Delivering Value. Scientifically.

Agile _SCRUM Project Management

Presenter Name : **Dipti Trivedi**

Getting To Know You

Sr. No	Participant Details
1	Are you aware of the concepts of projects and Project Management?
2	How long have you been writing schedules?
3	What do you think is the most challenging aspect of writing an effective schedule?
4	What were/are the concern areas ….
5	What is your expectation from this workshop

Your Facilitators

Dipti Trivedi

SAFe Agilist
SCRUM Master Certified
PMP, PMI, USA
Microsoft Certified Professional - MCTS
[Project SERVER]

Overall **15+ yrs** Industry experience

Conducted **10,000+ hours** training on **Project Management** & **PM Tools**

Trained 5000+ professional across the globe.

Sr. Manager, Cybage-Pune

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

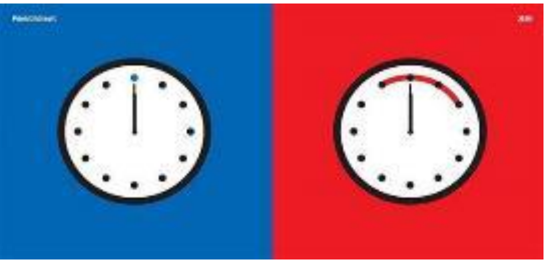
Lead-Technical Services at Softbridge Solutions Pvt. Ltd.

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

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

CYBAGE
Delivering Value. Scientifically.

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!

MY WaY!

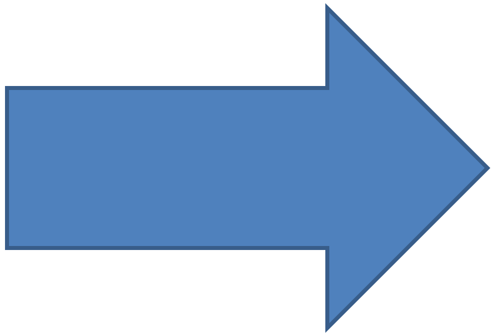
CONVINCE ME...

Have Question…

Anytime… ☺

Agenda Topics (Scope):

- Why Agile
- Agile Methodology
- Agile Manifesto
- 12 Principles of Agile
- Agile Frameworks: SCRUM and KANBAN
- What is SCRUM?
- Scrum in a nutshell
- SCRUM Vs Waterfall



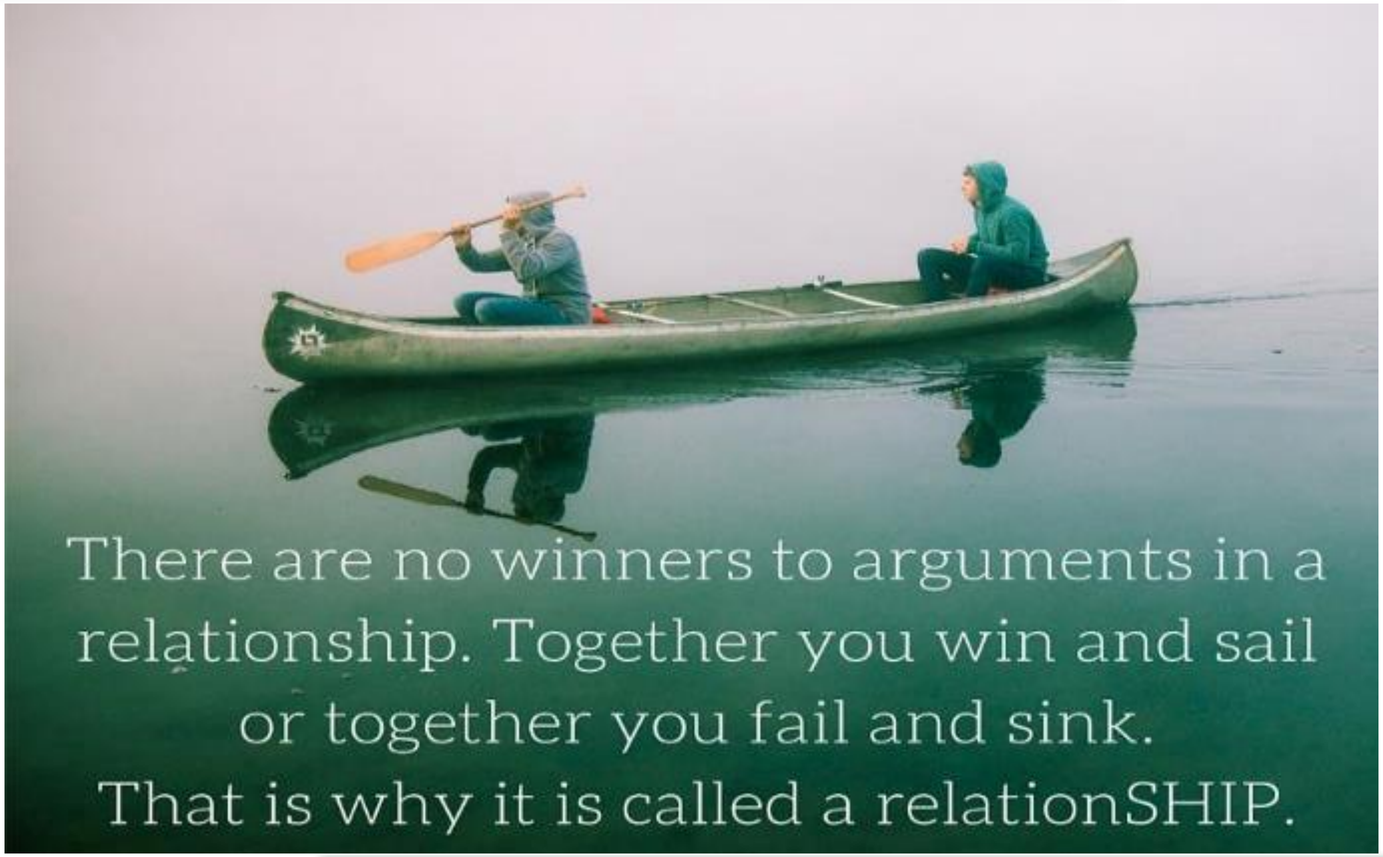
- Scrum Roles
  - PO –SM-Team
    - User Stories- Use case- Requirements
    - Velocity
- Scrum Meetings
- Scrum Artifacts
- Sprint Burndown Chart
- Scrum Estimation [optional]

- SCRUM & A road Ahead

ACTIVITY

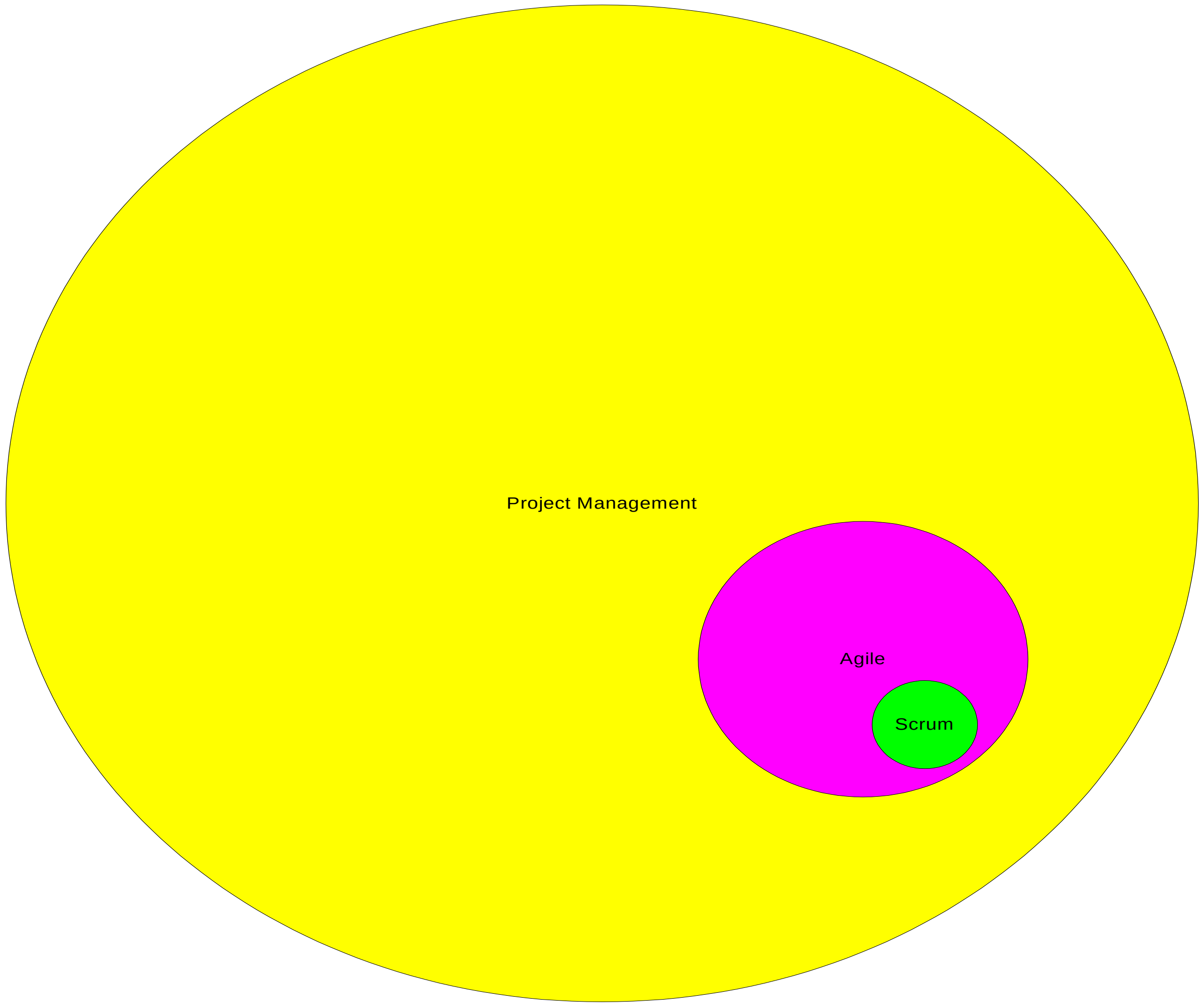
Being AGILE… Doing Agile: ☺

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.

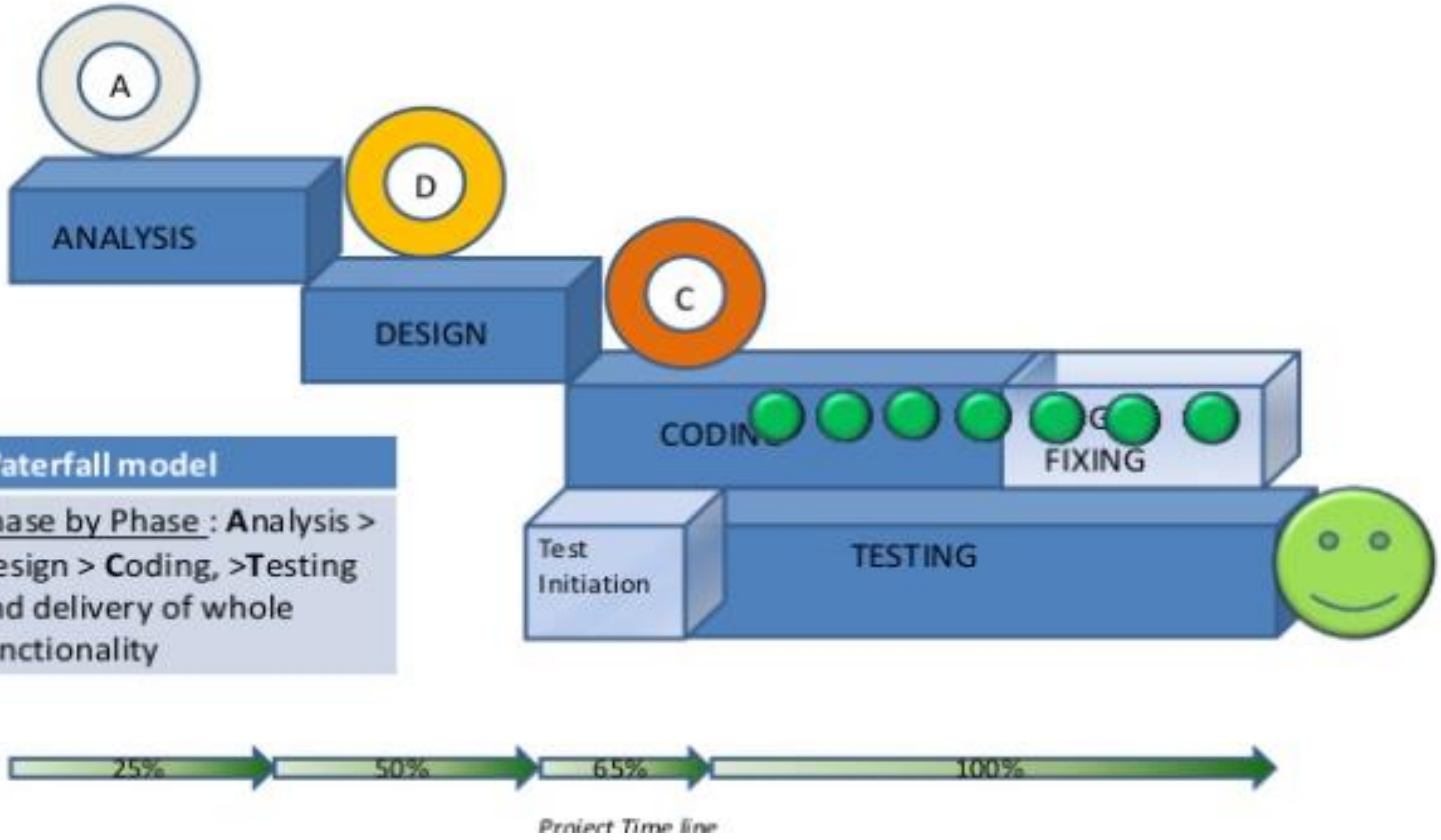
A joint accountability to make it SUCCESS

Management Sea



CYBAGE
Delivering Value. Scientifically.

WATERFALL Model



Waterfall model

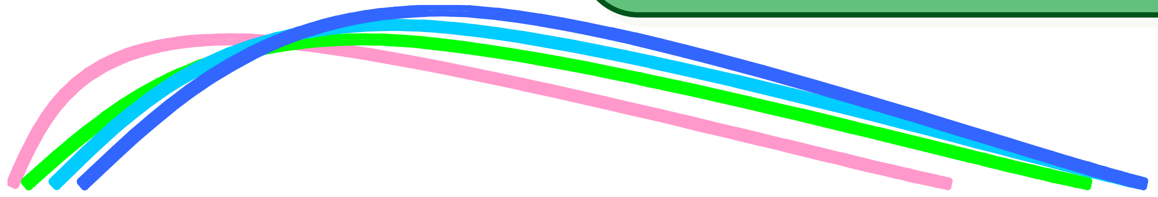
Phase by Phase : **A**nalysis > **D**esign > **C**oding, >**T**esting and delivery of whole functionality

Sequential vs. overlapping development

Requirements | Design | Code | Test

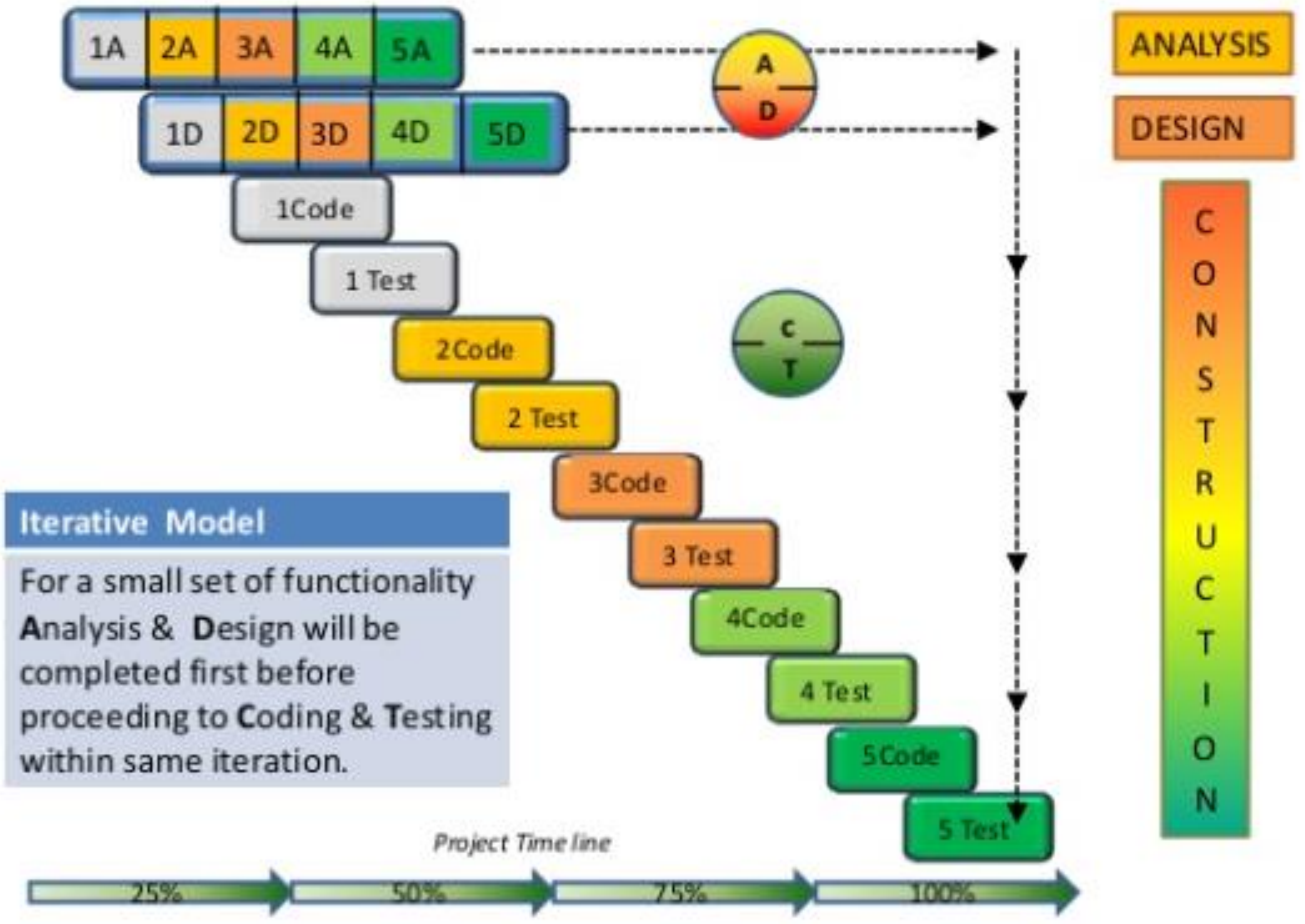
Rather than doing all of one thing at a time...

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



Source Courtesy: HBR

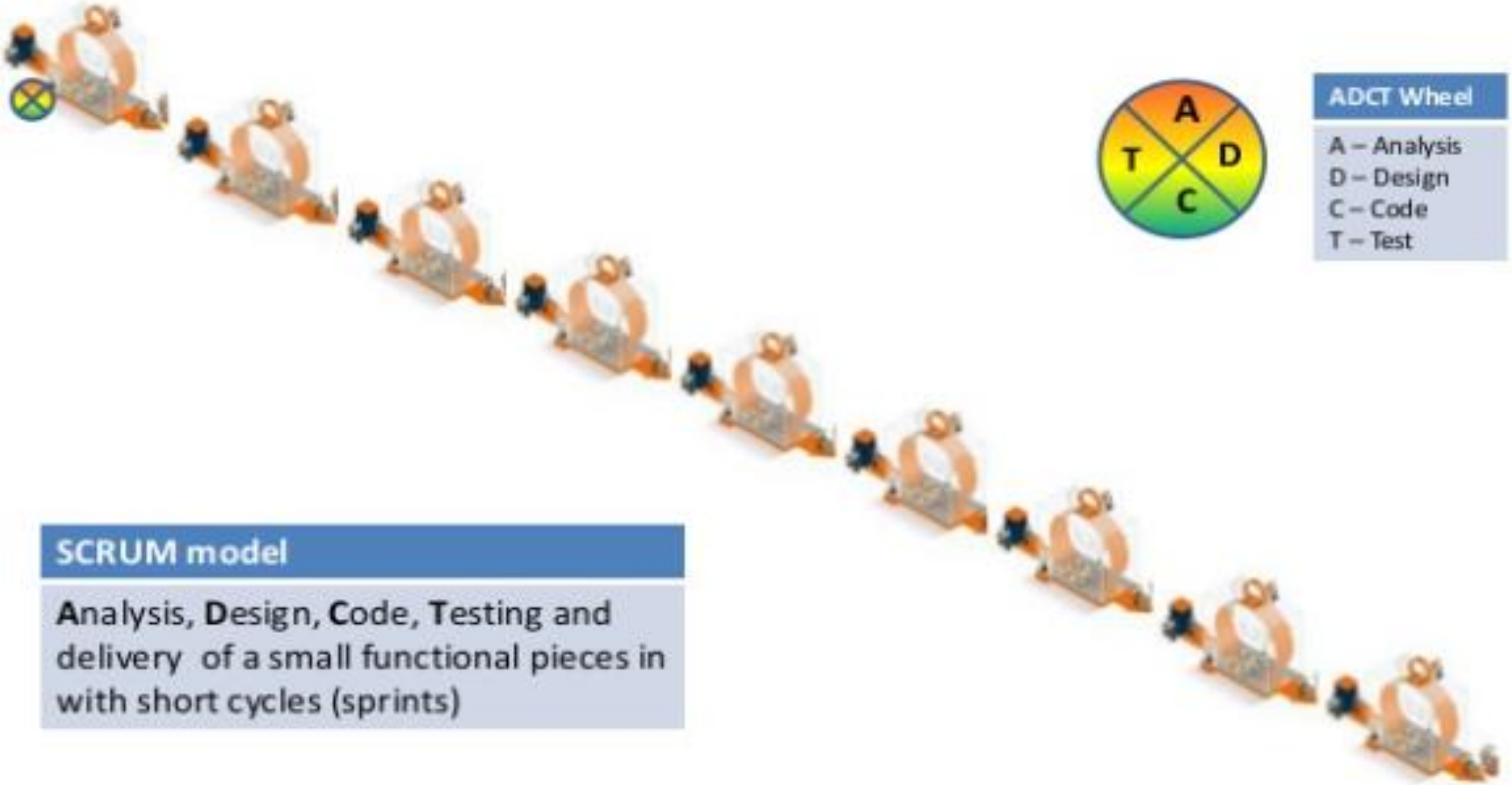
ITERATIVE Model



Iterative Model

For a small set of functionality Analysis & Design will be completed first before proceeding to Coding & Testing within same iteration.

