Jyothi chundi

Scrum Alliance
CSM
CERTIFIED

Course Book

CERTIFIED SCRUM MASTER® (CSM)

www.knowledgehut.com

knowledgehut

Certified ScrumMaster Workshop

PARTICIPANT HANDBOOK

FACILITATED BY

CERTIFIED SCRUM TRAINER

Vivek Angiras

Contents

Instructor Introduction

Vivek Angiras
Trainer and Coach

vivangiras@gmail.com
@vivangiras
vivekangiras

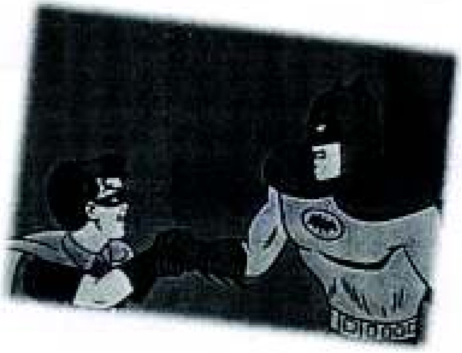
CST
CAL
CSM
CSPO
CSD
CSP-SM
CSP-PO
KeBS
Certified Practitioner
SCALED AGILE FRAMEWORK PROGRAM CONSULTANT
Kanban Coaching Professional Masterclass
PMI
TRAINING FROM THE BACK OF THE ROOM
Trained Facilitator
Wyeth
Pfizer
UNOS UNITED NETWORK FOR ORGAN SHARING
RBS
FreddieMac We make home possible
STATE STREET GLOBAL ADVISORS.
CapitalOne
LOWE'S
AIG

- Based out of Washington DC, USA
- MS in Electrical Engineering
- 12+ years in Information Technology

Who Am I?

Find a partner (someone not sitting at your table)

- Start telling them about yourself. As soon as they hear something you both have in common, they will say "Me Too!" and find a new partner



Where do I stand?

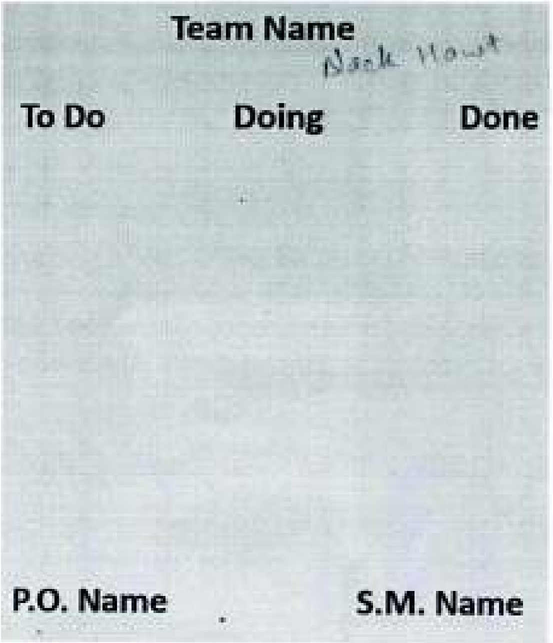
Self-organize to line up by level of Scrum experience – (least to most)

Choose your team roles

- Invent a Team Name
- Select a Product Owner
- Select a Scrum Master

CONSTRAINT: If you are currently a SM or PO, please do NOT play that role here

Definition of Done: A PO, SM, and Team Name have been chosen AND a flip chart has been created labeled according to the template on the right



Team Name

Dark Horse

To Do **Doing** **Done**

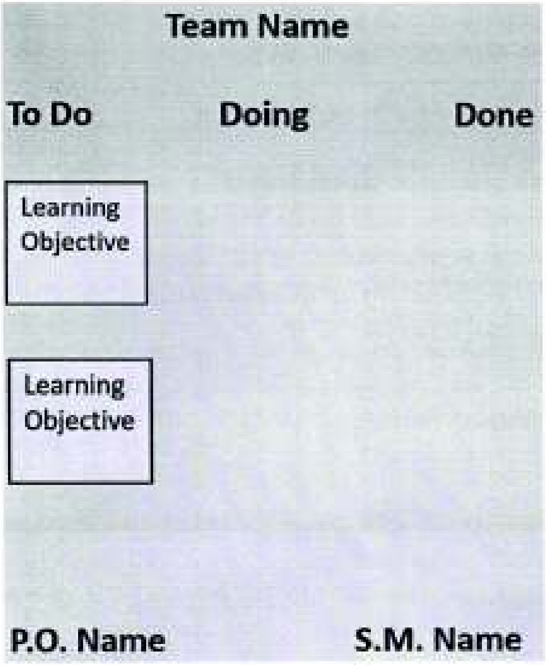
P.O. Name **S.M. Name**

Create your Learning Backlog

Create a Learning Backlog: What do you hope to get out of the class individually and as a team?

Each objective goes on its own sticky note

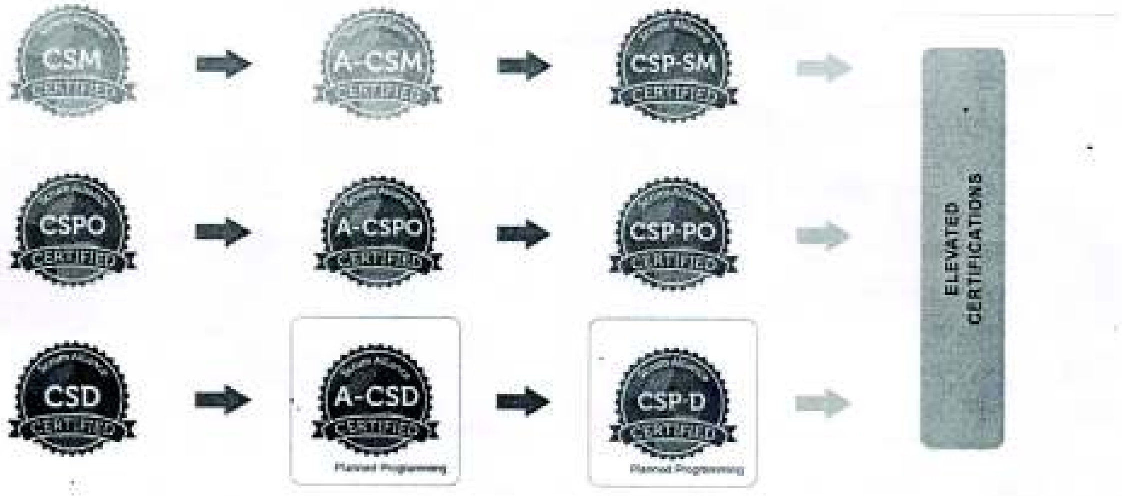
Definition of Done: All the sticky notes are in the 'Do' column AND your Learning Backlog is posted on a wall



About the CSM Test

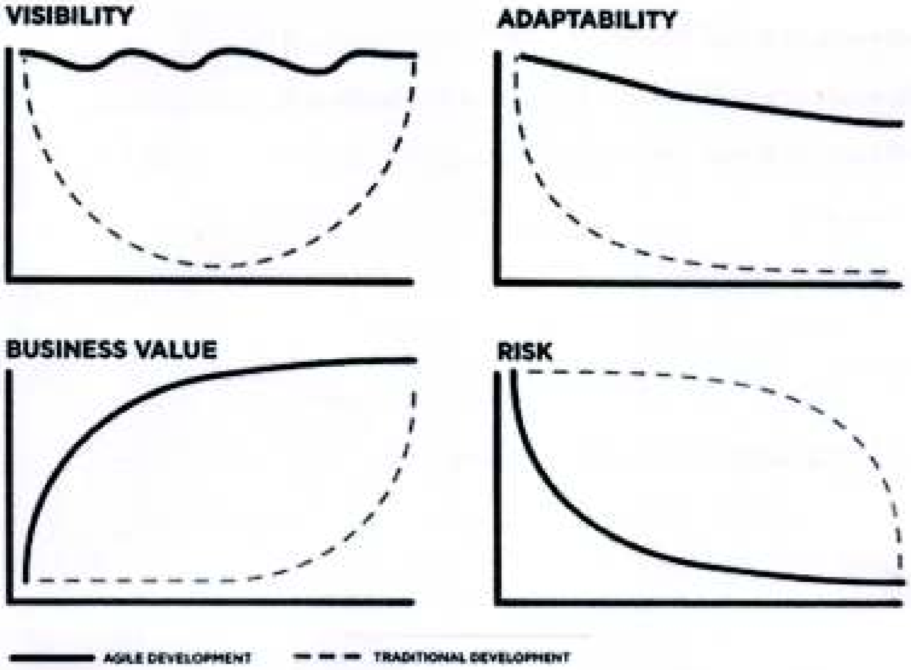
- You will receive email instructions for the test after class
- Passing score is >=24/35 correctly answered questions
- You will receive email from Scrum Alliance with instruction about how to take the test
- Certification is good for 2 years

Foundational and Advanced Certifications



www.scrumalliance.org

Why Agile?



Source : Version One

Ball Point Game (Scrum in Action!)

Goal: Maximize Throughput! → running avg

Rules:

- Everyone must touch the ball at least once
- Cannot pass to directly to your left or right
- During each pass, ball must have air time
- Dropped ball is a defect (but can be reused)

Ball Point Game Debrief

Did you enjoy the game? Yes

What was the learning for you?

1) Planning with Time & Resource

2) Execution and understanding the ~~Ba~~ Issues & replanning

3) Replanning ~~keeps~~ on ~~Bn~~ add advantage to the Delivery 4

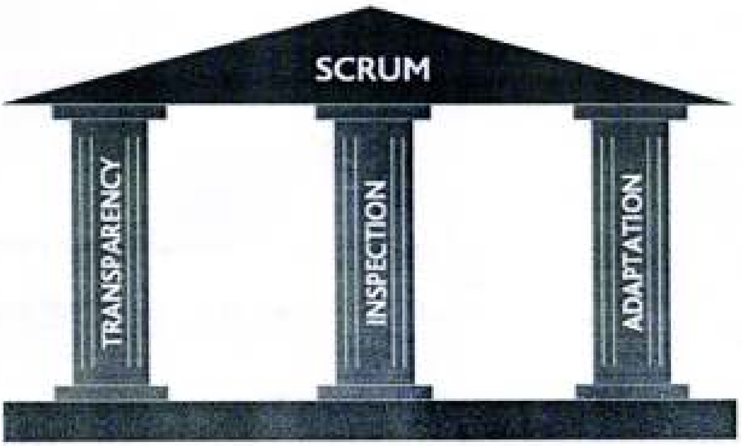
What would be different if the game was executed using a "sequential" approach i.e. plan all first, then execute?

1) replanning and Improvement of approach cannot be happen.

What attributes should not change in this game for predictable outcomes?

1) Rules 2) Resource 3) Team 4) Time

Empiricism



Why are the following required in Scrum?

- Transparency

- Inspection

- Adaptation

Iterative and Incremental Delivery

Incremental development is distinctly different from iterative development. Teams get into trouble by doing one and not the other, or by trying to manage them the same way.

The word increment fundamentally means add onto. The word iterate fundamentally means re-do. Incremental development gives you opportunities to improve your development process, as well as adjust the requirements to the changing world. Iterative development helps you improve your product quality. Yes, it is rework, and yes, you probably need to do some rework to make your product shine.

Incremental development is a staging and scheduling strategy in which various parts of the system are developed at different times or rates and integrated as they are completed.

The alternative strategy to incremental development is to develop the entire system with a big-bang integration at the end.

Iterative development is a rework scheduling strategy in which time is set aside to revise and improve parts of the system. The alternative strategy to iterative development is to plan to get everything right the first time.

It is important to notice that neither strategy presupposes, requires, or implies the other. It is possible to do either alone, both, or neither.

In practice, it is advisable to do both in different quantities. If you only increment, there tends to be an unpleasant surprise at the end when the quality is not good enough. If you iterate the entire system, ripple effects of the changes easily get out of control.

