



Certified ScrumMaster® Workshop

Siew Kok Ewe

Certified Scrum Trainer®

kok.ewe.siew@gmail.com



TABLE OF CONTENTS

Introduction	6
Course Agenda	7
A Couple of Big Problems in SOftware Development	8
Why Projects Fail?	10
Stacey Matrix	11
Waterfall Development Model.....	12
Agile Development.....	13
An Example	14
Manifesto for Agile Software Development	15
12 Agile Principles.....	16
The GOAL.....	19
Process Models	20
Stacey Matrix Revisited	21
Scrum Introduction.....	22
Scrum Values.....	23
Scrum Attributes	24
The Elements of Scrum	25
Scrum is a Framework.....	26
Scrum in a Picture	27
Product Backlog.....	28
Product Backlog Evolves	29
Creating Product Backlog Items	30
Product Backlog Items	31
Estimation.....	32
Planning Poker	33

Make Your Product Backlog DEEP	34
Product Backlog Refinement.....	35
Responsibilities around the Product Backlog.....	36
Tracking Progress with a Release Burnup Chart.....	37
Developers.....	39
Role of the Manager	40
Scrum Roles	41
Who is Responsible?.....	42
Definition of Done	44
Product Owner	45
Scenario Exercise: "David"	46
Scenario Exercise: "Shared DBA"	47
Scrum Master	48
Working Together	49
Commonly Asked Questions about Roles	50
Sprint Planning	51
Sprint Execution Quiz	52
Daily Scrum	53
Sprint Backlog	54
Tracking Progress	55
Sprint Burndown Exercise	56
Sprint Review	57
Sprint Retrospective	58
Improvement Actions.....	59
Sprint Events Summary	60
Scrum Quiz – Round 1	61

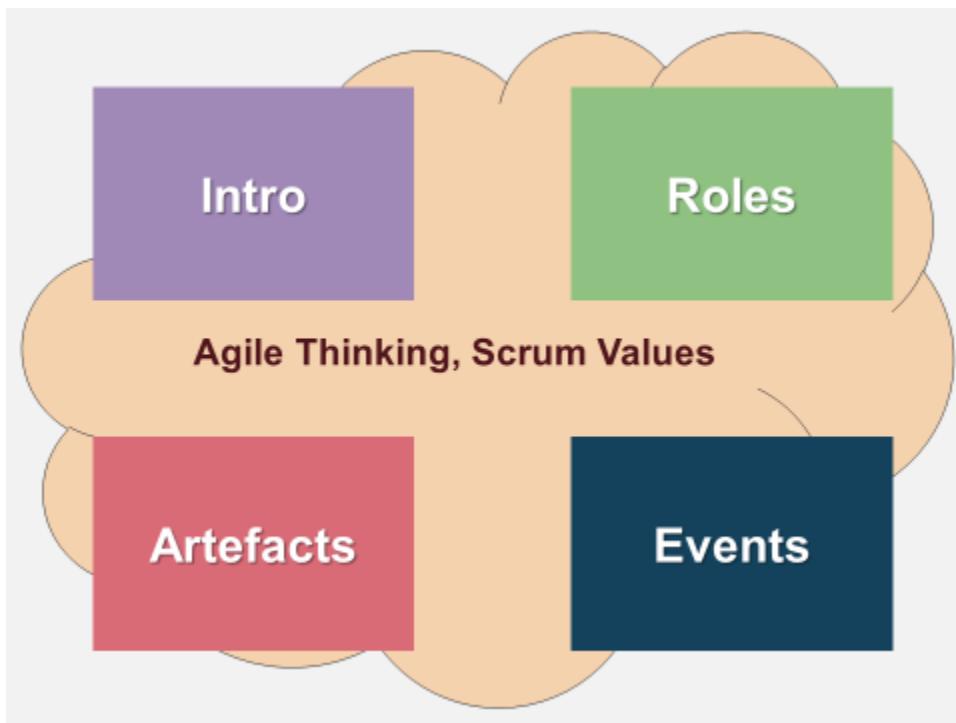
Scrum Quiz – Round 2.....	62
Appendix A – The Scrum Guide	63
Purpose of the Scrum Guide	63
Scrum Definition.....	64
Scrum Theory	64
Transparency	65
Inspection.....	65
Adaptation	65
Scrum Values.....	65
Scrum Team	66
Developers.....	67
Product Owner.....	67
Scrum Master.....	68
Scrum Events	69
The Sprint	69
Sprint Planning.....	70
Daily Scrum	71
Sprint Review	71
Sprint Retrospective	72
Scrum Artifacts	72
Product Backlog	73
Sprint Backlog.....	73
Increment	74
End Note	75
Acknowledgements	75
People.....	75

Scrum Guide History	75
Appendix B – “Scrum is Hard and Disruptive”.....	77
Appendix C – Sprint Execution Quiz Answers	79
Appendix D – References.....	80
Books	80
Articles	81
Videos	81
Self-Evaluation	83
Feedback Form	85

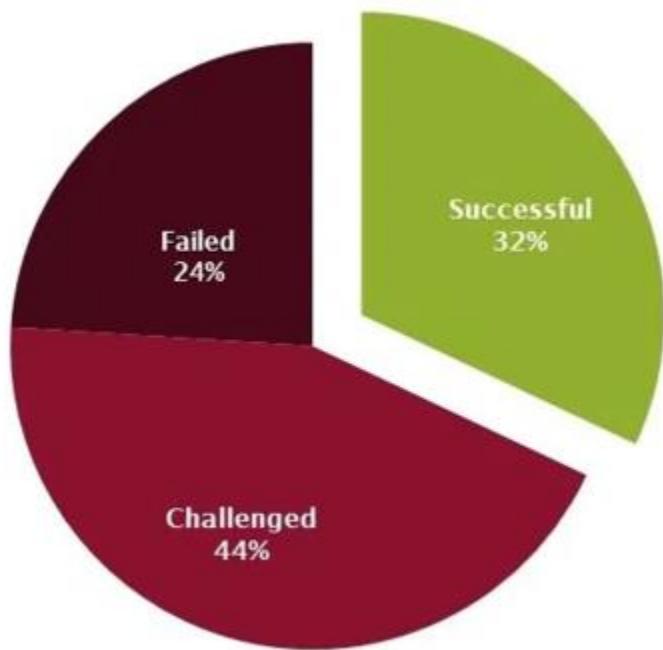
INTRODUCTION



COURSE AGENDA



A COUPLE OF BIG PROBLEMS IN SOFTWARE DEVELOPMENT



Source: CHAOS Report 2009, Standish Group

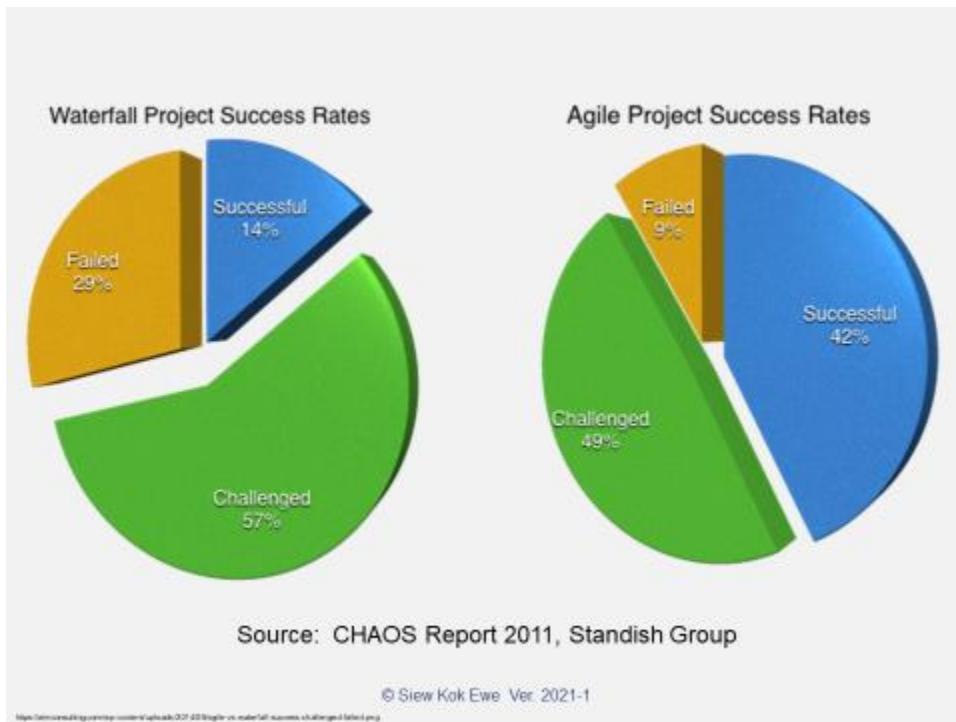
Software Projects Track Record

MODERN RESOLUTION FOR ALL PROJECTS

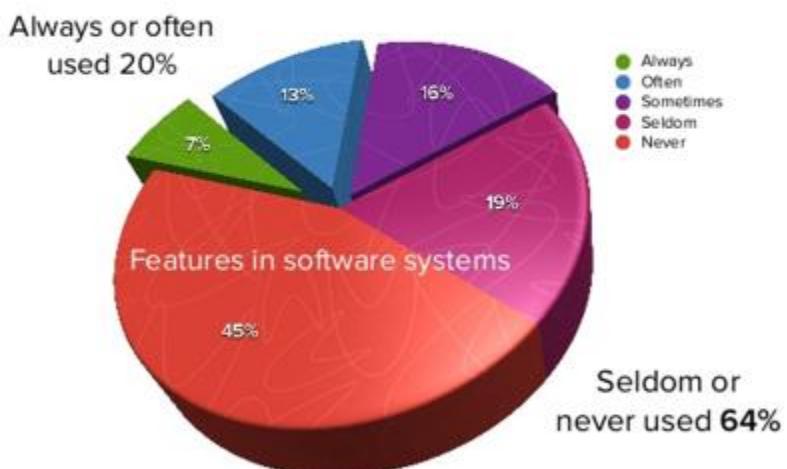
	2011	2012	2013	2014	2015
SUCCESSFUL	29%	27%	31%	28%	29%
CHALLENGED	49%	56%	50%	55%	52%
FAILED	22%	17%	19%	17%	19%

The Modern Resolution (OnTime, OnBudget, with a satisfactory result) of all software projects from FY2011–2015 within the new CHAOS database. Please note that for the rest of this report CHAOS Resolution will refer to the Modern Resolution definition not the Traditional Resolution definition.

