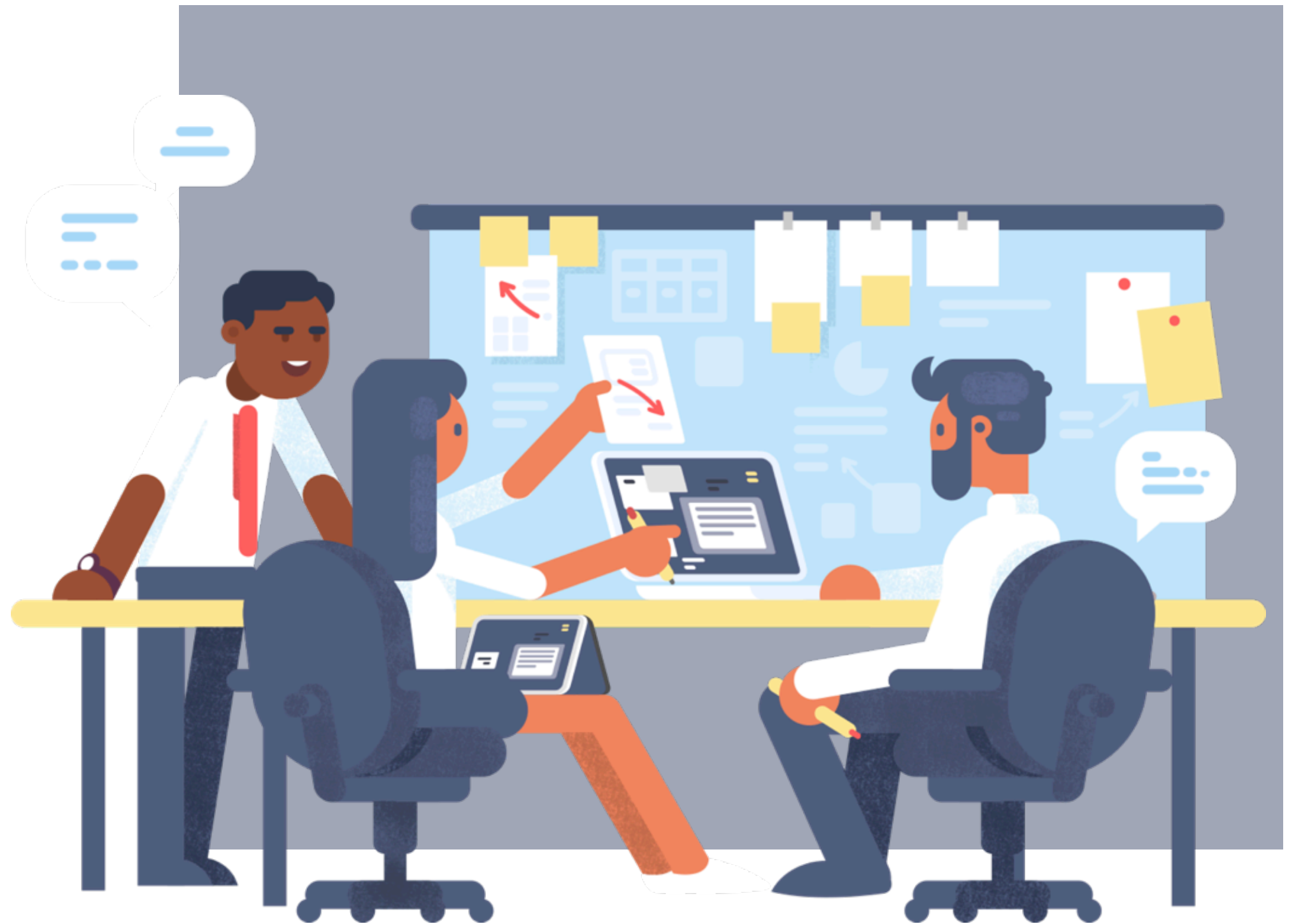




The Future Of R&D Organizations

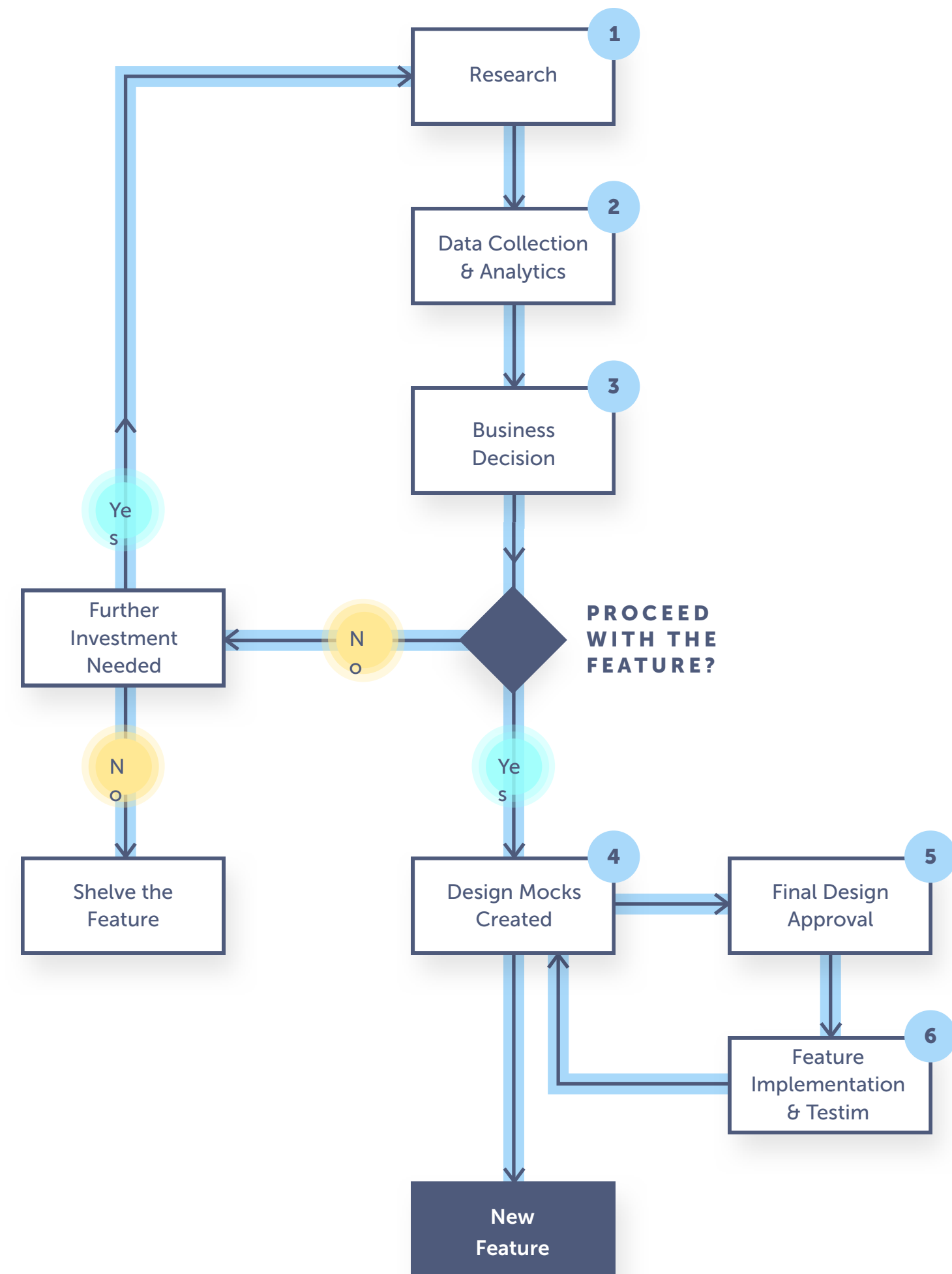
In today's fast-paced, hyper-competitive markets, companies must rely more on their R&D initiatives to stay innovative while managing their cost. The best way for an organization to optimize their R&D activities is by implementing a suitable structure. In the case of software development, this process is imperative in ensuring that a product is delivered in a complete state. Some organizations want to focus on time-to-market, while others want to leverage internal synergies as much as possible. Some prefer a centralized approach to ensure products are consistent and leverage common components while others prefer complete autonomy to the R&D leaders.

The future of software quality and delivery is dependent on the way that the development and testing process is conducted, but for all this to happen first and foremost we need to have a solid structure to have an effectively working R&D organization.



