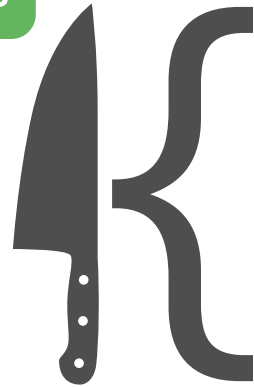


we make BIG websites

FOUR
KITCHENS



Draft: 01 August 2012

WORKING WITH THE WEB CHEFS: A GUIDEBOOK

Contact info

email shout@fourkitchens.com

phone 512 454 6659 fax 888 243 9124

office 3415 Greystone Dr. #210, Austin, TX 78731



So, you're about to start working with the Web Chefs? Let's get cookin'!

You already know that we're world-class web developers who are passionate about performance, design, and open source. Aside from the technical expertise we bring, much of our success is attributed to how we approach work. At Four Kitchens, the approach we use is called scrum.

Four Kitchens has three certified scrum masters on staff: Suzy, Andrew, and Leah. Our founders, Aaron and Todd, are certified product owners.

If you haven't used, or even heard of, scrum before, don't worry: the Four Kitchens team are scrum experts. We've also put together this little guide to get you started.

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, stakeholders change, and budgets change. Always. With scrum, the focus is placed on getting the most important stuff done first. The focus is also on getting to demonstrable, releasable features more frequently.

Why Scrum?

The main reason we scrum is that it jibes well with our people-first approach. Like The Agile Manifesto (agilemanifesto.org) states, we value "individuals and interactions over processes and tools" and "customer collaboration over contract negotiation." We also value "working software over comprehensive documentation" and "responding to change over following a plan."

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. We emphasize getting functions and features to a demonstrable, releasable state through a series of development sprints. Overall, a majority of product features go unused. Stated another way, a majority of feature development often holds up a release. With that in mind, we work on the most important features first, trying to move rapidly to a first release, getting your product in customers' hands sooner.



We work with our clients to determine which features are a priority and which features can wait until later releases. 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. It also means that we're not building infrastructure that will never be used.

From the client perspective, scrum lets you see what we're working on more often. We do demos every two weeks, open to any and all stakeholders. You will always know what has been completed and what's going to be worked on next. We also seek frequent feedback from our clients. This gives you the opportunity to tell us when we are and 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 product owner and client can, in turn, use the framework to plan for what features the team should focus on next and for defining how the development of product will pan out when discussing progress with stakeholders.

If you want to understand more about the scrum philosophy, we can direct you to some excellent resources. Now that we've covered some of the whys, let's take a look at the whos and whats.

The People of Scrum

There are three roles that you'll hear about 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 will be building your project.

The final role is that of the product owner. As a client working with Four Kitchens, the product owner, typically, is you. This means you'll be responsible for defining the product that the team is building. If you've



never worked on a scrum or agile project before, this job takes on a slightly different shape than with other methods of project management. In textbook scrum, 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. If time is not a luxury you have, we can appoint a *project owner* within Four Kitchens who can hold the product knowledge, prioritize features, and help maintain the backlog. Depending on the level of help required, the scrum master and scrum team can assist with the story writing and ordering of features in the backlog. More on stories and the backlog in a minute.

Sprints

In scrum, we get ever closer to a product launch through a series of development sprints. Sprints at Four Kitchens are on a two week cycle: 9 days of development and 1 day reserved for demos, feature deployment, review, and planning. Every two weeks, then, we will invite you to a series of "scrum ceremonies" to demonstrate what we've committed to working on, 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 we dive into development, we'll take some time to introduce you to the team and the team to the project during a project kick-off meeting. Once the team is familiar with the goals of the project, we'll start to work with you to load up the project backlog.

Loading up the project backlog

The project backlog is a list of features that you want to include in the project. There are two twists to this feature list, however: 1) the features

Product backlog:

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.

When we write user stories, we also take the time to write a brief "How to Demo" script. This helps everyone get on the same page by articulating what steps will demonstrate that the story is complete.



are prioritized and 2) the features are defined in independent, “vertical” slices of functionality. This way of describing features is called a user story. Once we have the product described in user stories, it’s time to plan the first development sprint.