© Siew Kok Ewe Ver. 2021-1



Feature usage



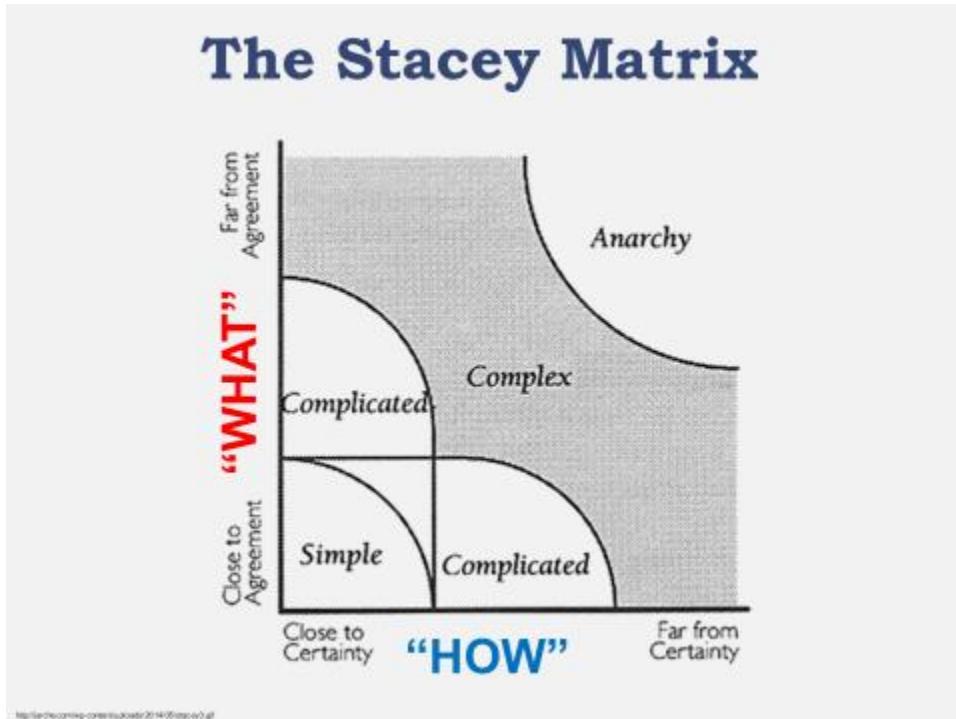
WHY PROJECTS FAIL?

Why Projects Fail?



From your experience, what contributes to product development project failures?

STACEY MATRIX



Typically, product development projects are COMPLEX work.

WATERFALL DEVELOPMENT MODEL

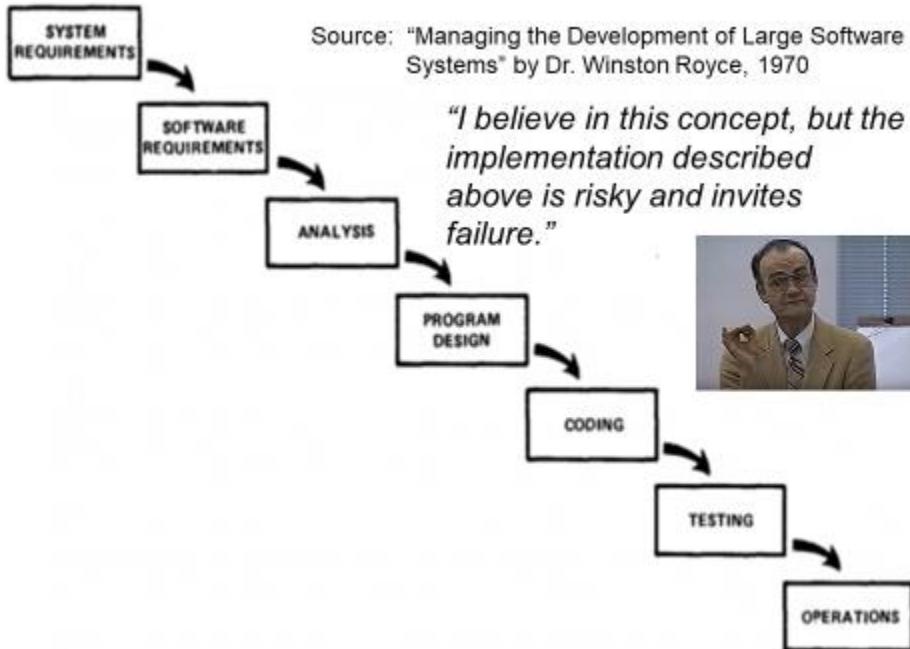
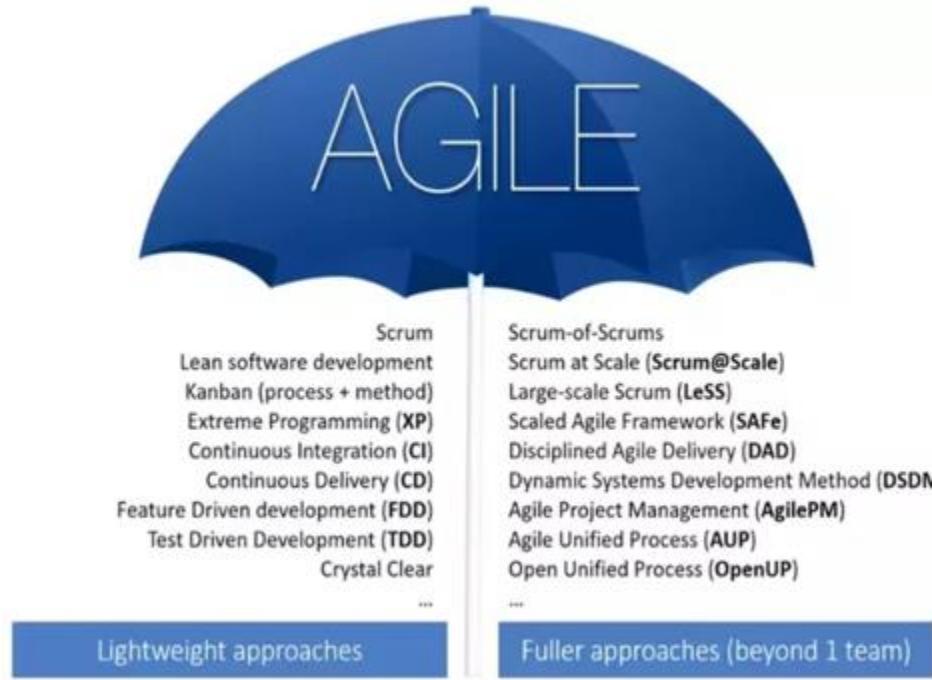


Figure 2. Implementation steps to develop a large computer program for delivery to a customer.

What are some disadvantages of the waterfall development model when applied to product development? Brainstorm with your table group and list them down.

AGILE DEVELOPMENT



Agile is a general term for various approaches to developing software that have emerged in response to the dismal results of the waterfall approach.

Currently the most popular Agile approach in the world is SCRUM.

All these approaches share similar thinking behind them, as described by the Agile Manifesto.

AN EXAMPLE



<https://www.youtube.com/watch?v=2NFH3VC6LNs>

This video illustrates an Agile approach to doing work.

In what ways is this different from the waterfall model?

MANIFESTO FOR AGILE SOFTWARE DEVELOPMENT

www.agilemanifesto.org



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 & 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.



12 AGILE PRINCIPLES

Agile Principle? (Y/N)	Statement
	Our highest priority is to deliver all the project scope on-time and on-budget.
Y	Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
	To avoid project delay and cost overrun, effective change control is vital to minimize scope creep.
Y	Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
Y	Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
	Deploy working software only when the entire product has been developed, since deployment is usually time consuming and requires a lot of effort.
	Once the project scope has been defined, minimize the involvement of the business people so that the developers can focus on developing the product.
Y	Business people and developers must work together daily throughout the project.
Y	Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
	Dynamically allocate the best resources to each project. The team's work must be closely monitored to ensure project success.

Y	The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
	The most efficient and effective communication is achieved when everyone uses the same document formats and tools.
	Keep all reports, dashboards and plans up-to-date to maintain a true picture of project progress.
Y	Working software is the primary measure of progress.
Y	Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
	Agile processes promote a fast pace of development. Goals and deadlines must be audacious enough to motivate the team to succeed.
Y	Continuous attention to technical excellence and good design enhances agility.
	Decisions regarding architecture, technical practices, tools, etc. should be made only after a thorough requirements analysis, to minimize the need to change these during project execution.
Y	Simplicity – the art of maximizing the amount of work not done – is essential.
	Product requirements, design, execution plans and development processes must be as comprehensive as possible in anticipation of all possible future needs and risks.
	The best architectures, requirements and design should be determined by the most experienced resources and given to the project team to implement.
Y	The best architectures, requirements, and designs emerge from self-organizing teams.

	A post-mortem at the end of the project is essential for the team to learn and improve their execution in the future.
Y	At regular intervals, the team reflects on how to become more effective, then tunes and adjust its behavior accordingly.

Creating Value is What Matters



© Siew Kok Ewe Ver. 2021-1

The GOAL of Agile Development is to:

TO CREATE THE HIGHEST BUSINESS
VALUE IN THE SHORTEST TIME,
IN A SUSTAINABLE WAY.

PROCESS MODELS

Waterfall



- Defined Process
- "Predict and Control"
- Success = deliver all scope within budget and schedule

http://checkincontrol.com/rapid-iteration-agile-processes-commits/2013/11/unintact_Civil_War_6666_1907.jpg

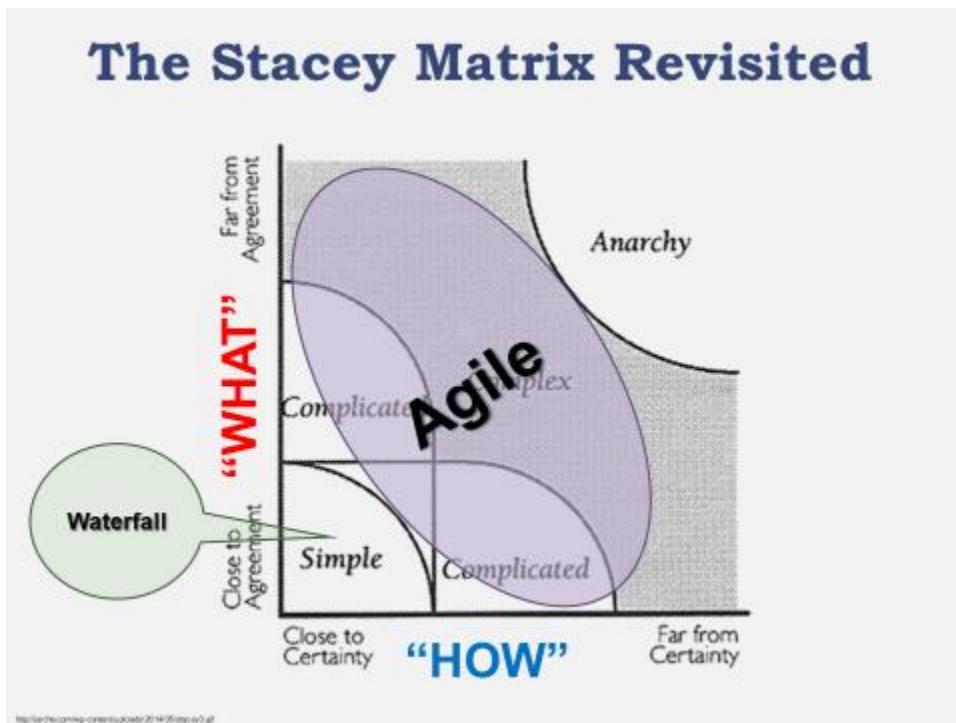
Agile (Incremental & Iterative)



- Empirical Process
- "Inspect and Adapt"
- Success = Objective met

<https://www.accessabc.com/no-communities/2019/07/19/what-is-the-difference/>

STACEY MATRIX REVISITED



What type(s) of project is the waterfall model applicable?

What type(s) of project is the Agile approach applicable?

Waterfall is not “wrong” per se. It’s just that we apply it to the wrong kind of problem.

SCRUM INTRODUCTION



Scrum is not an acronym. Ken Schwaber and Jeff Sutherland took the name from rugby. They were inspired by the 1986 Harvard Business Review article *The New New Product Development Game* by Takeuchi and Nonaka, who observed that product development is like playing rugby, rather than a relay race.

