Your guide to DevOps

Bring developers, IT, and the latest tools together to create a smarter, leaner, more successful coding machine
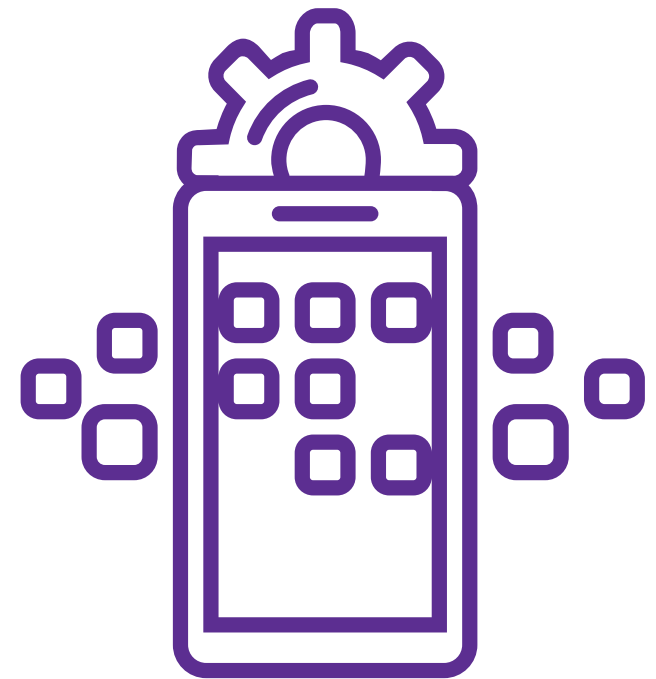
Introduction

The move to DevOps involves more than new processes and the latest tools. It's a whole new philosophy that brings developers, IT, and the business itself together to build and deploy apps continuously, with much higher quality than ever before.

Getting there requires shifting the way an organization thinks about developing new apps. It requires a mindset of collaboration and continuous evolution in which all stakeholders are involved from the beginning. It requires tools and technologies that can support the entire process and automate wherever it makes sense.

And the journey is worth it. By understanding and implementing DevOps, teams of all sizes can get applications into the hands of users faster than ever, while continually increasing quality and user satisfaction.

Want to know more? Read on.

Table of contents

Part one: Harmonize people, processes, and technologies

Part two: Master the art of testing

Part three: See the big picture, manage complexity, and tame large codebases

Part one:
Harmonize people, processes, and technologies

Get on the same page to build software—and value

Part one:
Harmonize people, processes, and technologies

What is DevOps? It's the union of people, process, and tools to enable continuous delivery of value to our end users. The fundamental promise of DevOps is better collaboration, faster cycle times, and higher quality.

Even though the term is new and subject to debate, its roots go as far back as software development itself. The intent behind building software is to solve a challenge, address a need, and create value for users. DevOps is really a method for doing that in the most natural possible way, by involving all roles and stakeholders from the beginning.

As such DevOps provides a frame of reference for how you and your organization continuously deliver value in the form of software to your end customers—here and now, in the real world.

Defining your approach

It's that focus on the realities of each organization that makes DevOps as a whole so hard to define. What are your organization's IT capabilities? Are you deploying on-premises, in the cloud, or using a hybrid scenario? Which processes should you automate, and where does manual input make more sense?

DevOps teaches you to treat your IT as part of your code, so any app you write can take advantage of its capabilities, while avoiding pitfalls. Going deeper, the broader organization also plays a part. You must consider how many developers you have, the size of your budget, the needs of your business stakeholders, and the landscape of the industry in which you work.

Any of these factors could make or break a software project. That's why DevOps seeks to bring all stakeholders in from the beginning, especially the IT Ops department, so the entire project is grounded in what's possible and potential challenges are identified from the start. Remember: It's all about collaboration.

Defining your approach (cont'd)

Learn more about building a DevOps organization, including how to frame it for your unique needs, and which processes to automate. In this episode of "Breakpoint," hosts Paul Laberge, Jonathan Rozenblit, and Anthony Bartolo discuss the DevOps discipline and how to start putting it to work.

Watch video
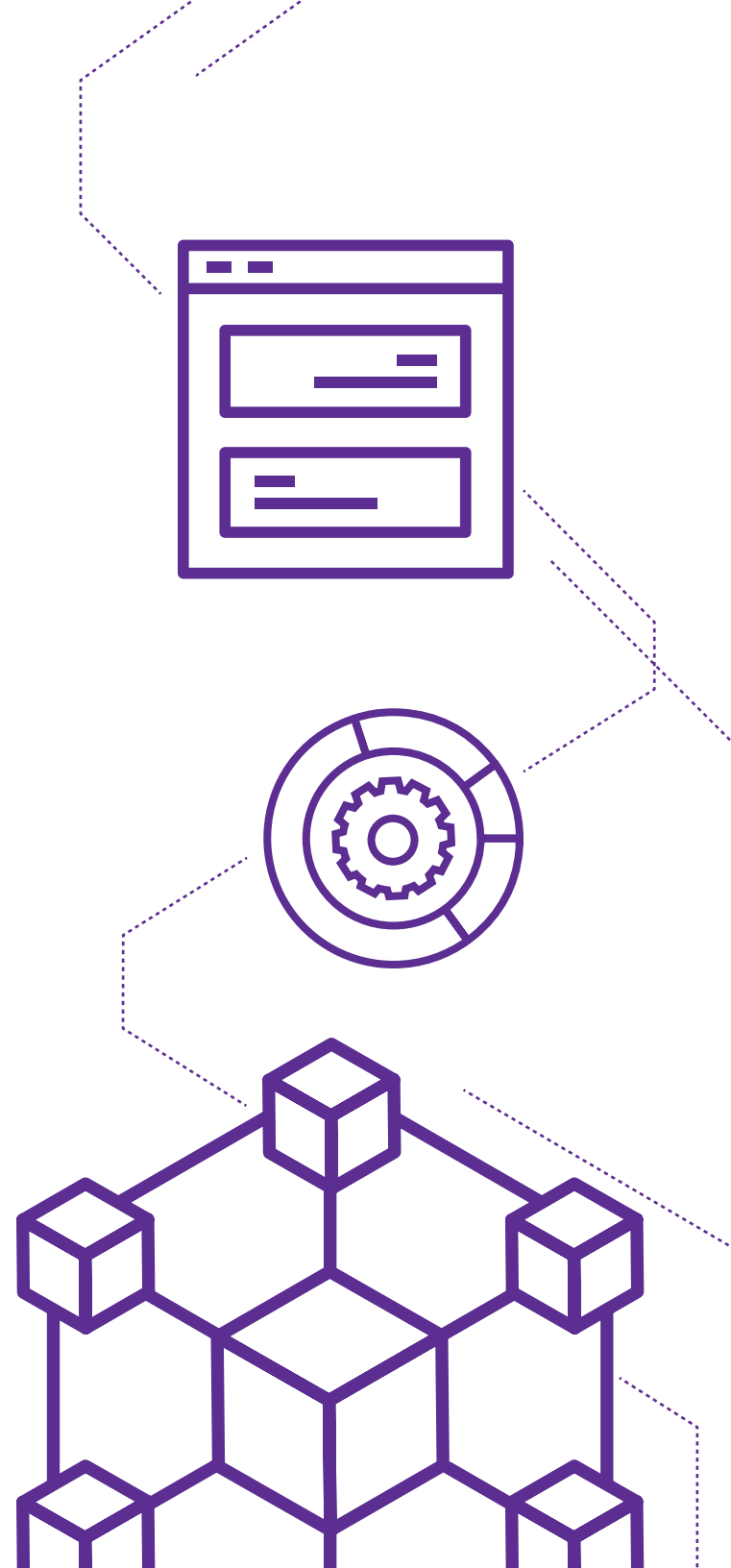
Building an effective release pipeline

