Enabling Students to Learn How to Learn through Tech Talks

How can something that's constantly changing be taught to students? Software engineering students are often most keenly interested in learning the modern frameworks and libraries that software is built on. This is important: potential employers may want to see that new employees have the skills to hit the ground running. Yet, at the same time, these frameworks and libraries are constantly changing.

My core inspiration came from a visit with a professor at University of Waterloo. In his class, students were given assignments in which they researched a topic. This research was then collated into sections of a book that was being written, that students could read and could be shared with future students as well as others. In essence, students learned to become experts and teach other students.

In my teaching, I adapt this approach through a Tech Talk Assignment. Students first choose one UI framework, out of a list of popular frameworks, or suggest their own. In small groups of 3 or 4, students become an expert in this framework, finding online resources to learn about the framework and try it out themselves. Students share what they learn through a 15 min Tech Talk.

This offers several important learning experiences. Students learn how to learn, spending time identifying and reviewing learning resources to find something that is high quality. Students are given some structure to what they should learn through the structure of the tech talk itself, focusing on some of the big ideas around its design and architecture and how to use it. Another key emphasis of the tech talk is a comparative look at other frameworks. As students must eventually learn how to choose between competing frameworks, students are asked to explicitly compare and contrast the framework against other similar frameworks, highlighting its key motivation (why it was initially created) and the pros and cons against other frameworks. And at the end, students get practice in teaching all these skills to others, a practice they will need to do in industry to help mentor more junior engineers.

Students listening to the tech talk also learn. Rather than just learning about a single framework, which will soon change, might go away, and might not be the one they use in industry, they are instead exposed to a whole space of frameworks, learning a little about the key concepts and ideas, seeing how these are similar and different, and learning about how they might choose between them if they are ever asked to look for a new framework to adopt.

Example Assignment - Tech Talk Signup

Each student is responsible for giving a 15 minute Tech Talk as part of a two person team. Tech Talks must cover a front-end web technology (i.e., a technology which executes client-side in a browser and helps in constructing a user interface). Tech Talks must include:

1. What can you do with the technology. What is its reason for existing and key compelling use case.
2. Walkthrough building a really simple app that highlights a few key steps and considerations (note: it's ok and expected to use an existing demo app as an example)
3. Overview of design and architecture (key abstractions, patterns, etc.)
4. Status - commercial / open source, beta or current version, approximate size of userbase (if available)

5. Nearest competitors - what other technologies might one choose instead
6. Summary. When should you use it? What is it good for? What isn't it good for?

Here is a list of popular front end web technologies (you are also free to select others):

d3.js	https://d3js.org/
Vega-Lite	https://vega.github.io/vega-lite/
Angular	https://angular.io/
Laravel	https://laravel.com/
Ruby on Rails	http://rubyonrails.org/
Django	https://www.djangoproject.com/
Vue.js	https://vuejs.org/
Relay	https://facebook.github.io/relay/
React Router	https://reacttraining.com/react-router/
Redux	https://redux.js.org/
Foundation	https://foundation.zurb.com/sites.html
Bootstrap	https://getbootstrap.com/
Pure	https://purecss.io/
SASS	http://sass-lang.com/
Electron	https://electronjs.org/
Web Components	https://developer.mozilla.org/en-US/docs/Web/Web_Components

---

This approach can be extended by having students serve as an on-call expert for a topic. In Software Project Laboratory, I ask students to each select a practice that they want to become an expert in. Rather than teach specific practices, I teach strategies for learning practices. Each student is assigned a practice and is responsible for learning it and becoming its champion by teaching and overseeing its use by other students in the course project. Each student independently researches the practice, writes and updates a GitHub FAQ page on its use, answers questions from other students, and writes a public blog post on their experiences with it.