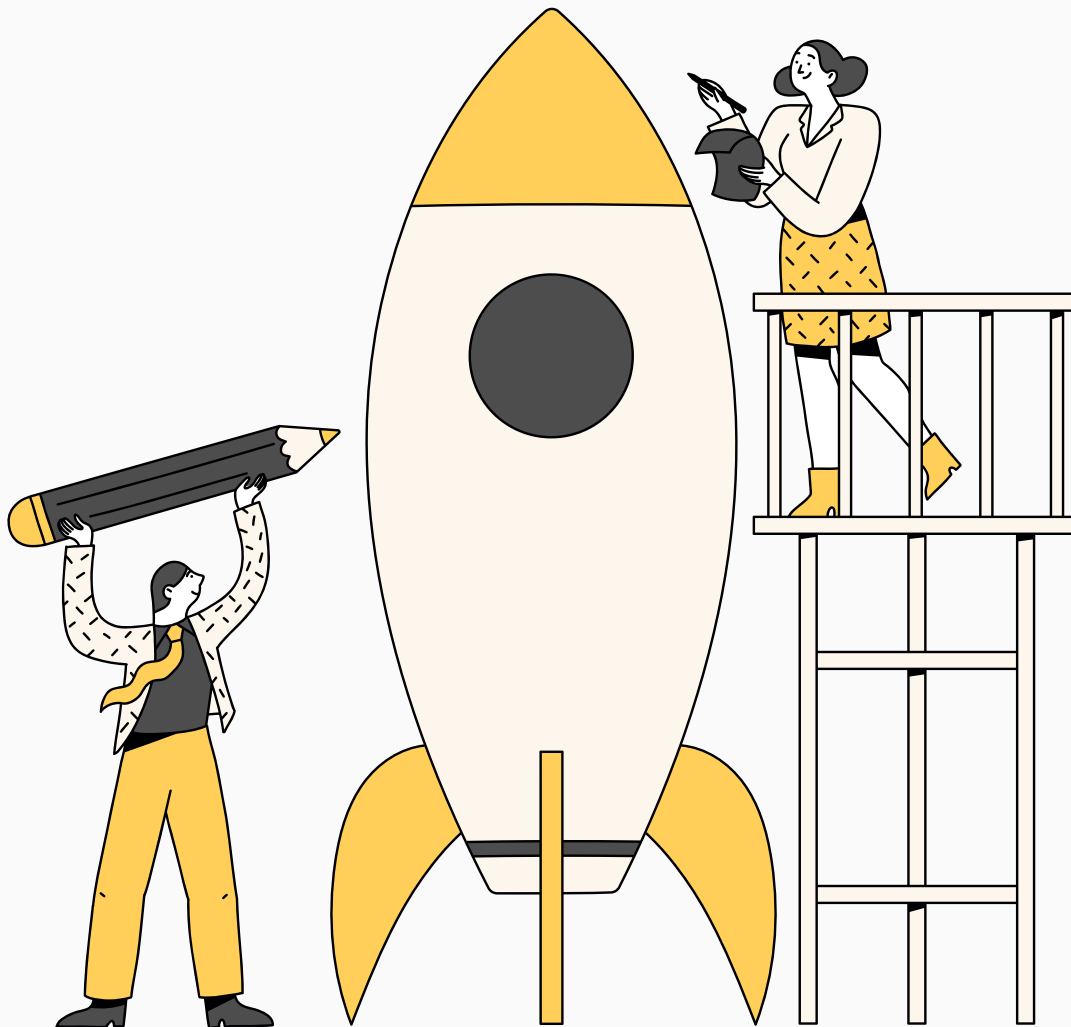


Agile Growth Plan



Agile Growth Plan Guide

Often, growth plans are lengthy documents with all manner of medium- and long-term planning, tracking, statements of purpose, categorization of work and so forth. The problem is that most people who develop this kind of plan spend a couple of weeks lovingly crafting it and then stick it in a drawer for six months. It hardly ever gets used to drive 1-on-1 conversations and it quickly drifts from the reality of the development that individuals experience day by day.

The Agile Growth Plan was developed as an antidote to this phenomenon. It is meant to be a living document stripped bare of everything non-essential so that it takes a few seconds to review it and keep it up-to-date. Additionally, it embodies a number of valuable ideas from product management, GTD and kanban.

Let's start with a sample of the end product...

