



Agile Mojo Booster

Product Owner Handbook



SilverStripe
Create the web





So you're going to be a Product Owner?

Congratulations, a Product Owner plays a critical role in shaping business success. If this is the first time you have been in this role you probably have a few questions, some of which we hope to answer in this manual.

In Agile projects, and particularly in Scrum projects, there are three roles:

The Team

Traditionally 'The Team' consists of developers and testers, but it can also extend to designers, business analysts, content writers — anyone who contributes work that progresses the project.

The Scrum Master

This is who makes sure everyone is free to work to the best of their ability. They are often referred to as a 'Servant Leader', someone who is there to serve the team, but also acts in a leadership role for the team by facilitating and reflecting back the team's processes.

The Product Owner — You!

A Product Owner is one of the most critical roles on an Agile project. Your role is to represent the user and your business. You need to ensure that the team deliver a valuable product to your users that fulfil their genuine needs and will make their day-to-day life easier. No pressure!

Responsibilities

Share the vision

Not only is it important for you to understand the overall vision of the project, you must be able to translate that to the team.

Liaise with stakeholders

Behind every good Product Owner is a number of stakeholders who need to be kept informed throughout a project.

Create user stories

The Product Owner works with whoever they need (stakeholders, the team, customers etc) to describe the outcome that represents benefits to the end user.

Prioritise user stories

These stories need to be prioritised in terms of the value that they give to the customer so the team know they are working on the most important thing every time they pick something off the top of the backlog.

Maintain the product backlog

The backlog is your product scope. It needs to be continually reviewed and refined throughout the project lifespan.

Protect the team

The Product Owner is a single point of contact for the team, and always knows who the team needs to speak to if they need answers. You help enable the team to not be distracted by other

activities that are not specifically related to the current sprint deliverables by directing key stakeholders input through them.

Accept, reject and clarify

When the team delivers a story to the Product Owner it is up to you to check that the acceptance criteria have been met and the story's intent has been satisfied. You will also need to answer any questions that the team has about acceptance criteria as they work on the story.

Trust the team

The team are experts in their field, give them the space to do the work they need to do.



Three traits of a great Product Owner



Is available

You need the time to dedicate yourself to the project so the team has access to you to ensure the project can keep progressing. We would recommend at least 50% or more of your time to be dedicated to the project.



Has authority

You need to be able to make decisions without always having to consult a committee, otherwise each slow decision will cause the team to grind to a halt.



Has a vision and can communicate it

If you can't communicate the vision, and don't have a true understanding of it, how can the team know what they are trying to achieve?

What not to do

Present solutions to the team

The team need you to communicate the intent of each story, and it's up to them to figure out the solution that best fits the need. You give them the 'why', they give you the 'how'.

Add scope mid-sprint

Once a team has committed to a certain amount of work for a sprint, you put that commitment at risk by adding more work. Sprints are short for a reason; can your new priority wait until the next one?

Allocate tasks to team members

Nobody is a micro-manager on a Scrum project, the team is a self-organising group and chooses what tasks they work on, based on the sprint priorities.

Pressure the team to take on more work

The team are the ones who do the work and have a feel for what they can achieve each sprint. If you pressure them to take on more than they think they can realistically achieve, people will get burned out and quality will start to slip.

Tell the team what you want, then leave

The worst thing a team can hear is "here are the requirements, see you at the end of the sprint". They need access to you each day throughout the sprint to get questions answered and to make sure they are heading in the right direction.

Some common questions

Is it a Project Management role?

The Product Owner has a much more collaborative role in an Agile team than a traditional Project Manager may have. They work with stakeholders, customers and the team to find out the best way to achieve their vision, and to enable the best outcome for the user.

The traditional Project Manager responsibilities are shared across the team, Scrum Master and Product Owner roles.

Is it a single person role?

Yes, it is a single person role. The team needs that one point of contact so they always know who to go to when they need answers. That's not to say the Product Owner is alone though, they often enlist the support of a mini-team to help create the product backlog. In these cases, the Product Owner can work closely with a



Business Analyst, a User Experience Specialist as well as a Senior Developer or Architect to breakdown large features into smaller deliverable stories.

They may also have an internal Project Manager providing support with project governance, budget, other project interdependencies and delivery timeline. But ultimately, one person needs to be the decision maker — you!