Agile- SCRUM Model

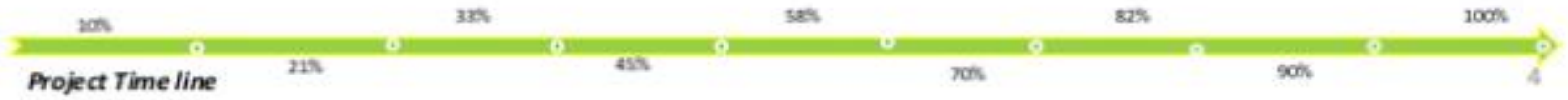


ADCT Wheel

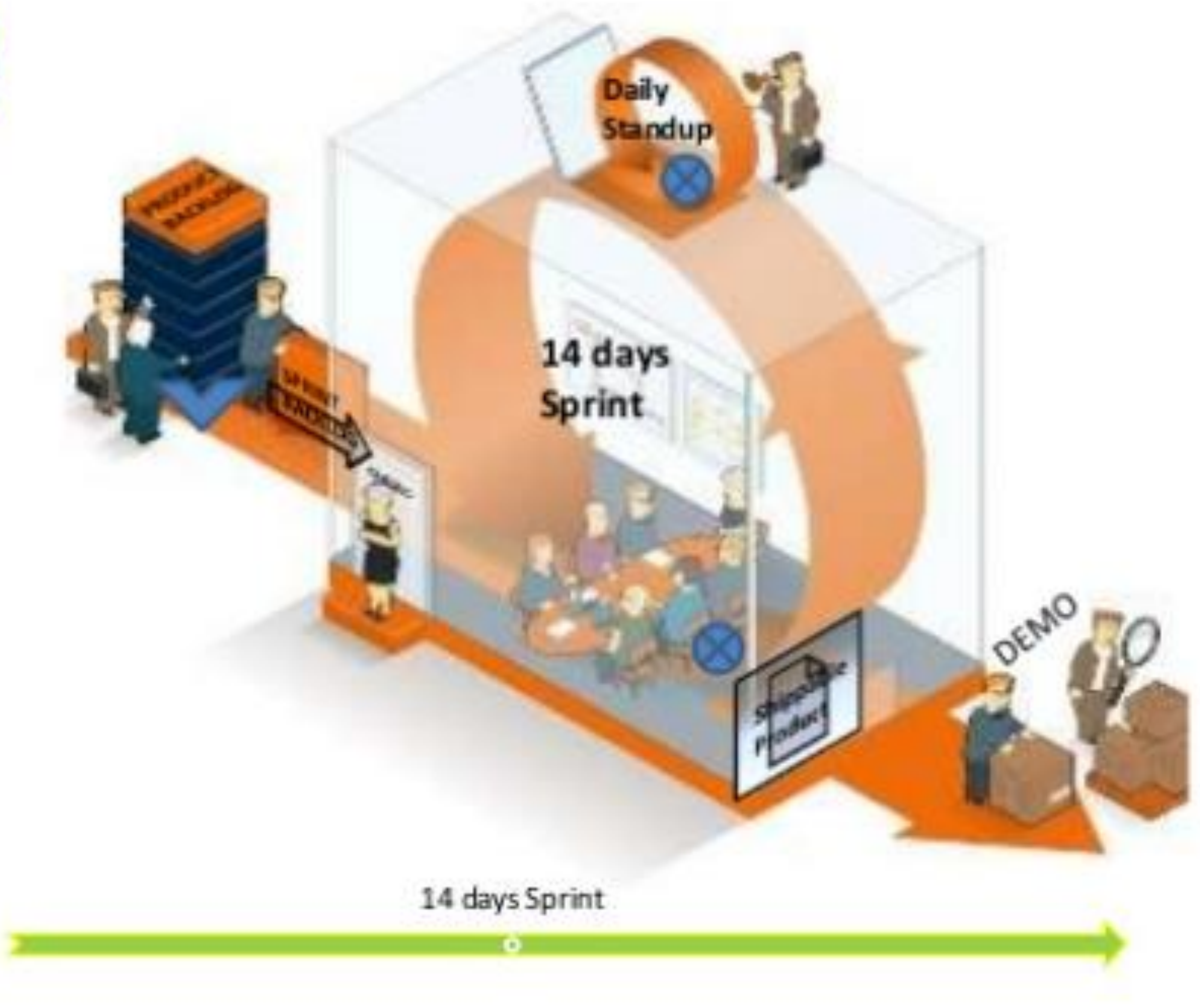
A – Analysis
D – Design
C – Code
T – Test

SCRUM model

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



SCRUM- SPRINT

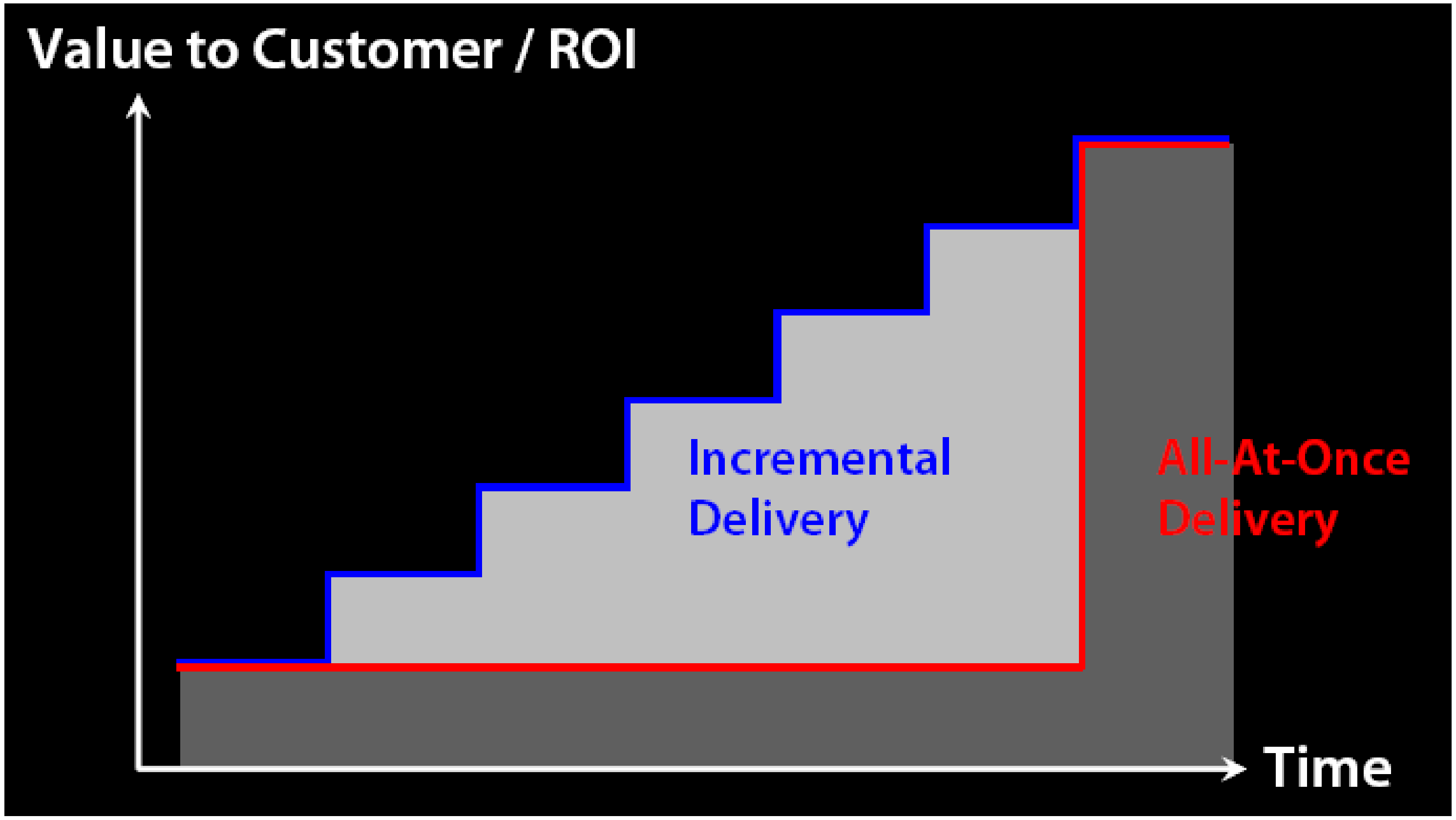


ADCT Wheel

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

CYBAGE
Delivering Value. Scientifically.

Value Realization- ROI: Waterfall Vs. Agile

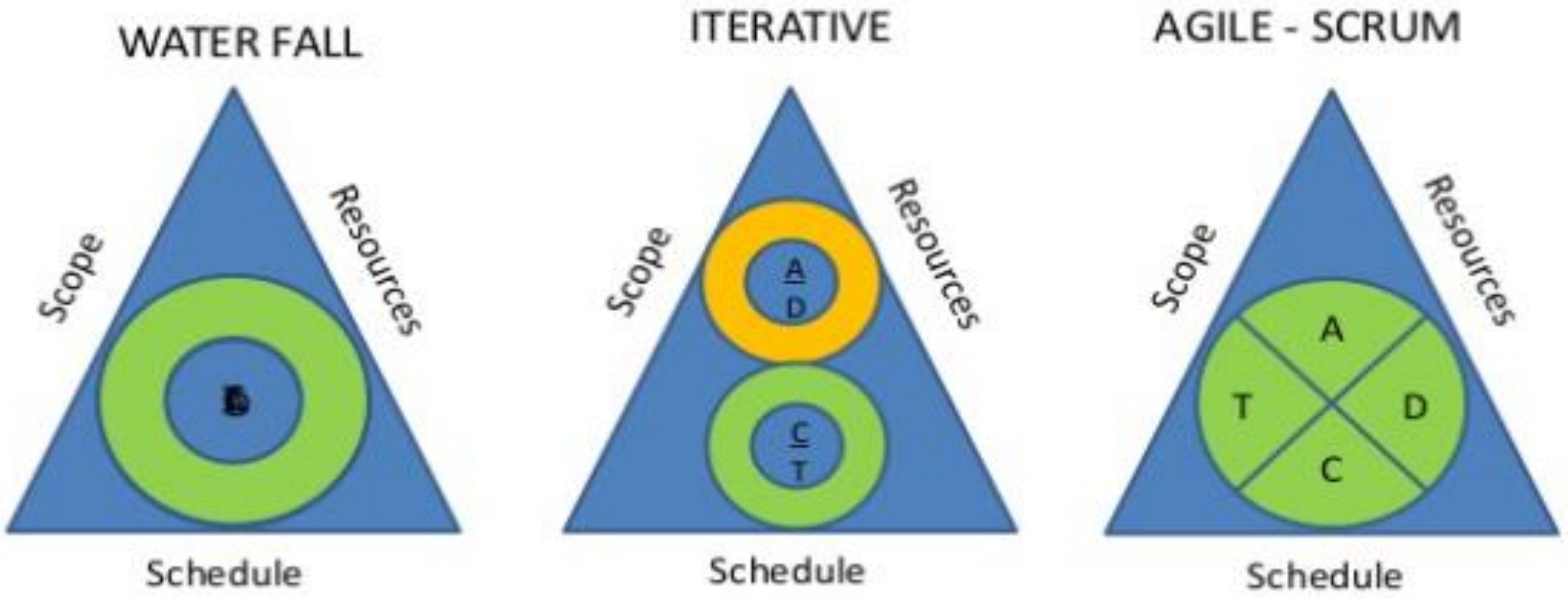


CYBAGE
Delivering Value. Scientifically.

Differences between Traditional Project Management Vs. Agile Project Management

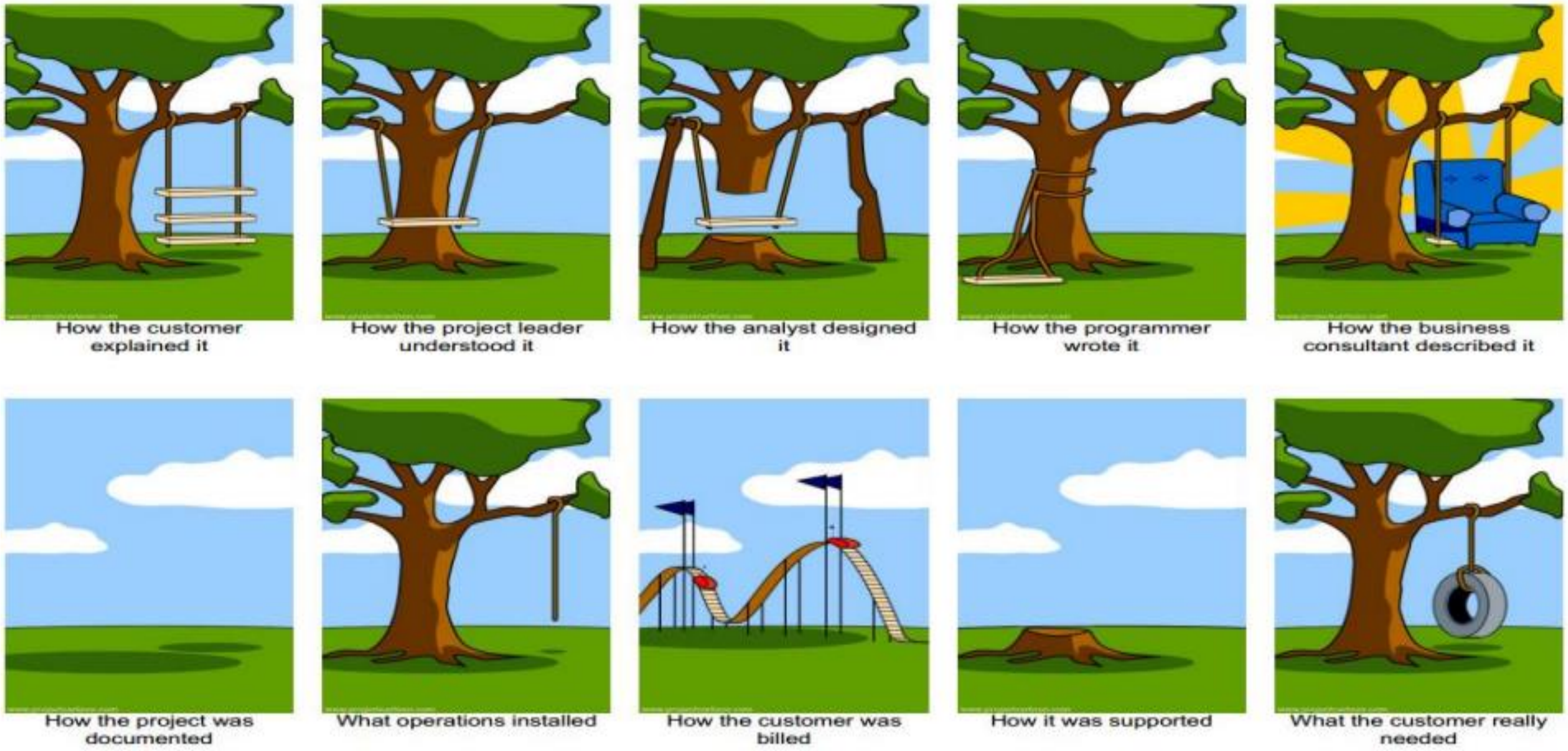
Traditional Project Management	Agile Project Management
Focus is more on 'Planning'	Focus is more on 'Executing'
Fixed scope; Time and Cost varies	Time are Cost are fixed; Scope varies
Project success implies meeting the triple constraints (Scope, Cost, Time)	Delivering business value is the criteria for project success
Scope inflexible once baseline defined; requires change request approval	Adaptable to change
Project Manager controls the project and directs the team members	Team is self-organizing and responsible for the project success
Interim deliverables created before the final product	Incremental and working product delivered frequently

Process vs Project Triangle

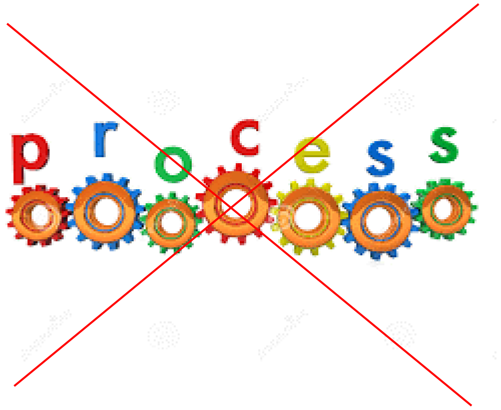


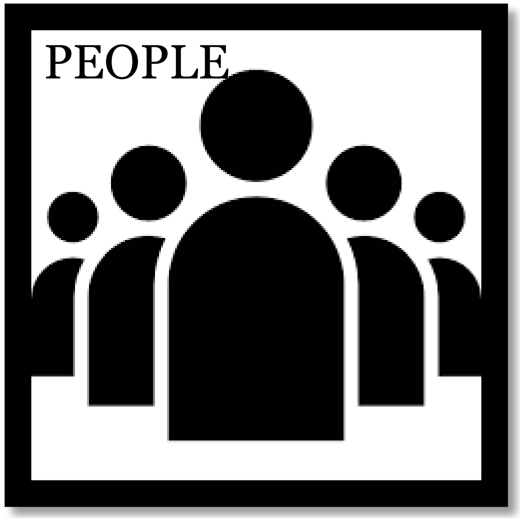
	Waterfall	Iterative	Agile
Format	Test Match: Strategic- Phase by Phase like Innings by Innings. Game for Specialists. Slow and Steady.	One Day: Strategic approach – First10/Middle/Slog overs. Mix of Specialists and All-Rounders. Result oriented.	T20: Lively , Dynamic, Full of Action. Game for All-Rounders.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

Real issue


How the customer explained it
How the project leader understood it
How the analyst designed it
How the programmer wrote it
How the business consultant described it
How the project was documented
What operations installed
How the customer was billed
How it was supported
What the customer really needed

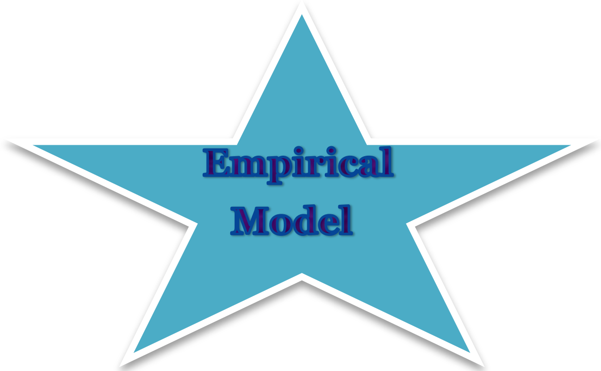
The problem



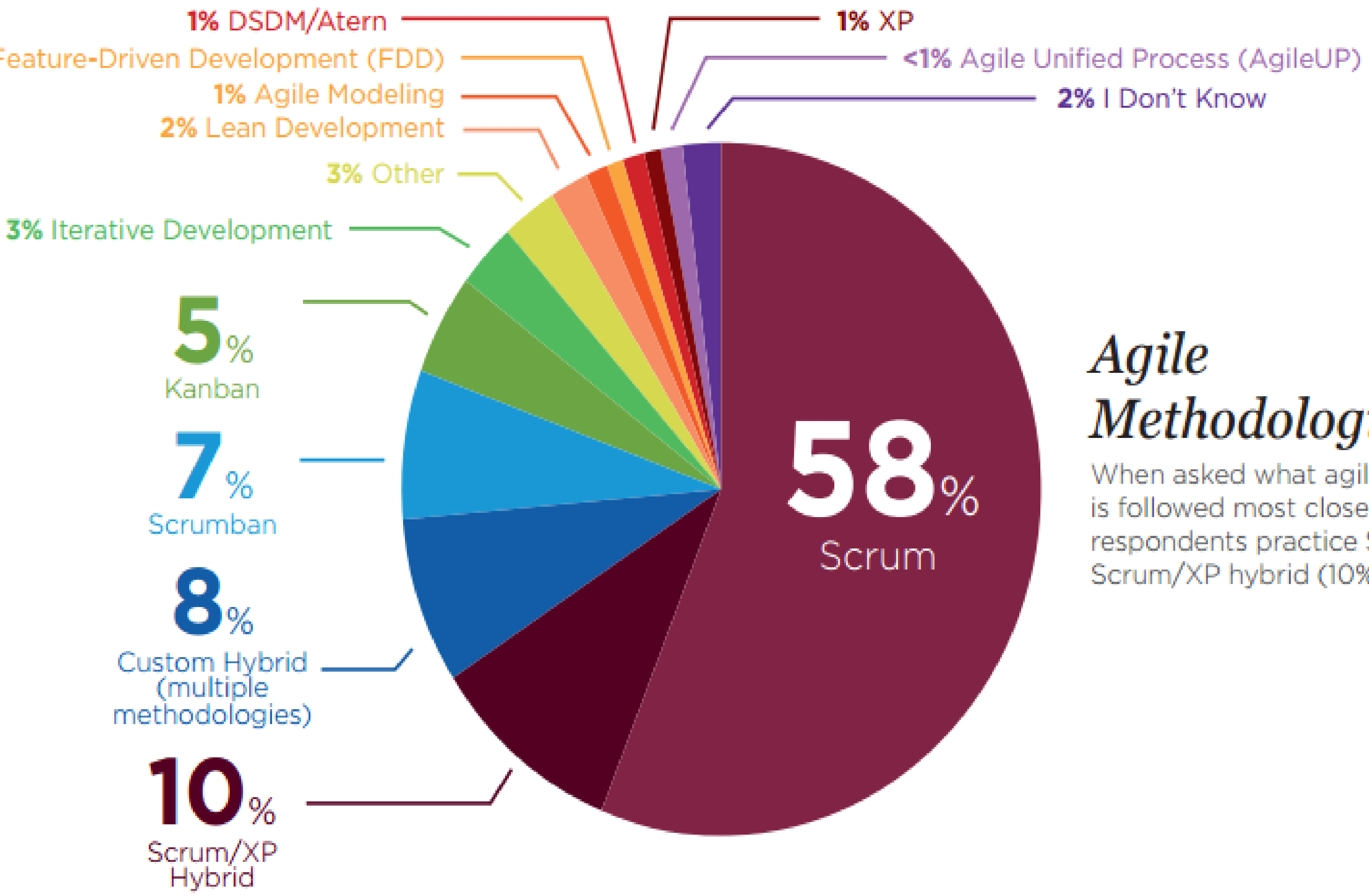


Scrum and Agile are based on the hypothesis that there is no meta-solution for software development. Just a framework within which we will be empirical – Inspect and Adapt

- Ken Schwaber



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%).

