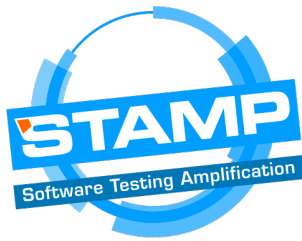


Title:	WP6 – D6.3 – Market Analysis
Date:	June 26, 2018
Writer:	Pascal Urso (Activeeon), Malena Donato (ATOS), Cédric Thomas (OW2)
Reviewers:	Benoit Baudry (KTH), Valentina Di Giacomo (ENG), Caroline Landry (INRIA)

Table Of Content

1.EXECUTIVE SUMMARY.....	2
2.REVISION HISTORY.....	2
3.OBJECTIVES.....	3
4.INTRODUCTION.....	3
5.REFERENCES.....	3
6.ACRONYMS.....	4
7.APPROACH.....	5
8.THE STAMP OFFERINGS.....	7
9.MARKET TRENDS: DEMAND SIDE ANALYSIS.....	9
10.INDUSTRY TRENDS: SUPPLY SIDE ANALYSIS.....	13
11.MARKET SIZING.....	15
12.BUSINESS OPPORTUNITIES.....	17
13.CONCLUSION.....	25
14.ANNEX: PANORAMA OF TOOLS IN THE SOFTWARE TESTING AND QA ENVIRONMENT.....	26



1. Executive Summary

This deliverable provides the first version of the STAMP Market analysis in the context of the WP6 (Dissemination & Exploitation). This report presents an overview of the STAMP project market environment. We already knew the technologies developed by the STAMP project were advancing the state of the art in software testing. What the market analysis shows is that these technologies are aligned with key market drivers such as digital transformation, the adoption of agile and DevOps methods which call for more automated testing technologies. The analysis also explores the competitive environment and identifies key players that could become STAMP partners or OEM adopters. The analysis of the industry suggest that STAMP should be promoted toward mid-tier technology-oriented vendors of testing platform rather than leading conventional global testing service vendors. A later version of the market analysis could drill down on some specifics of the market and pave the way for the development of successful exploitation plans.

2. Revision History

Date	Version	Author	Comments
15-feb-2018	0.01	Pascal Urso (Activeeon)	First draft of the deliverable
	0.02	Malena Donato (Atos)	First release.
	0.03	Caroline Landry (inria)	Correcting the date and the revision table
30-avril-2018		Pascal Urso (Activeeon)	Stakeholders
2-may-2018		Pascal Urso (Activeeon)	Test excellence centers
18-may-2018		Pascal Urso (Activeeon)	Test challenges
21-may-2018		Malena Donato (Atos)	Market Context
21-may-2018		Pascal Urso (Activeeon)	Test environment challenges
22-may-2018		Pascal Urso (Activeeon)	Test automation challenges
28-may-2018		Pascal Urso (Activeeon)	Environment
31-may-2018		Pascal Urso (Activeeon)	Competitor list
4-jun-2018		Pascal Urso (Activeeon)	Methodology
5-jun-2018		Pascal Urso (Activeeon)	Agile and DevOps adoption
7-jun-2018		Malena Donato (Atos)	Testing and automation market, value proposition
8-jun-2018		Pascal Urso (Activeeon)	SWOT
8-jun-2018		Malena Donato (Atos)	Testing market, executive summary, references, survey findings
11-jun-2018		Malena Donato (Atos)	Inputs to market and introduction, segments
11-jun-2018		Pascal Urso (Activeeon)	Prepare final version for revision
15-jun-2018		Valentina Giacomo (ENG) Benoit Baudry (KTH)	Reviewers
18-jun-2018		Pascal Urso (Activeeon) Malena Donato (Atos)	Overall revision
22-jun-2018	0.10	Cedric Thomas	Market trends, market sizing, industry analysis and overall rewriting
25-Jun-2018	0.11	Caroline Landry	Typos, formatting styles, addressing table of

			content formatting issue Updating sections IDE and Continuous integration
26-Jun-2018	1.0	Caroline Landry, Cédric Thomas	Integrating review comments Correcting links, formatting annex

3. Objectives

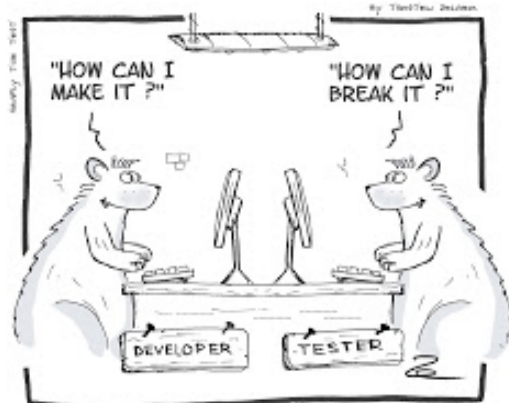
The objective of this deliverable is to provide a global overview of the current software testing and quality assurance market, and to understand the trends and challenges that drive the evolution of this market. Findings in this deliverable should provide relevant market insights to support the drafting of successful business exploitation plans of the technologies developed by STAMP.

4. Introduction

The present deliverable is the first deliverable in WP6 that deals with market and trends. It covers software testing and quality assurance markets context and the trends in the IT and software industry related to STAMP project.

This report starts by defining the status of the STAMP offering, then it presents a demand-side analysis made of an overview of key trends, drivers and challenges in the software testing market, followed by a supply-side analysis including a segmentation of the market and an identification of key industry players. These analysis are completed by quantitative market indicators gathered from diverse market analysts firms. The last section introduces an understanding of specific business conditions facing the STAMP projects.

This deliverable is not about technology, it is about understanding STAMP's business environment, how user needs evolve, who are the vendors, what are the trends, who are the players in the market and what is at stake for them, etc.



They were not so much different, but they had different goals

"When a tester didn't find any cool bugs, then he basically has two options to think about.

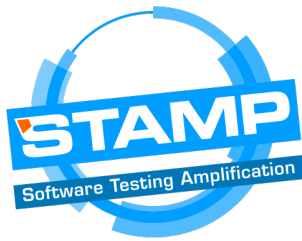
- a) The software is good enough to go live
- b) The software wasn't tested well enough

Typically, the developer thinks it is "a". And, of course, the tester always fears it could be option "b".¹

5. References

[0] D6.3 – Market Analysis: [d63_market_analysis.pdf](#)
A link to the most recent version of this document.

1 <http://simply-the-test.blogspot.com/2010/04/different-goals.html>



- [1] Mohanty, Hrushiksha, et al. Trends in Software Testing. Springer Singapore, 2017.
- [2] Moorthy, Anand, et al. World Quality Report 2017-18, Ninth Edition. CapGemini, Micro Focus, Sogeti, 2017.
- [3] Testing Trends in 2016: A Survey of Software Professionals. Sauce Labs, 2016.
- [4] Java tools and Technologies Landscape 2016. Rebellabs, 2016.
- [5] Urso, Pascal et al. D5.5 Use Cases Validation Report V1. STAMP H2020 project, 2018.
- [6] Petrovic, Goran, and Marko Ivankovic. "State of Mutation Testing at Google." Proceedings of the International Conference on Software Engineering—Software Engineering in Practice (ICSE SEIP). 2018.
- [7] Xuan, Jifeng, Xiaoyuan Xie, and Martin Monperrus. "Crash reproduction via test case mutation: Let existing test cases help." Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering. ACM, 2015.
- [8] Chen, Ning, and Sunghun Kim. "Star: Stack trace based automatic crash reproduction via symbolic execution." IEEE transactions on software engineering 41.2 (2015): 198-220.
- [9] Nayrolles, Mathieu, et al. "JCHARMING: A bug reproduction approach using crash traces and directed model checking." Software Analysis, Evolution and Reengineering (SANER), 2015 IEEE 22nd International Conference on. IEEE, 2015.
- [10] Soltani, Mozhan, Annibale Panichella, and Arie van Deursen. "A guided genetic algorithm for automated crash reproduction." Proceedings of the 39th International Conference on Software Engineering. IEEE Press, 2017.
- [11] Mao, Ke, Mark Harman, and Yue Jia. "Sapienz: Multi-objective automated testing for Android applications." Proceedings of the 25th International Symposium on Software Testing and Analysis. ACM, 2016.
- [12] Global Software Testing Market 2017-2021, Technavio, Feb 2017.
- [13] Global Bug Tracking Software Market 2017-2021, Technavio, May 2017.
- [14] Magic Quadrant for Software Test Automation 2017, Gartner.
- [15] World Quality Report 2017-2018, CapGemini (1660 respondents).
- [16] Testing Trends in 2018: A Survey of Software Professionals, Sauce Labs (1100 respondents)
- [17] Global Software Testing Market 2017-2021, Technavio.
- [18] Cruz, Jesus G. et al. D5.2 – Validation Roadmap and framework V1. STAMP H2020 project.
- [19] Software Testing Services: The Impact of Digital and DevOps, by Dominique Raviart, Oct 2017. Nelson Hall.
- [20] Christina Cardoza, SDTimes, <https://sdtimes.com/agile/continuous-testing-fast-paced-agile-devops-world/>

6. Acronyms

EC	European Commission
AI	Artificial Intelligence
ALM	Application Lifecycle Management
CAGR	Compound Annual Growth Rate
CEO	Chief Executive officer

CI	Continuous Integration
DevOps	Development & Operation
DSDM	Dynamic Systems Development Method
IDE	Integrated Development Environment
IoT	Internet of Things
IT	Information Technology
QA	Quality Assurance
Rol	Return of Investment
RDBMS	<i>Relational DataBase Management System</i>
SaaS	Software as a Service
SAFe	Scaled Agile Framework
SWOT	Strength, Weaknesses, Opportunities and Threats.
TCoE	Test Center of Excellence
TEC	Test Excellence Center
UI	User Interface
UX	User experience
WP	Work package

7. Approach

Our approach is supported by the following methodology. First we take a top-down approach on the testing market by analysing existing sources such as market reports published by independent market research firms and market and tools overviews published by dedicated web publishing sources. The top down approach provides us an understanding of the testing market trends and segmentation. Second we take a STAMP-specific bottom-up approach by directly approaching IT professionals with our own questionnaire. The bottom-up approach helps us confirm by ourselves results from the top-down approach and develop our own understanding of appropriate business strategy for the exploitation of STAMP results. In this paragraph we introduce our external sources and our questionnaire.

