



Software Quality for startups

Why startups fail
and how software quality
can help them succeed

According to Forbes, 70% of the “Next Billion Dollar Startups” for 2017 were technology startups (fintech, healthcare and biotech, data and analytics, logistics and new space industries). At the same time, 94% of participants in the State of Startup 2018 survey think now is a good time to start a company. Today’s digitization and growing startup ecosystems put software on center stage. It is no longer enough to have the right idea and develop it quickly, it now needs to be of the right quality from the start. This article highlights how startups need to balance speed with quality and discusses the necessary tools and processes that can help along the way.



Rali Genova MSc
is a Senior Consultant at KPMG
Advisory.
genova.rali@kpmg.nl

INTRODUCTION

Globalization has opened the floodgates to innovation. For the past ten years the world has seen a large expansion of startup and venture capital activity. According to [Flor19], global venture capital investment rose more than 200 percent, from \$52 billion in 2010 to \$171 billion in 2017. The annual venture capital invested in 2018 in the United States alone surpassed \$100 billion for the first time since the dot-com era to reach the astounding amount of \$130.9 billion invested across 8,948 deals ([PIBO19]). What is maybe less surprising is the fact that from the top 5 US IPOs (Initial Public Offerings) for 2018, four were tech companies (and the largest deal was a biotech company). Technology (and software in particular) has become prevalent in the startup world, which makes software quality more visible than ever.

Let's take the European startup scene as an example. According to a recent report on European early-stage technology startup activity by Tech.eu [Novi18], investment in early-stage technology companies across Europe has increased four-fold, from roughly €875 million invested in H1 of 2015 to more than €3.6 billion in H1 of 2018. The total early-stage investment in Europe for 2017 was €7.3 billion, of which FinTech and ICT investments accounted for 46.5% (see Figure 1).

And, while globalization, alongside technological advances and the proliferation of social media platforms have enabled fast and seamless communication, they also have had an impact on consumer behavior. Edward Deming, universally acclaimed as one of the Founding Fathers of Total Quality Management, wrote in his book *Out of the crisis* [Demi82] in 1982: "a dissatisfied customer does not complain: he just switches". The dissatisfied customer of today, however, will not simply switch, but will, most likely, complain all over social media. This is the reality in which startups must operate today. Therefore,

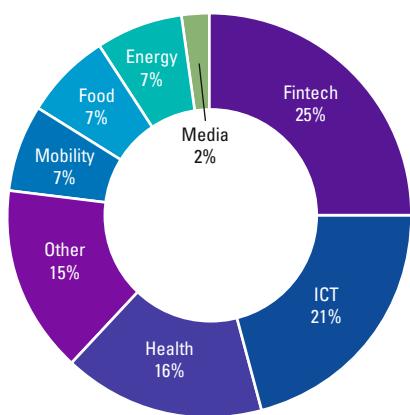


Figure 1. Sectors distribution by investment amount in 2017; chart recreated from [EBAN17].

A dissatisfied customer does not complain: he just switches

it is no longer enough to just have a cool idea, a garage and some cash to make it in the startup scene. Today, you need to be able to deliver quality, and fast, otherwise no one past the early adopters would use your product. And even early adopters, while initially lenient to the first few hiccups, may switch to a competitor if they consistently experience poor quality, version after version.

Startups must develop their products fast and get them on the market as quickly as possible, while still embedding quality into the development process in order to ensure that they are keeping their customers happy so that they don't switch products. CBInsights, a machine intelligence platform, gathering data on startups, analyzed the most common reasons why startups fail today. Of the top ten reasons, three were related to software quality, according to [CBI18]. So how can startups avoid failure, build great products and deliver quickly?

This article explores the need to find a balance between speed and quality. The next section outlines a methodology for how to minimize time-to-market and learn quickly about a startup's customers and their needs, developed by Eric Ries – the Lean Startup. However, in order to deliver quality, we also look at how to embed software quality tools and practices into the Lean Startup in order to maximize their chances of delivering quality software, fast.

