

# AGILE

## ROADMAPPING

A not-boring guide for  
real-world product managers

INCLUDING 4 READY-TO-USE  
**TEMPLATES**



## About Roadmunk

Roadmunk was inspired by a problem our founders experienced personally: there was no simple way for product managers to build, share and align on the roadmap. As we got to know our users, we discovered that roadmapping is not one-size-fits-all. Companies make roadmaps of all shapes, sizes and types—it's a powerful and essential tool for alignment. Our goal is to make strategic roadmapping quick, effective and collaborative across an entire organization.

[Learn more](#)

Button not working? [Click here.](#)

# Introduction

Agile is basically the status quo for tech companies. But in reality, it's rare to find agile teams that are ... well, *actually agile*. Too many real-world constraints prevent teams from embracing a pure agile mentality. Since starting Roadmunk, I've noticed agile product managers face similar struggles—particularly when creating a roadmap.

A giant question mark hovers around agile roadmaps. Can roadmaps play an effective role in agile? Aren't the two concepts counterintuitive? Isn't an "agile roadmap" an oxymoron?! So we set off on a mission at Roadmunk to find some answers.

The goal: discover what place roadmapping has in agile. How are real-world organizations using agile, and where do their roadmaps fit? Speaking to dozens of agile PMs, we've compiled our research together to present this guide to agile roadmapping. Here's what you'll find:

- Real-world case studies of different agile roadmaps being built at Shopify, Frank + Oak, Lola Travel and more
- Insights from expert agile product managers from Electronic Arts, Expedia, Goodreads and more
- A guide for planning your agile roadmap
- Best practices for creating an agile roadmap
- 4 ready-to-use agile roadmap templates that you can make your own

This ebook is here to provide the insights and advice you need to approach roadmapping confidently—no matter how your team operates in agile.

Happy reading (and roadmapping)!



**LATIF NANJI**  
CEO AT ROADMUNK



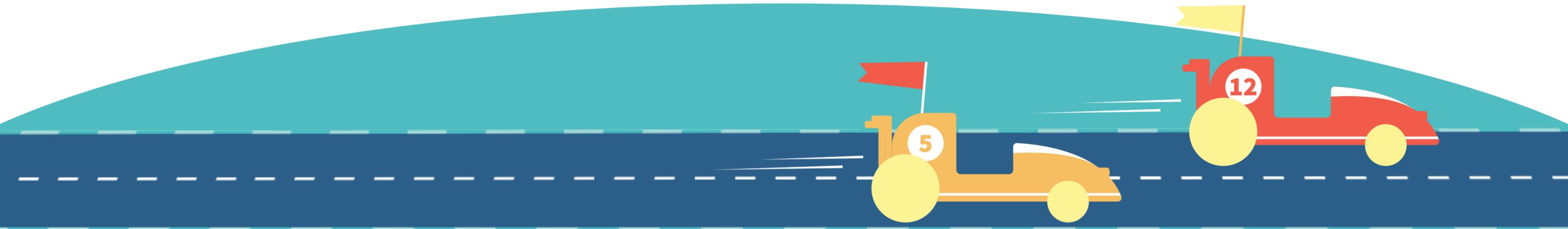
Start building your own  
roadmap with over 20  
customizable templates

Explore our template library

Button not working? [Click here.](#)

# Table of Contents

CH		PG
1	Can agile and roadmapping work together?	4
2	What you need to know before creating an agile roadmap	8
3	How to visualize your agile roadmap	11
4	Guidelines for your agile roadmap	18
	Conclusion	21
	Templates	22



## CHAPTER 1

# Can agile and roadmapping work together?

As a methodology, agile says: forget about planning on timelines—they're unhelpful and hard to meet! On the other hand, roadmaps help you visualize plans, often on a timeline. **So is the idea of an agile product roadmap counterintuitive?**

Let's look at the facts. Agile indicates that teams are very flexible and willing to change their plans on a daily or weekly basis. **That means project deadlines and dates are pretty insignificant.** Yeah, you can add dates to your plan, but if you're being *truly* agile (i.e. flexible), expect those dates to change.

Meanwhile, a roadmap usually asks: "Where the heck are we going to be in x amount of time?" **Not everyone roadmaps along a timeline,** but a roadmap almost always looks to the future beyond the day-to-day and week-to-week, even if not visually presented on a timeline.

So one concept (agile) sheds the constraints of time and another (roadmapping) often appears inherently time-based. Seems pretty counterintuitive. **But we have to respectfully disagree.** An agile product roadmap is a valuable tool and quite frankly, the concept of roadmapping fits rather nicely in agile environments. Here are some reasons why a roadmap does work for agile.

Working in agile doesn't mean you no longer have to plan ahead. It just signifies a mindset that lets you change direction faster based on incoming feedback from your market, customers and stakeholders.



