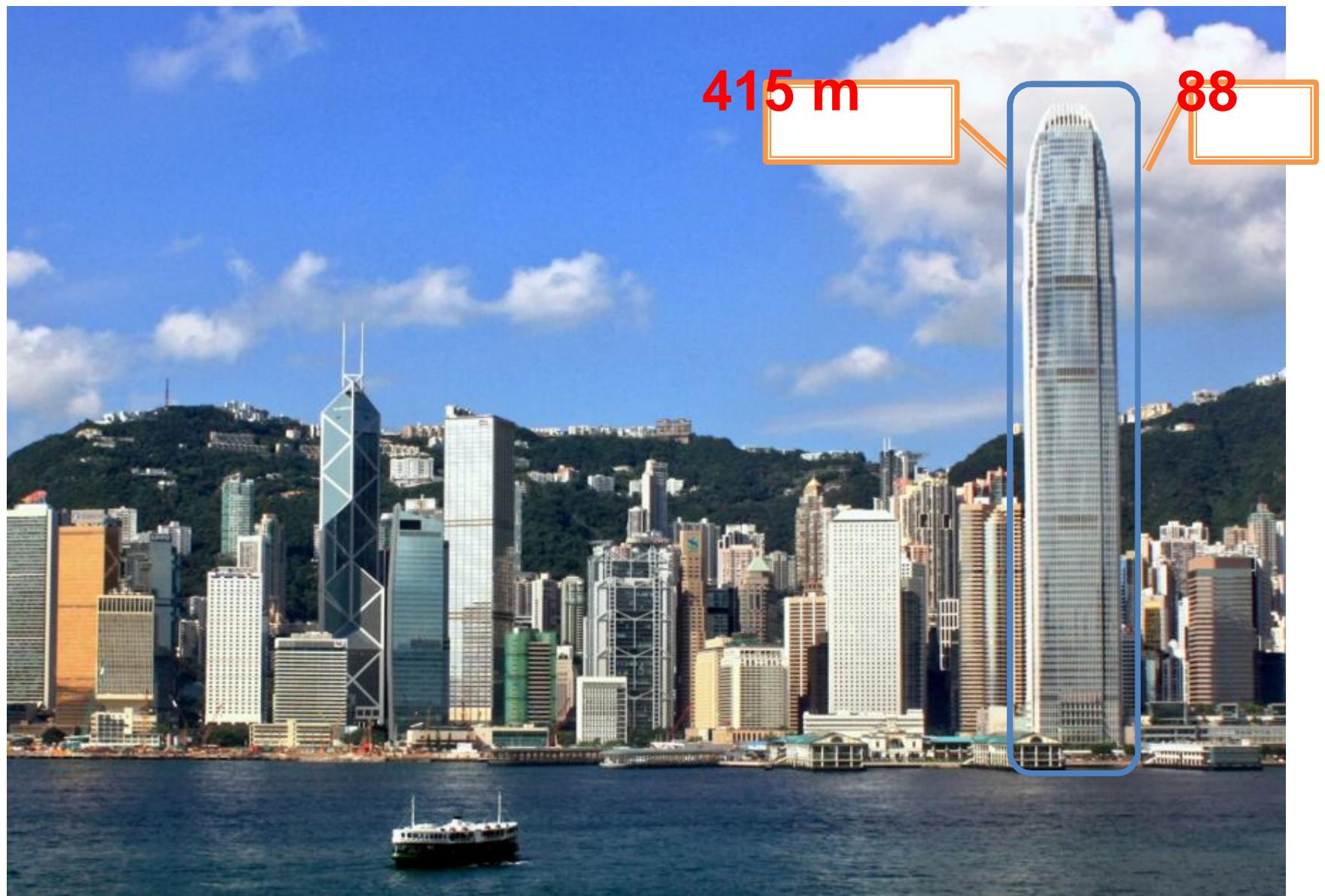
The Product Owner from each team will now conduct a Release Planning Meeting to explain the project requirements to the team, decide on the length of the Sprint, and discuss deadlines and the Product Backlog.

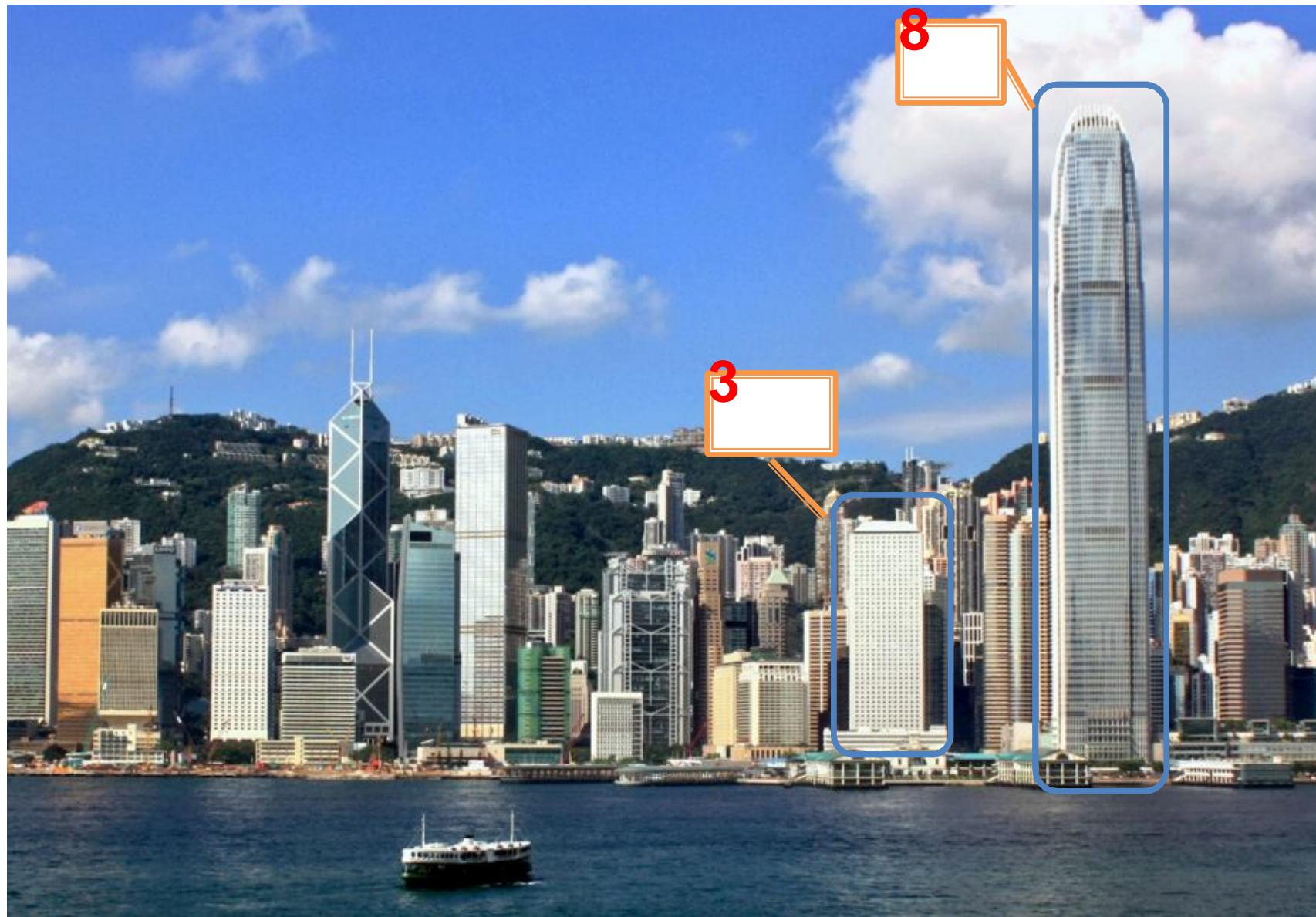
Product backlog grooming.

- ▶ Not officially recognized as part of scrum ceremonies.
- ▶ The product owner is responsible for product backlog grooming.
- ▶ It is a collaborative effort by the customer/stakeholders, product owner and the scrum team.
- ▶ Product owner collaborates: first with stakeholders and then with the development team.
- ▶ Thumb rule is to allocate 10% teams time for grooming every sprint.
- ▶ Requirements are not handed over to the team; the team members co-author them.
 - PO, dev team and scrum master work together to split large work items to smaller slices.
 - Ideally, small enough that 1-2 people could finish them in 3-4 days

A well groomed backlog is pre-requisite for a successful sprint planning meeting.

Estimation Techniques





Planning poker

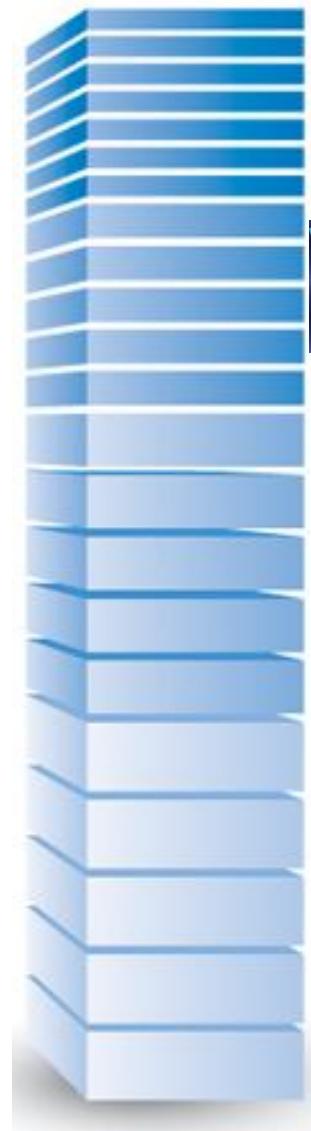
- ▶ Story points is the measure used to estimate
- ▶ *Fibonacci-like sequence:* 1, 2, 3, 5, 8, 13, 20, 40, 60, 80, 100
- ▶ Use hand or planning poker card.
- ▶ First a base line story is estimated.
- ▶ Complexity of a user story in relation to baseline story or other stories.
- ▶ User stories can be estimated based on their complexity, size or unknown.
- ▶ No more than 3 rounds to agree on numbers.



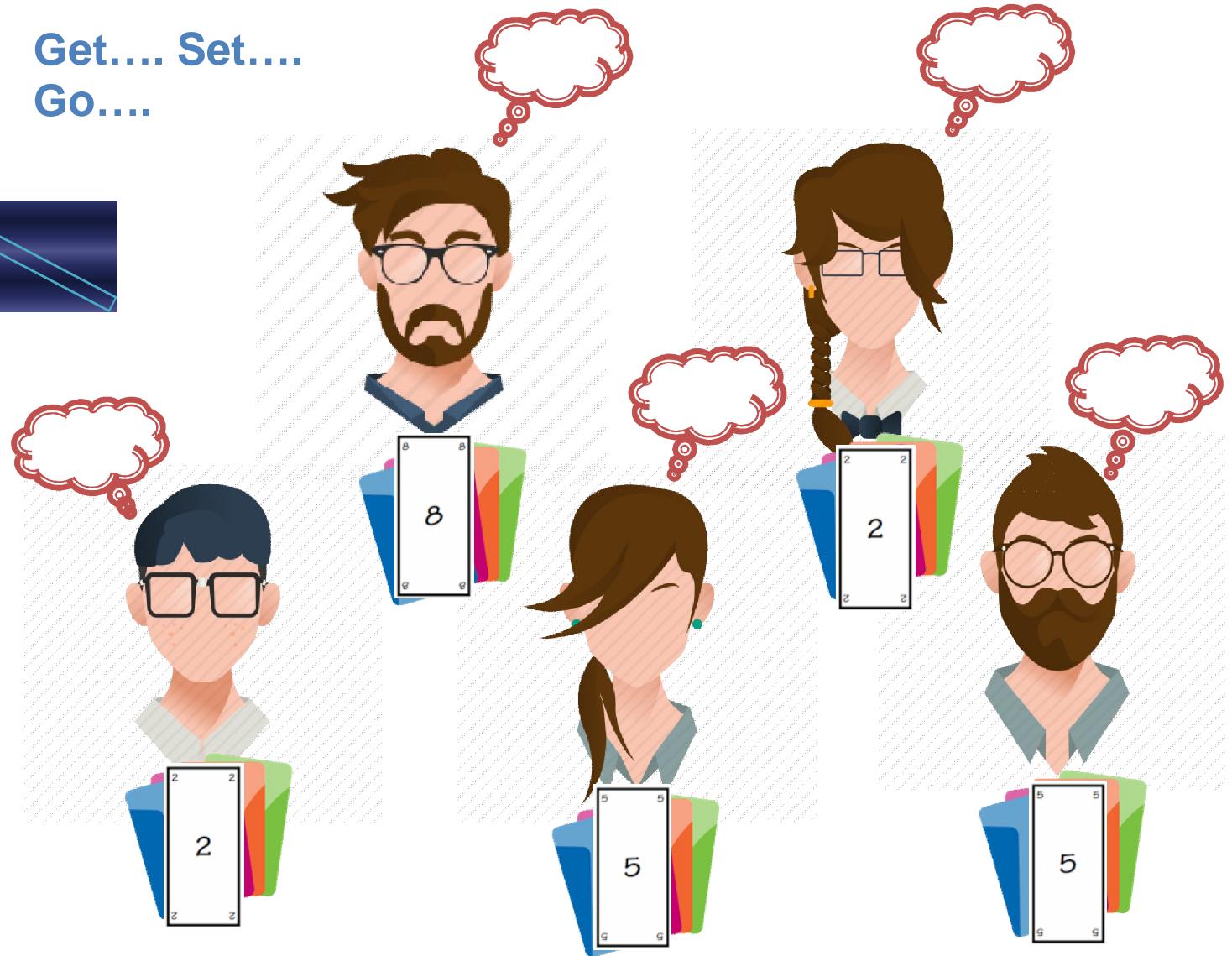
Planning poker contd...



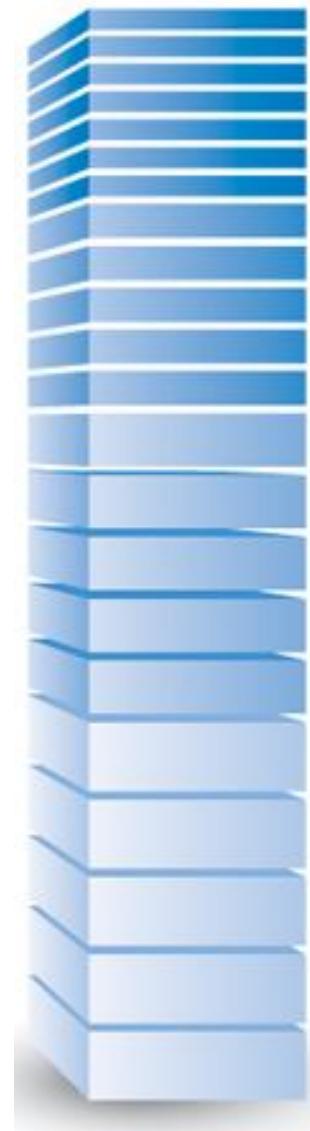
Planning poker contd...



Get.... Set....
Go....



Planning poker contd...



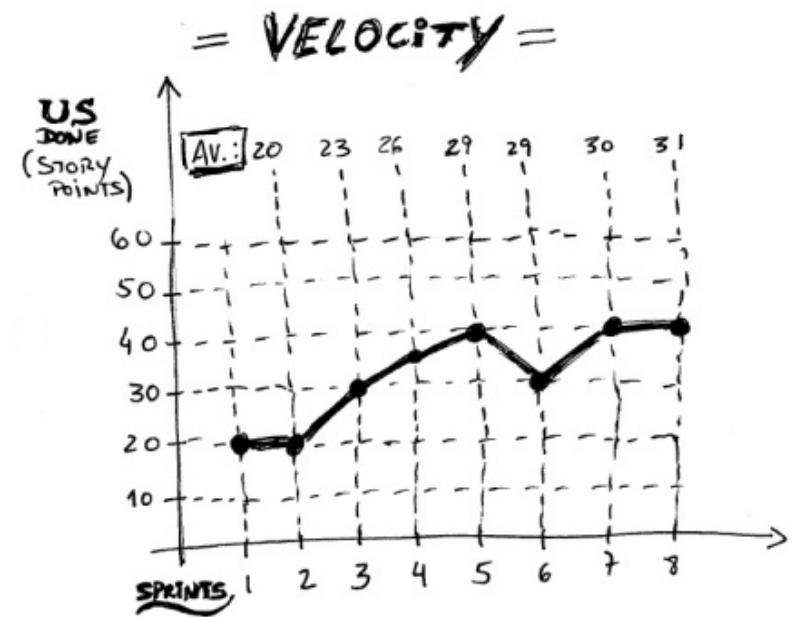
Get.... Set....
Go....