There are many challenges in releasing software that can affect quality, increase costs, or even derail projects altogether. Manual cycles require time-consuming outages and increase the possibility for errors. A lack of standards for different projects can lead to inconsistencies. Having multiple processes and environments creates inefficiency.

That's why managing releases is one of the most important areas for automation in a DevOps organization. Continuous, automated delivery is becoming the standard to solve all of these issues.

If you want to add this important field to your skillset, there is a Microsoft Virtual Academy course available for free as a series of videos on Channel 9. Microsoft Senior Premier Field Engineers Krithika Sambamoorthy and Micheal Learned walk through all the necessary skills to build an effective release pipeline using the Release Management functionality provided by Visual Studio and Team Foundation Server.

Topics covered include:

- Release Management architecture
- Visual Studio ALM and DevOps overview
- Release pipelines
- Deployment recipes
- Advanced topics
- Recommended resources and next steps

Building an effective release pipeline (cont'd)

Get started with the full online course. There are several lessons that will take you a few hours to complete.

Watch video

Take a look at an overview of how Visual Studio 2015 adds an open, web-based management interface that lets you share your existing assets across build and test tools, even if they're on multiple platforms**.**

Watch video
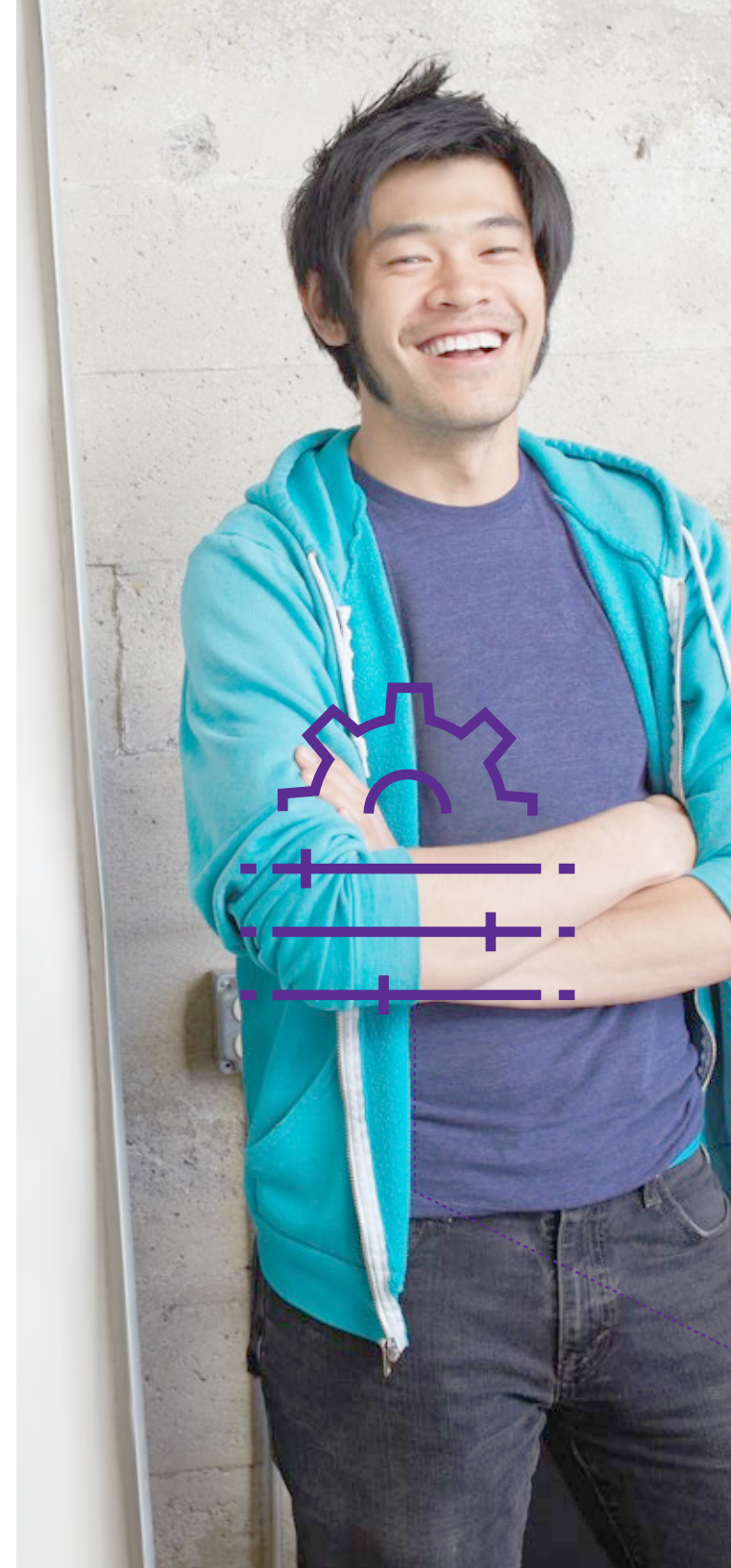
Part one:
Harmonize people, processes, and technologies

Microsoft Virtual Academy: Visual Studio advanced features

Whether you are working on greenfield software development projects or legacy code, the advanced features and tools in Visual Studio Enterprise can simplify your efforts. Look for the Microsoft Virtual Academy training videos throughout this ebook to discover how these advanced tools can help you design and build applications that are ready for modern continuous integration and continuous delivery pipelines, and easily transform older applications into modern ones.

Modules include the following:

- **Design.** Discover how to build well-architected software using Visual Studio Enterprise architecture tools such as Code Map and the Layer Diagram.
- **Debug.** Understand the importance of good tooling for rapidly finding the root cause of bugs. Learn how to use IntelliTrace, IntelliTrace in Production, Code Map, and the .NET Memory Dump Debugger.
- **Test.** Quality starts with good design and really becomes critical during coding. Learn how to use IntelliTest, Fakes and Shims, Coded UI, and Code Coverage with manual tests.
- **Tune.** Application performance is critical to the success of nearly all applications. Learn how to use web performance testing and load testing to find bottlenecks and identify opportunities for improvement.
- **Release.** See how the new Release Management can dramatically simplify the creation of continuous integration and continuous delivery pipelines that include build, unit test, package, deploy, and integration test steps.