External sources

Although we have had full access to some external sources, most of them have been used indirectly since we did not purchase any market report: we refer to them via press announcement, quotes in presentations, etc.; they include industry analysts reports, overview papers and publications. The main external sources studied are

1. "[Word Quality Report 2017-18](#)" [2]. This report is the ninth issue of a yearly study built by two major IT consulting firms: CapGemini and Micro Focus. They conducted interviews based on a questionnaire of 43 questions, and obtained responses from 1660 IT industry executives from 32 countries. The reports present their analysis in term of trends and challenges as well as a sector analysis. Please note, that the respondents of this survey are only executives and only from organization with more than a thousand employees, thus reflecting only a specific point-of-view.
2. "[Testing Trends in 2016: A Survey of Software Professionals](#)" [3]. This report is conducted by Dimensional Research and sponsored by Sauce Labs, an enterprise that provide a cloud-based platform for automated testing. This survey focus more on technical and methodological aspects of the software testing, and had interviewed 520 software professionals.
3. "[Software Testing Services: The Impact of Digital and DevOps](#)" [19], by Dominique Raviart, published on Oct 13, 2017. Nelson Hall. This NelsonHall Vendor Evaluation & Assessment Tool (NEAT) analyzes the performance of vendors offering digital software testing services. The NEAT tool allows

strategic sourcing managers to assess the capability of vendors across a range of criteria and business situations and identify the best performing vendors overall, and with a specific focus on digital delivery and DevOps & Agile.

4. "[Magic Quadrant for Software Test Automation](#)".[14]. Published: 20 November 2017 ID: G00320337. Analysts: Joachim Herschmann, Thomas E. Murphy.

STAMP Market Questionnaire

To obtain insights closer to STAMP objectives that are not presented in the above studies, we designed our own questionnaire of 23 questions. We obtained 34 responses from IT professionals. The questionnaire was available online² and we advertised internally and externally to our partners in order to obtain responses.

Despite a lower number of responses compared to some of the above studies, our results were consistent with them (see for instance "Test Automation" in Section 9). Respondents profiles vary from researchers to management and developers, and include an important part of specialists in QA and tests (45.5%). A majority of respondents (63.6%) belong to an organisation that employs less than 50 developers.

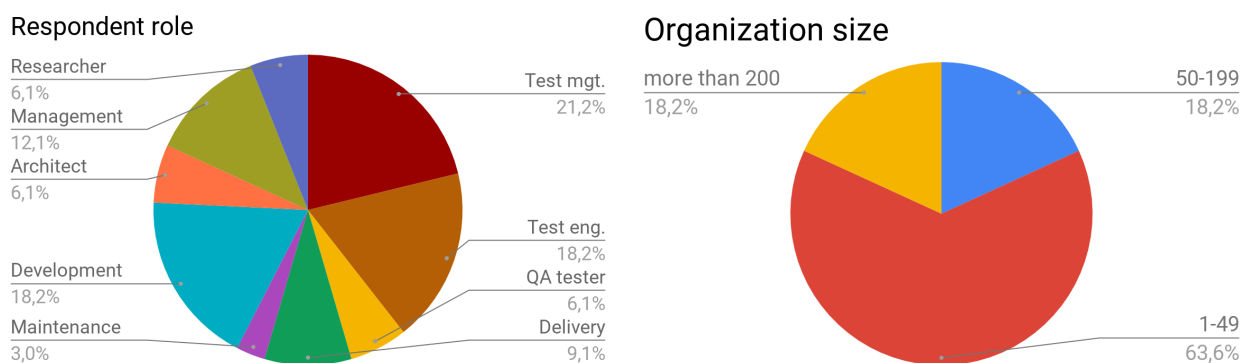


Figure 1: Respondents of STAMP survey

The main development language of the STAMP survey respondent is Java for 62.2%, contrary to the main developer population among which 14% to 16% are considered as using primarily Java.^{3,4} Since most of the STAMP outcomes are Java based, this set of respondents is particularly interesting for our market analysis.

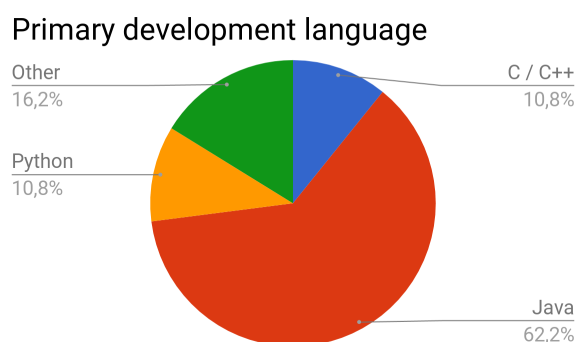


Figure 2: Primary development language of STAMP survey respondents

² <https://goo.gl/forms/NZ7A5iP7ufVRk2eH2>

³ <https://www.benfrederickson.com/ranking-programming-languages-by-github-users/>

⁴ <https://www.tiobe.com/tiobe-index/>

8. The STAMP Offerings

In this section we identify where STAMP stands from a market perspective. First we formalise its value proposal, what needs the project addresses and the benefits it brings to its users, then we identify the tangible offering which could support a business exploitation.

The Value Proposal

STAMP is about improving software testing, particularly in cloud application development, through automation. STAMP stands for Software Testing AMPLification, it combines ground breaking research in mutation testing and state-of-the-art systems and techniques in automatic test generation.

Stemming from the hypothesis that software test cases are often already available, but manually developed and with a quality that is, most of the time, insufficient, STAMP offers testers automatic generation of new variants in test cases so as to cover more inputs, and a wider range of behaviors and observation points.

The value proposal can be summarized in one sentence as follows: STAMP automatically amplifies existing software testing assets to detect more regression bugs prior to production and drives down the costs of software testing.

STAMP addresses the need of the increasingly popular DevOps way of developing and deploying software by focusing on three stages in the DevOps infinite loop as illustrated below: Unit testing, Configuration testing and Online testing.

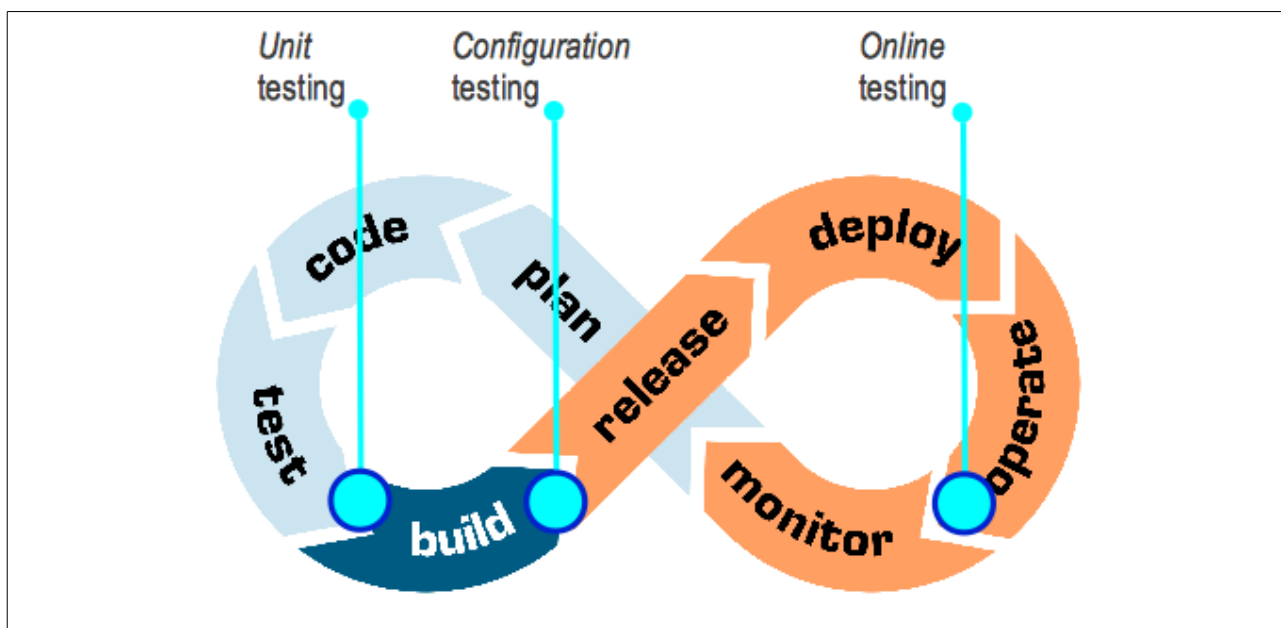
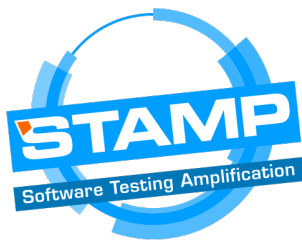


Figure 3: STAMP and the Infinite DevOps Loop

A key advantage of STAMP is that it doesn't start from scratch because its tools leverage existing test assets such as unit test cases, configuration files, and production logs. STAMP generate new test cases that detect a greater number of regression bugs, cover more more lines of code and even provide recommendations for more efficient test suites. STAMP also amplifies configuration testing. It uses existing configuration files to generate amplifications in Resource limitation, Horizontal scaling, Dependency twisting and Instrumentation. It enhances the detection of environment-specific bugs and performance regressions by increasing the number of configuration variations which are automatically generated, deployed and tested. And once a configuration of services is deployed and in production, STAMP automatically optimizes and analyzes production logs. Online test amplification automatically extracts information from production logs and generates new tests, replicating failures, crashes, anomalies and other incidents. This results in new and better test cases that can replay actual production conditions.



In summary, STAMP's key benefit is to boost test automation in the DevOps software development and execution chain by:

- detecting more regression bugs,
- detecting more configuration and scalability bugs
- and improving existing test suite with direct feedback from production.

In marketing terms one can stress that STAMP literally takes automation in DevOps to the next level.

The tangible product

As a research project, and at the time of writing, the STAMP offering is essentially comprised of technologies destined to be packaged as micro services and integrated in software testing chains. STAMP aims at delivering well documented working testing software services at TRL6.

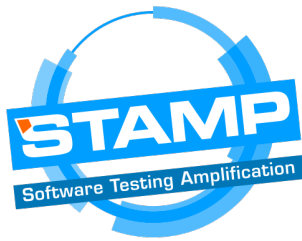
At this stage, and as its name suggests, STAMP is, technically, not a stand-alone product but rather a *tool set* used in the tuning and sharpening of existing testing environments. STAMP is a suite of tools packaged in a micro service architecture, convenient for DevOps environments. However it is planned that STAMP tools will also be integrated in Jenkins and will be available as plug-ins for the Eclipse IDE and Maven. STAMP tool set includes four tools, DSPOT, Descartes, CAMP and Evocrash.

- **DSPOT** : DSpot is a tool that generates missing assertions in JUnit tests. DSpot takes as input a Java project with an existing test suite and produces new test as output.
- **Descartes** : Descartes evaluates the capability of your test suite to detect bugs using extreme mutation testing. Descartes takes as input a Java project with an existing test suite and compute a mutation coverage score and report as output.
- **CAMP** : CAMP (Configuration AMPlification) takes as input a sample testing configuration and generates automatically a number of diverse configurations where to apply system tests. The current version of CAMP uses Docker files or docker-compose file as input and output configurations description.
- **Evocrash** : EvoCrash is a search-based approach to automated crash reproduction. EvoCrash receives a Java crash stack trace, and searches for a unit test case that can reproduce the crash.

