

An Introduction to Scrum

- 18th January 2020

Welcome to the session



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



Any Questions

An Introduction to Scrum

18th January 2020

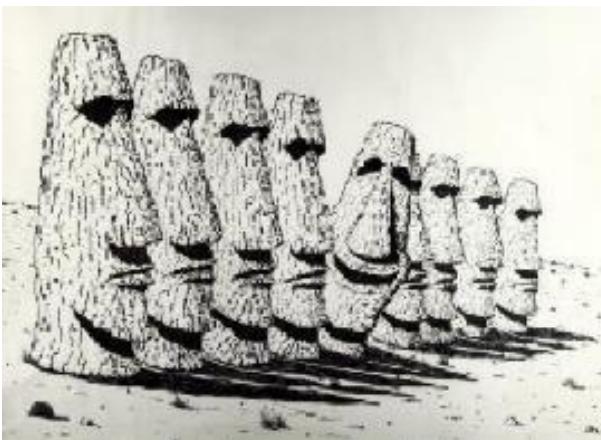
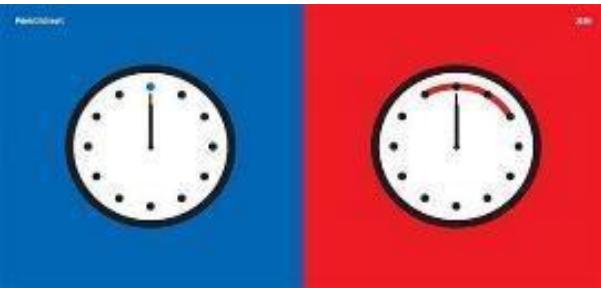
Presented by

Uday Uttarwar

(CSP, ICP-ACC, Safe Agilist, CSM)

Total 19+ years of IT Experience

Our Expectations

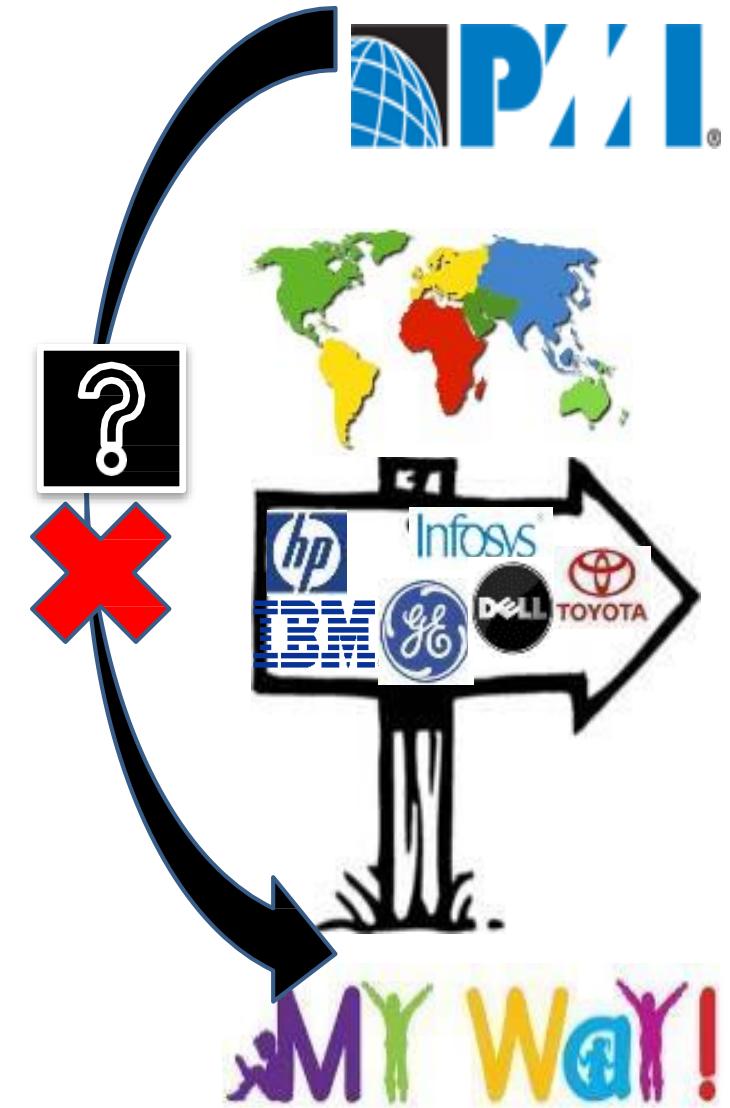


Punctuality: Start on time, end on time is the best policy. Concentration starts dropping after sunset.

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...

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]



ACTIVITY

Being AGILE... Doing Agile: □

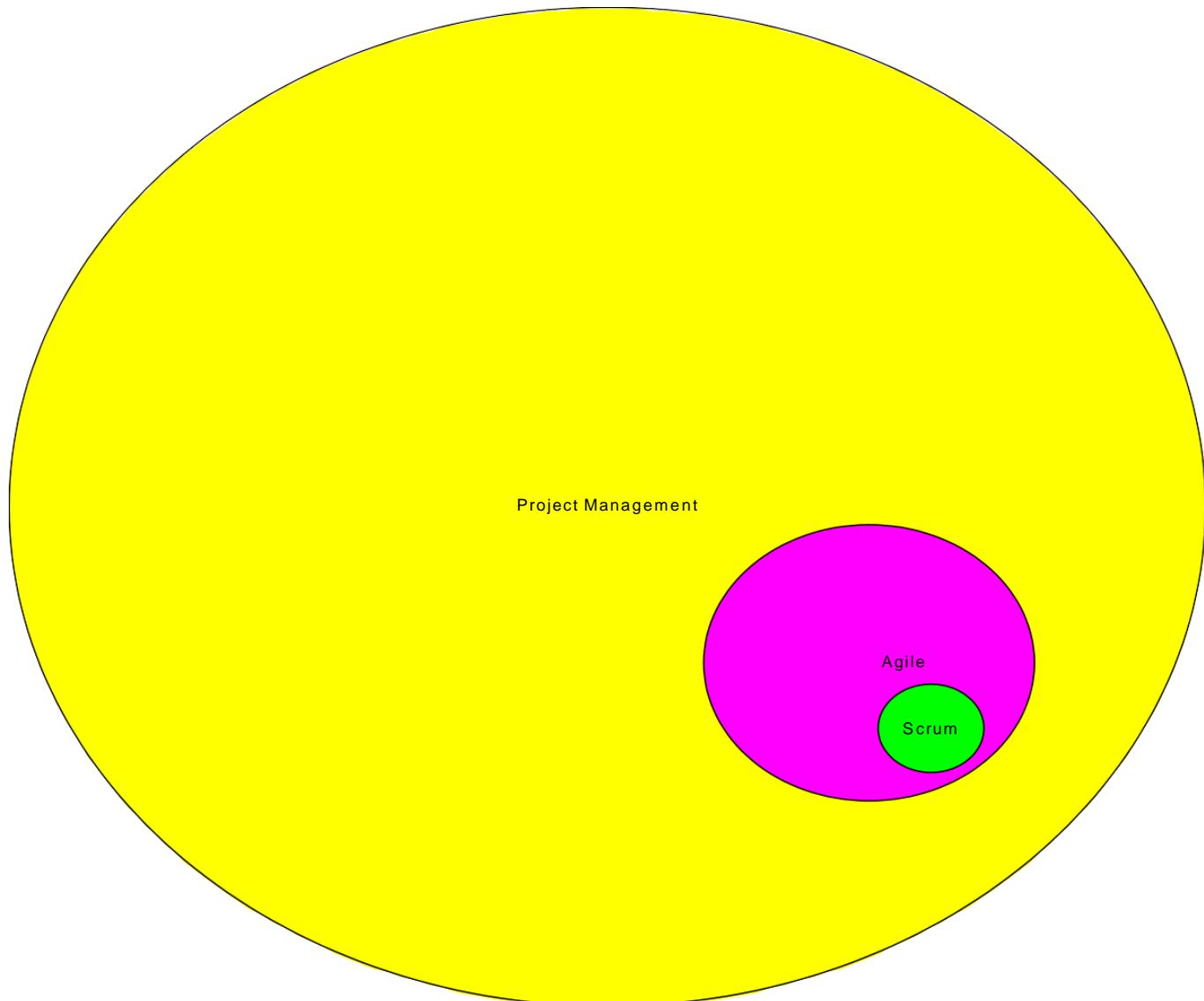
Sail Together or Sink Together



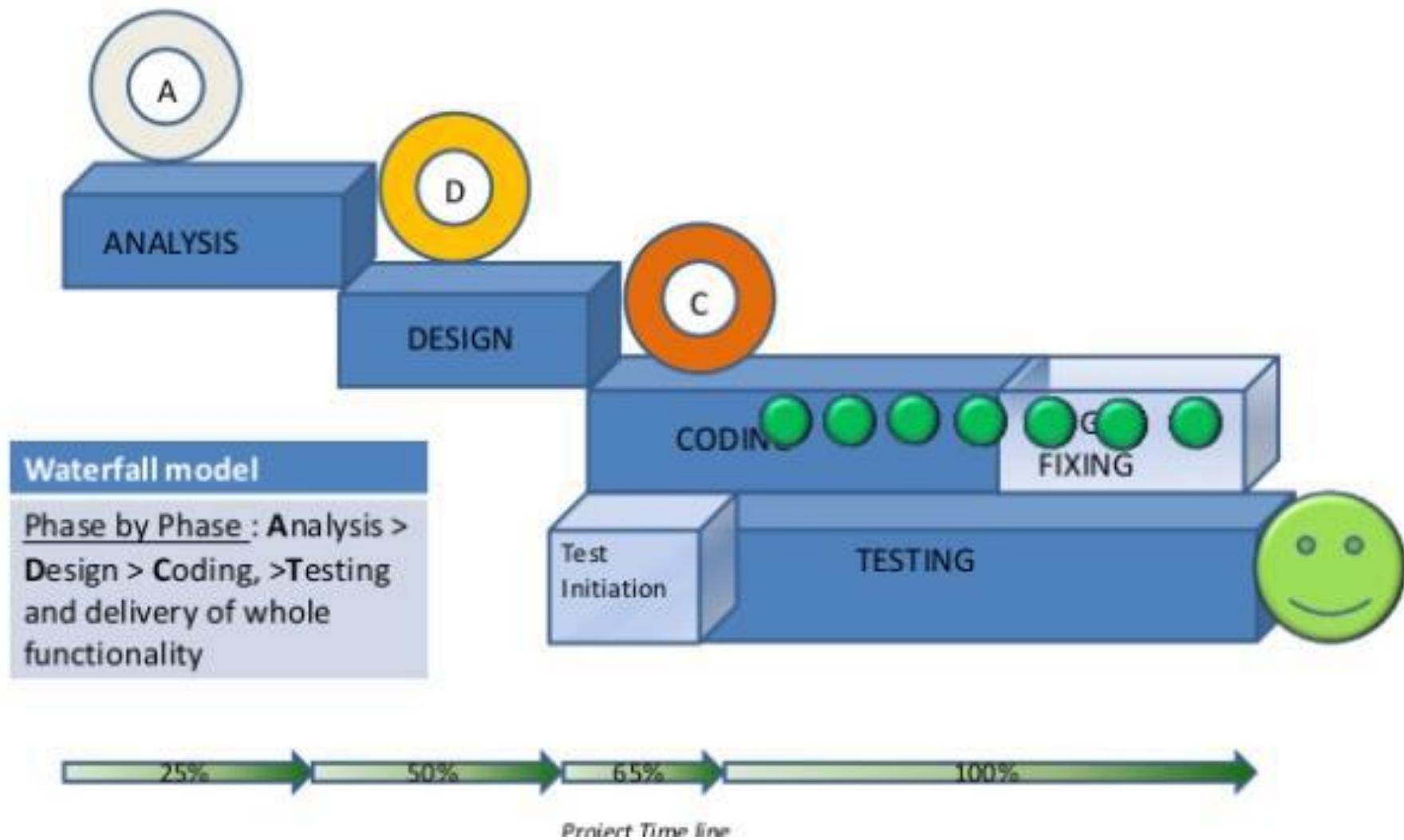
There are no winners to arguments in a relationship. Together you win and sail or together you fail and sink.
That is why it is called a relationSHIP.

