

Your fastest way to learn. Why wait?



Firebrand's training to prepare you for Scrum.org's Professional Scrum Master Certification

Courseware Slides & Notes

Version 2.5

Scrum Master Foundations

Theory, Practice & Assessment

1



Introductions

- Your name
- Your role
- Your experience
- Your expectations
- Your motivation



2



Course Structure

- This evening
 - Introduction, orientation and preparatory reading
- Tomorrow
 - Scrum Theory & Practice
 - Scrum Tools,
 - Approaches & Exercises
- Final Day
 - Scaling Scrum
 - Practice Exercises (if time available)
 - SCRUM.org PSM I Examination

3



Scrum Master Foundations

Part One - Theory

4



History

- Systems Engineering
- The rise of 'Agile'
- Scrum as part of the Agile 'family'

5



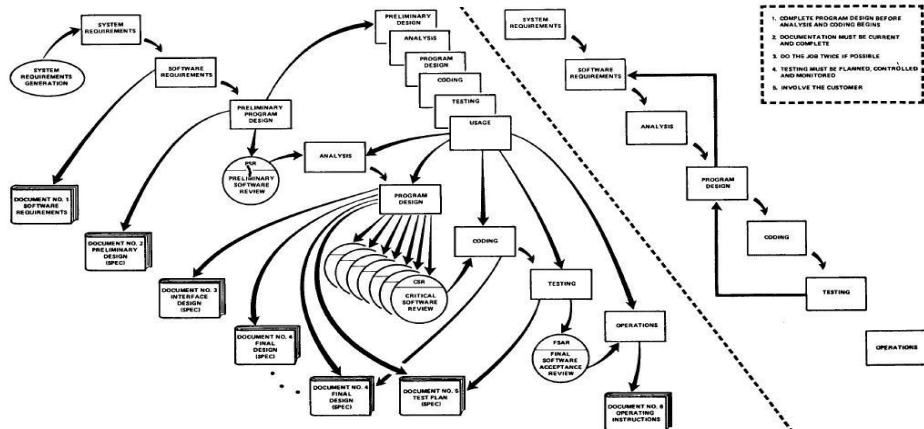
Waterfall

- Royce paper of 1970 is the classic expression of the systems engineering approach to software development
- Emphasis in a sequence of steps, each manageable in scope, refining the requirement, design, coding etc... and generating predefined outcomes.

6



Waterfall



10252016

7

Reaction to Waterfall

- Became apparent during 1980s and into 1990s that the traditional approach was too rigid and unresponsive to changes or ambiguity in requirements
- A number of approaches began to appear; RAD, JAD, Extreme Programming, DSDM

10252016

8

Agile

- A class of approaches to software development that emphasise
 - Repeated cycles (iterations)
 - Small delivery steps (increments)
 - Evolutionary requirements & solutions
 - Self-organizing & cross-functional teams

9



Agile Manifesto

- 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 **processes and tools**
 - **Working software** over **comprehensive documentation**
 - **Customer collaboration** over **contract negotiation**
 - **Responding to change** over **following a plan**
- That is, while there is value in the items on the right, we value the items on the left more

10



Scrum

- Origins of 'Scrum' in product development
- The Scrum Guide and scrum.org
- Scrum Theory & Scrum Framework
- Roles, Rules, Events & Artefacts

11



Origin of the 'Scrum' Concept

- "The New Product Development Game: Stop Running The Relay Race And Take Up Rugby"
- Hirotaka Takeuchi and Ikujiro Nonaka, Harvard Business Review, Jan/Feb 1986
- A report on new product development at companies such as Xerox, Canon, Honda, Brother, and Hewlett-Packard

12



Origin of the 'Scrum' Concept

- Six characteristics identified in this new development process:
 - Built-in instability
 - Self-organizing project teams
 - Overlapping development phases
 - “Multilearning”
 - Subtle control
 - Organizational transfer of learning

13



OOPSLA '95 & After

- A presentation by Ken Schwaber and Jeff Sutherland at the OOPSLA '95 conference
- Scrum Alliance formed in 2004
- Scrum.org founded 2009

14



Scrum Guide

- Available from www.scrumguides.org
- Released by Ken Schwaber and Jeff Sutherland in October 2011
- Current version **July 2013**
- “The Definitive Guide to Scrum: The Rules of the Game”

10252016

15



Scrum Theory

- “Scrum is founded on empirical process control theory...”
- If a process can be fully defined, with all things known about it so that it can be designed and run repeatedly with predictable results, it is known as a Defined Process and can be automated.

10252016

16



Empirical Process

- If a process can not be fully defined it can only be treated as a 'black box'.
- All that the can be known are inputs to and outputs from the process with its inner mechanisms unknowable.
- Such a process is called 'Empirical'

17



Empirical Process

- An Empirical Process can appear 'chaotic' and requires close monitoring and control at frequent intervals
- Scrum accepts that the (software) development process is unpredictable and therefore applies iterative, incremental, empirical controls that allow it to be 'agile'
- The desire is to overcome the inherent complexity of a situation by seeking to impose predictability upon it; to treat an empirical process as a defined one
- Agile Thinking accepts the complexity and complication of the development process and seeks to respond to it

18



Scrum vs. Traditional Methods

When to use Scrum	When to use traditional methods
<ul style="list-style-type: none"> • Scope is not clearly defined at project start • The (final) product will be defined during the project 	<ul style="list-style-type: none"> • Scope is clearly defined upfront. Clear product description is available upfront • Similar projects were done before
<ul style="list-style-type: none"> • Change of requirements will be very likely • Customer learns more about what he wants as the project matures 	<ul style="list-style-type: none"> • Requirements are well defined up front. • Just few changes are expected during the project • Products are not expected to change during project execution (with formal change process)
<ul style="list-style-type: none"> • Activities to produce scope cannot be well defined upfront • Effort and duration estimating is difficult 	<ul style="list-style-type: none"> • Activities to produce scope can be well defined upfront • Estimating is possible and reliable
<ul style="list-style-type: none"> • Process is iterative (numerous cycles) • Each cycle depends on the results and experience of the previous ones 	<ul style="list-style-type: none"> • Process is more long term • Project might be split into phases
<ul style="list-style-type: none"> • Success is mostly measured by customer satisfaction 	<ul style="list-style-type: none"> • Success is mostly measured by achieving the project goals for time, cost, scope...
<ul style="list-style-type: none"> • Incremental results have value and can be used by users 	<ul style="list-style-type: none"> • Users normally do not start using the products until the project is complete (e.g. a bridge)

19/03/2016

19



Scrum - fiction and facts

Fiction	Fact
<ul style="list-style-type: none"> • Developers are free to do what they want. • There is no discipline, structure 	<ul style="list-style-type: none"> • Developers work in a productive and predefined framework • Scrum Master makes sure everyone follows Scrum. • Scrum affords a high discipline by all participants
<ul style="list-style-type: none"> • Scrum gets rid of all paper work and allows the team to start developing right away. 	<ul style="list-style-type: none"> • The paper work should be reduced to what is absolutely necessary • Scrum has re-are defined planning steps involved in every Scrum project • Development can only start when the Sprint Backlog has been defined.
<ul style="list-style-type: none"> • All requirements (User Stories) must be agreed before the Development Team is allowed to start working on the product. 	<ul style="list-style-type: none"> • The Development Team can start working as soon as there is an initial Product Backlog defined. • The content of this initial Backlogs creates value.
<ul style="list-style-type: none"> • Scrum is very easy to implement, even without training. 	<ul style="list-style-type: none"> • Using Scrum is a big change in an organisation and the mind set of all participants. People doing Scrum must have a very good understanding of Scrum to be able to run their projects well.

19/03/2016

20



Scrum - fiction and facts

Fiction	Fact
<ul style="list-style-type: none"> The Scrum Master is like a project manager. 	<ul style="list-style-type: none"> There is no role similar to a traditional project manager in Scrum. The Scrum Master makes sure the Scrum framework is followed. The role of a Scrum Master is more of a "Process Manager"
<ul style="list-style-type: none"> Scrum does not require a Business Case. 	<ul style="list-style-type: none"> There should be a justified and documented reason WHY money and effort should be spent. The Product Owner is responsible for ensuring that there is a clear VISION of a feasible product/result. The PO must know its value.
<ul style="list-style-type: none"> Scrum allows the Development Team to decide what will be delivered. 	<ul style="list-style-type: none"> A Development Team only decides on HOW to deliver It is the Product Owner to determine WHAT will be delivered.

21



Scrum - fiction and facts

Fiction	Fact
<ul style="list-style-type: none"> The Product Owner is the project manager. 	<ul style="list-style-type: none"> The Product Owner creates and maintains the Product Backlog, but does not manage the day to day activities of the Team.
<ul style="list-style-type: none"> Scrum tells us everything about managing projects. 	<ul style="list-style-type: none"> Scrum mostly deals with the definition and delivery of the products. Many of the business oriented aspects of the project are done outside Scrum.
<ul style="list-style-type: none"> The Product Owner is a representative from the customer. 	<ul style="list-style-type: none"> The Product Owner is one of the people from the performing organization (the organization in charge of producing the final product of the project; a contractor in many cases), and the contact point with the customer.
<ul style="list-style-type: none"> Scrum can only be applied with in-house software development projects 	<ul style="list-style-type: none"> Scrum as a framework can be applied to almost every project that is not completely predictable.

