Consultancy Scrum

Agile development for clients and vendors

The Four Kitchens team developed the Consultancy Scrum approach to better integrate agile development methodologies with client/vendor relationships. Four Kitchens is a design and development firm located in Austin, Texas, that works exclusively with Drupal and other open-source web software. Unlike many firms, Four Kitchens is a one-stop shop for large-scale web projects. Branding, design, information and systems architecture, usability, development, and project management are all handled in-house by our team of talented Web Chefs using agile methods in close collaboration with clients.

Welcome to the Web Chefs' recipe for using Scrum in collaboration with clients. Let's get cookin'!

What is Scrum?

Scrum is a simple framework for managing projects. It's an approach to adapting to ever-changing needs in software development. Requirements change, timelines change, ideas change, stakeholder priorities change, and budgets change. Always. With scrum, the focus is placed on getting the most important stuff done first with more frequent, demonstrable, releasable features.

Protip: Glossary
For those new to scrum, there's a useful glossary at the back of this book for quick and easy reference.

Last updated: 10/2/2014

Scrum challenges

Organizations created Scrum and other agile methodologies to manage and develop their own products using their own resources. As a result, they're inward-facing. The stakeholders, Product Owner, Scrum Master, and team are all part of a single company. Budgets and resources are often managed by a single entity. All parties are invested in the process, which makes communication between stakeholders and the team relatively easy.

Vendors, however, are external to the client, so their Scrum must be outward-facing. Vendors cannot rely on the traditional client to be good stakeholders or Product Owners. Because clients control the budget and vendors control the resources, not all parties are invested in the development process. Often, the client has hired the vendor to simply make a problem "go away," and they don't care what methods are used to manage the project.

In short, Scrum was designed for *products* and internal teams and stakeholders. Consultancies work on *projects* with external teams and stakeholders.

Consultancy Scrum

This approach to agile development is a roadmap for integrating clients into the vendor's process. In Scrum Classic, the Product Owner is the internal person responsible for guiding the roadmap, backlog, releases and iterations. In Consultancy Scrum, the Product Owner is someone on the client side. It may take some training on the part of the vendor, but an integrated, invested client can be the development team's greatest resource.

Protip: Training Clients
Clients who are new to the Product Owner role will need to learn how to write user stories, how to prioritize them, what to look for in stories, and why accepting stories is important.

Most importantly, clients need to understand how something can be "done" when it's only one piece of the puzzle.

Why Scrum?

Scrum is a project management framework that takes an incremental and iterative approach to software development with a goal of constant improvement: constant improvement to the product and constant improvement to the process itself. Scrum helps teams build the most important features first, move rapidly to a first release, and get a product in customers' hands sooner.

Using Scrum encourages clients to determine which features are a priority and which features can wait until later releases. Features are built in "thin, vertical slices." Like a slice of cake, all the layers are there for the full taste experience, even without the whole cake. In tech terms, this means that all the components needed for a feature (design, backend, frontend) are created at once. This approach does have some overhead, but it also means that any feature that is done is never waiting on another layer to be completed in order to be released. Done is done.

From the client perspective, Scrum gives frequent and consistent visibility to the team's status more often. With demos every two weeks, open to any and all stakeholders, the client will always know what has been completed and what's going to be worked on next. The process also provides frequent feedback between clients and vendors, giving clients the opportunity to communicate if things aren't on the right track.

Scrum isn't just a framework, it's a philosophy of adaptability, trust, and making commitments. The Scrum team can use it as a framework for discussions that would otherwise be difficult. The client Product Owner can, in turn, use the framework to plan which features the team should focus on next. Clients can also better define expectations for other stakeholders.

The people of Scrum

There are three primary roles in scrum: the **Scrum Master**, the **Scrum team**, and the **Product Owner**.

The **Scrum Master**'s main duty is to ensure that the Scrum team has the best possible environment to be productive. As the name indicates, the Scrum Master is also responsible for enforcing the Scrum process.

The **Scrum team** is the group of developers, designers, and systems administrators who build the product.