A joint accountability to make it SUCCESS

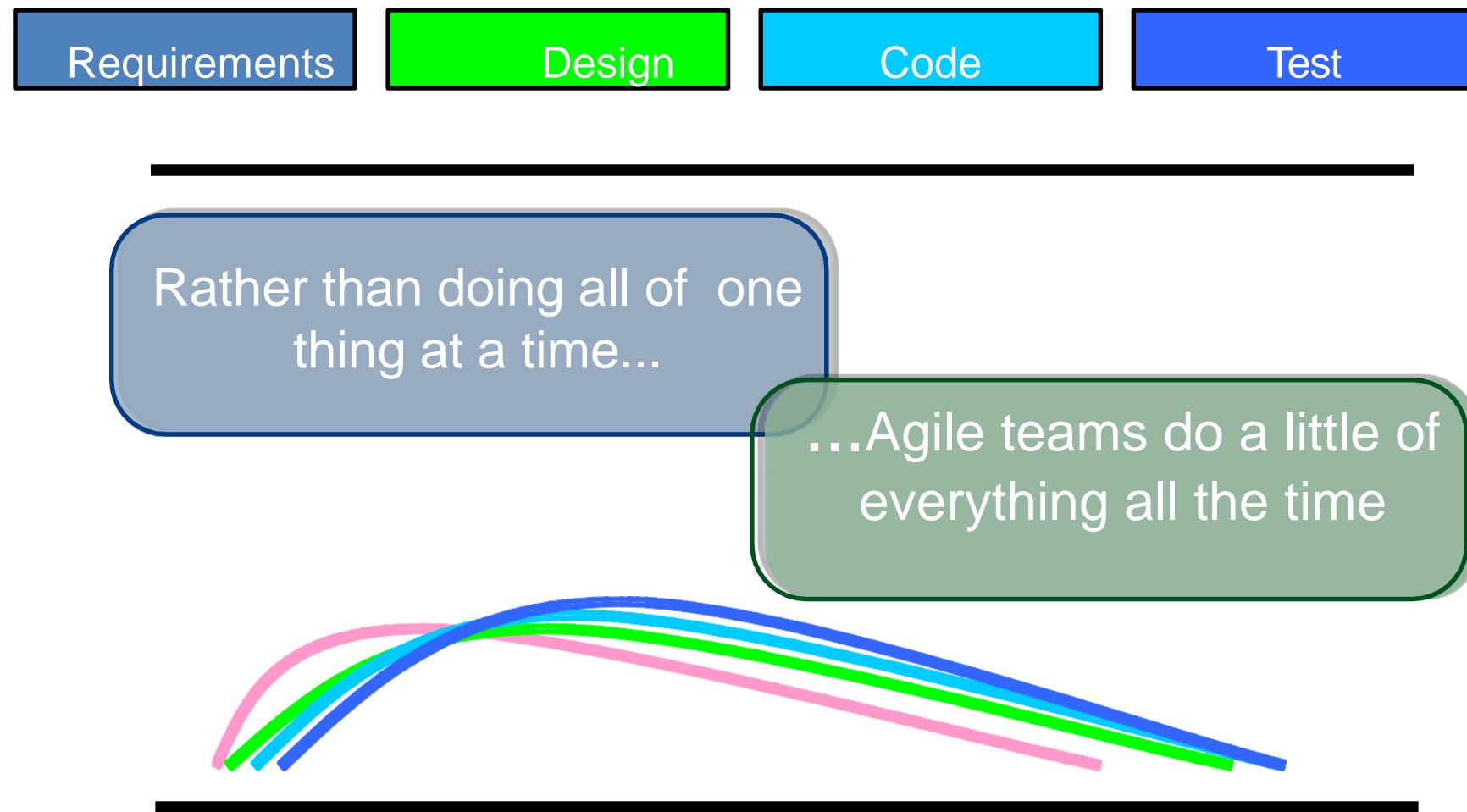
Management Sea



WATERFALL Model

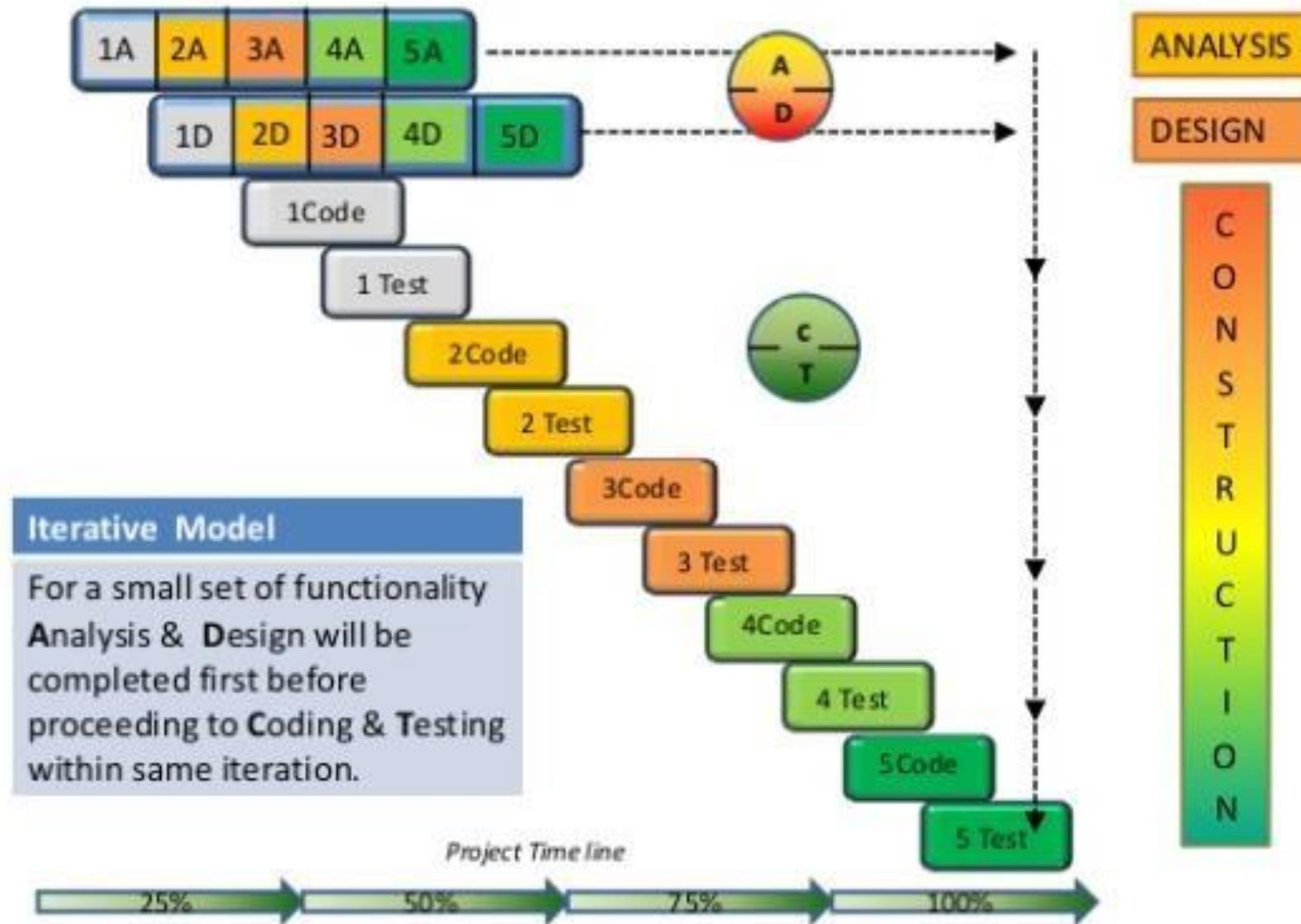


Sequential vs. overlapping development

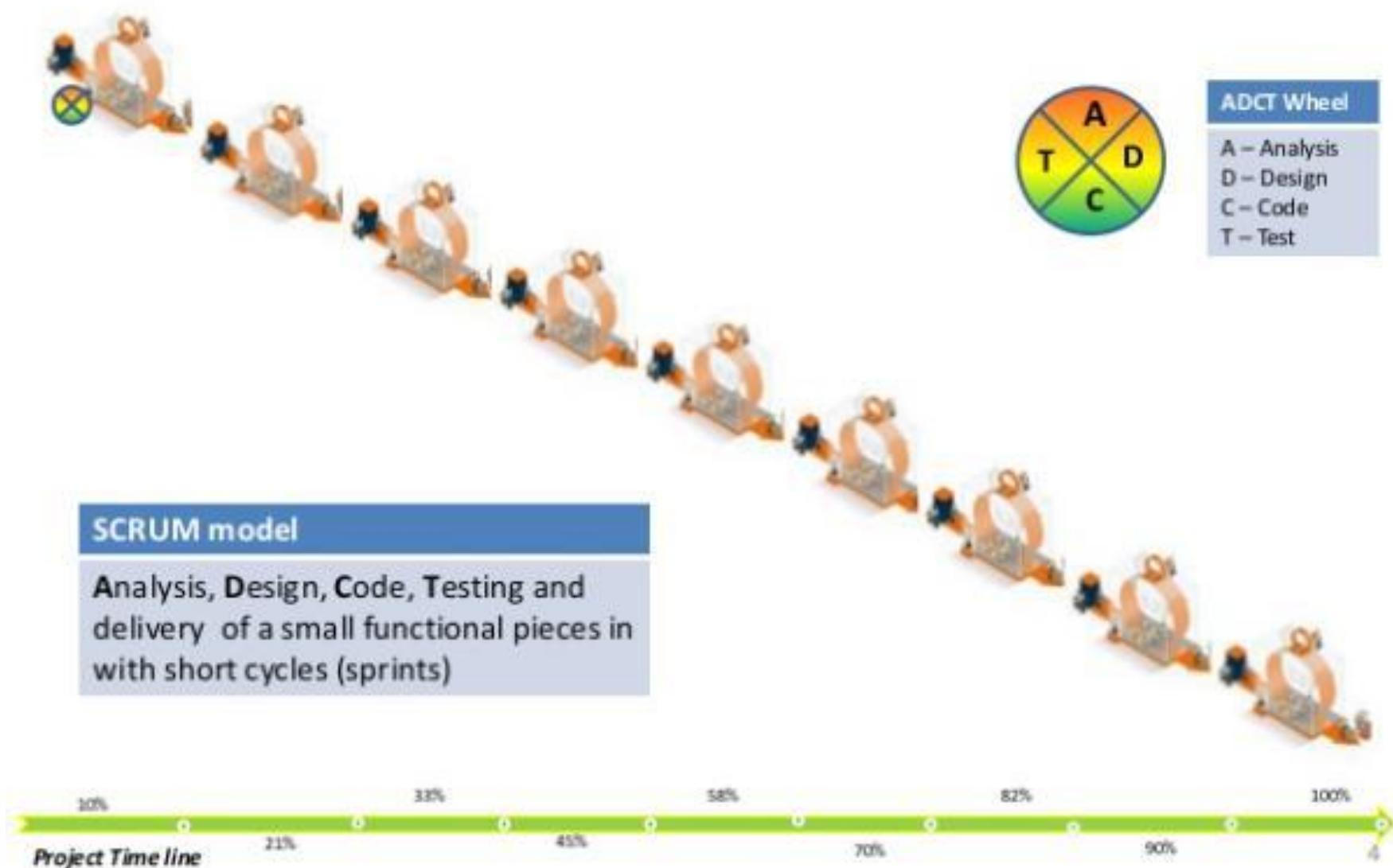


Source: "The New Product Development Game" by Takeuchi and Nonaka. *Harvard Business Review*, January 1986.

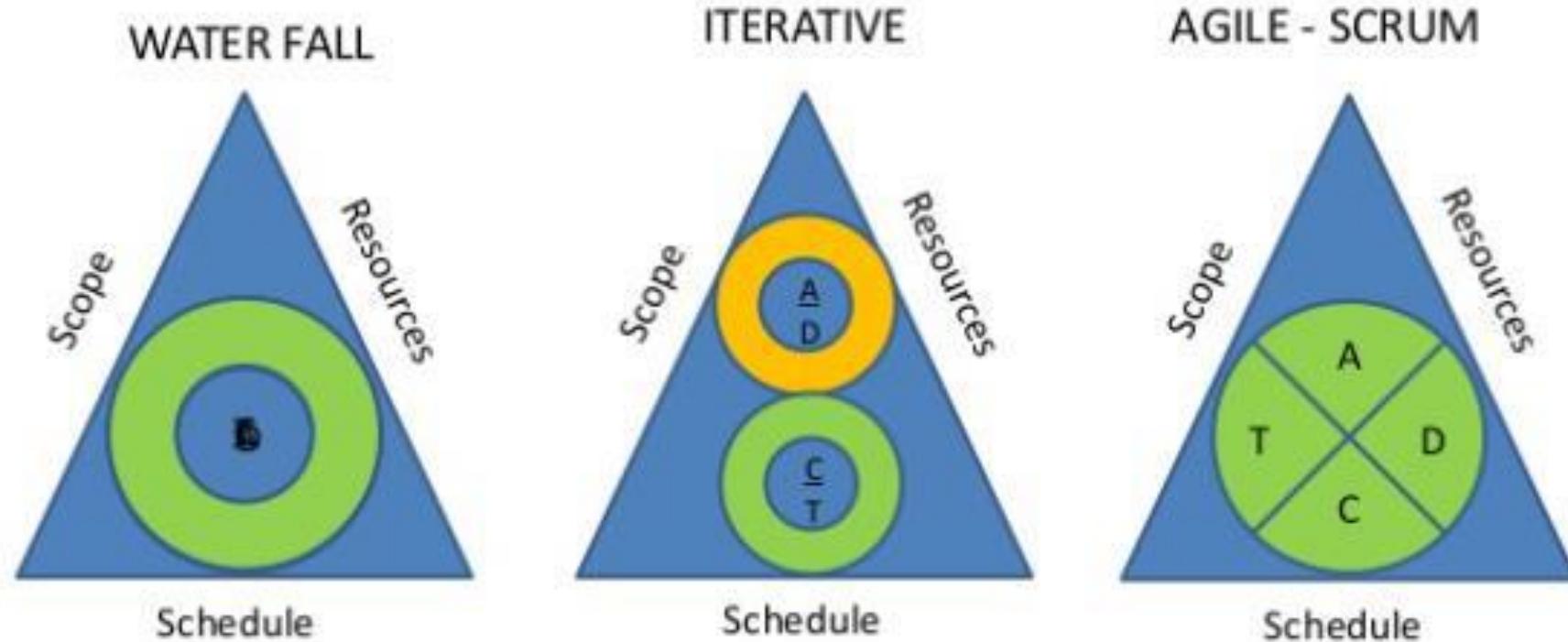
ITERATIVE Model



Agile- SCRUM Model



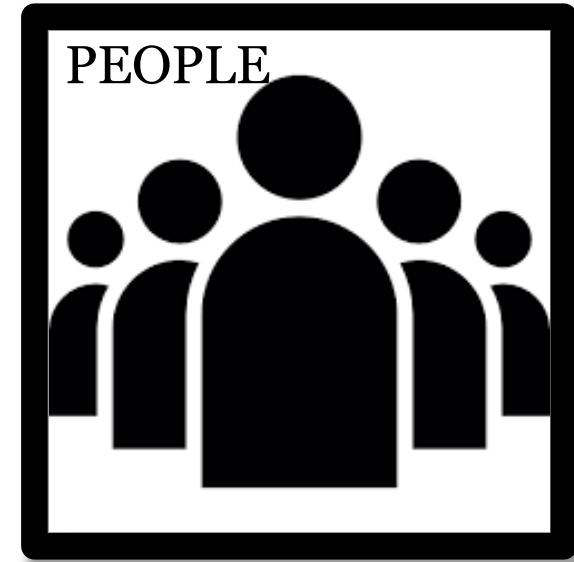
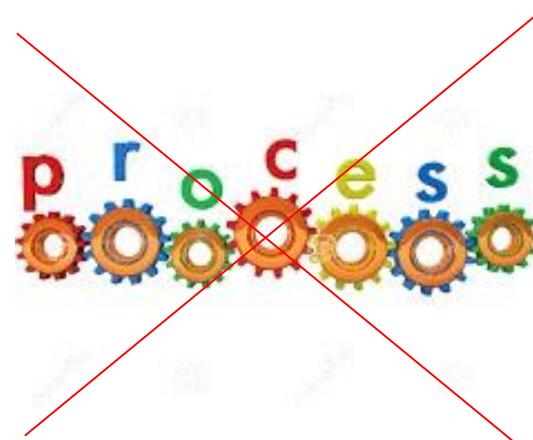
Process vs Project Triangle



