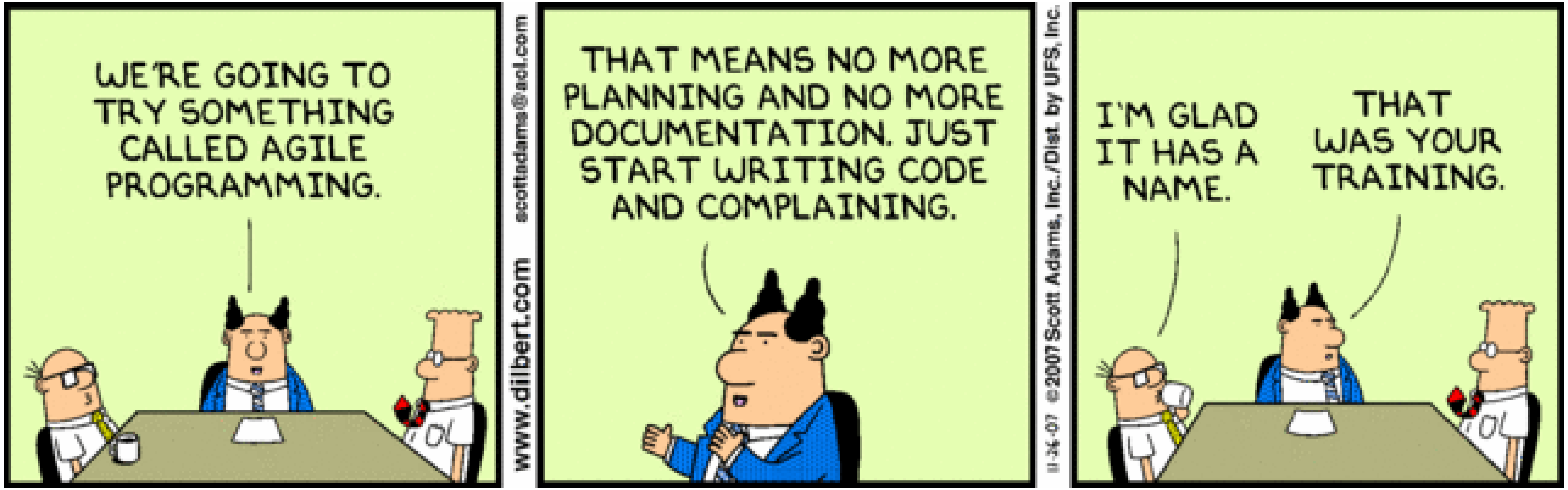
Agile in Brief

By Naseem Amjad (Agile/DevOps Coach)

naseem@technologist.com

August 03, 2017

Training Summary

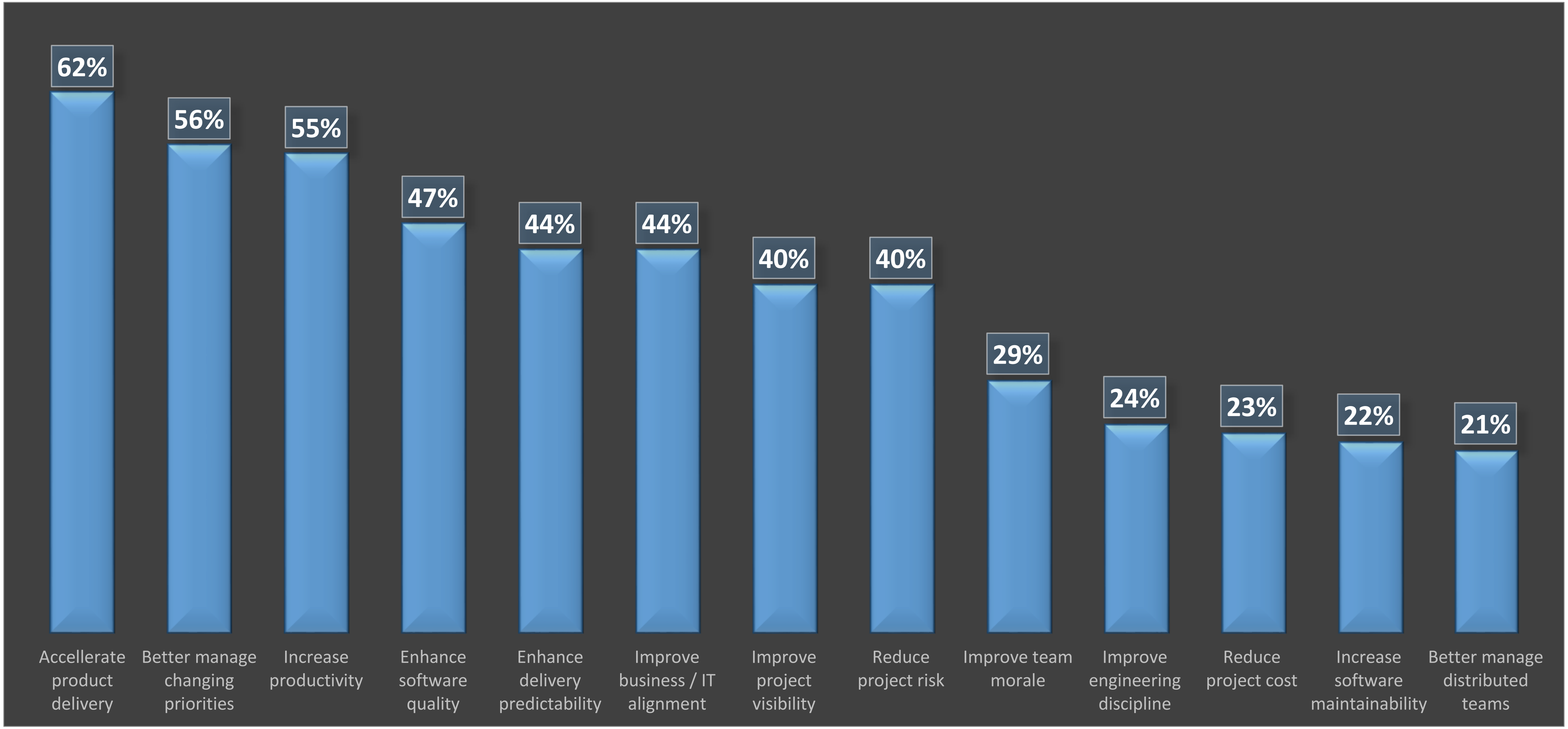

WE'RE GOING TO TRY SOMETHING CALLED AGILE PROGRAMMING.
THAT MEANS NO MORE PLANNING AND NO MORE DOCUMENTATION. JUST START WRITING CODE AND COMPLAINING.
I'M GLAD IT HAS A NAME.
THAT WAS YOUR TRAINING.
www.dilbert.com scottadams@aol.com
11-26-07 ©2007 Scott Adams, Inc./Dist. by UFS, Inc.

Overview of Agile

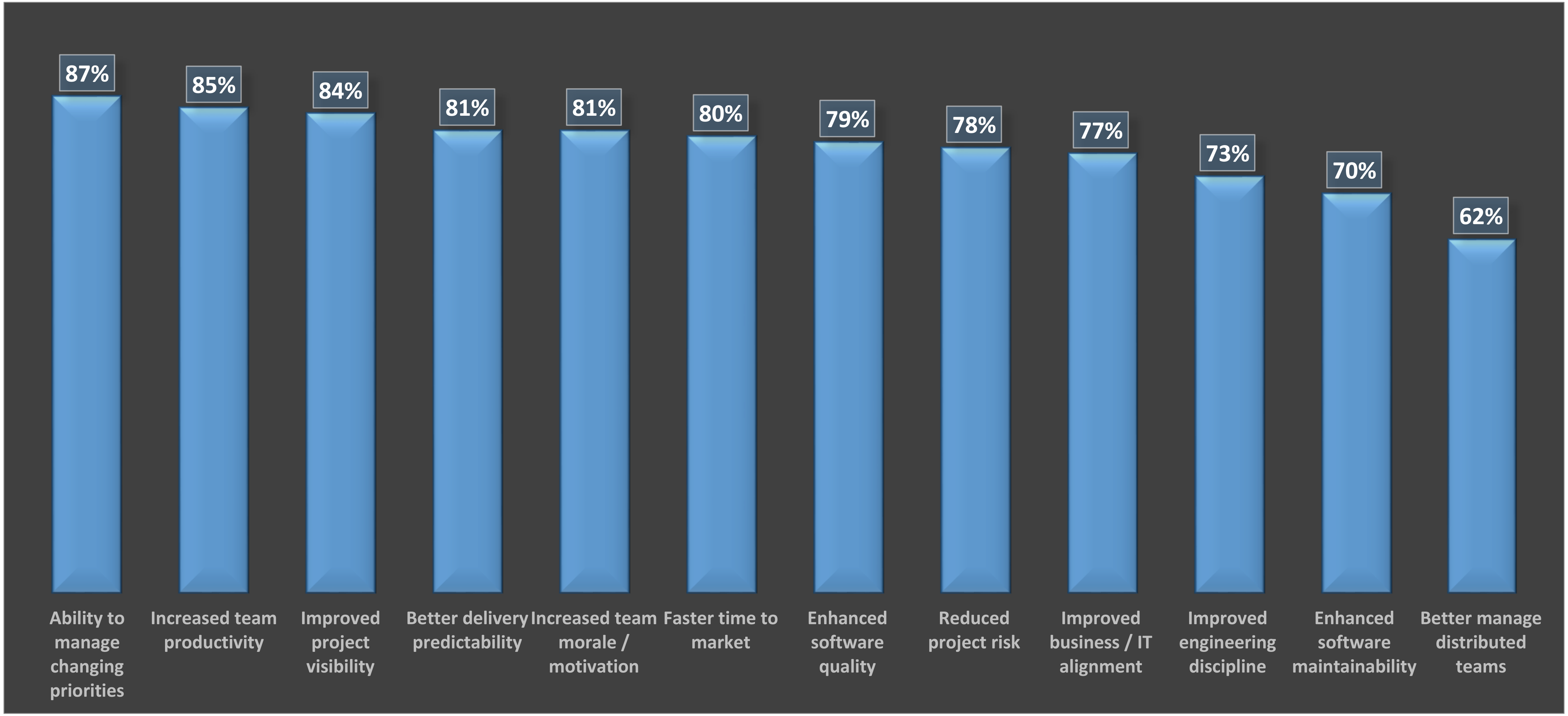
Brief history

- Conceptualised in 1986:
  - An article was written about how Honda, Canon and Fuji-Xerox produced world class results using a scalable, team-based approach
- Scrum process was created in 1993 by Jeff Sutherland
- The first paper on Agile Scrum was written by Ken Schwaber in 1995
  - This was a 14 page document
  - This is still the foundation for Agile scrum (16 pages to-date)

Why did companies adopt Agile?



What did Agile improve?



Who is using Agile?

