

AGILE **HANDBOOK**

OVERVIEW

WHAT IS THIS?

This handbook is meant to be a quick-starter guide to Agile Project Management. It is meant for the following people:

Someone who is looking for a quick overview on what agile is and why it is awesome

Someone who needs help getting their head around agile project management

Someone who is scared to introduce agile on their next project

Someone who needs help selling agile to their boss or client

This guide is not meant to be the end-all-be-all to agile. Far from it. It is meant to give busy people an overview of the framework and its benefits in 15 minutes or less. The resources section lists recommended books and companies that can provide more robust training on how to implement it.

WHO AM I?



And who am I to write about agile? My name is Emerson Taymor and I'm one of the co-founders of **philosophie**. We build better solutions to digital problems. We help startups, agencies and big companies with design and development. And we practice agile. Over the years, I've seen waterfall and agile projects succeed and fail. I've learned what makes them successful, and I've fallen in love with the agile way. I hope to share some of what I have learned within this short handbook.

emerson@gophilosophie.com

415.516.8341

@etaymor

[linkedin.com/in/etaymor/](https://www.linkedin.com/in/etaymor/)

AN AGILE OVERVIEW

Agile is a way to manage projects. It can be used for virtually anything, but it was founded in software development. This handbook focuses on agile for software development, but many of the principles can be expanded to other fields.

Agile breaks down larger projects into small, manageable chunks called iterations. At the end of each iteration (which generally takes place over a consistent time interval) something of value is produced. The product produced during each iteration should be able to be put into the world to gain feedback from users or stakeholders.

Unlike Waterfall project management, which is strictly sequenced: you don't start design until research is done and you don't start development until the designs are signed off on; agile has designers, developers and business people working together simultaneously.

AGILE GOALS

As made popular by the “Agile Manifesto”, agile values:

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

Agile realizes that software (and marketing) projects are inherently unpredictable. Over the course of any project there are likely to be changes. Be it market changes or feature changes as the product comes to life. Agile embraces this unpredictability. By breaking down projects into small chunks, it makes it easy to prioritize and add or drop features mid project. Something that is impossible in traditional waterfall projects.

THE 12 PRINCIPLES

- 1** Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- 2** Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

Welcome changing requirements, even late in development.
- 3** Agile processes harness change for the customer's competitive advantage.
- 4** Business people and developers must work together daily throughout the project.

- Build projects around motivated individuals. Give them
- 5** the environment and support they need, and trust them to get the job done.
 - 6** The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
 - 7** Working software is the primary measure of progress.
 - 8** Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
 - 9** Continuous attention to technical excellence and good design enhances agility.

- 10** Simplicity--the art of maximizing the amount of work not done--is essential.
- 11** The best architectures, requirements, and designs emerge from self-organizing teams.
- 12** At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

**WHAT
STANDS OUT**

WHY AGILE ROCKS

SPEED TO MARKET

Agile lets you get your concept to your users as quickly as possible. During every sprint an agile project delivers something of value. At any point, you may determine you want to launch what has been delivered and start building a user base or testing your hypothesis.

FLEXIBLE

Agile is based on accommodating change. Software projects consistently change. As a product comes to life or the market expands, you should be able to react and update the product accordingly. Agile also realizes that great ideas are bound to come mid-project and being locked into a scope doesn't let you take advantage of these realizations.

RISK MANAGEMENT

Incremental releases means that the product can be used early in the process by stakeholders and users. This lets you identify issues and feature deficits early in the process. Being adaptable to change means it isn't a problem to change the scope midway through the project, something that would be impossible in a waterfall style project.

COST CONTROL

Unlike a fixed budget project, agile is flexible with regard to scope. More often than not, our clients realize features they originally requested are no longer necessary. This allows them to launch sooner and pay less. Agile isn't about paying a lot with uncertainty, it's about paying for only what you need. Need to stick within a budget? No problem! We can rearrange the product backlog so that critical new features are implemented at the expense of less important features, not your budget.

QUALITY

Agile integrates testing throughout the process. Consistently delivering tested software means higher overall quality and less time spent on QAing the full application.

RIGHT PRODUCT

Incremental releases let you test your product early and often. Even if you don't release it to the public, it's much easier to locate flaws and things that can be improved when you have an actual product to play with vs a series of designs.

TRANSPARENCY

Agile lets you see, feel and use a project consistently throughout the project. You don't see things in compartmentalized silos; you see how things work together.

AGILE MISCONCEPTIONS

IT'S DIFFERENT

"I've never used agile before and I'm scared it will be too hard to get my whole team on board with it."

We've heard it before. Too many times. And we realize agile may be new to you and your company. But while it might take a slight rewiring of how stakeholders think about projects from the onset-- and how designers and developers are used to working at your firm-- it quickly becomes apparent that projects consistently run smoother on agile. And better results are produced.

Plus, you likely can admit that waterfall isn't the perfect process. While it might feel like it is more under control because everything is mapped out from the beginning, projects undoubtedly take longer than they need to and cost more than they should. Waterfall also doesn't allow the flexibility to change things mid-project as new insights come to life.

