Learning Practical

SCRUM

in 21 Minutes

A Practical Handbook for Product Owner, Scrum Master, and The Team

Daniel Foo

DanielCoding.net

Published by: DanielCoding.net
Email: hello@danielcoding.net
Website: http://danielcoding.net

October 2015

DanielCoding.net

The topics outlined in this book might not be suitable for every individual and are not guaranteed or warranted, whether express or implied, to produce any particular results. The strategies, advice, ideas and suggestions are written as general guide and specific professional advice may be necessary.

By reading this book, you agree to the following. You understand that the information presented contained on this book is for personal entertainment purpose only. It is not to be considered legal or personal advice. The Publisher and Author have made every effort to ensure all materials provided are accurate and updated at the time of writing however do not guarantee or warranty the information would remain valid at the time of reading. The Author and Publisher specifically disclaim any responsibility for any liability, losses or damages of any kind as a consequence, directly or indirectly, arising from the use and application of any of the contents of this book.

Table of Content

Foreword

Scrum - the buzz word everyone in the software development industry has been using. Today, many software development teams adopt Scrum as the default approach to build and ship software. Some adopted Scrum because they realized the prescriptive ceremonies are able to help resolving very classical software development problems; while some adopted Scrum because it is the “cool” thing to do.

“Scrum doesn’t work!” is a common claim for the second group who practice Scrum because it was “cool”. They were not able to experience the benefits for adopting Scrum not because Scrum does not work, it is simply because they lack the fundamental understanding about Scrum.

This book is specifically written for development teams who have heard about Scrum, read a few articles on Scrum and have practiced Scrum to certain extend but somehow do not (or not yet) see how Scrum can help in their development process.

The objective of this book is to provide practical insight for development team members on HOW to practice Scrum and WHY do we practice Scrum in certain manner. Once development team members understand the reasons why the Scrum ceremonies are designed in specific manner,

the members will appreciate Scrum better. In fact, when development team members understand the WHY, the HOW will come naturally.

There are 3 chapters in the book: Agile Manifesto, Scrum Ceremonies & The Team. Agile Manifesto is the foundation of Scrum. By understanding the fundamental of Agile, it greatly improve the team attitude and spirit on how to approach an Agile development cycle. Scrum Ceremonies are the heart of Scrum. The prescriptive ceremonies are the activities that make a Scrum team, Scrum. This section goes beyond introducing what are the ceremonies. It covers WHY do we do certain ceremonies so that the team members can better appreciate each ceremony. The Team is element which gives life to Scrum. The Scrum team members are the main drivers to make or break the Scrum team. This section cover extensively what each role in Scrum team responsibility are from a practical aspect.

I hope this guide will serve you well in your quest to build a stronger and more matured Scrum team in your organization.

Cheers!

Daniel Foo

Introduction

Scrum is an Agile framework for managing complex projects. Scrum is mainly used in software development projects, although it can also be applied to any complex, innovative scope of work.

Scrum is a simple framework but often it is hard to grasp a good understanding. This book is designed to provide the underlying understanding for Scrum to equip readers with practical actions.

This is a quick overview on the chronological ceremonies in Scrum:

1. Product Owner creates a prioritized to-do list known as product backlog.
2. During Sprint Planning, The Team pulls a certain number of product backlog from the top of that list. The Team will decides how to implement the backlogs.
3. The Team has a certain amount of time in a Sprint to complete its work. Typically it is 2-4 weeks. The Team members meets each day to assess the progress through a Daily Standup Meeting.
4. Along the Sprint, Scrum Master keeps the team focused on its goal, which is to complete all committed stories.

5. At the end of the Sprint, the work should be potentially shippable. The decision to ship the product or otherwise is entirely up to Product Owner.
6. The Sprint ends with a Sprint Review and Sprint Retrospective participated by all Scrum team members.
7. As the next Sprint begins, The Team pulls another collection of the product backlog and begins working again.

Before we go into each ceremony in Scrum, let us take a step back to understand Agile, the foundation of Scrum.

Agile Manifesto

While Scrum is an iterative and incremental agile software development methodology for managing product (software) development, we should not discuss about Scrum without first understanding Agile. In fact, many teams could not run Scrum successfully because they do not have a clear fundamental understanding about Agile manifesto. When we understand the idea of Agile, all prescriptive ceremonies in Scrum would make much better sense.

Now, let us look the 4 manifesto for Agile software development:

1. Individuals and interactions over processes and tools
2. Working software over comprehensive documentation
3. Customer collaboration over contract negotiation
4. Responding to change over following a plan

Individuals and Interactions Over Processes and Tools

In development team, Agile encourages team members to interact with each other such as talking face to face, instant messaging, or email communication – instead of heavily depend on project management tool such as TFS, Trello, Asana, JIRA, etc. For example, if a team member notices there is a defect in the code written by another team member, he is encouraged to talk the other team member directly (maybe to discuss the root cause and possible solutions) instead of creating a ticket, assign to him in JIRA and call it a day.

