Refactoring Education

Internal Blog Post by Erica Lauer Vose 4/10/14

We know the true value of education lies in helping us do what we do *better*.
But it's not always obvious when that's happening in a classroom setting. The real test is in application of learning.

So why not make application the focal point of a training? This week, the Ed-Team partnered with APM team member, Katie Miller to create a refactoring workshop that featured pairing on real problems and real work in their own code base.

While a more traditional training format can take hours to produce, the set-up and planning took less than 20 minutes of Katie's time. The minimal planning requirements involved in this format for engineers can go a long way towards addressing concerns about engineer time. And the hands-on practice time ensures that engineers use training time to solidify what they learned *while* moving the company's product goals forward.

Here's how it works:

1. Take a bite-sized piece of a current, relevant problem the team can benefit from working on together.
2. Invite one or two team members who have an interest or particular skill in that area to present a short bit about the problem. The presentation should be simple and fairly straight forward. Ideas include:
   a. share a favorite resource (Katie used a 12 minute Rails Cast and a blog post)
   b. show examples (Katie showed her work on a problem with before and after examples)
   c. list best practices or a favorite method
3. Spend about 30 - 40 minutes working on real problems in pairs, small groups, or mob-refactor style. It's a good idea to establish at the start that you are not keeping the code so everyone feels comfortable experimenting and exploring. Obviously, the team can decide to keep something if it turns out particularly well. (I know at least 2 pull requests came out of this week's workshop)
4. Share back to the whole group with demos and code review. This is the most important part; by taking time to show off what each pair worked on, everyone gets to learn from each other and get new ideas or perspectives on the code.

There's more to this than meets the eye; this workshop format is a highly efficient way to share information amongst team members so that everyone is more consistent in both the code and team processes, it promotes team building, it is a concrete way to reinvest in employees through education, and it offers opportunities for career growth by letting individuals stand out and shine amongst their peers.