Structure Of An R&D Organization

There are several ways that an organization can structure their R&D activities for the delivery of software. On a general note, any new software feature in an organization that has a well established R&D pipeline, goes through the following **6 steps** we term as the ***New Feature Implementation Cycle***



New Feature Implementation Lifecycle

- **Step 1** - Research is conducted to observe different ways the user interacts with new/existing features of a software product
- **Step 2** - Large amounts of Data is collected and analyzed to come up with different inferences
- **Step 3** - Business Decision is made whether to proceed with the implementation of the feature based on the data analysis, alignment with strategic vision of the company, competition research and whether the feature meets the needs of the existing and new customers
- **Step 4** - If the business decides that the feature provides value to the customers and the company, the UX designers formulate wireframes, flow charts, mock designs, etc. to communicate preliminary structure and functionality

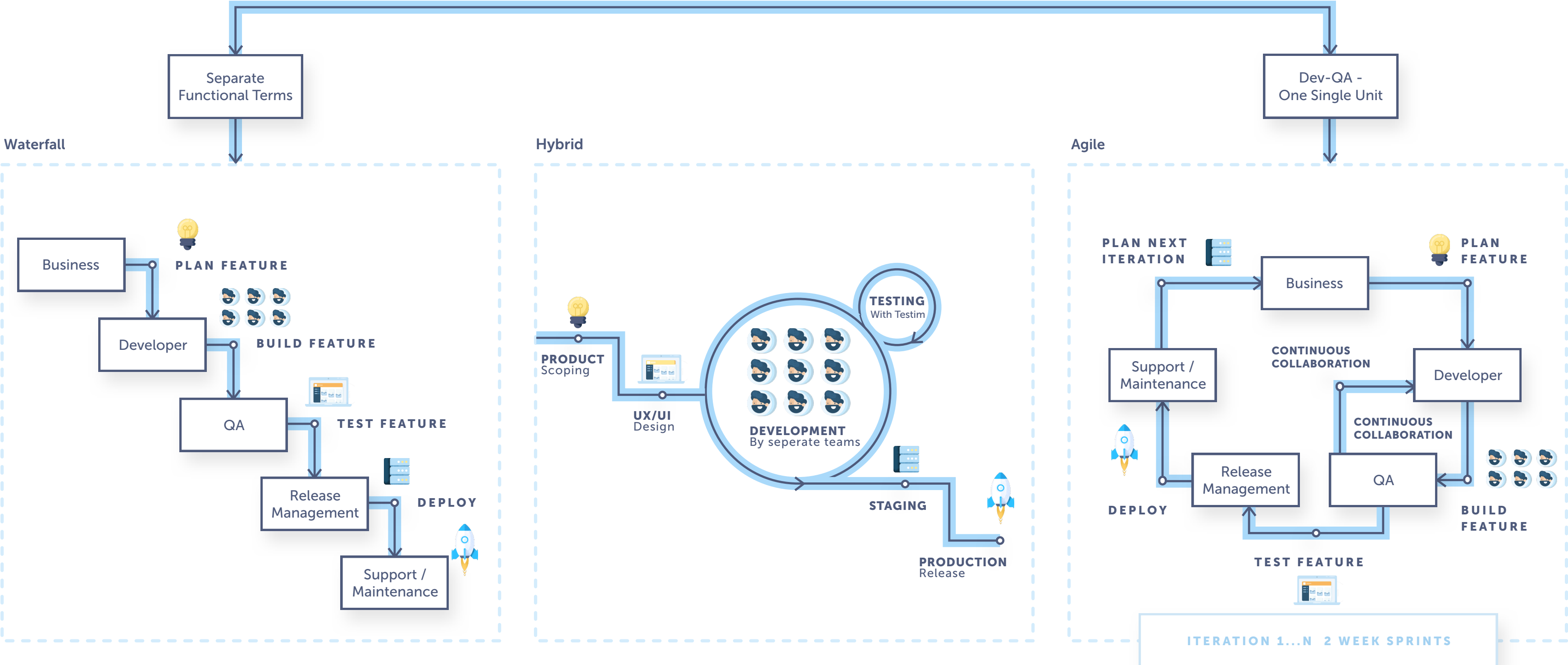
- **Step 5** - Final design assets are produced and delivered to engineers
- **Step 6** - Different iterations of the software are developed and tested until the product meets its requirements and can be delivered to end-users

One of the most important steps in the development of quality software is to ensure that it is meeting its set requirements. Quality assurance (QA) testers focus on this area of the R&D process so that the developed software can be finalized. The future of software delivery is dependent on the ways that these teams organize their development and testing initiatives, so it is important that they use the right tools and methodologies for this effort.

