European Commission
Horizon 2020
Communications Networks, Content & Technology
Cloud & Software
Research & Innovation Action
Grant agreement n° 731529

Title:	WP6 – D6.4 – Exploitation Plan
Date:	June 20, 2019
Writer:	Pascal Urso (Activeeon), Veronika Tsiareshchanka (Activeeon), Malena Donato (ATOS)
Reviewers:	Benoit Baudry (KTH), Ciro Formisano (Eng), Caroline Landry (INRIA), Cédric Thomas (OW2)

Table of Content

List of figures

List of tables

1. Executive Summary

This deliverable covers the exploitation plan of the STAMP consortium within and beyond the project's period. It contains the description of the exploitable results identified by the partners, the exploitation activities and means to achieve sustainability of the results along with some initial information on foreseen commercialisation strategies.

Some STAMP results are already deployed in professional software development pipelines and validated in real-life production environments. This reflects the technology readiness of the results and sets the stage for their long-term exploitation. The STAMP partners plan to exploit the results of the project essentially in large scale service offerings within their company or their organization, and in academic courses and professional trainings.

This deliverable will be completed by deliverable 6.5 "Business Plan", that will analyse further the commercialisation strategies.

2. Revision History

Date	Version	Author	Comments
15-jan-2019	0.01	Pascal Urso, Veronika Tsiareshchanka (Activeeon)	First draft of the deliverable
22-mar-2019	0.02	Pascal Urso, Veronika Tsiareshchanka (Activeeon), Malena Donato (Atos)	Second release.
27-mar-2019	0.03	Pascal Urso, Veronika Tsiareshchanka (Activeeon)	Objectives, KER, exploitation options
22-apr-2019	0.04	Pascal Urso (Activeeon)	Licenses first analysis
15-apr-2019 to 02-jun-2019	0.05	Daniele Gagliardi (ENG) Benoit Baudry (KTH) Caroline Landry (INRIA) Andy Zaidman (TUDelft) Caleb James DeLisle, Vincent Massol (Xwiki) Malena Donato (Atos) Pascal Urso (Activeeon) Hui Song (SINTEF) Lars Thomas Boye (Tellu) Olivier Bouzereau (OW2)	All Exploitation plans
24-may-2019	0.06	Pascal Urso (Activeeon)	Licence analysis, Roadmap for exploitation
30-may-2019	0.07	Malena Donato (Atos)	Preliminary business models
5-jun-2019	0.08	Pascal Urso (Activeeon) Malena Donato (Atos)	Sustainability map, introduction, risks, executive summary, prepare final version for revision
6-jun-2019	0.09	Ciro Formisano (Eng) Benoit Baudry (KTH)	Reviewers
10-jun-2019	1.00	Pascal Urso (Activeeon) Malena Donato (Atos)	Revision
10-Jun-2019	1.01	Caroline Landry (Inria)	Adding INRIA contribution Completing references section
17-jun-2019	1.02	Cedric Thomas (OW2)	Review

19-jun-2019	1.03	Pascal Urso (Activeeon)	Overall revision
20-jun-2019	1.04	Cedric Thomas (OW2)	Final review
21-jun-2019	1.05	Benoit Baudry (KTH), Caroline Landry (INRIA)	Review and edition

3. Objectives

The present document contains the analysis performed by all STAMP partners in the context of the work package 6 (WP6) to agree on the common aspects of their exploitation activities and business issues.

The aim of STAMP is to develop and validate test amplification techniques. Specifically, STAMP aims at developing 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. The developed testing amplification tools aim at helping DevOps in three main areas:

- Detect more regression bugs on continuous integration servers, before functional testing.
- Detect more scalability bugs, before going to production and experiencing bad behaviour (trashing, freezing) with high user load.
- Reproduce more production bugs in edge cases thanks to semantic logging.

Therefore, we aim at adding value to human labour and eventually reducing the number of bugs in production while limiting the risks associated with code updates**.**

The objective of this deliverable is to share the exploitation plans and agreement of the consortium partners on the following five key issues:

I. To identify which elements developed in the STAMP project are best candidate for exploitation activities.

II. To verify the suitability of open source licenses chosen for the STAMP developments, to support a large number kind of exploitation schemes, including commercial ones.

III. To clarify the exploitation intentions of the partners and to ensure that these intentions make a good usage of exploitation capabilities of the partners.

IV. To plan the sustainability of the STAMP result after the end of the project.

V. To identify preliminary business models that could be created around STAMP results.

The process of shaping this deliverable was based on a combination of standard innovation management and business strategy tools such as value proposition, business model, and exploitation and innovation questionnaires prepared ad hoc to understand each partner exploitation intention. The document also includes STAMP business models and ideas for exploitation.

4. Introduction

This deliverable presents the *key exploitable results* (KER) of the STAMP project, plans for exploitation of these results and preliminary business models. The KER are the four main tools developed within STAMP: DSpot, Descartes, CAMP and Botsing. They are complemented by plugins facilitating their usage in software development and deployment pipelines.

A particularity of the STAMP project is to put the results of the project in production. More in details, STAMP tools are integrated into the software engineering processes and platforms managing the main source code base of different partners. This process ensures an immediate validation of the results and a good perception of the added value. Furthermore, these tools are not only integrated as added value to the software engineering practices of the partners, but in some cases, are critical mandatory parts of their continuous integration processes. This integration effort is critical for a proper assessment of the tools (WP5) and it also already ensures effective exploitation of the STAMP results after the end the project.

Moreover, the added value on the corresponding software development processes has been qualified by the validation effort conducted and result obtained in WP5. **Thus, we consider that the Descartes STAMP tool has fully achieved TRL 8 during the project through its successful complete integration into XWiki main software production system.** And we expect that the integration and validation results will bring the other STAMP tools to a similar effective maturity level.