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.

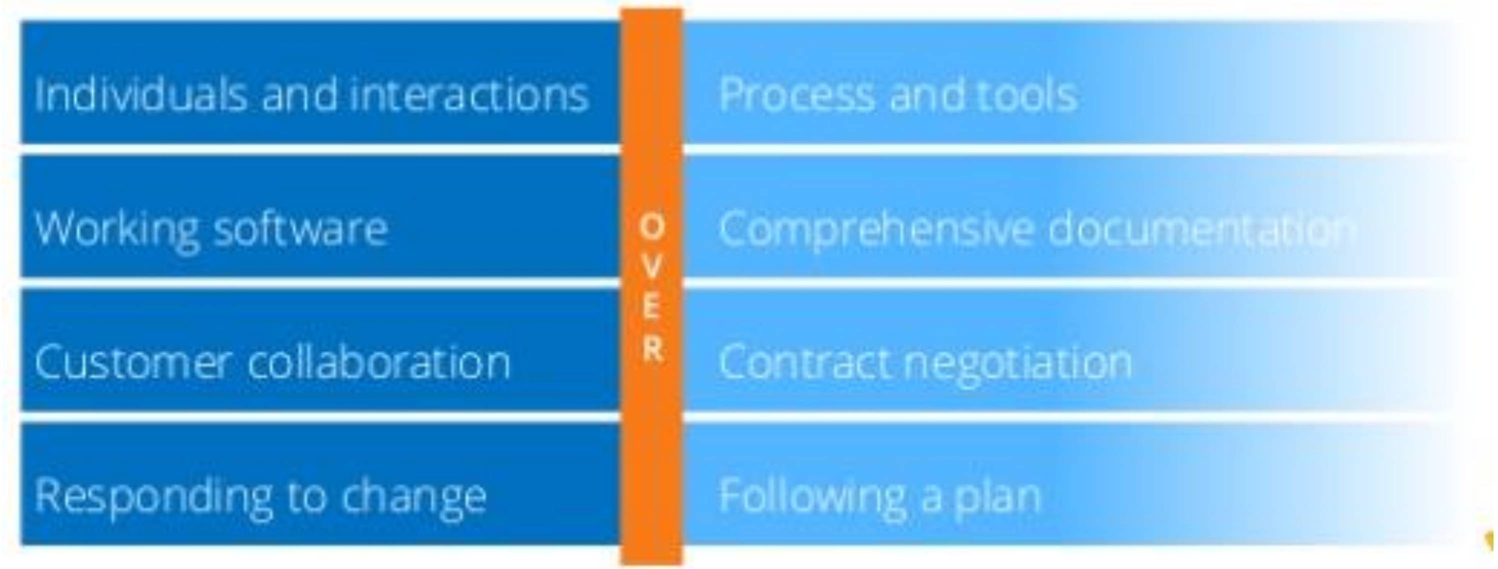
- Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockbur,n Ward Cunningham, Martin Fowler,, James Grenning Jim Highsmith, Andrew Hun,t Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwabe,r Jeff Sutherland, Dave Thomas

Reference: http://agilemanifesto.org/

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:



Individuals and interactions	OVER	Process and tools
Working software		Comprehensive documentation
Customer collaboration		Contract negotiation
Responding to change		Following a plan

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:

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Business people and developers must work together daily throughout the project.
- Build projects around self-motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- Working software is the primary measure of progress.
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- Continuous attention to technical excellence and good design enhances agility.
- Simplicity--the art of maximizing the amount of work not done--is essential.
- The best architectures, requirements, and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Reference: http://agilemanifesto.org/

Agile Benefits

Improved Customer Satisfaction

Early Surprises

Higher Quality

Team moral is high and enhanced work satisfaction

Productivity increase

Summary

We have discussed the following

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

Quiz

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. Project and operation is one and the same

A. True
B. False

Quiz

4. At what stage of the project life cycle the cost be lowest?

A. Concept
B. Development
C. Implementation
D. Closeout

Quiz

5. Who is assigned the most power in a projectized organization for waterfall methodology project….

A. The project team
B. Project Coordinator
C. The Project Manager
D. The functional manager

Quiz

6. All of the following are principles of agile except…

A. Incremental final product
B. Fixed Scope
C. Fixed Time
D. Fixed Cost

Quiz

8. The manifesto for agile software development authored by the founding members of

A. Agile alliance
B. Scrum alliance
C. Agile program leadership network
D. Agile project leadership network

Quiz

9. Which of the role is not found in agile projects

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

Quiz

10. Agile project management is both reliable and predictable

A. True
B. False

CYBAGE
Delivering Value. Scientifically.

SCRUM Project Management

Agile Methodologies

MODULE - 2

CYBAGE
Delivering Value. Scientifically.

Various Agile Models

Applying New Agile Practices
Variance in agile methods and approaches

Scrum	XP	Kanban
DSDM	LEAN	FDD
Crystal	RUP	TDD/ ATDD

Crystal

- The Crystal family of methodologies focuses on efficiency, osmotic communication between team members, and feedback-based learning for future operations.

Scrum

- Scrum is the most popularly adopted Agile methodology. It focuses on getting work done in time-boxed intervals called Sprints. Besides the Sprint, Scrum has a series of meetings on regular intervals (in addition to a daily meeting) to help stakeholders track project progress.

Dynamic Systems Development Method

- DSDM is an Agile framework that embraces dynamic customer involvement in project delivery. DSDM focuses on projects with tight schedules and budgets. DSDM subscribes to the Atern philosophy and uses these principles to direct the team in the delivery process.

Extreme Programming

- Extreme Programming is a methodology designed to improve software quality and responsiveness to changing customer requirements. Extreme Programming promotes a flat management structure. It has a few definitive features such as pair programming, unit testing of all code, and frequent communication with customers and programmers.

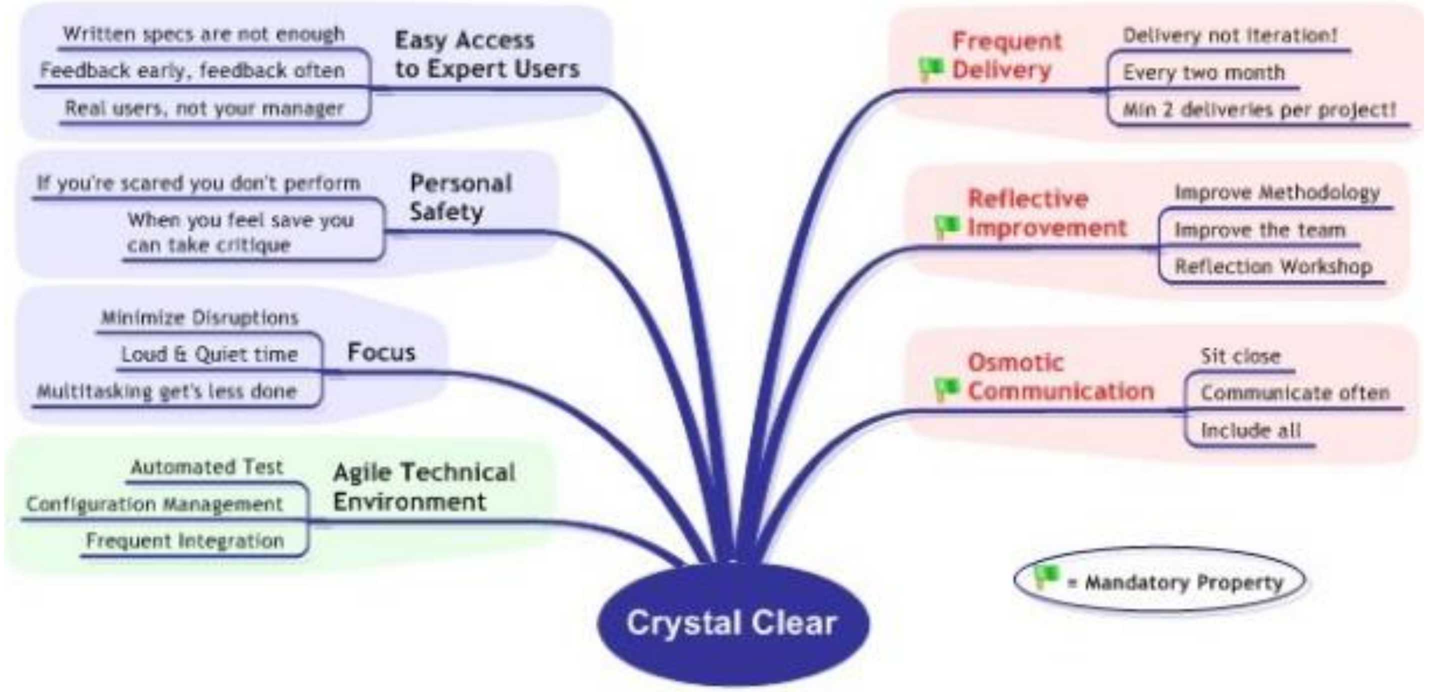
Kanban

- Kanban emphasizes just-in-time delivery for developing products and processes. Work is queued according to the desired priority to ensure relevant incremental delivery. The entire process of work completion is made transparent to help stakeholders track project progress.

Lean Product Development

- Lean Product Development uses the lean manufacturing and lean IT principles deployed in the Toyota Production System in the software development domain. Lean Product Development practices are slightly different from the other Agile methods, making use of value stream mapping, queuing theory, and other techniques for product development.

The 7 Properties of Crystal Clear



Amazon Deploys to Production Every 11.6 Seconds

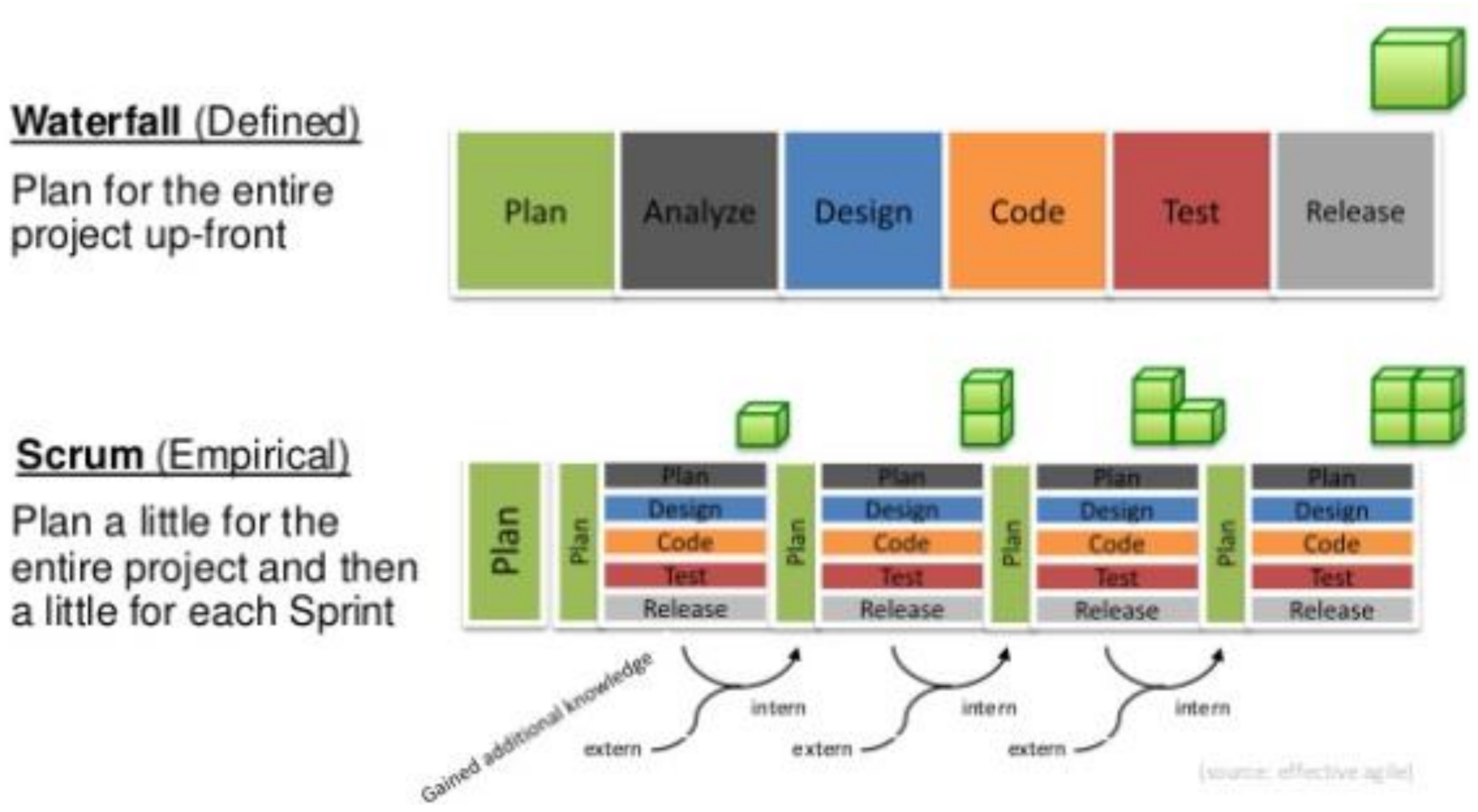
Source: http://joshuaseiden.com/blog/2013/12/amazon-deploys-to-production-every-11-6-seconds/

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
  - Impediment
- Sprint reports
  - Burndown Chart
- Scrum Estimation [optional]

CYBAGE
Delivering Value. Scientifically.

Defined Vs. Empirical


Waterfall (Defined)

Plan for the entire project up-front

Scrum (Empirical)

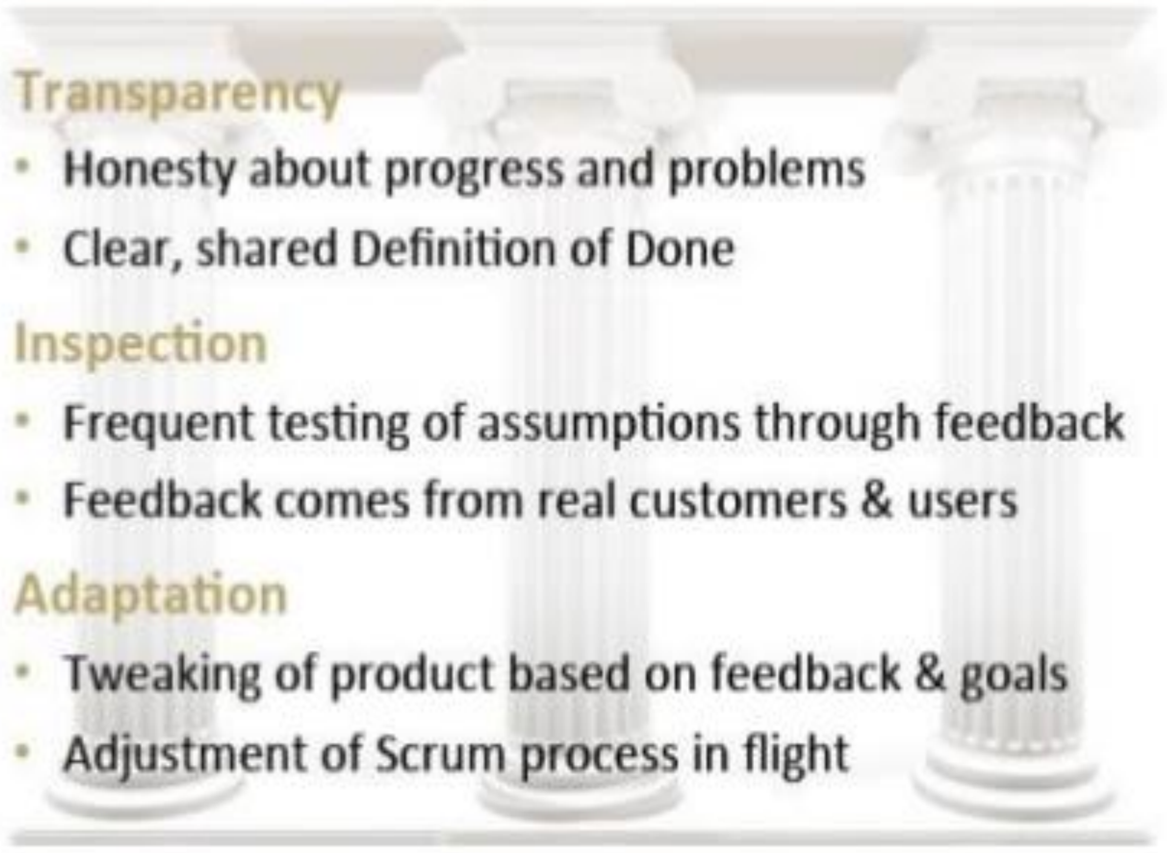
Plan a little for the entire project and then a little for each Sprint

CYBAGE
Delivering Value. Scientifically.

Empirical Process Control

Three pillars uphold every implementation of empirical process control:

The Scrum Framework
Transparency
Inspection
Adaptation
Three Pillars Of Scrum



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

CYBAGE
Delivering Value. Scientifically.

Scrum in 100 words

- Scrum is an agile process that allows us to focus on delivering the highest business value in the shortest time.
- It allows us to rapidly and repeatedly inspect actual working software (every two weeks to one month).
- The business sets the priorities. Teams self-organize to determine the best way to deliver the highest priority features.
- Every two weeks to a month anyone can see real working software and decide to release it as is or continue to enhance it for another sprint.

Characteristics

- **Self-organizing** teams
- Product progresses in a series of month-long “**sprints**”
- Requirements are captured as items in a list of “**product backlog**”
- No specific engineering practices prescribed
- Uses generative rules to create an agile environment for delivering projects
- One of the “agile processes”

Sprints

Sprint
30 Days

- Scrum projects make progress in series of “sprints”
- Typical duration is a calendar month **at most**
- A constant duration leads to better rhythm
- Product is designed, coded and tested during the sprint

Rapid Fire

Process of making Burger at fast food center is?

Defined process as it produces burger of acceptable quality repeatedly

Does scrum recommends specific engineering practices?

No, but scrum team uses some XP practices

Is Scrum Master is the same like Project Manager

No

Is the sprint and iteration are the same

Yes, in a certain way, both are a time boxed event of one month or less

Agile- A Mindset…

Agile Mindset

is the differentiator between

‘Doing’ Agile

&

‘Being’ Agile

Being AGILE… Doing Agile: ☺

- **Doing agile** is like using a compass and paper maps, you may get to the destination without getting lost, but not sure thats the destination you are hoping for.
- **Being agile** is like using the *GPS turn by turn navigation*, it alarm you about next sharp turn ahead, it inform you to change the path of your journey in case of traffic congestion ahead.

And even if you get lost its just re-calculate the route for you….

It keeps a track of your expected arrival time to the destination, keep on monitoring your speed to which you are travelling.

CYBAGE
Delivering Value. Scientifically.

