

# Welcome to Quest for Value!

Quest for Value is a tabletop-inspired Scrum game, that involves participants going on an imaginative journey into the world of project work. Throughout the game, your team will possess different roles, with different abilities and responsibilities, hoping to complete the best project you can dream up!

Quest for Value is designed to be played with 3-6 participants, and takes around 2 hours, with setup and introduction included.

The following pages include the rules for the game, as well as the individual rules for the three Scrum roles; The Scrum Master, the Product Owner, and the Developers.

If you have questions about the game, feel free to contact the creator at [magc@itu.dk](mailto:magc@itu.dk).

## If you're new to tabletop games...

Tabletop games are board games wherein players explore an imaginative world, described by the Facilitator (sometimes called Dungeon Master in traditional tabletop games), where the rules merely exist to help them determine the success of their adventures, instead of defining what they are able to do.

Tabletop games do not have turns or winners. Instead, players play together to overcome challenges, and achieve a common purpose, through any means they can imagine!

Do you want to slay the dragon? Run away from it? Sneak in and steal its treasure? Make friends with it?

Only your imagination is the limit!

You can watch a short video on Quest for Value in action here, if you want an example:

[https://youtu.be/\\_sLlxWVgoBk](https://youtu.be/_sLlxWVgoBk)

## If you're new to Scrum...

Scrum is a cooperative framework for developing, delivering, and sustaining complex products.

It involves participants taking on different roles to work together in iterations (sprints), completing their project in increments.

While Quest for Value is not made to test your knowledge of Scrum, having a basic understanding of how Scrum works, what sprints are, and the roles that participants take on, will be helpful.

You can read more about Scrum at:

<https://www.scrumguides.org/scrum-guide.html>



Introduction

Welcome to the Quest for Value! Your team will over 6 sprints attempt to create the most valuable product, relying on the roles and abilities of the members of your scrum team.

Your Team

Your team will consist of members across the three Scrum roles - the Product Owner, the Scrum Master, and several members of the development team. Each role will have different abilities and focuses, and it’s up to your team to find the best way to work together to produce the best product!

Sprints

Your project will run for 6 sprints. Each sprint will move you closer to finishing your product.

Every sprint your team will spend story points to complete tasks (see below). Each team member has 4 points per sprint.

When a sprint ends, a review, retrospective and planning phase begins.

Review, Retrospective, Planning

Between sprints, three phases will occur.

In the review phase, the Facilitator summarizes incremental progress

In the retrospective phase, you will talk about process, either in-game or off-game.

In the planning phase, you will plan a goal for the next upcoming sprint.

Review  
Retro  
Plan

Tasks

In order to progress your project, your team will discover tasks along the way. Tasks will naturally come up as your team attempts to do things. Your facilitator will let you know when you’re encountering a task, and your team will have to try to complete it successfully, in the best way you can think of!

In principle, any task can be undertaken by anyone, but the abilities of each role might help you complete them.

Completing a task costs story points, depending on the size of the task (see completing tasks).

Completing Tasks

Work doesn’t always go the way we would like it to! Whenever your team attempts to complete a task, whoever is completing the task must roll for task success.

When deciding to take on a task, your facilitator will identify the estimated size of the task, low (1), medium (2), or high (3). Should you decide to attempt to complete the task, your character will lose that amount of story points for the current sprint.

A task roll succeeds on a die roll of 4 or higher. If a task roll fails, a disaster occurs!

+1 can be added to the roll when using an appropriate skill (as given by your role).

Overworking

If a team member runs out of points to complete work, they can overwork themselves, by borrowing points from the next sprint.

A team member that is overworking gets -1 to die rolls, and will have fewer story points in the next sprint, equal to the amount spent on overworking.

An Example!

*John decides to try and implement a search feature on the web-site the team is building.*

*The facilitator tells John that this is a task, and that it has an estimate of 2 story points.*

*John spends 2 of his points for the Sprint, in order to work on the task.*

*John rolls a 2, but uses his “deliver value” ability, to add 1 to the roll, because John is working on exactly that - delivering value. He succeeds!*

Optional!

Plan Changes

During a sprint, your team might want to complete tasks that don’t align with your decided sprint goal. This is decided by the facilitator, when your team attempts to take on a task.

In order to complete these tasks, your team must first make a change of plan roll, before rolling for the task.