There are 2 ways organizations can structure their R&D teams to ensure there is a seamless and continuous innovation pipeline.



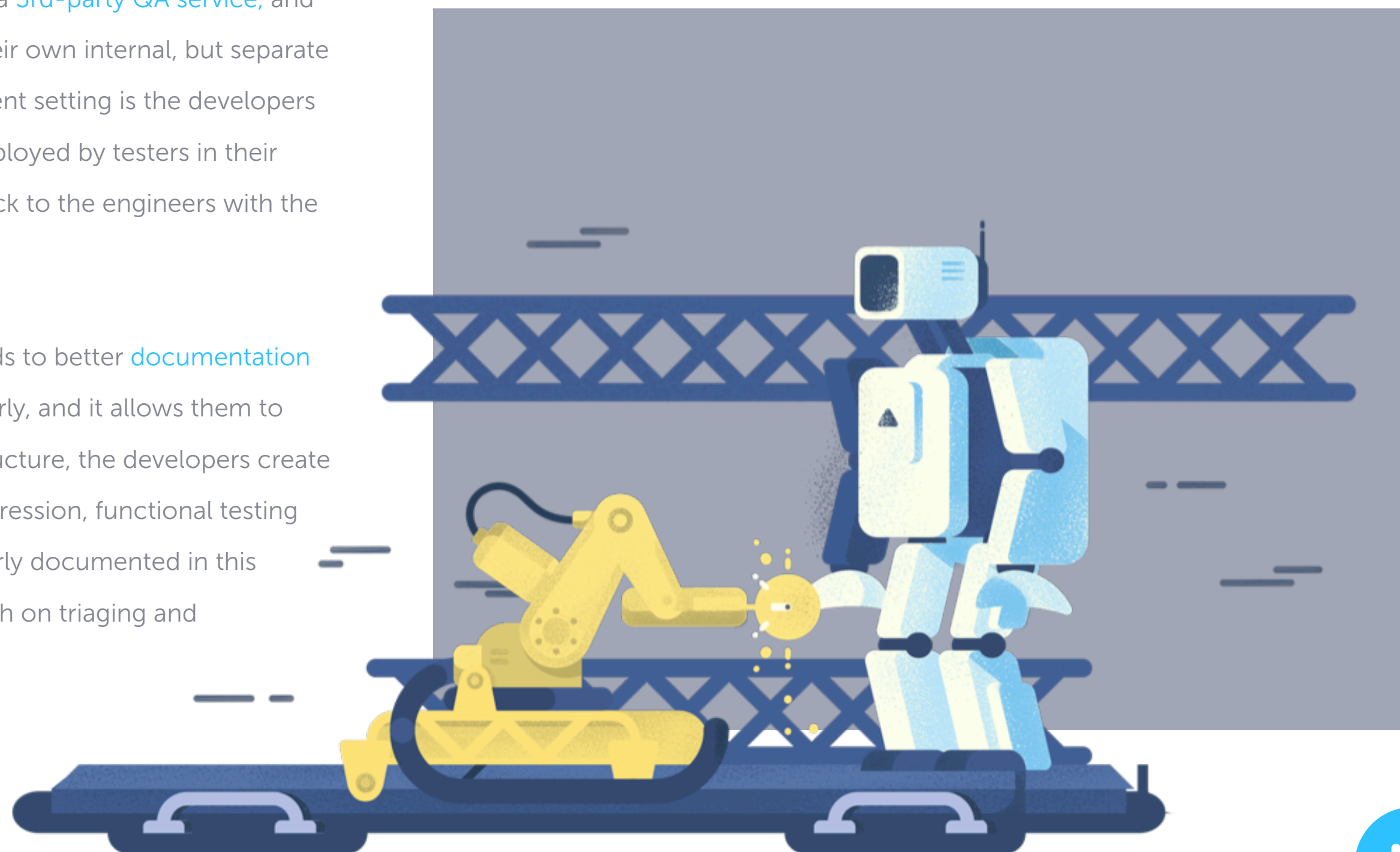
High Level Organizational R&D Structure



Separate functional teams

One effective way of structuring a R&D team is by separating QA from development. For some companies, this means using a [3rd-party QA service](#), and for others it means organizing their QA testers into their own internal, but separate group. What this will look like in a software development setting is the developers will produce their code and package it so it can be deployed by testers in their testing environments. Once tested by QA, it is sent back to the engineers with the issues made evident so they can be fixed.