The **Product Owner** is responsible for defining the product that the team is building. In Consultancy Scrum, the Product Owner is someone from the client side. It may require some training to bring a client unfamiliar with the Scrum process up to speed, but the payoff is a client who is informed, involved, and integral to development.

The Product Owner's primary responsibility is to keep a ready backlog of features. The Product Owner determines what features are required in each release, the priority of the features, and holds the product vision.

To be ultimately effective, a Product Owner needs to have **time, knowledge, and authority**: time to attend Scrum ceremonies, knowledge about the product's features, and the authority to make decisions about the product. Depending on the Product Owner's level of Scrum expertise, the Scrum Master and Scrum team can assist with the story writing and ordering of features in the backlog.

Sprints

In Scrum, a team gets ever closer to a product launch through a series of development sprints. Sprint lengths can vary, but for the sake of this book, assume sprints are on a two week cycle: nine days of development and one day reserved for demos, feature deployment, review, and planning. Every two weeks, then, the team will demonstrate the features it committed to build, review how well the previous two weeks went, and plan for the next two weeks of work.

Getting to work!

First things first: Kick off!

Before diving into development, take some time to introduce the client to the team and the team to the project during a project kickoff meeting. Once the team is familiar with the goals of the project, start loading up the project backlog.

Building up the project backlog

The project backlog is a list of features that the Product Owner wants to include in the final product. There are two twists to this feature list, however: 1) the features are prioritized and 2) the features are defined in independent, vertical slices of functionality. This way of describing features is called a user story.

Definition: User Story
A user story is the description of a feature told from the user's point of view. It also provides the business value of the feature. Example: As a site visitor, I want to see the avocado crop report so that I know when to buy the best avocados.

Writing requirements in this way creates thin, vertical slices of functionality. The requirement then includes all of the necessary components to make it work: database, back-end business logic, user experience design, and visual design.

How to Demo

When writing user stories, take the time to write a brief “How to Demo” script. This helps the Product Owner and development team get on the same page by articulating what steps will demonstrate that the story is complete.

Once the product is described in user stories, it’s time to plan the first development sprint.

Taking the time to plan

Each sprint begins with a planning meeting. This meeting usually takes about two hours. During the meeting, the team discusses the highest priority user stories with the Product Owner. Discuss what each story is about, possible technical challenges, and talk at a high level about possible solutions. Next, determine the size of the story with a game called Planning Poker. This helps the team determine how much work they’ll commit to for the sprint.

Planning Poker is an estimation card game the team plays to determine the size of stories.

1. For each story, the team discusses what the feature is about.

2. Once everyone has a good sense of what the feature is, the team is ready to play their hands. Everyone selects a card with the story point value they want to assign the story. The cards have the values of a pseudo-Fibonacci sequence on them: 0, 1/2, 1, 2, 3, 5, 8, 13, 20, and 40. Sometimes, team members will also use ? and infinity to indicate they don't fully understand the story.

3. At the count of three, everyone shows their card at the same time. This ensures that everyone is estimating independently.

4. If everyone agrees on the size, record the value and move on to the next story. If the sizes vary, discuss it a bit more and then either reach a consensus or vote again.

One more thing: the Scrum Master and the Product Owner don't get to play—they can help facilitate discussion and answer questions about expectations, but they don't get to influence the team's decision.

Protip: Story Points
Story points are a valuation of the relative size of a story when compared to other stories in the same project. The size doesn't always translate into a time estimate. This helps the team determine how much work they can commit to in each sprint.

Why use story points instead of just estimating hours?

In development, like many things, it can be hard to judge how long, specifically, it might take to complete a task. Judging relative effort is a bit easier, though. Mowing the lawn takes about twice as long as cleaning the bathrooms. A German shepherd is about twice as big as a border collie.

When working on a project, developers can get a sense of how much work features are going to require relative to other things they've seen before. As the project continues and the team gains experience working together, estimates get more accurate.