SCRUM VALUES

F S

FOCUS

O p n s

OPENNESS

R P T

RESPECT

C u r g

COURAGE

C m t n

COMMITMENT

SCRUM ATTRIBUTES

E m _ _ g _ n _ e EMERGENCE

I _ s p _ _ t & a _ _ p t INSPECT & ADAPT

S _ _ f - m _ _ _ g _ m _ n t SELF-MANAGEMENT

A _ k _ t h _ t _ _ m ASK THE TEAM

V _ l _ _ - d r _ _ _ n VALUE-DRIVEN

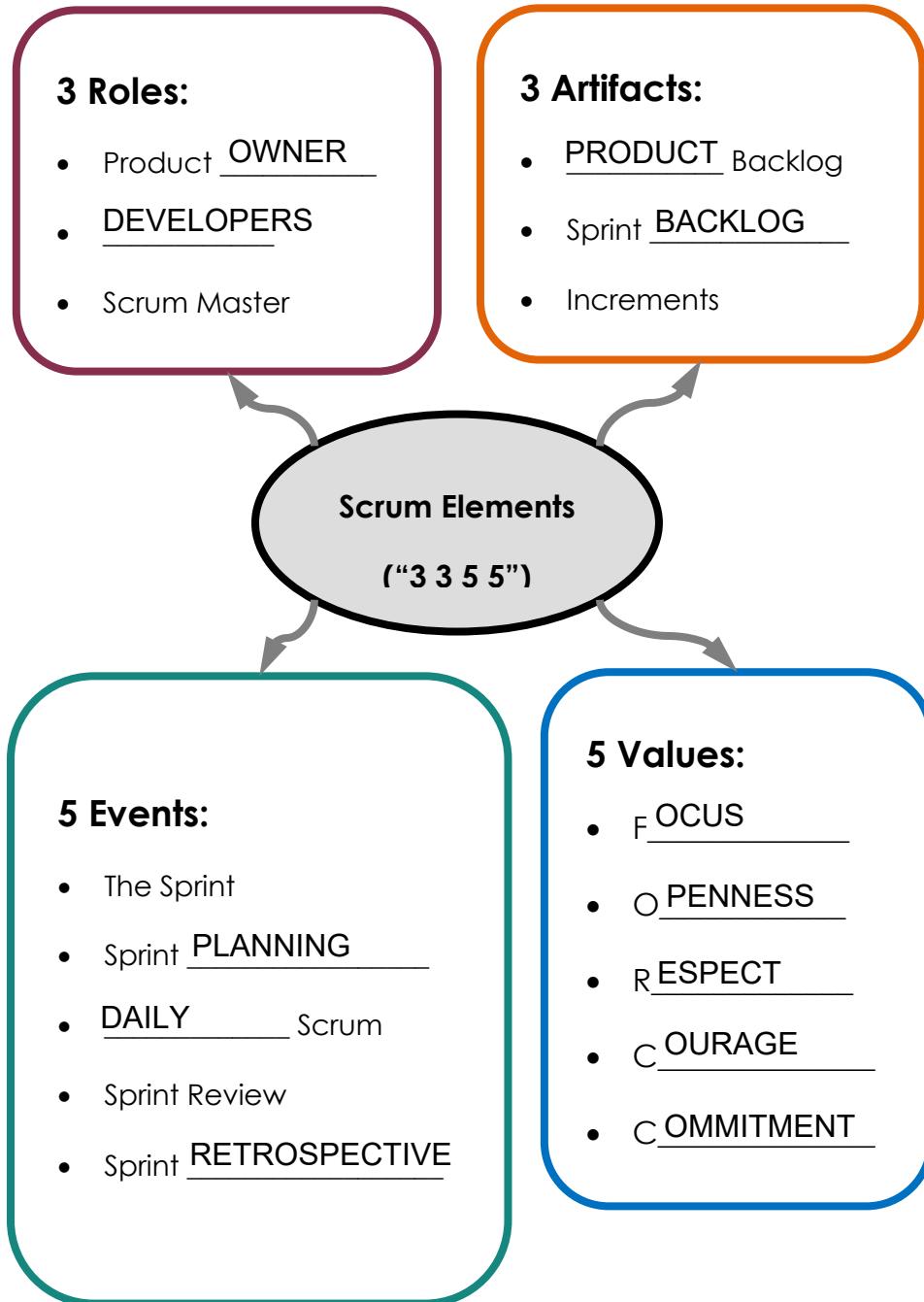
T _ _ _ - B _ X _ G TIME-BOXING

T r _ _ s p _ r _ _ y TRANSPARENCY

What are the "three pillars of Scrum"?

1. TRANSPARENCY
2. INSPECTION
3. ADAPTATION

THE ELEMENTS OF SCRUM



SCRUM IS A FRAMEWORK



Scrum – An Agile Framework

© Siew Kok Ewe Ver. 2021-1

Scrum is a **generic framework** for solving **complex** problems such as product development.

Scrum works on **3 key principles**: 1) transparency; 2) inspection; 3) adaptation. That is, to solve a complex problem, organizations must constantly inspect their reality and adapt to it; and that is only possible in a transparent environment. These 3 are often referred to as the “3 pillars of Scrum”.

Practitioners will find that Scrum is actually a framework for organizational improvement “disguised” as a project management framework, which most organizations use Scrum for.

Why is Scrum not a process (or methodology)? What's the difference between a process and a framework?

SCRUM IN A PICTURE

Draw a picture of the Scrum framework here:

PRODUCT BACKLOG



Product Backlog

© Siew Kok Ewe Ver. 2021-1

What is a Product Backlog?

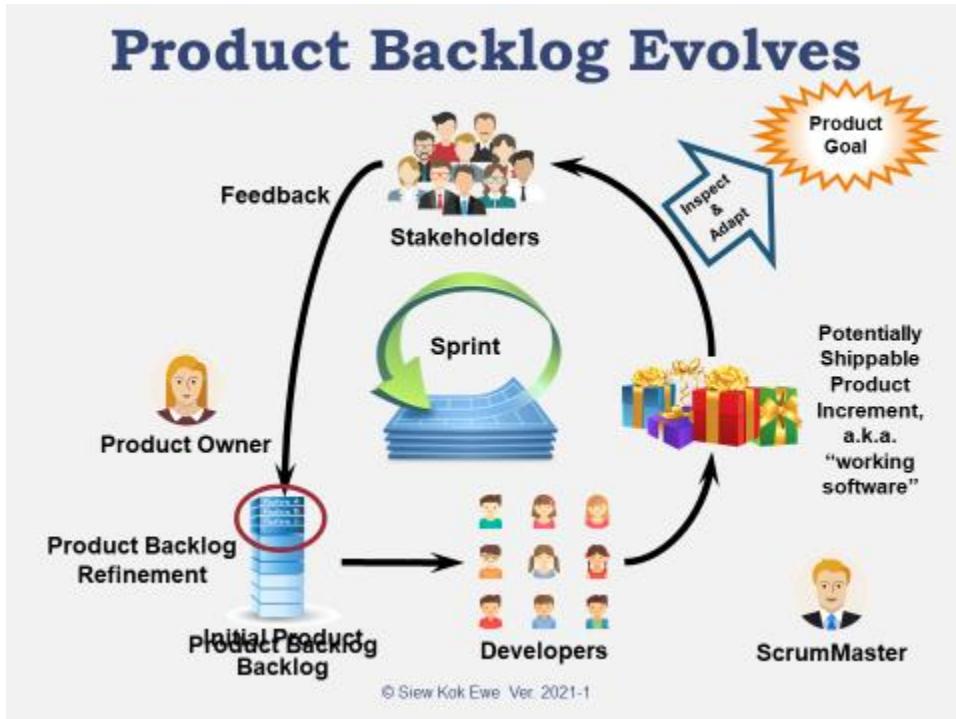
Generally, a Product Backlog is a list of valuable things to accomplish. A Product Backlog is a list of Product Backlog Items (PBI's). When a PBI is delivered, some value is created.

If we are developing a product, the Product Backlog is a

DYNAMIC list of USER - VALUED

features that MIGHT appear in a product.

PRODUCT BACKLOG EVOLVES



Product Goal = a future state of the product = target for Scrum Team.

The Product Backlog **emerges** to define “what” will fulfill the Product Goal.

The Product Backlog continuously evolves based on the Scrum Team's learning and stakeholder feedback on the product.

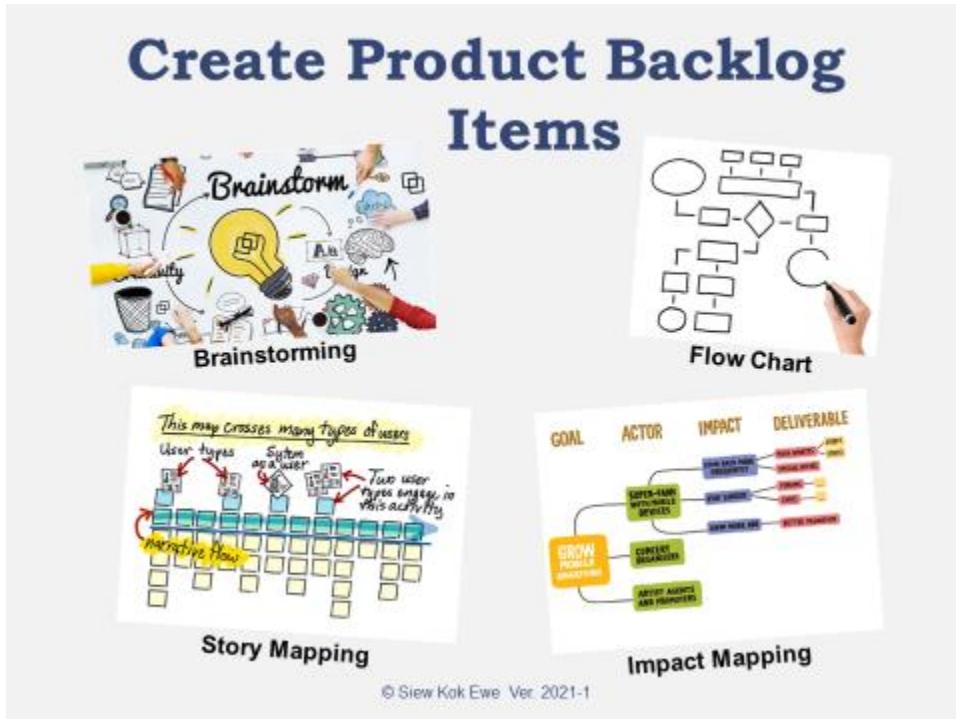
The Product Backlog is continuously kept up-to-date, through the Product Backlog Refinement activity.

Product Backlog Refinement is a collaboration among:

- PRODUCT OWNER
- DEVELOPERS
- STAKEHOLDERS

It may be facilitated by a SCRUM MASTER as needed.

CREATING PRODUCT BACKLOG ITEMS



There are many techniques for creating Product Backlog Items. All we are doing is creating a list of important things to achieve.

Besides features to develop, PBI's can include other work the Team needs to accomplish, e.g. bug fixes, improving the code, reducing technical debt, data migration, platform upgrades, etc. All the work the Team needs to do should be visible in the Product Backlog.

Product Backlog Items

As a customer,
I want to pay for my flights with
my credit card,
so that I don't have to send a
check.

Acceptance Criteria:

User Interface

Business Logic

Database

A big item is split into smaller items **vertically**.

© Siew Kok Ewe Ver. 2018-1

A popular format for Product Backlog Items is the *User Story* format. Therefore, PBI's are also commonly called *stories* or *user stories*. Note that the User Story format is not a part of the Scrum framework.

