Agile _ SCRUM Project Management

Presenter Name : **Dipti Trivedi**

Presentation Date : 26-08-20

Your Facilitator

Dipti Trivedi

PMI ACP, PMI- USA
ICP-ACC **[Certified Agile Coach]**
SAFe Agilist [SA]
SCRUM Master Certified
PMP, PMI, USA
Microsoft Certified Professional - MCTS
[Project SERVER]

Overall **18+ yrs** Industry experience

Conducted **30,000+ hours** training on **Project Management & PM Tools**

Trained 7000+ professional across the globe.

Group Manager, Cybage-Pune

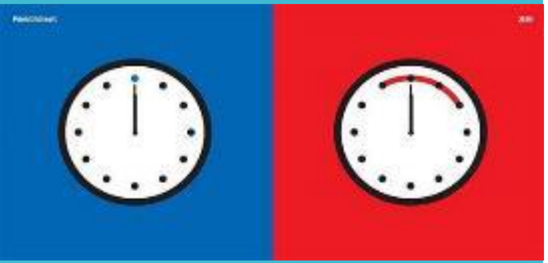
Sr. Consultant, SABCONS Software Consultant- REP, PMI-USA

Lead-Technical Services at Softbridge Solutions Pvt. Ltd.

Project Manager (IT) at AT&T Technology Park and

Ex- PMI India Champion & Associated with PMI Pune Deccan India Chapter

Our Expectations

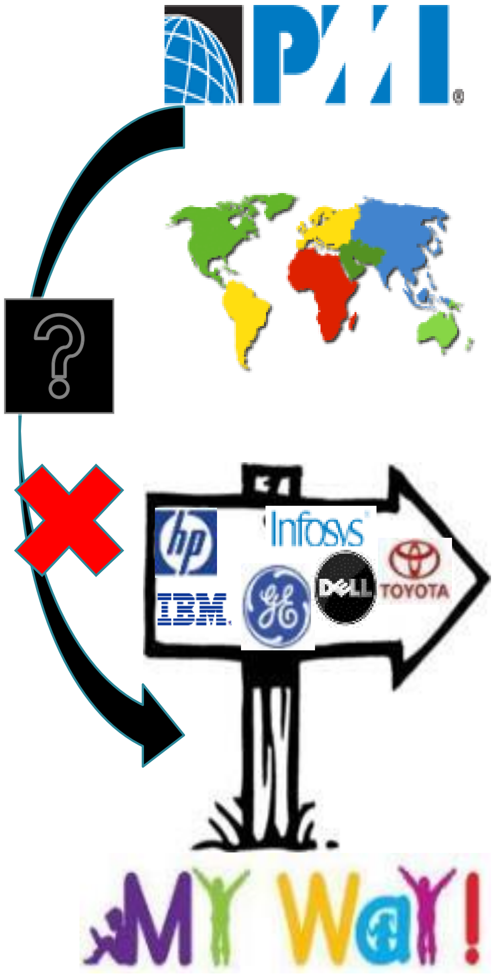


Punctuality: Start on time, end on time is the best policy. Concentration starts dropping after sunset.

Cell-phones/Laptops: Unwelcome in the Training Room. One of the Trainers in Bangalore is a very successful entrepreneur - if he can activate his voice-mail, so can the participants.

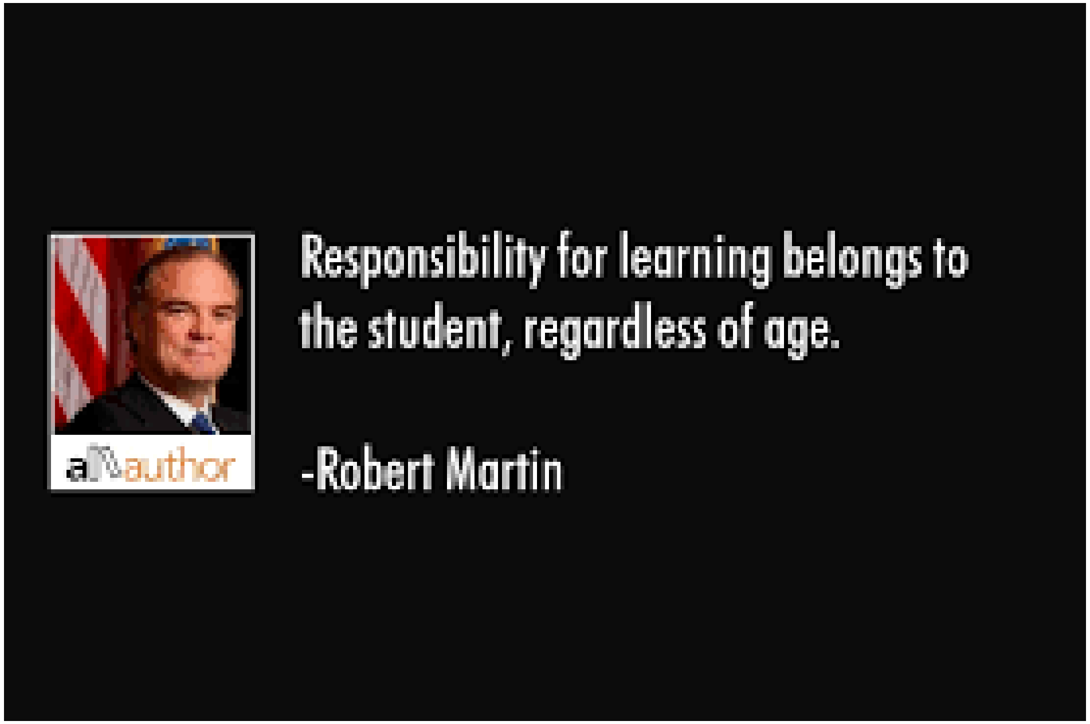
Unsolicited visitors: Unwelcome in the Training Room.

Expectation: The objective of a Training Program is not to give silver bullets but to create an awareness and motivate participants to think differently - they still have to do the thinking!



CONVINCE ME...

The facilitation alone is my responsibility, the learning is all YOURS….!



Responsibility for learning belongs to the student, regardless of age.

-Robert Martin

Have Question…

Anytime… ☺

Agenda Topics

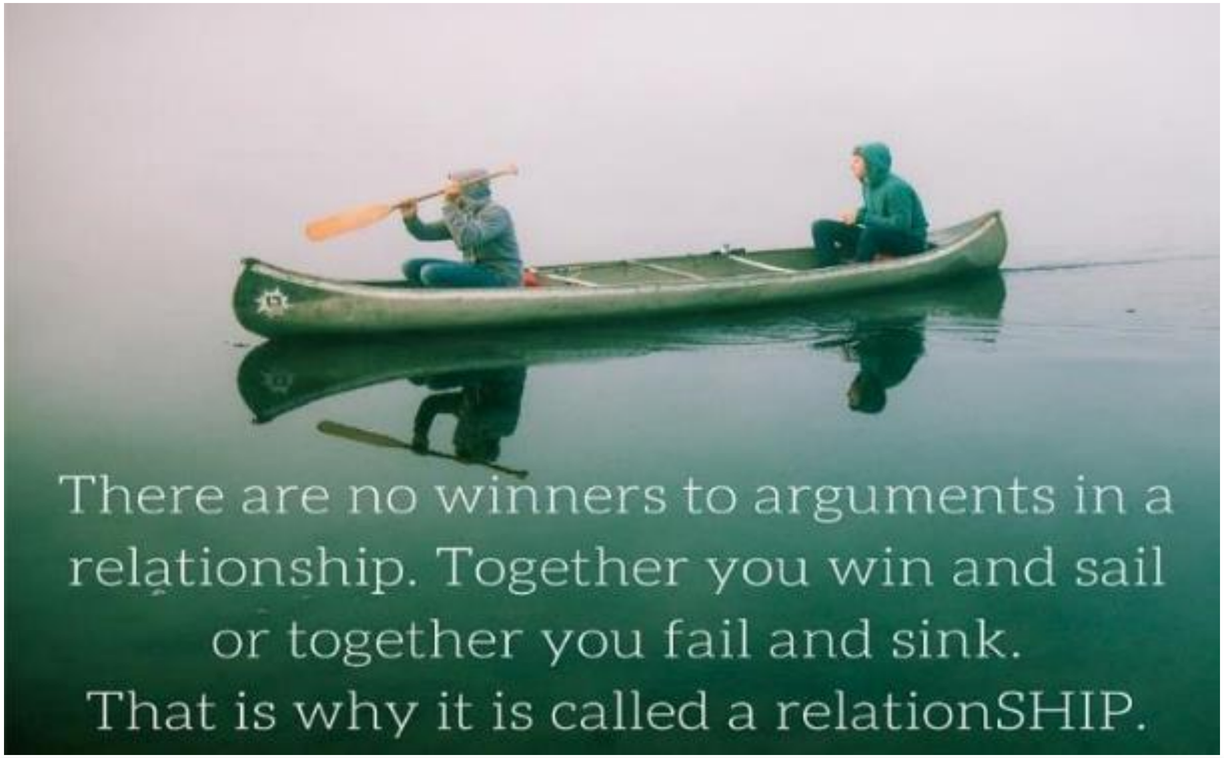
- Why Agile
- Agile Methodology
- Agile Manifesto- 4 values & 12 Principles
- Agile Frameworks
- Agile Vs Waterfall
  - What is SCRUM?
  - Scrum Roles
    - PO –SM- Dev Team
  - Scrum Meetings
    - Release Planning- Sprint Planning- Daily Stand-Ups, Review , Retro
  - Scrum Artifacts
    - PB- SB- Increment
  - Scrum Estimation [optional]
- SCRUM & A road Ahead

ACTIVITY

https://forms.office.com/Pages/ResponsePage.aspx?id=-5xwN7LzSEauoZyCiohhR-AmNT9NboxBkJzF-FBsVMlUNEpXWUNPWVFNVThGMlRYTVBHRjA0MExSRi4u

Being AGILE... Doing Agile: ☺

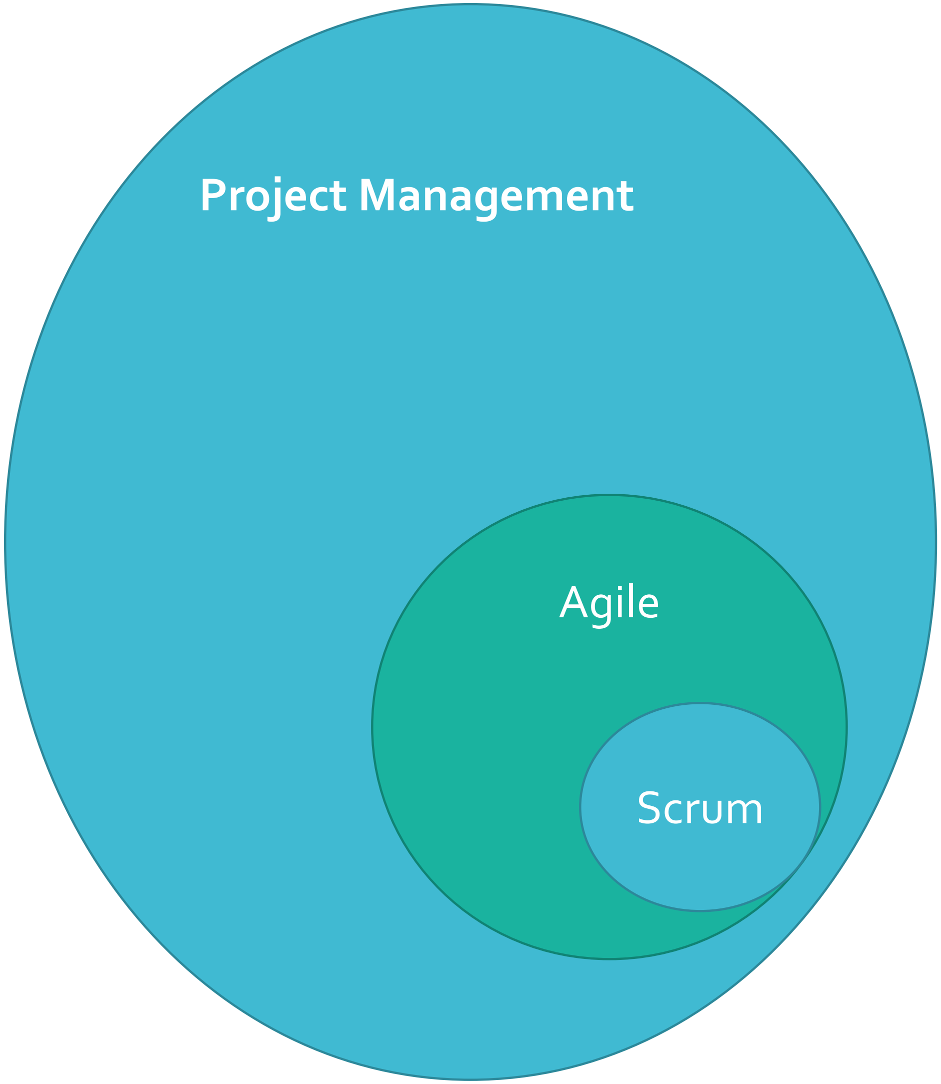
Sail Together or Sink Together



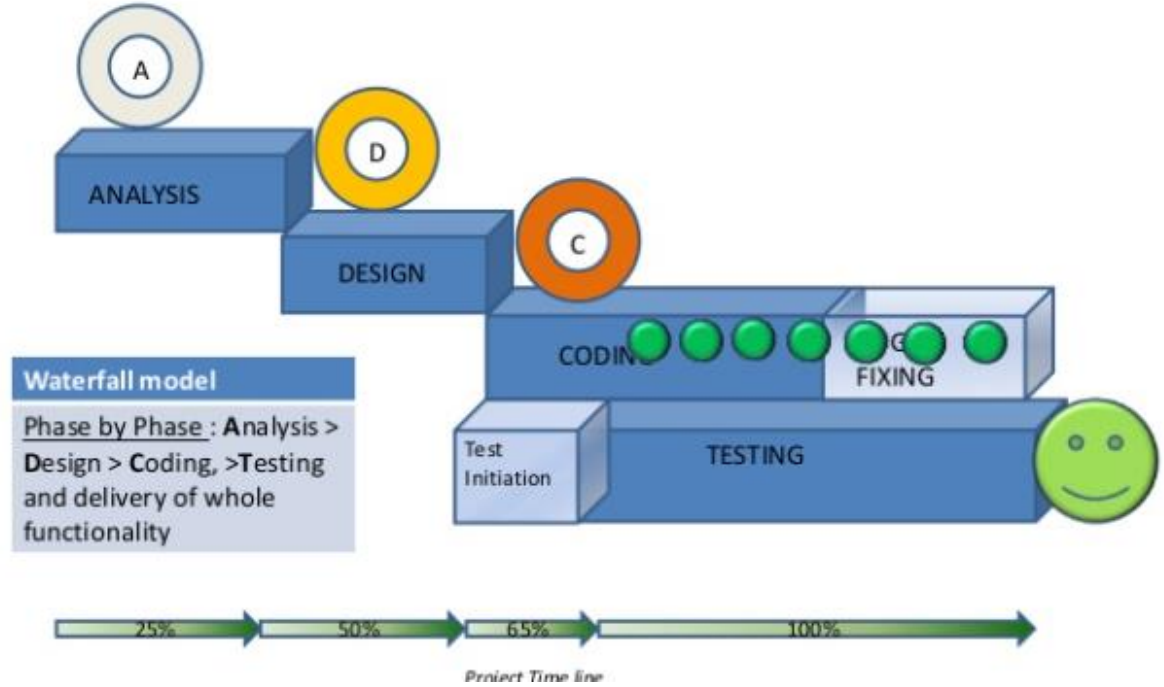
There are no winners to arguments in a relationship. Together you win and sail or together you fail and sink.
That is why it is called a relationSHIP.

A joint accountability to make it SUCCESS

Project Management Sea



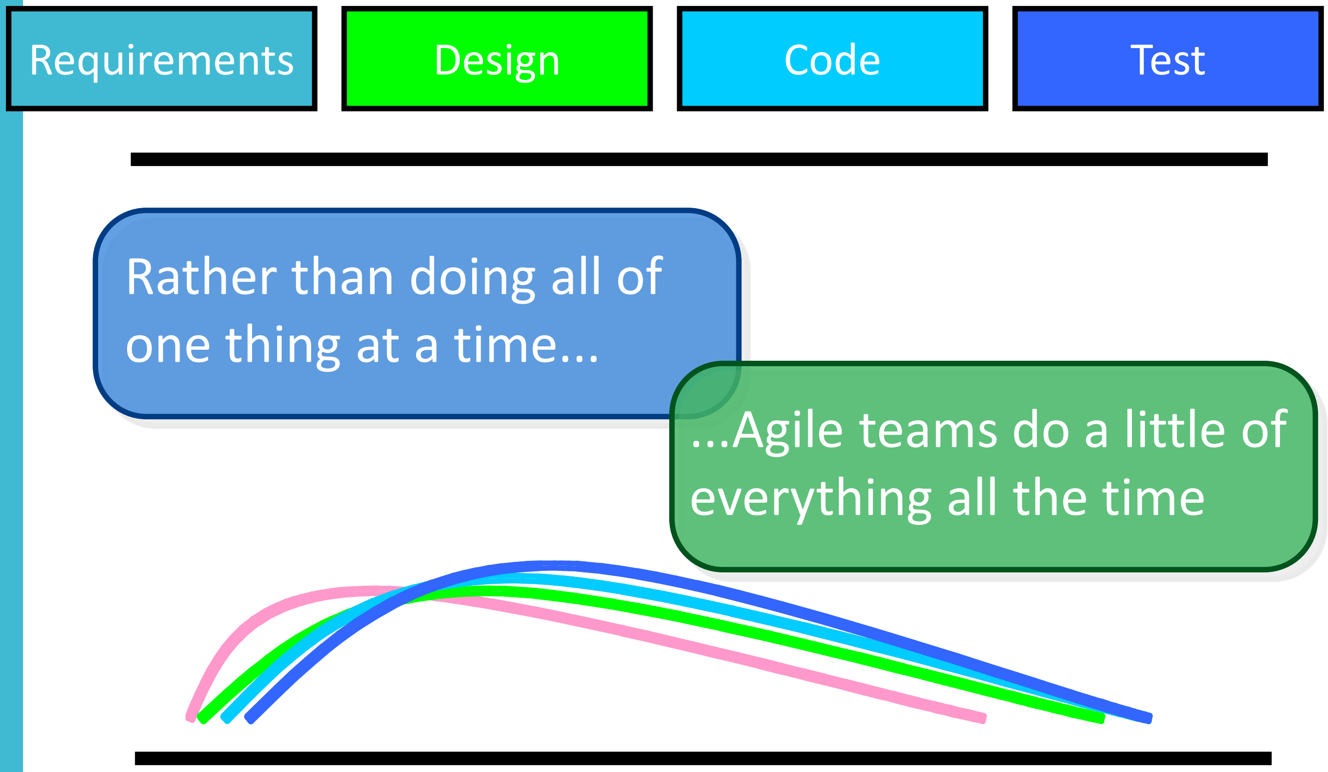
WATERFALL Model



Waterfall model

Phase by Phase : **A**nalysis > **D**esign > **C**oding, >**T**esting and delivery of whole functionality

Sequential vs. Overlapping development



Requirements

Design

Code

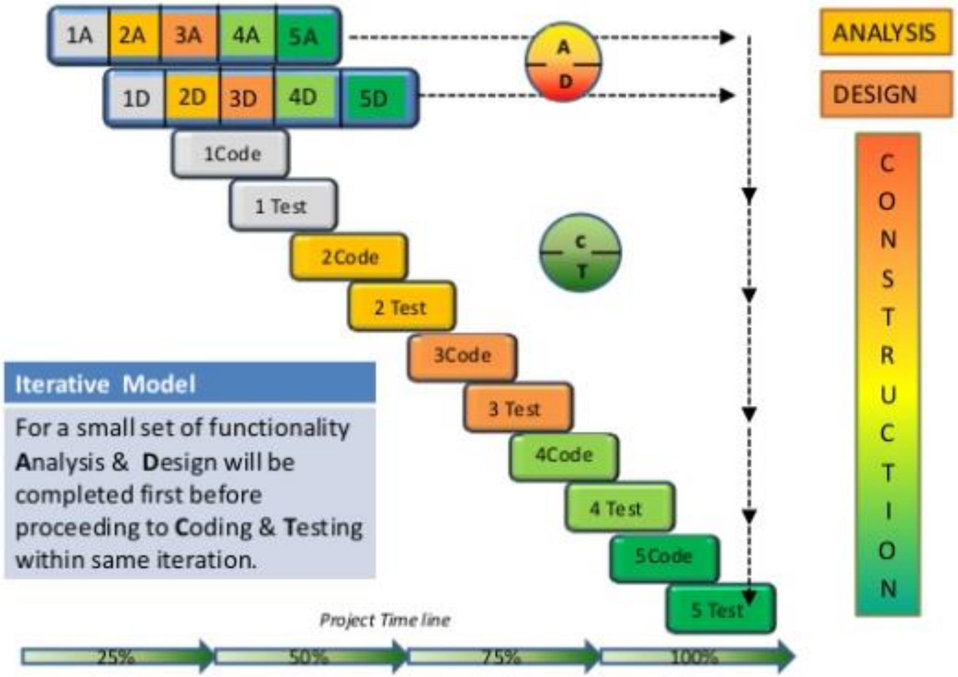
Test

Rather than doing all of one thing at a time...

...Agile teams do a little of everything all the time

Source Courtesy: HBR

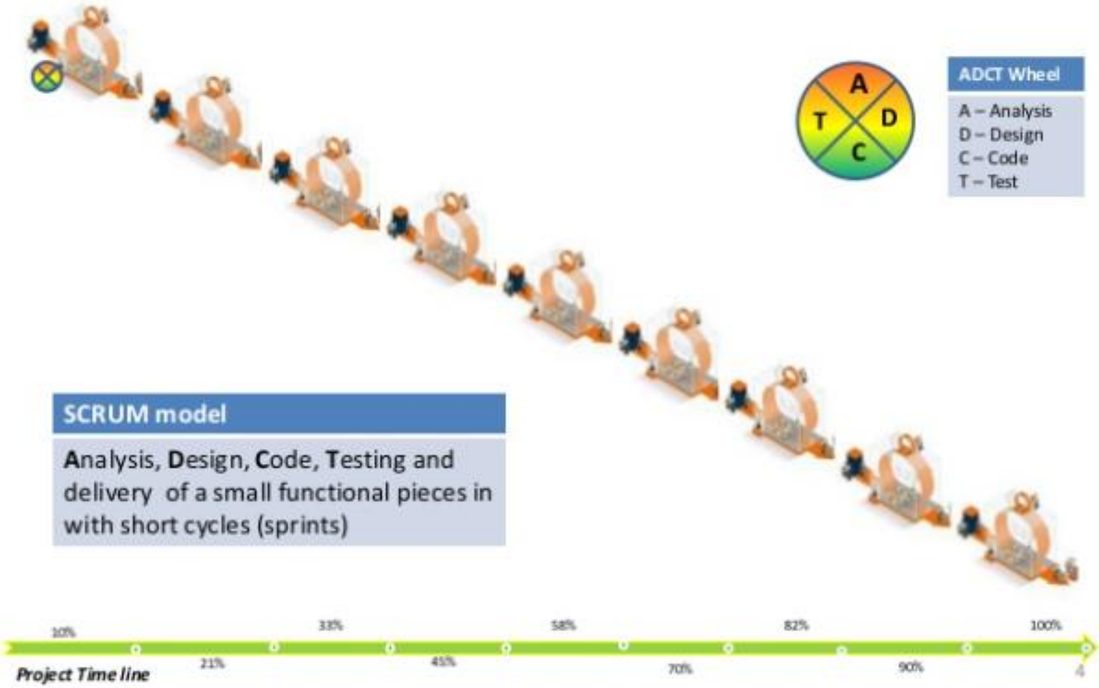
ITERATIVE Model



Iterative Model

For a small set of functionality **A**nalysis & **D**esign will be completed first before proceeding to **C**oding & **T**esting within same iteration.

Agile- SCRUM Model

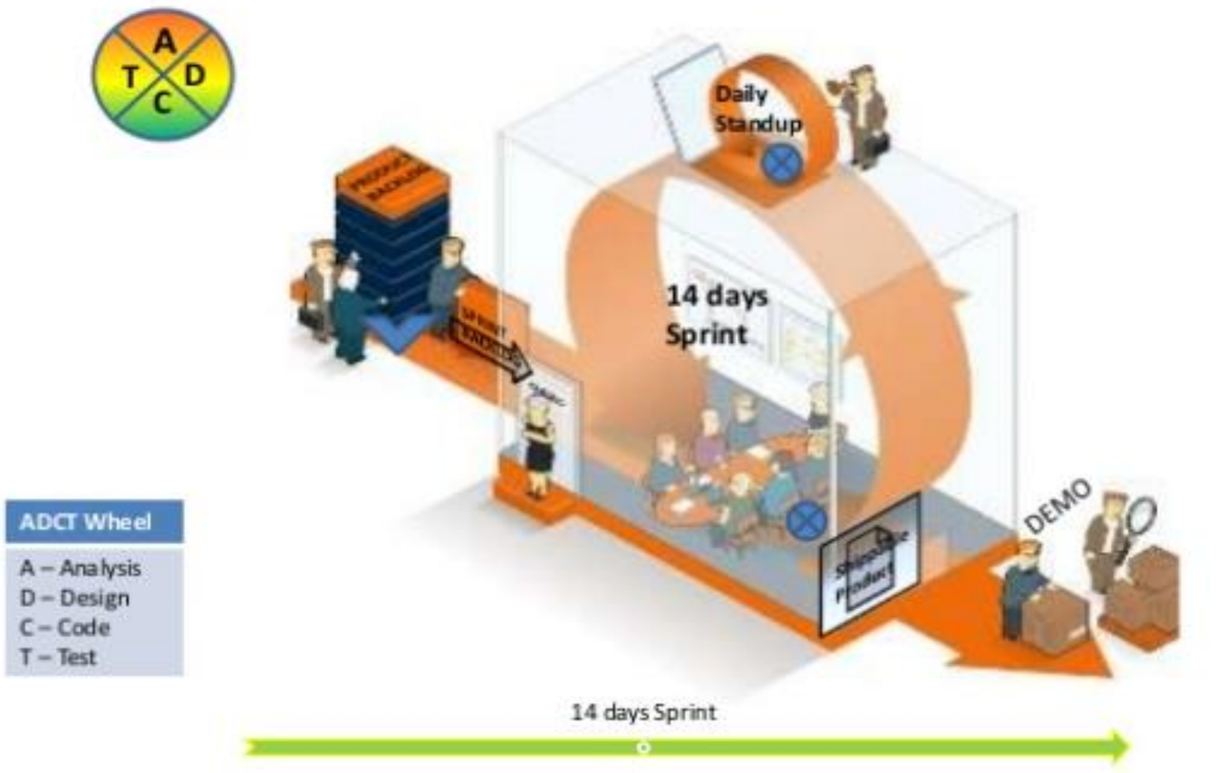


ADCT Wheel

A – Analysis
D – Design
C – Code
T – Test

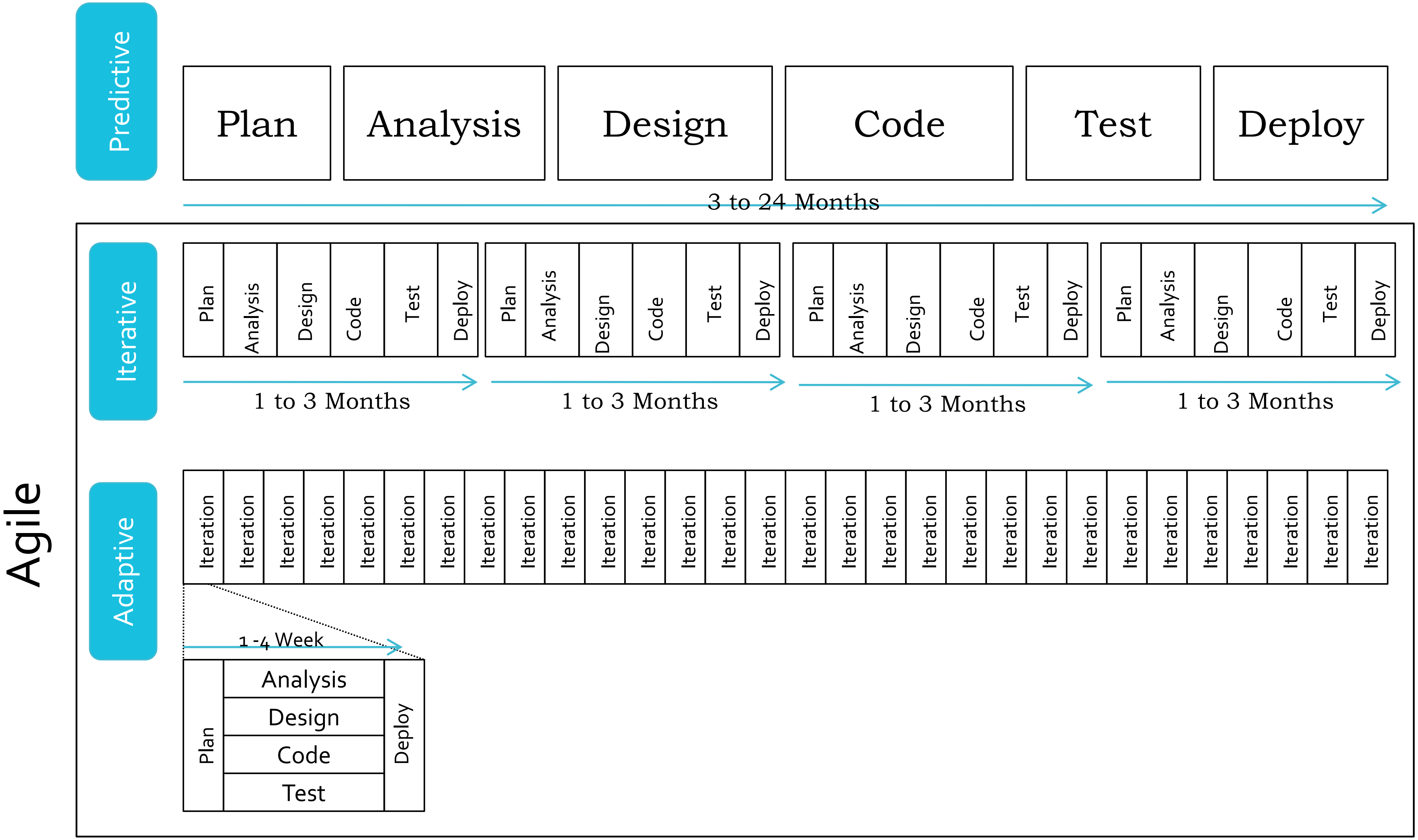
SCRUM model

Analysis, **D**esign, **C**ode, **T**esting and delivery of a small functional pieces in with short cycles (sprints)