Visual Studio Advanced Features:
- Monitoring apps in production
- Release Management

Watch videos

Part Two:
Master the art of testing

Building an effective DevOps organization
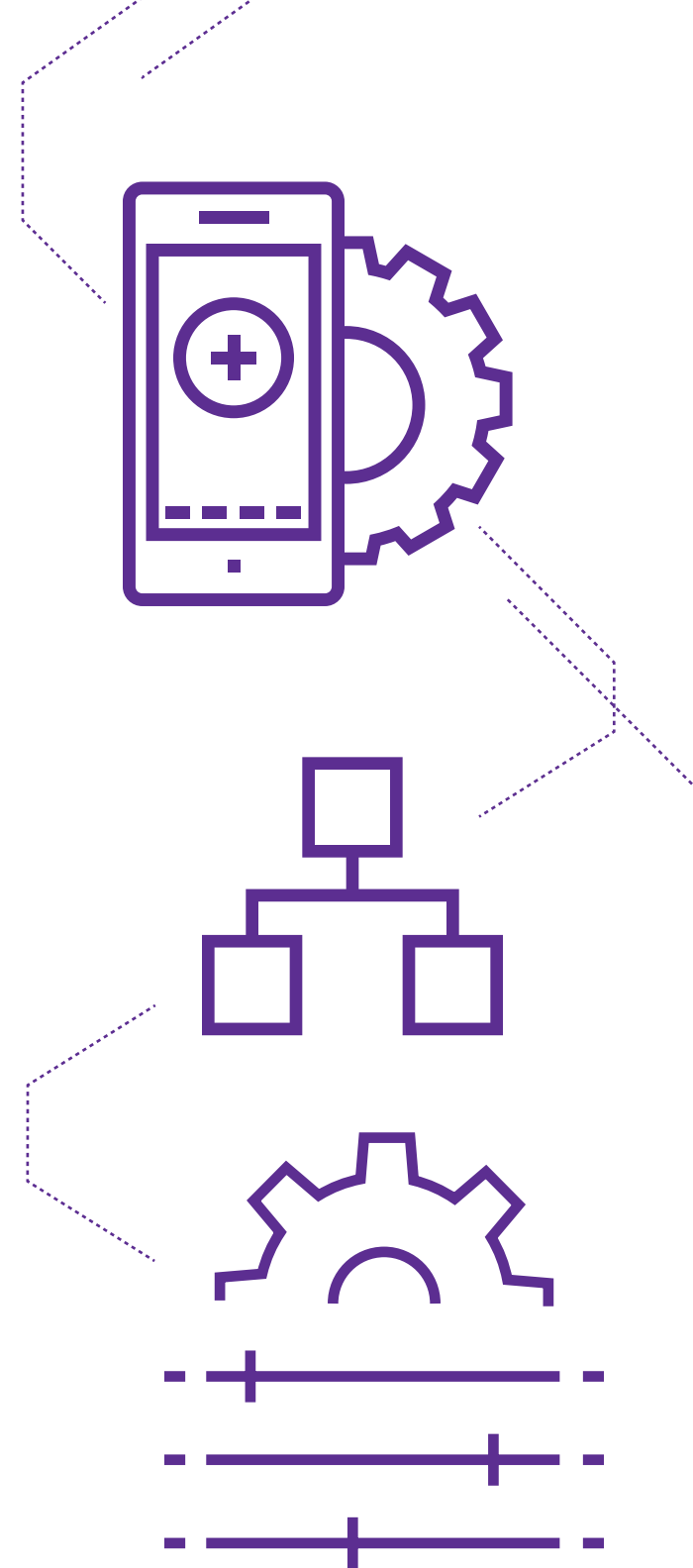
Part two:
Master the art of testing

It may seem as if DevOps has the potential to reduce the emphasis and importance of testing in the software development process. The reality is that testing and QA are just as important as always, if not more so. The difference in testing with DevOps lies in when it is conducted and why.

With traditional development, "big bang" testing is often left to the end of the cycle. This brings inherent problems, quality concerns, and delays as entire batches of code must be shipped back and straightened out.

Since the goal of DevOps is continuous integration and continuous delivery of new code, testing becomes continuous as well. The sooner testing is integrated into the cycle, the more effective it is, and virtually any time code is moved, merged, or integrated, testing should be performed, especially during the last phases of development.

Thus, the ability to master the when, where, why, and how of testing code is a critical element of building a mature and effective DevOps organization.

Fortunately, it's not all about process. Today's advanced tools for manual and automated testing can help you deliver top-quality work, all the time.

The importance of manual testing

Manual testing is still one of the most important quality assurance practices for any development house. It's also an art—a skill that requires practice.

One of the reasons manual testing is still so effective is the tester's own knowledge of both the business logic for the app and the technical knowledge of the infrastructure that supports it. Understanding this junction between why an app is performing a certain function and how that plays out behind the scenes enables the tester to better infer the probable cause behind any quirks brought up by the test.

In this episode of "Breakpoint," hosts Jonathan Rozenblit and Paul Laberge explain the importance of manual testing, how to implement it, and strategies you can use to integrate quality into your software project. Later, they show how the Visual Studio suite of manual testing tools makes the whole process easy and even allows you to leverage data provided by the tools to gain insight into the overall health of your project.