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

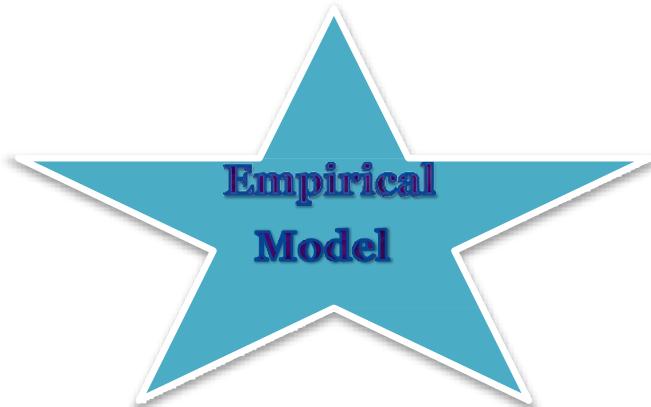


The problem



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

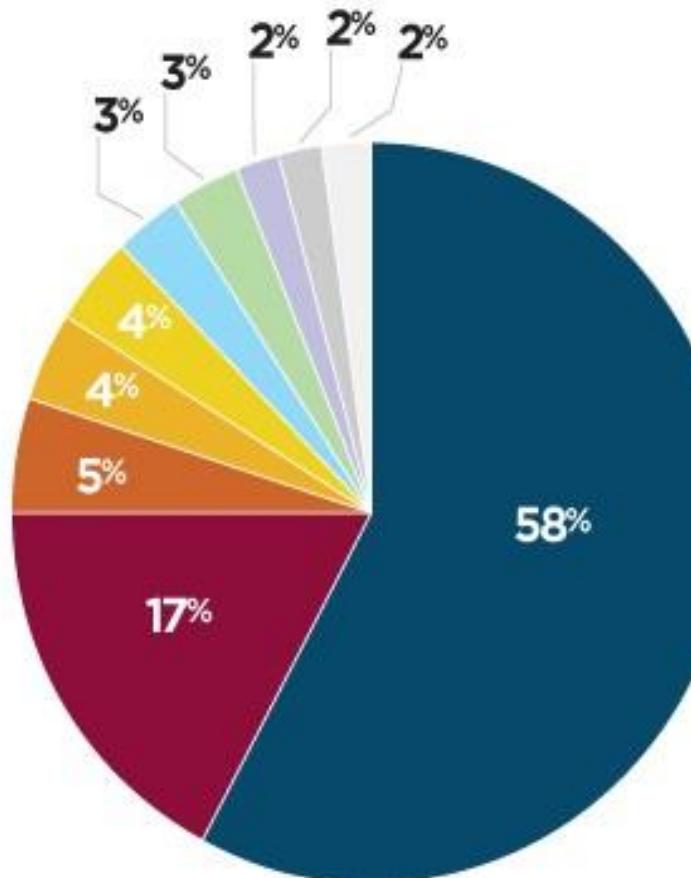
- Ken Schwaber



Agile Methods Contribution



AGILE METHODS & PRACTICES

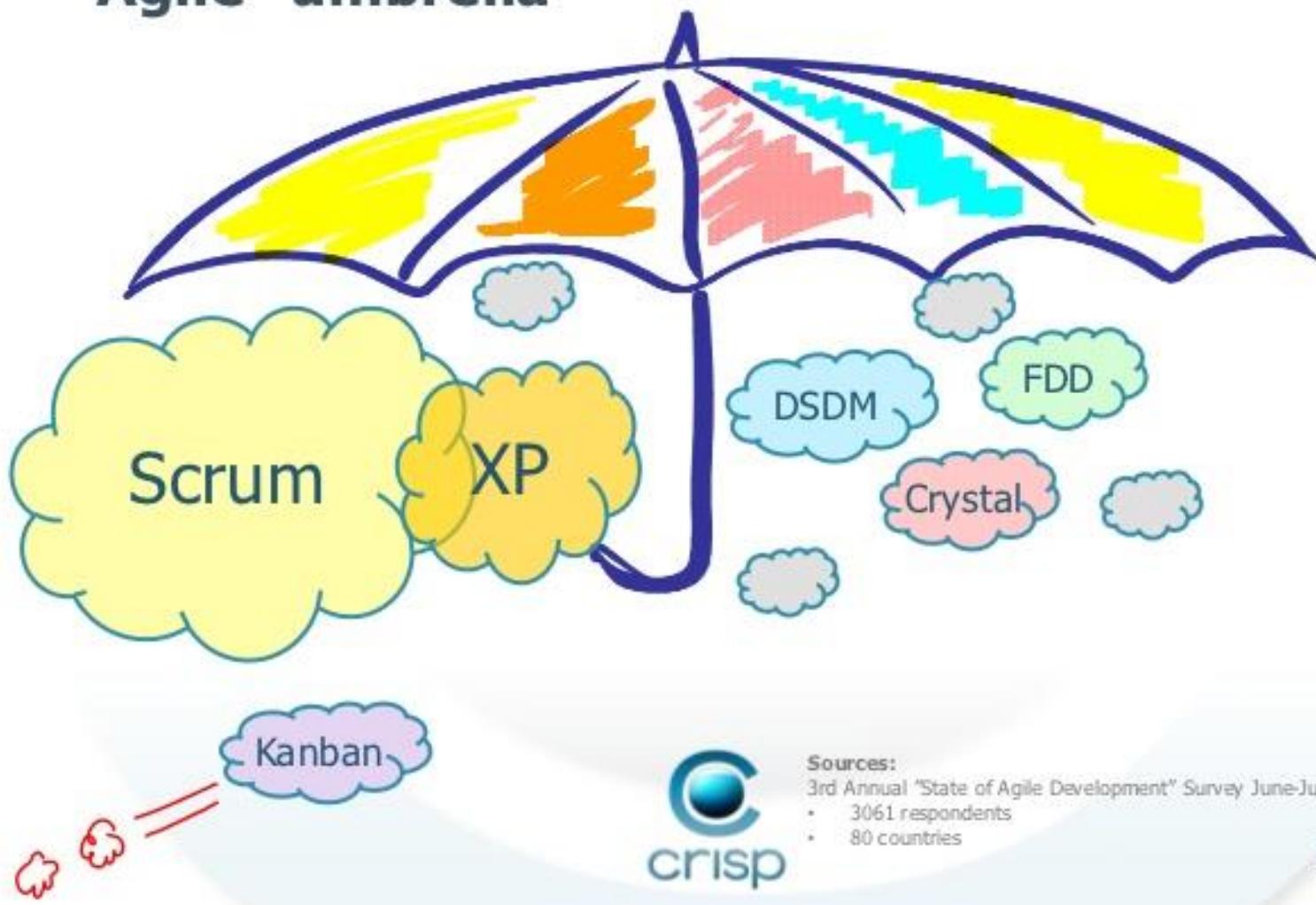


AGILE METHODOLOGY MOST CLOSELY FOLLOWED

Scrum or Scrum variants were by far the most common agile methodologies employed.

- SCRUM
- SCRUM/XP HYBRID
- CUSTOM HYBRID
- OTHER
- EXTREME PROGRAMMING (XP)
- DON'T KNOW
- SCRUMBAN
- LEAN
- FEATURE DRIVEN DEVELOPMENT (FDD)
- AGILE UP

Agile "umbrella"



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 Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas

The Agile Manifesto—a statement of values

Individuals and interactions

over

Process and tools

Working software

over

Comprehensive documentation

Customer collaboration

over

Contract negotiation

Responding to change

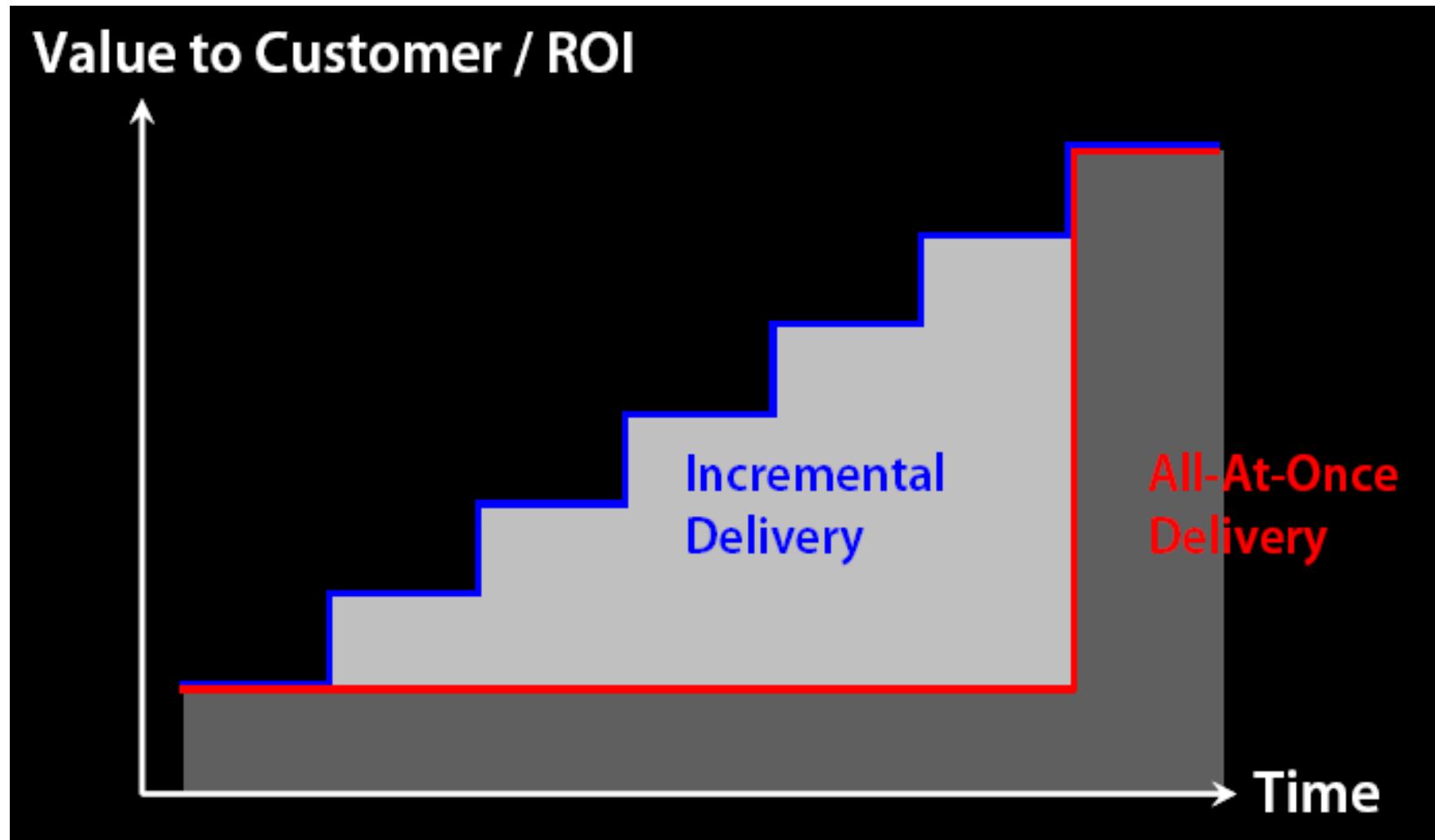
over

Following a plan

Principles behind the Agile Manifesto

- 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.

Value Realization- ROI: Waterfall Vs. Agile



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.

Agile Benefits



Improved
Customer
Satisfaction

Productivity
increase

Higher
Quality

Team moral is
high and
enhanced work
satisfaction

Early
Surprises

"Sh*t Bad Scrum Masters Say"

Summary

We have discussed the following

- Project Management
- Why Agile
- Benefits of agile
- Traditional Vs Agile

Quiz

1. In Project Life Cycle, All of the following is true except

 - A. Project Phases are typically completed sequentially
 - B. Project Phases can overlap in some situations
 - C. Any project phase can be skipped at any point in time
 - D. All phases are mandatory in a successful project completion

Quiz

2. Project and operation is one and the same

- A. True
- B. False

Quiz

3. A balanced matrix structure recognizes the need of the project manager
- 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

- 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

7. Who organizes the project charter

- A. End User
- B. The Project Sponsor
- C. The client
- D. Senior manager

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

SCRUM Project Management

Agile Methodologies



MODULE - 2

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.