SCRUM- SPRINT

Software Development Life Cycle



Predictive

Plan | Analysis | Design | Code | Test | Deploy

3 to 24 Months

Agile

Iterative

Plan | Analysis | Design | Code | Test | Deploy | Plan | Analysis | Design | Code | Test | Deploy | Plan | Analysis | Design | Code | Test | Deploy | Plan | Analysis | Design | Code | Test | Deploy

1 to 3 Months

1 to 3 Months

1 to 3 Months

1 to 3 Months

Adaptive

Iteration | Iteration | Iteration | Iteration | Iteration | Iteration | Iteration | Iteration | Iteration | Iteration | Iteration | Iteration | Iteration | Iteration | Iteration | Iteration | Iteration | Iteration | Iteration | Iteration | Iteration | Iteration | Iteration | Iteration | Iteration | Iteration | Iteration | Iteration | Iteration | Iteration

1 -4 Week

Plan

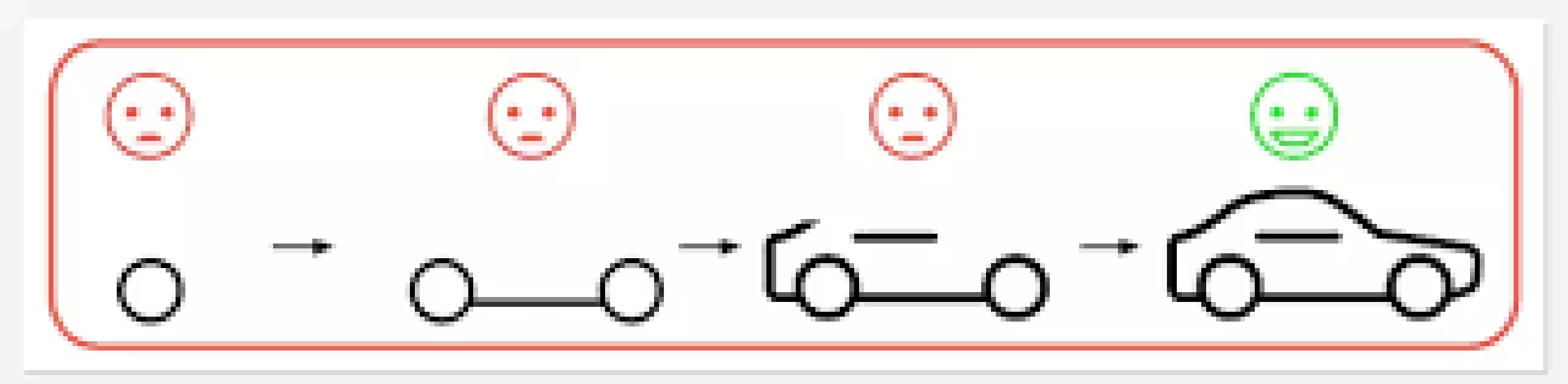
Analysis

Design

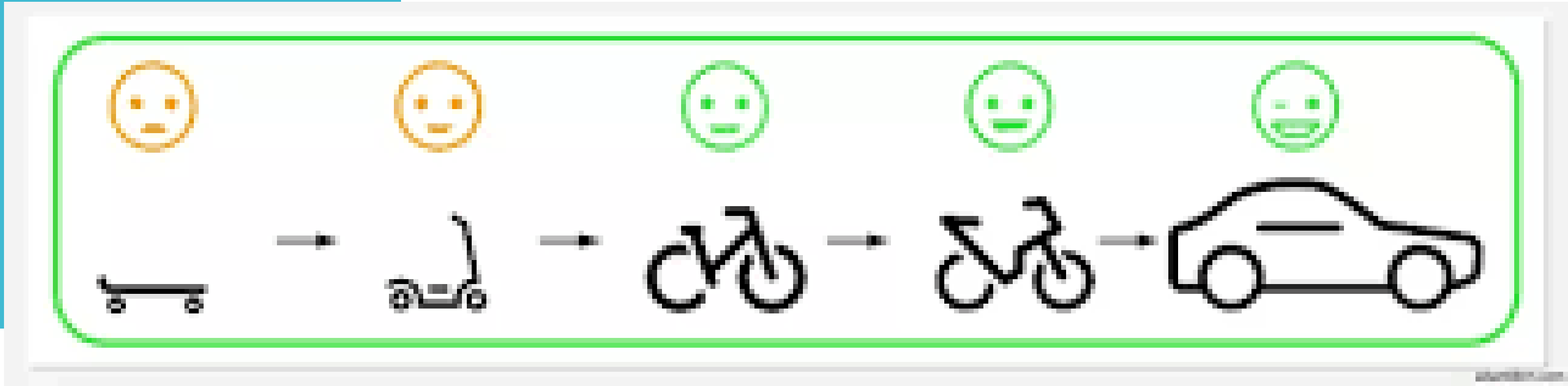
Code

Test

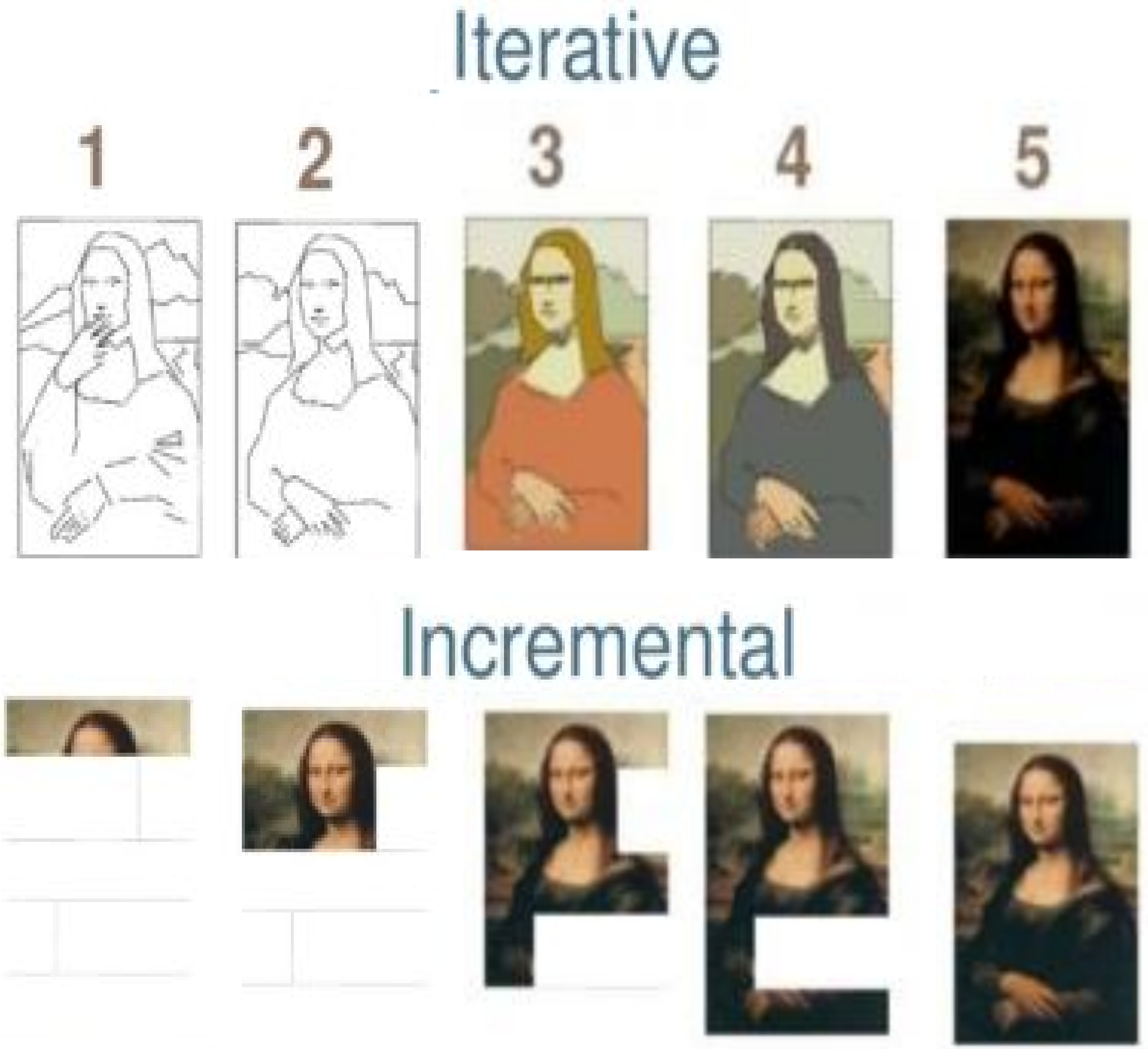
Deploy

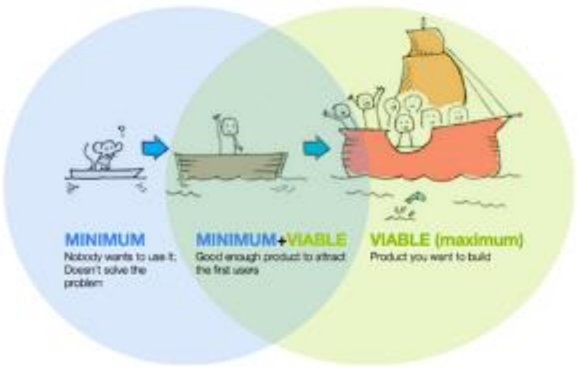


Adaptive Devp. Vs. Incremental Devp.

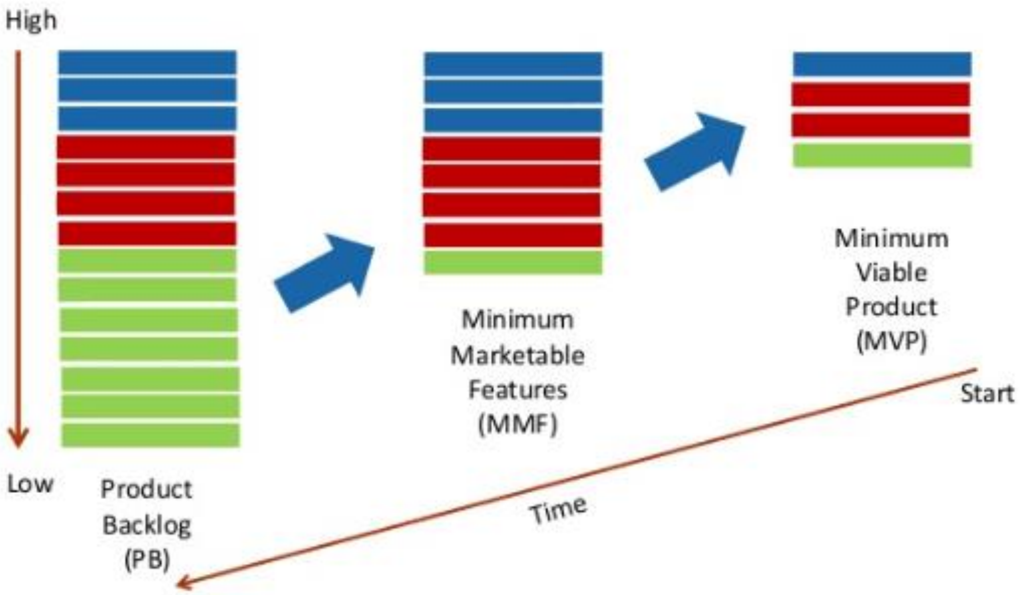


Iterative Vs. Incremental Development

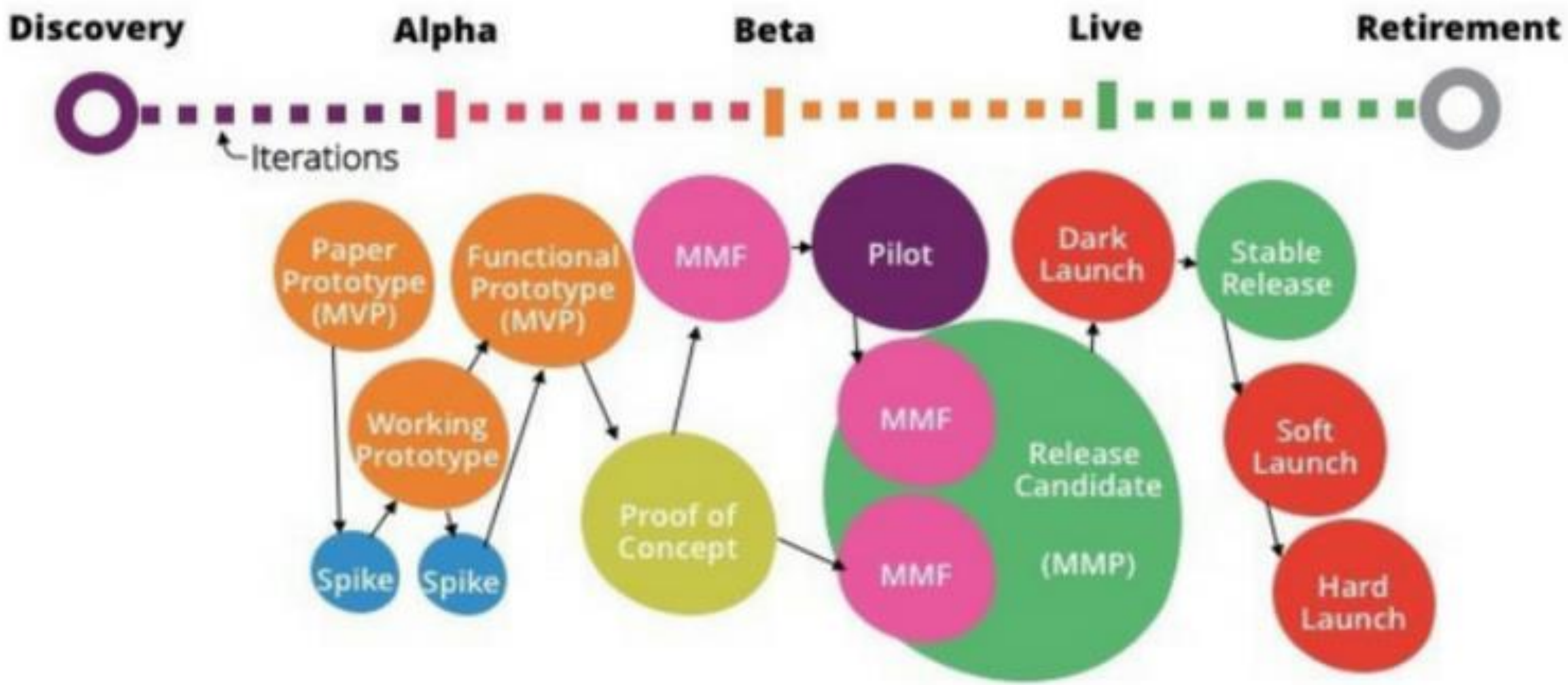


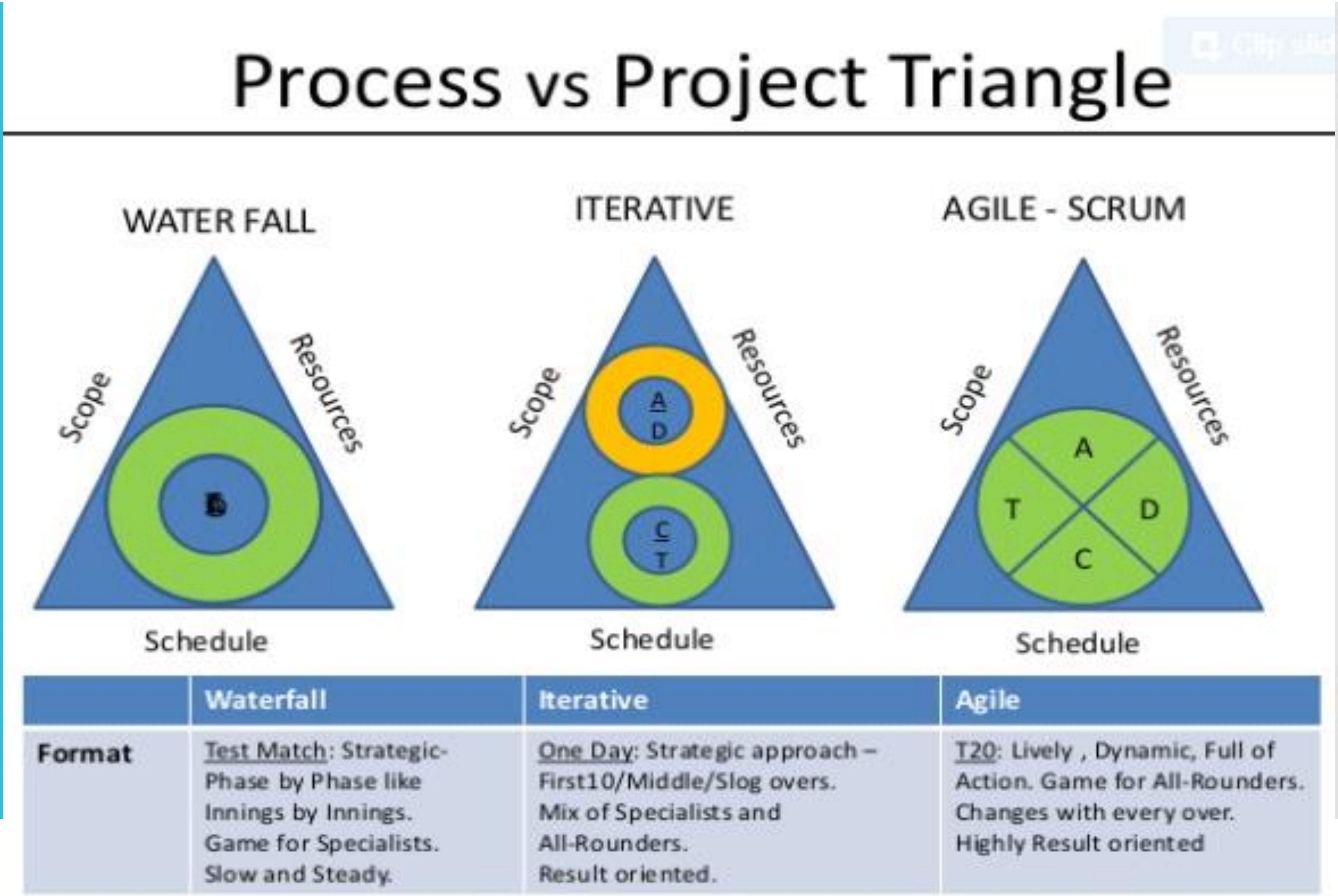

MINIMUM
Nobody wants to use it; Doesn't solve the problem
MINIMUM+VIABLE
Good enough product to attract the first users
VIABLE (maximum)
Product you want to build

MVP or MMF


High
Low
Product Backlog (PB)
Minimum Marketable Features (MMF)
Minimum Viable Product (MVP)
Start
Time

Product Lifecycle: Discovery to Retirement

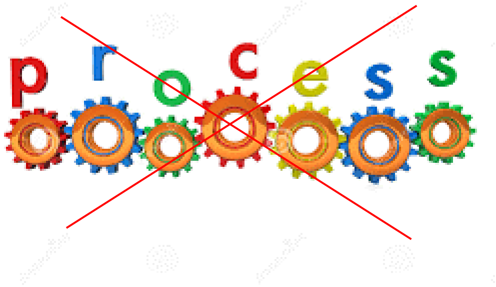




Process vs Project Triangle

	Waterfall	Iterative	Agile
Format	Test Match: Strategic- Phase by Phase like Innings by Innings. Game for Specialists. Slow and Steady.	One Day: Strategic approach – First10/Middle/Slog overs. Mix of Specialists and All-Rounders. Result oriented.	T20: Lively , Dynamic, Full of Action. Game for All-Rounders. Changes with every over. Highly Result oriented

The problem



Scrum and Agile are based on the hypothesis that there is no meta-solution for software development. Just a framework within which we will be empirical – Inspect and Adapt

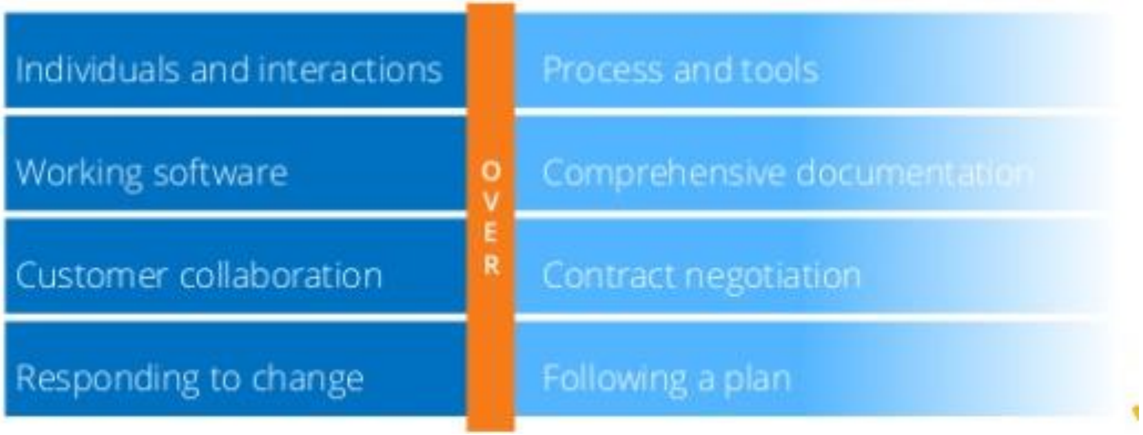
Empirical Model

- Ken Schwaber

Agile Manifesto Origin

- On February 11-13, 2001, at The Lodge at Snowbird ski resort in the Wasatch mountains of Utah, **17 people** met to **talk, ski, relax, and try to find common ground** and of course, to eat. What emerged was the Agile Software Development Manifesto.
- Representatives from Extreme Programming, SCRUM, DSDM, Adaptive Software Development, Crystal, Feature-Driven Development, Pragmatic Programming, and others sympathetic to the need for an alternative to documentation driven, heavyweight software development processes convened.
- Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockbur,n Ward Cunningham, Martin Fowler,, James Grenning Jim Highsmith, Andrew Hun,t Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwabe,r Jeff Sutherland, Dave Thomas

Reference: http://agilemanifesto.org/



Individuals and interactions	O V E R	Process and tools
Working software		Comprehensive documentation
Customer collaboration		Contract negotiation
Responding to change		Following a plan

Manifesto for Agile

We are uncovering better ways of developing software by doing it and helping others do it.

Through this work we have come to value:

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

Reference: http://agilemanifesto.org/

Principles behind the Agile Manifesto

We follow these 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 to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around self-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.
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.

Reference: http://agilemanifesto.org/

Agile Mind set

What is a differentiator between

‘**Doing**’ Agile

&

‘**Being**’ Agile

Agile mindset, Intent-based Management

200%+ benefit

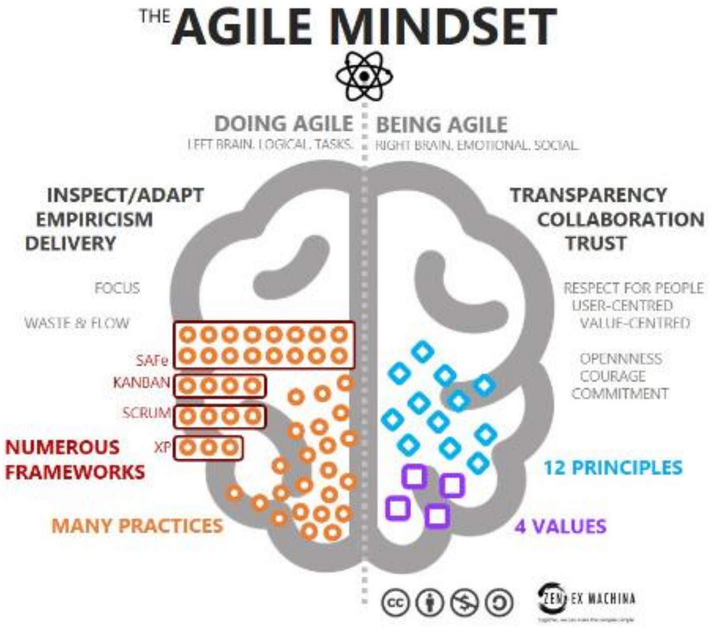
- Employee engagement
- Leadership at all levels
- Customer delight
- Continuous learning

Doing Agile ≠ Being Agile

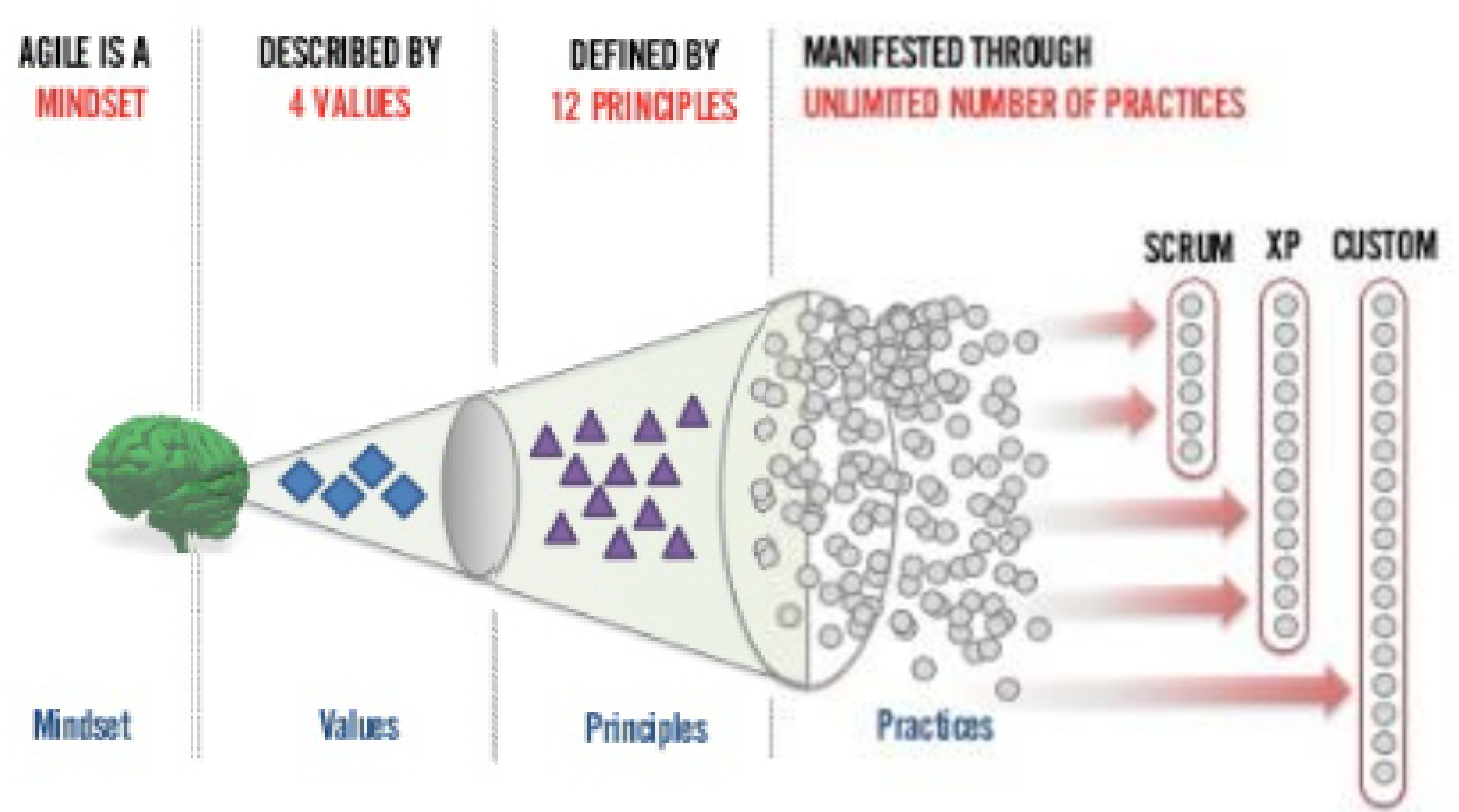
Kanban boards, daily scrums, Sprints

20% or so benefit

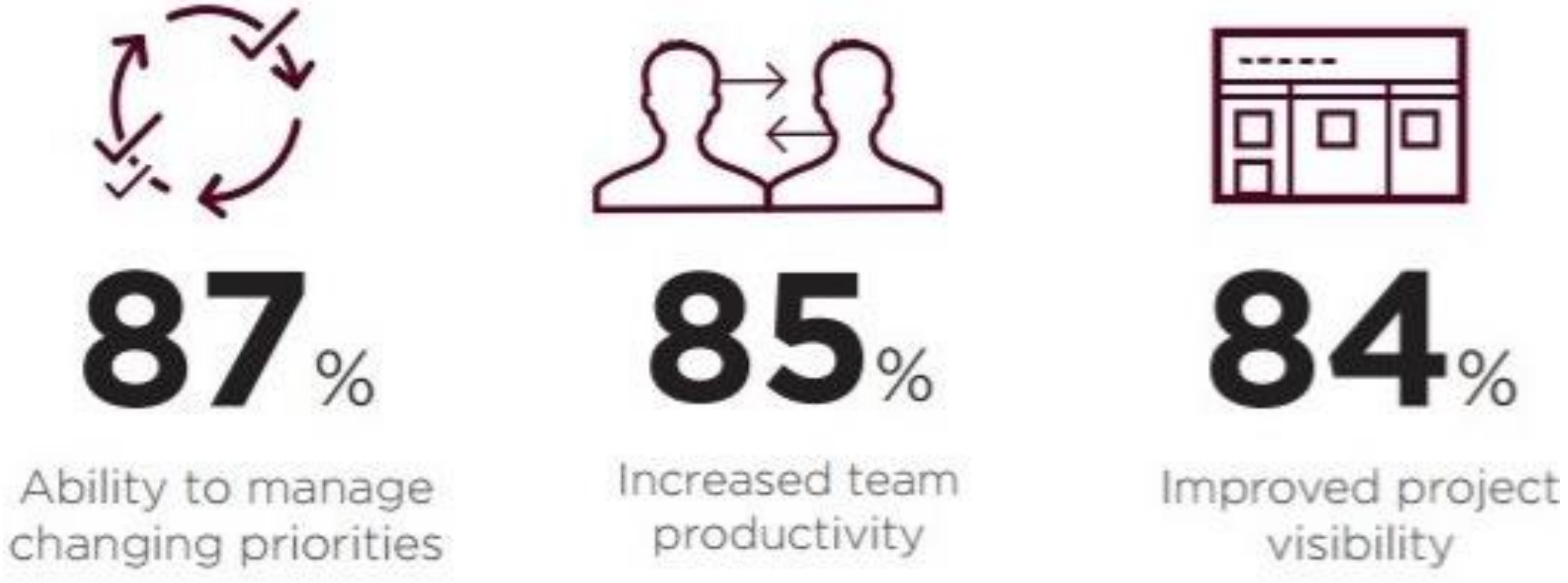
- Improved visibility, communication
- Increased productivity
- Some ability to adapt to changing priorities