Additionally, other activities are already exploiting and plan to exploit the KER of the project. The efforts by three partners, ATOS, Engineering, and OW2, to develop service offerings based on the STAMP tools represent a significant direction with regard to the exploitation of the project’s results. These services are internal services proposed to other departments of the partners or affiliated members helping and encouraging the developers of these groups to improve their software testing quality and process using the STAMP KER. Finally, STAMP KERs are exploited in academic and industrial courses.

There is evidence that adoption of software is facilitated by open source licenses. We expect that part of the developers that will adopt the STAMP tools through service offerings and courses, will participate in its sustainability through open source contributions. Such contributions would strengthen the sustainability effort committed by the STAMP project members.

The rest of the document is structured into four sections as follows: Section 7 – STAMP Key exploitable results – presents and details the main results of the STAMP project that will be exploited after the end of the project. Section 8 – STAMP license policy – explains the software licenses we chose for these KER, and how these licenses affect exploitation. Section 9 – Roadmap for exploitation – presents the exploitation activities that are conducted upon these KERs as well as ways to ensure their sustainability and mitigate risk issues. And Section 10 – Preliminary business models – presents the preliminary business model that can be constructed upon STAMP results. In the appendix are presented the individual exploitation plans of the partners.

5. References

[1] Online and latest version :
https://gitlab.ow2.org/stamp/h2020/blob/master/deliverables/wp6/d64_exploitation_plan.pdf
[Che2006] Chesbrough Henry, “Open Business Models”, Harvard Business School press, 2006.
[CMS2018] World Quality Report 2018-19, sponsored by CapGemini, Micro Focus, Sogeti

[Dan2007] Dan, MA. The business model of "Software-as-a-Service". (2007). IEEE International Conference on Services Computing SCC 2007: 701-702. http://ink.library.smu.edu.sg/sis_research/572.

[TRL2014] The TRL Scale as a Research & Innovation Policy Tool, EARTO Recommendations, april 2014. https://www.earto.eu/wp-content/uploads/The_TRL_Scale_as_a_R_I_Policy_Tool_-_EARTO_Recommendations_-_Final.pdf

[FS2017] 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

[GSE2017] Geissdoerfer, M., Savaget, P., Evans, S. (2017). "The Cambridge Business Model Innovation Process". Procedia Manufacturing. 8: 262–269. doi:10.1016/j.promfg.2017.02.033. ISSN 2351-9789

[Nie2016] Niedermayr, R., Juergens, E., & Wagner, S. (2016, may). Will my tests tell me if I break this code?. In *Proceedings of the International Workshop on Continuous Software Evolution and Delivery* (pp. 23-29). ACM.

[Par2018] Paraschiv, E. (2018, nov) The State of Java in 2018. *Baeldung*. https://www.baeldung.com/java-in-2018

[Pet2019] Petrović, G., & Ivanković, M. (2018, may). State of mutation testing at google. In *Proceedings of the 40th International Conference on Software Engineering: Software Engineering in Practice* (pp. 163-171). ACM.

[Map2016] Maple S., (206, jul). Java Tools and Technologies Landscape Report 2016. *RebelLabs*. https://jrebel.com/rebellabs/java-tools-and-technologies-landscape-2016/

6. Acronyms

EC	European Commission
KER	Key Exploitable Result
CD	Continuous Delivery
CI	Continuous Integration
DevOps	Development & Operation
HHRR	Human Resources
IDE	Integrated Development Environment
IT	Information Technology
IP	Intellectual Property
PAYGM	Pay as you go model
QA	Quality Assurance
RoI	Return of Investment
SaaS	Software as a Service
UI	User Interface
WP	Work package

7. STAMP Key Exploitable Results (KER)

The STAMP project is articulated around the novel concept of test amplification, which we explore at three DevOps phases: unit testing amplification, configuration testing, and online testing. Our research activities are based on the development of four main tools that constitute the main exploitable results of the project: DSpot, Descartes, CAMP, and Botsing. These tools were designed and constructed following the methodological pillars of the project: continuous test amplification, technology development, and continuous validation. The continuous test amplification pillar ensures scientific and innovation soundness of the tools, technology development ensures their integration into existing software engineering practices and environments, continuous validation ensures their relevance against existing software assets but also challenge and validate their exploitability in real and complex software production settings.

More specifically, STAMP value proposition is to improve the software engineering processes and products developed by:

- increasing test coverage and quality with less manual effort than traditional methods,
- facilitating testing the systems in different configurations, representative of relevant cloud hosting environments,
- supporting DevOps and allowing to release new versions of systems more often, through automation of testing which ensures the quality and reliability of the code is maintained and no regressions are introduced,
- assessing and providing feedback about the quality of the tests,
- helping developers lower the cost of developing tests by generating new tests from existing tests,
- promoting and increasing the adoption of container-based test configurations, increases the coverage of existing test suites,
- helping developers replicating bugs escaped from existing test cases and revealed by execution logs,

These tools are not the only technical results of the STAMP project. Each KER is accompanied by a series of plugins that facilitate their integration in the continuous integration / continuous delivery (CI/CD) pipelines. These additional developments are essential for the industrial exploitation of the STAMP results. Also, a large number of complementary tools were produced (such as a flaky test detection mechanism), that support the core KER with optional features.

The following paragraphs describe the value proposition of the STAMP results and the four KER of the project. Each KER description is accompanied by a characterisation table that details information (e.g. TRL, IP, competition, …) about this result. To establish the technological maturity level (TRL) of a result, we use the recommendations made by the European Association of Research and Technology Organisations **(**EARTO) [TRL2014].

