1

No single Product Owner for one team

I see this phenomenon happening quite often in organisations that implemented SAFe (Scaling Agile Framework) in the past, dropped and returned to simple Scrum. In the SAFe framework, you have two roles: Product Manager and Product Owner. The Product Manager is the one thinking more in a strategically way; the Product Owner is the one thinking more about the daily activities. When companies go back to Scrum there is no more two roles, there is only Product Owner, but companies do not want to fire anyone, so they assign two product owners to the team, even if they still use the same mindset: assigning one person to the strategically part and another to the daily activities I see this as a failure. I believe it’s tough to work with this scenario; you must have one single person responsible for the whole product, like a friend of mine used to say “one single neck”.

Product Owner Anti Patterns by Luis Goncalves / Card-Deck Layout by @UdoWiegaertner / V 0.1

2

Interference with the daily work

I saw this happening in several organisations but mainly because of two significant causes. First, cause the Product Owner in his past was a typical project manager where he was chasing people to get a job done. Second reason relies on the fact the Scrum Master is not active, and the Product Owner must step into and do a lot of the activities of the Scrum Master, taking the role to serious ending up controlling and interfering more than helping. In this situation is normal to see Product Owner taking a quite active role on the dailies, manipulating the normal process. Another problem that I see quite often relies on the fact of Product Owners track what developers are doing. I saw several examples of Product Owner talking directly with developers asking what they are doing because they are not assigned to any task. Now imagine how developers felt when suffering this kind of control.

Product Owner Anti Patterns by Luis Goncalves / Card-Deck Layout by @UdoWiegaertner / V 0.1

3

Stories built around product layers

This is a tricky topic. Over the last months, I have been working with teams that cannot deliver as fast as they could if they would use other approaches. They made a quite typical mistake; they have user stories that are built around layers of the product. As an outcome user stories are dependent on each others causing a lot of delays to deliver a product that can be usable by the customer.

Product Owner Anti Patterns by Luis Goncalves / Card-Deck Layout by @UdoWiegaertner / V 0.1

4

No projection of the product backlog

Usually, the Scrum Master produces burn down charts showing how the sprint went. This is quite common, what is not so common is the Release/Program burn down or burn up. This is a quite common problem in many of the organisations where I worked for. Not having this kind of information make the life of Product Owner harder because it gets harder to know if the team is on track or not. With this kind of information available, the Product Owner can take better decisions about the scope to deliver the product on time, bringing the biggest amount of value to the customer.

Product Owner Anti Patterns by Luis Goncalves / Card-Deck Layout by @UdoWiegaertner / V 0.1

5

Backlog is filled with a bunch of “Strings”, not proper user stories

This problem is by far one of the most commons that I observed during the last years. Most of the times the product owner does not write the stories in a proper format, he just writes few key words, and that´s it. The problem with this approach relies on the fact that everyone outside of Product Owner head will not understand what the user story is about. And to be honest, I saw some examples where not even the product owner knew what he wanted to solve with some of the user stories that he wrote.

Product Owner Anti Patterns by Luis Goncalves / Card-Deck Layout by @UdoWiegaertner / V 0.1

6

The Product Owner is underpowered and does not act as the guard of the backlog

I saw this situation happening in some of the companies where I worked. Unfortunately, companies do not realise how dangerous this is. In these companies, the product owner did not have the power to decide anything and managers could change the backlog as much as they wanted. This situation leads the Product Owners to no longer care about the product. They started to not care with the ROI, and they completely lost all their entrepreneurial behaviour. Usually, this happens with traditional management, where they want to control everything that is part of the Product Backlog, not giving any freedom or trust to the Product Owner.

Product Owner Anti Patterns by Luis Goncalves / Card-Deck Layout by @UdoWiegaertner / V 0.1

Solution

In general, a good user story should be written covering all the layers of the product. Should be an end to end user story. The solution for this problem is the coach the Product Owner and the team about this. For me works quite well when I explain to developers that do not matter if we deliver 200 back end user stories when in reality the customer cannot use them without the front end part. Below there is a picture from Ángel Medinilla that shows in a great way how stories should be built.

Solution

I believe the solution for this case is to find a strong and experienced Scrum Master. An experienced Scrum Master understands quite well both roles (Scrum Master and Product Owner) and can coach the Product Owner to allow the team to take responsibility for their tasks. One of the roles of Scrum Masters is to protect the team from intrusions, having a strong and experience strong master within the team will allow the team to remain isolated and protected from unwanted interferences, and at the same time will allow Product Owners to be coached on right behaviours.