Sample Growth Plan

Focuses

<p>Speak up more in meetings [1]</p>	<p>What: I'd like my voice to be more prominent when the team is discussing strategy.</p> <p>Why: I often have useful things to say but fail to offer them at the moment when they'd have the most impact.</p> <p>Components: Trust myself. Lean in. Doesn't have to be perfect. Read the audience. Come prepared. Think like a facilitator.</p>
<p>Learn how databases work [2]</p>	<p>What: Understand the various technologies for storing data, what their tradeoffs are, why we use the ones we do and how to develop software with specific databases in mind.</p> <p>Why: I've realized in the last 6 months that most of the really difficult problems we've faced have been around how we manage data. I want to help more effectively with those kinds of problems.</p> <p>Components: SQL, NOSQL, caching, indexes, contention, replication, storage latency, DB APIs.</p>
<p>Cook more for myself [3]</p>	<p>What: I've relied too much on take-out and pre-made meals for my food and want to have more independence and confidence in making my own meals.</p> <p>Why: I need to lose some weight and increase my overall health and my relationship with food is a big part of that.</p> <p>Components: Types of cooking (roasting, baking, boiling, grilling), common ingredients, quick meals, creating flavors, cutting techniques, presentation.</p>

Backlog

- ~~Read chapter 2 of "Everything's An Offer", [1]~~
- Read chapter 3 of "Everything's An Offer". [1]
- Ask Tim how he manages to be so succinct when he speaks in meetings. [1]
- Make a stir-fry and learn how to cut peppers and onions properly. [3]
- Install MongoDB at home and run through [exercise 0](#). [2]
- See what happens if I bake a batch of brownies low and slow. [3]
- Make a shortlist of coaches who can help me learn how to think like a facilitator. [1]

Icebox Of Future Focuses

- Understand how organizational design can affect team morale.
- Achieve basic proficiency in Rust.
- Become a good technical interviewer.

Creation Guide

Step 1: Generate Your List Of Potential Focuses

Start by taking a few minutes to write down your mental backlog of ways-you'd-like-to-develop. Each item on the list can be of any size and complexity (no standardization required). For example, you could have these two focuses sit quite comfortably next to each other...

1. Become an expert software engineer.
2. Understand our team's product manager better.

The first focus might seem enormous in scope compared with the second one but it's important to recognize that we never really know where our learning journey will take us. We can only guess at the amount of effort and growth associated with items like these. The key thing is that each focus implies some learning / growing that's important to you.

For the best results, ensure that the word pattern used for each focus...

- Starts with an active verb like "become" or "achieve" (avoiding words like "try").
- Is under 10 words (and is ideally closer to 5).

Step 2: Pick Your Top Three Focuses

Imagine your list of focuses as though it listed a set of mines wherein you plan to dig for gold. Trying to dig in all of them at once would make it very difficult to do so properly. Instead, Agile Growth Plans rely on the selection of three of those mines that are 'active dig sites' at any given time. So pick your top three.

The Agile Growth Plan framework is un-opinionated on how you should prioritize your growth. Maybe you want to choose the things that seem most urgent. Maybe you'd prefer to bias towards long-term value. It's possible you have a few that are mostly complete and you want to get them across the finishing line. This is entirely up to you.

As to the focuses you don't pick, it's a good idea to maintain an 'icebox' of future focuses that you can swap in when an existing focus feels like it has been 'mined out'.

Step 3: Unpack The Selected Focuses

In order to make our focuses useful, we need to unpack them a little. This involves constructing three sentences for each focus that address the following questions...

- What is this thing?
 - In seeking to describe what this focus is about, how might you go one level deeper than the few words in the summary phrase?
- Why is it important to you and how will you know when you're done?
 - What's really motivating this focus and (as such) how will you know when you've gotten 'enough' value out of digging here?
 - Note that this is not the same question (necessarily) as 'why is this important'. The 'you' is the key here.
- What components of the learning / growth seem most important?
 - What's the 'word cloud' of concepts / ideas / situations associated with this focus that you think you're going to need to pay attention to on your learning journey?

With the three focuses selected and unpacked in this way, you will have a 'personal charter'. This is a compact, high-level reminder of your ultimate growth aims.

Some people stop at this point, stick this charter to the back of their wallet / the bottom of their monitor and rely on their instincts to do the rest. For most people, however, this high-level charter needs to be married with a practical, granular list of specific things to achieve...

Step 4: Construct The Backlog

Build a concise, flat, prioritized list of things you know you'll be doing soon (2-4 weeks) in service of your focuses. In order to avoid procrastination (combining principles from both agile development and GTD), the list itself should be...

- Flat: do not divide the list into groupings by focus etc.
 - When you use the list, you will want to avoid the mental speedbump of having to choose a section and then choose an item.
- Prioritized: how you prioritize is up to you.
 - When you use the list, you will always pull from the top.
- Short but not too short: aim for about 10 things.
 - If you're dipping below 5, you're not showing the list enough love.
 - If you're trending towards 20 you've likely got too much upfront-design going on.

