Flutter – A Sweet Dish, but Not all “Peaches and Cream”

Lessons I’ve learned, 3rd party add-ins I’ve used, customizations I’ve made, and problems I’ve suffered through while building a real-world Flutter app over the past few months,

Flutter – the Marketing Promise

- Everything you need “in the box” to build whatever you need
- Super-fast development times – “we built an amazing app with just a couple of devs in a couple of months”
- “Customers want a <u>single</u> UI for *both* platforms”, and Flutter provides that…with native UI elements for either platform as well, you know, just in case...
- Modern, advanced approach to ‘programming’ the UI
- Amazing development/debug/testing (IDE) tools

The Delivered Product – Quite the Sweet Deal

- The 1.0 release (even the betas) is very strong – very complete, very stable.
- Installation was fairly quick, easy and painless.
  - Hardest part is ensuring that Android is fully installed, up to date, and for Windows, that the emulator works
- There are hundreds of videos and blogs that demonstrate the various bits of Flutter. Many of them are exciting to watch (if you're in to these types of things ☺)
- The 'out-of-the-box' component (widget) selection is outstanding.
- The 3rd-party 'market' for custom/special components is impressive and growing.
- The on-line community seems to be 'big' and there seems to be answers out there for nearly everything. I have received plenty of help so far.
- Building an app has been mostly successful so far – my one code base is essentially platform-agnostic and runs equally great on both iOS and Android.
- Dart (new to me) is easily approachable and seems to be quite potent.

Is it *really* that good… That ”easy”…That fast…that complete, yada, yada, yada…?

- Short answer: No

So…then…what’s the deal?

- Numerous ‘basic’ functions/capabilities are not “in the box”.
- No built-in ‘architecture’ for MVC or MVVM, or the like.
- Some of the built-in widgets, and 3rd party widgets still need customizing to be fully useful.
- Delivering ‘appropriate’ UI on each platform is not exactly smooth.
- The number of ‘blue-screen crashes’ has been steadily increasing.
- Builds are taking longer and longer – copy and launch to connected devices is taking longer and longer and frequently does not ‘finish’.

Wait a minute…more issues?

- Yet another new language (for most developers)
- 3rd-party IDEs which have non-ideal degrees of integration with the framework/language itself
- A “step back in time” approach to programming whereby the ‘layout’ code is intermixed with the ‘logic code’ – what developers used to deride the original Microsoft ASP approach with: “Spaghetti code.”
- Unintuitive mix of imperative and reactive programming models.
- Incomplete set of programmatic interaction model with widgets.
- A huge set of delivered widgets (isn’t that a “good” thing?)

Numerous ‘Basic’ functions are not ‘In the Box’

Here is a partial list of the 3rd party add-ins I am using:

- Flutter_Secure_Storage (for writing global prefs/values to persisted storage on device)
- Progress_HUD
- Scoped_Model
- HTTP (from Dart, and, does NOT support Request Cookies!)
- Connectivity (isn’t even 100% useful !)
- Intl (for date formatting and such)
- Google_Maps_Flutter
- URL_Launcher
- Image_Picker
- Image_Crop
- Autocomplete_Textfield
- Dropdown_Formfield (for validating form fields which are dropdowns)

No delivered architecture for “Separation of Concerns”

- Yes, ‘stateful’ widgets have their ‘state’, which is kinda almost similar to a ‘model’.
- Numerous 3rd-party solutions
  - How to choose the ”right one”?
    - Answers are incomplete, ranging from “…it depends on your app…” to “…this one is easier to understand than the others…” to “…this one was started by Google…”
    - In other words, nobody knows…
  - Choosing any one can result in a substantial [re] development investment, and may not yield the desired results.
- I chose the Scoped_Model option.
  - It billed itself as the ‘easiest’ and ‘lightest’ while still providing ‘good’ benefits
  - I have had mixed results – some success, but some problems, resulting in re-development efforts…

The whole "one UI for both platforms" thing

- On the one hand, yes, some developers and/or customers prefer a single UI for both platforms. If you've seen it on Android, then you've seen it on iOS, and vice-a-versa.
  - Great concept – I agree with the 'core' of it.
- On the other hand…Apple (at least in the past) would 'deny' apps that did not 'look like' a true iOS app
  - Dialogs
  - Buttons
  - Progress indicators
  - Pickers
  - Form elements.
- Flutter provides TWO sets of widgets: one for Material (a gazillion options), and one for 'Cupertino' (a very small set of options).
  - Sometimes both sets of 'similar' widgets have the same programming surface, sometimes not.
  - Makes it quite challenging to 'write once'…
  - I have impletments and extended the 'Platform_Widgets' project, as well as my own 'Platform DateTime Pickers' widget.

The whole “speed of development” thing

- Their claim to fame: Hot reload.
  - Yes, that is a *great* feature. Generally very fast.
- The rest of the ‘system’ though is no faster or slower than any other system I’ve ever used.
  - You still have to write code
  - You still have to define layout and elements, etc..
  - You still have set breakpoints and debug, etc…
- From *my* prospective, the “speed of development” has been *slower* than other tools/system due to one key issue: The knowledge/effort to know when and how to use each widget, as well as the debugging effort when there are problems.