The exhibit below summarizes the STAMP tool set and its software dependencies for installation and execution.

Tool	Installation	Execution stack	Data
DSPOT	maven	java <i>optionally : PIT, JaCoCo, OpenClover, maven</i>	Java code JUnit tests
Descartes	maven	PIT maven, gradle or ant java	Java code JUnit tests
CAMP	-	docker python	docker configuration files
Evocrash	-	java	Java stack trace

Table 1: Components of the STAMP toolset



9. Market trends: Demand side analysis

This section provides an analysis of STAMP's market context from a demand-side perspective. It highlights market drivers and challenges that may have an impact on the extent to which STAMP is aligned with user needs. Drivers are changes in the market context that are positive for STAMP, whereas challenges are issues and potential difficulties that should be addressed or worked around, when considering STAMP's market development. This section concentrates on drivers and challenges that are relevant to STAMP, for example it does not include market trends such as crowd testing and industry specific testing that are very much front-end or UX-oriented whereas STAMP is more back-end and infrastructure oriented.

Driver: Digital transformation

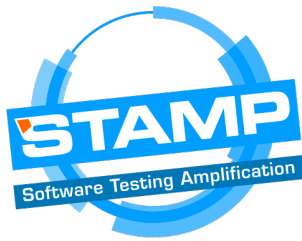
The much talked about digital transformation of businesses points to a world where companies not only have to become software developers to compete, but deliver higher quality software to market faster. Digital transformation means business processes are becoming software-based and relations with partners and customers software driven. Digital transformation provides a whole new context to software development and generates radical changes in the testing market. For starters, the global software testing market experiences changes that shift its focus from being product or application centric to customer centric. In order to support business transformation, testing services are increasingly shifting to quality assurance. Quality assurance focuses efforts on client's satisfaction rather than strictly technical variables. Then, in a digital transformation context, companies can't take a year anymore to get a product out the door; they have to be much more agile. To compete in a world where high-quality application experiences are the norm and in order to respond to permanent competitive moves and ever changing customer needs, online applications are being updated constantly. Testing must be as fast and as reactive as the agile development processes in order to avoid delays in developing and shipping software to operations. Software developers no longer have to write code, test it, and then wait to find issues; they must be able to constantly take pieces of the solution and constantly test, instead of waiting for a new release cycle. In this context, traditional or legacy testing tools and processes don't comply with speed; testing needs to be efficient and effective, and that requires automation and a continuous testing approach. In summary, the digitization of businesses increases the dependence on software for running business operations. The failure of an application or software product at a crucial point in the process or transaction can lead to significant financial losses amounting to millions as well as the loss of customer goodwill. This leads more companies to focus on software testing services within the development process.

Driver: Agile and DevOps

There is a growing adoption of agile and DevOps principles by organizations seeking to achieve the speed, quality and scale needed to succeed. On one hand, agile software development becomes the norm and on the other the DevOps approach has a clear impact on testing processes and methodologies. There seems to be a continuum from Agile to Testing via DevOps. While it accelerates the speed of development and brings value to customers faster, agile development needs to take operations into consideration. And suddenly it's not just agile development but DevOps. But as soon as DevOps is taken into consideration, there is an unavoidable segment in the middle that needs to be taken care of, which is testing. Changes toward agile software development processes have a big impact on testing and QA. For a start, agile implies a shift in QA and Testing to earlier stages in the lifecycle and involvement of testing teams in the inception and sprint planning phases. In this perspective, testing becomes more iterative, progressive, and integrated from the beginning of the development cycle. With the DevOps resulting in the integration of development and operations, software testing must now be conducted at every stage. Testing not only covers requirements verification but also the development of automation frameworks and the configuration of tool sets for integration and delivery. As Aruna Ravichandran, vice president of DevOps product and solutions marketing at CA Technologies, comments: *"In many ways, testing is the final frontier of DevOps and continuous delivery. To release the high-quality code that enables businesses to stay ahead of their competition faster, organizations must adopt next-generation testing practices that enable them to test early, often, automatically and continuously."*⁵ ⁶

⁵ <https://sdtimes.com/agile/continuous-testing-fast-paced-agile-devops-world/>

⁶ <https://sdtimes.com/agile/continuous-testing-fast-paced-agile-devops-world/>



Driver: Automated testing

As of today, the main utilization of test automation now focuses on UI testing, however Gartner highlights that testing automation is now about being able to create a set of effective tests that can be maintained with minimal effort and executed without any human intervention, as part of a DevOps toolchain. Automation is a key competitive factor in large online services. If too many person-hours are spent testing an application, updates will arrive late and the service is more likely to overlook a critical test due to release deadlines. Software testing automation includes tools for performing static code analysis, functional testing, and load and performance testing in an automated way. As industry analyst Gartner stresses: "Test automation tools are fundamental to achieving the continuous testing approach required by DevOps."⁷ Automated testing tools represent a logical first step. Organizations are discovering how shifting their focus upstream through automation can create repeatable and consistent processes to ensure a software quality standard, as new applications and processes are implemented. Automation is implemented for defect prevention, not detection since test automation right from the beginning of the process will prevent defects to get into the code. Building this kind of automation helps organization enforce a quality process without taking up extra resources or needing a separate security team. Automation increases because even if manual testing serves as an important factor, these tests do not scale. Automated test must fill the gap that manual test cannot achieve such as accurate test, repetitive, parallel check and expected behaviour and even amplification of tests. Once the code is analysed it must be packaged and deployed in various servers and resources to demonstrate that the application behaves properly end-to-end. The goal for automated testing is to ensure that the capabilities behave as planned/designed for. Also, testing ensures that there are no problems with environment specific settings or even with other software dependencies. Test automation tools are now an important way of ensuring software quality, moreover test automation is the key to continuous testing. Automation will go hand in hand with integration: to support smart testing and analytics, data has to be gathered from different sources and phases in software development, such as requirements management systems, change control systems, task management systems, and test environments. We will see test automation and management tools offering features to integrate with various ALM tool sets and test environments. This integration allows smarter decision-making concerning software testing and quality.

Driver: Continuous testing

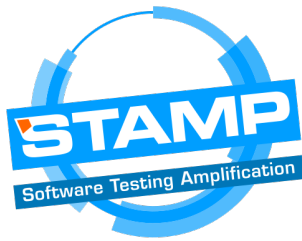
Continuous testing is the process of executing automated tests as part of the software delivery pipeline, to obtain immediate feedback on the business risks associated with a software release candidate. It draws enough attention for industry analyst Gartner to produce a "Continuous testing" Magic Quadrant. Continuous testing requires to integrate with and leverage different environments, including but not limited to development tools, continuous integration (CI), quality assurance (QA), pre-production, performance testing, staging and even production (e.g., by leveraging production monitoring). Continuous testing is a DevOps requirement. As Mr. Timothy Coats, Director of Applied Innovation, Trace3 explains: "It is critical to rapidly and effectively test integration points for functionality, operational efficiency, and security requirements. You can't really continuously deploy code, but you can continuously test. You need tools that enable the creation of testing environments, execution of test plans, and provide real-time feedback for all testing metrics. These tools also need to be able to validate deployments, identify when deployment issues arise, and reduce the mean-time-to-restore should you need to rollback unsuccessful code"⁸.

Driver: Cloud testing

Organizations are increasingly investing in cloud based solutions for their QA and testing requirements. A study by Technavio, suggests that there's a shift from usual bug tracking systems, to cloud-based bug tracking software. Cloud-based testing has greatly helped many mobile applications testing firms to reduce the cost involved in testing. Testing on cloud provides on demand testing services which enable handling of projects that are critical in nature and require faster time to market. The benefits are that cloud testing incorporates automated testing tools and covers a scope that goes beyond functional testing. In fact, one study published by Technavio [13] suggests that current bug tracking software are evolving towards cloud-based deployment models. This trend seems to have good adoption considering there are big players such as Google Drive and Microsoft Azure which have enhanced their offerings of cloud-based solutions. This shift

⁷ <https://www.gartner.com/doc/reprints?id=1-4KT1O8L&ct=171120&st=sb>

⁸ <https://devopsdigest.com/devsecops-security-tools-1>



avoids customers the creation of dedicated test infrastructure. Moreover, cloud-based bug tracking software help customers focus on their businesses and solutions instead of other issues. The shift toward cloud-based bug tracking software helps companies remain competitive and provide end-users the latest technologies. A report published in February 2018, quotes Charles Ramsey, CEO of Sauce Labs saying: “It’s clear from this year’s results that our customers not only need a fast and reliable cloud to execute their automated tests, they also need tools to help them debug faster”. A survey by Sauce Labs⁹ found that, in the past few years, the way development teams use cloud services for testing has changed dramatically. Teams are becoming more efficient and prescriptive when it comes to how they are using 3rd party cloud services for testing. In the first year of this study, most respondents (91%) reported that they used cloud services for all tests, but as cloud-based service offerings have matured and development processes have evolved, there has been a significant increase in the number of respondents who now report using 3rd party cloud services for specific tests. More than half (54%) are using cloud services for running specific tests, up dramatically from 9% in 2015. A similar trend is seen with spillover tests, which didn’t account for any cloud-based testing in 2015 but is now almost a fifth (19%). This increased focus may be reflective of the sophistication and maturation of the software development teams that are fully embracing healthy DevOps and Continuous Delivery practices.

Driver: Open Source tools

Open source has a strong impact upon the market. IT organizations increasingly seek open source testing tools, which are often less expensive from a total cost of ownership perspective and offer equivalent functionality, and also, several State-of-the-art testing tools are open source tools. Currently, open-source solutions are mainly oriented toward developers and focused on web and mobile technologies, making them somewhat specialized and not appropriate for all testing needs. Open source tools are also commonly used to achieve enhanced performance in terms of test automation with DevOps and Agile. The increased dependence of most software development teams on open source tools will drive innovation and expansion of new open source tools. It will also attract greater involvement from open source support communities. Prominent testing tool vendors, such as Atlassian, Microsoft, and Selenium, rely on open source communities and offer open source versions of their products. Gartner predicts that, by 2020, Selenium WebDriver will become the standard for functional test execution. Open source tools will make testing more readily accessible to development and DevOps teams to the point that they will adopt the “release early, release often” open source mantra as “test early, test often”. Industry analyst Gartner expects that most organizations will utilize more than one solution for the next five years. This echoes a survey by KPMG in which 64% of respondents declared using more open source tools than proprietary ones in their testing activity. Gartner also expects that open source will be a core element of a growing amount of cloud-delivered test automation services.

