

# 2020-06-24 Meeting notes

## Date

24 Jun 2020

## Attendees

- [Craig McNally](#)
- [Tod Olson](#)
- [Stephen Pampell](#)
- [Zak\\_Burke](#)
- [Vince Bareau](#)
- [Mark Veksler](#)
- [Marc Johnson](#)
- [Anton Emelianov](#)

## Discussion items

Time	Item	Who	Notes
5 min	Security Question	<a href="#">Craig McNally</a>	<ul style="list-style-type: none"><li>• Github alerts when there are known vulnerabilities in a repository... In some cases these have gone ignored for quite some time.<ul style="list-style-type: none"><li>• <b>Should the TC mandate that these be addressed in a "timely" manner?</b> Answer - No. Rely on Security Team's process.</li><li>• Leave it up to the security team to decide on a case-by-case basis?</li></ul></li><li>• <a href="#">Craig McNally</a> will raise for discussion at the next Security Team meeting (Friday)<ul style="list-style-type: none"><li>• Review and ensure JIRAs exist for all existing vulnerabilities reported by github</li><li>• Triage/prioritize each and assign to teams as needed.</li><li>• Investigate how to manage this better going forward - is there some automated way to get these into JIRA?</li></ul></li></ul>
10 min	UI Testing	<a href="#">Mike Gorrell</a>	<p>Discuss the need for investigation/review of our UI testing strategy and recommendations/requirements based on current tech/framework and state of coverage, level of effort, etc.</p> <ul style="list-style-type: none"><li>• Broadly, there are two types of UI tests: integration tests (full-stack) that operate against a production UI build backed by a live backend, and functional or unit tests (UI-only) that operate against a mocked backend.<ul style="list-style-type: none"><li>• integration tests are written in <a href="#">NightmareJS</a>; it is not currently maintained and uses an outdated version of the Electron browser</li><li>• unit tests are mostly written in <a href="#">BigTest v1</a>, a framework that is now officially deprecated</li></ul></li><li>• We have substantial investment in both Nightmare and BigTest, but given their lack of maintenance, we are a bit stuck.</li><li>• Additional unit testing challenges we face regardless of the framework:<ul style="list-style-type: none"><li>• We don't have a good way to manage mocks for unit tests in a way that keeps them in sync with backend changes.</li><li>• We don't have a good way to manage mocks for interfaces in a single place when multiple UI apps rely on the same interface. Currently, we maintain separate copies of mocks in separate UI repositories.</li></ul></li></ul> <p>Two real questions for now:</p> <p>1) Should we review the technology, process and expectations (80%) for creation Unit tests. Consensus is yes. We'll review the requirements for both UI and backend <a href="#">Anton Emelianov</a> volunteers to kick this effort off. Will follow up in 3 weeks.</p> <p>2) What do we do with modules that don't meet the test strategy requirements?</p> <ul style="list-style-type: none"><li>▪ Note that MARCcat is being taken up by @Cult and Scanbit - the expectations are that they'll have 80% coverage for Unit tests.</li><li>▪ Any module that has less than 80% needs to have "special exception" noted by Tech Council in order to be included in community builds.</li><li>▪ We need to have a process to document that special exception.</li><li>▪ UI-Courses had a "grandfathered" special exception because we thought tests were coming in Q2. It will not be included in the community releases in Q3 unless its code coverage meets our standards.</li></ul>
45 min	Architectural Blueprint candidates	TC Champions	We didn't get to this - update shifted to next week (July 1). Update the TC on <a href="#">efforts and discussions on their items</a> .