FLUTTER BASICS: THE GOOD AND THE BAD

Flutter has risen quickly as an app development tool. Originally released by Google in May 2017, Flutter has been used by two million developers since. LinkedIn reports Flutter is the fastest-growing skill among software engineers.

This excellent growth is fueled by users' hopes that it's an elixir to cure the coding experience of all maladies. Like anything, of course, Flutter has its shortcomings. Let's take a look.

What is Flutter?

Flutter is built on the Dart programming language. Developed by Google, Dart was first unveiled in 2011. The language covers the major hot points that a modern language should: it is object-oriented, class-based, and has an added garbage-collector. It has the async, future options out-of-the-box. It has C-style syntax, so should look familiar to JavaScript devs—in fact, devs report they pick up the language quickly.

Dart is intentionally simple. Ease comes with costs, so Dart can be executing extra, or less-refined, work in the background. Compared to writing the native code, Dart can be slower and less reliable than a native language. Dart is to JavaScript what Python is to C++.

Flutter is an open-source tool for building UIs, particularly on mobile. An essential concept to Flutter is its widgets. Their motto, *everything is a widget,* is entirely true. All things are widgets. From building layouts with Scaffold and Material App widgets, to BLoC patterns and Provider Widgets, Flutter is built of widgets. Its layouts need to be hand-built, but a few developers created some layout

playgrounds to let you build and print the code:

- mutisya.com
- flutterstudio.com

In this code, you can see how a Text() widget is inside an AppBar() widget is inside a Scaffold() widget.

```
28
29 class _MyHomePageState extends State<MyHomePage> {
30     @override
31     Widget build(BuildContext context) {
32       return new Scaffold(
33         appBar: new AppBar(
34           title: new Text('App Name'),
35           ),
36       );
37     }
38 }
```

The Flutter project is only two years old and the team is hard at work. Their GitHub sits at 7,500 open issues, almost 30,000 closed, and 93,000 stars. According to an April 2020 update, Flutter has:

- 500,000 monthly users
- ~50,000 apps on the Play Store, including 10,000 uploaded in March 2020.

The benefits of Flutter

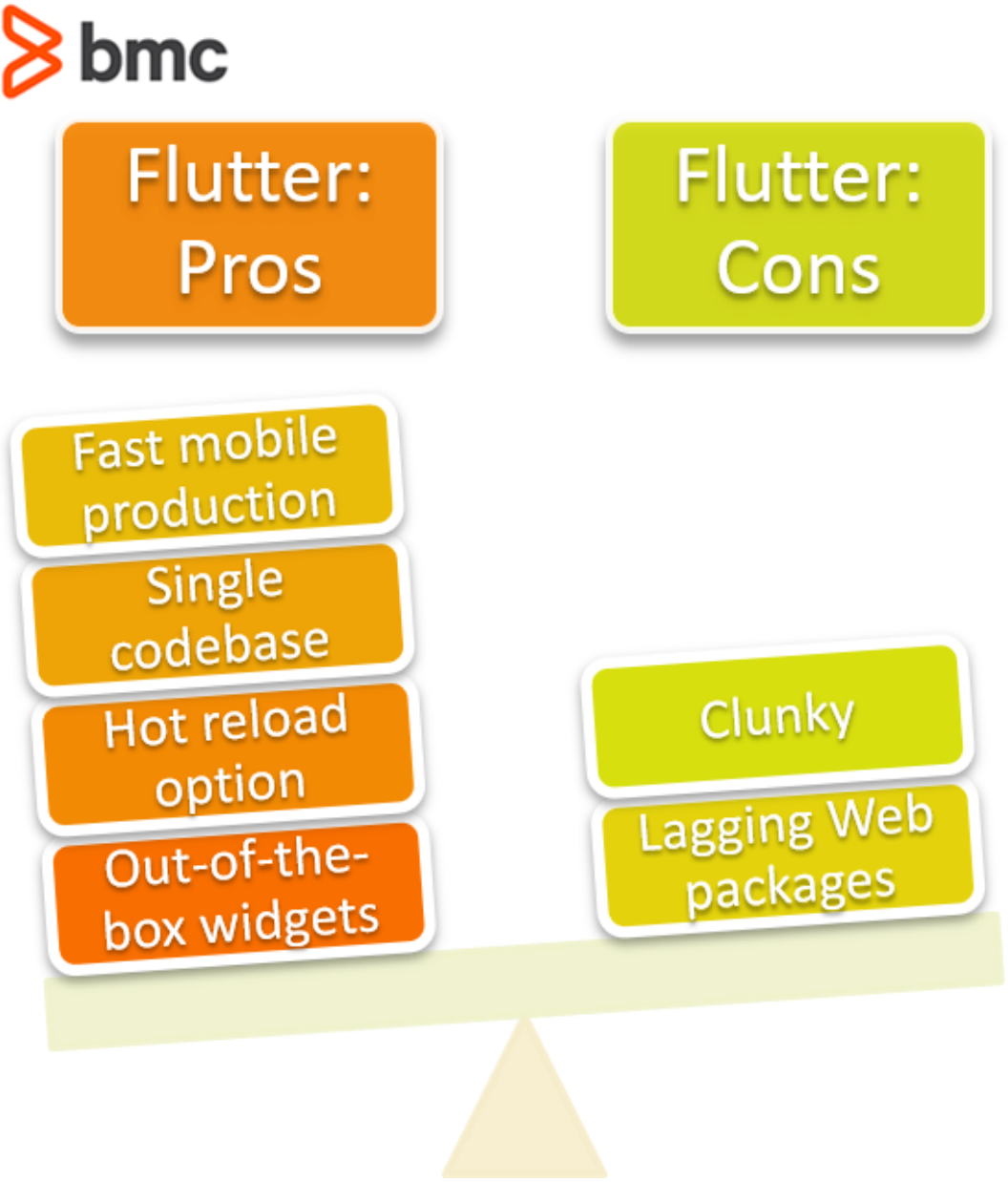
So, why all the hype about Flutter? Here are the big pluses:

- Flutter lets a developer make a product quickly.
- Flutter lets a developer create cross-platform apps with a single body of code.
- Flutter has a hot reload option, so developers can quickly see their changes on an emulator.
- Many out-of-the-box widgets to let a dev quickly stand their app up.

The drawbacks of Flutter

Of course, there are some cons as well:

- Flutter can be slow and clunky.
- The iOS and Android versions are well developed, but its Web packages still lag—due primarily to its newness.



Flutter in practice

Flutter can be used in a variety of use cases. The Flutter team reports a few major companies are using the app to prove big players have adopted the tech and it works for their uses. Among them are Square, Tencent, The New York Times, and Realtor.com.

It appears that those who use Flutter don't use it for their entire site. Flutter is being used to provide simple apps adjacent to their primary products:

- Realtor.com is not relying entirely on Flutter for its site. Instead, Realtor has adopted Flutter to serve a specific use-case, A-B testing, in their development process.
- The New York Times uses Flutter for the KenKen Puzzle.
- Flutter helps power apps on Google Assistant.

It is also fair to say, because it is new, companies cannot transition all their code over to using Flutter at once. It's possible what we are seeing are companies slowly getting their feet wet with Flutter.

Here are some ways you might experiment with Flutter:

Hackathons

Lately, Flutter is gaining popularity at hack-a-thons, where developers are pressured to create demos in just a few days. Flutter lets developers spend more time creating a feature-rich product, and less time writing boiler-plate code to put it in action.

A/B testing

Developers use Flutter to A/B test new product features in tandem with native applications. The idea is that Flutter can be used to create quick tests. When a feature proves to be a good fit with a community, then the feature can be written with stronger code.

Simple products with few features

Flutter is being adopted to provide low-risk appendage, or bolt-on, applications, such as a simple game like the NYT KenKen, or to provide basic image detection functionality to a user.

Getting started with Flutter

Jump in with Flutter. Build your first app, test it on a device, and make some demo apps. Slowly lean on it, and test how much you can trust it. It looks like a project that is here to stay and will become more and more reliable in the coming months.

Additional resources

For more on application development, check out these BMC Blogs:

- Application Developer Roles and Responsibilities
- The Software Development Lifecycle (SDLC): An Introduction
- 3 Critical End User Experience Metrics for Application Performance