- In April 2010, Apple launched the iPad
  - It wasn't a fully featured tablet PC, but it contained minimal marketable features (MMFs)
  - 3 million devices were sold in 3 months
  - 15 million devices were sold in 8 months, capturing 75% market share

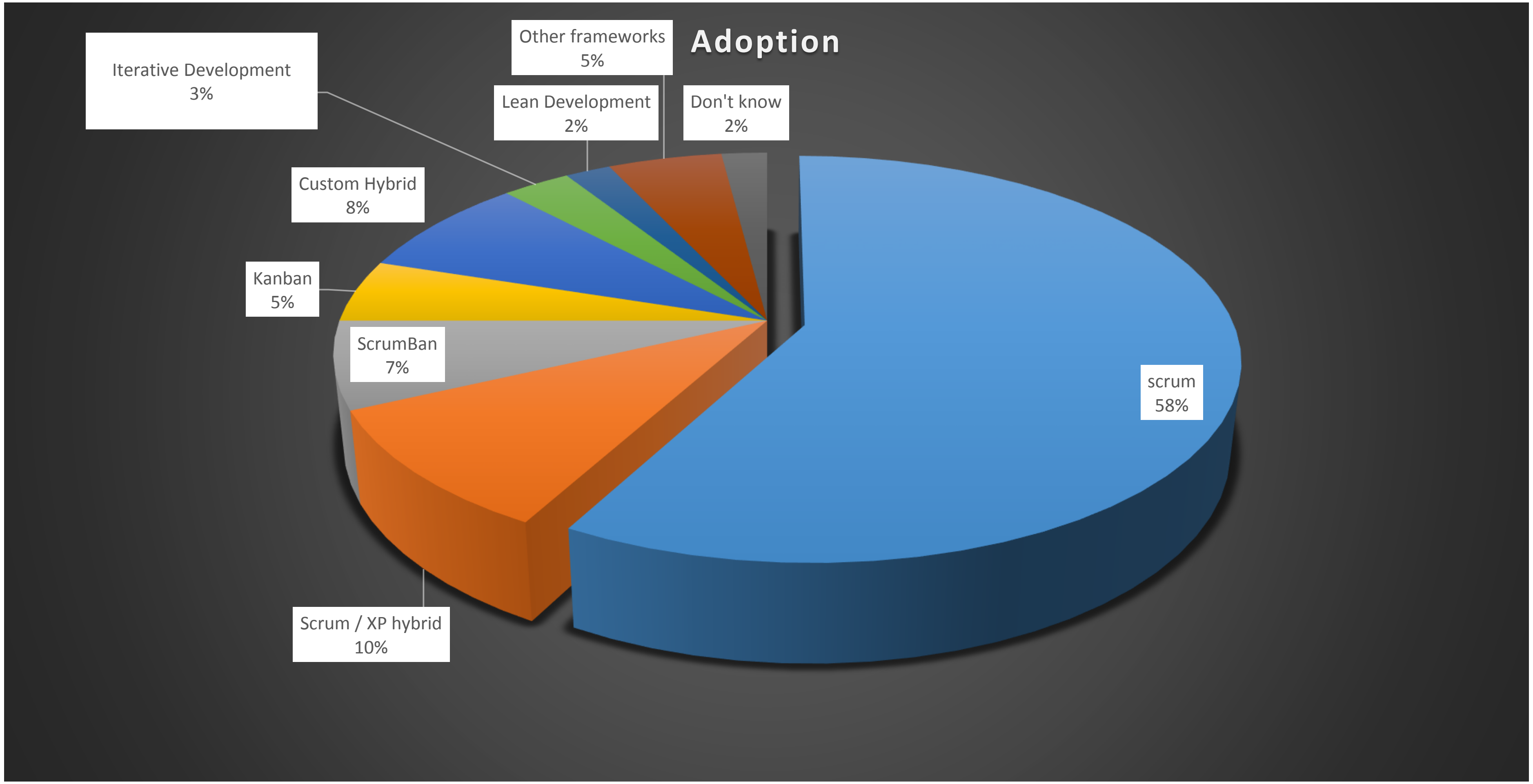
- The major reason for success was not the number of features but the addition of those features which mattered to the customer

Who is using Agile?

- Apple
- Atlassian (Jira, confluence, etc.)
- Dell
- DHL
- Facebook
- FedEx Corp
- General Motors Corp
- Google
- Harvard Business School
- Hewlett-Packard Corp
- IBM
- Marriott International
- Microsoft
- Rockwell Colins
- SalesForce.com
- Siemens Healthcare
- Sony Ericsson
- Toshiba
- Toyota
- UNICEF
- Verizon Wireless
- Warner Music Group
- Yahoo
- Twitter
- And thousands more …

What are they using?

- 95% of software development companies use agile in some form, 80% scrum related

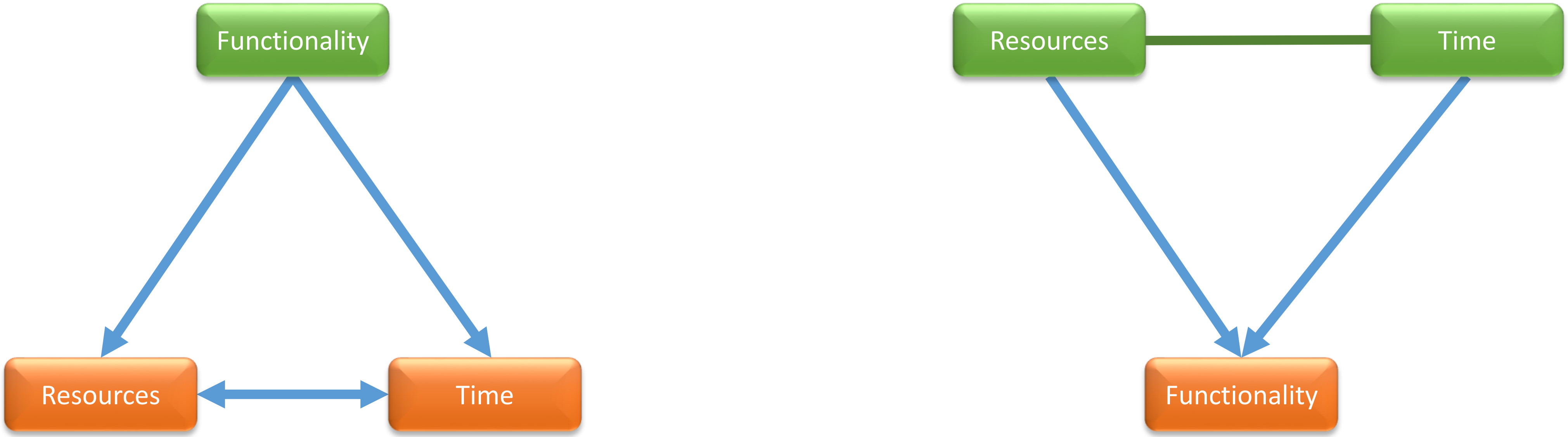


Fundamentals

Agile vs Waterfall

Waterfall

Agile



Functionality

Resources

Time

Resources

Time

Functionality

Legend

🟩 Fixed — Fixed

🟧 Flexible ↔ Drives

Agile Manifesto

- **Individuals and interactions**
- **Working Software**
- **Customer collaboration**
- **Responding to change**

>

- Processes and tools
- Comprehensive documentation
- Contract Negotiation
- Following a plan

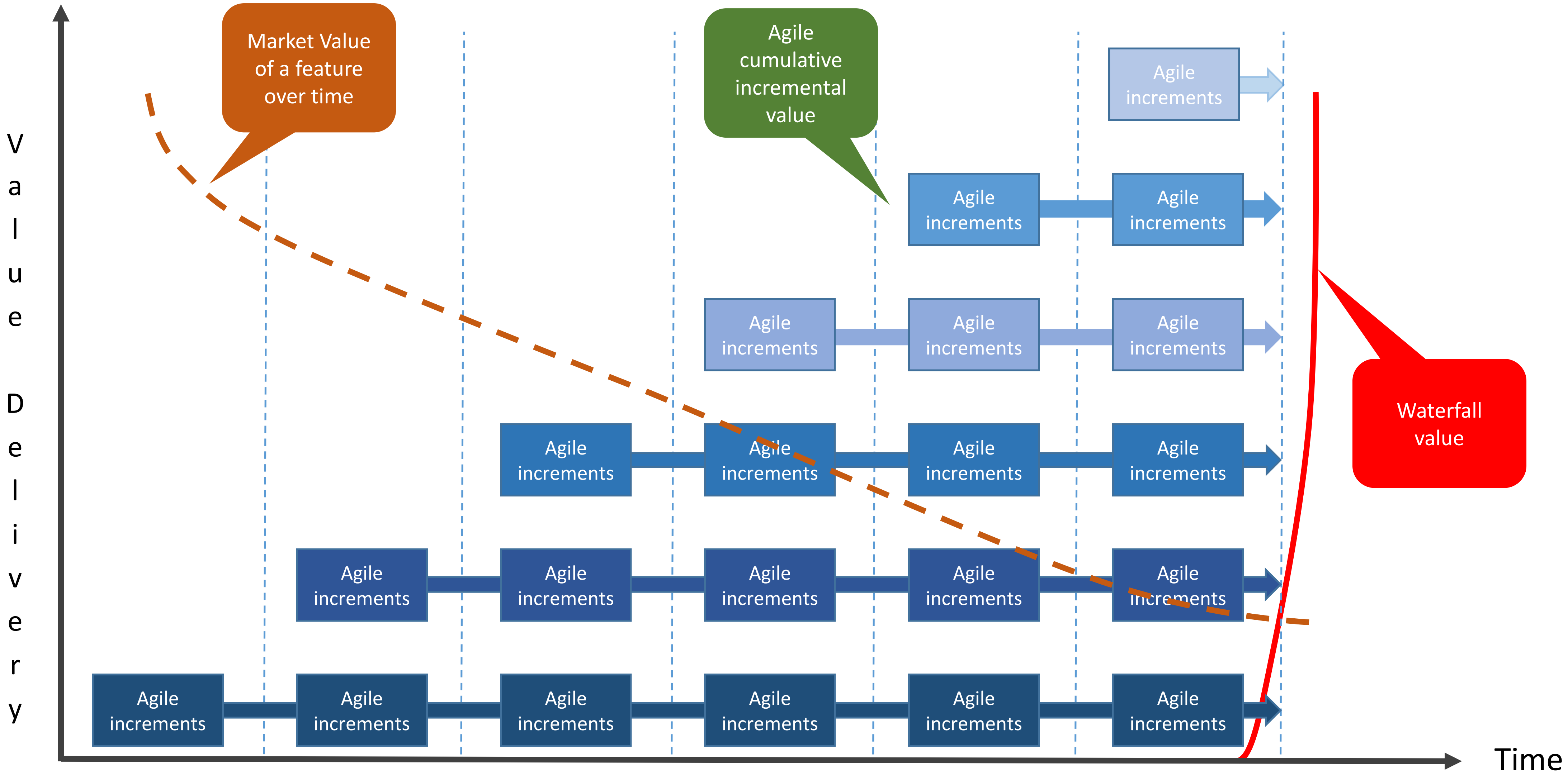
Important:

