This page contains old annual roadmaps, which are provided for historical context.

2021

Areas of Focus

Null safety

We will be introducing Dart's sound null safety to Flutter, and shepherding the migration of the plugin and package ecosystem to null safety, including migrating the packages and plugins directly maintained by the Flutter team.

As part of this we plan to provide a migration tool, samples, and documentation to aid migration of existing code.

Android and iOS

We are continuing to address jank-on-startup performance issues.

We will work on supporting incremental downloads of assets and code from the stores (subject to each platform's limitations), allowing the initial download of applications to be much smaller than the full download, with data fetched on demand.

We will also seek to improve the performance and ergonomics, and reduce the overhead, of embedding Flutter in existing applications on Android and iOS.

In addition, as usual, we plan to add support for new features of the iOS and Android operating systems.

Web and Desktop

Our goal for 2021 is to deliver production-quality support for Web, macOS, Windows and Linux, in addition to iOS and Android, enabling developers to create apps across six separate platforms using the same SDK.

For Web specifically, our focus will be on fidelity and performance, rather than new features, as we drive to prove that Flutter can provide a high quality experience on the Web.

For desktop, in addition to ensuring a quality experience, we will also be completing our work on the accessibility layer, and adding support for showing multiple independent windows.

Improving the developer experience

We will continue to focus on removing friction points. One area of research will be around reducing the boilerplate needed to achieve common goals in Flutter. We will also build on our investment in migration tooling for null safety to

investigate the possibility of creating tooling that enables us to make breaking changes easier for developers to manage, which would enable us to make some long-desired improvements to our APIs that we have so far avoided due to their breaking nature.

Ecosystem

In 2021, we will continue to work with the community on the Flutter-team-supported plugins. The goal will be to bring the pre-release plugins up to production quality and maintain them at that level by being increasingly responsive to issues and PRs. We also plan specifically to make significant improvements to the WebView plugin.

Quality

We will have efforts around improving Flutter's memory usage, application download size overhead, runtime performance, battery usage, and jank, based on experiences with real Flutter-based applications. These may take the form of engine or framework fixes, as well as documentation or videos describing best practices. We also intend to improve our tooling to help debug issues around memory usage.

In addition, we will continue to address bug reports. In 2020, we resolved over 17,000 issues during the year, and our goal is to have at least that level of impact in 2021.

New features

While in 2020 we primarily focused on fixing bugs, in 2021 we plan to also add significant new features. Some are listed above. We also intend to make improvements to our table widgets and introduce some tree widgets, with support for large numbers of columns, rows and/or tree levels, and column- or row-spanning cells.

Release Channels and Cadence

Flutter offers three "channels" from which developers can receive updates: master, beta and stable, with increasing levels of stability and confidence of quality but longer lead times for changes to propagate. We plan to release one beta build each month, typically near the start of the month, and about four stable releases throughout the year. We recommend that you use the stable channel for apps released to end-users. For more details on our release process, see the Flutter build release channels wiki page.

We used to also have a *dev* channel which represented a level of stability between master and beta. At the end of 2021, we retired this channel; it is no longer updated.

2020

Areas of focus

Web and Desktop

At our Flutter Interact event in December 2019, we announced that our support for Web had progressed to beta-level quality. We intend to continue this work with the goal of having Web be supported as an equal peer to Android and iOS. We hope to similarly continue our work in making Flutter the best way to create desktop applications.

Our goal for this year is that you should be able to run `flutter create; flutter run` and have your application run on Web browsers, macOS, Windows, Android, Fuchsia, and iOS, with support for hot reload, plugins, testing, and release mode builds. We intend to ensure that our Material Design widget library works well on all these platforms.

We don't intend to provide desktop-equivalents of the Cupertino widget library in 2020.

Quality

Our other main goal is to improve Flutter's quality, fixing bugs and addressing a few of the most-highly requested features. This covers a wide range of areas but we have a particular focus on our Cupertino library and iOS fidelity, our support for the long tail of Android devices, and the development experience.

We intend to deliver on long-anticipated features such as our router refactor, instance state saving and restoring, and an improved internationalization workflow.

In general in 2020 we intend to primarily focus on fixing bugs rather than adding new features.

We mainly use the "Thumbs-Up" emoji reactions on the first comment of an issue to determine its importance. See the Issue hygiene wiki page for more details on our prioritization strategy.