Additionally, each item on the list should be...

- Related to a focus: indicate this by tagging each item with the number of the focus it relates to.
 - This is an important cross-check that your bottom-up and top-down thinking are in sync.
 - It also gives you a quick indication of coverage (and whether some of your focuses are starved for attention).
- Unambiguous: requiring no more decision-making in order to get started.
 - Procrastination is very easy if you look at your next backlog item and have to start with the question “what do I actually need to do here?”.
 - Rather than “read a book on the Ruby programming language”, you want something more like “read chapter 3 of Practical Ruby For System Administration”.
 - To extend this example, if you didn’t know what book to read then the task would become “ask 3 colleagues for a Ruby book recommendation”. In other words, you would work backwards to the next unit of unambiguous work.
- Bitesize: taking 15-60 minutes.
 - The main thing is to size tasks to fit in whatever time block you carve out in one, uninterrupted sitting.
 - The Pomodoro method can work well here.
- Deletable: nothing on this list is sacred.
 - This list isn’t supposed to be an aspirational account of the things you think you’re supposed to be doing.
 - If you start jumping past items to do others, stop and think about deleting / reprioritizing as needed.
- Learning-centered: don’t let this turn into a simple todo-list.
 - The point of this list is to drive learning / growth rather than delivery.
 - To keep yourself honest here, experiment with using this phrasing for some / all of the items: “In order to learn about X, I will Y”.

Step 5: Use It Actively

The only thing worse than constructing a traditional, long-form growth plan and then leaving it in a drawer for a year would be to construct an Agile Growth Plan and do the same thing with it.

Every day, take 2 minutes to glance at your plan and...

- If something's been done, strike it through (or move it to a 'done' section if you don't like crossing things out).
- If the backlog is starting to look a little thin (5 items or fewer), add some new ones to get you back to around 10.
- Remember to think about the 'why' of your focuses and notice if it's starting to get hard to find value in one or more of them.

Every week (ideally as part of 1-on-1s you may have with a mentor / coach)...

- Take stock of all the items that have been completed since last time and ask yourself whether this 'velocity' feels right / is normal for you. If your velocity has dropped noticeably, ask yourself whether you're spending enough time on your development or whether the focuses and backlog truly reflect your goals.
- Once you've processed the completed items, either delete them or archive them off somewhere if you'd like to track them historically.

Every month (again, ideally as part of those 1-on-1s)...

- Consider your focuses. Do they still best-capture the growth you both want and need to reach for? Have you had any feedback that might affect the answer to that question?
- Don't forget your 'icebox' of future focuses. Maybe it's time for a swap?

FAQs

Where should this growth plan live?

Where you keep it and how you share it are ultimately up to you as it's your document. Since you'll want to use it in your 1-on-1s, we recommend keeping it in a tool that allows you to share it easily with your mentor / coach and that allows them to collaborate on it effectively. A Google Doc, for example, would be a good choice. We've also seen teams benefit from sharing their plans in a common area, allowing them to find commonalities in their learning journeys.

What happens if the focuses that feel right to me don't line up well with my current job?

In the end, this is a *personal* growth plan. You should take account of all the input you're getting when you think about your focuses. That said, If it then turns out that the areas in which you most want to grow have very little to do with your day job, that's an important signal to consider in reflecting on your future career choices.

What if my mentor / coach disagrees with the value of my focuses?

A big part of your mentor / coach's role in this process is to offer honest opinions on whether they think a particular focus will yield the adaptation you need. You're holding the steering wheel but they're often holding the best map. So listen openly and weigh their advice carefully.

Can I add a section to track XYZ?

It's your plan so add any features you need. Just make sure that the focuses and backlog are the first things you see when you open the document. They should always be the heart of the document.

Are there any non-document-y ways to model this plan?

Because of the backlog-centric nature of this design, it is easy to model your plan using a personal kanban board (if that's your jam). Some people's Agile Growth Plans live entirely in Trello.

What if I need to build a new habit? How can I model that using this approach?

If you're trying to build a habit, include a backlog item with the action / activity in question and a countdown that can be updated: "At the next 10 stand-up meetings, remember to give at least one sentence of context for my work". With such an item, each time you do this, you'll be able to decrement the number by one (so 10 becomes 9 and so forth). This approach is useful as it creates an end-state (when you hit 0) for what would otherwise be an open-ended process.

Are there any ways to increase the impact when whole teams are using this approach?

If you're mentoring multiple people, consider helping them use common phrases to describe similar focuses - it will make it easier for you and them to identify learning communities amongst that group. For example, if a few of your mentees want to grow their ability to help others to perform better, you might encourage them all to have the focus "set clear expectations" - making the shared goal obvious.