Running agile programs (without losing your mind)

Some may think transitioning to agile means losing sight of the bigger picture. We couldn't disagree more.

BY DAN RADIGAN

Early adopters of agile development were small, self-contained teams working on small, self-contained projects. They proved the agile model can work, to the joy and betterment of software makers around the world. More recently, larger organizations are scaling agile beyond single teams or projects, and seeking ways to apply it to whole programs.

This is not without its challenges. But that doesn't mean it can't be done! But first, let's hear how Gilt, an online fashion retailer uses agile to transform their business.

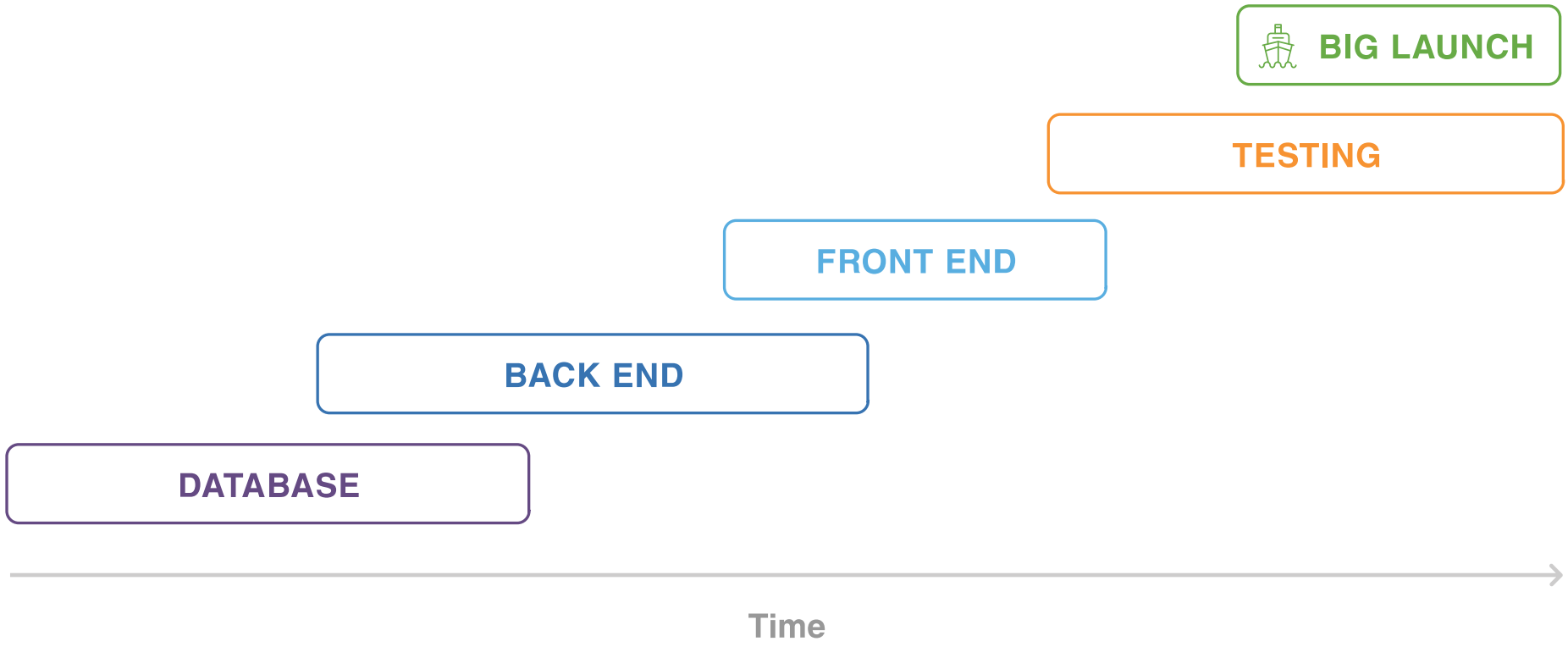
Up Next
Workflow →

Waterfall versus agile

Let's start with the basics–like what makes agile different.