22



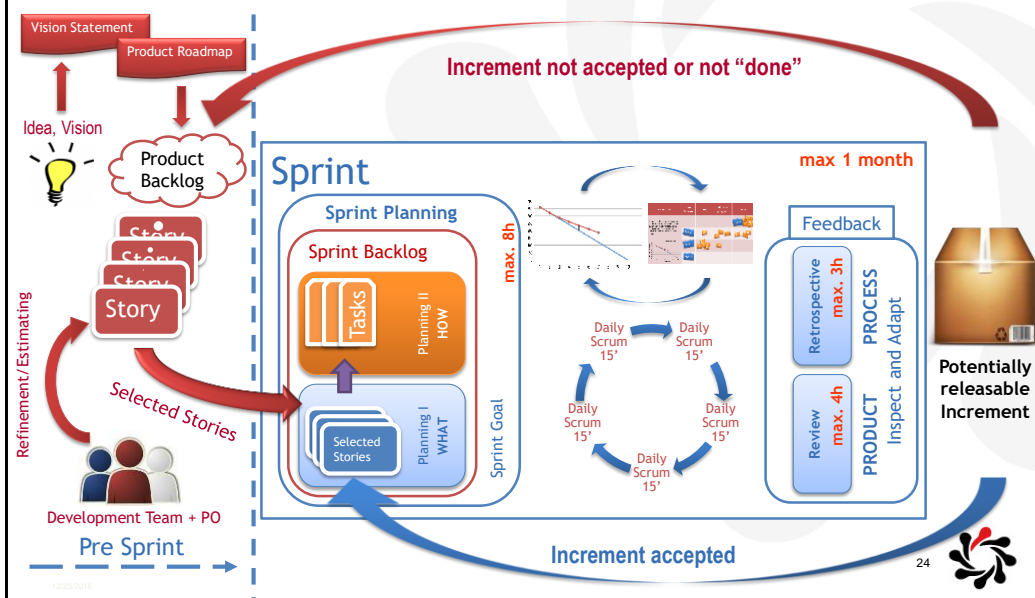
5 Scrum Values

- ❑ Commitment
 - People personally commit to achieving the goals of the Scrum Team.
- ❑ Courage
 - The Scrum Team members have courage to do the right thing and work on tough problems.
- ❑ Focus
 - Everyone focuses on the work of the Sprint and the goals of the Scrum Team.
- ❑ Openness
 - The Scrum Team and its stakeholders agree to be open about all the work and the challenges with performing the work.
- ❑ Respect
 - Scrum Team members respect each other to be capable, independent people

23



Scrum - Flow



24



Scrum

□ What happens prior to Sprints:

- The Vision Statement provides a concise description of the strategic objectives of the project to help the team keeping the focus
- The Product Roadmap is an initial and visual timeline of major product features to be delivered and is normally created by the Product Owner
- User requirements are gathered and turned into User Stories, normally written by the Product Owner based upon customer (user) requirements.
- The User stories make up the Product Backlog. The Product Backlog does not need to be 100% complete to start the Sprints
- Sprints are being started as soon as the Product Backlog is mature enough for the near future. The Product Backlog is being updated all the way through the project.

25



Scrum

□ What happens prior to Sprints:

- The Product Backlog must be estimated (by Development Team). The Product Owner supports as he/she knows best its content based on user requirements
- The Product Backlog (User Stories) must be refined prior to the Sprints. This is done by the Development Team together with the Product Owner initial before the first Sprint starts, and is done repeatedly and iterative in advance to the Sprints for those User Stories that should be selected for the next Sprints.

26



Scrum

□ Sprint Activities:

- Sprint Planning defines WHAT will be done within a Sprint and HOW it will be done. The Product Owner prioritises these User Stories (requirements) and therefore decides indirectly on the contents of the Sprint Backlog.
- The Team breaks down (expands) these stories into tasks.
- These User Stories (features, functionalities, or deliverables) AND the Tasks (the work to deliver) make up the Sprint Backlog, a list of everything that will be done and delivered during the next sprint.
- The Development Team then takes up to a maximum of one calendar month to deliver an agreed amount of stories.

27



Scrum

□ Sprint Activities:

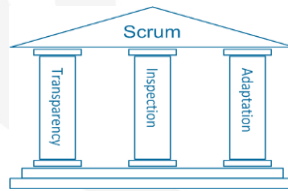
- The Team holds a Daily Scrum meeting of 15 minutes each day to collaborate with each other.
- At the end of the Sprint, the Team demonstrates an increment of the product (the completed stories) to the customer and stakeholder in a Sprint Review meeting.
- The last activity is the Scrum Retrospective meeting, where the team reviews the Sprint processes and looks for ways of improvement (lessons learned).
- The Scrum Master makes sure the Scrum process is followed entirely and offers coaching to everyone involved.

28



Three Pillars

- “Three pillars uphold every implementation of empirical process control:
 - Transparency
 - Inspection
 - Adaptation
- That is the centrality of communication, review and improvement



29



Scrum Framework

Roles, Events, Artefacts

30



Scrum Framework

- ❑ Scrum is not a process or a technique, but a framework within which processes and frameworks can be deployed more effectively
- ❑ It consists of roles, rules, events and artefacts

31



Scrum Roles

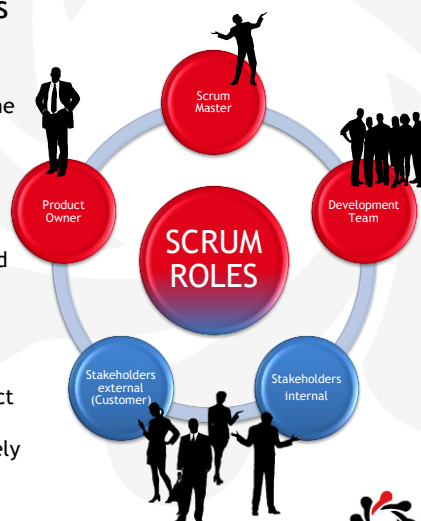
- ❑ The Roles in Scrum are defined as

- **The Scrum Team**

- Self-organizing; chooses how to accomplish the work without external direction
- Cross-functional; has all the competencies necessary within it to accomplish its work without depending on others outside
- Designed to optimise flexibility, creativity and productivity

- **(Stakeholders)**

- persons external to the Scrum Team with a specific interest in and knowledge of a product that is required for incremental discovery. Represented by the Product Owner and actively engaged with the Scrum Team at Sprint Review.

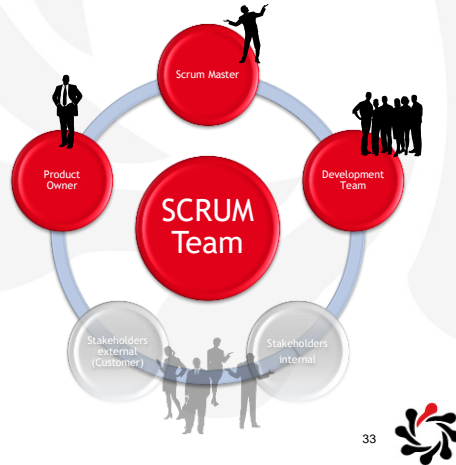


32



Scrum Team

- Responsible for delivering increments of 'Done' product; potentially useful versions of working product
- Has three sub-roles
 - Product Owner
 - Developer
 - Scrum Master



Product Owner

- The individual responsible for maximizing the business value of the work of the Development Team and of the products they produce
- Sole responsibility for requirements and their priorities
- Owns the Product Backlog
- They also measure the performance of the project, forecast the completion date, and make this information transparent to all stakeholders.



Development Team

- Joint responsibility for the development of 'Done' Increments in product
- 'Developer' is the only title in the team
- No one tells the team how to organise its work
- The ideal size for the Development Team is 6 ± 3 members



35



Scrum Master

- The Scrum Master is a servant-leader for the Scrum Team. His/her role is to facilitate Scrum practice within the Scrum Team and within the larger organization
- The Scrum Master is a management position, he/she manages the Scrum process, rather than the Scrum Team.
- The Scrum Master is not the manager of the Development Team which is self-organising
- He/she should also help those outside the Scrum Team understand the appropriate interactions with the Scrum Team to maximise the value created by the Scrum Team. The Scrum Master usually leads the organisation in its effort to adopt Scrum.



