

About me

I'm Tom, I'm Kineo's open-source learning technologies developer, and I work full-time on the open-source project (predominantly the authoring tool).

I've been working with Adapt since it was first prototyped (and not open-source). It wasn't a pretty sight back then!

Before Adapt, I worked on bespoke flash elearning courses (if anyone here still remembers flash).

I'm going to give you a bit of an introduction to the Adapt authoring tool, where it came from, what it does, and where we're looking to go in the future.

Hopefully you'll be able to take something away regardless of whether you're new to the tool, or already have some experience.

The market in 2013

Before I go into the where we are now, I thought it worth mentioning quickly setting some context for where we came from.

If we cast our minds back to 2013, the elearning landscape was slightly different.

A few years before, Steve Jobs announced Adobe Flash content would not be supported on the iPhone, which marked a big attitude shift away from

Flash and towards HTML.

In 2015, Adobe announced rebranded Flash, to mark a move away from Flash to HTML – they're actually still actively supporting flash until 2020

However, even in 2013 mobile elearning was still very much a nice-to-have – many companies had invested heavily in their Flash-based courses, and still required support for legacy systems, which HTML5 wasn't yet mature enough to support in the way that Flash could.

Our solution

Pete's already given a good overview of the project's philosophy and goals, but these 3 points are the most important in the context of the authoring tool:

Project aims:

- Market disruption: No product was ready to call itself 'future-proof' partly due to the transitional period of the time (shift from Flash to HTML5 on the web, HTML5 was still very much a buzzword) – no one knew who would be crowned champion – there were tools outputting HTML5, but no one was doing 'well enough'
- Open collaboration: still not something that's being done really anywhere else in the market, but even more so at the time – as I said in the previous slide, the big names in elearning authoring were expensive monolithic proprietary systems – the idea of giving something away for free when you could charge for it was unheard of – crazy considering the success of other learning tech projects like Moodle. Seems like a no brainer to pool development with like-

minded companies.

- This leads on to the next point: the technology itself became a level playing field, so competitors had to differentiate in other ways (often by providing additional services around the tool)

Our authoring tool:

- Modern technology: mongo, express, backbone, node. Considered other technologies like php etc. and decided that the modern MEAN stack lent itself better to heavily interactive web apps. Wanted technology to be shared across project, to allow shifting between workstreams.
- Browser-based: at the time this was fairly unusual for an elearning authoring tool, most were still desktop based, single user installs. The idea that you could handle multiple users in the same system hadn't been done to great success.
- FOSS: the key point, and differentiator for Adapt. By being free and open-source, Adapt raises the yard-stick for everyone else in the market – alternative tools need to go that bit further to justify a price tag.

Typist

Just before the 0.4.0 release we had a post from a member of the community asking about the 'project health' of the authoring tool – whether the project was still supported, and was still worth investing time in (as a user).

(slightly ironic due to the fact that the last 6 months has probably seen the most significant progress since the authoring tool project was first started – over 300 commits, and 150,000 code changes)

It's important to note that although we're 'only' on release 0.4, if we look beneath the surface of the releases at what's going on, we'll see that what we have today is the culmination of 13 releases, nearly 2,500 commits and almost 1500 issues closed.

We've also recently added a new collaborator to the project who is able to commit 2 extra developers to the authoring tool.

So for anyone with concerns about the rate of development...don't worry! I've also got a slide a bit further on which goes into more detail about our upcoming plans.

Version numbers

For anyone interested – we use the semantic versioning system to choose our release numbers:

Number split into 3 parts:

- Major when you break the API for existing users (not backwards compatible)
- Minor when you add functionality (backwards-compatible)
- Patch for small bug fixes (backwards compatible)

4 minor releases, 9 patch (7 v1, 2 v2)

You'll notice we haven't had any major releases – this means that the authoring tool is still essentially pre-release software, but only in the sense that we haven't yet hit our Minimum Viable Product (i.e. the met all the requirements we set out when first outlining what was 'good-enough' for a version 1.0.0 release)

Where we are today

As I've said, after 13 releases here's where we are:

(I won't go over all of these points, but to pick out a few)

- Latest framework: we always endeavour to support the latest release of the framework as it goes live (although this isn't always possible)
- No programming: Always very important from the start that the authoring tool targets non-technical users first and foremost, and so shouldn't require any experience coding
- Single source: there's no need to maintain different versions of your courses for different devices, one course works for all
- Extend: Additional functionality in the form of components, extensions, themes and menus can be loaded into the tool – including those developed by the community (i.e. not the core team)
- User management: a more recent addition is multi-user support,

A more competitive marketplace

Taking a step back and looking at the market now, in 2017, we can all agree that as elearning authors, we now have a lot more choice when it comes to the software we choose.