**CATHERINE SHYU**  
PRODUCT MANAGER AT  
FULLCONTACT INC.

## Roadmaps change as often as the agile world

Agile is an umbrella term. You've got your scrum shops, your extreme programming teams, your kanban processes. They all fall under "agile." Each of these methodologies have their own nuances and ways of operating, but ultimately they all share a few major themes:

- You're operating in short cycles
- Change comes quickly, so you *always* have to be prepared for it
- There is a ton of feedback because you're operating in cycles that include checkpoints for evaluating your work

These themes are all rooted in the core idea that when functioning in agile, **you must be flexible.**

Here at Roadmunk, we've stressed many, *many* times before, a roadmap is a statement of intent. *It's not a literal roadmap.* Your roadmap should not be an amalgamation of hard deadlines that must be met. It should be an incremental plan that openly embraces change. **Plans can and will alter, and as a result so will your roadmap.**

This flexibility of a roadmap pairs extremely well with the nature of agile methodologies. Roadmaps change frequently, allowing them to easily fit into agile settings—which also get reoriented often.



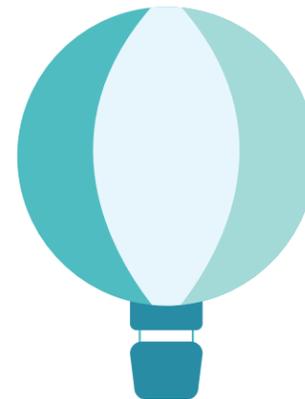
## Roadmapping and agile both cut the bullshit

Agile is all about the hard truth—whether you like it or not. Agile provides realistic versions of how projects will play out. If a project is derailing, you're told straight up it's derailing. This transparency is part of why agile has picked up so much steam: it cuts the bullshit about what can or cannot be delivered.

Just as agile environments function on transparency, **roadmaps create transparency.** You build a roadmap to clarify where the hell

you're headed. It's a communication asset designed specifically to align various parties on the what, when and how of your project. And when that plan changes, you present a shiny, updated version.

A roadmap's function (**showing transparency**) directly ties into one of the major motivations of adopting an agile methodology (**operating with transparency**). They're not in conflict; they fit nicely together.



“An agile product roadmap is done with the intention of change being planned into the document itself and into the process.”



**AHSAN NANJI**  
DIRECTOR OF PRODUCT MANAGEMENT  
AT ELECTRONIC ARTS

## You can't escape long-term planning

One thing we've noticed when speaking to various PMs is that many of those who claim to operate in agile actually aren't. Okay, they're operating in some form of agile, but waterfall elements are still mixed in—so it's not purely agile. The reason: **as your company grows, the harder it gets to ignore long-term planning.**

Plenty of startups have successfully adopted an agile mentality that dismisses long-term timelines. But as these companies expand, stakeholders increase, user needs intensify and communication needs arise. PMs start juggling way more variables, which requires a **deeper degree of planning.**

Your dev and product teams may function in speedy cycles, but the rest of your growing company probably doesn't. **Other stakeholders don't give a damn about your daily or weekly psychology.** They want longer-term visions to plan hiring, budgeting and more. A roadmap addresses long-term goals and tackles planning pressures that inevitably creep up.

“What it comes down to is that companies aren't as agile as they say they are. I've heard lots of different terms where it's always a hybrid. Where the agile aspects come into play is within a project; how you're executing.”



**BREANNA HUGHES**  
SENIOR PRODUCT MANAGER AT  
LEAGUE INC.

# Mythbusters:

## THE AGILE ROADMAP EDITION



**Myth:** Working in agile means planning is not necessary.



**Fact:** People like to think you wing things in agile, because plans change so often. Say sayonara to that thinking because planning is actually critical in agile. You actually end up updating your plan more often.



**Myth:** Timelines don't exist in agile.



**Fact:** Wrong! Yes—time constraints might not be as demanding as they are in waterfall. Yes—you might not be as concerned with dates. But no—you still need to consider time at *some point!* Agile teams can't escape long-term planning, so they also can't escape timelines altogether.



**Myth:** Creating a roadmap in agile means you have to incorporate dates/deadlines in your plan.



**Fact:** Nope. Not at all true. There are actually a few ways to create an agile roadmap that don't incorporate any strict timelines, dates or milestones at all. Buh-bye myth.

# Why you need a roadmap, especially for your agile environment

We've debunked the argument that roadmaps don't *belong* in agile environments, but why do you actually *need* one? We'll get into what an agile roadmap looks like in Chapter 3, but for now here are some reasons for the not-yet-convinced.

## Align better with your “makers”

When you say you're operating in agile, you're most likely referring to your “makers.” Your **developers**, **product team** and **designers**—the ones who assemble your product piece-by-piece. Long-term plans are not pressing priorities for makers. They're more concerned with what they can accomplish in the next hour, day or week.

By presenting an agile roadmap, you signal to your makers that you understand how they function. You're committing to short-term priorities, but also pointing out that the future is “fuzzy.” It'll be much easier to align with your makers if you position your roadmap **as a living document that aligns short-term priorities and embraces the future's unpredictability.**

## More cross-company cohesion

While your makers embody agile principles, other teams probably don't. Sales is working in lengthy cycles, marketing is concerned with the next release, customer success has quarterly goals to hit. **It's a mishmash of timeframes across the company.** And guess who has to align these different psychologies of time? You got it—the PM! (Yay...)