FIXED BUDGET

I have a fixed budget that doesn't work with agile!

Au contraire! Nothing about agile says it can't meet a strict budget. Agile gives you dedicated resources. Generally, there is a fixed cost to a sprint that includes X team members. An agile team can estimate approximately how long it will take to complete the goals that you have outlined and that will give you a budget. As the project evolves and you choose to add a new feature, agile lets you drop a similarly sized feature so that you can stick to the initial budget.

IT'S UNPREDICTABLE

Agile can be unpredictable. But all projects are unpredictable. It is impossible to know exactly what your end users want. Agile embraces this unpredictability and leverages it to produce better results.

DEVELOPERS MAKE ALL THE FEATURES

Another common misconception of agile is that the developers get to choose what is important and what is implemented when. That could not be further from the truth. Before each sprint begins there is a comprehensive sprint planning meeting where all the key stakeholders determine which features will be implemented in that sprint. This meeting includes developers, designers, business people and anyone else involved in the product. Not just developers determining what to build willy-nilly.

IT DOESN'T CARE ABOUT THE LONG TERM

Some people believe that because agile focuses on short-iterative releases it doesn't take into account the long-term needs and goals. Agile actually benefits the long term. At a minimum, it is just a different means to get to the end. By having something that you can actually test earlier in the process, it lets you make better decisions for the long-term.

REQUIRES MORE TEAMWORK

Agile requires collaboration between designers and developers. Fortunately, most designers and developers love to collaborate. While there can be a bit more upfront work to get everyone on the same page, the end result is a better product, faster and for less money.

HOW AGILE WORKS

THE AGILE LIFECYCLE

There are many different flavors of agile. Ultimately, it is up to your team to come up with the best process for you. Generally they all follow a short life cycle, which repeats during each iteration. This guide focuses on Scrum, but many of the features are universal.

Scrum projects are broken down into short iterations (generally 1 - 3 weeks) called sprints. The lifecycle of each sprint includes:

Planning

Execution

Review

Rinse & Repeat

KICK-OFF / SPRINT PLANNING MEETING

Each scrum project begins with a kick-off meeting. The first meeting is generally the most extensive as the initial project backlog needs to be created and the project team introduced. Additionally, before each of the future sprints there is a sprint planning meeting.

First, the kick-off meeting. The kick-off meeting's goals are

- An overview of the project and the goals

- Who will be working on the project

- Determining the point person for client sign-off

- Creating project backlog (if you haven't already)

- Determining which features you will be working on

- Getting on same page

PROJECT BACKLOG

Behind every project is a project backlog. The project backlog is a list of all the product features generally defined by "user stories". User stories define everything potential users want to do on the site. They are defined for each of the user groups on the site and are structured like:

As a **[user type]** I want to **[do what]** so that I can **[purpose]**

For example:

As a **teacher** I want to **post test grades** so that I can **keep track of students**.

There are many tools to keep track of your project backlog, both analog and digital options. The important thing is that the backlog is always accessible and easy to track. In its most basic form it might be post-it notes on a wall. In fact, one of the best ways to create the initial project backlog is to write all of the user stories on post it notes during the kick-off meeting. Post-it notes are easy to rearrange so make a perfect analog solution to creating a backlog. If you prefer keeping things online, there are a number of tools listed in the Resources section.

After all the user stories have been dreamed up, they are ranked in order of priority. Part of this ranking is also grouping stories together. Some stories will naturally lend themselves to being built with others, which will expedite the process.

Remember that the project backlog is always fluid and never locked in. The project lead will be in charge of reprioritizing the backlog between sprints. And if new features are dreamed up or requested by users, they are encouraged to be added to the backlog. The one exception to the fluid backlog is during a sprint. While the sprint is in session, it is important to not add features. That keeps the team focused and makes sure that the project can be properly tracked.

FEATURE ESTIMATION

To be able to estimate what can get done per sprint and how long the full project will take, it is necessary to estimate how long each user story will take. Because one of the major challenges in development is accurately predicting how long things will take to get done, agile uses relative estimation.

Features are rated on a 1, 2 or 3 point scale. More precise estimation is more challenging and ends up less accurate. It is easy to compare things relatively on a scale of 3. And if something is particularly challenging that you don't think it fits within the "3 point" bucket, it should be broken down into smaller features that can each fit into the respective buckets.

There are a number of ways to handle feature estimation. It can be as simple as just talking about it or it can be slightly more complex using "planning poker".

It's also important to determine the sprint velocity of the team working on the project. That is how many "points" the team can complete per sprint. This velocity is averaged over time. And in typical average time value- the more sprints you do, the estimates and velocity become more and more accurate. That is to say that in some sprints you may not hit your goal number and other sprints you may exceed it. Over the course of a standard project, this averages out.

THE SPRINT

Agile projects are broken down into small, consistent time intervals. These intervals are referred to as sprints. They can be as short as a few days and generally are no longer than 3 - 4 weeks. We typically work in 1 - 3 week sprints depending on the extent of the overall project. During a sprint there is a dedicated team that includes designers, developers and business people working together.