When individual interaction take precedence, team members build the bonding to cover each other's back. When processes and tools take precedence, team members would only look at their own plate and teamwork rarely happens. Having said that, it does not mean processes are not important. They are important to facilitate the operation but processes and tools should not be the central of development team. It is the team members who are building the product (software), the processes and tools are merely to facilitate them to get their job done.

Working Software Over Comprehensive Documentation

Would you rather to have a working software, or comprehensive documentation? Theoretically it is best to have both. Unfortunately projects often running behind the most precious commodity – time. If the team has to choose either one, it should always be a working software. First, documentations are often outdated and team members are unconsciously trained to not rely on documentations because the perception is documentations are inaccurate anyway. Second, it is much easier to navigate a working software than to navigate a 200 pages functional specification document. Documentations are often the poor fella sitting at the shared drive or intranet that people rarely looks at.

Always choose to build a working software over writing comprehensive documentation when facing the challenge of limited time. After all, what is the purpose of documentation if the software is not working?

Customer Collaboration Over Contract Negotiation

This might not be directly relevant to the team members who are actively building the product. It is a mindset the business decision makers should have. When business negotiates a contract with customer (regardless internal customers or external customers), a set of objective or functional requirement is often agreed upon. However with today's market speed of change, circumstances tend to change faster than business could anticipate. It is impractically for business to re-negotiate a new contract

whenever a change take place. Both the development team and business must understand that collaboration is what makes great software. Customer has the understanding of the market to describe the requirements; while development team has the skill to transform requirements into features.

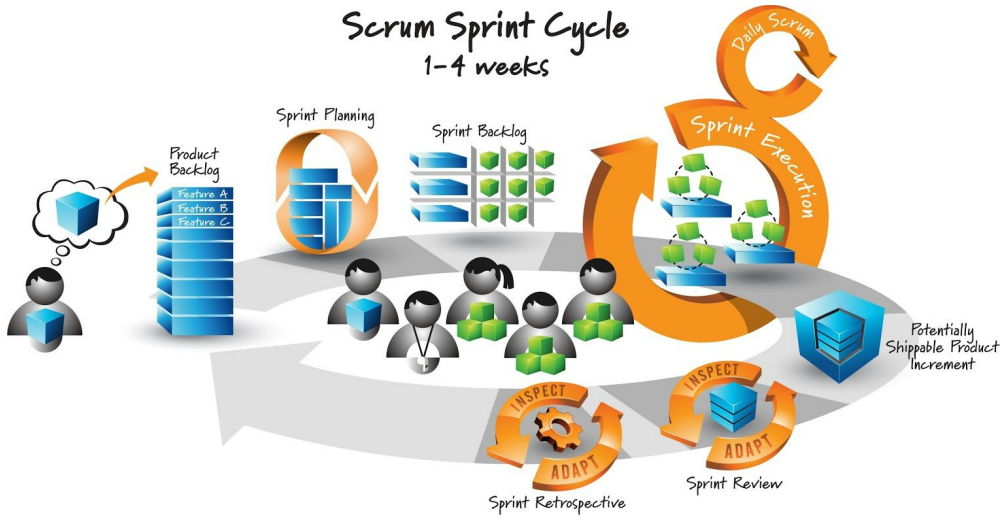
Collaboration with customers is one of the fastest ways to make customers happy. Generally customers do find greater satisfaction if customers are aware that development team put their business success in their best interests over milking cash from them with a contract.

Responding to Change Over Following a Plan

If Plan A doesn't work, don't worry, the alphabet has 25 more letters. The ability to respond to change is the heart of Agile. Thanks to today's speed of change, the only constant in the market is change. If internal speed of change is slower than external, the business is doomed to fail. If things tend to change so often, why should anyone stubbornly stick to a plan? A plan is made based on best knowledge and known facts at that point of time decision is made. It provides a guideline to keep team members on track. Team members might (in most cases, will) learn higher quality knowledge and identify invalid facts along the way. The team would make use of the new knowledge or more accurate facts to deal with the situation more efficiently.

Managers should be the leaders to head the change response but ironically, it is also the managers who are often very reluctant to change and claim “We have a plan, we need to follow our plan”. This could happen for various reasons and the justification could be valid or otherwise. Not all changes happen for a better outcome however if the team could identify a change is happening (or has happened) and an appropriate response is required, the team should respond by introducing a new plan. There is little value in protecting the ego and refusing to change. Responding to change might be painful in the beginning, but it always yield better return for the product on the long term.

Scrum Ceremonies



In previous chapter, we discussed the fundamental of Agile development which covers 4 Agile Manifesto. Lacking understand of Agile Manifesto often lead team members to practice Scrum ceremonies for the sake of practicing – without realizing the purpose of each ceremony. If your intention is to run a successful Scrum team, I highly recommend you to grasp a solid understanding of the 4 Agile Manifesto before moving to the prescriptive ceremonies in Scrum.