An agile roadmap better aligns your makers with teams working along different timelines. Teams working under more traditional time constraints may have a lot of questions for agile teams. “Why isn't this done?” “When should we expect this?” A roadmap adapted for your agile environment communicates realistic goals to internal stakeholders prying for answers.

## Managing ALL the expectations

The beauty of the agile product roadmap is that it **gives a PM more flexibility.** Not only does it encompass the intent of a typical roadmap, it also captures the flexibility of agile methodologies. A PM can whip out their roadmap and easily set expectations. “This isn't set in stone and it will probably change, but it's what we're proposing.”

With a baseline understanding established from the start, PMs' lives are made a hell of a lot easier in the long run. When unavoidable changes happen and a PM must make a tough decision, people will cause less of a fuss because, well, their expectations were managed at the start. This reduces conflict and prepares team members for change.

“I know it'll always be challenging to evangelize what agile is to non-technical teams, but it comes down to incorporating those other teams to understand agile processes.”



**SARAH PYO**  
PRODUCT MANAGER AT  
EXPEDIA



**ARAN RASMUSSEN**  
DIRECTOR OF PRODUCT MANAGEMENT AT FRANK + OAK

“Structure your roadmap in a way that everyone's clear on where there is and isn't flexibility.”



## CHAPTER 2

# What you need to know before creating an agile roadmap

Before we get to what an agile roadmap looks like and how you can build one, let's talk about what you need to know beforehand.

Behind every effective agile roadmap is a well-thought out plan. As true (and cheesy) as that statement is, it's also a bit funny since a roadmap is meant to be a (flexible and transparent) plan that aligns stakeholders across your organization. So what we're saying is: **you need to plan for your plan.**

You might be asking: why plan for an agile roadmap when it will ultimately change? You're not wrong—your agile roadmap should be a statement of intent that updates often to reflect changes brought on by your agile setting, organization and market; NOT because you poorly planned your roadmap. Planning can prevent unnecessary changes and even complete overhauls of your agile roadmap.

Roadmap planning is a key step in the workflow of all the agile PMs we've chatted with. So before creating your agile roadmap, **here are five steps you should take for effective agile roadmap planning.**

“Even the best laid plans go awry and you should be prepared to scrap parts of it. Knowing that helps keep the balance in not over-planning for your roadmap.”



**JESSICA JOHNSON**  
SENIOR PRODUCT MANAGER AT  
GOODREADS

1

## Set your goals

Your agile roadmap **communicates and aligns stakeholders**, but *what* are you addressing? Even if you have a list of impressive features to release, the big picture must be considered. Your objective might be achieving product-market fit, or launching “sticky” features. Whatever the goal, your roadmap should convey the plan for achieving your high-level strategy.

Without established objectives, expect stakeholders to walk away from your roadmap with different priorities (the TOTAL opposite of what you want). **Your roadmap should always tie directly back to your primary goals** so stakeholders are clear about your product strategy. Ask yourself if the document speaks to your **company and product vision**.

2

## Evaluate your resources

If you have ten developers, but two are on vacation and one is shared by marketing, how much can you ship in your next sprint? The answer: probably not as much as you could with a fully available team of ten developers. Planning for your agile roadmap should address your **available resources and their constraints**.

Another thing to plan against: **the speed at which resources operate**. You’re going to get pushback on how fast (or slow) your team is delivering on your plan—even in agile. Plan a cycle length that works best for your team—especially if you’re representing time on your roadmap. Determining your team’s speed will let you develop a plan grounded in reality.

3

## Check in with your users

It boggles our minds when organizations build roadmaps based solely on quantitative market data. User needs go much deeper than statistical research. **Speaking first-hand to customers during your planning** provides insights into disregarded investments. Before committing to anything on your roadmap, it makes sense to understand if your users, you know, actually want these things.

Additionally, user research makes a really, really strong case when getting buy-in. User research shows stakeholders that your agile roadmap isn’t just a random, aimless plan you pulled out of thin air, but one developed from **authentic user needs**. It’s pretty hard to refute real user needs.

4

## Determine your building blocks

Next up: establish your **major themes**, and the features that fall under each of them. Then break them down into **epics and user stories**. From there develop high-confidence estimates on your features, epics and stories, and identify any **dependencies** that could hold you back. Hack away until you have all the building blocks you need *before* assembling your roadmap.

If this seems like a lot of work *before* creating an agile roadmap, we say: planning for these elements will prepare you to address (and defend) your choices when changes do pop up. And if changes are needed, you can speak to impacts and trade-offs that must be made.

5

## Hold planning sessions

This step isn’t so much isolated, but more ongoing and works in conjunction with the previous steps. **Make agile roadmap planning a team effort** by putting your whole product team in one room. This is your opportunity to really understand together your use cases, discovery work and objectives before making any roadmap investments.

Think of it as a way to get buy-in for the plan for your plan (again—strange, we know). Early buy-in during your planning stage will be beneficial when getting **official buy-in** and public affirmations from your stakeholders.