Reference: http://alistair.cockburn.us/Using+both+incremental+and+iterative+development

Three Wastes in Lean Thinking: 3 M's

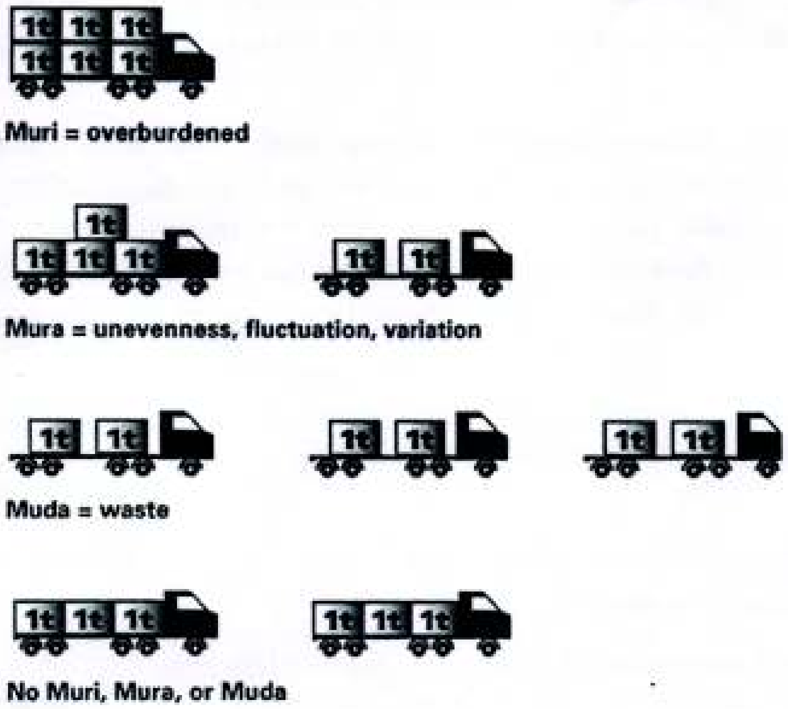


Image Courtesy: lean.org

Did you see any wastes in the video? Discuss and share in pairs.

Chd , carers, - Resource capacity   pressure

overburdened

unevennes.

Requirements.

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:

Individual and Interactions over s/w ~~product~~ processes and tools

Working s/w Product over comprehensive documentation

Customer collaboration over Contract negotiation

Responding to Change over following a plan

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

1 Individuals ✓

3 collaboration

5 processes

4. change

following a plan

2 Software/product ✓

contract

4, 6, 12

12 Principles

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

2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

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

5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

7. Working software is the primary measure of progress.

8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

9. Continuous attention to technical excellence and good design enhances agility.

10. Simplicity--the art of maximizing the amount of work not done--is essential.

11. The best architectures, requirements, and designs emerge from self-organizing teams.

12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Scrum Origins

India's version of a joyous scrum after their win

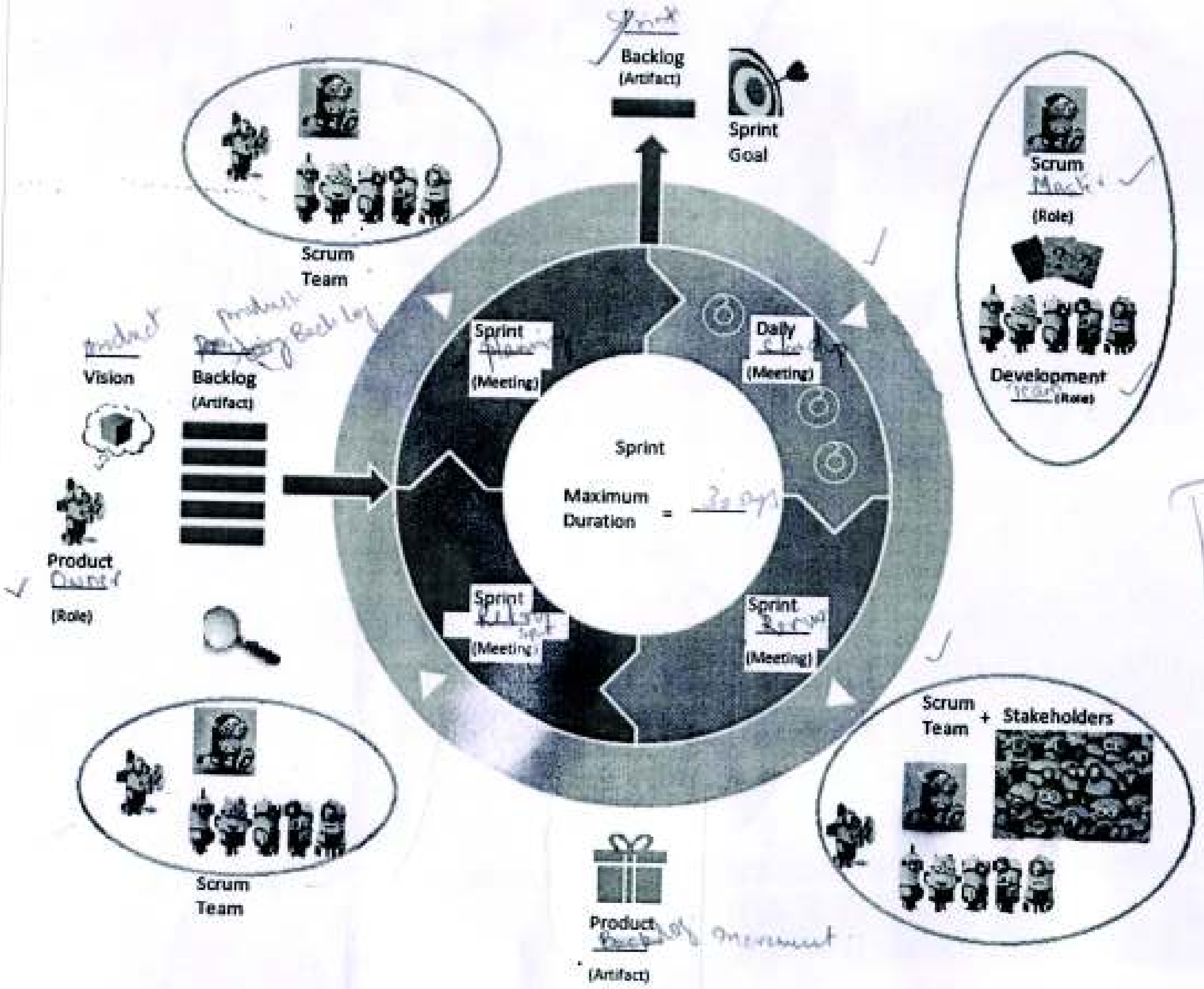
Associated Press

ESPN Cricinfo pic: Nidahas Trophy Mar 2018

"The 'relay race' approach to product development may conflict with the goals of maximum speed and flexibility. Instead a holistic or 'rugby' approach – where a team tries to go the distance as a unit, passing the ball back and forth – may better serve today's competitive requirements."

- Hirotaka Tekeuchi and Ikujiro Nonaka
  "The New New Product Development Game"
  Harvard Business Review, January 1986

Fill in the blanks


Sprint Backlog (Artifact)
Sprint Goal
Scrum Team
Scrum Master (Role)
Development team (Role)
product Vision
~~Daily~~ product Backlog (Artifact)
Product Owner (Role)
Sprint Planning (Meeting)
Daily Standup (Meeting)
Sprint
Maximum Duration = 30 days
Sprint Retrospect (Meeting)
Sprint Review (Meeting)
Scrum Team
Scrum Team + Stakeholders
Product Backlog increment (Artifact)

2w sprint.

3 → roles
4 → Meeting
3 → Artifacts

Scrum Values

FOCUS → planning, Standup,

OPENNESS → sprint planning, Daily Standup

RESPECT →

COMMITMENT → Standup,

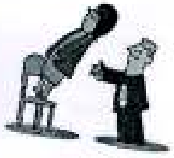
COURAGE Retro

3 Roles.
4 Metting
3 Artifacts

How does each value relate to Scrum framework?

Which value can be hardest to embrace in your organization?

What is the foundation of these 5 values? → Trust.

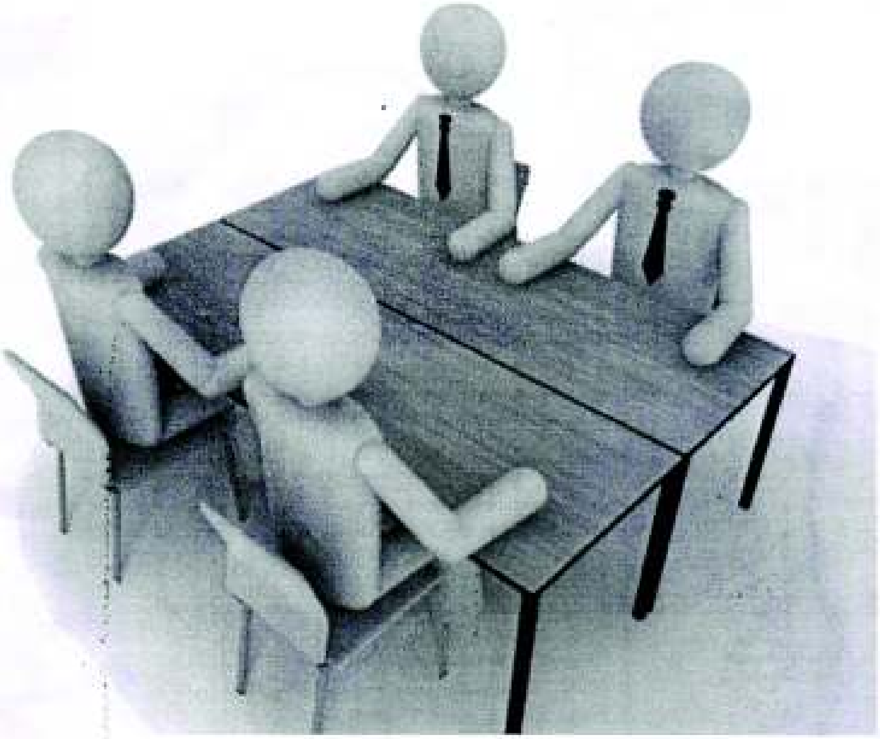


Pair Share

Shyam.

- Pair up with someone from a different table
- Think of your favorite boss or teacher and what was it you admired about them

Anatomy of a great Scrum Master

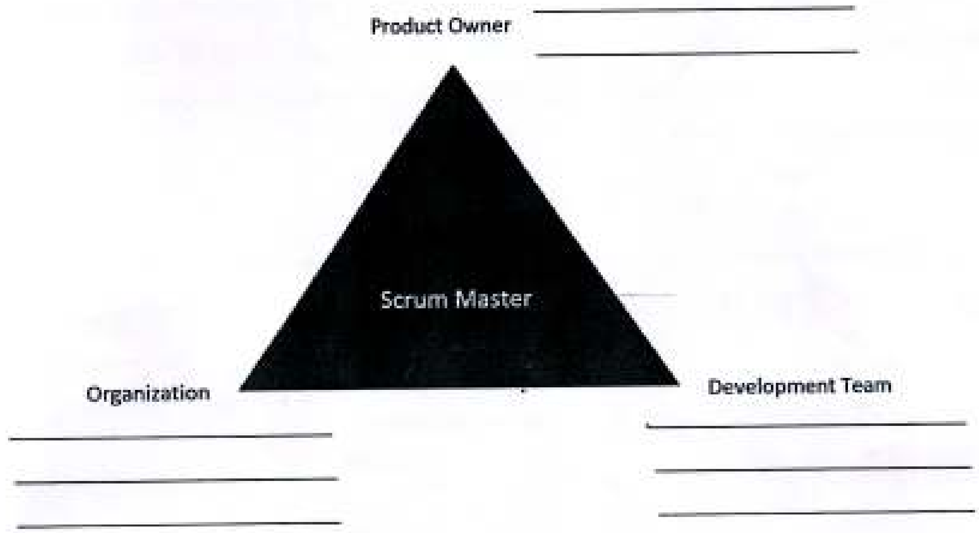


Scrum Master

Evidance Based approach.