The Agile Mindset. Zen Ex Machina (2017)



Agile Mindset –Ahmed Sidky



87%

Ability to manage changing priorities

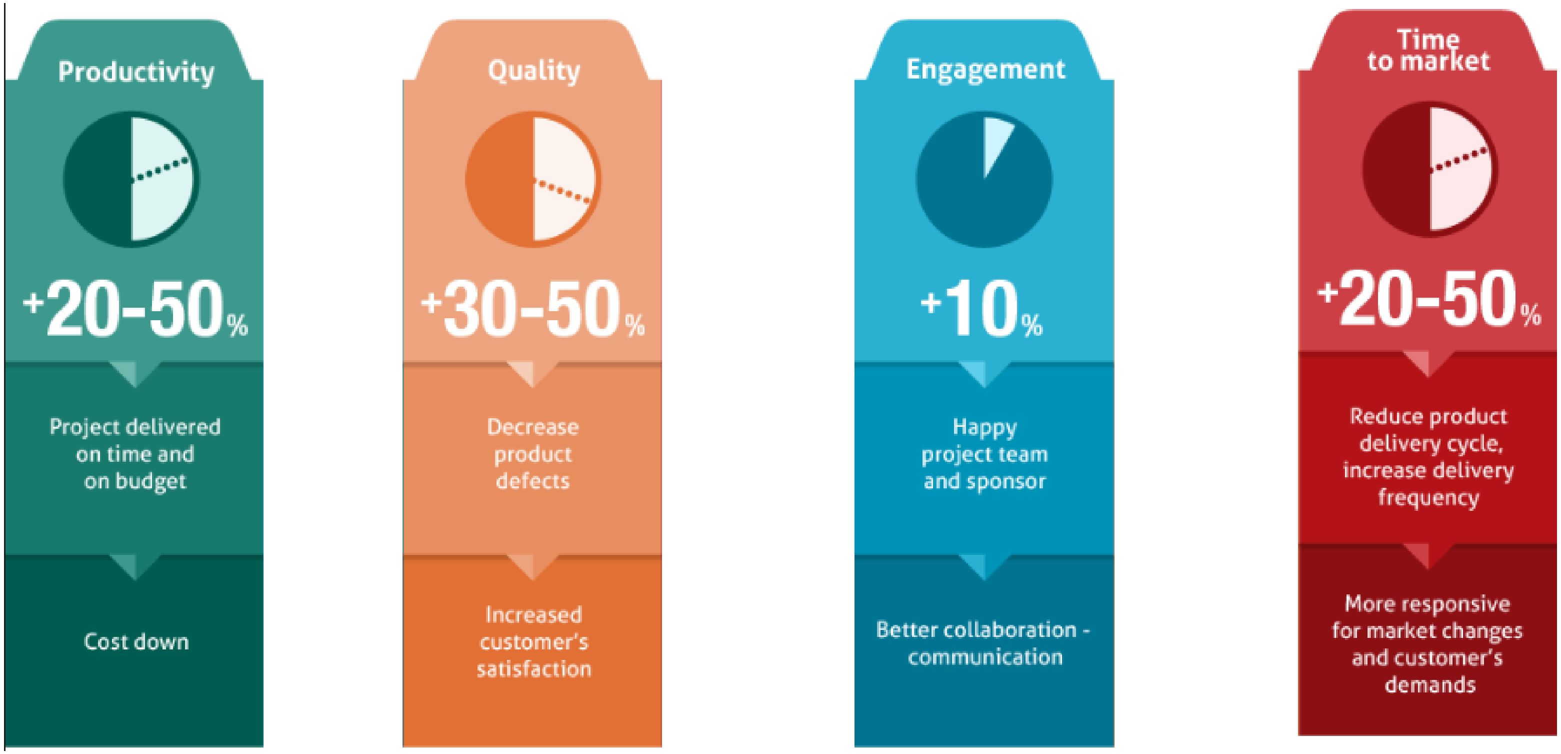
85%

Increased team productivity

84%

Improved project visibility

Agile Benefits @Team level



Productivity

+20-50%

Project delivered on time and on budget

Cost down

Quality

+30-50%

Decrease product defects

Increased customer's satisfaction

Engagement

+10%

Happy project team and sponsor

Better collaboration - communication

Time to market

+20-50%

Reduce product delivery cycle, increase delivery frequency

More responsive for market changes and customer's demands

Agile Benefits @Enterprise Level

Summary

We have discussed the following

- Project Management
- Traditional Vs Agile
- Why Agile
- Agile Manifesto & Origin
- Benefits of agile @Team & Enterprise

1. Which of the following is NOT a principle from the Agile Manifesto?

A. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software

B. Business people and developers must work together daily throughout the project.

C. Project manager should control the project execution

D. Working software is the primary measure of progress.

Quiz

2. Which of the role is not found in agile projects?

A. Project Director
B. Product Owner
C. Scrum Master
D. Team

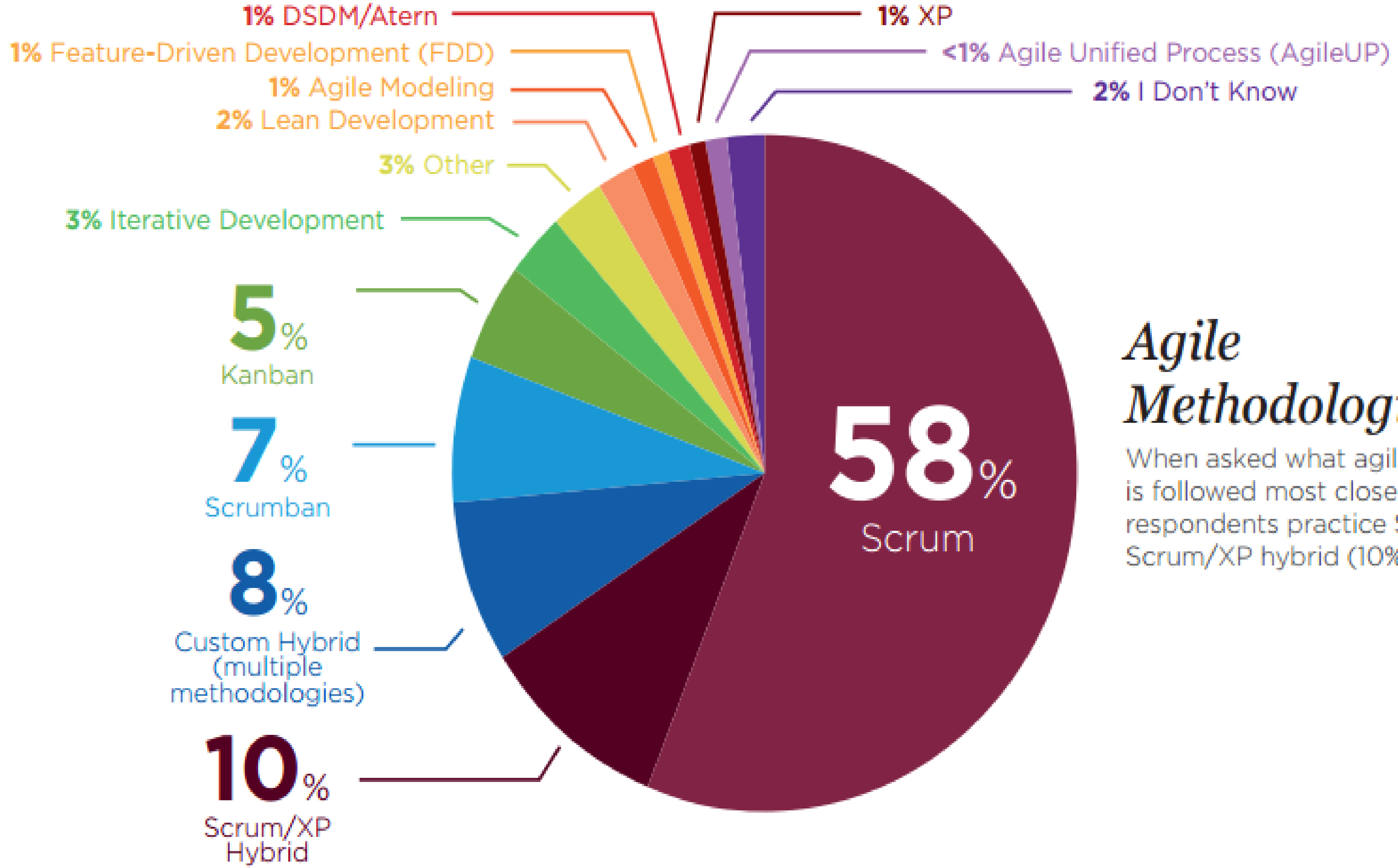
Quiz

Agile Methodologies

SCRUM
Project Management

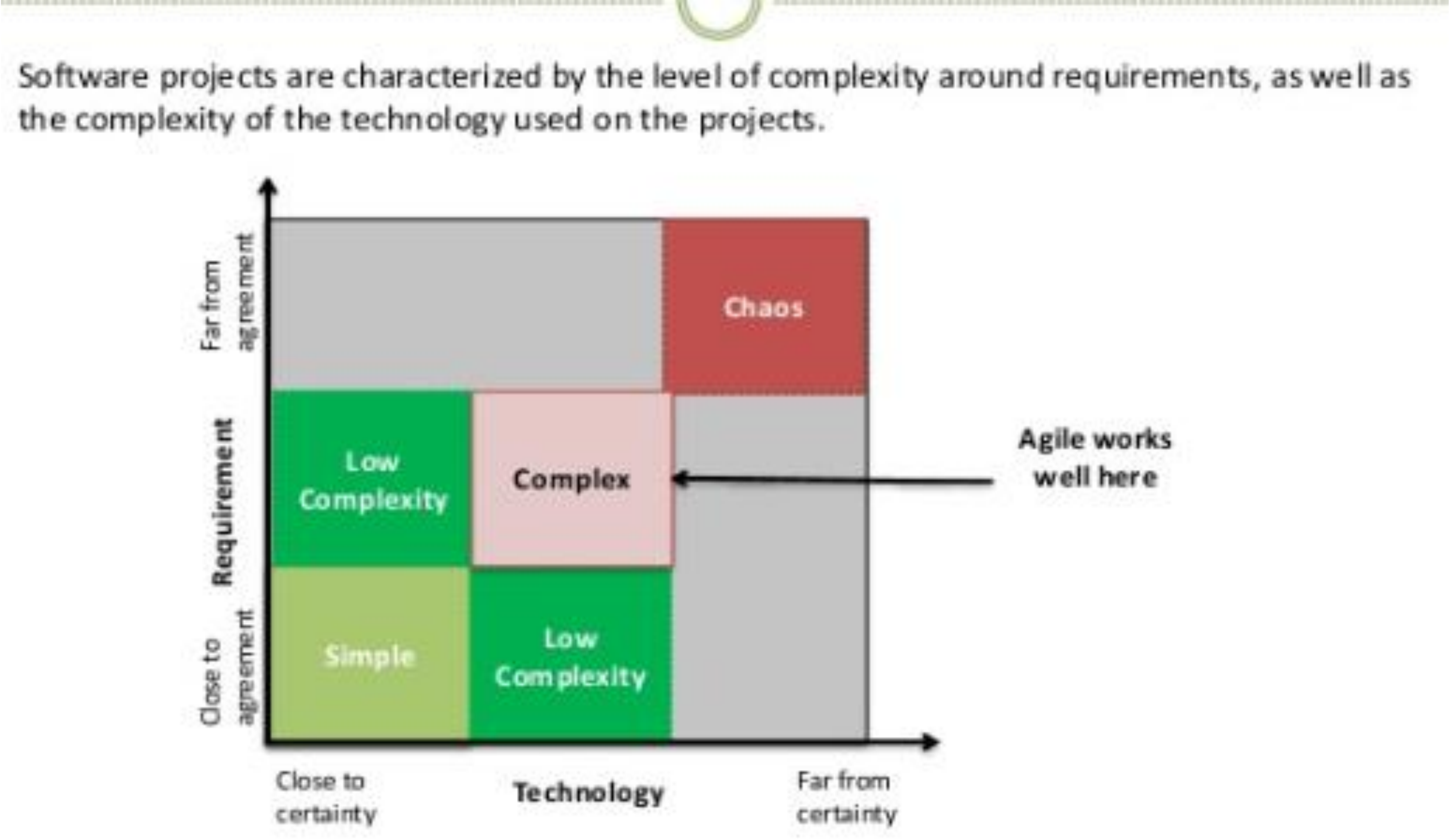
MODULE - 2

AGILE METHODS AND PRACTICES



Agile Methodologies Used

When asked what agile methodology is followed most closely, nearly 70% of respondents practice Scrum (58%) or Scrum/XP hybrid (10%).

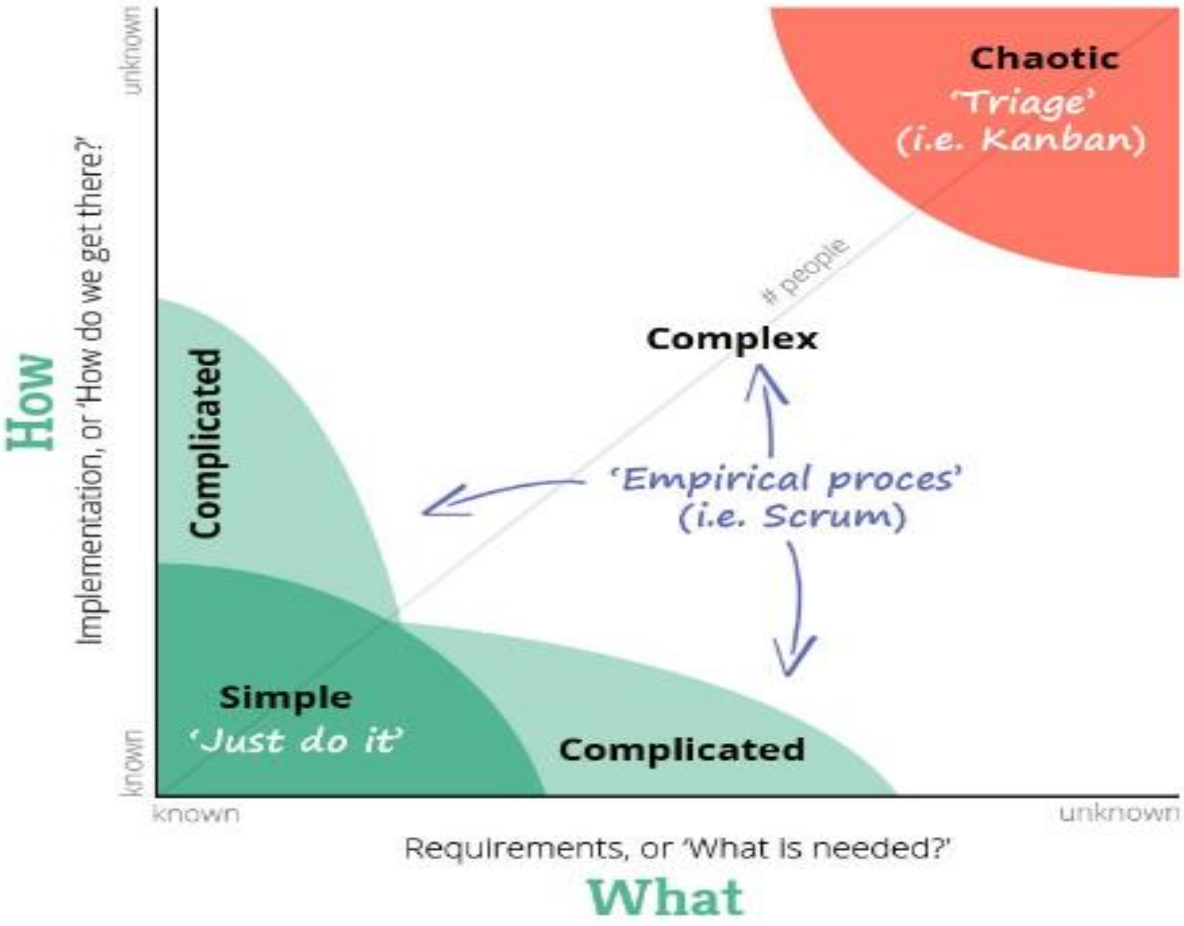


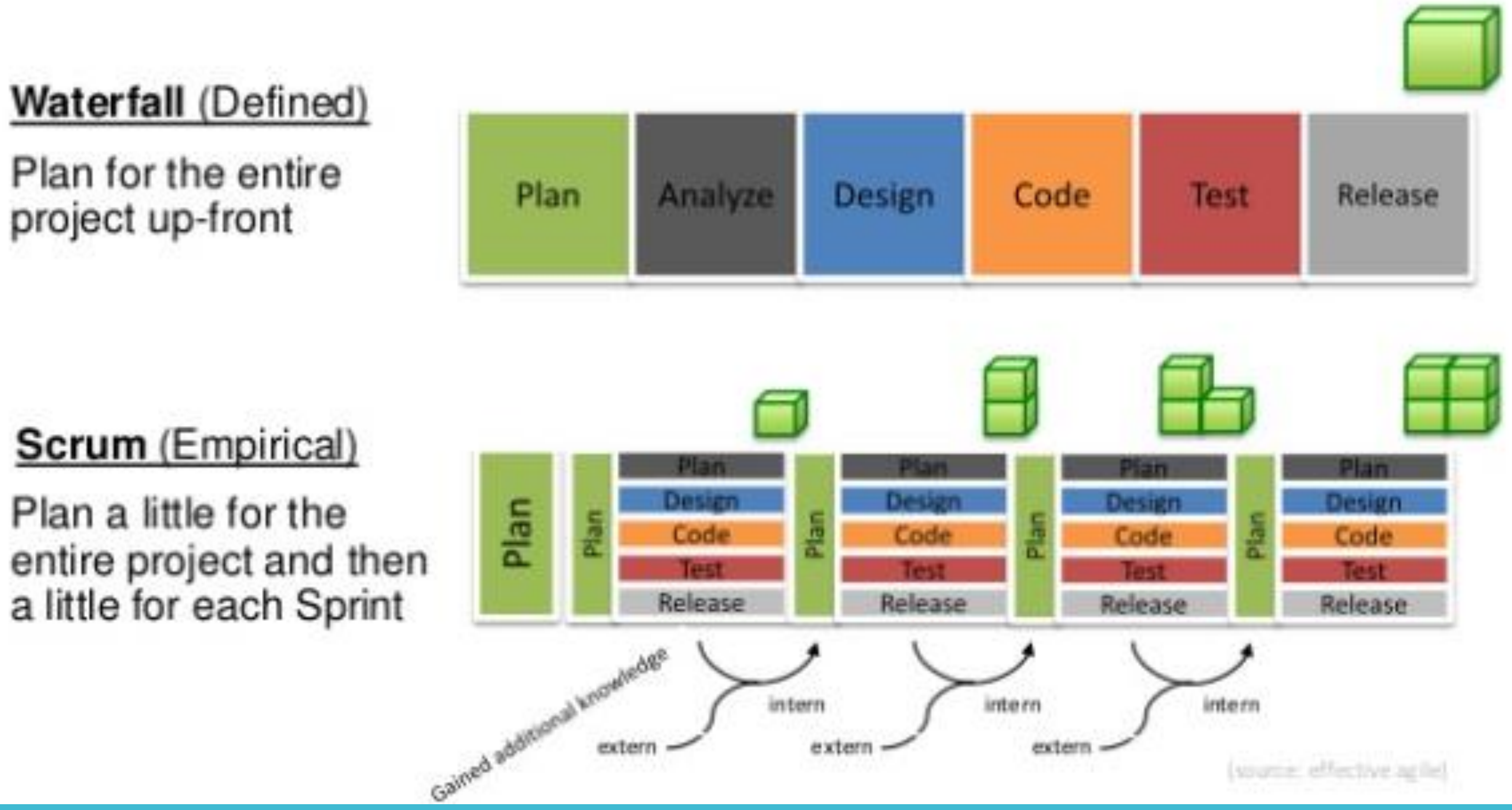
Software projects are characterized by the level of complexity around requirements, as well as the complexity of the technology used on the projects.

How do you decide…?!!!

Stacey Matrix

adapted for software development





Waterfall (Defined)

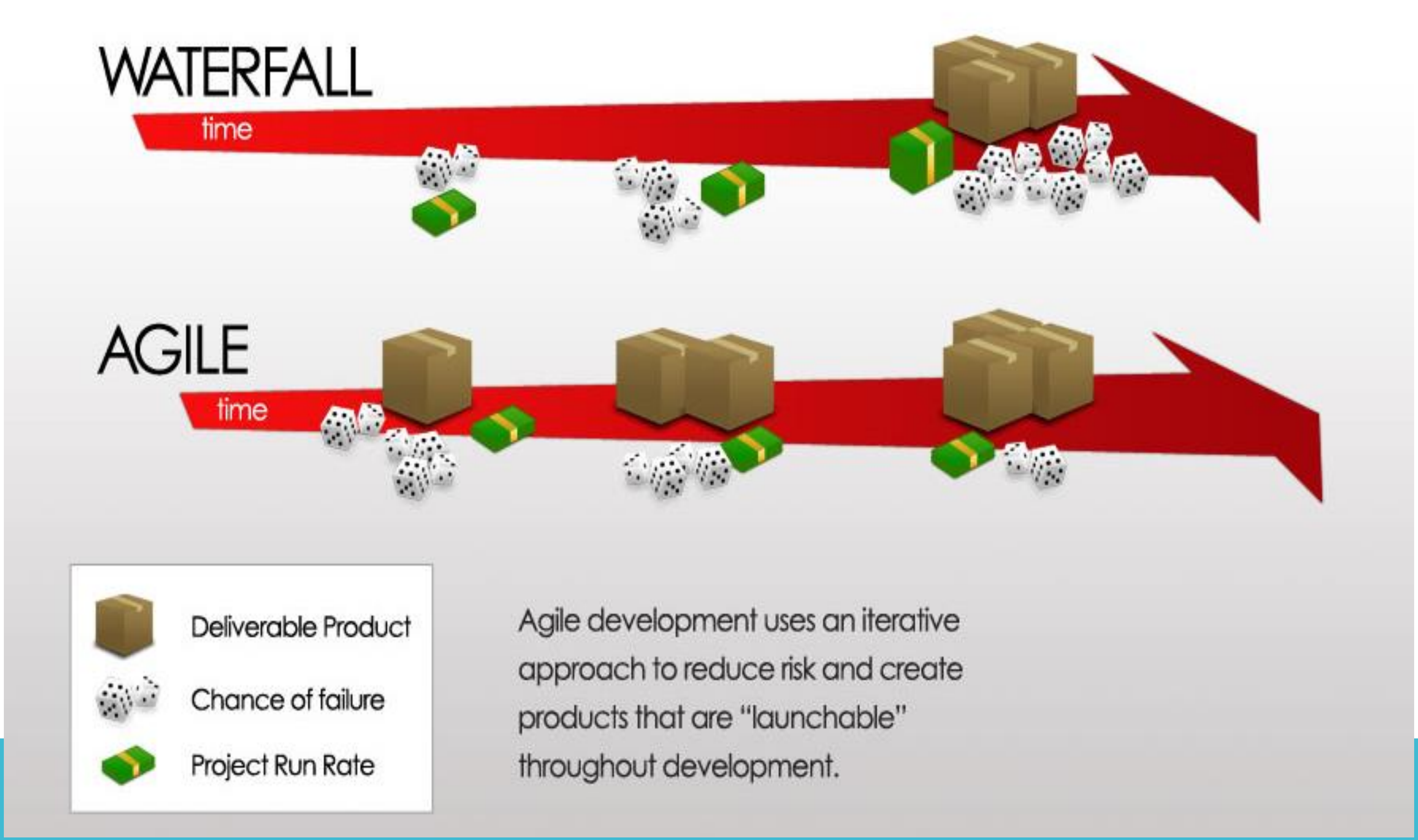
Plan for the entire project up-front

Scrum (Empirical)

Plan a little for the entire project and then a little for each Sprint

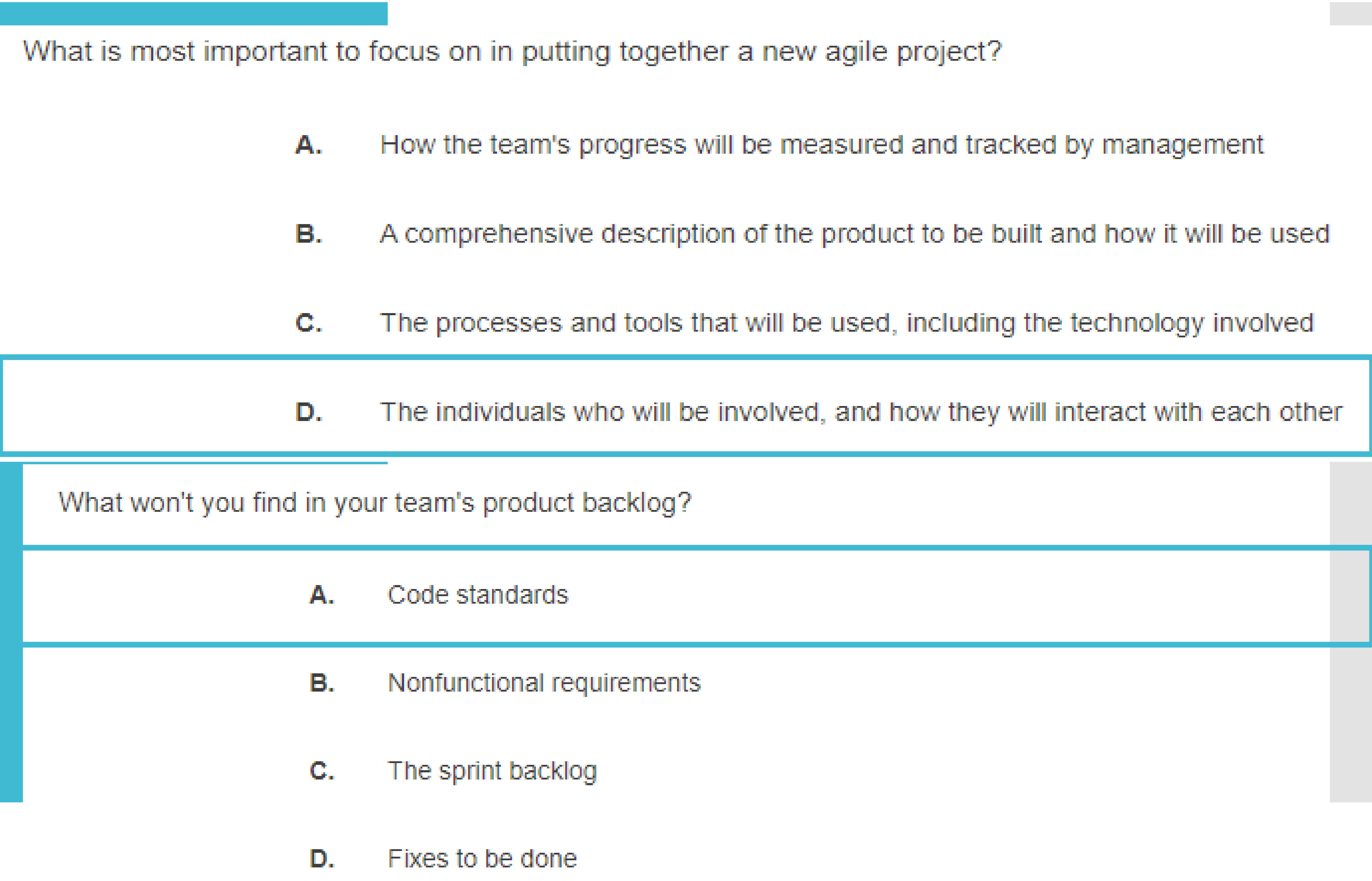
(source: effective agile)

Defined Vs. Empirical



Agile development uses an iterative approach to reduce risk and create products that are "launchable" throughout development.

Risk & Delivery Rate: Waterfall Vs. Agile



What is most important to focus on in putting together a new agile project?

A. How the team's progress will be measured and tracked by management

B. A comprehensive description of the product to be built and how it will be used

C. The processes and tools that will be used, including the technology involved

[✓ D. The individuals who will be involved, and how they will interact with each other]

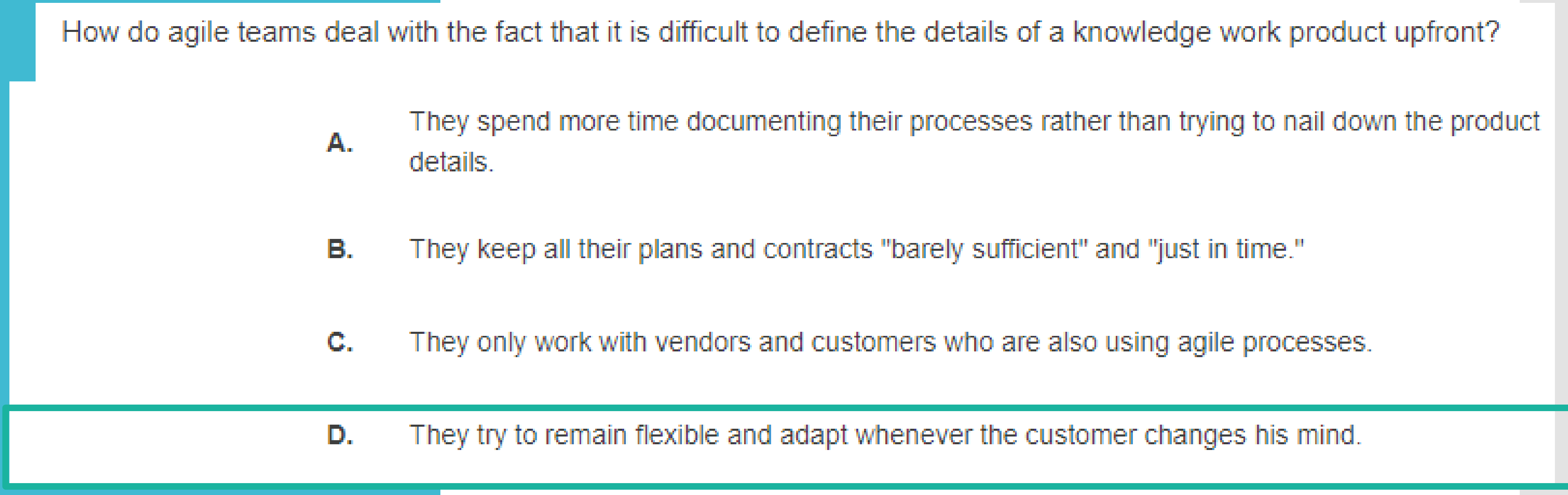
What won't you find in your team's product backlog?