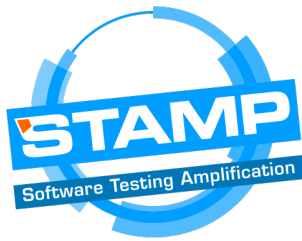
Challenge: Software complexity

Digital transformation adds more complexity to the testing world because devices are starting to take many different forms, provide more capabilities, and interconnect between features and functions. Because of this, more teams are starting to move to a software architecture that is extremely agile such as service-oriented architecture or micro-service architecture. Moreover In modern, online services, software is deployed across a multitude of operating systems, mobile environments, mobile devices, and IoT devices. This software complexity brings greater value within shorter delivery cycles, but it changes the conditions in which software is delivered and tested. While bringing clear advantages in terms of agility from an architectural and developer perspective, this approach in increases the heterogeneity and the number dependencies in the environment to be tested. In such complex software environment, problems can take many hours of lost business to fix, DevOps teams must be able to anticipate problems and this means something should always be being tested, whether it is unit tests, interoperability, or white box/gray box testing. Testing loops need to be automated and process loops need to be automated.

Challenge: IoT, mobility and security testing

The increasing development of mobile applications and enterprise mobility increases the demand for next generation mobile application testing environment. With digital transformation, many business are adopting processes that integrate Internet of Things (IoT) either within their back-end operations or in their relations

⁹ <https://saucelabs.com/news/Sauce-Labs-Releases-Third-Annual-State-of-Testing-Survey-Results>



with customers. As IoT becomes strategic, IoT devices and applications must be tested for security, performance and usability. All the IoT devices require Internet connectivity, thus vulnerable to security defects and risks. This ensures the need for IoT Testing. Mobile devices have become a part of everyday life and has evolved from a platform to communicate and access content to one that support business transactions. The development cycle of mobile applications is relatively short compared to web applications. There is a growing need for testing functionality, performance and security of mobile applications and devices. However, IoT and mobile solutions can be quite complex to test because they rely on a cloud-based integration layer that is often provided by a third party. Testing integration with these solutions can be quite challenging, the challenges of IoT testing increases every year. Due to the potentially limitless scope of situations encountered by IoT based applications, testing technologies AI and deep learning capabilities may be required to test them thoroughly.

Challenge: AI and deep learning

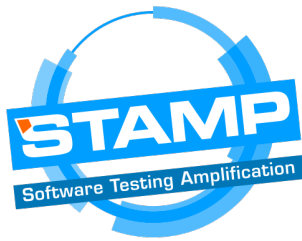
AI carries great expectations in software testing, specially a massive increase productivity by automating test-case creation to cover all user behaviors. But AI is also a target for testing since businesses are increasingly seeking to make competitive breakthroughs by utilizing artificial intelligence (AI) and machine learning to better understand and leverage captured data from user and customer behaviors and from their market environment. The need here is not so much to test AI itself but the applications using the outcomes of artificial intelligence. As a result, major software vendors will increasingly embed, or will be expected to embed machine learning within their core applications. This will challenge the testing community as new ways of testing and validating results from AI-enabled application and online services need to be developed. Gartner evaluates that by 2021, 50% of enterprises will leverage intelligent test automation driven by AI and machine learning. Already a company such as TestPlant offers Eggplant AI which uses artificial intelligence (AI) and machine learning to auto-generate test cases and optimize test execution to find defects and get broader coverage of the user experience, and RainForest QA offers an on-demand QA service with machine learning-verified results through a proprietary algorithm of over 70 million data points.

Challenge: Platform availability

In May 2017, WhatsApp, the world's most popular cross-platform instant messaging application went down, leaving many of its 1.2 billion users unable to send or receive messages. It's believed that an update to the application caused the downtime. A month earlier, the British Airways website went down for eight and a half hours due to a crash caused by an overnight database upgrade. Customers were unable to check in or access any flight details via the site. With many platforms offering similar capabilities, how will consumers decide where to spend their money? Differentiation of services will come down to availability, ease of use and a consistent, high quality experience. The 'uberisation' of services is based on a mesh of interconnected services governed by an API economy. In this context, users expect service when and where they want, there is no allowance for a breakdown in supply. Although some businesses are achieving great things in testing and monitoring terms, many organizations are still focusing efforts on speed instead of quality, security and resilience. Companies need to address the overall quality of their services to remain competitive. Consequently, there will be a shift in focus to monitoring the customer experience, as well as the need for thorough end-to-end (cross-services and cross-partner) testing, embedded within the delivery lifecycle.

Challenge: Software testers training

As the press puts it, "The world of software testing is currently experiencing a sea change that demands every software tester and Quality Assurance (QA) engineer to keep up with the latest trends." The continuing era of business digitalization and mobilization brings the new technologies in software development, along with the testing trends in quality assurance industry. Testing itself is an essential stage of software development where all the QA initiatives save developers' time and company's money by finding problems or bugs fast and effectively. Organizations need to invest in their developers, testers and release managers, and train them on what it means to be continuously testing. Software testers and QA engineers have to keep up with the latest testing trends to be able to present a high-grade product that clients demand. In order to do so, they must always be aware of what emerging technologies in software engineering are currently popular and worth the attention. For example, mobile application testing greatly differs from traditional desktop



testing due to its complexity and new challenges. Software testers will also need to adopt advanced programming skills to build libraries and automation frameworks to run their tests.

Challenge: Evolution of the test Center of Excellence

A conventional organization of testing is to place it under the responsibility or test Center of Excellence (CoE). The CoE's role is typically to provide a test team that delivers test as a service to the development team. The organization of the CoEs (which are the delivery engine for managed testing services contracts) needs to be adapted to the digital world. CoEs are now aiming to maintain their characteristics (offshore delivery, low costs, and delivery centralization) to adapt to agile development and testing projects. Growing adoption of DevOps erodes boundaries between developers and QA/testers roles and developers and operation managers start to accept testing activity as a part of their daily task. In real life developers love to catch bugs before it goes to the production environment, with this in mind, it is probable that in some organizations developers and testers role will merge in upcoming years. As a result, centralized CoEs may tend to become obsolete. One key reason is that with a CoE communication isn't as connected as it should be, causing testers to be informed late about changes and not responsive enough in a business context that requires quick reaction. Another reason is that the CoE may not be relevant anymore. Agile development requires cross-platform teams and the DevOps approach's key principle is that all departments developing and running software must collaborate seamlessly. This means that not only are teams made up of QA engineers, developers, and project managers, but the responsibilities are shared amongst the team members, the test automation developers become a part of the agile teams. Continuous testing makes testing activities part of everyone's role, from the business analysts, to the developers, to the performance engineer, to the operations manager, etc.

10. Industry trends: Supply side analysis

This section provides an analysis of STAMP's market context from the supply-side perspective. It introduces a segmentation of the market as per how analysts report how vendors structure their offerings and it identifies the main industry players.

Key products and services segments

This paragraph outlines the segmentation of the market following analysis by different market analysts firms. The exhibit below provides an example of how a leading vendor of software testing services lists its activities.

Application Testing. Our application testing practice offers a comprehensive suite of services in testing, consulting and engineering. Our quality engineering and assurance transformation services help customers develop deep, agile capabilities that create or extend their competitive advantage. Our business-aligned services in the areas of system and integration testing, package testing, user acceptance, automation, performance testing and test data management address our customers' critical quality needs. Consulting and infrastructure solutions in quality management, test tools and test infrastructure enable our customers to capitalize on emerging opportunities.¹⁰

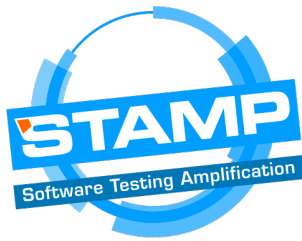
Exhibit 1: Cognizant's summary of their testing activities

The software test automation market offerings are comprised of tools, technologies, components and services that together constitute the critical elements of automated testing. It includes tools for performing static code analysis, functional testing, and load and performance testing in an automated way.

According to TMR¹¹ the global test automation market can be bifurcated into products and services, based on the type of components. Of the two, the services segment accounts for a majority of share in the market and is expected to retain its leading share by showing maximum growth due to the string trend toward managed testing services.

¹⁰ Source: Cognizant SEC filing Form 10k

¹¹ <https://www.transparencymarketresearch.com/test-automation-market.html>



The services segment itself can be further divided into advisory and assessment services and managed services. Similarly, the segment of products can be further broadly divided into test automation tools and type. The test automation tools can be segregated again into configuration testing, functional testing, web services testing, compatibility testing, security testing, acceptance testing, integration testing, load testing, mobile testing, migration testing, platform testing, usability testing, network testing, and QA process design. Of them, the functional testing segment accounted for most of the revenue in the global test automation market in 2016. This is because of the surging popularity of agile development practices that support a trend toward a higher degree of automation in the functional testing process.

According to Zion Marker Research,¹² on the basis of test types, global test automation market is divided into functional testing, configuration testing, web services testing, acceptance testing, compatibility testing, integration testing, load testing, security testing, mobile testing, migration testing, platform testing, usability testing, network testing and QA process design. Functional testing dominated the test automation market in 2016 and it expected grow during forecast period. It accounted for around 24% share of the total market in terms of revenue. Functional testing deals with the testing between the application on one side and rest of the system. Advanced properties and methods for functional testing of .NET, Java and web applications demands the functional testing tools which simultaneously increases the demand of test automation market.

In its directory of testing tools, QA Symphony suggests a detailed segmentation focused on testing tools: Test Automation tools which include Functional testing and Integration/API testing, Performance Testing tools, Test Management tools, Specialty testing tools (security, usability, compatibility, etc.), Test Setup and Infrastructure Testing tools, Environment Management Software Testing tools, Mobile Testing tools, Bug Tracking tools, Test Reporting and Test Metrics tools, Crowd testing tools, and Beta Management Testing tools.

Key industry players

According to independent analyst Gartner, leading testing service providers include Accenture, Applause, Atos, Capgemini, CGI, Cigniti Technologies, Cognizant, Deloitte, DXC Technology, IBM, Infosys, NTT Data, Planit, QualiTest Group, SQS, Tata Consultancy Services, Tech Mahindra, Wipro. Leading players in the global software testing market as identified by 1MarketResearch¹³ include: IBM, Capgemini, Wipro, Cognizant, HPE, Infosys, TCS and Hexaware.

Who are the main player more precisely in the Software Test Automation space?

According to Gartner¹⁴, they are: CA Technologies, IBM, MicroFocus, Microsoft, Parasoft, Ranorex, SmartBear, TestPlant and Tricentis. TMR¹⁵ provide a slightly different list with IBM, Wipro, Capgemini, Cognizant, Infosys, HP, TCS (Tata Consultancy Services), Hexaware, Wipro, Zensar, QASymphony, SmartBear, and Tricentis. And for Zion Market Research, key industry participants include IBM, Wipro., Capgemini., Cognizant, Infosys, HP and TCS. And for Marketsandmarkets key vendors providing automation testing solutions and services to commercial clients across the globe include companies, such as IBM, CA Technologies, Micro Focus, Capgemini, Microsoft, Tricentis, SmartBear, Parasoft, Cigniti Technologies, Ranorex and TestPlant.