Velocity

- ▶ Extremely simple, powerful method to measure the rate at which scrum teams can deliver business value.
- ▶ Sum of all the story points delivered / number of sprints

Sprint	Story points	Velocity
Sprint 1	20	20
Sprint 2	23	$(20 + 23) / 2 = 21.5$
Sprint 3	26	$(20 + 23 + 26) / 3 = 23$
Sprint 4	29	24.5
Sprint 5	30	25.6
Sprint 6	31	26.5



Role Play: User Story Estimation

With your team, estimate the groomed and prioritized user stories available in the product backlog using Planning Poker estimation technique.

Item #	Epic Description	User Story Description	Priority	Estimate
1	Login feature	As a registered user I want to login into the system so that I can view and shop grocery items	Could	3
2	Access to payment gateways	As a registered user I want to access to payment gateways so that I can make payments online	Should	5
3	Generic browsing and shopping cart facility	As a public user I want browse and add items to shopping cart so that I can place order online	Must	8

Sprint Goal

- A short statement of what the work will be focused on during the sprint
- Examples:
 - ~~Make the application run on SQL server in addition to Oracle.~~
 - Make the website open nicely on mobile devices.
 - Support more technical indicators than company ABC with real-time streaming data.
 - ~~Bug fixing, hmmm....~~

Sprint planning meeting



Sprint planning meeting

- ▶ 2 part meeting
- ▶ Product owner, scrum master and the team.
- ▶ Stakeholders may attend the meeting by invitation, although it is rare.
- ▶ Time boxed to 4 hours (2 weeks sprint)

- Beginning of iteration
- Review vision and roadmap
- Review development status, architecture and previous iterations
- Determine velocity
- Review team availability & capacity
- Review definition of done
- **Review product backlog & select items for iteration / sprint**
- **Identify tasks & estimates**
- **Identify challenges & risks**
- Review capacity required
- **COMMIT** (planning not over until entire team commits to sprint plan)

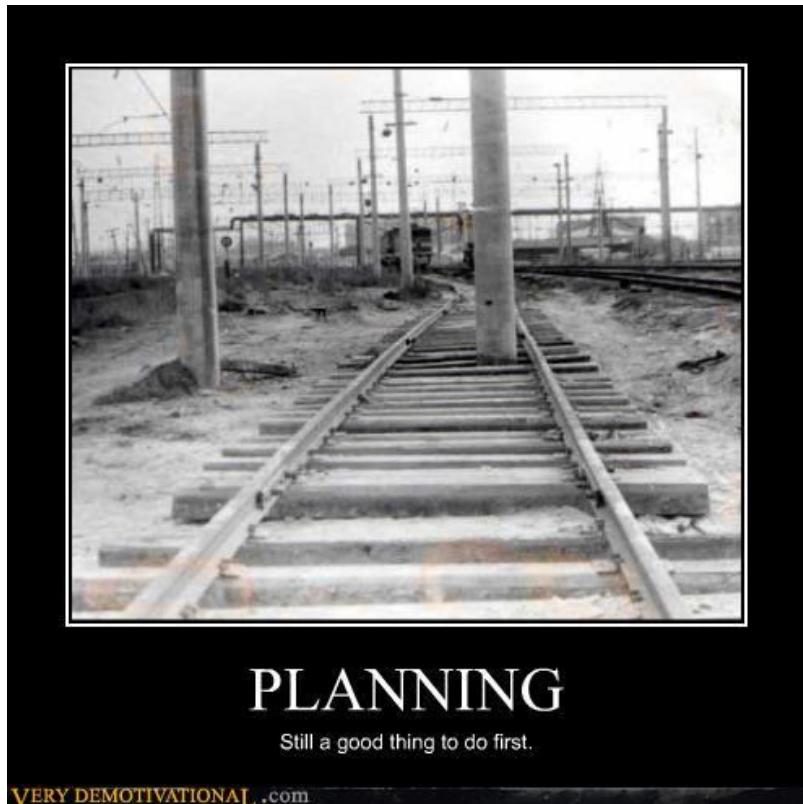
1st part of sprint planning meeting

- ▶ The product owner describes what he/she wants to be build for the sprint
- ▶ The team asks questions back and forth to get more clarity on the user stories/bugs
- ▶ The team and product owner negotiates on what needs to go into the current sprint.
- ▶ The team decides what they can commit for the current sprint.
- ▶ At end the team comes up with a one sentence goal for the sprint



2nd part of sprint planning meeting

- ▶ The committed items for the current sprint are decomposed into smaller tasks.
- ▶ The product owner should be available for the meeting but not mandatory to be present. In essence he should be available for any doubt clearance.
- ▶ If the product owner is physical present, the scrum master should make sure the product owner is not influencing the team on the tasks and their times.
- ▶ The team assigns hours against each task.
- ▶ Output is sprint backlog



Sprint backlog



Sprint backlog contd..

- ▶ Outcome of sprint planning.
- ▶ List of all user stories and tasks committed for a sprint.
- ▶ Each task usually estimated in hours.
- ▶ Teams commitment for the sprint.
- ▶ Sub task has both dev, QA, DBA, BA or any other tasks needed to complete a user story.

Role Play: Sprint Planning Meeting

Using the existing Prioritized Product Backlog (initially provided by the Product Owner and refined by you during Release Planning), plan your next Sprint with the faculty as your Product Owner, commit to the scope of your next Sprint, and create a Sprint Backlog. List all the tasks related to each user story and estimate them. Create an initial Burndown chart.

Item #	Epic Description	User Story Description	Priority	Estimate
	Sprint 1			16
1	Login feature	As a registered user I want to login into the system so that I can view and shop grocery items	Could	3
2	Access to payment gateways	As a registered user I want to access to payment gateways so that I can make payments online	Should	5
3	Generic browsing and shopping cart facility	As a public user I want browse and add items to shopping cart so that I can place order online	Must	8
	Sprint 2			

Role Play: Paper Product Development

We do a mock product development using the technique of paper product development. The technique will be explained to you by your faculty member. You will use it to develop the features committed to in the Sprint Planning Meeting.

In this role-play, the duration of a simulated workday will be time-boxed to 5 minutes.

A Sprint will consist of 3 working days of 7 minutes (2 minutes Daily Standup, 5 minutes development).

Daily stand-up



Daily stand-up contd..

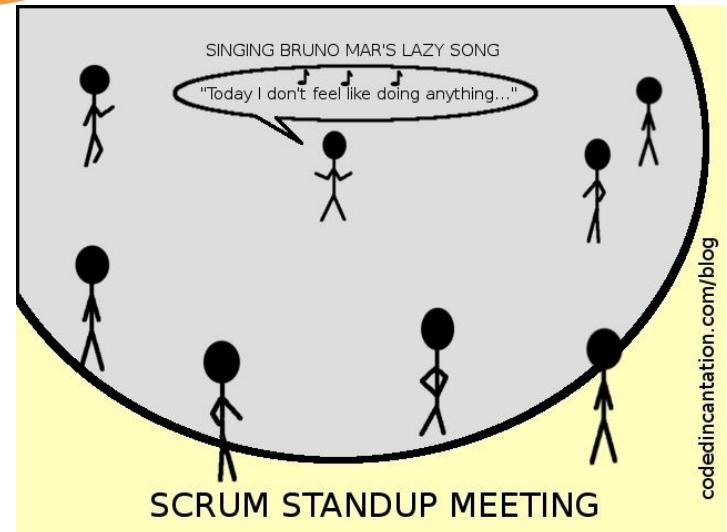
- ▶ Everyone standing (whenever possible) and time boxed to 15 minutes.
- ▶ Attendees: the team, scrum master and product owner
- ▶ Stakeholders are optional, can join the meeting but only Pigs are allowed to talk.
- ▶ This is team's daily update meeting to the team.



Daily stand-up contd...

- ▶ Each team member answers the following three questions:

- **What have you done since yesterday?**
- **What are you planning to do today?**
- **Do you have any problems preventing you from accomplishing your goal?**



Role Play: Daily Standup Meeting

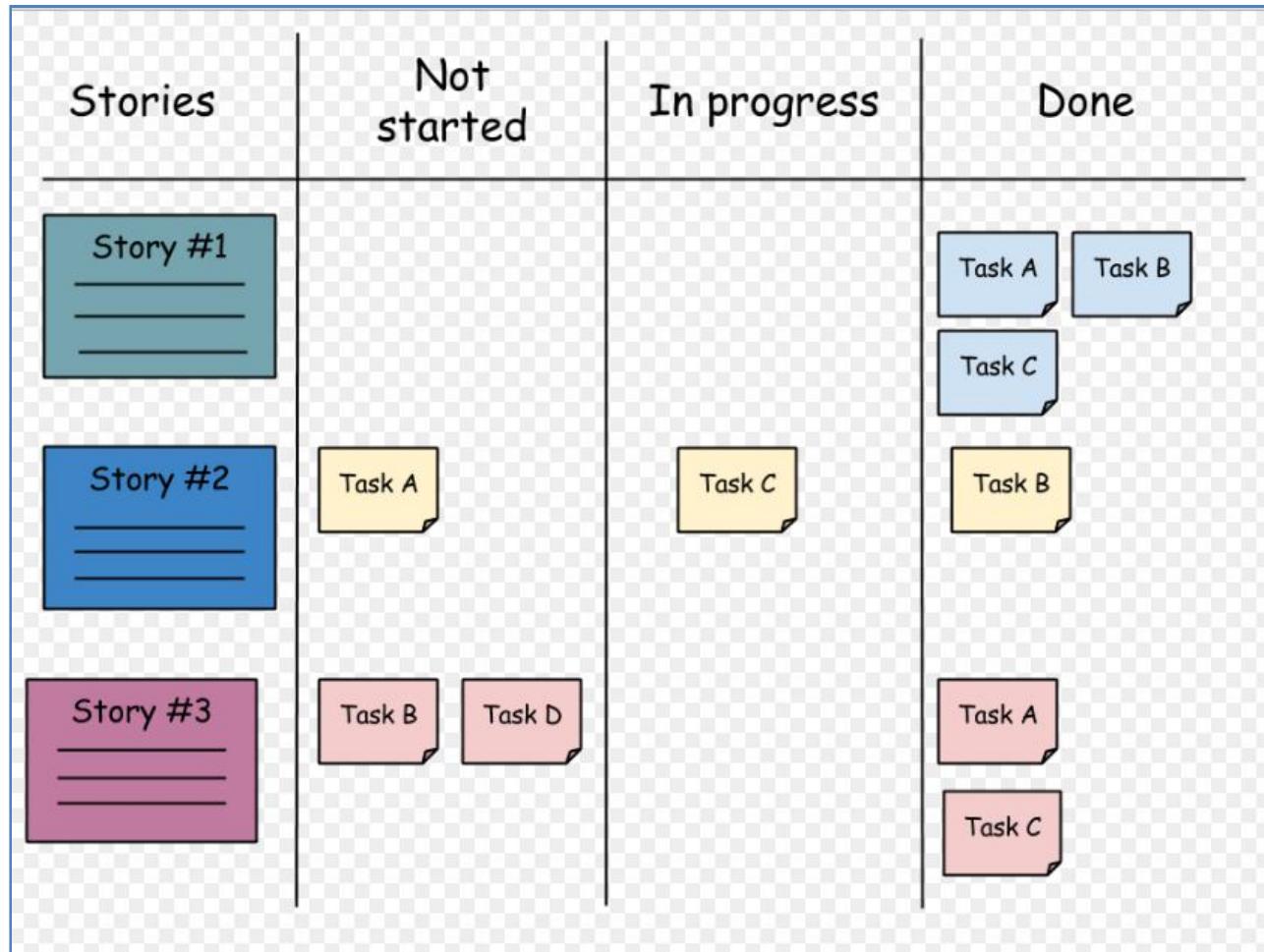
Perform a Daily Standup Meeting. Based upon the previously simulated workday, each team member answers three relevant questions:

What did I complete yesterday?

What will I complete today?

Are there any impediments?

Sprint board



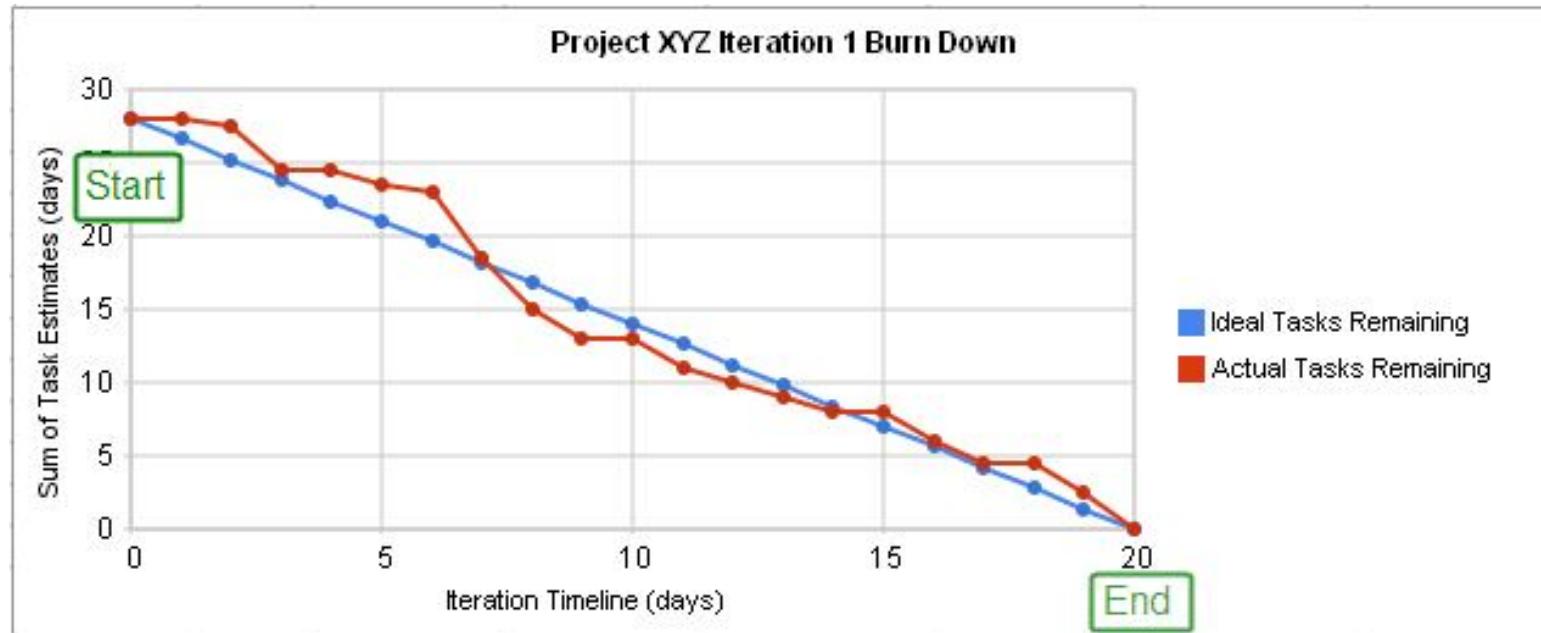
Burn down chart



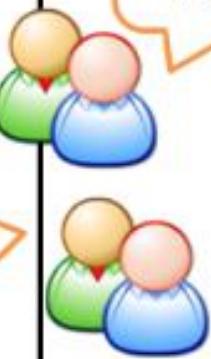
**Burndown/up
Charts**

Burn down chart contd...

