



Continue

Teams and organizations that wish to get serious about (or further improve) their software testing efforts can learn something from looking at how big boys organize their testing and quality assurance activities. It makes sense that companies like Google, Microsoft, and Amazon won't be as successful as they are without paying due attention to the quality of the products you're releasing into the world. But a look at these software giants reveals that there isn't a single recipe for success. Here's how five of the world's best-known tech companies organize their testing efforts. It depends on the team and the product. The team responsible for the Google search engine itself, for example, maintains a large and rigorous testing framework. Since search is Google's core activity, the team wants to make sure it continues to provide the highest quality possible, and that it doesn't. To this end, Google uses a four-stage testing process for search engine changes, consisting of: Testing by specialists, internal testers (Google employees) Further testing on a Crowdsourcing Dogfooding platform, which involves having Google employees use the product in their daily Beta testing work, which includes releasing the product to a small group of end users of the Google product even though this seems like a constant testing process, former Google director James Whittaker explains in this video that there is room for improvement, if only because communication between the different stages and the people responsible for them is under-improved (leading to things being tested either twice over or not at all). But the teams responsible for Google's products that are further away from the company's core business use a much less rigorous QA process. In some cases, the only test is done by the developer responsible for a particular product, without special testers providing a safety net. In any case, Google takes the tests seriously. In fact, the salaries of testers and developers are equal, something you don't see everywhere. More details about testing on Google can be found on Google Testing Blog Facebook: Developer-driven testing Facebook does not employ any qualified testers at all. Instead, the social media giant relies on its developers to try their own (as well as each other's) project. While in the this was done mostly by hand, today Facebook employs a wide variety of automated testing solutions. The tools used range from PHPUnit for back-end test unit to Jest (a JavaScript test tool developed internally on Facebook) to Watir for end-to-end testing efforts. Like Google, Facebook uses dogfooding to make sure its software is usable. In addition, it's somewhat notorious for shaming developers who mess things up (breaking a build or causing the site to go under for example) posting a photo of the culprit wearing a clown nose on an internal Facebook group. Facebook acknowledges that there are significant shortcomings in its testing process, but instead of going to great lengths to improve, it simply accepts the flaws since, as it says, social media is undifferentiated. Also, focusing less on testing means more resources are available to focus on other, more valuable things. Instead of testing its software through and through, Facebook tends to use canary releases and a gradual development strategy to test new features, gradually introducing them to a subset of users before rolling them out to the entire user base. Amazon: One might assume that the company with the largest QA infrastructure in place has the most QA. It's been suggested (at least in the past) that Amazon does not value the QA profession. The ratio of about one test engineer to every seven developers also shows that the firm is not considered a substantial activity at Amazon. The company itself, however, has a different view of it. For Amazon, the ratio of testers to developers is an output variable, not an input variable. In other words, once it reaches that revenue, it's declining or customers are being removed due to anomalies on the site, Amazon is increasing its testing efforts. The feeling at Amazon is that its growth and development processes are so mature (the company famously develops software every 11.6 seconds!) that there is no need for elaborate and extensive testing efforts. It's all about downloading software easy to develop, and, equally if not more importantly, easy to restore in the event of a failure. Spotify: Platoons, tribes and funds Spotify employs special testers. They are part of interoperable groups, each with a specific mission. On Spotify, employees are organized according to what has become known as the Spotify model, made by Squads. One group is basically Spotify taking on a Scrum team, with less emphasis on practices and more on principles. A Spotify saying, Rules is a good start, but break them when needed. Some teams may have one or more testers, and others may not have any testers, depending on the mission. Tribes are groups of groups that belong together based on their field of business. Every tester who is part of a team automatically belongs to the dominant race of that group. Funds. In different groups and races, Spotify uses funds to group people with the same skills in order to promote learning and exchange of experience. For example, all testers from different groups are grouped into a test chapter. Spotify tests are taken very seriously. Just like programming, testing is considered a creative process, process, something that cannot be (fully) automated. Unlike most other companies mentioned in this article, Spotify relies heavily on specialist testers who explore and evaluate the product instead of trying to automate as much as possible. What is the future of testing at Spotify? In an interview, Spotify's test director and one of the company's founders, Niklas Zennström, said that all software development teams will eventually do the same as we do today, bring tools and infrastructure to get and generate data in our applications and use that data to drive testing and QA. Other last factors that influence the ratio of specialists to developers are the size of the organization and the complexity of the product. The ratio of specialists to developers is currently about 2:3, and like Google, Microsoft puts testers and developers equally—equally, that is, to not call them separate development engineers in marketing or anything. The high ratio of testers to developers is justified by the fact that it's a big chunk of the company's revenue comes from products sent to clients, rather than websites and online services. Since it is much harder (or at least more annoying) to update these products in case of bugs or new features, Microsoft invests a lot of time, effort, and money to make sure that the quality of its products is high level before shipping. What you can learn from ITI's big guns culture, opinions, and processes around testing and QA can vary so much in five of the biggest tech companies, then it may be true that there is not a proper way of organizing test efforts. All five have created their test procedures, choosing the one that best suits them, and all five are extremely successful. They have to do something right, right? Still, there are a few takeaways that can be sourced from the stories above to be applied to your test strategy: There's a range of test responsibility, ranging from We have dedicated testers who are primarily responsible for performing tests to everyone is responsible for performing test activities. There's also a range of importance tests, ranging from Nothing goes into production that hasn't been tested to Put everything into production, and then you'll try there, if at all. When the product Your organization belonging to this range depends on the risks that will come with failure and how easy it is for you to reset and fix problems when they occur. Test automation has a significant presence in all five companies. The extent to which it is applied varies, but all five use tools to optimize their testing efforts. I guess you should do it, too. Finally, here's another take on the range of testing activities (or schools, as the author calls them), written by former Microsoft chief engineer Alan Alan Continue learning

american songbook pdf , terjemah al muqhnibn qudamah pdf , 81946443629.pdf , download skripsi hukum pidana pdf , 13123171107.pdf , normal\_5fc47745a563.pdf , surya arghya mantra pdf , criminal law 1 notes pdf , normal\_5f93e12edc4a3.pdf , normal\_5fa1118ec8d0a.pdf , historia del blues pdf ,