- Ensures Scrum is understood and enacted
- Servant Leader
- Removes Impediments
- Facilitator – Meeting, Conflict Mangm. Mapping
- Coach and Mentor
- Protector

Ensures Scrum is Understood. coach + Mentor



What impediments will the Scrum Master face when she/he is helping the organization adopt Scrum? How will she/he go about removing those impediments?

Cultue shift, planning Budget, People avalibility, Support from leadership / first start will less risk team with practise which give support for scrum, resistance for new adop tion.

What skills and tools are needed for effective facilitation?

conflict Mangment, Networking, openness, Be in the Background

Timeboxing

The Scrum Master should show the Matrix for non availability of the P.O the Delay of the Decision Making,

Table Talk - Scrum Master Roleplay

ScrumMaster Role Play Instructions:

As a team, read the scenario below and answer the questions following the scenario:

You are the ScrumMaster for a Scrum team. The team is half-way through the current sprint. You are on your way to get some coffee, when the company's vice president of sales (who is a stakeholder and not the product owner) stops you in the hallway. The vice president of sales proceeds to tell you that they just got off the phone with the company's biggest customer. As it turns out the customer wants a new feature added immediately to the product. The vice president of sales made a commitment to meet the customer's request and now wants you to get the team to deliver during the current sprint.

1. What is the first thing that you would say to the vice president of sales?

   Lets talk to the product owner. facilitate the Discussion
   Some to wait for a week so to the VP when the change to be incoorporate

Trade off
Communication

2. How would you present this situation to the team?

3. What would the team's reaction be? How would you deal with their reaction as the ScrumMaster?

   Explaining the Business value to the Team
   Interaction of the Team with V.P
   Trade off Not to overloaded
   or Bring the product owner
   Scrum should have some Budget

Table Talk - Scrum Master Combo

Please discuss at your table pros and cons for each situation:

Situation 1:

Scrum Master is also a part of the team

pos: 1) understanding situation Better

2) Might Not be Nutral

3) estimation

Narrow Vision

Availability will be reduced

P

Situation 2:

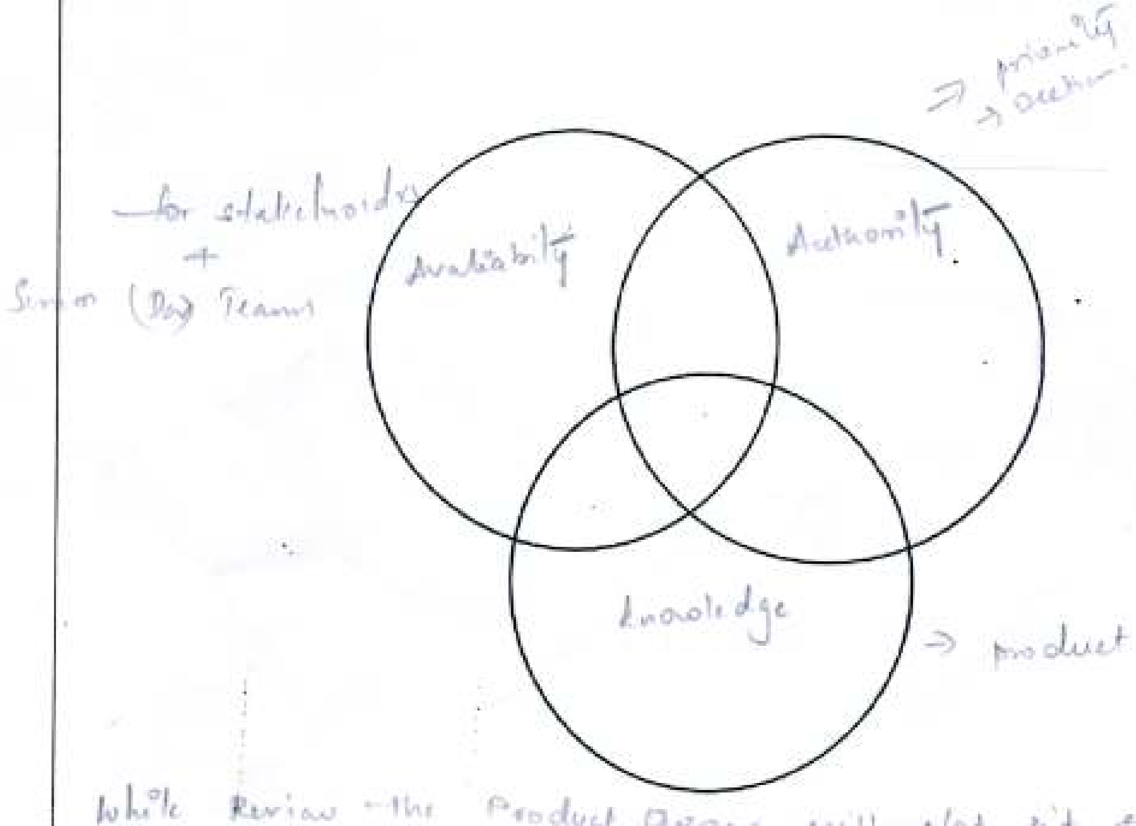
Scrum Master is also the Product Owner

Make Sure that time to have Next Sprint decision of Making Release

Product Owner

BusinessAnalyst is a proxy product Owner. Product Owner to take the Responsibility of Analist Make if needed.

- The Product Owner is responsible for maximizing the value of the product resulting from work of the Development Team
- Responsible for owning and managing the Product Backlog
- Product Owner is an individual and not a committee
- Serves the Development team and the ScrumMaster
- Collaborates with the Stakeholders
- Owns the Product Vision
- Responsible for maximizing ROI



While Review the Product Owner will Not sit side of Development Team.

Delay in Decision Make for NON availabity.

Dev. Team for all the Documents.

The selection of Dev Team Can be done by P.O, SM or product M anger again it happen differently in different Organisation.

You should be complain whether Team is Need
so to use Scrum

Table Talk – Product Owner Role Play

As a team, read the scenario below and answer the following questions:

The team, Product Owner, and ScrumMaster have gathered together for their second Sprint Planning meeting. In addition, several key stakeholders have joined as well. Present in the room are the directors of marketing, sales, and procurement. The Product Owner for this product works in information technology and was appointed by the chief information officer. In prior meetings, each of the directors have been pushing their own agendas. They came to this Sprint Planning meeting to make sure that their backlog items were presented to the team for implementation. To the surprise of each of the directors, the Product Owner presents product backlog items that they knew nothing about. The product backlog items presented appear to be information technology specific. The directors confront the product owner about their decision and a verbal argument breaks out.

1. What should the Product Owner do in this situation?

2. What should the ScrumMaster do in this situation?

3. What should the team do in this situation?

Manager Employee Exercise

Development Team

- Delivers potentially releasable Increment of "Done" product at the end of each Sprint.
- Size 3 – 9 team members (5-10 # Members) Dev. Team + P.O + SM
- Self-Organizing (SM & PO will Not dictate about the work to do)
- Cross Functional
- Own and manage Sprint backlog
- Own "How" and "How Much"

What's my role?

PRIMARY ROLE	SECONDARY ROLE	NOT MY ROLE
Scrum Master	Can be ~~Scrum~~ Product Owner or Development Team	

Table Talk

Project Manager is not a role on a Scrum Team. Who is responsible for the project management activities?

Scrum Master & Product Owner & Development Team.

What happens to specialists? → can be part of Dev Team. Cross functional Team needed for Better approach.

- Architect
- BA
- DBA
- Developer
- Other Specialists

Product Manger → stakeholder ← Product Owner
prioritization
Budgeting.
Decission Making.

Strategy
acceptance.
→ Priority

* Scrum is Modular in Nature

Scrum Events

→ 4 Inputs → 2 outputs

Sprint Planning

I/P

O/P
Sprint Backlog
Sprint Goal

- Sprint Planning meeting is to plan the work for the Sprint
- Meeting input is the Product Backlog, latest product Increment, projected capacity of the Development Team and its past performance
- Time-boxed to maximum eight hours for a month of Sprint
- It answers the following:
  - What can be delivered in the Increment resulting from the upcoming Sprint?
  - How will the work needed to deliver the Increment be achieved?

Product Backlog
capacity (looking about katen).
Velocity (Past performance)
* latest product Increments

What about bugs or production issues?

Product Owner — Dev Team   How & what

Sprint Goal

- An objective set for the Sprint that can be met through the implementation of the Product Backlog
- Created during Sprint Planning by Scrum Team
- All work done during the Sprint by the Development Team should be meant to satisfy the Sprint Goal
- Examples of Sprint Goals

Import candy Sketch.

Ship a basic version of shopping portal

Decrease landing page load times by 25%

Increase DB performance by 15%

For 1st Sprint Addln work.

product Backlog Refinment [priority, estimates]
product Initiation Needed.

then only we can start Scrum

priority of the Sprint [length of the Sprint] done at the ~~th~~ time of the product Initiation

2 – 8 weeks to be taken for Initiation

Scrum Master should find
→ Level of Maturity of the team.

Standup is XP Term

Any clarification is Needed It will be diffcult for sprint planning. The clarificatn done at product Backlog [Req, user stories etc]

Daily Scrum

Ask me about Tips and Tricks

- Daily Synchronization meeting for development team
- Timeboxed to 15 mins

SM will provide format
Doubt can be raised.
Impediments can be sorted out

Max-time 15 Mins

Each team member can share

1. What did I do to yesterday to help dev team accomplish the sprint goal?

2. What am I planning to do today to help dev team accomplish the sprint goal?

3. Are there any impediments?

SM should be able adopt for difficult situation
Team to self Organise. SM is just facilators.
SM will not check for user stories. SM will not give feedback
The PO or any other should not Interfere in Daily Scrum

Sprint Review

- Sprint review is to review the increment developed during the sprint
- Timeboxed to 4 hours for a month-long sprint
- Stakeholders and Scrum team collaborate during the meeting
- Scrum Team showcases the potentially releasable product increment to stakeholders

4 Meeting planning
15 x 1
last Day Review & Retrospective

Sprint Retrospective

- It is the final meeting before the beginning of the next sprint
- Timeboxed to 3 hours for a month-long sprint
- This meeting is only for the Scrum team
- The agenda for the meeting is to discuss: (About the Team Meeting with healthy atmosphere)

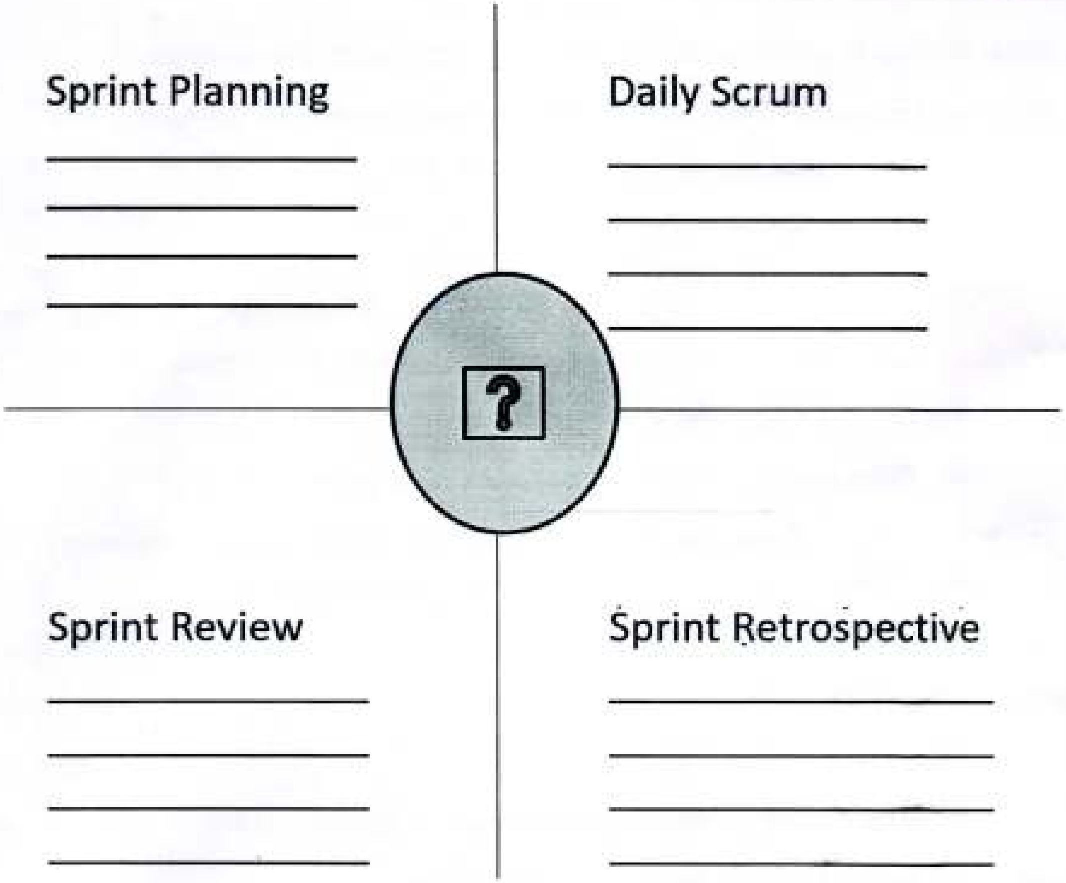
- What went well?