- ▶ Work left to do versus time.
- ▶ Outstanding work on vertical axis and time left on horizontal axis.
- ▶ Easy way to find outstanding work.
- ▶ Helps in predicting when work will be completed.



Impediment Log

Issues / Impediments (3)		
Identified	In Progress	Resolved
<p>Recurring problem with X - need more tests</p> <p>Interface to payments engine keeps changing</p> <p>Users not available for demo's</p>	<p>We've hit our WIP limit on Issues! I'll work with the users to get them at the next demo.</p>  <p>We'll talk to payments team and agree some acceptance tests</p> 	<p>We'll focus on building more tests for X.</p> 

Shippable product



Finished Work

Shippable product

- ▶ At the end of sprint the team has to showcase working features.
- ▶ By shippable product it means a finished feature which can be shipped with minimal effort.
- ▶ Also known as Working software, Shippable feature, product increment.



Sprint Cancellation

- A Sprint can be cancelled before the Sprint time-box is over.
- A Sprint can be cancelled if the Sprint Goal becomes obsolete.
- Only the PO has the authority to cancel the Sprint
- Sprint cancellations are often traumatic to the Scrum Team
- When a Sprint is cancelled, any completed and “Done” Product Backlog items are reviewed. If part of the work is potentially releasable, the Product Owner typically accepts it. All incomplete Product Backlog Items are re-estimated and put back on the Product Backlog.



Sprint Review



Sprint Review

Sprint review contd..

- ▶ Held at the end to inspect the product increment.
- ▶ Chance to show the accomplishment of the scrum team.
- ▶ Informal demo of the completed features.
- ▶ No power point slides
- ▶ Time boxed to 15 mins to 2 hours.
- ▶ Customer gets a chance to see a ROI and give feedback.



Role Play: Sprint Review Meeting

Sprint Review Meetings are a part of the Demonstrate and Validate Process. At the end of the Sprint, demonstrate the items your team has completed in the preceding Sprint to the Product Owner (represented by the faculty or a member from another team or rest of all team members).

Sprint retrospective



Sprint retrospective contd...

- ▶ Three things are talked about
 - What went well?
 - What needs improvement?
 - Action items for improvements
- ▶ Facilitated by the scrum master.
- ▶ Attended by the team, scrum master and product owner.
- ▶ Time boxed to 2 hours (2 weeks sprint)



If the new ideas don't work throw it out and try something else.

Role Play: Sprint Retrospective Meeting

Complete a retrospective meeting of the preceding Sprint. Discuss what went well and what did not. Identify improvement opportunities.

Recap

Inputs from Executives,
Team, Stakeholders,
Customers, Users



Product Owner



Product Backlog

Team selects starting at top as much as it can commit to deliver by end of Sprint

Sprint Planning Meeting

Task Breakout

Sprint Backlog



Daily Scrum Meeting



Sprint Review



Finished Work

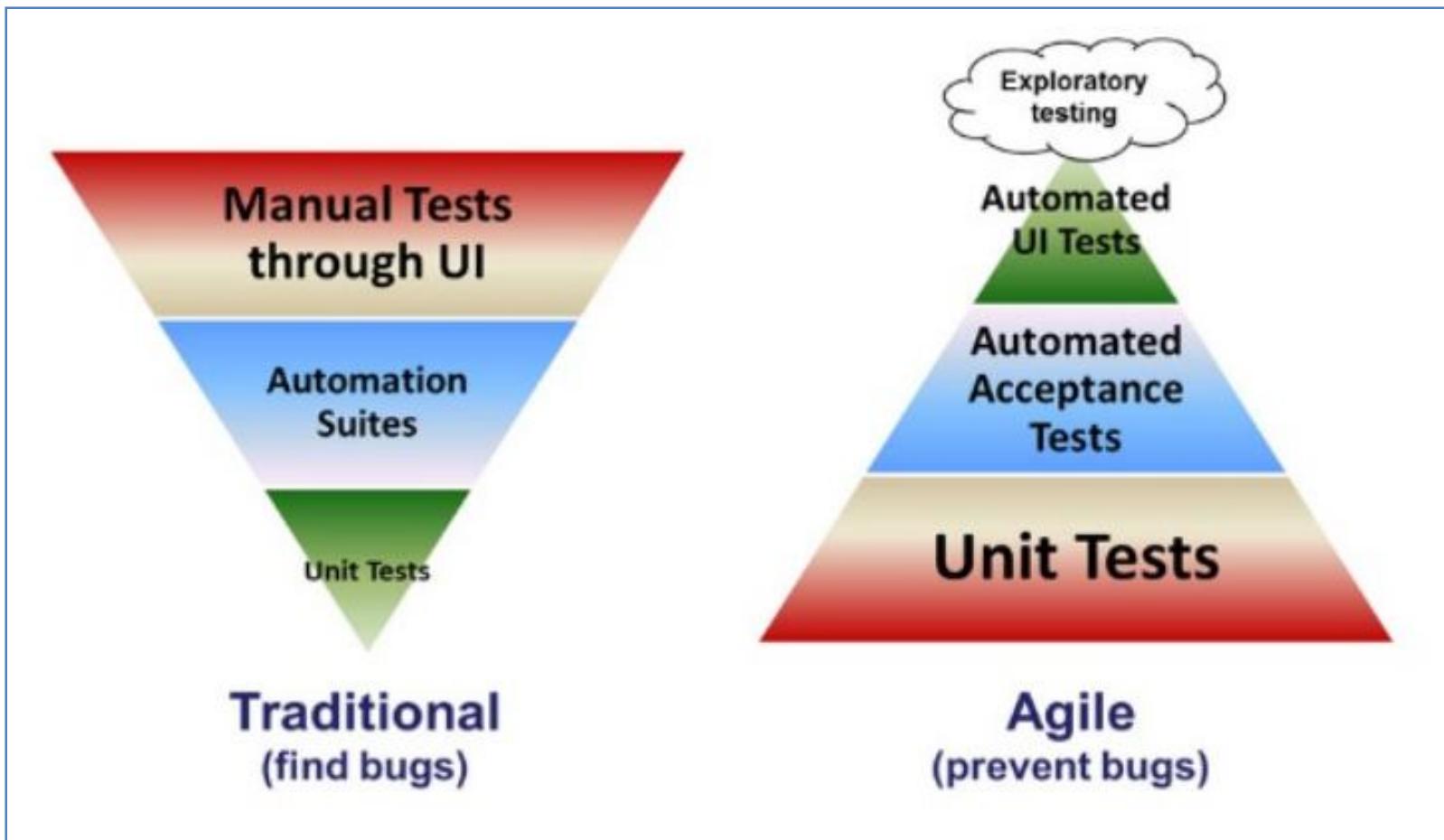
Sprint Retrospective

Agile Testing

Outline

- ▶ Traditional vs Agile Testing
- ▶ Agile Testing Concepts
- ▶ Exploratory Testing
- ▶ Agile Testing Approach
- ▶ Agile Testing Matrix
- ▶ Key success factors

Traditional vs Agile Testing

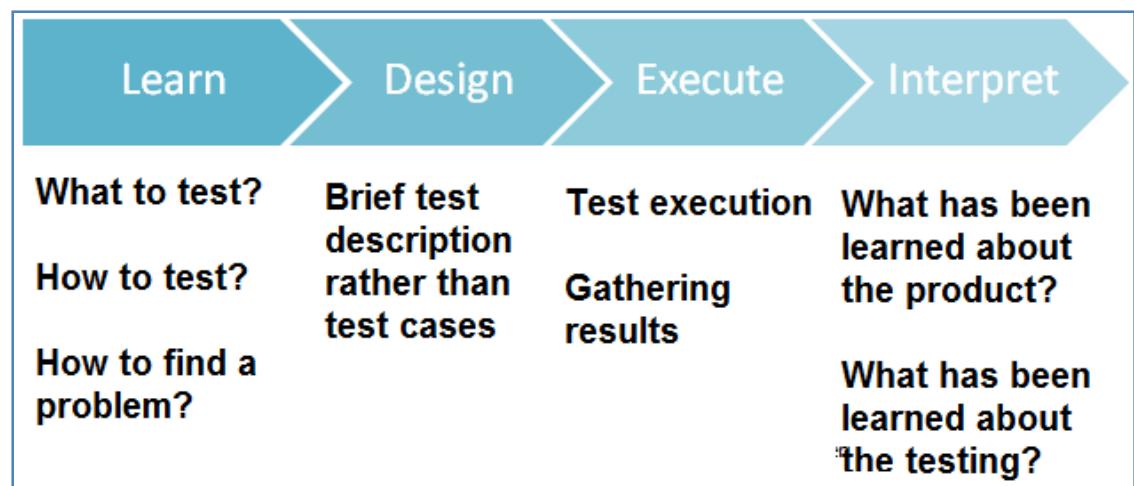
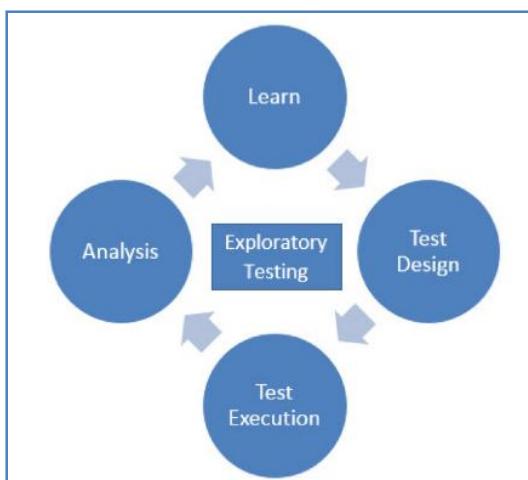


Agile Testing Concepts

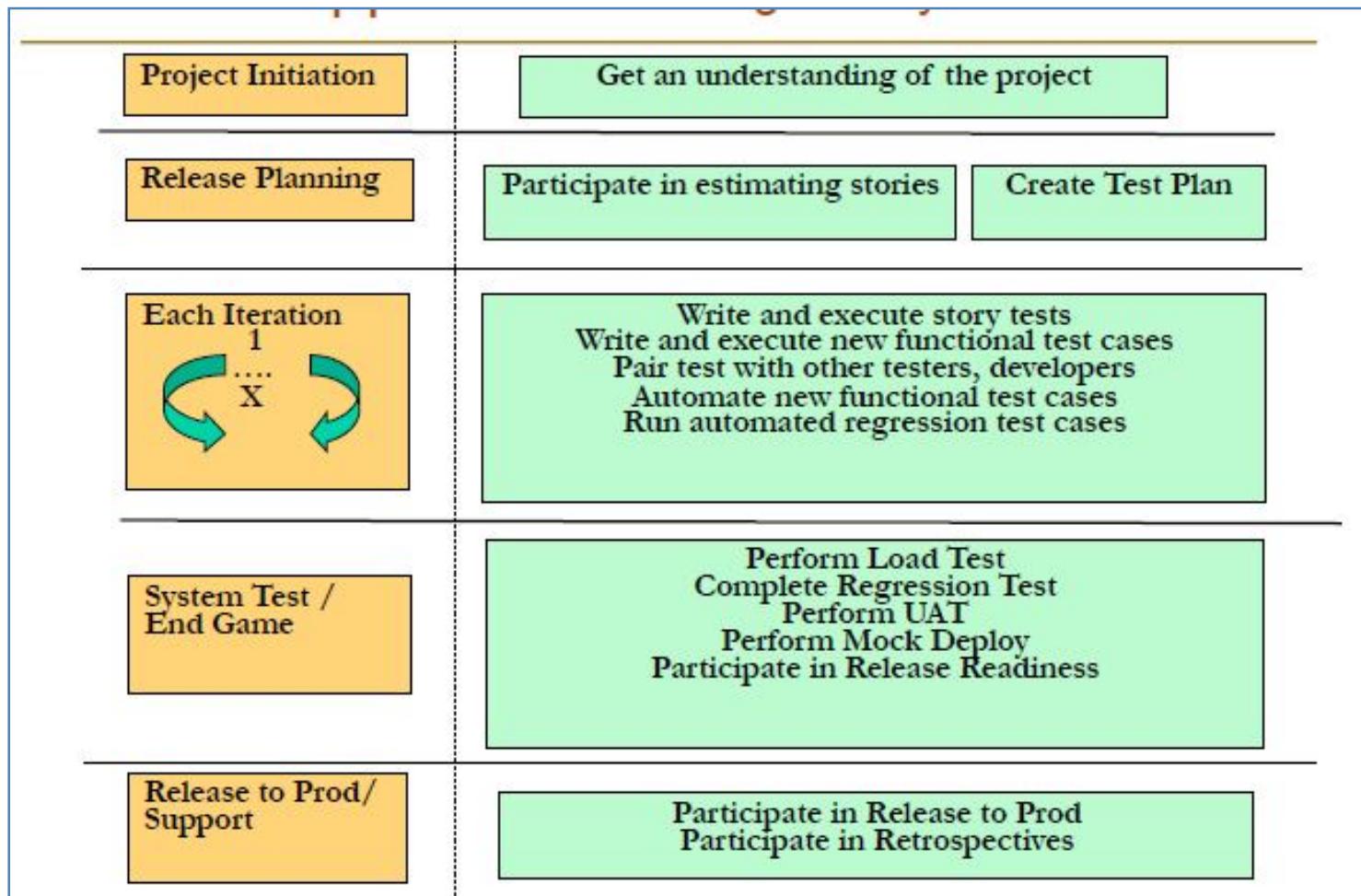
- Communication over Documentation
- Automation over Manual
- Exploratory over Structured Testing
- Testers as Quality Coach

Exploratory Testing

- Exploratory testing is a form of software testing in which the individual tester can design and run tests in a freer form
- Instead of following detailed test scripts, the tester explores the system under tests based on the user stories
- As the tester learns how the system behaves the tester can optimize the testing work and focus more on testing than documenting the test process

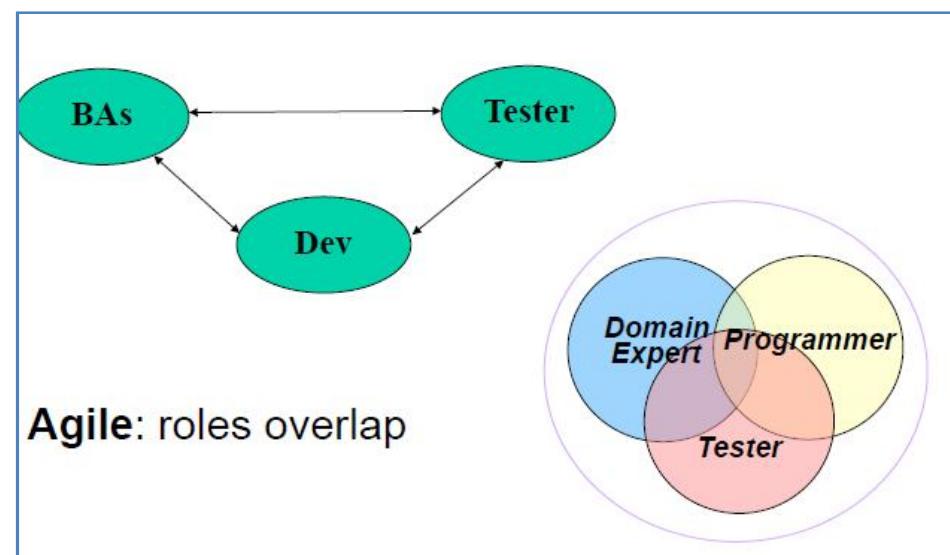


Agile Testing Approach



Agile Testing Approach

- Whole Team Approach
- Coding and testing are one process
- Feedback, collaboration key
- TDD/ATDD practices
- Test-infected developers, better tools, better-designed tests