[✓ A. Code standards]

B. Nonfunctional requirements

C. The sprint backlog

D. Fixes to be done



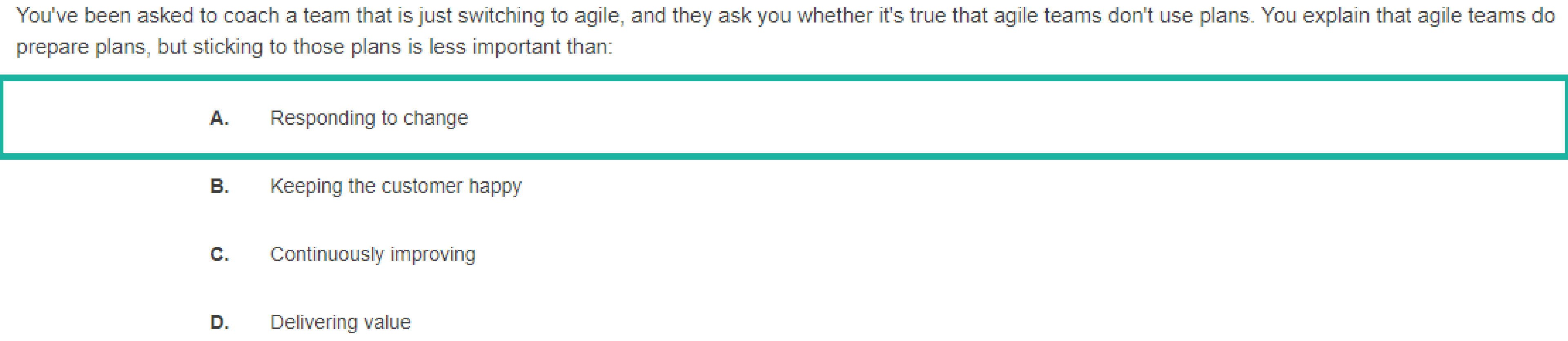
How do agile teams deal with the fact that it is difficult to define the details of a knowledge work product upfront?

A. They spend more time documenting their processes rather than trying to nail down the product details.

B. They keep all their plans and contracts "barely sufficient" and "just in time."

C. They only work with vendors and customers who are also using agile processes.

[✓ D. They try to remain flexible and adapt whenever the customer changes his mind.]



You've been asked to coach a team that is just switching to agile, and they ask you whether it's true that agile teams don't use plans. You explain that agile teams do prepare plans, but sticking to those plans is less important than:

[✓ A. Responding to change]

B. Keeping the customer happy

C. Continuously improving

D. Delivering value

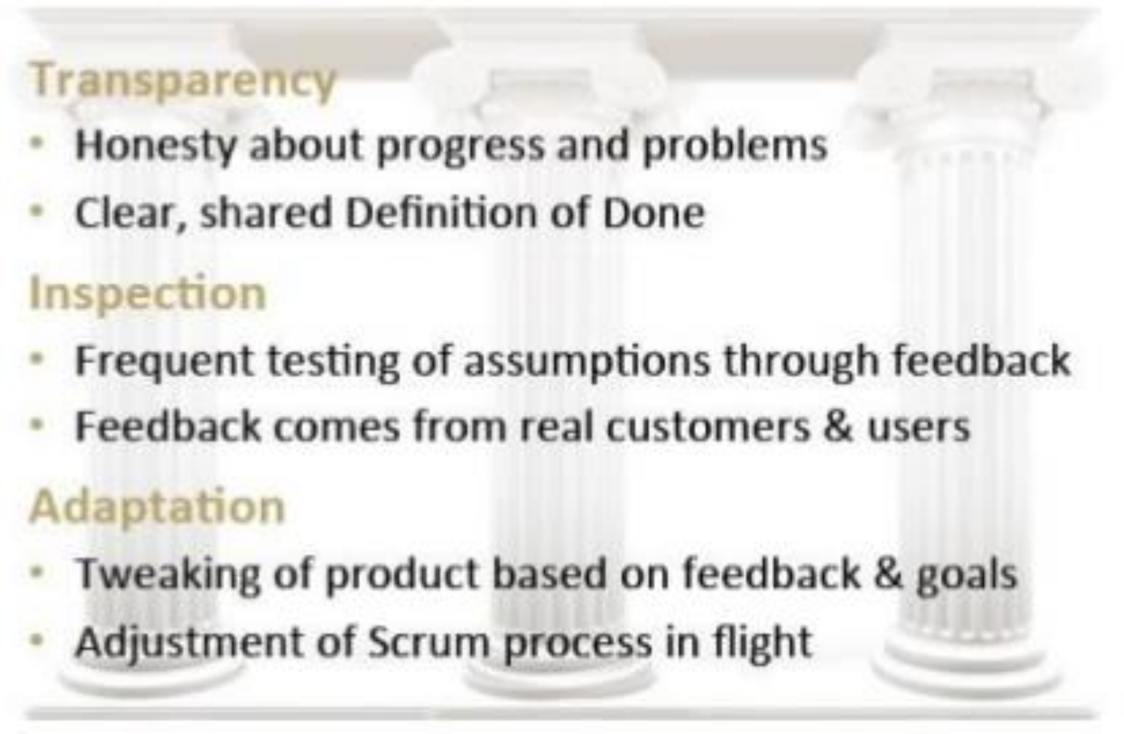
SCRUM

- Empirical Process
- Scrum Roles
  - PO –SM-Team
    - User Stories- Use case- Requirements
    - Velocity
  - Scrum Meetings
    - Daily Stand-up
    - Sprint Planning
    - Sprint Demo
    - Sprint retro
- Scrum Artifacts
  - PB
  - SB
  - Increment
- Sprint reports
  - Burndown Chart

Empirical Process Control

The Scrum Framework
Transparency
Inspection
Adaptation

Three pillars
uphold every implementation of
Empirical process control



Transparency

- Honesty about progress and problems
- Clear, shared Definition of Done

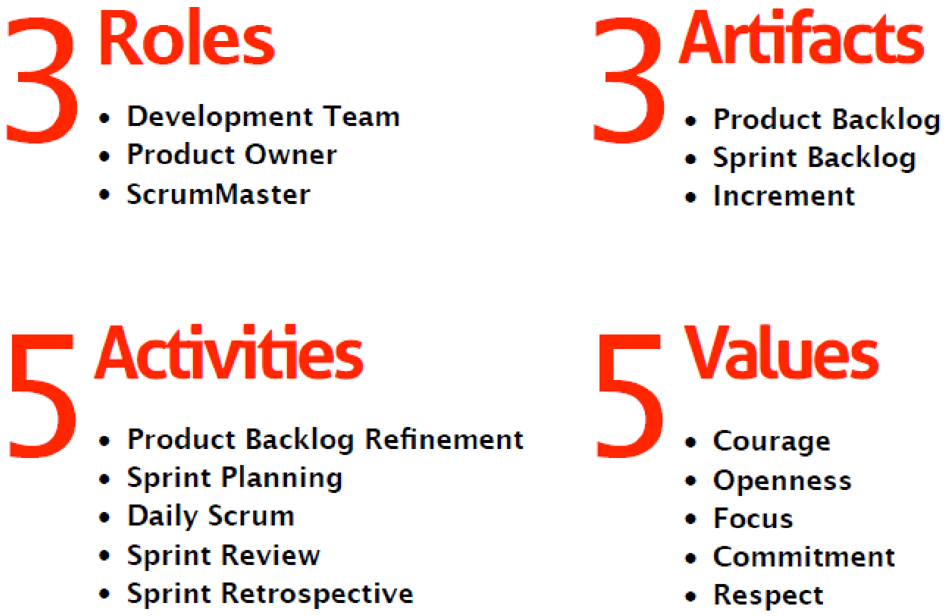
Inspection

- Frequent testing of assumptions through feedback
- Feedback comes from real customers & users

Adaptation

- Tweaking of product based on feedback & goals
- Adjustment of Scrum process in flight

Scrum Framework



3 Roles

- Development Team
- Product Owner
- ScrumMaster

3 Artifacts

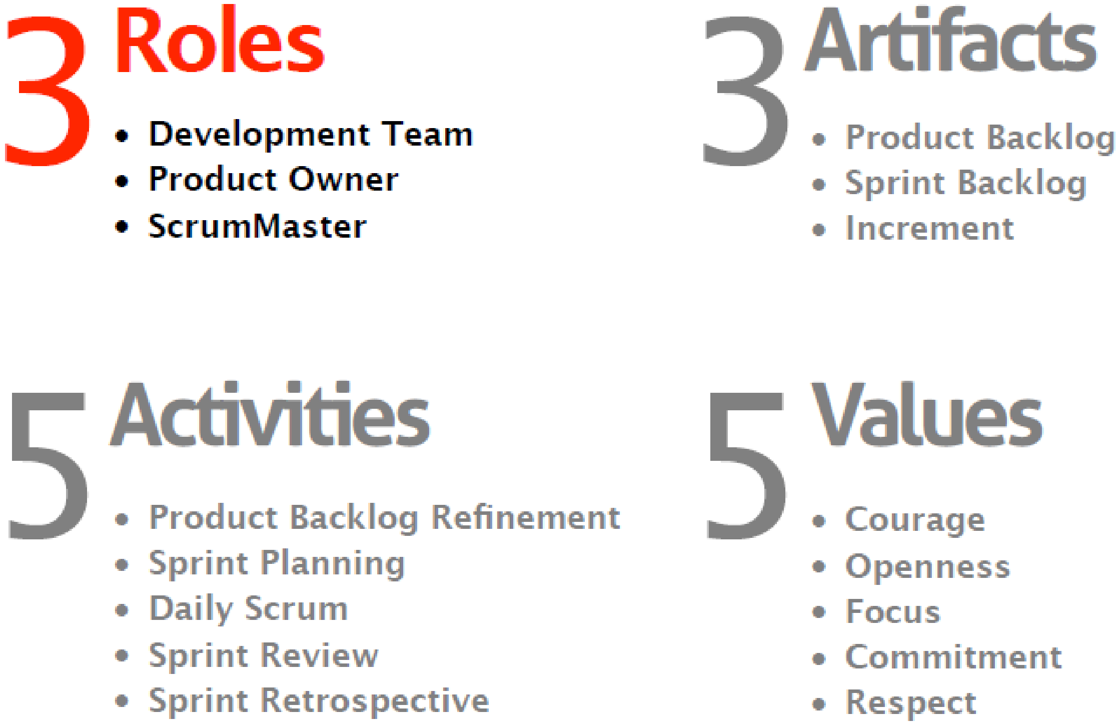
- Product Backlog
- Sprint Backlog
- Increment

5 Activities

- Product Backlog Refinement
- Sprint Planning
- Daily Scrum
- Sprint Review
- Sprint Retrospective

5 Values

- Courage
- Openness
- Focus
- Commitment
- Respect



3 Roles

- Development Team
- Product Owner
- ScrumMaster

3 Artifacts

- Product Backlog
- Sprint Backlog
- Increment

5 Activities

- Product Backlog Refinement
- Sprint Planning
- Daily Scrum
- Sprint Review
- Sprint Retrospective

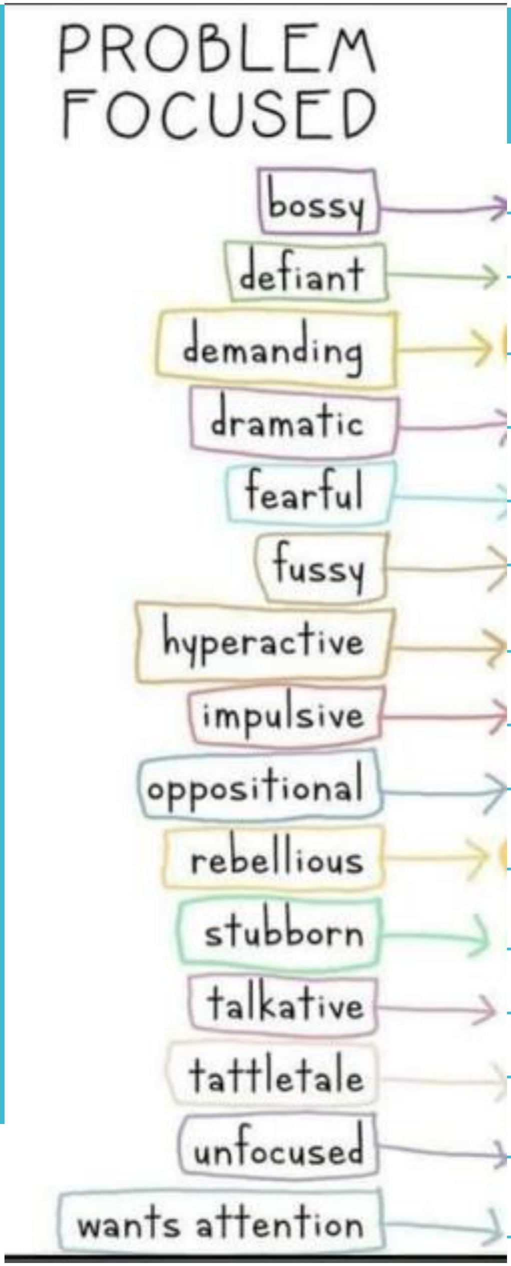
5 Values

- Courage
- Openness
- Focus
- Commitment
- Respect

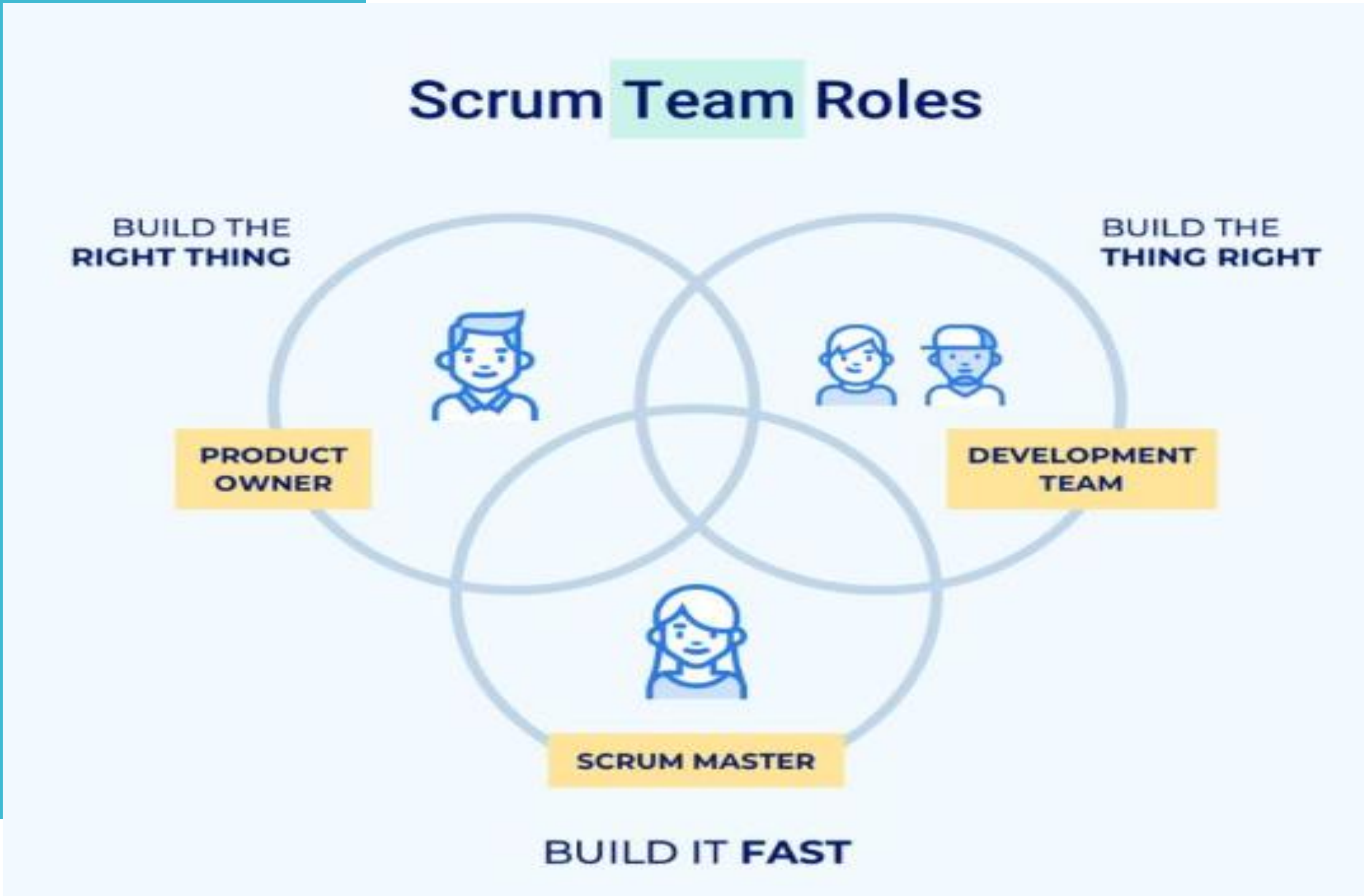
Scrum Framework

Are you PF or SF?

Problem Focused	Solution Focused
What's wrong	What's needed
What needs fixing	What’s work
Blame	Progress
Control	Influence
Causes in the past	‘Counters’ in the past
The expert knows all	Collaboration
Deficits and Weaknesses	Resources and Strengths
Complications	Simplicity
Definitions	Actions

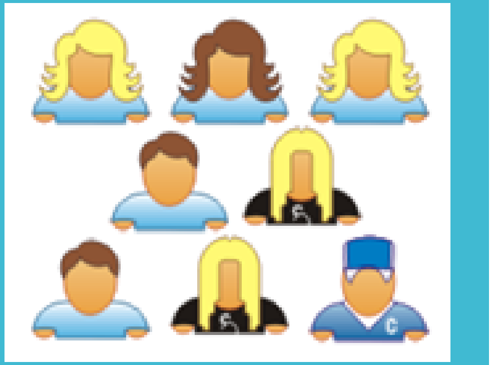


PROBLEM FOCUSED	Solution Focused
bossy	Natural Leader, Visionary
defiant	Holds strong beliefs, Bold, Determined
demanding	Knows what they want, Outspoken
dramatic	Expressive, Passionate
fearful	Cautious, Careful
fussy	Has strong preferences
hyperactive	Energetic, Enthusiastic, On the go
impulsive	Spontaneous, instinctive
oppositional	Advocates for a different perspective
rebellious	Is finding their ways
stubborn	Persistent, Determined, Steadfast
talkative	Enjoys Communicating
tattletale	Seeks Justice, Respects Rules
unfocused	Seeks Justice, Respects Rules
wants attention	Advocates for needs, Seeks Connection



Scrum Team Roles

The Development Team

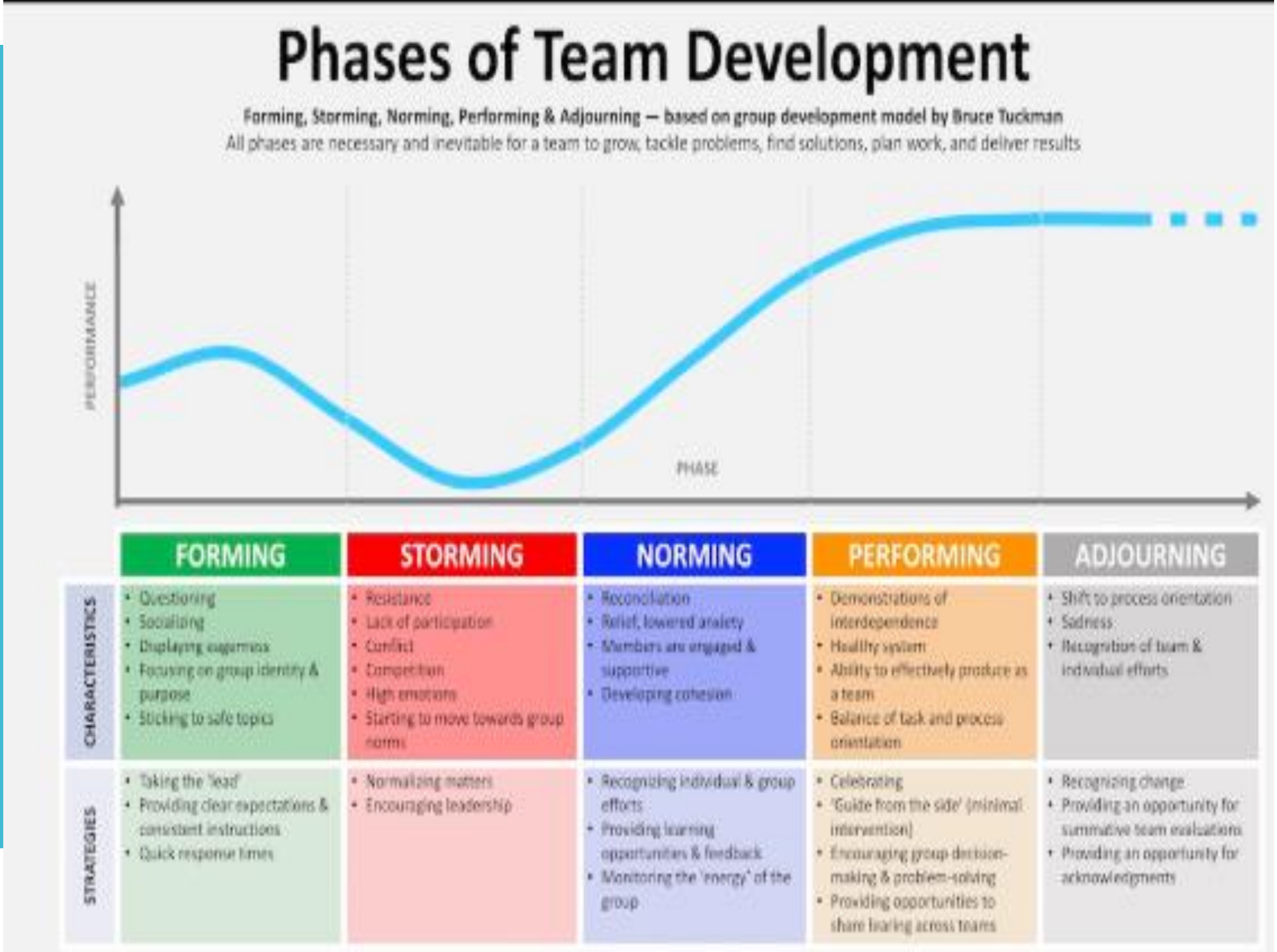


Characteristics

- Teams are self organizing,
- Ideally no titles but rarely a possibility
- Cross-Functional
- Self Managed
- Full time (100% dedicated) &
- sit together

Mission

- Decides how much work to take on in a sprint
- Collectively responsible for reaching the sprint goal and meeting the commitment
- Delivers PSPI each sprint without sacrificing quality and sustainable pace
- Manages the sprint backlog and keeps tracking the progress
- Makes continuers self-improvements



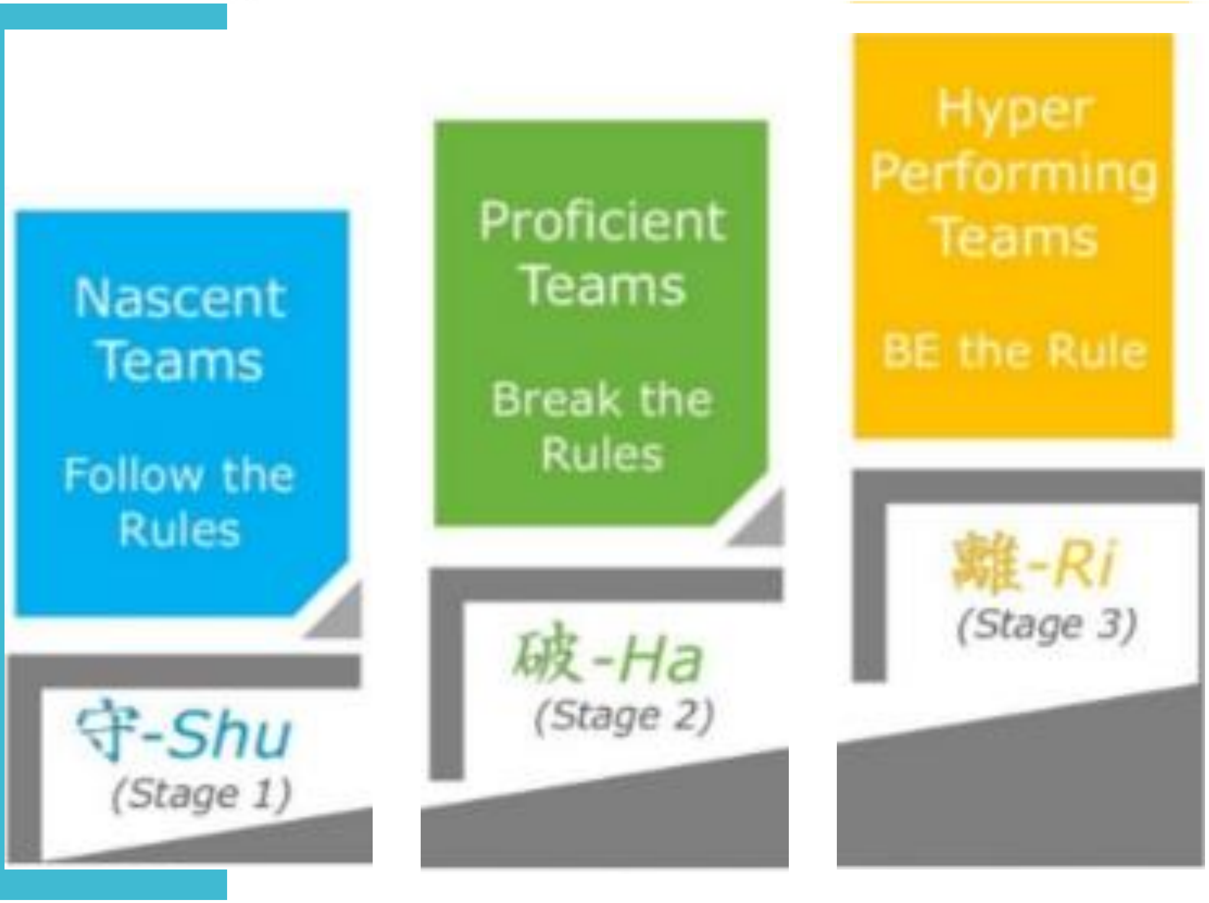
Phases of Team Development

Forming, Storming, Norming, Performing & Adjourning — based on group development model by Bruce Tuckman

All phases are necessary and inevitable for a team to grow, tackle problems, find solutions, plan work, and deliver results

	FORMING	STORMING	NORMING	PERFORMING	ADJOURNING
CHARACTERISTICS	• Questioning • Socializing • Displaying eagerness • Focusing on group identity & purpose • Sticking to safe topics	• Resistance • Lack of participation • Conflict • Competition • High emotions • Starting to move towards group norms	• Reconciliation • Relief, lowered anxiety • Members are engaged & supportive • Developing cohesion	• Demonstrations of interdependence • Healthy system • Ability to effectively produce as a team • Balance of task and process orientation	• Shift to process orientation • Sadness • Recognition of team & individual efforts
STRATEGIES	• Taking the 'lead' • Providing clear expectations & consistent instructions • Quick response times	• Normalizing matters • Encouraging leadership	• Recognizing individual & group efforts • Providing learning opportunities & feedback • Monitoring the 'energy' of the group	• Celebrating • 'Guide from the side' (minimal intervention) • Encouraging group decision-making & problem-solving • Providing opportunities to share learning across teams	• Recognizing change • Providing an opportunity for summative team evaluations • Providing an opportunity for acknowledgments

Shu-Ha-Ri
Applied to Agile team



Nascent Teams

Follow the Rules

守-Shu
(Stage 1)

Proficient Teams

Break the Rules

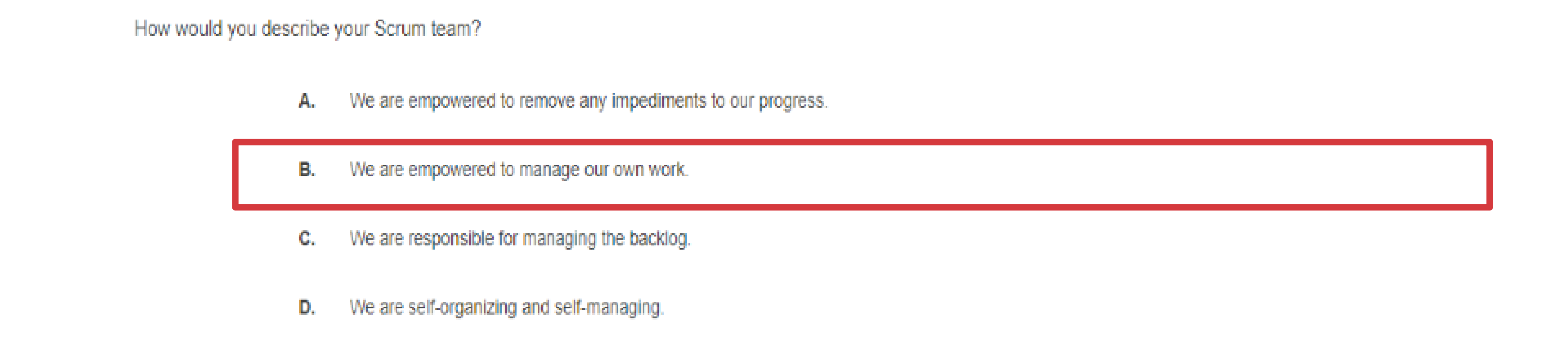
破-Ha
(Stage 2)

Hyper Performing Teams

BE the Rule

離-Ri
(Stage 3)

Source: http://www.slideshare.net/cognizant/shuhari-measuring-agile-adoption-maturity



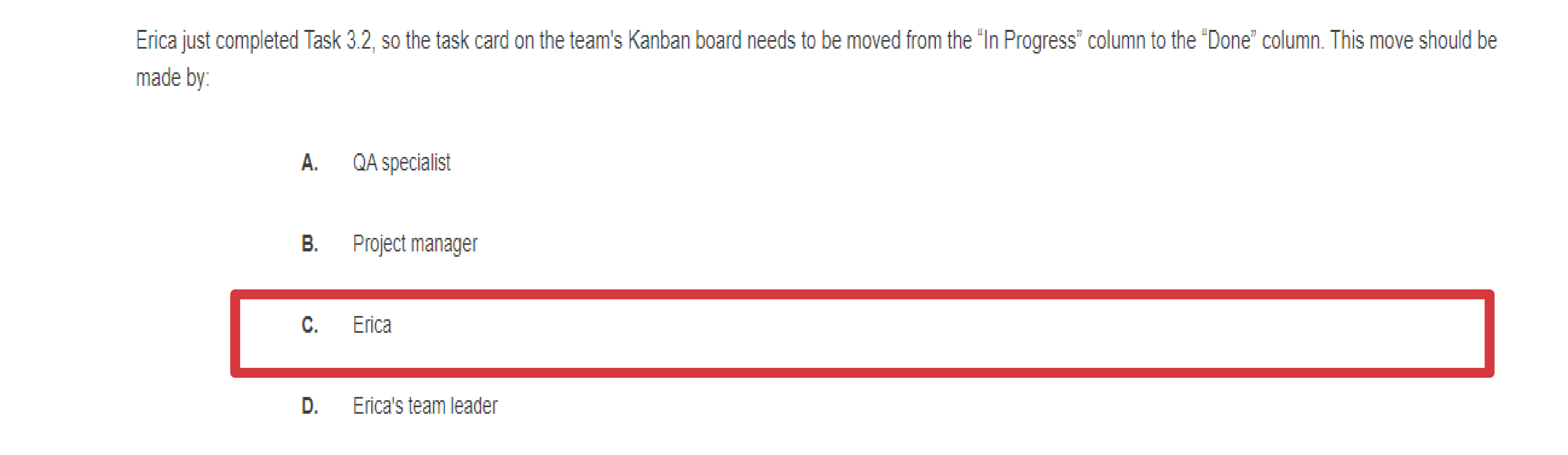
How would you describe your Scrum team?