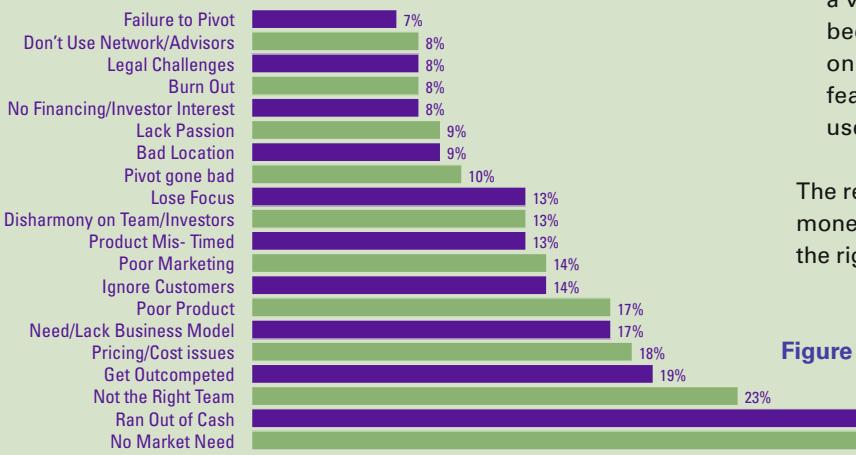
THE LEAN STARTUP – AN OVERVIEW

In 2011, Eric Ries published his book *The Lean Startup, a methodology for developing businesses and products* [Ries11]. It provides a scientific approach to product development and focuses on delivering results to the customers faster via an iterative process. The Lean Startup helps entrepreneurs answer two important questions: "Should this product be built?" and "Can we build a sustainable busi-

Why startups fail

Everyone is an entrepreneur nowadays, Sweden even offers unpaid sabbaticals to start your own business (which could explain why they have the most startups in Europe and are second only to Silicon Valley regarding the number of billion-dollar tech companies per capita, according to [Flor19]). Digitization of our lives means almost every startup needs software to power up their product (or in a lot of cases the software is the product) and with the increase of importance of software we see an increase of importance in software quality.

A study by CB Insights [CBI18] (a machine intelligence platform, gathering data on startups) of 101 startup post-mortems, revealed the twenty most common reasons for startup failure.



Of those twenty, three of the top ten can be linked directly to lacking software quality:

- #4 *Get outcompeted* – if an idea is hot, there might be more entrants to the market competing for customers. In this case, it is useful to watch what the competition is doing and build a product that not only solves a customer need, but also is easy to use (*usability*);
- #6 *Poor product* – Ignoring the customer's needs comes at a cost; 17% of startups failed because they developed something that most customers had trouble using (*usability*);
- #9 *Ignore customers* – building something just because you think it is cool might not be a viable business case. 14% of startups failed because they did not seek customer feedback on what they were building and instead put features on the market only they thought were useful (*functional usability*).

The rest of the reasons ranged from running out of money or not finding investors at all, to not having the right teams or even location for the startup.

Figure 2. CBInsights Top 20 reasons startups fail ([CBI18]).

ness around this set of products and services?" ([Ries11]). Instead of adopting a "Just do it" mentality, startup founders have the choice to use a structured approach to gain control over the chaos of initial product development and test their vision continuously. While the initial target audience was startups, the methodology can also be applicable to non-startup companies that want to innovate with new products or move into new markets. The Build-Measure-Learn loop, which we will discuss in more detail below, is a tool that can prove useful in directing product development for any company, regardless of the phase they are in – be it a startup, or an already established, recognized name.

Five principles of Lean Startup

The Lean Startup has five main principles which guide the methodology (see Figure 3 for an overview of each). Each of the principles focuses on an aspect of the startup approach, which, when done right, will lead to the success of the startup. The focus of the remainder of the article is the fifth principle – the Build-Measure-Learn loop and how to embed quality tools and practices from the start.

The Build-Measure-Learn loop

One of the core concepts of the Lean Startup is the Build-Measure-Learn loop – the way to continuously test hypotheses and measure results in order to improve and learn. According to the Lean startup model, the Build-Measure-Learn loop will minimize the time it takes a startup to push its initial product to market and later facilitate its refinement, based on learnings from early customer adoption.