7.1. KER A - DSpot

The DSpot tool takes Java unit test cases as input and aims to automatically synthetize test improvements. Test improvements are modifications that improve the quality of a test case according to a given metric (code coverage or mutation score). In DSpot, improvements are obtained by transforming the input of the tests to trigger new execution paths during test execution or by adding new assertions which strengthen the validity of the unit test.

The automatically generated modifications can be integrated in the test code as modifications of existing tests or as new tests.

Creating new unit tests, while highly expected by test professional and business analysts [CMS2018], is still not a mainstream software engineering activity. Only high-level tests (functional tests) are somewhat now designed with semi-automated tools. Automatic unit test generation is still a niche market. It is addressed only by a few commercial propositions (e.g. Agitar[1], Parasoft JTest[2])

[1] http://www.agitar.com/

[2] https://www.parasoft.com/products/jtest

and research prototypes and these existing solutions have their own limitations compared to STAMP since they do not exploit existing tests written by the developer i.e. they do not amplify existing knowledge.

KER - A - DSpot	
Description of the result	DSpot is a tool that automatically detects and generates missing assertions for JUnit test cases (also known as test amplification) or new JUnit test cases. DSpot can run against test cases related to code changes or even against new test cases provided along with the push operation on GitHub.
Main delivery model	Open source delivery. Releases are available in MavenCentral using a dedicated Maven plugin and source code and Jenkins pipeline script available on Github for free download (digital delivery). The script provides statistics about test coverage.
Innovativeness introduced compared to already existing products/services	DSpot enriches test cases written by developers, assessing and automatically increasing their robustness. It gives additional test coverage not otherwise achieved manually. Allows detecting more bugs.
Unique selling point	Automatic generation of missing test assertions for better test coverage within CI/CD
Product/service market size	A segment in the automated functional testing market (part of software test automation market, which is $34 bn)
Market trends and public acceptance	Increased adoption of DevOps demands higher levels of automation in continuous integration and delivery and QA. This will further increase the need for automated testing. Open source continues to have a strong impact upon the market. The main insight to explore is the time allocated for testing activities.
Product/service positioning	Open source tool for automated test amplification and maximized test coverage for early adopters among experienced software testers and QA professionals
Legal, normative and ethical requirements (standards, authorizations)	N/A
Competitors	Agitar, Parasoft JTest, EvoSuite (academic prototype), Randoop (academic prototype)
Prospects/customers	Software developers; DevOps
Current Technological Readiness Level	*TRL 6* *Technology demonstrated in relevant environment*

Adequateness of consortium staff	Yes
External experts/partners to be involved	*None*
Background IP (type and partner owner)	Java code and scripts developed by INRIA who provides expertise in software testing technology. Source code available on github:https://github.com/STAMP-project/dspot
Foreground IP (type and partner owner)	INRIA, Engineering, Atos, XWiki : development of additional plug-ins/extensions
IP exploitation forms (type and partner owner) e.g. direct industrial use, patenting, technology transfer, license agreement, publications, standards, etc.	Direct usage of Jenkins pipeline script under MIT license. DSpot Eclipse/Maven plugins also available (*different license*)
Partners' involved expectations (what each partner is going to do with the product)	INRIA: Further R&D. troubleshooting for users. Maintenance and upgrades. Engineering: integrating within production infrastructure, develop further extensions/plug-ins as need by adopters TellU: DSpot to automatically generate unit tests, increasing test coverage for our TelluCloud production code base, helping to ensure its quality and speeding up the process of releasing new versions for our customers Use case partners (Activeeon, OW2, Atos, XWiki): usage in internal unit testing routines in software development process. KTH: using background and foreground knowledge and developed assets for teaching in corresponding modules

7.2. KER B - Descartes

The objective of the Descartes tool is to evaluate the quality of Java unit tests cases. The tool fully implements the concept of mutation testing highly regarded by firms like Google [Pet2018]. The general idea of mutation testing is to change (to mutate) part of the code which should be tested and to verify if the evaluated unit test is able to detect these changes, i.e. if the test fails after the changes. More precisely, Descartes operates "extreme mutations" [Nie2016] by performing coarse grain transformations like removing the whole body of a method. If no test is able to detect any extreme mutant of a given method, the method is qualified as "pseudo-tested", and "partially tested" if only some of the mutants are detected. Usage of extreme mutation allows to reduce the computation time required to produce mutations and process them.

Descartes is designed as a mutation engine for PITest[3], a mature and open-source testing tool for Java programs. PITtest uses a mutation engine to discover and create mutants against which it runs the tests and computes results reports (including a mutation score). The objective of Descartes can thus be compared to more classical code coverage approaches and tools that measure which portion of the code is actually executed when the tests are run. Mutation score can be seen as complementary to classical code coverage metrics. In particular, it is a unique technology to assess the quality of test oracles (beyond the quality of test inputs).

KER - B - Descartes	
Description of the result	Descartes evaluates the capability of existing Java test suites to detect bugs using extreme mutation testing. Descartes provides a mutation coverage score of the code, and detailed reports about pseudo or partially tested method and related tests. A Jenkins plugin able to show detailed reports and trend report about Descartes execution in a Jenkins freestyle job.
Main delivery model	Open source delivery. Releases are available in MavenCentral using a dedicated Maven plugin and source code is available on Github (digital distribution). Extension to PiTest software. A Jenkins plugin is available to work along with Jenkins.
Innovativeness introduced compared to already existing products/services	Descartes computes a mutation score faster than comparable tools, and it finds pseudo-tested and partially-tested methods, pin-pointing what to improve to increase test quality.
Unique selling point	Descartes is the only tool which implements extreme mutation and so offers an effective way to assess existing test cases strength and quality by detecting weakly tested pieces.
Product/service market size	A segment in the Application Development Life Cycle Management (ALM) Tools (as Atlassian Clover for instance). Total ALM market size is $2.4 bn
Market trends and public acceptance	Mutation testing is a mode of testing that is not yet largely adopted, and extreme mutation as an innovative approach should come as a next level of it. Increased adoption of DevOps demands higher levels of automation in continuous integration and delivery and QA. This will further increase the need for automated testing. Open source continues to have a strong impact upon the market.
Product/service positioning	An extension to PiTest software for extreme mutation testing to get more actionable and faster mutation analysis results for early adopters: developers and software testers.