And concerning the outsourced software testing market Technavio identifies Accenture, Amdocs, HP and IBM as global leaders ahead of other prominent vendors in the market such as Atos, Amdocs, CGI, Cigniti, CSC, HCL, HP, L&T Infotech, Logica, Mindtree, Software Quality Systems, Tech Mahindra, and Thinksoft Global Services.

These different lists suggest a significant portion of software testing offering comes out of India. Beside companies originated from the region such as Hexaware, Wipro, TCS and Infosys, even the testing services offering from global leaders such as IBM, Capgemini, Logica, Cognizant and HP are provided through operations based in India. This clearly reflects the dominant proportion in manual testing provided by competitively priced Indian workforce.

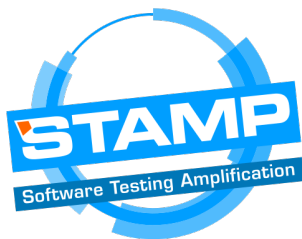
The software test automation market consists of independent vendors and test automation products that are part of larger vendors' portfolios or solutions. Clients can purchase any test automation product as a stand-

¹² <https://www.zionmarketresearch.com/news/test-automation-market>

¹³ <https://www.1marketresearch.com/market-reports/software-testing-market-125594/>

¹⁴ <https://www.gartner.com/doc/reprints?id=1-4KT1O8L&ct=171120&st=sb>

¹⁵ <https://www.transparencymarketresearch.com/test-automation-market.html>



alone offering. Some products have many optional and related components and most offer "free for evaluation" or open-source tools, which can make up a significant percentage of a vendor's user base. Gartner notes that many testing service providers have a test automation offering however, these are generally not offered independently of services. Over the years, a number of independent vendors in this space have been acquired by larger vendors, which demonstrates the criticality of automated testing in building a comprehensive testing tools and services offering. However, despite signals that may suggest consolidation of the industry, there are still dozens of independent vendors, as well as new vendors entering the market and established vendors transforming their products to offer enhanced test automation capabilities. Independent vendors in the software testing automation market have adopted partnerships, agreements, and collaborations as the key strategies to enhance their market reach.

With this perspective and using overviews provided by market analysts firms, we'd like to highlight vendors who seem to be potential targets, as partners, integrators or contributors, for STAMP. Two criteria seem most relevant in categorizing the main vendors: their orientation toward developing tools such as software testing automation or continuous testing platforms and their size (here we have the implicit hypothesis that large global established leaders will be less accessible than mid-tier companies). We identify three groups of companies:

- global leaders positioned by Gartner as providers of software test automation: IBM, CA, and Microsoft.
- global leaders in testing service: Capgemini, TCS, Infosys, Cognizant, TCS and Wipro.
- the mid-tier players including continuous testing platform vendors such as: Tricentis (QA Symphony was acquired by Tricentis), SmartBear, Parasoft, Testplant, MicroFocus and Sauce Labs and pure play testing services providers such as Planit and QualiTest.

11. Market sizing

Global market

Opinions differ quite significantly among analysts as to the size of the global market. While differences may come from different definitions of the market they mainly result from the fact that global leaders do not clearly identify their revenues in software testing products and services. Difference are not only found in the size of the market but also in its growth rates.

According to Nelson Hall¹⁶, the software testing services market is a maturing market, expected to grow by 6.5% over the 2016-2021 period, from \$20.4bn to \$27.9bn. More optimistically, a report published by Technavio predicts "the global software testing market to grow steadily at a CAGR of around 14% by 2021"¹⁷. According to another report by Technavio, the global pure play software testing services market is expected to reach USD 9.08 billion by 2021, growing at a CAGR of more than 19%.

At this stage in our analysis, we realized that Tricentis refers to Technavio's market evaluations in their presentations. Since Tricentis is a company with a technology positioning quite compatible with STAMP, i.e. oriented more toward the development of automation testing technology than selling large multi-year contracts of managed services, we decide to go along the Technavio's evaluations as shown below.

¹⁶ <https://research.nelson-hall.com/sourcing-expertise/it-services/software-testing/?avpage-views=article&id=80030&fv=1>

¹⁷ <https://www.technavio.com/report/global-it-professional-services-global-software-testing-market-2017-2021>

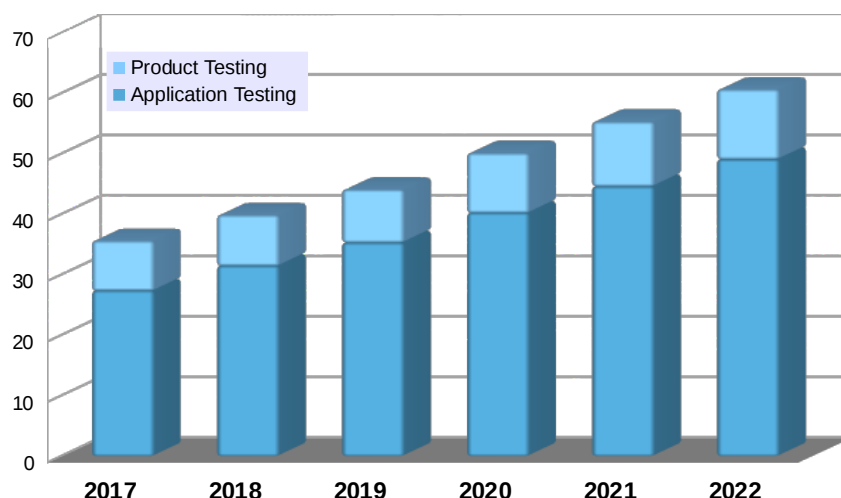


Figure 4: Evolution of the global software testing market (source: Technavio)

Key drivers responsible for the growth of this market include the rise in agile testing services, test automation services and cloud-based services that can be tuned to customer requirements.

In terms of geography, the Americas accounted for the maximum market share during 2016 and will continue to dominate the market for the next few years. The banking and telecom sectors formed a major portion of the demand for testing services in the region. Increased consumerization of data services and location-based applications in the telecom sector is also a key driver for the growth of software testing market in the Americas. Furthermore, the growing demand for cloud services will lead to the entry of new providers in the cloud-based testing services space, which in turn, will augment the growth of the software testing market during the forecast period.

Another source¹⁸ reports that with the slowdown in world economic growth, the Software Testing industry has also suffered a certain impact, but still maintained a relatively optimistic growth, the past four years, Software Testing market size to maintain the average annual growth rate of 2.87% from 914 million \$ in 2014 to 995 million \$ in 2017, BisReport analysts believe that in the next few years, Software Testing market size will be further expanded, and expect that by 2022, the size of the software testing market will reach 1094 million \$.

Market by geographies

According to TMR¹⁹, from a geographical standpoint, North America grosses maximum revenue in the global test automation market. In 2016, it contributed about 46.0% share in the revenue in the overall market. In the years ahead too, North America is slated to hold on to its leading share owing to the increasing traction of QA within organizations, as more transformational projects are implemented to bring down the time-to-market and augment returns on investment (ROI). Europe trails North America in the global test automation market on account of the rising sophistication and maturity of the testing/QA sector.

In terms of growth rate, however, Asia Pacific is expected to surpass all other regions by clocking a strong 16.5% CAGR between 2017 and 2025. The growth in the region will mainly be brought about increasing investments in internal professional testing resources, existing tools, and external outsourced resources.

All market analysts seem to agree on North America being the largest geography at more or less 45% of the global market as for markets in Europe and Asia Pacific, while opinions vary as to which one is the largest, analysts concur in giving them more or less the same size. In terms of growth rate however their all seem to agree on Asia Pacific being the fastest growing region at more than 15% CAGR.

¹⁸ <https://www.wiseguyreports.com/reports/3203733-global-software-testing-market-report-2018>

¹⁹ <https://www.transparencymarketresearch.com/test-automation-market.html>

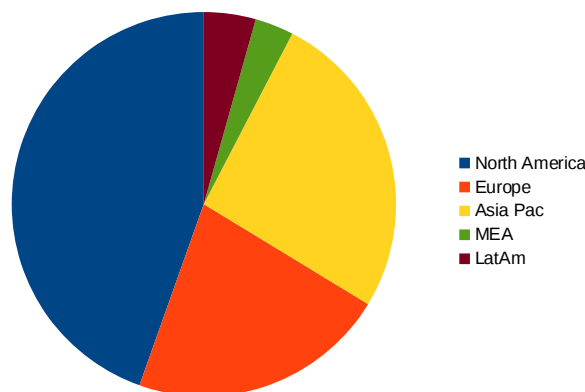


Figure 5: Global market breakdown by region (source: OW2 synthesis)

According to a report by Zion Market Research, the global test automation market was valued at approximately USD 15.87 billion in 2016 and is expected to reach approximately USD 54.98 billion by 2022, growing at a CAGR of 23.01% between 2017 and 2022. The speedy deployment of mobile applications and Software as a Service (SaaS) on cloud is one of the major reason for the growth of the test automation market in North America.

According to a report by Marketsandmarkets, the cloud testing market accounted for USD 5.13 Billion in 2016 and is projected to reach USD 10.24 Billion by 2022, at a CAGR of 13.01% during the 2017-2022 period.

Market by verticals

Main vertical sectors identified by market research companies in the domain of software testing are: bank, Financial Services and Insurance (BFSI), Information Technology, Telecommunications, Retail, and other sectors.

Technavio found that, in 2015, the financial services sector led the software testing services market with more than 34% of the total market share. Almost all the financial application development and maintenance requires a high degree of testing services throughout the software development lifecycle. Intense competition among global leaders and the shift to specialized services are pushing financial institutions to seek for enterprise-grade testing services.

12. Business opportunities

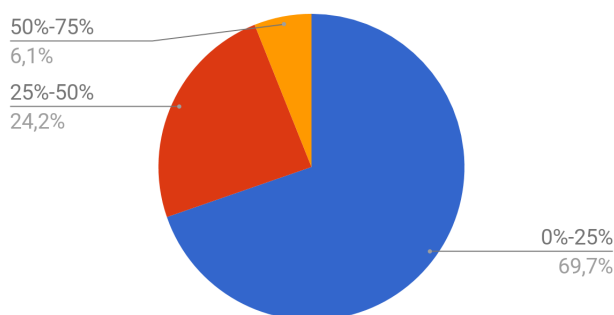
This section provides the bottom up analysis of STAMP's business opportunities from the standpoint of the results of the STAMP-specific questionnaire introduced in section 7. As indicated, an online survey was submitted to respondents including IT specialists approached via events or recommended by STAMP's partners either professionals from industry (i.e ATOS, Engineering) or academic (i.e University Rey Juan Carlos in Spain), and other EU-funded projects (i.e [Elastest](#)). Questions were designed to capture key information on current experiences, challenges, and trends in software testing. Participants included a different range of roles, vertical industries or interests.

