



<b>Title</b>	WP6 – D6.5 – Business Plan
<b>Date</b>	November 29, 2019
<b>Writers</b>	Malena Donato, Atos Vincent Massol, XWiki Cedric Thomas, OW2 Pascal Urso, Activeeon
<b>Reviewers</b>	Benoit Baudry, KTH Ciro Formisano, ENG Caroline Landry, INRIA
<b>Online version</b>	<a href="https://github.com/STAMP-project/docs-forum/blob/master/docs/d65_business_plan.pdf">https://github.com/STAMP-project/docs- forum/blob/master/docs/d65_business_plan.pdf</a>



## Table Of Content

<b>1</b>	<b>EXECUTIVE SUMMARY</b>	<b>4</b>
<b>2</b>	<b>OBJECTIVES</b>	<b>6</b>
<b>3</b>	<b>INTRODUCTION</b>	<b>6</b>
<b>4</b>	<b>REFERENCES</b>	<b>6</b>
<b>5</b>	<b>ACRONYMS</b>	<b>7</b>
<b>6</b>	<b>MARKET OPPORTUNITY</b>	<b>8</b>
6.1	Aligning with Relevant Market Trends	8
6.2	The GitHub Market Opportunity	8
6.3	The Java Market Opportunity	9
<b>7</b>	<b>MISSION, TECHNOLOGY AND OFFERING</b>	<b>11</b>
7.1	Mission and Value Proposal	11
7.2	Technology	11
7.3	Market Entry with Descartes	12
<b>8</b>	<b>GO-TO-MARKET: THE GITHUB MARKETPLACE</b>	<b>13</b>
8.1	GitHub Apps	13
8.2	Revenue Model	13
8.3	Advantages of GitHub app model	14
8.4	Publication and Verification process	14
8.5	Implementing the GitHub App Model	14
8.6	Scalable Delivery model	15
<b>9</b>	<b>ORGANIZATION AND RESOURCES</b>	<b>15</b>
9.1	Stand-Alone Start-Up	15
9.2	Business Model Canvas	16
<b>10</b>	<b>COMPETITION AND MARKETING</b>	<b>17</b>
10.1	Competitive advantage	17
10.1.1	<i>Codacy</i>	18
10.1.2	<i>Coveralls</i>	18
10.1.3	<i>LGTM</i>	18
10.1.4	<i>TestQuality</i>	18
10.2	Competition Pricing Analysis	18
10.3	Marketing plan	19
10.3.1	<i>Web design</i>	19
10.3.2	<i>Marketplace</i>	19
10.3.3	<i>Social media and Content marketing</i>	20
10.3.4	<i>Developer's events</i>	20
10.3.5	<i>Free trials</i>	20
10.3.6	<i>Upsell</i>	20



10.3.7	<i>SEO and Affiliate program</i>	20
10.3.8	<i>Analyst endorsement</i>	20
11	<b>OPERATIONS AND GROWTH PLAN</b>	21
11.1	<b>Sales Forecast</b>	21
11.2	<b>Research, Development and IP</b>	22
11.2.1	<i>Descartes tool</i>	22
11.2.2	<i>SaaS Architecture</i>	22
11.2.3	<i>IP Management</i>	23
11.3	<b>Personnel</b>	23
11.4	<b>Costs of Operations</b>	23
12	<b>FINANCIAL PROJECTION AND FUNDING REQUIREMENTS</b>	25
13	<b>CONCLUSION</b>	26
<b>ANNEX I</b>		27



# 1 Executive Summary

This deliverable presents the final document prepared in the context of the STAMP WP6 and deals with exploitation and business planning activities. The objective of this deliverable is to present a realistic business plan for a start-up to exploit commercially the results of the STAMP projects.

The STAMP project has developed tools and methodologies for enriching and amplifying test assets in software systems. The test amplification techniques allow now to transform existing test assets in order to detect regression bugs before production and drive down the cost of software testing. STAMP automatically transforms testing assets that are manually written by human developers to improve testing effectiveness. For the design of a sound business plan, we focus on the Descartes tool, which is the most mature and stable tool available at TRL 8 at the end of the project (significantly more mature than the TRL6 initially planned in the STAMP project proposal). Thereby, we propose a business plan for a start-up that will distribute Descartes, in a software as a service delivery model through the GitHub marketplace. The document includes a financial projection and considers the commercial exploitation of Descartes only. Moreover, the market analysis has been updated to allow us to provide a market segmentation where STAMP/Descartes may present value. In addition, we provide an analysis for both the Descartes tool and STAMP in general, considering the current features and future evolutions of the STAMP tools. STAMP advantages are included as well, and the main features offered by each tool as well as the marketing plan, sales forecast and operational and financial projections.

## 1 Revision History

Date	Version	Author	Comments
30-jul-2019	0.01	Pascal Urso (Activeeon)	First TOC version
11-sep-2019	0.02	Pascal Urso (Activeeon)	STAMP Offering
27-sep-2019	0.03	Pascal Urso (Activeeon)	Introduction
2-oct-2019	0.03	Vincent Massol (XWiki)	Competitive Advantage
2-oct-2019	0.04	Cedric Thomas (OW2)	Marketing Strategy
7-oct-2019	0.05	Pascal Urso (Activeeon)	Product, R&D and staffing
9-oct-2019	0.06	Pascal Urso (Activeeon), Cedric Thomas (OW2)	Financial assumption
15-oct-2019	0.07	Malena Donato (Atos)	Market section, SWOT
18-oct-2019	0.08	Malena Donato (Atos)	Agile format and focused market
22-oct-2019	0.09	Vincent Massol (XWiki)	Survey
7-nov-2019	0.10	Malena Donato (Atos) Cedric Thomas (OW2)	Marketing strategy, STAMP offering, market segmentation
11-nov-2019	0.11	Malena Donato (Atos)	Objectives, introduction
12-nov-2019	0.12	Vincent Massol (XWiki)	SWOT, competitive advantage
12-nov-2019	0.13	Malena Donato (Atos)	Conclusion, business description
13-nov-2019	0.14	Malena Donato (Atos)	Release version for revision
15-nov-2019	0.15	Ciro Formisano (ENG)	Peer-revision - Overall revision and suggestions for improvement
19-nov-2019	0.16	Benoit Baudry (KTH)	Peer-revision - Suggestions for changes/questions
22-nov-2019	0.17	Cedric Thomas (OW2)	Corrections and improvements
25-nov-2019	0.18	Vincent Massol (XWiki) Cedric Thomas (OW2) Malena Donato (Atos)	Benefits to clients, general editions



25-nov-2019	0.19	Caroline Landry (INRIA)	Fixing title numbering issue and updating section 8.
26-nov-2019	0.20	Cedric Thomas (OW2)	Restructuring of the document
27-nov-2019	v1.0	Malena Donato (Atos)	Final version to send to coordinator & EC.
27-nov-2019	v1.1	Caroline Landry (INRIA)	Quality check and submission to EC Fixing section 8.5



## 2 Objectives

The present document contains the analysis done by all STAMP partners in the context of the work package 6 to agree on the common business issues and approaches. **The objective of this deliverable is to describe a realistic and scalable business plan for commercializing Descartes tool and analysing the context for STAMP outcomes.** The business plan needs to focus the business opportunity and the market target identified in the preceding Market Analysis deliverable according to the results obtained in the project. This target as well as the delivery mode and the operation of the business plan must be appropriate to STAMP results, i.e. new technologies and new testing methodologies, which concerns just emerge in the software engineering practices. STAMP advantages are included as well, and the main features offered by each tool.

STAMP has developed and validated test amplification techniques. Specifically, STAMP developed test amplification tools to increase the level of automation in software testing. STAMP focuses on test amplification in the context of DevOps and targets the early detection of regression bugs. In this context, the Descartes tool has been developed: Descartes evaluates the capability of tests suite in order to detect bugs using extreme mutation testing. Thus, we propose a business plan for a potential start-up based on this tool alone. The delivery model would be as SaaS - software as a service- through the GitHub marketplace.

Therefore, the financial estimation, marketing plan, sales forecast, and operational aspects consider the commercial exploitation of Descartes only. Moreover, the market analysis has been updated to allow us to provide a market segmentation where STAMP/Descartes may present value.

## 3 Introduction

As presented in the STAMP Exploitation Plan (Deliverable 6.4), all the partners of the STAMP project are engaged in the exploitation of its results. Academics partners are engaged in further development and research around the results obtained in the project, and the industrial partners have already integrated the STAMP tools in their continuous integration process and platforms. Concerning business exploitation, the partners which are specialists in consulting (Atos and Engineering) are offering consulting services around test design and automation that includes STAMP results.