[3] http://pitest.org/

Legal, normative and ethical requirements (standards, authorizations)	N/A
Competitors	Default PITest mutation engine, Major
Prospects/customers	Software developers / DevOps / PiTest users / Mutation testing evangelists
Current Technological Readiness Level	*TRL 8* *System complete and qualified*
Adequateness of consortium staff	Yes
External experts/partners to be involved	Pit developer - Henry Coles - has been involved as base mutation testing technology expert. R. Niedermayr [Nie2016] uses and contributes to Descartes and its Maven plugin PitMP.
Background IP (type and partner owner)	Mutation testing technology (open source) developed by Henry Coles for PiTest software. Extreme mutation testing technology developed by INRIA as an extension and completion to mutation testing methods.
Foreground IP (type and partner owner)	INRIA
IP exploitation forms (type and partner owner) e.g. direct industrial use, patenting, technology transfer, license agreement, publications, standards, etc.	*Direct use under MIT License. Extension to PiTest software.*
Partners' involved expectations	INRIA: Further R&D. Troubleshooting for users. Maintenance and upgrades. Engineering: Provide developers with guidelines to adopt Descartes within production infrastructure, *develop further extensions/plug-ins required by adopters* TellU: exploiting Descartes and plan to integrate it in our DevOps process Use case partners: process enhancement: usage in internal unit testing routines in software development process (DevOps processes). XWiki: testing tests to ensure test quality KTH: teaching, tutorials, empirical analyses with other partners
Sources of financing	

foreseen after the end of the project	

7.3. KER C - CAMP

CAMP is a configuration testing tool. The objective of CAMP is to know in which environment a given software can be executed without errors. CAMP automatically generates different system configurations (databases, operating systems, browser, …) around the software and runs the system test suite of the software against the different configurations. CAMP uses a model of system environment expressed as docker and docker-compose files that it amplifies to produce different configurations. Then, the software system tests are executed automatically in the different containers produced and the results are consolidated and presented to the developers.

Albeit automated configuration generation for containerized programs is not yet a common software engineering activity, according to [CMS2018], model-based automated test case design is highly expected by software companies (68% expectation), and large adoption of DevOps (99%) and automated mobile app testing coverage usage (e.g. Appium[4], Espresso[5], XCUITest[6], …) opens the way to such practices.

KER - C - CAMP	
Description of the result	CAMP is a configuration amplification tool based on Docker. CAMP is executed when test configurations/environments change. CAMP takes as input a sample testing configuration and generates automatically a number of diverse configurations. The generation is guided by predefined features and constraints, and utilizes a set of reusable pieces. The current version of CAMP is focused on the Docker environment, and the input and output configurations are specified as Dockerfiles or docker-compose files. It generates new possible configurations based on a basic one in a form of YAML file, then it transforms these new configurations in actual Docker files. CAMP then deploys the amplified Docker containers to run the test suite in different environments.
Main delivery model	Jenkins pipeline script. Open source model. Source code in python available for free use on GitHub. The fastest way to get started with CAMP is to use pre-built Docker image.
Innovativeness introduced compared to already existing products/services	CAMP facilitates micro-service and system testing by the generation of new system configurations, deploying these and testing against them, making it feasible to do more extensive system testing and thereby raising the quality of the system and enabling more frequent

[4] http://appium.io/

[5] https://developer.android.com/training/testing/espresso

[6] https://developer.apple.com/documentation/xctest/user_interface_tests

	releases of new versions.
Unique selling point	Automated container-based testing environment enabler, allowing to test a maximum of possible configurations for exhaustive system testing.
Product/service market size	Software Testing System Integration market, CAGR of 14% by 2019
Market trends and public acceptance	As the number of large projects increases with different teams being involved, complexity of software development increases. Companies tend to increase adoption of agile techniques and DevOps which demands a higher level of automation in unit testing, CI and QA. Open source continues to have a strong impact upon the market. Consulting companies are the main influencer on adopting new products for CI and testing automation. Docker technology in being largely adopted for environment and configuration management.
Product/service positioning	A docker-based configuration amplifier for effective system testing
Legal, normative and ethical requirements (standards, authorizations)	N/A
Competitors	Tox: Python version Testen: NodeJS Chef: decrire configuration Mobile test: e.g. Xamarin CI: Travis (diff conf) wercker IBM, Accenture, Capgemini, Wipro, TCS, Computer Sciences Corporation (CSC), Cigniti Technologies, Gallop Solutions, Infosys, NTT Data, Steria, Tech Machindra, UST Global
Prospects/customers	DevOps / QA professionals / plutôt Ops
Current Technological Readiness Level	*TRL 6* *Technology demonstrated in relevant environment*
Adequateness of consortium staff	Yes
External experts/partners to be involved	None
Background IP (type and partner owner)	Code, scripts and pre-built docker image configuration developed by SINTEF. Python, docker and Java expertise. Source code available on Github: https://github.com/STAMP-project/camp