Watch video
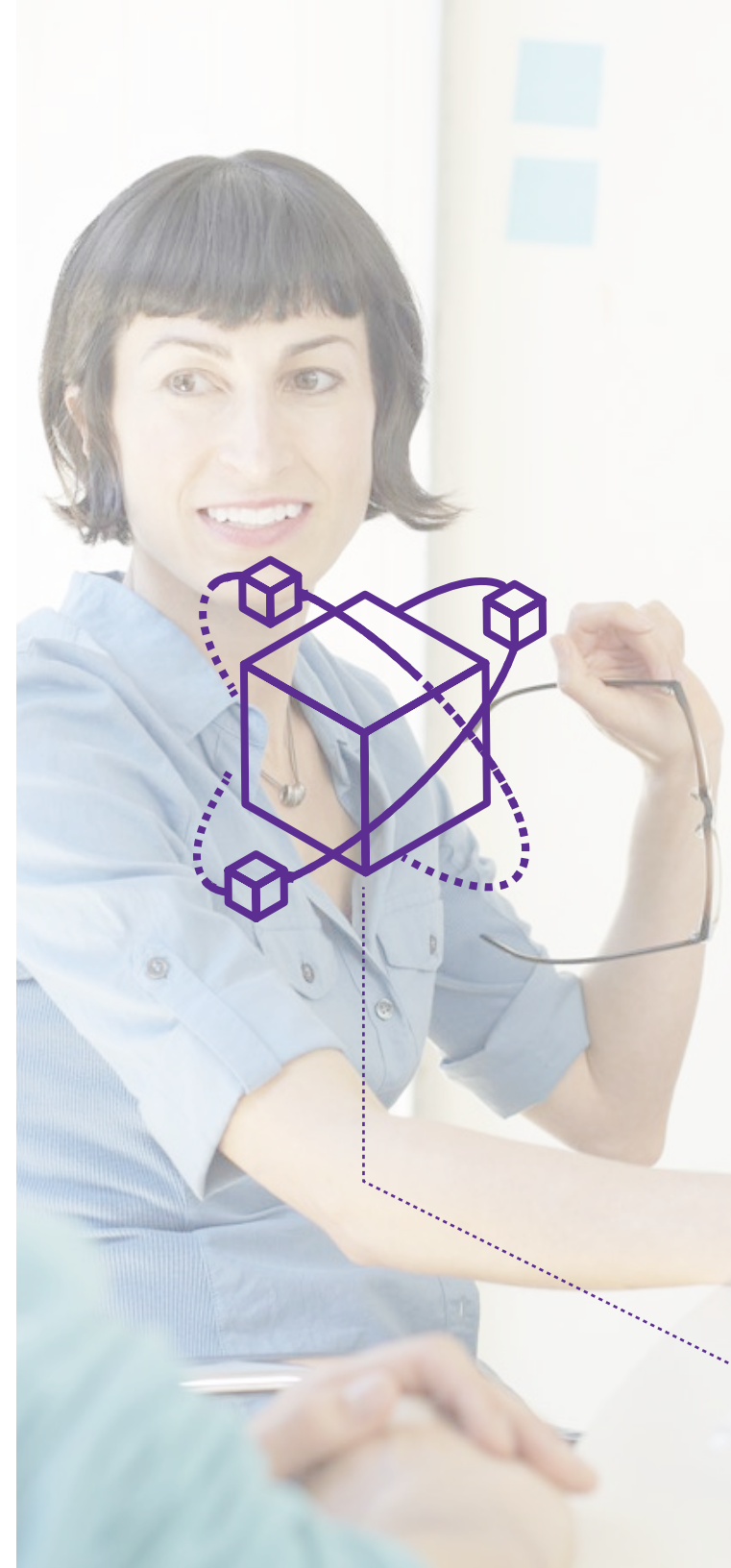
Breaking code down into testable units

Unit testing is the process of breaking an app's code down into the smallest testable pieces of functionality, isolating each piece from the rest of the code, and determining whether it behaves the way you expect. In a DevOps environment, the ability to do this effectively is one of the most valuable building blocks for creating better code.

So why aren't more developers doing more of it?

Many developers find that creating good, effective unit tests is a challenge. A good unit test is like a good scientific experiment: It isolates as many variables as possible and then validates or rejects a specific hypothesis about what happens when the isolated variable changes.

It can be hard to wrap your mind around unit testing at first, but getting a handle on this critical skill will provide immediate and long-lasting benefits to any DevOps organization.
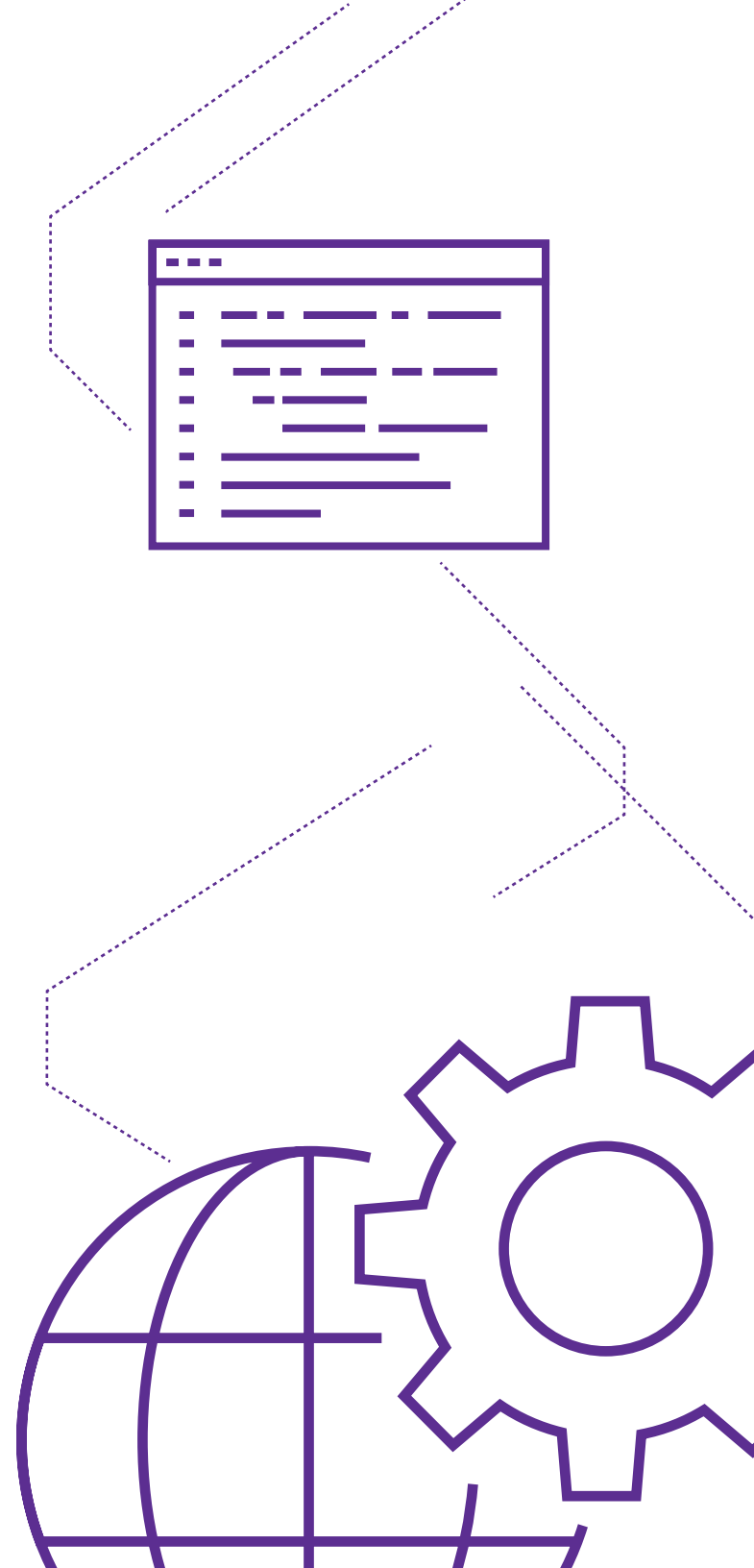
Breaking code down into testable units (cont'd)

Learning a consistent process for creating fast, reliable, and automated tests will add significantly to overall code quality. To understand the basics of doing this in a repeatable, consistent fashion, take a look at this three-part video series from Microsoft's Robert Green:

Part 1 | Part 2 | Part 3

Use IntelliTest to automate unit testing. Visual Studio Enterprise Edition includes the IntelliTest feature that can automate the process and make unit testing easier than ever. IntelliTest explores your .NET code to generate test data and a suite of unit tests. Running IntelliTest allows you to see which tests are failing and fix them quickly.