A change of plan roll costs the entire team 1 story point each (coordination is hard!), and is initially rolled with 6 dice, requiring any die to land on a 5 or 6 to succeed. If the roll succeeds, your team comes up with a new plan that fits the task. If the change of plans fails, you cannot complete the task.

But plans get harder to change as the project moves along! Your team rolls with 1 less die per iteration.



Introduction

Welcome to the Quest for Value, Scrum Master!

As the Scrum Master, it’s your job to guide the process of the team, ensuring teamwork and communication, while protecting the team from outside intervention.

Your Abilities

As a Scrum Master, your abilities involve helping your team to meet their goals, through well thought out and effective team work.

Your abilities have different costs, that you can spend your estimation points on every sprint.

It is not required (nor is it neccesarily always the best use of your time) to spend all of your points each Sprint.

Using an ability depends on the context. Abilities should not be seen as rules for what you can do, but as support for what you want to do. As a player, you can take any ation you deem useful in the given situation - however, if the action you take might be backed up by one of your abilities, you’re probably better off using it, reaping its benefits! This is simply done by declaring that you’re performing an action, followed by some statement like “I suggest we ..., using my ability Improve Process”.

You will likely find that the retrospective phase is where you have the most obvious role - don’t let this hold you back! The role of Scrum Master is one that has a little hand in everything, so feel free to interact and guide the others, whether you’re using an ability or not!

Rolling for Success

Some of your skills, or attempts at performing other tasks, might require a success roll, in order to succeed. This works exactly like completing tasks succesfully, requiring a roll of 3 or higher on a six-sided die.

Helper Dice

*The Scrum Master is all about supporting the process of the rest of the team. One of the ways in which they can do this, is through helper dice, gained from their “Improve Process!” ability. A helper die is a die that can be given to a team member, allowing them to use (and roll) it for any future roll they might encounter, adding the helper die to their normal roll. Only one helper die can be held by each team member at a given time.*

Scrum Master Example

*The Scrum master looks at their team and goes: “Team, it seems like some people are quicker than others to just grab at the tasks - maybe we should let everyone pick one task for the start of the sprint, before everyone does their work?” The team agrees, and after improving their process, the Scrum Master gives a helper die to John, which he can use for an upcoming roll.*

Your Role

Ability 1

Ability 2

Scrum Master



Supporting the Process  
Protecting the Team  
Shared Understanding

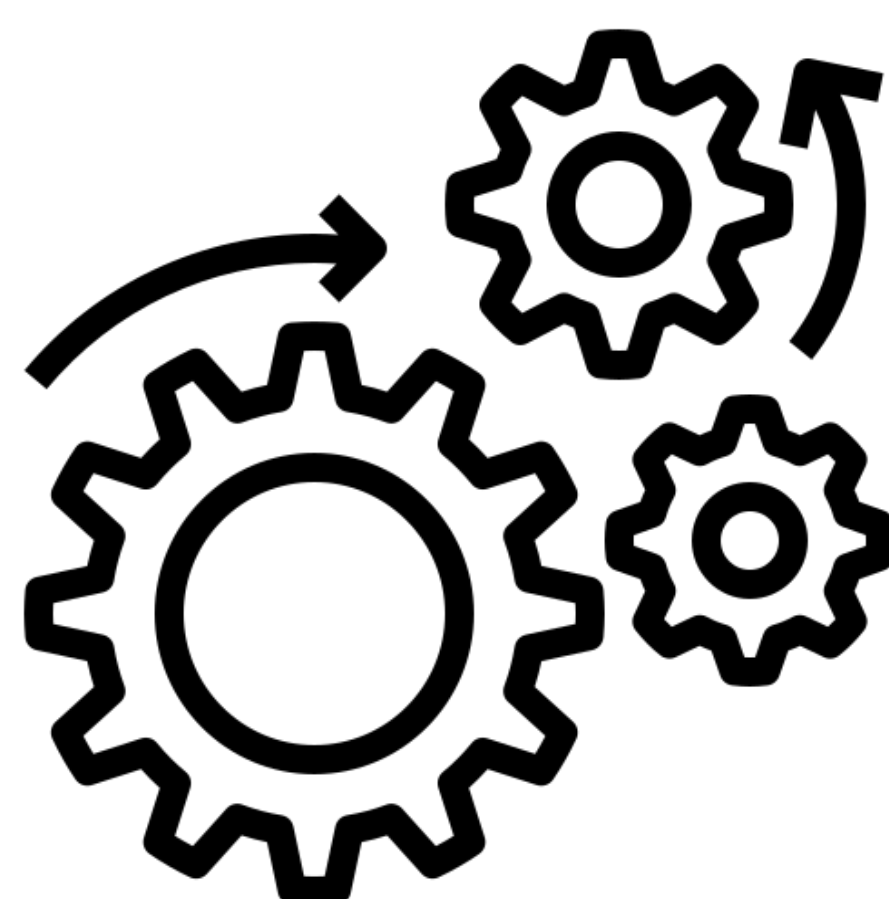
Mitigate Disaster!