A. We are empowered to remove any impediments to our progress.

[✓ **B.** We are empowered to manage our own work.]

C. We are responsible for managing the backlog.

D. We are self-organizing and self-managing.



Erica just completed Task 3.2, so the task card on the team's Kanban board needs to be moved from the "In Progress" column to the "Done" column. This move should be made by:

A. QA specialist

B. Project manager

[✓ **C.** Erica]

D. Erica's team leader

BOLD

Product Owner

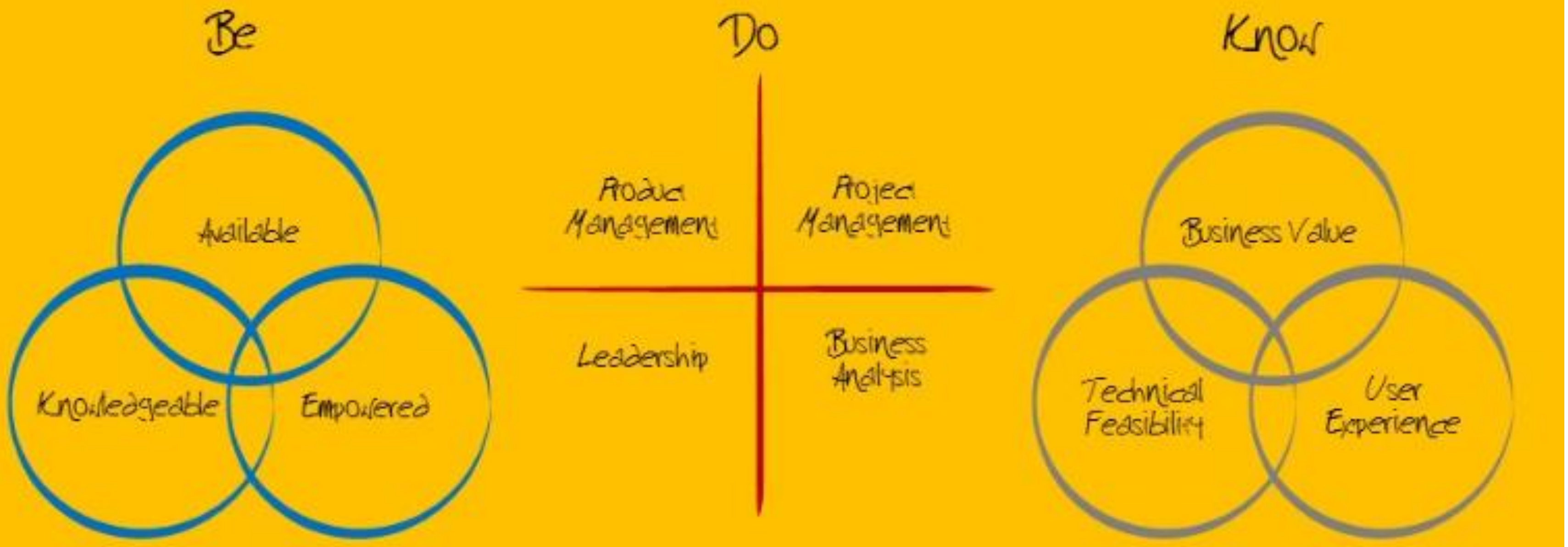
Characteristics

- One person playing the role
- Drives product success
- Represents project to the stakeholders
- Represents stakeholders to the project
- Collaborates with everyone
- Typically played by customer or customer representative
- Part of the team - tightly engaged through the sprint

Mission

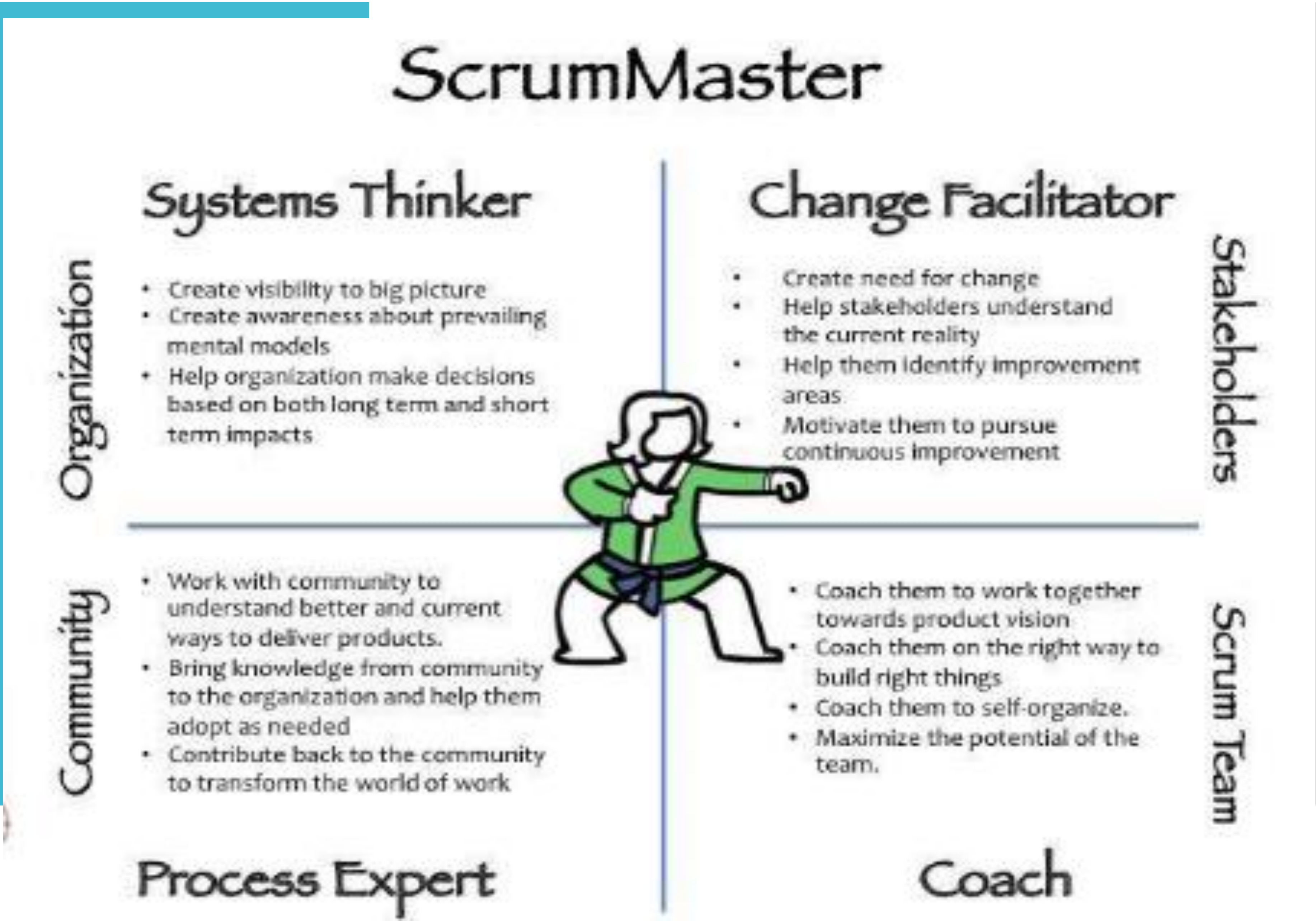
- Creates product vision
- Defines the feature of the product
- Responsible for ROI (Return of Investment)
- Prioritizes feature according to market value
- Adjusts feature/priority according to the market feedback
- Accepts/rejects work result
- Ensures the readiness of sprint input

What a Product Owner Should Be, Do, and Know



MasteringBusinessAnalysis.com

CRACK



ScrumMaster

Systems Thinker

Organization

- Create visibility to big picture
- Create awareness about prevailing mental models
- Help organization make decisions based on both long term and short term impacts

Change Facilitator

Stakeholders

- Create need for change
- Help stakeholders understand the current reality
- Help them identify improvement areas
- Motivate them to pursue continuous improvement

Community

- Work with community to understand better and current ways to deliver products.
- Bring knowledge from community to the organization and help them adopt as needed
- Contribute back to the community to transform the world of work

Process Expert

Scrum Team

- Coach them to work together towards product vision
- Coach them on the right way to build right things
- Coach them to self-organize.
- Maximize the potential of the team.

Coach

Scrum Master

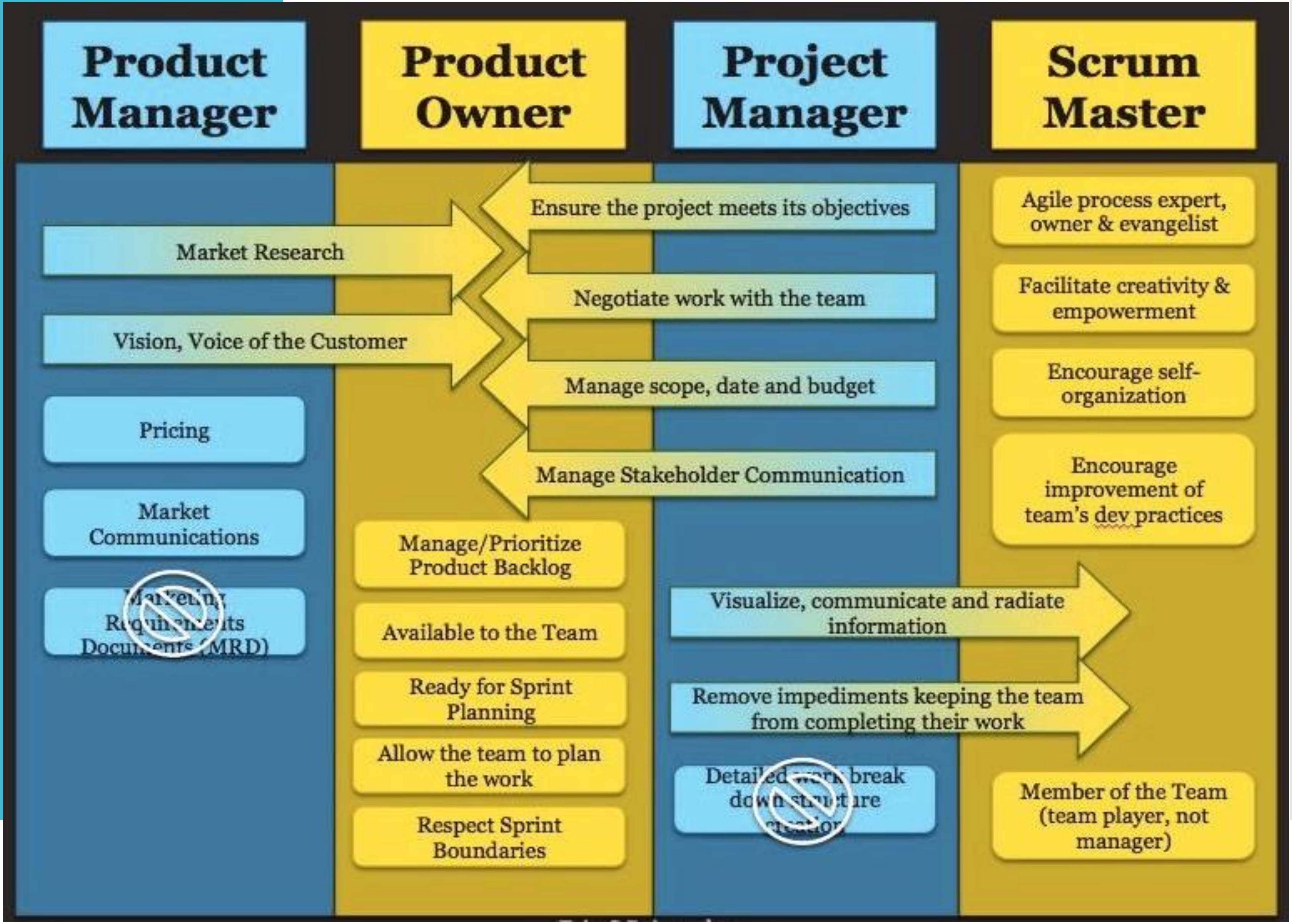
Characteristics

- Represent management to the project
- Responsible for enacting Scrum values and practices
- Removes impediments
- Ensures team is fully functional and productive
- Shield the team from external interferences
- Process check master
- Performance feedback

Mission

- Helps the Team remove obstacles and impediments
- Protects the Team from disruption and other threats
- Coaches the Team on their practices to make continuous improvements
- Facilitates the interactions within the Team/between the Team and the PO
- Teaches Scrum to the Team, PO and other people
- Being a change agent in growing the organization to deliver early and often &
- remove waste

Overlapping Responsibilities



Product Manager

- Market Research
- Vision, Voice of the Customer
- Pricing
- Market Communications
- Marketing Requirements Documents (MRD)

Product Owner

- Manage/Prioritize Product Backlog
- Available to the Team
- Ready for Sprint Planning
- Allow the team to plan the work
- Respect Sprint Boundaries

Project Manager

- Ensure the project meets its objectives
- Negotiate work with the team
- Manage scope, date and budget
- Manage Stakeholder Communication
- Visualize, communicate and radiate information
- Remove impediments keeping the team from completing their work
- Detailed work break down structure creation

Scrum Master

- Agile process expert, owner & evangelist
- Facilitate creativity & empowerment
- Encourage self-organization
- Encourage improvement of team's dev practices
- Member of the Team (team player, not manager)

RICH

3 Roles

- Development Team
- Product Owner
- ScrumMaster

3 Artifacts

- Product Backlog
- Sprint Backlog
- Increment

5 Activities

- Product Backlog Refinement
- Sprint Planning
- Daily Scrum
- Sprint Review
- Sprint Retrospective

5 Values

- Courage
- Openness
- Focus
- Commitment
- Respect

Product Backlog

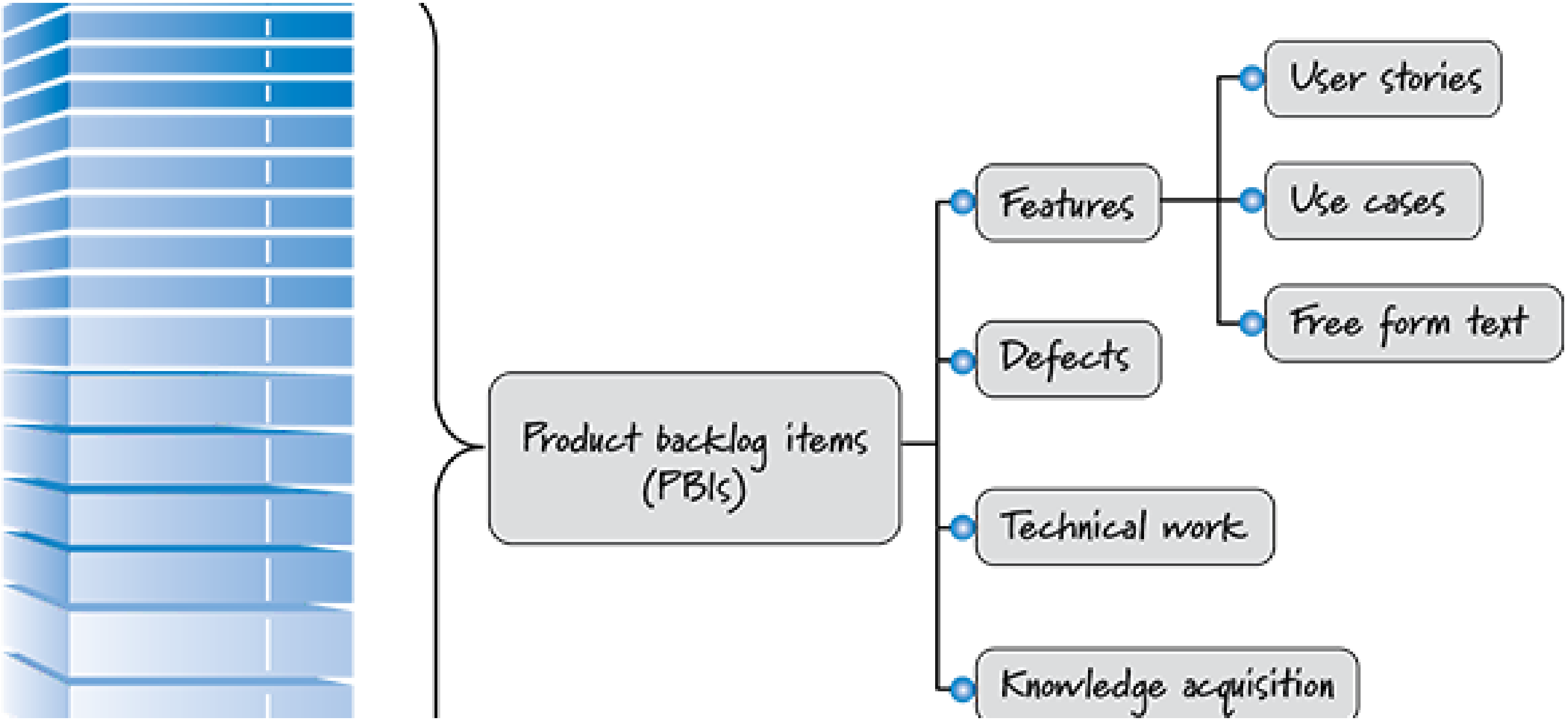
Product Backlog

Requirements

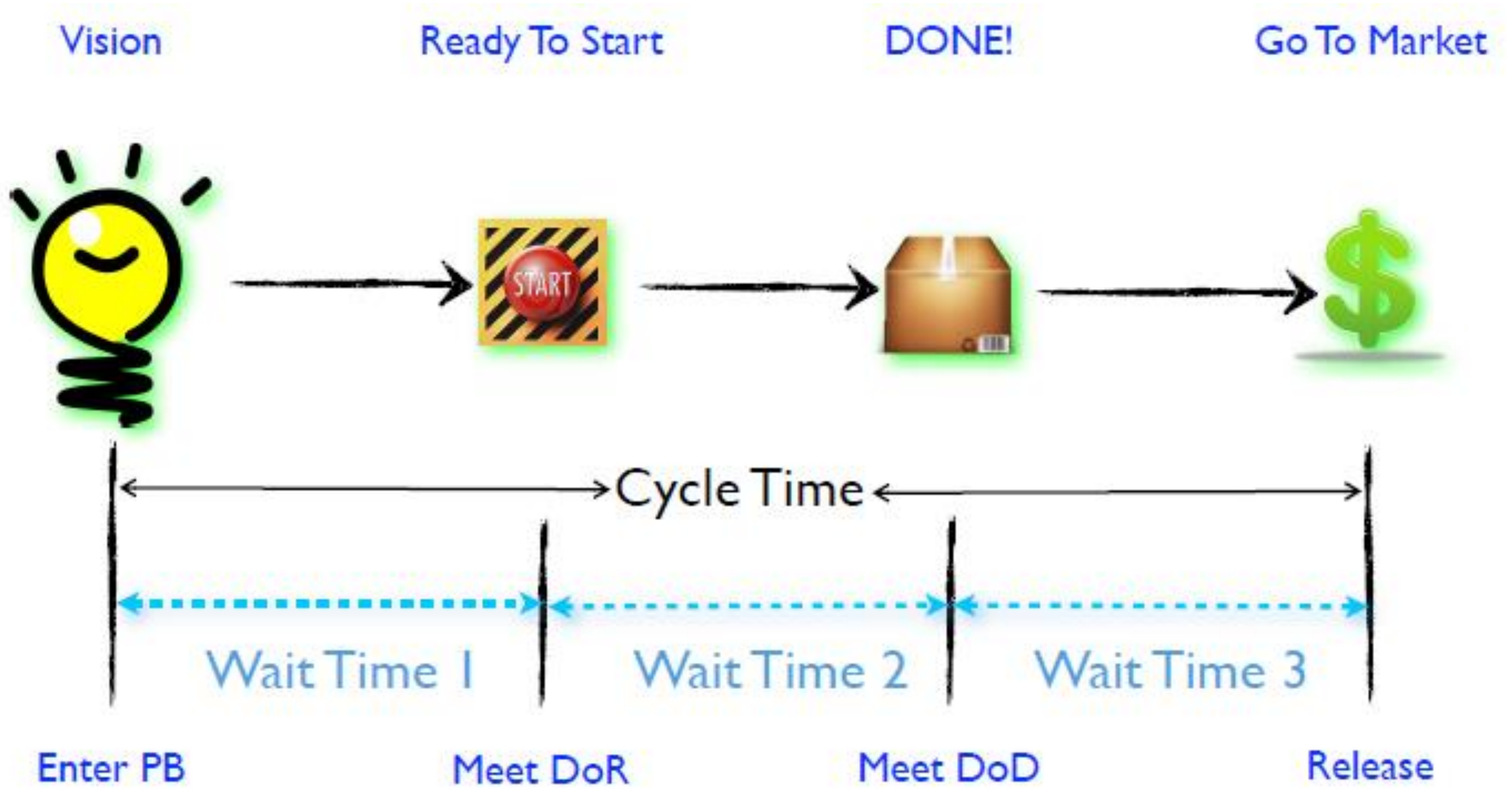
A list of all desired work on the project

Prioritized by the Product Owner

Reprioritized at the start of each sprint



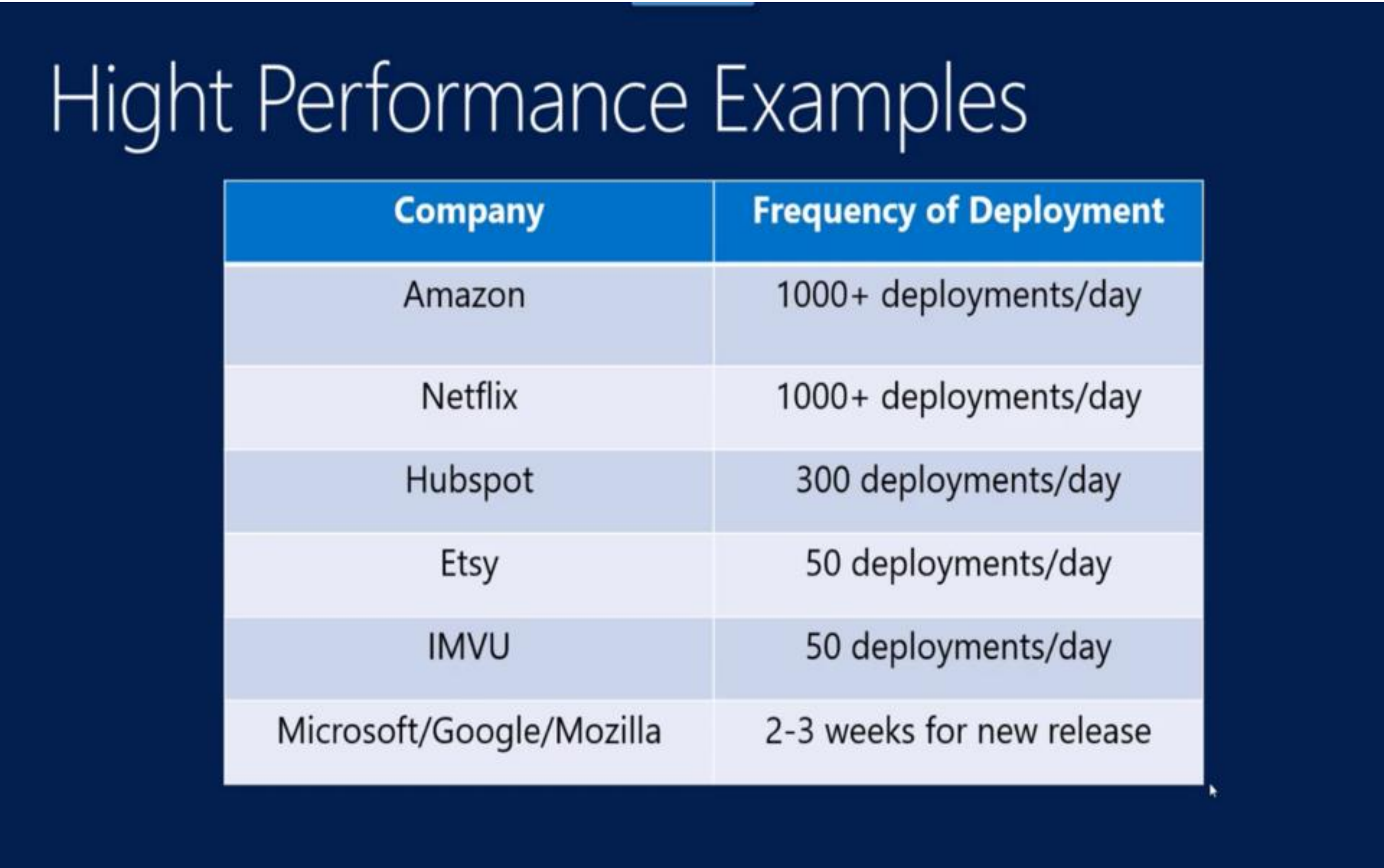
Product Backlog



Cycle Time for a feature

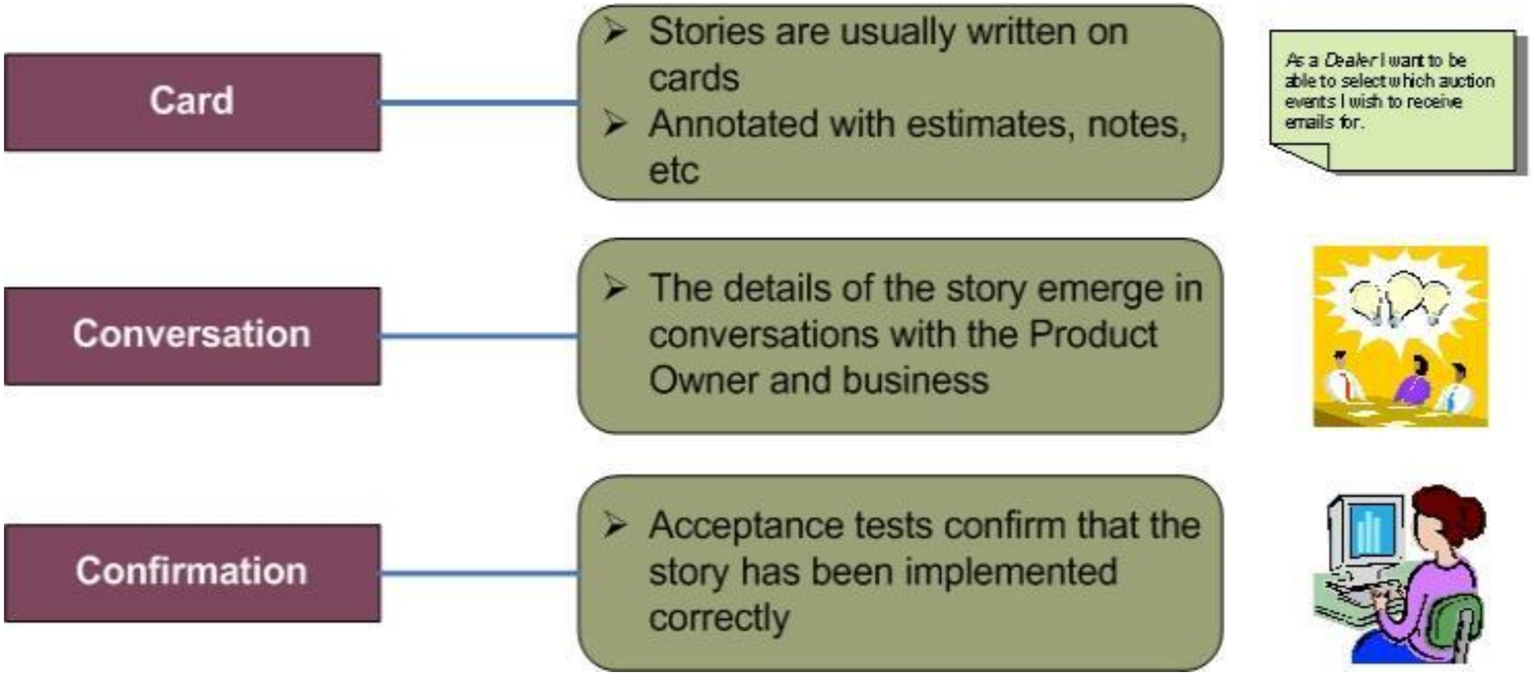
Amazon Deploys to Production Every 11.6 Seconds

Source: http://joshuaseiden.com/blog/2013/12/amazon-deploys-to-production-every-11-6-seconds/



Hight Performance Examples

Company	Frequency of Deployment
Amazon	1000+ deployments/day
Netflix	1000+ deployments/day
Hubspot	300 deployments/day
Etsy	50 deployments/day
IMVU	50 deployments/day
Microsoft/Google/Mozilla	2-3 weeks for new release



Card
- Stories are usually written on cards
- Annotated with estimates, notes, etc

As a *Dealer* I want to be able to select which auction events I wish to receive emails for.