- Agile sees value in the items on the right
- However, Agile values the items on the left more

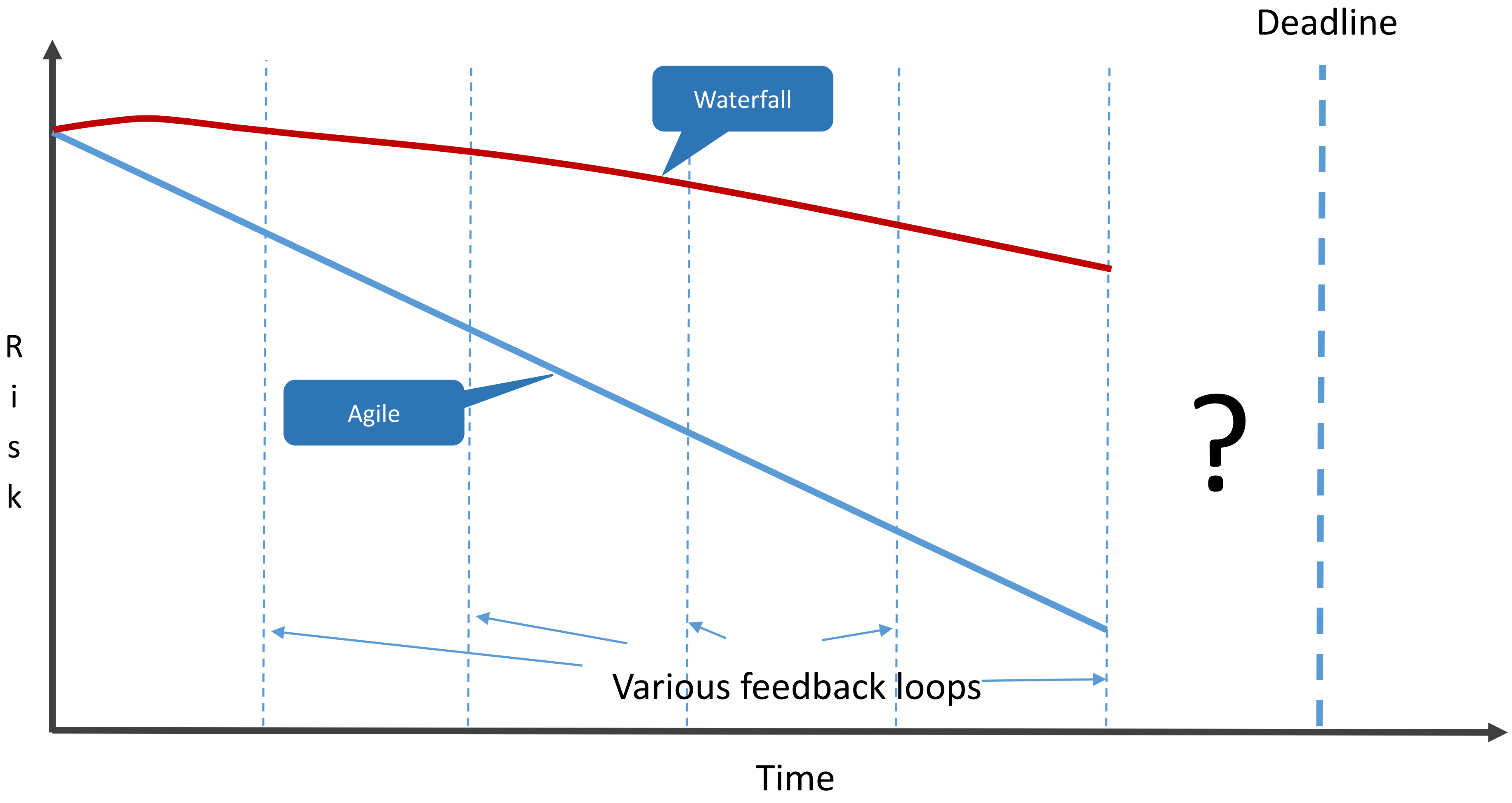
Fundamentals

- Agile is about open and frequent communication
- Agile is about adopting to change, when/where it makes sense
- Agile is about ownership and empowerment
- Agile is about teamwork
- Agile is transparent

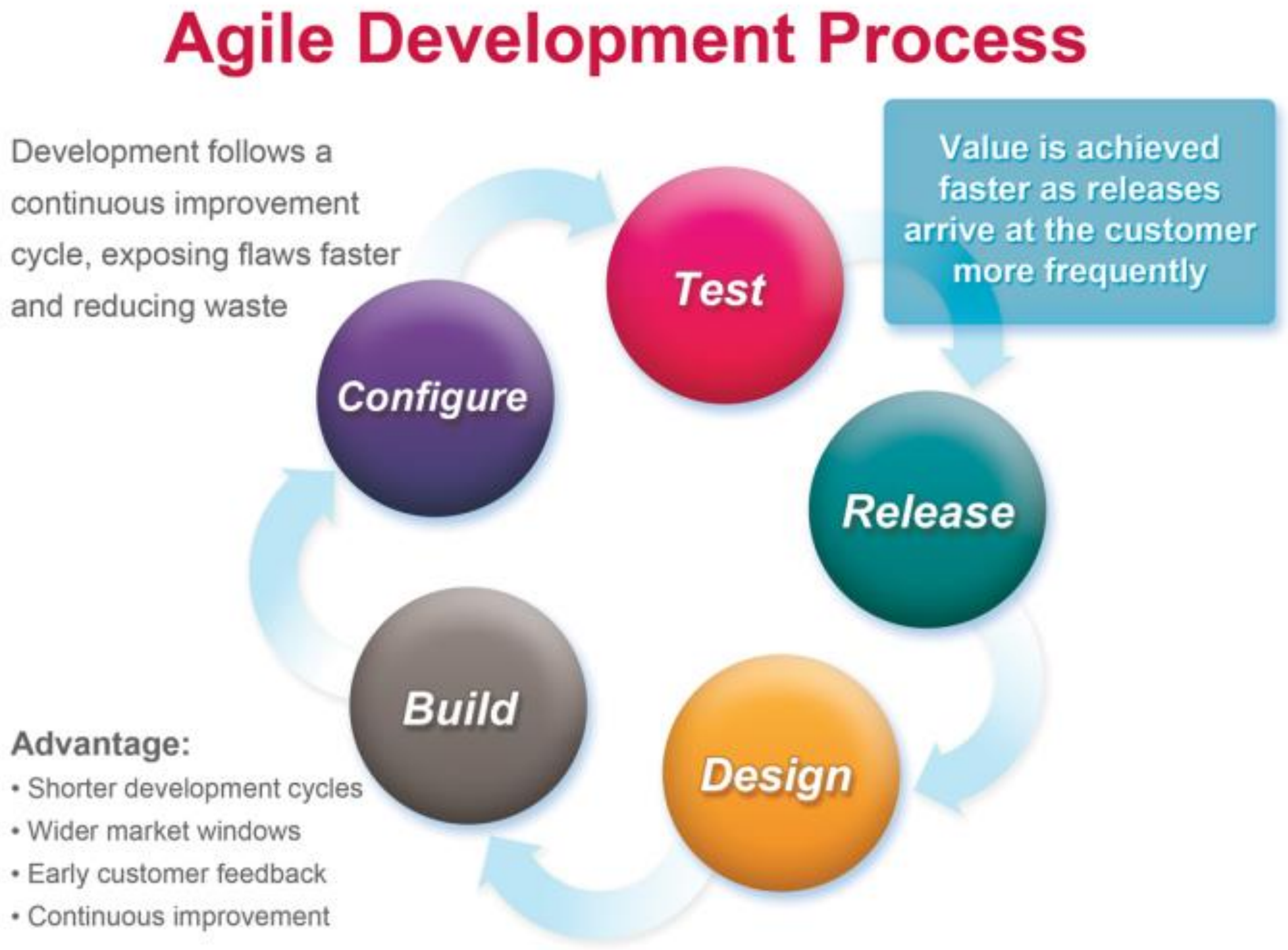
Provide value early on


Market Value of a feature over time
Agile cumulative incremental value
Waterfall value
Value Delivery
Time
Agile increments

Reduce Risk


Deadline
Waterfall
Agile
R
i
s
k
?
Various feedback loops
Time

Reduction Waste / Technical Debt



Agile Development Process

Development follows a continuous improvement cycle, exposing flaws faster and reducing waste

Value is achieved faster as releases arrive at the customer more frequently

Advantage:

- Shorter development cycles
- Wider market windows
- Early customer feedback
- Continuous improvement

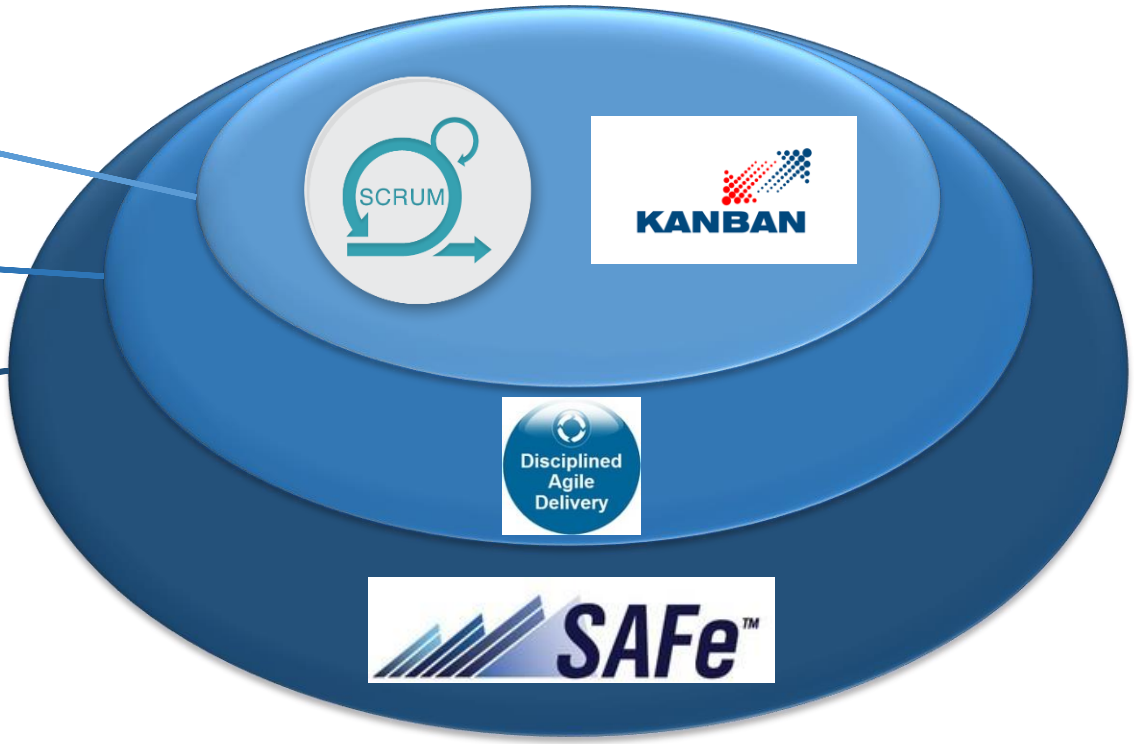
Common Misconceptions

- Agile is about speed / delivering applications quickly
  - Speed comes secondary to quality
  - Agile is about continuously delivering quality software increments at regular intervals
- Agile gives instant benefits
  - Agile gives huge benefits once properly implemented but transformation means a learning curve. This can have an initial downwards effect on productivity which is recovered later on.
- Agile is easy to implement
  - Agile is about transformation, of the processes and delivery lifecycle as well as of the company culture and how people work together in their daily work environment, typically implementation takes between 12 – 18 months.
- Agile means no planning
  - Especially when using D.A. planning is very much a part of agile development projects, even in scrum planning is performed in sprint 0s
  - Detailed planning is as essential to agile being effective as it is to waterfall.
- Individuals get to do what they want
  - Agile requires the development team (which means DEV, QA, BA, DBA, Architects, etc.) to work together as a team; the team is measured as a whole and as a whole responsible for the deliverables.