Can you have a ‘proxy’ Product Owner?

A proxy Product Owner is useful if the actual Product Owner is time-poor or away, but they still need to have a full understanding of the vision, the time to be available to the team and the ability and remit to make decisions quickly.

How much time am I expected to spend on the project?

You are the key decision maker throughout the project and therefore need to work with the team on a daily basis.

You will be expected to attend:

- Sprint planning (once a sprint)
- Sprint review and retrospective (once a sprint)
- Demos (as required throughout the sprint as each story is completed)
- Daily stand ups (up to 15 minutes every day — optional)
- Backlog refinement (once a sprint)

The more engaged you are throughout the sprint, the better the end result will be. The success of a project is usually proportionate to the time and interest the Product Owner has put in.

What you need to start a project

A business objective and an agreed budget is all you need to start a project. Don't worry about preparing all the details up front — we'll gather them together as we go.

A great exercise to kick off your project might be story mapping — a form of a workshop with the whole team, where together you discuss what features you have in mind, how they logically fit together, and what user stories might be created for the backlog in order to achieve your objectives.

User stories

User stories tell the team what to build, and also why to build it, which helps them find the best solution for a given problem. They are written from the perspective of a user and follow a set template:

As a <type of user>, I want <goal> so that I <benefit>

They also help build shared understanding. It's very easy to misunderstand a written document, but user stories are like holiday photos — they are quick to absorb, but necessitate conversation to give them context. Acceptance criteria are also added to help clarify and limit the scope of each story — they tell developers when they have done enough to achieve the outcome.

Writing user stories can be difficult to master, but if you follow the 'INVEST' model, it can make your life a little easier:

- “I” ndependent (the stories should be independent of each other)
- “N” egotiable (it's a conversation, not a contract)
- “V” aluable (each story creates some value for the customer)
- “E” stimatable (if you can't estimate it, there are too many unknowns)
- “S” mall (the stories need to fit into a sprint)
- “T” estable (if you can't test it, how do you know when it's finished?)

How do I prioritise my stories?

One useful technique might be to always imagine what you would want to build if the next sprint was your last. The team should always be working on the most important thing.



Understanding the basic Agile/Lean concepts

At SilverStripe we believe that, although it is no silver bullet that will solve all your problems, Agile builds on so many pragmatic elements that it has a much higher chance of setting your project up to be a success.

Most Agile and Lean methodologies will be based on the following principles:

Creating customer value

The Product Owner role is essential when it comes to identifying and prioritising the backlog stories to ensure the team is always working on the items that create the most value for the customer.

Removing waste

Anything that is not directly adding customer value is waste. Some of the most typical types of waste in software development building are work-in-progress (software that is developed but not yet available to the customer), unused (or little used) features and functionality, defects and delays.

Frequent delivery

When we build software in an iterative and incremental manner it allows us to get quick feedback from users and validate that we are building the right things. In SilverStripe, where Scrum is the most commonly used Agile methodology, our sprints are typically 2 weeks,

after which we should have some working software in a production-like environment.

Pull-based system

Using a pull-based system gives a number of advantages compared to a push-based system. In a supermarket you don't try to cram more stock onto the shelves than there is room for. You wait until stock is sold, and only then replace it. In Scrum we pull items off the backlog once we complete a higher priority item.

Building in quality

At SilverStripe, we strongly feel that building in, or 'baking in', the quality means a better outcome overall. Baking in quality means building and testing features to the best of our ability throughout the sprint cycle. This means, as a Product Owner, the urge to put as many features as possible into your sprints can be detrimental. Trust the team when they say 'we need this time to make sure these things are done right'. A good ratio to aim for is 60% new features, 20% bug fixing & polish, and 20% refactoring & architectural changes to enable the next sprint or to bed in the new features.

Continuous learning and improvements

'Inspect and Adapt' is an important element of Agile practices. Retrospectives, where the team reflect over their recent work, are a mandatory part of most Agile methodologies. They allow the learnings from one sprint to be immediately incorporated into the next!

Defer commitment

Most of us are usually trying to take decisions as early possible to give us most time to carry them out (e.g. when we sign off a big requirements specification before development starts). In Agile, we

actually try the opposite by deferring commitment. The later we can wait making a commitment, the more we have learned and the less likely it is that we will have to reconsider. This is why we don't refine the whole product backlog and why we only commit to a limited set of stories per sprint.