Taking the time to plan

At the beginning of each sprint, we’ll ask you to join us for a planning meeting. This meeting usually takes about 2 hours. During the meeting, the team will discuss the highest priority user stories with you. We’ll discuss what each story is about, possible technical challenges, and talk at a high-level about possible solutions. Then we 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, we’ll also use ? and infinity when someone really doesn’t understand.
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, the value is recorded and we moved on. If the sizes vary, we 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.

Now that we’ve agreed which stories are the top priority, the team will make a commitment to a certain number of story points. This is the amount of work the team is willing to say will definitely be done by the

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 be relative to other things they’ve seen before. As the project continues and the team gains experience working together, estimates get more accurate.



demo. During the first sprint, the team will commit to a number of story points they feel comfortable committing to. 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 the team commits to, 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.

Daily Scrum Meeting

Every day, the project team will convene in the middle of the day 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. As a product owner, we will need you to attend this meeting. The scrum master on your project will provide you with a call-in number. This is a great time for the team to ask you questions, get clarification, and generally remove the obstacles and blockers that are keeping them from completing a task.

Pre-Demo

If the team wants feedback from the product owner or other project stakeholders before the formal demo, they'll organize a pre-demo via screen share. Like a demo, the team will demonstrate various features according to the "How to Demo" script written during planning, solicit questions about the implementation, and ask for feedback.

Sprint Demo

At the end of the sprint, the team will hold a public demo. A public demo is open to any member of the Four Kitchens staff and any member of the client team. While stakeholders in the client organization are not required to attend, their presence is welcome.

Demos are pretty straightforward:

1. The team will demo each of the stories according to the "How to Demo" script written during planning.
2. The team will answer any questions about the functionality that the

Protip: Trust

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 the trust, is honest with the product owner about the amount of work they can complete. They don't under- or over-commit. They will commit to what they believe they can definitely get done. If things move faster than expected, the team can choose to pull the next highest priority item from the backlog. With appropriate and reasonable commitments, expectations are appropriately set and all of the work that is expected to get done is completed.

In a lot of projects, the process is simultaneously feature- and deadline-driven. All the features need to be done by a certain date. This causes a lot of Product Owners and project managers to over-commit their team and teams to feel pressured into over-committing. At the end of the development cycle, however, everyone is disappointed because things didn't get done, there are a lot of bugs, and trust between the team and product owner was broken.



product owner or other stakeholders might have.

3. If the story has been completed successfully (or nearly successfully), the team will ask the product owner if the story is accepted.
4. Continue demonstrating.

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.

Sprint Retrospective

At the end of each sprint, after the demo, you may be invited to join 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 will pose 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.

The format of the retrospective isn't set in stone, however, so be prepared for a variety of activities led by your scrum master.

Release planning: When will my project be done?

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

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.

Protip: Rolling stories
When a story isn't quite done.

During a demo, sometimes it's tempting to ask for small (or big) changes to a feature before accepting it. In this situation, there are a couple of considerations to make. 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, we generally encourage the product owner to accept the story as complete, add the changes as a new story in the backlog, and prioritize the changes against other features. Sometimes that change won't seem so important, after all.



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. With scrum, a different tack is taken. 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).

Early in the process, we'll schedule a release planning meeting to review what features must be complete before the first release. This will help you schedule your release once the team has established their velocity.

Project Retrospective

Similar to the sprint retrospective, we will likely invite you to a project retrospective when your project has launched. This gives us a chance to inspect our process with you. We seek to constantly improve, not just on a sprint-by-sprint basis, but also in the longer-term sense. The project retrospective helps us do this.

In Conclusion

In conclusion, we're excited to be working with you. If you have any questions about the scrum process, feel free to ask your project's scrum master or any of the web chefs. We are enthusiastic about scrum and love to talk about it.

GLOSSARY:

Backlog -

Planning meeting

Product Owner



Scrum Master

Scrum meeting

Sprint

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

Sprint Demo

Sprint Retrospective

Story points

User Story

DRAFT

thank you!