- What didn't go well?

- What can we do to improve?

Product Owner
→ -Proven for Backlog (~~Product~~ Backlog Remaining Work)
High viable, functional Non-functional etc
BRD is Traditional approach

Knowledge Check – 10 minutes!



Sprint Planning

____

____

____

____

Daily Scrum

____

____

____

____

Sprint Review

____

____

____

____

Sprint Retrospective

____

____

____

____

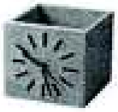
→ Dev Team will owne the Sprint Backlog
→ Sprint Backlog will update once in a day Depends on line Item

→ UnDone works goes Back to product Backlogs
Scrum will be useful if we maintain Timebox
Collabrolively the script backlog creation

Sprint

In Scrum, all events are time-boxed events, such that every event has a ~~fixed~~ Maximum duration. Once the Sprint begins its duration is fixed / fixed. and cannot be shortened or lengthened. Sprints are protected. Only PO can cancel the Sprint.

What are the benefits of Timeboxing?



Focus
No Time waste
Decision Making quicker
Team is Organised

What factors would influence selecting the duration of the Sprint?

~~capacity + velocity + product Back log~~
~~+ P.O + Dev Team.~~

Team size (capacity)
Product complex
How Frequency the Business changes the mind

Midweek the Sprint start

U Never Do product Backlog Refinement for current Sprint
1st week

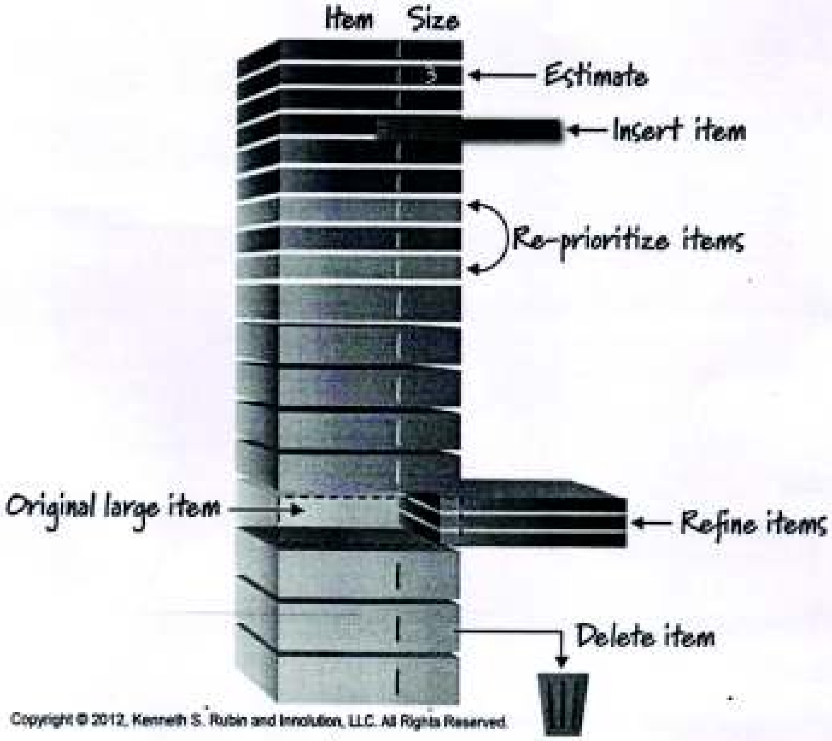
Product Backlog Refinement

- Product Backlog refinement is the act of adding detail, estimates, and order to items in the Product Backlog
- Assists team in preparing for upcoming sprints
- Forward Looking
- Refinement usually consumes no more than 10% of the capacity of the Development Team

True or False (T/F)

- Only the product owner must worry about Refinement F
- Refinement happens once every release F
- During Refinement, the team helps prioritize and estimates product backlog items T
- New backlog items can be created during Refinement sessions T
- Every team does Refinement exactly alike F
- Refinement includes refining backlog items to include more detail, identify dependencies, break into smaller pieces, and/or increasing understanding of the work T
- Meeting to refine the product backlog in the middle of a sprint disrupts the team and provides little value F →

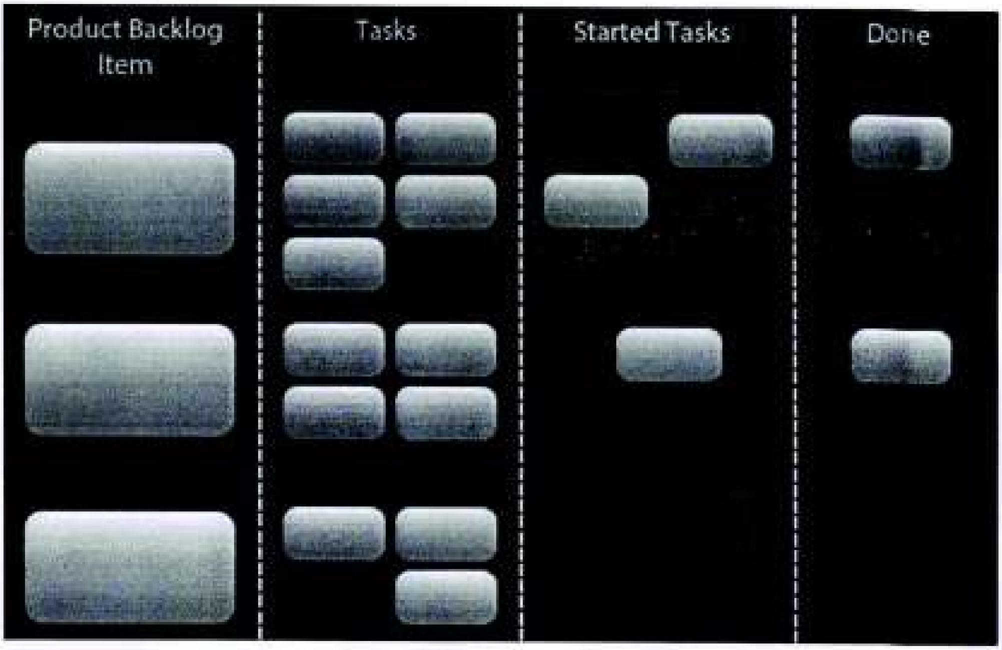
Product Backlog



Copyright © 2012, Kenneth S. Rubin and Innolution, LLC. All Rights Reserved.

- An ordered list of items for the product aka Product Backlog Items (PBIs)
- Product Owner is responsible and accountable for maintaining the Product Backlog
- Order of items can change at any time in the product backlog
- Product Backlog will have the features, architectural, infrastructural, etc

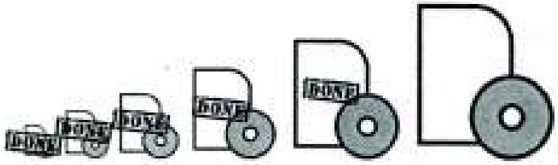
Sprint Backlog



- Subset of Product Backlog items selected for the Sprint
- The Sprint Backlog makes visible all the work that the Team identifies as necessary to meet the Sprint Goal
- The Sprint Backlog is a highly visible, real-time picture of the work
- Owned and managed by the team

PBI s → Product backlog Elins

Increment → Delivery value to the customer



Increment is the sum of all the PBIs completed during a Sprint and the value of the increments of all previous Sprints. Each increment must be "Done" (no work remaining), and is potentially usable by customer. A "Done" Increment provides transparency to everyone.

What are the various "things" that should be completed if the user (customer) should use the product after every Sprint?

Sprint Review with status "Done"

What are the advantages of having a "Releasable Product Increment" every Sprint?

Transparency to everyone

→ Business value

→ Trust from the customer → estimation problems Resolves

→

Estimation = Stable Team + Time

Definition of Done is Owned by Scrum Team

For one team Done can be Dev + Testing

2nd " Dev + Testing + UAT

3rd " " + Production

But Not at the line item count

Definition of Done

; the better quanlity product

Definition of Done (DoD) for a PBI (or a user story) is the set of quality standards that apply broadly to all (or at least most) PBIs, whereas acceptance criteria (aka Conditions of Satisfaction) apply to a single PBI or user story.

- Code is checked in and tested
- Automated test coverage >80%
- Passed UAT and accepted by PO
- All changes (build/deploy/configuration) implemented and documented
- Relevant documentation/diagrams produced and/or updated

– the Done is updated . Sprint Retro

Sprint Ready

Sample criteria for Sprint readiness:

- PBI has acceptance criteria
- Small enough to be completed in a Sprint
- PBI can be estimated by the team
- Dependencies/assumptions are identified

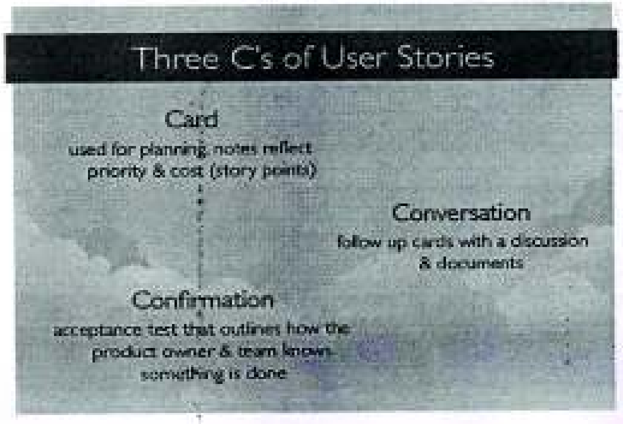
Technical Debt

Technical Debt includes those *internal* things that you choose not to do now, but which will impede future development if left undone. This includes deferred refactoring.

Technical Debt is a measure of how untidy or out-of-date the development work area for a product is. – Dave Smith

What is a User Story?

A user story is a short-written description, told from the perspective of a user, used for planning the system and as a reminder to discuss with your customer the details of the functionality.