36



Scrum events

- Each event is time-boxed; it has a maximum duration
- Each event is an opportunity to provide transparency and to inspect and adapt products and processes
- Scrum Events (within a Sprint)
 - Sprint Planning
 - Daily Scrum
 - Sprint Review
 - Sprint Retrospective



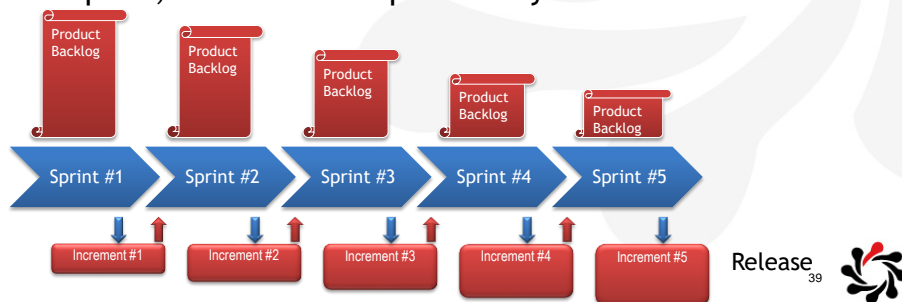
Sprint

- The Sprint event is the container for all other events; the definition and design of what is to be built, the plan and work for building it and the resultant product
- It is time-boxed at a one-month maximum, less if risk or circumstances dictate, but their duration should be consistent across the project
- The scope of a Sprint can be clarified and re-negotiated between Product Owner and Development Team, but no changes are made that would alter the Sprint Goal



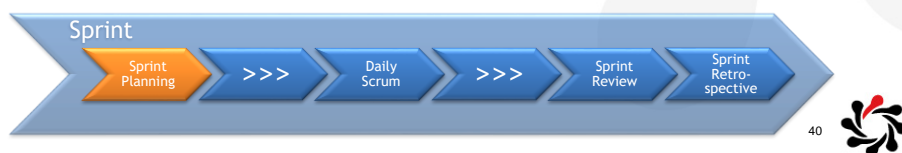
Sprint

- ❑ A Sprint can be cancelled, but only by the Product Owner and only if the Sprint Goal becomes obsolete
- ❑ Each Scrum project delivers the final product (Release) after a number of Sprints.
- ❑ An Increment may or may not be released at the end of the Sprint, but it must be potentially releasable.



Sprint Planning

- ❑ The meeting is proportionately time-boxed at 2 hours per week of Sprint (i.e. 8 hours for a 4 week Sprint)
- ❑ The whole Scrum Team collaborates
- ❑ The meeting is divided into two, equally time-boxed, sessions; the first covering 'what' and the second 'how'



Sprint Planning I - “What“

- The Development Team forecasts the functionality that will be developed during the Sprint based on
 - The ordered Product Backlog items
 - The latest product Increment
 - The projected capacity and past performance of the Development Team

41



Sprint Goal

- Identified as part of the first part of the Sprint Planning Meeting
- Defines the business objective that will be met, within the Sprint, through the implementation of the selected Product Backlog items; it defines the value of building the Increment

This is a sample Sprint Goal:

“The sprint goal is to make the purchasing part of the website mature enough, so that customer are able to handle the entire purchasing process, which brings the website in a functional status to create revenue. “

42



Sprint Planning II - “How“

- The Development Team decides how to build the selected functionality into ‘Done’ Increment and plans the work in sufficient detail to forecast what it can do during the Sprint by decomposing the User Stories into Tasks
- The Product Owner and Scrum Master may be present, but only to assist and not direct

A single Backlog Item (User Story)

Story Title	Payment Gateway
ID	113
Estimate	7
Importance	80
Demo	When user have something in the cart and clicks on the check-out button, they will be one; they would be able to pay, and will then be directed to the thank you page
Tasks	113-1, 113-2, 113-3, 113-4

Decomposition



Tasks needed for getting the story “done”
A plan for developing the item

43



Daily Scrum

- A daily, 15-minute meeting at which the Development Team synchronises work, inspects and adapts products and processes, and creates a plan for the next 24 hours
- The meeting is for the Development Team, it is not a status meeting for the Scrum Master or Product Owner



44



Daily Scrum

- ❑ At the Daily Scrum each Developer answers the ‘three questions’
 - What has been accomplished since the last meeting
 - What will be done before the next meeting
 - What obstacles are in the way
- ❑ Each task completed must be ‘Done’ in order to be completed; i.e., the code must be ‘clean’
- ❑ The code must be ‘built’ each day; the ‘Done’ of the code must be transparent
- ❑ Daily Scrum is not a problem-solving meeting; any issues raised are addressed outside the meeting (usually immediately afterwards)



10255004

Daily Scrum

- ❑ Task Board to be updated during the Daily Scrum

Sprint Goal	User Stories	Tasks	Work in progress	Done
<p>The sprint goal is to make the purchasing part of the website mature enough, so that customer are able to handle the entire purchasing process, which brings the website in a functional status to create revenue.</p>	<p>Story #1</p> <p>Story #2</p> <p>Story #3</p> <p>Story #4</p>	<p>T 2.7</p> <p>T 3.5</p> <p>T 3.2</p> <p>T 3.6</p> <p>T 3.4</p> <p>T 3.3</p>	<p>T 2.4</p> <p>T 2.5</p> <p>T 2.6</p> <p>T 3.1</p>	<p>T 1.3</p> <p>T 1.4</p> <p>T 1.5</p> <p>T 1.6</p> <p>T 2.1</p> <p>T 2.2</p> <p>T 2.3</p>



10255004

Sprint Review

- ❑ The Scrum Team and stakeholders meet to accept completed work and identify new requirements
- ❑ This is an opportunity for demonstration and feedback not judgement; acceptance testing should have been completed prior to the meeting
- ❑ The meeting is proportionately time-boxed at 1 hour per week of Sprint (i.e. 4 hours for a 4 week Sprint)



47



Sprint Review

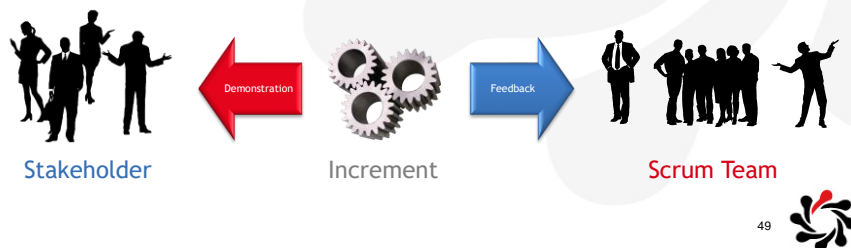
- ❑ The whole Scrum Team collaborates with Stakeholders on revising the Product Backlog based on the output of the Sprint and the feedback received from the customer to identify what was done during the Sprint and on the things that can be done in the next one
- ❑ Present and inspect the “Done” items (the Increment) from the current Sprint and adapt the Product Backlog by marking off “Done” items as complete and add new items or change the existing ones if necessary.

48



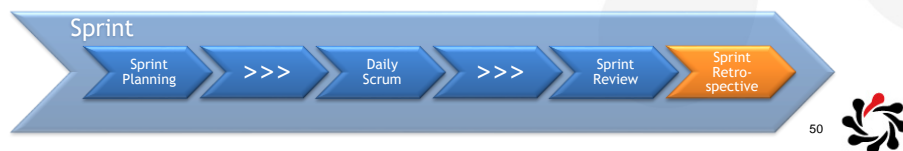
Sprint Review

- ❑ Changes are welcome as they may increase the satisfaction of the customer and will create a final product that better matches the needs of the customer.
- ❑ The Product Owner discusses the status of the Product Backlog and the likely completion dates based on the progress.



Sprint Retrospective I

- ❑ This meeting follows the Sprint Review and is time boxed at ¼-hour per week of Sprint
- ❑ The Scrum Master facilitates the Scrum team in inspecting and adapting itself
- ❑ Always look for ways to improve. This meeting is a formal opportunity for improvement, even though we do not limit our improvement to the results of this meeting. Review (inspect) the Sprint, with regards to people, relationships, processes, and tools, and identify ways of improving them in the next Sprint.



Sprint Retrospective II

- Set the stage
- Gather data
- Generate insights
- Decide what to do
- Close the retrospective

51



Scrum Artefacts

- The purpose of the Artefacts is to promote transparency, to provide mechanisms to enhance communication and mutual understanding
 - Product Backlog
 - Sprint Backlogs
 - Increment
- Not an official artefacts in Scrum Guide™
 - Definition of Done
 - Monitoring tools (Task Board, Sprint Burn-Down, Release Burn-Down)

52



Product Backlog

- ❑ “...an ordered list of everything that might be needed in a product...”
- ❑ “...the single source of requirements for any changes to be made to the product”
- ❑ It is never complete, a “living artefact”
- ❑ Content is defined by User-Stories

53



Sprint Backlog

- ❑ The Sprint Backlog
 - consists of the following:
 - Selected items from the Product Backlog, to be delivered through the Sprint
 - A detailed plan for turning the selected items into “Done” Increment of the product and to realise the Sprint Goal (the plan would not be completely detailed in the Sprint Planning meeting)
 - “...defines the work the Development Team will perform to turn Product Backlog items into a ‘Done’ Increment”
 - It evolves during the Sprint
- ❑ Keep in mind: The Sprint goal is defined during Sprint Planning BUT is NOT part of the Sprint Backlog!

54



Sprint Backlog

- ❑ The Sprint Backlog is frozen after the Sprint Planning regarding selected User Stories
- ❑ Development Team will focus on delivering an Increment of “Done” based on this plan. Items (stories) in the Sprint Backlog cannot be added or removed during the Sprint.
- ❑ However, it might be necessary to re-negotiate, justify, or clear some of the items during the Sprint, which should be done in the presence of the Product Owner.
- ❑ The detailed plan which is normally not complete at the end of the Sprint Planning will become more complete as the Sprint continues.