## BONUS QUESTION!

If you want to get *reallyyy* specific, also ask yourself what key performance indicators you want to focus on. Planning your main KPIs will allow you to create a roadmap that explains how you’ll hit those targets.

# A Handy Checklist

## FOR YOUR AGILE ROADMAP PLANNING

To give some structure to this process, find below a handy checklist meant to make the agile roadmap planning process just a tad easier. Follow this checklist for steps you should take, as well as questions you should be asking and key takeaways from each step.

### Step 1:

#### SET YOUR GOALS

##### QUESTIONS TO ASK

- What is the big-picture goal we are trying to achieve?
- What are the KPIs I want to address?

##### KEY TAKEAWAYS

- Company vision
- Product vision
- Key performance indicators

### Step 2:

#### EVALUATE YOUR RESOURCES

##### QUESTIONS TO ASK

- What resources do I have available to me?
- At what velocity can our resources operate?

##### KEY TAKEAWAYS

- Schedule of team

### Step 3:

#### CHECK IN WITH YOUR USERS

##### QUESTIONS TO ASK

- What features do my users actually want?
- What qualitative data can we gather on our users?

##### KEY TAKEAWAYS

- Quantitative user data on the market
- Qualitative user data
- Use cases

### Step 4:

#### DETERMINE YOUR BUILDING BLOCKS

##### QUESTIONS TO ASK

- What are the big initiatives I want to address on this roadmap?
- What estimations can we have on these items?

##### KEY TAKEAWAYS

- Themes
- Features
- Epics
- User stories

### Step 5:

#### HOLD PLANNING SESSIONS

##### QUESTIONS TO ASK

- Do we all understand the key takeaways from steps 1-4?

##### KEY TAKEAWAYS

- Buy-in from product team

CHAPTER 3

# How to visualize your agile roadmap



Let's get to the meaty stuff now: [what does an agile roadmap actually look like?](#)

Through all our chats with agile PMs, we've discovered there is no one way to create an agile roadmap. Swimlanes, timelines, dates, no dates, a myriad of colours—there *might* be too many options, honestly. However, we couldn't help but notice a few agile roadmapping trends that kept popping up. From these patterns, we identified four powerful ways to visualize your agile roadmap:



Theme-Based



Sprint



Fuzzy Time



"Agile-ish"

Implemented and executed by real-life companies, these agile roadmaps are best suited for different agile scenarios. We asked [six awesome product managers](#) (whose agile roadmapping processes align with at least one of the four buckets) to share how they're actually using agile, how they visualize their agile roadmap and why their chosen roadmap works best for them.

“The roadmap itself is a secret recipe that we have. It's the secret recipe of how we build products.”



**CASEY MCKINNON**

VP, PRODUCT AT ECOBEE

# Theme-Based Roadmap

Since agile is known for **shedding constraints of time**, it's no surprise many agile PMs create roadmaps absolutely void of time. You'll see their roadmaps organized by theme—not dates or timelines.

This is typical amongst your 'pure' agile-ists, or smaller startups that find it more helpful to ignore time constraints. But there are exceptions. For instance, [Avrum Laurie](#), VP of Product for the DIY accounting software [FreshBooks](#), approaches roadmapping with a 'no-date' mentality.



	INFRASTRUCTURE	NEW FEATURES	STICKINESS	IMPROVEMENTS
WEB APP	<ul style="list-style-type: none"> <li>Metrics</li> <li>Automated Tests</li> </ul>	<ul style="list-style-type: none"> <li>Feature A Scope</li> <li>Integrated Prototype</li> <li>Undo Function</li> </ul>	<ul style="list-style-type: none"> <li>Onboarding Flow</li> <li>Desktop Delighter (TBD)</li> </ul>	<ul style="list-style-type: none"> <li>Dialogue Styling</li> <li>Import Engine</li> <li>Update Navigation</li> </ul>
MOBILE APP	<ul style="list-style-type: none"> <li>Demo Staging</li> <li>Regression</li> <li>Back-End Analytics</li> </ul>	<ul style="list-style-type: none"> <li>Feature Requirements</li> <li>Front-End Prototype</li> <li>Feature B Scope</li> </ul>	<ul style="list-style-type: none"> <li>Reward (Progress Bar)</li> <li>Mobile Delighter (TBD)</li> <li>Gamification</li> </ul>	<ul style="list-style-type: none"> <li>Performance Overhaul</li> </ul>

“We don't really do anything like a yearly roadmap. I've tried a more traditional roadmap, but it's kind of just fortune telling,” says Avrum. “You look back and see what was delivered, tossed, changed entirely. I'd say 95% of it falls into the last two buckets. It wasn't particularly productive and worth over-planning that far.”

Instead of focusing on longer timelines, Avrum hones in on themes and plans in quarters—like so:

- Avrum's team establishes yearly themes. “We have a set of strategic imperatives that are described in terms of a **product or customer-theme**,” says Avrum.
- Once yearly themes are established, the team turns them into quarterly OKRs: a popular technique that lists out an organization's objectives and (usually) 3-4 key measurable results for each objective.
- At a quarterly level, a more traditional roadmap may come into play. But don't expect time to be represented. “We don't typically prescribe very specific timelines to things,” says Avrum.