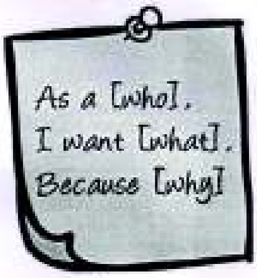


Three C's of User Stories

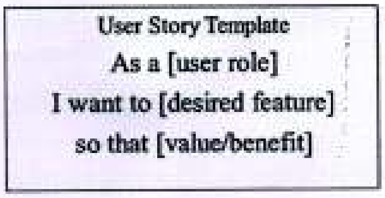
Card
used for planning notes reflect priority & cost (story points)

Conversation
follow up cards with a discussion & documents

Confirmation
acceptance test that outlines how the product owner & team known something is done



As a [who],
I want [what],
Because [why]



User Story Template
As a [user role]
I want to [desired feature]
so that [value/benefit]

Example

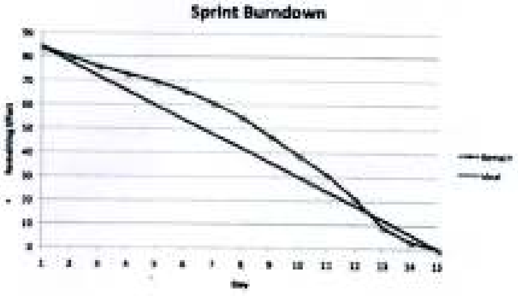
As a customer, I want to be able to pay with a credit card so that I can buy products online.

Acceptance Criteria:

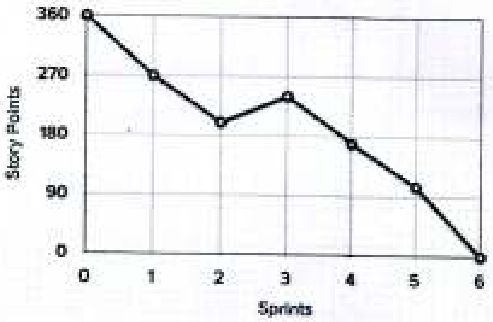
Validate that MasterCard, VISA and American Express are accepted.

Validate that no other cards are accepted.

Burndown Charts



Release Burndown



- Sprint Burndown chart is an information radiator owned and managed by the team
- ScrumMaster ensures the chart is updated at least once a day
- A tool to increase visibility
- Release Burndown chart is an information radiator owned and managed by the PO

1 3 5 8 13

→ Buckets

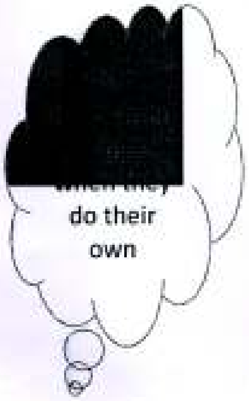
quickly Below 2 Mins	cups 4 Mins	TUB
1-10 capacity Bigger the size →	2-30 capacity	5-100 is harder merely

Pair Share

- Pair Up with someone else from the class.
- How do you currently estimate?
- How is it working for you?

Estimation Basics

- Estimate size/complexity and derive duration empirically within an established range
- Goal is to get to a useful estimate with minimal effort
- Estimates are informed guesses, not commitments
- Use estimates to
  - Plan how many stories can fit in a Sprint
  - Plan how many stories can fit in a release
  - Derive cost of the release



Estimate Size derived Duration + Range

↓ product Backlog Refinment

↓ Sprint planning

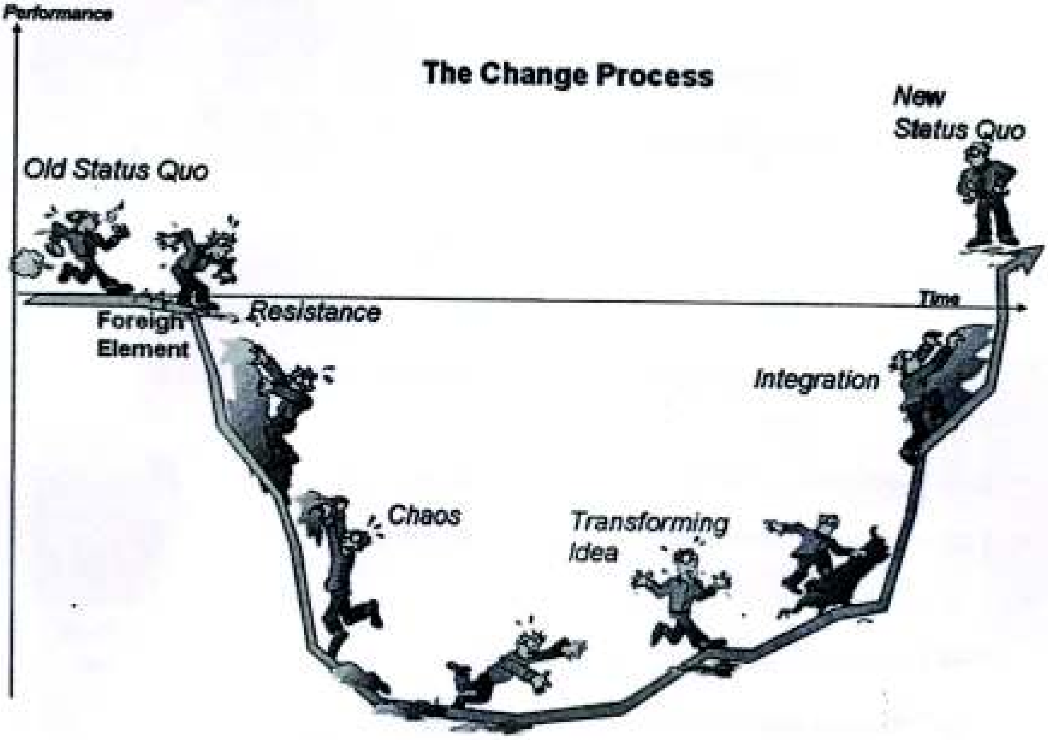
upper limit
low limit

Ldr Scrum Master

Mountain goat software

website

Remember!


Performance
The Change Process
New Status Quo
Old Status Quo
Time
Foreign Element
Resistance
Integration
Chaos
Transforming Idea

Stay in Touch!

Vivek Angiras
Trainer and Coach

vivangiras@gmail.com
@vivangiras
vivekangiras

CST
CAL
CSM
CSPO
CSD
CSP-SM
CSP-PO
LeSS
Certified Practitioner
SCALED AGILE FRAMEWORK PROGRAM CONSULTANT
Kanban Coaching Professional Masterclass
PMI
TRAINING FROM THE BACK OF THE ROOM
trained facilitator
Wyeth
Pfizer
UNOS UNITED NETWORK FOR ORGAN SHARING
RBS
Freddie Mac
We make home possible
STATE STREET GLOBAL ADVISORS
Capital One
LOWE'S
AIG

Appendix

Reference Materials --- Add to the list as we discuss during the class

- Agile Estimation and Planning – Mike Cohn
- Flow: The Psychology of Optimal Experience –Mihaly Csikezentmihalyi
- Nonviolent Communication: A language of life: Life changing tools for healthy relationships –Marshall Rosenberg
- Agile Retrospectives: Making Good Teams Great – Ester Derby and Diana Larsen
- Refactoring: Improving the Design of Existing Code - Martin Fowler
- Agile Project Management with Scrum – Ken Schwaber

Steve Blank mvp

- ______________________________
- ______________________________
- ______________________________
- ______________________________
- ______________________________
- ______________________________

Check for Understanding (team: standing)

"Am I exploring if my colleagues understand my point clearly?"

"Am I exploring if I understand my colleagues' point?"

"Am I *really actively* listening?"

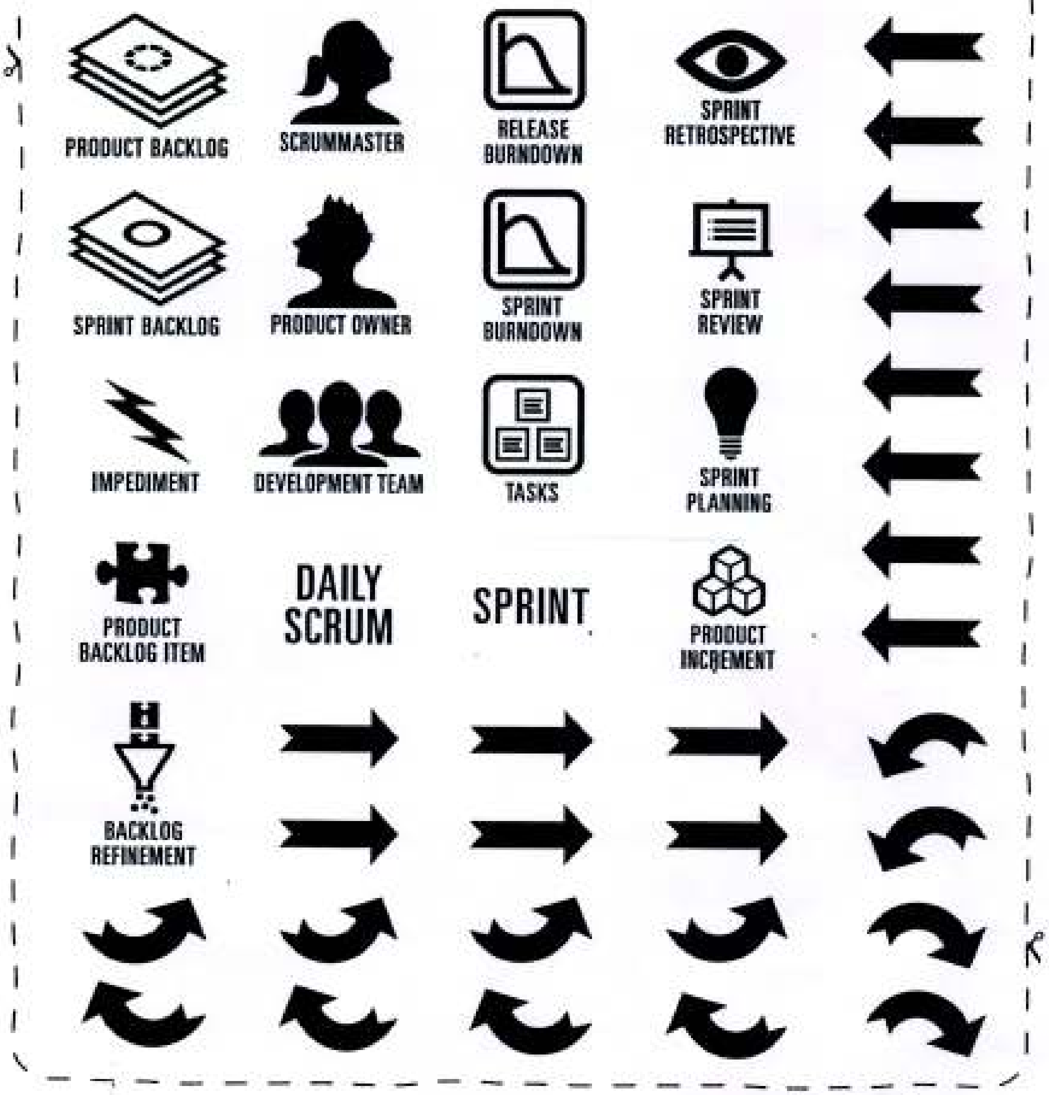
"Am I trying to help everyone feel heard and included?"

Reflections (team: standing)

Each person: most **noteworthy** or **interesting idea** since last?

Each person: then, write it on a post-it and post it on the wall

Build your own Scrum



Use this exercise in your next course. Download it from weisbart.com/byos

Can be

Yes..

should be

x10

→ ScrumMaster has to get the cross-functional Team

⇒ should convence the P.O

→ Scrum will Not go for hiring process

⇒ But ScrumMaster + P.O can involve in hiring process to be good.

Fill In The Blanks – Terminology Worksheet

Use these words to complete the sentences below:

ScrumMaster, The Development Team, Sprint Retrospective, Product Backlog, Sprint Backlog, Sprint Planning, Daily Scrum, Product Increment, Impediment, Product Owner, Sprint Review, Sprint

1. A Sprint is a time-box of one month or less during which a "Done", useable, and potentially releasable Product Increment is created.

