ReviewNinja

Initial User Research Findings, June 30, 2014

Edited August 29, 2014 for GitHub.com (names removed)

Rachel Reynard
rachel.reynard@sap.com

User Research objectives

- To understand why and how developers conduct code reviews.
- To understand how code reviews fit into the development process.

Approach

- We selected teams from SuccessFactors and SAP developing cloud applications for Web or Mobile, using agile methods.
- We interviewed developers individually about development and code review.
- If a tool was used for code review, each interviewee was asked to demonstrate how they use it.

Users by broad preferences

1. “In-person” – much prefer in-person code reviews, and take a dim view of tools, considering them an impediment to meaningful communication about code.
2. “Hybrid” – value in-person code reviews, but see many benefits of code review tools for asynchronous communication, broadcasting impactful changes, and for more tactical changes.
3. “In-tool” – very comfortable using a tool for code reviews.

Representative quotes: hows & whys

“An old manager took a long time [with code reviews] because he wanted to make sure my coding style was compatible and consistent with the rest of the team. I think it's helpful for senior developers to bring that discipline to junior developers.”

“Coding is a discipline and an art, it should have some structure… …code reviews help improve structure.”

“[We do code reviews] primarily to have a second set of eyes and to flush out bugs. It's also a good way to have discussions about coding style and architecture.”

“If some things have been left out, some logic or something in the code, you want someone else to be able to double check these things.”

“With [tool] I'll invite the person who knows the code I’m working on best: typically one person. Sometimes I’ll pick someone random so they get familiar with the code in case I’m out of the office.”

“For something really critical (like a payment module) we like to have 3-4 eyes on it. Sometimes we'll add more reviewers if it impacts their area.”

Representative quotes: pain points

"If you're only reviewing the code, you're doing too little too late. The early discussions will have more impact on the end result."

"It's frustrating to have maybe 100 files all with the same change and they're all in the diff. If I'm making a small CSS change it's not a diff I need a review on, but the tool can't distinguish."

"It takes me 5 mins each time to remember how to post to [tool], it's a pain."

"One of the downsides of online reviews is that the reviewer may take a long time to respond."

"It would be useful to have more of an overview comment and then reference specifics."

"The way it works is if you click through files, they're set as complete [whether you've actually reviewed them or not]. It's helpful but it's open to abuse."

"Code review doesn't fit in very well to our process. It comes up a lot that it's something we should do more often, but it's cumbersome and too time consuming. The way we're doing it takes about an hour, because the reviewer has to pull down the changes, do a build and then review."

Key Results

1. Everyone without exception thought that code reviews were important and valuable. There were differing opinions about how and when to conduct them, but their perceived importance was unanimous.
2. Most interviewees thought they should be doing more code reviewing than they actually are.
3. Code reviews are optional for the majority of teams in the sense that you can “self-approve” with [tool 1], or choose not to initiate a code review with [tool 2]. In general, junior developers are expected to have their code reviewed.
4. Code reviews are conducted on an “as needed” basis, but mostly a daily activity.
5. Code reviews are widely used for communication and training.
6. Teams self manage and configure their process, environment, and code reviews to suit their needs. The teams we spoke to were well established, knew each other and each others’ area of responsibility well.
7. The enterprise teams we interviewed use the code review tool they’re given.
8. Someone external to the team selected the code review tool.

Next Steps

- Identify and engage external development teams.
- Develop UX Flows & Sketches (paper prototype) to test with developers we've already met, plus external development teams.
- Identify target *customers* and conduct follow-up interviews to understand their needs (distinct from the needs of end users).