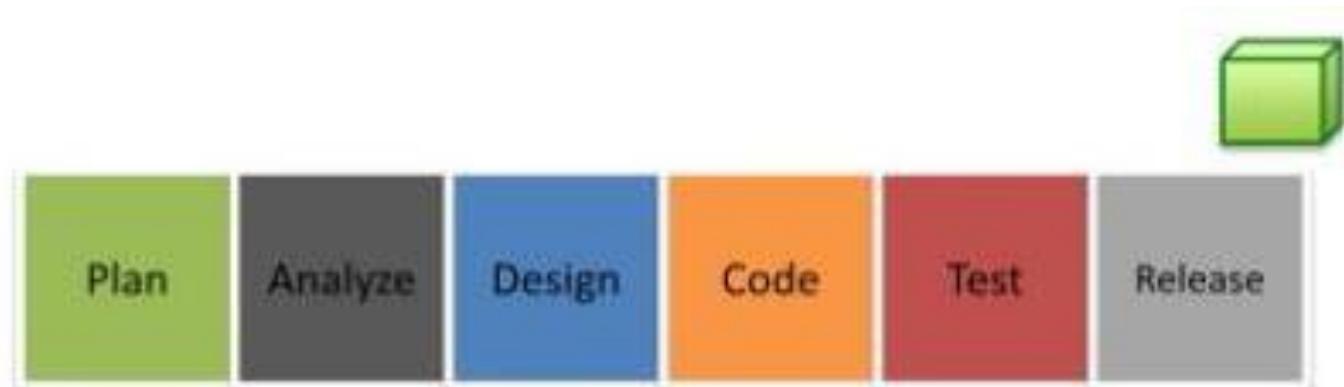
SCRUM

- ❖ 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]

Defined Vs. Empirical

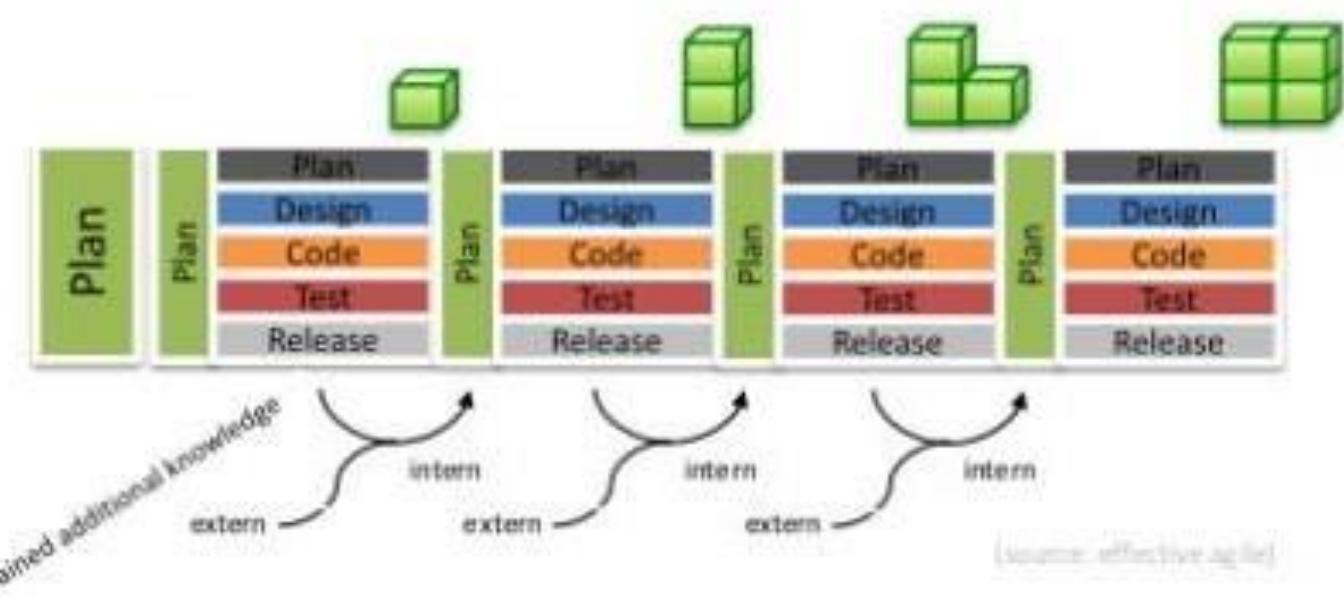
Waterfall (Defined)

Plan for the entire project up-front



Scrum (Empirical)

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



[Source: effectiveagile.com]

Empirical Process Control

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

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



- 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 Quiz

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 same as project manager

No

Is the sprint and iteration are the same

Yes it is a time boxed event of one month or less

Agile- A Mindset...

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

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

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 continuous 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

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

RICH

3 Roles

- Development Team
- Product Owner
- ScrumMaster

5 Activities

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

3 Artifacts

- Product Backlog
- Sprint Backlog
- Increment

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)

User story Examples:

1. Pushing Data to AWS from various input sources(on premise, other cloud, mobile)
2. Giving Access to AWS bucket
3. Sending notification- SNS and SES

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 **level of detail** of each.

Use cases

- 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.

User stories

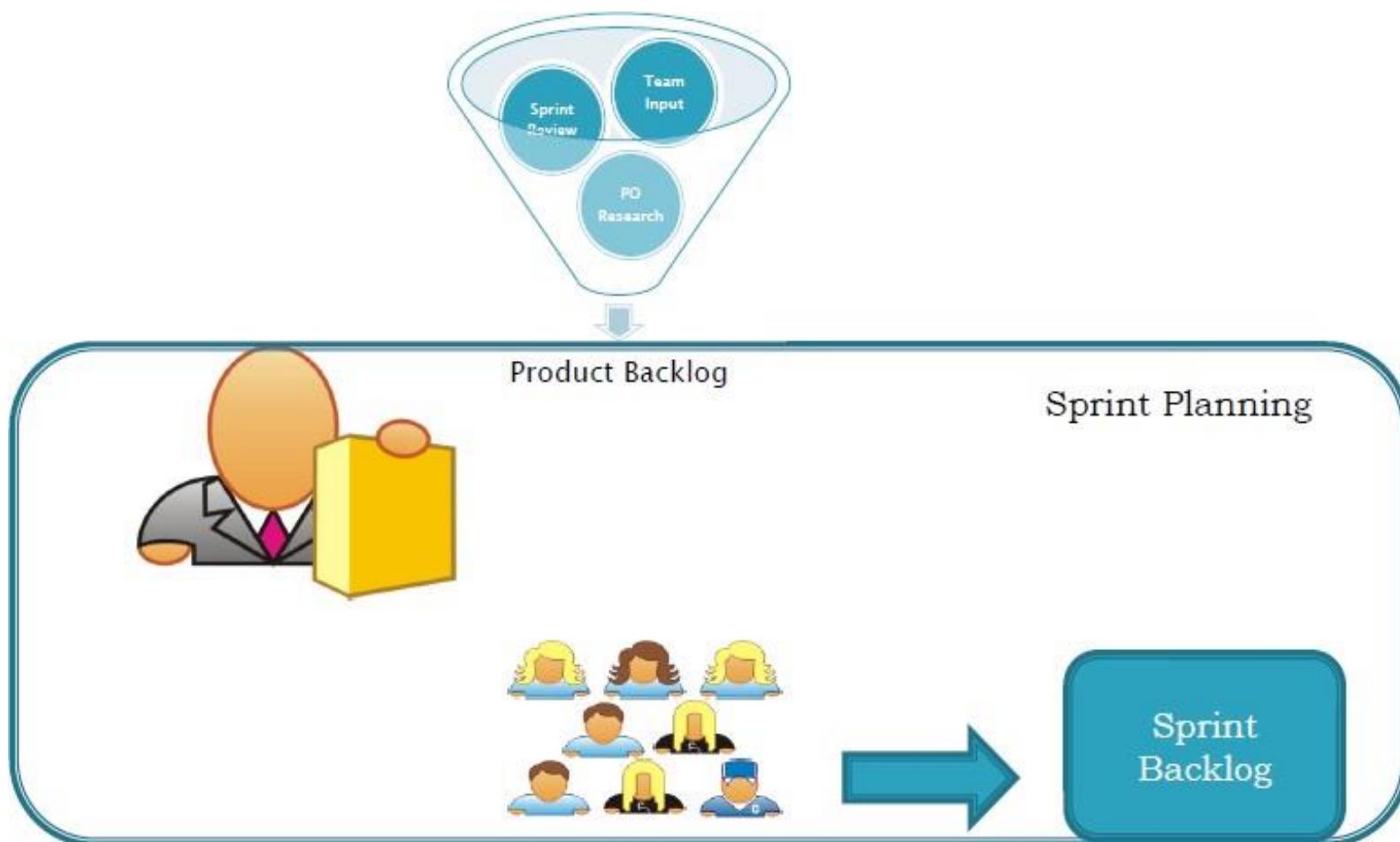
- are short and
- briefly state
 - the type of **user**,
 - the **feature** needed, and
 - the **benefit**.

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.