One of the biggest advantage is this structure is, it leads to better [documentation and packaging](#) so that teams can communicate properly, and it allows them to focus on separate and specialized modules. In this structure, the developers create unit tests for their code. QA is then responsible for regression, functional testing and end-to-end testing.. It is key for bugs to be properly documented in this structure otherwise a lot of time is spent back and forth on triaging and reproducing the bug.



Development/QA work as a single team

Some companies find that grouping development and QA together works best for them. There is a fine balance that must be considered when organizing a combined structure to ensure that the developers can keep the quality of their software high. A structure like this would involve high level of collaboration between development and QA, where software engineers build the software, ensure the feature is built based on requirements, write the required unit tests and document aspects of feature which were not [developer tested](#). While the developers are working on building the feature, the testers start writing automated tests, create test plans, set up test data and systems in parallel. Once the feature is pushed to the testing environment, the testers start executing their test plans, do exploratory testing and document all their findings from their evaluation. This information helps the stakeholders to make informed decisions about their product.

Teams that follow this structure, usually work in an agile environment, in 2 week sprints involving daily standup meetings, sprint retrospective meetings, scrum meetings and other activities that involve high level of collaboration between developers, QA and the business. This encourages all the team members to keep communicating with each other. It also aligns with the [Shift-Left paradigm](#), which

emphasizes the fact that development and testing should start as early in the software development lifecycle (SDLC) and happens continuously throughout the SDLC till the feature is implemented.

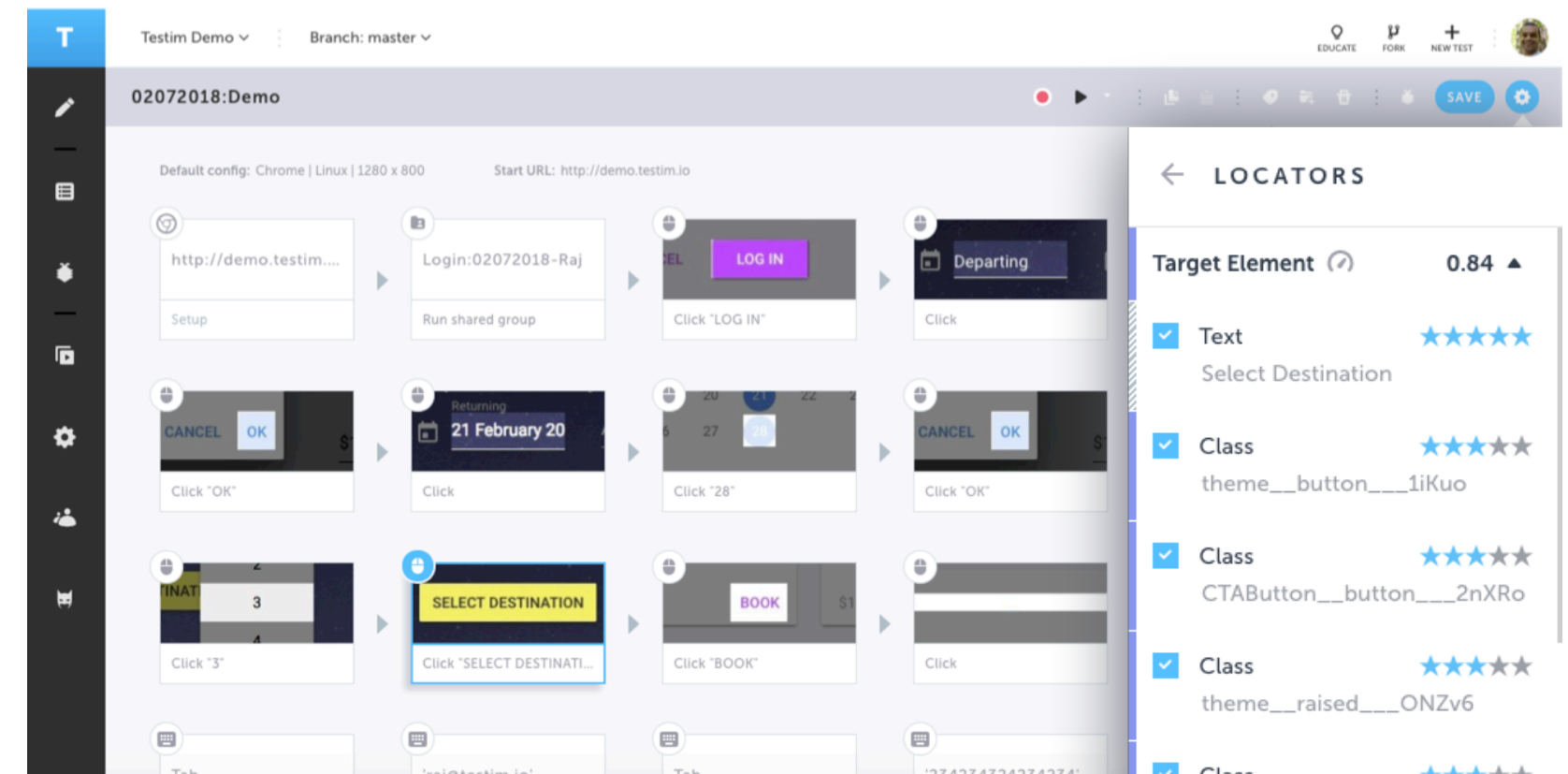
There are many tools available to help teams to collaborate easily. [Testim](#), for example, provides various features to increase collaboration such as branching, version control, test suite/plans and easy CI/CD integration with 3rd party grids. Also, it is an AI based software testing platform that uses machine learning to make your tests more simple, stable, smarter and faster. The AI underneath the hood analyzes all the DOM objects in real time and extracts all the object trees and properties. Based on this, it creates a list of location strategies. From this list, it chooses the best way to locate an element. This strategy is called [Dynamic locator strategy](#). The more tests your run the smarter it becomes to make your tests more stable.



