I want to formerly request a change to the development life cycle.

I propose we are given one day at the end of each release (once per month) for training, product changes, and general research. Initiate a code freeze to the core development branch. This will allow QA more time to test changes that will be pushed out, without any developer changes to interfere. The first day of the next sprint would allow QA to test any committed changes from the developers they completed during their training day. Engineers would create testable epics and stories, outside of the core development branch to keep the training day changes separate from that of our committed sprint work.

This would help sharpen skills, improve on areas of the product that engineers and clients care about, and help motivate developers to take ownership of the product. I volunteer to take ownership of outlining a full process around source control management, documentation, outlining acceptable research areas for each role, and management approval. Working with the product team and developer management team for an approval process around epics, stories and general product changes.

If an engineer doesn’t have a research topic or an approved story, they can choose a story from the bug backlog for their training day.

With the added responsibilities of design, development, testing, and automation, engineers have a lot of responsibility, but they are not motivated outside of that of an assembly line worker. These changes could give engineers the motivation to take ownership of their work and respective products. Let’s invite the company leaders to see programmers as active participants in the business who can drive business growth, just as the marketing or sales departments do within their teams.

There are companies seeing success by following this practice. Atlassian gives their programmers the opportunity to pursue a project of their choosing once a month, so long as they are related to a business objective. Sonatype gives their engineers an opportunity to train or work on one project or subject of their choosing, one day each sprint with the requirement that they update a confluence page each training day. More examples can be provided.

We could see positive impacts around client retention, performance improvements, product quality and reliability, motivation, engagement, and overall job satisfaction from all members of the cycle. No one would feel constrained or feel tempted to sneak in fixes.