Agile Scrum

Agile Structure

- Operational level
- Enterprise level
- International level



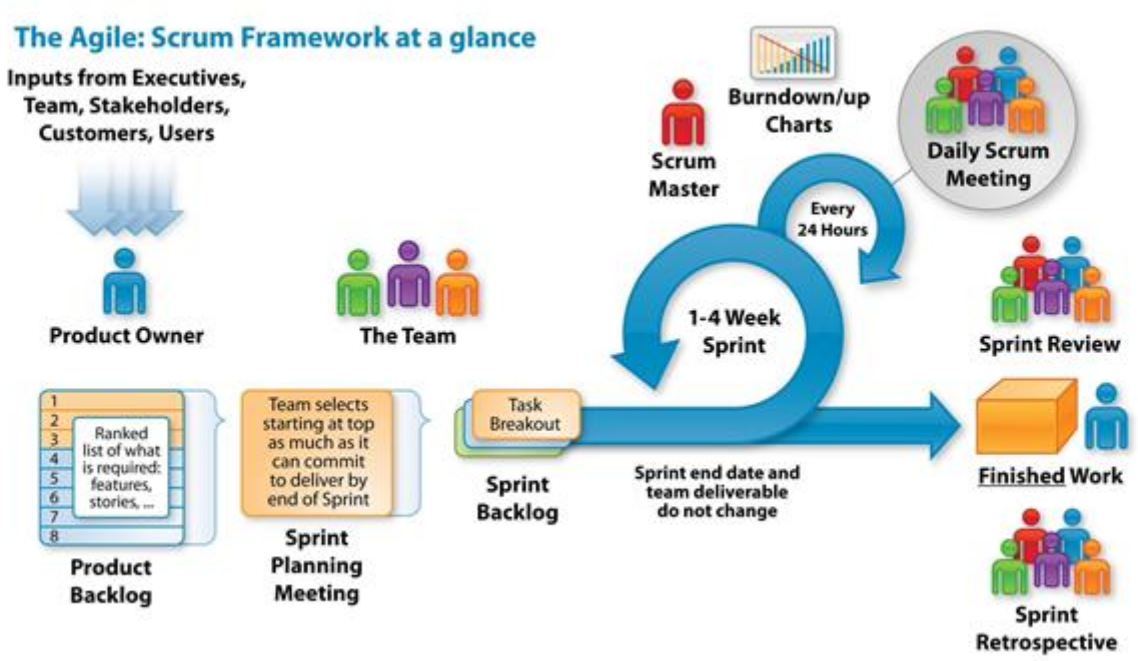
So what is Scrum?

- Scrum is a framework structured to support complex product development.

- Scrum consists of Scrum teams and associated
  - Roles
  - Events
  - Artifacts
  - Rules

- Each of these serves a specific purpose and is essential to Scrum's success and usage

Scrum Framework



The Agile: Scrum Framework at a glance
Inputs from Executives, Team, Stakeholders, Customers, Users
Product Owner
The Team
Scrum Master
Burndown/up Charts
Daily Scrum Meeting
Every 24 Hours
1-4 Week Sprint
1
2
3
4
5
6
7
8
Ranked list of what is required: features, stories, ...
Product Backlog
Team selects starting at top as much as it can commit to deliver by end of Sprint
Sprint Planning Meeting
Task Breakout
Sprint Backlog
Sprint end date and team deliverable do not change
Sprint Review
Finished Work
Sprint Retrospective

The Team

- Is self regulating, self managing and self governing
- Is cross functional
- Is trusted and self organising
- Creates solutions by designing, developing, testing, and whatever else is required to build the solution.
- Turns the product backlog into increments of potentially shippable functionality, every sprint.
- Maximum team size is 9 people
- The team is co-located

Roles: The Scrum Team

The Team

- Members of QA, DEV, DBA, BA, architects, etc.
- Select stories and define tasks for each story
- Size stories and tasks
- As a team responsible for delivering potentially shippable product increments at the end of each sprint
- Responsible for the sprint backlog
- Self regulating, self governing

Product Owner

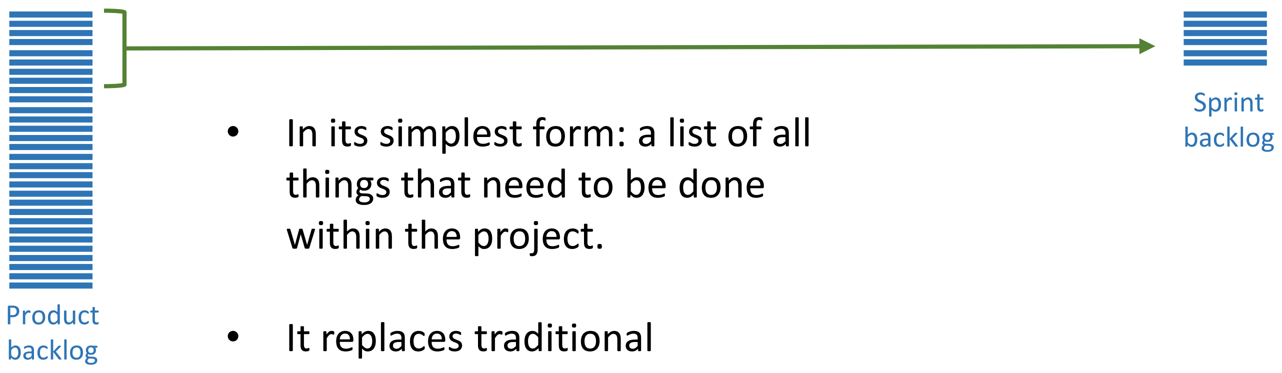
- Members of BAG, PM, pre-sales, etc.
- Owns the product backlog
- Voice of the customer
- Writes stories and acceptance criteria
- Responsible for maximising value of the product and work of the team
- Responsible for creating and maintaining the product backlog
- Often this role is played by the customer

Scrum Master

- Team lead, team member, diplomat
- Servant leader
- Ensures rules are understood and followed
- Resolves impediments
- Protects team from outside interference
- Improves productivity
- Improves development practices
- Responsible for administration and coordination

Artifacts

Product Backlog



- In its simplest form: a list of all things that need to be done within the project.
- It replaces traditional requirements specification artefacts
- They can be technical in nature or user centric (epics, themes and stories)
- It is a <u>prioritised</u> features list, numerical and MoSCoW
- It contains epics and stories

Product Owner

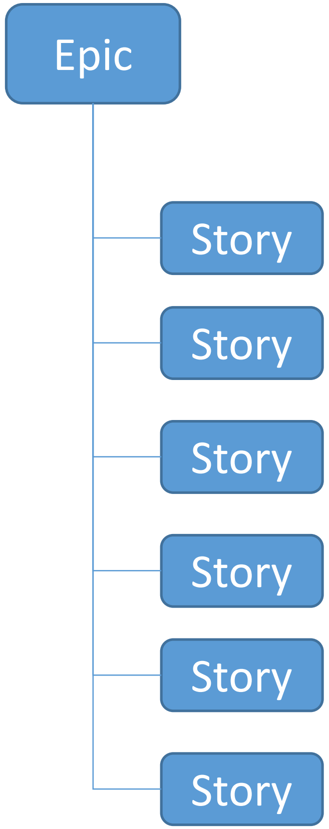
Sprint Backlog

- A list of the items that a t commits to complete wit current sprint
- It contains Stories and tas
- Stories are prioritised usi MoSCo
- Stories are sized (story pc
- Task are estimated (man-hours)

The Team

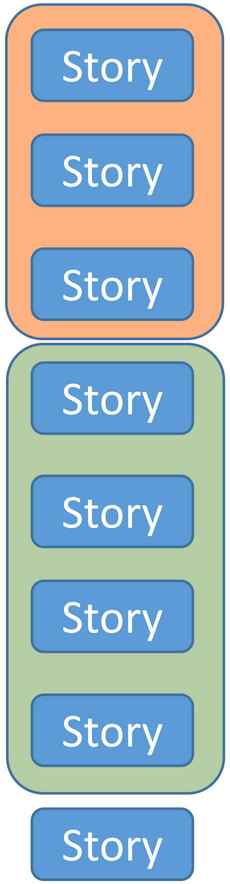
Epics and Themes

Epic



- A story that is too large in and off itself
- Epics can exist in the product backlog but not the sprint backlog
- They need to be broken down into multiple user stories

Theme



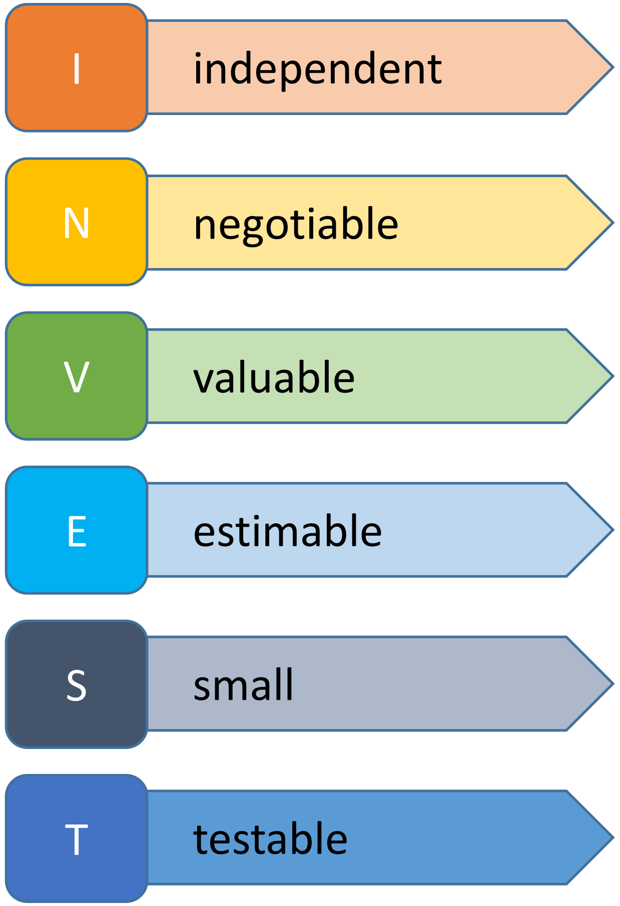
- A collection of related stories
- Often when an epic is broken into stories, that collection of stories can be considered a theme
- Themes can also be related to modules or common sets of functions

User Stories

EXTREME PROGRAMMING
I CAN'T GIVE YOU ALL OF THESE FEATURES IN THE FIRST VERSION.
scottadams@aol.com
www.dilbert.com
AND EACH FEATURE NEEDS TO HAVE WHAT WE CALL A "USER STORY."
© 2002 United Feature Syndicate, Inc.
1/10/03
OKAY, HERE'S A STORY: YOU GIVE ME ALL OF MY FEATURES OR I'LL RUIN YOUR LIFE.