Guidelines for Scrum Team

Each Sprint typically runs for 2 weeks. The duration of a Sprint is suggested to be 2 weeks to 4 weeks. Through experience, a 1-week Sprint is too short that leads to rushing and too much overhead while

3-week or 4-week Sprint is too long that leads to unnecessary idle time and big-bang implementation and deployment which makes Sprint harder to manage. A 2-week Sprint is ideal in most situation.

An ideal size of a Scrum team consists of 1 Product Owner, 1 Scrum Master and 7 (with +/- 2) team members. A team with less than 5 team members is too small to build anything significant; while a team of more than 9 team members are little too large to manage. Depending on the scope of the project and resource availability in the organization, 7 (+/- 2) team members is not a stone-carved law to follow but rather a best-practice guideline.

Sprint Grooming

Sprint Grooming is a pre-sprint activity that takes place before Sprint starts. During grooming session, team members would clarify the stories written by Product Owner. Stories writing must have been completed by Product Owner, to allow team members to clarify issues regarding implementation, either functional or technical or both.

The following is the format of how a story should be written:

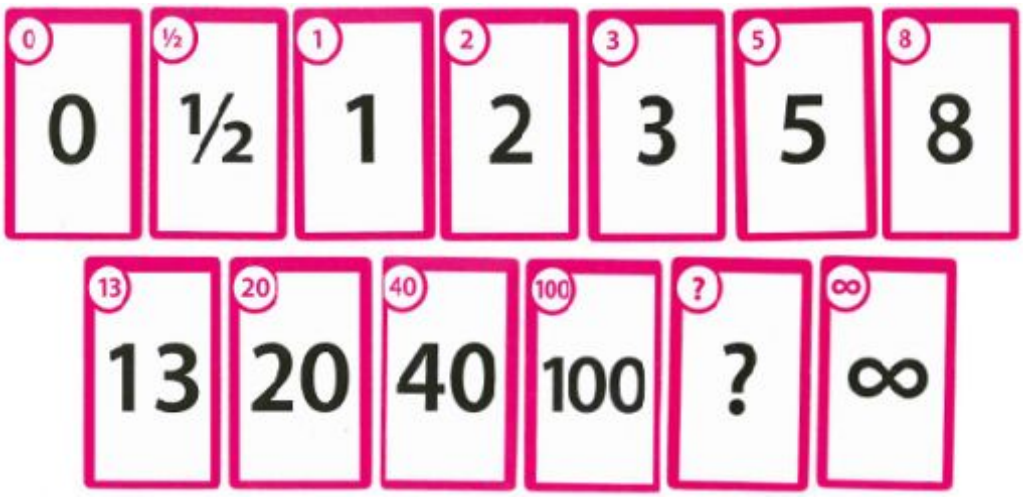
As a <type of business user>,

I want <to perform certain task(s)>

so that I can <achieve some goals / benefits / values>

During this stage, the team might need some time to carry out research to evaluate whether a story could be implemented and how much effort is required. On the other hand, Product Owner could encounter edge scenarios asked by The Team. Product Owner potentially need time to check with the respective stakeholders to finalize a decision. Although such situation is expected, a matured Scrum team and a seasoned Product Owner who understand business use cases well would keep such “get-back-to-you-later” discussion to minimal. Sprint grooming could take place for multiple sessions depending on how productive each session went.

Estimation



Before Sprint Planning begins, team members would have all questions in stories answered and have finalized the story point for each story. Note that estimation should NOT be hour estimation. Estimation done in hours

defeat the purpose of Scrum. Scrum estimation is to be done using relative measurement. Tools such as Poker card or T-shirt size are good options to measure the complexity of story. In order for relative estimation to work, team members must have a common ground on the complexity of base point 1. For example, if 1 point is equivalent to X amount of complexity, a story that carries 5 points means 5 times more complex than the 1 point story. 5 times more complex does not necessary means it will takes 5 times the effort to complete it. Again, the base point of 1 is vitally important to form the base understanding among team members. Without defining base point of 1, story point estimation would be meaningless.

If you have a smartphone, you can download Scrum Poker Cards app from **Google Playstore** or **iTune**.

Sprint Planning

During Sprint Planning, team members and Product Owner would finalize which story to be committed in the coming Sprint. Depending on the team velocity, total story point and priority of each story, Product Owner (with the feedback and influence of team members) would select the best combination of stories to be committed for the coming Sprint. Note that requirement clarification should be kept minimum during Sprint Planning as requirement clarification should have been completed during earlier Sprint Grooming session. The more requirements was clarified during

Sprint Planning session, the less accurate the story point estimation represents.