2. The product Backlog is an ordered list of everything that might be needed in the product.

3. The sum of all the Product Backlog items completed during a Sprint and the value of the increments of all previous Sprints is called a product Increment

4. The Sprint Backlog is a set of Product Backlog items selected for the Sprint, plus a plan for delivering the product Increment and realizing the Sprint Goal.

5. A risk or issue that is impeding the team's progress is called a Impediment.

6. The Scrum Master is responsible for ensuring Scrum is understood and enacted.

7. The Product Owner is responsible for maximizing the value of the product and the work of the Development Team.

8. The The Developmend Team consists of professionals who do the work of delivering a potentially releasable Increment of "Done" product at the end of each Sprint.

9. Work to be performed in the Sprint is planned during the Sprint planning meeting.

10. 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

11. A Sprint Review is held at the end of the Sprint to inspect the Increment and adapt the Product Backlog if needed

12. The Sprint Retrospective is an opportunity for the Scrum Team to inspect itself and create a plan for improvements to be enacted during the next Sprint.

Burn-Down Charts:

A burn-down chart is a graphical depiction of a team's progress towards completing their sprint goal. Typically, hours, tasks, or story points are charted on the "Y" axis and time (in days) is charted on the "X" axis. Every day the team's progress (aka velocity) is plotted. The burn-down chart becomes a graphical representation as to whether the team expects to finish on-time or not. The real value of the burn-down chart is being alerted to the need to "have a conversation" about where the team stands.

Build A Burn-Down Chart Instructions:

As an individual, create a burn-down chart utilizing the following data (Note: The data below reflects the 1st week's progress of a 2-week long sprint).

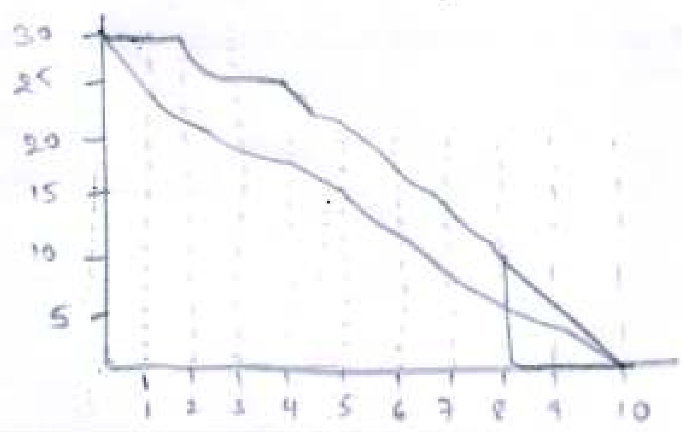
Day	Planned	Work Remaining	Daily Completed
0	30	30	0
1	27	30	0
2	24	27	3
3	21	27	0
4	18	25	2
5	15	21	4

As a team, analyze the burn-down chart and discuss the results. Is the sprint on-track? Yes or no? Why?

Complete your burn-down chart by plotting week 2's data.

Day	Planned	Work Remaining	Daily Completed
6	12	15	6
7	9	12	3
8	6	6	6
9	3	0	6
10	0	0	0

As a team, analyze the burn-down chart and discuss the results. Did the sprint conclude on-track? Yes or no? Why?



HW

TEAM

The Scrum Guide™

The Definitive Guide to Scrum: The Rules of the Game

Who is Responsibity for hiring

November 2017

Ken Schwaber

Developed and sustained by Scrum creators: Ken Schwaber and Jeff Sutherland

Table of Contents

Purpose of the Scrum Guide

Scrum is a framework for developing, delivering, and sustaining complex products. This Guide contains the definition of Scrum. This definition consists of Scrum's roles, events, artifacts, and the rules that bind them together. Ken Schwaber and Jeff Sutherland developed Scrum; the Scrum Guide is written and provided by them. Together, they stand behind the Scrum Guide.

Definition of Scrum

Scrum (n): A framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value.

Scrum is:

- Lightweight
- Simple to understand
- Difficult to master

Scrum is a process framework that has been used to manage work on complex products since the early 1990s. Scrum is not a process, technique, or definitive method. Rather, it is a framework within which you can employ various processes and techniques. Scrum makes clear the relative efficacy of your product management and work techniques so that you can continuously improve the product, the team, and the working environment.

The Scrum framework consists of Scrum Teams and their associated roles, events, artifacts, and rules. Each component within the framework serves a specific purpose and is essential to Scrum's success and usage.

The rules of Scrum bind together the roles, events, and artifacts, governing the relationships and interaction between them. The rules of Scrum are described throughout the body of this document.

Specific tactics for using the Scrum framework vary and are described elsewhere.

Uses of Scrum

Scrum was initially developed for managing and developing products. Starting in the early 1990s, Scrum has been used extensively, worldwide, to:

1. Research and identify viable markets, technologies, and product capabilities;
2. Develop products and enhancements;
3. Release products and enhancements, as frequently as many times per day;
4. Develop and sustain Cloud (online, secure, on-demand) and other operational environments for product use; and,
5. Sustain and renew products.

Scrum has been used to develop software, hardware, embedded software, networks of interacting function, autonomous vehicles, schools, government, marketing, managing the operation of organizations and almost everything we use in our daily lives, as individuals and societies.

As technology, market, and environmental complexities and their interactions have rapidly increased, Scrum's utility in dealing with complexity is proven daily.

Scrum proved especially effective in iterative and incremental knowledge transfer. Scrum is now widely used for products, services, and the management of the parent organization.

The essence of Scrum is a small team of people. The individual team is highly flexible and adaptive. These strengths continue operating in single, several, many, and networks of teams that develop, release, operate and sustain the work and work products of thousands of people. They collaborate and interoperate through sophisticated development architectures and target release environments.

When the words "develop" and "development" are used in the Scrum Guide, they refer to complex work, such as those types identified above.

Scrum Theory

Scrum is founded on empirical process control theory, or empiricism. Empiricism asserts that knowledge comes from experience *and* making decisions based on what is known. Scrum employs an iterative, incremental approach to optimize predictability and control risk.

Three pillars uphold every implementation of empirical process control: transparency, inspection, and adaptation.

Transparency

Significant aspects of the process must be visible to those responsible for the outcome. Transparency requires those aspects be defined by a common standard so observers share a common understanding of what is being seen.

For example

- A common language referring to the process must be shared by all participants; and,
- Those performing the work and those inspecting the resulting increment must share a common definition of "Done".

Inspection

Scrum users must frequently inspect Scrum artifacts and progress toward a Sprint Goal to detect undesirable variances. Their inspection should not be so frequent that inspection gets in the way of the work. Inspections are most beneficial when diligently performed by skilled inspectors at the point of work.

Adaptation

If an inspector determines that one or more aspects of a process deviate outside acceptable limits, and that the resulting product will be unacceptable, the process or the material being processed must be adjusted. An adjustment must be made as soon as possible to minimize further deviation.

Scrum prescribes four formal events for inspection and adaptation, as described in the *Scrum Events* section of this document:

- Sprint Planning
- Daily Scrum
- Sprint Review
- Sprint Retrospective

Scrum Values

When the values of commitment, courage, focus, openness and respect are embodied and lived by the Scrum Team, the Scrum pillars of transparency, inspection, and adaptation come to life and build trust for everyone. The Scrum Team members learn and explore those values as they work with the Scrum roles, events, and artifacts.

Successful use of Scrum depends on people becoming more proficient in living these five values. People personally commit to achieving the goals of the Scrum Team. The Scrum Team members have courage to do the right thing and work on tough problems. Everyone focuses on the work of the Sprint and the goals of the Scrum Team. The Scrum Team and its stakeholders agree to be open about all the work and the challenges with performing the work. Scrum Team members respect each other to be capable, independent people.

The Scrum Team

The Scrum Team consists of a Product Owner, the Development Team, and a Scrum Master. Scrum Teams are self-organizing and cross-functional. Self-organizing teams choose how best to accomplish their work, rather than being directed by others outside the team. Cross-functional teams have all competencies needed to accomplish the work without depending on others not part of the team. The team model in Scrum is designed to optimize flexibility, creativity, and productivity. The Scrum Team has proven itself to be increasingly effective for all the earlier stated uses, and any complex work.

Scrum Teams deliver products iteratively and incrementally, maximizing opportunities for feedback. Incremental deliveries of "Done" product ensure a potentially useful version of working product is always available.

The Product Owner

The Product Owner is responsible for maximizing the value of the product resulting from work of the Development Team. How this is done may vary widely across organizations, Scrum Teams, and individuals.

The Product Owner is the sole person responsible for managing the Product Backlog. Product Backlog management includes:

- Clearly expressing Product Backlog items;
- Ordering the items in the Product Backlog to best achieve goals and missions;
- Optimizing the value of the work the Development Team performs;
- Ensuring that the Product Backlog is visible, transparent, and clear to all, and shows what the Scrum Team will work on next; and,
- Ensuring the Development Team understands items in the Product Backlog to the level needed.

The Product Owner may do the above work, or have the Development Team do it. However, the Product Owner remains accountable.

The Product Owner is one person, not a committee. The Product Owner may represent the desires of a committee in the Product Backlog, but those wanting to change a Product Backlog item's priority must address the Product Owner.

For the Product Owner to succeed, the entire organization must respect his or her decisions. The Product Owner's decisions are visible in the content and ordering of the Product Backlog. No one can force the Development Team to work from a different set of requirements.

The Development Team

The Development Team consists of professionals who do the work of delivering a potentially releasable Increment of "Done" product at the end of each Sprint. A "Done" increment is required at the Sprint Review. Only members of the Development Team create the Increment.

Development Teams are structured and empowered by the organization to organize and manage their own work. The resulting synergy optimizes the Development Team's overall efficiency and effectiveness.

Development Teams have the following characteristics:

- They are self-organizing. No one (not even the Scrum Master) tells the Development Team how to turn Product Backlog into Increments of potentially releasable functionality;
- Development Teams are cross-functional, with all the skills as a team necessary to create a product Increment;
- Scrum recognizes no titles for Development Team members, regardless of the work being performed by the person;
- Scrum recognizes no sub-teams in the Development Team, regardless of domains that need to be addressed like testing, architecture, operations, or business analysis; and,
- Individual Development Team members may have specialized skills and areas of focus, but accountability belongs to the Development Team as a whole.

Development Team Size

Optimal Development Team size is small enough to remain nimble and large enough to complete significant work within a Sprint. Fewer than three Development Team members decrease interaction and results in smaller productivity gains. Smaller Development Teams may encounter skill constraints during the Sprint, causing the Development Team to be unable to deliver a potentially releasable Increment. Having more than nine members requires too much coordination. Large Development Teams generate too much complexity for an empirical process to be useful. The Product Owner and Scrum Master roles are not included in this count unless they are also executing the work of the Sprint Backlog.

The Scrum Master

The Scrum Master is responsible for promoting and supporting Scrum as defined in the Scrum Guide. Scrum Masters do this by helping everyone understand Scrum theory, practices, rules, and values.

The Scrum Master is a servant-leader for the Scrum Team. The Scrum Master helps those outside the Scrum Team understand which of their interactions with the Scrum Team are helpful and which aren't. The Scrum Master helps everyone change these interactions to maximize the value created by the Scrum Team.

Scrum Master Service to the Product Owner

The Scrum Master serves the Product Owner in several ways, including:

- Ensuring that goals, scope, and product domain are understood by everyone on the Scrum Team as well as possible;
- Finding techniques for effective Product Backlog management;
- Helping the Scrum Team understand the need for clear and concise Product Backlog items;
- Understanding product planning in an empirical environment;
- Ensuring the Product Owner knows how to arrange the Product Backlog to maximize value;
- Understanding and practicing agility; and,
- Facilitating Scrum events as requested or needed.