However, behind these exploitation activities, to achieve an important economic fallout from the STAMP project results, we need to consider the opportunity and the risk of commercializing directly such innovative results. Indeed, the STAMP Market Analysis (Deliverable 6.3) shows that automated test execution, code quality concerns, as well as DevOps practices only begin to be mature. Pushing further automation, by creating automatically new tests, or evaluating the quality of the test themselves, is really seen as potential but pioneering software engineering activities. This constitutes both a risk and an opportunity, when proposing a brand-new market segment.

To assess these risks and opportunities we propose a lightweight and focused business plan able to

- be put in operation quickly and with agility
- address the most adequate focused market segment
- be commercialized and distributed with a minimum overhead
- scale up efficiently in case of success.

For all these reasons, which are detailed below, **we propose a business plan for a start-up that will distribute Descartes, in a software as a service delivery model through the GitHub marketplace.**

## 4 References

- [1] Blog, G. (2018). Retrieved from <https://github.blog/2018-11-08-100m-repos/>
- [2] Cosentino, V. I. (2017). *A systematic mapping study of software development with GitHub*. *IEEE Access*, 5, 7173-7192.
- [3] Datanyze. (2018). *Datanyze*. Retrieved from <https://www.datanyze.com/market-share/source-code-management/github>



- [4] Elizabeth Wasserman How to write a business plan, Harvard business school.
- [5] Foss, N.J., Saebi, T., Business models and business model innovation: Between wicked and paradigmatic problems, Long Range Planning (2017), <http://dx.doi.org/10.1016/j.lrp.2017.07.006>
- [6]
- [7] GitHub, W. (2019). Retrieved from <https://en.wikipedia.org/wiki/GitHub>
- [8] Rifkin, J. (2014). *The Zero Marginal Cost Society. The Internet of Things, The Collaborative Commons, And the Eclipse of Capitalism.* Palgrave Macmillan New York et al. 2014.
- [9] Schmitt, A. R.-e. (2018). Schmitt, A., Rosing, K., Zhang, S. X., & Leatherbee, M. (2018). *A dynamic model of entrepreneurial uncertainty and business opportunity identification: exploration as a mediator and entrepreneurial self-efficacy as a moderator.* Entrepreneurship Theory and.
- [10] TravisCI. (2019). *Travis CI GitHub*. Retrieved from GitHub Travis CI: <https://github.com/apps/travis-ci>
- [11] Warren, T. (2018). *Microsoft confirms it will acquire github for \$7.5 billion.* <https://www.theverge.com/2018/6/4/17422788/microsoft-github-acquisition-official-dea>

## 5 Acronyms

EC	European Commission
KER	Key Exploitable Result
CAGR	compound annual growth rate
CD	Continuous Delivery
CI	Continuous Integration
DevOps	Development & Operation
IDE	Integrated Development Environment
IT	Information Technology
QA	Quality Assurance
RoI	Return of Investment
SaaS	Software as a Service
SEO	Search Engine Optimization
TRL	Technology Readiness Levels
UI	User Interface
WP	Work package



## 6 Market Opportunity

### 6.1 *Aligning with Relevant Market Trends*

The software testing market has been experiencing an increase in investments over the years because the need to offer excellent apps without errors is nowadays crucial for companies and their survival. The testing market experienced a profound transformation due to the massive number of services and software needs such as new features, constant updates and tests to run correctly. The increased need for new advanced products and applications that happens in a constant disrupting environment forces companies to constantly reinvent and consolidate their testing tool offers.

Our previous Market Analysis survey (Deliverable 6.3), found that STAMP was aligned with key market drivers such as digital transformation and the adoption of agile and DevOps methods. However, the market analysis also found that the level of test automation as well as DevOps practices was only beginning to mature. Thus, STAMP exploitable results that push automation to a higher level, but require an existing test base, have to be proposed to a focused market target able to adopt them.

From this observation we see a market opportunity for STAMP defined and driven by two combined trends: the growth of GitHub the globally dominant collaborative software engineering platform and that of the Java language and the number of Java developers.

In a nutshell, GitHub users are the best candidates to be the early adopters. GitHub users, are a promising population of early adopters for innovative software engineering tools [CITATION Cos17 \l 1034]. This is demonstrated by the existence of GitHub marketplace, the first platform that allows to discover and purchase services that extend software engineering workflows already provided by the GitHub platform.

Proposing an app on this marketplace, will bring direct access, with very limited marketing resources to that ecosystem.

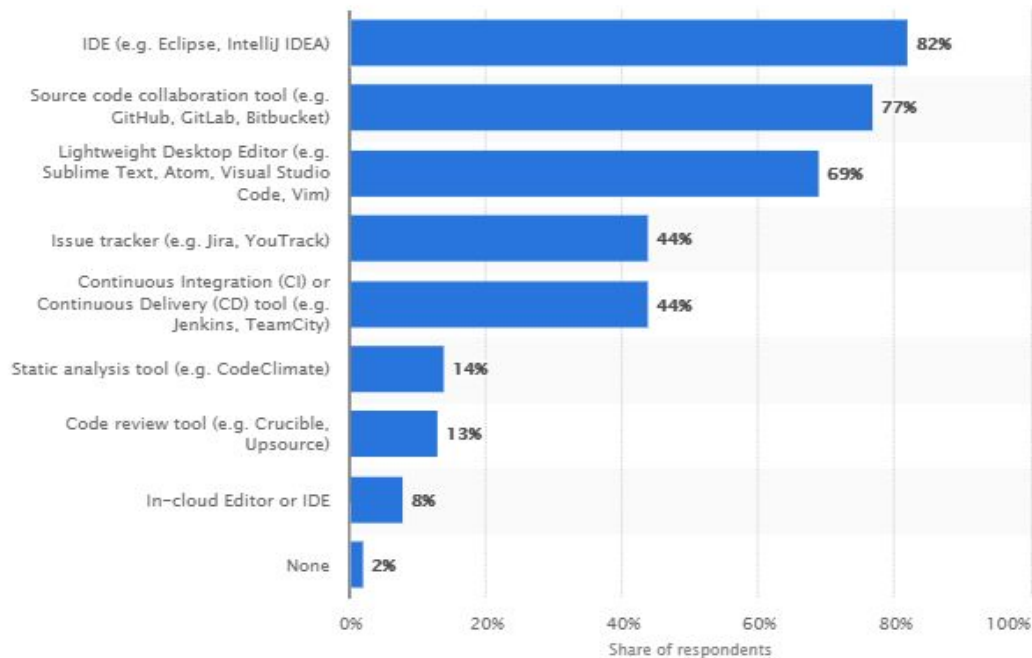
### 6.2 *The GitHub Market Opportunity*

GitHub.com, the open source project repository hosting, is one of the most used tools in open source and developer communities. In fact GitHub is so dominant that it represents a market in itself. Its growth seems to be unstoppable. In April 2016, GitHub.com, **had over 12 million users and 31 million public repositories** [ (Blog, 2018)]. And, according to Wikipedia (GitHub, 2019), in May 2019, GitHub had over **37 million users and more than 100 million repositories which 28 million are public and it is the largest source code platform of the world.**

In June 2018, Microsoft acquired GitHub for 7.5 billion USD. Moreover, GitHub has a total of 67 million repositories, 53 million monthly visitors, 1.5 million teams, 100 million pull requests, 1.5 billion commits a year (considering 2017), all totalling around half a Petabyte of code (Warren, 2018).

Considering the number of developers in the world, who potentially may use GitHub, based on the study from the Evans Data Corporation **there were 23 million software developers (in 2018)** and this number is expected to **reach 26,4 million by the end of 2019 and 27,7 million by 2023**. According to another analysis done by Datanyze (Datanyze, 2018) Git alone has almost 38% market share considering the 23 million software developers in the world. During 2019 GitHub has already 24 million users, although many of them likely aren't actual coders. The figure below shows the preferred tools used among programmers; the category **"source code collaboration platforms"** are used by 77% of software developers. The top platforms considered are: GitHub, Gitlab and Bitbucket and account by the most used tools in the world during 2018.