Once everyone has agreed which stories are the top priority, the team will commit to completing a certain number of story points. This is the amount of work the team is willing to say will definitely be done by the end of sprint demo. The team should only commit to a number of story points they believe they can finish in a single sprint. If they end up completing all of the work, they can pull stories in from the backlog. In future sprints, the velocity, or number of story points to which the team commits, is dictated by the number of points completed in the last sprint.

With the sprint commitment made, the planning session is complete and the team can get to work.

Protip: Velocity
In Scrum, the Product Owner must trust that the team is working at a sustainable pace. The team determines their velocity over the course of several sprints and only commits to that much work for the next sprint. The team, in exchange for this trust, is honest with the Product Owner about the amount of work they can complete. They don't under-commit or over-commit.

If things move faster than expected, the team can choose to pull the highest priority item from the backlog into the sprint. With appropriate and reasonable commitments, expectations are appropriately set and all of the work that is expected to get done is completed.

Protip: Pick Two
Encourage the Product Owner and stakeholders to pick their two highest priorities: Budget, Timeline, or Features. The backlog should always be fine-tuned to meet the most important of the two, allowing the third to be flexible. Note: Budget is always one of the two, even if the client says it isn't. When working for an outside stakeholder, budget is a driver and should be reviewed with the client and team on a consistent basis. This minimizes the risk of surprises. No one wants to run out of money before the project is complete.

Daily Scrum Meeting (aka "Stand Up")

Every day, the project team convenes for a quick, 15-minute Scrum meeting. This is when the team tells everyone what they completed since the last scrum, what they are planning to complete by the next scrum, and anything that is preventing them from doing their work. This provides a daily opportunity for the client side Product Owner to communicate with the team and is a great time for the team to ask questions, get clarification, and generally remove the obstacles and blockers that are keeping them from completing a task.

Protip: Daily Scrum
The daily Scrum meeting isn't a status meeting in the traditional sense. It's a standard time that the team can get together and ensure that they are on track for the sprint. The team can also ask the Scrum Master and Product Owner to clear the path of any issues that might be preventing them from doing work. The Product Owner and Scrum Master must resist talking in the meeting unless answering a question. This time is for the team to ensure they have everything they need.

Pre-Demo

If the team wants feedback from the Product Owner or other project stakeholders before the formal demo, they can organize a pre-demo. Like a demo, the team demonstrates various features according to the “How to Demo” script written during planning, solicits questions about the implementation, and asks for feedback. This is a great way for the team and the Product Owner to touch base and make certain development is proceeding along the right track.

Sprint Demo

At the end of the sprint, the team holds a demo that is open to any member of the client team. While stakeholders in the client organization are not required to attend, their presence is welcome.

Demo Structure:

1. The team demonstrates each of the stories according to the “How to Demo” script written during planning.
2. The team answers any questions about the functionality that the Product Owner or other stakeholders might have.
3. If the story has been completed successfully, the team asks the Product Owner if the story is accepted.
4. Continue demonstrating.

Protip: Rolling stories

During a demo, sometimes it's tempting to ask for small (or large) changes to a feature before accepting it. In this situation, there are several factors to consider.

- Any change to the feature is more work that needs to be done.
- Does the feature, as demonstrated, function well enough to meet the requirement or does it miss the mark entirely?
    - If it genuinely fails and needs to be reworked in the next sprint, the story should fail acceptance and be addressed in the next sprint.
    - If it functions well enough, the Product Owner accepts the story as complete, adds the changes as a new story in the backlog, and prioritizes the changes against other features. Sometimes that change won't seem so important, after all.

Backlog Grooming

During the sprint, the Scrum Master may call a backlog grooming meeting. Sometimes, this meeting is also referred to as a "pre-planning meeting." Similar to a planning meeting, the backlog grooming meeting is a time for the developers to discuss user stories with the Product Owner. They can discuss the intended use and behavior and its priority relative to other stories in the project backlog. The team can write description notes, steps for "how to demo" the feature, and size the story if there is enough information.

Protip: Weekly Backlog Review
Schedule a regular backlog review meeting between the Product Owner and someone on the vendor side, ideally the Scrum Master or a project manager. Having a client and vendor representative working together to groom the backlog regularly makes for a healthy, prioritized backlog.