User Stories

- Story format
  - Story:
    - As a: (user)
    - I want: (function / action)
    - So that: (result / desired goal)
  - Acceptance criteria:
    - Given: (an event occurs)
    - When: (I do an action)
    - Then: (I get an expected result)
  - Definition of Done
- Can contain:
  - Wireframes
  - Personas
  - Use cases
- Follow INVEST principles



I independent

N negotiable

V valuable

E estimable

S small

T testable

User Stories

I independent

- Of order of user story delivery
- Of internal and external dependencies

N negotiable

- Flexible scope
- Non-specific language
- Explain the intention, not the implementation

V valuable

- Value is clear to everyone
- Persona matches benefit and goal will deliver the benefit
- Avoid technical / role specific language

E estimable

- Clear and concise explanation
- Breakable into tasks

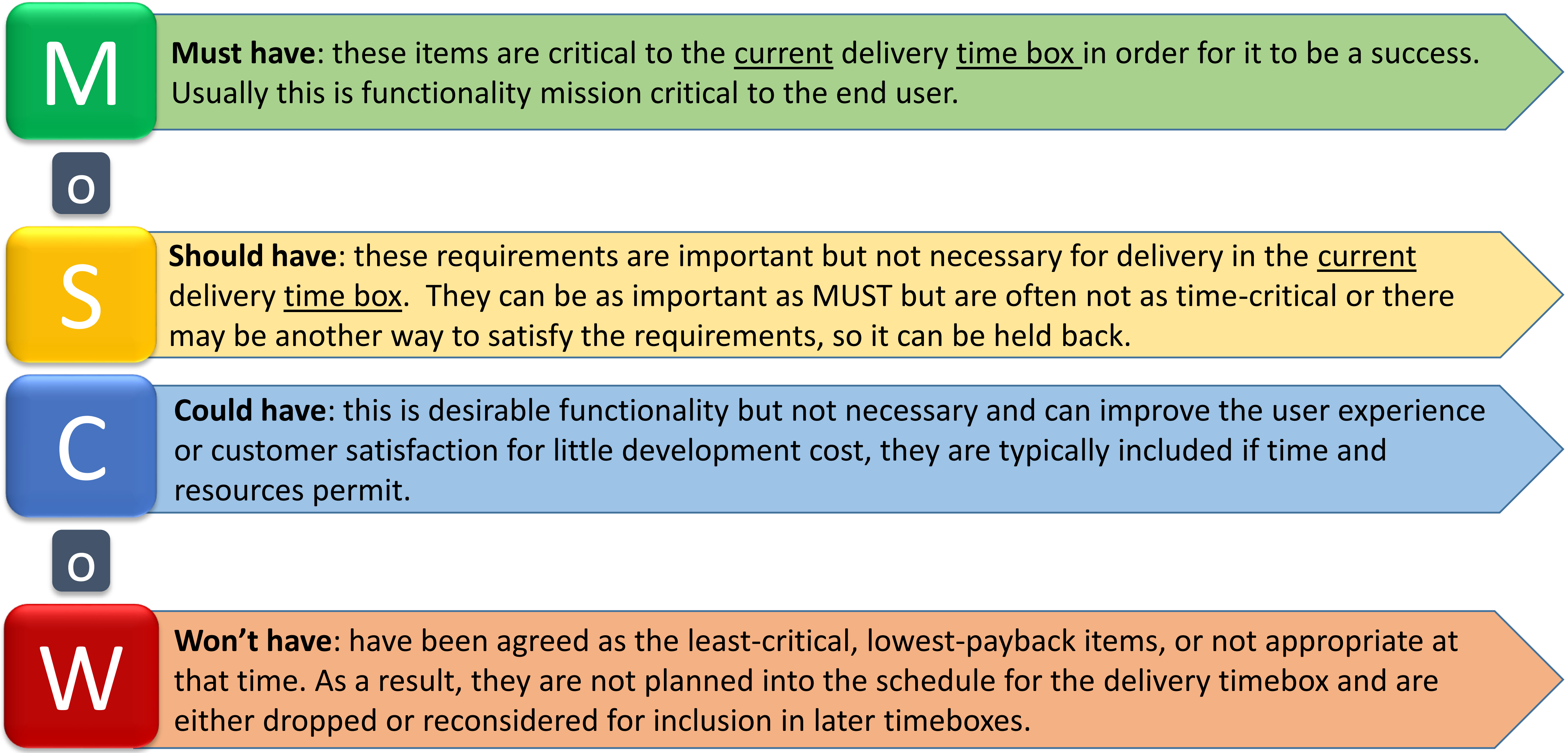
S small

- Easily fits into a sprint <20% of velocity
- Breakable into tasks that are not more than 8 hours each

T testable

- Can be automated
- Avoids external testing / long test suites

Prioritisation



M

o

S

C

o

W

Must have: these items are critical to the current delivery time box in order for it to be a success. Usually this is functionality mission critical to the end user.

Should have: these requirements are important but not necessary for delivery in the current delivery time box. They can be as important as MUST but are often not as time-critical or there may be another way to satisfy the requirements, so it can be held back.

Could have: this is desirable functionality but not necessary and can improve the user experience or customer satisfaction for little development cost, they are typically included if time and resources permit.

Won’t have: have been agreed as the least-critical, lowest-payback items, or not appropriate at that time. As a result, they are not planned into the schedule for the delivery timebox and are either dropped or reconsidered for inclusion in later timeboxes.

Personas

- A Persona is a hypothetical user of the system
- They represent the end user and are used as a point of reference when the end user is not available
- They are compiled by the team and used across multiple sprints
- They force the team to think about the users and help gain insight into who the user will be
- They provoke discussion about users
- They put a human face to the end user
- They capture important requirements and preferences

Acceptance Criteria

- Unique and apply to each story
- Defined by the product team
- The measure by which the product owner will accept a story as “done”
- Follow a set structure:
  - Given: (an event occurs)
  - When: (I do an action)
  - Then: (I get an expected result)
- Used as a basis for test cases
- A story is finished when it is “done and done”

Definition of Done (DoD)

- The team defines DoD for features and sprints, however DoD can apply to 3 different levels:
  - DoD for a feature (story or backlog item)
  - DoD for a sprint (collection of features)
  - DoD for a release (collection of sprints)

- There are multiple factors that influence whether an activity belongs in the DoD of which state, defined as follows:
  - Can we do this activity for each feature? If not, then:
  - Can we do this activity for each sprint? If not, then:
  - We <u>have</u> to do this activity for our release.

Definition of Done (DoD)

- The team as a collective defines the DoD, following these minimal standards:
  - Code has been produced (all to-do items in the code have been completed)
  - Code has been commented, checked in and run against current version in source control
  - Peer reviewed and meets all development standards, duly checked
  - Builds run without errors
  - Unit tests have been written, are of quality and all pass (including pre-existing unit tests)
  - Deployed to test environment and has passed all tests
  - Passed UAT and signed off as meeting the requirements
  - Any build / deployment / configuration change implemented, documented and communicated
  - Relevant documentation, prototypes and diagrams produced and / or updated
  - Remaining hours for tasks set to zero and tasks closed on the scrum board.

- Once a story complies with the acceptance criteria (done) and the DoD (done) it is considered complete (done and done)

Sample: User Story

M

8

User story title: Customer withdraws cash.

- As a customer,
- I want to withdraw cash from an ATM
- So that I don't have to wait in line at the bank.

Acceptance Criterion 1:

Given that the account is creditworthy
 And the card is valid
 And the dispenser contains cash,
When the customer requests the cash
Then ensure the account is debited
 And ensure cash is dispensed
 And ensure the card is returned.

Acceptance Criterion 2:

Given that the account is overdrawn
 And the card is valid,
When the customer requests the cash
Then ensure the rejection message is displayed
 And ensure cash is not dispensed.

Sample: Personas

Kulsoom is ordinary lady, She is a middle-class retiree living on a fixed income. She has been a customer of the bank for 17 years although has never used an automated teller machine (ATM) but is interested in using it.

Name-: Kulsoom

Age-: 58

Marital Status-: Married

Responsibility -: House Hold..

Naseem Ahmad is a self employed and well educated person. He is father of 1 boy. He likes to play/ watch cricket. He also likes travelling and hang around with family

Peter is with NBP from last 10 years.

Name-: Naseem Ahmad

Age-: 52

Marital Status-: Married

Responsibility -: General User of ATM machine within country and outside country as well.

Javaria is an employee of Bank. She has 1 baby girl. She likes to spend time with her family. She also likes to watch movies and read Books.

He is with NBP from last 5 years.

Name-: Javaria

Age-: 35

Marital Status-: Married

Responsibility-: She is working as Customer service representative , in-addition she can generate different requests in Banking system on request of customers.

Amir is working as System Administrator in the Bank.He is father of 2 children's. He likes to play/ watch Football. He also likes to watch movies and read Books.

He is with NBP from last 8 years.

Name-: Amir Ishtiaq

Age-: 42

Marital Status-: Married

Responsibility -: His one of the major responsibility is to perform administration related activities.