Read more
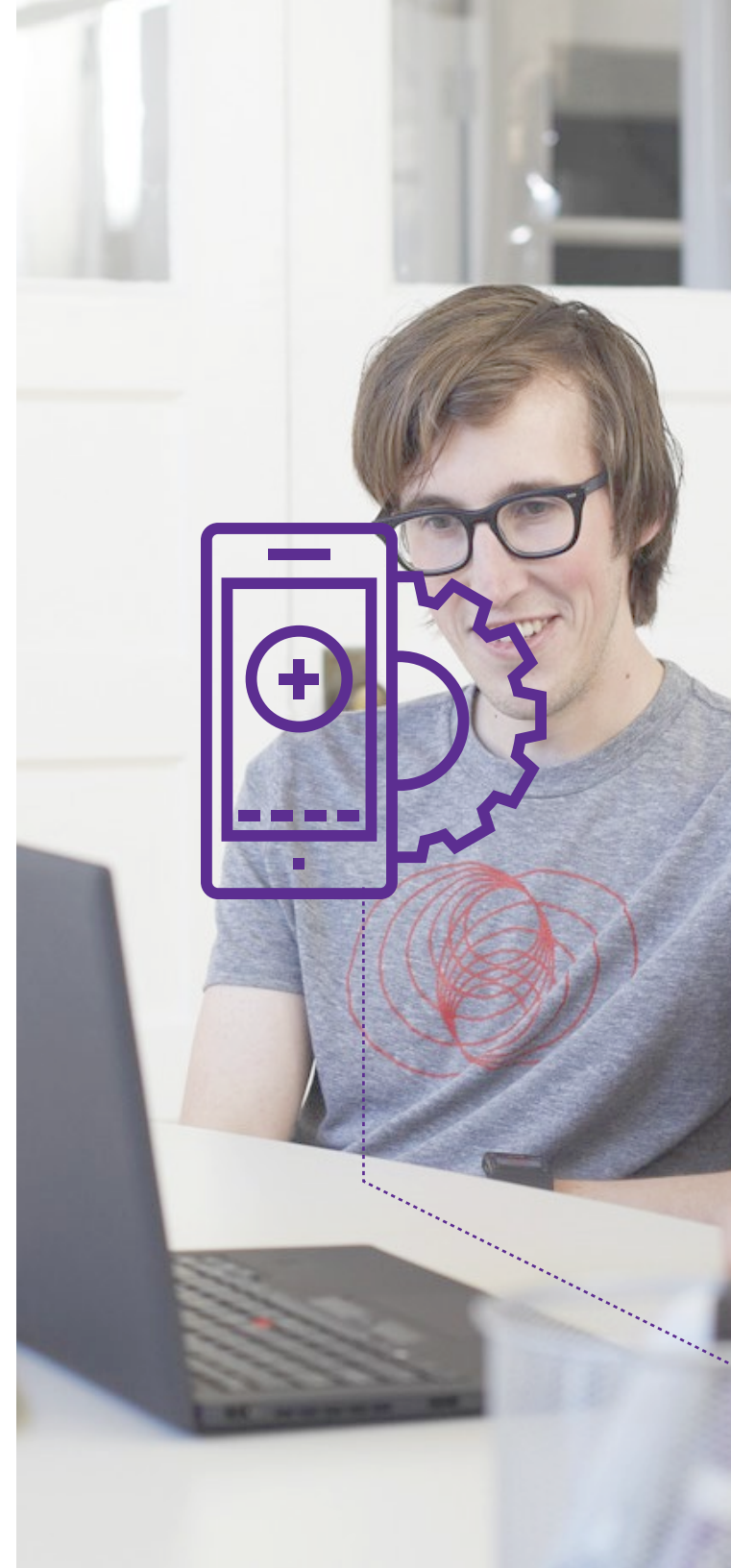
Load testing made easy

At the other end of the testing spectrum is load testing on an entire build. Load testing offers its own challenges, in that it's often expensive in terms of time and resources. As such, much like unit testing, it may not always be performed as often or as well as possible.

Visual Studio Team Services has several features to make advanced load testing much easier, including expanding your available infrastructure using the Microsoft Azure cloud.

In this video demo, Senior Program Manager Charles Sterling walks through some of the features in Visual Studio designed to facilitate load testing and ensure your apps work the way they should at scale.

Watch video
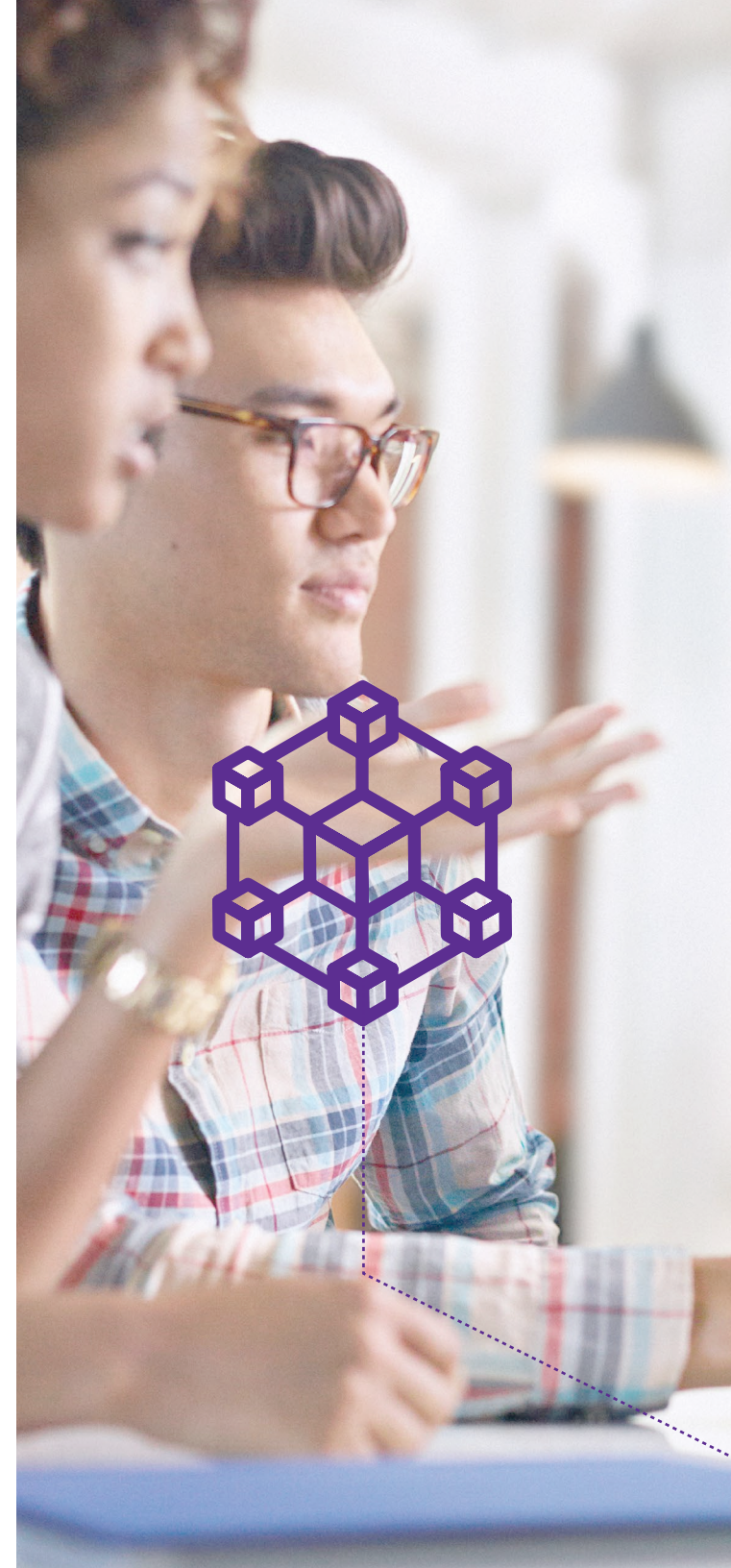
Continuous monitoring: Become an app whisperer to improve with every release