What is not immediately evident from the model is the Pivot, one of the core concepts of the Lean Startup. Once the company goes through the Build-Measure-Learn loop, they are faced with a choice – "pivot or persevere". If data backs up the initial hypothesis (e.g. customers want a "Select All" button for their image browser app), the startup can "persevere" – continue with the next iteration in the same direction. If, however, data shows the hypothesis was wrong (e.g. customers don't even use that screen in the app, let alone care for a specific button on that screen), then the startup should "pivot", or change direction and explore other avenues. The pivot could be as small as covering a set of features to

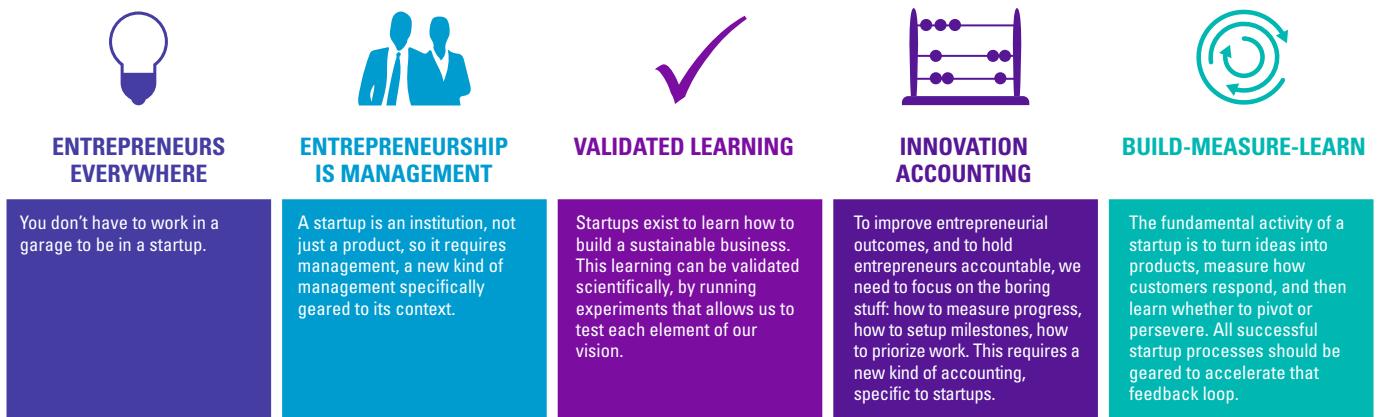


Figure 3. Five Lean Startup Principles. Adapted from [Ries11].

be developed, or as big as changing the strategic direction of the whole company. The pivot provides startup with a tool to always follow input from the market in their decisions on which direction to take.

By providing a structured approach, with a built-in validated learning, the Build-Measure-Learn loop enables startups to develop products (and sustainable businesses) by testing each element of their vision. The loop's main goal is to minimize the time startups spend between generating an idea and releasing it to the market. However, as we saw in the previous section, releasing a poor-quality product can kill any startup. By embedding software quality aspects into each phase of the loop, startups can enable a faster delivery while not compromising on quality. In the sections that follow, we will examine (per loop phase) the various software quality tools that can be used to strengthen the product quality.

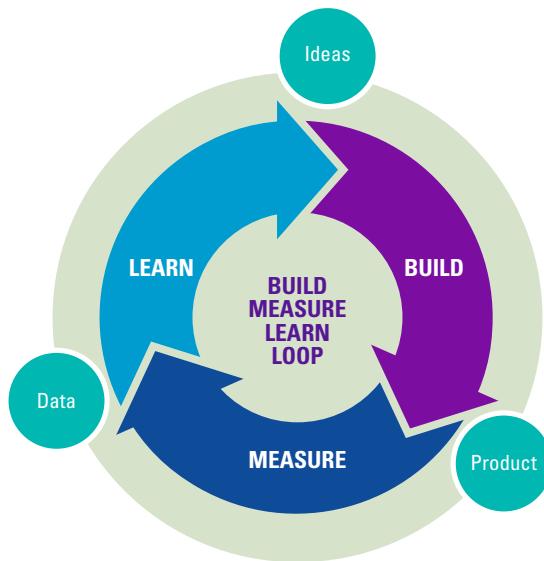


Figure 4. Lean startup 'Build-Measure-Learn' model.