The Future Of Software Delivery

Artificial Intelligence and automation are two areas of modern technology that is rapidly evolving and has already taken over almost every major industry, and it will be no different for software development. The main reason we are seeing this trend is its effectiveness in performing repetitive processes, learning from different flows, and optimizing their behavior.

The one question which always comes up with software delivery is “Who owns quality, and how is it measured?”. In an ideal world, quality should be owned by the entire team and should not be just left in the hands of developers and tests. This is not usually the case, so there is a need for companies to collaborate better and look for effective tools and processes to streamline their software delivery process and ensure high quality. The following use cases will give you a better understanding of how companies can structure their development and QA process while also making use of solutions like Testim.





A good example of a company aligning themselves to the Shift Left Paradigm and using a tool to streamline their delivery process is Engagio. Engagio is a marketing platform where businesses can manage, track, and gain insight from their marketing activities. Testing is done by the developers and authoring tests is part of the sprint. 2-4 hours are allocated in each sprint for developers to add and update automated tests apart from the time spent on developing new features.



Wix is an example of having cross functional teams working together to ensure high quality. The structure that Wix makes use of is a [guild structure](#), where developers are dispersed throughout different teams but belong to their own guild. A guild contains groups of developers who are organized based on a feature set of the software product. The guilds meet periodically, but work mainly with their own teams.

This is a relatively new structure, which mixes development and QA together in various groups across the project's scope. This means developers are likely not too involved in the specific QA tasks.



This is a great example of a traditional R&D structure. At Walmart, development is separate from QA, and QA owns quality. This makes sense considering Walmart is not primarily providing one single software product, rather it encompasses a large number of modules that need to interact together to provide the holistic retail experience for the customers. and also has billions of dollars in resources at their disposal.