Avrum finds a theme-based approach keeps his team's expectations in-check. “At the year-level we ask: **what themes do we want to focus on?**” says Avrum. “At a quarter level, a more typical roadmap comes into play. A simple roadmap (without specific timelines) sets more realistic expectations and de-risks things.”

Working from a top-down level, FreshBooks' product team lets themes (rather than time) drive planning—and by extension, their agile roadmap. This theme-based approach to roadmapping caters to teams embracing the 'dateless' mentality of pure agile.



**Dates aren't your thing?  
Build a theme-based roadmap**

**Try this template**

Button not working? [Click here.](#)

# Sprint Roadmap

While there is an appeal to being dateless, it's not realistic for most agile teams. Long-term planning becomes necessary, different stakeholders need appeasing, and, if your team is growing, good luck balancing multiple priorities with zero timelines in mind!

Real-world demands spur many agile teams to shift to the more structured **scrum** model. Working in short **one- or two-week sprints** allows teams to be agile, while adhering to a loose concept of time. If you're an agile team that likes the structure of scrum, creating an agile roadmap organized around your sprints may be the best way to meet your roadmapping needs.

	1.1	1.2	1.3	2.1
NEW FEATURES	<ul style="list-style-type: none"> <li>Feature Requirements</li> <li>Front-End Prototype</li> </ul>	<ul style="list-style-type: none"> <li>Feature A Scope</li> <li>Archiving</li> </ul>	<ul style="list-style-type: none"> <li>Integrated Prototype</li> <li>MVP Requirements</li> </ul>	<ul style="list-style-type: none"> <li>Feature B Scope</li> <li>Search</li> </ul>
STICKINESS	<ul style="list-style-type: none"> <li>Status Updates</li> </ul>	<ul style="list-style-type: none"> <li>Reward (Progress Bar)</li> <li>Onboarding Flow</li> </ul>	<ul style="list-style-type: none"> <li>Mobile Delighter (TBD)</li> </ul>	<ul style="list-style-type: none"> <li>Gamification</li> </ul>
INTEGRATIONS	<ul style="list-style-type: none"> <li>Slack</li> <li>Salesforce</li> </ul>	<ul style="list-style-type: none"> <li>JIRA</li> </ul>	<ul style="list-style-type: none"> <li>Trello</li> <li>HubSpot</li> </ul>	<ul style="list-style-type: none"> <li>Zendesk</li> </ul>
INFRASTRUCTURE	<ul style="list-style-type: none"> <li>Demo Staging</li> </ul>	<ul style="list-style-type: none"> <li>Regression</li> <li>Automated Tests</li> </ul>	<ul style="list-style-type: none"> <li>Design Process</li> </ul>	<ul style="list-style-type: none"> <li>Back-End Analytics</li> </ul>



ELLEN CHISA

**Ellen Chisa**, VP of Product at the personalized travel company **Lola Travel**, recently shifted to sprint-based roadmapping. Ellen says, “We started and it was literally me working by myself and then we grew to 50 people. People felt like they were missing out in not being able to collaborate with more of their engineering colleagues, so we moved to two-week sprints.”

**Sprints inherently suggest there is a timeline—without setting a hard deadline.** Working in sprints means working your ass off to complete

the priorities for your specified timeframe. Once that cycle is finished, on to the next cycle. Teams are aware of their time box and work within it, so dates don't have to be outlined on your agile roadmap.

Ellen's roadmap perfectly embodies this structured—but not so structured that you need hard dates—agile mentality. “Basically, we have a backlog view that's a list of tasks. We assign tasks into sprints and name sprints by whichever team is working on them,” says Ellen. “We prioritize things we're working on—or the next most important things—towards the top and assign them into specific sprints.”



	1.1
NEW FEATURES	<ul style="list-style-type: none"> <li>Feature Requirements</li> <li>Front-End Prototype</li> </ul>
STICKINESS	<ul style="list-style-type: none"> <li>Status Updates</li> </ul>

Ellen maps her sprints against scrum teams, but **you can map against themes, features, or anything else**. This format works for Ellen because as she says, “Travel’s a pretty complex industry; every team operates a little autonomously. One team focuses on managing our supply of hotel and flight information. Another team focuses on tools for travel agents. They’re somewhat connected, but more separated—which is why this format works for us.”

Ellen’s roadmap doesn’t incorporate dates, but by allocating all items and tasks to two-week sprints, the team instills their agile process with *some* structure—without getting *too* rigid.

**Add structure (but not *too* much) with a sprint roadmap**

**Try this template**

Button not working? [Click here.](#)



## Fuzzy Time Roadmap

Fuzzy time: one of the most popular ways to organize agile roadmaps. Rather than stating explicit dates, fuzzy time roadmaps include loose time buckets like **In Progress**, **Future** and **Completed**. Since growing agile teams ultimately cannot ignore time, a “fuzzy” approach is a compromise for teams not wanting to be held to specific dates, but still needing rough time frames.