Transparency and trust

Transparency is required for building trust between the team and you, the Product Owner.

In fixed price/fixed scope contracts (typically found in Waterfall type projects) there is a built-in conflict between customer and vendor. The risk is on the vendor, and this is often managed by resisting or charging for change. In an Agile project it is more common to have fixed price/variable scope. The client and vendor now have the same goal of creating the most customer value within the given budget, and are free to make changes where they are of benefit.

Respect for people

In Agile we believe in creating the right environment for a project to succeed. If we work collaboratively towards the same goal the project will have a high chance of success.



What does success look like

Features/Time/Cost — what is your Iron Triangle?

Generally speaking, you will always have an ultimate driver behind your project or deliverable. Maybe the most important thing is to build all of the features and no one cares how long it takes or how much it costs.

You might have a tight or fixed deadline, so it's more important to do things in priority order and release on time above all else. A fixed budget means you can 'buy' so much time, and what you achieve for that is what you prioritise based on value.

CLASSIC APPROACH

AGILE APPROACH

fixed



variable

For example, fixed timeframe and fixed features means no room to move. You 'have' to build what was promised, and it 'must' be done by that date at the expense of quality.

The diagram opposite outlines a typical approach for both a traditional project approach, versus an Agile view of managing a project.

Defining your features in priority order means that the high value items are achieved first and foremost, and some more 'nice to have' items may or may not be delivered in the timeframe. As a Product Owner, you want to be able to come up with new valuable ideas as things progress and have the ability to add these into your priority without compromising the whole deliverable. Once you understand your constraints, you can then begin working on how to best deliver the outcomes.

Defining your success — what is the business driver?

Being clear on what the business driver is will help define the goal of the project or programme of work.

Features and benefits — how do you know what will be the most important thing for success?

But how do you know what will add the most value? Comparing the features and assessing what the benefit to the user will be, will help really identify value and therefore how much effort you would be prepared to use to achieve the outcome. If a feature benefit means spending \$10,000 dollars, how much value would it need to add? 10-fold return? 100-fold return? It will always be relative to your overall project, as \$10,000 may be a little or a lot. The main objective is to analyse whether a feature is worth moving forward with. You often get passionate people who can really sell their idea across, although it is based on gut feel or something shiny and new rather than actual return on investment.





Cost of pixel perfection

As a Product Owner, you will need to make some hard calls. Many of those come down to what level of perfection you need to achieve. You may 'want' to achieve this, but it does come at some cost to you. If that is the most important thing, then it is well worth the time and money, but it needs to be acknowledged and managed so you get the best value for money.

For each line of code written, there will be unit tests, peer review, manual testing, regression scripts, maintenance and depreciation over time. This is good to remember when considering how many browsers and devices to test on your project, and to what level you will worry about fixing minor things.

Sometimes, the drive to get 'exactly what was bought and paid for' can influence this, leaving no room for working on the most important and valuable items, and having to spend the same amount of effort on ensuring pixel perfection on a device that has 5% of users. Pragmatically speaking, no site goes live without some minor issues still remaining, as the cost of delay in getting the high value features in front of people is not tenable.

It is crucial to remember that as Product Owner, you need to understand what is most important; usability, visual look and feel, feature depth or a range of features, or how many devices/ browsers they can be used on. For each, they come with an associated effort and cost, which you will need to balance.

Ways to break it down

Time spent polishing less important things can mean missing out on more essential functionality in other features.

Good, better or best

Not all scope of a feature is equal, or needs the same amount of priority or attention. When writing user stories, it is very useful for the team to understand what types of things you are expecting. It can seem quite obvious, or implied, but just so everyone understands what is ACTUALLY required, it is best to put as much detail as possible. In saying this, try not to specify exactly how and where, as these types of decisions are made as part of building the features.

Separating stories into essential items, and then creating other stories for less important functionality that could come in a later sprint means you will be getting the most important thing delivered at all times.

Example

The page opposite shows three separate stories for an image/video 'lightbox' presentation feature.

The return on investment for this feature may be worth the 'Best' scenario to your business, so you can prioritise the entire scope. If only the 'Good' scenario gets you to the outcome, then prioritise only that story, and add the other two lower down in the backlog in relation to the other stories.