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

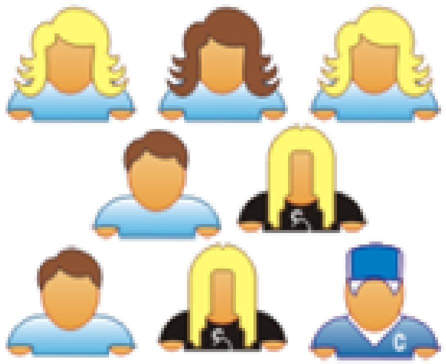
- Product Backlog Refinement
- Sprint Planning
- Daily Scrum
- Sprint Review
- Sprint Retrospective

5 Values

- Courage
- Openness
- Focus
- Commitment
- Respect

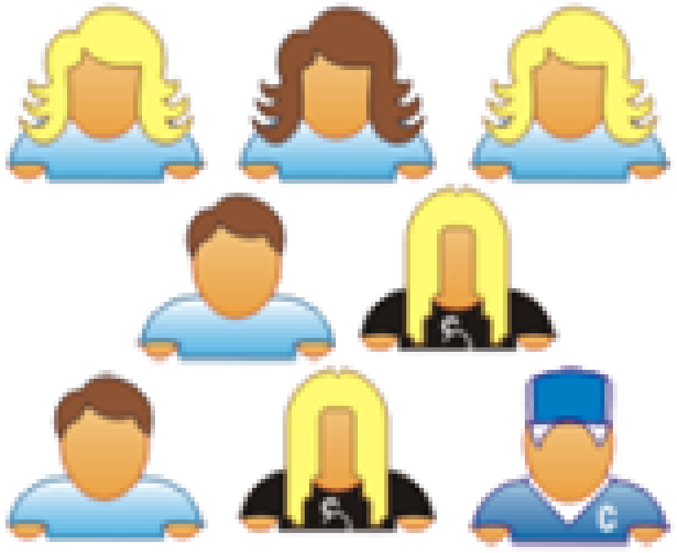
CYBAGE
Delivering Value. Scientifically.

The Team - Characteristics



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

The Team - 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

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

Product Owner - 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

CRACK

"Sh*t Bad Scrum Masters Say"

CYBAGE
Delivering Value. Scientifically.

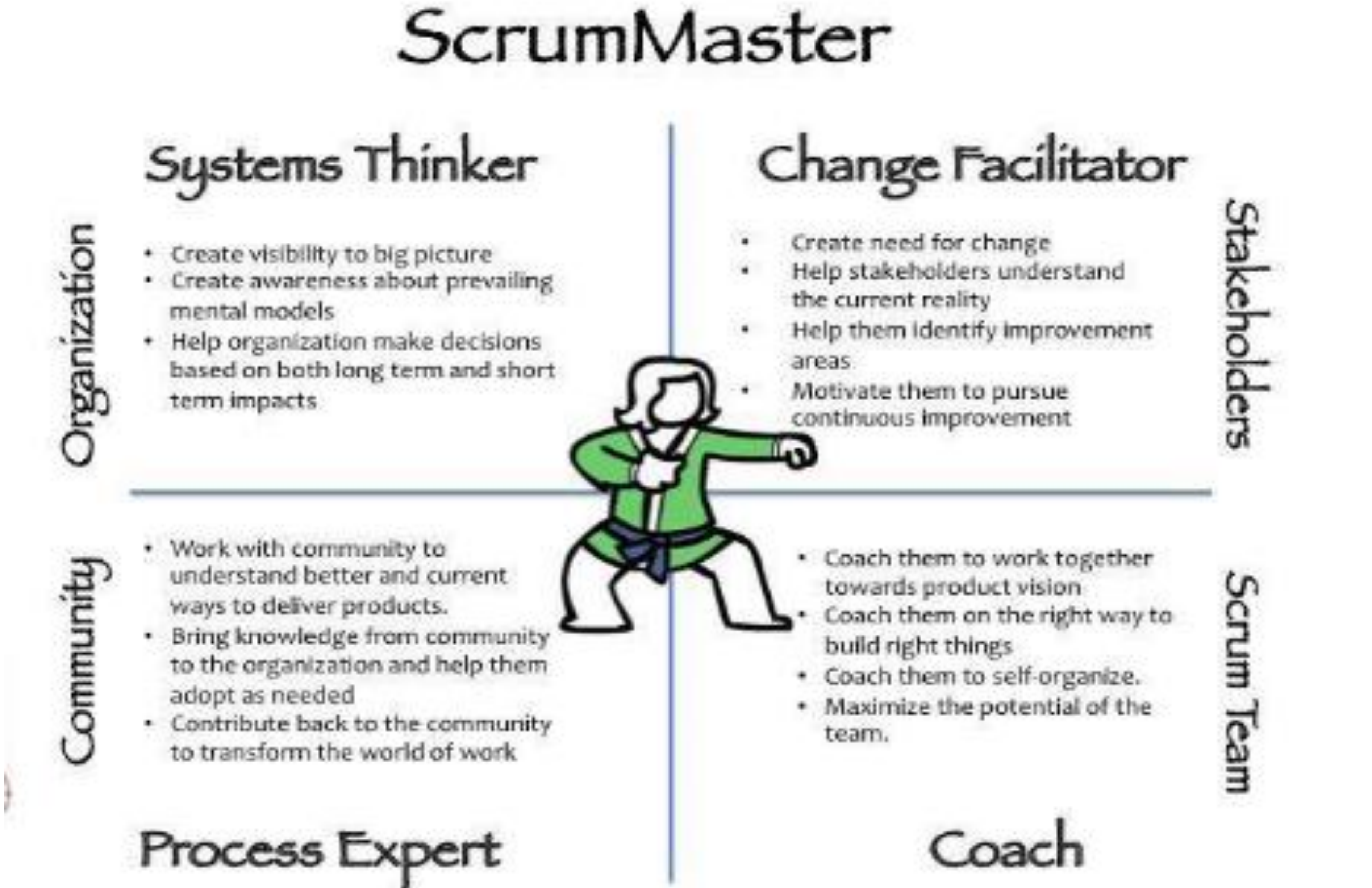
Scrum Master - Characteristics

- 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

Scrum Master - 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, and
- remove waste

CYBAGE
Delivering Value. Scientifically.



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

Process Expert

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

Coach

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.

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

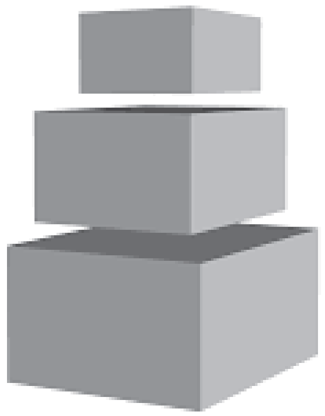
5 Values

- Courage
- Openness
- Focus
- Commitment
- Respect

Product Backlog

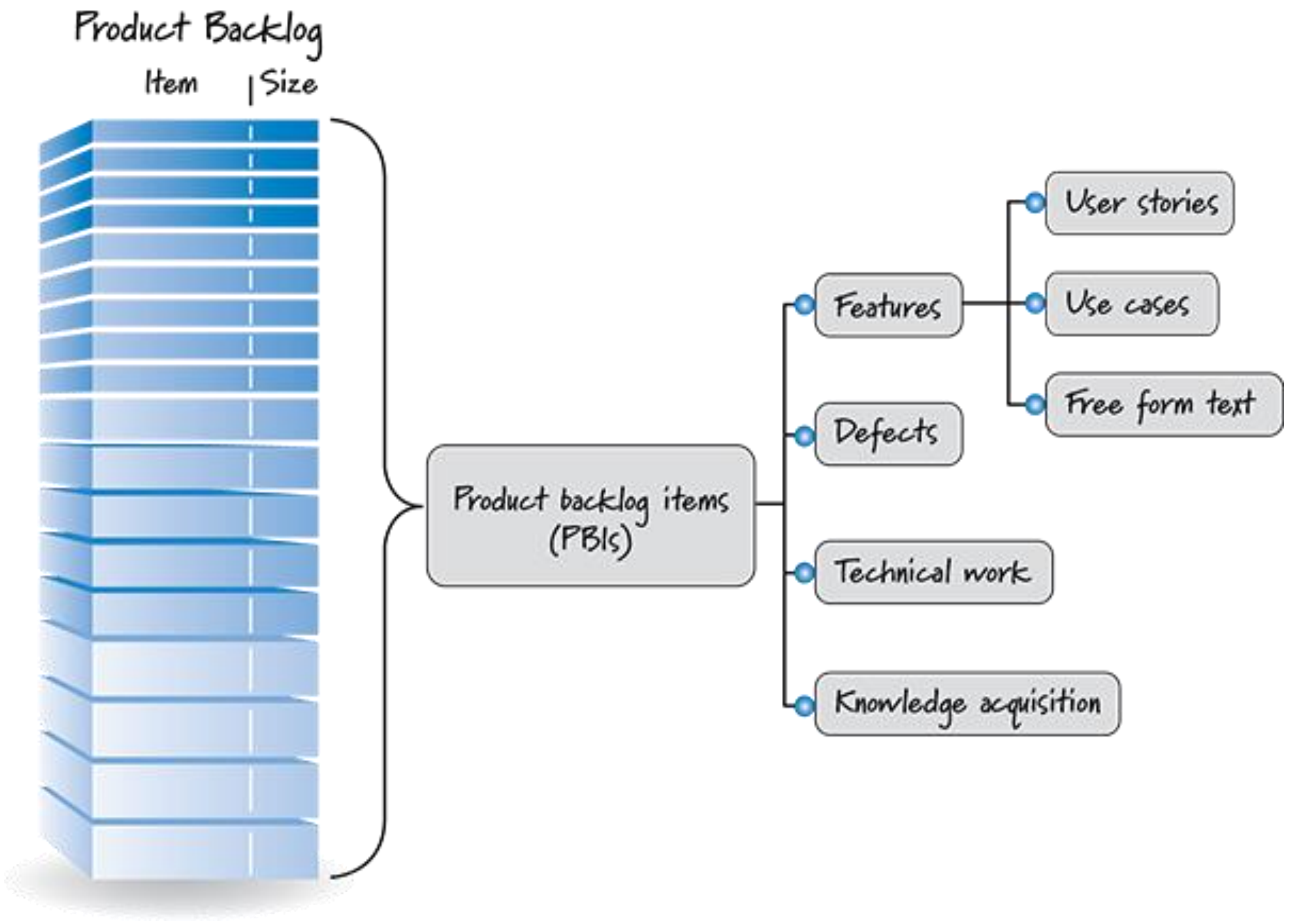
Product Backlog

- Requirements
- A list of all desired work on the project
- Prioritized by the Product Owner
- Reprioritized at the start of each sprint



Product Backlog
(Features)

Product Backlog



Difference between a user story and a use case?

- Both "user stories" and "use cases" are terms used in gathering requirements from customers in a software development project,
- but there are some major differences, most notably in the
- **<u>level of detail</u>** of each.

Use cases	User stories
• Became a popular way in the 1990's of documenting requirements for object-oriented systems. • Describe the interaction between one or more "actors" with the system.	• are short and • briefly state – the type of **user**, – the **feature** needed, and – the **benefit**.

CYBAGE
Delivering Value. Scientifically.

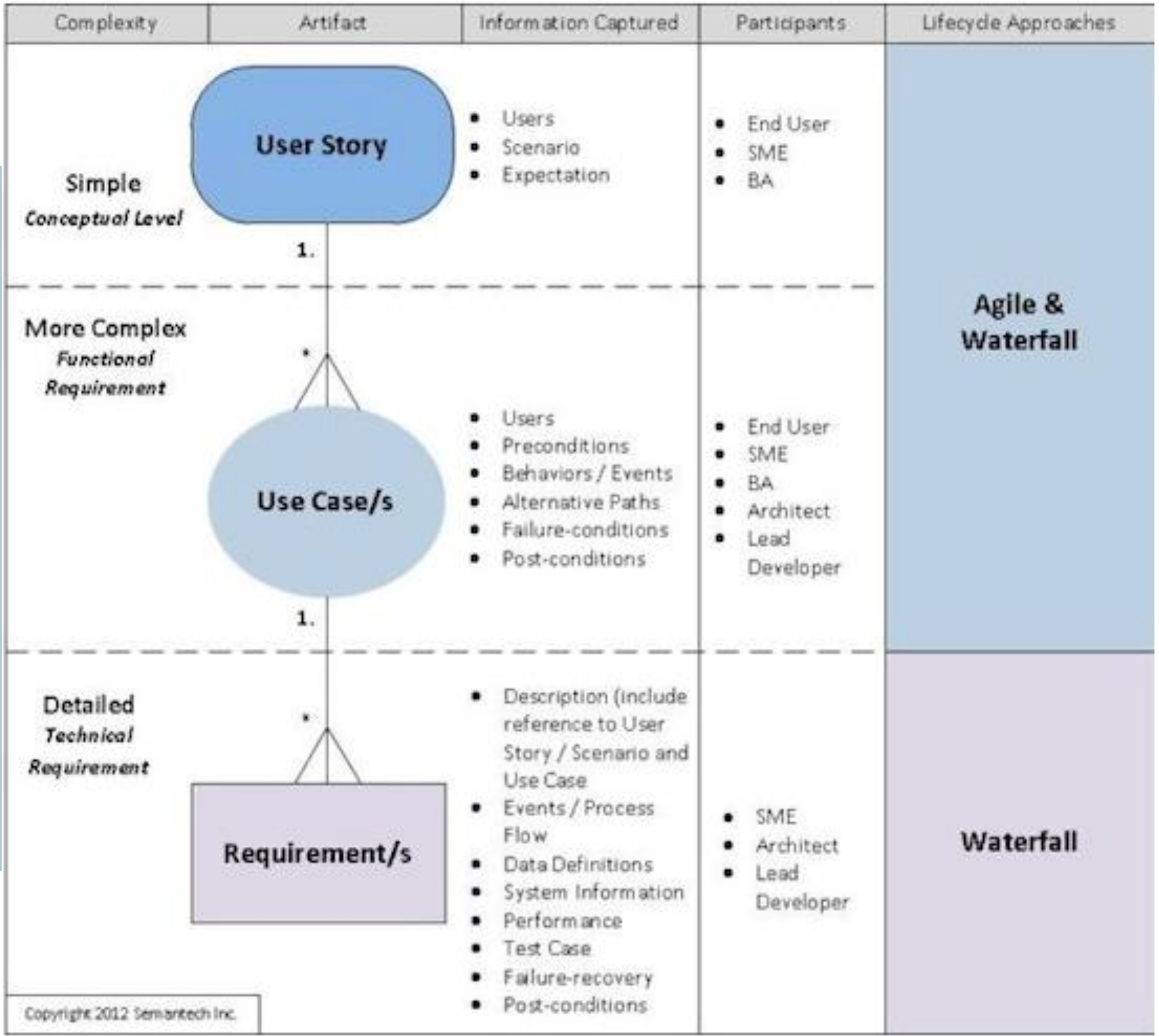
Use case

Vs.

User Story

Vs.

Requirements



Complexity	Artifact	Information Captured	Participants	Lifecycle Approaches
Simple *Conceptual Level*	User Story	• Users • Scenario • Expectation	• End User • SME • BA	Agile & Waterfall
More Complex *Functional Requirement*	Use Case/s	• Users • Preconditions • Behaviors / Events • Alternative Paths • Failure-conditions • Post-conditions	• End User • SME • BA • Architect • Lead Developer	
Detailed *Technical Requirement*	Requirement/s	• Description (include reference to User Story / Scenario and Use Case • Events / Process Flow • Data Definitions • System Information • Performance • Test Case • Failure-recovery • Post-conditions	• SME • Architect • Lead Developer	Waterfall

Story Points - Estimating the Size

➢ Let’s talk Estimating

- 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” size
- Story Points
  - Complexity
  - Effort
  - Doubt
- Entire Team consensus

Relative Sizing

Story Points Are Relative

- Story points are a unit of measure for expressing the overall size of a user story, feature, or other piece of work. When we estimate with story points, we assign a point value to each item. The raw values we assign are unimportant. What matters are the relative values. A story that is assigned a two should be twice as much as a story that is assigned a one. It should also be two-thirds of a story that is estimated as three story points.

Points Are Size, not Duration

- It is important for understanding points to divorce the concepts of size and duration for the thing being estimated. This is explained further in the following section(s) but it is worth repeating many times because it is a paradigm shift that is commonly missed when communicating about points. In fact, it is generally acknowledged that the point-based estimation technique's most common failure or shortcoming is that people (including those doing the estimation) fail to comprehend and apply this timeless property of "the point".

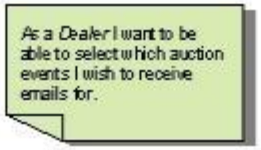
- A common question at this point becomes "if points are not duration-based estimates, how do I know how long something estimated as n points will take to do?" To understand how points address this question, let's use an analogy. Assume you are tasked with painting the walls of the rooms in a house that has the floor plan shown next slide

User Stories

User Stories are an established method of clear communication between the team & the business

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.

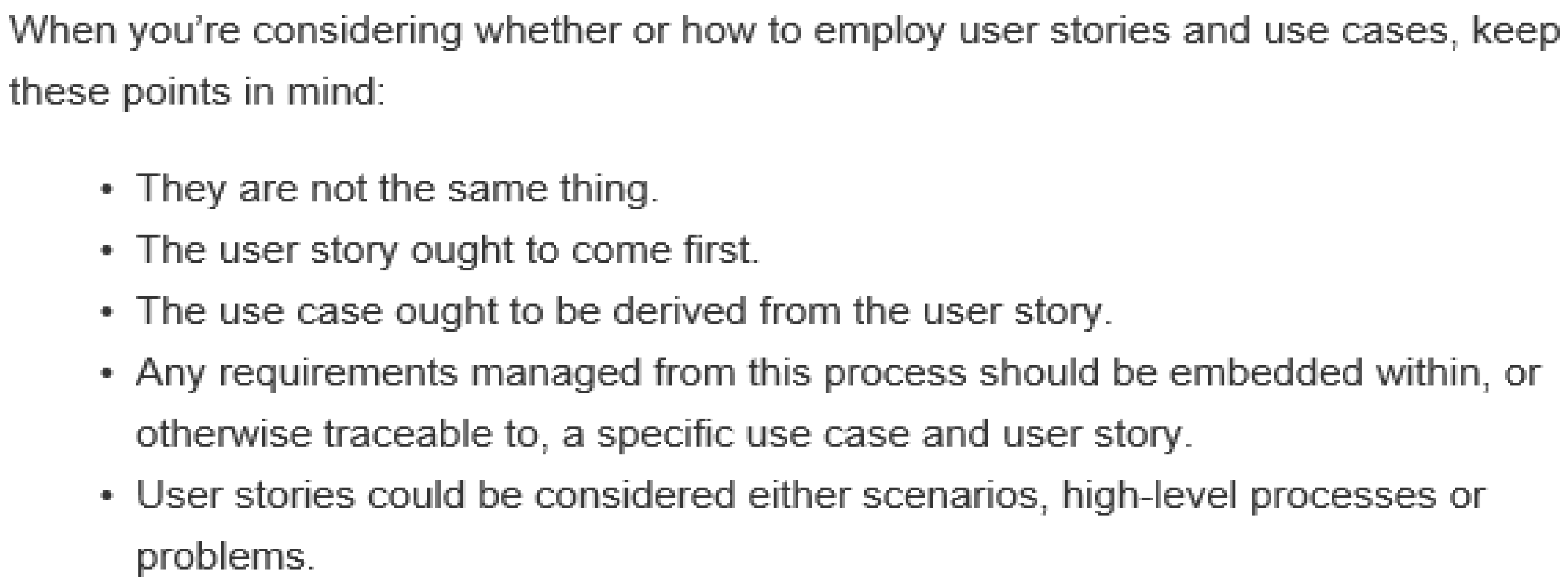
Conversation

- 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

Difference between a user story and a use case?

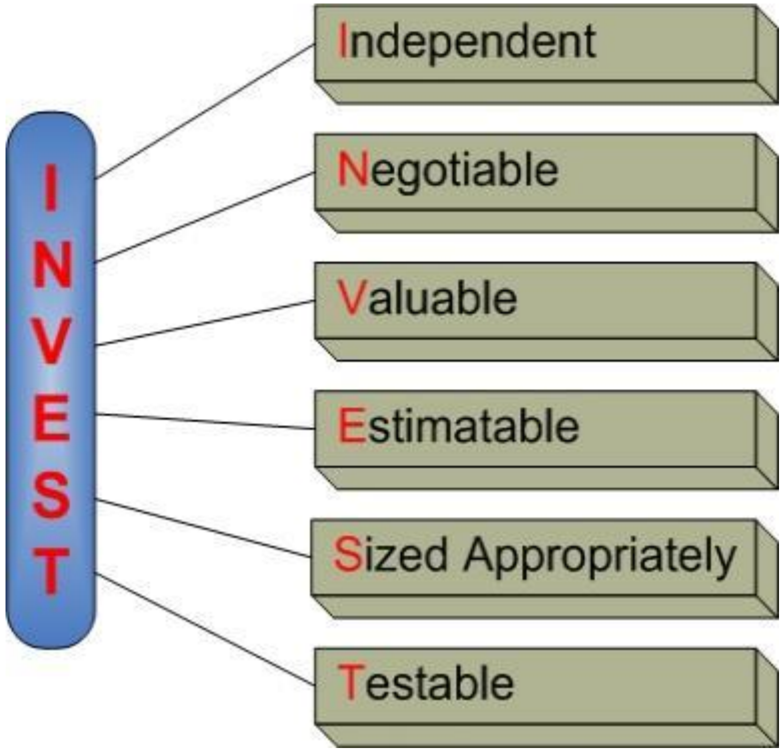


When you're considering whether or how to employ user stories and use cases, keep these points in mind:

- They are not the same thing.
- The user story ought to come first.
- The use case ought to be derived from the user story.
- Any requirements managed from this process should be embedded within, or otherwise traceable to, a specific use case and user story.
- User stories could be considered either scenarios, high-level processes or problems.