**Figure 1. Programmer tools by popularity**

[1] GitHub.com users and repositories: <https://en.wikipedia.org/wiki/GitHub>

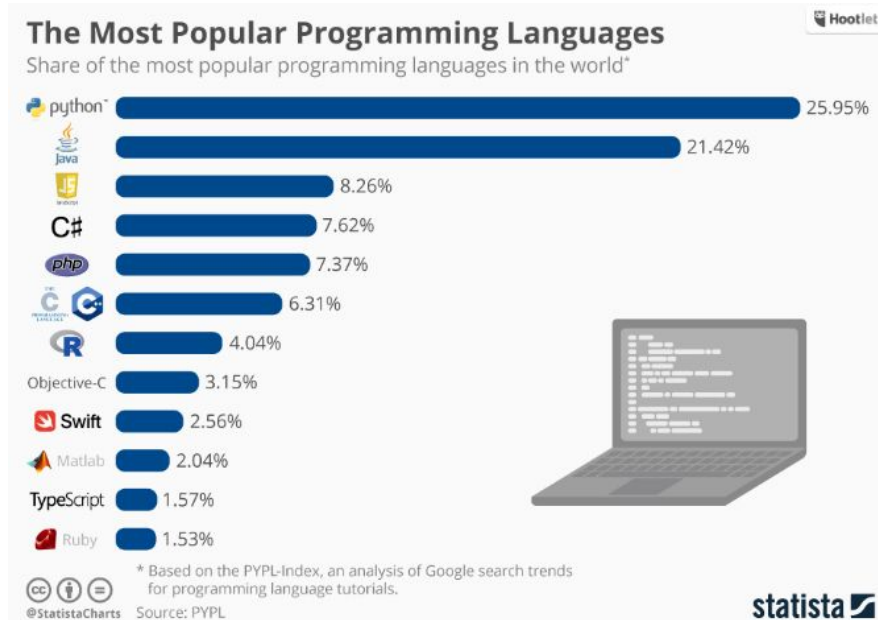
More precisely the GitHub service market is about half of GitHub's \$200 million in annual revenue and it comes from private businesses. GitHub offers different services available such as code libraries, project spaces and forums that are free to anyone who wants to create open-source software projects. Currently open source projects are available and any company that is willing to pay for a service can access it. The GitHub Enterprise offers additional features, including private workspaces, 24-7 support, and more dynamic cloud hosting options. For now, Descartes tool is available for free as a GitHub Marketplace app but could be placed in more premium services. Also, in the case of DSpot that uses Travis continuous integration (TravisCI, 2019) to build and test software projects hosted at GitHub, for the other tools in their CI/CD Jenkins is primarily used. In fact, we have run a survey asking the community of developers if they are willing to pay 5€ a month for using the tool that verifies the effectiveness of your tests and gives you indications and options on how to improve these. Out of 66 answers, 39% of respondents have answered yes, that they are willing to use the tool and pay for it.



**Figure 2. Survey run online for adoption/usage of Descartes tool**

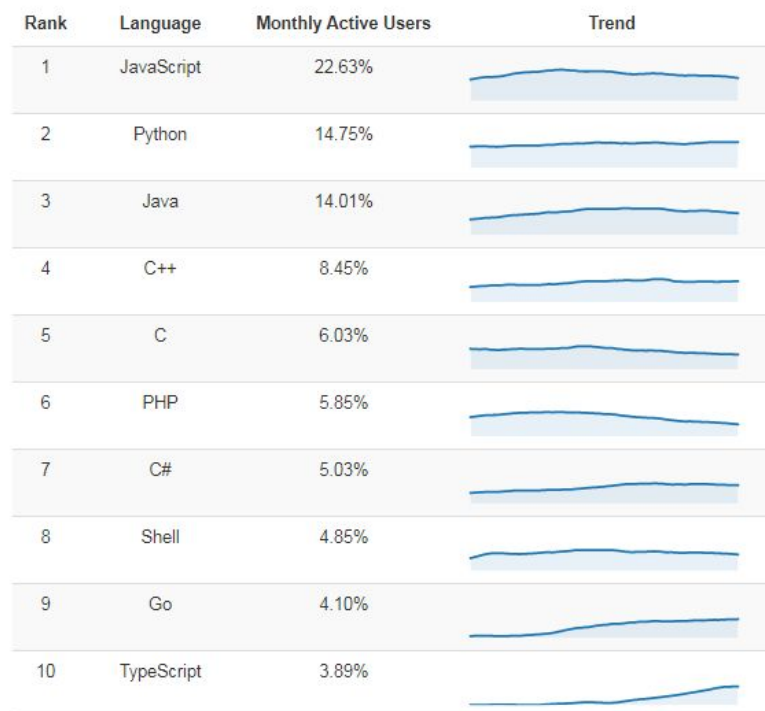
### 6.3 The Java Market Opportunity

Nowadays nine million developers use Java, and Java is currently running in 10 billion devices. The growth of Java usage is without a doubt a market. As shown in Figure 7, provided by [Statista](#), the second most used programming language is Java and accounts used by 21,42% of total users.

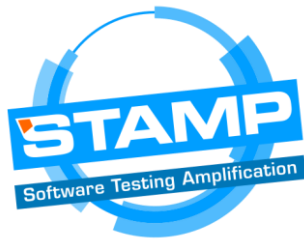


**Figure 3. Most popular programming languages (word)**

Moreover, according to the analyst firm [market research future](#), the microservice architecture market is expected to reach \$32.01 billion by 2023 and growing at a compound annual growth rate (CAGR) of aprox. 16.17%. Moreover, the firm IDC predicts that by the year 2022, 90% of new enterprise applications will be based on microservices. Already this year 2019, many IT departments will look on how to configure their virtual machine infrastructures and will be refactored/adapted to support microservices-based option and JavaScript appears to be the top one.



**Figure 4. Most popular programming language per month (active users)**



## 7 Mission, Technology and Offering

### 7.1 Mission and Value Proposal

Our overall mission is to improve the quality of software testing in a DevOps environment by leveraging STAMP.

The main benefit is to find weak spots in existing test suites and find places where the tests are not really testing what they think they're testing. The tool highlights these places by providing a report with detailed information about how to improve these tests.

It should be noted that the ROI for our tool and service is high. It should be enough for our tool to find a single badly written test that was causing a bug, to pay back the cost of the subscription for 10 years! That is our value proposal. Indeed, our service can prove that the underlying code is not actually tested properly, thus allowing developers to fix the test and prevent a potential bug in production. The cost of a bug in production is very high (usually one to several person/days: one day to fix the issue and release again in production plus analysis and reproduction times). Assuming the cost of the service is 60 euros per year, and that a developer costs 500 euros per day, that's close to 10 years of subscription.

The STAMP tools and services are targeting companies who have already invested in QA and already have tests in place. These companies already have in place a QA strategy, with automated tests covering a good portion of their code base. They're now interested by the next step of QA which is twofold:

- Ensure that their existing tests are properly executed and designed. It's indeed easy to have a good coverage but that gives very little indication as to how effective the tests are (it's possible to have 100% coverage and yet have the software not working or working badly).
- Build on top of their existing test assets to generate new complementary tests automatically.

The emergence of DevOps is a strong incentive for software development teams to implement test suites that can be executed automatically with each commit. In the continuous integration server, application of this technique increases trust in the commit. STAMP project developed four open source tools that integrate into build chains and instantiate the idea of amplification for Java, Junit, and Docker environment.

### 7.2 Technology

STAMP has developed a new technology called test amplification that helps developers improve their test and thereby the quality of their software. This approach starts with the code produced by the developers: unit tests case, APIs, and configuration files. Then, the amplification automatically analyzes and transforms this code to generate variants which will allow more intensive testing.

Amplification consists in applying automatic transformations to an existing set of manually written test assets, to significantly improve the efficiency of the tests and to detect anomalies as soon as possible. More precisely, STAMP focuses on:

- unit tests: detection of regressions on continuous integration servers before functional tests;
- configuration tests: detect compatibility bugs (operating system, language version, browser used, database used, memory size, etc.) before being put into production;
- production tests: reproduce the production bugs thanks to the call stacks.

Figure 2 positions these three amplification phases in a DevOps process. We present in the following, the tools that allow to automate these phases of amplification. These amplification techniques are currently implemented and **available for download** on the website <https://www.stamp-project.eu/view/main/> as **open source tools**. The **Descartes**, **Dspot**, and **Botsing** tools work on Java sources and Junit test suites, and **CAMP** is based on a Docker environment.