Another primary principle of DevOps is the notion that you are continually improving your apps and processes with every release. But how do you know that your app is hitting the mark with users?

While you may have received good feedback, there are also plenty of quantitative metrics to be had "under the hood." Tapping into those is important to understand whether you're focusing on the most important functionality.

Facilitating this is the function of Application Insights. Hosted on Azure, Application Insights allows you to see how your app is being used and how it's performing—directly from the app itself. With this new tool, you don't have to wonder which functionality is most important.
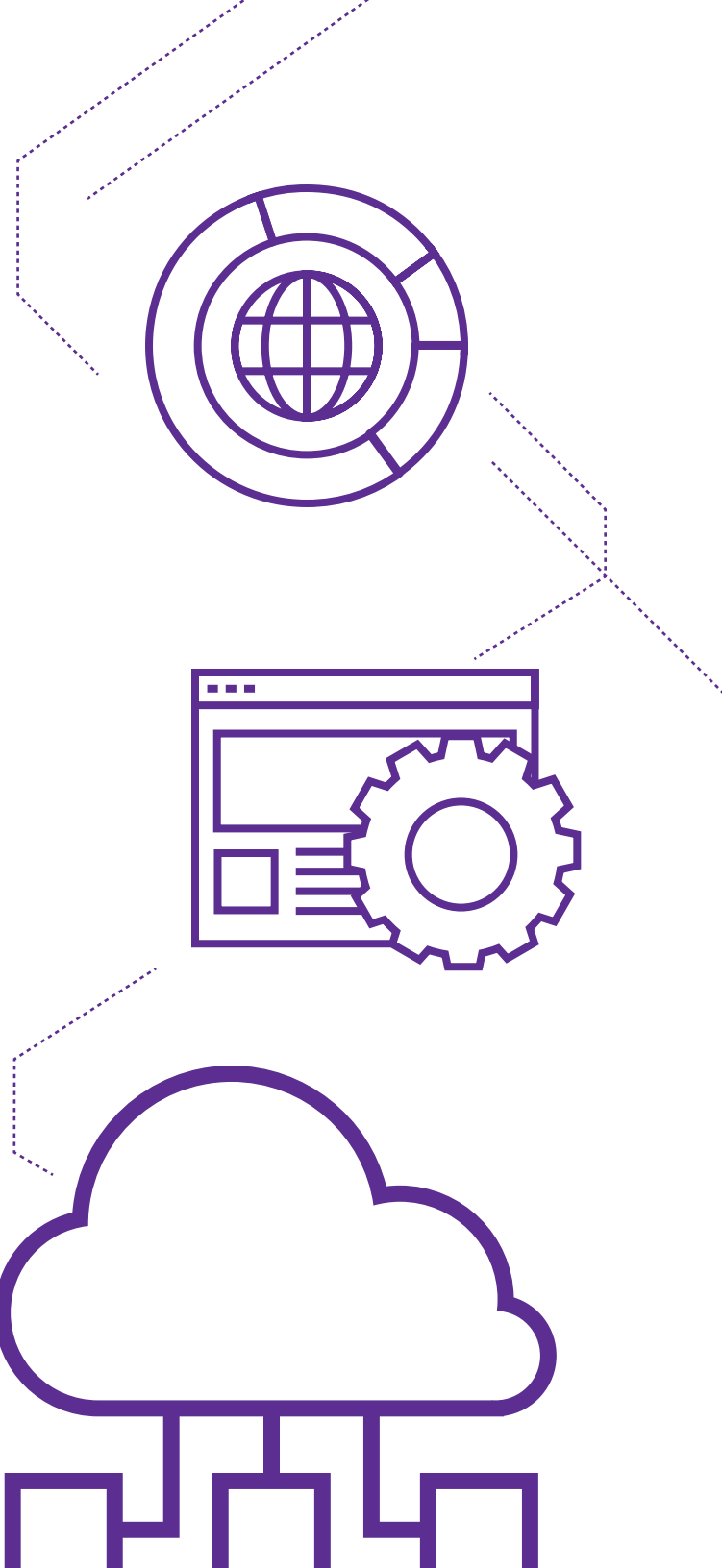
For a more detailed overview and demo of how Application Insights works, take a look at this episode of "Breakpoint Shorts."

Watch video

Visual Studio Advanced Features:
- Cloud load testing
- Quality tools for developers

Watch videos

Part three:
See the big picture, manage complexity, and tame large codebases

Tools, services, and resources for better team collaboration

Part three:
See the big picture, manage complexity, and tame large codebases

Just as it's important to continually test code at the smallest possible level, it's also critical to see the big picture and ensure everything comes together as a whole.

This is a bigger challenge than ever, as today's apps are getting much larger than in years past—and with today's more versatile devices that put the cloud's powerful computation in the palm of your hand, that trend is only going to continue.