The Scrum Master mitigates a disaster as it happens, lessening the impact!

2

Improve Process!



The Scrum Master helps improve a team process.  
Give a helper die to a team member on a succesful roll.

2



Introduction

Welcome to the Quest for Value, Product Owner!

As the product owner, it’s your job to maximize the value of the product resulting from the work of the Development Team.

Your Abilities

As a Product Owner, your abilities involve helping your team to produce the best possible product, delivering the highest amount of value to the client.

Your abilities have different costs, that you can spend your estimation points on every sprint.  
It is not required (nor is it neccesarilly always the best use of your time) to spend all of your points each Sprint.

As a product owner, one of your abilities has a varying cost (denoted by the X on the ability). You chose the value of X when using the ability, which will change the depth of the outcome.  
Your “Build Backlog” ability allows you to define critical work to be done by the development team, specifying a more structured way for them to work, than them discovering tasks as they work. It is recommended that you write tasks defined this way down on a post-it, or something similar!

Using an ability depends on the context. Abilities should not be seen as rules for what you can do, but as support for what you want to do. As a player, you can take any ation you deem useful in the given situation - however, if the action you take might be backed up by one of your abilities, you’re probably better off using it! This is simply done by declaring that you’re performing an action, followed by some statement like “I call the client, asking about..., using my ability Discovery!”.

The Product Vision Table

As a product owner, you are the one with the vision for the product! When the session starts, the facilitator will ask you to create a product vision. The tables below will help you design your product vision - the *thing* your team will attempt to build over the course of the scenario. It is up to you whether you want to roll for it, or simply select the parts that you find most interesting!

You will design your product vision by making die rolls in the tables below, and filling the results into the following sentence:

In the field of (A) I want to make a (B), so that I can (C).

Table A - Domains

●	Medicine
●●	Finance
●●●	Government
●●●●	Education
●●●●●	Management
●●●●●●	Travel

Table B - Solution

●	Mobile App
●●	Video Game
●●●	Web App
●●●●	IOT Device
●●●●●	VR Experience
●●●●●●	AI