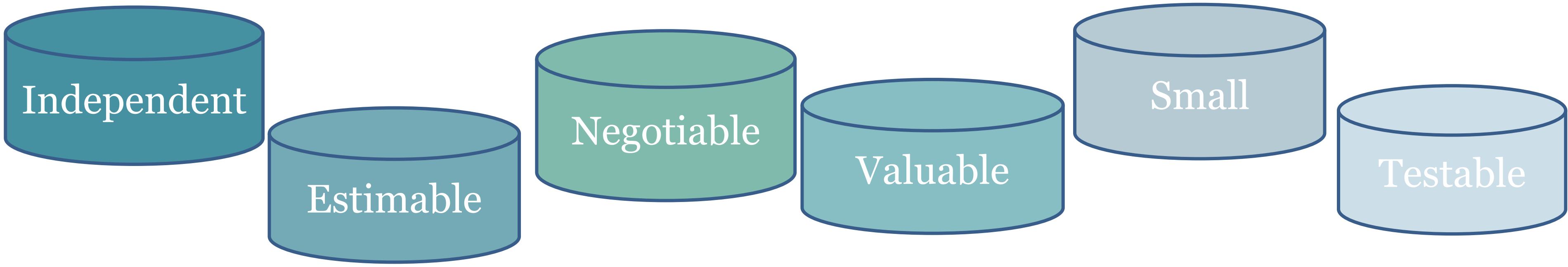
What makes a good story?

- **INVEST** acronym can be found in Mike Cohn’s “User Stories Applied”



Ref: www.mountaingoatsoftware.com

Six Features of a Good User Story



Independent - One user story should be independent of another (as much as possible). Dependencies between stories make planning, prioritization, and estimation much more difficult. Often enough, dependencies can be reduced by either combining stories into one or by splitting the stories differently.

Negotiable - A user story is negotiable. The "Card" of the story is just a short description of the story which do not include details. The details are worked out during the "Conversation" phase. A "Card" with too much detail on it actually limits conversation with the customer.

Valuable - Each story has to be of value to the customer (either the user or the purchaser). One very good way of making stories valuable is to get the customer to write them. Once a customer realizes that a user story is not a contract and is negotiable, they will be much more comfortable writing stories.

Estimable - The developers need to be able to estimate (at a ballpark even) a user story to allow prioritization and planning of the story. Problems that can keep developers from estimating a story are: lack of domain knowledge (in which case there is a need for more Negotiation/Conversation); or if the story is too big (in which case the story needs to be broken down into smaller stories).

Small - A good story should be small in effort, typically representing no more, than 2-3 person weeks of effort. A story which is more than that in effort can have more errors associated with scoping and estimation.

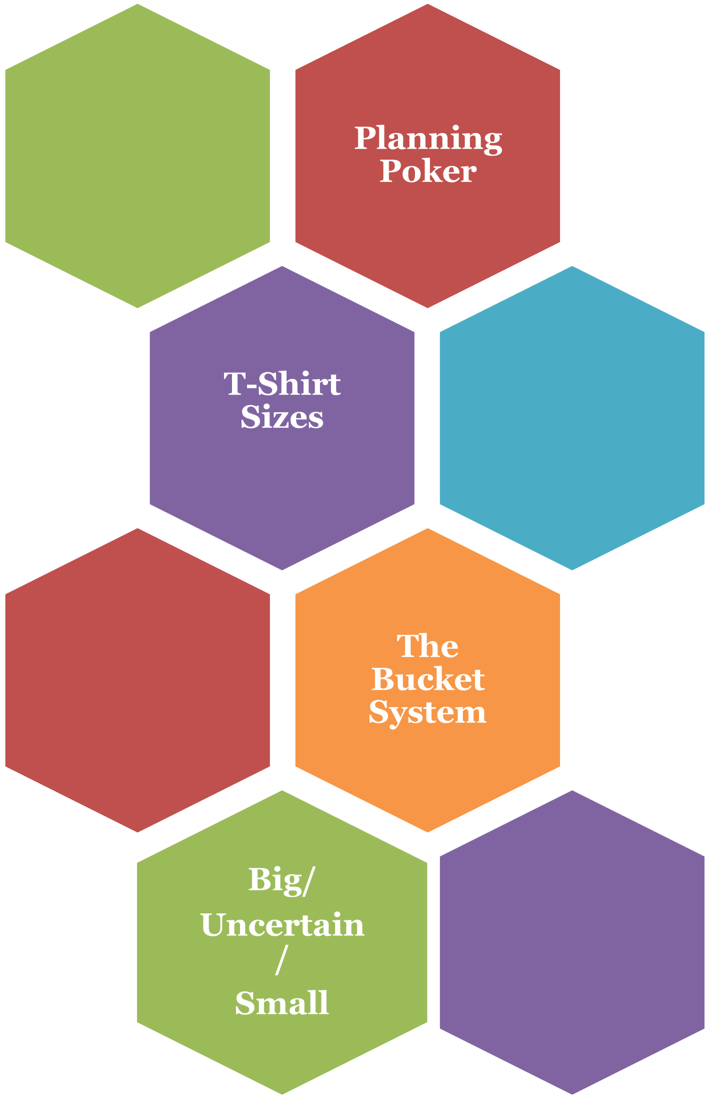
Testable - A story needs to be testable for the "Confirmation" to take place. Remember, we do not develop what we cannot test. If you can't test it then you will never know when you are done. An example of non-testable story: "software should be easy to use".

Ref: www.mountaingoatsoftware.com

Delivering Value. Scientifically.

Agile estimation techniques

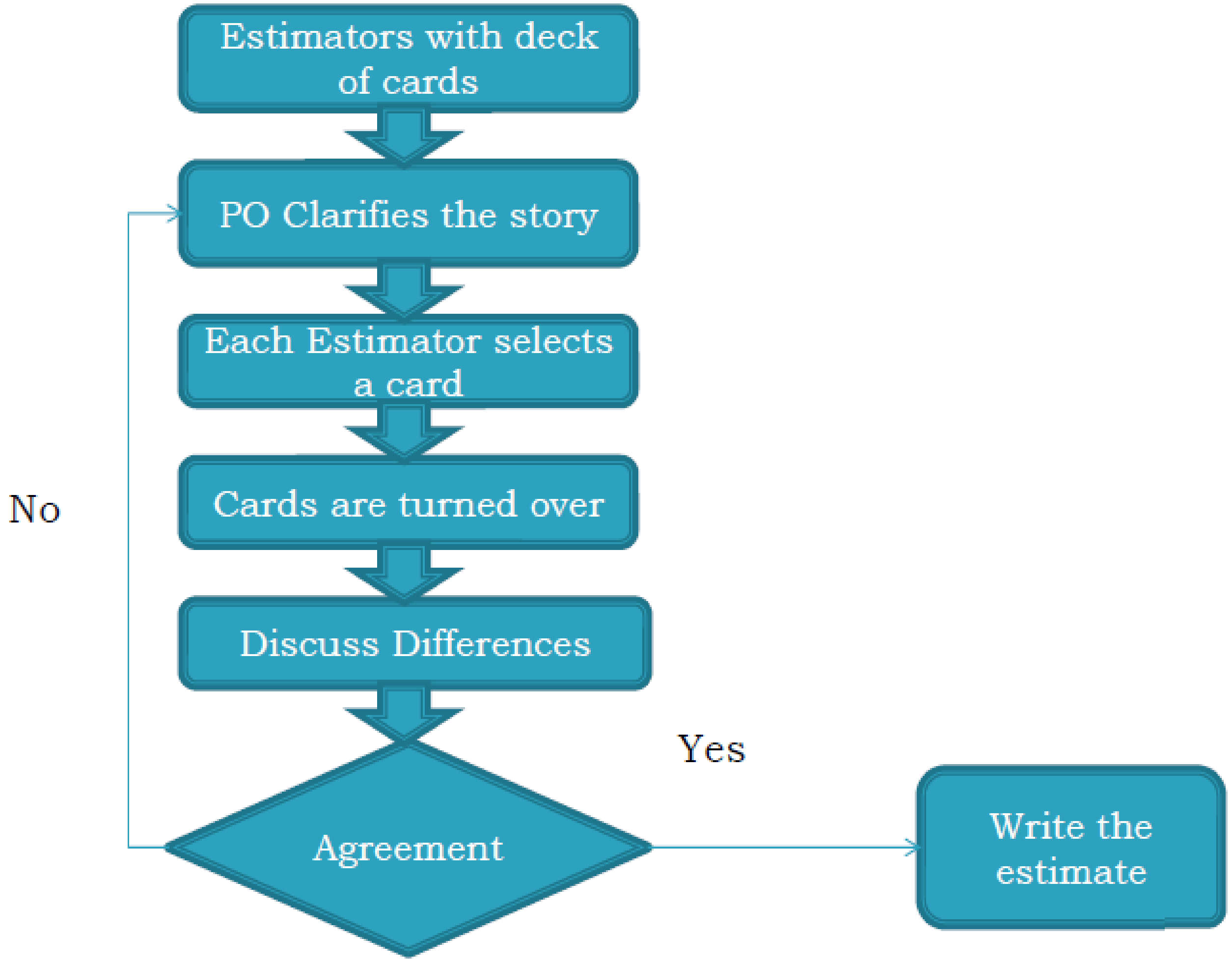
- Many people have used a variation of Planning Poker to do Agile estimation.
- Here is a reference of **9 different Agile estimation techniques** for different circumstances.
- I have seen all of these techniques work in practice, except one.
- Try a new one each Sprint!



Planning poker

- 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



Ref: Internal – Org Library

CYBAGE
Delivering Value. Scientifically.

ATM User story Example:

ATM User

I want to WITHDRAW funds from my bank account
So that : I can increase my cash on hand

I want to PRINT a bank statement
So that : I know how much cash I have in my account

ATM User

ATM Operator

I want to set the ATM's withdrawal parameters
So that : AT M will provide funds to customer but protect against fraudulent activities

I want to restock the ATM money
So that : AT M will be available for customer to withdraw funds

Bank Owner

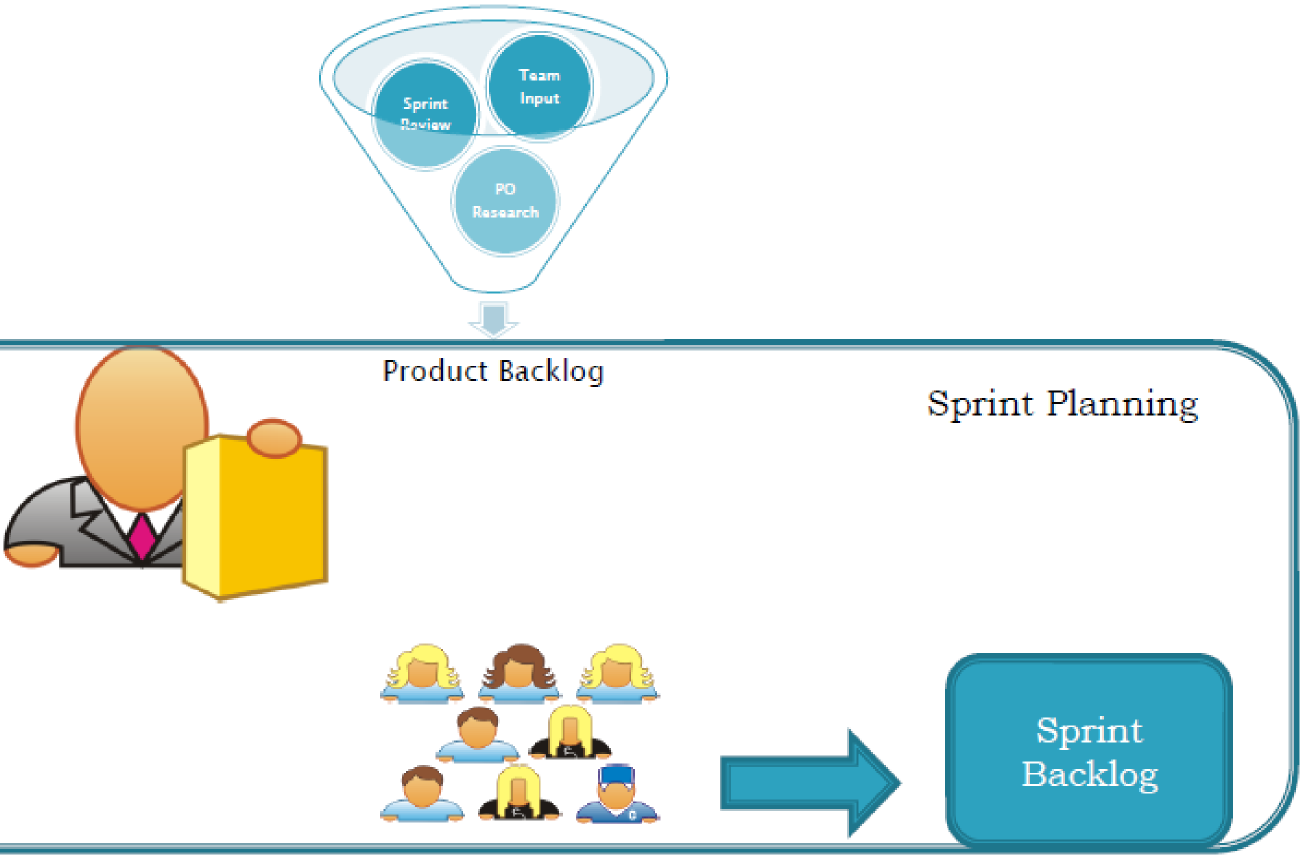
Definition of “Ready”- DoR

- Meets all acceptance criteria
- Story is fully groomed with the whole team including but not limited to Dev, QA, DA, DM, Architect, PO, and appropriate SMEs
- All the acceptance criteria is defined
- 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

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

Product Backlog



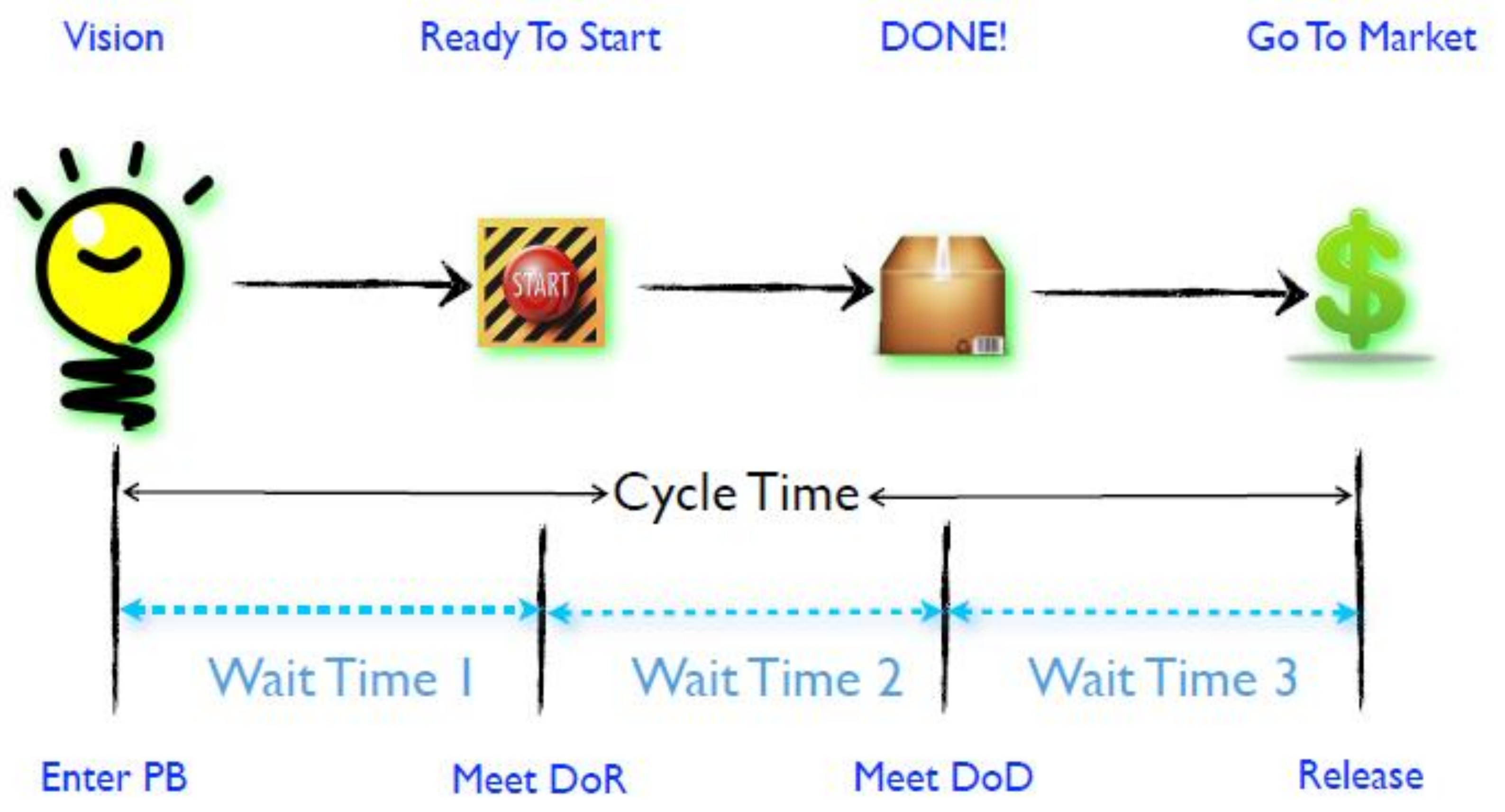
Good Product Backlog

- **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

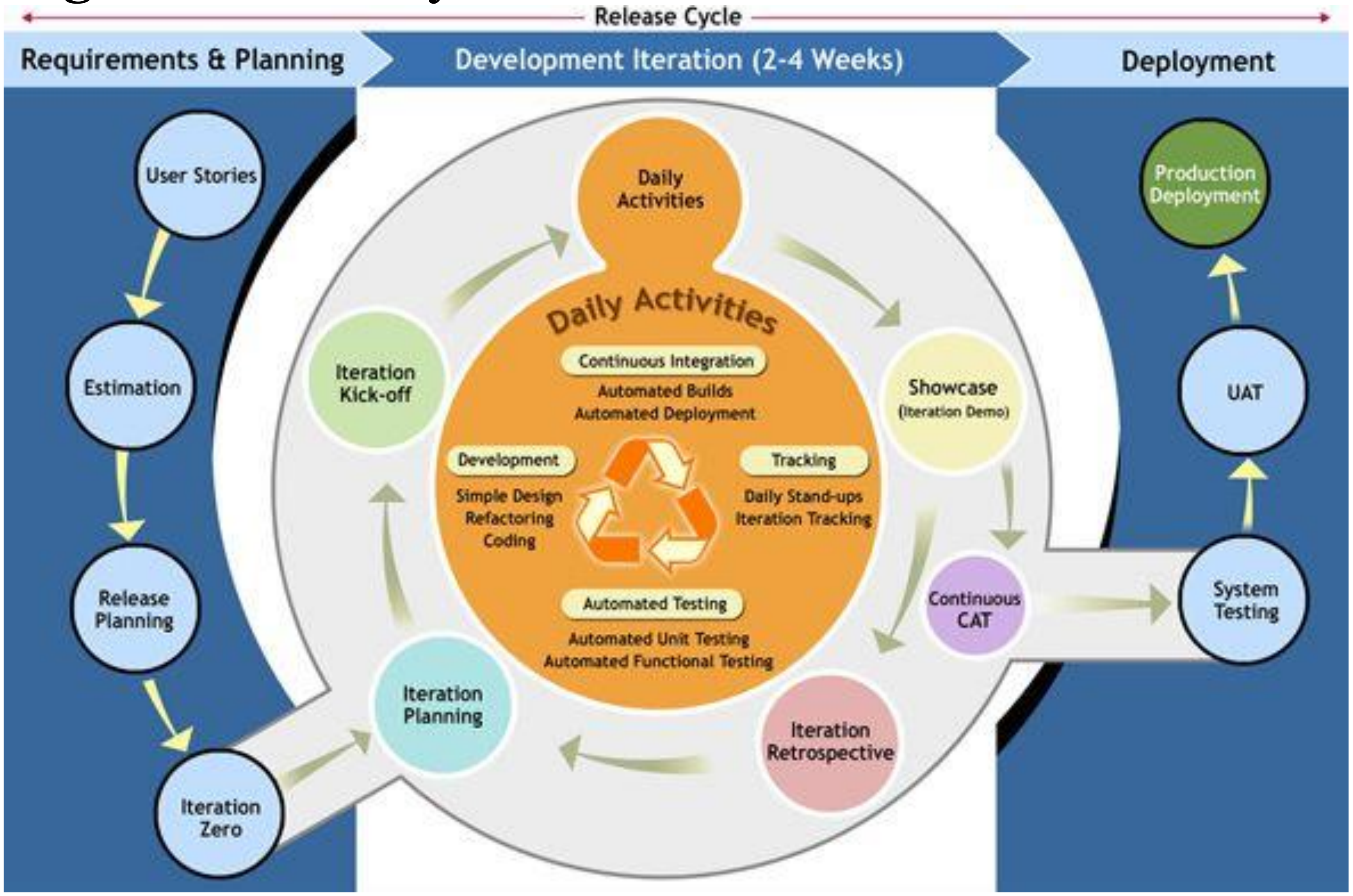
DEEP

Cycle Time for a feature



CYBAGE
Delivering Value. Scientifically.