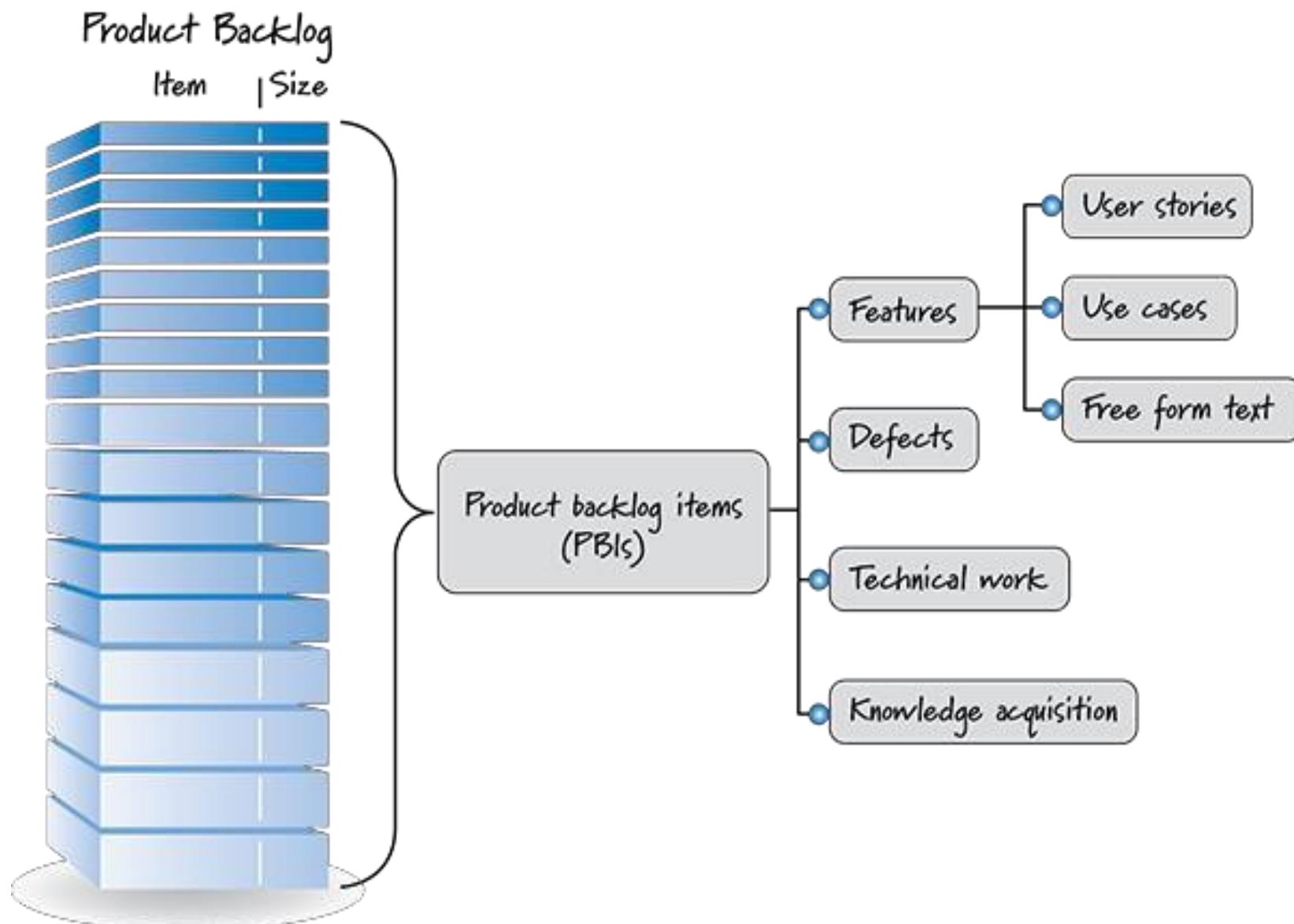
Product Backlog



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

Product Backlog



DEEP

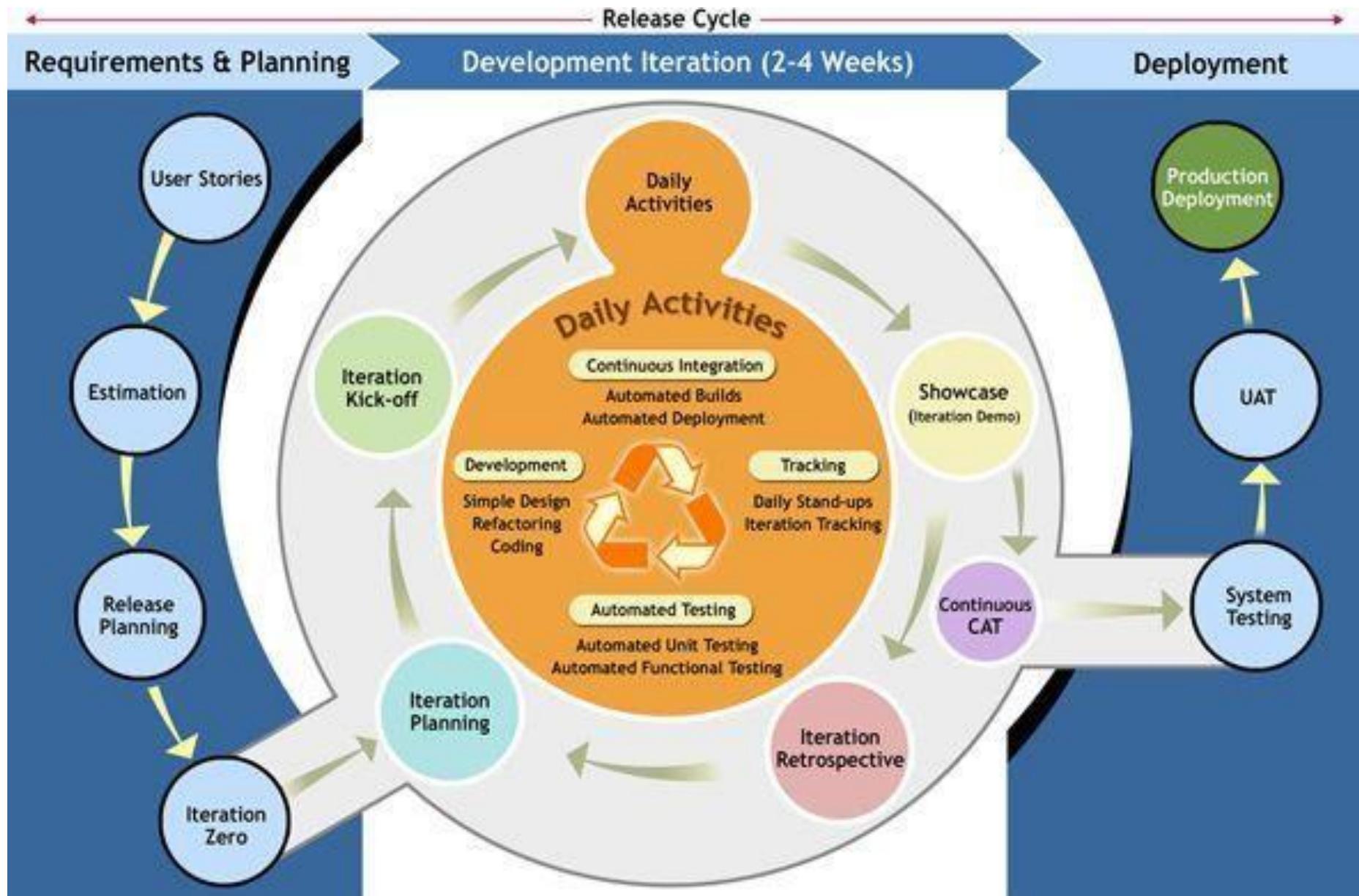
Definition of “Ready”- DoR

- 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

Agile Release Cycle



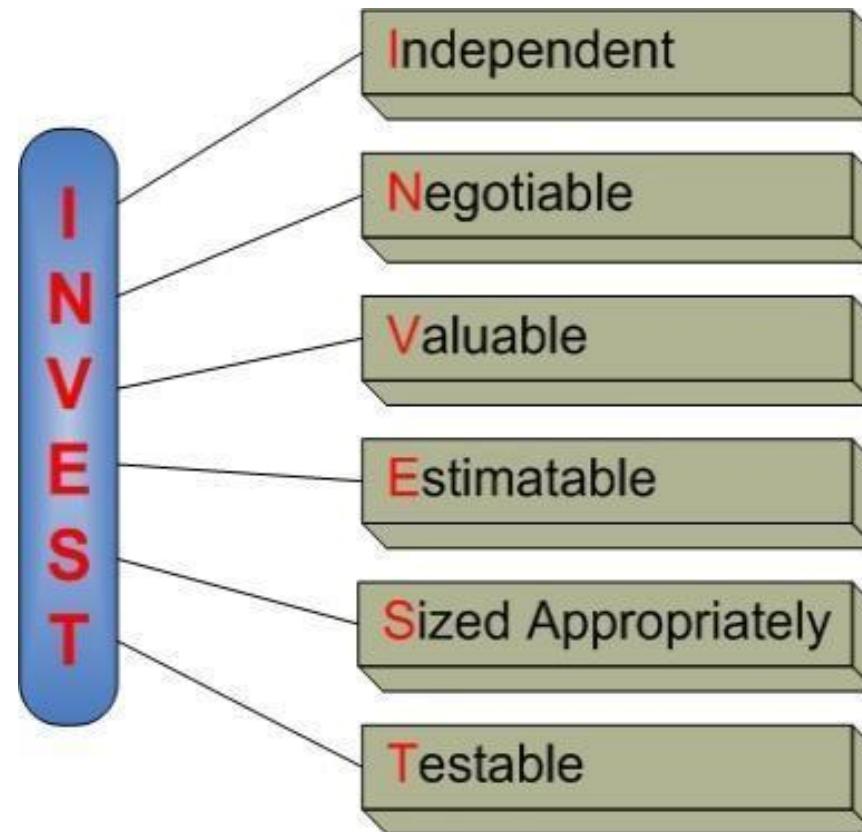
Sprint Backlog

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

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".

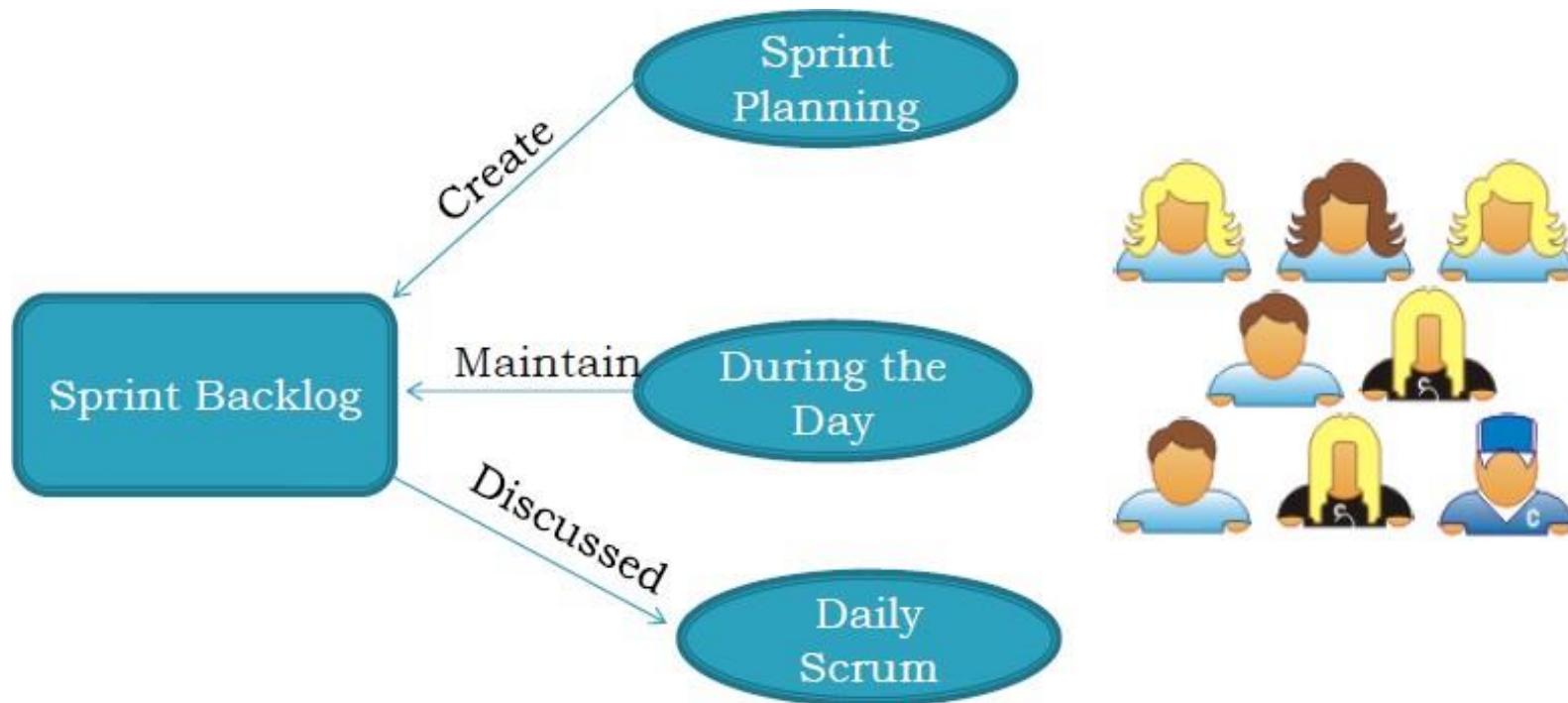
Story Points - Estimating the Size

➤ 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

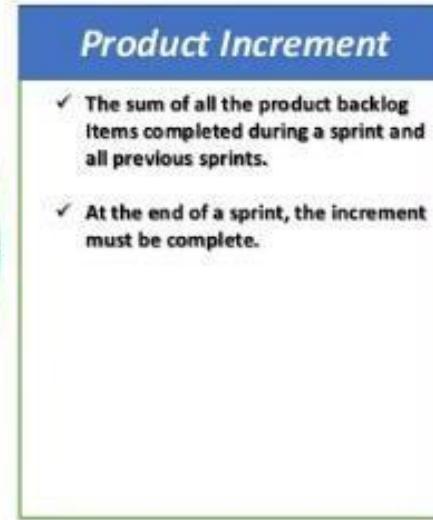
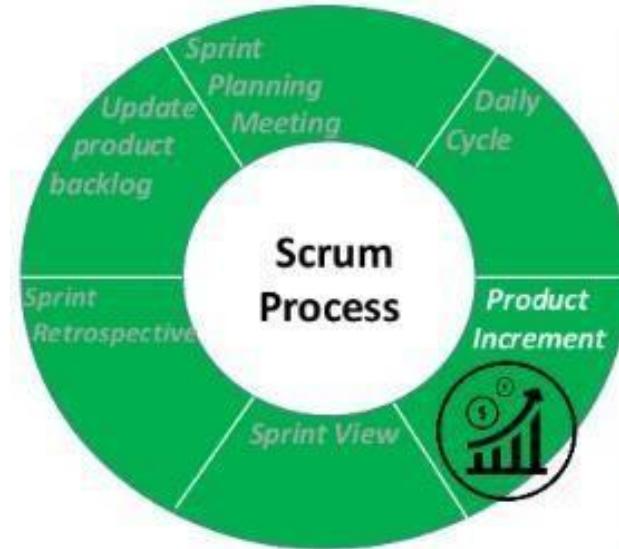