Daily Standup Meeting

The purpose of daily stand up provides a platform for the team to expose potential problems early. During stand up meeting, 3 questions would be covered by each team member:

1. What did I accomplish yesterday?
2. What will I do today?
3. What impediment am I facing?

Question #1 (What did I accomplish yesterday?) is designed to ensure team member accomplished what he has committed (to the rest of the team) on the previous working day. Having the team member to proactively report his progress is a much more motivating approach compared to having a traditional project manager to ask “Have you completed Task X?”. It gives a sense of pride to the team member for accomplishing tasks and announcing them to the rest of the team.

Question #2 (What will I do today?) is designed to ensure team members are moving forward everyday throughout the Sprint. By having team members to commit to story on their own will, they would naturally have a higher sense of ownership and accountability for the stories.

If everyone is moving forward together, then success takes care of itself. -Henry Ford

The purpose of Question #1 and #2 are to ensure the team members are being productive everyday. When Scrum Master notices a team member has been stuck with the same task for longer than expected, it is a signal that the team member is experiencing difficulty. A seasoned Scrum Master would spot this early to resolve it quickly rather than getting unpleasant surprises at the end of the Sprint.

Question #3 (What impediment am I facing?) is designed for team members to raise any obstacle they are facing. Often in a less matured Scrum team, all members would say “No impediment” during Daily Standup meeting. A “No impediment” response does not necessary mean no problem. The team members are sometimes too shy to raise the obstacle during daily standup meeting. This is a common observation for newly established Scrum team or when a new team member joined the team. In more matured Scrum team, team member would proactively raise obstacles during stand up and quickly resolve them among themselves.

An important note to remember during Daily Standup meeting is the “people” element. The focus should be placed on the team members’ ability to keep moving forward. A poorly run Scrum team shift the focus

away from the team members to the status of tasks completion. This is due to traditional project managers who are not familiar with Scrum approach feel more comfortable getting tasks status update instead. Through experience, focusing too much on tasks status update is a less efficient approach to manage Scrum team which often lead to 11th hour additional tasks because team members are “trained” to *close* a task rather than to *complete* a task. A more efficient approach is to focus on the team members’ ability to progress. Highly motivated Scrum team members would proactively help each other and do whatever it takes to complete all committed stories. At times, the team would resolve obstacles even before the Scrum Master is aware of the obstacles.

One of the most important roles for Scrum Master is to create a supportive and safe environment for team members to speak up during daily standup. Blaming particular team member or finger pointing must be avoided at all cost. However, that does not mean being lenient while dealing with troubles. The guideline is to be critical on issues, be kind to people. Scrum Master should not dominate Daily Standup discussion. Scrum Master is playing the role of a “referee” to ensure team members are getting the most value out of Scrum values. A Scrum Master in a matured Scrum team would merely serve as an observer and only speak up when help is needed. A Scrum Master is also known as a servant leader. In other words, the Scrum Master is to serve the team members to complete the Sprint.

Daily stand up is designed to run for 15 minutes or less – this is why team members should physically stand up during the session. Sitting comfortably encourages team member to drag the meeting longer than required. The act of standing up to conduct the meeting encourages team members to finish the meeting in 15 minutes or less because it would be too painful to stand for too long.

Often, team members are tempted to discuss specific issue with another team member during stand up, which potentially be a waste of other team members' time. Story specific discussion should be taken offline among the team members working on the specific story. Scrum Master has to be quick to call a cut when story specific discussion happens, unless that discussion is applicable to all team members.

Although all team members standing up together physically might not be feasible involving remote or off-shore team members, the fundamental objectives of daily stand up should not be neglected due to geographical limitation.

Sprint Review

Sprint Review is the activity where the whole Scrum team demonstrates the completed stories to Product Owner. The team member who

implemented the respective stories is responsible to run the demonstration. Stakeholders or Subject Matter Experts would often join the review / demo session to understand the newly implemented stories and provide appropriate feedback.

The objective of Sprint Review is to demonstrate the product incremental changes to Product Owner and stakeholders, so that there will be no big-bang surprises. Imagine in a non-Scrum approach of development, a team works on a product for 9 months and when they finally demonstrate the working product to customers, the customers feedback “No, this is not what we want”. 9 months of development effort would go down to drain. Sprint Review serves as a milestone and checkpoint to allow Product Owner and stakeholders to acknowledge the team is moving towards the right direction and they are happy with each milestones.

At the end of Sprint Review, Product Owner would provide feedback whether he is happy with the work done in the last Sprint and decide whether to ship / release / deploy the product. Note that in Scrum, at the end of every Sprint there MUST be a potentially shippable product. However, whether to ship the product or otherwise is entirely up the Product Owner.

Sprint Retrospective

Sprint Retrospective is the meeting where team members reflect back how well did the last Sprint go and discuss if there is any room for improvement.