Good PBI's meet the *INVEST* criteria:

- **I NDEPENDENT**
- **N EGOTIABLE**
- **V ALUABLE**
- **E STIMABLE**
- **S MALL**
- **T ESTABLE**

ESTIMATION



What 2 pieces of information the Product Owner needs about each PBI, in order to prioritize them?

- BUSINESS VALUE
- SIZE

The only purpose of the estimates is to help the Product Owner make
DECISIONS.

In an Agile project, we prefer to use *Relative Estimation*. Why?
IT'S 1) EASIER AND 2) EASIER FOR US TO COME TO THE SAME ANSWERS.

PLANNING POKER

Planning Poker*

- Fibonacci – less argument
- Clarification on difference, not on similarity
- Everybody involved
 - Not dictated by most-knowledgeable person
- Fast
- Reliable
- Fun



* Not part of Scrum but complements it

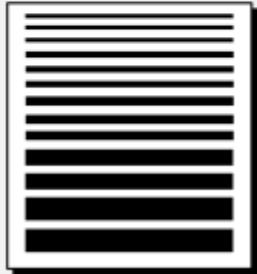
© Siew Kok Ewe

Note that not only Planning Poker is not part of Scrum, estimation is also optional in Scrum.

Product Backlog Properties

Detailed Appropriately
Estimated
Emergent
Prioritized (ordered)

High Priority



Fine-grained, detailed items ready
to be worked on in the next sprint

(Fine enough that the Team can
deliver at least 4 items in a sprint.)

Low Priority

Large, coarse-grained items

© Siew Kok Ewe Ver. 2021-1

Every item has the following elements:

- A description
- Order in the backlog
- Business value
- Size estimate

The items near the top of the Product Backlog are:

- HIGHER priority
- MORE detailed

PRODUCT BACKLOG REFINEMENT

Product Backlog Refinement

- Add or remove items
- Split items
- (Re-)order the items
- (Re-)estimate items
- Add details, e.g. acceptance criteria



5-10% of the sprint is spent on Product Backlog Refinement.

© Siew Kok Ewe Ver. 2021-1

Product Backlog Refinement happens every sprint. It can be done either in a meeting, or on an ad-hoc basis.

Two main goals of Product Backlog Refinement:

1. TO KEEP THE PRODUCT BACKLOG UP-TO-DATE.
2. TO MAKE SURE THE TEAM IS READY FOR THE WORK IN THE NEXT SPRINTS.

Two reasons why the Scrum Team spends no more than 10% of the sprint on Product Backlog Refinement:

1. THE DEVS NEED TIME TO WORK ON THEIR CURRENT SPRINT WORK.
2. COMING UP WITH TOO MUCH DETAIL COULD BE A WASTE OF TIME.

Negative impact if Product Backlog is NOT refined properly:

1. THE DEVS MAY NOT BE READY FOR THE NEXT SPRINT'S WORK.
2. THE DEVS MAY WORK ON THE WRONG PRIORITY.
3. WE LOSE TRANSPARENCY AND THE ABILITY TO INSPECT AND ADAPT.

Product Backlog Responsibilities

Responsibility	Who
Overall responsibility for the Product Backlog	PO
Provides estimates (item size)	DEVS
Provides details for each item	PO, SH
Introduces tools & techniques for Product Backlog management as needed	SM
Decides what gets included in the list	PO
Writes PBI's	PO, SH, DEVS
Refining the Product Backlog	PO, SH, DEVS
Facilitate collaboration as needed	SM
Decides on the priority of items	PO
Split PBI's	PO, SH, DEVS
Makes sure the Product Backlog is visible to everyone	SM, PO

© Siew Kok Ewe Ver. 2018-1

TRACKING PROGRESS WITH A RELEASE BURNUP CHART

On the chart in the next page, draw a Release Burnup for the following situation:

At the start of a project, the total estimated Product Backlog is 100 story points.

At the end of sprint 1, the team has completed 10 story points, but the Product Backlog has also grown by 10 story points due to new features being added.

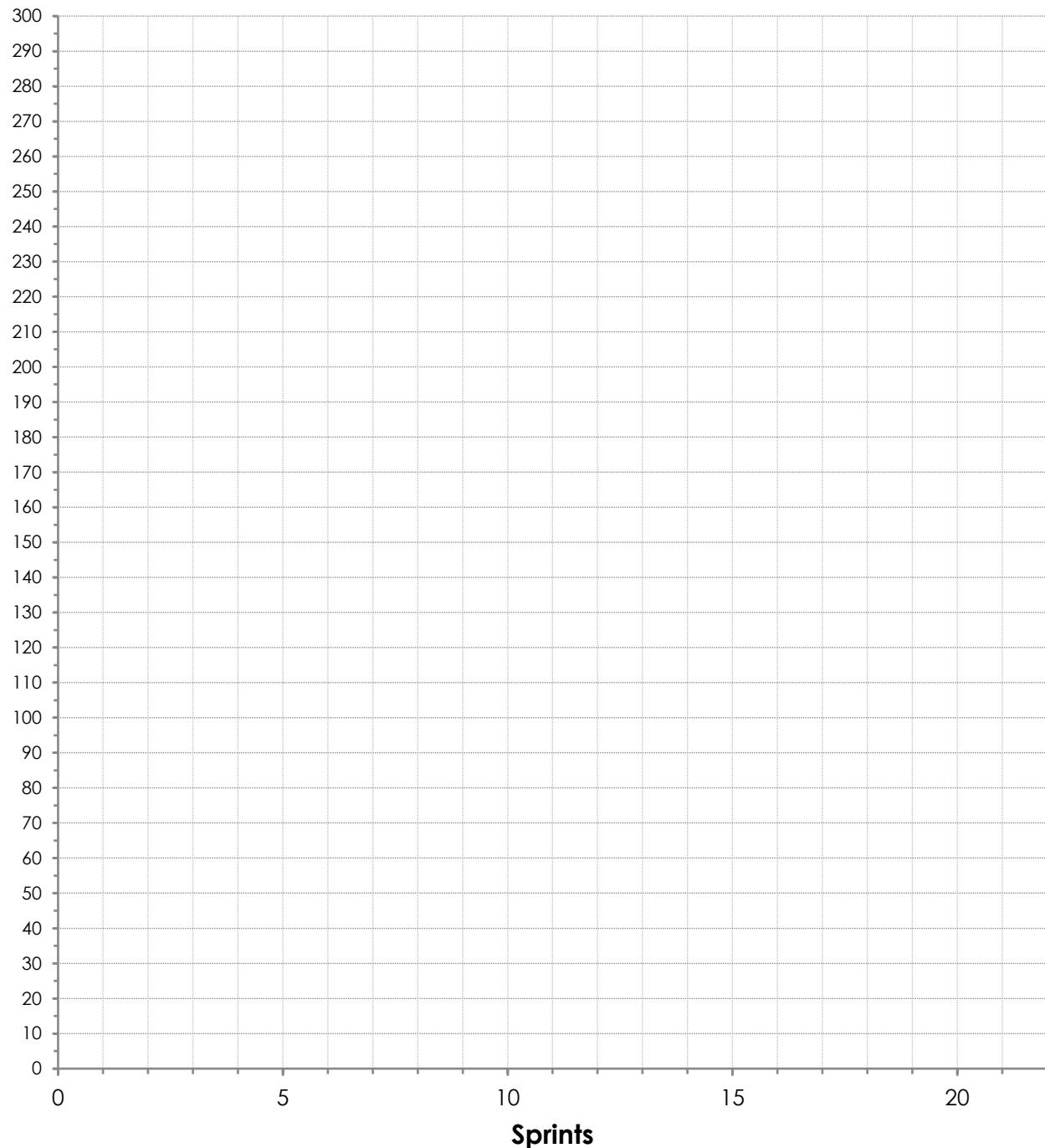
At the end of sprint 2, the team completes a further 20 story points, and no new items are added to the Product Backlog.

At the end of sprint 3, the team completes a further 15 story points, the amount of work now remaining to be done on the Product Backlog is 85, i.e. 20 story points have been added this sprint.

Use the Release Burnup to answer the following questions:

1. *What is the team's average velocity?* **15 POINTS PER SPRINT**
2. *At what rate is the Product Backlog growing (on average)?* **10 POINTS PER SPRINT**
3. *How many story points will the team complete in 8 sprints?* **120 POINTS**
4. *How many sprints will it take until all the work is complete, assuming the current trends continue?* **ABOUT 18 SPRINTS**

Release Burnup



Developers



Characteristics

- Self-managing
- Cross-functional
- No titles
- No sub-teams
- 3 – 9 people
- Stable, full-time, dedicated
- T-shaped; generalizing specialists

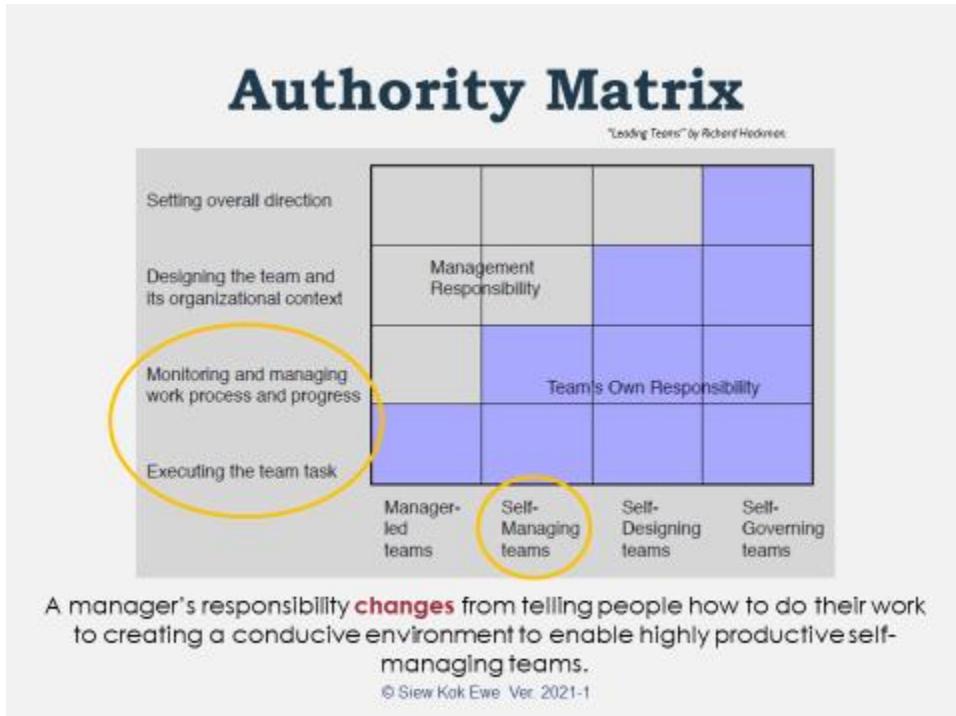
Responsibilities

- Decide how to do the work (process)
- Track progress in sprint
- Produce “done” increment every sprint
- Improve engineering practices
- Estimate work
- Collaborate with PO and SH to maintain Product Backlog

© Siew Kok Ewe Ver. 2021-1

The Developers only work on what the Product Owner wants (in the Product Backlog), even though they may get requests from other people.
Why is this important?