Agile Release Cycle

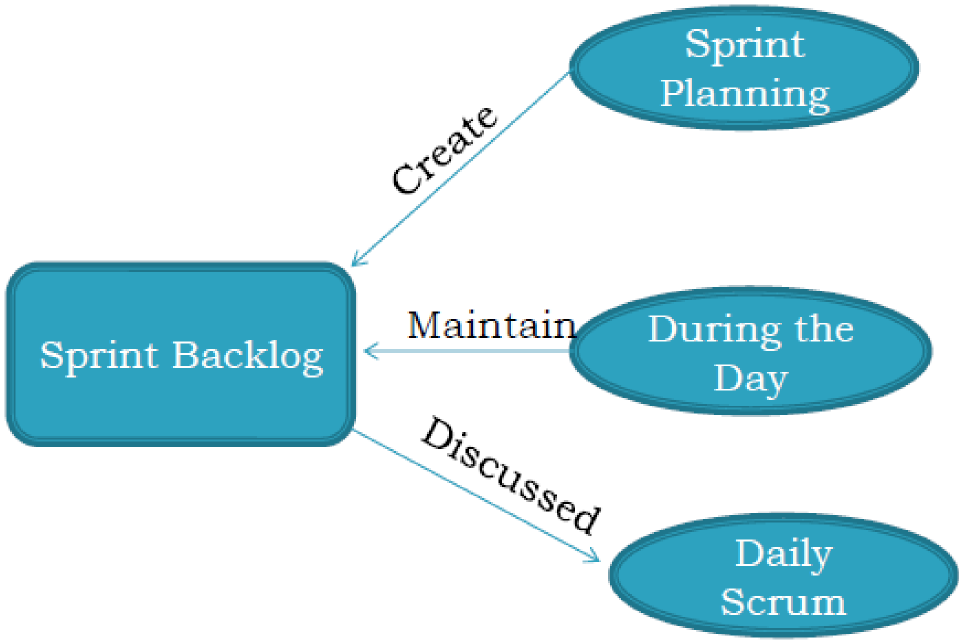


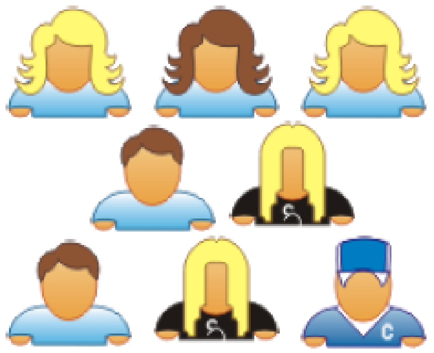
Sprint Backlog

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

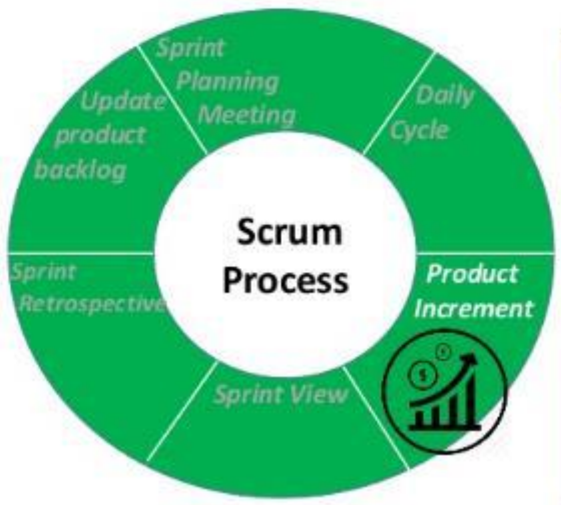
Sprint Backlog





Increment

Increment

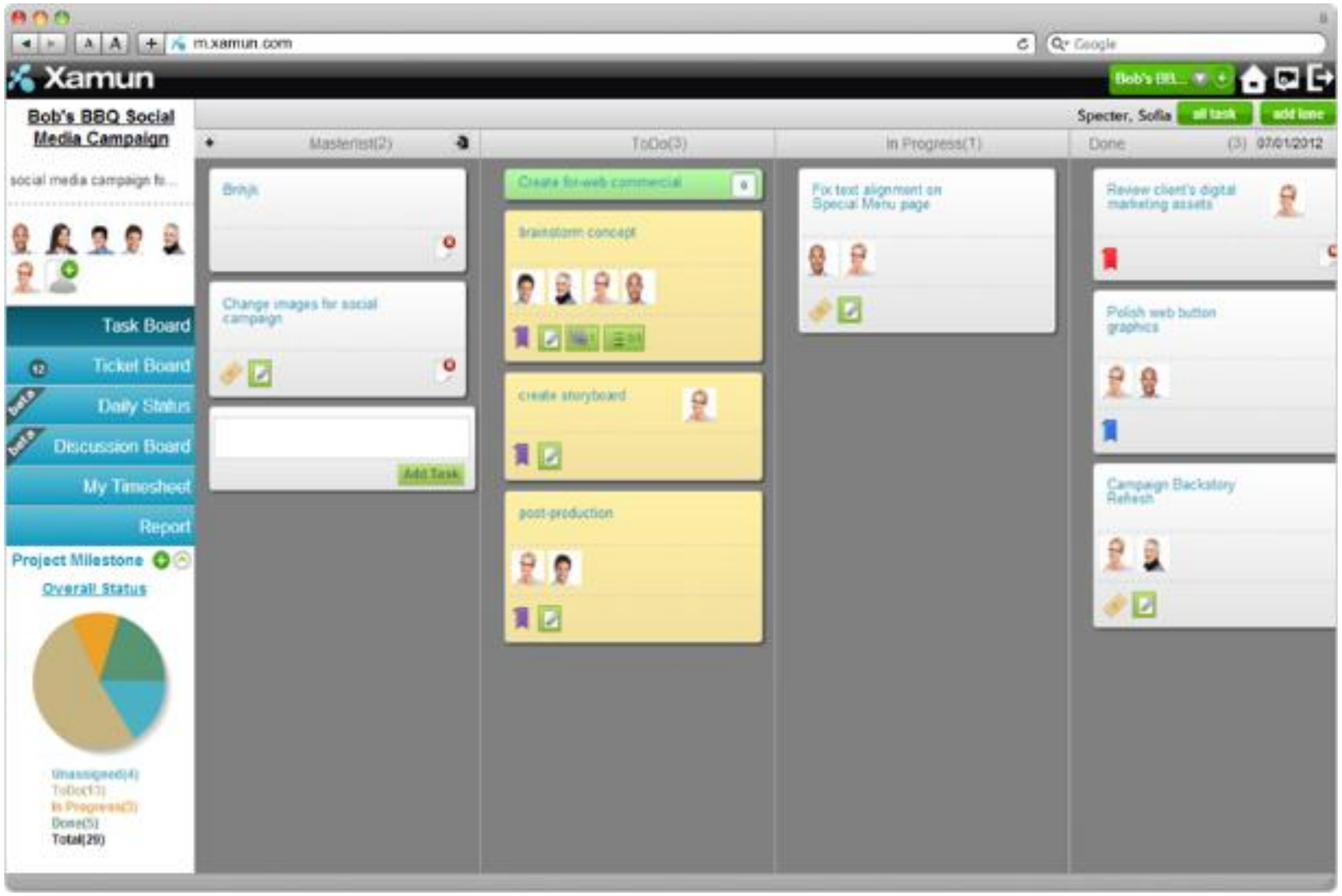


Product Increment

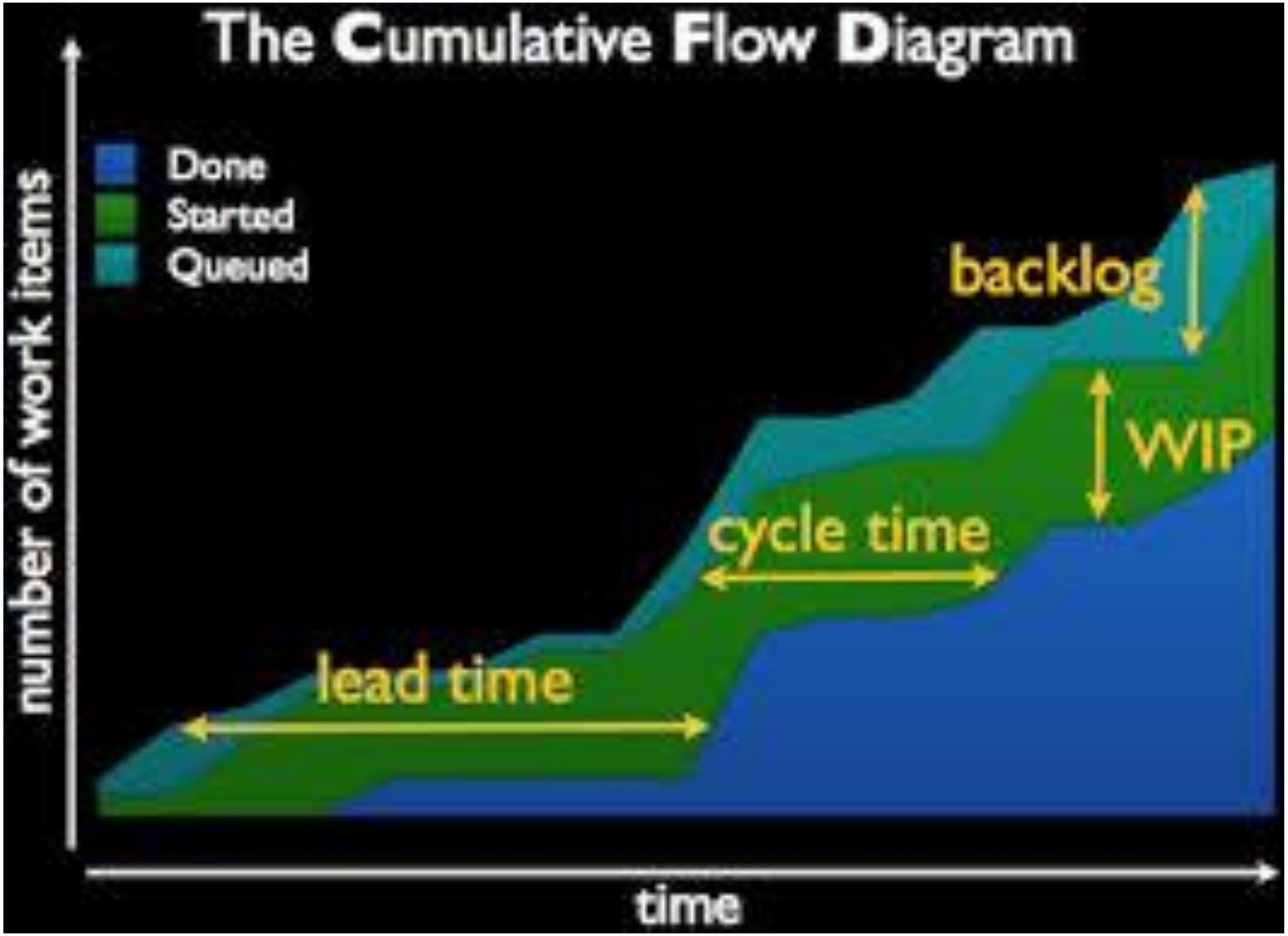
- ✓ The sum of all the product backlog Items completed during a sprint and all previous sprints.
- ✓ At the end of a sprint, the increment must be complete.

- Increment 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

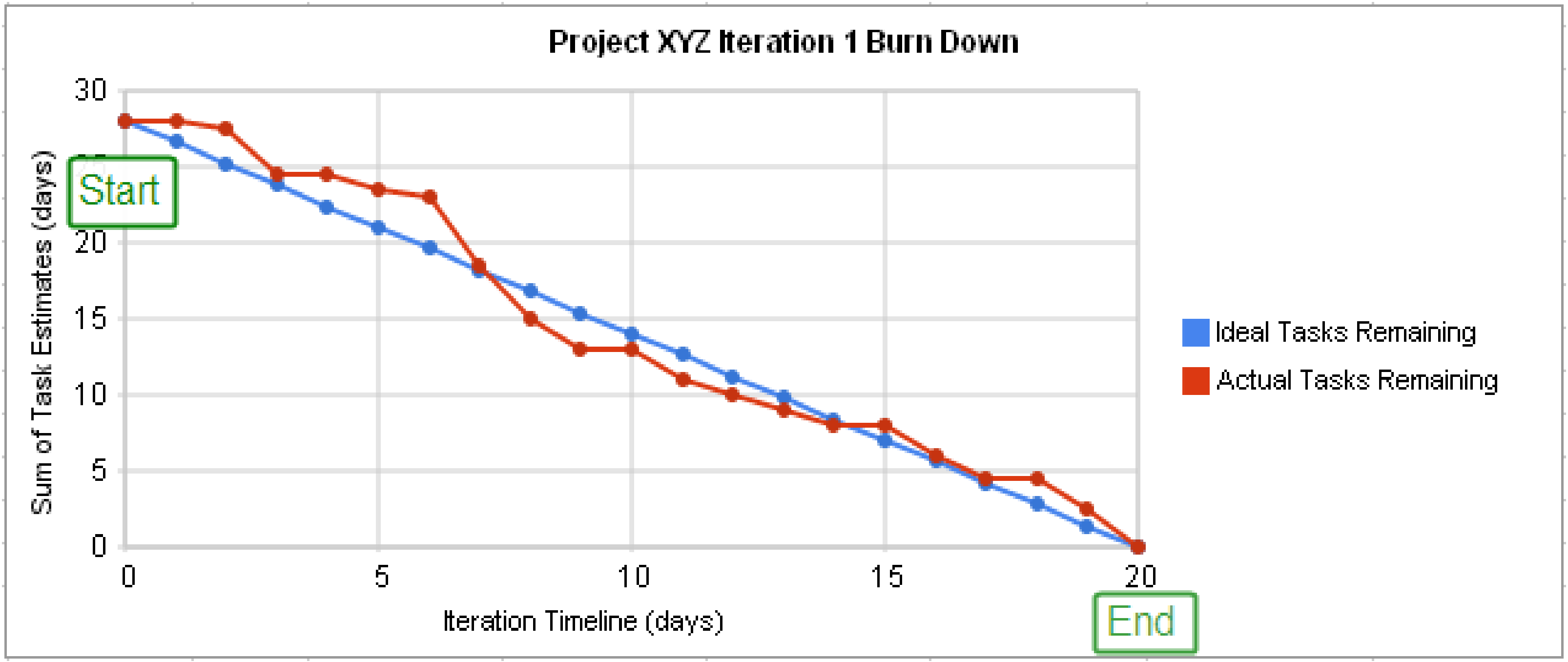
Increments – Task Board



- 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
backlog
WIP
cycle time
lead time
number of work items
time

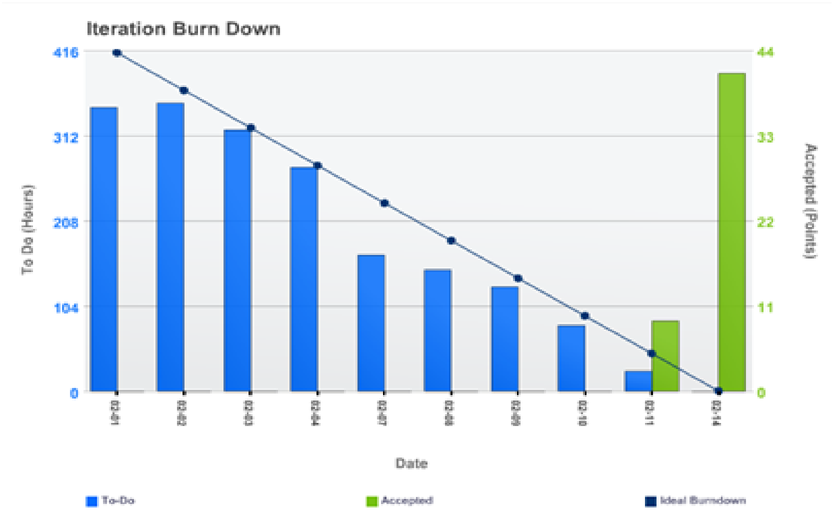
Increments – Burn Down Charts

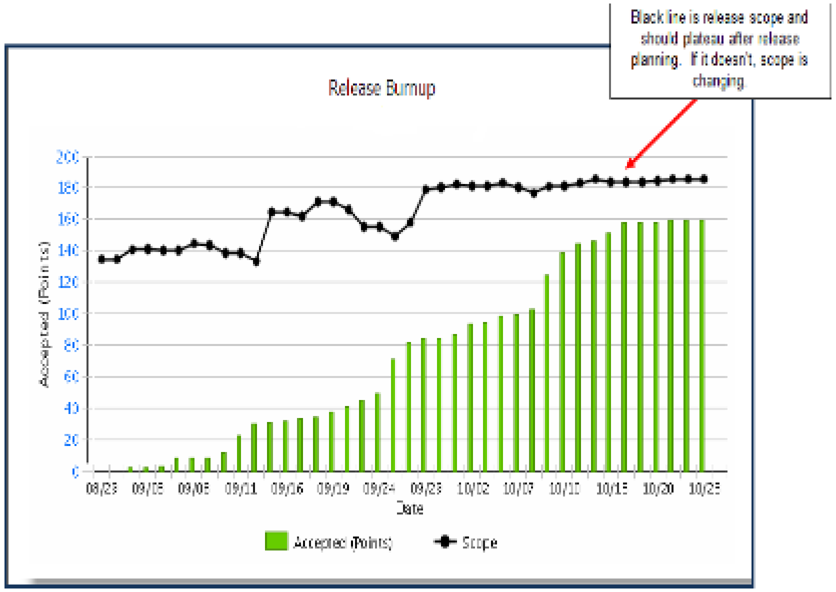


- Updated daily, usually during the daily scrum
- Represents the total amount of work remaining
- Track **DONE** only

Burn Charts (Up & Down)

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.



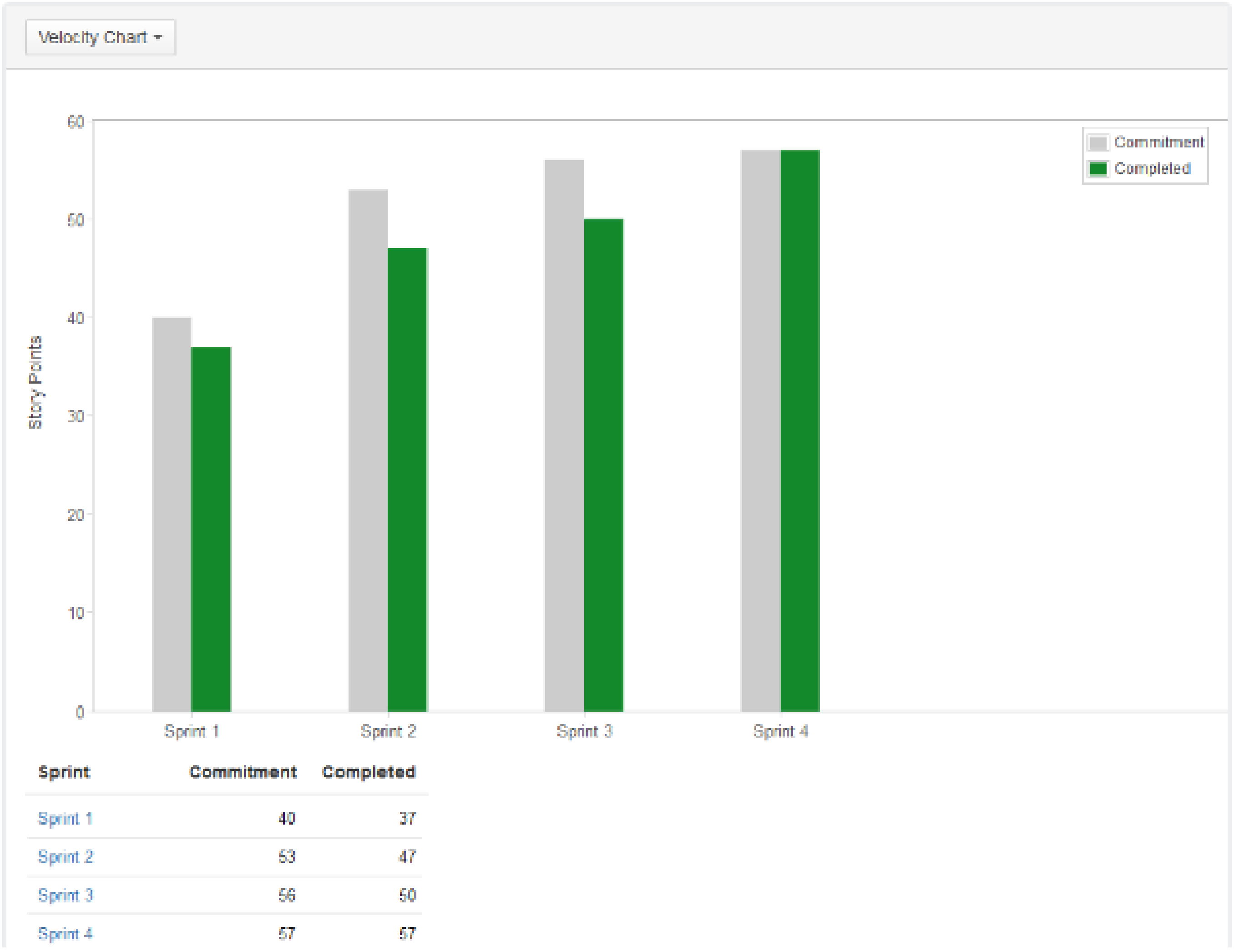


A **burn up** chart, tracks progress towards a projects completion. The Release Burnup displays the work delivered so far in the release versus the scope planned.

It is reviewed to proactively (after every sprint optimally) to anticipate whether the release scope will be delivered.

Ref: Internal – Org Library

Team Velocity



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

CYBAGE
Delivering Value. Scientifically.