Table C - Goals

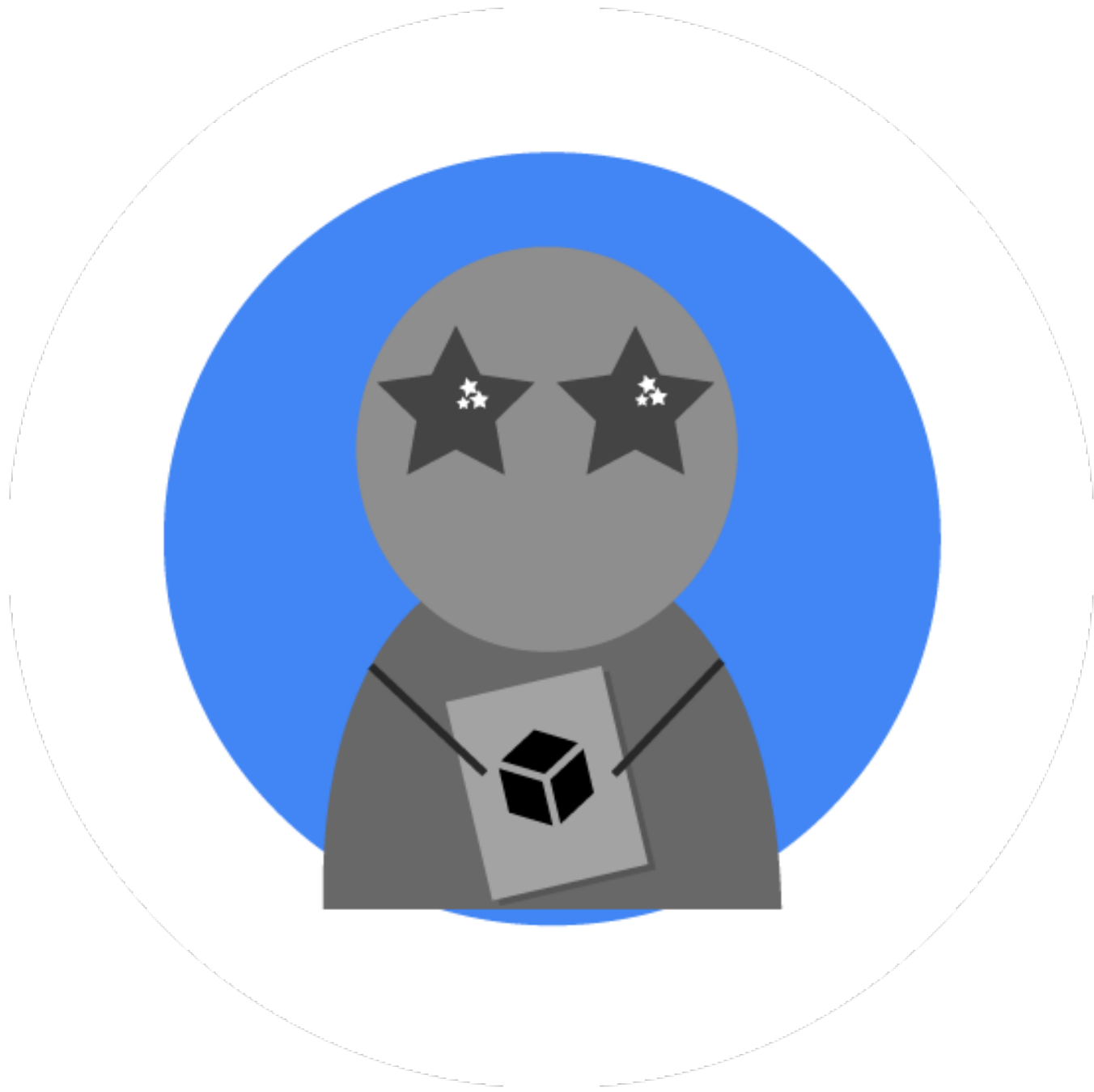
●	Change the world for the better
●●	Save people a lot of time
●●●	Make international collaboration easier
●●●●	Improve communication
●●●●●	Make the experience more fun and engaging
●●●●●●	Share information and create awareness