Although the survey was answered by only 33 specialists, results point to the same trends and similar considerations as in the current general figures of the testing market mentioned in the top-down analysis. Main results of the survey are presented and, when relevant, balanced with results from a comparable survey reported by Capgemini.

Testing adoption

Survey's answers suggest the same need and focus on test automation as top-down analysis. Respondents in the STAMP survey all agree that there's an increased investment in effort spent in testing and in QA. The present figures and charts below support the findings along the overall trends mentioned before. The first figure shows that almost 70% of respondents see that there is a stable investment in QA and testing efforts. At the same time, the second chart shows that 63% think that the future and changes in QA and testing are very stable and foresee more investments.

Effort spent in QA and testing



Future change in QA and testing effort

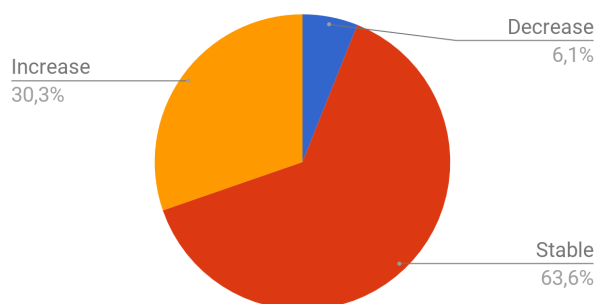
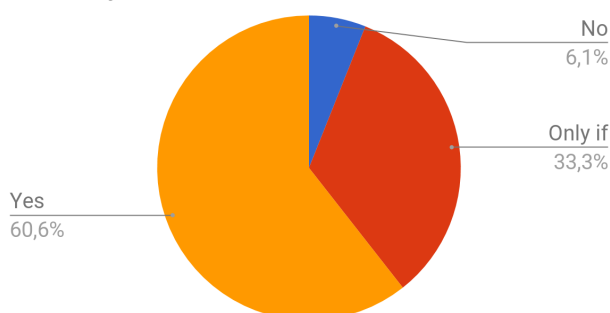


Figure 6: Effort spent on testing and QA (STAMP survey)

Also, 94% are in favor of trying out new tool: this represents an immense opportunity to position STAMP in this respect and offer our amplification for further market uptake, if possible. Albeit, a third will consider it "only if" this new tools are integrated in the tools they are already using, e.g. presented as a plug in for their continuation integration factory.

Would you consider new tools?



Figures 7: New tools adoption (STAMP survey)

Considering the way these tools could be delivered, 82% of the survey respondents are in favor of SaaS delivery (with 33% of them who already use SaaS for testing and QA). Paying software licenses for testing and QA is not an obstacle for 76% of STAMP respondents (with 30% of them who already pay licenses). On the other hand, almost half (48%) of the respondents will not consider external consulting for such purposes. I.e., respondents do not envision to externalize their software testing and quality assurance processes to another company.

For your QA & testing, would you

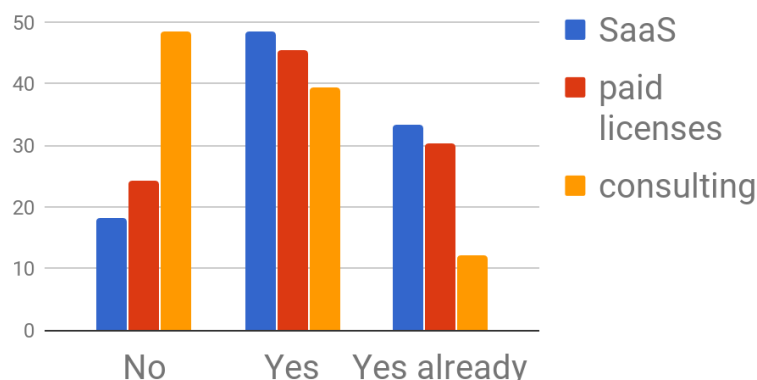


Figure 8: Consideration of delivery in QA and testing processes

The most suitable delivery method for STAMP tools to market will be established in the following month when building the STAMP exploitation and business plans. Anyway, cloud-based applications and SaaS tool election support our theory and intuition when launching the survey. STAMP survey respondent are willing to accept to use SaaS and to pay license for their testing and QA tools.²⁰

Agile adoption

With a 77% agile adoption rate, results in the STAMP survey are more mitigated than other comparable surveys. For instance the already mentioned Capgemini report states that 96% of organizations have adopted agile methodology for at least some of their project in 2017 [2]. The SauceLabs report while more nuanced, describes an 88% agile adoption rate in 2016 [3], and finds that only 21% of agile development team are reported having adopted the five agile testing practices considered (partnership between development and QA, real-time communication between development and QA, immediate bug fixing, high test automation, iterative development and testing).

Software Development Process

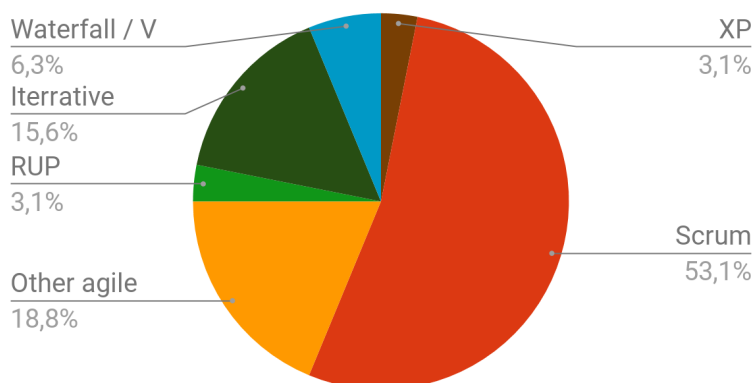


Figure 9: Software development process of STAMP survey respondents

Combining agile approaches with waterfall methods is the preferred standard for testing in agile context. Bottom-up finding from the STAMP questionnaire are quite consistent however with the executive-oriented Capgemini's survey which states that the Scaled Agile Framework (SAFe) and the Dynamic Systems

²⁰ Even if this observation does not implies that all tools must be delivered in this way.

Development Method (DSDM) are the most implemented agile testing methodologies and are rapidly growing, with respectively 59% and 50% adoption reported.

Considering the DevOps adoption, numbers are less impressive, demonstrating a maturing phase ongoing with a better appreciation of the real purpose and application of DevOps approach. DevOps first effects on delivery pipeline automation and tool implementation are now adopted but breaking down the silos of business, development, quality and operations requires a cultural change in software engineering. Albeit implementation of Quality Assurance in the development cycle could be buried among other DevOps implementation challenges, QA guideline and toolkit for DevOps can be the enabling glue that keeps engineering teams focused on customer value and business relevance.

The Capgemini report states that 88% of organizations have adopted DevOps for at least some of their projects in 2017 [2]. However, respondents shifted from a massive declared usage of DevOps principles in 2015 (58% of respondents declared to use DevOps for more than 50% of their projects) to a more focused usage in 2017 (47% of respondents declared to use DevOps for less than 20% of their projects).

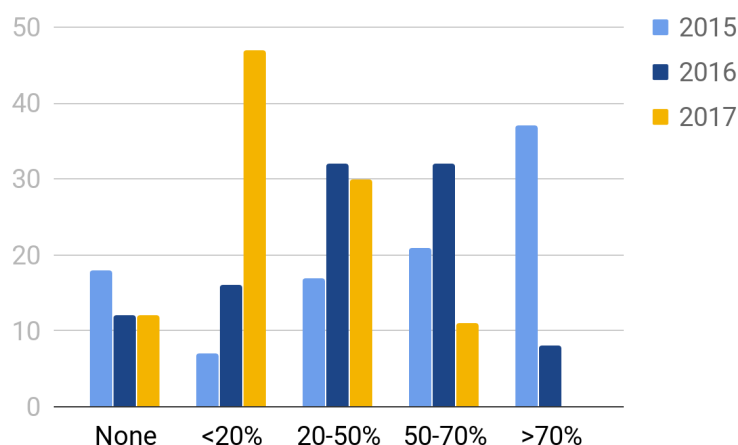


Figure 10: Proportion of projects using DevOps principles within respondent organization [2]

Test environments

Test environments allow software tester to execute the designed test cases; they require computing capabilities, dedicated software and test data. The environments can be deployed on in-house (50%) or cloud servers (45%) that can be permanent (27%), virtualized (58%), or containerized (15%). However, we notice that 70% of respondents still also use local devices (such as desktop or mobile phone) to run tests [2].

The management of these environments involve challenges on their own. The respondent cite the “inability to manage excess needs for test environment” (48%), “lack of visibility of test environment availability” (47%), and “lack of availability of right tools for testing” (43%) [2]. Respondents of STAMP survey particularly cited “UI Testing”, “multi-browser testing”, “RDBMS mock” as features needed in their test tools.

This context does not impact STAMP fundamental scientific results directly, but impact their future exploitation and business plan. Indeed, the STAMP tools exploitation should be realized into scalable, monitorable and accessible environments. Hopefully, the integration of STAMP in both Jenkins and ProActive scheduler constitute an opportunity for STAMP exploitation. Also, containerization, that recently and strongly appears constitute an opportunity and challenge for STAMP tool exploitation as need to ensure that our tools are able to run in a containerized environment. First results in this domain are encouraging, especially with configuration testing that already rely on docker for both input and execution.

Test automation

By pushing test automation to new standards (automatic test case generation, automatic test case validation, and automatic configuration testing), STAMP project address test automation challenges with strength. However, according to the different studies, basic test automation (non-manual testing) is far than complete.

Only 26% of respondents of the Sauce Labs survey consider their tests as “mostly automated” or “fully automated” (3%). The STAMP survey obtains a average 4,03 test automation level on a scale from 0 to 10. More surprisingly, 12% of respondent reporting no automation at all.

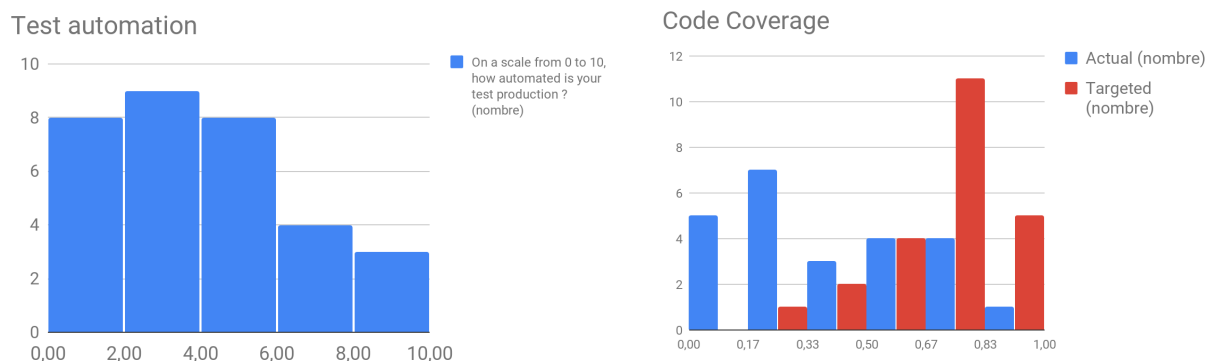


Figure 11: Test automation perception in STAMP survey

The code coverage level reported by STAMP survey respondent is consistent with the perceived automation degree, since the average code branch coverage reported is 41% (when known), while the targeted code branch coverage is 73%. The test automation degree reported is also consistent with the one reported in the SauceLabs survey.