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

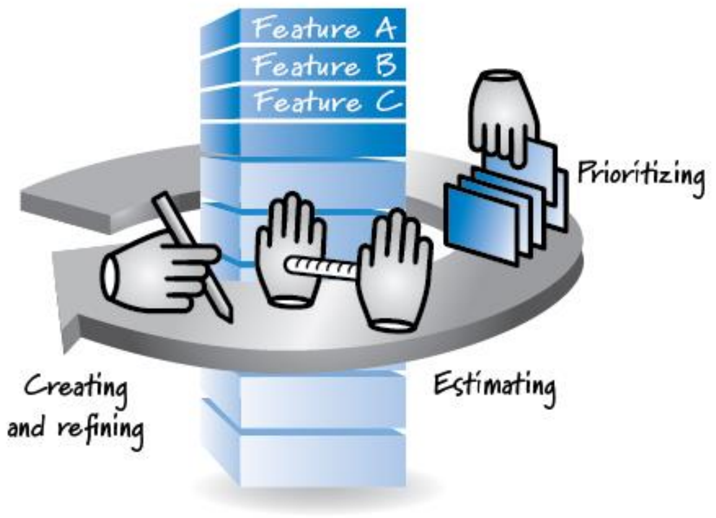
5 Values

- Courage
- Openness
- Focus
- Commitment
- Respect

Product Backlog Refinement

Product Backlog Refinement

Product Backlog refinement



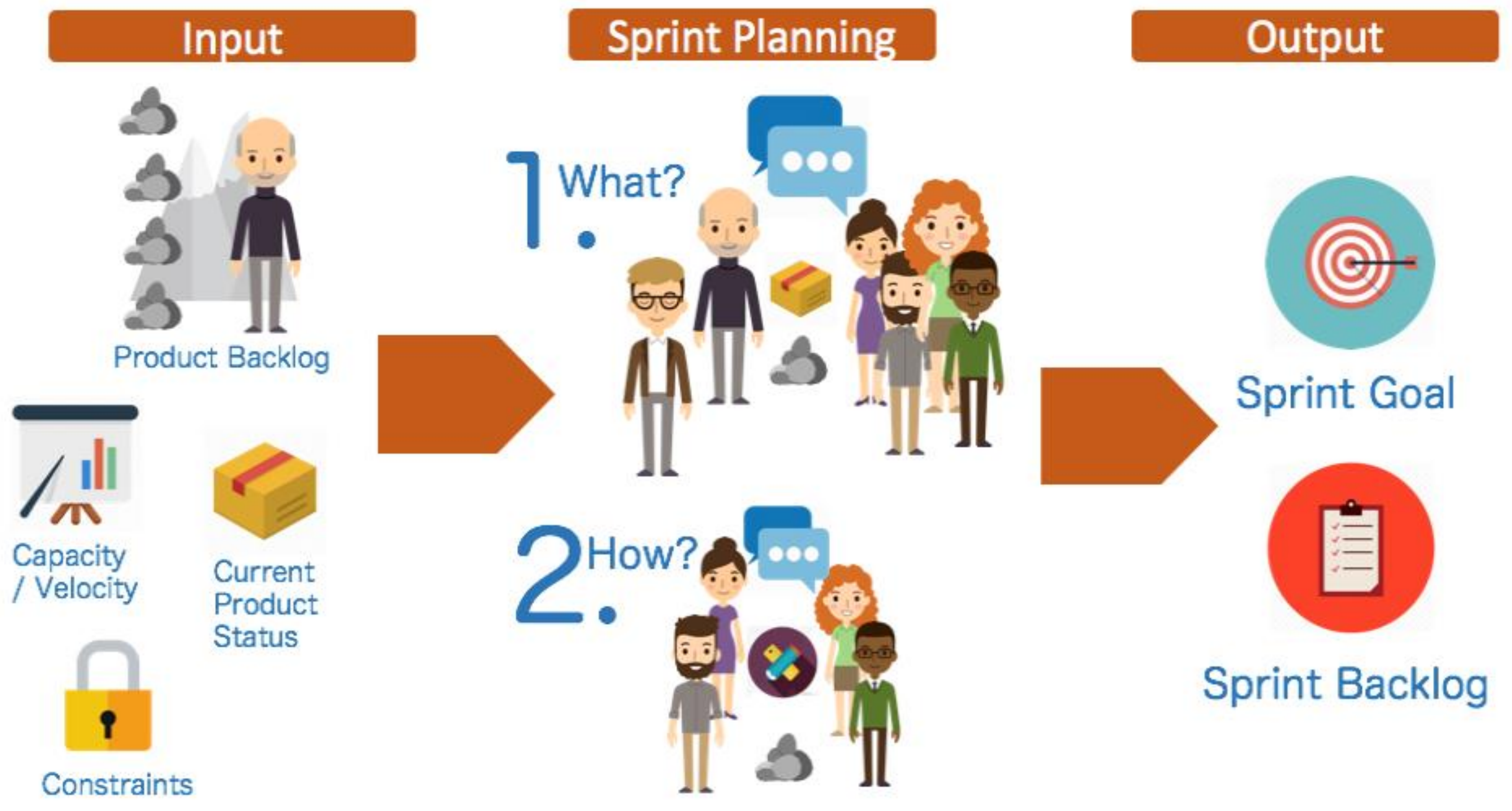
- Keep the Product Backlog in order
- Remove or demote Product Backlog Items that no longer seem important
- Add or promote Product Backlog Items that raise or become more important
- Split Product Backlog Items into smaller items
- Merge Product Backlog Items into larger items
- Estimate Product Backlog

Product Backlog refinement 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 Planning

Sprint Planning



Velocity

- Measures the amount of work delivered by a team over a period of time
- Not a measure of productivity
- It is the sum of story points/ideal time accomplished during an iteration cycle
- Can change with every iteration/ sprint
- **Factors affecting velocity**
  - Team size
  - Resource availability
  - Team member experience/capability
  - Team cohesiveness
- Should not include bugs and rejected stories
- Helps to make commitments of future work
- Velocity cannot be compared against other teams

Ways to improve velocity:

- Reduce technical debt
- Protect team from distractions
- Better customer involvement
- Support energized work
- Right resources at the right time
- Eliminate bottlenecks

Formula

Velocity = Estimated # of iterations * Total size in story points

Sprint Planning

- The Team and the Product Owner collaborate to help the Team determine how much Product Backlog it can turn into functionality during the upcoming sprint
- The teams create plans (Sprint Backlog) by identifying tasks for converting selected Product Backlog into functionality

Sprint Planning

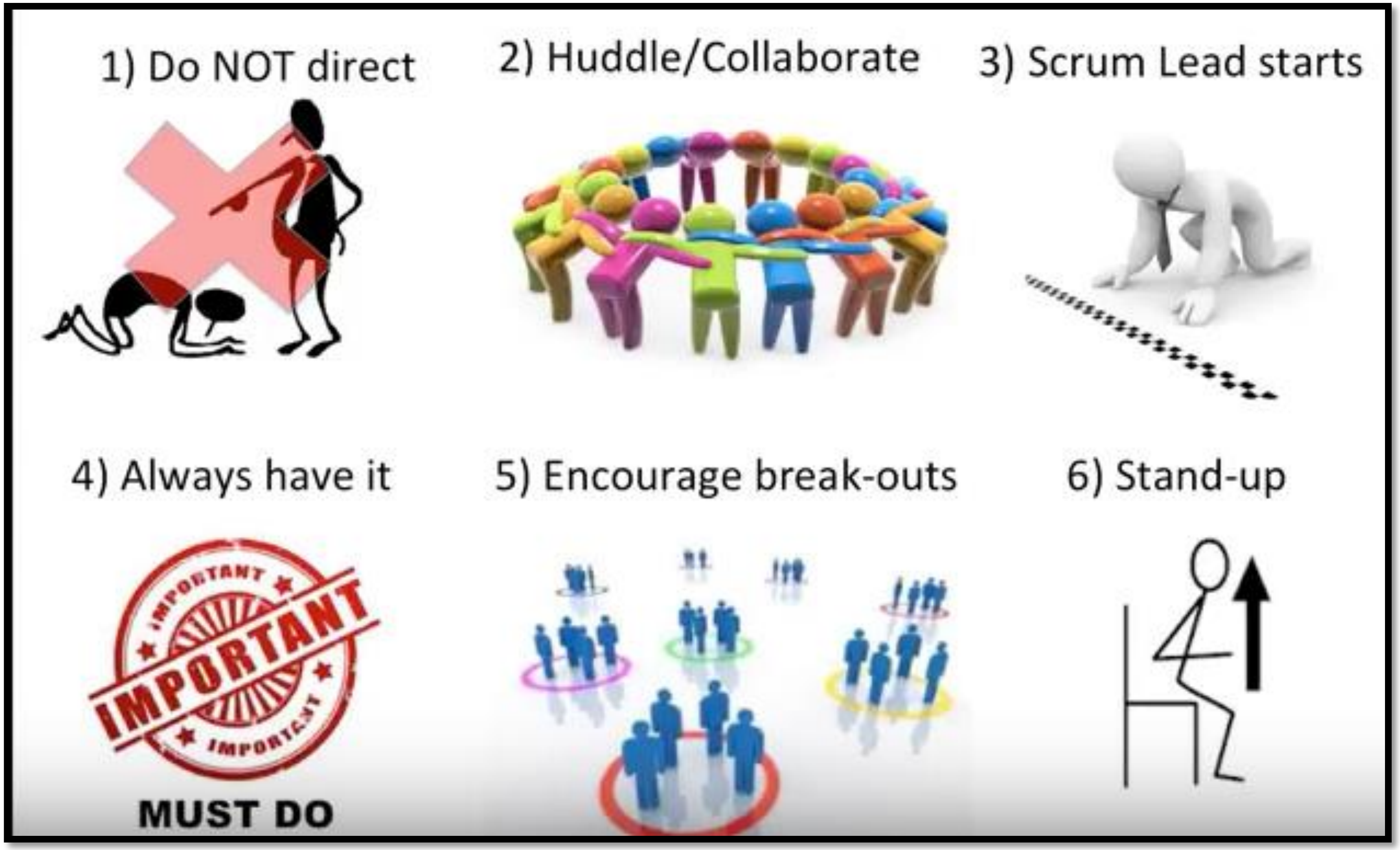
- Team selects items from the product backlog they can commit to completing
- Sprint backlog is created
    - Task are identified and each are estimated (1 – 16 hours)
    - Collaboratively, not done ***alone*** by the Scrum Master
- High-level design is considered

Daily Scrum Meeting

Why Daily Scrum?

- Fine-grain coordination
- Daily commitment
- Raising impediments
- Peer pressure
- Access progress towards sprint goal

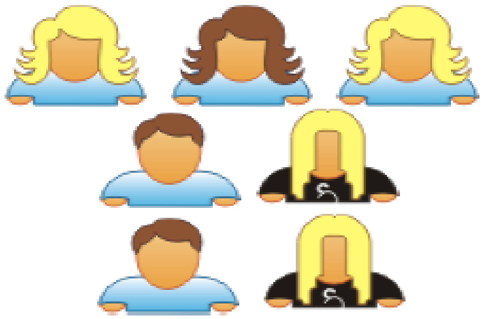
Daily Scrum Meeting



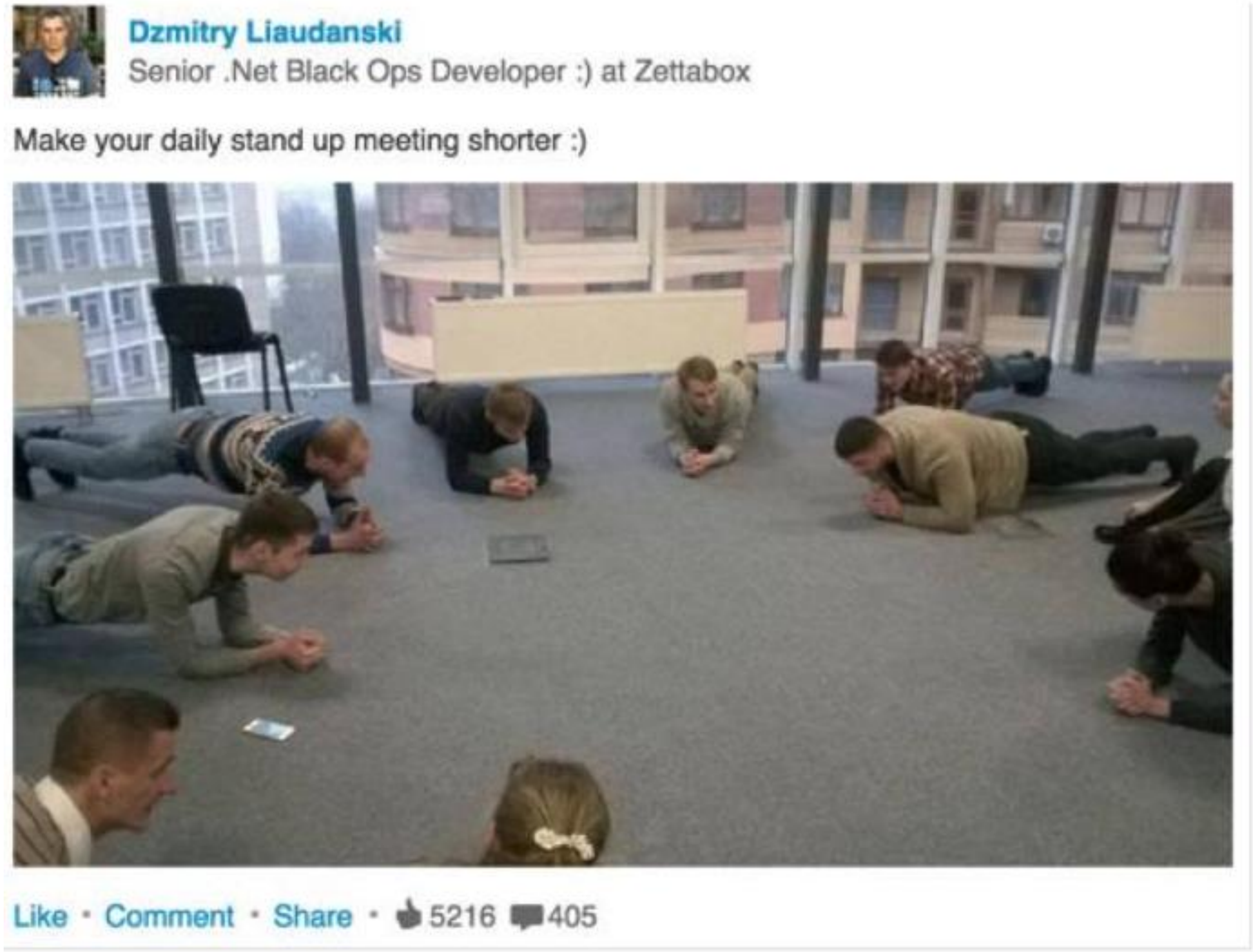
The Daily Scrum

1. What did you do yesterday?
2. What will you do today?
3. Is anything in your way?

- These are *not* status for the Scrum Master
  - They are commitments in front of peers



The Daily Scrum



Dzmitry Liaudanski

Senior .Net Black Ops Developer :) at Zettabox

Make your daily stand up meeting shorter :)

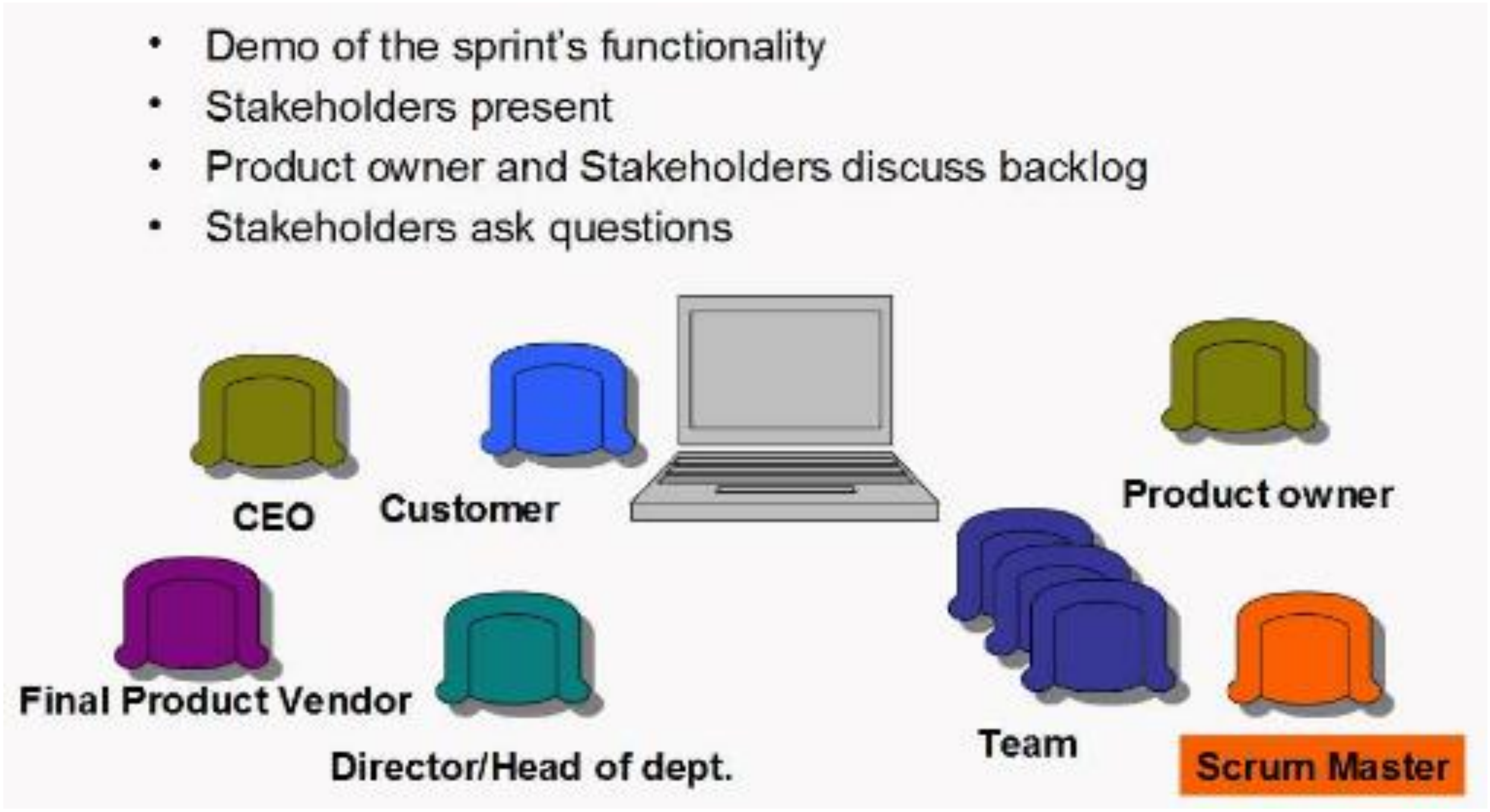
Like · Comment · Share · 5216 405

The Daily Scrum

- Parameters
  - Daily
  - 15-minutes
  - Stand-up
- Not for problem solving
  - Team members, Scrum Master, Product Owner can talk
- Help avoid other unnecessary meetings
- Team is responsible for conducting the meeting

Sprint Review

Sprint Review



- Demo of the sprint's functionality
- Stakeholders present
- Product owner and Stakeholders discuss backlog
- Stakeholders ask questions

The Sprint Review

- Team present what is accomplished during the sprint
- Typically takes the form of demo of new features
- Informal
  - 2-hour prep time rule
  - No slides
- Whole team participates
- Product owner accept or reject

Sprint Review

Feedback on Product
Demo
Feedback on Product
Facilitate

Sprint Retrospective

Retrospectives

- The sprint retrospective is a meeting facilitated by the Scrum Master at which the team discusses the just-concluded sprint and determines what could be changed that might make the next sprint more productive.
- Recommendation is to conduct it as a start-stop-continue meeting.
- This is perhaps the simplest, but often the most effective way to conduct a retrospective.
- Using this approach each team member is asked to identify specific things that the team should:
  - Start doing
  - Stop doing
  - Continue doing

What Went Well
What Could Have Been Improved
Actions

Sprint Retrospective

- Periodically take a look at what is and what is not working
- Done after every sprint
- Participants:
  - Scrum master
  - Team
  - Product Owner (Optional)

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

QUIZ…

5. Who write the user stories?

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

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…

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

a. Velocity

b. Cycle time

c. Burn rate

d. Capacity

QUIZ…

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

3 Roles

- Development Team
- Product Owner
- ScrumMaster

3 Artifacts

- Product Backlog
- Sprint Backlog
- Increment

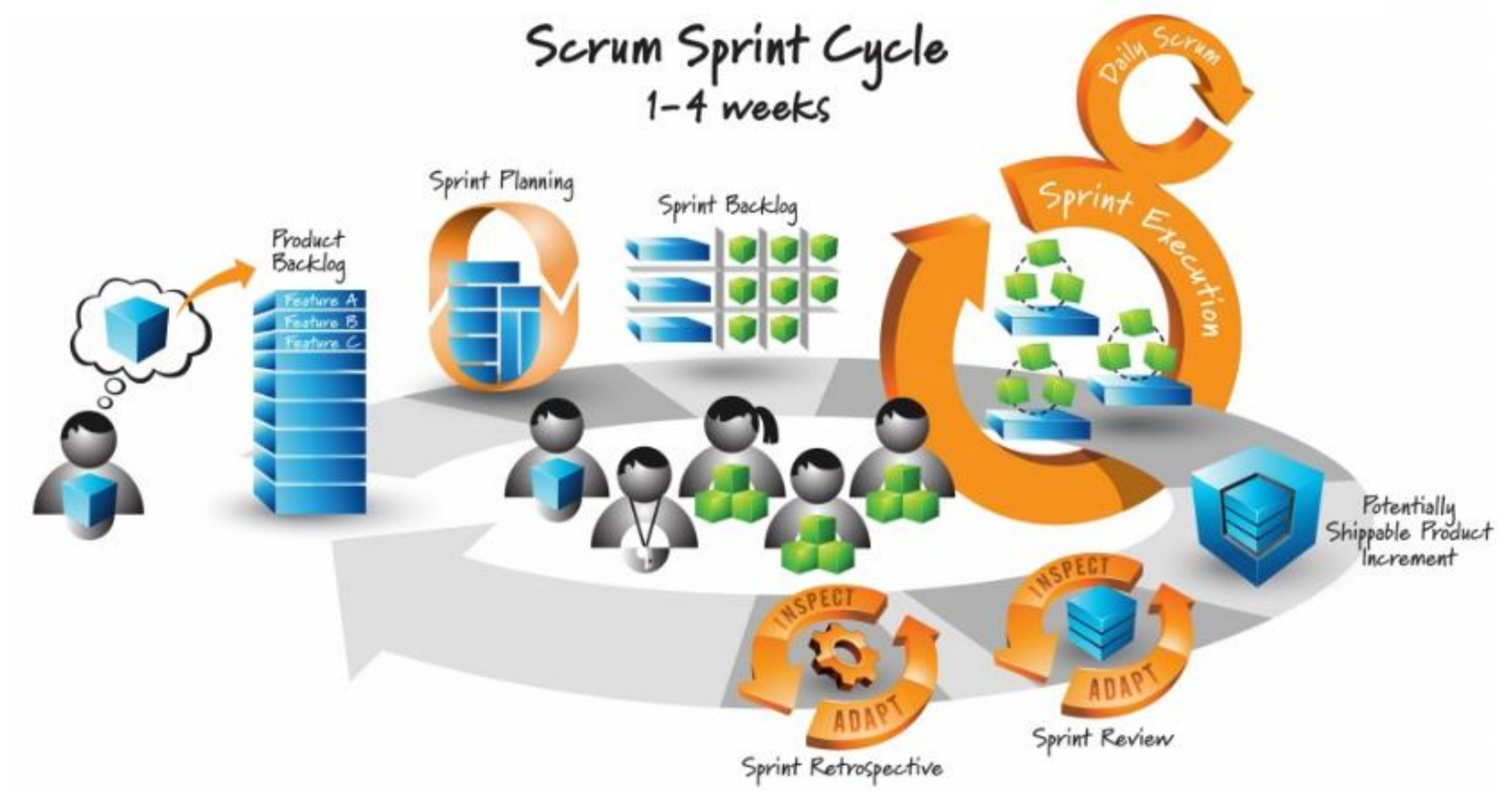
5 Activities

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

5 Values

- Courage
- Openness
- Focus
- Commitment
- Respect

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

CYBAGE
Delivering Value. Scientifically.