Famous Business Pivots

Did you know that some of the most well-known companies were also a startup at one point and had to pivot in order to come to its current state? In 2013 Forbes contributor Jason Nazar published the list "14 Famous Business Pivots" [Naz13]. Below, we will take a look at three of those pivots – all from different industries, showing that pivoting can be successful for any startup, not only in the field of software.



An ex-Google employee started the predecessor of Twitter – Odeo. This was supposed to be a podcast platform, the team soon found out however that Apple was moving in that space and they knew they could never compete with iTunes, so the company decided to pivot – they had two weeks to find a new direction for their product offering. And so, Twitter was born – the status-updating micro-blogging platform we all know today.



Suzuki was not always known for their motorcycles and sports cars. In the beginning it was a weaving looms company, powering Japan's silk industry. However, the founder, Michio Suzuki, was an aspiring inventor and was looking for diversification. His pivot took him to producing small cars, and later the now famous motorcycles.

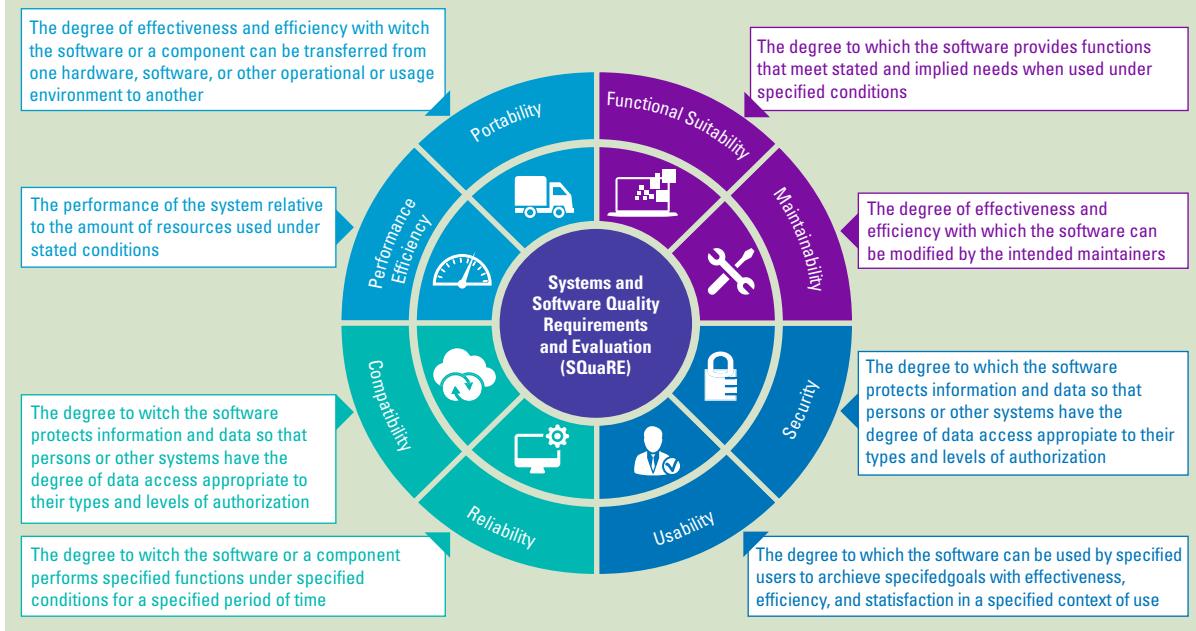


One of the most recognizable brands in American history, the company which gave us JuicyFruit, Spearmint and Doublemint did not always sell gum. In fact, in the beginning, Mr. Wrigley sold soap and baking powder, while giving the gum away for free. As soon as he recognized the value of the product he was giving away, he started manufacturing his own chewing gum and built the company around it.