Fortunately, modern development environments such as Visual Studio are making it much easier to manage, navigate, and monitor the behavior of complex code. In this section, see how advanced tools for code-mapping and architecture discovery make it easy to manage dependencies and large codebases.
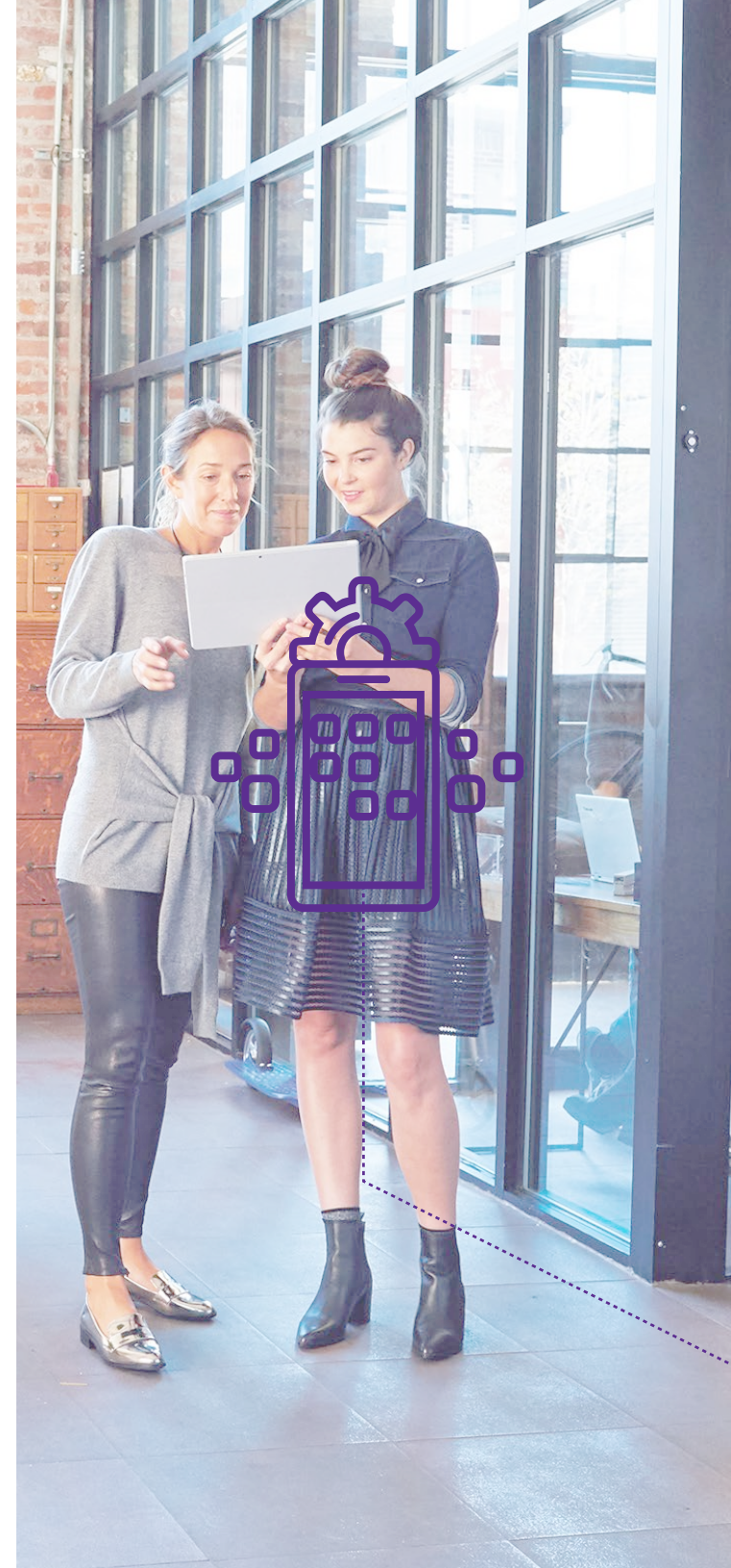
Part three:
See the big picture, manage complexity, and tame large codebases

## The critical role of the architect

Whether your organization has dozens of developers or just one, somebody has to design the software. Even if you're not an architect, chances are, at some point in your career, you will act as one.

And here's the thing: It's not easy. Beyond just designing the app, the architect is the hub between the developers who build it, the testers who put it through its paces, the operations team that implements it, and the business users themselves.

Fortunately, for anyone interested in this critical role in software development, there are several relatively unheralded tools in Visual Studio that offer support and guidance on designing the best software, and understanding how effective your team is in building it.

In the episode of "Breakpoint" below, hosts Jonathan Rozenblit and Paul Laberge explore the architectural tools in Visual Studio and how they can help your team build the best possible software.

Watch video
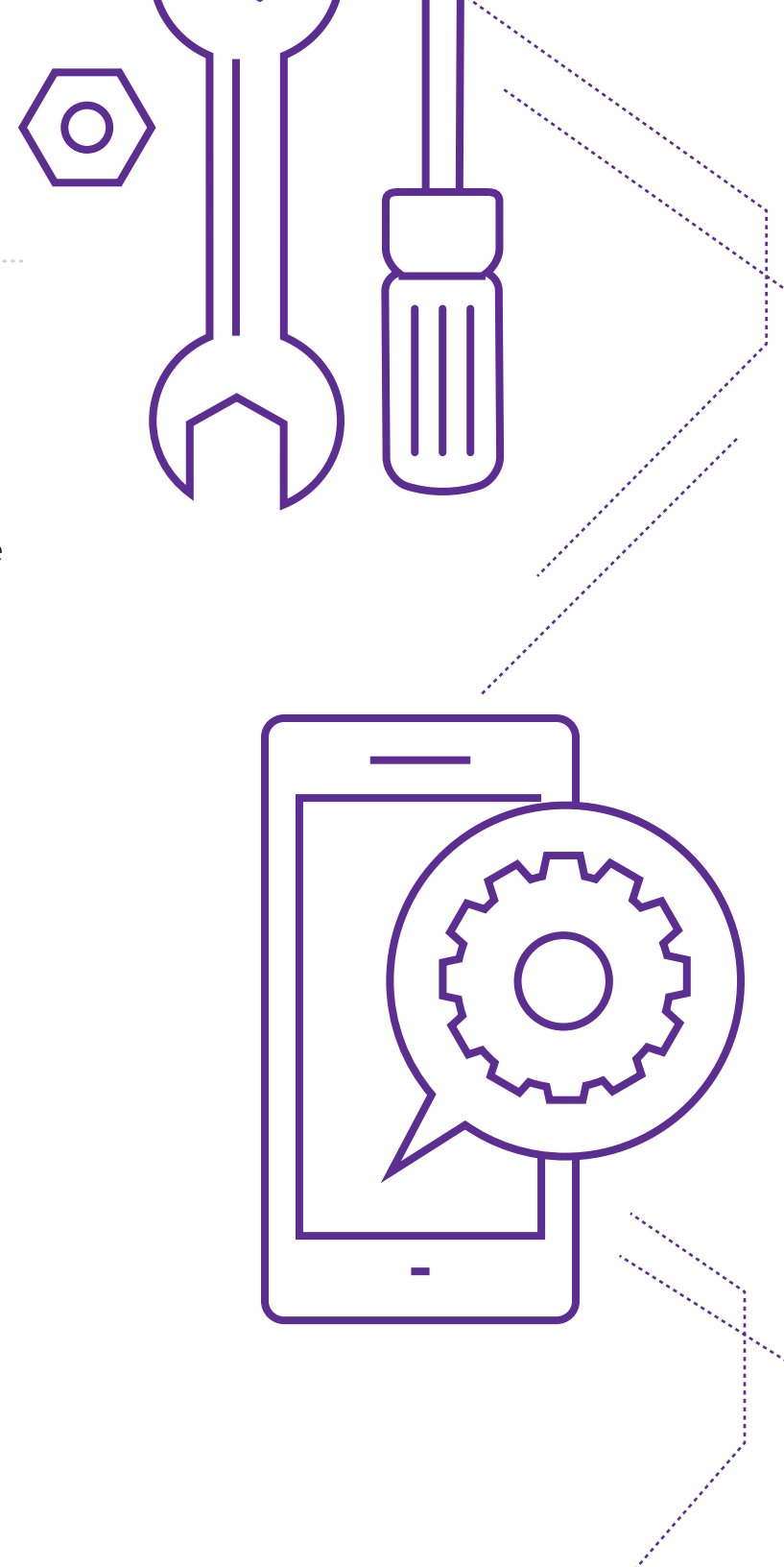
Part three:
See the big picture, manage complexity, and tame large codebases