Sprint Backlog



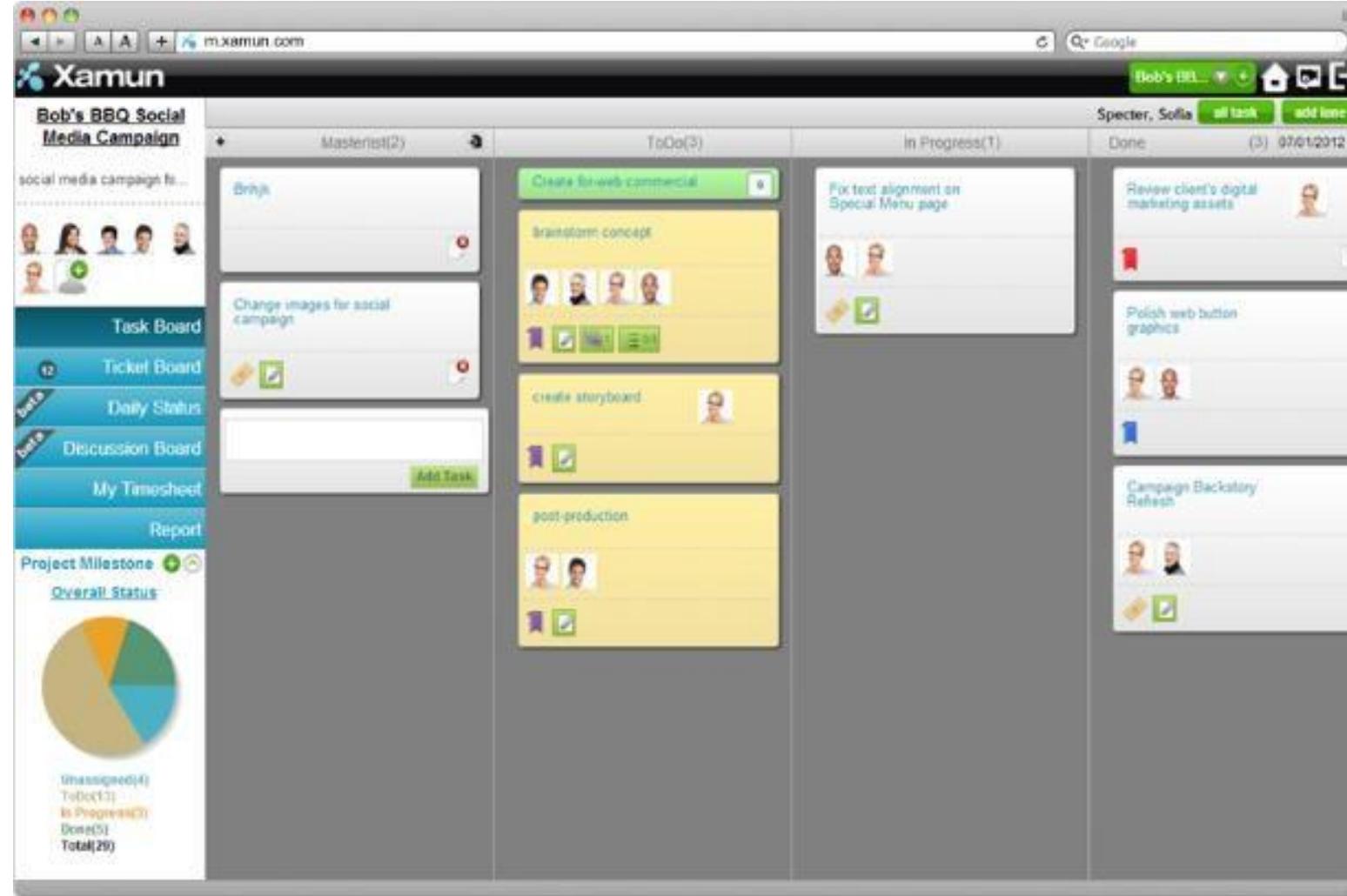
Increment

Increment



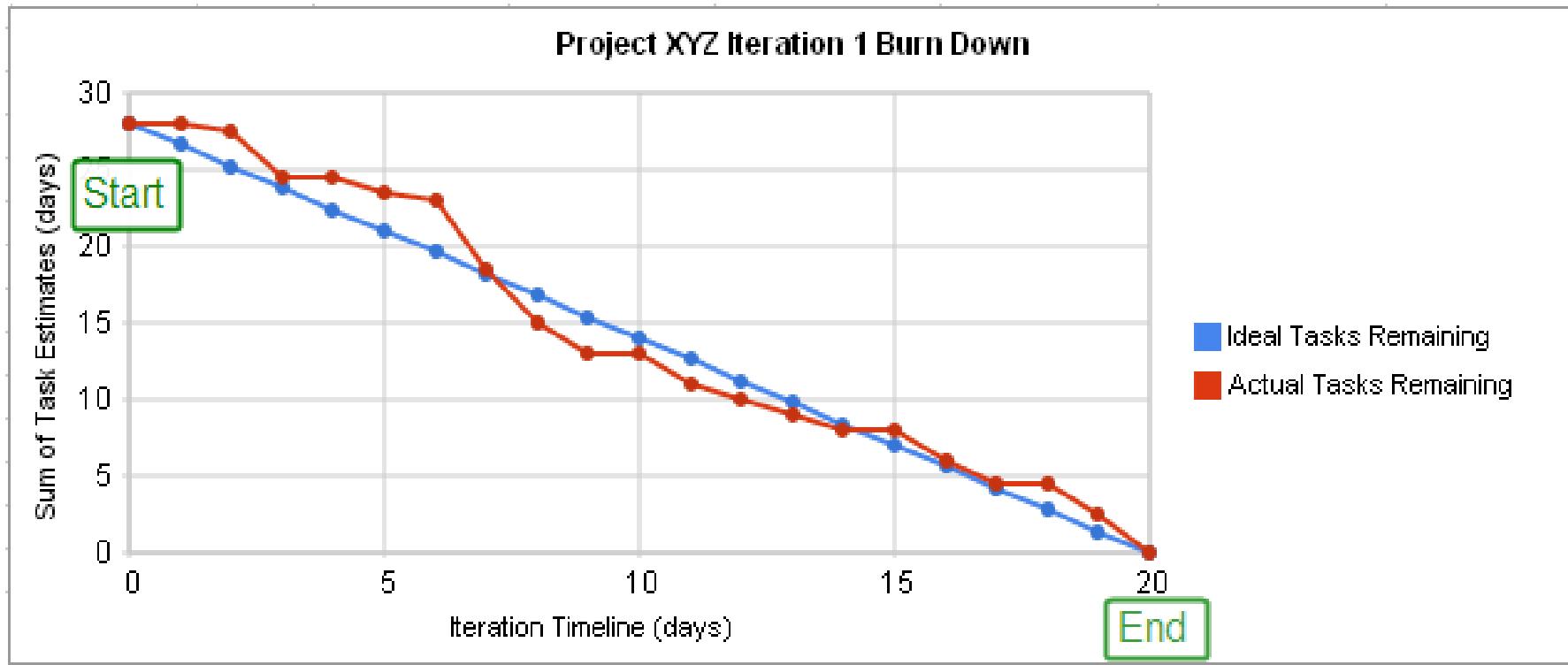
- Increment is the sum of all Product Backlog items completed during a sprint and all previous sprints
- At the end of a Sprint, the new Increment should “Done”
- It must be useable condition (Potentially Shippable Product)
- Release increments early and frequently Vs deliver the finished product in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

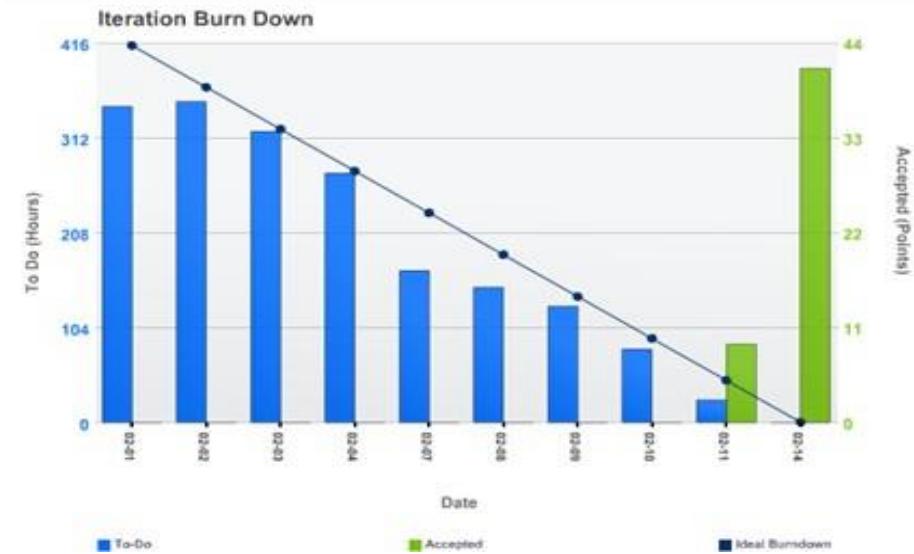
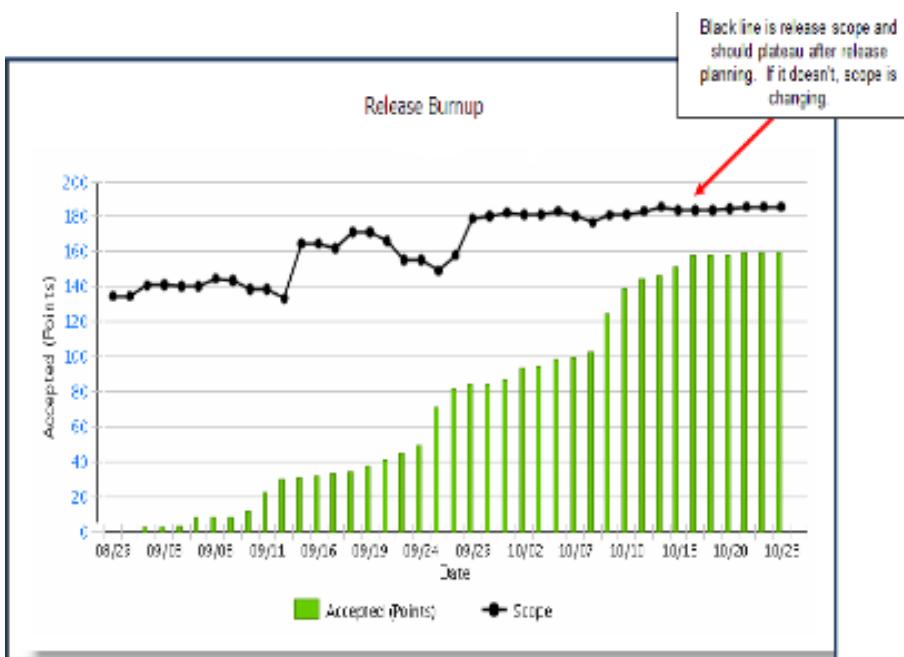
Increments – Burn Down Charts



- Updated daily, usually during the dailyscrum
- 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

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

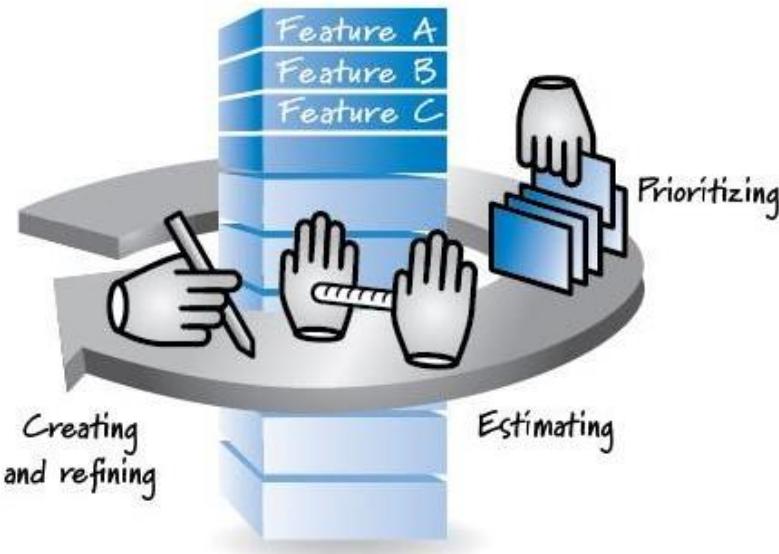
5 Values

- Courage
- Openness
- Focus
- Commitment
- Respect

Product Backlog Refinement

Product Backlog Refinement

Product Backlog refinement



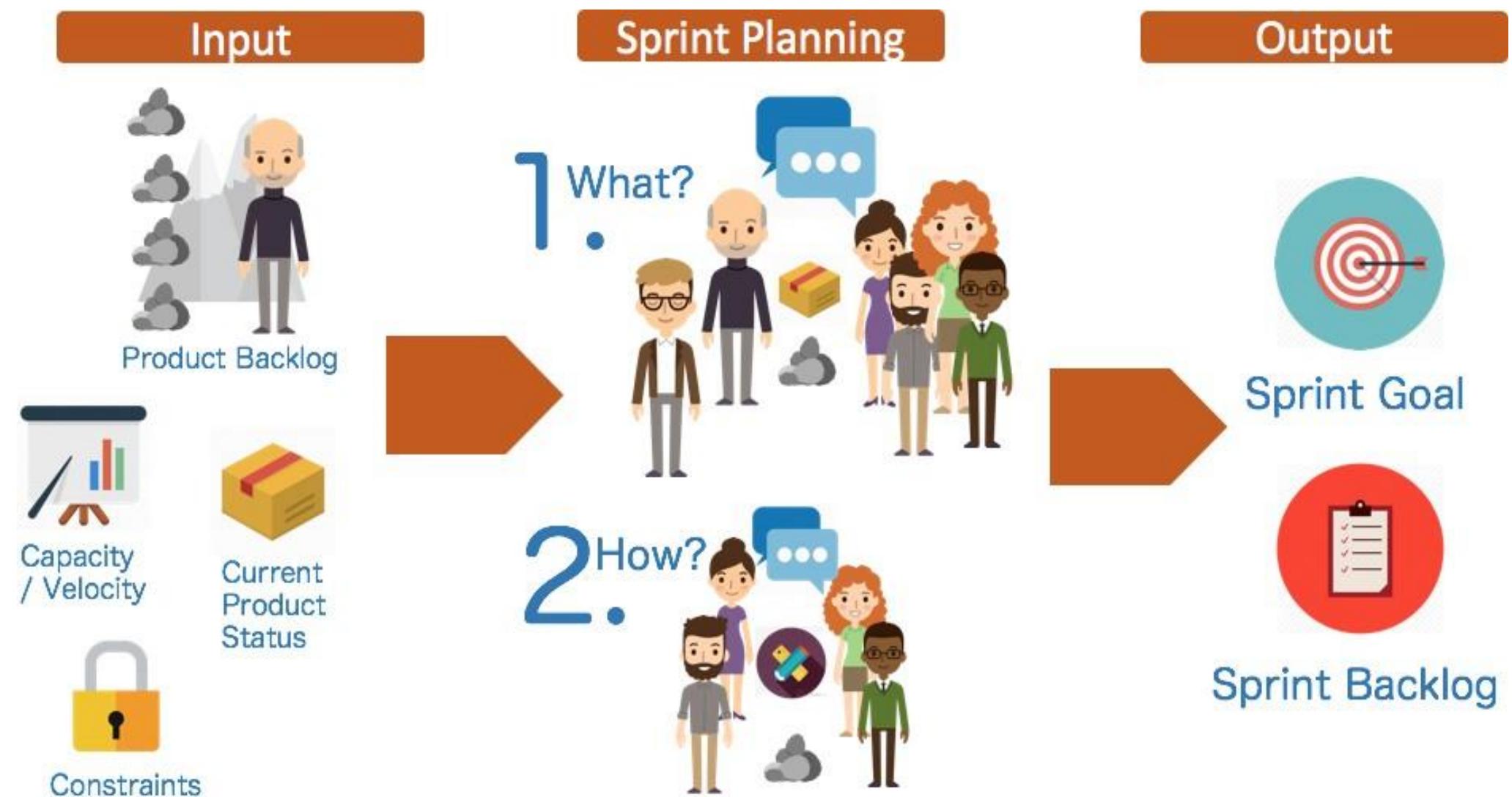
- Keep the Product Backlog in order
- Remove or demote Product Backlog Items that no longer seem important
- Add or promote Product Backlog Items that raise or become more important
- Split Product Backlog Items into smaller items
- Merge Product Backlog Items into larger items
- Estimate Product Backlog

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

Formula

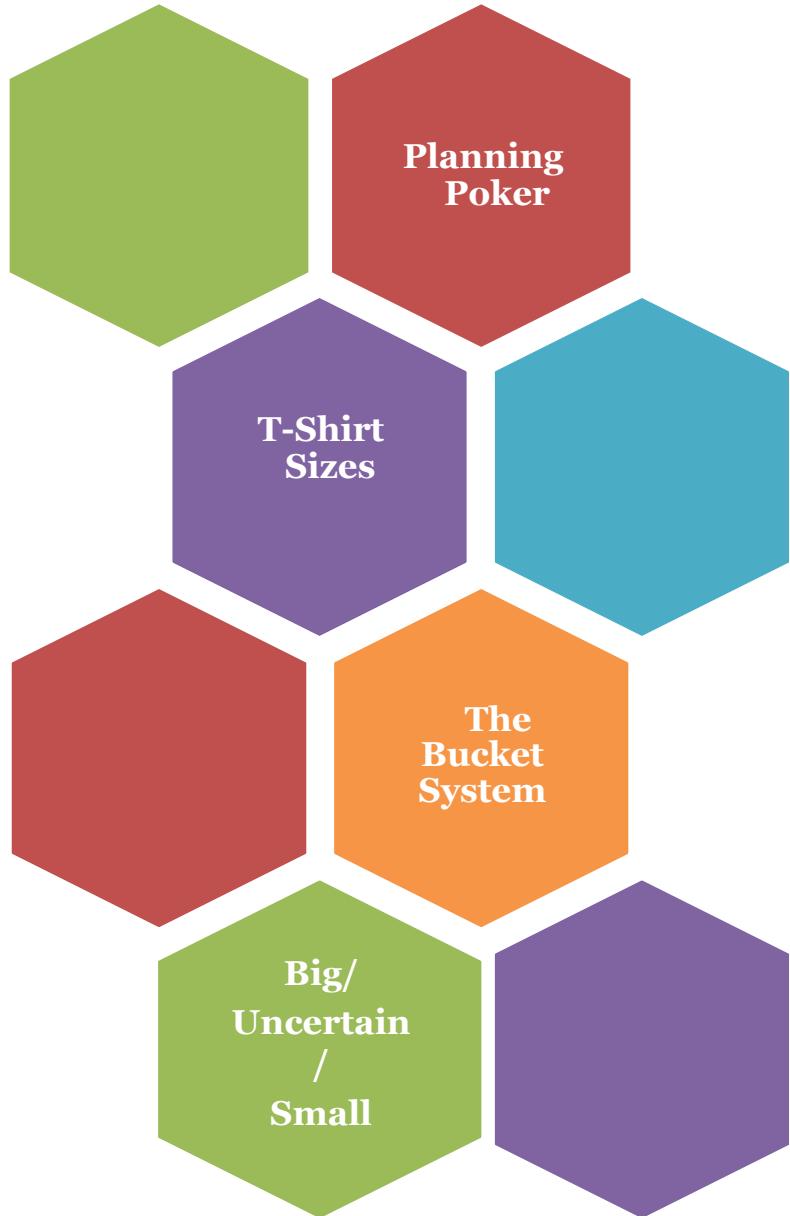
Velocity = Estimated # of iterations * Total size in story points