Samples

- User Story: https://netsolpk.atlassian.net/browse/MBSAA-1
- Velocity chart: https://netsolpk.atlassian.net/secure/RapidBoard.jspa?rapidView=221&projectKey=MBSAA&view=reporting&chart=velocityChart&useStoredSettings=true
- Burndown chart: https://netsolpk.atlassian.net/secure/RapidBoard.jspa?rapidView=221&projectKey=MBSAA&view=reporting&chart=burndownChart&sprint=91
- Scrum board: https://netsolpk.atlassian.net/secure/RapidBoard.jspa?rapidView=221&projectKey=MBSAA

Estimation and Sizing

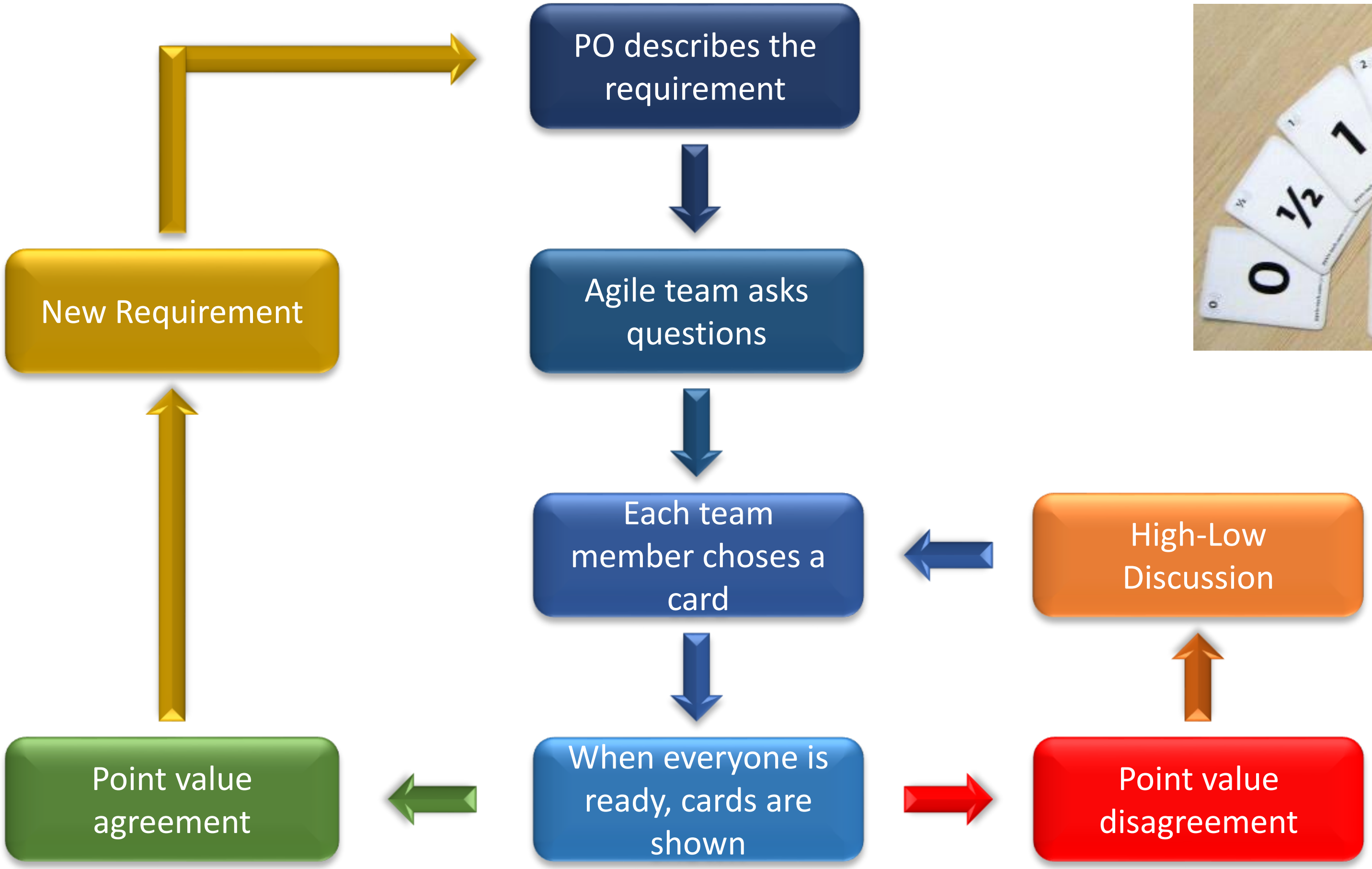
Estimation and Sizing

- Estimation in Agile is done through Story points
  - Story points are an abstract
  - They are used to determine the complexity of a story, not the time the story will take to be developed
  - They offer a quick estimation technique (max 8 minutes / story)

- Story point define the velocity of a team
  - Each team’s velocity is unique to that team
  - Velocity is measured over time (multiple sprints)
  - Velocity is used to do release planning

- Stories are broken down into tasks
  - Tasks are sized by the team in man-hours

Planning Poker



New Requirement

PO describes the requirement

Agile team asks questions

Each team member choses a card

When everyone is ready, cards are shown

Point value agreement

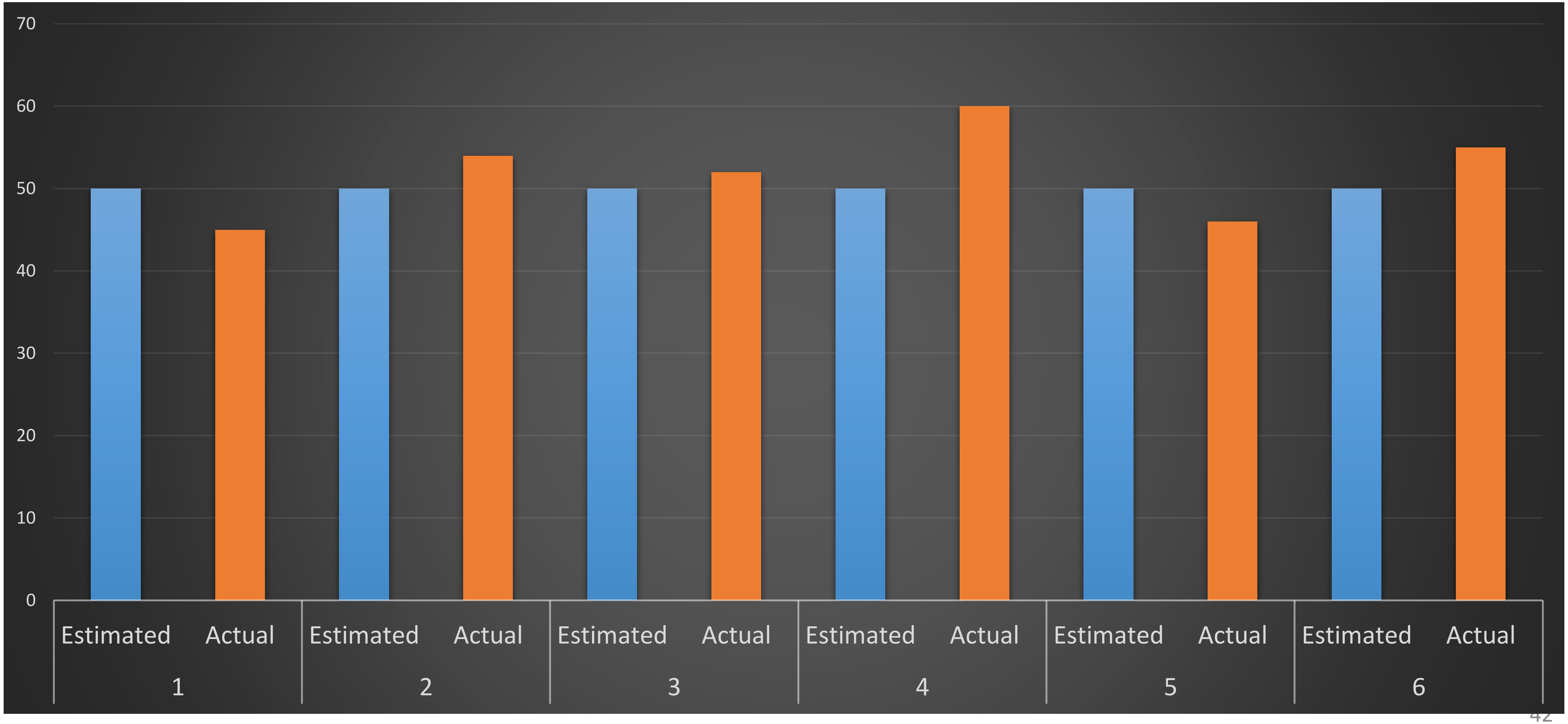
Point value disagreement

High-Low Discussion

Team velocity

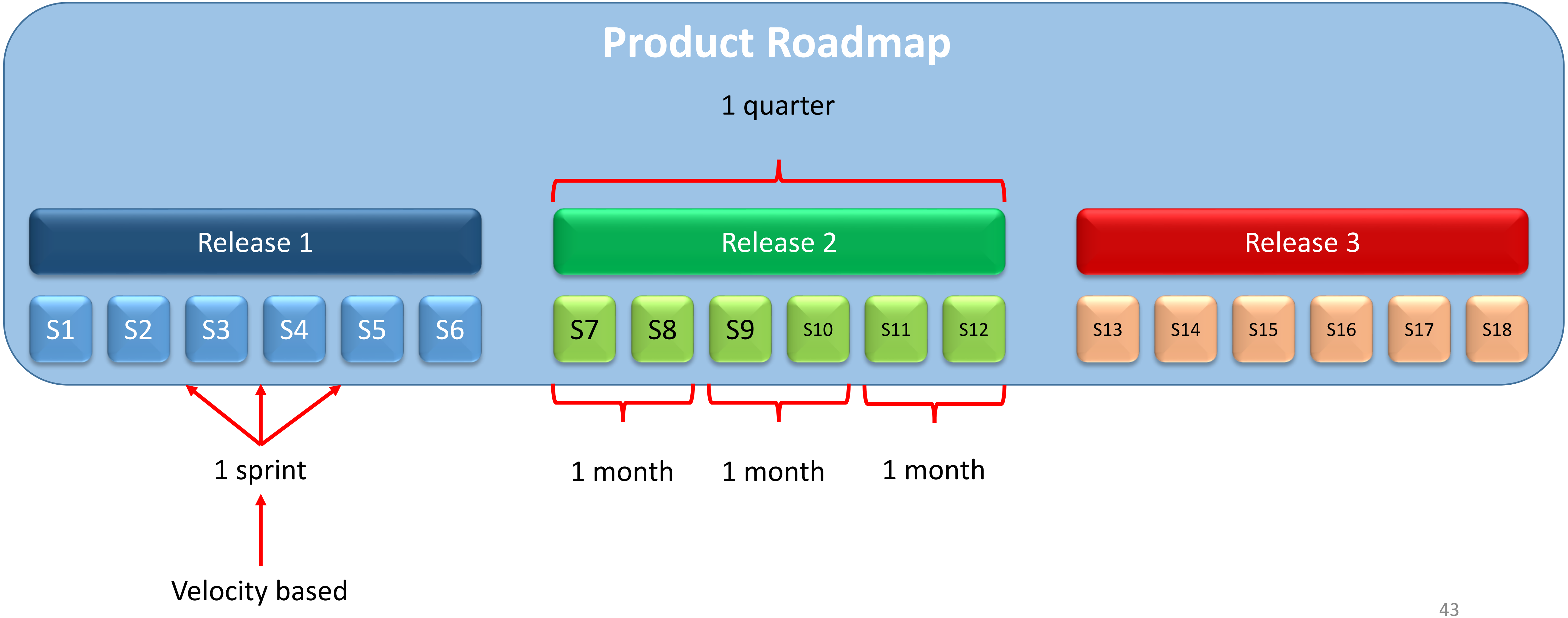
- Velocity is the total of the story points for those stories that were completed during the past sprint
- Stories <u>have</u> to adhere to done-and-done
- Stories not completed do not count towards velocity
  - They are returned to the backlog and velocity is counted in the sprint they are completed
- Velocity is measured over time, I.e. multiple sprints to determine the teams average velocity
- Velocity is unique to a team, never compared across teams