	IN PROGRESS	SOON	FUTURE	COMPLETED
NEW FEATURES	<ul style="list-style-type: none"> <li>Undo Function</li> <li>Feature B Scope</li> <li>Search</li> </ul>	<ul style="list-style-type: none"> <li>Integrated Prototype</li> <li>MVP Requirements</li> </ul>	<ul style="list-style-type: none"> <li>Archiving</li> <li>Front-End Prototype</li> </ul>	<ul style="list-style-type: none"> <li>Feature A Scope</li> <li>Feature Requirements</li> </ul>
STICKINESS	<ul style="list-style-type: none"> <li>Gamification</li> <li>Desktop Delighter (TBD)</li> </ul>	<ul style="list-style-type: none"> <li>Mobile Delighter (TBD)</li> </ul>	<ul style="list-style-type: none"> <li>Reward (Progress Bar)</li> <li>Onboarding Flow</li> </ul>	<ul style="list-style-type: none"> <li>Status Updates</li> </ul>
INTEGRATIONS	<ul style="list-style-type: none"> <li>Zendesk</li> <li>Marketo</li> </ul>	<ul style="list-style-type: none"> <li>Trello</li> <li>HubSpot</li> </ul>	<ul style="list-style-type: none"> <li>JIRA</li> <li>HipChat</li> </ul>	<ul style="list-style-type: none"> <li>Slack</li> <li>Salesforce</li> </ul>
INFRASTRUCTURE	<ul style="list-style-type: none"> <li>Design Process</li> <li>Metrics</li> </ul>	<ul style="list-style-type: none"> <li>Automated Tests</li> </ul>	<ul style="list-style-type: none"> <li>Regression</li> <li>Back-End Analytics</li> </ul>	<ul style="list-style-type: none"> <li>Demo Staging</li> </ul>

Adopting a kanban-style model to keep up with speedy market demands, [Aran Rasmussen](#), Director of Product Management at the fashion-forward online retailer [Frank + Oak](#), follows an agile roadmap completely based on fuzzy time.



ARAN RASMUSSEN

“We’ve moved over to a kanban-style environment and away from two-week sprints. I find it to be a better agile framework for companies that have

a product already live and in the world,” explains Aran. “Especially, I find in retail, where things change fast and we need to be ultra-responsive to marketing needs and opportunities.”

Aran oversees three website development teams for front-end, back-end and mobile. These teams are *always* deploying. A fuzzy time roadmap helps him manage multiple priorities—all while still having a realistic and flexible outlook on task timelines.



MACKENZIE EWING

The professional networking startup [Ten Thousand Coffees](#) also turns to fuzzy time. Product Manager, [Mackenzie Ewing](#), explains how agile roadmaps with loose time structures bring clarity to their intense development process.

“Our roadmap has ‘Backlog,’ which is for items not prioritized or ready to be actioned. ‘To Do,’ which is for

items actually ready to go. ‘In Progress’ to track what devs are working on. We have ‘Under Test’ to say what is ready and in pre-production. And then we have ‘Done.’ And that’s obviously: it’s in production,” says Mackenzie.

Just like Aran, Mackenzie finds fuzzy time works best for his team because it **lays out really friggin’ realistic expectations**. “At a high level, it gives us a framework and cadence to go forward. ‘This is what I have to execute and these are the days I have.’ Having that expectation laid out is really important to get your team bought-in to the larger vision of how we’re getting from point A to B.”



NEW FEATURES

NOW	SOON	LONG-TERM	DONE
Undo Function	Integrated Prototype	Archiving	Feature A Scope
Feature B Scope	MVP Requirements	Front-End Prototype	Feature Requirements
Search			

We asked Aran for a clearer idea of how his agile roadmap incorporates fuzzy time. “I have a swimlane and three columns—one for each team. On the left, I have time buckets. I’ve got my ‘Now,’ ‘Soon,’ ‘Long-Term’ and ‘Done’ rows. Some things have deadlines and some don’t. I try to be as real as possible with everybody about those deadlines.”

Track progress, not deadlines, using a fuzzy time roadmap

Try this template

Button not working? [Click here.](#)

# “Agile-ish” Roadmap

And then there are agile teams bound by time. We’ve deemed them “agile-ish.” The perfect moniker because, while most agile teams believe they’re lean, mean pure agility machines, many teams have to incorporate waterfall processes thanks to strict timelines (even if their development team is working in pure agile).