Sprint Retrospective includes 3 main topics for discussion:

1. What went well?
2. What went wrong?
3. What can be improved?

Question #1 (What went well?) is designed to acknowledge the good doing happened during the Sprint. A team member who has gone the extra miles to help another team member should be acknowledged. An unforeseen obstacle during the Sprint has been proactively resolved by the team should be acknowledged. The idea is to acknowledge the team members who has done an outstanding job during the Sprint and encourage the team to continue the good teamwork.

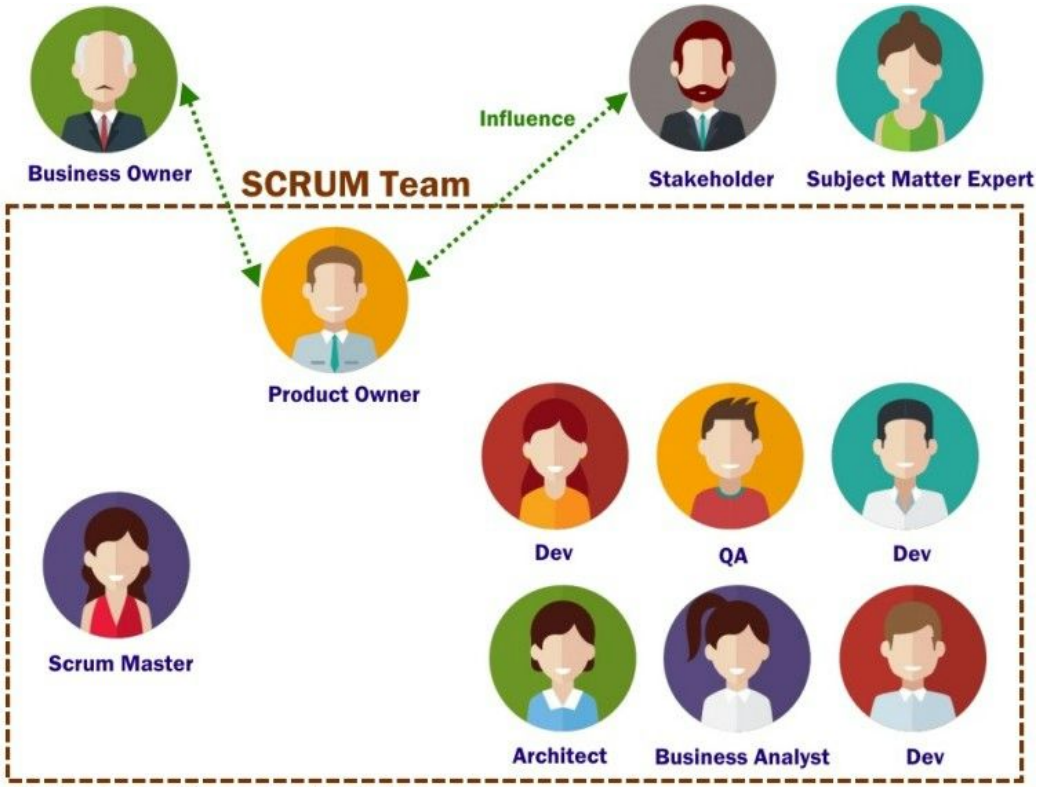
Question #2 (What went wrong?) is designed to identify areas for improvement. No problems should ever be swept under the carpet. If there was an unpleasant incident due to process inefficiency, it should be highlighted. If there was an external interruption that caused the reduction in team productivity, it should also be highlighted. The idea is to identify what are the pain points for the Scrum team, so that the team could optimize and increase The Team velocity.

Question #3 (What can be improved?) is a continuation of Question #2. The issues identified earlier should be discussed and identify a solution by the team, so that the same problem would not happen in the coming Sprint. The idea of Question #2 and #3 are to strike for continuous improvement in the Scrum team. No matter how good a Scrum team is, there is always room for improvement. This concept also known as Kaizen (改善) which means “improvement for the better”.

Often in less matured Scrum team, team members are less likely to speak up what is in their mind. Scrum Master could consider enforcing every team member to take turn answering all three questions above to get team members be more comfortable speaking during Sprint Retrospective. Even if the team member says something like “Same with the other guy” is a good start. In more matured Scrum teams, team members would proactively prepared a list of topic to bring into Sprint Retrospective for discussion.

While Scrum ceremonies are the activities that make a Scrum team, team that practices Scrum ceremonies for the sake of practicing gains little value for running project in Scrum. It is important to remember the underlying objective of each ceremony to gain the most out of these carefully designed ceremonies. In next chapter, we will discuss more about The Team (people) in Scrum.

The Team



SCRUM team members consist of 3 main roles: Product Owner, Scrum Master and The Team. Each role are closely inter-related. It is recommended for Scrum team members to be physically sitting together in the same location whenever possible.