Defining software quality

Quality is subjective and as such it can mean different things in different contexts. The same applies to software quality. Is software, which is free of defects, but is not easy to use by its customers of good quality? Or is easy to use, but unreliable in performance? To define software quality in more than one aspect, the International Standards Organization published the standard for software quality in 1991 – ISO 9126. It was later updated to ISO/IEC 25010 in 2011. The standard is a quality model that determines which quality characteristics will be considered when evaluating the properties of a software product. It further defines eight quality aspects of software that should be kept in mind during all phases of software development – functional suitability, maintainability, security, usability, reliability, compatibility, performance efficiency and portability. In addition, the model specifies five characteristics of software – safety, satisfaction, effectiveness, efficiency and usability. The standard can be used to evaluate the quality of a software product and each team can define which aspects they would like to focus on during the development lifecycle, depending on their context.

Figure 5. ISO/IEC 25010 on software quality.



Embedding quality into the Loop

The Build Phase

Before diving into the build phase, every startup should formulate their hypothesis: the market need they will be solving and how. Then comes the build phase - the first one in the Lean Startup methodology. Its main goal is to create a Minimum Viable Product (MVP) with which to test the main hypothesis. The MVP should have the bare minimum set of features needed to attract the interest of the early adopters of the product. What an MVP is going to look like depends on the type of startup – a simple landing page, an app with only the basic features, or, in the case of Dropbox, a video describing the upcoming features! Whether the startup chooses to build a software product for the MVP, or only once the MVP is successful,

it is worth implementing the following software quality elements in order to optimize the build phase: continuous code integration, continuous code quality monitoring, sandboxing and just-in-time scalability (or automated provisioning).

The tools and software quality practices recommended in the build phase help teams manage development and test environments, establish quality metrics (and monitoring source code quality on a continuous basis) and early and often integration of the code. Availability of sandboxes, so developers can experiment with different sets of features, speeds up the overall development cycle and allows the simultaneous testing of multiple hypotheses. There are great free tools out there (or for nominal fees at the first tier of users), so startups don't have to spend a fortune to ensure quality.

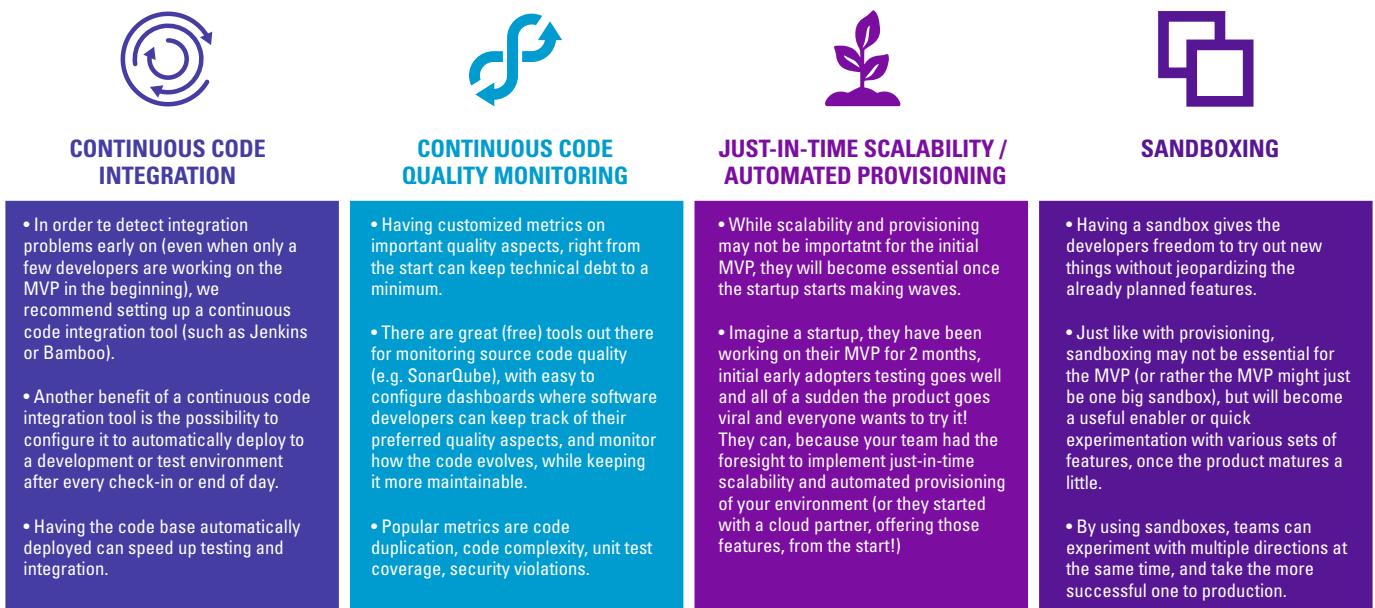


Figure 6. The Build Phase.