	Q3 2017	Q4 2017	SOON	FUTURE
NEW FEATURES	<ul style="list-style-type: none"> <li>Feature Requirements</li> <li>Front-End Prototype</li> </ul>	<ul style="list-style-type: none"> <li>Feature A Scope</li> <li>Archiving</li> </ul>	<ul style="list-style-type: none"> <li>Integrated Prototype</li> <li>MVP Requirements</li> </ul>	<ul style="list-style-type: none"> <li>Undo Function</li> <li>Feature B Scope</li> <li>Scope</li> </ul>
STICKINESS	<ul style="list-style-type: none"> <li>Status Updates</li> <li>Onboarding Flow</li> </ul>	<ul style="list-style-type: none"> <li>Reward (Progress Bar)</li> </ul>	<ul style="list-style-type: none"> <li>Mobile Delighter (TBD)</li> </ul>	<ul style="list-style-type: none"> <li>Gamification</li> <li>Desktop Delighter (TBD)</li> </ul>
INTEGRATIONS	<ul style="list-style-type: none"> <li>Slack</li> <li>Salesforce</li> </ul>	<ul style="list-style-type: none"> <li>JIRA</li> </ul>	<ul style="list-style-type: none"> <li>Trello</li> <li>HubSpot</li> </ul>	<ul style="list-style-type: none"> <li>Hipchat</li> <li>Zendesk</li> <li>Marketo</li> </ul>
INFRASTRUCTURE	<ul style="list-style-type: none"> <li>Demo Staging</li> </ul>	<ul style="list-style-type: none"> <li>Regression</li> <li>Automated Tests</li> </ul>	<ul style="list-style-type: none"> <li>Design Process</li> </ul>	<ul style="list-style-type: none"> <li>Metrics</li> <li>Back-End Analytics</li> </ul>

So how do you create this strange, hybrid roadmap? An “agile-ish” roadmap will have dates with a caveat. Dates closer to the present are more granular, whereas the further into the future you go, the more abstract (i.e. fuzzy) the time headers become. This way, your dates aren’t the be-all-end-all and you’re not over planning the future (since far-off time is fuzzy).

Casey McKinnon, VP, Product at the hardware company [ecobee](#)—the builders of smart WiFi thermostats and sensors (how cool is that?)—is one PM who functions on a hybrid agile roadmap.



**CASEY MCKINNON**

“Agility to us means something completely different than it would to a software company. We work in phases,” says Casey. “There are nine phases we pass through and within each phase we follow agile process. But when you emerge from one phase, you commit to what you’ve created in that stage before moving on to the next.”

“With consumer electronics you have one window. Every October, stores load inventory for the holidays. So hitting that deadline is absolutely critical,” explains Casey. “We determine how many units we need and work back on when to start mass production. Everything prior to that is all sprints and iterating inside a cycle.”

See what we mean? While ecobee’s development process is agile, due to market demands, the team can’t release an MVP and just build on it. They have to get it right the first time, otherwise that’s a huge setback in keeping up with the market. **They can’t escape time!**



This combination of methodologies strongly influences ecobee’s agile roadmap. Casey says, “It’s a calendar. Down the side you see the different phases we pass through. Underneath each phase are items corresponding to outputs from that stage. Essentially on the roadmap, each phase gets a block of time and within that you are sprinting.”

**42%** of companies use both agile and waterfall \*

\* Product Focus’ 2017 Product Management and Product Marketing survey

This roadmapping process ensures Casey’s teams are very aware about what they’re doing up until a certain point (i.e. end of a phase). Beyond that, **future tasks are less certain and only addressed when entering a phase.**

ecobee isn’t the only company that’s agile roadmap is affected by real-world realities. **Rahul Kulkarni**, Product Manager at **Shopify** (specifically the Point of Sales team), finds his agile roadmap impacted by time too.

“We have two major events in a year: our annual developer conference, Unite, and Black Friday/Cyber Monday. We have roadmaps specifically planning for those two dates,” says Rahul. “Because we’re at scale now, we also introduced timelines that incorporate joint marketing pushes. So we have some deadlines that are marketing- and growth-oriented.”



**RAHUL KULKARNI**

	Q3 2017	Q4 2017	SOON	FUTURE
NEW FEATURES	Feature Requirements	Feature A Scope	Integrated Prototype	Undo Function
	Front-End Prototype	Archiving	MVP Requirements	Feature B Scope
				Scope

Thanks to multiple timelines, Rahul’s agile roadmapping process becomes more complex. Rahul explains, “**We definitely use the hybrid approach to agile.** We have two-week sprints and a monthly release cycle. There’s a very granular view of our monthly release cycles— which you can call our roadmap. It’s here we have a backlog of functions and use cases and different things we want to accomplish this upcoming release cycle.”

But as Rahul’s team looks beyond their upcoming release cycle, things become less granular and get relegated to a lengthy spreadsheet. “On top of that we have a spreadsheet full of really strong high-level ideas (i.e. concepts of projects and products) that we’ll continue to revisit. And if a really good opportunity comes up, we can put it on our more granular roadmap,” Rahul says.

The closer Rahul’s team is to the present, the more certainty there is around what they can deliver. **The further into the future they plan, the more uncertainty there is.** That’s the reality of agile. You can speak to what’s around the corner very well, but the rest of it... not so much.

**The future is uncertain.  
Create an “agile-ish” roadmap**

**Try this template**

Button not working? [Click here.](#)

## CHAPTER 4

# Guidelines for your agile roadmap

You've now got four powerful ways to visualize your agile roadmap. But no matter how you choose to represent your roadmap, there are certain standards you *just gotta follow*.



## Treat an agile roadmap as a statement of intent

First thing's first: **approach your agile roadmap as a statement of intent**. We covered this in Chapter 1 and we're going to keep *drilling it into your minds*. Treat your agile roadmap as a living document that can (and 100% will) change.