Ways to improve velocity:

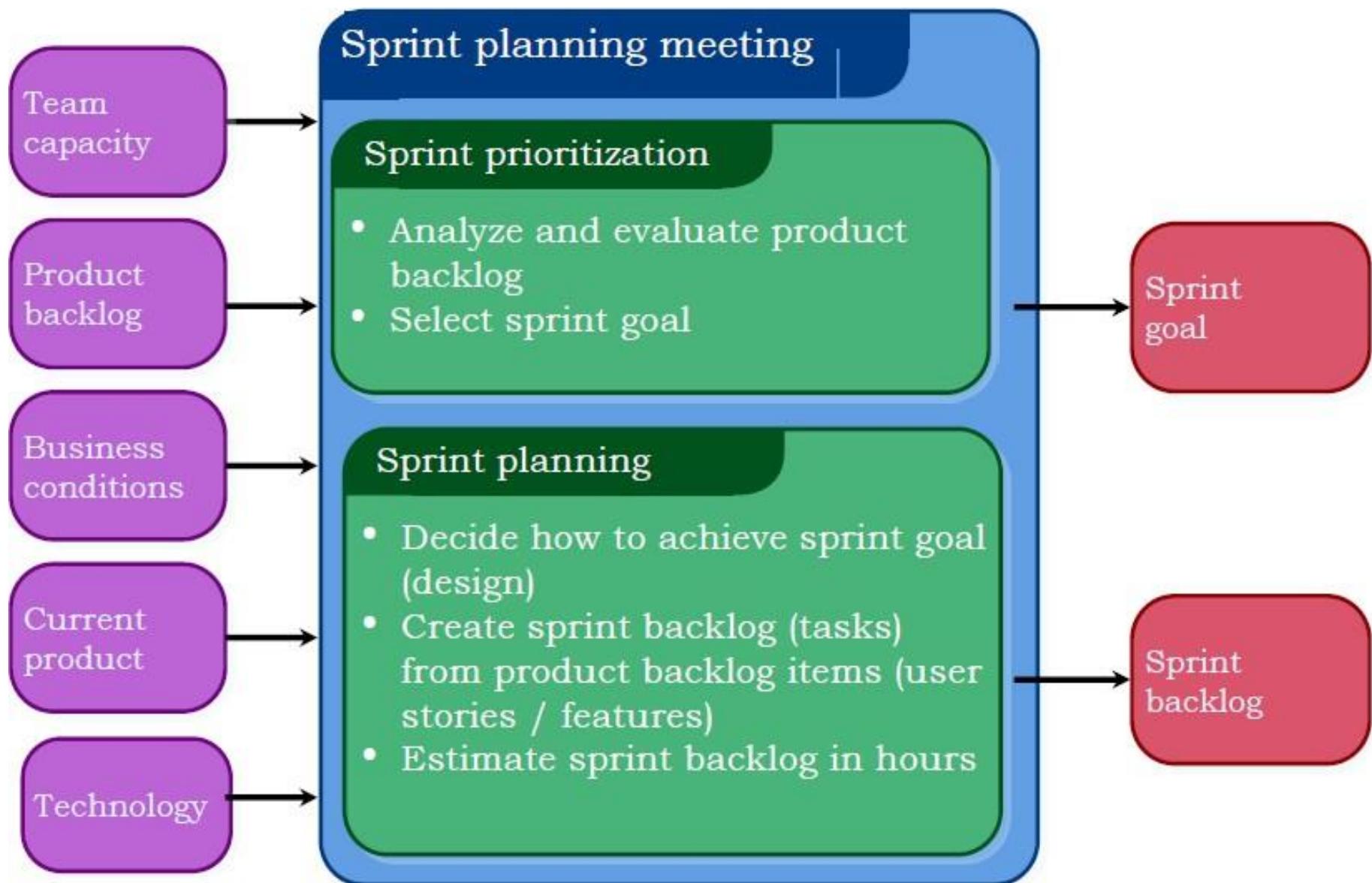
- Reduce technical debt
- Protect team from distractions
- Better customer involvement
- Support energized work
- Right resources at the right time
- Eliminate bottlenecks

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!



Sprint Planning Meeting



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 team 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

The daily scrum

- Parameters
 - Daily
 - 15-minutes
 - Stand-up
- Not for problem solving
 - Whole world is invited
 - Only team members, ScrumMaster, product owner, can talk
- Helps avoid other unnecessary meetings



Everyone answers 3 questions

1

What did you do yesterday?

2

What will you do today?

3

Is anything in your way?

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

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



Sprint Review

The sprint review

- Team presents what it accomplished during the sprint
- Typically takes the form of a demo of new features or underlying architecture
- Informal
 - 2-hour prep time rule
 - No slides
- Whole team participates
- Invite the world



Sprint Review



Sprint Retrospective

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)

Sprint Retrospective Meeting

WHAT WENT WELL

WHAT COULD BE IMPROVED



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

Scrum time boxes



Sprint Planning Meeting

- This is time-boxed to **eight hours for a one-month Sprint**.
- For shorter Sprints, the event is proportionately shorter.

Daily Scrum

- The Daily Scrum is a **15-minute time-boxed** event for the Development Team to synchronize activities and
- create a plan for the next 24 hours.

Sprint Review

- This is a **four-hour time-boxed meeting** for one-month Sprints.
- Proportionately less time is allocated for shorter Sprints

Sprint Retrospective

- This is a **three-hour time-boxed meeting** for one-month Sprints.
- Proportionately less time is allocated for shorter Sprints.



3 Roles

- Development Team
- Product Owner
- ScrumMaster

5 Activities

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

3 Artifacts

- Product Backlog
- Sprint Backlog
- Increment

5 Values

- Courage
- Openness
- Focus
- Commitment
- Respect

Scrum Values

All work performed in Scrum needs a set of values as the foundation for the team's processes and interactions. And by embracing these five values, the team makes them even more instrumental to its health and success.

Focus

Because we focus on only a few things at a time, we work well together and produce excellent work. We deliver valuable items sooner.

Courage

Because we work as a team, we feel supported and have more resources at our disposal. This gives us the courage to undertake greater challenges.

Openness

As we work together, we express how we're doing, what's in our way, and our concerns so they can be addressed.

Commitment

Because we have great control over our own destiny, we are more committed to success.

Respect

As we work together, sharing successes and failures, we come to respect each other and to help each other become worthy of respect.

Sample user stories

#	Backlog Item (User Story)	Story Point
1	As a Teller I want to be able to find clients by last name, so that I can find their profile faster	4
2	As a System Admin I want to be able to configure user settings so that I can control access	2
3	As a System Administrator I want to be able to add new users when required so that...	2
4	As a data entry clerk, I want the system to automatically check my spelling so that...	1

Sample user stories

#0001	USER LOGIN	Fibonacci Size # 3
As a [registered user], I want to [log in], so I can [access subscriber content].		
<small>For new features, annotated wireframe. For bugs, steps to reproduce with screenshot. For non-functional stories, explain scope/standards.</small>		

User Login

Username: User's email address.
Validate format.

Password: Authenticate against SRS
using new web service.

Remember me

[message]

Login

Forgot password?

Store cookie if ticked and login successful.

Display message here if not successful.
(see confirmation scenarios over)

Go to forgotten password page.

Further information is attached to this story on VSTS Product Backlog.

As a customer, I want to be able to search for flights between two cities to see which ones have the best price and route.
Estimate: 1.0 points
Priority: 2 - High

Example Story Card

Artifacts





**Thank
You!!!**

Is it so easy?

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



12 AGILE PRINCIPLES IN 27 WORDS

WHAT?

Satisfy customer
Welcome changes
Deliver frequently

HOW?