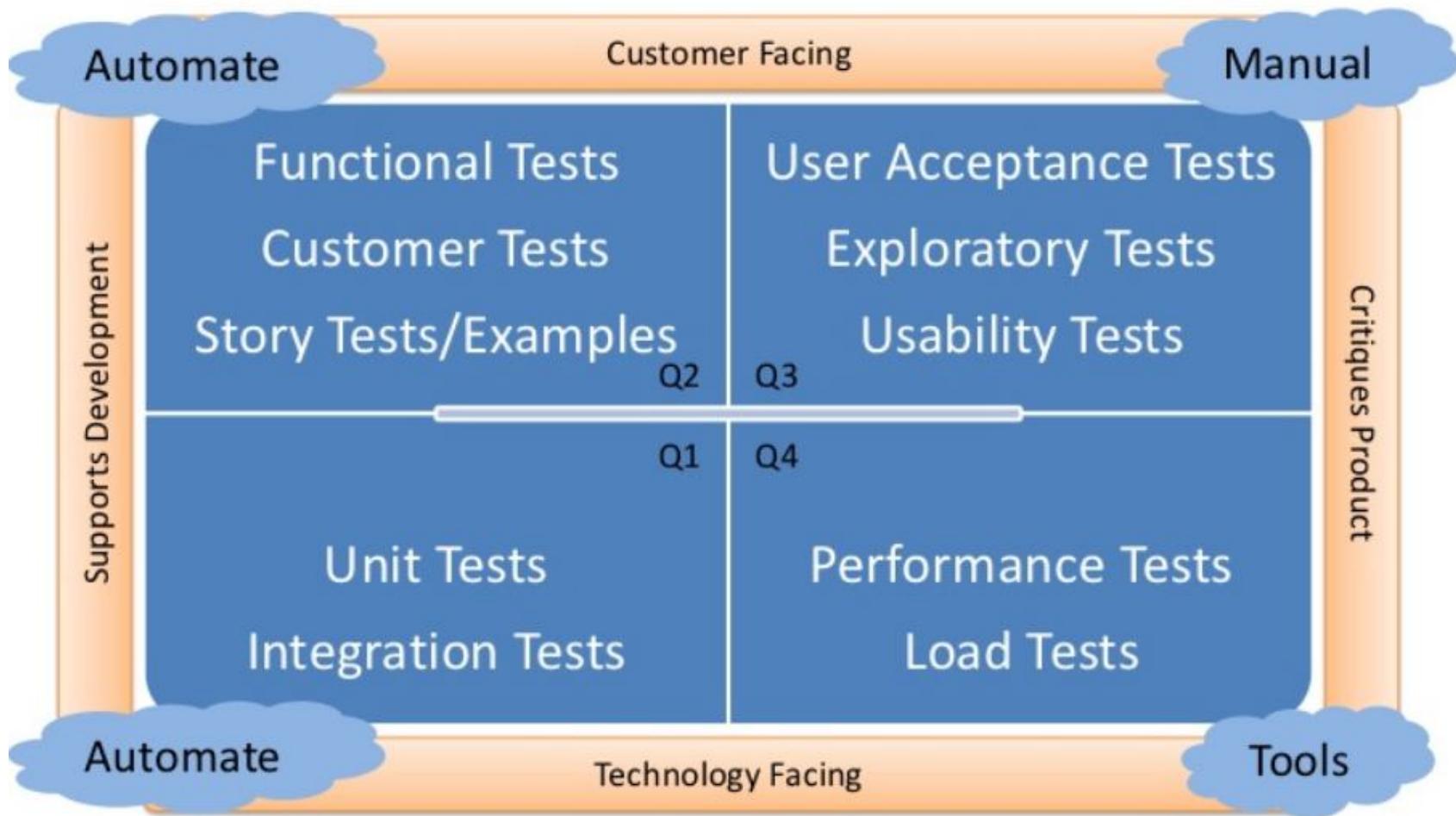


Agile Testing Practices

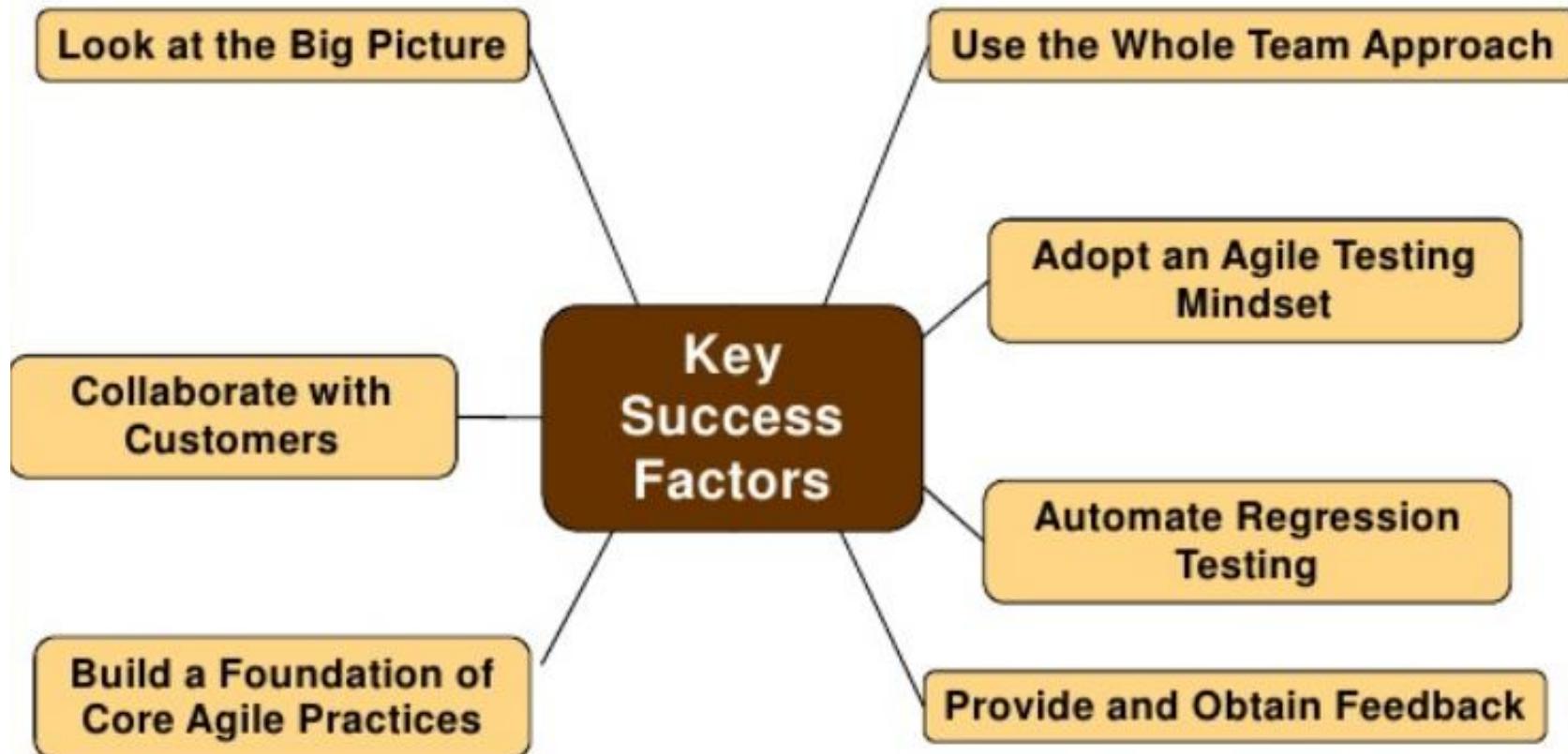
- Team writes acceptance tests together
- Programmers and testers pair to automate tests
- ROI guides automation
- Testers do exploratory testing
- Customers participate
 - Pair with customers for ET
 - Frequent demos



Agile Testing Matrix



Key Success Factors

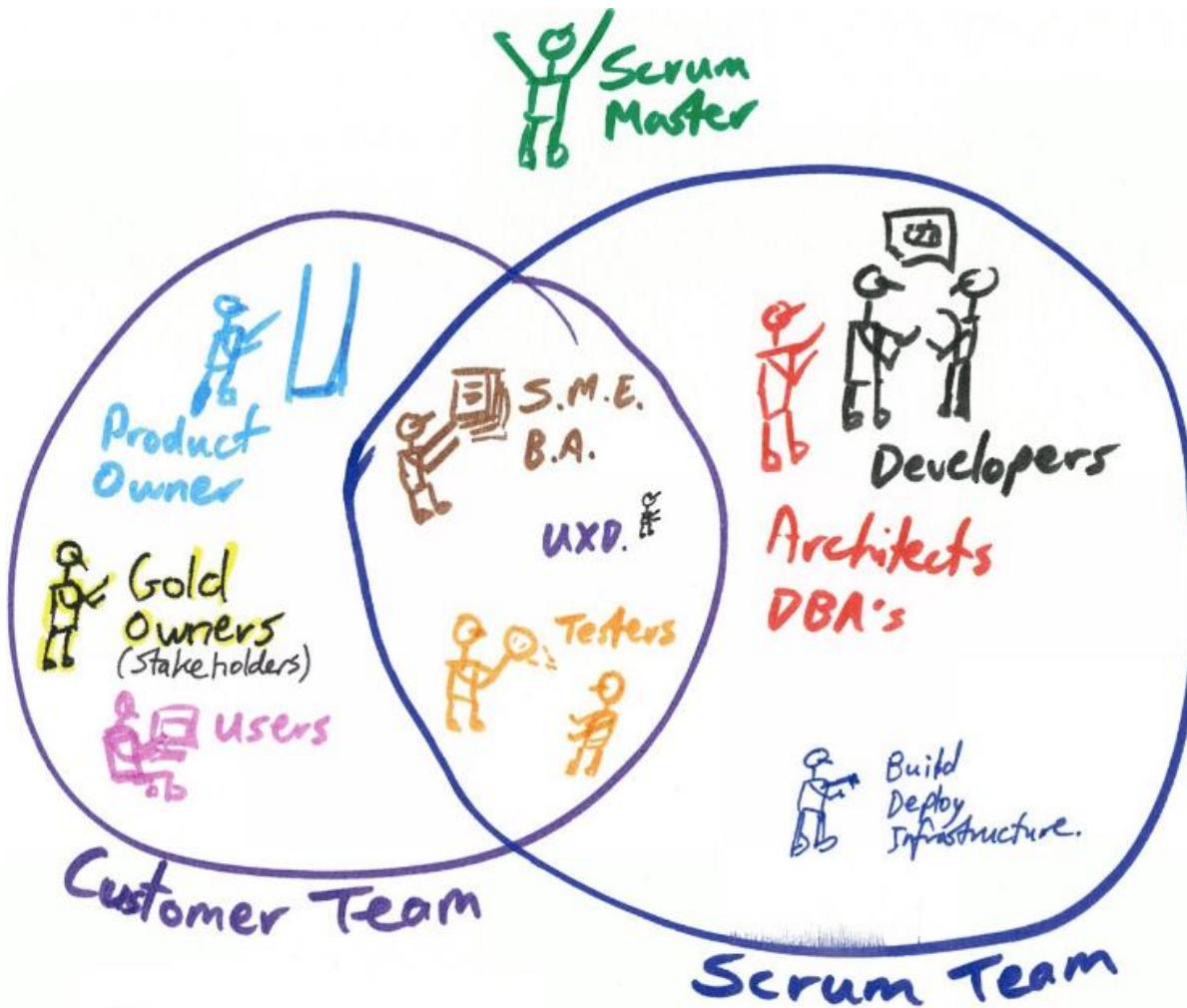


Scrum – Team Structure and Agile Culture

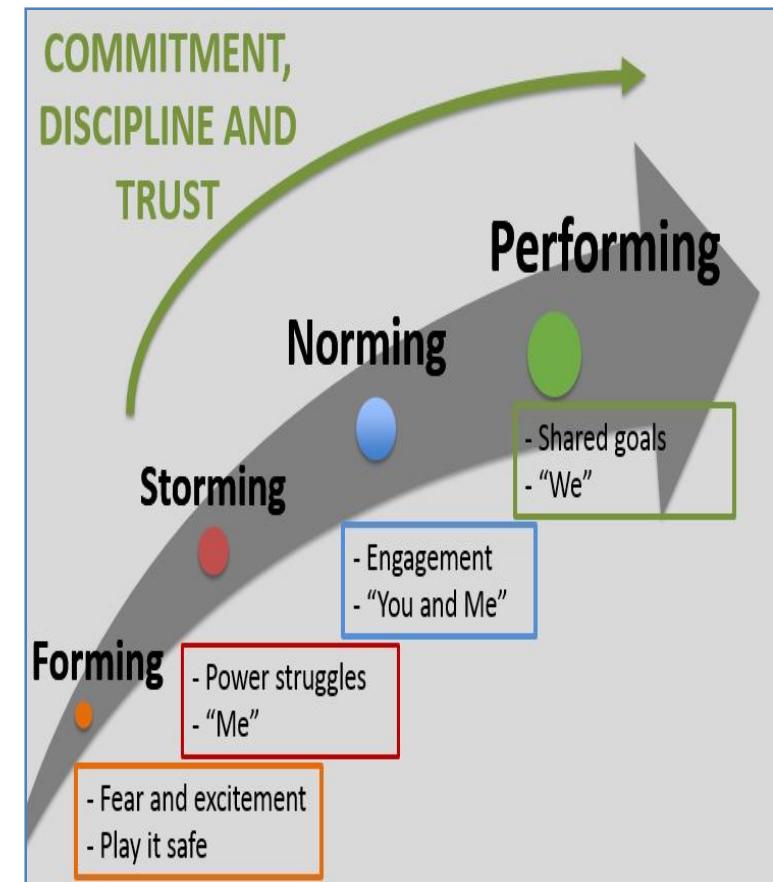
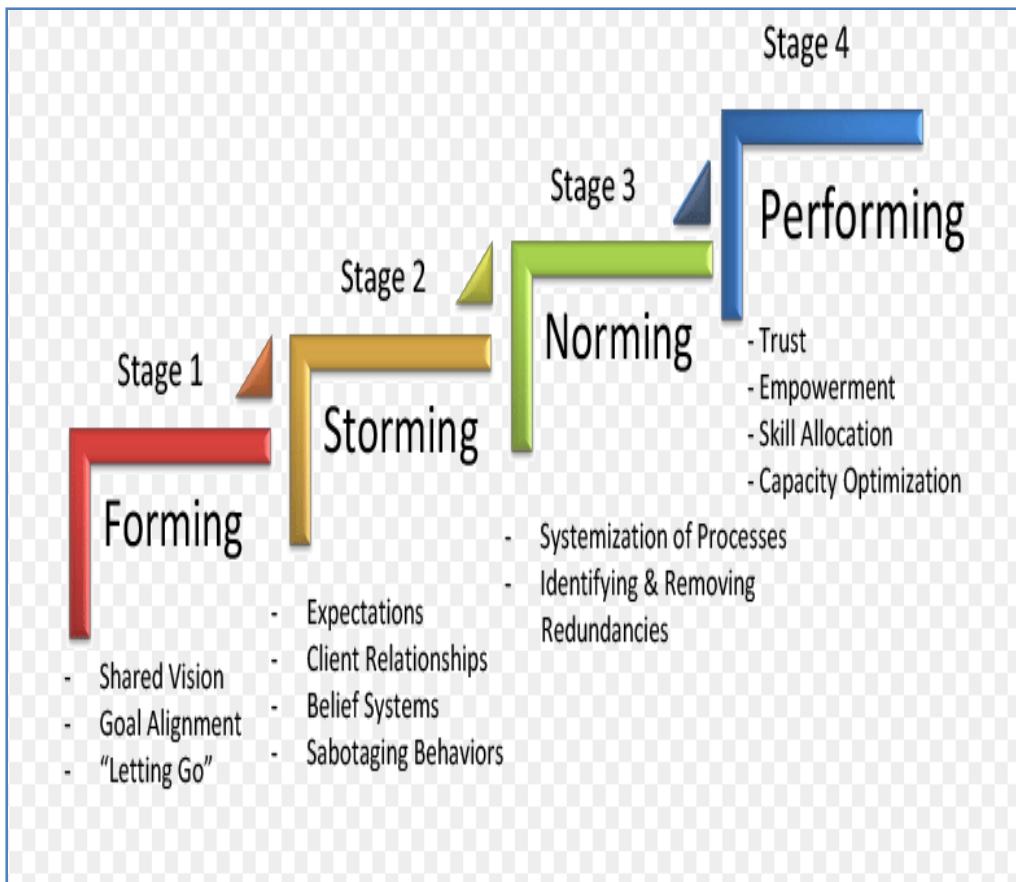
Outline