Foreground IP (type and partner owner)	Core: SINTEF Atos, Engineering
IP exploitation forms (type and partner owner) e.g. direct industrial use, patenting, technology transfer, license agreement, publications, standards, etc.	*Direct use* Under MIT license. Potentially, transfer of the tool ownership to industry partners through the SINTEF Technology Transfer Office
Partners' involved expectations	SINTEF to produce publications in major venues based on tool extensions. Integration of tools in Software Engineering courses at the University of Oslo. Contribution to the open-source community through the GitHub service. Consultancy on the tool usage. Potentially, transfer of the tool ownership to industry partners through the SINTEF Technology Transfer Office. SINTEF for further R&D in order to allow CAMP to produce recommendations for the most relevant configurations for the current system. Engineering: Making it available within Engineering Group production infrastructure, to promote among developers the usage of container-based testing environment, thanks to the automation that CAMP offers generating test configurations and executing test suites against them. Atos is using CAMP to test CityGo application against different system configurations. XWiki uses CAMP to set up XWiki collaborative platform in various environments. TellU: Exploit CAMP to facilitate micro-service and system testing of our TelluCloud system. CAMP will be used to generate new system configurations, deploying these and testing against them, making it feasible to do more extensive system testing and thereby raising the quality of the system and enabling more frequent releases of new versions. Activeeon: use CAMP to enable testing new configurations and increase quality of the architecture. OW2 to promote CAMP to its projects to integrate it in their DevOps processes and CI

7.4. KER D - Botsing

Botsing is a Java crash reproduction tool. Botsing takes as input a stack trace of an exception that caused a software defect, and automatically generates a JUnit test that reproduces the crashing behavior. The objective is two-fold: to allow the developers to better understand the scenario that causes the software defect (debugging) and to obtain a non-regression test case that will fail at first, but which should pass when the bug is corrected. Botsing uses evolutionary search to obtain the test case scenario and relies on EvoSuite, a search based test generator, as an external dependency.

The software engineering activity of debugging using crash information is often categorized as "root cause analysis". Such kind of functionality is often provided as a feature of the broader application and service monitoring software. Major actors such as Oracle (Log Analytics) and more specialized enterprises such as DataDog[7], Dynatrace[8] or OverOps[9] provide such monitoring systems with less or more advanced and automated root cause analysis features (sometimes only retrieving the log or the trace that is relative to the defect, but never generating new unit tests). Please note that competitor companies raised tens of millions of dollars in founding to be competitive in the market (e.g. $49.5M for OverOps or $147.9M for DataDog).[10]

KER - D - Botsing	
Description of the result	Botsing supports developers to speed up the bug fixing process, reproducing automatically an error with the generation of a test case able to replicate it. Botsing (Dutch for "crash") is a Java framework for crash reproduction based on Evosuite. The Botsing framework enables developers to automatically create test cases based on stack traces of runtime exceptions and crashed position in source files.
Main delivery model	Open source model. *Recommended use as Jira plugin*. Botsing is a complete re-implementation of EvoCrash. Source code available on GitHub. Pull requests are welcome. https://GitHub.com/STAMP-project/botsing
Innovativeness introduced compared to already existing products/services	Botsing eases debugging and maintenance via automatic crash reproduction. In practice today, crash reproduction is manually conducted by developers. Based on such reproduction, developers can further understand the root cause of crashes and fix the bug behind crashes. Botsing reproduces **automatically** an error by generating a test case able to replicate it, which speeds up debugging and allows to optimize software.
Unique selling point	Botsing allows developers to easily and quickly find the root cause of an in-production crash to get a clear insight about what causes in-production bugs and **automatically** produce relevant test cases. Botsing is designed to save time spent on debugging and human resources.
Product/service market size	Software Testing Automation market

[7] https://www.datadoghq.com/

[8] https://www.dynatrace.fr/

[9] https://www.overops.com/

[10] https://www.crunchbase.com/organization/overops and https://www.crunchbase.com/organization/datadog

Market trends and public acceptance	*Log Analysis* *Based on metaheuristic search techniques.* This involves defining a search space, usually too large to be explores exhaustively. For Botsing, the search space is the set of all the possible executions of the software. Search-based software engineering does not yet experience wide industry acceptance. Software engineers are reluctant to adopt tools that are unlike those that humans produce. Developers need to be confident that any automatically produced modification does not generate unexpected behaviour outside the scope of a system's requirements and testing environment.
Product/service positioning	Botsing is a framework for automated crash reproduction and test cases creation. Based on EvoSuite technology, it helps developers optimize their software in terms of debugging and maintenance.
Legal, normative and ethical requirements (standards, authorizations)	NA
Competitors	Root Cause Analysis : OverOps Stack trace analysis - STAT CRAY Sapienz
Prospects/customers	Developers / Support Engineers involved in debugging and customer support
Current Technological Readiness Level	*TRL 5* *Technology validated in relevant environment*
Adequateness of consortium staff	Yes
External experts/partners to be involved	None
Background IP (type and partner owner)	EvoSuite technology LGPL used externally Framework created by TUDelft. Jira knowledge
Foreground IP (type and partner owner)	Botsing : TUDelft only Code, documentation and tutorials produced by TUDelft. Use cases and analyses produced by TUDelft with use case providers.
IP exploitation forms (type and partner owner) e.g. direct industrial use, patenting, technology transfer, license agreement, publications,	*Apache 2.0 in order to facilitate adoption in industry and academia.* *Jira plugin for enterprise users*

standards, etc.	
Partners' involved expectations	TUDelft: further R&D. TUDelft will use documentation related to Botsing for teaching, tutorials and empirical analyses with other industry partners. Other use case partners (Atos, XWiki, Activeeon, OW2, TellU) to integrate Botsing in their debugging processes. XWiki: human readable tests

7.5. Integration

To ensure applicability and leverage exploitation of the main STAMP tools in real life software engineering practices, we released for each tool a collection of plugins that allows to integrate them in the most prominent tools used in each step of a Java software engineering. These plugins concerns:

- *Integrated development environment* (IDE), with an eclipse IDE plugin,
- *Build tools*, with plugins for both Maven and Gradle,
- *CI system*, with a Jenkins plugin,
- *Issue tracking,* with a Jira plugin for Botsing,

More details about typical software engineering scenarios and integration of STAMP tools are available in STAMP deliverable 4.3 *"Second public version of API and implementation of services and courseware"*.