Collaborate daily
Motivate, support, trust
Interact in-person

BUILD

Working software
Sustainable pace
Technical excellence

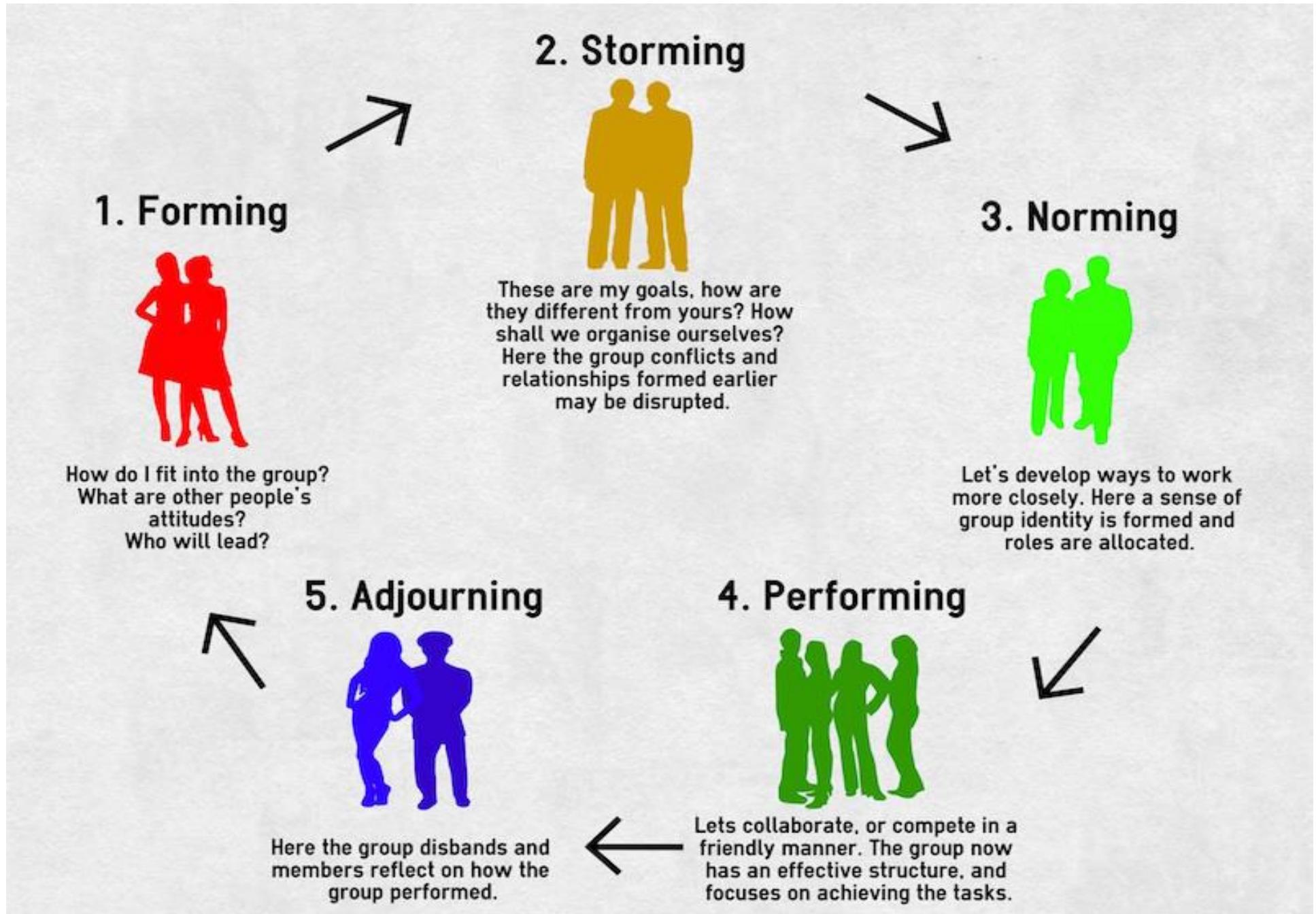
TEAM

Simplicity
Self-organize
Reflect together

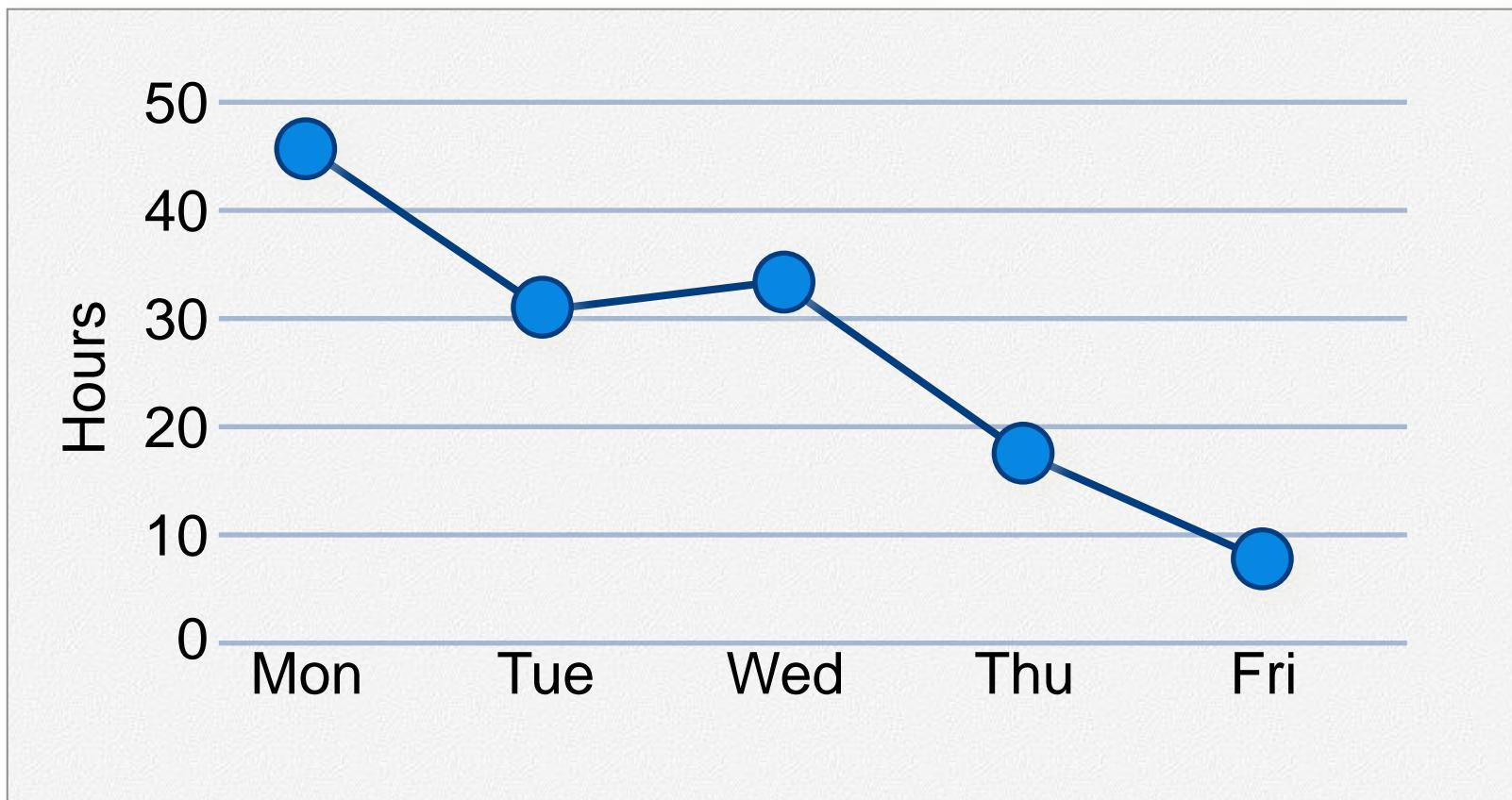
Where to go next

- www.mountaingoatsoftware.com/scrum
- www.scrumalliance.org
- www.controlchaos.com
- scrumdevelopment@yahoogroups.com

Group formation theory –Tuckman's rule

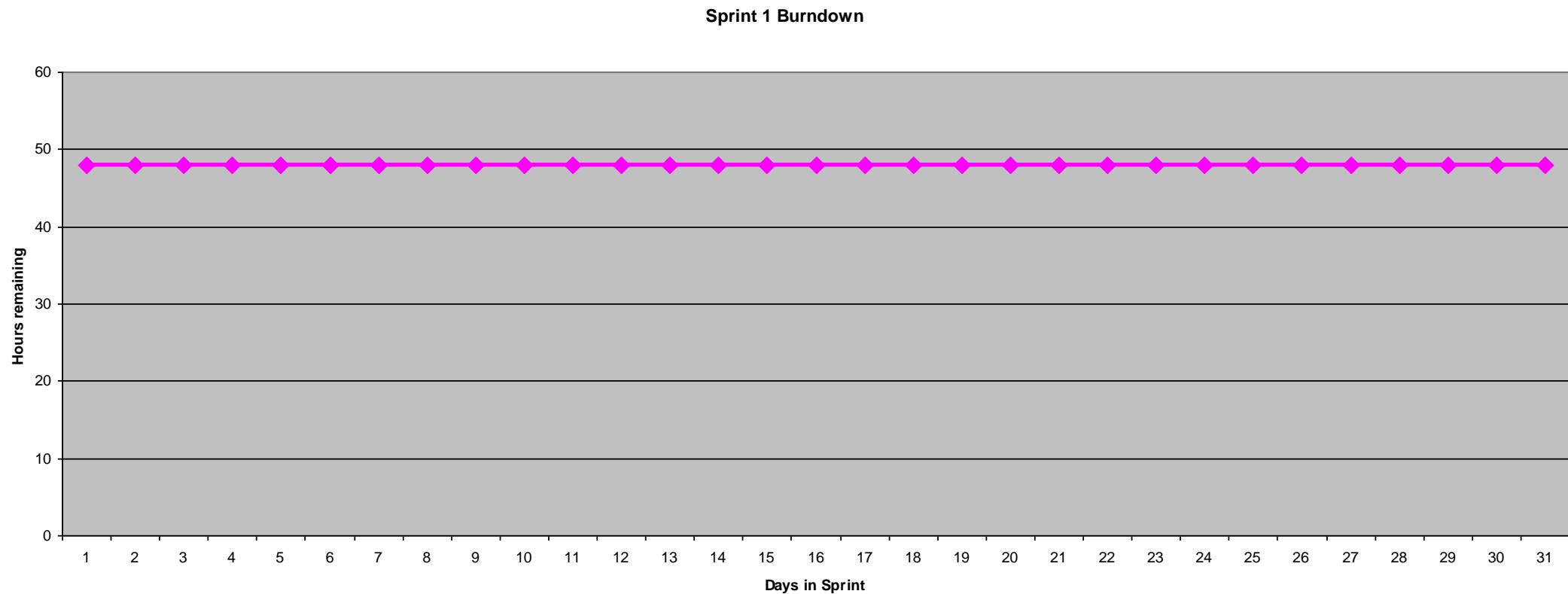


Tasks	Mon	Tue	Wed	Thu	Fri
Code the user interface	8	4	8		
Code the middle tier	16	12	10	7	
Test the middle tier	8	16	16	11	8
Write online help	12				



Burndown Example 1

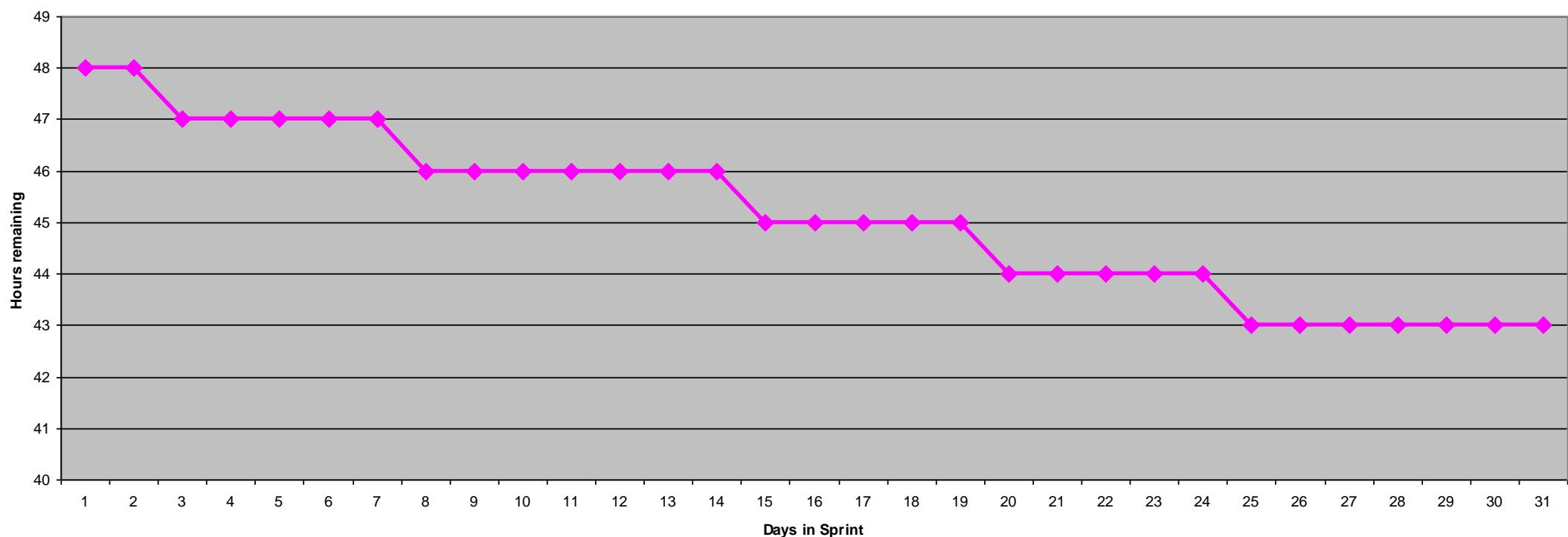
No work being performed



Burndown Example 2

Work being performed, but not fast enough

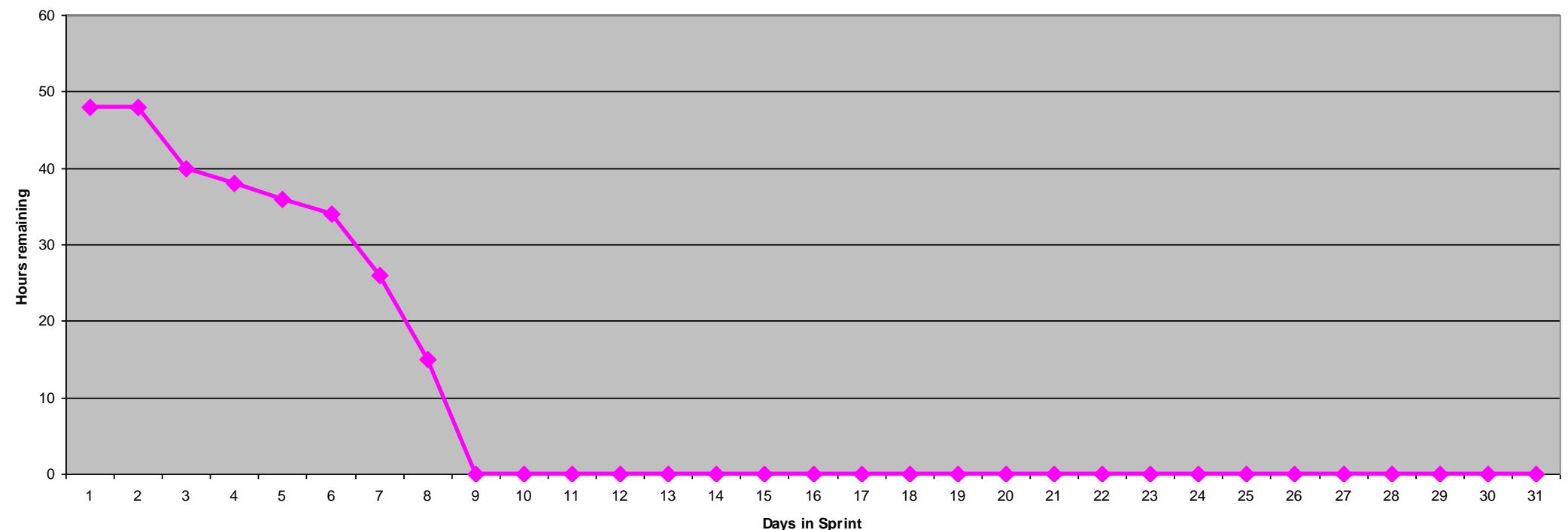
Sprint 1 Burndown



Burndown Example 3

Work being performed, but too fast!

Sprint 1 Burndown



Scrum vs. Other Models

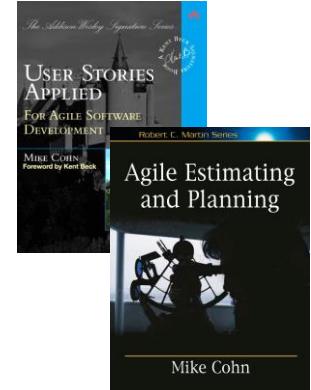
Process Comparison

	Waterfall	Spiral	Iterative	SCRUM
Defined processes	Required	Required	Required	Planning & Closure only
Final product	Determined during planning	Determined during planning	Set during project	Set during project
Project cost	Determined during planning	Partially variable	Set during project	Set during project
Completion date	Determined during planning	Partially variable	Set during project	Set during project
Responsiveness to environment	Planning only	Planning primarily	At end of each iteration	Throughout
Team flexibility, creativity	Limited - cookbook approach	Limited - cookbook approach	Limited - cookbook approach	Unlimited during iterations
Knowledge transfer	Training prior to project	Training prior to project	Training prior to project	Teamwork during project
Probability of success	Low	Medium Low	Medium	High

Credits, References

- Mike Cohn, Mountain Goat Software
www.mountaingoatsoftware.com
- *Scrum and The Enterprise* by Ken Schwaber
- *Succeeding with Agile* by Mike Cohn
- *Agile Software Development Ecosystems* by Jim Highsmith
- *Agile Software Development with Scrum* by K. Schwaber and M. Beedle
- *User Stories Applied for Agile Software Development* by Mike Cohn

- www.agilescrum.com/
- www.objectmentor.com
- jeffsutherland.com/
- www.controlchaos.com/scrumwp.htm
- agilealliance.com/articles/articles/InventingScrum.pdf



Types of organization

Organizational Structure

Type #1: Functional Organization

Functional Organization is the most common type of organizational structures in the industry. It is adopted by many multinational and big companies, in many industries.

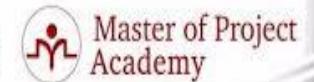
Type #2: Projectized Organization

In Projectized Organization, the entire company is organized by projects. So the resources of the project are fully dedicated to the project activities. Project Managers have control on the projects. Resources only report to project manager and the project manager has the ultimate control of resources.

Type #3: Matrix Organization

Matrix Organization attempts to get strengths of Projectized and Functional Organizations. Projectized Organizations ensure the dedication of project resources to the project.

Organizational Influences & Project Lifecycle



Comparative Overview of Organizational Structures

Project Characteristics	Organization Structure	Functional	Matrix			Projectized
			Weak Matrix	Balanced Matrix	Strong Matrix	
Project Manager's Authority	Little or None	Low	Low to Moderate	Moderate to High	High to Almost Total	
Resource Availability	Little or None	Low	Low to Moderate	Moderate to High	High to Almost Total	
Who manages the project budget	Functional Manager	Functional Manager	Mixed	Project Manager	Project Manager	
Project Manager's Role	Part-time	Part-time	Full-time	Full-time	Full-time	
Project Management Administrative Staff	Part-time	Part-time	Part-time	Full-time	Full-time	