In the interest of transparency, we want to share high-level details of our roadmap, so that others can see our priorities and make plans based off the work we are doing.

Our plans will evolve over time based on customer feedback and new market opportunities. We use our quarterly surveys and feedback on GitHub issues to prioritize work. The list here shouldn't be viewed either as exhaustive, nor a promise that we will complete all this work. If you have feedback about what you think we should be working on, we encourage you to get in touch (e.g. by filing an issue, or using the "thumbs-up" emoji reaction on an issue's first comment). Flutter is an open source project, we invite contributions both towards the themes presented below and in other areas.

If you are a contributor or team of contributors with long-term plans for contributing to Flutter, and would like your planned efforts reflected in the roadmap, please reach out to Hixie (ian@hixie.ch).

2022

You may also be interested in Google's discussion of its strategy for Flutter in 2022.

Areas of Focus

Developer experience

The area where we will spend most of our focus is the developer experience. It is our intent to create an SDK that developers love. This will manifest in a myriad of different areas, for example creating widgets or plugins that solve common scenarios, cleaning up existing APIs, introducing new APIs to simplify frequently-seen patterns, improving error messages, evolving our developer tools and IDE plugins, creating new lints, fixing bugs in the framework and engine, improving API documentation, creating more useful samples, hot reload on the web, and improving stack traces in Dart-to-JS scenarios.

Desktop

In 2022 we plan to bring our desktop support to the stable channel. We plan on focusing on testing and announcing one platform at a time, as they become ready, starting with Windows, then Linux, and macOS. A significant part of this effort is expanding our regression test suite to give us the confidence that enables us to expand on these efforts without breaking existing code.

Web

Regarding Flutter for web in particular, we plan to work on improving performance, plugin quality, accessibility, and consistency across browsers. We

also intend to make it much easier to embed Flutter applications inside other, non-Flutter, HTML pages.

Framework and engine

We will update the Material library to support Material 3. This is primarily motivated by our goal to improve fidelity with Android, though it is not limited to that platform. We intend to implement cross-widget text selection. This is motivated by our goal of achieving good fidelity with the web platform, though again it is not limited to the web.

We intend to improve the text editing experience on various platforms, for example improving our fidelity with desktop text editing conventions and our integration with iPadOS handwriting recognition.

For desktop and web we will provide a solution for menus (context menus and menu bars), including integration with the host OS (which is particularly relevant for macOS).

Finally, also motivated by desktop though again not limited just to that platform, we intend to experiment with supporting rendering to multiple windows from a single Isolate.

Dart

We plan to continue to evolve the language at a deliberately slow but steady pace. We expect to introduce one major feature in 2022 (probably static metaprogramming; we will make decisions based on our confidence that the feature will improve the language), as well as some minor language improvements, probably including improving the import syntax for packages.

We also plan to expand Dart's compilation toolchain to support compiling to Wasm, contingent on the timely standardisation of WasmGC.

Jank

In 2021 we resolved a number of issues around jank, but our conclusion was that we needed to entirely rethink how we used shaders. As a result, we have been rewriting our graphics backend. In 2022, we intend to migrate Flutter on iOS to this new architecture, and then, based on our experience with this, begin work on porting this solution to other platforms. In addition, we will also implement other performance improvements and performance introspection features, such as those which our new DisplayList system has made possible.

Planned deprecations

We plan to drop support for 32bit iOS in 2022.

Infrastructure

In 2022 we will increase our investment in supply chain security, with the intent to eventually bring our infrastructure in line with the requirements described in SLSA level 4.

Note: We maintain an archive of roadmaps from previous years in a separate page.