Product Owner

Product Owner (PO) is the person who is ultimately responsible for the product. The product could be a system, a small piece of software or even a component. He often understands the historical evolution of the product,

well versed with the product current functionality, and define the vision of what the product eventually will be.

One of the most important guideline in Scrum is to build a potentially shippable product at the end of every Sprint. PO is the person who ultimately decides whether or not to ship the product. Whatever decision PO made, he is answerable to the business in the larger scope. The decision made by Product Owner are often influenced by other Stakeholders or Subject Matter Experts and the business owner.

Ideally, PO is the person who is much closer to the customers (both internal customers or external customers). He understands the business challenges the customers are facing. He will communicate these business challenges and solutions to The Team by writing user stories so that The Team can build the necessary features.

A user story follows the following format:

> ***As a*** *<type of business user>,*
>
> ***I want*** *<to perform certain task(s)>*
>
> ***so that I can*** *<achieve some goals / benefits / values>*

For example:

> ***As a*** *content editor*

I want *to be able to update my company website content without changing the source code*

so that I can *frequently let the visitors know the latest promotion*

During Sprint Planning and Spring Grooming session, PO will provide as much details as necessary to The Team related to the stories. At this point, PO would have completed writing each story and identify what is the priority for each story so that The Team can determine which story to pick first in Sprint Planning.

Some Product Owners who have a strong technical background might choose to write code together with The Team. However, he might not be able to commit on complicated stories or large number of stories due to his other obligations within the business. There are also some Product Owners who do not code but will participant in testing to work very closely with The Team.

In some organization structure, a business analyst might play a portion of Product Owner's role due to the inaccessibility to Product Owner. Issues such as time zone difference or other business commitment might not allow the Product Owner to have high availability for The Team. Business analyst will step in to clarify the questions The Team has. Whenever business analyst is in doubt, he will clarify the matters with Product Owner.

Scrum Master

Scrum Master is responsible to resolve any problem for the team members throughout a Sprint. Scrum Master is expected to understand different Scrum ceremonies better than the rest of the team members so that he can make the right call when conflicts arise. Scrum Master is the person who enforces the prescriptive guidelines in Scrum to ensure the Scrum team is able to reap the most values for adopting Scrum.

Having said that, Scrum Master does not have to be the most senior or most technically competent person in the Scrum team. In fact, Scrum Master might not know technical at all. Scrum Master's main responsibility is to clear out obstacles for The Team and enforce Scrum guidelines throughout the Sprint.

Let's walk through Scrum Master's specific responsibilities throughout the Sprint.

Sprint Grooming

Prior to the Sprint Grooming, Scrum Master has to ensure Product Owner has completed writing every story and priority has been placed on every story.

Scrum Master has to call for a Sprint Grooming few days prior to Sprint Planning. A few days of buffer should be allocated before Sprint Planning because Sprint Grooming sometimes cannot be completed in a single session. During Sprint Grooming Scrum Master needs to ensure The Team fully understands the stories written by Product Owner and ensure The Team has all necessary resources to kick start the Sprint.

After Sprint Grooming, Scrum Master has to call for another session for The Team to estimate the story point for each story. Often in Scrum team there are 1 or 2 team members who are more senior or has deeper domain knowledge with the product. They will indirectly influence the story point estimation given by other team members as the less senior team members tend to look up to them. Scrum Master has to spot this early and encourage the junior team members to independently provide story point estimation and justify why a story should have a higher (or lower) story point. Scrum Master has the responsibility to ensure all team members are working in harmony rather than dominated by senior team members.

Sprint Planning

During Sprint Planning, Scrum Master has to facilitate the Scrum team to finalize the stories to be committed in the coming Sprint.

In the perspective of Scrum, stories to be committed are always in accordance to priority defined by Product Owner. Among all prioritized stories, together with their story point, The Team will commit the stories with highest priority and make a cut off at the point of hitting The Team capacity (velocity). Occasionally, 1 or 2 stories will be reshuffled for the best match of priority and story point to fit the team capacity within one Sprint. This has to be done with the acknowledgement of Product Owner.

Scrum Master's responsibility during Sprint Planning is to ensure the above guidelines are followed. During Sprint Planning, Scrum Master has no say over whether to commit or not to commit certain story. The decision is entirely depend on the discussion between Product Owner and The Team. Again, Scrum Master role is to facilitate this process and to conduct a smooth planning session.

Daily Standup Meeting

Daily stand up is a quick 15 minutes catch up for The Team. Scrum Master has to ensure the following questions are answered by each team member:

1. What did I accomplish yesterday?
2. What will I do today?
3. What impediment am I facing?

On Question 1 (What did I accomplish yesterday?): Under healthy progress, everyday every team member would have completed certain task to contribute to the completion of the stories. However, if Scrum Master notices a team member has not been able to accomplish anything in the last working day, it's a sign that the team member is stuck at certain task. Scrum Master has the responsibility to find out what caused it and assist the team member to resolve it. Note that since Scrum Master might not be the most technically competent person, he might not be able to directly help the team member who is stucked on a technical task. However Scrum Master has to use his resourcefulness to arrange the necessary resources (people, documents, etc) to help the team member to resolve his impediment.