“From just a mechanic standpoint, making sure you have a **roadmap that can actually be updated** is really important,” says Ahsan Nanji, Director of Product Management at Electronic Arts (no relation to Latif Nanji, our CEO). “I’ve seen PMs struggle with creating a plan that doesn’t build any flexibility into it. So you want to—with either the software or whatever process you’re using—be very adaptable to change.”

If you're too hung up on dates or milestones, you restrict your roadmap and ultimately set yourself up for failure. Give yourself breathing room, so that way you can **sync your roadmap nicely with the flexibility and changes you'll experience in an agile environment**.



## Determine if time is right for you

When we asked Casey McKinnon (VP, Product at ecobee from Chapter 3) about his agile roadmap best practices, the very first thing he brought up was evaluating the role of time for your roadmap.

“Well, first of all I would consider if time’s a factor,” he says. “If the idea is that you need to hit a specific date, you need to compare that to something where you can put your product out into the world and there are no consequences for when it’s launched.”

Determine how time plays a role (if at all) in your day-to-day function. If it doesn’t play a role: awesome, your roadmap won’t incorporate time (i.e. theme-based). If time does have a place, decide how to represent time. “Agile-ish?” Fuzzy time? Sprints? Knowing whether time applies, and its particular role in your environment, will lead to a more effective agile roadmap.



## Remember who’s seeing your agile roadmap and tailor to them

Creating a roadmap without an audience in mind is pretty useless. The whole point of a roadmap is to communicate and align with your stakeholders. One of the very first questions you should be asking is: **who’s going to see this?**

“You want to understand your audience first, because they want to see different things,” says Latif (CEO here at Roadmunk). “If you’re building this agile roadmap for engineers you can get a lot more granular about information, whereas sales and marketing are more concerned about the ‘what’ and ‘when.’”

“One of the worst things you can do is have everyone see the same roadmap,” says Latif. If you know each stakeholder has individual goals and concerns, you can’t show up to each meeting with

the same exact plan. Offer something relevant to them. The result: **expectations are managed and stakeholders can feel assured that their needs are being addressed.**



## Know your product market. Like really know it.

This seems obvious. Of course a PM should know their product market! But we're talking *really* knowing what **upcoming features users want**, beforehand (planning step 4 in Chapter 2). In agile this will change—and quickly. But a strong market knowledge makes your roadmap much more adaptable to inevitable changes.



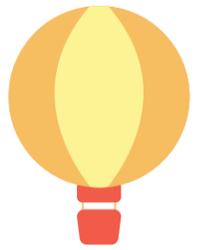
## Check-in with your agile roadmap (as a team)

This should be the easiest guideline to remember because it's a natural outcome of agile. Since you're working in short cycles, you're immersed in a constant feedback loop. You're regularly forced to stop and ask yourself, "So what worked and didn't work this last cycle?"

This introspective behavior pushes you to **check in with your roadmap often**. Your document should always be up-to-date with your product strategy and if you've created (and you should have) a living document that can change,

it'll be easy to update to reflect recent changes.

Casey McKinnon not only agrees with this practice, but emphasizes getting the whole team involved (planning step number 5 in Chapter 2): "**You should continue to revisit your roadmap with your whole team**. Include all the disciplines on the roadmap and don't let it be an artifact of product management, but make it an artifact of the whole company. This way we all agree, we all bought in, and everyone's really excited about the roadmap."



“Sarah Pyo, Product Manager at Expedia, puts it this way: “Understanding how your product works technically will help with prioritization, because you’ll know different levers to pull to drive your product. What new, futuristic features can we push out there and what is the demand?”

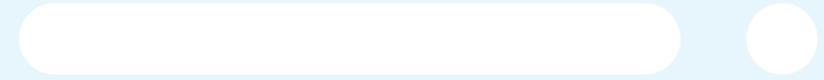
“Recommending a user-centric method to collect market knowledge for your agile roadmap, Rahul Kulkarni (Product Manager at Shopify from Chapter 3) says, “Validate low-fidelity mockups with customers as often as possible. Get real feedback and parse out what’s relevant.”

Chances are your market will be disrupted (many times), and as a result, so will your roadmap. **Keeping tabs on users' wants and needs will prepare you to be just as agile**; allowing you to update your roadmap swiftly to reflect any market shifts.





Agile roadmaps have solid benefits that fit nicely into any agile environment. Whatever your agile style, the key to reaping the rewards of an agile roadmap comes down to treating it as a **statement of intent** (we told you we wouldn't let you forget this idea). Your agile roadmap requires flexibility, attention and frequent re-evaluation to be truly effective. Build a living document that can and *will* change, and watch as your agile environment gets injected with some much-needed alignment.



To build an effective agile roadmap, build a statement of intent



# Kickstart your roadmap with our templates

Don't stop at agile roadmaps. Join our many customers and choose from over 20 customizable templates you can make your own.

**citibank**

**Adobe**

**amazon**

**#slack**

**mastercard.**

Explore our template library

Button not working? [Click here.](#)