Velocity chart



https://netsolpk.atlassian.net/secure/RapidBoard.jspa?rapidView=221&projectKey=MBSAA&view=reporting&chart=velocityChart&useStoredSettings=true

Roadmap


Product Roadmap
1 quarter
Release 1
Release 2
Release 3
S1
S2
S3
S4
S5
S6
S7
S8
S9
S10
S11
S12
S13
S14
S15
S16
S17
S18
1 sprint
1 month
1 month
1 month
Velocity based

Planning in Agile

5 Levels of planning

Level	Frequency	Main Actors	Output
Product Vision	1-2 times / year	Product Owner (PO)	- Vision
Roadmap	2-3 times / year	PO	- Roadmap - Initial product backlog
Release Plan	3-4 times / year	PO & Team & SM	- Release plan
Sprint Plan	Every sprint	PO & Team	- Sprint backlog
Daily Plan	Every day	Team	- Updated stories - Impediments

Events

The Sprint

Week 1

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
09.00am	Sprint Planning (4 hours)	Daily Scrum – 15 Mins	Daily Scrum – 15 Mins	Daily Scrum – 15 Mins	Daily Scrum – 15 Mins	09.00am
14.00pm						14.00pm

Week 2

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
09.00am	Daily Scrum – 15 Mins	Daily Scrum – 15 Mins	Daily Scrum – 15 Mins	Daily Scrum – 15 Mins	Daily Scrum – 15 Mins	09.00am
14.00pm	Backlog Grooming (2 hours)				Show and Tell (2 hours)	14.00pm
					Retrospective (2 hours)	

🟩 = Team, SC, PO

🟨 = Team & SC, others can observe

🟩 = PO, Stakeholders, others & Team member

🟪 = Team and SC only

Sprint Planning

- Suggested Duration: 4 hours
- Purpose: 2 part meeting to define the work the team will commit to for the coming sprint
- Agenda:
  - Part 1: What?
    - PO presents and clarifies the top priority stories to the team (from the product backlog)
    - PO and team review, discuss and define which ones will be part of the sprint (decided by the team)
    - They agree on the sprint goal
  - Part 2: How?
    - The team identifies tasks relating to each story
    - Stories are estimated (story points) and tasks are sized (ideal man-hours)
    - The sprint backlog is compiled

Moderator:	Product Owner
Master of Ceremonies:	Scrum Master
Participants:	Product owner, Scrum Master, team members.

Daily Scrum

- **Maximum Duration:** 15 Minutes
- **Purpose:** Stand-up is designed to quickly inform everyone of what's going on across the team.
- **Agenda:** Every team member will answer 3 questions
  1. What did you do yesterday?
  2. What will you do today?
  3. Are there any impediments in your way?

Moderator: Scrum Master

Participants: Scrum Master and Team. Product Owner can also attend this meeting but cannot contribute.

Others: Outside participants are allowed but do not contribute

Backlog Grooming

- Maximum Duration: 2 hours
- Purpose: is to clean up the backlog and provide the team with insight in future sprints, as well as provide the PO with in sights into story complexity
- Agenda: PO presents future stories to the team, release planning can take place, sizing can take place, story refinement

Moderator:	Product Owner
Participants:	Product owner, Team and Scrum Master.
Others:	Outside participants are allowed but do not contribute

Sprint review / show and tell

- **Maximum Duration:** 2 Hours
- **Purpose:** is to show what team has planned and accomplished during the sprint.
- **Agenda:** Power point slides and system demo can be used to cover below topics:
    - What was the sprint goal and which stories were selected?
    - How many stories are completed in this sprint?
    - Sprint Burn Down Chart
    - Actual Demo of the system
    - PO will explain what’s coming next based on backlog grooming session during the sprint.

Moderator: Product Owner

Participants: Product owner, Team and Scrum Master. But other stakeholder like management, customers and developers from other projects are encouraged to attend this meeting.

Retrospective

- Maximum Duration: 2 Hours
- Purpose: is to continuously improve productivity of the team.
- Agenda: Every one will answer below questions
  - What went well during the sprint cycle?
  - What went wrong during the sprint cycle?
  - What could we do differently to improve?

- This is an internal, confidential meeting by the team for the team

Moderator: Scrum Master

Participants: Team and Scrum Master.

Scrum-of-scrum

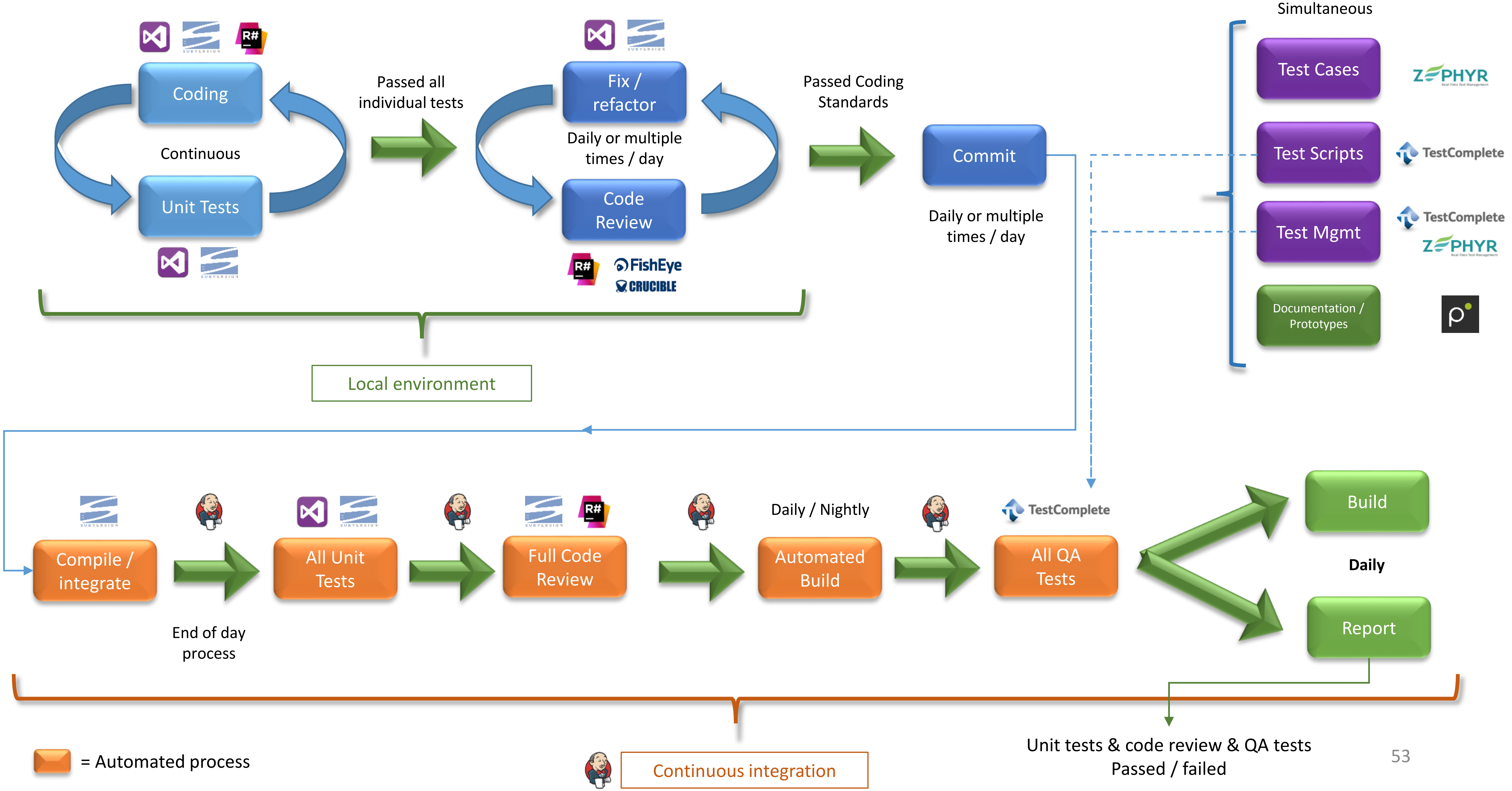
- Suggested Duration: 1 Hour / 2 times per week
- Purpose: alignment across teams / scaling across large project teams
- Agenda: Every one will answer below questions
  - What has your team done since we last met
  - What will your team do before we meet again
  - Is anything slowing your team down or getting in their way
  - Are you about to put something in another team's way?

This meeting is to discuss and resolve problems

Moderator: Scrum Coach or Product Director

Participants: Product owners, Scrum Masters, individual team members.

Development Process (tobe - Scrum)