- ▶ Team Structure
- ▶ Think Holistically, Work Incrementally
- ▶ Communication Effectiveness

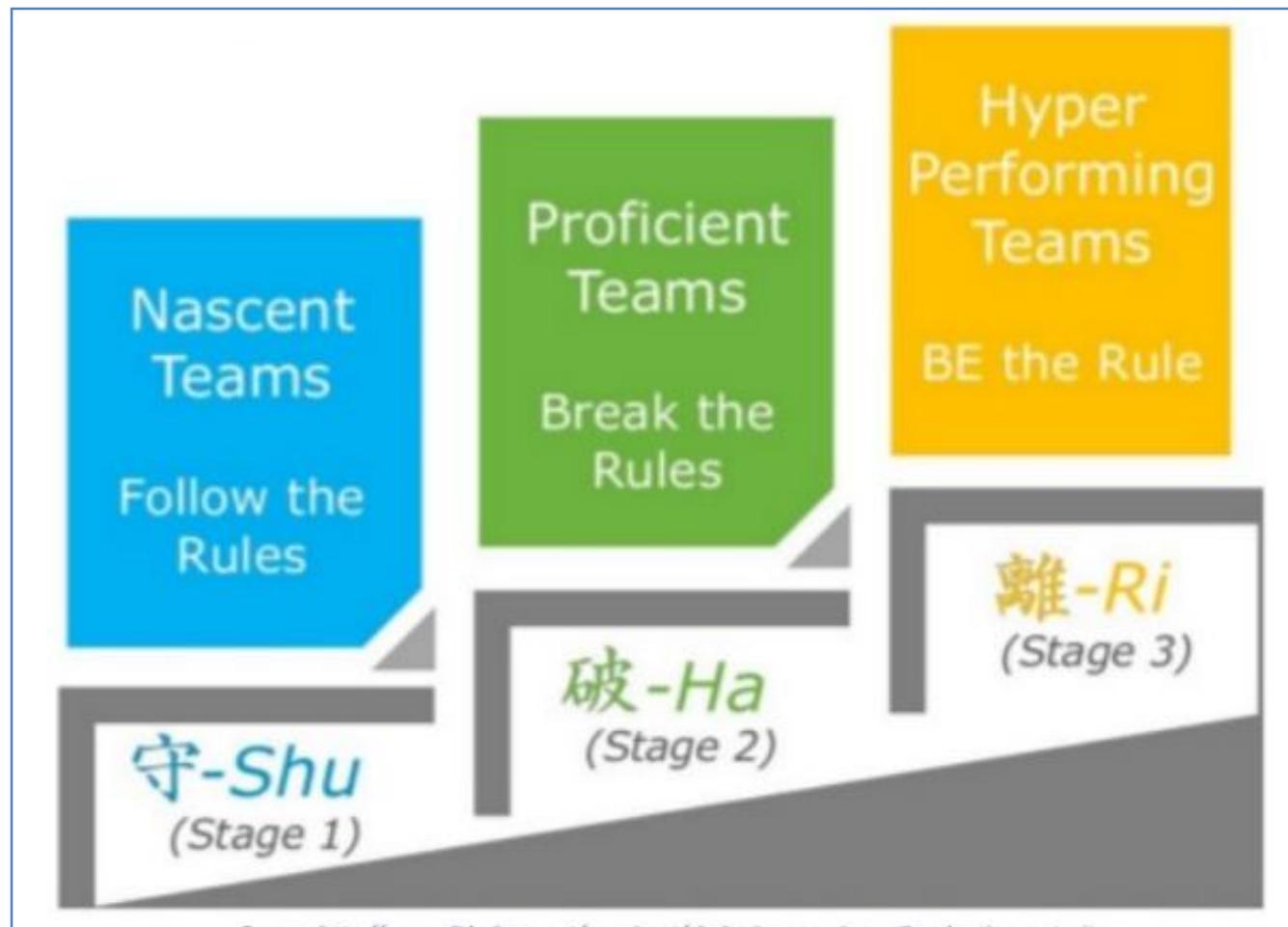
The Team Structure



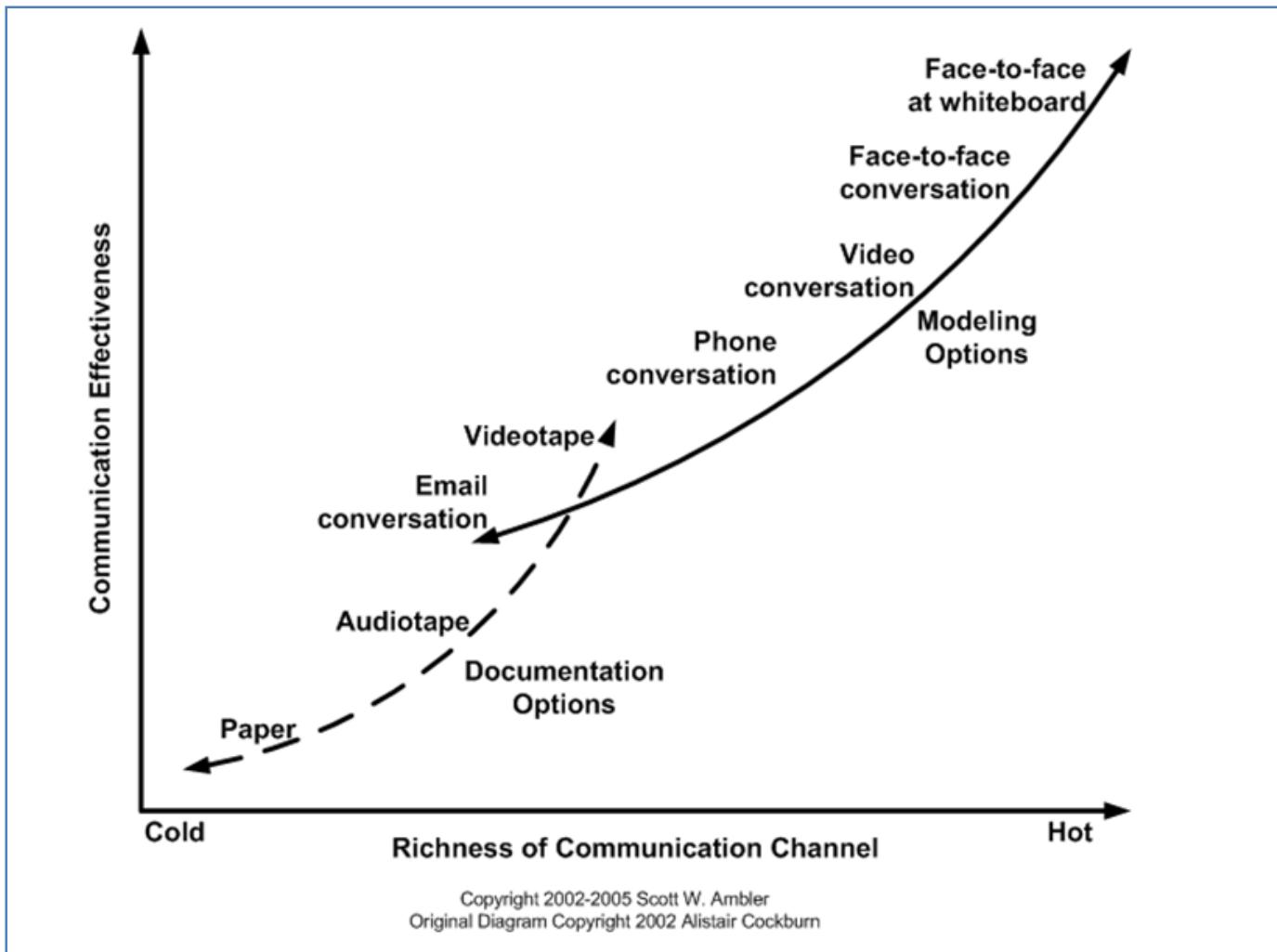
Team Formation Stages



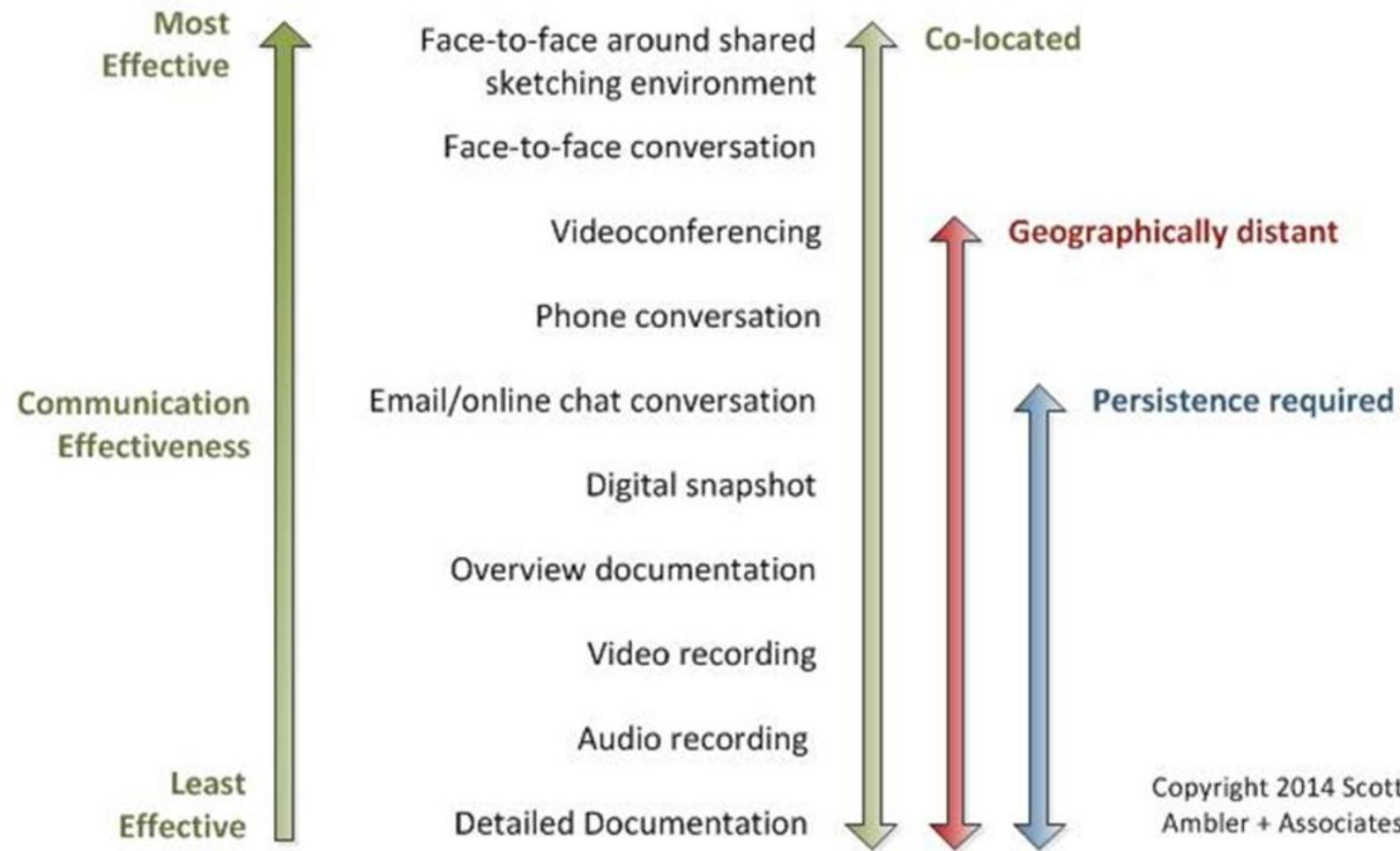
Team Evolution



Effective Communication



Effective Communication



Engineering Practices and Agile Tools

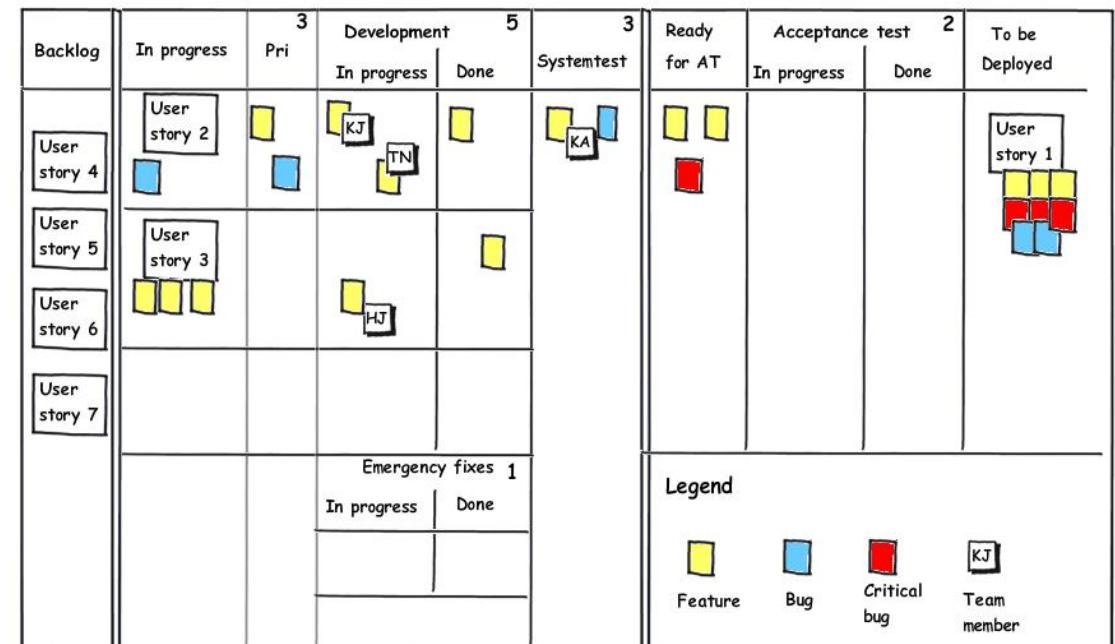
Outline

- ▶ Agile Methods and Engineering Practices
 - Kanban Overview
 - Extreme Programming Overview
 - Feature Driven Development (FDD)
 - Test Driven Development (TDD)
 - Behavior Driven Development (BDD)
 - Pair Programming
 - Refactoring
 - Continuous Integration / Deployment
- ▶ Agile Tools Overview

Kanban

- ▶ Emphasis is on JIT (Just In Time) delivery
- ▶ Visual process-management system that tells
 - What to produce?
 - When to produce?
 - How to produce?
- ▶ Again inspired from TPS
- ▶ Kanban in Japanese means “Signboard” or “Billboard”

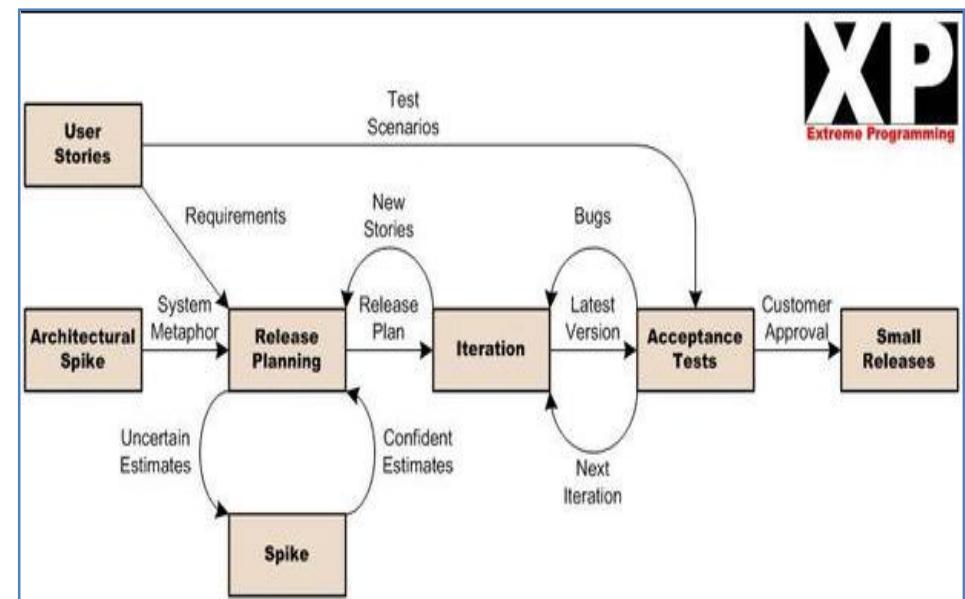
Kanban board



Extreme Programming (XP)