Conversation
- The details of the story emerge in conversations with the Product Owner and business

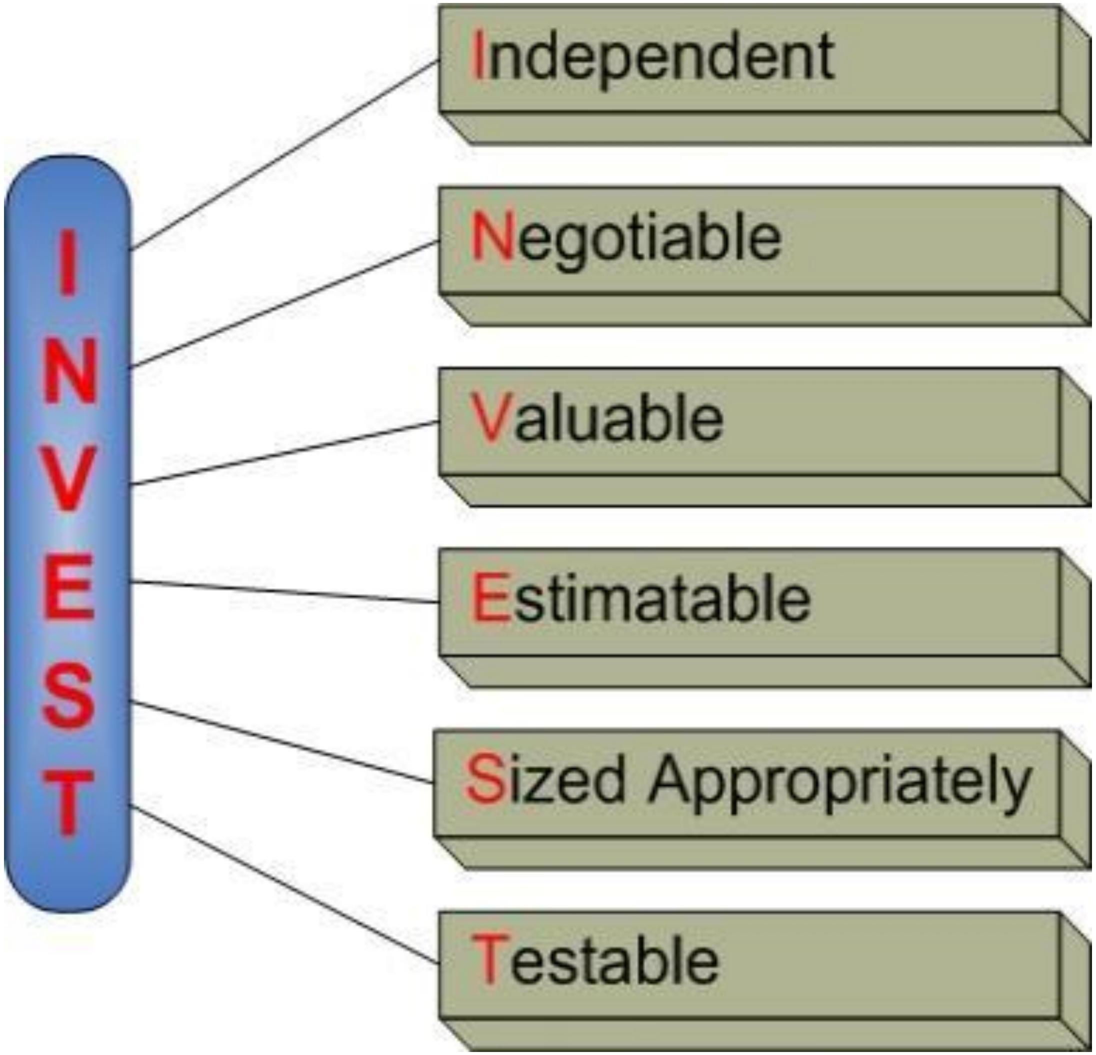
Confirmation
- Acceptance tests confirm that the story has been implemented correctly

User Stories

User Stories are an established method of clear communication between

the team & the business

What makes a good story?



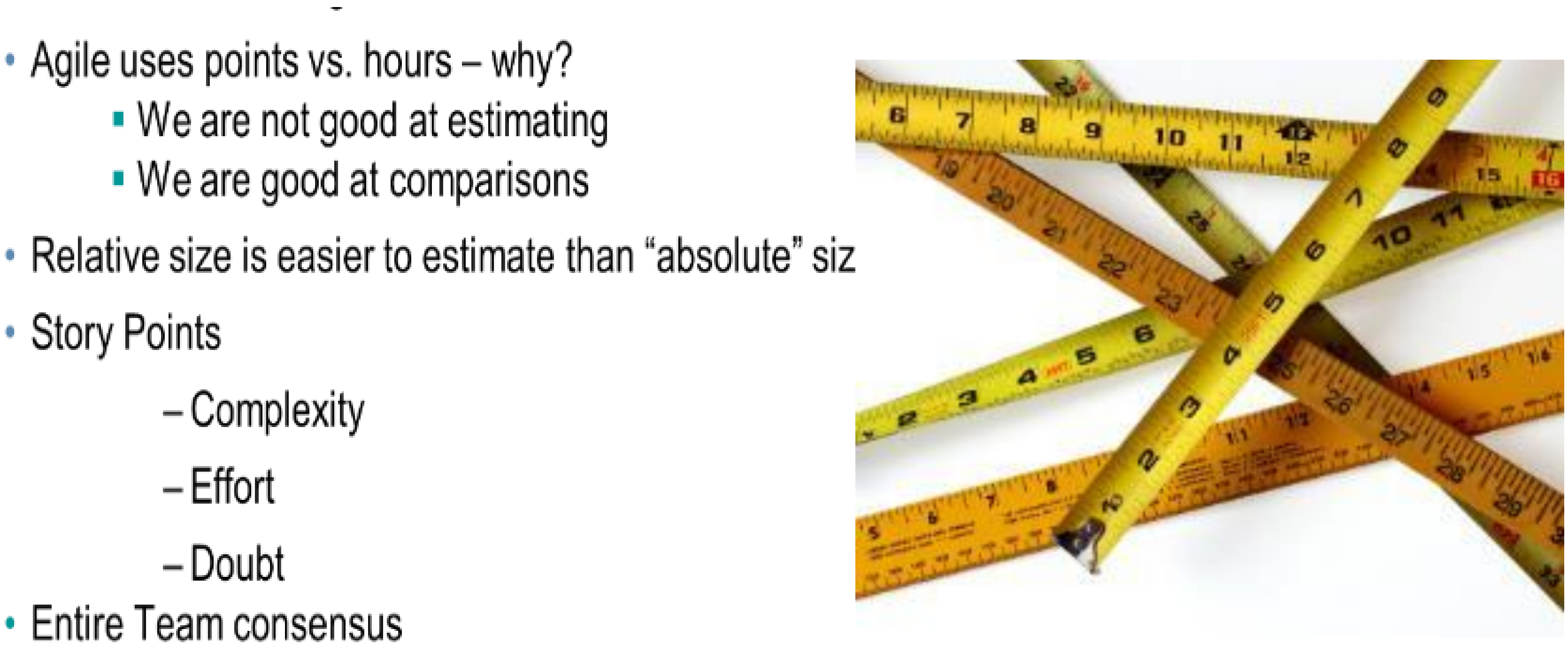
INVEST

- Independent
- Negotiable
- Valuable
- Estimatable
- Sized Appropriately
- Testable

Ref: www.mountaingoatsoftware.com

INVEST acronym can be found in Mike Cohn's "User Stories Applied"

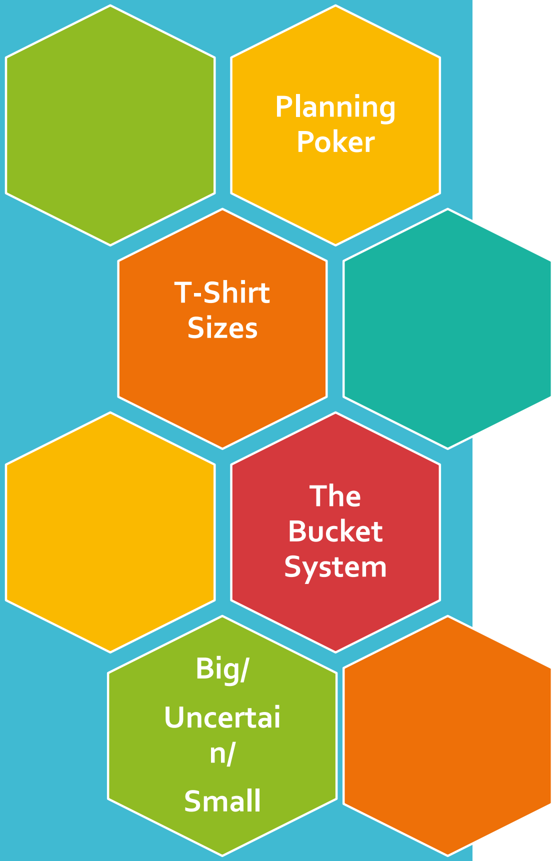
Story Points - Estimating the Size



- Agile uses points vs. hours – why?
  - We are not good at estimating
  - We are good at comparisons
- Relative size is easier to estimate than “absolute” siz
- Story Points
  - – Complexity
  - – Effort
  - – Doubt
- Entire Team consensus

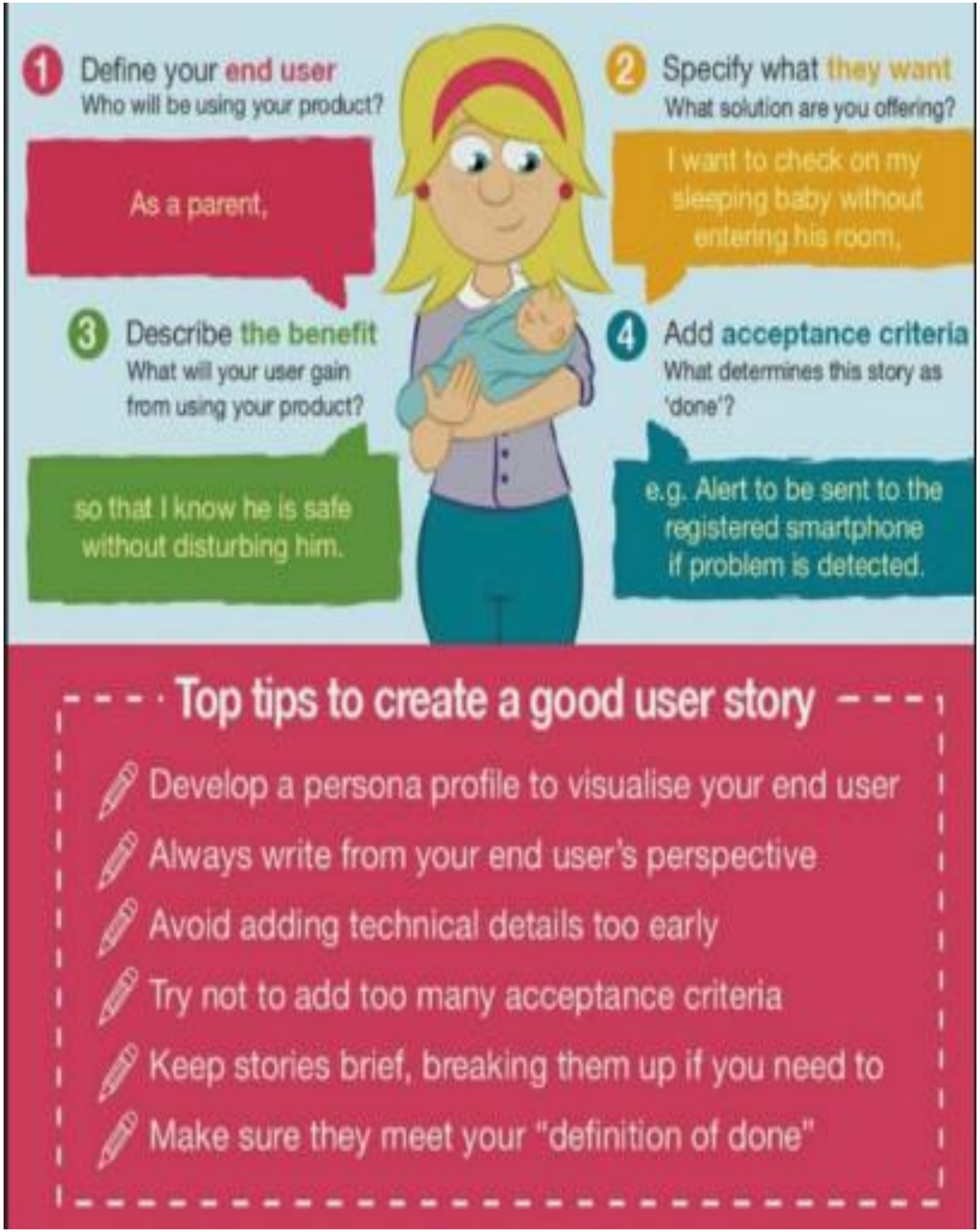
Ref: Internal – Org Library

Agile Estimation Techniques



- Many people have used a variation of Planning Poker to do Agile estimation.
- Here is a reference of **9 different Agile estimation techniques** for different circumstances.
- I have seen all of these techniques work in practice, except one.
- Try a new one each Sprint!

How to write an Agile User Story



1 Define your **end user**
Who will be using your product?

As a parent,

2 Specify what **they want**
What solution are you offering?

I want to check on my sleeping baby without entering his room,

3 Describe **the benefit**
What will your user gain from using your product?

so that I know he is safe without disturbing him.

4 Add **acceptance criteria**
What determines this story as 'done'?

e.g. Alert to be sent to the registered smartphone if problem is detected.

Top tips to create a good user story

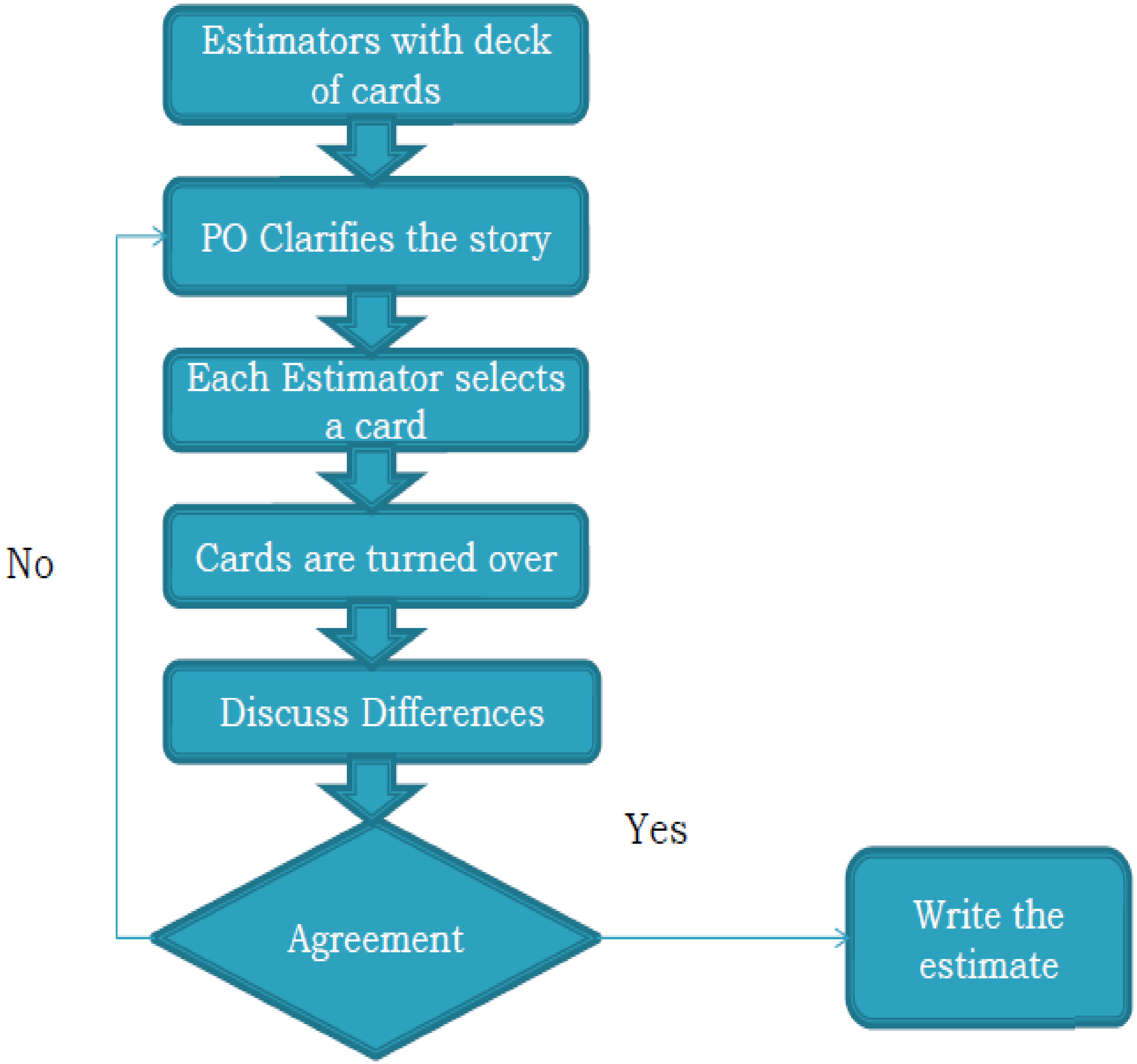
- Develop a persona profile to visualise your end user
- Always write from your end user's perspective
- Avoid adding technical details too early
- Try not to add too many acceptance criteria
- Keep stories brief, breaking them up if you need to
- Make sure they meet your "definition of done"

Planning poker [www.pointingpoker.com]

- A variation of the wide band Delphi method
- Team participates and user story points or ideal time
- The moderator(*) reads out the story (*product owner)
- Team discusses and each estimator turns over a card that represents their estimate
- High and low estimates are reconciled/clarified
- Future rounds show convergence, otherwise choose one of:
  - Choose majority estimate
  - Choose high estimate(s)
  - Average of the estimates
  - Adopt three point (PERT) averaging
    - [(P+ 4M + O) /6], Pessimistic, Most Likely, Optimistic

Planning Poker

Estimation flow



Estimators with deck of cards

PO Clarifies the story

Each Estimator selects a card

Cards are turned over

Discuss Differences

Agreement

No

Yes

Write the estimate

Multiple Vendors: Dev + QA + ALM + DevOps + UI +…

When multiple agile teams work on a project, they benefit from a common baseline for estimating story points.

The Product Owner introduces the user story by talking about:

- The motivation for doing the user story.
- The intended outcome and benefits.
- The scope of the user story (including what is not in scope with this user story).
- Other relevant considerations.

At this point, the team can ask questions like:

- What should happen in a given scenario?
- What should happen in some negative case or edge case (i.e. plausible but not common scenario)?
- Do we need to build this for one type, several user types or all users?
- Do we need to track any new performance metrics in order to understand if this user story is working as expected?

Of course, the team can ask as many questions as they want! These are just ideas of the type of things the team will ask the Product Owner.

Definition of "Ready"- DoR

- All the acceptance criteria is defined
- Story is fully groomed with the whole team including but not limited to Dev, QA, DA, DM, Architect, PO, and appropriate SMEs
- QA, Performance and UAT representatives should have had an opportunity to voice their concerns and provide input.
- Story is sized using the Normalized Uses Story Guidance
- Story can feasibly be completed within the sprint, including all tests, bug remediation, artifacts required, and enough time to demo the story successfully
- The tasks should be clear, time box-able, and as a collective, the tasks should add-up to the story point size being delivered.
- Dependencies and Risks are identified and captured
- Within TFS, stories with dependencies to other stories are to be tied using successor/predecessor linkages
- **A story cannot be pulled into a sprint unless it will be free of dependencies during the sprint and have sufficient time to complete all tasks and demo successfully**

Examples of DoR…

- Big Picture/Why are we doing it?
- Description?
- Any specific NFRs for this story?
- Acceptance Criteria listed
- 8 SP or under
- Agreed Acceptance Criteria
- Resources in Story, API URLs, Contracts, documents if known
- Every Team member understands story
- How will we test it?
- What is NOT in scope for this story?
- List any existing questions
- Unknowns accepted or removed
- Assumptions Listed and accepted by team
- Design review task (does it need one?)
- Known refactoring included? tests that need adding? changing?
- Known tech debt we need to fix?

Definition of "Done" -DoD

- Meets all acceptance criteria
- All coding has completed related to User Story
  - Coding Standards are followed
  - Code Review suggestions are either implemented or scheduled or future implementation
- All tasks are completed
- User stories are tested in appropriate environments
- No critical or high severity defects
- Required documentation
- Story has been successfully demo'ed (reviewed) and acceptance to the Product Owner

Good
Product Backlog

A DYNAMIC list of functionalities the product **MIGHT** include

Good Product backlog should be **DEEP**

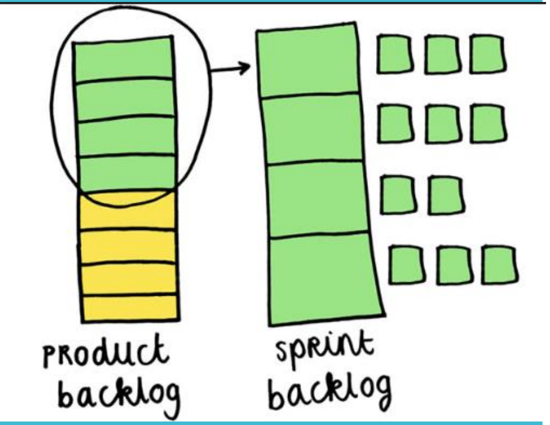
- ***Detailed appropriately***
- ***Emergent***
- ***Estimated***
- ***Prioritized***

Open to all but ultimately groomed by Product Owner

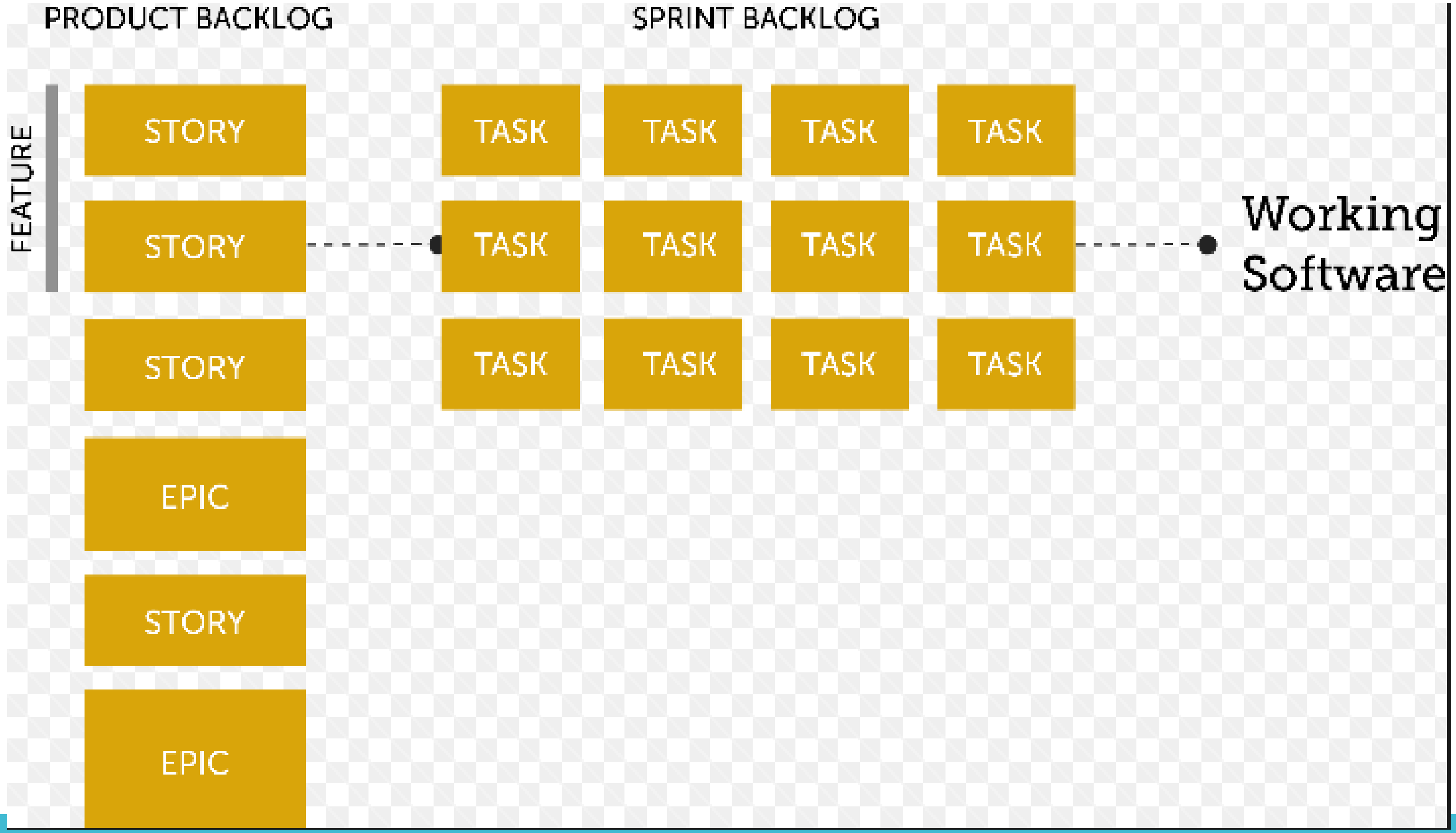
- **Coined by Roman Pichler and Mike**

DEEP

Sprint Backlog



- Sprint Backlog defines the work that the team will perform to turn selected Product Backlog into a "Done" increment
- The list emerges during the sprint
- Each task has information about estimated amount of work remaining on the task on any given day during the sprint



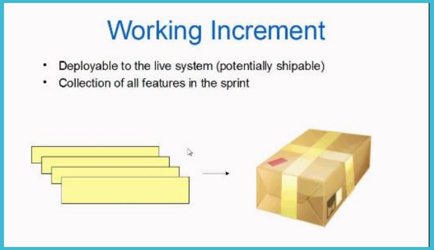
Sprint Backlog

Sprint Planning Meeting
Daily Scrum
Update product backlog
Scrum Process
Product increment
Sprint Retrospective
Sprint Review

Product Increment

- The sum of all the product backlog items completed during a sprint and all previous sprints.
- At the end of a sprint, the increment must be complete.