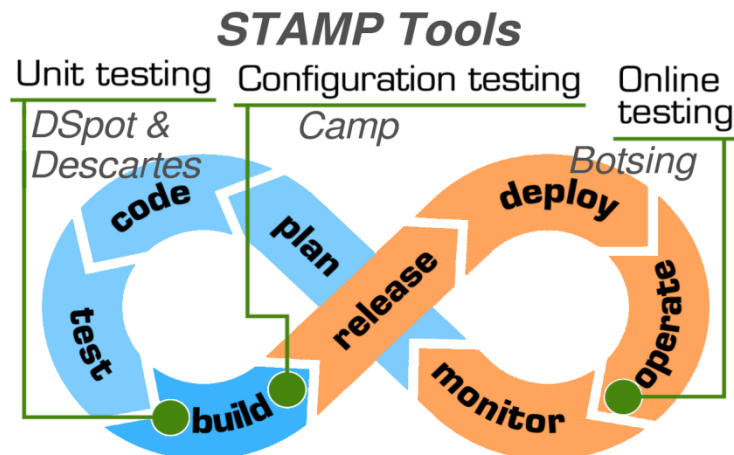


Figure 5. Positioning of the STAMP tools on the DevOps loop

- **Descartes** assesses the ability of a test suite to detect bugs, and helps developers to improve their test suites by reporting weak points in the covered code, that is, untested or incorrectly tested. More in details, **Descartes is a service that spots weak test cases and suggests improvements**: it is designed for software developers who wish to strengthen their test suite. Unlike code coverage tools which provide feedback about the breadth of the testing, Descartes focuses on providing feedback about the effectiveness of the testing.
- **DSpot automatically enhances existing JUnit test suites**. It automatically generates either new assertions in existing test cases or new test cases and will increase coverage and the overall “quality” of the tests (higher mutation score). For software developers and QA teams who wish to consolidate their test suite, DSpot is a service that automatically generates variants of existing test cases to cover more behaviours. Unlike Evosuite or AFL, DSpot starts from existing tests written by the developers. This way, the generated tests have a clear provenance and developers can understand the intention of these tests.
- **CAMP amplifies configuration testing**. From a basic Docker configuration and possible alternatives, CAMP runs systems test in all possible configurations. It helps software developers to ship their applications to a large number of platforms and environments. Using CAMP, **software teams can formalize the variations that their applications need to support such as different databases, application servers**, etc. While other configuration tests only address a small number of fixed configurations, **CAMP automatically explores the space of valid configurations, selects a relevant subset, automates the tests and produces aggregated reports**.
- **Botsing automatically generates a test that can reproduce a crash**, basing on the Java stack trace generated by the program during the crash. **Botsing is for software developers who wish to reproduce a runtime exception**. Botsing is a service that fully and automatically generates a test case that reproduces a crash starting from a stack trace, helping developers find out where the bug lies and thus increasing productivity.

### 7.3 Market Entry with Descartes

STAMP's key exploitable results is a tool box of four testing tools. They are however not all at the same state of market readiness. We will begin by concentrating on Descartes, the tool most ready for commercial exploitation. We identified for each of the tools the following possible scenarios for integration into the GitHub platform:

- DSpot can be triggered when a commit occurs. The app would check if the commit occurs on the test directory, and try to generate new tests from this new one, creating automatically a pull request.



- Descartes, like other code quality tools, will produce a report on test quality, on a regular basis when a commit occurs.
- Botsing can be triggered when an issue containing a stack trace is submitted, to generate a pull request that contains a test that reproduce the issue
- CAMP can be integrated in CI process, using the docker files present in the repository to run systems test nightly or immediately after a commit as usual in CI workflows.

As presented in the introduction, **all these STAMP tools could be proposed through one or several GitHub apps**, however in this deliverable **we consider a single, focused offer around the sole Descartes tool in order to reach the market quickly and efficiently**. The reason is that we consider the most matured tool to make an exercise and prepare a business plan. And at the same time the tool has a high level of maturity and can perfectly fit in the GitHub marketplace, whereas the other tools mentioned before would need more development and initial investment. We consider Descartes as the most adequate tool to be proposed as a service through a GitHub app. Indeed, this tool has been evaluated as the most mature during STAMP validation process and the one that offer the most added value to the developer (see STAMP Validation deliverables). This is also the tool that integrates the most straightforwardly into the GitHub software engineering workflows as demonstrated by the several code quality app already proposed in the GitHub marketplace.

## 8 Go-to-market: the GitHub Marketplace

### 8.1 GitHub Apps

GitHub apps<sup>1</sup> are tools provided in a Software-as-a-Service model that add functionalities to GitHub repository management and improve the users' software engineering process. Indeed, the basic functionalities provided by GitHub (source code management with access control, bug tracking, feature request, simple task management, and wiki) do not cover all the software engineering activities, and even less the continuous integration and continuous delivery processes.

Other functionalities can be provided by the enterprise or organization on-premise computing infrastructure and software, but also by GitHub apps. Services provided by existing GitHub apps are for instance: packaging and publishing, deployment, static code analysis, code formatting, test execution, external messaging, bug tracking actions, online IDE, etc. These services usually require minimum administrative or integration effort and can be used on-demand, regularly (e.g. each night), or automatically (e.g. for each commit).

### 8.2 Revenue Model

The GitHub revenue share consists in 75% of subscription fee reversed to the application developer in USD without reduction for Taxes except for any withholding taxes that are required under applicable law. The remaining 25% being allotted and retained by GitHub.

The pricing model of GitHub apps is diverse. The most typical pricing model is similar to GitHub pricing model, which is free usage for open-source repositories, and a paid usage billed monthly either per organization, per repository or per user (or a combination of such parameters).

Albeit, different pricing models exist, notably, some GitHub app providers propose only free usage or only paid usage, or even propose pricing according to the number of resources required to provide the service. E.g. Travis CI GMBH the company that is the main maintainer of the famous Travis CI open-source software, propose a GitHub App which pricing is

- free for open source project
- \$69 for up to one building job at a time
- \$199 for up to three building jobs at a time
- \$349 for up to six building jobs at a time

1

<https://github.com/marketplace>





In the code quality segment, pricing is usually a monthly fee per user, per repository or per organization. For instance, the TestQuality<sup>2</sup> app that propose different features including test analytics is free for open source repository or cost 50\$ per month for a team of five users. The Coveralls<sup>3</sup> app that feature test coverage reports cost 5\$ for one repository, \$25 for ten repositories or \$100 for an unlimited number of repositories.

### **8.3 Advantages of GitHub app model**

For software developers, the advantage of using GitHub apps can be seen as:

1. a better integration between CI/CD tools
2. and easier CI/CD software management especially for distributed teams
3. a free or pay-as-you-use pricing model to remove the fixed cost of CI infrastructure.

For the developer of GitHub apps, the advantage of proposing functionality through GitHub is the immediate access to the millions of users of the platform, and an integration effort of their software limited to standardized APIs (contrary to potential multiple on-premise environments). It also offers a showcase for enterprises willing to also sell licenses of enterprise versions of their software.

This business model is particularly well suited for open-source tools dedicated to improving software engineering processes and CI/CD workflows, such as STAMP tools

### **8.4 Publication and Verification process**

Anyone can publish an app on the GitHub marketplace. Albeit, there is a verification process that allows to obtain the “verified” app status. Verified apps appear specifically in the marketplace, and can propose paid plans, while unverified app can only propose free plan.

The verification process includes verification of user experience requirements, billing flow compliance and a security review.<sup>4</sup> Notably, security aspects concern authentication, data protection, logging capability, incident response plan, and vulnerability management workflow. The review process is based on the documentation (officially signed by the developing organisation) of the different security-related processes.

### **8.5 Implementing the GitHub App Model**

**The initial product of the start-up will be a code quality tool based on Descartes provided as a GitHub app, commercialized through GitHub marketplace.**

All GitHub app requires authorisations to access and potentially modify the repository. These authorisation can be *read-only* access to a repository or can include rights to add or modify elements in the repository.

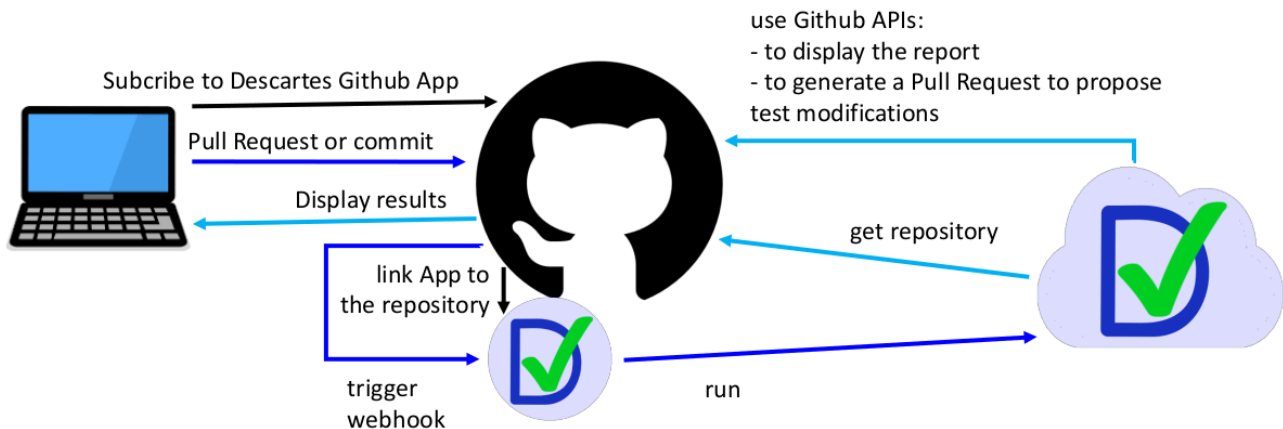
The workflow managing the Descartes App requires only read-only rights to the code source of the repository. Then the GitHub platform registers the app for web-hooking. For each commit on the repository, the GitHub platform will send request to the App. The App will then call the Descartes tool, and prepare a web report for the User. The Figure 6 illustrates this workflow of interactions between GitHub and Descartes.