Traditional software engineering practices (daily interactions, working software, testing, etc.) taken to so-called "extreme" levels, leads to a process that is more responsive to customer needs ("agile"), while creating software of better quality

- ▶ Kent Beck
- ▶ Shorter life cycle (2 days - a week)
- ▶ On-Site customer
- ▶ Pair Programming
- ▶ TDD
- ▶ BDD
- ▶ Continuous integration

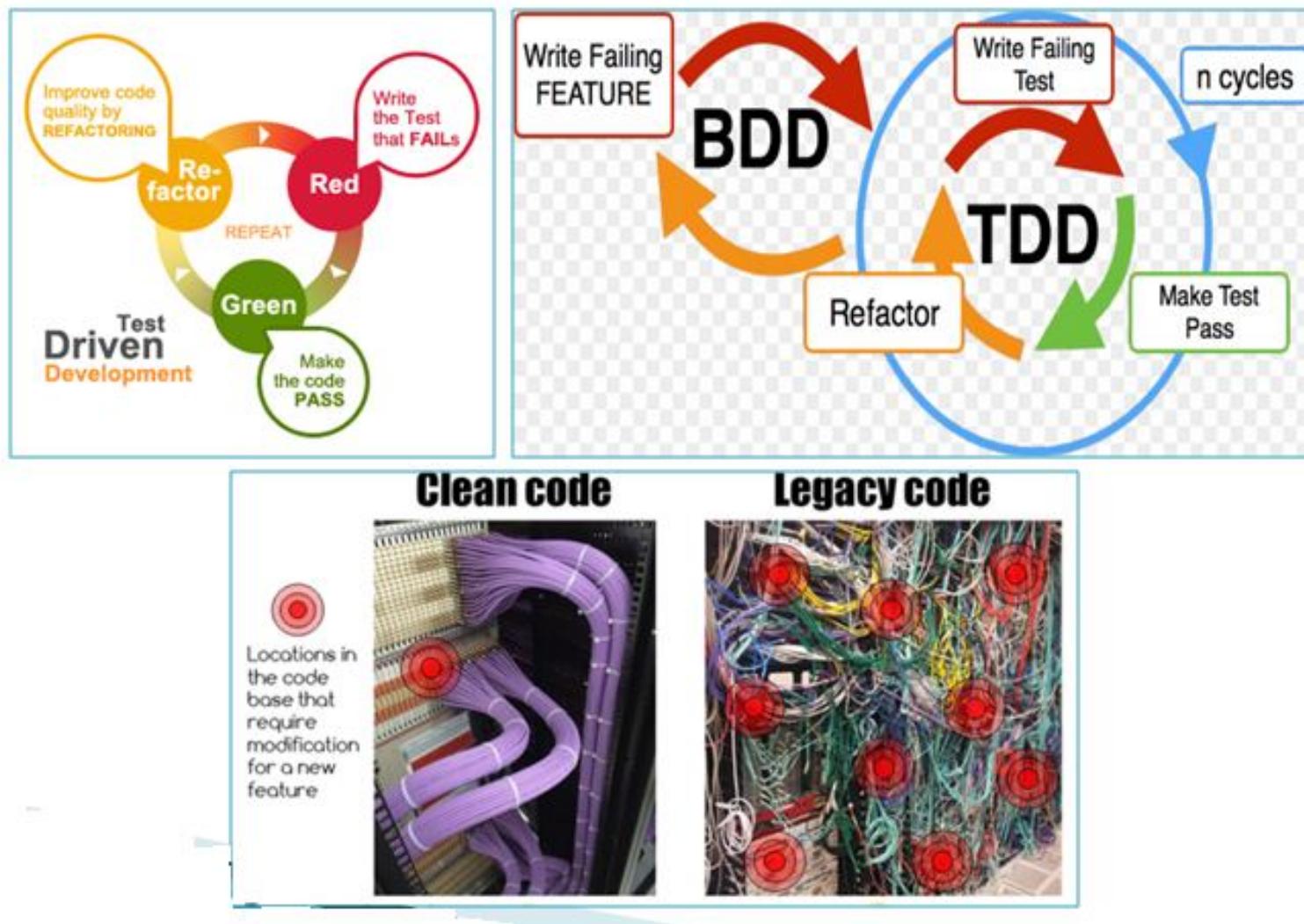


Lean Software Development (LSD)

- ▶ Adapted from Toyota Production System.
 - ▶ Value stream focussed
 - Select only the truly valuable features
 - Work product comes from customer request
 - ▶ Lean principles
 1. Eliminate waste (Muda)
 2. Amplify learning
 3. Decide as late as possible
 4. Deliver as fast as possible
 5. Empower the team
 6. Build integrity in
 7. See the whole



Engineering Practices

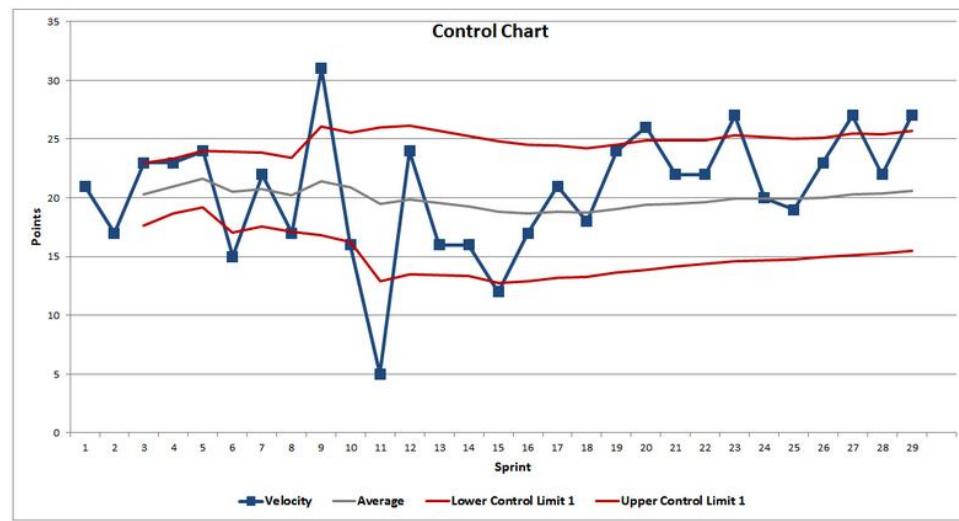
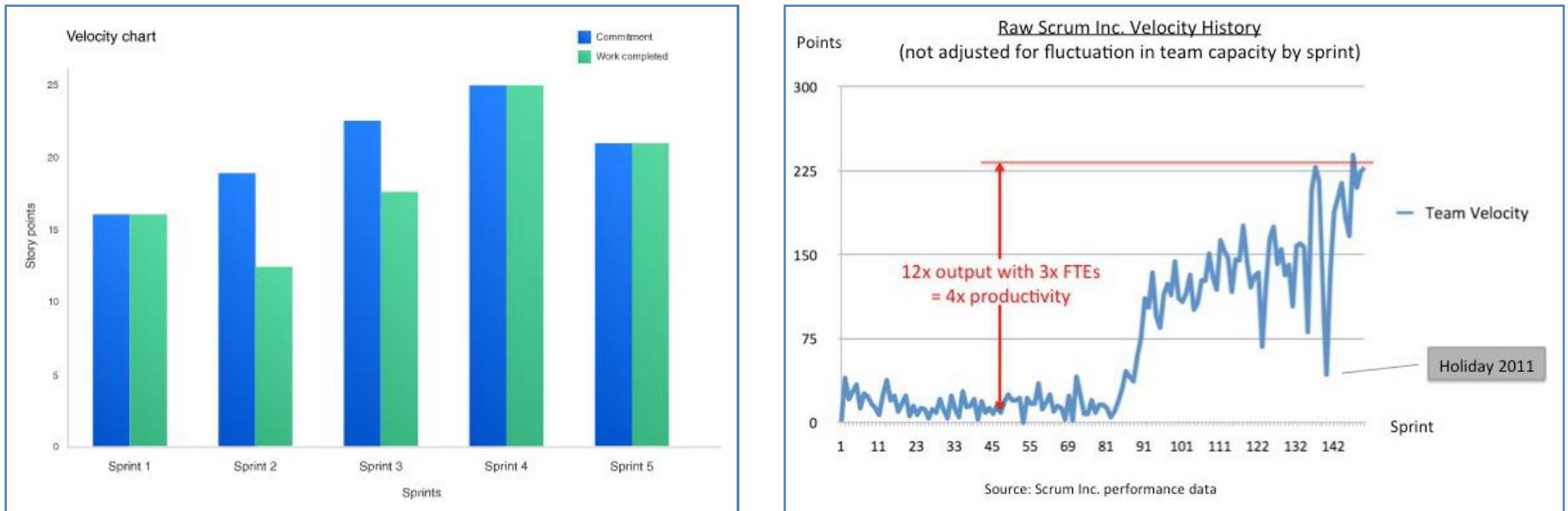


Agile Scrum Metrics and Scaling

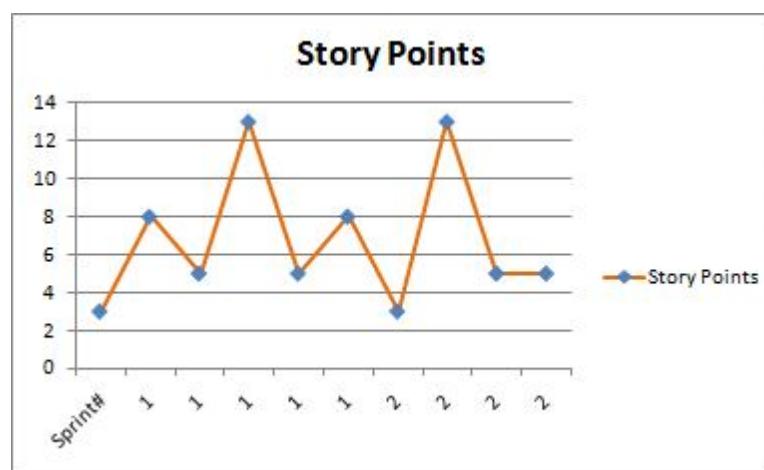
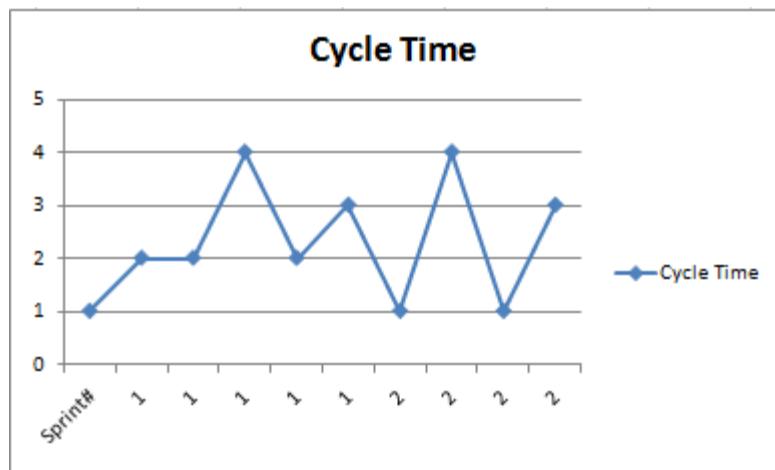
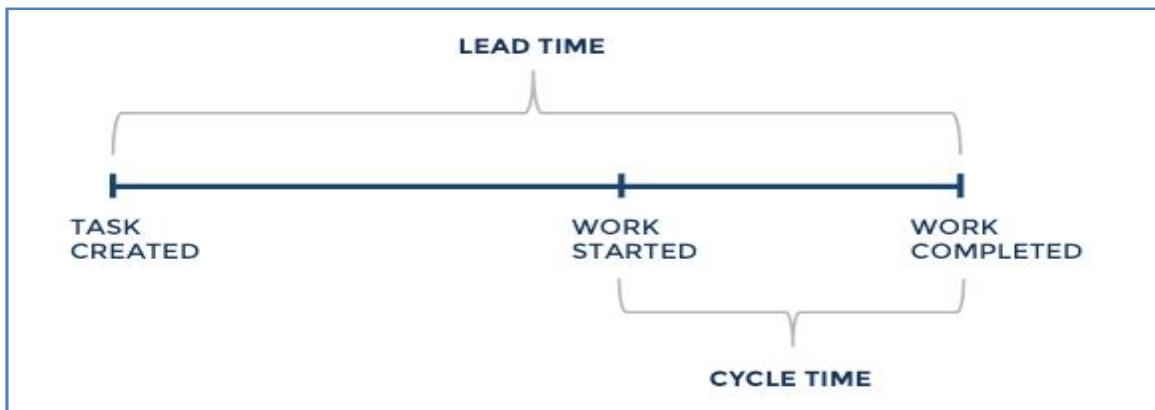
Outline

- ▶ Productivity Metrics
 - Velocity Chart
 - Cycle Time
- ▶ Quality Metrics
 - Defect Density
 - Defect Detection Percentage
- ▶ Scaling Agile Scrum