ROLE OF THE MANAGER



There is no role called "manager" in Scrum. Most organizations still have managers after adopting Scrum. *How will the role of the manager change in a Scrum environment?*

From: CONTROLLING/MONITORING HOW WORK GETS DONE

To: CREATING A GREAT ORGANISATION WITHIN WHICH GREAT TEAMS SELF-MANAGE TO DELIVER GREAT RESULTS.

SCRUM ROLES

Scrum Roles



<http://www.vectorforall.net/preview/office-man-vector-1988>

<http://www.dher.com/clipart/v/v/kn5/people-in-the-office-md.png>

<http://softcavapla.yk.com/wp-content/uploads/2012/08/business-man-holding>

© Siew Kok Ewe Ver. 2021-1

The three roles in Scrum are:

1. PRODUCT OWNER
2. DEVELOPERS
3. SCRUM MASTER

WHO IS RESPONSIBLE?

Put an X under the role(s) who is/are responsible for each task.

Task	Scrum Master	Product Owner	Developers
Estimates the size of items on the Product Backlog			X
Participates in Sprint Planning		X	X
Updates the Release Burnup chart		X	
Ensures everybody follows Scrum practices	X		
Updates the Sprint Backlog when tasks are done			X
Ensures impediments are removed	X		
Manages the budget and return on investment of the product		X	
Assigns tasks to developers			X
Accepts the delivery of the sprint		X	
Must attend the Daily Scrum			X
Communicates status of the release to stakeholders		X	
Updates the Sprint Burndown chart			X
Educes the organisation about Scrum	X		

Decides how much will be delivered in a sprint			X
Ensures each story meets the Definition of Done			X

Additional notes on what you have learnt about the roles:

Two Different Conditions of Satisfaction

Acceptance Criteria

- “What”
- What the PBI needs to meet for it to be accepted
- Different for each PBI

Definition of “Done”

- “How”
- What work the Team can do in a sprint for a PBI to be called “done”
- Same for all PBI's
- “Shippable” is a special kind of “Done”

© Siew Kok Ewe Ver. 2021-1

Product Owner

Characteristics

- One person
- Authority over product decisions
- Accountable for product success
- “Single wring-able neck”



Responsibilities

- On-going visioning
- Clearly express PBI's
- Order PBI's
- Optimize value of Developers' work
- Manage Product Backlog; maintain transparency
- Ensure Developers have necessary details to work on

© Siew Kok Ewe Ver. 2021-1

Why is the Product Owner a single person and not a committee?

SCENARIO EXERCISE: “DAVID”

What are the lessons from this scenario exercise?

SCENARIO EXERCISE: “SHARED DBA”

What are the lessons from this scenario exercise?

Scrum Master



Characteristics

- "Leader who serves"
- No authority
- Leads by influence
- Role models Scrum values
- Agent of change
- Not a manager
- Not an admin
- System thinker

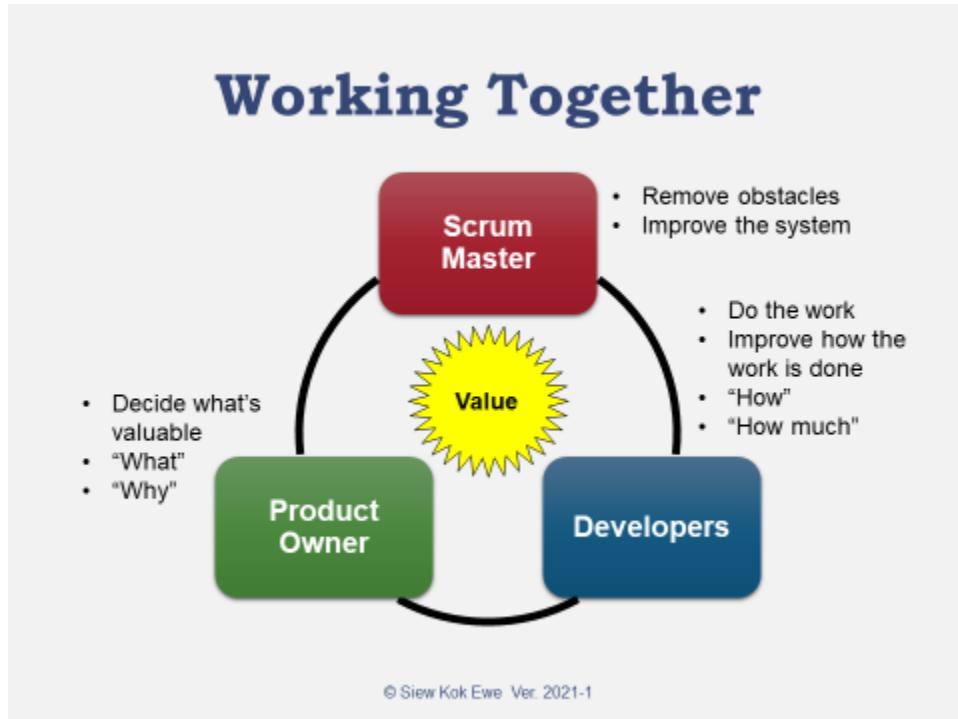
Responsibilities

- Help people understand Scrum
- Increase visibility
- Foster collaboration
- Protect the team
- Remove impediments
- Leads change
- Facilitate as needed
- Improve the system (org)

© Siew Kok Ewe Ver. 2021-1

Why does the Scrum Master have no authority and needs to lead through influence instead?

- PEOPLE WILL DO WHAT YOU SAY EVEN IF THEY DON'T WANT TO.
- THE TEAM WILL FAIL TO SELF-MANAGE.
- THERE WILL BE LESS OPENNESS.



All three roles work together towards creating the highest value in the shortest time.

COMMONLY ASKED QUESTIONS ABOUT ROLES

1. *There is no PM in Scrum. How do we run a project without a PM?*

PM RESPONSIBILITIES ARE NOW SPREAD ACROSS THE THREE ROLES.

2. *Can one person play two roles in Scrum?*

NOT A GOOD PRACTICE. EACH ROLE IS FULL-TIME AND HAS DIFFERENT INTEREST AND FOCUS.

3. *Can the line manager be the Product Owner or Scrum Master?*

NOT A GOOD PRACTICE DUE TO A LACK OF OPENNESS BETWEEN THE DEVELOPERS AND THE PRODUCT OWNER OR SCRUM MASTER.

4. *Can you have team leads in Scrum?*

ASK THE TEAM.

5. *Where do architects fit in Scrum?*

CAN BE A DEVELOPER.

6. *How many Scrum Teams can a Scrum Master work with?*

IDEALLY ONE. MAYBE TWO. NOT TOO MANY - HARD TO FOCUS.

Sprint Planning Part 1

- A.k.a. "selection meeting"
- "WHY" & "WHAT"
 - Why is this Sprint valuable; i.e. the Sprint Goal
 - What can be Done?
- Developers select the PBI's to deliver
- Understand acceptance criteria
- Developers decide how much work to commit to
- When
 - 1st thing in the sprint
- Who
 - Product Owner
 - Developers
 - SM (optional)
 - Others can be invited
- How long
 - 1 hour per week of sprint

© Siew Kok Ewe Ver. 2021-1

Sprint Planning Part 2

- A.k.a. "design meeting"
- "HOW"
 - How will the chosen work get done?
- Developers plan for work to turn PBI's into valuable increments
- → Sprint Backlog
- The Scrum Team commits to the Sprint Goal
- When
 - After SP1
- Who
 - Developers
 - PO (optional but available if needed)
 - SM (optional)
- How long
 - 1 hour per week of sprint

© Siew Kok Ewe Ver. 2021-1

SPRINT EXECUTION QUIZ

Decide if the following statements are True or False:

1. The duration of a sprint can change from sprint to sprint as long as it is under 4 weeks. _____
2. A sprint can be terminated if the sprint goal is no longer valuable.

3. The authority to terminate a sprint lies with the ScrumMaster. _____
4. The mechanisms for monitoring progress during the sprint in Scrum are the Daily Scrum, Sprint Backlog and Sprint Burndown. _____
5. The task board and Sprint Burndown are maintained by the ScrumMaster. _____
6. The task board or Sprint Backlog only contains items for the current sprint. _____
7. Each team member in a cross-functional team needs to be able to do all tasks in a story. _____
8. Adding team members will always increase velocity. _____
9. ScrumMasters should keep an impediment log of all issues affecting the team and aim to resolve most impediments within 24 hours.

10. Tasks must be estimated in hours. _____
11. Teams need slack to make improvements to their process. _____
12. Co-location means team members need to be within 6m of every other person in their team. _____
13. Under pressure to work faster or harder developers unconsciously compromise quality. _____

(Answers are in Appendix C, page 79.)

Daily Scrum

- “Are we on track?”
- For the Developers to
 - Inspect & adapt their sprint plan
 - Renew commitment to one another
- 15-minute time-box
- No problem-solving
- Developers' meeting
 - Others welcome but it's not for them
- 3 questions (1 way to do it)
 - What did I do yesterday?
 - What am I going to do today?
 - What help do I need?



© Siew Kok Ewe Ver. 2021-1

<http://www.ag.com.my/visualmanagement/wp-content/uploads/itsmday2.jpg>

Why daily?

SPRINT BACKLOG

Sprint Backlog

- Sprint Goal (why) + selected PBI's (what) + plan work (how)
- A plan by and for the Developers
- Sprint Goal (why)
 - Single objective for the Sprint; keeps the Scrum Team focused
 - Should stay fixed as much as possible
- Selected PBI's (what)
 - May change; Developers & PO can negotiate
- Planned work (how)
 - Exact work will likely change; Developers inspect and adapt

© Siew Kok Ewe Ver. 2021-1

Tasks (Optional)

User Story	To Do	In Progress	Blocked	Done
Story #1	Task 2 Task 4 Task 6	Task 5 Task 3		
Story #2	Task 4 Task 5	Task 2	Task 3	Task 1
Story #3	Task 1 Task 2 Task 4 Task 5			

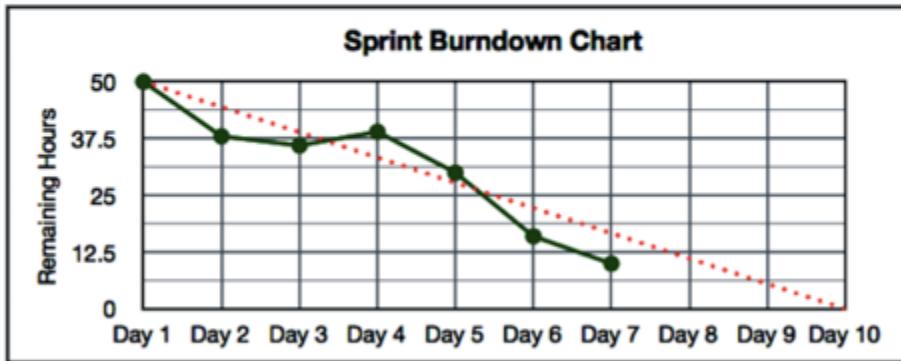
- Tasks - very small pieces of work (< half day)
- Owned by the Developers
- Can be estimated in hours (not a must)
- Don't pre-assign tasks during Sprint Planning

© Siew Kok Ewe Ver. 2021-1

TRACKING PROGRESS

Tracking Progress

- Responsibility of the Developers



© Siew Kok Ewe Ver. 2021-1

The Sprint Burndown chart tracks **remaining** task hours (or remaining number of tasks), rather than actual hours spent (or number of tasks completed). Why?