HOW WOULD YOU CHARACTERIZE THE AMOUNT OF EFFORT THAT IS PUT INTO AUTOMATED AND MANUAL TESTING?

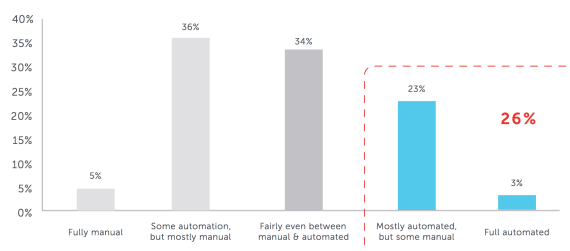


Figure 12: Test automation perception in [3]

At the level of software framework to automate tests execution we notice an adoption of JUnit (71% for all versions of JUnit) and Selenium (42%). The most used framework for mocking is Mockito (50%), while 20% of respondents do not report usage of a mocking framework.

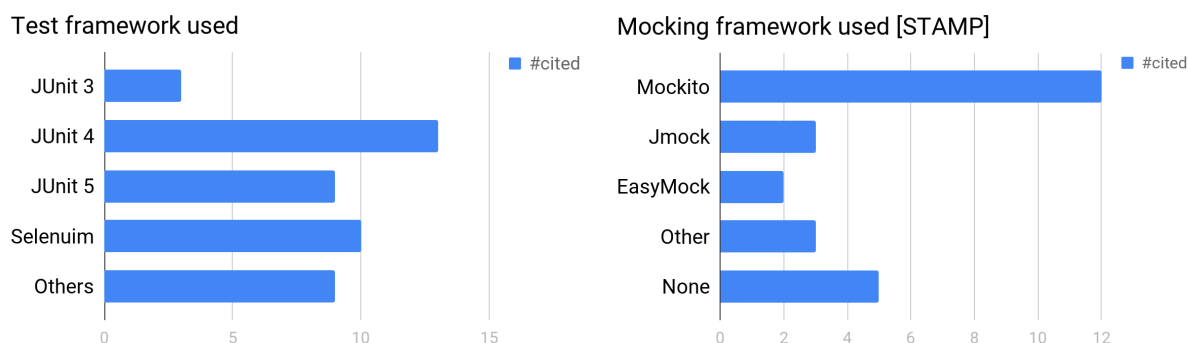
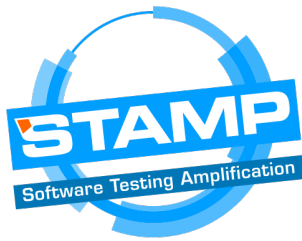


Figure 13: Automated test tools usage (STAMP survey)



Such an automation degree is both an opportunity and a treat for STAMP outcomes. It's an opportunity for DSPOT mutation testing tool and Evocrash that are able to increase code coverage. However, it's also a treat, since DSPOT and Descartes tools require existing JUnit tests to generate and validating test cases, if there is no automatic test case, or non-JUnit test suites, as reported by 33% of respondents, these STAMP tools are not usable. Benefits expected from automation are important, CapGemini survey reports that users expect better detection of defects, reuse of test case, or code coverage with reduction of test cycle time and costs.

Technologies

This paragraph introduces, finding from a number of source, on widespread technologies that have an important ratio of market share in 2018 and that must be taken into consideration in a STAMP exploitation plan:

- Java is used by 14% of GitHub active users (ranked 3rd),²¹ and is consistently ranked 1st in the TIOBE index, with 16% rating in May 2018²² (the ranking difference might be due to higher usage of java in closed source / enterprise software)
- JUnit is consistently reported as the de facto standard for unit testing in Java. For instance, an analysis of the most starred pure java project in GitHub, shows that 42% of them use JUnit.²³ Our
- Docker is consistently reported as by far the leading container technology, in growing adoption.²⁴
- Maven is also reported as the most used build tool in the Java community. RebelLabs survey reports a 68% adoption rate, the second being Gradle with 16% [4]. An other analysis reports a growing adoption of Gradle, almost to the level of Maven.²⁵ However, the STAMP tools may consider to allow installation with Gradle without such much effort.
- Considering PIT, since mutation testing adoption is at its beginning, it is hard to derive significant numbers. Using a quick analysis of GitHub repository through Google BigQuery, we obtain that 10 out of 5062 sample GitHub maven configuration files (pom.xml) refer to PIT. Its usage is probably greater than that since PIT can be used with a command line without modifying the pom.xml file. This result compares to none for Grester and Jumble, the only other known mutation testing tools that support maven. However, we may not consider this low number of references as a treat for the STAMP tool adoption. Contrary to JUnit, Maven or Docker, we do not require that developers change their usual tools to match our own requirements, but only add a new library which is open-source and freely available.

```
select count(*) from [bigquery-public-data:github_repos.sample_contents]
where sample_path LIKE '%pom.xml' AND content LIKE '%org.pitest%'
```

Exhibit 2: BigQuery script to estimate proportion of GitHub projects that use PIT

²¹ <https://www.benfrederickson.com/ranking-programming-languages-by-github-users/>

²² <https://www.tiobe.com/tiobe-index/>

²³ <https://blog.takipi.com/the-top-100-java-libraries-in-2017-based-on-259885-source-files/>

²⁴ <https://www.datadoghq.com/docker-adoption/>

²⁵ <http://glaforge.appspot.com/article/gradle-vs-maven-and-gradle-in-kotlin-or-groovy>

Benefits of test automation

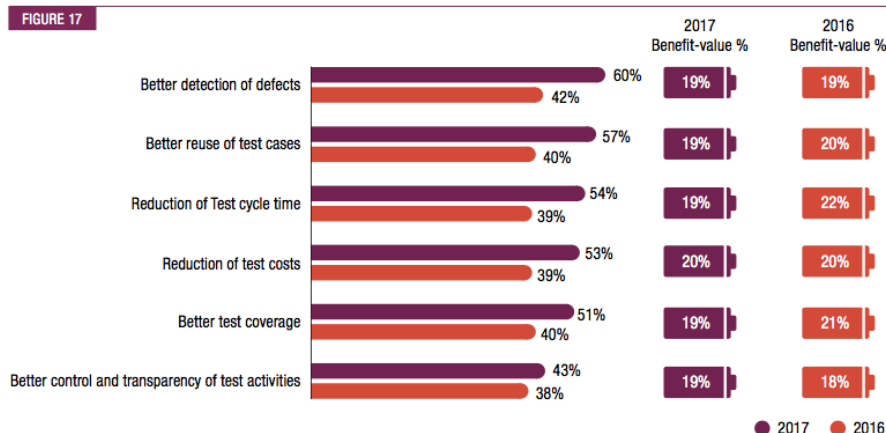


Figure 14: Benefits of test automation [2]

There is also expectations for automatic test case generation, with 12% of STAMP survey respondents freely suggest some sort of automatic test case generation as a needed feature, and 48% of Capgemini survey's respondent citing "test design automation" as an emerging automation technique. Note that the STAMP survey respondents expect to obtain this automation from very different means: "machine learning", "mutation testing" or "impact analysis".

STAMP integration

STAMP is a tool set destined to be integrated in a "test environment"; selecting the most suitable target environment to integrate is more an exploitation plan issue, but observations suggest the following.

Integrated Development Environment. IntelliJ IDEA is reported as most adopted IDE (Integrated Development Environment) for the first time in the RebelLabs survey series [4]. IntelliJ is adopted by 46% of respondent, the second (and formerly the first) one being Eclipse by 41%. The growing market share of IntelliJ is quite a good news for STAMP tool adoption. Indeed, STAMP tools are well integrated with maven for installation but also for execution, and maven usage is more convenient in IntelliJ than Eclipse. The integration in Eclipse through maven issue was reported during the first validation phase of WP6 [5]. So far Eclipse plugins for Descartes and DSpot, developed by Atos, are available on the STAMP Github (<https://github.com/STAMP-project/eclipse-ide>).

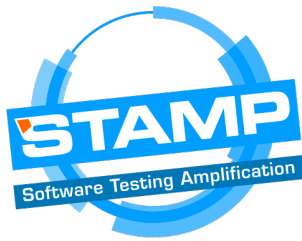
Continuous integration. Jenkins is by far the most used Continuous Integration Server, by 60% of RebelLabs survey respondents [4] and 68% of respondents of sauce labs survey [3]. XWiki STAMP partner are now integrating the STAMP tools into Jenkins through Jenkins Pipelines [5], demonstrating integration of, at least, Descartes and DSPOT, into Jenkins. A prototype of providing Descartes as a service on a pull request in Github is also available as a Github application.

Direct competition

As of today, STAMP represent a tool suite with no direct competition. However several research efforts are targeting the same functional space. This section, presents a quick overview of direct competition to the STAMP tool set component by component.

DSpot. The idea of amplification, i.e., start from existing test cases and automatically generate new ones that complement the existing ones, is completely new. To our best knowledge, there is no commercial or known open source tool proposing test amplification.

Descartes. Descartes extends the PIT tool for mutation testing. PIT and Descartes differ by the kind of mutation they perform on application code to evaluate the quality of a test suite. Descartes uses coarse grained mutations, it generates less mutants than Pitest, hence scales better than default PIT engine on large programs.



By now, there only exist research products in the domain of mutation testing, thus this constitute an opportunity. However, big company like Google [6] begin to use internally mutation testing engine at very large scale. If such company release their engine in a commercial or open-source product, that will be an important threat to Descartes adoption.

CAMP. CAMP is based on two ideas: 1) applying lightweight virtualization technologies to formalize possible configurations and execute software, e.g. Docker 2) capturing variations with help of well-defined formalism in software product lines, e.g. features. To our knowledge, there is not any well-known commercial or open-source tool which combines those two methods to capture variations, generate concrete executable configurations, e.g. docker images and docker-compose files, deploy these configurations and execute test cases to verify functionality or certain properties of a product.

Evocrash Evocrash automatically generates test cases that reproduce a crash execution using genetic algorithms. By now only academic research approach propose equivalent feature. MuCrash uses mutation testing [7], STAR uses symbolic execution [8] and JCharming uses model checking [9]. EvoCrash outperforms them in terms of the number of crashes it can reproduce [10]. We must also point out that none of those tools can be found online and it seems that they are not maintained anymore.