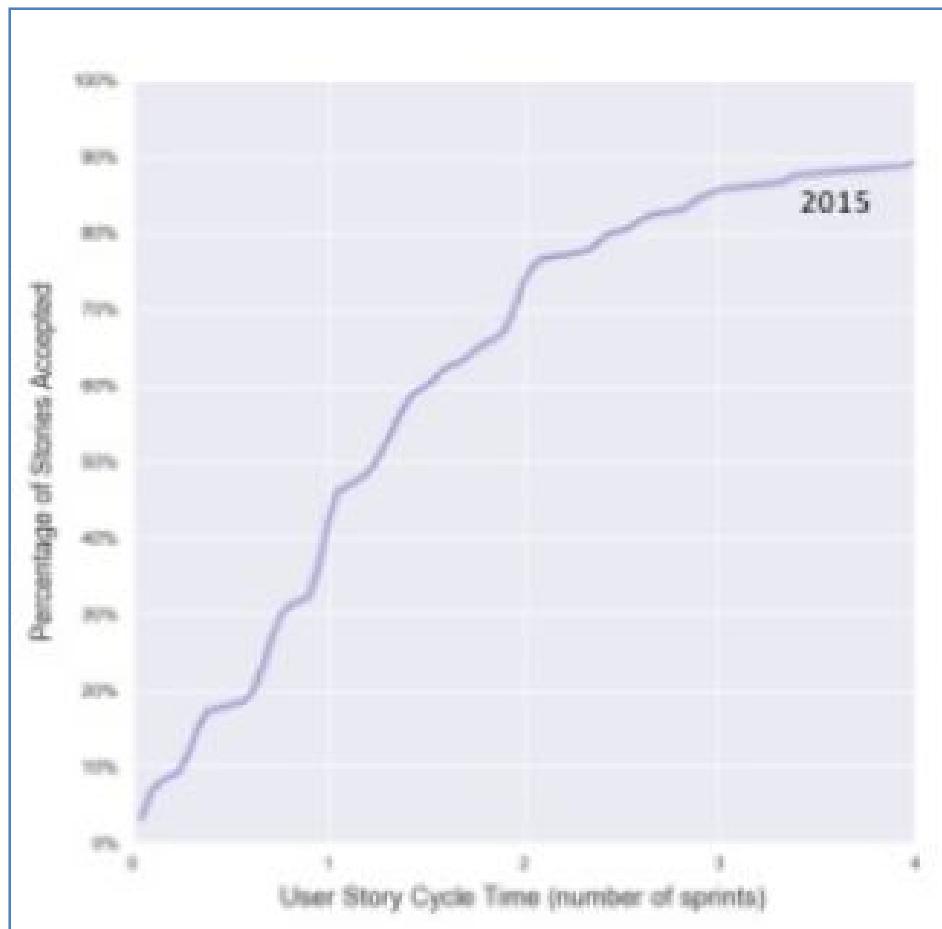
Velocity Chart



Cycle Time



Measuring Predictability



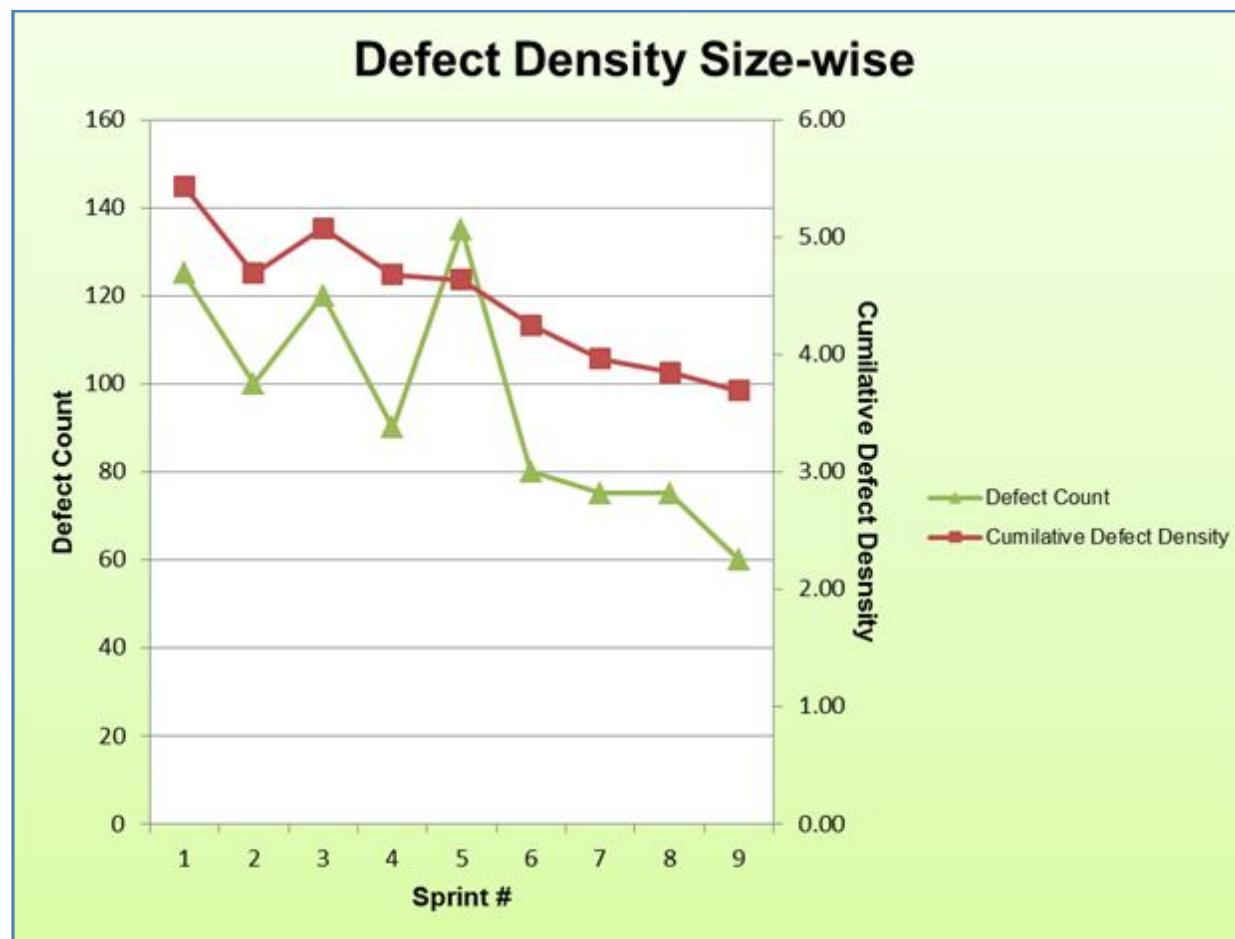
About the Chart:

- This chart shows the cumulative percentage of 2015 stories that were accepted over time.
- Note that the x-axis is in units of sprints rather than days.
- In the ideal, all stories that were committed to in a given sprint are completed within that sprint.
- At the 1 sprint line, the larger the percentage of completed stories the closer the teams are to the ideal.

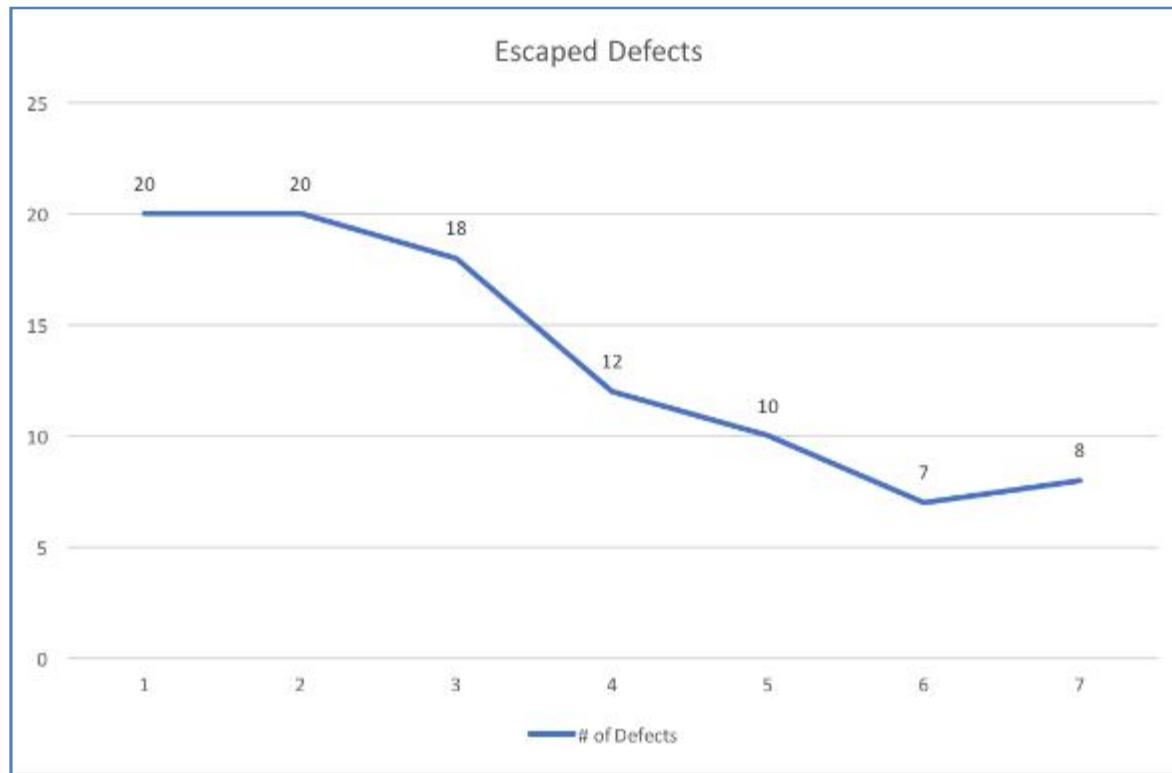
Of Note:

- Only about 43% of the user stories were completed within 1 sprint.
- After 2 sprints the completion percentage is at 74%.
 - This means that over one quarter of the user stories took longer than 2 sprints to complete
- Based on this data, **the likelihood that any particular story would be completed within one sprint is well short of 50-50.**

Quality Metrics – Defect Density



Quality Metrics – Escaped Defects

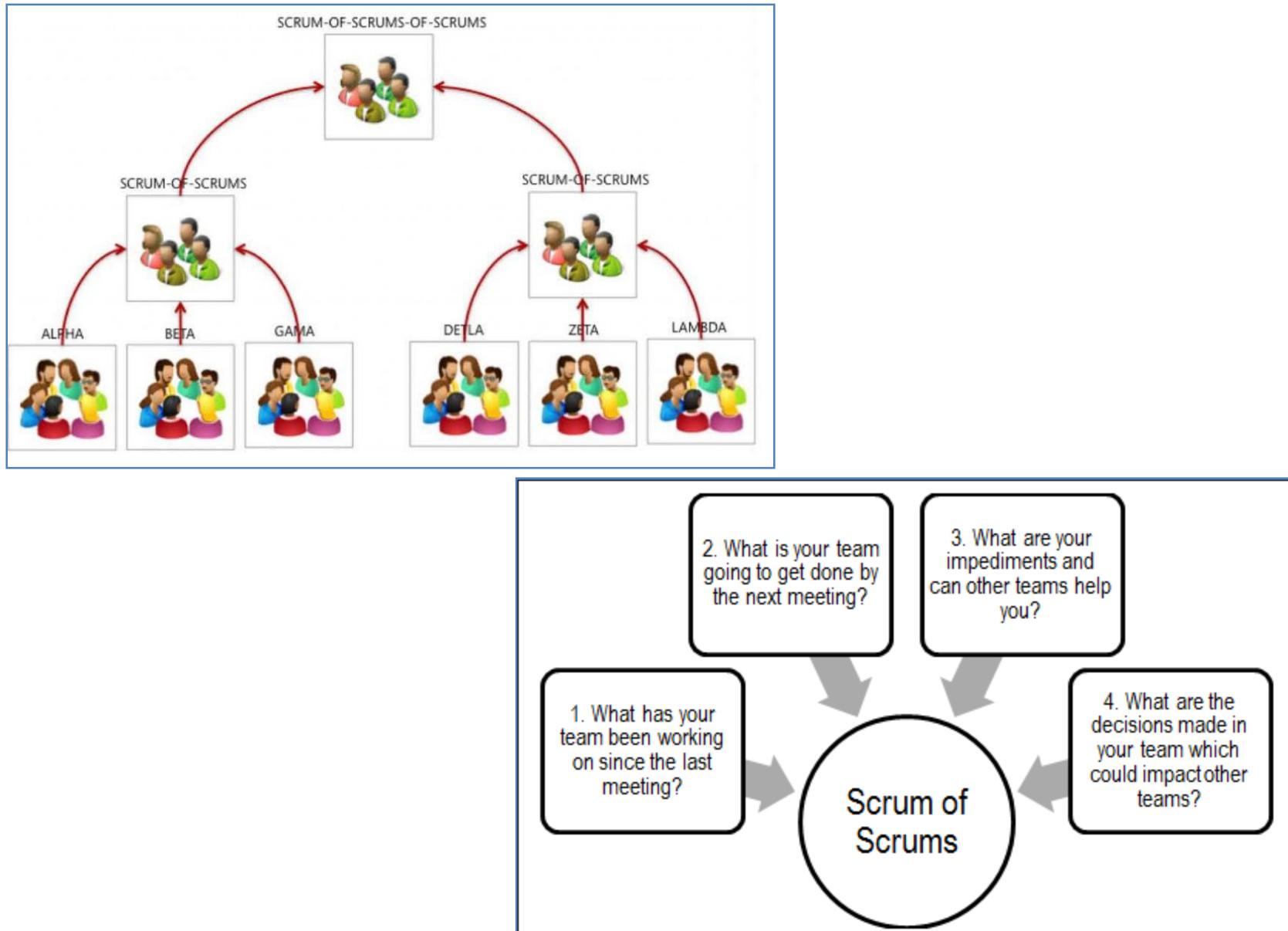


Quality Metrics – Defect Leakage Ratio

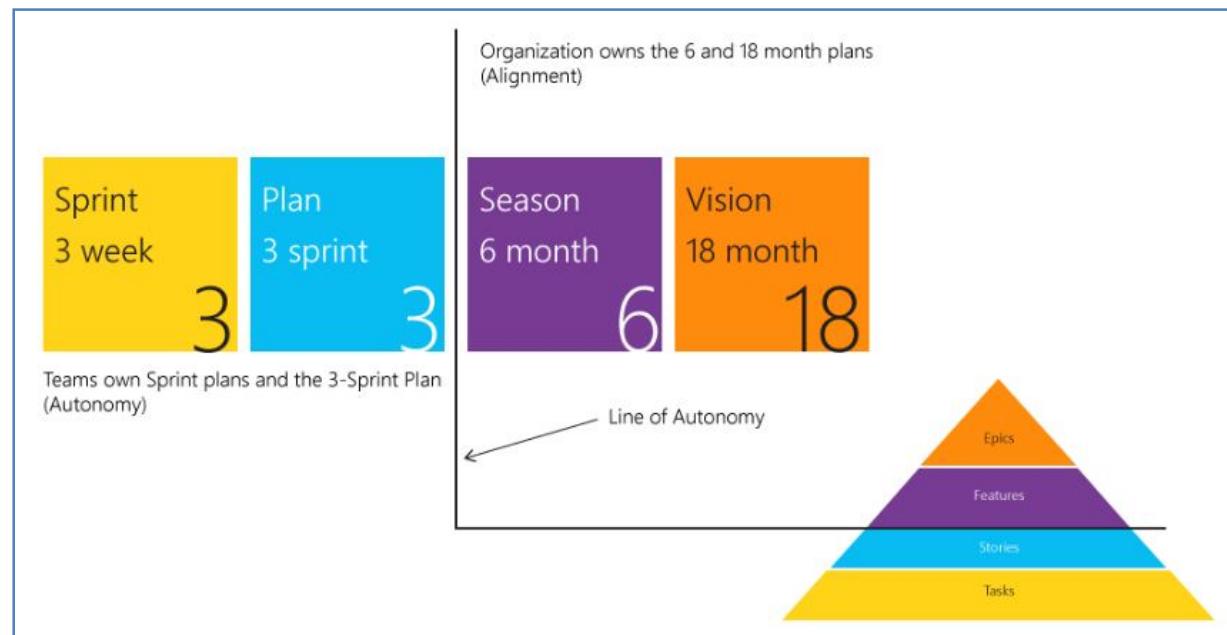
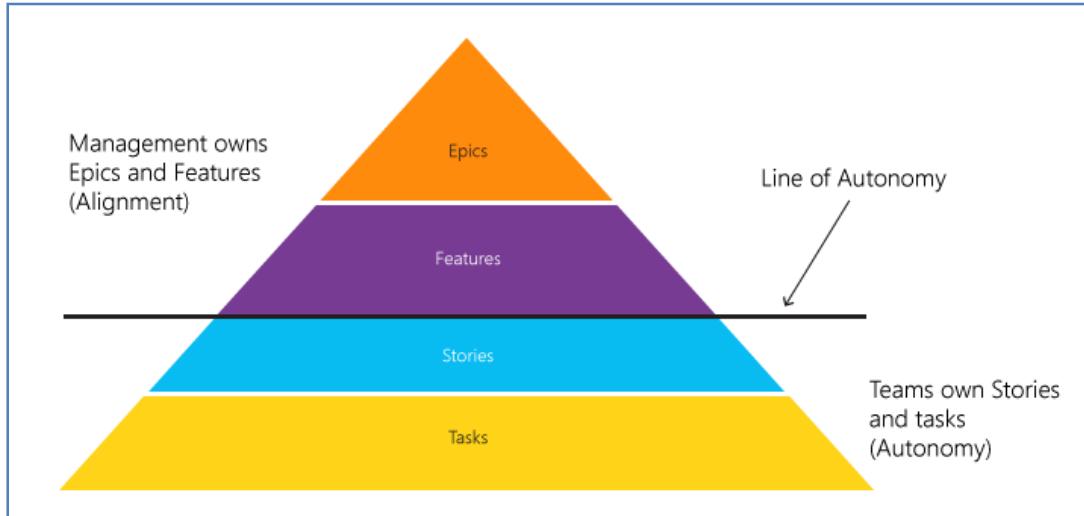
Defect Leakage Ratio =

No of External Defects / Total Defects (Internal + External)