55



Sprint Backlog

- ❑ The Sprint Backlog is represented e.g. in the Task Board by the selected User Stories and the referring Tasks as a plan for the work to meet the Sprint Goal

Sprint Goal	User Stories	Tasks	Work in progress	Done
The sprint goal is to make the purchasing part of the website mature enough, so that customer are able to handle the entire purchasing process, which brings the website in a functional status to create revenue.	Story #1	T 1.1, T 1.2, T 1.3, T 1.4, T 1.5, T 1.6		
	Story #2	T 2.1, T 2.2, T 2.3, T 2.4, T 2.5, T 2.6, T 2.7		
	Story #3	T 3.1, T 3.2, T 3.3, T 3.4, T 3.5, T 3.6		
	Story #4	T 4.1, T 4.2, T 4.3, T 4.4		

56



Increment

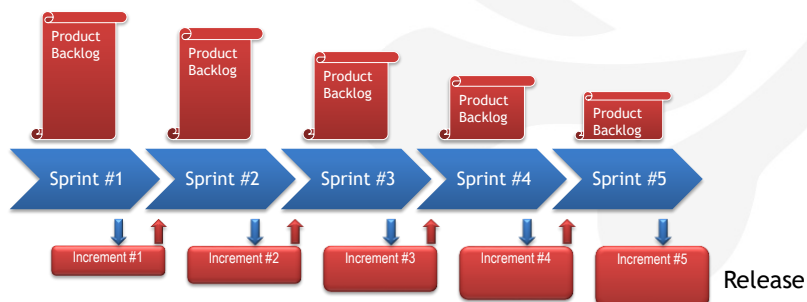
- “The Increment is the sum of all the Product Backlog items completed during a Sprint and all previous Sprints”
- “The purpose of each Sprint is to deliver Increments of potentially releasable functionality that adhere to the Scrum Team’s current Definition of ‘Done’”

57



Increment

- The number of stories in the Product Backlog decreases Sprint by Sprint, as the number of features in the Increments increases.
- Note that the Increment concept is cumulative: each Increment also contains the features of the previous ones.



58



Scrum Master Foundations

Part Two - Practice

59



User Stories & Product Backlog

- A User Story is a simple statement, in the language of the system user, of an element of system functionality that would be of value to the user
- Stories can be prioritised, estimated, inspected and adapted

60



User Stories

- A User Story represents a requirement, it does not document it
- The details of the requirement are determined through discussions between users and developers and recorded as tests that show that the functionality has been delivered

61



The Tree Elements - “C’s”

- Card; the short, simple statement of the requirement (e.g. “A user can place an item in a basket”)
- Conversation; the discussions between users and developers
- Confirmation; the tests that will be applied to show that the functionality has been delivered

62



Structure of a 'Good' Story

- ❑ It should specify a major goal that a user has for interacting with the system
- ❑ It should provide an element of end-to-end functionality, lead to a meaningful outcome, allows the user to accomplish something
- ❑ Uses the 'active voice'; "I, as a (role) want to (function) so that (value)"

63



Structure of a 'Good Story'

- ❑ Follows the INVEST acronym
 - Independent
 - Negotiable
 - Valued
 - Estimable
 - Small
 - Testable

64



Epic

- A story that is too big to be delivered in a single Sprint
- A 'compound' epic comprises smaller stories and should be 'split'
- A 'complex' epic cannot easily be split and so should be 'spiked'

65



Constraint

- A 'Constraint' is a User Story that must be obeyed rather than implemented
- It is not estimated or scheduled because it does not represent work to be done but limitations within which work is carried out and functionality delivered

66



Gathering Stories

- Ensuring wide points-of-view
 - User-Role Modelling (what would a 'customer' want to do with the system?)
 - Personas (what would 'Mary, full-time working single mother' want to do with the system?)
 - Extreme Characters (what would a 'serial armed robber' want to do with the system?)



10252016

Gathering Stories

- Techniques
 - User Interviews, Questionnaires, Observation, Story-Writing Workshops
- User Proxies
 - User Manager, Development Manager, Domain Experts, Salesperson, Marketer, Former User, Customer, Support Technician, Business Analyst



10252016

Product Backlog

- ❑ Items that can be “Done” within a single Sprint are called “ready” or “actionable”
- ❑ Refinement or “Grooming” is the process of adding detail, order and estimates to items so that they become “ready”. It is an on going activity and is not time boxed! Should consume not more than 10% of the Development Team’s time.
- ❑ A single Product Backlog is used even when the project consists of multiple teams



69

Prioritization

- ❑ There are four factors that can be used by the Product Owner to prioritise work
 - Value
 - Cost
 - Knowledge
 - Risk



70

Prioritization

- Requirements (User Stories) can be prioritised using the MoSCoW Model:
 - **M**ust Have - must be done, the strategic objective of the project cannot be reached without it. Cannot be negotiated -> **PRIORITY 1**
 - o
 - **S**hould Have - should be done under “normal” circumstances. Can be negotiated if circumstances afford this -> **PRIORITY 2**
 - **C**ould Have - could be done if time and budget allows. Can be negotiated first if circumstances afford this -> **PRIORITY 3**
 - o
 - **W**on't Have - out of scope. Will not be done in the project/product

71



Product Backlog - Exercise

- Exercise 1:
 - **Part A:** Create Product Backlogs of User Stories for the following product development projects:
 - A promotional box of matches
 - A coat for a large dog
 - A dating web site for Scrum Masters
 - An android app for identifying impossible project deadlines
 - **Part B:** Prioritise the one of the Backlogs using the MoSCoW model
 - **Part C:** Create acceptance tests for each story in the prioritised Backlog
 - **Part D:** Create an outline Release Plan for the prioritised Backlog

72



Product Backlog

- ❑ Each Product Backlog item also has a work estimate
- ❑ These estimates are solely done by the Development Team
- ❑ They are used in comparison to the capacity of the Development Team in a single Sprint, to determine the number of items that will be selected for that certain Sprint.
- ❑ Many additional information might be added to each item to help the Scrum Team take control.

73



Estimating

- ❑ Estimates of effort and duration can be based on three basic techniques
 - Comparative; tasks are compared with previous similar tasks
 - Analytic; tasks are decomposed into sub-tasks small enough to estimate duration directly
 - Parametric; tasks durations are derived from size parameters

74



Estimating From Size

- Scrum uses estimates based on task size for Release and Sprint Planning and estimates based on analysis for Daily Planning
- User Stories can be sized in one of two ways
 - Story Points
 - Ideal Days

75



Story Points

- Story Points are an abstract, relative expression of the size of a story compared to a “reference story”
- The points are abstract in that they do not relate to any physical characteristic of a story; they should be an expression of an amalgamation of effort involved, complexity, risk etc...
- They are relative in that a story assigned 4 points should be four-times larger than a story assigned 1 point
- The process of assignment either uses a base story of 1 or 2 point(s) to be compared

76



Ideal Days

- ❑ Like Story Points, a measure of relative size not an estimate of duration
- ❑ The time a task would take in real time is called the 'elapsed time'. 'Ideal Days' should be an estimate of the expected effort in an idealised world, with no interruptions, dependencies, risks, mistakes, etc...

77



10252016

Estimating Techniques

- ❑ There are diminishing returns on time spent in estimating; the objective is to produce an adequate figure, one that is useful for the task in hand
- ❑ Estimating is a collaborative process emphasizing that responsibility for estimating the work lies with those performing the work

78



10252016

Estimating Techniques

- A number of techniques can be used to structure collaborative estimating
- Wideband Delphi is a consensus-based technique for estimating effort
- “Planning Poker” is used by development teams and “The Planning Game” by Product Owners with developers
- Magic Estimation: see next slides

79



Magic Estimating

- This isn't inherently complex 😊, we will need a
 - Floor
 - People
 - Planning Poker cards
 - Product Backlog

- That's it!**

80



Magic Estimating

- Start with the Product Backlog of user stories
 - Team will play, Product Owner will watch (and learn)
 - Lay the estimation cards down on the floor, spaced out as per their Story Point values e.g. 1, 2, 3, 5, 8, 13, 20 etc...
 - Hand out user stories to team - stories should be distributed evenly among the team members.
 - Explain rules: no talking, no non-verbal communication during estimating
 - Each team member estimates his/her own stories, places the stories at points
 - If a Story is not clear, it is put on the “?”. After explanation by the PO, the story will be estimated again.
 - Each team member checks all estimates of the other team members, re-estimates and moves if desired (once all their own cards are down)
 - Product Owner marks fall-outs (Big ones and Bouncers) to be discussed and clarified

81



Magic Estimating

- Big ones (Stories where estimates end up very large)
- Bouncers (Stories where team cannot agree on the estimate)
- Product Owner picks these Stories up and the team does Planning Poker on them after Magic Estimation is complete



82



Estimating - Exercise

□ Exercise 2:

- **Part A:** Read the following text and create a short presentation explaining its content
 - (taken from http://www.scrumalliance.org/pages/what_is_scrum)
- **Part B:** Study the other URLs that the Instructor will now give you and consider the work required to create a presentation for each
 - http://en.wikipedia.org/wiki/Planning_poker
 - http://en.wikipedia.org/wiki/Net_present_value
 - http://en.wikipedia.org/wiki/Critical_path_method
 - [http://en.wikipedia.org/wiki/Scrum_\(software_development\)](http://en.wikipedia.org/wiki/Scrum_(software_development))
- **Part C:** Using 'Planning Poker', estimate as a team the size of task required to create each presentation using the original text as the base unit.