Sapienz [11] could be seen as a more serious competitor for EvoCrash engine. It was developed by Mark Harman (formerly of UCL), now at Facebook (Facebook explicitly bought him and the technology).²⁶ However, Sapienz is more of a competitor to EvoSuite, the genetic algorithms technology that EvoCrash builds on to replicate crashes, i.e., it doesn't start from a crash, but from a piece of software (with or without tests). What is particular in the case of Sapienz is that it relies heavily on seeding.

Related, but no dedicated to crash replication, we also have **Infer** (also at Facebook)²⁷. The tool uses static analysis to discover paths in a method that can trigger an exception. As for any other static analysis tool, they have some difficulties to scale and, for now, they only support NullPointerException.

SWOT

The following Strengths, Weaknesses, Opportunities and Threats (SWOT) matrix is a first synthesis of factors that can influence the development of STAMP's market position. The SWOT analysis involves specifying the objective of the business venture or project and identifying the internal and external factors that are favorable and unfavorable to achieve that objective.

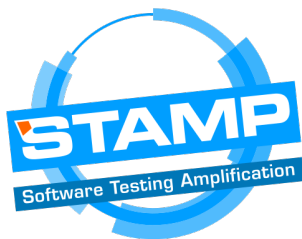
1. **Strengths:** characteristics of the business or team that give it an advantage over others in the industry.
2. **Weaknesses:** are characteristics that place the firm at a disadvantage relative to others.
3. **Opportunities:** external chances to make greater sales or profits in the environment.
4. **Threats:** external elements in the environment that could cause trouble for the business.

The following table presents a first version of the STAMP's SWOT.

Strengths	Weaknesses
<ul style="list-style-type: none">• STAMP focus on DevOps and CI• Need for more automation• STAMP processes and tools are adapted to agile delivery models• Integration into CI tools (Jenkins)• STAMP tools are built upon leading technologies• No serious competitors in mutation and online testing	<ul style="list-style-type: none">• Not addressing UI testing• Not addressing web testing• Not addressing testing data issues

²⁶ <https://code.facebook.com/posts/134560877297803>

²⁷ <http://fbinfer.com>

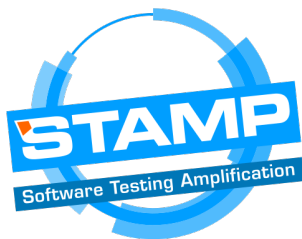


Opportunities <ul style="list-style-type: none">• Maturity of DevOps adoption in organizations• Emergence of test excellence center• Automation level is still low (unsatisfactory testing tools or insufficient automation)• Test design automation viewed as a emerging technique• Growing market share of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

Table 3: SWOT STAMP market analysis

13. Conclusion

This report presents an overview of the STAMP project market environment. We already knew the technologies developed by the STAMP project were advancing the state of the art in software testing. What the market analysis shows is that these technologies are aligned with key market drivers such as digital transformation, the adoption of agile and DevOps methods. The analysis also explores the competitive environment and identifies key players that could become STAMP partners or OEM adopters. A later version of the market analysis could drill down on some specifics of the market and pave the way for the development of successful exploitation plans.



14. Annex: Panorama of tools in the software testing and QA environment.

Testing tools are offered by platform vendors, service vendors and they also come as freely accessible open source software. Testing tools come in a wide variety of flavors and specialties. QA Symphony offers a directory of over 100 testing tools organised in eight following eight categories.²⁸ Capterra²⁹ is another provider of automated testing software directory. Here is an excerpt of the QA Symphony directory:

Test Automation Tools.

Test automation is one of the most mature software testing segments. Currently, the rise of good open source options and the push toward DevOps and multichannel models are transforming this segment. We can break down test automation tools into two subcategories: Functional testing and Integration/API testing. Functional Testing tools drive automated tests by interacting with the UI layer of an application. These tools usually provide a record and playback interface, making it easier for nontechnical testers to provide automation coverage. However, these tests can be slow and brittle. Functional testing tools include: Katalon Studio, HP Unified Functional Testing, IBM Rational Functional Tester, Tricentis Tosca Testsuite, Worksoft Certify, TestPlant eggPlant Functional, Ranorex,

Integration/API testing tools send various types of payloads through APIs to make sure they accept valid inputs and reject improper ones with proper messages. Some of these software testing tools can also provide testing for load. Integration/API testing is especially important for cloud, mobile and PaaS. Integration/API testing tools include: CA Technologies Application Test, IBM Rational Test Workbench, Parasoft SOAtest, SmartBear Ready! API, Crosscheck Networks SOAPSonar.

There are also several open source options for integration and API testing of which QA Symphony lists a dozen: Apache Jmeter, SoapUI, PowerMock, EasyMock, REST Assured, WebInject, Mockito, Jmockit, mocktail, WireMock, Betamax and Mockserver.

Performance Testing Tools

Performance testing tools take a simple automation script and run it through hundreds or thousands of machines to simulate how an application will perform under load. These tests are typically expensive and are run less frequently than functional automation tests. Performance testing is particularly important for cloud applications and there are two main types to consider: Pure Play and Extension Tools. Examples of performance testing tools include: Automation Anywhere Testing Anywhere, BlazeMeter, Borland Silk Performer, CA Technologies Application Test, HP LoadRunner, Performance Center & StormRunner, IBM Rational Performance Tester, Neotys NeoLoad, Parasoft Load Test, Radview WebLOAD, SmartBear LoadComplete, Soasta CloudTest, TestPlant eggPlant Performance,

Test Management Tools

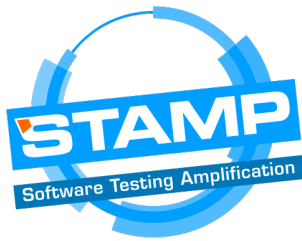
Test management is a mature testing segment currently being transformed by the widespread adoption of Agile, DevOps and Test Driven development. Most test management tools are either geared toward Agile or Waterfall development methodologies. Examples include: qTest Manager by QASymphony, SmartBear QAComplete, Zephyr, Testuff. QA Symphony also lists the following open source tools: Trantula, Testopia, QABook, TestLink, Xqual, TestMaster and Testitool.

Specialty testing tools

They include tools for testing security, usability, compatibility, etc. It is more of an emerging segment featuring typically expensive and specialized tools. These tools are evolving, becoming easier to use and offering improved service, to assist smaller teams who cannot yet invest in dedicated resources.

²⁸ <https://www.qasymphony.com/blog/100-plus-best-software-testing-tools/>

²⁹ <https://www.capterra.com/automated-testing-software/>



Examples of Security Testing Tools include: HP Fortify On Demand, Veracode and IBM Application Security APPScan. Examples of Usability Testing Tools: UserTesting, Validately. Compatibility Testing Tools: Browsera, Browserling,

Test Setup & Infrastructure Testing Tools

Test setup and infrastructure is a rapidly changing segment due to trends in mobile, web (cloud), NoSQL and containers, with many new companies emerging. QA Symphony breaks down test setup and infrastructure tools into four subcategories: Test cloud, Mobile device farm, Test data management and Environment management.

Test cloud tools provide a network of virtual machines where desktop and web testing can occur at a lower cost, saving organizations money. Containers and cloud providers are lowering the barrier to entry for this market. Examples of Test Cloud tools include: Sauce Labs, HP StormRunner Load, IBM Smart Business Development and Test Cloud, Borland Silk Central Connect, BrowserTap,

Mobile device farms provide a network of mobile devices for testing to occur, saving device acquisition, management and maintenance costs. A recent entrance by Amazon with a fixed price offering has shaken up the market. Offerings include: Amazon Web Service Device Farm, Borland Silk Mobile Testing, Experitest SeeTestCloud, HP Mobile Center, Keynote Mobile Testing, Mobile Labs deviceConnect, Perfecto Mobile Monitoring, Sauce Labs, Smart Software Testing Solutions pCloudy, Soasta TouchTest, TestPlant eggCrate, Xamarin Platform.

Test data management tools automate the process of deploying “gold” test data to databases. The introduction of open source databases first shook up the market, and it is facing change again due to the rise of NoSQL/Big Data stores. Examples include: Informatica Test Data Management (TDM), CA Technologies Test Data Manager, HP Test Data Management, IBM InfoSphere Optim, Delphix Engine and Data Masking,

Environment Management Software Testing Tools

Environment management tools automate the process of deploying software into a test or staging environment. These tools are increasing in importance alongside the rise of DevOps due to quicker builds and the need for automation. Docker, Puppet and Chef have paved the way for newer, cheaper and better environment management solutions. Examples of Environment Management tools include: Plutora Test Environment Manager (TEMS), TestPlant eggPlant Manager, TEMS Omnium,

Mobile Testing Tools

Mobile testing has become increasingly critical as mobile device usage grows ubiquitous. Given the variety of application types (native, hybrid, mobile web) and operating systems, testing mobile applications can prove difficult. Mobile testing tools use automated testing frameworks to help simplify this process. Examples of Mobile Testing tools include: Keynote DeviceAnywhere, TestPlant eggPlant Mobile, Appium, Selendroid, ios-driver,

Bug Tracking Tools

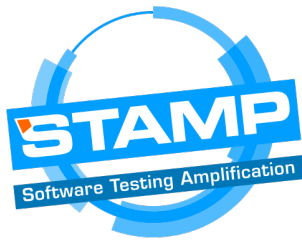
Bug tracking tools simplify the process of identifying, managing and reporting on software bugs. These tools strengthen collaboration, streamline recording and reporting on bugs and ensure consistency across the board, especially when the alternative is a spreadsheet. Examples include: Bugzilla, MantisBT, Atlassian JIRA, FogCreek FogBugz, Bontq, BugAware,

Niche Testing

Niche categories are emerging to handle the new problems being created from Agile, mobile, cloud, DevOps and best of breed movements.

Test Reporting & Test Metrics Tools centralize and standardize the reporting around testing activity from independent tools, bridging the Agile/Waterfall, cross-platform and manual vs. automated testing metrics divides. Examples: qTest Insights by QASymphony, Testuff.

Logging/debugging tools collect information around technical and functional errors from pre-production and production environments. Production logging is mostly useful in mobile and cloud applications. Examples:



QASymphony qTest eXplorer, HP Sprinter, Instabug, TestFairy, Browserling, BrowserStack, CrossBrowserTesting,

Crowd testing tools use “the crowd” to assist in software testing through web-enabled platforms. Crowd testing reduces the general cost of labor as well as the cost of access to specialized devices, networks and skill sets. Crowd testing communities are either managed or unmanaged. A managed community is one where the vendor manages interactions (e.g. bug verification) with the crowd. These communities minimize the effort required from your business and provide a much higher quality service, but they are also more expensive. Meanwhile, an unmanaged community is one where your business manages interactions with the crowd. These communities require more effort from your business, but they are also less expensive. QA Symphony list over 20 different tools.

Beta Management Testing Tools

Beta management tools control the beta testing process from app deployment to monitoring by collecting feedback and instructing beta participants how and where to test the application. They include: BetaEasy, Applause Mobile Beta Management,