Our Process

/Revelry

An AI-Powered, Lean Agile Software Development Process to Help Businesses Launch, Scale & Succeed

revelry.co

Unleash Your Potential

Building something new – or improving what you have – can be challenging, especially in the world of software development. However, with Revelry as your partner, the product development process becomes a clear and exciting part of your journey to success. We bring proven expertise, a commitment to your product's success, and a unique AI-powered, Lean Agile process.

Here, we offer a closer look at how Revelry has been getting things done for our partners worldwide since 2012. These are our processes and tools that enable our team to deliver scalable software solutions over and over again.

But First, A Note From Our Founder and CEO

Building software can be hard. At Revelry, we've optimized our product delivery workflows into a process we call Lean Agile; we apply it to just about every part of our business. We've also developed an arsenal of tools – frequently leveraging AI – that de-risk various delivery tasks; collectively they're referred to as our Innovation Stack.

The combination of our process and stack eliminates as many meetings as possible and enables productive and time-saving async communication, so product design and engineering teams have the headspace to do their best work and scale in a remote-first world. We're sharing it here with you, because we've learned so much from our industry peers and believe in open collaboration.

No company is the same, but we hope you'll find ideas and information here to help make your processes more efficient. If you'd like to chat, let's connect and talk about what we can build together.

– Gerard Ramos

Let's dig in...

Step 1: Assemble Your Product Delivery Team

We understand that choosing the right team members is crucial for a successful project kickoff. You have the flexibility to choose the experts you need, whether it's product managers, engineers, designers, or other professionals. You can request specific resources with a few clicks in our partner platform, or we can work with you one one-on-one to curate a team composition tailored to your project's needs. Another important member of your team is the Product Owner, someone from your organization who serves as the ultimate decision-maker on your project, working with our team on a day to day basis. If you don't have someone from your team who can work this closely with us, we will still ensure regular touch points for approvals and reviews throughout the product development process.

As your project progresses, we understand that your needs may evolve. You can go into our partner platform and request changes to your resourcing plan at any time. Our team closely monitors the progress and dynamics of your project and regularly provides recommendations for optimizing your team composition. As we go this journey together, it's important to have the right resources in place to drive your project forward.

Step 2: Project Setup

Once we have our team in place, we begin by setting up communication channels and tools to manage your project. The first thing we do is set up a Slack channel, so that the whole team sees all conversations and decisions as they unfold. We then go on to set up your project. We can fit into your existing project management tooling (Jira, Asana, etc.), or if you don't have one, we set it up for you in a matter of seconds using our automated stack.

Our Preferred Project Management Stack and Setup

We prefer to use GitHub as our project management tool, because it serves as both a project management tool and a code repository. Our setup begins with a simple Slackbot command – a first taste of the project efficiencies created by Revelry's tools. The bot (aka Slax) ensures each project is set up in the same way and that no part of the process is left to chance. For new products, this can mean that the first 20 to 30 tasks required to set up a project are defined within a few seconds. For existing projects, we rely on the issue and PR lookup, so we can easily access the defined work right in our shared Slack channel.

Another reason we prefer Github is that it is tightly integrated with ProdOps, our AI driven product operations tool. We use ProdOps to supercharge our ability to generate roadmaps and user stories, leveraging the context of your project's code repository and your project data with the power of LLMs.

Step 3: Develop a Cadence

In pursuit of transparent communication throughout your project, we have communication standards as well as regularly scheduled meetings that keep us moving fast, together.

Communication Standards

- When it comes to *anything project related* (aka not personal), we communicate in the open, in our project Slack channel.
- We immortalize our decisions and tasks into Github issues. ANYONE can write issues, not just one person on the team.
- Clear is kind. Craft your language to your audience and intention.
- When we get blocked on anything, we reach out (in the Slack channel). We don't wait until a meeting to tell people we are blocked or need help.
- If it's easier to get on a call, we get on a call. Text communication can only go so far.