HOWEVER, none of them free and open-source!

It's important to note that because I think we're still relevant and needed in the market – again one of our initial goals was to target educational institutions and non-profits, and no one's doing that better than us.

To name a few, we have users championing Adapt from:

- Cambridge University
- King's College London
- The Open Data Institute

What's next for Adapt?

A couple of weeks ago (8th), we held a roadmap meeting among the core development team to discuss our next priorities

The main talking points coming out of that are:

- We're aiming for a few quick-win releases in the next few months for almost complete features
- Beyond that, we've got a few big ticket items that will likely run into the latter half of the year to 2019. Most of these are dependent on some reworking of the core code - to make things a bit more modular, so we're going to be focussing on that for at least the next release.

Tenant management

- Host multiple separate groups of users (e.g. clients) in a single install
- Separate content: assets, courses
- Feature includes some back-end code changes, but mostly adds a UI to administer tenants (similar to user management)

Course theming

This is probably the most requested feature.

We've had a prototype for this for about a year, but haven't pushed it out into the open in part because we were waiting for the front-end refactor which came out as v0.4.

We haven't completely pinned down how we want this to work, but it will likely introduce two levels of theme customisation:

1. Course-level changes: settings that affect an entire course (this could be general theme colours, fonts, font-sizes etc.)
2. Content-level changes: settings that only target individual bits of content (so this might be something like block background)

We're also considering the ability to add theme presets, which will allow you to save a state/collection of settings for reuse in other courses.

This could allow you to build a single theme for all your clients, and easily switch between different brand colours without the need to apply same the settings each time.

Getting involved

If you're new to Adapt or the authoring tool, like what you've heard, and haven't yet tried it out, there are a few ways you can get involved:

- The first thing is to obviously download it and give it a try. It may not be for you. For pointers on installing, there are a number of

sessions spaced around the two days.

- The next step is to join the community discussion on the forums and gitter (the links for these are on adaptablearning.org) – especially if you’ve got specific questions. The community is always very eager to help out newcomers, so don’t be afraid to ask for help! Also do have a good search through the existing forum posts, as someone else has probably already encountered your problem.
- If all that you’d like to contribute back, get in touch with someone from the core team, and we’ll point you to where your skills are best suited.

Getting involved: content creators

If you’re an instructional designer or content creator, the best first step is to get building courses, and feed back to the core team via GitHub issues or the forums/gitter.

We’re also always interested to hear feedback on the learning experience (LX), and areas that could be improved to help the end-learners.

As always, keep in touch via gitter.

Getting involved: developers

If you’re a coder, one of the best ways to get a feel for Adapt is to start writing plugins to extend existing functionality.

If you’re struggling for things to work on, it’s also worth heading over to GitHub and have a look through the open issues to in case there’s something that you can help out on.

As always, keep in touch via the community forums/gitter.

Join the team

If you're interested in getting involved in a bigger way, do get in touch with someone from the core team, and we'll be more than willing to discuss how you can give back.

In particular, we could use more expertise in UI and UX design (wireframing, mockups, prototypes) and testing (user testing and usability testing)

What you can expect from joining the team:

- We'll give you a point of contact/someone to get in touch with directly with questions/issues, and will give feedback.
- Suggest tasks based on interests/experience
- As with any core developer, we'll do code reviews on any contributions.

Questions

- For those that haven't yet tried, what's stopping you?
- What other authoring tools have people tried?
- Has anyone had any bad experiences?
- If you could choose to add one new feature, what would it be?

Adapt jam

We want to hold a more interactive session tomorrow (if there's interest).