On Question 2 (What will I do today?): Under healthy progress, everyday every team member will pick a new task or continue working on the remaining task from yesterday. However, if Scrum Master notices a team member do not have the next task to move to, it is a sign The Team has committed less story than The Team velocity allows. If such situation persists, Scrum Master has to consider increasing The Team velocity in the coming Sprint by committing more stories. In some occasions, a team member dares not to pick up certain task because it is too difficult / not enough domain knowledge / not sure what is the solution, Scrum Master has to identify the appropriate resource and make necessary arrangement to help the team member to proceed with the story.

On Question 3 (What impediment am I facing?): Under most circumstances, team members will report there is no impediment. When a team member raises an impediment, Scrum Master should not dismiss the impediment lightly and should pay serious attention to resolve the impediment quickly. It is Scrum Master's responsibility to make use of his resourcefulness to make the necessary arrangement to get the impediment resolved as soon as possible to allow The Team to continue their progress.

Scrum Master should not dominate Daily Standup meeting. A good Scrum Master will maintain his silence during Daily Standup meeting but he is also quick to voice up when there are team members diverting from the objective of Daily Standup meeting.

Sprint Review

Sprint Review is a session driven by The Team to demonstrate the deliverable (completed stories) to Product Owner. Scrum Master involvement during Sprint Review is often minimal.

Sprint Retrospective

Sprint Retrospective is the session to reflect what happened in the last Sprint. Scrum Master needs to get The Team to cover the following topics:

1. What went well?
2. What went wrong?
3. What can be improved?

During the Sprint Retrospective, Scrum Master has to maintain a safe and harmonious environment for team members to speak up.

Scrum Master also has to ensure the team members understand the purpose of Sprint Retrospective is to provide continuous improvement for the coming Sprints rather than finding fault happened in the last Sprint. Therefore finger pointing should be avoided. Scrum Master has to ensure The Team is carrying out the session by constructively identify the root cause of “what went wrong” and openly identifying “what can be improved” for the future Sprints.

Scrum Master has to take note of the issues being discussed during Sprint Retrospective. Sometimes, The Team will treat Sprint Retrospective as a good chat but forget about what has been agreed to do by the next Sprint. In this situation, Scrum Master has to step in to remind The Team those items that have been agreed upon and The Team should honor the agreement made during Sprint Retrospective.

In newly established Scrum team, The Team might be uncomfortable to speak up during Sprint Retrospective. A workaround for this scenario is for

Scrum Master to request every team member taking turn to give at least 1 point for the 3 topics. This is also useful for matured Scrum team as The Team members often have too many points to bring up. Having the team members to take turn to speak is an information flood control.

Of all the ceremonies in Scrum, a Scrum Master should NEVER dominate any session. In fact, Scrum Master is known as the servant leader. Scrum Master is the person to serve the team to ensure all team members are able to carry out their tasks efficiently.

In matured Scrum team, Scrum Master often play a passive role to monitor The Team is following Scrum guidelines. However in a newly established Scrum team (or team members who are less familiar with Scrum), Scrum Master has to play an active role to ensure The Team members are following the appropriate Scrum guidelines. If team members are diverting from Scrum guidelines, Scrum Master have to be quick to speak up to enforce Scrum guidelines and explain the purpose of the specific guidelines.

The Team

The Team is the main driver in a Sprint. The Team typically consists of Developer, Quality Assurance, Architect, Business Analyst and sometimes

even Graphic Designer, Database Administrator to work together to deliver the potentially shippable product at the end of Sprint.

The Team is cross functional. The Team is able to step into different functional areas to ensure an user story is fully implemented.

In an ideal Scrum team, everyone in The Team is equal. There is no difference in title, role or seniority. The ideal Scrum does not recognize whether a member is a Developer, Quality Assurance, Architect, Business Analyst, Database Administrator or Designer. Everyone in The Team is equivalently qualified to pick up any story. Everyone in The Team should be able to perform designing work, coding, testing, writing documentation, etc. The idea is to take on the full stack of tasks in a user story. However, this is often impractical in many today's organizations.

Today, many organizations hiring process is role base. For example, the hiring manager might put up a job vacancy looking for *"Developer with 5 years of ASP.NET & C#.NET experience"* or *"Quality Assurance with 1 years experience in building automation using Selenium"*. Many development teams are not ready to embrace the ideal Scrum ideology where every member can do full stack of task in a story.

With that, a less drastic and practical approach to embrace Scrum value of “everyone is equal” is by actively encouraging team members to step into different roles within The Team.