Scrum Master Service to the Development Team

The Scrum Master serves the Development Team in several ways, including:

- Coaching the Development Team in self-organization and cross-functionality;
- Helping the Development Team to create high-value products;
- Removing impediments to the Development Team's progress;
- Facilitating Scrum events as requested or needed; and,
- Coaching the Development Team in organizational environments in which Scrum is not yet fully adopted and understood.

Scrum Master Service to the Organization

The Scrum Master serves the organization in several ways, including:

- Leading and coaching the organization in its Scrum adoption;
- Planning Scrum implementations within the organization;
- Helping employees and stakeholders understand and enact Scrum and empirical product development;
- Causing change that increases the productivity of the Scrum Team; and,
- Working with other Scrum Masters to increase the effectiveness of the application of Scrum in the organization.

Scrum Events

Prescribed events are used in Scrum to create regularity and to minimize the need for meetings not defined in Scrum. All events are time-boxed events, such that every event has a maximum duration. Once a Sprint begins, its duration is fixed and cannot be shortened or lengthened. The remaining events may end whenever the purpose of the event is achieved, ensuring an appropriate amount of time is spent without allowing waste in the process.

Other than the Sprint itself, which is a container for all other events, each event in Scrum is a formal opportunity to inspect and adapt something. These events are specifically designed to enable critical transparency and inspection. Failure to include any of these events results in reduced transparency and is a lost opportunity to inspect and adapt.

The Sprint

The heart of Scrum is a Sprint, a time-box of one month or less during which a "Done", useable, and potentially releasable product Increment is created. Sprints have consistent durations throughout a development effort. A new Sprint starts immediately after the conclusion of the previous Sprint.

Sprints contain and consist of the Sprint Planning, Daily Scrums, the development work, the Sprint Review, and the Sprint Retrospective.

During the Sprint:

- No changes are made that would endanger the Sprint Goal;
- Quality goals do not decrease; and,
- Scope may be clarified and re-negotiated between the Product Owner and Development Team as more is learned.

Each Sprint may be considered a project with no more than a one-month horizon. Like projects, Sprints are used to accomplish something. Each Sprint has a goal of what is to be built, a design and flexible plan that will guide building it, the work, and the resultant product increment.

Sprints are limited to one calendar month. When a Sprint's horizon is too long the definition of what is being built may change, complexity may rise, and risk may increase. Sprints enable predictability by ensuring inspection and adaptation of progress toward a Sprint Goal at least every calendar month. Sprints also limit risk to one calendar month of cost.

Cancelling a Sprint

A Sprint can be cancelled before the Sprint time-box is over. Only the Product Owner has the authority to cancel the Sprint, although he or she may do so under influence from the stakeholders, the Development Team, or the Scrum Master.

A Sprint would be cancelled if the Sprint Goal becomes obsolete. This might occur if the company changes direction or if market or technology conditions change. In general, a Sprint should be cancelled if it no longer makes sense given the circumstances. But, due to the short duration of Sprints, cancellation rarely makes sense.

When a Sprint is cancelled, any completed and "Done" Product Backlog items are reviewed. If part of the work is potentially releasable, the Product Owner typically accepts it. All incomplete Product Backlog Items are re-estimated and put back on the Product Backlog. The work done on them depreciates quickly and must be frequently re-estimated.

Sprint cancellations consume resources, since everyone regroups in another Sprint Planning to start another Sprint. Sprint cancellations are often traumatic to the Scrum Team, and are very uncommon.

Sprint Planning

The work to be performed in the Sprint is planned at the Sprint Planning. This plan is created by the collaborative work of the entire Scrum Team.

Sprint Planning is time-boxed to a maximum of eight hours for a one-month Sprint. For shorter Sprints, the event is usually shorter. The Scrum Master ensures that the event takes place and that attendants understand its purpose. The Scrum Master teaches the Scrum Team to keep it within the time-box.

Sprint Planning answers the following:

- What can be delivered in the Increment resulting from the upcoming Sprint?
- How will the work needed to deliver the Increment be achieved?

Topic One: What can be done this Sprint?

The Development Team works to forecast the functionality that will be developed during the Sprint. The Product Owner discusses the objective that the Sprint should achieve and the Product Backlog items that, if completed in the Sprint, would achieve the Sprint Goal. The entire Scrum Team collaborates on understanding the work of the Sprint.

The input to this meeting is the Product Backlog, the latest product Increment, projected capacity of the Development Team during the Sprint, and past performance of the Development Team. The number of items selected from the Product Backlog for the Sprint is solely up to the Development Team. Only the Development Team can assess what it can accomplish over the upcoming Sprint.

During Sprint Planning the Scrum Team also crafts a Sprint Goal. The Sprint Goal is an objective that will be met within the Sprint through the implementation of the Product Backlog, and it provides guidance to the Development Team on why it is building the Increment.

Topic Two: How will the chosen work get done?

Having set the Sprint Goal and selected the Product Backlog items for the Sprint, the Development Team decides how it will build this functionality into a "Done" product Increment during the Sprint. The Product Backlog items selected for this Sprint plus the plan for delivering them is called the Sprint Backlog.

The Development Team usually starts by designing the system and the work needed to convert the Product Backlog into a working product Increment. Work may be of varying size, or estimated effort. However, enough work is planned during Sprint Planning for the Development Team to forecast what it believes it can do in the upcoming Sprint. Work planned for the first days of the Sprint by the Development Team is decomposed by the end of this meeting, often to units of one day or less. The Development Team self-organizes to undertake the work in the Sprint Backlog, both during Sprint Planning and as needed throughout the Sprint.

The Product Owner can help to clarify the selected Product Backlog items and make trade-offs. If the Development Team determines it has too much or too little work, it may renegotiate the selected Product Backlog items with the Product Owner. The Development Team may also invite other people to attend to provide technical or domain advice.

By the end of the Sprint Planning, the Development Team should be able to explain to the Product Owner and Scrum Master how it intends to work as a self-organizing team to accomplish the Sprint Goal and create the anticipated Increment.

Sprint Goal

The Sprint Goal is an objective set for the Sprint that can be met through the implementation of Product Backlog. It provides guidance to the Development Team on why it is building the Increment. It is created during the Sprint Planning meeting. The Sprint Goal gives the Development Team some flexibility regarding the functionality implemented within the Sprint. The selected Product Backlog items deliver one coherent function, which can be the Sprint Goal. The Sprint Goal can be any other coherence that causes the Development Team to work together rather than on separate initiatives.

As the Development Team works, it keeps the Sprint Goal in mind. In order to satisfy the Sprint Goal, it implements functionality and technology. If the work turns out to be different than the Development Team expected, they collaborate with the Product Owner to negotiate the scope of Sprint Backlog within the Sprint.

Daily Scrum

The Daily Scrum is a 15-minute time-boxed event for the Development Team. The Daily Scrum is held every day of the Sprint. At it, the Development Team plans work for the next 24 hours. This optimizes team collaboration and performance by inspecting the work since the last Daily Scrum and forecasting upcoming Sprint work. The Daily Scrum is held at the same time and place each day to reduce complexity.

The Development Team uses the Daily Scrum to inspect progress toward the Sprint Goal and to inspect how progress is trending toward completing the work in the Sprint Backlog. The Daily Scrum optimizes the probability that the Development Team will meet the Sprint Goal. Every day, the Development Team should understand how it intends to work together as a self-organizing team to accomplish the Sprint Goal and create the anticipated Increment by the end of the Sprint.

The structure of the meeting is set by the Development Team and can be conducted in different ways if it focuses on progress toward the Sprint Goal. Some Development Teams will use questions, some will be more discussion based. Here is an example of what might be used:

- What did I do yesterday that helped the Development Team meet the Sprint Goal?
- What will I do today to help the Development Team meet the Sprint Goal?
- Do I see any impediment that prevents me or the Development Team from meeting the Sprint Goal?

The Development Team or team members often meet immediately after the Daily Scrum for detailed discussions, or to adapt, or replan, the rest of the Sprint's work.

The Scrum Master ensures that the Development Team has the meeting, but the Development Team is responsible for conducting the Daily Scrum. The Scrum Master teaches the Development Team to keep the Daily Scrum within the 15-minute time-box.

The Daily Scrum is an internal meeting for the Development Team. If others are present, the Scrum Master ensures that they do not disrupt the meeting.

Daily Scrums improve communications, eliminate other meetings, identify impediments to development for removal, highlight and promote quick decision-making, and improve the Development Team's level of knowledge. This is a key inspect and adapt meeting.

Sprint Review

A Sprint Review is held at the end of the Sprint to inspect the Increment and adapt the Product Backlog if needed. During the Sprint Review, the Scrum Team and stakeholders collaborate about what was done in the Sprint. Based on that and any changes to the Product Backlog during the Sprint, attendees collaborate on the next things that could be done to optimize value. This is an informal meeting, not a status meeting, and the presentation of the Increment is intended to elicit feedback and foster collaboration.

This is at most a four-hour meeting for one-month Sprints. For shorter Sprints, the event is usually shorter. The Scrum Master ensures that the event takes place and that attendees understand its purpose. The Scrum Master teaches everyone involved to keep it within the time-box.

The Sprint Review includes the following elements:

- Attendees include the Scrum Team and key stakeholders invited by the Product Owner;
- The Product Owner explains what Product Backlog items have been "Done" and what has not been "Done";
- The Development Team discusses what went well during the Sprint, what problems it ran into, and how those problems were solved;
- The Development Team demonstrates the work that it has "Done" and answers questions about the Increment;
- The Product Owner discusses the Product Backlog as it stands. He or she projects likely target and delivery dates based on progress to date (if needed);
- The entire group collaborates on what to do next, so that the Sprint Review provides valuable input to subsequent Sprint Planning;
- Review of how the marketplace or potential use of the product might have changed what is the most valuable thing to do next; and,
- Review of the timeline, budget, potential capabilities, and marketplace for the next anticipated releases of functionality or capability of the product.

The result of the Sprint Review is a revised Product Backlog that defines the probable Product Backlog items for the next Sprint. The Product Backlog may also be adjusted overall to meet new opportunities.

Sprint Retrospective

The Sprint Retrospective is an opportunity for the Scrum Team to inspect itself and create a plan for improvements to be enacted during the next Sprint.

The Sprint Retrospective occurs after the Sprint Review and prior to the next Sprint Planning. This is at most a three-hour meeting for one-month Sprints. For shorter Sprints, the event is usually shorter. The Scrum Master ensures that the event takes place and that attendants understand its purpose.

The Scrum Master ensures that the meeting is positive and productive. The Scrum Master teaches all to keep it within the time-box. The Scrum Master participates as a peer team member in the meeting from the accountability over the Scrum process.

The purpose of the Sprint Retrospective is to:

- Inspect how the last Sprint went with regards to people, relationships, process, and tools;
- Identify and order the major items that went well and potential improvements; and,
- Create a plan for implementing improvements to the way the Scrum Team does its work.

The Scrum Master encourages the Scrum Team to improve, within the Scrum process framework, its development process and practices to make it more effective and enjoyable for the next Sprint. During each Sprint Retrospective, the Scrum Team plans ways to increase product quality by improving work processes or adapting the definition of "Done", if appropriate and not in conflict with product or organizational standards.

By the end of the Sprint Retrospective, the Scrum Team should have identified improvements that it will implement in the next Sprint. Implementing these improvements in the next Sprint is the adaptation to the inspection of the Scrum Team itself. Although improvements may be implemented at any time, the Sprint Retrospective provides a formal opportunity to focus on inspection and adaptation.

Scrum Artifacts

Scrum's artifacts represent work or value to provide transparency and opportunities for inspection and adaptation. Artifacts defined by Scrum are specifically designed to maximize transparency of key information so that everybody has the same understanding of the artifact.

Product Backlog

The Product Backlog is an ordered list of everything that is known to be needed in the product. It is the single source of requirements for any changes to be made to the product. The Product Owner is responsible for the Product Backlog, including its content, availability, and ordering.