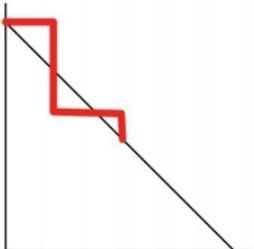
TO KNOW WHETHER THEY ARE ABLE TO REACH THE SPRINT GOAL, THE DEVELOPER NEED TO KNOW HOW MUCH WORK STILL REMAINS.

Why is it normal to see the line going up (instead of going down) especially near the beginning of the sprint?

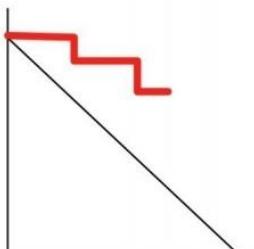
NEW TASKS MAY EMERGE AS THE DEVELOPERS WORK ON THE ITEMS.
HENCE THE LINE GOES UP.

SPRINT BURNDOWN EXERCISE

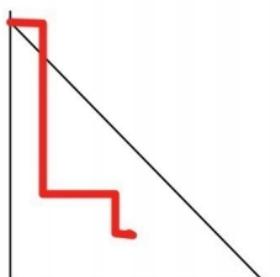
What might be happening in each of these cases below, and what would the Developers do?



GOING WELL. KEEP GOING.

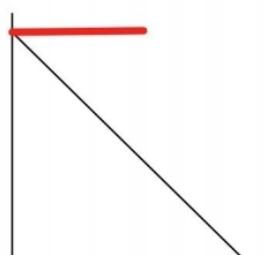


NOT GOING WELL. FIND OUT THE ROOT CAUSE, ADDRESS THE PROBLEM, AND GET BACK ON TRACK.



GOING EXCEEDINGLY WELL. FINISH THE REMAINING WORK THEN DECIDE WHETHER TO TAKE IN THE NEXT ITEM IF THERE IS TIME.

DISCUSS IN RETROSPECTIVE AND LEARN FROM THIS.



NO TRANSPARENCY - DON'T KNOW WHAT IS GOING ON.

FIND OUT WHAT IS REALLY HAPPENING AND UPDATE THE CHART. CATCH UP IF LATE.

Sprint Review

- To inspect and adapt the PRODUCT
- Take < 1 hour to prepare
- Only show working software
 - Or DONE stories
- No PowerPoint
- Get feedback on the current state of the product
- When
 - 2nd last event in the sprint
- Who
 - Product Owner
 - Developers
 - Stakeholders (anybody with feedback)
 - Scrum Master (optional)
- How long
 - 1 hour per week of sprint

© Siew Kok Ewe Ver. 2021-1

Make sure feedback on the product is generated in the Sprint Review. Otherwise, we have not achieved the purpose of the Sprint Review.

The main audience of the Sprint Review are the stakeholders, not the Product Owner. Why?

THE PRODUCT OWNER WORKS CLOSELY WITH THE DEVELOPERS DURING THE SPRINT. SHE ALREADY KNOWS WHAT THE DEVELOPERS HAVE BEEN ABLE TO ACCOMPLISH.

What happens to work that is not completed in the Sprint?

THE DEVELOPERS SHOULD STRIVE TO GET ALL ITEMS DONE IN THE SPRINT.

IF AN ITEM CANNOT BE COMPLETED, WORK WITH THE PRODUCT OWNER TO SPLIT OFF A PORTION OF THE ITEM AND GET THAT DONE. THE OTHER PORTION CAN GO BACK TO THE PRODUCT BACKLOG.

Sprint Retrospective

- To inspect and adapt the PROCESS
- “How are we working as a Team?”
- Phases
 - Gather data
 - Generate insights
 - Decide what to do
- 1-2 improvement actions
- Scrum values apply
- When
 - Last event in the sprint
- Who
 - Developers
 - Scrum Master (usually, but optional)
 - Any other people the Developers invite
- How long
 - 45 minutes per week of sprint

© Siew Kok Ewe Ver. 2021-1

Make sure the Developers generate improvement actions for the very next Sprint (not some 3 months from now) in the Retrospective. Otherwise, we have not achieved the purpose of the Retrospective.

Improvement Actions

- What is the improvement action?
- Why are we taking this action?
- Who will do what and when?
- How will we know we have
 - Succeeded
 - Failed



© Siew Kok Ewe Ver. 2020-1

SPRINT EVENTS SUMMARY

Event/Activity	Focus	Purpose	Outcome	Participants	When	Timebox
Sprint Planning Part 1 <i>a.k.a. "Selection Meeting"</i>	"What" to deliver; and "Why"	To establish Sprint Goal & select PBI's to deliver	Sprint Goal & a list of PBI's to deliver	Product Owner & Developers	First event in the Sprint	1 hour per week of sprint
Sprint Planning Part 2 <i>a.k.a. "Design Meeting"</i>	"How" to meet the Sprint Goal	To determine solutions & commit to a Sprint Backlog	Sprint Backlog (Sprint Goal + selected PBI's + plan)	Developers PO is "optional but available"	Right after Sprint Planning Part 1	1 hour per week of sprint
Daily Scrum	Progress towards the Sprint Goal	To inspect & adapt towards Sprint Goal	Understand how to collaborate	Developers Others are welcome but it's not for them	Daily	15 minutes
Product Backlog Refinement	Product Backlog	To update the PB; get ready for next 1-3 sprints	Updated PB; Dev's are ready for next 1-3 Sprints' work	Product Owner, Dev's, Stakeholders	Anywhere in the Sprint	5-10% of the sprint
Sprint Review	Product	To inspect & adapt the product ("what")	Feedback on the product & revised Product Backlog	Product Owner, Dev's, and Stakeholders especially	2 nd last event in the Sprint	1 hour per week of sprint
Sprint Retrospective	Process	To inspect & adapt the process ("how")	Process improvement actions for the next Sprint	Developers, ScrumMaster (usually), anybody the Dev's invite	Last event in the Sprint	45 minutes per week of sprint

SCRUM QUIZ – ROUND 1

Answer the following 5 questions as a table group. Don't refer to your workbook, answer them from your memory. When you have written down your 5 answers bring your answers to the instructor for scoring.

1. How often should you hold a retrospective?

EVERY SPRINT, AT THE END.

2. Is Scrum a defined or empirical process?

EMPIRICAL

3. Who is responsible for updating the Sprint Backlog and Sprint Burndown?

DEVELOPERS

4. Name the 5 Scrum values.

FOCUS, OPENNESS, RESPECT, COURAGE, COMMITMENT.

5. Answer TRUE or FALSE

If there is a Product Backlog Item (story) that does not meet the Definition of Done at the end of the sprint, the sprint is extended until that item is complete.

FALSE

SCRUM QUIZ – ROUND 2

Answer the following 5 questions as a table group. Don't refer to your workbook, answer them from your memory. When you have written down your 5 answers bring your answers to the instructor for scoring.

6. Which area of the Stacey Matrix is Agile best suited to?

COMPLEX

7. What is the focus of the sprint review?

THE PRODUCT

8. Complete the following statement from the agile manifesto:

Customer Collaboration over

CONTRACT NEGOTIATION

9. What is the primary measure of progress in agile?

THE PRODUCT

10. Where would you find the following two statements about a story:

a. "The page shall take no more than 1 second to refresh"

i. In the definition of done?

ii. In the story's acceptance criteria?

b. "User acceptance testing must be done"

i. In the definition of done?

ii. In the story's acceptance criteria?

APPENDIX A – THE SCRUM GUIDE

Below is the November 2020 version of the Scrum Guide, the official definition of Scrum, maintained by its co-creators, Jeff Sutherland and Ken Schwaber. The Guide is updated by Jeff and Ken from time to time. For the latest version, go to <http://www.scrumguides.org/>.

PURPOSE OF THE SCRUM GUIDE

We developed Scrum in the early 1990s. We wrote the first version of the Scrum Guide in 2010 to help people worldwide understand Scrum. We have evolved the Guide since then through small, functional updates. Together, we stand behind it.

The Scrum Guide contains the definition of Scrum. Each element of the framework serves a specific purpose that is essential to the overall value and results realized with Scrum. Changing the core design or ideas of Scrum, leaving out elements, or not following the rules of Scrum, covers up problems and limits the benefits of Scrum, potentially even rendering it useless.

We follow the growing use of Scrum within an ever-growing complex world. We are humbled to see Scrum being adopted in many domains holding essentially complex work, beyond software product development where Scrum has its roots. As Scrum's use spreads, developers, researchers, analysts, scientists, and other specialists do the work. We use the word "developers" in Scrum not to exclude, but to simplify. If you get value from Scrum, consider yourself included.

As Scrum is being used, patterns, processes, and insights that fit the Scrum framework as described in this document, may be found, applied and devised. Their description is beyond the purpose of the Scrum Guide because they are context sensitive and differ widely between Scrum uses. Such tactics for using within the Scrum framework vary widely and are described elsewhere.

Ken Schwaber & Jeff Sutherland November 2020

SCRUM DEFINITION

Scrum is a lightweight framework that helps people, teams and organizations generate value through adaptive solutions for complex problems.

In a nutshell, Scrum requires a Scrum Master to foster an environment where:

1. A Product Owner orders the work for a complex problem into a Product Backlog.
2. The Scrum Team turns a selection of the work into an Increment of value during a Sprint.
3. The Scrum Team and its stakeholders inspect the results and adjust for the next Sprint.
4. Repeat

Scrum is simple. Try it as is and determine if its philosophy, theory, and structure help to achieve goals and create value. The Scrum framework is purposefully incomplete, only defining the parts required to implement Scrum theory. Scrum is built upon by the collective intelligence of the people using it. Rather than provide people with detailed instructions, the rules of Scrum guide their relationships and interactions.

Various processes, techniques and methods can be employed within the framework. Scrum wraps around existing practices or renders them unnecessary. Scrum makes visible the relative efficacy of current management, environment, and work techniques, so that improvements can be made.

SCRUM THEORY

Scrum is founded on empiricism and lean thinking. Empiricism asserts that knowledge comes from experience and making decisions based on what is observed. Lean thinking reduces waste and focuses on the essentials.

Scrum employs an iterative, incremental approach to optimize predictability and to control risk. Scrum engages groups of people who collectively have all the skills and expertise to do the work and share or acquire such skills as needed.

Scrum combines four formal events for inspection and adaptation within a containing event, the Sprint. These events work because they implement the empirical Scrum pillars of transparency, inspection, and adaptation.

TRANSPARENCY

The emergent process and work must be visible to those performing the work as well as those receiving the work. With Scrum, important decisions are based on the perceived state of its three formal artifacts. Artifacts that have low transparency can lead to decisions that diminish value and increase risk.

Transparency enables inspection. Inspection without transparency is misleading and wasteful.

INSPECTION

The Scrum artifacts and the progress toward agreed goals must be inspected frequently and diligently to detect potentially undesirable variances or problems. To help with inspection, Scrum provides cadence in the form of its five events.

Inspection enables adaptation. Inspection without adaptation is considered pointless. Scrum events are designed to provoke change.

ADAPTATION

If any aspects of a process deviate outside acceptable limits or if the resulting product is unacceptable, the process being applied or the materials being produced must be adjusted. The adjustment must be made as soon as possible to minimize further deviation.

Adaptation becomes more difficult when the people involved are not empowered or self-managing. A Scrum Team is expected to adapt the moment it learns anything new through inspection.

SCRUM VALUES

Successful use of Scrum depends on people becoming more proficient in living five values:

Commitment, Focus, Openness, Respect, and Courage

The Scrum Team commits to achieving its goals and to supporting each other. Their primary focus is on the work of the Sprint to make the best

possible progress toward these goals. The Scrum Team and its stakeholders are open about the work and the challenges. Scrum Team members respect each other to be capable, independent people, and are respected as such by the people with whom they work. The Scrum Team members have the courage to do the right thing, to work on tough problems.