Coding

Continuous

Unit Tests

Passed all individual tests

Fix / refactor

Daily or multiple times / day

Code Review

Passed Coding Standards

Commit

Daily or multiple times / day

Simultaneous

Test Cases

Test Scripts

Test Mgmt

Documentation / Prototypes

Local environment

Compile / integrate

End of day process

All Unit Tests

Full Code Review

Daily / Nightly

Automated Build

All QA Tests

Build

Daily

Report

Unit tests & code review & QA tests Passed / failed

🟧 = Automated process

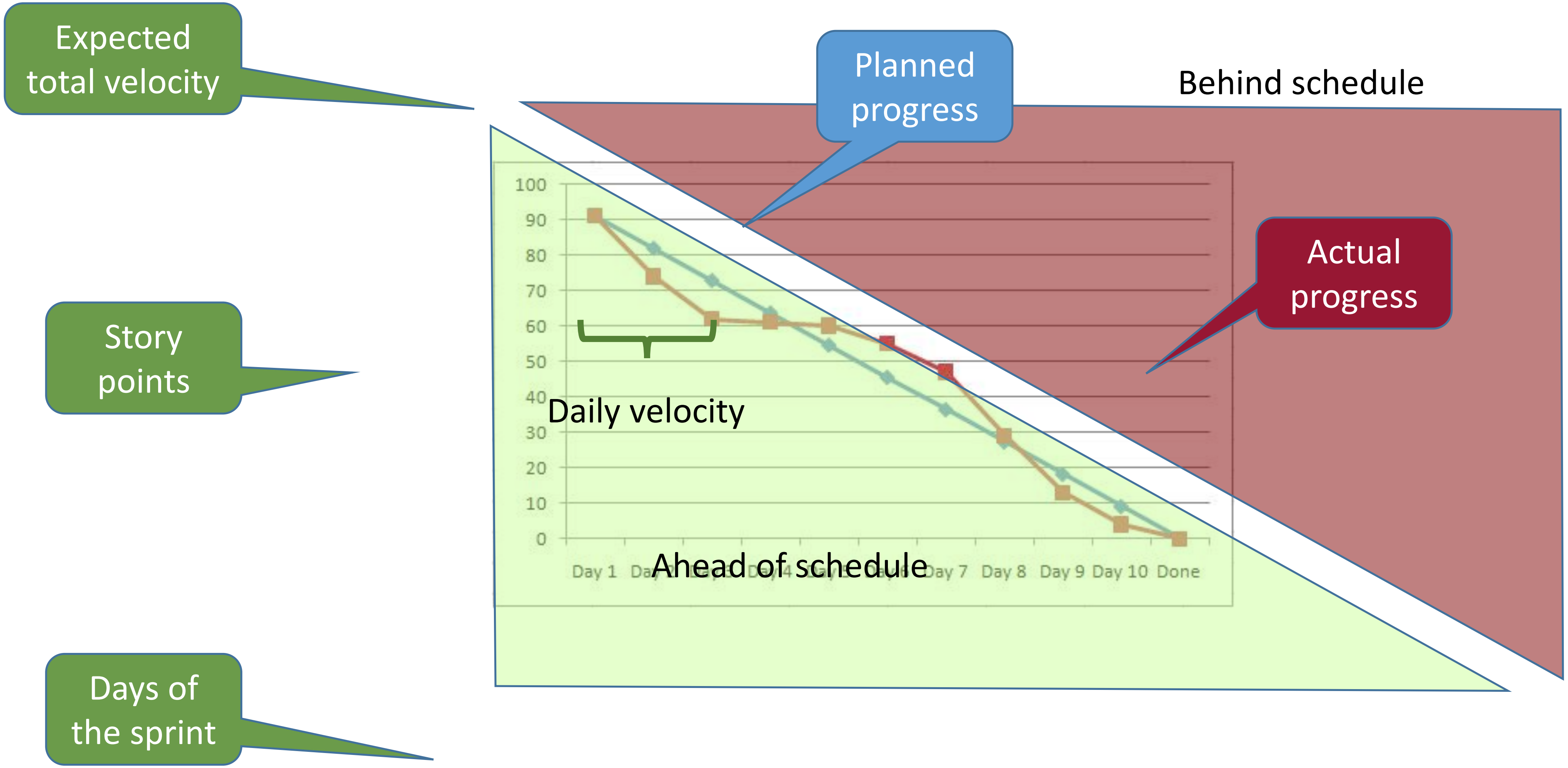
Continuous integration

Information Radiators

Information Radiators

- Visual displays that concentrate important project information for anyone to review:
  - Jira
  - Scrum Board
  - Burn up / down charts
  - Release burn down charts

Sample burn down chart



Expected total velocity

Story points

Days of the sprint

Planned progress

Behind schedule

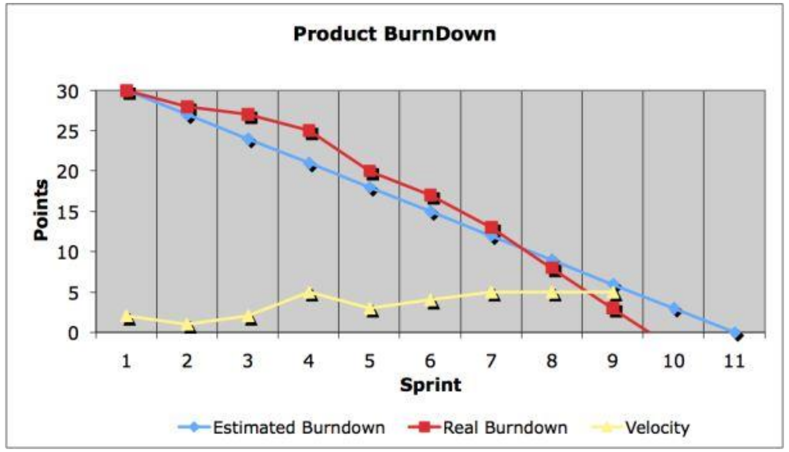
Actual progress

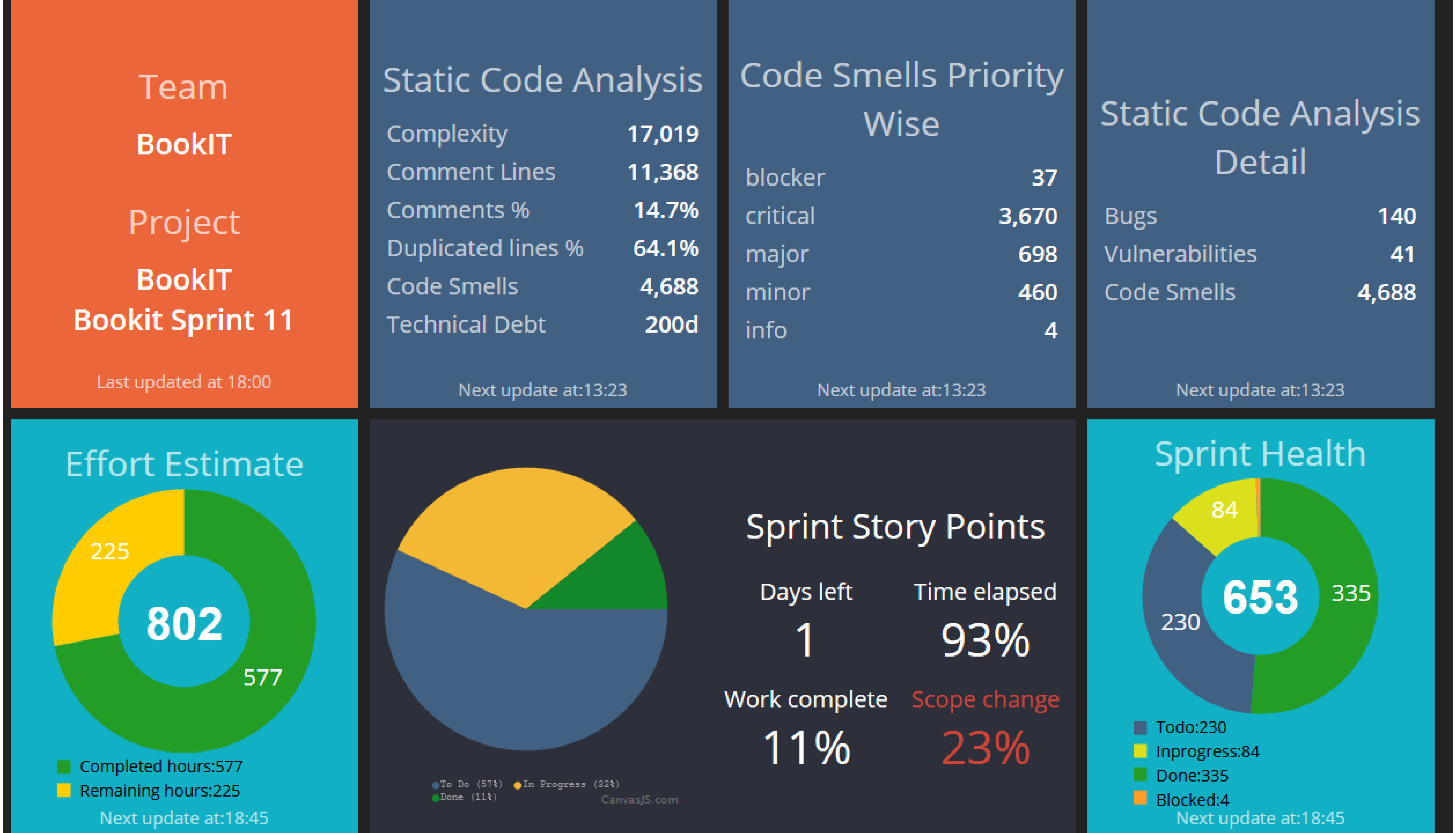
Daily velocity

Ahead of schedule

https://netsolpk.atlassian.net/secure/RapidBoard.jspa?rapidView=221&projectKey=MBSAA&view=reporting&chart=burndownChart&sprint=91

Sample Product burn down chart





Team

BookIT

Project

BookIT
Bookit Sprint 11

Last updated at 18:00

Static Code Analysis

Complexity	17,019
Comment Lines	11,368
Comments %	14.7%
Duplicated lines %	64.1%
Code Smells	4,688
Technical Debt	200d

Next update at:13:23

Code Smells Priority Wise

blocker	37
critical	3,670
major	698
minor	460
info	4

Next update at:13:23

Static Code Analysis Detail

Bugs	140
Vulnerabilities	41
Code Smells	4,688

Next update at:13:23

Effort Estimate

🟩 Completed hours:577
🟨 Remaining hours:225

Next update at:18:45

Sprint Story Points

Days left	Time elapsed
1	93%
Work complete	**Scope change**
11%	23%

Sprint Health

▨ Todo:230
🟨 Inprogress:84
🟩 Done:335
🟧 Blocked:4

Next update at:18:45

Thank You

Useful information

Useful Information

Agile:

https://www.scrumalliance.org/ - Agile Scrum

www.agilemanifesto.org/ - The Agile Manifesto

http://agilemethodology.org/ - Agile Scrum

https://www.agilealliance.org/ - Agile Scrum

http://disciplinedagiledelivery.com/ - D.A. 2.0

https://www.mountaingoatsoftware.com// - Agile Scrum

https://www.atlassian.com/agile/kanban/ - Agile Kanban

http://www.scaledagileframework.com/ - SAFe

These websites cover the various frameworks, role definitions, responsibilities, team structures, planning, etc. from small sized organisations to large sized organisations.

Specific to story points:

https://agilefaq.wordpress.com/2007/11/13/what-is-a-story-point/

https://www.scrumalliance.org/community/articles/2012/august/story-points-versus-task-hours
https://www.scrumalliance.org/community/spotlight/mike-cohn/june-2014/how-many-hours-is-a-story-point-worth

User Stories

- http://www.romanpichler.com/blog/10-tips-writing-good-user-stories/
- https://www.scrumalliance.org/community/articles/2014/march/stories-versus-themes-versus-epics
- http://searchsoftwarequality.techtarget.com/definition/user-story
- http://agileforall.com/new-to-agile-invest-in-good-user-stories/
- http://ronjeffries.com/xprog/articles/expcardconversationconfirmation/
- https://www.mountaingoatsoftware.com/uploads/articles/User-Stories-Applied-Mike-Cohn.pdf
- http://www.romanpichler.com/blog/epics-and-ready-stories/
- https://help.rallydev.com/writing-great-user-story
- http://www.agilemodeling.com/artifacts/userStory.htm
- https://www.scrumalliance.org/community/articles/2013/september/agile-user-stories
- http://www.boost.co.nz/blog/2010/09/acceptance-criteria/
- http://nomad8.com/acceptance_criteria/
- http://www.seguetech.com/blog/2013/03/25/characteristics-good-agile-acceptance-criteria
- https://www.mountaingoatsoftware.com/agile/user-stories