The Measure Phase

The main goal of the Lean Startup methodology is to give startups a structured approach to creating a viable business by continuously measuring against customer needs. During the measure phase of the Build-Measure-Learn loop, teams focus on collecting data from their experiment (hypothesis), which they built in the previous phase. The data will tell them whether they are moving

in the right direction and whether what they are building is solving a real customer need. One way to collect the data is via APM (Application Performance Monitoring) tools which continuously monitor various metrics that have been deemed important. Examples of such metrics could be page response times, number of logged in users or how many times certain resources have been downloaded by the customer.

Code quality tools

Examples of continuous integration tools, available for free (or “fremium”) are:



Jenkins

Jenkins – an Open Source continuous integration tool, written in Java.



Bamboo

Bamboo – a continuous integration server by Atlassian.



TeamCity

TeamCity is a Java-based build management and continuous integration server from JetBrains. A Freemium license for up to a hundred build configurations and three free Build Agent licenses is available, while Open Source projects can request a free license.



Apache Gump (only for Open Source projects) is an open source continuous integration system, which aims to build and test all the open source Java projects. To join Gump, a project must provide two XML files – one describes how to access the live CVS or Subversion repository; the other what to build from the repository, and the artifacts that need to be produced.

Depending on the programming language, there are various code inspection tools available.

One example of a continuous code quality monitoring, supporting multiple languages is SonarQube, which can be used in the cloud as a service, or installed locally and supports ABAP, Android (Java), C, C++, CSS, Objective-C, COBOL, C#, Flex, Forms, Groovy, Java, JavaScript, Natural, PHP, PL/SQL, Swift, Visual Basic 6, Web, XML, Python, Ruby, Go, Scala, Kotlin, Apex.

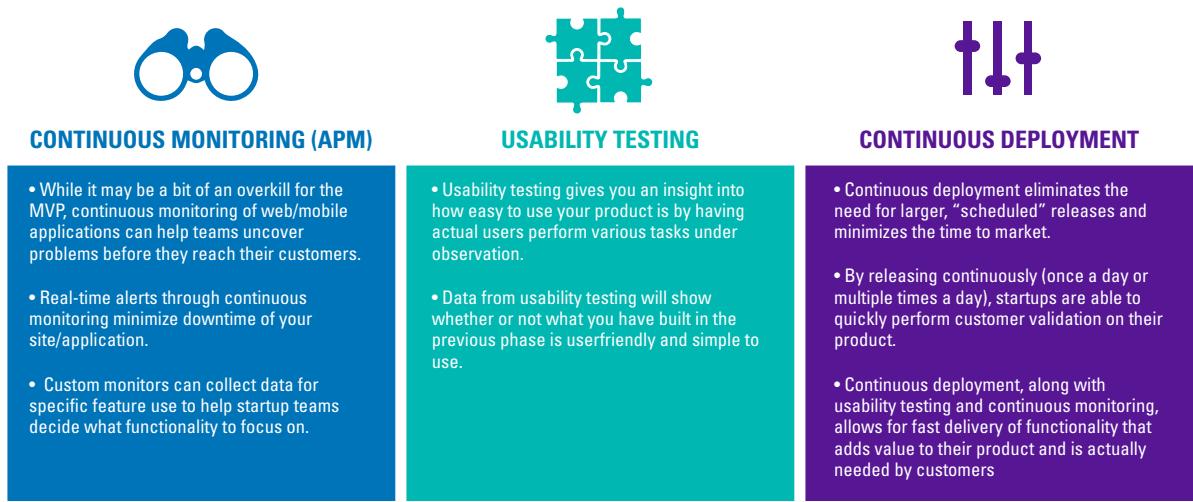


Figure 7. The Measure Phase.