A Product Backlog is never complete. The earliest development of it lays out the initially known and best-understood requirements. The Product Backlog evolves as the product and the environment in which it will be used evolves. The Product Backlog is dynamic; it constantly changes to identify what the product needs to be appropriate, competitive, and useful. If a product exists, its Product Backlog also exists.

The Product Backlog lists all features, functions, requirements, enhancements, and fixes that constitute the changes to be made to the product in future releases. Product Backlog items have the attributes of a description, order, estimate, and value. Product Backlog items often include test descriptions that will prove its completeness when "Done."

As a product is used and gains value, and the marketplace provides feedback, the Product Backlog becomes a larger and more exhaustive list. Requirements never stop changing, so a Product Backlog is a living artifact. Changes in business requirements, market conditions, or technology may cause changes in the Product Backlog.

Multiple Scrum Teams often work together on the same product. One Product Backlog is used to describe the upcoming work on the product. A Product Backlog attribute that groups items may then be employed.

Product Backlog refinement is the act of adding detail, estimates, and order to items in the Product Backlog. This is an ongoing process in which the Product Owner and the Development Team collaborate on the details of Product Backlog items. During Product Backlog refinement, items are reviewed and revised. The Scrum Team decides how and when refinement is done. Refinement usually consumes no more than 10% of the capacity of the Development Team. However, Product Backlog items can be updated at any time by the Product Owner or at the Product Owner's discretion.

Higher ordered Product Backlog items are usually clearer and more detailed than lower ordered ones. More precise estimates are made based on the greater clarity and increased detail; the lower the order, the less detail. Product Backlog items that will occupy the Development Team for the upcoming Sprint are refined so that any one item can reasonably be "Done" within the Sprint time-box. Product Backlog items that can be "Done" by the Development Team within one Sprint are deemed "Ready" for selection in a Sprint Planning. Product Backlog items usually acquire this degree of transparency through the above described refining activities.

The Development Team is responsible for all estimates. The Product Owner may influence the Development Team by helping it understand and select trade-offs, but the people who will perform the work make the final estimate.

Monitoring Progress Toward Goals

At any point in time, the total work remaining to reach a goal can be summed. The Product Owner tracks this total work remaining at least every Sprint Review. The Product Owner compares this amount with work remaining at previous Sprint Reviews to assess progress toward completing projected work by the desired time for the goal. This information is made transparent to all stakeholders.

Various projective practices upon trending have been used to forecast progress, like burn-downs, burn-ups, or cumulative flows. These have proven useful. However, these do not replace the importance of empiricism. In complex environments, what will happen is unknown. Only what has already happened may be used for forward-looking decision-making.

Sprint Backlog

The Sprint Backlog is the set of Product Backlog items selected for the Sprint, plus a plan for delivering the product Increment and realizing the Sprint Goal. The Sprint Backlog is a forecast by the Development Team about what functionality will be in the next Increment and the work needed to deliver that functionality into a "Done" Increment.

The Sprint Backlog makes visible all the work that the Development Team identifies as necessary to meet the Sprint Goal. To ensure continuous improvement, it includes at least one high priority process improvement identified in the previous Retrospective meeting.

The Sprint Backlog is a plan with enough detail that changes in progress can be understood in the Daily Scrum. The Development Team modifies the Sprint Backlog throughout the Sprint, and the Sprint Backlog emerges during the Sprint. This emergence occurs as the Development Team works through the plan and learns more about the work needed to achieve the Sprint Goal.

As new work is required, the Development Team adds it to the Sprint Backlog. As work is performed or completed, the estimated remaining work is updated. When elements of the plan are deemed unnecessary, they are removed. Only the Development Team can change its Sprint Backlog during a Sprint. The Sprint Backlog is a highly visible, real-time picture of the work that the Development Team plans to accomplish during the Sprint, and it belongs solely to the Development Team.

Monitoring Sprint Progress

At any point in time in a Sprint, the total work remaining in the Sprint Backlog can be summed. The Development Team tracks this total work remaining at least for every Daily Scrum to project the likelihood of achieving the Sprint Goal. By tracking the remaining work throughout the Sprint, the Development Team can manage its progress.

Increment

The Increment is the sum of all the Product Backlog items completed during a Sprint and the value of the increments of all previous Sprints. At the end of a Sprint, the new Increment must be "Done," which means it must be in useable condition and meet the Scrum Team's definition of "Done." An increment is a body of inspectable, done work that supports empiricism at the end of the Sprint. The increment is a step toward a vision or goal. The increment must be in useable condition regardless of whether the Product Owner decides to release it.

Artifact Transparency

Scrum relies on transparency. Decisions to optimize value and control risk are made based on the perceived state of the artifacts. To the extent that transparency is complete, these decisions have a sound basis. To the extent that the artifacts are incompletely transparent, these decisions can be flawed, value may diminish and risk may increase.

The Scrum Master must work with the Product Owner, Development Team, and other involved parties to understand if the artifacts are completely transparent. There are practices for coping with incomplete transparency; the Scrum Master must help everyone apply the most appropriate practices in the absence of complete transparency. A Scrum Master can detect incomplete transparency by inspecting the artifacts, sensing patterns, listening closely to what is being said, and detecting differences between expected and real results.

The Scrum Master's job is to work with the Scrum Team and the organization to increase the transparency of the artifacts. This work usually involves learning, convincing, and change. Transparency doesn't occur overnight, but is a path.

Definition of "Done"

When a Product Backlog item or an Increment is described as "Done", everyone must understand what "Done" means. Although this may vary significantly per Scrum Team, members must have a shared understanding of what it means for work to be complete, to ensure transparency. This is the definition of "Done" for the Scrum Team and is used to assess when work is complete on the product Increment.

The same definition guides the Development Team in knowing how many Product Backlog items it can select during a Sprint Planning. The purpose of each Sprint is to deliver Increments of potentially releasable functionality that adhere to the Scrum Team's current definition of "Done."

Development Teams deliver an Increment of product functionality every Sprint. This Increment is useable, so a Product Owner may choose to immediately release it. If the definition of "Done" for an increment is part of the conventions, standards or guidelines of the development organization, all Scrum Teams must follow it as a minimum.

If "Done" for an increment is **not** a convention of the development organization, the Development Team of the Scrum Team must define a definition of "Done" appropriate for the product. If there are multiple Scrum Teams working on the system or product release, the Development Teams on all the Scrum Teams must mutually define the definition of "Done."

Each Increment is additive to all prior Increments and thoroughly tested, ensuring that all Increments work together.

As Scrum Teams mature, it is expected that their definitions of "Done" will expand to include more stringent criteria for higher quality. New definitions, as used, may uncover work to be done in previously "Done" increments. Any one product or system should have a definition of "Done" that is a standard for any work done on it.

End Note

Scrum is free and offered in this Guide. Scrum's roles, events, artifacts, and rules are immutable and although implementing only parts of Scrum is possible, the result is not Scrum. Scrum exists only in its entirety and functions well as a container for other techniques, methodologies, and practices.

Acknowledgements

People

Of the thousands of people who have contributed to Scrum, we should single out those who were instrumental at the start: Jeff Sutherland worked with Jeff McKenna and John Scumniotales, and Ken Schwaber worked with Mike Smith and Chris Martin, and all of them worked together. Many others contributed in the ensuing years and without their help Scrum would not be refined as it is today.

History

Ken Schwaber and Jeff Sutherland worked on Scrum until 1995, when they co-presented Scrum at the OOPSLA Conference in 1995. This presentation essentially documented the learning that Ken and Jeff gained over the previous few years, and made public the first formal definition of Scrum.

The history of Scrum is described elsewhere. To honor the first places where it was tried and refined, we recognize Individual, Inc., Newspage, Fidelity Investments, and IDX (now GE Medical).

The Scrum Guide documents Scrum as developed, evolved, and sustained for 20-plus years by Jeff Sutherland and Ken Schwaber. Other sources provide you with patterns, processes, and insights that complement the Scrum framework. These may increase productivity, value, creativity, and satisfaction with the results.

Authorized Bodies

Scrum Alliance
PMI
Project Management Institute
Scrum.org
DevOps
ICAgile
International Consortium for Agile
PEOPLECERT
PARTNER
AXELOS
CERTIFIED PARTNER
IIBA Endorsed Education Provider
The Open Group
TOGAF 9
Accredited Training Course
DISCIPLINEDAGILE
CONSORTIUM
SCALED AGILE
ISTQB
Certified Tester
Foundation Level

Our Popular Courses

AGILE MANAGEMENT

- Leading SAFe® 4.5
- SAFe® 4.5 Scrum Master
- SAFe® 4.5 Advanced Scrum Master
- SAFe® 4.5 for Teams
- SAFe® 4.5 Product Owner/Product Manager
- SAFe® 4.5 DevOps
- Certified Agile Leadership ®(CAL I)
- Certified ScrumMaster® (CSM)
- Certified Scrum Product Owner ® (CSPO)
- Certified Scrum Developer ® (CSD)
- Advanced Certified Scrum Master™ (A-CSM™)
- Advanced Certified Scrum Product Owner™ (A-CSPO™)
- Agile and Scrum
- Agile User Stories
- Agile Estimating and Planning
- PMI Agile Certified Practitioner (PMI-ACP) ®
- iSQI ® Certified Agile Essentials (CAE)
- Professional Scrum Developer™ (PSD)
- Professional Scrum Foundations™ (PSF)
- Professional Scrum Product Owner™ (PSPO)
- ICAgile–Foundation of DevOps (ICP-FDO)
- ICAgile – Agile Coaching (ICP-ACC)

Project Management

- Certified Associate in Project Management (CAPM®
- Project Management Professional (PMP) ®
- Portfolio Management Professional (PfMP) ®
- PMI Risk Management Professional (PMI-RMP) ®
- Program Management Professional (PgMP) ®
- Project Management Tools and Techniques
- PRINCE2 ® Foundation and Practitioner
- Microsoft ® Project 2007/2010/2013/2016
- Practical PMP with MS Project 2010/2013/2016
- Certified Green Project Manager Level GPM-b™
- Oracle Primavera P6V8

Business Management

- Certified Business Analysis Professional (CBAP ®)
- Certification of Capability in Business Analysis (CCBA®)
- Entry Certificate in Business Analysis (ECBA ™)
- PMI Professional in Business Analysis (PMI-PBA)®
- Business Analysis 101
- Business Case Writing

Cloud Computing

- AWS Developer Associate
- AWS Technical Essentials
- Microsoft Azure for AWS Experts

Big Data and Hadoop

- Big Data Analytics Course
- Introduction to Data Science
- Big Data and Hadoop
- Data Harvesting and R Essentials

Testing

- Certified Agile Tester
- ISTQB ® Foundation
- Selenium Essentials
- ICAgile - Agile Testing (ICP-TST)

Information Technology

- ITIL ® Foundation
- ITIL ® Intermediate (All Modules)
- ITIL ® Practitioner
- Blockchain 101
- Implement Blockchain using Hyperledger
- Technology of Blockchain - Ethereum Network
- Introduction to Cryptocurrency
- Introduction to Artificial Intelligence
- Mastering Python
- Ruby on Rails
- Python 101
- Angular JS
- Ansible Essentials
- Chef Advanced
- Android Development
- Angular JS 2.0
- Data Visualization with Tableau

IT Security

- CISM®
- CISSP®
- COBIT ® 5 Foundation
- CISA®
- PCI-Data Security Standard (PCI-DSS)

Get In Touch

Email Us

Support	: support@knowledgehut.com
Corporate	: corporate@knowledgehut.com
APAC	: apac@knowledgehut.com
MEA	: mea@knowledgehut.com
Europe	: europe@knowledgehut.com
Americas	: americas@knowledgehut.com
India	: india@knowledgehut.com

Call Us

USA	: 1-469-442-0620, 1-832-684-0080
UK	: +44 20 3519 3467
AUSTRALIA	: +612 9037 4228
SINGAPORE	: +65 6494 4930
INDIA	: +91 9036007392, +91 9036007399

www.knowledgehut.com

knowledgehut