---

2 <https://github.com/marketplace/testquality>

3 <https://github.com/marketplace/coveralls>

4 <https://developer.github.com/marketplace/getting-started/requirements-for-listing-an-app-on-github-marketplace/>



**Figure 6. Interaction between GitHub and Descartes**

The other STAMP tools can be integrated into the App. The GitHub marketplace ecosystem and the GitHub API allow several different interactions between GitHub and the app of the marketplace. For instance, the following workflow can be built:

- An app proposing Botsing, should be triggered when an issue is added on a repository. When the test reproducing the issue is created, the issue can be annotated by this test. This can be realized asynchronously using the GitHub API or synchronously as a response for the webhook trigger. Indeed, GitHub webhooks rely on POST request, including the triggering element, which response can include a modified version of that element.
- An app proposing DSpot should be triggered when a commit is produced, and should produce a merge request containing the tests generated.

The CAMP tool integration in GitHub is slightly more complex, and requires more R&D work since it requires to be integrated into a CI process. However, CI applications (such as Travis CI) exist in the marketplace, so this can be achieved using either another CI application or directly with integration into the repository owner CI platform.

## 8.6 Scalable Delivery model

GitHub marketplace allows users to find apps to use across the whole development process, from continuous integration to project management and code review. These apps are integrated in the GitHub process in a software as a service delivery model. First a user associates an app to one of its repositories or its organization, giving permissions to the app to interact with it. Then, the app extends the GitHub workflow by handling event occurring in the workflow, for instance an issue submission, a commit, or a merge request.

A GitHub app runs on a server managed by the app developers. The app registers to the GitHub platform through webhooks that trigger requests to the app server. The app replies to this request (e.g. by changing the status of the received issue) and can also interact on the repository through the GitHub API, according to the given permissions.

This delivery model offers several benefits to both users and app vendors. From the user point of view, it does not require any software installation or operation to obtain the service, and it integrates directly to its usual software engineering workflow. For the app vendor, it represents a unique integration effort to a platform, and coupled with a cloud-based stateless architecture, it can scale to an unlimited number of clients and requests. Indeed, the data being hosted by the GitHub platform, a simple load balancer (or cloud autoscaling group) and horizontal VM scaling management can handle such an amount of work, as well as function as a service solution (like AWS Lambda or Google Cloud Platform).

## 9 Organization and Resources

### 9.1 Stand-Alone Start-Up

Among the partners of the STAMP project, none is suitable to commercialize (externally) directly the results of the project. The large services companies' partners (Atos and Engineering) are proposing consulting services around the STAMP results, but do not have the agility to launch quickly such innovative and cutting-edge business model. However, they plan to incorporate the STAMP results in their own processes to serve all their QA teams. The independent software editors (XWiki and Activeeon) must focus their activities in the specific field they already address. Finally, the other academic or organization partners, can support the construction of a start-up (by providing incubation or provision of personnel), but do not intend to directly operate such business model.

In this context, the start-up company format is the ideal format to seek, effectively develop, and validate a scalable business model. This is especially true when addressing an emerging market with high uncertainty and potential failure, but also it allows to grow the business plan beyond the solo founder, have employees, and intend to scale to become large and influential [ (Schmitt, 2018)].

## 9.2 Business Model Canvas

We chose to use the Business Model Canvas approach to describe the overall operations foreseen. The business model Canvas for STAMP is shown below. The tool helps for strategic management and it is based on the "lean startup template" for developing new business models. We describe the STAMP value proposition, infrastructure, customers, and revenues options. The value proposition is clear to "evaluate the quality of the test in a given development/QA workflow". Some activities are offered as SaaS, providing user friendly information and reports on testing. The customer relation is through providing automated services and support for using the technology.

The customer segment is composed of GitHub developers, and test engineers/managers. In this case the GitHub revenue sharing option could work by keeping some core services available by a certain price and at the same time offering premium features with some specific, more sophisticated options of effectiveness of the testing/components that could have fees. The costs structure is clearly identified the personnel costs, marketing and the cloud maintenance/infrastructure. We have agreed on this common approach for using STAMP results and how these will be offered and commercialized based on this model. The figure 1 below summarizes the core ideas of our business plan.

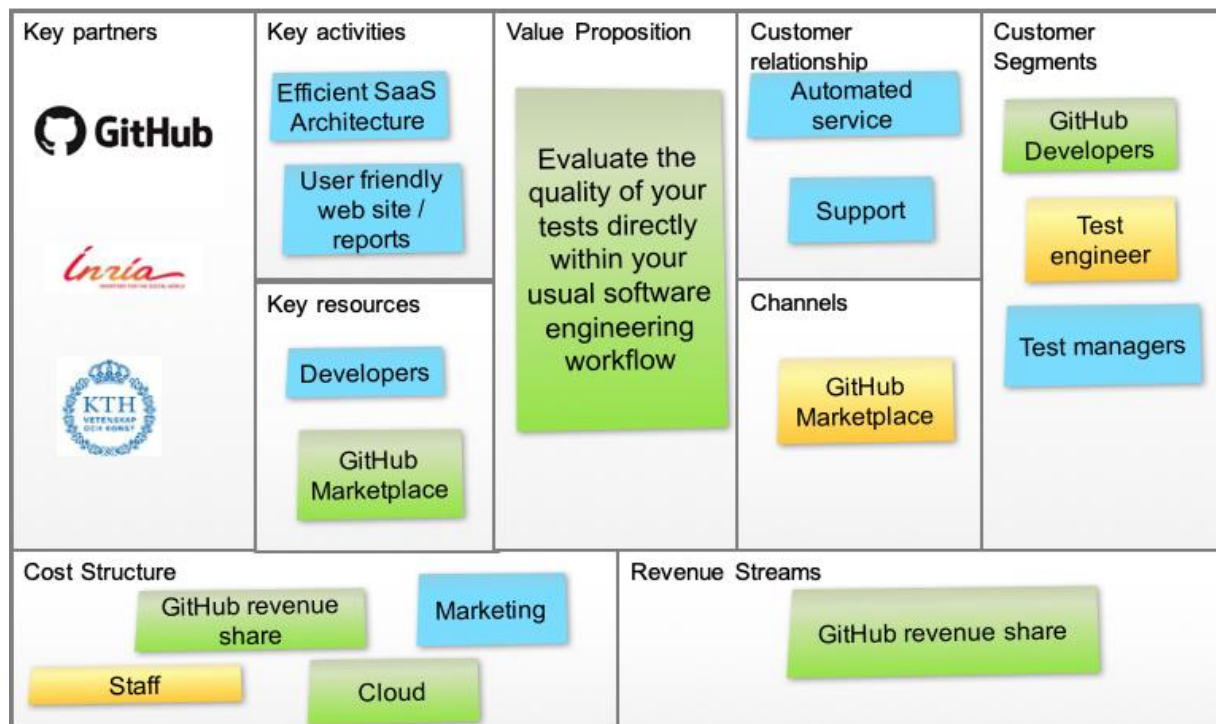


Figure 7. STAMP business canvas



As we are considering a business plan focused on Descartes we also provide the business model Canvas for the tool below. We describe the Descartes value proposition, infrastructure, customers, and revenues options. The value proposition is clear to “evaluate the quality of the test and provide a solution to it”. Activities are focused on providing mutate source code to verify the quality of those test cases. The customer relation is through providing automated services and support for using the technology. The customer relationship is based on GitHub. Customer segments are software providers, open source tools and any industry that has QA teams/developers, etc. The costs structure is clearly identified the R&D personnel costs, sales and marketing and operational costs based on the infrastructure and support. The revenues stream may come from GitHub services, consultancy and training. We will describe the approach and with more details for using Descartes and commercialize services based on this model. The figure 2 summarizes the core ideas of our business plan for Descartes.