83



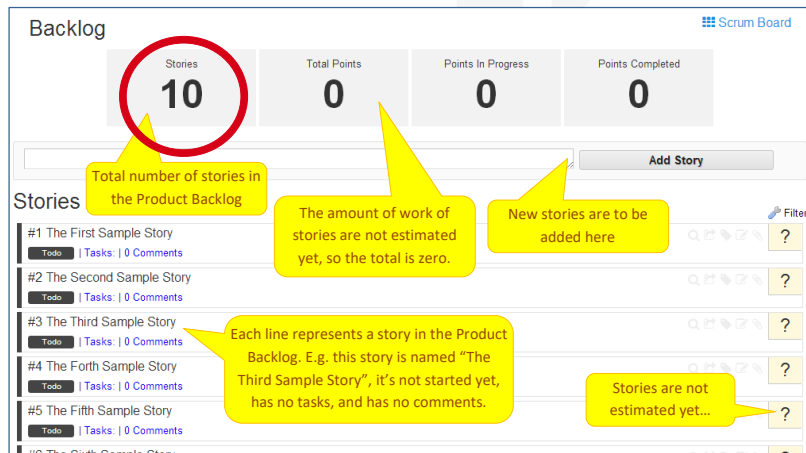
Product Backlog - Example

- The next slides show the type of information available for a single Product Backlog item in a typical Scrum tool. This is also a good example of a Scrum tool.
- Here are examples for User Stories. Do you see what is missing?
 - <https://www.mountangoatsoftware.com/agile/scrum/product-backlog/example>
- You can see in the example how the Product Backlog and the Sprint Backlog evolve

84



Product Backlog - Example



85



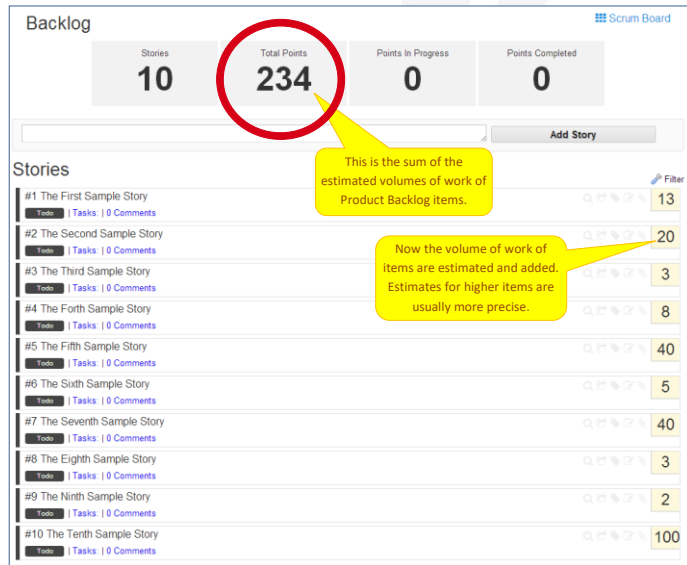
Product Backlog - Example

- This shows a sample Product Backlog created in an online Scrum tool. The Product Owner has identified the stories, but the estimates are not yet done (question marks on the right side of the rows).
- Current status of the sample:
 - **Identified stories** yes
 - Estimates no
 - Sprint plan no
 - Tasks no
 - Completed tasks and stories no

86



Product Backlog - Example



Product Backlog - Example

- Product Backlog tool after the addition of the estimates.
- Current status of the sample:
 - Identified stories yes
 - **Estimates** yes
 - Sprint plan no
 - Tasks no
 - Completed tasks and stories no
- The total amount of work in this sample is 234 points. Very large stories (normally stories with estimates higher than 13 story points) are often Epics and must be split into smaller stories later.

88

Sprint Backlog - Example

Iteration Planning

Project Velocity	Last Iteration Stories	Last Iteration Velocity	Planned Iteration Stories	Planned Iteration Points
-	-	-	-	-

Backlog

- #1 The First Sample Story (13 points)
- #2 The Second Sample Story (20 points)
- #3 The Third Sample Story (3 points)
- #4 The Forth Sample Story (8 points)
- #5 The Fifth Sample Story (40 points)
- #6 The Sixth Sample Story (5 points)
- #7 The Seventh Sample Story (40 points)
- #8 The Eighth Sample Story (2 points)

The First Sprint
Choose an iteration...
The First Sprint

List of all defined Sprints. It is time to select the next Sprint and define its backlog.

Available items in the Product Backlog to be added to the Sprint

A sample screenshot from online Scrum tool, [ScrumDo](#)



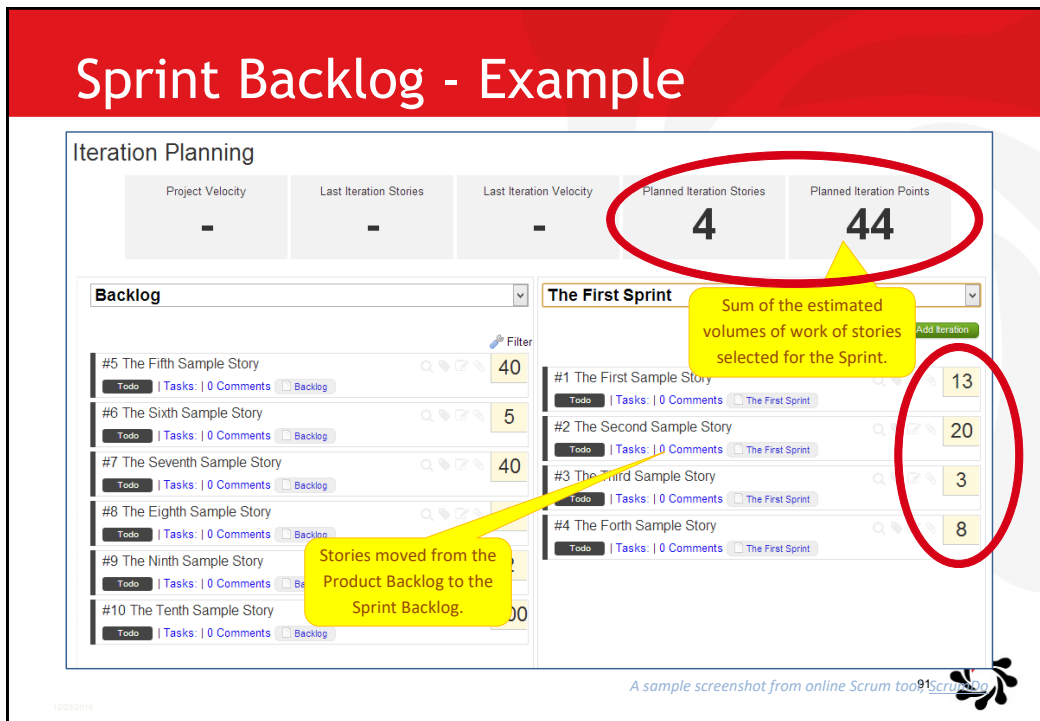
Sprint Backlog - Example

- Sample project in Sprint Planning: a new Sprint, called “The First Sprint”. Scrum Team are now ready to select items from the top of the Product Backlog to be assigned to this Sprint.
- Suppose the estimated team capacity by 50 points/Sprint
- Current status of the sample:
 - Identified stories yes
 - Estimates yes
 - **Sprint** **plan on-going**
 - Tasks no
 - Completed tasks and stories no

90



Sprint Backlog - Example

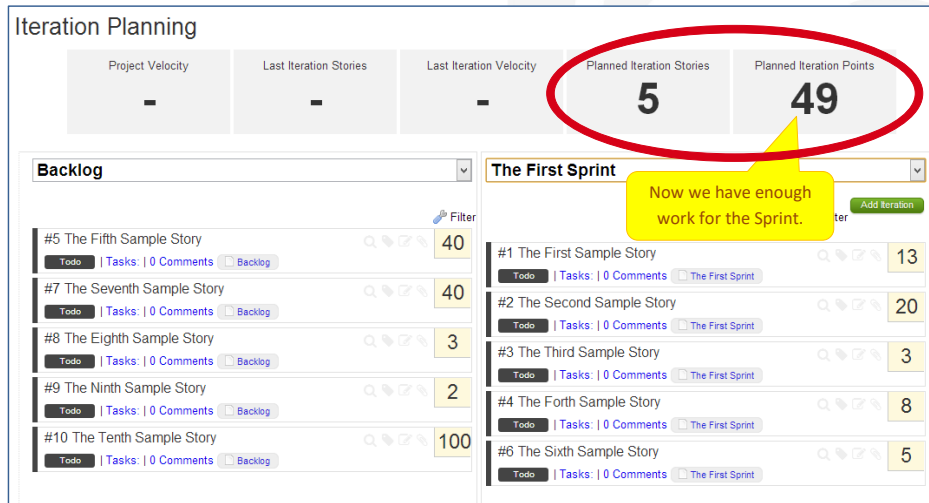


Sprint Backlog - Example

- ❑ There are four items with an estimate of 44 points. It is not possible to add the next Product Backlog item, because it has 40 points and there is only about 6 points free in our Sprint capacity.
- ❑ You can split the next large story/epic into smaller ones (should normally have been done during refinement in advance)
- ❑ It is common for the Product Owner in such cases to reorder the backlog; for example to bring The 6th Sample Story above The 5th Sample Story, and so we can add it to the Sprint Backlog (next figure).