New Language + New GUI paradigm = Hard

- The Dart language is ‘similar’ but different than every other language I’ve used.
- Similar to JS and C# and probably others, but different none-the-less.
- Can it ‘do anything’ – From what I’ve seen, yes. It just does it differently, meaning lots of reference lookups… Not a blocker, just a speed-bump…
- The range of GUI widgets is mind-boggling at times. Knowing which widget to use, in which situation, is frustrating at a minimum, maddening at its worse.
- Knowing *how* to use each widget is similarly frustrating and difficult.
  - Which property will result in what happening? Some are obvious, some are not.
- The videos all make the last two points seem trivial… ah, to be a Flutter master….

IDE integration is less than ideal – Debug is hard!

- Visual Studio Code is ‘light weight and fast’, but not perfect.
  - This is what I use.
- Android Studio (basically IntelliJ IDEA + Flutter plug-in) is ‘heavier’ and ‘slower’, but more ‘complete’.
- It is *very* hard to ‘debug’ layout issues
  - The device typically displays ‘repeated error fragments/suggestions’ on its screen.
  - The debugger also outputs the errors/suggestions.
  - Both are somewhat cryptic and non-positionally indicative as to ‘where or what’ is not ‘working’. Most of the time I have no idea what ‘went wrong’ or ‘how or where’ to fix things. Extremely frustrating.
- Programming bugs are frequently impossible to isolate or ‘prepare for’ (with breakpoints). It seems that in many cases breakpoints are ignored or errors are generated before reaching them (yet are ‘caused’ by stuff ‘after’ them)… Illogical, I know…

Increased App size is leading to longer build times and run-debug problems

- Videos typically show build-times in under 10 seconds… I started out that way as well…
- Build times are now taking over 2 *minutes* sometimes. Is that good? It doesn’t ‘feel’ good, considering that my app is probably less than 10% complete/developed.
- While some ’crashes’ are ‘fixed’ with a ‘hot re-load’, many of my crashes are now requiring a full ‘stop-and-start’. Besides it taking *minutes* to stop-and-start, far too often that new ‘start’ does NOT result in the app starting up. Sometimes ‘nothing’ happens. Sometimes I get a ‘white screen’ with no debug output. I typically have to ask the OS to ‘stop’ the app, and then launch it again…VERY frustrating…

Assorted Bitching and Moaning…

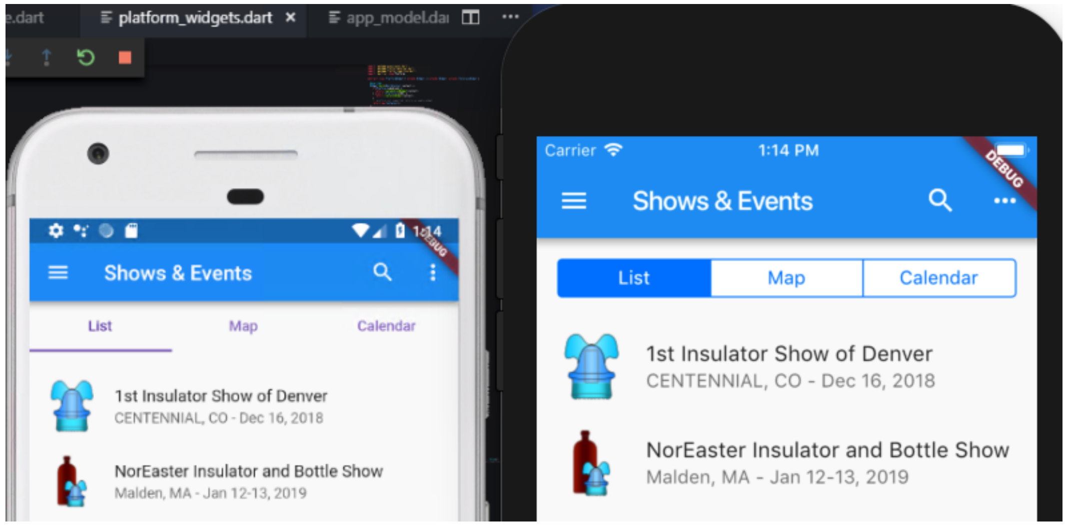
- Incomplete programmatic interaction model with visual widgets/elements:
  - *HyperCard* had this in 1988!!
  - Visual Basic has had this since 1995
  - iOS Xcode has had this since it was in the NeXT machine (1990?)…
  - HTML DOM/JS has had this in varying degrees since 2000
  - Pretty much every 'visual development' tool has had this.
  - How can Flutter have such a TINY subset of this?!?!?
- The list is too long to go into here.
  - Text fields (why two types?) should have onFocus, onBlur, onChange, onKeyDown, onKeyUp, etc… WAY too hard/messy to deal with text widget events and such…
  - Things like 'tab bars' should have onTabChangeBegin(cancel), onTabChangeEnd, etc…
  - Not enough time to cover all the elements and their missing events.