8. STAMP License Policy

In this section, we describe the open source Software Licensing policy of the STAMP results, and we clarify licenses compatibility and the impact of this policy on all potential exploitation activities.

8.1. Licenses Compatibility

The STAMP key exploitable results licenses are described in the following table. The different KER have different open sources licenses according the guidelines of the main institution driving their development (Inria for DSpot and Descartes, SINTEF for CAMP, and TUDelft for Botsing). The licenses of the plugins are chosen according their respective ecosystem, i.e. Jenkins, Eclipse and Jira plugin having the same licenses than the software they extend.

Result / Tool	License
DSpot	LGPL v3.0
Descartes	LGPL v3.0
CAMP	MIT
Botsing	Apache 2.0
Maven plugins	*Same license as main tool*

Jenkins plugins	MIT
Eclipse plugins	EPL
Jira plugin	Apache 2.0

Table 1 STAMP KER Licenses

All these licenses allow commercial use, distribution, modification and private use. EPL and LGPL are considered as “weak copyleft” license, meaning that every person is allowed to redistribute or modifying the protected software but the resulting modification or copies are bounds with the same license agreement. They are considered as “weak” as, anyone is authorized to distribute a software linked to the protected software with a different license. MIT and Apache licenses are considered as “permissive” since they allow sub-licensing, i.e. anyone is authorized to redistribute the licensed software with a different license (e.g. a commercial license). Albeit, Apache 2.0 license requires the authors of modifications to document their changes.

Unification of licenses for compatibility reason was envisioned, but is not necessary. Indeed, in the STAMP project, the different key exploitable results share the same objective and value proposition (improving code quality trough test amplification) but are used independently[11] at different steps of the CI/CD process. So, they are not considered as components of the same software, and do not jeopardize license compatibility.

Compatibility of licenses between plugin and STAMP main tools is ensured. First, none of the work released by the STAMP project have a strong copyleft license (e.g. GPL), so no plugin is “contaminated”. Second, plugins are not distributed with tools but have only a dependency to them, so they can propose a more or similarly permissive license (Apache/MIT) than the tool they depend on. The only remaining issue could be between Jira plugin and CAMP, since Apache 2.0 license of Jira being a little more restrictive than the MIT license of CAMP. However, no Jira plugin is envisioned for CAMP, since Jira plugins are suited to acquire crash traces that can be exploited by tools like Botsing.

8.2. License and Exploitation

As stated since the STAMP project proposal, the licenses of the STAMP result are open source licenses. Open licenses facilitate the exploitation of the results inside the STAMP consortium and outside as advocated by the European Commission[12].

Indeed, we notice a strong preference for open licenses in the Java software engineering community, especially when addressing testing, DevOps and CI/CD practices [Par2018]. For instance, the three main tools for building Java software (Maven, Gradle, and Ant) are released under Apache 2.0 License. The main continuous integration server [Map2016], Jenkins is released under a MIT License, and JUnit, the main unit testing framework is released under Eclipse Public License 1.0 (for JUnit 4) or 2.0 (for JUnit 5). In the DevOps area, we can cite the main Java application server (Tomcat, Apache 2.0), version control system (Git, GPL 2.0), containerization environment (Docker, Apache 2.0), and configuration management tools (Ansible, GPL 3.0; Chef, Apache 2.0; and Puppet, Apache 2.0).

[11] Except DSpot and Descartes, that can be used in combination (DSpot can use Descartes as its test quality measurement tool), but anyway share the same licence.

[12] http://ec.europa.eu/research/participants/data/ref/h2020/other/events/2017-03-01/8_result-dissemination-exploitation.pdf

Open source software licenses are very well suited to ensure mass adoption of new technology and they do not prevent commercial exploitation of results by member of the consortium or third party that would be interested into commercialize the results for the STAMP project. Indeed, if we analyse the recent successful stories in the domain of DevOps enabling software (Docker, Ansible, Chef and Puppet) we observe that open source tends to be the norm in emerging software practices like DevOps where new technologies arrive very quickly and professionals expect to, or need to test new tools without barriers. Direct commercial exploitation can be successful with different licenses (including GPL 3.0) and different business models:

- Docker is the massively prominent containerization solution maintained and commercialized by Docker, Inc. Core elements (docker compose, docker community edition) are released under Apache 2.0 license. Docker Inc commercialize a proprietary enterprise edition (Docker EE) that propose additional features (certification, supervision, security scan) and support.
- Ansible is a configuration management tool maintained by RedHat, Inc. The Ansible platform is released under GPL 3.0 License. RedHat, Inc. commercialize the Ansible Tower edition that provide a graphical user interface and different support plans.
- Chef is a configuration management tool maintained and commercialized by the Chef enterprise. Formerly having a dual edition model similar to Ansible and Puppet, the enterprise recently (april 2019) decided to release all the features of their software under Apache 2.0 license[13]. They now sell subscription that provide enterprise distributions, including updates, support and access to expertise on DevOps and Chef products.
- Puppet is a configuration management tool maintained and commercialized by the Puppet company. They provide they core engine under Apache 2.0 license, and commercialize Puppet Enterprise providing additional functionalities (e.g. access control, supervision, orchestration workflows, ...) and support.

Finally, in the case of software testing tools, all kind of open source licenses facilitate usage adoption of the tools by third parties. Indeed, whatever the open source license, such third party will be able to use the software freely, without limitation, and still able to release its developed software without any link or dependency to the considered software testing tools.