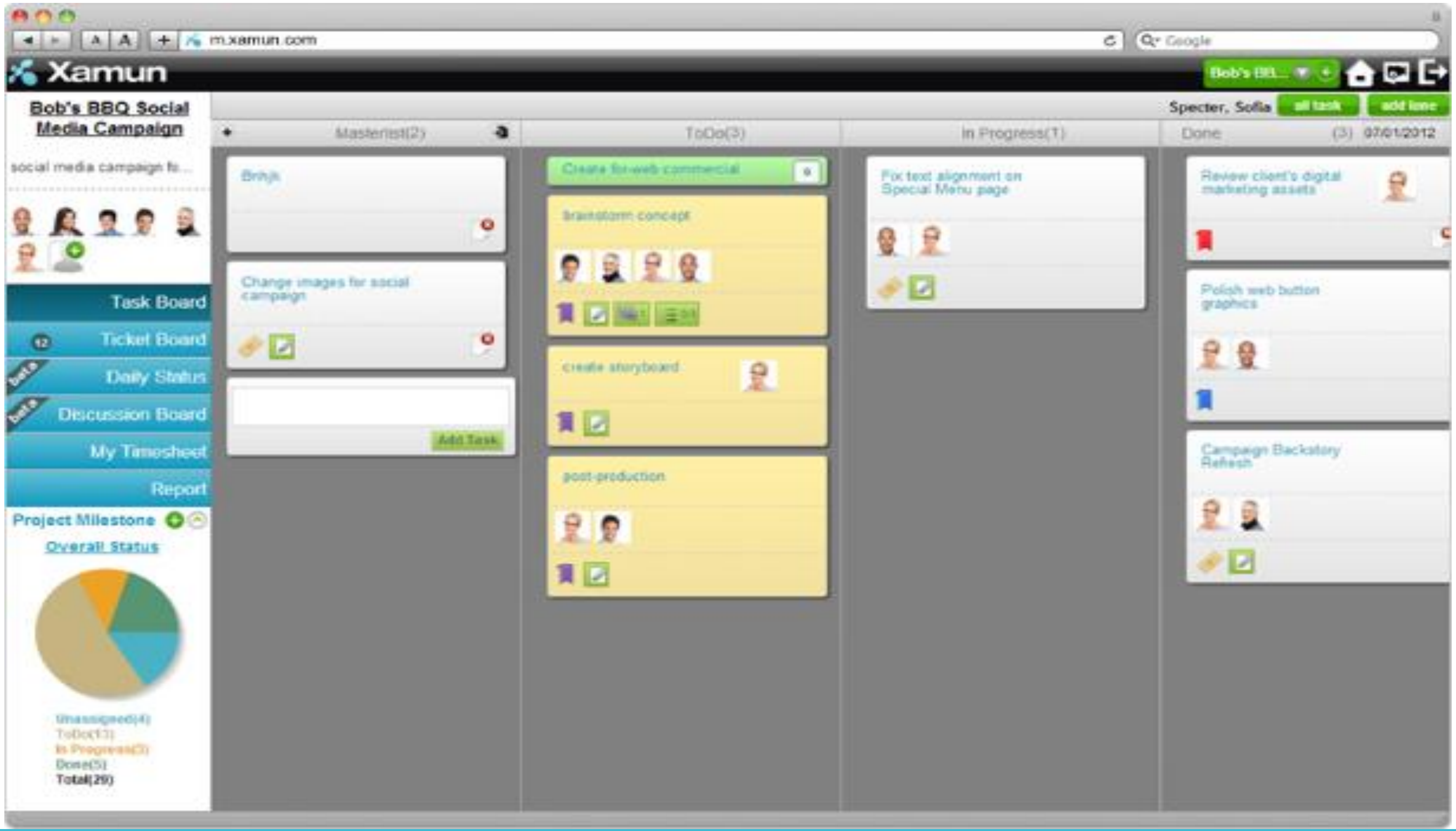
Increment



Working Increment

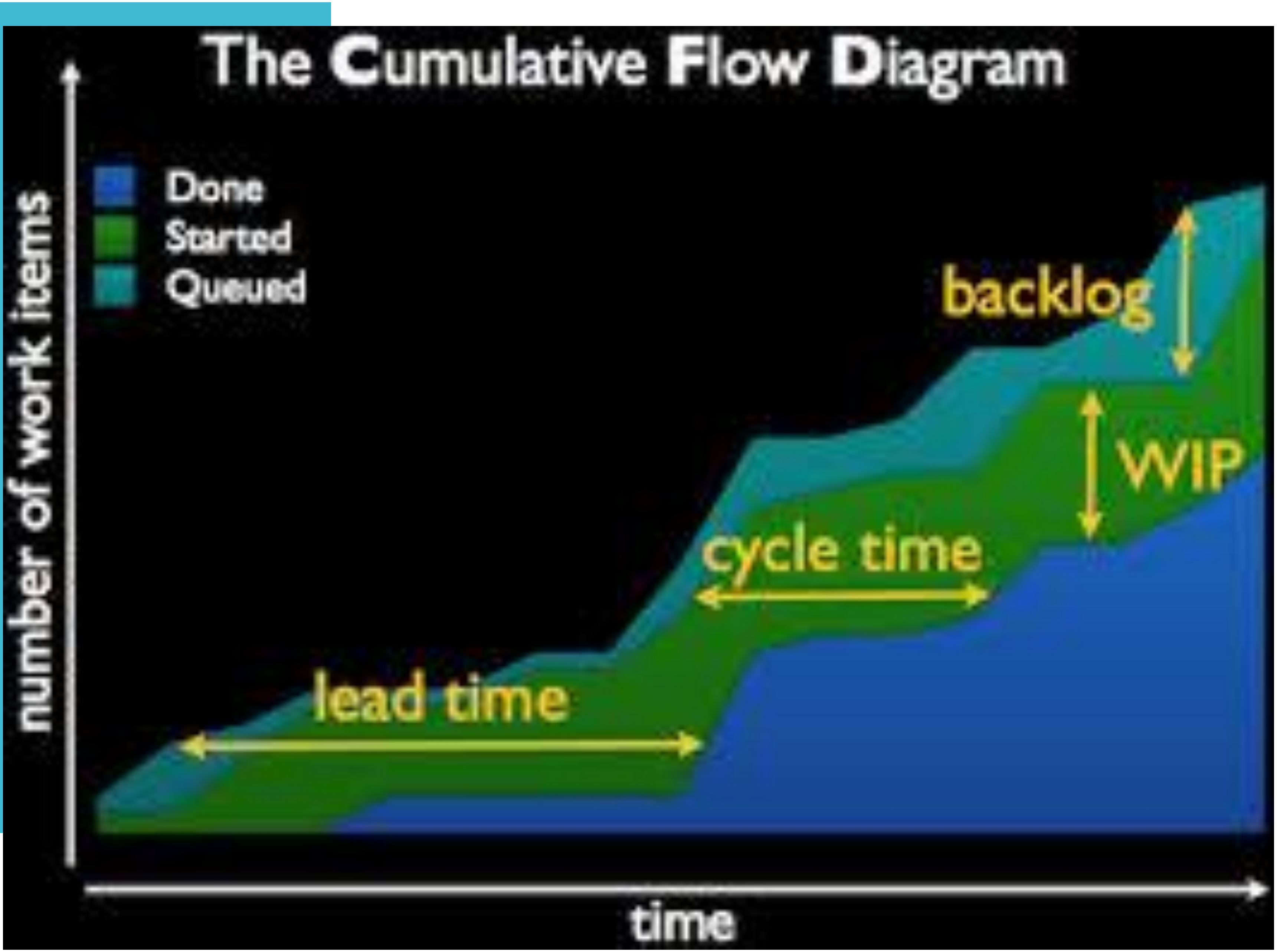
- Deployable to the live system (potentially shipable)
- Collection of all features in the sprint

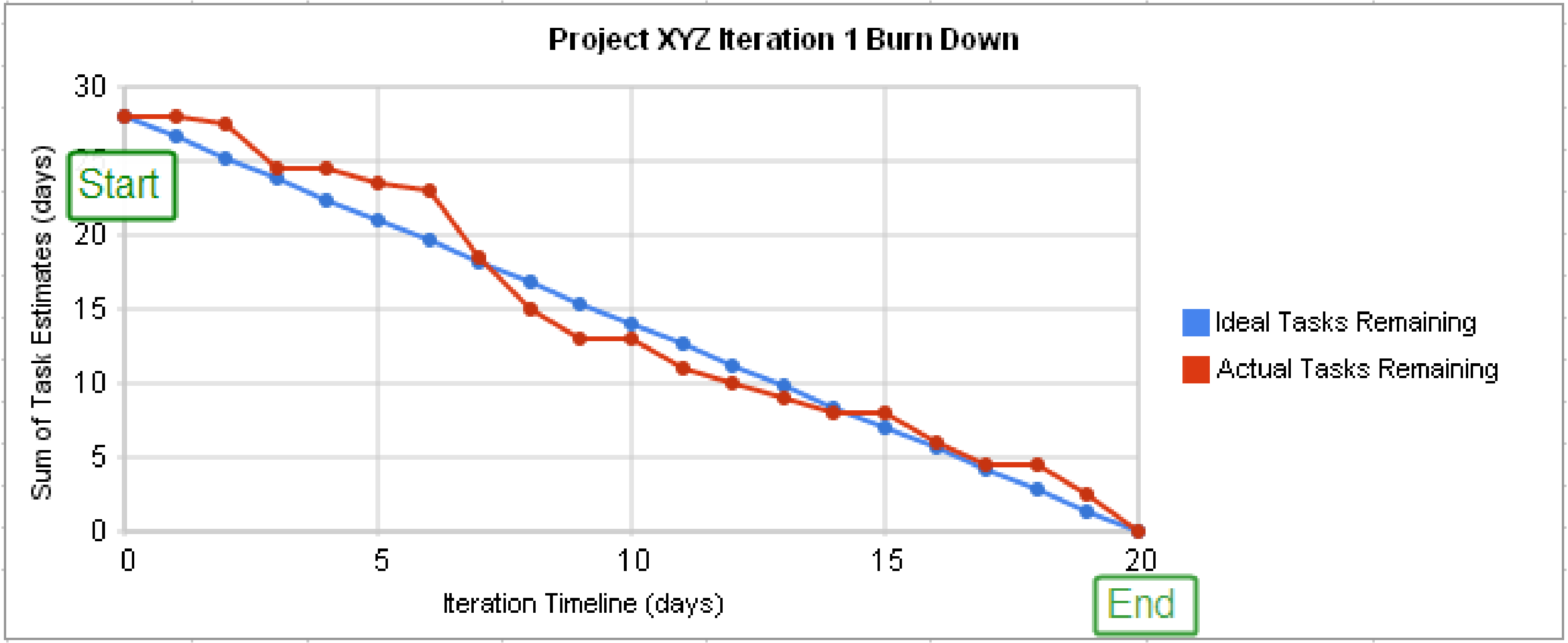
- Increment is the sum of all Product Backlog items completed during a sprint and all previous sprints
- At the end of a Sprint, the new Increment should "Done"
- It must be useable condition (Potentially Shippable Product)
- Release increments early and frequently Vs deliver the finished product in one go



Increments-Task Board

- Update in real time by picking work then move them on status change
- Track **DONE** only by having features only


The Cumulative Flow Diagram
Done
Started
Queued
number of work items
time
lead time
cycle time
WIP
backlog



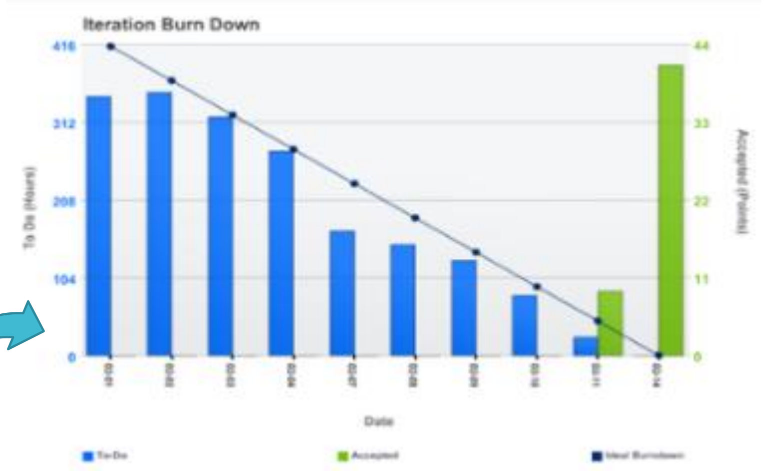
Increments
Burn Down Charts

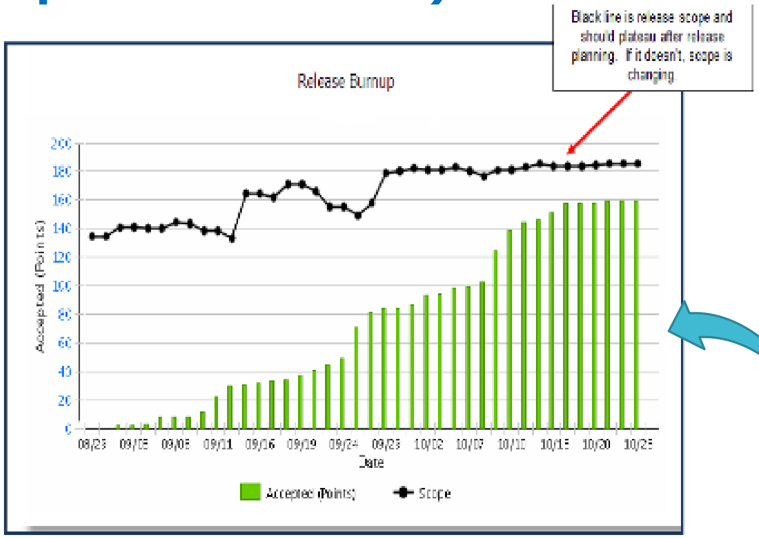
Updated daily, usually during the daily scrum

Represents the total amount of work remaining

Track **DONE** only

Burn Charts (Up & Down)

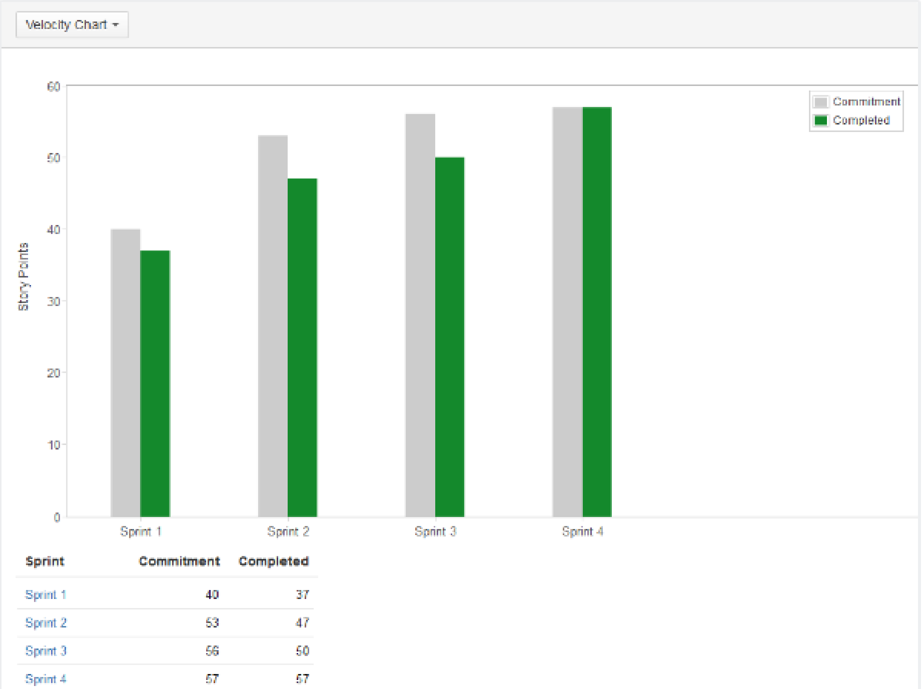



Black line is release scope and should plateau after release planning. If it doesn't, scope is changing

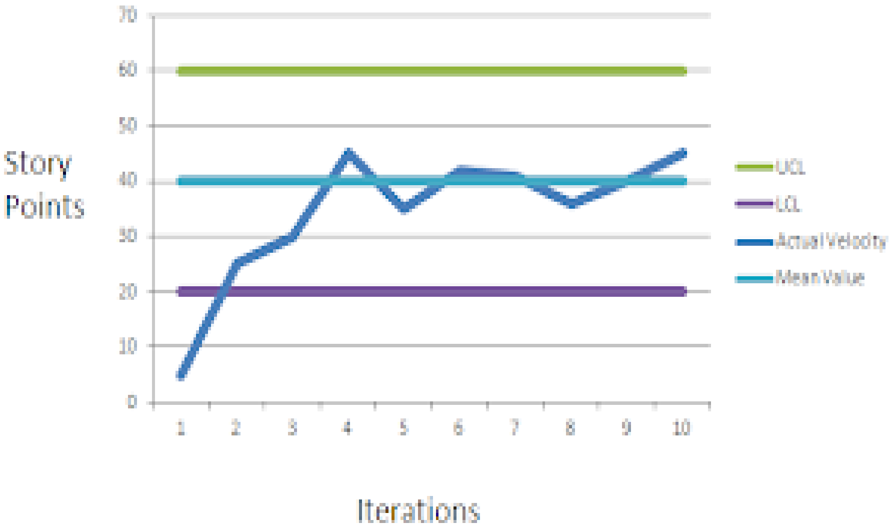
- A **burn down** chart is a graphical representation of work left to do versus time.
- The outstanding work (or backlog) is often on the vertical axis, with time along the horizontal.
- That is, it is a run chart of outstanding work. It is useful for predicting when all of the work will be completed.

- A **burn up** chart, tracks progress towards a projects completion.
- The Release Burnup displays the work delivered so far in the release versus the scope planned.
- It is reviewed to proactively (after every sprint optimally) to anticipate whether the release scope will be delivered.

Ref: Internal – Org Library

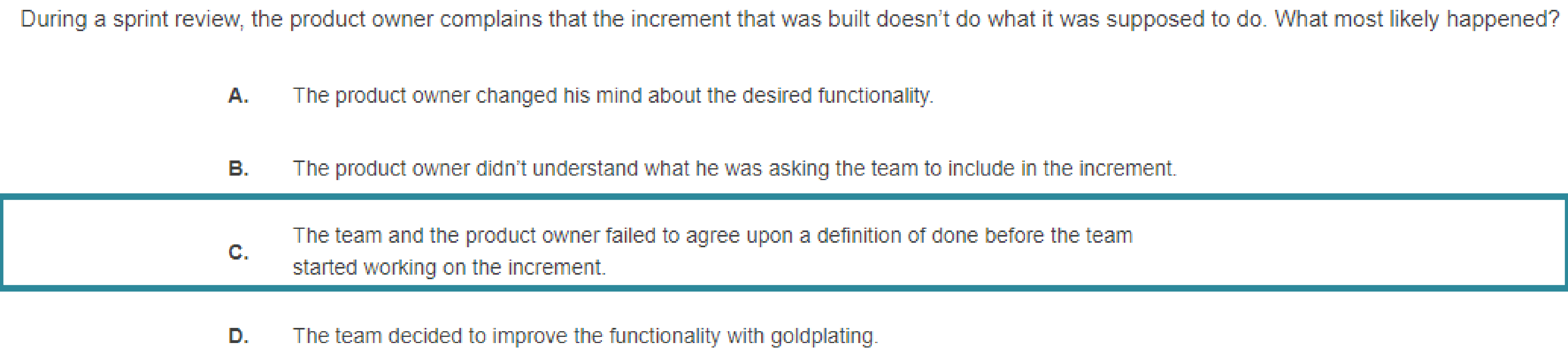


Sprint	Commitment	Completed
Sprint 1	40	37
Sprint 2	53	47
Sprint 3	56	50
Sprint 4	57	57



Velocity

is a measure of work the **Team** completes during a single Sprint and is calculated at Sprints end by totaling the Points of completed User Stories



During a sprint review, the product owner complains that the increment that was built doesn't do what it was supposed to do. What most likely happened?

A. The product owner changed his mind about the desired functionality.

B. The product owner didn't understand what he was asking the team to include in the increment.

[✓ C. The team and the product owner failed to agree upon a definition of done before the team started working on the increment.]

D. The team decided to improve the functionality with goldplating.

3 Roles

- Development Team
- Product Owner
- ScrumMaster

3 Artifacts

- Product Backlog
- Sprint Backlog
- Increment

5 Activities

- Product Backlog Refinement
- Sprint Planning
- Daily Scrum
- Sprint Review
- Sprint Retrospective

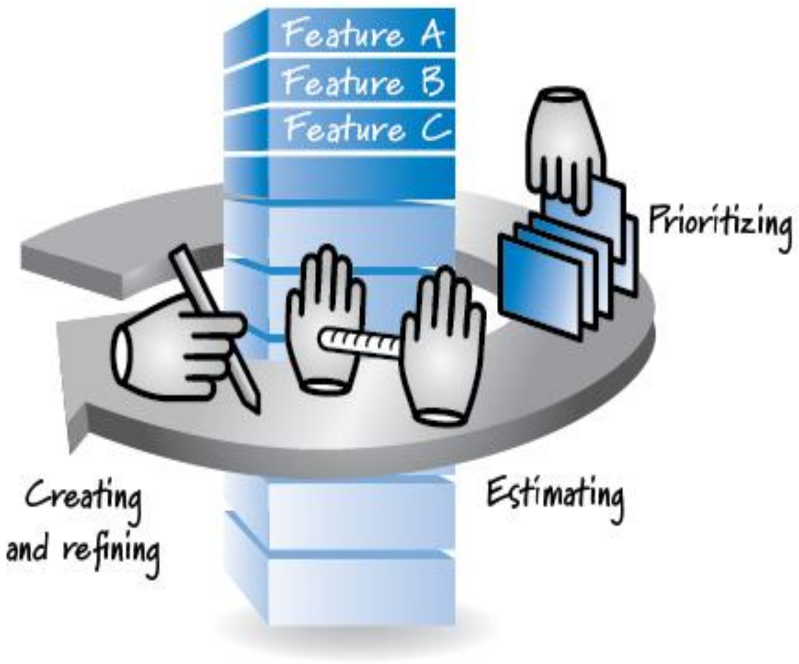
5 Values

- Courage
- Openness
- Focus
- Commitment
- Respect

Product Backlog Refinement

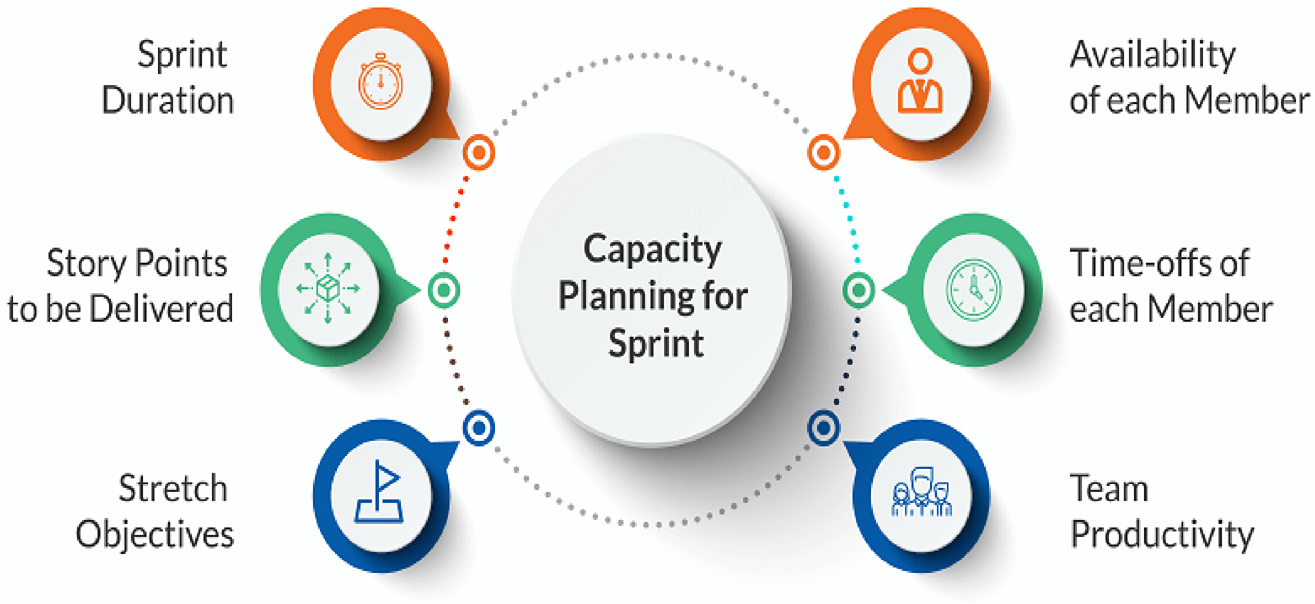
- Keep the Product Backlog in order
- Remove or demote Product Backlog Items that no longer seem important
- Add or promote Product Backlog Items that raise or become more important
- Split Product Backlog Items into smaller items
- Merge Product Backlog Items into larger items
- Estimate Product Backlog

Product Backlog refinement

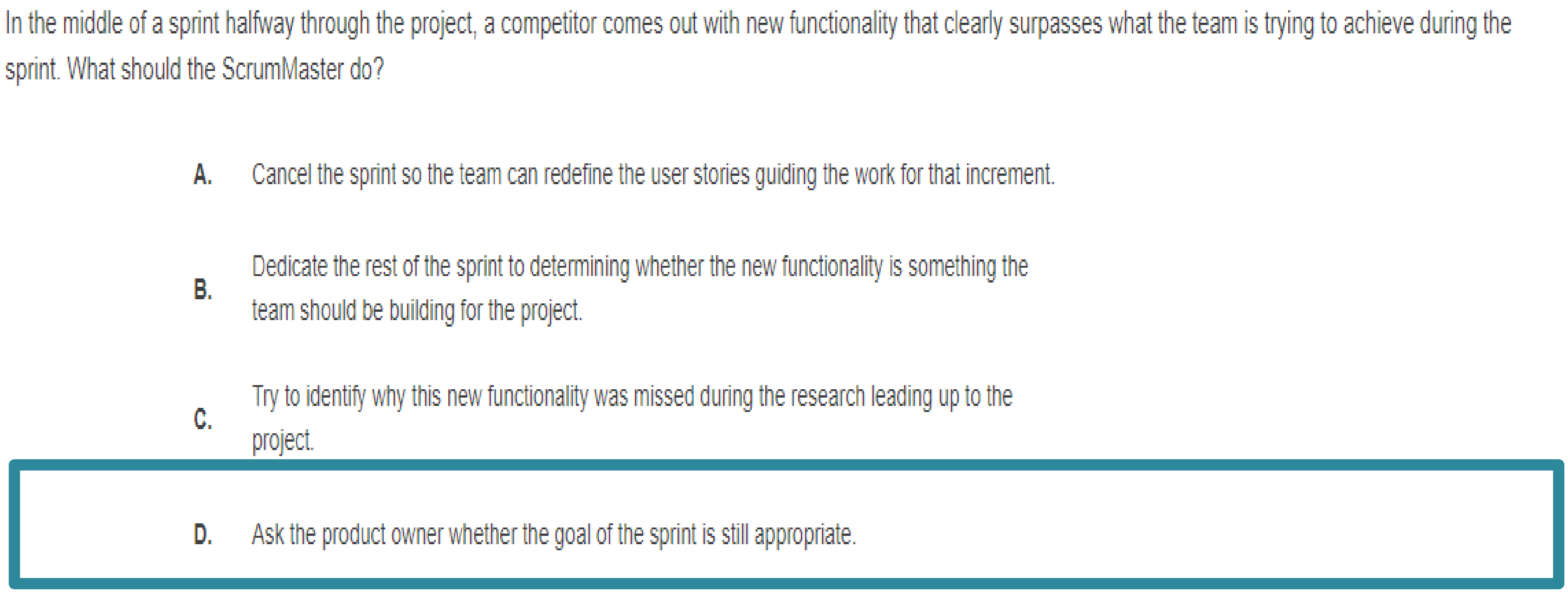


Product Backlog Refinement vs. Delivery

- A continuous activity effort than a formal sprint activity - NOT A TIME BOX
- **PO AND DEVELOPMENT TEAM** work together to prepare for the upcoming Sprints
- Typical goals of Product Backlog Refinement activity:
  - Everyone is clear about the requirement - backlog meets DoR
  - Product Backlog Items targeted to the next sprint are SMALL ENOUGH


Sprint Duration
Story Points to be Delivered
Stretch Objectives
Capacity Planning for Sprint
Availability of each Member
Time-offs of each Member
Team Productivity

Capacity Planning



In the middle of a sprint halfway through the project, a competitor comes out with new functionality that clearly surpasses what the team is trying to achieve during the sprint. What should the ScrumMaster do?

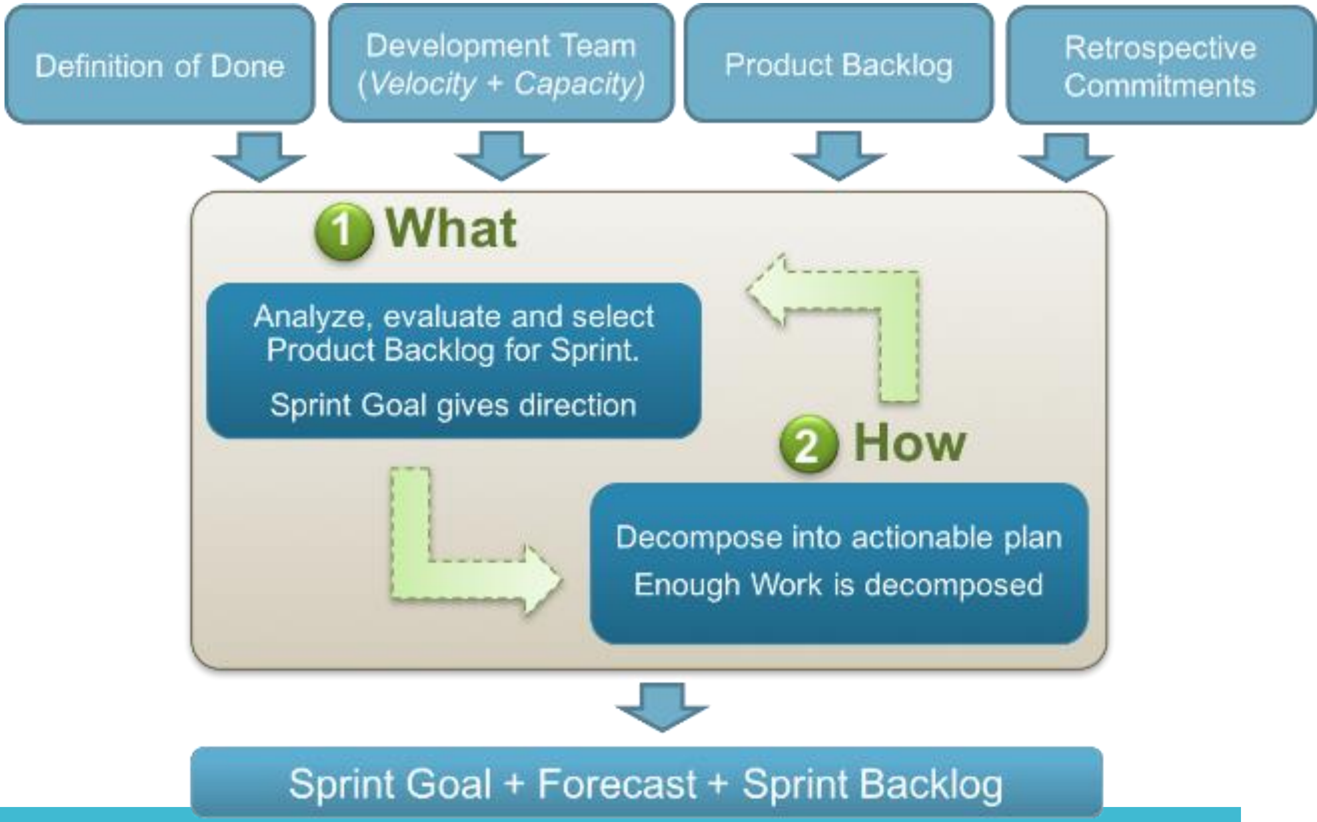
A. Cancel the sprint so the team can redefine the user stories guiding the work for that increment.

B. Dedicate the rest of the sprint to determining whether the new functionality is something the team should be building for the project.

C. Try to identify why this new functionality was missed during the research leading up to the project.

[✓ **D.**] Ask the product owner whether the goal of the sprint is still appropriate.

Sprint Planning



Definition of Done

Development Team (*Velocity + Capacity)*

Product Backlog

Retrospective Commitments

1 What

Analyze, evaluate and select Product Backlog for Sprint.

Sprint Goal gives direction

2 How

Decompose into actionable plan

Enough Work is decomposed

Sprint Goal + Forecast + Sprint Backlog

Sprint Planning

Sprint Planning

- Team selects items from the product backlog they can commit to completing
- Sprint backlog is created
  - Task are identified and each are estimated (1 – 16 hours)
  - Collaboratively, not done ***alone*** by the Scrum Master
- High-level design is considered

Daily Scrum Meeting

1) Do NOT direct

2) Huddle/Collaborate

3) Scrum Lead starts

4) Always have it

5) Encourage break-outs

6) Stand-up

IMPORTANT
IMPORTANT
IMPORTANT
MUST DO

Daily Scrum Meeting

Why Daily Scrum?