In order to minimize their time inside the loop, teams should focus on continuous deployment (to deliver code to production faster), usability testing (to validate that code by customers faster) and continuous monitoring (to collect usage data and anticipate downtime/unavailability problems).

The Learn Phase

In the learn phase, teams use the information gathered in the previous phase to make evidence-based decisions on where to go next. There are two choices, as discussed previously – persevere or pivot. Based on which decision is made, the teams can use either split (A/B) testing, in order to test a new pivot, or smoke

testing,¹ to verify whether existing functionality is correct. In addition, UX design can be used as a tool to help adapt the user experience based on results from earlier customer testing.

Since this phase is focused more on making (evidence-based) business decisions, the focus on software quality is not as big as in the build phase, for example. The learn phase is also a way to close the Build-Measure-Learn loop and prepare for the next build phase by analyzing the collected data and adjusting the hypotheses.

¹ Eric Ries uses the term “smoke testing” in the Lean Startup methodology as a way of testing a lack of customer demand for a specific business model. The smoke test in the context of the Lean Startup may be simply a web page describing the services the startup is planning to offer, with a form to be filled out if people are interested. We have used “smoke testing” in its software context – testing to reveal severe enough defects, which will render the release unusable.

Figure 8. The Learn Phase.



Building quality software quickly and efficiently can make or break a startup

CONCLUSION

Venture capital investment in early-stage startups is at an all-time high and projected to keep rising in the upcoming year. Startups are becoming more global, and the share of tech startups has been growing steadily over the past few years. Now that everyone wants a piece of this pie, it may be more difficult than ever to succeed, especially in technology. Software has become an intrinsic part of our lives and building quality software quickly and efficiently can make or break a startup.

While startups are trying to find whether they have the right product, they shouldn't be wondering whether it is also of the right quality. Embedding software quality tools and practices from the beginning will empower startups to focus on what is important – fail fast, learn from it and improve. Building a sustainable business is a hurdle that many startups fail to overcome. Building quality software, however, should not be an unsurmountable challenge if startups make it a priority from the start. The tools and practices discussed in this article can help startups embed quality even in their early prototypes, so they can scale quickly when needed, or experiment with a new direction. It is important to note that many of the software quality practices, accepted as standard by larger companies, are not a good fit for startups (take for instance automated test suites – while they improve maintainability and stability of established software products, they would only slow down the evolving product development of a startup). When deciding which tools and processes to implement, startups should always ask themselves if they would impact their speed and flexibility to build, measure and learn until they zero in on a successful product.

References

- [CBI18] CBInsights. (2018, February 2). *20 reasons why startups fail*. Retrieved from cbinsights.com: <https://s3-us-west-2.amazonaws.com/cbi-content/research-reports/The-20-Reasons-Startups-Fail.pdf>
- [Demi82] Deming, E. (1982). *Out of the Crisis*. Cambridge: The MIT Press.
- [EBAN17] EBAN. (2017). *EBAN statistics compendium 2017*. EBAN.
- [Flori19] Florida, R. (2019). *Rise of the Global Startup City*. Retrieved from startupusa.org: <http://startupsusa.org/global-startup-cities/>
- [Nazari13] Nazar, J. (2013, October 8). *14 Famous Business Pivots*. Retrieved from Forbes.com: <https://www.forbes.com/sites/jasonnazar/2013/10/08/14-famous-business-pivots/#73a892c35797>
- [Novir18] Novick, N. (2018). *Seed the Future: A deep dive into european early-stage tech startup activity*. Tech.eu.
- [PIBO19] 4Q 2018 PitchBook-NVCA Venture Monitor. (2019, January 9). Retrieved from PitchBook: <https://pitchbook.com/news/reports/4q-2018-pitchbook-nvca-venture-monitor/>
- [Ries11] Ries, E. (2011). *Lean Startup*. Crown Publishing Group.

About the author

Rali Genova MSc is Senior Consultant at KPMG Digital Enablement. For over ten years, she has had different roles within the software development lifecycle – from developer, to business analyst, to release manager, test coordinator and product owner. Rali is also a key member of KPMG's Software Quality group.