Weekly Kickoffs

We have an initial project kickoff the first week of our project, and kickoffs every week of the project thereafter. Kickoffs are where we make sure that the team knows exactly what it needs to focus on in order to continue driving on the most important objectives. We review our accomplishments from the previous week, talk about what we are working on next, and work through any blockers to those priorities. Product Owners are encouraged to actively

Daily Standups

Standups are an integral part of our workflow. Every morning at 9:30 CT, we have a standup in Slack, detailing what we worked on yesterday, what we'll work on today, and what our blockers are for work in progress. We link tickets to our standups, too, so it's easy to see what issues are being referred to.

Standups provide vital insight to all team members about the status of our collective ongoing work, provide an opportunity for all team members (including Product Owners) to unblock each other and set (or reset) our day's intentions and expectations. Other benefits of our Slack standups include being quicker and more cost effective than getting everyone on a call, and managers are able to quickly skim through the notes on multiple projects to get their pulse.

Product Owner Syncs

We like to have weekly time for you and your product manager to align and adjust your project's roadmap and priorities, to touch base on in-progress or recently completed work, and to ensure that work is defined as expected through requirements review and approval. These crucial meetings make sure we are in lock step on your project's initiatives.

Retrospectives

Every other week the team has a retrospective conversation. We discuss what felt bad, what was awesome, and specific things we can do to improve upon those pain points as we move forward. Retrospectives are where we can transform what went poorly into a newly found product delivery superpower.

Step 4: Discovery and Alignment

After your team is in place, your project is set up, and communication channels have been established, we dig further into your project's needs to find alignment and get to work.

In order to have a clear path to success, we focus on solidifying our initial roadmap and generating our first issues to work on, with the assistance of our in house AI product operations tool, ProdOps.

Identify

For new products, we'll begin to frame the most impactful problems to solve in order to generate an actionable and aligned roadmap. For established products, our goal is much the same – to audit, identify, and roadmap work that maximizes value and fosters growth, all the while minimizing the disruption that often accompanies substantial "upgrades". Whether your project needs a code refactor or UX audit, design thinking, user testing, or competitive analysis, we use the right tools and processes to identify these areas for impact.

Prioritize

Once we have sight on some different areas of impact, we work together to prioritize solutions that serve the unique needs and goals of your business and users. Our approach is always flexible and adaptable, with room to make any necessary changes in priority from week to week, or from day to day.

Prioritization is an art and science that we love. Together we take into account the potential impact, effort required, desired appetite, and dependencies of each item in your backlog to guide the journey to product success. Revelry's team of product managers, designers, and engineers collaborates closely to identify and discuss the risks, complexities, and dependencies of each item in the backlog. This allows us to assess the feasibility and scope of delivering certain features, enabling us to make impactful decisions. We believe in maintaining a living roadmap, working with you regularly to reassess and reprioritize. This allows us to quickly and easily respond to evolving user needs and changes in the market, providing you with a product that not only meets your current requirements, but also anticipates future needs for long-term success and sustainability.

Step 5: Building

Once your roadmap and issues are generated in ProdOps and exported to your project management tool, we start to track that work in our project board to bring greater visibility to its status. Again, we can always adjust by moving things on or off of the board on a weekly or daily basis.

A Note on Issues

Issues are essential for all projects, as they contain the building blocks of your product. We write issues for everything we do, ensuring that each feature or task is well-documented and addressed. This includes everything from research tasks, POCs, epics, user stories, and bugs. Creating issues as soon as possible allows us to maintain clarity and organization throughout the development process. And, remember, issues aren't just for us, you can create them, too. We encourage you to generate your own issues in ProdOps.

Working Issues

Here's a glimpse at how we manage our work on the GitHub project board and move it through to completion, step by step:

- **Write:** This column contains our backlog of written issues that contain the work that we know we want to do.