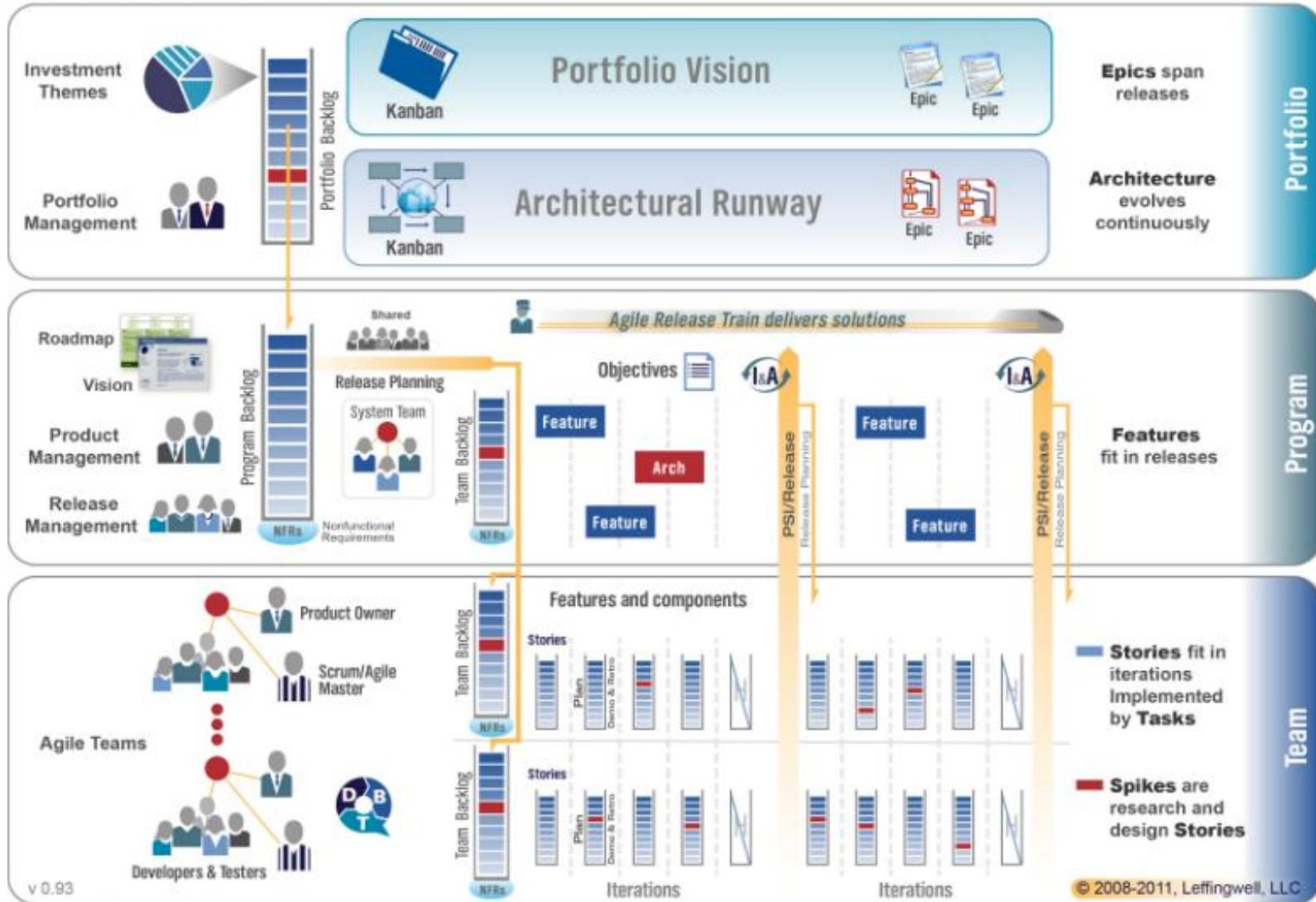
Scaling Agile Scrum – Scrum of Scrums



Scaling Agile Scrum – Autonomy and Alignment



Scaled Agile Framework™ Big Picture



See scalingsoftwareagilityblog.com and Leffingwell, D. *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*. Addison-Wesley (Pub. 2011)

Titbits

Top 5 Agile Techniques

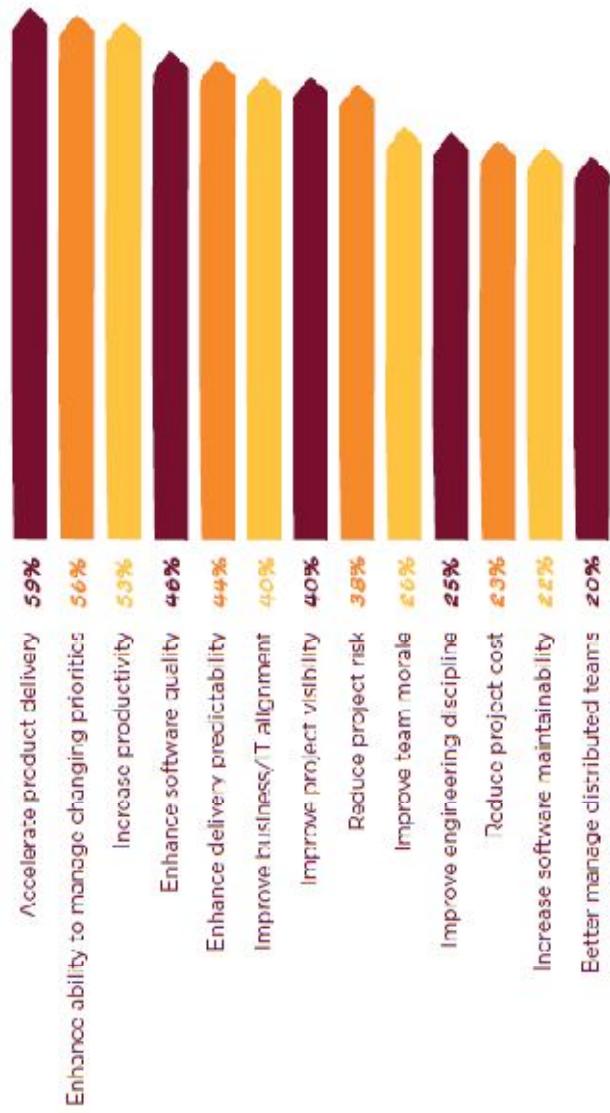


Percent of 100

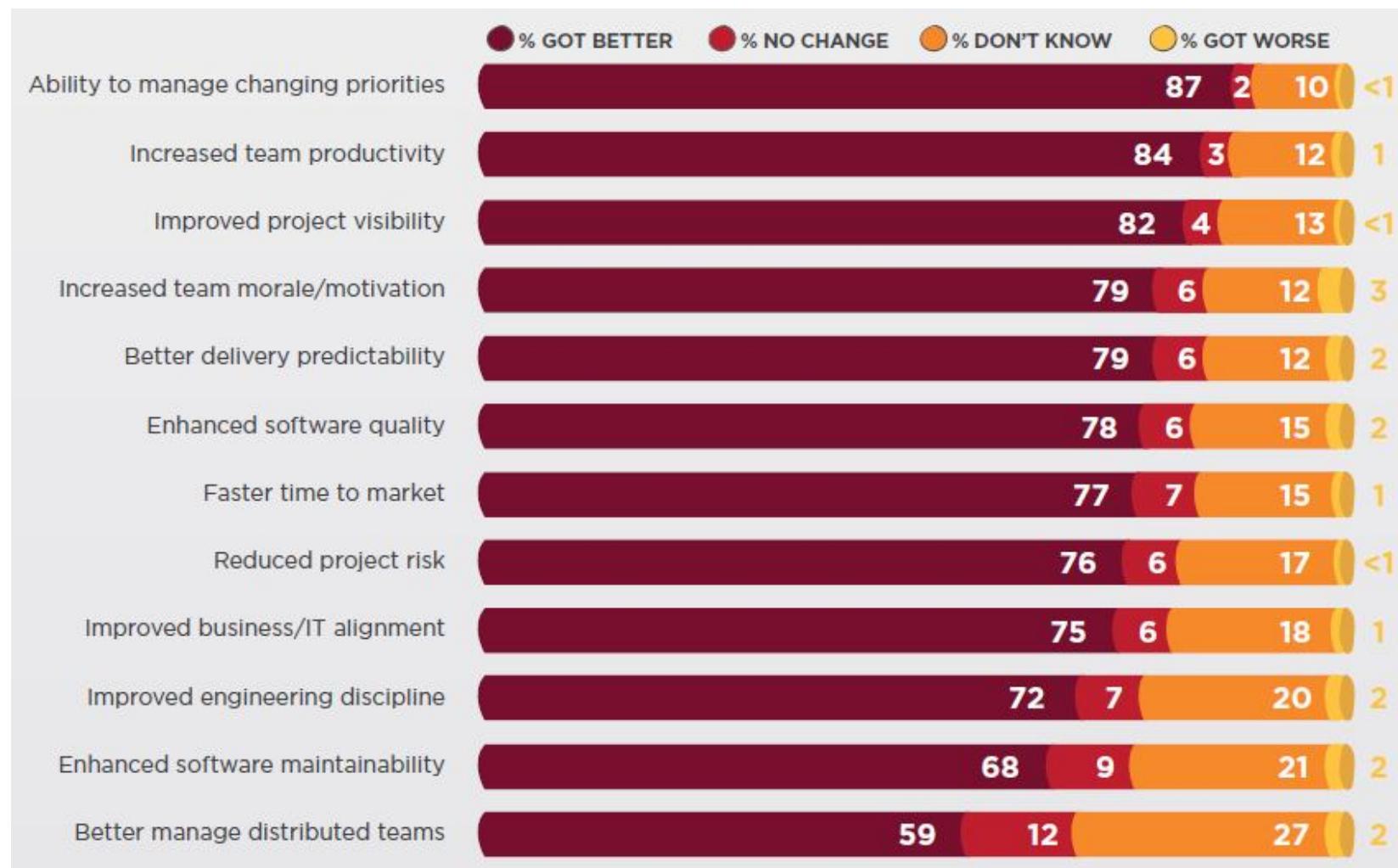
- 80%** Daily standup
- 79%** Short iterations
- 79%** Prioritized backlogs
- 71%** Iteration planning
- 69%** Retrospectives
- 65%** Release planning
- 65%** Unit testing
- 56%** Team-based estimation
- 53%** Iteration reviews
- 53%** Taskboard
- 50%** Continuous integration
- 48%** Dedicated product owner
- 46%** Single team (integrated dev & testing)
- 43%** Coding standards

- 38%** Open work area
- 36%** Refactoring
- 34%** Test-Driven Development (TDD)
- 31%** Kanban
- 29%** Story mapping
- 27%** Collective code ownership
- 24%** Automated acceptance testing
- 24%** Continuous deployment
- 21%** Pair programming
- 13%** Agile games
- 9%** Behavior-Driven Development (BDD)

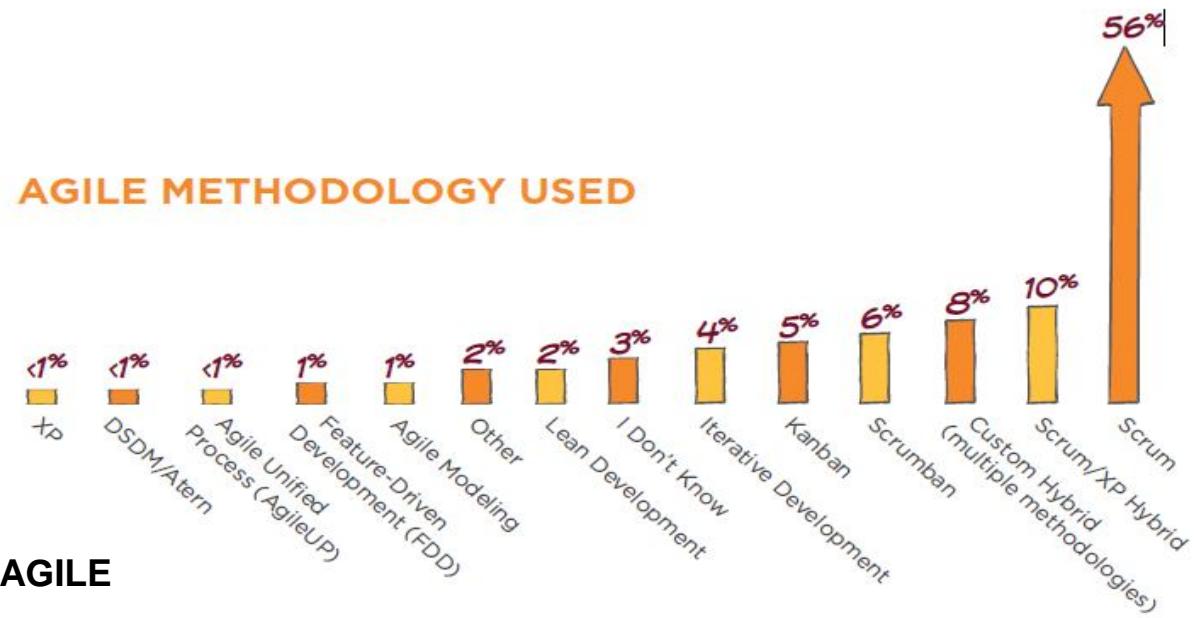
REASONS FOR ADOPTING AGILE



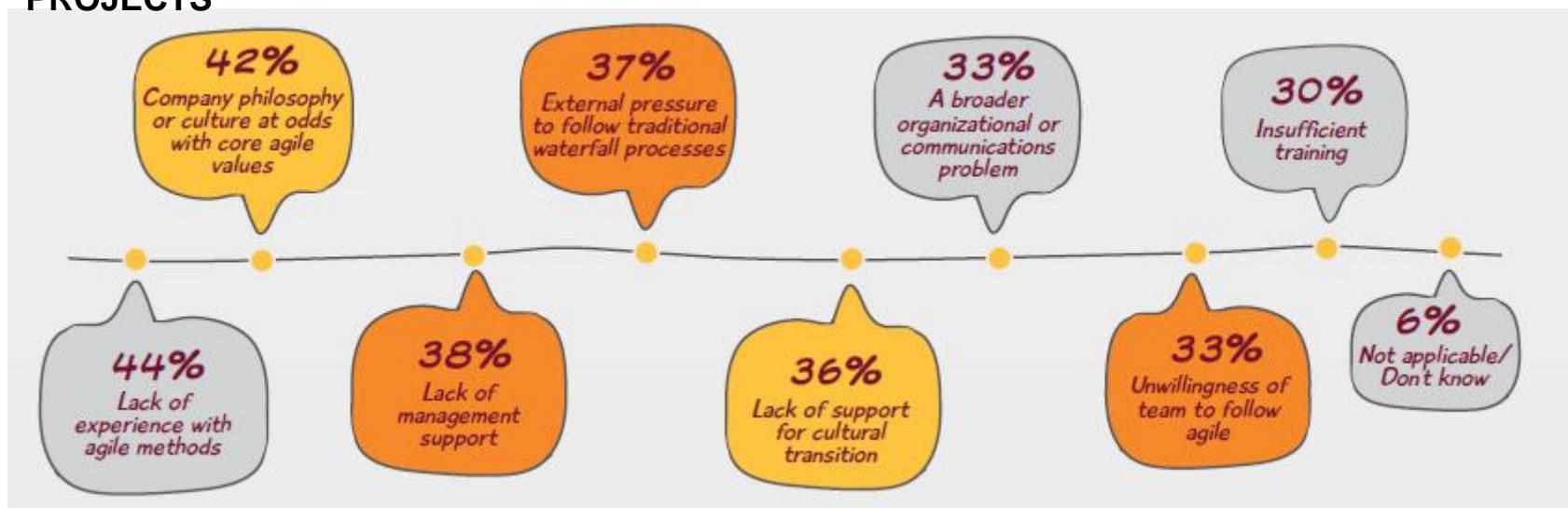
IMPROVEMENTS FROM IMPLEMENTING AGILE



Continued



LEADING CAUSES OF FAILED AGILE PROJECTS



Thank You!