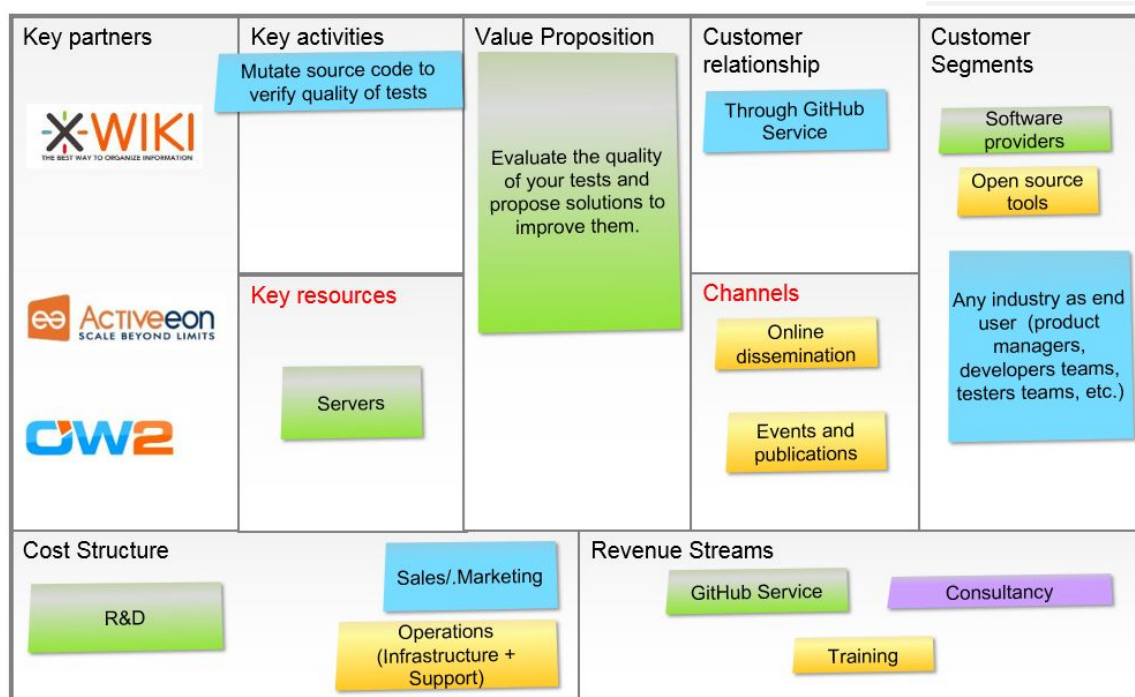


Figure 8. DESCARTES business canvas

## 10 Competition and Marketing

### 10.1 Competitive advantage

There are other GitHub applications claiming to offer functionalities similar to STAMP tools. They are in the “code quality”, “code review”, and “testing” GitHub marketplace categories.

All these tools provide quality indicators about the test but do not document how effective they are. For example, it is possible to have 100% coverage but without the tests testing anything (imagine some tests not having any assertions). The STAMP Descartes Application provides information about the effectiveness of your tests.

The following apps are the ones which propose features more similar to ours.



### 10.1.1 Codacy

- URLs
  - Website: <https://www.codacy.com/>
  - GitHub app marketplace: <https://github.com/marketplace/codacy>
- Features: Automatically identify issues through static code review analysis. Get notified on security issues, code coverage, code duplication, and code complexity. Triggered in GitHub.
- CI integration: in every commit and pull request, directly from your current workflow.
- Pricing: <https://www.codacy.com/pricing> & <https://github.com/marketplace/codacy>
- Competitive advantage we have: They don't support testing tests. Tests can show 100% coverage and still do nothing.
- Positioning vs them: Same positioning, for code quality

### 10.1.2 Coveralls

- URLs:
  - <https://coveralls.io/>
  - <https://github.com/marketplace/coveralls>
- Features: We help you deliver code confidently by showing which parts of your code aren't covered by your test suite.
- CI Integration: in every commit and pull request, directly from your current workflow.
- Pricing: <https://github.com/marketplace/coveralls>
- Competitive advantage we have: They don't support testing tests. Tests can show 100% coverage and still do nothing.
- Positioning vs them: Same positioning, for code quality

### 10.1.3 LGTM

- URLs:
  - <https://github.com/marketplace/lgtm>
- Features: LGTM automatically runs 1600+ standard analyses (static code analysis and build execution). Compare project quality between them.
- CI Integration: regularly on demand on the whole codebase + on PRs
- Pricing: <https://github.com/marketplace/lgtm>
- Competitive advantage we have: They don't support testing tests. Tests can show 100% coverage and still do nothing.
- Positioning vs them: Same positioning, for code quality

### 10.1.4 TestQuality

- URLs:
  - <http://www.testquality.com/>
  - <https://github.com/marketplace/testquality>
- Features: test plan management helping developers and testers create, run, coordinate and monitor software testing tasks.
- Integrates with GitHub issue tracker
- Pricing: <https://github.com/marketplace/testquality>
- Competitive advantage we have: They're not in the same domain as TestQuality focuses on test plans and test execution while we focus on gauging quality of tests.

## 10.2 Competition Pricing Analysis



Online applications are usually charged through monthly subscriptions and according to a range of packages, which are called “pricing plans”, representing levels of consumption. There are usually three pricing levels defined by the number of repositories covered by the contract.

The following Table 1 identifies pricing plans of comparable applications. We could say that the first level is for individual usage and covers a limited number of repositories, the second level is for an intermediate level with a number of repositories suitable for a team of developers, and the third level is the high end one adapted to corporate-wide usage with a high or unlimited number of repositories.

From this analysis we derive a range of pricing plans for STAMP. The initial pricing plan will be adapted depending on market reaction. We chose a pricing plan that is aggressive at the corporate level by having a low threshold for the “unlimited” pricing plan.

	Plan 0 (open source)	Plan 1	Plan 2	Plan 3
#repository	1	1-20	20-50	50 and more
Codacy	0			18
Coveralls	0	5	25	100
LGTM	0			
TestQuality	0		50	
BetterCode Hub	0	9		20
Code beat	0			20
Code Factor	0	19	39	199
Average	0	11	38	71,4
STAMP Pricing Plans	0	1	10	Unlimited

**Table 1. Monthly pricing plans of comparable applications**

### 10.3 Marketing plan

The marketing strategic plan leverages a combination of online (website, social media and marketplace) and face-to-face (industry events and conferences) resources and activities.

#### 10.3.1 Web design

We will pay particular attention to web design as a marketing tactic, it influences the amount of time and attention a user will spend on our page. Our website is the center of all our digital marketing efforts, so our page needs to reflect the state of the art in web design among our direct competitors. The web site will be clean, easy to read, and interesting. We consider hiring a professional web designer hiring someone to create a website that is modern, attention-grabbing, and mobile friendly.

#### 10.3.2 Marketplace

For faster setup, international outreach and ease of business, all transactions will be done online. In the first couple of years we will leverage the GitHub marketplace for apps. We will use it to showcase the different testing services plans and to manage user preferences and payments. As a front end to our services and our customer engagement, the marketplace will be an extension of the web site. We will evaluate how to make the



marketplace become a community place by connecting it to a forum and ways for users to share questions, concerns and best practices.

### **10.3.3 Social media and Content marketing**

Social media are an unavoidable communication channel in the developer community. We see content marketing as a high-end way to promote our services. The objective is to create and distribute valuable, relevant, and consistent content to attract and retain developers and their managers. We will work with a public relations professional to get our services into articles and news stories on topics related to DevOps, testing and software quality. The objective is to help establish STAMP as a trusted resource, and it will help broaden the scope of IT professionals to whom STAMP will be introduced.

### **10.3.4 Developer's events**

There are plenty of industry exhibitions and conferences targeting the developer and DevOps community. Beside international events such as Cloud Expo, DevOxx, FOSDEM, and OSCON, each country has its own local events. We will develop an event plan to reach out face to face to as many developers as possible. We will use a variety of event participation formats: both with running demos, presentations in the conference programs and contests or hackathons – developers love contests and giveaways.

### **10.3.5 Free trials**

A survey found that, although free trials conversion rates are usually low, a growing number of SaaS companies depends on free trials and freemium to acquire new business customers. We will develop free trials as a marketing strategy, by offering free trials for a limited period of tests. The free trial period must be of value, so we estimate that a whole month is both long enough to test the service and verify its value in real life environment and generate repeat use and skills investment and short enough to trigger a purchase. Free trials can be bound by a limited quantity, but we will need to run experiments as to what would constitute a relevant limited quantity.

### **10.3.6 Upsell**

While the first priority will be to grow the customer base with all sales efforts on the acquisition of new customers, we are well aware of the critical role of managing revenue from existing customers in the SaaS business model. SaaS company growth rates are strongly influenced by customer retention and upsell. We will pay attention to metrics on churn and upsell. Upselling will include offering an increase in compute resources available online, additional features (such as statistics, dashboards, for individual and group usage, etc.) and online support and expertise.

### **10.3.7 SEO and Affiliate program**

We will implement search engine optimization (SEO) and Affiliate program tactics. SEO is part of what helps make our business appear higher on a list of search engine results. With a strong SEO strategy, STAMP's website will become associated with the keywords used to find testing services. We will develop an affiliate program with other services or companies so as to enhance STAMP's profile and visibility online. This increases chances of being the online service a developer chooses to work with after searching online.