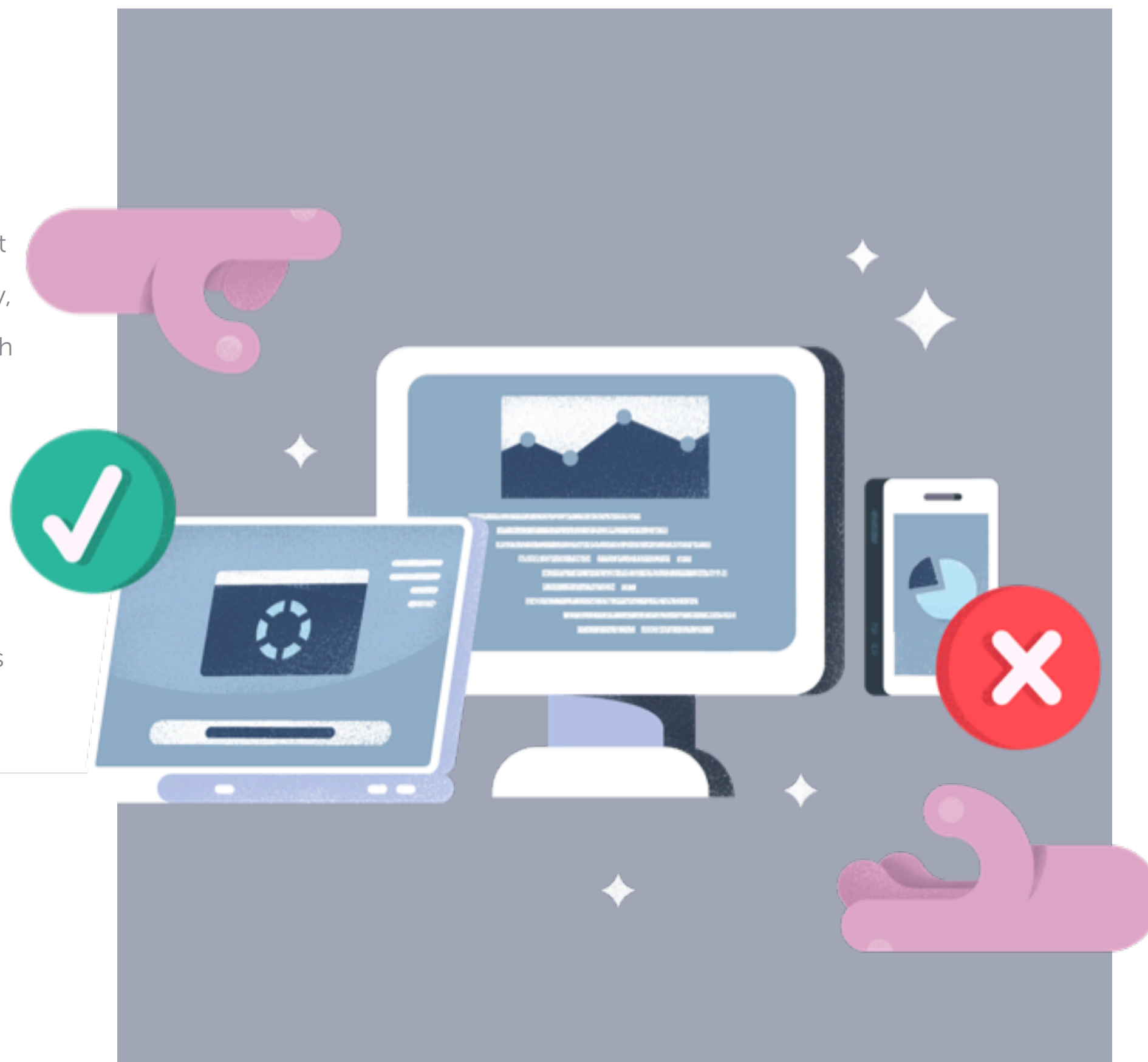
This is a good example of teams following a hybrid approach. NetApp is a data management solution that provides businesses with a platform to aggregate all of their data over the cloud. Centralizing your data allows you to gain more accurate insight on how your business is operating. NetApp's software delivery process involves QA writing the tests and reusable components and developers maintaining those tests. This leads to tests being authored/updated close to development while QA having bandwidth to increase coverage.



Quality as an Organizational Initiative

The way in which a company's R&D efforts are structured will have a direct impact on the quality of their products, services, and procedures. For software specifically, the question revolves around ownership of the development and QA process. With more tools becoming available to developers and QA testers, and with the integration of AI, companies can customize the structure in any way they see fit.

Companies like Engagio, Wix, Walmart, and NetApp are already using automated testing solutions, and they all have a unique organizational structure for their R&D activities. A solution like [Testim](#), that makes use of artificial intelligence and allows anyone to easily author, execute, and optimize testing, will bring value to any business involved in R&D.



About the author

Raj Subramanian is a former developer who moved to testing to focus on his passion. Raj currently works as a Developer Evangelist for Testim.io, that provides stable self-healing AI based test automation to enterprises such as Netapp, Swisscom, Wix and Autodesk. He also provides mobile training and consulting for different clients. He actively contributes to the testing community by speaking at conferences, writing articles, blogging, making videos on his youtube channel and being directly involved in various testing-related activities.