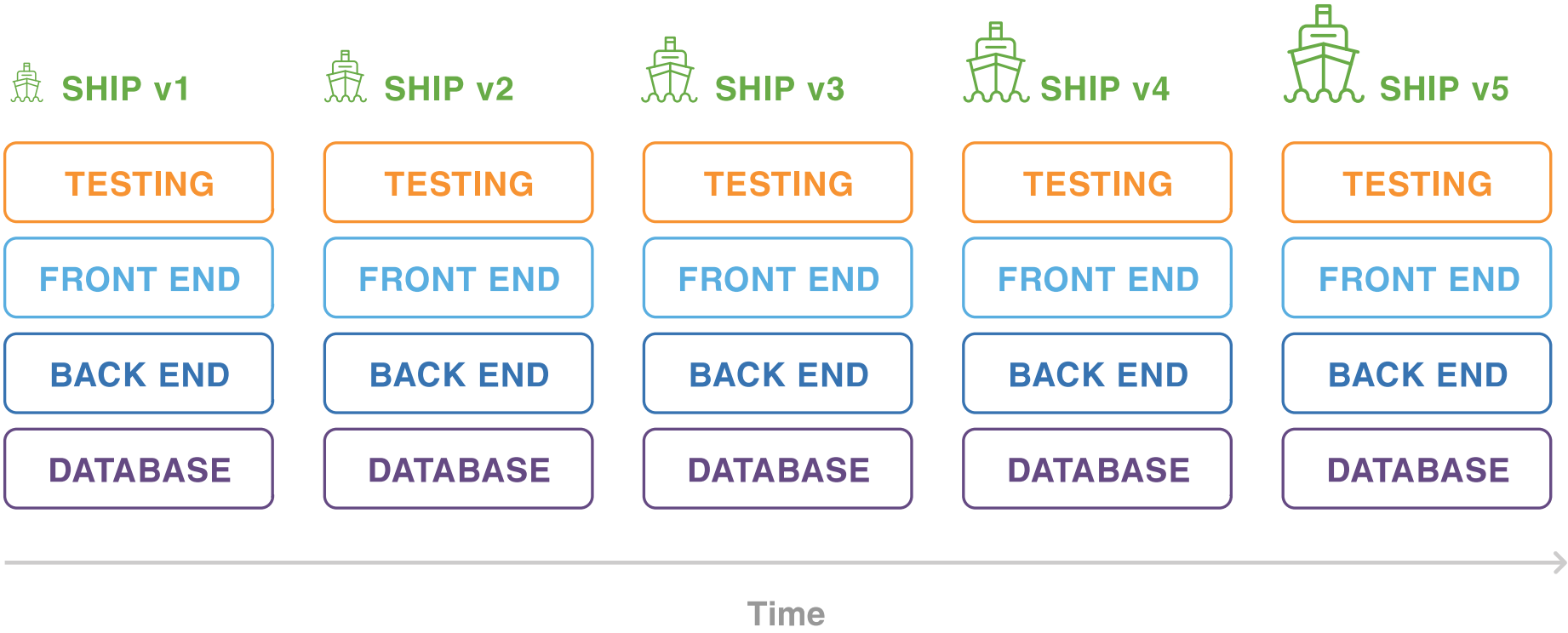
Traditional project management styles, like waterfall, build in phases. Below is an illustration of a

bang", high-risk release. Once a project passes one phase, it's painful to revisit it because teams are always pressing forward to the next stage.



Traditional project management styles often create "critical paths", where the project can't move forward until a blocking issue is resolved. To add insult to injury, the end customer can't interact with the product until it's fully complete. Thus, important issues in the product's design, and code, go undiscovered until release.

Let's contrast that with an agile project management style, which takes an iterative approach to development with regular feedback intervals. These iterations allow for the team to be diverted to (and productive in) another area of the project while a blocking issue is resolved.