Good (most important)

Acceptance Criteria:

- I can login to the CMS as an admin
 - I can click on an area in the CMS that shows the admin of the lightbox area
 - I can load X number of images in order of presentation in the lightbox
 - I can load image files only
-

Better (ideal if we can do this as well)

Acceptance Criteria:

- I can load videos
 - I can load JPEGs, PNGs, TIFFs, Youtube and Vimeo links
-

Best (nice to have if we get time)

Acceptance Criteria:

- I can see the image size limit in text where I load the images
- I get an error if the size is too big
- I can upload two different videos or images and based on which geo-location they are in, show the user one or the other

Working with the Team

Understanding the importance of cross-functional teams

Increasing the cross-functionality (mixing skill sets and knowledge areas) across team members helps to create awesome teams that are great at working together to rapidly deliver your product.

If teams are not cross-functional then:

- The team is less resilient to unplanned leave or unplanned changes
- Work must often pass between a number of individuals before it can be completed
- You are more likely to have bottlenecks that prevent work from being completed
- Roadblocks are harder to resolve — other team members are less able to help
- You can't foster teamwork, in extreme cases you only have a team of individuals

Creating the right environment for team performance

Scrum teams work best when they are able to focus on a defined set of deliverables for a given period.

As Product Owner, you can help the team perform at their best by:

- Ensuring the stories are well thought through
- Not changing priorities mid-sprint, and being clear on priorities going into a sprint planning session
- Not over-loading the sprint

Refactoring is not a dirty word

Refactoring — altering the way that existing features have been coded to pave the way for future changes — within a sprint cycle is a good thing. No matter what methodology is used, no developer can predict what future requirements you will have, or whether what they are building now will be appropriate in an ever-changing web or technology environment. When a development team adds refactoring of technical debt stories into your backlog, it is them caring about future-proofing your product for as long as possible. Encouraging this behaviour means a much better long-term outcome.



The end of a project

What happens if I still have lots in my backlog?

As Product Owner, you've had the hard task of prioritising the backlog for your team. Without your prioritisation, some really important features may be missed. You need to ensure your team are able to concentrate on the critical pieces of work.

If items in your backlog are non-negotiable from a business point-of-view, discuss this with your Account Manager and a Scrum Master as early as possible. Remember, though, support teams work differently to Scrum teams and the details of this should be discussed with your Account Manager.

Warranty periods

As a Product Owner, you should be aware of any warranty periods that might be included in your Statement of Work. In general, this kind of warranty will cover previously undiscovered P1 and P2 issues/bugs for a set amount of time. This warranty period will generally be separate to any kind of ongoing maintenance agreement. You should also ensure you are aware of who is looking after warranty type issues for you — your project team or a support team — so you know who to contact.

Once my project is complete, what do I need to think about within my organisation?

As Product Owner, you may or may not be responsible for your site after it goes live. You should consider what introductions need to be made in terms of who will need to communicate with the agency during the support phase. These people may need to be added to helpdesk systems or similar and informed on the ins and outs of the support process.

How is the support phase different from the project?

It's important to remember that once a project is finished, you no longer have a completely dedicated development team. While these support teams are always there to help, a Product Owner should be aware of the differences and limitations involved with this new phase.

Meeting your new team should be a priority for your organisation. This handover is important to establish the new relationship and well as cover the mutual expectations during support.







Summary

It's not easy being a Product Owner, but it's very rewarding and can be lots of fun. Having a team you can trust and be part of is key. At SilverStripe, we value our Product Owner's input, and go to great lengths to help you do your best work for your company. Your Scrum Master will work with you as a coach, to guide you along the journey. There will be ups and downs, but knowing you are in good hands and you can ask for help at any time will mean someone always has your back. Please contact us with any questions you may have at solutions@silverstripe.com.

Good Luck!

The SilverStripe Solutions Team

SilverStripe values



Honesty over comfort

We believe open communication creates better results.



Adaptability over guarantees

We need to be open to change if new information reveals better solutions.



Collaboration over control

We prefer working in teams and inviting feedback, even if this means we have less control.



Continuous improvement over perfection

We don't expect to be perfect. We release early, often and learn from mistakes quickly.



Camaraderie over corporate culture

We want to know each other as real people not "colleagues".



Level 5, 101 - 103 Courtenay Place, Wellington 6011, New Zealand

PH +64 4 978 7330

www.silverstripe.com