Your Role

Ability 1

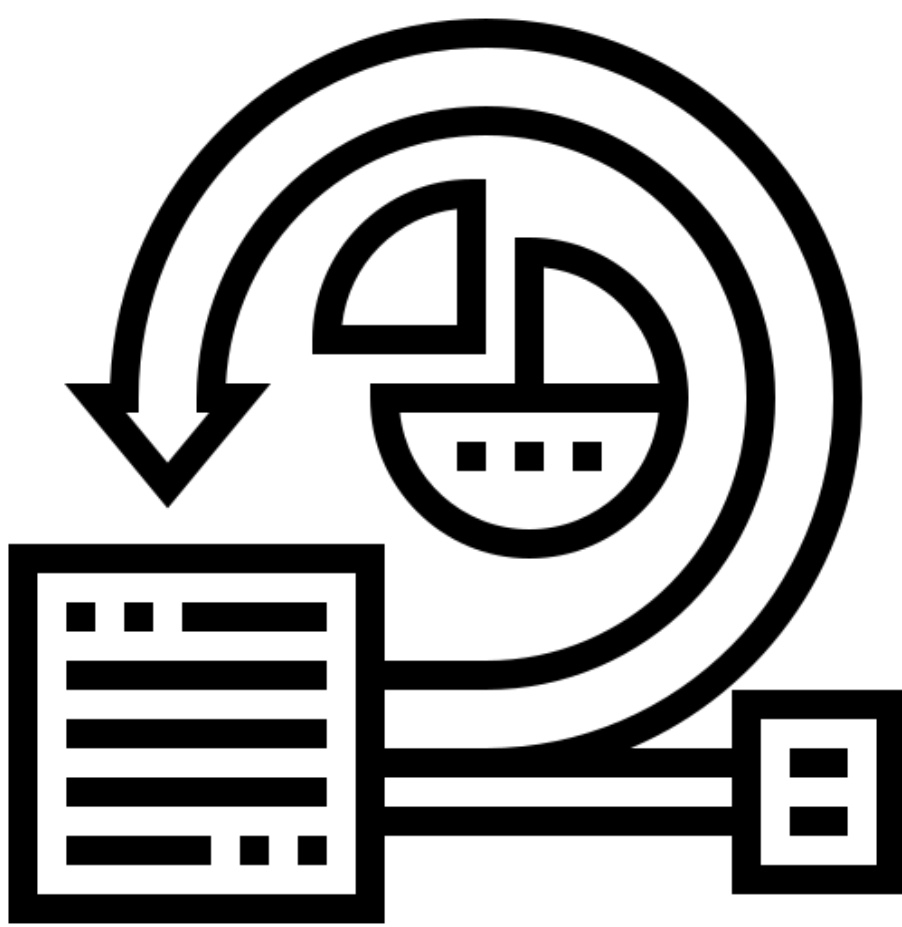
Ability 2

Product Owner



Maximizing Value  
Prioritizing Work  
Directing the Vision

Build Backlog



Define a critical task for your team to undertake! Backlog tasks cost 1 less story points to complete.

1

Discovery!



The Product owner discovers new value in the product vision. This can for example stem from talking with customers/users.

X



Introduction

Welcome to the Quest for Value, Developer!

As the developer, it’s your job to deliver valuable software in incremental slices, utilizing your expert knowledge of your domain.

Your Abilities

As a Developer, your abilities involve building the blocks of the product your team is working on, while supplying expert knowledge about the technical side of things.

Your abilities have different costs, that you can spend your estimation points on every sprint.  
It is not required (nor is it neccesarily always the best use of your time) to spend all of your points each Sprint.

As a developer, one of your abilities has a varying estimation point cost (denoted by the X on the ability). The value of X is determined by the Facilitator, depending on the size of the work you’re trying to complete, on a scale from 1-3, provided the task is deemed possible - attempting to build Facebook in a week might net you nothing but a questioning look from your Facilitator.

The “Domain Expert” ability is a passive ability that developers have. At the start of the session, you chose your domain of expertise. Whenever you make rolls in this domain, you get “advantage” - you roll twice, and pick the better roll.

Using an ability depends on the context. Abilities should not be seen as rules for what you can do, but as support for what you want to do. As a player, you can take any ation you deem useful in the given situation - however, if the action you take might be backed up by one of your abilities, you’re probably better off using it!  
This is simply done by declaring that you’re performing an action, followed by some statement like “I want to build a search function into the app, using my ability Deliver Value!”.

Developer Example

*After hearing the Product Owner’s pitch, the developer thinks for a bit, then asks a fellow developer: “In order to make that app, I think we might need to model an architecture first. I suggest we try and build a simple three-tier stack, and start out just implementing the simple shells for each.”*  
*The other developer agrees.*  
*The first developer tells the Facilitator that they would like to use their “deliver value” ability.*  
*The facilitator argues that this is a medium level task, costing them 2 estimation points.*  
*The developer rolls for success, but uses two dice rolls, since the developer chose the domain “Architecture” at the start of the session.*  
*The developer rolls a 2 and a 5, picking the 5 as the better roll.*  
*The developer is successful, and the Facilitator narrates how the work succeeds.*