These values give direction to the Scrum Team with regard to their work, actions, and behavior. The decisions that are made, the steps taken, and the way Scrum is used should reinforce these values, not diminish or undermine them. The Scrum Team members learn and explore the values as they work with the Scrum events and artifacts. When these values are embodied by the Scrum Team and the people they work with, the empirical Scrum pillars of transparency, inspection, and adaptation come to life building trust.

SCRUM TEAM

The fundamental unit of Scrum is a small team of people, a Scrum Team. The Scrum Team consists of one Scrum Master, one Product Owner, and Developers. Within a Scrum Team, there are no sub-teams or hierarchies. It is a cohesive unit of professionals focused on one objective at a time, the Product Goal.

Scrum Teams are cross-functional, meaning the members have all the skills necessary to create value each Sprint. They are also self-managing, meaning they internally decide who does what, when, and how.

The Scrum Team is small enough to remain nimble and large enough to complete significant work within a Sprint, typically 10 or fewer people. In general, we have found that smaller teams communicate better and are more productive. If Scrum Teams become too large, they should consider reorganizing into multiple cohesive Scrum Teams, each focused on the same product. Therefore, they should share the same Product Goal, Product Backlog, and Product Owner.

The Scrum Team is responsible for all product-related activities from stakeholder collaboration, verification, maintenance, operation, experimentation, research and development, and anything else that might be required. They are structured and empowered by the organization to manage their own work. Working in Sprints at a sustainable pace improves the Scrum Team's focus and consistency.

The entire Scrum Team is accountable for creating a valuable, useful Increment every Sprint. Scrum defines three specific accountabilities within the Scrum Team: the Developers, the Product Owner, and the Scrum Master.

DEVELOPERS

Developers are the people in the Scrum Team that are committed to creating any aspect of a usable Increment each Sprint.

The specific skills needed by the Developers are often broad and will vary with the domain of work. However, the Developers are always accountable for:

- Creating a plan for the Sprint, the Sprint Backlog;
- Instilling quality by adhering to a Definition of Done;
- Adapting their plan each day toward the Sprint Goal; and,
- Holding each other accountable as professionals.

PRODUCT OWNER

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

The Product Owner is also accountable for effective Product Backlog management, which includes:

- Developing and explicitly communicating the Product Goal;
- Creating and clearly communicating Product Backlog items;
- Ordering Product Backlog items; and,
- Ensuring that the Product Backlog is transparent, visible and understood.

The Product Owner may do the above work or may delegate the responsibility to others. Regardless, the Product Owner remains accountable.

For Product Owners to succeed, the entire organization must respect their decisions. These decisions are visible in the content and ordering of the Product Backlog, and through the inspectable Increment at the Sprint Review.

The Product Owner is one person, not a committee. The Product Owner may represent the needs of many stakeholders in the Product Backlog. Those wanting to change the Product Backlog can do so by trying to convince the Product Owner.

SCRUM MASTER

The Scrum Master is accountable for establishing Scrum as defined in the Scrum Guide. They do this by helping everyone understand Scrum theory and practice, both within the Scrum Team and the organization.

The Scrum Master is accountable for the Scrum Team's effectiveness. They do this by enabling the Scrum Team to improve its practices, within the Scrum framework.

Scrum Masters are true leaders who serve the Scrum Team and the larger organization.

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

- Coaching the team members in self-management and cross-functionality;
- Helping the Scrum Team focus on creating high-value Increments that meet the Definition of Done;
- Causing the removal of impediments to the Scrum Team's progress; and,
- Ensuring that all Scrum events take place and are positive, productive, and kept within the timebox.

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

- Helping find techniques for effective Product Goal definition and Product Backlog management;
- Helping the Scrum Team understand the need for clear and concise Product Backlog items;
- Helping establish empirical product planning for a complex environment; and,
- Facilitating stakeholder collaboration as requested or needed.

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

- Leading, training, and coaching the organization in its Scrum adoption;
- Planning and advising Scrum implementations within the organization;

- Helping employees and stakeholders understand and enact an empirical approach for complex work; and,
- Removing barriers between stakeholders and Scrum Teams.

SCRUM EVENTS

The Sprint is a container for all other events. Each event in Scrum is a formal opportunity to inspect and adapt Scrum artifacts. These events are specifically designed to enable the transparency required. Failure to operate any events as prescribed results in lost opportunities to inspect and adapt. Events are used in Scrum to create regularity and to minimize the need for meetings not defined in Scrum. Optimally, all events are held at the same time and place to reduce complexity.

THE SPRINT

Sprints are the heartbeat of Scrum, where ideas are turned into value.

They are fixed length events of one month or less to create consistency. A new Sprint starts immediately after the conclusion of the previous Sprint.

All the work necessary to achieve the Product Goal, including Sprint Planning, Daily Scrums, Sprint Review, and Sprint Retrospective, happen within Sprints.

During the Sprint:

- No changes are made that would endanger the Sprint Goal;
- Quality does not decrease;
- The Product Backlog is refined as needed; and,
- Scope may be clarified and renegotiated with the Product Owner as more is learned.

Sprints enable predictability by ensuring inspection and adaptation of progress toward a Product Goal at least every calendar month. When a Sprint's horizon is too long the Sprint Goal may become invalid, complexity may rise, and risk may increase. Shorter Sprints can be employed to generate more learning cycles and limit risk of cost and effort to a smaller time frame. Each Sprint may be considered a short project.

Various practices exist to forecast progress, like burn-downs, burn-ups, or cumulative flows. While proven useful, 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.

A Sprint could be cancelled if the Sprint Goal becomes obsolete. Only the Product Owner has the authority to cancel the Sprint.

SPRINT PLANNING

Sprint Planning initiates the Sprint by laying out the work to be performed for the Sprint. This resulting plan is created by the collaborative work of the entire Scrum Team.

The Product Owner ensures that attendees are prepared to discuss the most important Product Backlog items and how they map to the Product Goal. The Scrum Team may also invite other people to attend Sprint Planning to provide advice.

Sprint Planning addresses the following topics:

Topic One: Why is this Sprint valuable?

The Product Owner proposes how the product could increase its value and utility in the current Sprint. The whole Scrum Team then collaborates to define a Sprint Goal that communicates why the Sprint is valuable to stakeholders. The Sprint Goal must be finalized prior to the end of Sprint Planning.

Topic Two: What can be Done this Sprint?

Through discussion with the Product Owner, the Developers select items from the Product Backlog to include in the current Sprint. The Scrum Team may refine these items during this process, which increases understanding and confidence.

Selecting how much can be completed within a Sprint may be challenging. However, the more the Developers know about their past performance, their upcoming capacity, and their Definition of Done, the more confident they will be in their Sprint forecasts.

Topic Three: How will the chosen work get done?

For each selected Product Backlog item, the Developers plan the work necessary to create an Increment that meets the Definition of Done. This is often done by decomposing Product Backlog items into smaller work items of one day or less. How this is done is at the sole discretion of the Developers. No one else tells them how to turn Product Backlog items into Increments of value.

The Sprint Goal, the Product Backlog items selected for the Sprint, plus the plan for delivering them are together referred to as the Sprint Backlog.

Sprint Planning is timeboxed to a maximum of eight hours for a one-month Sprint. For shorter Sprints, the event is usually shorter.

DAILY SCRUM

The purpose of the Daily Scrum is to inspect progress toward the Sprint Goal and adapt the Sprint Backlog as necessary, adjusting the upcoming planned work.

The Daily Scrum is a 15-minute event for the Developers of the Scrum Team. To reduce complexity, it is held at the same time and place every working day of the Sprint. If the Product Owner or Scrum Master are actively working on items in the Sprint Backlog, they participate as Developers.

The Developers can select whatever structure and techniques they want, as long as their Daily Scrum focuses on progress toward the Sprint Goal and produces an actionable plan for the next day of work. This creates focus and improves self-management.

Daily Scrums improve communications, identify impediments, promote quick decision-making, and consequently eliminate the need for other meetings.

The Daily Scrum is not the only time Developers are allowed to adjust their plan. They often meet throughout the day for more detailed discussions about adapting or re-planning the rest of the Sprint's work.

SPRINT REVIEW

The purpose of the Sprint Review is to inspect the outcome of the Sprint and determine future adaptations. The Scrum Team presents the results of their work to key stakeholders and progress toward the Product Goal is discussed.

During the event, the Scrum Team and stakeholders review what was accomplished in the Sprint and what has changed in their environment. Based on this information, attendees collaborate on what to do next. The Product Backlog may also be adjusted to meet new opportunities. The

Sprint Review is a working session and the Scrum Team should avoid limiting it to a presentation.

The Sprint Review is the second to last event of the Sprint and is timeboxed to a maximum of four hours for a one-month Sprint. For shorter Sprints, the event is usually shorter.

SPRINT RETROSPECTIVE

The purpose of the Sprint Retrospective is to plan ways to increase quality and effectiveness.

The Scrum Team inspects how the last Sprint went with regards to individuals, interactions, processes, tools, and their Definition of Done. Inspected elements often vary with the domain of work. Assumptions that led them astray are identified and their origins explored. The Scrum Team discusses what went well during the Sprint, what problems it encountered, and how those problems were (or were not) solved.

The Scrum Team identifies the most helpful changes to improve its effectiveness. The most impactful improvements are addressed as soon as possible. They may even be added to the Sprint Backlog for the next Sprint.

The Sprint Retrospective concludes the Sprint. It is timeboxed to a maximum of three hours for a one-month Sprint. For shorter Sprints, the event is usually shorter.

SCRUM ARTIFACTS

Scrum's artifacts represent work or value. They are designed to maximize transparency of key information. Thus, everyone inspecting them has the same basis for adaptation.

Each artifact contains a commitment to ensure it provides information that enhances transparency and focus against which progress can be measured:

- For the Product Backlog it is the Product Goal.
- For the Sprint Backlog it is the Sprint Goal.
- For the Increment it is the Definition of Done.

These commitments exist to reinforce empiricism and the Scrum values for the Scrum Team and their stakeholders.

PRODUCT BACKLOG

The Product Backlog is an emergent, ordered list of what is needed to improve the product. It is the single source of work undertaken by the Scrum Team.

Product Backlog items that can be Done by the Scrum Team within one Sprint are deemed ready for selection in a Sprint Planning event. They usually acquire this degree of transparency after refining activities.

Product Backlog refinement is the act of breaking down and further defining Product Backlog items into smaller more precise items. This is an ongoing activity to add details, such as a description, order, and size.

Attributes often vary with the domain of work.

The Developers who will be doing the work are responsible for the sizing. The Product Owner may influence the Developers by helping them understand and select trade-offs.

Commitment: Product Goal

The Product Goal describes a future state of the product which can serve as a target for the Scrum Team to plan against. The Product Goal is in the Product Backlog. The rest of the Product Backlog emerges to define "what" will fulfill the Product Goal.

A product is a vehicle to deliver value. It has a clear boundary, known stakeholders, well-defined users or customers. A product could be a service, a physical product, or something more abstract.

The Product Goal is the long-term objective for the Scrum Team. They must fulfill (or abandon) one objective before taking on the next.

SPRINT BACKLOG

The Sprint Backlog is composed of the Sprint Goal (why), the set of Product Backlog items selected for the Sprint (what), as well as an actionable plan for delivering the Increment (how).