Before each sprint, there is a sprint planning meeting (often combined with the sprint review meeting). This meeting determines what the goals are for that sprint. Based on the team velocity, a set of features are pulled from the top of the backlog. During the sprint, no features are added and the sprint goals don't change. The only exception to this is if the team finishes a sprint early. Client communication is generally limited to the daily standup results, but some firms allow for an open dialog via a chatroom.

DAILY STANDUP

Every morning of the sprint the project team gets together for a short (under 15 minute) meeting. This meeting takes place at the same time every day and includes everyone on the project. Everyone stands up for the meeting to keep everyone focused and to keep the meeting short. Often a timer is set so that the meeting does not run long.

Each person on the team is tasked to answer 3 simple questions:

What did you work on yesterday?

What are you working on today?

Do you need any help or are there any blockers in your way?

These three questions allow for complete transparency. Everyone on the team is in the loop, and the answers make people accountable for what they say they will deliver. The results of this meeting are typically shared with the client. This daily communication makes sure that if something is holding up the team, they can get a response quickly.

SPRINT REVIEW MEETING

At the end of every sprint something of value is produced. Something that theoretically could be launched. The sprint review meeting brings together the project team and other project stakeholders like the client to present the work that was completed.

The sprint review always starts with a functional demo. A conversation then takes place on how it can be improved and if there needs to be any reprioritizing of the project backlog. Then the team collectively plans out the next sprint.

KEYS TO SUCCESS

COMMUNICATION

Any project benefits from good communication. Agile projects are no exception. If you haven't run an agile project before, communication is especially important. Being kept in the loop about what is ahead of schedule and what is behind schedule can help alleviate concerns with unpredictability. A transparent process keeps people at ease and lets them focus on what is important: delivering the best possible product to their users.

DEDICATED TEAMS

Agile works best with a dedicated team of people who are willing and want to collaborate. The better the collaboration, the better the product.

GOOD PLANNING

For an agile project to succeed, it requires good planning. This doesn't mean planning everything down to the day or week like in a traditional waterfall project, but it does mean thinking through the project ahead of time: coming up with a robust project backlog and estimating features as best you can.

CONCLUSION

GET AFTER IT

Agile is meant to improve your life, not complicate it. It is meant to help you release your products faster, better and for less money. It is meant to be less risky than waterfall. It is meant to help teams work together better to generate their best work. Give it a whirl. Start with a small project that you can handle, and I bet you will enjoy it.

GLOSSARY

SPRINT

A sprint is a set amount of time where the work is accomplished.

PROJECT BACKLOG

The project backlog contains all of the user stories (or features) for the project ranked by priority. Each story has an estimated value of 1 to 3 points.

FEATURE ESTIMATION

Feature Estimation is the process of estimating how long each user story will take. You assign each story a relative point value of 1 to 3 points. If it is likely to take longer than 3 points, you break it down into smaller chunks.

PLANNING POKER (SCRUM POKER)

Planning Poker is a technique used for feature estimation. One issue with feature estimation is that by speaking the recommended point value, people may influence others in the group. Planning Poker solves this by using cards that are flipped over simultaneously.

DAILY STANDUP

Every day at the same time, the entire project team stands up and has a short meeting to review what was accomplished and what will be worked on.

USER STORIES

User Stories are one to two sentences that describe what a specific type of user needs to do to accomplish a goal on the site. They are formatted like:

As a [user type] I want to [do what] so that I can [purpose]

WATERFALL

Waterfall is a traditional type of project management that is sequenced. For example, once you complete the designs, you start development.

BURNDOWN CHART

A burndown chart is a graphical chart that is used to show the amount of work left vs time left.

VELOCITY

Sprint velocity is how much work a project team can get done per sprint. It is typically used to estimate how many features can be accomplished each sprint (based on the feature points)

SCRUM

Scrum is one flavor of agile development that focuses more on the management of the project as opposed to on what is accomplished.

SCRUMMASTER

The scrummaster is a member of the team that facilitates the meetings. Their goal is to remove any impediments that the team has.

RESOURCES

THE AGILE SAMURAI

pragprog.com/book/jtrap/the-agile-samurai

One of my favorite books on Agile: it provides not only a great overview, but also an indepth look at how to run agile properly and successfully.

THE AGILE MANIFESTO

agilemanifesto.org

The founding fathers of Agile

AGILE AND LEAN SOFTWARE DEVELOPMENT LINKEDIN GROUP

linkedin.com/groups/Agile-Lean-Software-Development-37631

Good discussions and Q&A

LEAN STARTUP

theleanstartup.com

Book and website dedicated to Lean thinking. Agile and the Lean methodology complement each other perfectly

PIVOTAL TRACKER

www.pivotaltracker.com

The popular tracking tool that helps you manage your Agile projects. The feature set is robust, but the User Interface could be improved to make this a go-to tool.

TRELLO

trello.com

A simple online tool to help you manage the project backlog. Trello is like a set of digital post-it notes that you can easily rearrange.

ASANA

asana.com

Another popular management tool. Asana offers a more thorough feature set than Trello, but the interface isn't as intuitive.



PHILOSOPHIE

www.philosophie.is

acts@gophilosophie.com

310.314.0011