### **10.3.8 Analyst endorsement**

We see analysts' endorsements as a powerful way to speed up international growth. Industry analyst influence nearly 60% of all software sales. Industry analysts such as Gartner, Forrester and IDC influence high tech markets. However, we consider the big three to be out of reach and we would either go through the big three or through more affordable influencers such as 451, OVUM, or PAC.



Online sales: for the STAMP software components (only services), it may be possible to offer STAMP services via an online shop: GitHub website. We could develop more software that can be delivered remotely/over the cloud. Further discussions will be undertaken during the early stages of the commercial partnership we are establishing to clarify the sales strategy with other strong partners.

## 11 Operations and Growth Plan

### 11.1 Sales Forecast

According to the market segment analysis, we can estimate the number of **Java GitHub projects as 1M in 2020, with a growth of 40% per year**. We foresee an increase in interest in QA, testing and quality of testing of 20% per year (especially in new project).

The pricing of our application is defined according to the competitors pricing scheme (see above). However, since we will offer only one functionality at the beginning, we consider half of the price of comparable applications that offer different features. Once the other STAMP project results being integrated, the price could grow while maintaining a Descartes only commercial offer.

We consider that more of the half of the app subscriptions will be free open-source subscriptions, that will impact our operational cost but are useful for advertising.

	Revenue simulation		2020	2021	2022	2023	2024
a	# of Java developers on GitHub	units	1 000 000	1 400 000	1 820 000	2 366 000	2 839 200
b	% of a concerned by QA and testing	percent	15%	18%	22%	26%	30%
c	% of b concerned by quality of testing	percent	10%	12%	14%	17%	20%
d	Total Addressable Market on GitHub	units	15 000	30 240	56 609	105 973	170 352
e	single subscription sales price	euros	5	5	5	5	5
f	10-repository subscription price	euros	15	15	15	15	15
g	unlimited subscription sales price	euros	50	50	50	50	50
h	% market penetration assumption	percent	5%	7%	10%	14%	19%
i	# total number of subscriptions	units	750	2 100	5 500	14 500	32 700
j	% of free subscription	percent	80%	70%	65%	65%	60%
k	% of single repository subscription	percent	15%	20%	20%	20%	25%
l	% of 10-repository subscription	percent	4%	10%	10%	10%	10%
m	% of unlimited subscription sales	percent	1%	3%	5%	5%	5%
			100%	103%	100%	100%	100%
n	# of free subscription	units	600	1 480	3 610	9 450	19 600
o	# of single repository subscription	units	110	420	1 110	2 910	8 200
p	# of 10-repository subscription	units	30	210	550	1 450	3 300
q	# of unlimited subscription sales	units	10	60	280	730	1 600
r	Total sales assumption	euros	7 000	41 250	181 800	474 300	1 050 500

Table 2. Pricing and sales five-year forecast



## **11.2 Research, Development and IP**

Technically, the present Business Plan relies on two main elements: the Descartes tool and the SaaS architecture that responds to GitHub requests and call the Descartes tool.

### **11.2.1 Descartes tool**

As demonstrated by its successful integration into XWiki main's continuous integration pipeline and the positive feedback from all partners and external users, the Descartes tool has a high degree of maturity, established at TRL 8 (See Exploitation Plan deliverable). Moreover, the benefits of integrating Descartes in the software development processes have been confirmed by the validation effort conducted and results obtained in WP5.

However, further research will be conducted to improve the Descartes tools, such as:

- correcting automatically the weaknesses identified by Descartes by applying test amplification on targeted tests, or optimizing tests suites,
- customizing the mutation engine according to individual practices on automatic fixing of pseudo-tested methods.

Such research development will increase the value added by our GitHub app. Moreover, the research will be conducted by the concerned academic partners (INRIA and KTH), with no (or very limited) additional cost on the business plan start-up.

### **11.2.2 SaaS Architecture**

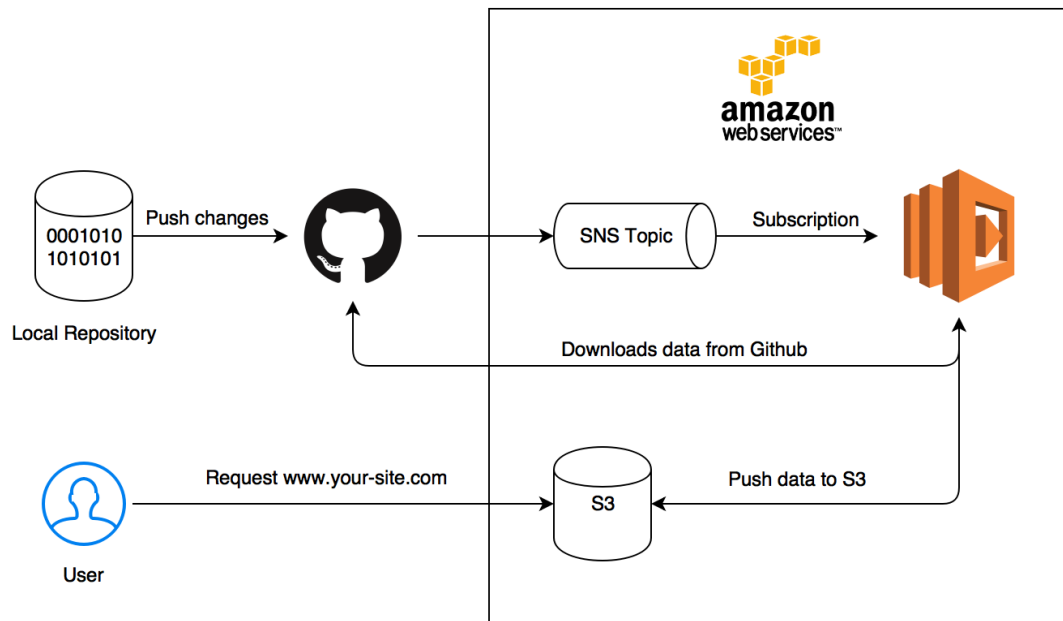
Development work has to be conducted to build the SaaS architecture that will operate the Descartes tool at scale. Thanks to the stateless nature of the application, this architecture can scale with a simple horizontal scalability model. Also, the mapping between GitHub request and Descartes execution requires a limited amount of work, as well as the integration into Github API. For instance, a basic GitHub app powered by AWS lambda FaaS platform, only requires a few hundred lines of code<sup>5</sup> <sup>6</sup>. Finally, the website that promotes our GitHub app as well as a comprehensive report generation when using the app must be developed.

---

5 <https://github.com/Glavin001/lambda-github-app>

6 <https://github.com/retgits/github-lambda>





**Figure 9. Example of serverless architecture for a DevOps process managed through GitHub**

### 11.2.3 IP Management

As explained in the exploitation plan (Deliverable 6.4), the STAMP consortium decided to apply an open-source license policy. More particularly the Descartes source code is released with LGPL V3.0 license. Such licensing allows the startup to exploit commercially an app that use Descartes as a tool (or even a library as stated by the license). Only AGPL (Aferro General Public License) would restrain such remote commercial exploitation by enforcing publication of the source code of the app. The brand under which the app will be marketed has to be registered.

### 11.3 Personnel

We understand staff must be kept to a minimum in the beginning so as to keep costs as low as possible. The technical requirements of the venture are the ones that cannot be overlooked. They are the priority for staffing. Sales and marketing will come when the operations has proven its potential. Administrative support is minimal with such small staff and can be outsourced.

Therefore, given the technical task required as described above, the start-up personnel will be exclusively technical. To produce this architecture, we will leverage part of the work conducted during the project we consider that a part-time DevOps engineer (for the architecture) and full-time web developer (for the website and web report).

Therefore, we will start with two persons:

- one engineer specialist of Descartes, that will be part-time in developing the SaaS infrastructure and part-time in support operations, advocate of the product,
- one engineer for the Web and user interface aspects, both for the promotion of the service and for the reporting,

### 11.4 Costs of Operations

Operation costs relative to this business plan can be divided into the following categories:

- Premises: offices for the start-up team. Free or reduced cost solution through universities, or industrial partnership will be targeted. Elsewhere, we count 5k€ for yearly rental fees in an

incubator for a staff of three. For more than three persons it may be more efficient to rent conventional office space.