Sprint Backlog - Example



A sample screenshot from online Scrum tool, [ScrumPlas](#)

93



Sprint Backlog - Example

□ Current status of the sample:

- Identified stories yes
- Estimates yes
- **Sprint plan** yes
- Tasks no
- Completed tasks and stories no

94



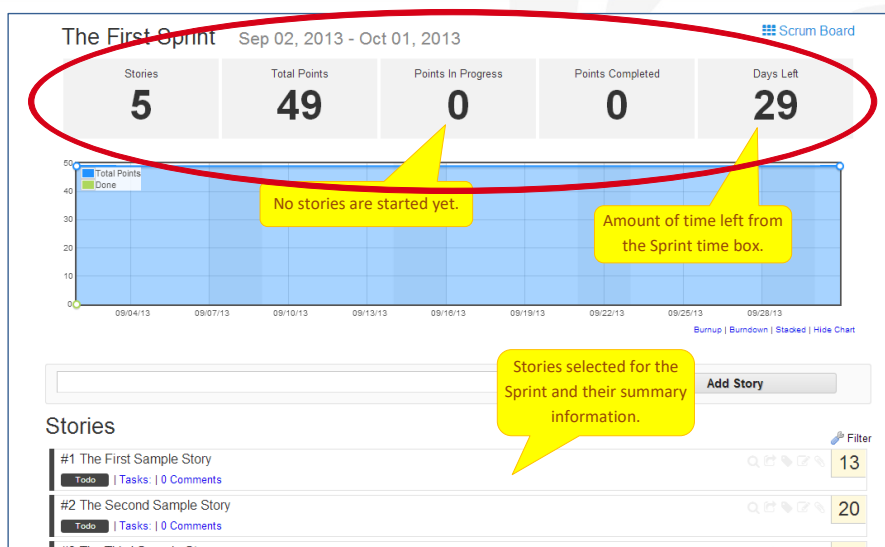
Sprint Backlog - Example

- ❑ The next figure shows the Sprint items, along with a summary, and a Burn-Down chart.
- ❑ Current status of the sample:
 - Identified stories yes
 - Estimates yes
 - Sprint plan yes
 - Tasks no
 - Completed tasks and stories no



10232016

Sprint Backlog - Example



Sprint Backlog - Example

Stories

Story ID	Story Name	Tasks	Comments	Estimate
#1	The First Sample Story	5	3	13
#2	The Second Sample Story	7	2	20
#3	The Third Sample Story	3	0	3
#4	The Forth Sample Story	0	0	8
#6	The Sixth Sample Story	0	0	5

A sample screenshot from online Scrum tool, [ScrumDo](#)

A sample screenshot from online Scrum tool, [ScrumDo](#)

97



Sprint Backlog - Example

- Current status of the sample:
 - Identified stories yes
 - Estimates yes
 - Sprint plan yes
 - **Tasks** **yes (for some stories)**
 - Completed tasks and stories no

98



Sprint Backlog - Example

- ❑ The Scrum tools usually update the Burn-Down chart as we progress through the Sprint.

Stories

Story ID	Status	Tasks	Comments	Estimate
#1 The First Sample Story	Done	6	3	13
#2 The Second Sample Story	Done	7	4	20
#3 The Third Sample Story	Todo	6 (3 finished)	2	3
#4 The Fourth Sample Story	Todo	4	1	8
#6 The Sixth Sample Story	Todo	6	2	5

A sample screenshot from online Scrum tool, [ScrumDo](#)

A sample screenshot from online Scrum tool, [ScrumDo](#)



99

Sprint Backlog - Example

- ❑ As we go through the Sprint, some tasks and items get Done and more items are detailed.
- ❑ Current status of the sample:
 - Identified stories yes
 - Estimates yes
 - Sprint plan yes
 - Tasks yes
 - **Completed tasks and stories yes (some of them)**



100

Quality Testing & Done

- ❑ Quality: the extent to which a product or process is fit for purpose
- ❑ Inspection, the continuous assessment of the quality of products and processes, is a function of each Scrum Event
- ❑ Testing of software during development is a focus of agile approaches
- ❑ Verification: testing to ensure that software conforms to specification
- ❑ Validation: testing to ensure that software conforms to need; i.e. acceptance



101

Definition of Done I

- ❑ The “Definition of Done”: the agreement by Scrum Team members and significant stakeholders that software has been satisfactorily verified and validated - usually in the form of a clear and concise list of requirements that a software Increment must adhere to for the team to call it complete. Until this list is satisfied, a product Increment is not done.
- ❑ The Scrum Guide™ describes the Definition of Done (DoD) as a tool for bringing transparency to the work a Scrum Team is performing. It is related more to the *quality* of a product, rather than its functionality.
- ❑ Having a clear Definition of Done helps Scrum Teams work together more collaboratively, increases transparency, and ultimately results in the development of consistently higher quality software.



102

Definition of Done II

- ❑ During the Sprint Planning meeting, the Scrum Team develops or reconfirms its DoD, which enables the Development Team to know how much work to select for a given Sprint. Further, a common DoD helps to:
 - Baseline progress on work items
 - Enable transparency within the Scrum Team
 - Expose work items that need attention
 - Determine when an Increment is ready for release
- ❑ The Definition of Done is not changed during a Sprint, but should change periodically between Sprints to reflect improvements the Development Team has made in its processes and capabilities to deliver software.
- ❑ The adoption of DoD happens during Sprint Retrospective



103

Definition of Done III

- ❑ When multiple Scrum Teams are working on a single project, it might not be possible to use the same definition of “Done” for all teams, because they might be working on items of different natures.
- ❑ In such a case, each Scrum Team will define its own definition of “Done” and delivers its items based on that definition.
- ❑ However, the integration of those definitions of “Done” should be capable of creating a potentially releasable Increment in the project level.



104

Monitoring Progress

- 'Inspect & Adapt' applies as much to the progress of the project as it does to the products produced
 - Daily Scrum
 - Velocity
 - Burn-Down
 - Quality, Testing & 'Done'
- Progress is monitored within and between Sprints*
 - The **Development Team** tracks the work remaining of the **Sprint** with the Daily Scrum
 - The **Product Owner** tracks the work remaining to reach a goal at each **Sprint Review**

105



Monitoring Tools

- Use the Burn-Down chart to visualise the progress of development during a Sprint (sprint Burn-Down) and visualise the progress of the whole project (project Burn-Down).
- The Product Owner is responsible to monitor the progress of the whole project toward its goal. This should be done at least once per Sprint Review. The Product Owner determines the amount of remaining work and compares it to the remaining work of the previous Sprints, and forecasts the completion date of the project. All stakeholders should have access to this information.
- The project Burn-Down chart shows the amount of remaining work, instead of the amount of completed work

106

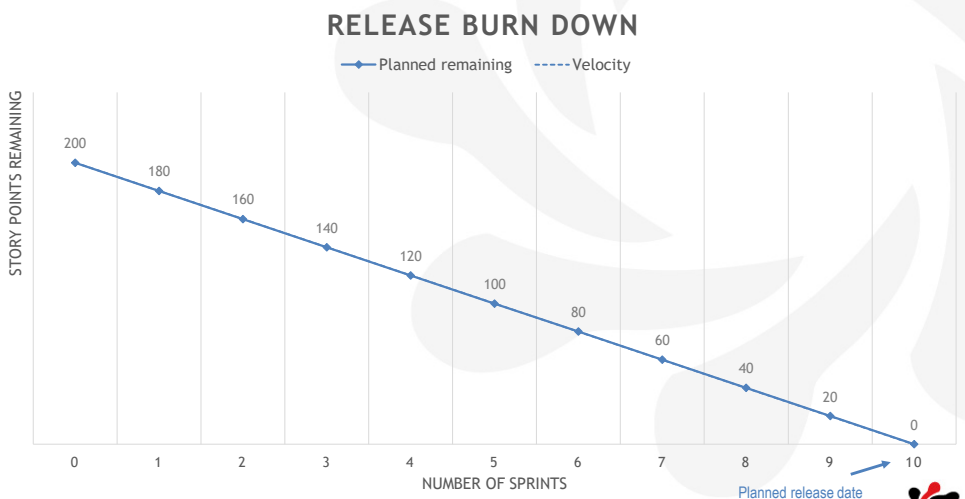


Burn-Down

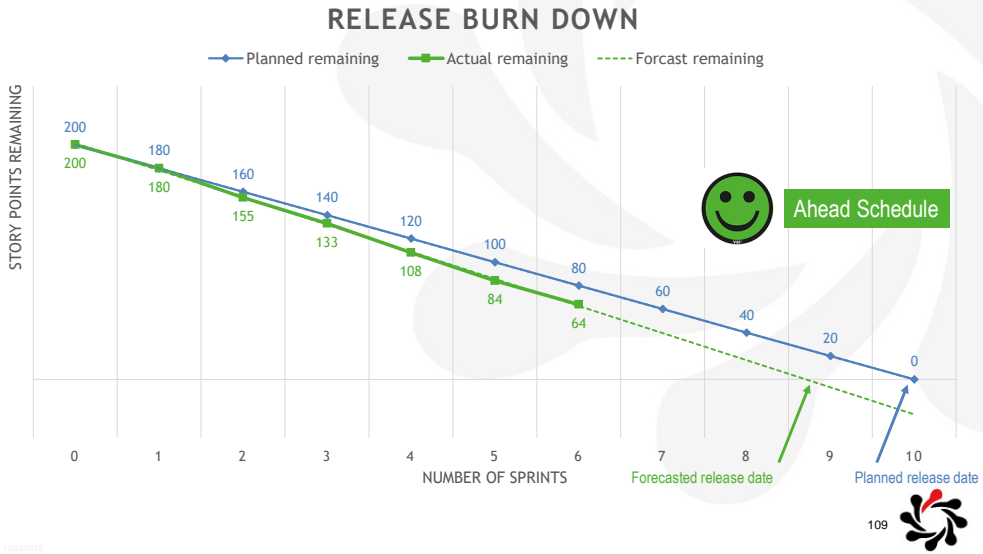
- 'Burn-Down' is a measure of project work outstanding
- "Scrum does not consider the time spent working on Sprint Backlog items. The work remaining and the date are the only variables of interest"