More Bitching and Moaning…

- The absence of ‘global’ variables in glaring. Yes, the Scoped_Model can suffice, but still, it is a 3rd party widget, and an explicit effort is needed to ‘use’ it wherever it is needed…
    - I understand that ‘reckless/lazy/stupid’ developers can ‘abuse’ globals, but ‘good’ developers can put them to good use.
    - This is not a ‘radical’ idea… ‘everyone’ else has them…
- Reactive programming is not ‘intuitive’ for me, and probably many other devs who have used the ‘imperative’ model in countless development systems over the past 30+ years…
    - Can I/Am I dealing with this? Yes. Doesn’t make it any ‘better’… Just sayin’…

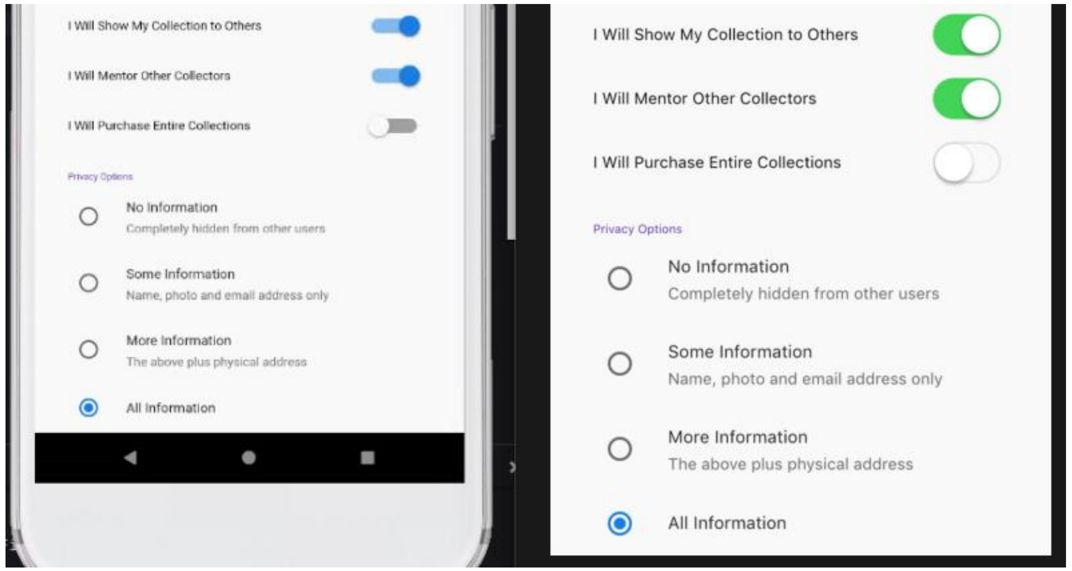
So if you hate it so much, why do you use it?

- First off, just like pretty much every other tool I’ve used in the past 35 years, I am in a ‘love-hate’ relationship with Flutter. There is plenty of it that I do love.
- It pretty much solves THE eternal problem: write once, run anywhere.
  - I develop on Mac, and my app ‘just runs’ in the iOS simulator and my OnePlus 3 Android phone without hardly any dickering around. It just works!
- There are a gazillion widgets which, if you can figure out where and how to use them, are quite impressive. And, you can tweak any/all of them you need for your own purposes.
- We’re *only* at version 1.0 folks… this ride is just beginning!
- Flutter will be [soon?] capable of outputting to DESKTOPS as well as mobile, not to mention the web. Talk about ‘write once, run anywhere’…



1st Insulator Show of Denver
CENTENNIAL, CO - Dec 16, 2018

NorEaster Insulator and Bottle Show
Malden, MA - Jan 12-13, 2019



I Will Show My Collection to Others ✓
I Will Mentor Other Collectors ✓
I Will Purchase Entire Collections

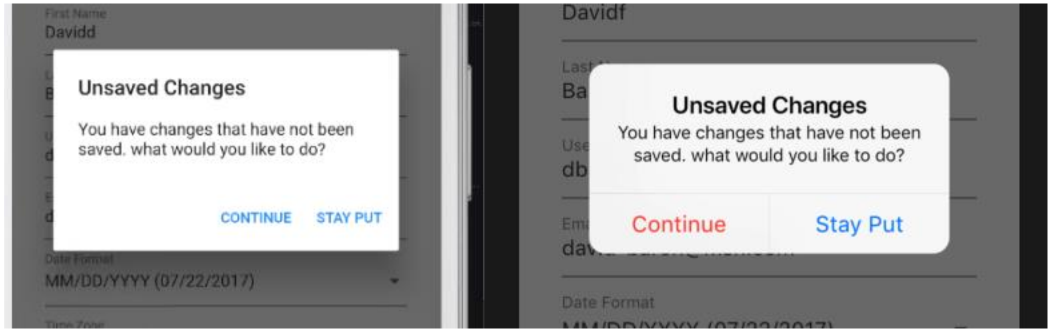
Privacy Options

No Information
Completely hidden from other users

Some Information
Name, photo and email address only

More Information
The above plus physical address

[✓ All Information]



Unsaved Changes

You have changes that have not been saved. what would you like to do?