- Staff: as presented above, we count two engineers. More staff will be added subsequently to integrate the other STAMP tools in as the success of the business model occurs. Equipment: only personal computers/laptops are considered, since the infrastructure will be cloud based.
- Communications facilities: included in incubator location fees.
- Legal and book-keeping cost: 10k€ as starting annual cost
- Marketing costs: as described in Section 9, marketing expenses will remain quite low up to 15k€ for the first year, including 5k€ for travel.
- Suppliers (cloud): cost for running Descartes on cloud instances. To compute this cost, we multiply the number of repositories targeted in the sales forecast (including free open source repository) by the number of commits per year and the cost of cloud instance. The average number of commits per month is computed by retrieving the statistic of the top trendy java repositories<sup>7</sup>, which we double to obtain a high limit. The cloud instances considered are AWS t2.small spot instance<sup>8</sup>; spot instances being well adapted to such non-synchronous work.
- Finally, total cost is estimated as twice the VM cost due to storage (HTML reports) and cloud scaling services.
- Commits per month: Below we show the XWiki's monthly figures as estimates, in order to get an estimation of commits per month. The figures from GitHub shows an activity of around 30-50 commits per week and 100-200 commits per month:

Repository	# commit per week	# commit per month
<a href="#">Xwiki platform</a>	30-50	100-200

**Table 3. Xwiki platform commits**

			Y1	Y2	Y3	Y4	Y5
a	Repository	units	1 170	4 870	15 570	40 760	90 900
b	Avg commit / repo / month	units	40	40	40	40	40
c	Commit per year (x1000)	units	562	2338	7474	19565	43632
d	Computation time per commit	min	3	3	3	3	3
e	Cost per hour	euros	0,0079	0,0083	0,0087	0,0091	0,0096
f	VM computing cost	euros	222	970	3255	8946	20949
g	Total cloud expenses	euros	444	1939	6509	17892	41898

**Table 4. Cloud production cost five-year estimates**

	Personnel cost		2020	2021	2022	2023	2024
b	Development/support engineer	euros	75 000	76 500	117 000	159 200	243 600
c	Web engineer	euros	75 000	76 500	78 000	119 400	121 800
d	Management and marketing	euros	0	0	39 000	79 600	162 400
	Total Personnel Cost	euros	150 000	153 000	234 000	358 200	527 800

**Table 5. Personnel cost five-year estimates**

	Operating expenses		2020	2021	2022	2023	2024
--	--------------------	--	------	------	------	------	------

<sup>7</sup> <https://github.com/trending/java> on 09/10/2019

<sup>8</sup> <https://aws.amazon.com/ec2/spot/>



a	Office rent	euros	6 000	6 300	9 900	13 900	21 900
e	Marketing expenses	euros	10 000	12 000	14 400	17 300	20 800
f	Travel expenses	euros	5 000	6 000	8 400	12 600	18 900
g	Legal and Book-keeping	euros	10 000	11 000	12 100	13 300	14 600
h	Cloud computing costs	euros	440	1 900	6 500	17 900	41 900
l	Other operating expenses	euros	29 600	14 500	36 500	28 300	41 000
j	<b>Total operating expenses</b>	<b>euros</b>	<b>31 440</b>	<b>37 200</b>	<b>51 300</b>	<b>75 000</b>	<b>118 100</b>

**Table 6. Operating expenses five-year estimates**

## 12 Financial Projection and Funding Requirements

Given our revenue and cost assumptions, establishing and launching a business venture exploiting the Descartes service would require a substantial startup investment before reaching profitability at the fifth year of operations. The table below gives a summary of our financial expectations.

	Operating result simulation		2020	2021	2022	2023	2024
r	Total sales assumption	euros	7 000	41 250	181 800	474 300	1 050 500
s	GitHub revenue share	euros	1 750	10 313	45 450	118 575	262 625
t	Total available revenue	euros	5 250	30 938	136 350	355 725	787 875
j	Total operating costs	euros	181 440	190 200	285 300	433 200	645 900
v	<b>Operating Result</b>	<b>euros</b>	<b>-176 190</b>	<b>-159 263</b>	<b>-148 950</b>	<b>-77 475</b>	<b>141 975</b>

**Table 7. Operating results five-year estimates**

Due to the costs needed to operate this business plan and the maturity of the Descartes tool, the investment required to set up is limited; assuming GitHub revenue share remains stable at 25%, our projections reveal a need for funding of €884800 to take the start-up from launch to profitability.

Funding should be optimally split in two rounds. If only one round the risk would be great for an investor and if more than two rounds the time spent on raising money would be inefficiently diverted from the necessary efforts to grow the business. The first round of funding should cover the first four years while the second round provides cash to get into the fifth year. The second-round amounts to 50% of the operational cost of the fifth year. The funding scheme looks as follows:

Funding Scheme	euros
<b>Cumulated five-year result</b>	<b>-419 900</b>
<b>First round. Covering four years</b>	<b>562 000</b>
<b>Second round. Getting into fifth year</b>	<b>323 000</b>
<b>Total required funding</b>	<b>885 000</b>

**Table 8. Start-up funding scheme**

This level of investment is not trivial and would probably not be personally provided by the founder(s) of the start-up, either the CEO or all the starting team. It will be necessary to explore alternative sources of financing from European research initiatives such as EIT Digital start-up program which is well adapted to such start-up funding, or Eurostar or FTI program for scaling up and integrating the other STAMP tools.

The above financial projections consider the commercial exploitation of Descartes only. As the other tools are evaluated less mature by the validation activities, they require a slightly higher amount of R&D to be provided



in a SaaS mode through a platform such as GitHub. We will leverage the net operating result from the start-up from year 4, and private investment and public subvention as soon as they are obtained.

As soon as the cash situation allows it, an extension to this business plan will evaluate focus on adding other STAMP tools to the service.

## 13 Conclusion

The project consortium is confident that a new start-up exploiting the STAMP tools can be profitably launched. Having carefully considered its value proposal, its market and its growth potential and leveraging the GitHub marketplace the start-up could reach profitability within five years by offering a Descartes service only.

For the sake of clarity, the current business plan is exclusively focused on offering Descartes as SaaS. More revenue could be derived from offering traditional services such as consulting and technical expertise. However traditional service revenue and cost grow linearly depending on the number of staff and their billing hours and would not have changed substantially the need for funding.

The start-up has the potential to create highly qualified jobs with an innovative testing service that has the potential to be highly regarded in the software engineering community. This could make the start-up an acquisition target which has to provide lucrative returns to potential investors.

## Annex I

To update and have the overall STAMP picture we have updated the SWOT sent out in year one, we share here a SWOT analysis is a strategic planning method used to evaluate the Strengths, Weaknesses, Opportunities, and Threats involved in both **in the case of our proposed business venture: Descartes and for STAMP as whole/project**. It involves specifying the objective of the business venture or project and identifying the internal and external factors that are favourable and unfavourable to achieve that objective.

### Descartes S.W.O.T. Analysis

<p><b>Strength</b></p> <ul style="list-style-type: none"> <li>• Descartes is open source</li> <li>• Strengthen test suites</li> <li>• Can work in SaaS</li> <li>• Detects weakness in code/test cases and suggests improvements</li> <li>• Provides feedback on effectiveness of testing</li> </ul>	<p><b>Weakness</b></p> <ul style="list-style-type: none"> <li>• Requires manual understanding of Descartes reports and a developer needs to perform the suggested actions on the existing tests</li> <li>• Quite fast but still takes some time to execute</li> </ul>
<p><b>Opportunity</b></p> <ul style="list-style-type: none"> <li>• Automated testing is a need in organisations</li> <li>• Deployment in SaaS mode</li> <li>• Most existing tools focus code coverage but none cover quality of tests themselves. It's a new domain and next step in QA</li> </ul>	<p><b>Threat</b></p> <ul style="list-style-type: none"> <li>• Requires users and companies who are already good at QA with good code coverage. Otherwise they won't be interested in testing their tests.</li> </ul>

Figure 10. Descartes SWOT analysis

### STAMP S.W.O.T. Analysis

<p><b>Strength</b></p> <ul style="list-style-type: none"> <li>• STAMP tools are open source</li> <li>• STAMP focus on DevOps and CI</li> <li>• Market need for more automation</li> <li>• STAMP processes and tools are adapted to agile delivery models</li> <li>• Integration into CI tools (Jenkins)</li> <li>• STAMP tools are built upon leading technologies</li> <li>• No serious competitors in mutation and online testing</li> </ul>	<p><b>Weakness</b></p> <ul style="list-style-type: none"> <li>• Not addressing UI testing</li> <li>• Not addressing web testing</li> <li>• Not addressing testing data issues</li> </ul>
--	--



<b>Opportunity</b>	<b>Threat</b>
<ul style="list-style-type: none"><li>• Automated testing is a need in organisations</li><li>• Maturity of DevOps adoption in organizations</li><li>• Emergence of test excellence center</li><li>• Automation level is still low (unsatisfactory testing tools or insufficient automation)</li><li>• Test design automation viewed as a emerging technique</li><li>• Growing market share of IDE interfaced with CI</li><li>• Deployment in SaaS mode</li></ul>	<ul style="list-style-type: none"><li>• Development skills requirement to run STAMP tools</li><li>• Some organizations do not have automated tests cases at all, so they cannot use most STAMP tools</li><li>• Possible competitors from Google (mutation testing) or Facebook (online testing)</li></ul>

Figure 11.STAMP SWOT analysis