

A Non-Jurassic Approach to Managing Projects

So you're ready to build an app, eh? Before you start coding, you need to make sure you have a project management strategy ready to go. In this ebook, we outline our agile approach to app development, and give you some tips and tools for keeping the process streamlined and efficient.

Table of Contents

Intro	1
Our approach	1
Iterations	2
Meetings and day to day	2-4
Toolbox	4
Estimates	5
Conclusion	6

Intro

Welcome to the Tyrannosaurus Tech project management ebook!

In this book you'll learn:

- How we work together with clients to achieve rapid, iterative small wins for one big win
- Which tools, meetings, and tasks we leverage to get those wins
- Our framework for gathering requirements and estimating the work to deliver those requirements
- How we deliver work over the course of a project

Our approach

At Tyrannosaurus Tech, we favor speed and communication over bulky processes and the traditional, waterfall-style approach to projects. To that end, we follow an agile style of management to ensure our clients are always in the loop and provide valuable feedback on deliverables.

We use a number of tools, touchpoints, and tactics to get this done, but ultimately these result in small sets of deliverables. We do this because rapid approvals and touchpoints allow us to move quickly and stay on target.

Let's dive more into the iterative approach!

Iterations

Whereas the old methodology of shipping software products was all about the waterfall approach (one large chunk of work in isolation at a time), the more modern approach leverages an "agile" style of management. Agile encourages stakeholders and project teams to communicate frequently, and it calls for more frequent review of deliverables and progress throughout the project as well.

The project is typically broken into two-week intervals, called a sprint or iteration. Inside of each iteration, a team picks up work they can complete within that interval and demos at the end to the group of stakeholders. This differs from the waterfall method in that with waterfall, stakeholders would usually only see the reviewable product/deliverable when it was fully completed based on the project requirements. The issue with this approach is that it leaves the precious feedback from those stakeholders out of the build process, resulting in pretty much zero visibility into the project's progress.

Let's talk more about what specifically goes into a sprint/iteration, and the essential tools and meetings needed for teams to execute.

Meetings and day-to-day

At a high level, a sprint or iteration usually contains the following meetings:

- Grooming
- Sprint planning
- Daily Standup
- Retrospective
- Demo

Grooming

Traditionally, grooming happens at the end of the previous sprint and is what sets up the planning meeting for success. Grooming calls for the team to review upcoming work and do the following:

1. The team must make sure the task is small enough to be completed in one sprint
2. The team should identify blockers and dependencies in that task
3. The team must prioritize the tasks amidst the broader backlog

Each task must be small enough to consume that sprint, and should look like this:

"A user can upload a photo from their phone's library to use as their profile picture"

NOT

"A user has a profile with photos, a bio and other info"

This is what we mean by a task's size. The former is a user story, whereas the latter is too broad in definition for the developer or designer to complete that story as intended.

Sprint Planning

First, a sprint is kicked off with its planning meetings. This usually takes 1-2 hours and the team will review what work is available (and hopefully prioritized) for them to pick up and complete that sprint. The project manager goes around the room with each team member, reviewing available tasks and assigning them to that team member for the sprint. The team member also estimates the level of effort to complete that task, but we'll get to that more in detail later.

How do we proceed after grooming and planning are done? The standup of course!

Daily Standup

In the daily standup, the project manager leads each team member through the following set of questions in less than 15 minutes:

- What did you do yesterday?
- What are you working on today?
- Do you have any blockers?

The PM follows up with team members separately on any topics outside of these three questions, and follows up separately to resolve those blockers/inhibitors as well to make sure the team can work smoothly.

Retrospective

After two weeks of work, the internal project team meets for the retrospective. This meeting helps teams talk about what went well, and what could be improved. That way, the teams receive some props for excellent work from each other, but also identify areas of improvement in the process to be implemented in the upcoming sprint.

Demo

Finally, there's the demo, which is exactly what it sounds like: the team demos the work to key stakeholders. The client provides feedback free form, but is also advised to record feedback separately in their own internal processes and return with the full set of feedback within 48 hours so the team can also incorporate some of that work in the upcoming sprint.

Toolbox

What about the tools though? Project managers have many tools at their disposal to manage the team and the deliverables, but at Tyrannosaurus Tech we use the following:

Waffle.io

Waffle integrates directly with GitHub to manage the product backlog, current sprint tasks, and delivered stories all in one place

Active Collab

Active Collab is used for design and client deliverables, and for broader conversations about project scope and status reporting. It is handy because it has no file size limits, so the teams can easily share large documents or design deliverables—which sometimes can be massive!

TeamGantt

The good ol' Gantt chart has been around forever, but TeamGantt makes it easy to build out the project timeline and invite teams to collaborate or review it.

Slack

Slack is essential to teams of all sizes and types these days. The day to day conversations and reviews are usually handled in slack as the primary messaging platform on projects.

Estimates

So what happens when a team member prepares to take on new work in a sprint, and needs to estimate the level of effort? The preferred option is to measure by complexity.

Estimates in complexity

Estimating by complexity allows stories to be contained to a certain size. Velocity is calculated as how many stories of that size can be completed within a two week sprint.

To calculate this estimate, we use a point system that is usually broken out like so:

Points	Example
1	Text change
2	Simple task, bug
3	Simple yet defined feature
5	As complex as is acceptable in a single sprint, touches multiple parts of the codebase
8	Takes entire sprint, very complex, needs to be broken down more

Over time, this system helps the team accurately predict what they can complete in any given sprint. For example, they may find that they can handle roughly two three-point stories and one one-point story per resource, per sprint.

Estimates in time

Rather than measuring by complexity of a story, teams can provide estimates based on time. For example:

"I think I can get this feature done in 2 days, so I will assign it a two."

However, time estimates cannot reflect the complexity of the project or the certainty of the estimate. In contrast, the Fibonacci point system used for our complexity estimate reflects the increasing uncertainty of the estimate relative to the sprint's growing complexity. In other words,, estimating by time doesn't help you measure the certainty of the estimate, whereas measuring by complexity does. For this reason, we recommend using complexity estimates for the best accuracy.

Conclusion

Once we have all of the above nailed down, the process enforces continuous delivery each sprint as deliverables are reviewed with the client and merged into the main body of work. In a designated amount of time, the product is taking shape and approaching the deadline with processes in place to maintain the timeline or beat it. Because this approach also enforces multiple touch points with clients to receive feedback, prioritize work, and remove blockers, agile projects tend to flow much more smoothly than standard waterfall projects.

TYRANNOSAURUS **TECH**

THANK YOU FOR READING!

From app development to technical consulting, we have the skills and expertise to help you use technology to drive your business forward. Check out tyrannosaurustech.com/services to learn more about how we can help.

Sign up for a free consultation by emailing us at info@tyrannosaurustech.com or calling 770-282-9628. We're ready when you are!