9. Roadmap for Exploitation

The consortium provides a plan for the exploitation of the main results for the next six months and after the end of the project. We list actions to be carried out, responsible partners and resources needed as well as the timeline are listed. The forecasted actions for exploitation are assigned per partners, as detailed in the individual exploitation plan presented in the appendix.

In the following, we detail the exploitation activities according to five categories identified as relevant by project partners for the STAMP results:

- direct: continuous exploitation of STAMP tools on the partner own main code base,
- service offering: professional environment built to proposes and empower STAMP tools usage to affiliates, clients, or other partner's departments,
- research: further research activities exploiting STAMP results,
- education: academic courses and professional training focused on STAMP results.

All these activities exploit the STAMP results and ensure their sustainability.

9.1. Direct Exploitation

[13] https://blog.chef.io/2019/04/02/chef-software-announces-the-enterprise-automation-stack/

All partners who provide a use case (Aeon, ATOS, OW2, TellU, XWiki) have already integrated the STAMP tools in the software engineering process in charge of evolving and maintaining their main code base. In some case, the STAMP tools are already an essential part of the software production pipeline of the partner, i.e. a commit or a release mandatory needs to be automatically validated by a STAMP tool. This integration added-value is qualified by the validation result obtained in WP5. In that sense, **the STAMP Descartes KER has achieved TRL 8 since the tool was qualified in the actual XWiki software engineering production system**. Such a maturity level is a successful result for a collaborative research project. We expect that the other integrations efforts will be similarly qualified before the end of the project.

The effort to integrate the STAMP tools in the software engineering process of the use case partner was conducted primarily because, at the current status, they already provide added value to their pipelines, but also for research reasons. Indeed, a qualification process conducted within a software production process is complex: in particular, the result of the evaluation could be masked by the continuous modification on the code base done by all the developers of the respective organisations. Indeed, in a completely controlled testing environment, typical of research projects, the researchers can perform their test on a fixed codebase and observe the results.

Moreover, applying manually such tools on a regular basis is more significant but requires the involvement and skills from the developers. Thus, it was decided starting from the first part of the project, to integrate the STAMP tools in the software engineering process of the use cases partners in order to:

1. ensure a regular usage of this tools,
2. facilitate their usage and adoption for all developers,
3. obtain a qualification of the added value of the tools.

This integration also ensures exploitation of the STAMP tools after the end of the project, and encourages development efforts on the tools by providing continuous testing and feedback. This also strengthens the sustainability of the tools and plugins used and developed by the partners, since these elements are required for an effective execution of the software engineering process.

The direct exploitation activities have different degrees of maturity according the use case partner requirements and practices and the KER considered:

- Activeeon has integrated Descartes and DSpot on its Jenkins platform, is using CAMP and Botsing on a regular basis and intends to integrate Botsing and CAMP on its Jenkins platform and to enforce CAMP execution on their release management and quality assurance process.
- ATOS is regularly using all STAMP tool on the CityGO and SUPERSEDE IF open source code bases, and plan to integrate CAMP tool into software engineering process of their Canopy Compose commercial product.
- TellU has integrated Descartes into its DevOps process and intend to integrate similarly DSpot and CAMP.
- XWiki has integrated Descartes deeply into its CI platform, a build can succeed only if the mutation score returned by Descartes does not decrease compared to the previous build. This process is automatically handled by the Jenkins server for every modification (commit) on any part of XWiki standard (core of XWiki code base, which is more than 600K LoCs). CAMP/TestContainers is also used in production. XWiki plans to integrate similarly DSpot on its CI platform to apply it on every modification and to integrate Botsing for automatic execution on its Jira platform.

More details about the STAMP tool adoption by the different use case partners, and the impact of this exploitation activities can be found in the deliverables 5.5 and 5.6 (Use Cases validation).

9.2. Service Offering

The service offering is an organized effort conducted by some partners to propose STAMP results as a service to software developers who are not in scope the STAMP project. These activities intend to integrate the STAMP results on the software engineering process of different organizations or departments. Partners cannot enforce the adoption of STAMP tools by developers and organizations, but if these service offerings are successful, the final result of this activity will be technically and organizationally similar to the "direct" exploitation scheme result, but at an even bigger scale.

In particular, the partners that adopted this category of exploitation perform the following activities:

- ATOS will incorporate STAMP tools usage into its Testing Factory in Seville and Valladolid. These factories provide testing services to many institutional and private consumers. They employ already many different testing techniques. STAMP KER tools will be used to amplify tests and their execution will automatize in the CI-CD platform used to manage the concerned projects. Specifically, ATOS aims to use the tools CAMP and Botsing to perform quality assurance testing in third-party applications. Also, as ATOS must ensure the best quality for mission critical software to their clients, the CAMP tool will be used in the factory to identify the optimal configuration and maximise the performance of software configuration as defined in the stress tests.
- Engineering is integrating STAMP tools in their main software engineering platform available for all Engineering business units. For instance, Descartes tool is integrated into the Jenkins CI platform of the Engineering Software Laboratories (ESL), to complete the static analysis process already performed by a SonarCube instance with mutation scores. Relevant ESL teams using Jenkins pipelines and having significant unit tests will provide feedbacks before the end of the STAMP project. Similarly, the CAMP tool will be integrated in Engineering CI platform, and Botsing will be integrated into the Jira instance of the Engineering corporate infrastructure. Moreover, Engineering as already integrated STAMP technologies into its business portfolio offered to all its consumer.
- OW2 is integrating Descartes as an online service into the software engineering infrastructure offered to its hosted open source projects. Such an online service can be activated on-demand by the projects owners. The integration of other STAMP tools will be evaluated. Moreover, OW2 uses a maturity assessment framework to publicly evaluate all hosted projects. The maturity level of the hosted projects is computed using a project quality dashboard containing several criteria.[14] Usage of STAMP tools will be added to these criteria thus encouraging OW2 projects to improve their test practices and advocating STAMP KER exploitation. OW2 will provide a support for the usage of these tools to their project community to facilitate adoption.