The Sprint Backlog is a plan by and for the Developers. It is a highly visible, real-time picture of the work that the Developers plan to accomplish during the Sprint in order to achieve the Sprint Goal. Consequently, the Sprint Backlog is updated throughout the Sprint as more is learned. It should have enough detail that they can inspect their progress in the Daily Scrum.

Commitment: Sprint Goal

The Sprint Goal is the single objective for the Sprint. Although the Sprint Goal is a commitment by the Developers, it provides flexibility in terms of the exact work needed to achieve it. The Sprint Goal also creates coherence and focus, encouraging the Scrum Team to work together rather than on separate initiatives.

The Sprint Goal is created during the Sprint Planning event and then added to the Sprint Backlog. As the Developers work during the Sprint, they keep the Sprint Goal in mind. If the work turns out to be different than they expected, they collaborate with the Product Owner to negotiate the scope of the Sprint Backlog within the Sprint without affecting the Sprint Goal.

INCREMENT

An Increment is a concrete stepping stone toward the Product Goal. Each Increment is additive to all prior Increments and thoroughly verified, ensuring that all Increments work together. In order to provide value, the Increment must be usable.

Multiple Increments may be created within a Sprint. The sum of the Increments is presented at the Sprint Review thus supporting empiricism. However, an Increment may be delivered to stakeholders prior to the end of the Sprint. The Sprint Review should never be considered a gate to releasing value.

Work cannot be considered part of an Increment unless it meets the Definition of Done.

Commitment: Definition of Done

The Definition of Done is a formal description of the state of the Increment when it meets the quality measures required for the product.

The moment a Product Backlog item meets the Definition of Done, an Increment is born.

The Definition of Done creates transparency by providing everyone a shared understanding of what work was completed as part of the Increment. If a Product Backlog item does not meet the Definition of Done, it cannot be released or even presented at the Sprint Review. Instead, it returns to the Product Backlog for future consideration.

If the Definition of Done for an increment is part of the standards of the organization, all Scrum Teams must follow it as a minimum. If it is not an organizational standard, the Scrum Team must create a Definition of Done appropriate for the product.

The Developers are required to conform to the Definition of Done. If there are multiple Scrum Teams working together on a product, they must mutually define and comply with the same Definition of Done.

END NOTE

Scrum is free and offered in this Guide. The Scrum framework, as outlined herein, is immutable. While 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.

SCRUM GUIDE HISTORY

Ken Schwaber and Jeff Sutherland first co-presented Scrum at the OOPSLA Conference in 1995. It essentially documented the learning that

Ken and Jeff gained over the previous few years and made public the first formal definition of Scrum.

The Scrum Guide documents Scrum as developed, evolved, and sustained for 30-plus years by Jeff Sutherland and Ken Schwaber. Other sources provide patterns, processes, and insights that complement the Scrum framework. These may increase productivity, value, creativity, and satisfaction with the results.

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

© 2020 Ken Schwaber and Jeff Sutherland

This publication is offered for license under the Attribution Share-Alike license of Creative Commons, accessible at <https://creativecommons.org/licenses/by-sa/4.0/legalcode> and also described in summary form at <https://creativecommons.org/licenses/by-sa/4.0/>. By utilizing this Scrum Guide, you acknowledge and agree that you have read and agree to be bound by the terms of the Attribution Share-Alike license of Creative Commons.

Source:

<http://www.scrumguides.org/>

APPENDIX B – “SCRUM IS HARD AND DISRUPTIVE”

By Ken Schwaber, 2006.

1. Scrum is a framework for iterative, incremental development using cross-functional, self-managing teams. It is built on industry best practices, lean thinking, and empirical process control.
2. Scrum is optimized for high yield product management and product development. Scrum is particularly appropriate for high risk, complex, large projects and can be used when other parts of the endeavor are hardware or even waterfall development.
3. If waterfall suits current needs, continue using it.
4. An enterprise can use Scrum as a tool to become the best product development and management organization in its market. Scrum will highlight every deficiency and impediment that the enterprise has so the enterprise can fix them and change into such an organization.
5. Whenever an enterprise modifies or only partially implements Scrum, it is hiding or obscuring one or more dysfunctions that restrict its competence in product development and management.
6. The iterative, incremental nature of Scrum puts stress on the product development organization to improve its engineering skills and on the product management organization to optimize the return on investment of every release and project. The phrase, “That can't be done here” really means that it will be very difficult to do so. The gap between current practices and target practices is a measure of incompetence and competitive risk.
7. The use of Scrum to become an optimized product development and management organization is a change process that must be led from the top and requires change by everyone within the enterprise. Change is extremely difficult and fraught with conflict, and may take many years of sustained effort. Turnover of staff and management can be expected.
8. The most serious impediments to using Scrum are habits of waterfall, predictive thinking over the last twenty to thirty years; these have spawned command and control management, belief that demanding something will make it happen, and the willingness of

development to cut quality to meet dates. These are inbred habits that we aren't even aware of anymore.

9. The focus of using Scrum is the change from old habits to new ways of doing business. Scrum is not implemented or rolled-out as a process; it is used to foment change.
10. Scrum is not a methodology that needs enhancing. That is how we got into trouble in the first place, thinking that the problem was not having a perfect methodology. Effort centers on the changes in the enterprise that is needed.
11. Iterative, incremental development is much harder than waterfall development; everything that was hard in waterfall engineering practices now has to be done every iteration, and this is incredibly hard. It is not impossible, but has to be worked toward over time.
12. Managing a release or project to deliver only the highest value functionality and not deliver the rest optimizes value [and] is the job of product management and customers.
13. Self-managing teams are extremely productive. When they work closely with the customer to derive the best solution to a need, they and the customer are even more productive.
14. A team consists of people under pressure to do their best. Conflict is natural and the team needs to know how to deal with the conflict and have resources to draw on when needed.
15. The role of an enterprises management changes from telling people what to do to leading and helping everyone do their best to achieve goals. People aren't resources and managers aren't bosses.

Source:

<http://www.controlchaos.com/storage/scrum-articles/Scrum%20Is%20Hard%20and%20Disruptive.pdf>

APPENDIX C – SPRINT EXECUTION QUIZ ANSWERS

1. False. A sprint is a timebox. Its duration is fixed from sprint to sprint. However, the Scrum Team may change the sprint length if it is not suitable. Shorter sprints are preferred.
2. True, although that should be rare.
3. False. The Product Owner has the authority to terminate a sprint.
4. True.
5. False, the Team should maintain the task board and burn-down so that they own their own progress.
6. True, items for the next sprint are on the Product Backlog.
7. False, cross-functional teams need to have all the skills in the team, but not necessarily in every member of the team.
8. False, changing team composition will impact velocity, but it is impossible to tell if it will increase or decrease.
9. True, although it is difficult to resolve some impediments the ScrumMaster should try to solve them all as quickly as possible.
10. False, although this is common, some teams choose not to estimate tasks at all.
11. True, teams always focused on delivery and working at maximum capacity will not have down time to think about or implement improvements.
12. True, it is important for people to be close enough that you can see the expressions on their face when seated at your desk.
13. True, this is a commonly observed behaviour.

APPENDIX D – REFERENCES

BOOKS

1. *Essential Scrum: A Practical Guide to the Most Popular Agile Process*, by Kenneth S. Rubin.
2. *Agile Project Management with Scrum (Developer Best Practices)*, by Ken Schwaber.
3. *Succeeding with Agile: Software Development Using Scrum*, by Mike Cohn.
4. *Scrum: The Art of Doing Twice the Work in Half the Time*, by Jeff Sutherland.
5. *Implementing Lean Software Development: From Concept to Cash*, by Mary Poppendieck and Tom Poppendieck.
6. *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*, by Eric Ries.
7. *Agile Product Management with Scrum: Creating Products that Customers Love*, by Roman Pichler.
8. *The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win*, by Gene Kim, Kevin Behr, George Spafford.
9. *Continuous Delivery: Reliable Software Releases through Build, Test and Deployment Automation*, by Jez Humble and David Farley.
10. *Scaling Lean and Agile Development: Thinking and Organizational Tools for Large-Scale Scrum*, by Craig Larman and Bas Vodde.
11. *Management 3.0: Leading Agile Developers, Developing Agile Leaders*, by Jurgen Appelo.
12. *Coaching Agile Teams: A Companion for ScrumMasters, Agile Coaches, and Project Managers in Transition*, by Lyssa Adkins
13. *Agile Retrospectives: Making Good Teams Great*, by Esther Derby and Diana Larsen

14. *The Five Dysfunctions of a Team: A Leadership Fable*, by Patrick Lencioni.
15. *The Toyota Way*, by Jeff Liker.
16. *Extreme Programming Explained*, by Kent Beck and Cynthia Andres.
17. *The Mythical Man-Month*, by Frederick P. Brooks Jr.
18. *Refactoring: Improving the Design of Existing Code*, by Martin Fowler et al.
19. *Working Effectively with Legacy Code*, by Michael Feathers.
20. *Specification by Example: How Successful Teams Deliver the Right Software*, by Gojko Adzic.
21. *User Stories Applied: For Agile Software Development*, by Mike Cohn.
22. *The Principle of Product Development Flow: Second Generation Lean Product Development*, by Donald G. Reinertsen.
23. *Reinventing Organizations*, by Frederic Laloux.

ARTICLES

1. *The New New Product Development Game*, by Hirotaka Takeuchi and Ikujiro Nonaka. Harvard Business Review, January–February 1986.
2. *The Scrum Guide*, by Jeff Sutherland and Ken Schwaber.
www.scrumguides.org.
3. *The Scrum Primer*, by Pete Deemer, Gabrielle Benefield, Craig Larman and Bas Vodde. www.scrumprimer.org.

VIDEOS

1. *Scrum Training Series*, by Michael James.
www.scrumtrainingseries.com.

2. *TED Talk about Wikispeed*, by Joe Justice.
www.youtube.com/watch?v=x8jdx-lf2Dw.
3. *The Power of an Agile Mindset*, by Linda Rising.
www.agilealliance.org/resources/learning-center/keynote-the-power-of-an-agile-mindset
4. *Drive: The Surprising Truth about What Motivates Us*, by Daniel Pink.
www.youtube.com/watch?v=u6XAPnuFjJc
5. *I, Tomato: Morning Star's Radical Approach to Management*.
ReasonTV. www.youtube.com/watch?v=qqUBdX1d3ok
6. *Agile Product Ownership in a Nutshell*, by Henrik Kniberg.
www.youtube.com/watch?v=502ILHjX9EE
7. *Spotify Engineering Culture*, by Henrik Kniberg.
www.youtube.com/watch?v=R2o-Xm3UVjs
8. *Agile Scrum Working Agreements*, by Tirrell Payton.
www.youtube.com/watch?v=CStypsb3GKI
9. *Scrum Repair Guide: Grooming the Product Backlog*, by Mike Cohn.
www.youtube.com/watch?v=kXJuss2w39w
10. *Joy, Inc.*, by Richard Sheridan.
www.youtube.com/watch?v=Oe8VTi3m8U8
11. *Reinventing Organizations*, by Frederic Laloux.
www.youtube.com/watch?v=gcS04BI2sbk

SELF-EVALUATION

What: What have I learned?

So What: What will change because of what I learned?

Now What: What will I do now because of it?

FEEDBACK FORM

Date: _____

Name (optional): _____

1. Rate this course out of 10. Circle one number.

1 2 3 4 5 6 7 8 9 10

2. Tell us what you liked about the course.

3. Tell us what would make the course a perfect 10 for you.