Solution

Companies must understand that is not possible to have this scenario and they must provide an environment where each team has one single product owner.

Solution

My experience tells me this situation is quite difficult to resolve because most of the time this problem comes from managers above the Product Owner. In this case, the Scrum Master or the Agile Coach has a fundamental role to play. He is responsible for talking with the management (or with anyone else who is behaving in this way) and explaining how this behaviour has an impact on the company and the quality of the product. I have witnessed several situations where the Agile Coach was able to explain to Senior Management what was happening and was able to help management to change their behaviour.

Solution

To point out to the solution, I am using a blog post from Roman Pichler. "10 Tips to write good user stories".

Solution

The solution here is quite easy. The Product Owner should build a product burn down/burn up exactly as the Scrum Master does, but instead of being focused on the sprint, the Product Owner focuses on the release. A possible example can be observed below.

7

The Product Owner that uses estimates to set deadline demos for clients

Here we have a typical problem that is common in people who do not understand the basics of software development. This is much deeper than Agile software development. The people who do this do not understand how software development works, so their estimates are just an approximate calculation; they do not guarantee that everything will be done by a certain date.

Product Owner Anti Patterns by Luis Goncalves / Card-Deck Layout by @UdoWiegaertner / V 0.1

8

The Product Owner is too busy for the team; he has too many teams

I see this problem too often, Product Owners who are completely overloaded with work: the overworked Product Owner.

Product Owner Anti Patterns by Luis Goncalves / Card-Deck Layout by @UdoWiegaertner / V 0.1

9

The Product Owner is too busy for the team. He spends a lot of time in meetings with clients

This case is a bit different than the previous one, but the result is the same (for the team). In both cases, the team does not have the support of their Product Owner, making the task of developing the right software difficult.

Product Owner Anti Patterns by Luis Goncalves / Card-Deck Layout by @UdoWiegaertner / V 0.1

10

The Product Owner Proxy

This person acts as a placeholder for the actual product owner. This can lead to delayed decisions or even conflicts. The author of Agile Product Management with Scrum, Roman Pichler, refers to a case where the PO was too busy to take the role. Therefore a company got a proxy PO. In the end, there were various conflicts because the proxy PO was not empowered to make decisions and all decisions were supposed to be made by the official PO who was never available.

Product Owner Anti Patterns by Luis Goncalves / Card-Deck Layout by @UdoWiegaertner / V 0.1

11

Product owner community

This happens to companies that create a committee to be responsible for a product. When there is no one person responsible for the product, it can lead to delayed decisions. Often, most of the people involved want to reach an agreement between all parties, and this will lead to lengthy meetings without any proper outcome. This is a typical case of “too many chefs in the same kitchen”. This is quite similar to one of the problems that I raised above.

Product Owner Anti Patterns by Luis Goncalves / Card-Deck Layout by @UdoWiegaertner / V 0.1

12

The Product Owner sacrifices the quality of the product to ship the product faster

I see this happening quite often, especially when a Product Owner is under external pressure. Stakeholders push Product Owners to release the product faster, even if the quality is not the best. Of course, this approach will damage the company over the long term; everyone knows that poor quality software will sooner or later create problems and damage the provider’s reputation.

Product Owner Anti Patterns by Luis Goncalves / Card-Deck Layout by @UdoWiegaertner / V 0.1

Solution

The job of a Product Owner is not simple; it´s a tremendous and complex job, sometimes the product is so complex that one single person might not be able to manage everything. I've seen some companies that brought business analysts to the team; these people took the responsibility of talking with clients to see what the market needs, leaving the responsibility of the technical product to the actual Product Owner. I saw this technique in several companies, and it worked quite nicely.

Solution

A Product Owner should not have more than two teams. Of course, this is a general rule, and it might be possible to have more, but if he/she has more than that, it can be difficult to give full support to the teams.

Solution

People approach me telling me that sometimes we do not choose the release dates and we have deadlines to be met. Therefore, there is no way to behave in a different manner in this kind of setup; the customers just want to have the product. I worked for a mobile manufacturing company, so I understand what dates set by the market means. Every year we had Christmas sales where the phones needed to be out; there was no other solution, so how could we manage this? The answer is simple: follow basic agile principles. The PO must educate the client that after every sprint the team has got something to show (with this the PO gets his demos). During the sprint demo/review, the client has the opportunity to provide feedback on what was delivered. And more importantly, has the opportunity to select the most important things that are on the backlog, enabling the team to always deliver the most value to the client. At the same time (and this is one of the most frequently forgotten practices), the PO checks the overall release status using Releases Burn Downs. This practice allows the PO to know exactly where the team will land, and discuss with the client about what can be dropped in the event the team is not able to deliver on time. If the client is not open to all these suggestions, the company should start to reconsider if the client is someone with whom the company would like to work in the future. The problems that will arise from future interactions will be so complex that the relationship will not benefit any of the parties.