Besides removing critical paths, iterations let you interact with the product during development.

This, in turn, gives the team constant opportunities to build, deliver, learn, and adjust. Market changes don't catch you flat-footed, and teams are prepared to adapt quickly to new requirements.

An even greater benefit is shared skill sets among the software team. The team's overlapping skill sets add flexibility to the work in all parts of the team's code base. This way, work and time isn't wasted if the project direction changes. (For more on that, see our article on building great agile teams.)

How to build a great agile program

When a program transitions from traditional project management to agile, the team and the stakeholders must embrace two important concepts:

- The product owner's focus is to optimize the value of the development team's output. The development team relies on the product owner prioritizing the most important work first.
- The development team can only accept work as it has capacity for it. The product owner doesn't push work to the team or commit them to arbitrary deadlines. The development team pulls work from the program's backlog as it can accept new work.

Let's explore the mechanisms agile programs use to organize, run, and structure work in an iterative way.

Roadmaps

A roadmap outlines how a product or solution develops over time. Roadmaps are composed of initiatives, which are large areas of functionality, and include timelines that communicate when a feature will be available. As the program develops, it's accepted that the roadmap will change–sometimes subtly, sometimes broadly. The goal is to keep the roadmap focused on current market conditions and long-term goals.

Requirements

Each initiative in the roadmap breaks down into a set of requirements. Agile requirements are lightweight descriptions of required functionality, rather than the 100-page documents associated with traditional projects. They evolve over time and capitalize on the team's shared →

understanding of the customer and the desired product. Agile requirements remain lean while everyone on the team develops a shared understanding via ongoing conversation and collaboration. Only when implementation is about to begin are they are fleshed out with full details.

Backlog

The backlog sets the priorities for the agile program. The team includes all work items in the backlog: new features, bugs, enhancements, technical or architectural tasks, etc. The product owner prioritizes the work on the backlog for the engineering team. The development team then uses the prioritized backlog as its single source of truth for what work needs to be done.

Agile delivery vehicles

Agile can be implemented using various frameworks (like scrum and kanban) to deliver software. Scrum teams use sprints to guide development, and kanban teams often work without fixed work intervals. Both frameworks, however, use large delivery vehicles like epics and versions to structure development for a synchronized release cadence out to production.

Agile metrics

Agile teams thrive on metrics. Work in progress (WIP) limits keep the team, and the business, focused on delivering the highest priority work. Graphs like burndown and control charts help the team predict their delivery cadence, and continuous flow diagrams help identify bottlenecks. These metrics and artifacts keep everyone focused on the big goals and boost confidence in the team's ability to deliver future work.

Agile runs on trust

An agile program cannot function without a high level of trust amongst team members. It requires candor to have difficult conversations regarding what's right for the program and the product. Because conversations happen at regular intervals, ideas and concerns are regularly expressed. That means team members also have to be confident in each other's ability (and willingness) to execute on the decisions made during those conversations.

In summary, agile development is a structured and iterative approach to making software. It gives you the ability to respond to change without going off the rails. And that's good news for any program.

SHARE THIS ARTICLE

DAN RADIGAN

Agile has had a huge impact on me both professionally and personally as I've learned the best experiences are agile, both in code and in life. You'll often find me at the intersection of technology, photography, and motorcycling. Find me on Twitter! @danradigan

TUTORIAL

Learn scrum with Jira Software

A step-by-step guide on how to drive a scrum project, prioritize and organize your backlog into sprints, run the scrum ceremonies and more, all in Jira.

Try this tutorial →

ARTICLE

Get started building an agile workflow

Every software team has a process they use to complete work. Workflow is about bringing structure and scale to that process. Learn more about workflow management, process & scaling.

Read this article →

Agile Topics

Agile project management
Scrum
Kanban
Design

Software development
Product management
Teams
Agile at scale
DevOps

Sign up for more agile articles and tutorials.

Email

email@example.com

Subscribe