The idea is to allow team members to step into different roles and doing different tasks they are not “officially” responsible for. For example, an Architect is encouraged to actively code and implement user stories; a Developer is encouraged to help out in testing; a Business Analyst is encouraged to help out in testing, a Quality Assurance is encouraged to help out in documentation, etc.

When team members step into different roles within team, team members learn to see the development task from a different perspective and be able to communicate from the perspective of different roles. Not only this will keep team members’ job interesting, but also enhance team members’ career growth by having wider scope of development experience.

Let’s walk through the main responsibility for The Team during different phases.

Sprint Grooming

Every members in the team should get good comprehension of every story. All questions related to user story implementation should be clarified

by team members. Junior members tend to rely on the senior members to “figure out” the story. This should be avoided. Again, every team member is equal in a Scrum team. Every team member should understand every story the best they could because every team member is expected to be capable to pick up the story later to implement it.

Estimation

All team member has the equal right to decide what story point the story should have. Junior members tend to look up to senior members and follow their story point estimation. This should be avoided. Every team member is encouraged to provide independent story point estimation. When estimation is significant varied, the team member is encouraged to provide justification why the given story point is significant higher or lower. Junior team member might not necessary know less on specific scope of the story. The purpose of such explanation is to allow the rest of the team members to identify aspects they might have overlooked or unaware.

Sprint Planning

Assuming Sprint Grooming is fruitfully conducted and estimation is agreed earlier, Sprint Planning will be a smooth process for The Team to formally commit to a set of user story up to The Team capacity. Product Owner

might shuffle around a few stories, The Team will have to provide their insight whether such arrangement is feasible.

Daily Stand Up

Once the Sprint started, team members will pick up story and implement the full stack story. On daily basis, team members will gather at the same time to report the following questions to the rest of the team:

1. What did I accomplish yesterday?
2. What will I do today?
3. What impediment am I facing?

The purpose of Daily Standup Meeting is for team members to be aware of what each other are working on to avoid redundant effort. It is also a platform for The Team to expose any potential problem early enough for the whole Scrum Team to act on.

Sprint Review

Sprint Review is the time for team members to proudly demonstrate what they have accomplished in the Sprint. As the full stack of task for a particular story is implemented by 1 or 2 team members, the respective team members naturally have stronger sense of ownership on the story. It

is best for the respective team member(s) to demonstrate the completed stories to Product Owner during Sprint Review. Product Owner will provide feedback whether he is satisfied with the deliverable. Team members will listen to PO feedback and take necessary actions based on the feedback.

Sprint Retrospective

Sprint Retrospective is the time for team members to reflect back what happened throughout last Sprint by answering:

1. What went well?
2. What went wrong?
3. What can be improved?

Team members are encouraged to acknowledge other team members who has done well. This is not only subjected for the senior members to recognize the junior members effort. Junior members are also encouraged to identify other team members who have done a good job in the last Sprint. Remember, everyone is equal in a Scrum team.

While addressing what went wrong and what can be improved, team members are encouraged to be courageous to speak up yet be compassionate while speaking. A Scrum team is very much like a working “family”. If there are problems need to be addressed in a team, it is best to address the issues as soon as possible in humane manner. Improvement

such as processes, resources availability, better approach to deal with a issues are to be discussed during Sprint Retrospective.

Summary

You might notice throughout the book certain concept is being repeatedly mentioned. This is done on purpose to introduce the concept through different perspective. For example, the 3 questions to be answered by The Team during daily standup meeting. This book has discussed it through Scrum Master's perspective and also The Team's perspective because they have different responsibility during the session.

Running a successful Scrum team is not Scrum Master's sole responsibility. Product Owner, Scrum Master and The Team must also play an active role to enforce Scrum prescriptive guidelines. This is why, this book is designed to discuss each Scrum ceremonies and prescriptive guidelines from different perspective.

Congratulation for making this far. This is the end of the book. If you have any feedback on the book, or wish to discuss specific Scrum challenge you are facing, please contact me **here**. I hope you have gained new and useful insight on how to run a better Scrum team.

About The Author

Daniel Foo has been building software for 10 years. Throughout his career, he has played the role of programmer, tester, SEO specialist, product analyst, project manager, team lead, sales engineer & product support.

He started his software development by programming a Library Book Searching application for fun, which leads him to his first programming job in his university R&D Department. In his years of software development career, he has written codes in PHP, Java, Objective-C, and his favourite programming language is C#.NET. He has built desktop applications for Windows, iMac, Android and countless web applications.

He has special interest in database because he believes database is the heart of software application. He has acquired his MCSE (Microsoft Certified Solution Expert) in Business Intelligence. Apart from SQL Server, he is familiar with MySQL, Oracle and SQLite.

He also has a special interest in software development processes. He is a CSM (Certified Scrum Master). Compared to other software development frameworks such as RUP and Kanban, he believes Scrum is a simple yet well-thought framework for most modern software development.