Your Role

Ability 1

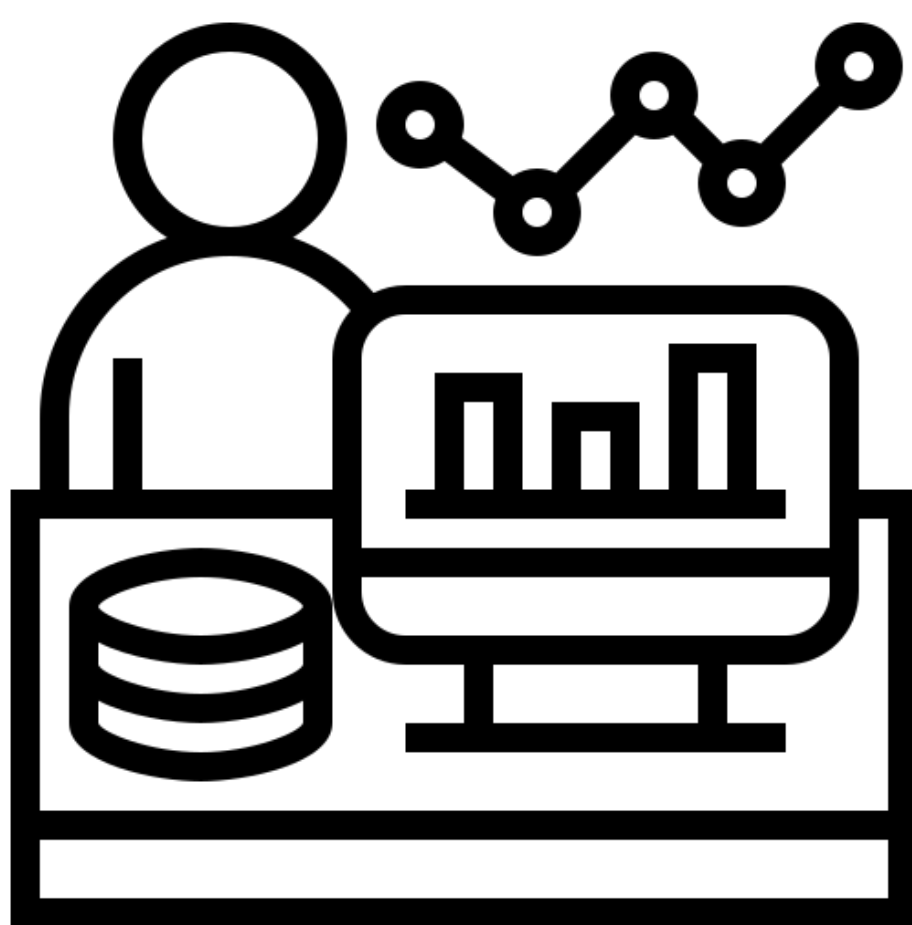
Ability 2

Developer



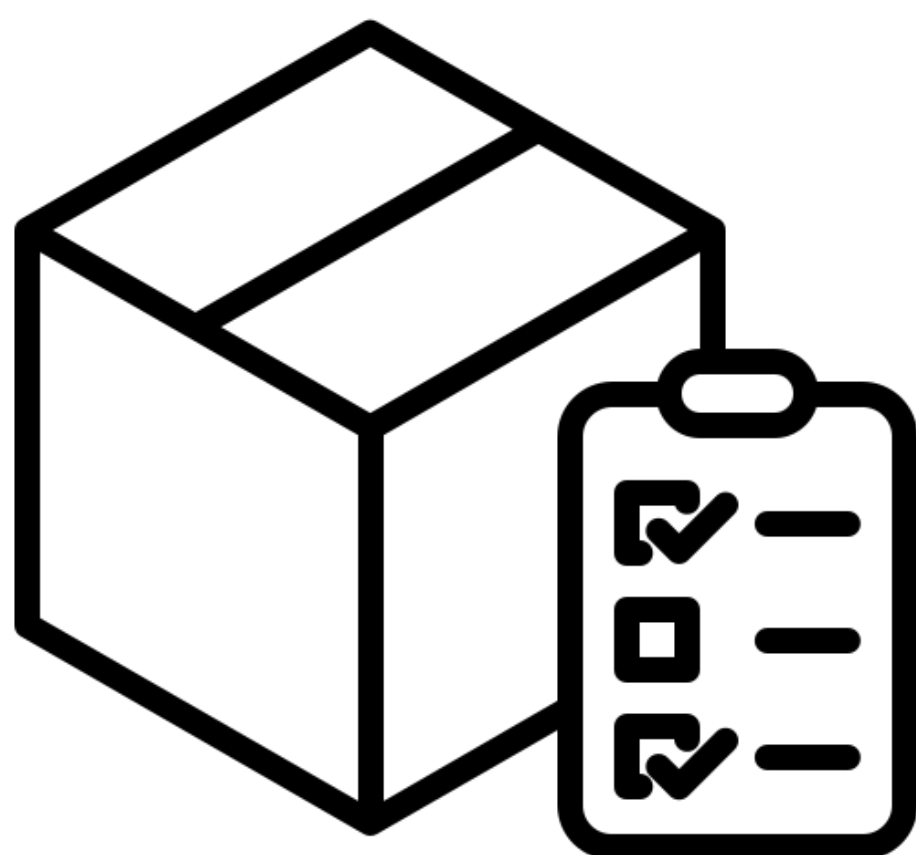
Delivering Value  
Self-organizing  
Expert Knowledge

Domain Expert



Developers get to chose a domain, and get advantage on rolls in that domain.

Deliver Value!



The developer produces valuable software! Roll for degree of success.