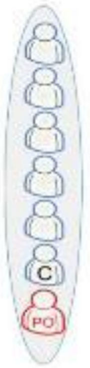
A road Ahead

Waterfall to Agile to Squads...

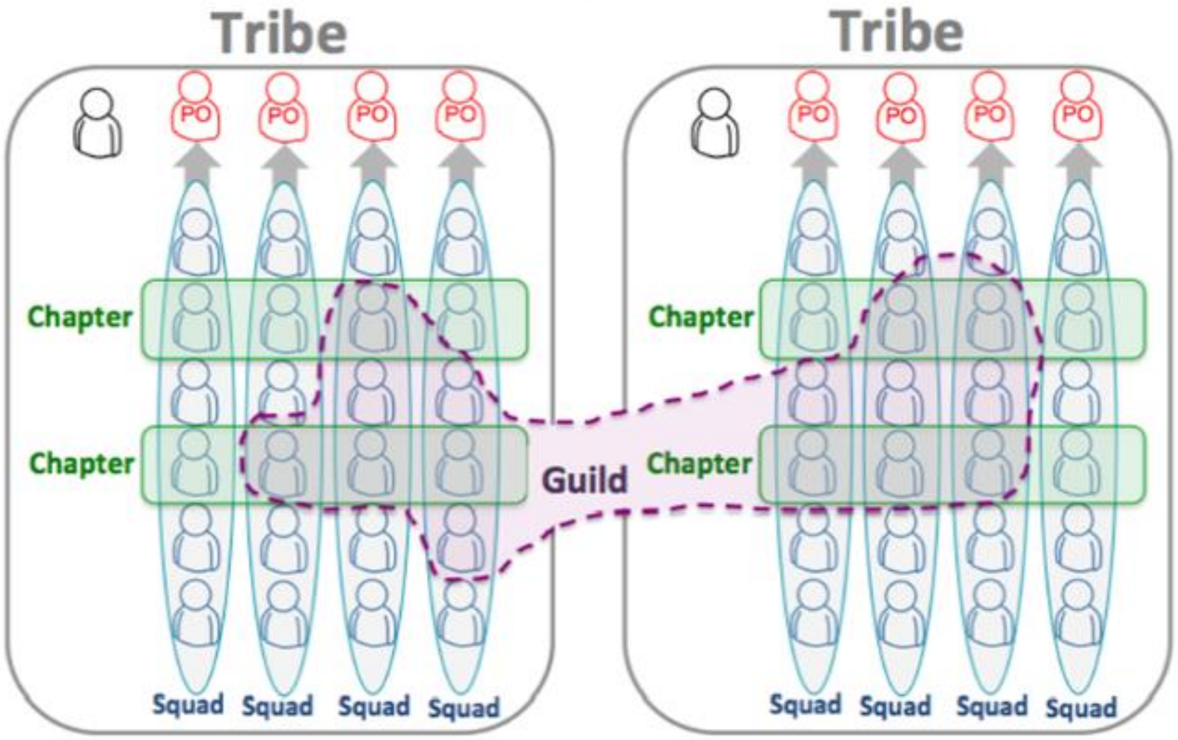
full-stack agile

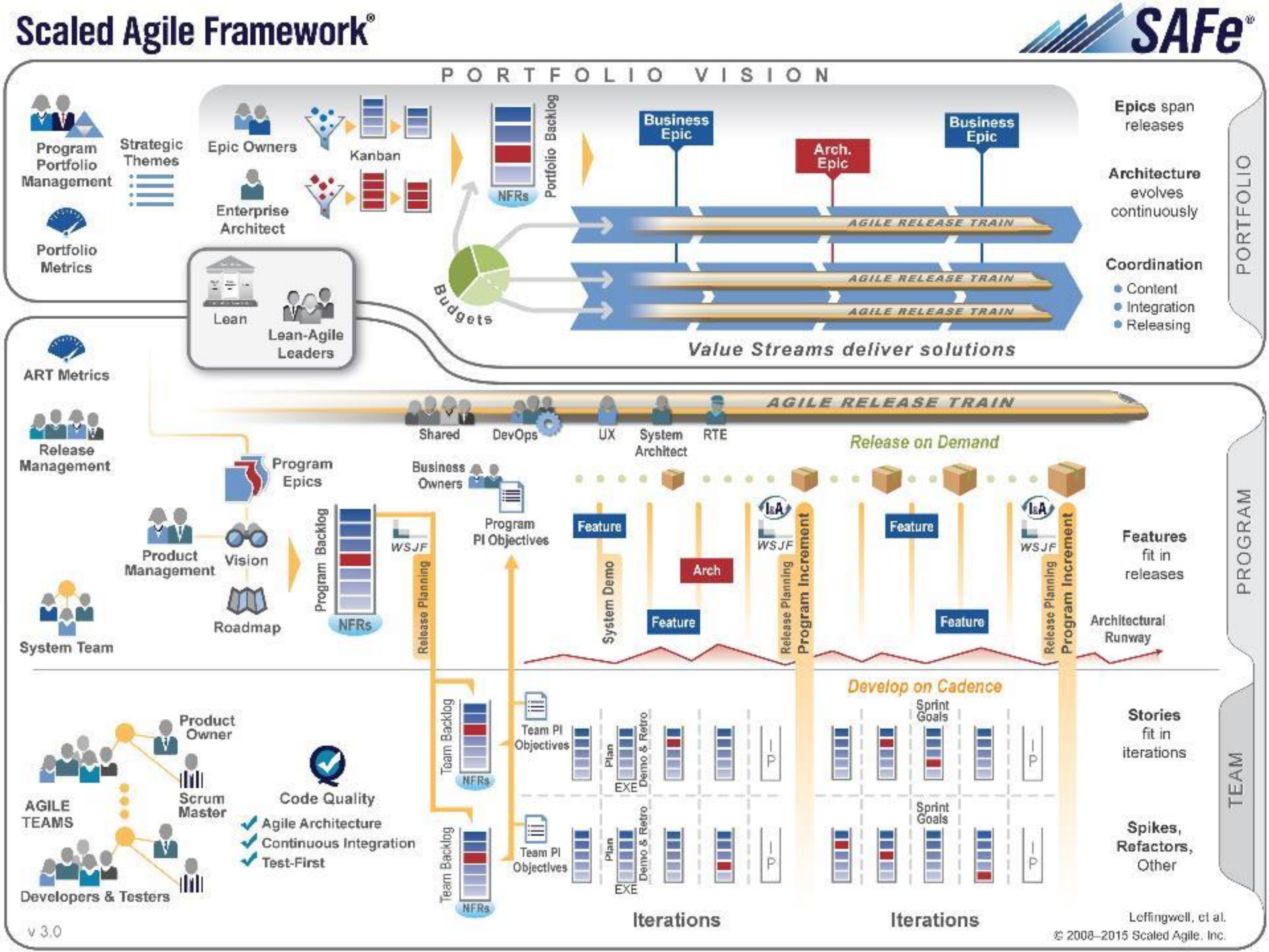
Mature Agile Organization Team Structure…

Squad



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

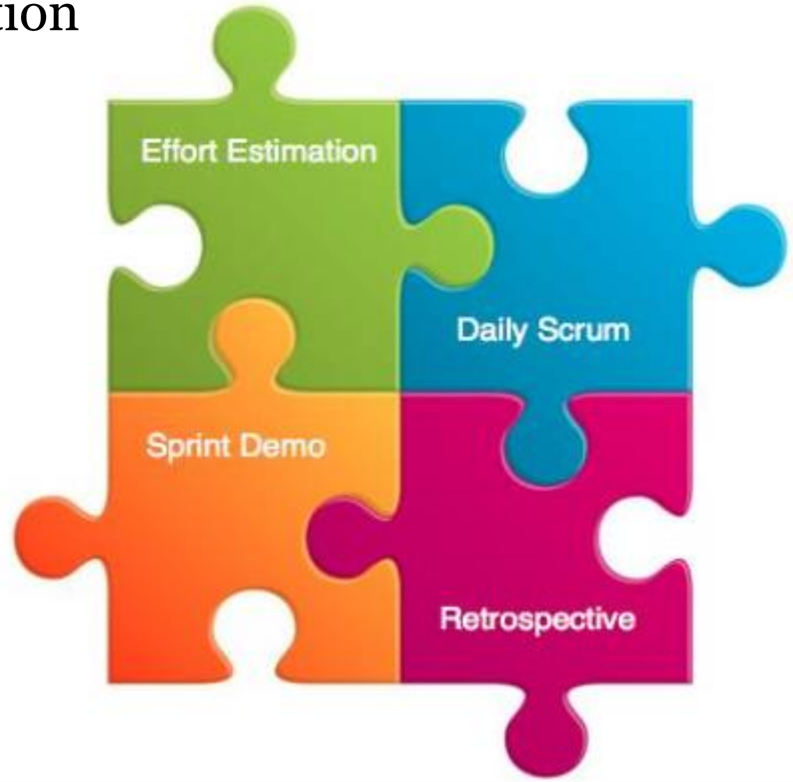



Scaled Agile Framework®
SAFe®
PORTFOLIO VISION
Program Portfolio Management
Strategic Themes
Portfolio Metrics
Epic Owners
Enterprise Architect
Kanban
Portfolio Backlog
NFRs
Business Epic
Arch. Epic
Business Epic
AGILE RELEASE TRAIN
AGILE RELEASE TRAIN
AGILE RELEASE TRAIN
Budgets
Lean
Lean-Agile Leaders
Value Streams deliver solutions
Epics span releases
Architecture evolves continuously
Coordination
• Content
• Integration
• Releasing
PORTFOLIO
ART Metrics
Release Management
System Team
AGILE RELEASE TRAIN
Shared
DevOps
UX
System Architect
RTE
Release on Demand
Program Epics
Product Management
Vision
Roadmap
Program Backlog
NFRs
WSJF
Release Planning
Business Owners
Program PI Objectives
Feature
System Demo
Arch
Feature
I&A
WSJF
Release Planning
Program Increment
Feature
Feature
I&A
WSJF
Release Planning
Program Increment
Architectural Runway
Features fit in releases
PROGRAM
Develop on Cadence
Product Owner
Scrum Master
AGILE TEAMS
Developers & Testers
Code Quality
✓ Agile Architecture
✓ Continuous Integration
✓ Test-First
Team Backlog
NFRs
Team PI Objectives
Plan
EXE
Demo & Retro
IP
Sprint Goals
Team Backlog
NFRs
Team PI Objectives
Plan
EXE
Demo & Retro
IP
Sprint Goals
Iterations
Iterations
Stories fit in iterations
Spikes, Refactors, Other
TEAM
v 3.0
Leffingwell, et al.
© 2008–2015 Scaled Agile, Inc.

Agile…

In agile all of its artefacts and activities are interconnected as lego pieces, failing to use any

of it will increase the risk of falling the whole structure on which the while agile foundation rest.



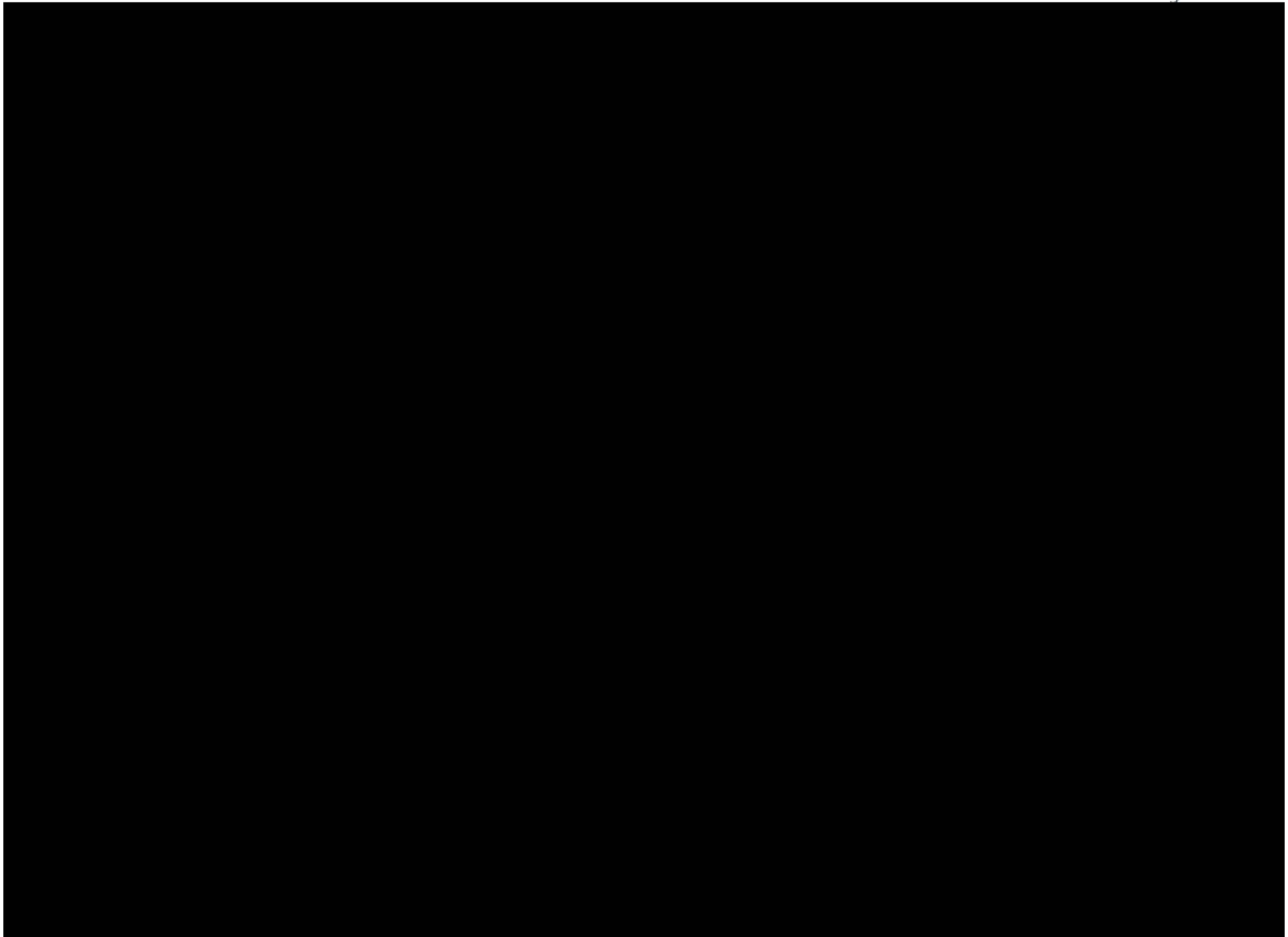
Agile…

- Detail sprint planning, effort estimation, planning poker, knowing team’s velocity, daily scrum are the activities which are interconnected to each other.
- Some team have assumption that they are doing agile because they have a **Product owner** who also happen to be a scrum master, have backlog and run sprint which have duration of 3-4 weeks, another **miss conception** is “**real purpose of agile to have no process overhead and meetings**”

Is it so easy?

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

CYBAGE
Delivering Value. Scientifically.



CYBAGE
Delivering Value. Scientifically.

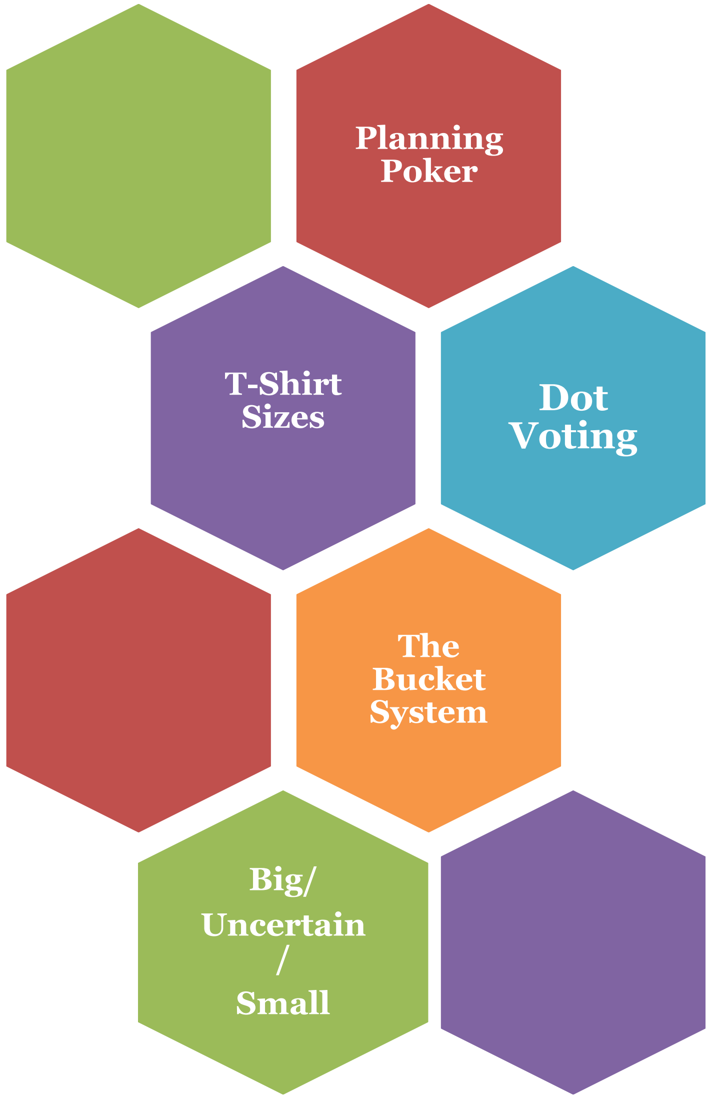
How was your journey towards being agile, share your thoughts….

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

CYBAGE
Delivering Value. Scientifically.

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!



Planning Poker

- Participants use specially-numbered playing cards to vote for an estimate of an item.
- Voting repeats with discussion until all votes are unanimous.
- There are lots of minor variations on Planning Poker.
- Good technique to estimate a very small number of items (2 to 10).

T-Shirt Sizes

- Items are categorized into t-shirt sizes: XS, S, M, L, XL.
- The sizes can, if needed, be given numerical values after the estimation is done.
- This is a very informal technique, and can be used quickly with a large number of items.
- Usually, the decisions about the size are based on open, collaborative discussion, possibly with the occasional vote to break a stalemate. There is a brief description of T-Shirt Sizes here.

The Bucket System

- Using the same sequence as Planning Poker, a group or a team estimate items by placing them in “buckets”.
- The Bucket System is a much faster Agile estimation technique than Planning Poker because there is a “divide-and-conquer” phase.
- The Bucket System can also be used with larger groups than Planning Poker and with very large numbers of items to be estimated (50 to 500).

Big/Uncertain/Small

- For super-fast Agile estimation, the items to be estimated are simply placed by the group in one of three categories:
  - big,
  - uncertain and
  - small.
- The group starts by discussing a few together, and then, like the Bucket System, uses divide-and-conquer to go through the rest of the items.

Dot Voting

- Dot voting is usually considered a decision-making tool, not an Agile estimation technique.
- However, for estimating small numbers of items, dot voting can be a super-simple and effective technique.
- Each person gets a small number of “dots” and uses them as votes to indicate the size of an item; more dots means bigger.

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.