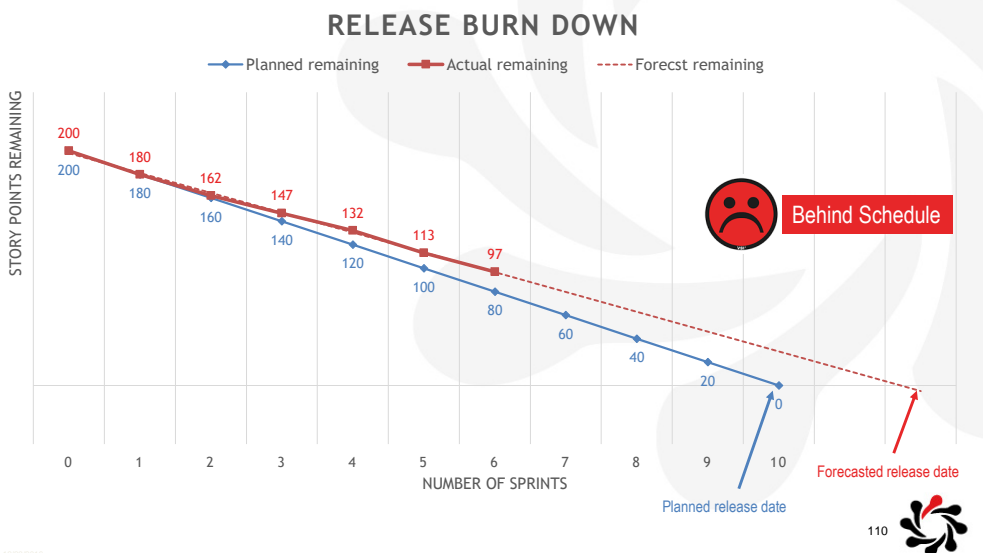
Monitoring Tools - Release Burn-Down



Monitoring Tools - Release Burn-Down



Monitoring Tools - Release Burn-Down



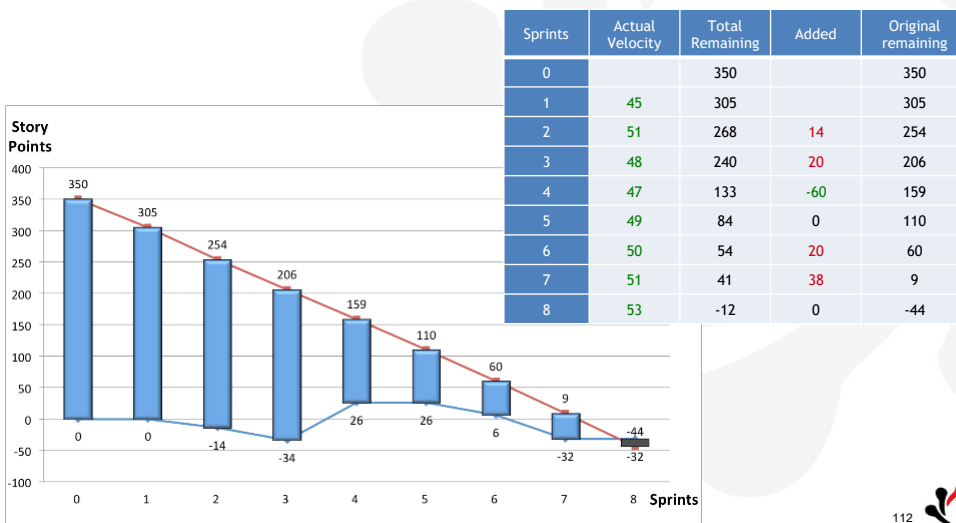
Velocity

- Velocity is the measure of a teams progress in terms of work completed, the number of Story Points or Ideal Days the team completes within a Sprint
 - Velocity across Sprints is measured in Points or (Ideal) Days
 - Velocity within Sprints is measured in (Ideal) Hours
- This figure, averaged over a number of Sprints, is then used to estimate the amount of work that it is sensible to assign to a Sprint



111

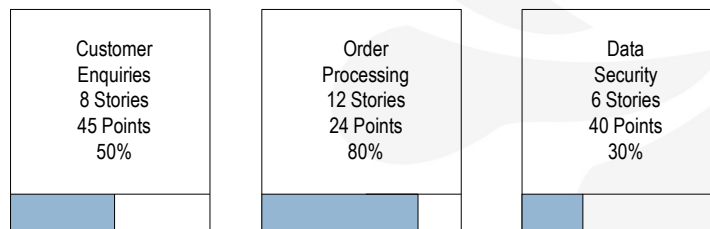
Burn-Down Bar-Chart - Example



112

Parking Lot Chart

- Both the Burn-down Chart and the Burn-down Bar Chart show aggregate progress within or across Sprints. The 'Parking Lot' Chart shows progress across themes (areas of functionality, groups of stories).



113



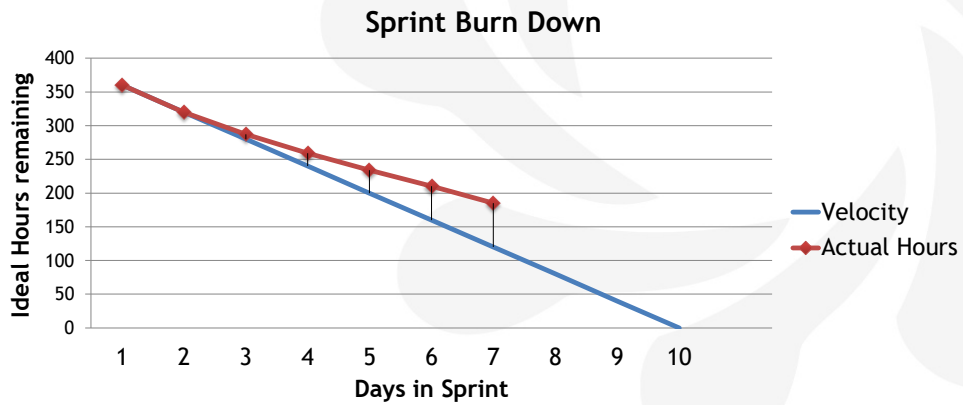
Monitoring Tools - Sprint Burn-Down

- Besides the monitoring done for the whole project, we should also monitor the progress of each single Sprint throughout its life. This is the responsibility of the Development Team and should be done at least once per Daily Scrum.
- This information is used to calculate the likelihood of achieving the Sprint Goal and completing all items of the Sprint Backlog.

114



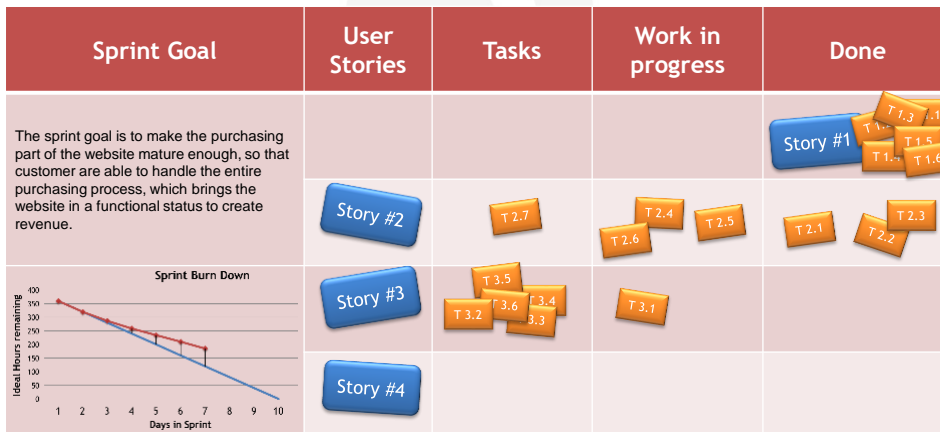
Monitoring Tools - Sprint Burn-Down



115

Monitoring Tools - Task Board

□ A Task Board can be used to visualise progress



116

Burn Down - Exercise

□ Exercise 3: Velocity & Burn-down

- **Part A:** A Sprint of 20 working days is scheduled to deliver 35 Story Points. Plot a Burn-down chart and determine the ideal daily velocity of the team.
- **Part B:** The team reports the following points done at the Daily Scrum.