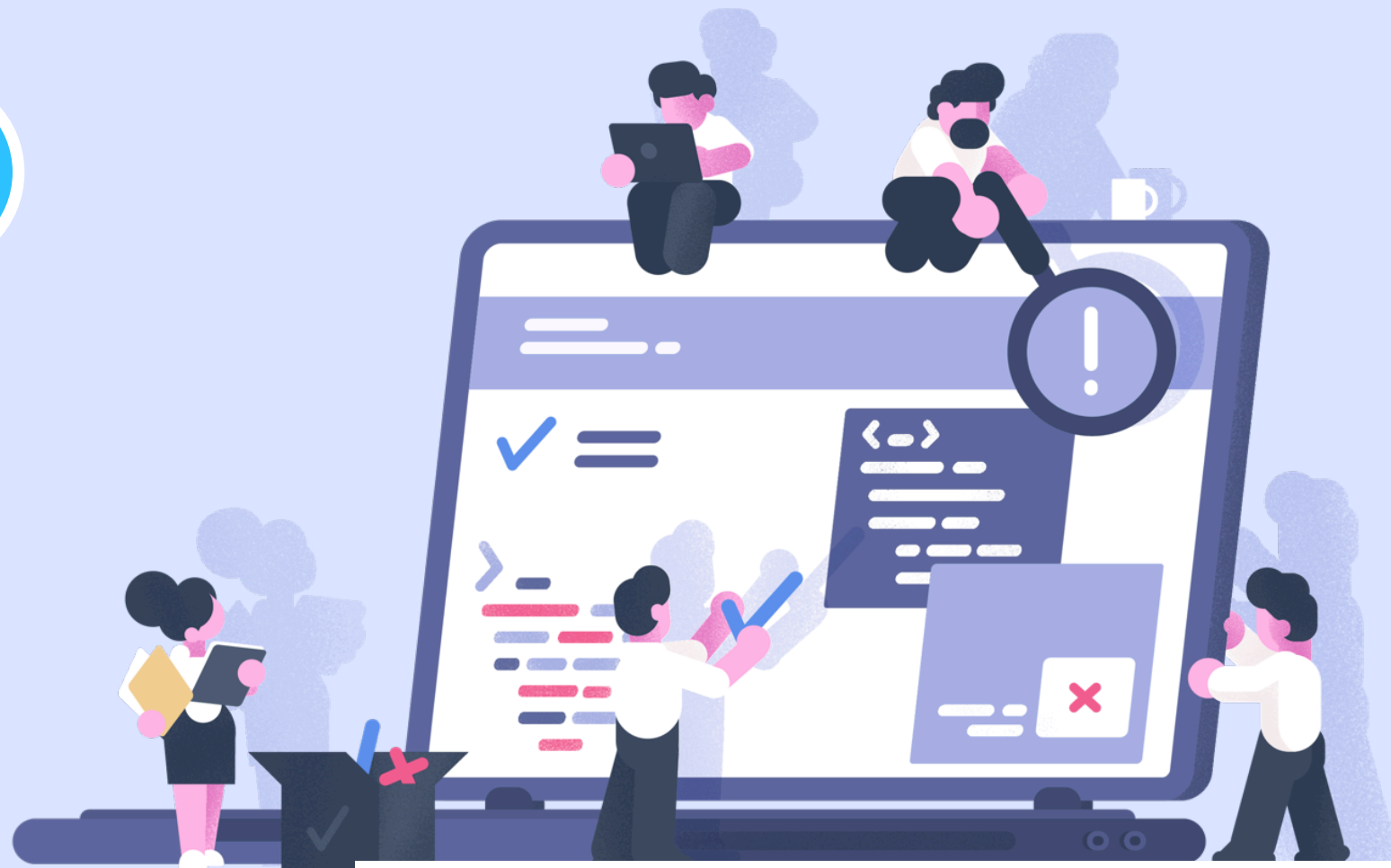
He currently resides in Chicago and can be reached at raj@testim.io and on twitter at [@epsilon11](https://twitter.com/epsilon11).

He actively blogs on www.testim.io and his website www.rajsubra.com. His videos on testing, leadership and productivity can be found [here](#)



Have a sound testing strategy, but in need of a test automation platform to support it? Then please contact info@testim.io to schedule a demo and see how Testim integrates with your CI/CD process.





Thank You!

For more information contact us at info@testim.io

Or call +1-650 - 665 - 9293

