

EBOOK

Harness the Power of AI for Code

diffblue
AI for Code



What exactly is AI for Code?

There's a lot of buzz around Artificial intelligence (AI) and its potential to revolutionize everything from retail to transportation to healthcare. But what exactly is 'AI for Code?' This white paper provides a few practical examples of how artificial intelligence (when harnessed properly) can provide scalable business-critical benefits. We'll guide you to the next steps for integrating this powerful technology into your development processes and leveraging it to outpace your competitors.

Defining Artificial Intelligence for Code

The hype around artificial intelligence (plus the ever-popular sub sector of 'machine learning') has grown exponentially over the years, and has resulted in equally big claims. By some estimates, AI-enabled tools alone will generate \$2.9 trillion² in business value by 2021 (in part, by saving an estimated 6.2 billion hours of worker productivity; you can read more about this in our whitepaper 'The ROI of AI for Code').

However, is there any merit to all of this hype? The one area that has earned a significant amount of media attention is AI for software development, and it's deserved. AI is becoming the fundamental building blocks of the software engineering process, and this is what we refer to as 'AI for Code'.

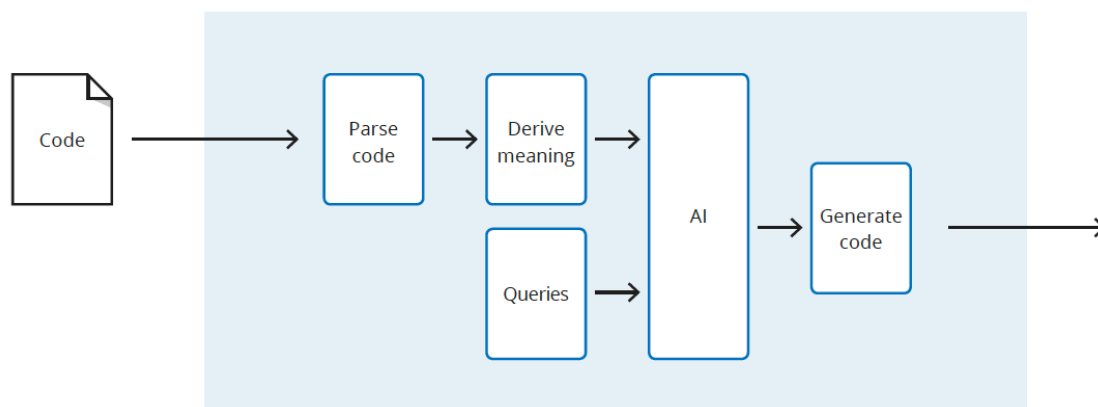
How does AI for Code work?

The very core of AI for Code is a mathematical reasoning and learning engine. It's a complex process which can be broken down into a few key steps:

1. Your code is parsed into an internal representation.
2. A semantic analysis constructs a formula which captures the behavior of all or part of the program.

3. The Diffblue Cover AI Engine automatically generates a series of queries about the code, such as which inputs are needed to reach line R, take branch B or to cause various kinds of exceptions. It uses automated reasoning and learning to understand the code and find answers to the queries.

5. In the case of Diffblue Cover, for example, answers from the AI engine are translated back into an execution trace of the program and these are used to build smart unit tests.



To take a more specific example, with Diffblue Cover—our tool for generating unit testing via AI for Code—we turn the executions into a suite of unit tests. We generate test cases for all behaviors of the code, from typical to non-standard and even very undesirable situations. This allows Diffblue Cover to make comprehensive improvements to your regression suite.

Example of a Diffblue Cover Generated Unit Test

To see this in action, it's helpful to understand how and why it's used. Diffblue Cover offers a pioneering approach to software testing that gives three clear benefits:

1. Automatically generate unit tests. This increases developer productivity and speeds up the development process.

2. Allow software architects, management and developers to understand the impact of changes to the source code, enabling them to make more informed decisions about development. This is especially helpful when considering changes or migrations to unknown legacy code.

3. Show where the existing code is not covered and automatically generate tests to increase coverage, thus improving the related regression suite.

Let us present an example of a unit test generated for the following function, taken from the open source Java Library, Swift:

```
265 public static boolean nameStartsWithAcronym(String name)
266 {
267     if (name.length() <= 1) {
268         return false;
269     }
270     if (Character.isUpperCase(name.charAt(0)) && Character.isUpperCase(name.charAt(1))) {
271         return true;
272     }
273     return false;
274 }
```

Diffblue Cover takes a function in from the source code and generates a suitable test case with appropriate variables:

```
/*
 * Test generated by Diffblue Cover.
 * This test case covers:
 * - conditional line 267 branch to line 270
 * - conditional line 270 branch to line 270
 * - conditional line 270 branch to line 271
 */
@Test
public void nameStartsWithAcronymInputNotNullOutputTrue() {
    // Arrange
    final String name = "BD";
    // Act
    final boolean retval = TemplateContextGenerator.nameStartsWithAcronym(name);
    // Assert result
    Assert.assertEquals(true, retval);
}
```

Who needs to harness the power of AI for Code?

Not all businesses have the budget or ability to hire the very best developers (or very many of them), and even the best developers have off days, or don't like working on some tasks, e.g. legacy code migrations.

Companies quickest to adopt AI for Code typically have some of the following criteria.

- Aggressive development timescales that cannot be met using conventional means
- Extremely fault-sensitive use cases where automation can help to ensure human errors are reduced
- Lack of context for legacy applications that need to be migrated or changed

Summary: AI for Code and Software Development

Here at Diffblue, we are revolutionizing software engineering through AI for code, making development teams wildly more productive. We do this by using AI to support and automate software development.

We're grounded in cutting edge research to help build the future of software development. We know that developers and existing development tools will not be able to keep pace with the future rate of development without assistance from AI.

[Learn more at www.diffblue.com](http://www.diffblue.com)

References

1. Titcomb, J. "AI Is the Biggest Risk We Face as a Civilisation." The Telegraph, Telegraph Media Group, 17 July 2017, www.telegraph.co.uk/technology/2017/07/17/ai-biggest-risk-facecivilisation-elon-musk-says/.

2. Biere, A., Cimatti, A., Clarke, E., & Zhu, Y. (1999). Symbolic Model Checking without BDDs. United States: Carnegie Mellon University.
3. Source code taken from: <https://github.com/facebook/swift> under the license <http://www.apache.org/licenses/LICENSE-2>.
4. "The evolution of artificial intelligence"
<https://www.ubs.com/microsites/artificial-intelligence/en/new-dawn.html>
5. Dsouza, M. "5 ways artificial intelligence is upgrading software engineering", Packt, 2 September, 2018, <https://hub.packtpub.com/5-ways-artificial-intelligence-is-upgrading-software-engineering/>
6. "AI to Drive Job Growth by 2020", 15 December 2017, <https://blogs.wsj.com/cio/2017/12/15/ai-to-drive-job-growth-by-2020-gartner/>