- **Approve:** This column is where issues go when they are ready to be approved for work by the Product Owner.
- **Estimate:** Before we start officially working on an approved issue, we like to talk about it as a team during a poker session. Poker sessions are where we score stories. We use this as a mechanism for surfacing and discussing unanswered questions about the work, complexities, and any risks. We also use it for tracking the velocity of the team. In story scoring, team members cast their personal votes and then the average score for each story is calculated. In years past, this democratic process took awhile; it wasn't uncommon to spend two hours on a scoring meeting, which consumed a lot of developer time. **We automated the process by building a Slack bot, /poker.** It's essentially a virtual, automated polling system that **eliminates more than an hour of inefficiency.** That time that really adds up when you use it every week for a team of 3-6 people.
- **Implement:** Always organized in priority order, this column contains the issues that are ready to be worked by the members of our team. Team members that are ready to take on a new issue should typically always pull from the top of the Implement column.
- **In Progress:** When a designer, engineer, or other team member begins working an issue, they assign themselves and move the ticket from Implement to In Progress. We see progress through their daily comment updates and linked pull requests.
- **Code Review:** Once the issue's requirements are complete, tickets with pull requests are moved to Code Review and the pull request is assigned

to another team member. Think of this as code quality assurance, where we check one another's work.

- **QA:** Once code has been reviewed and feedback has been addressed, we move the issue to the Quality Assurance (QA) column. We use automated deployment and continuous integration so that code ready for review and QA is continually available for review and testing. When in QA, another team member is able to pick up the ticket and test it. If they find the ticket does not meet the pre-approved requirements, they uncheck the box and send the issue back to the engineer with a comment. If an issue arises in QA that isn't represented by any existing requirements, we log a separate issue. Sometimes we play jazz and make small fixes that were not covered initially, but we typically try to stick to the requirements.
- **UAT:** Once QA is complete, we move the issue to the User Acceptance Testing (UAT) column. If there are problems with meeting the ticket requirements, the person doing UAT sends it back down the chain for needed changes. If there are other issues, we hash it out, document findings, and prioritize accordingly. Issues that don't meet the requirements coming back from QA or UAT will usually take the highest priority slot in comparison to other work in progress or unworked prioritized tickets. Side note: We LOVE it when our partners do UAT, so they can see the magic in action!
- **Done:** Once the issue has passed UAT, we mark the issue as Done. We'll then deploy the feature to production as soon as the next planned release, and in some cases immediately through automated

deployment. After we deploy, we still have work to do before considering something “Complete”. Our teams monitor deployments to make sure new work deploys cleanly and is working as expected in our production environments. We leverage different tools for different jobs, but this stage generally consists of monitoring and error tracking. Any unwanted behavior is addressed quickly.

A Closer Look: The Tools We Use

These are just a few of the tools we use to streamline our software development process.

- Amplitude & Google Analytics – Tracking to understand your product’s user behavior
- GitHub – Code Repositories, Project Tracking, and more
  - Actions – Enables our continuous integration and continuous deployment (CI / CD)
  - Issues – Where the work lives.
  - Pull Requests – How our designers and engineers submit code changes and get feedback on those changes
  - Project Boards – Automatically-updated visualization of Issue progress
  - Dependabot – Helps keep our project repos updated and secure
- Revelry Partner Platform – Your project hub for all things, including: resourcing, contracts and billing at Revelry
- ProdOps – Revelry’s AI-boosted software delivery copilot
- Slack – Communication tool supporting asynchronous work

    - Project Channel – Your project's central communication hub.
    - Reminders – Automated tooling that we use for daily standups and reminders to the team to leave progress updates on their tickets
- Slax — A custom built slackbot that streamlines accessing and creating GitHub issues from Slack
    - New Project command – Creates a new project (repository) in GitHub, adds our starter stories to the project, sets up webhooks, and creates the project's Slack channel.
    - Issue Bot — Slack bot for looking up issues and pull requests and creating issues
    - Poker Bot – Slack bot for efficient story scoring
- Rollbar & Sentry – Tools we use for error monitoring on projects

While these are some of the tools we like to leverage, we pride ourselves on being a right-tools-for-the-job kind of place. Tools are utilized based on the project / stack / partner needs, and this list is just the tip of the iceberg for many things like error reporting and performance monitoring, generating design artifacts, and tracking user behaviors.

*Let us know if you try our process and tools. We'd love to hear from you. You can reach us via email at **hello@revelry.co** or our **website.** Let's build something together.*