Solution

This problem is quite serious and, in my opinion, is something that is part of the company´s culture (not caring about quality). The method I found most useful to reduce this problem is the presence of a Definition of Done Checklist. A Definition of Done Checklist is a simple list of tasks defined by the team that represents all the activities that must be done before a story goes into production. Without a Definition of Done Checklist, it´s easier for the Product Owner to push the team to cut corners; if this list is there, on the other hand, it´s possible for developers to show that a story is not done until all tasks that are part of the Definition Of Done are completed. My experience tells me this helps to reduce the problem. If you are interested to know how can you Mature Your Definition Of Done check this blog.

Solution

Each product should have only one Product Owner. My experience tells me if we have too many people responsible for a product we have none. Never assign more than one Product Owner to a product. If you have several products that are connected with each other, you might need to have a program. In that case, you should get a Program Manager or a Chief Product Owner as some companies call it, but never more than one person responsible for a product.

Solution

Each case is different, but as a general solution, I believe that we could try to use the same approach as in the first example. Having a business analyst within the team could help the Product Owner to get more time to help the team to deliver the right product.

13

Confusing the Product Owner role with the Scrum Master Role

I saw this happen many times in companies where Scrum Masters were passive for a week or two. In this situation, Product Owners lack the necessary support from the Scrum Master, and they start to take care of all the activities inside of the team. At some point, the Product Owner no longer knows what his role is, and what the Scrum Master role is. This causes a lot of confusion and, worst of all, conflicts and mistrust between the Product Owner and Scrum Master.

Product Owner Anti Patterns by Luis Goncalves / Card-Deck Layout by @UdoWiegaertner / V 0.1

14

The Product Owner who expects a stretched commitment because "based on their experience it shouldn't take that long."

It is quite common to see this behaviour from old school project managers, where commitments were almost forced by project managers or stakeholders. Another possible cause lies in the fact that Product Owners are ex-tech leads or architects. In this situation, the team is dealing with someone who has great technical skills and who expects the team to solve the issues in a faster way.

Product Owner Anti Patterns by Luis Goncalves / Card-Deck Layout by @UdoWiegaertner / V 0.1

15

Being Product Owner because they're a department lead/ manager.

In some companies, the Product Owner role has some semblance of power, and many people will try to grab this position even if they do not have any interest in the product itself. They use this position as a platform to increase their political power. As you can imagine, most of the time the outcome of this is poor and lacking in quality, driving everyone to frustration.

Product Owner Anti Patterns by Luis Goncalves / Card-Deck Layout by @UdoWiegaertner / V 0.1

16

The PO who feels it is not their role to maintain the backlog or delegates all of the actual User Story writing to team and SM

I did not see this happening too often (fortunately) during my career, but it does happen. This type of Product Owner thinks his or her job is just to dump ideas into the backlog and that someone else should be doing the boring job of transforming the ideas into stories. This kind of Product Owner thinks that his job is too important to be spent on secondary tasks.

Product Owner Anti Patterns by Luis Goncalves / Card-Deck Layout by @UdoWiegaertner / V 0.1

17

The Product Owner who thinks a Scrum Master is just a project manager whose role is to pester people into doing their job

Sadly, there are still a lot of people do not understand anything about human behaviour. There are a lot of people who still think that punishments and pressure work quite nicely to motivate people. A lot of these Product Owners still believe that people are lazy and need project managers to keep them in line, and of course, the Scrum Master fits perfectly into this job (according to their point of view).

Product Owner Anti Patterns by Luis Goncalves / Card-Deck Layout by @UdoWiegaertner / V 0.1

18

The Product Owner who refuses to choose between five top priorities

I see this quite often; the Product Owner calls the most important story a Must, then there appears another story he says is more important, they become a Must-Must story, and so on. This clearly shows that the Product Owner does not understand what the single most important thing that must be achieved during the next sprint is. Having the teams working on several important issues is a dangerous factor. Not too long ago, some of my teams were working on different goals to be achieved in a single sprint. The result was poor, there was no focus, and while everyone worked on everything, the teams constantly failed to achieve their goals.