Using code map to navigate and debug

Whether you're creating new code or integrating from a legacy application, it can be easy to get lost. The Code Map feature in Visual Studio allows you to create visualizations of the complex relationships that exist in your source code so you can easily navigate and understand dependencies and interactions. This can be useful if, for instance, you identify a bug in a feature, but aren't sure where exactly the underlying code for that feature is in the overall codebase.

Using Code Map, you can navigate directly from a feature to its underlying code, making it easy to identify the root cause of a bug and fix it. Watch the video below to see how you can use Code Map to navigate through complex code and resolve bugs that may have seemed invincible before.

Watch video

**Using code map to navigate and debug (cont'd)**

Code Map also integrates directly with Visual Studio Debugger. Integrating Debugger with Code Map helps you understand dependencies by showing you exactly where in the code you've been, in relation to where you are now. Check out the video below for a demonstration of this simple yet powerful tool.

Watch video

Visual Studio Advanced Features:

- Documenting architecture
- Managing code
- Package management

Watch videos
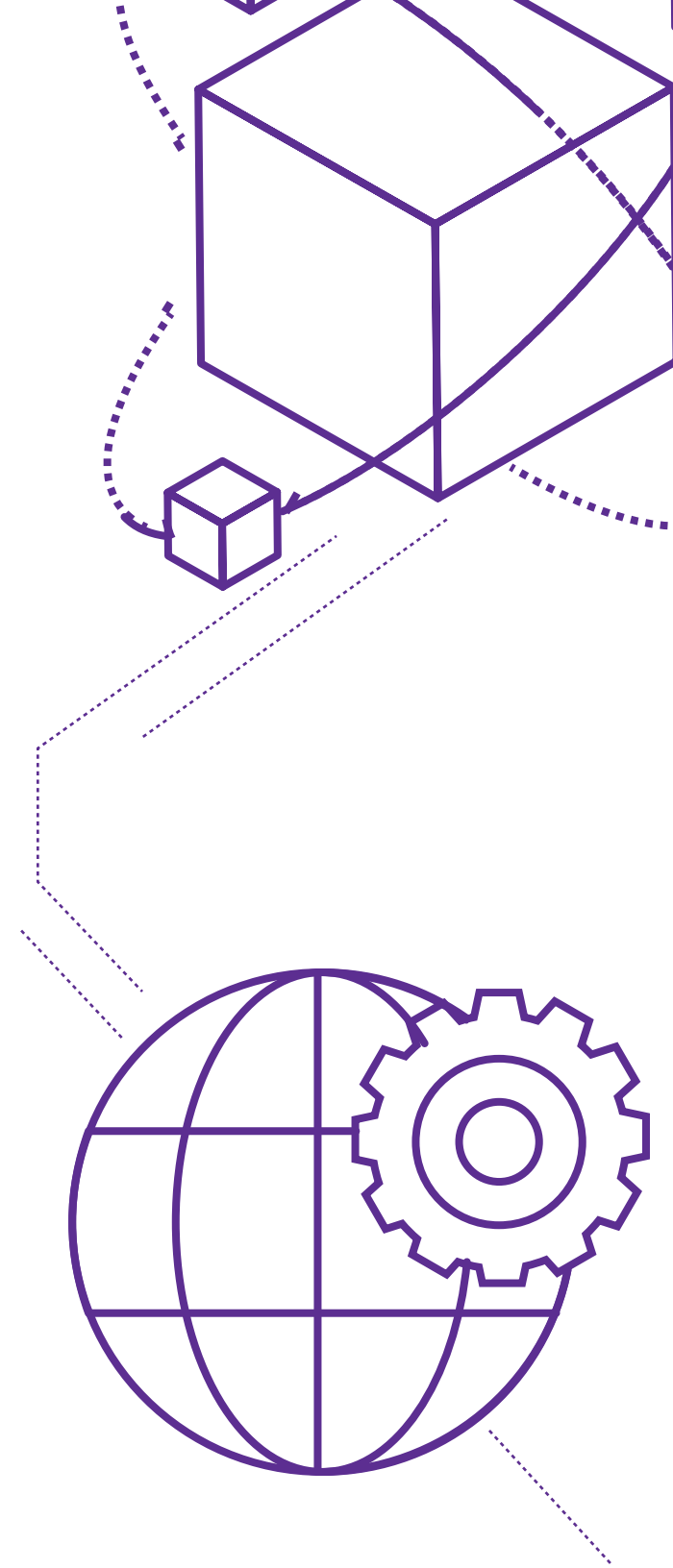
Is Visual Studio 2015 right for you?

Visual Studio 2015 is for developers and teams of any size that want to create great software for any device or platform. Check out the list of resources below and start your evaluation today:

Video introduction to Visual Studio Enterprise 2015
Watch this brief video to see more about this integrated, end-to-end solution for teams of any size.

Overview of Visual Studio Enterprise
Access product information, resources, purchasing info, and more.

Microsoft Virtual Academy: Visual Studio Advanced Features Overview
This five-part video series covers the advanced features and tools in Visual Studio Enterprise that will simplify your efforts every step of the way, from design to release.

Visual Studio Team Services
Share code, track work, and ship software—for any language, all in a single package.

Pricing info and purchasing options
Find the right Visual Studio edition for your team.

Download a free trial of Visual Studio Enterprise