Access progress towards sprint goal

Fine-grain coordination

Daily commitment

Raising impediments

Peer pressure

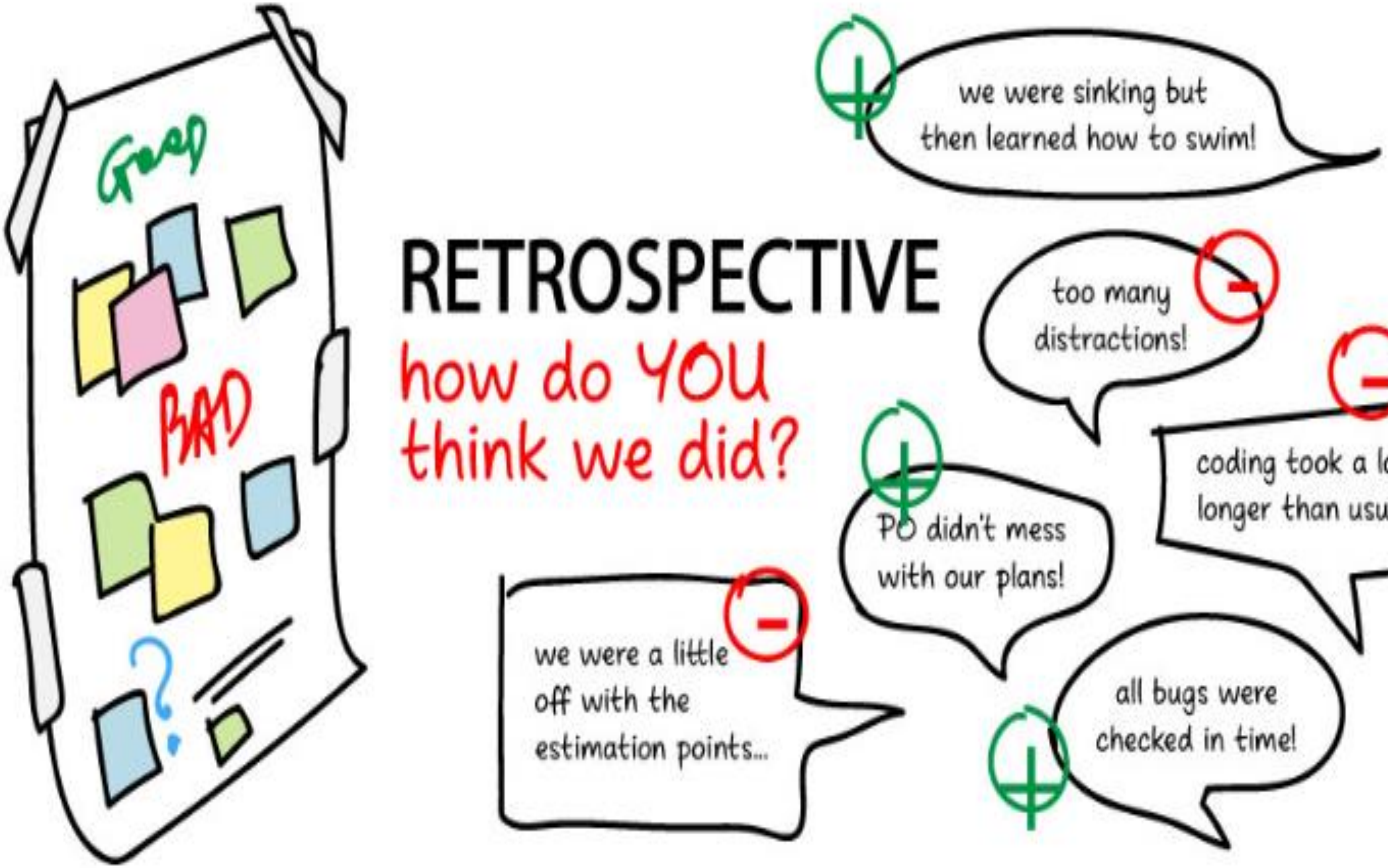
Dzmitry Liaudanski
Senior .Net Black Ops Developer :) at Zettabox

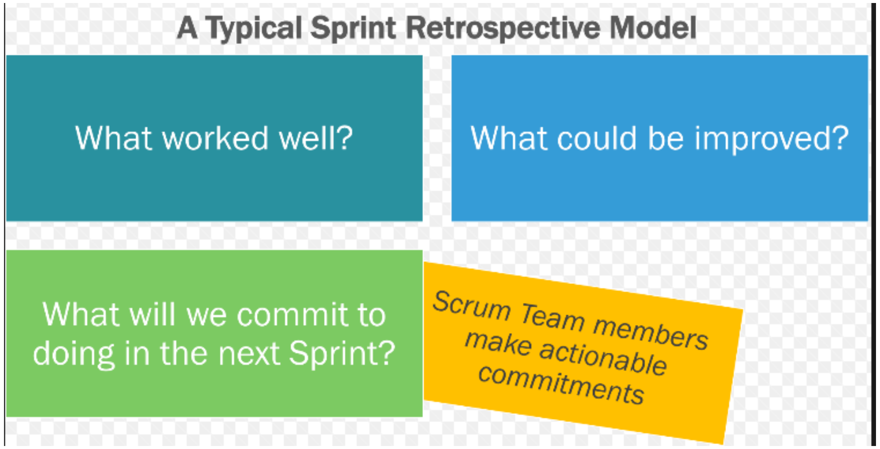
Make your daily stand up meeting shorter :)

Sprint Review

SPRINT REVIEW MEETING
SCRUM MASTER
PRODUCT OWNER
SCRUM TEAM
OR
...????
STAKEHOLDER AND SPONSORS
CUSTOMER

Sprint Retrospective


Good
BAD
?
RETROSPECTIVE
how do YOU think we did?
+ we were sinking but then learned how to swim!
− too many distractions!
− coding took a lot longer than usual
+ PO didn't mess with our plans!
− we were a little off with the estimation points...
+ all bugs were checked in time!



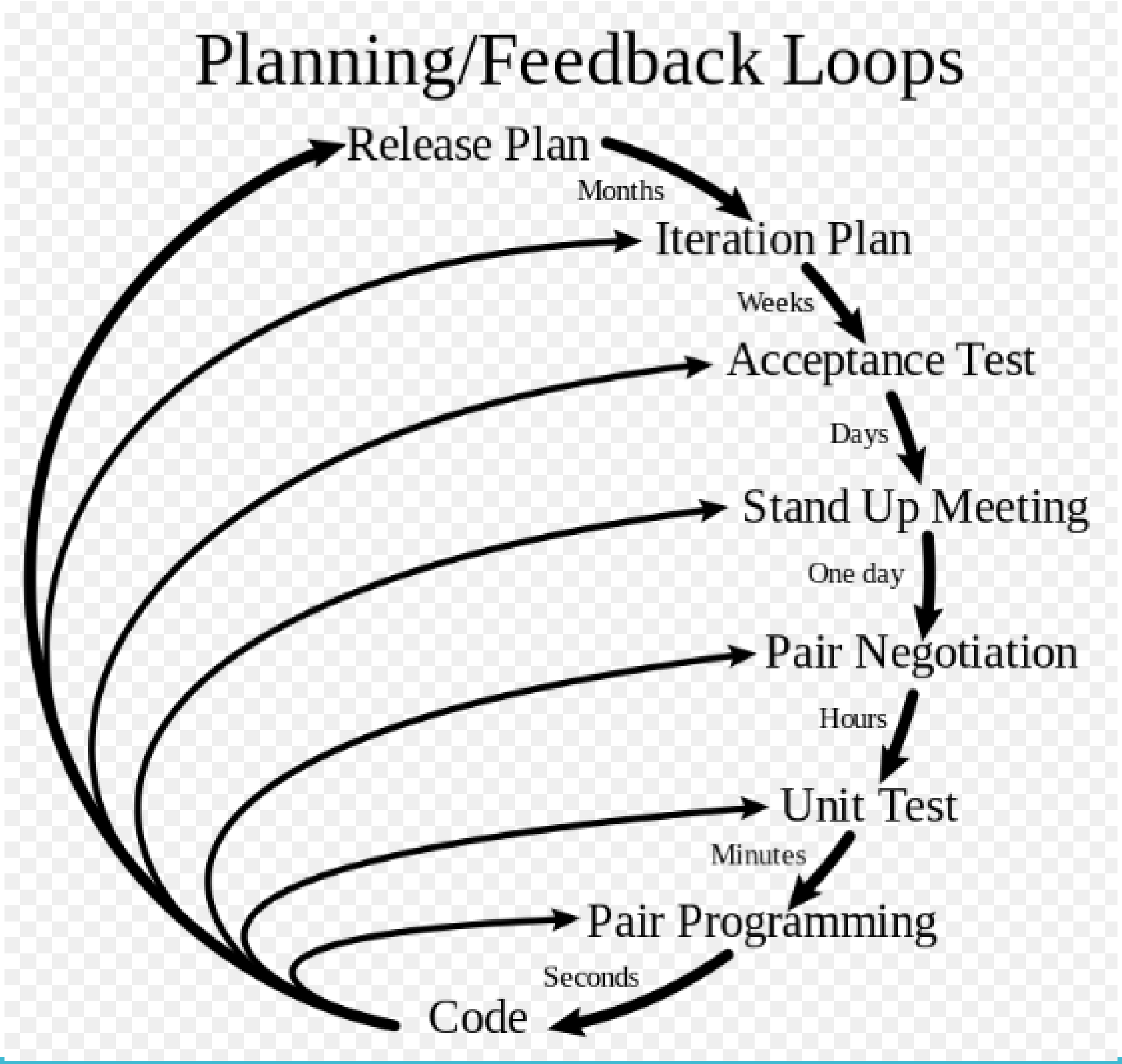
A Typical Sprint Retrospective Model

What worked well?

What could be improved?

What will we commit to doing in the next Sprint?

Scrum Team members make actionable commitments


Planning/Feedback Loops
Release Plan
Months
Iteration Plan
Weeks
Acceptance Test
Days
Stand Up Meeting
One day
Pair Negotiation
Hours
Unit Test
Minutes
Pair Programming
Seconds
Code

3 Roles

- Development Team
- Product Owner
- ScrumMaster

3 Artifacts

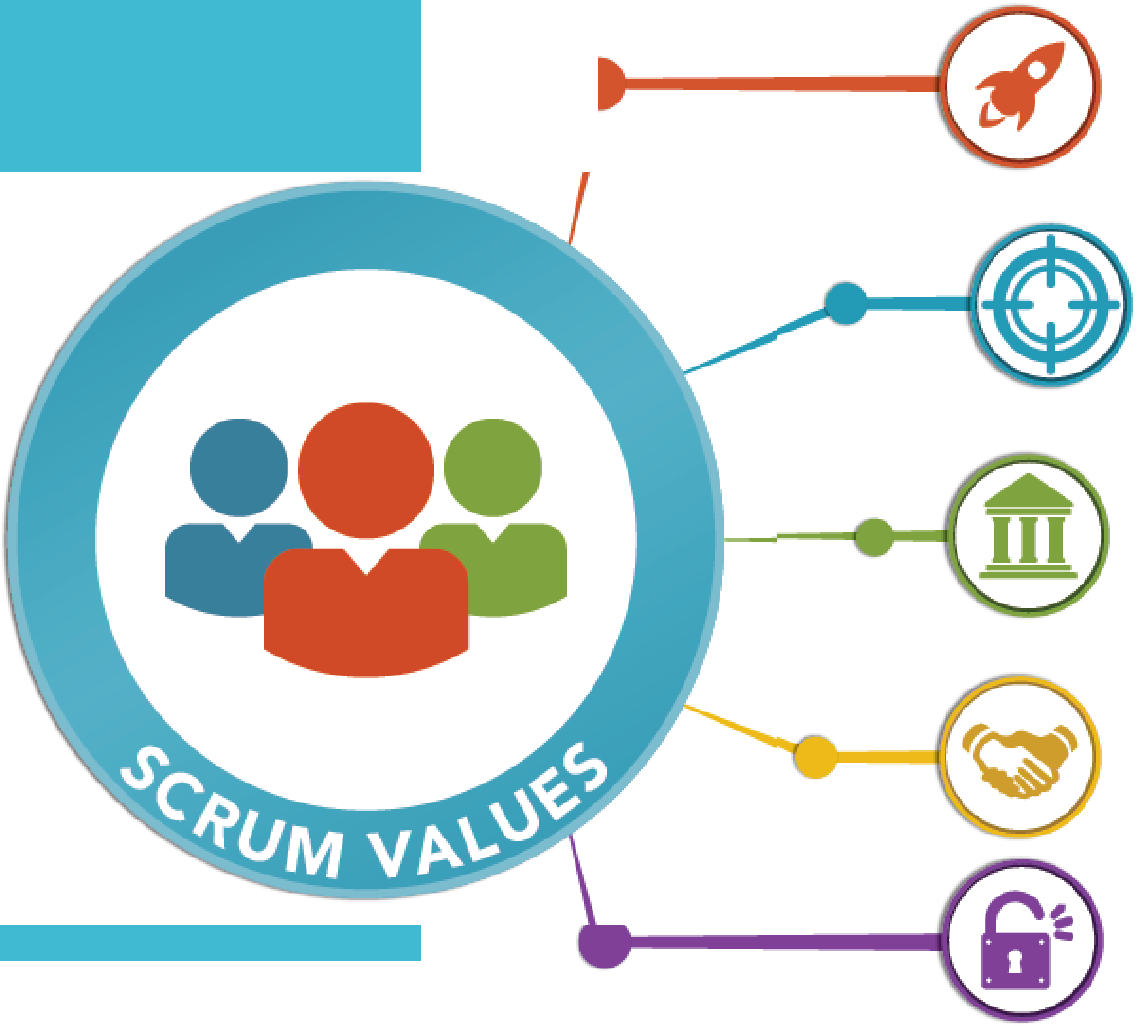
- Product Backlog
- Sprint Backlog
- Increment

5 Activities

- Product Backlog Refinement
- Sprint Planning
- Daily Scrum
- Sprint Review
- Sprint Retrospective

5 Values

- Courage
- Openness
- Focus
- Commitment
- Respect



COURAGE
Scrum Team members have courage to do the right thing and work on tough problems

FOCUS
Everyone focuses on the work of the Sprint and the goals of the Scrum Team

COMMITMENT
People personally commit to achieving the goals of the Scrum Team

RESPECT
Scrum Team members respect each other to be capable, independent people

OPENNESS
The Scrum Team and its stakeholders agree to be open about all the work and the challenges with performing the work

© Scrum.org

Rapid Fire

Sprint starts with which ceremony?

Sprint Planning Meeting

Sprints ends with which ceremony?

Sprint Retrospective

Which ceremony happens every day?

Daily Stand-up

Which scrum ceremony stakeholders allow to speak?

Sprint Demo

5. Who write the user stories?

A. The team
B. The customer
C. The users
D. The testers

QUIZ…

QUIZ…

6. Which of the following reports visually shows the remaining estimated workload over the course of the project?

A. Sprint Chart

B. Gantt Chart

C. Product Backlog Burn up report

D. Product Backlog Burndown Report

QUIZ…

7. Product Backlog is owned by?

A. Scrum Master

B. The Team

C. The Product Owner

D. None of the above

QUIZ…

9. The number of stories a team can deliver in an iteration is known as

   a. Velocity

   b. Cycle time

   c. Burn rate

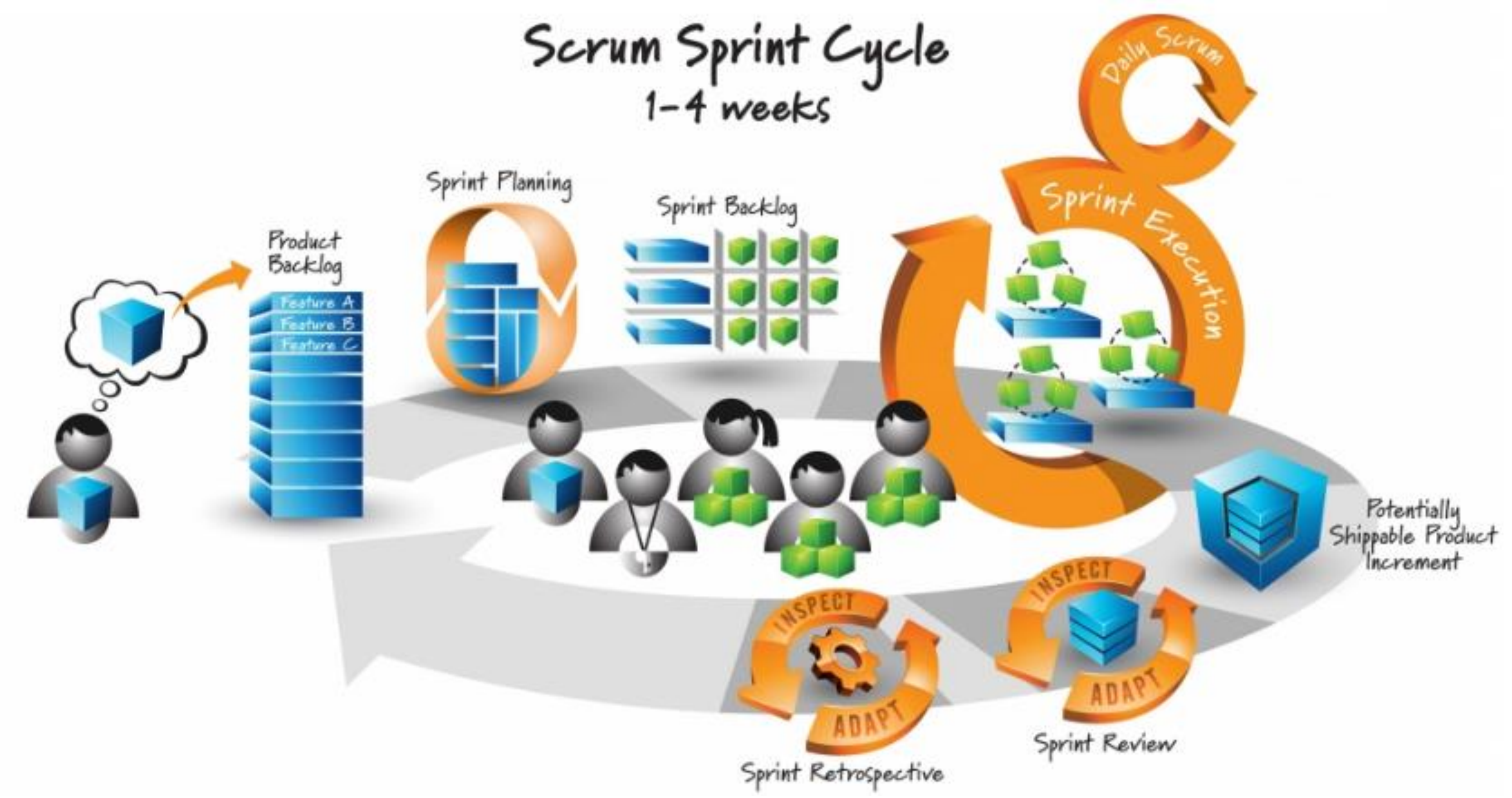
   d. Capacity

10. When a release burndown chart show a bump

A. Actual velocity is less than the planned velocity
B. Story got re-prioritized after first iteration
C. Work has been added to the release
D. Work has been removed from the release

QUIZ…

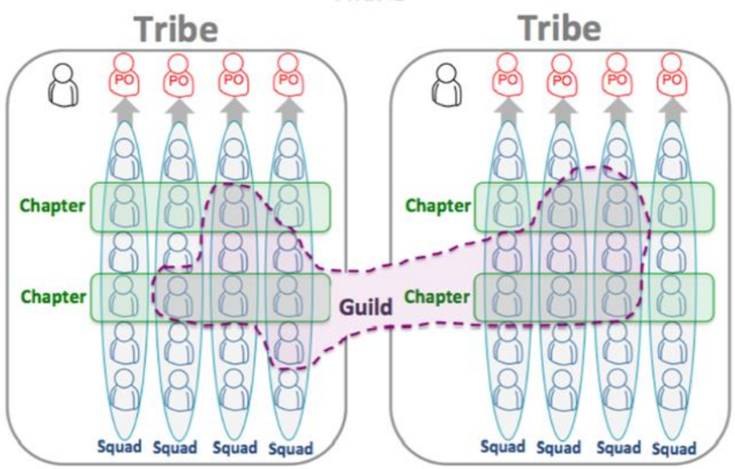
Artifacts


Scrum Sprint Cycle
1-4 weeks
Daily Scrum
Sprint Execution
Sprint Planning
Sprint Backlog
Product Backlog
Feature A
Feature B
Feature C
Potentially Shippable Product Increment
INSPECT
ADAPT
Sprint Review
INSPECT
ADAPT
Sprint Retrospective

A road Ahead

Waterfall to Agile to Squads…

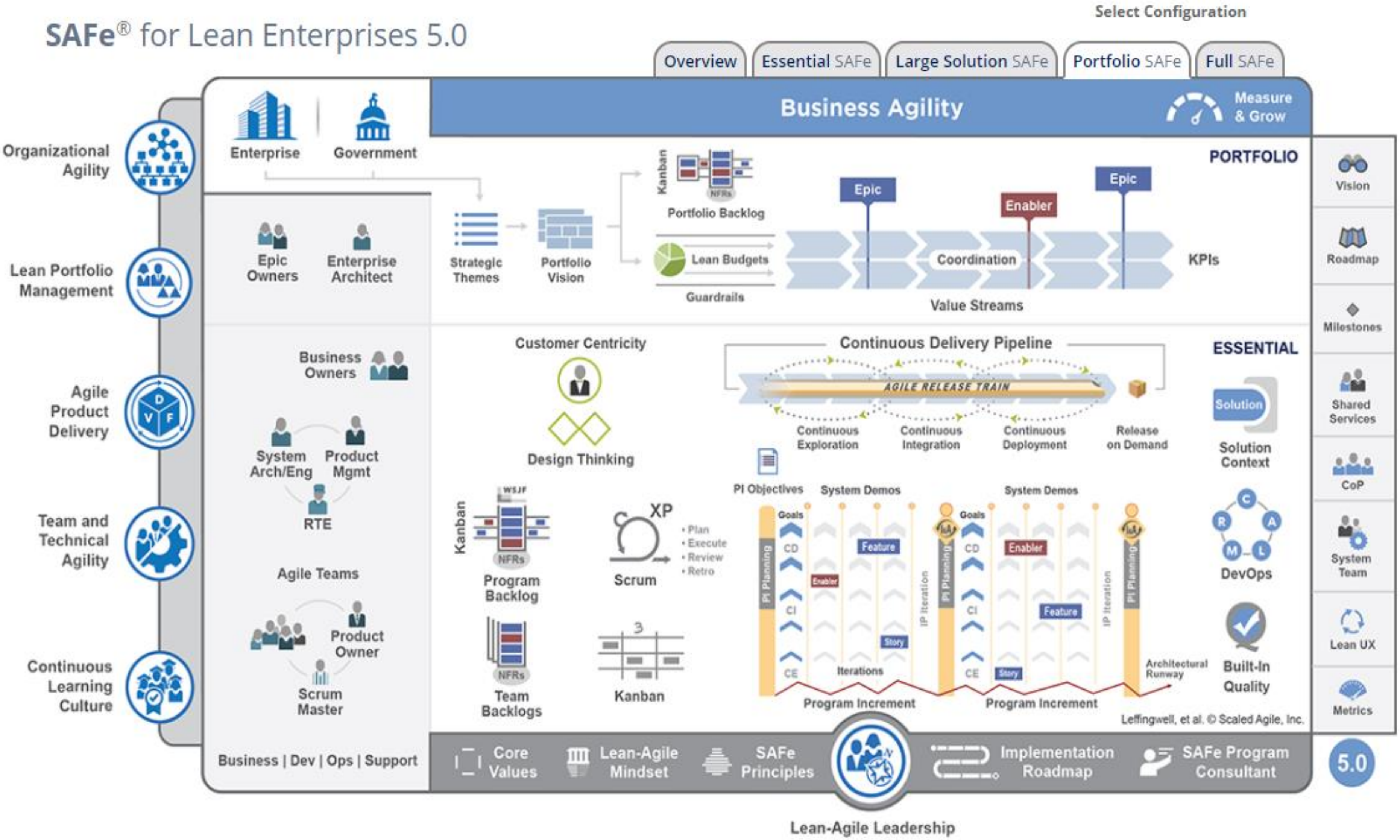
full-stack agile



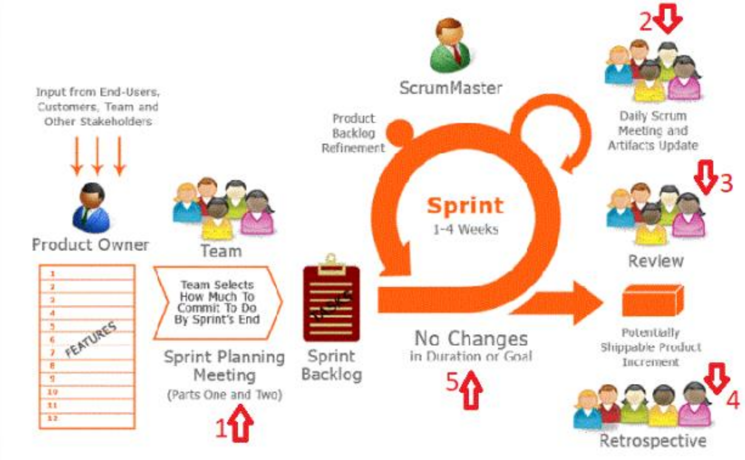
Squad

- ✓ "Feel like a mini-startup"
- ✓ Self-organizing
- ✓ Cross-functional
- ✓ 5-7 engineers, less than 10
- ✓ Stable

Mature Agile Organization Team Structure…



Enterprise Agile Implementation SAFe Framework


Input from End-Users, Customers, Team and Other Stakeholders
Product Owner
1
2
3
4
5
6
7
8
9
10
11
12
FEATURES
Team
Team Selects How Much To Commit To Do By Sprint's End
Sprint Planning Meeting
(Parts One and Two)
1
Sprint Backlog
ScrumMaster
Product Backlog Refinement
Sprint
1-4 Weeks
No Changes
in Duration or Goal
5
2
Daily Scrum Meeting and Artifacts Update
3
Review
Potentially Shippable Product Increment
4
Retrospective

Is it so easy?

Scrum is lightweight, simple to understand but extremely difficult to master and implement well

- Ken

“There is a perfect ant, a perfect bee, but man is perpetually unfinished…Moreover, the incurable unfinishedness keeps man perpetually immature, perpetually capable of learning and growing.”

– Eric Hoffer

CYBAGE

Delivering Value. Scientifically.

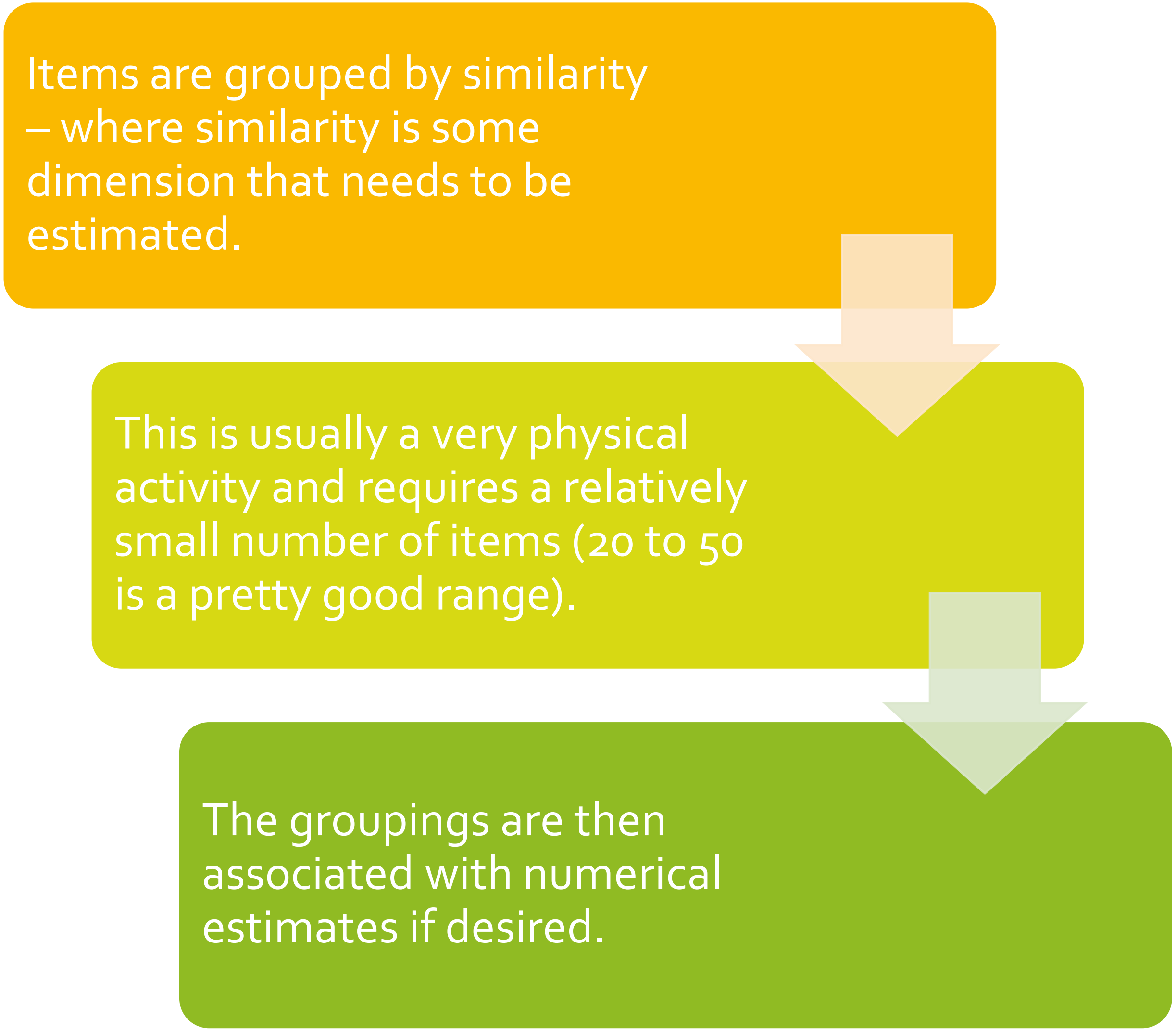
THANK YOU!

www.Cybage.com

How was your journey towards being agile, share your thoughts....

- Feel free to share....
  - Feedback
  - Comments
  - Suggestions...

Affinity Mapping



Items are grouped by similarity – where similarity is some dimension that needs to be estimated.

This is usually a very physical activity and requires a relatively small number of items (20 to 50 is a pretty good range).

The groupings are then associated with numerical estimates if desired.

Ordering Protocol

- Items are placed in a random order on a scale labeled simply “low” to “high”. Each person participating takes turns making a “move”.
- A move involves one of the following actions:
  - change the position of an item by one spot lower or one spot higher, talking about an item, or passing.
  - If everyone passes, the ordering is done.
- The Challenge, Estimate, Override and the Relative Mass Valuation methods are variations on the ordering protocol.

Divide until Maximum Size or Less

The group decides on a maximum size for items (e.g. 1 person-day of effort).

Each item is discussed to determine if it is already that size or less.

If the item is larger than the maximum size, then the group breaks the item into sub-items and repeats the process with the sub-items.

This continues until all items are in the allowed size range.