Product Owner Anti Patterns by Luis Goncalves / Card-Deck Layout by @UdoWiegaertner / V 0.1

Solution

In this case, I truly believe there is not much that I can suggest; management must understand that a Product Owner is extremely important for the company and only people who understand customers and the market can do a good job. Assigning people to these positions who do not understand the market or the customers is a recipe for disaster. I can only suggest that management assigns people based on quality and not based on political interest.

Solution

In both situations the Scrum Master's job is quite important; he must protect the team. He must explain to the Product Owner that the team owns their estimates. They are the ones deciding how much it takes to deliver that amount of work and there is not much he can do. The Scrum Master must explain to the Product Owner that the team's members are doing their best and he needs to trust them.

Solution

The solution here is quite simple, but not easy to implement at all. In this case, it is extremely important to get an experienced and strong Scrum Master. Great Scrum Masters can do their jobs in a well, but even more importantly, they can coach others in their jobs. Great Scrum Masters are great coaches as well, so having a strong and competent Scrum Master in this situation is fundamental to improving these issues.

Solution

Here it is quite important for the Product Owner to understand what the most important things are that must be delivered to the customers. Product Owners must understand the team cannot deliver everything at the same time, so they really must understand what is the top priority that should be achieved during the next sprint. When we applied this rule in some of my teams, teams got extremely focused, and they started to deliver much more stories per sprint. But more important than that, they started to deliver fully implemented features that could be shipped to the customer. Before, the teams were delivering a lot of things, but none of their features was ready, blocking the release to the customers.

Solution

The solution here is not easy to achieve at all. I truly believe this has more to do with mindset than with any agile process. I would suggest taking this Product Owner to a good agile training session to make sure that he understands what the roles are in Agile software development. To complement this, I would take him to a good training session about intrinsic motivators to help him understand what drives people to do a great job. If he would not like to attend any of these training sessions, maybe give him the book Drive by Daniel Pink. Hopefully, he will then understand that punishment or pressure will never motivate people. Hopefully, he will then understand that punishment or pressure will never motivate people.

Solution

I believe the Scrum Master and the team itself have an important role here. Together they must explain to the Product Owner that the aforementioned is his job. They must explain that without a proper backlog the team cannot do their jobs. The stories should be discussed between the Product Owner and the team. The Product Owner is the person ultimately responsible for keeping the product backlog in good shape. The Scrum Master and the team must show to the Product Owner that it is extremely difficult to do any work if the product backlog is not maintained. A good story must be clear, optimally written in the INVEST (Independent, Negotiable, Valuable, Estimable, Scalable and Testable) format with proper Acceptance Criteria.

19

The Product Owner who insists on architecting the solution design

I mentioned above that sometimes the Product Owner is an ex-architect or an ex-tech leader. As a result, it’s normal to have this kind of anti-patterns. The Product Owner wants to be responsible for the architecture of the product, driving the team crazy.

Product Owner Anti Patterns by Luis Goncalves / Card-Deck Layout by @UdoWiegaertner / V 0.1

20

The Product Owner who does not regard himself as a team member

I do not see this Product Owner Antipattern too often, but several people have mentioned it to me. Therefore I am adding it to my list. As the name suggests, the Product Owner does not feel himself part of the team. He feels that he should only be responsible for delivering the stories and nothing more. As you can imagine, the cooperation between the team members and Product Owners will not be the best, and at some point, the collaboration will become extremely hard.

Product Owner Anti Patterns by Luis Goncalves / Card-Deck Layout by @UdoWiegaertner / V 0.1

Solution

The Scrum Guide says: “The Scrum Team consists of a Product Owner, the Development Team, and a Scrum Master.” Just because the Scrum Guide says that, of course, there is no guarantee the Product Owner will consider himself part of the team—But, I think this could be a good start nevertheless. The Scrum Master or the team could explain to him that, according to the manual, the Product Owner is officially part of the team. On top of this argument, the team and Scrum Master could explain to the Product Owner they are not able to do a fantastic job unless the Product Owner starts to become more connected to them. This could be a nice topic for one of the Retrospectives.

Solution

The Scrum Master and the team must explain to the Product Owner that the design of architecture is not his responsibility. I always like to give this explanation: The Scrum Master and the team are responsible for “How”: How the product is made. The Product Owner is responsible for the “What” part, meaning that the Product Owner should be the one saying what is supposed to be built.