Day	1	2	3	4	5
Points done	1	2	1	3	1

Plot these figures on the chart created for Part A and determine the team's actual velocity



Burn Down - Exercise

□ Exercise 3: Velocity & Burn-down

- **Part C:** A team working on a project has been delivering a number of 'Done' points per sprint against a point target. Unfortunately, the Product Owner has also been adding points to Sprints and estimates for points have also been estimated up.

Sprint	1	2	3	4	5
Points Target	30	28	29	31	28
Points Done	28	30	25	28	29
Points Added	6	12	5	4	8

Plot the Burn-down chart for these figures. Plot a Burn-down bar Chart and determine the team's actual velocity.



Multi Level-Planning

- ❑ Planning is not about predicting the future, the development process is too chaotic for that
- ❑ Planning is about aligning expectations, within the Scrum Team and with Stakeholders
- ❑ The 'Planning Horizon' limits the value of detailed planning so the approach called 'Rolling Wave Planning' is used
- ❑ Scrum uses three planning horizons to handle progressive plan elaboration,
 - Release
 - Sprint
 - Day



119

Release Planning

- ❑ A Release Plan is a 'high-level' plan that covers a period greater than a Sprint
- ❑ It identifies the time and work is needed to create a releasable product
- ❑ It provides a forecast of delivery and timeframe
- ❑ It provides a framework for assessing Sprint performance



120

Product Development Roadmap

- This identifies in outline the main areas of development focus over the next few releases
- This may be 'feature-driven' or 'date-driven'
- This is the 'Product Breakdown Structure' for the project

121



Release Plan

- Using the estimates of prioritised stories and the forecasts of the amount of work that can be delivered in each Sprint, which Stories will be in which Sprints, can be 'roughed out'
- Following the principle of 'rolling-wave planning' specific functions are assigned to the next couple of Sprints only

122



Sprint Planning

- ❑ It is during the Sprint Planning Meeting that the Stories selected for the Sprint are broken-down into Tasks (the 'Work Breakdown Structure')
- ❑ The tasks derived from the Stories are estimated in Ideal Hours

123



1232016

Daily Planning

- ❑ Tasks are assigned during the Daily Scrum
- ❑ The 'Task Board' is a simple and transparent way to record and display a schedule, each row represents a Story and its associated tasks, the columns are used to show tasks waiting, in progress, completed etc...

124



1232016

Scaling Scrum

- ❑ When more than one Scrum Team work on a project, which is highly likely given the size of each team, it is referred to as a 'scaled project'
- ❑ The procedures put in place to coordinate scaled projects are called 'scaling mechanisms'

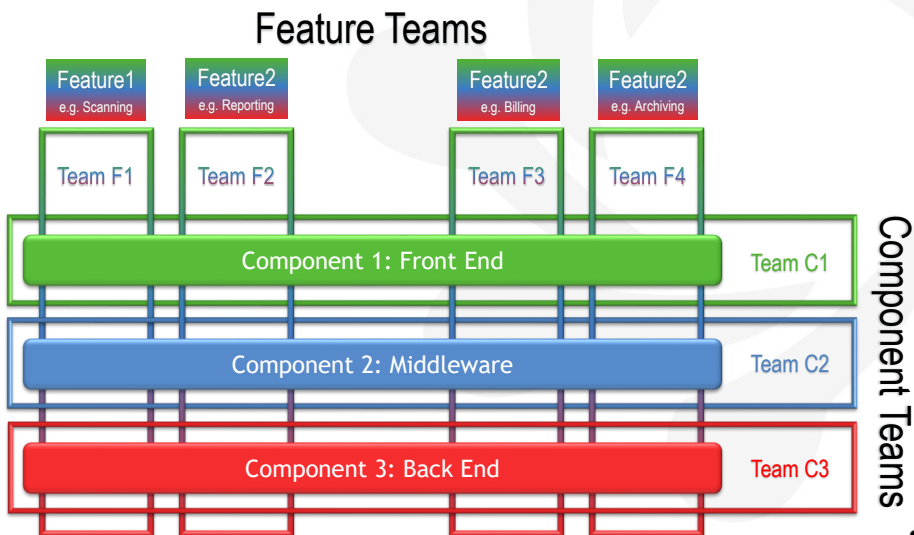


Scaling Scrum - Team Structure

- ❑ Cross-functional Team:
 - A team composed of members with all the functional skills (such as UI designers, developers, testers) and specialties necessary to complete work that requires more than a single discipline.
- ❑ Feature Teams:
 - A cross-functional and cross-component team that can pull end-customer features from the product backlog and complete them. See also *cross-functional team*. Contrast with [component team](#).
- ❑ Component Teams:
 - 1. A team that focuses on the creation of one or more components of a larger product that a customer would purchase. Component teams create assets or components that are then reused by other teams to assemble customer-valuable solutions.
 - 2. Team that is cross-functional (multi-disciplinary), single component focused. Contrast with [feature team](#).



Scaling Scrum - Team Structure



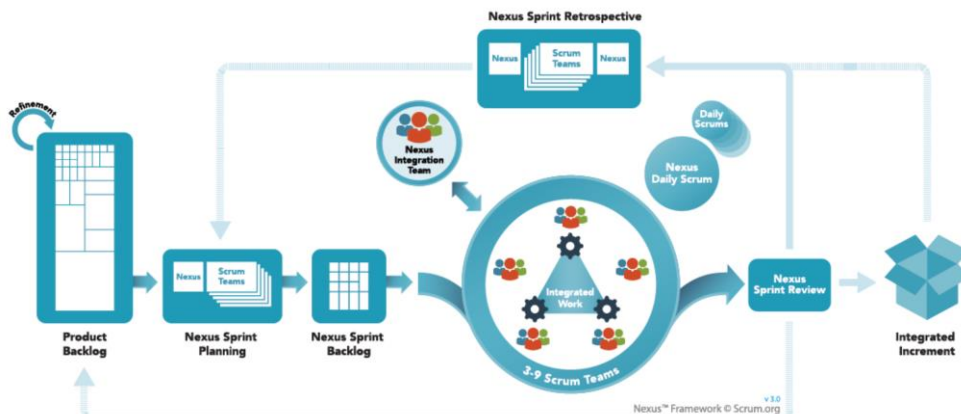
Scaling Scrum - Nexus Framework

- ❑ Scrum.org issued a new standard for scaling:
 - <https://www.scrum.org/Resources/The-Nexus-Guide/Downloads>
- ❑ Nexus is a framework that drives to the heart of scaling: cross-team dependencies and integration issues.
- ❑ It is an exoskeleton that rests on top of multiple Scrum Teams who work together to create an Integrated Increment. It builds on the Scrum framework and values.
- ❑ The result can be an effective development group of up to 100 people. For larger initiatives, there is Nexus+, a unification of more than one Nexus.

128

Scaling Infrastructure

□ Nexus Framework



129



Scaling Infrastructure

- The process of defining and prioritizing the requirements for scaling is called 'staging'
- Staging happens prior to the first Sprint and takes usually a single day
- A single team Sprints through the scaling infrastructure, with the other teams created once it is in place
- 'Sashimi': Each slice contains a little bit of everything, all the work and infrastructure required to deliver a complete product are performed together

130



Hierarchy of Backlogs

- A single 'master' Product Backlog is used across the project in order to coordinate work
- If requirements are derived from multiple user groups, regions, etc..., individual Product Backlogs are created and amalgamated into the master Backlog
- The emphasis is on transparency during the amalgamation process

131



Hierarchy of Scrums

- Where multiple teams are working on a project, a 'Scrum of Scrums' approach can be used
- These meetings of team representatives allow clusters of teams to discuss their work, focusing especially on areas of overlap and integration
- Both the Daily Scrum and Scrum of Scrums are not status reports
- Unlike the Daily Scrum which focuses on planning rather than problem management, the Scrum of Scrums is the opposite

132



Scrum Master Foundations

Part Three - Assessments



133

Scrum.org Assessments

- SCRUM.ORG offer a number of assessments that allow people to check and demonstrate their understanding of SCRUM
- We shall use two of these during this course
 - Open Assessment
 - PSM I Assessment
- The third assessment (PSM II) is beyond the scope of this course



134

Open Assessment

- ❑ “The Scrum Open assessment is available for free to anyone interested in testing their knowledge of Scrum. The Scrum Open assessment is also a useful tool for those preparing to take the Professional Scrum Master I assessment for certification”
- ❑ “The assessment consists of 30 questions randomly selected from a larger pool



PSM I

- ❑ Professional Scrum Master I (Fundamentals)
- ❑ “This assessment of fundamental Scrum knowledge is available to anyone who wishes to demonstrate his or her knowledge and achieve certification”
- ❑ 60 minute examination consisting of multiple choice questions with a pass mark of 85%



PSM II

- ❑ Professional Scrum Master II (Intermediate)
- ❑ “This assessment of intermediate Scrum knowledge and skill is open to anyone who has passed the PSM I assessment and wishes to demonstrate his or her ability to apply Scrum to solve complex problems”
- ❑ 120 minute examination consisting of multiple choice questions , case study questions, and essays with a pass mark of 85%

137



Scrum Master Foundations

End of the course

Thank you for your attention!

138