Sprint Retrospective

At the end of each sprint, after the demo, the Product Owner joins the team for a sprint retrospective. This is a time designated to review how well the team is working together and what steps can be taken to improve. In a textbook retrospective, the Scrum Master poses the same three questions to everyone on the team in a round robin fashion:

1. What went well?
2. What didn't go as well?
3. What can we do to improve?

Everyone is given the chance to speak and discussion is discouraged until everyone has had the chance to speak.

Release planning: When will the project be done?

Because Scrum takes an iterative approach to project development, deciding release dates is a little different from traditional approaches.

Classically, an arbitrary deadline is set at the beginning of the project. As the deadline approaches, teams scramble, work long hours, and often deliver buggy software. Scrum takes a different approach. As the project develops momentum, the team begins to work at a predictable pace. Their velocity, or number of story points they complete each sprint, becomes fairly standard. Once the team and client have a ready backlog, including all of the features needed for launch that have been sized, the team can predict when the minimally viable product (MVP) will be ready. For example: if there are 200 story points in the backlog and the team has a velocity of 40 points, the team can confidently say that they'll be ready to launch in 5 sprints (or about 10 weeks).

Protip: Define MVP ASAP
A fully formed backlog is essential to defining the full scope of the project before work begins so negotiations can happen early. Once MVP is defined, the client and vendor both have a clearly defined goal in sight.

Early in the process, schedule a release planning meeting to review what features must be complete before the first release. This way, the release can be scheduled based on the team's established velocity.

Project Retrospective

Similar to the sprint retrospective, the project retrospective takes place soon after the project has launched. This gives the team and Product Owner a chance to inspect their collaborative process. Agile development seeks to constantly improve the process, not just on a sprint-by-sprint basis, but also in the longer-term sense. The project retrospective is a chance to iterate on the Consultancy Scrum process.

Protip: Client Management
When the client is integrated into development, it's easy to think of them as another member of the team. Remember that they're still the client and have business and emotional needs that need to be met beyond the scope of the project. Appoint an account manager-like point of contact who isn't purely project focused. This person can have frank discussions with the client to make sure all of their needs are being met without any of the day-to-day project concerns getting in the way.

Learn More

Learn more by visiting www.consultancyscrum.org and keep the conversation going!

Last updated: 10/2/2014

Glossary

MVP - Stands for “Minimum Viable Product.” A collection of the fewest features necessary to release a product.

Planning meeting - Each sprint begins with a meeting in which the team discusses the highest priority user stories with the Product Owner.

Product Owner - The person responsible for defining the product that the team is building. In Consultancy Scrum, the Product Owner is someone from the client side.

Project Backlog - A list of features that the Product Owner wants to include in the final product.

Scrum Master - Ensures that the Scrum team has the best possible environment to be productive. As the name indicates, the Scrum Master is also responsible for enforcing the Scrum process.

Scrum meeting - A quick, 15-minute meeting in which the team tells everyone what they completed since the last Scrum meeting, what they are planning to complete by the next Scrum meeting, and anything that is preventing them from doing their work. This meeting is sometimes called a “stand up” because team members will avoiding settling in the way they might for a longer meeting.

Sprint - A series of short development periods (ideally around two weeks) during which the development team builds completed features to get ever closer to a product launch.

Sprint Backlog - The set of user stories that the team has committed to work on during the sprint.

Sprint Demo - At the end of the sprint, the team demonstrates all of the work completed during the sprint.

Sprint Retrospective - At the end of each sprint, after the demo, the Product Owner joins the team for a sprint retrospective. This is a time designated to review how well the team is working together and what steps can be taken to improve.

Story points - A valuation of the relative size of a user story when compared to other stories in the same project. These help the team determine how much work they can commit to in each sprint.

User Story - The description of a feature told from the end user's point of view. It also provides the business value of the feature. Example: As a site visitor, I want to see the avocado crop report so that I know when to buy the best avocados.