9.3. Research

STAMP research results and STAMP KER will be exploited and strengthen through further research activities around test amplification that will use STAMP results as a technological basis to new developments. The intensity of research activities is usually influenced by research grants, nevertheless in each research institution partner, several permanent researchers or long term non-permanent researchers are personally engaged in these research topics which are at the core for the corresponding research groups.

The partners that will pursue this category of exploitation are:

- Inria, which will continue research continue the research activities on Descartes and DSpot topics. The work done on these topics and the exchanges with industry have raised new research questions and identified some industry's needs which are in the scope of Descartes

[14] https://tc.ow2.org/view/wiki/Mature%20level

and DSpot topics, e.g. correcting automatically the weaknesses identified by Descartes by applying test amplification on targeted tests, or optimizing tests suites, or customizing the mutation engine according to individual practices. Inria will offer a permanent position to the PhD student involved in STAMP who will lead the research on the Descartes and DSpot topics, including supervising PhD thesis.

- KTH, which will continue research around Descartes and DSpot. Specifically, KTH will work on automatic fixing of pseudo-tested methods for Descartes and for DSpot, on automatic generation of comment that explain the amplification to developers, and a second version of DSpot that can amplify system tests (e.g. cucumber). KTH is hosting three Masters interns to develop new features and experiments about DSpot (choice of amplifiers, generation of messages for amplified tests pull requests) and in the following nine months. KTH will recruit a PhD student funded by the ITEA project Internet of DevOps.
- SINTEF, which will continue research around CAMP, by applying it to a number of open-source projects beyond STAMP, where SINTEF aims at collecting empirical evidence about forward/backward compatibility issues. A few proposals are currently being elaborated at SINTEF where CAMP will be SINTEF's baseline technology.
- TUDelft, which will continue to develop new approaches and new algorithms for search-based software testing. In this context, Botsing will be maintained, improve and extended to experiment with crash reproduction, model seeding, common and uncommon behavior reproduction. This research takes place in a broader context dedicated to the improvement of software testing using artificial intelligence (including search-based) approaches. TUDelft will provide support for Botsing and other STAMP technologies developed by TUDelft during the nine months following the project. Botsing is open source so if the tool remains popular, TUDelft will draw additional person power from the open source community. One PhD student will work and finish his PhD until August 2021. One postdoc is employed by TU Delft to support and further develop STAMP technologies until October 2020.

9.4. Academic Courses, Professional Training and Advocacy

Academic courses and professional training are more relative to dissemination. But thanks to the open source nature of the STAMP tools their adoption by a significant number of trainees will increase the sustainability of the STAMP result.

The partners that propose these activities are:

- Atos which plans to provide training such as webinars on all available STAMP tools to share knowledge on how to run tests and benefits of amplification of test cases. At the same time, Atos aims at incorporating STAMP knowledge as part of their own Innovation methodology for developing software in the Atos Research & innovation department.
- Engineering, which presents all STAMP tools in the “Test Process” course of the *Engineering School of ICT & Management* since the beginning the STAMP project. Every year, in three editions, there are 50-60 attendants which are developers, analysts, solution architects, testers and test managers from Engineering and also external customers.
- Inria, which presents Descartes and DSpot in the Master 2 course on “Validation & Verification”, a module on advanced testing technics. This course targets students from ISTIC at the Master 2 level. Descartes and DSpot fit well in the scope of the course and are example of state-of-the-art tools. Students have the possibility of learning, not just about testing problems, but they get to see actual solutions in practice and how they have been implemented and applied. Moreover, the DiverSE team will continue the dissemination of Descartes by giving talks at industry conferences such as POSS, OW2con, BreizhCamp and Devoxx. and by making specific presentations in organizations: already done with Kereval,

Les pages jaunes, OrangeLabs and EC/DGIT, and to be planned with SNCF and Boursorama.

- KTH, which covers DSpot and Descartes in its “Testing and DevOps” course.[15]
- TUDelft, which presents the different STAMP tools in the bachelor course on “software testing” and a master course on “advanced software testing”. Botsing will be particularly the subject of new course about “search-based software engineering”.

9.5. Sustainability Map

We have established a sustainability map to verify if all STAMP KER results will at least be maintained, and hopefully continue to evolve for the next year after the end of the project.

	DSpot	*Descartes*	*CAMP*	*Botsing*	*Maven plugins*	*Gradle plugins*	*Eclipse plugins*	*Jenkins plugins*	*GitLab plugins*
Activeeon			O			X		O	
ATOS			O	O			X		
Engineering		O			X			X	
KTH	X	X			O				
Inria	X	X			X				
OW2		O							X
SINTEF			X						
Tellu			O		O				
TUDelft				X	X				
XWiki		O	O		O			O	

Table 2 Sustainability Map

This table identifies with an “X” the partners that are engaged into ensuring sustainability for a STAMP KER result element for the next year after the end of the project; and with a “O” the partners that not only use the element in production, but make this element as a **mandatory** part of their software engineering production process (or their research experiment activities), that thus are much willing to help in sustainability efforts.

9.6. Risks

[15] https://GitHub.com/KTH/devops-course

In this section, we present the key exploitation risks we identified and their potential impact on the STAMP project. The risks have been collected thanks to the exploitation questionnaires filled out by all partners. We have identified risks and their probability of happening and worked on reducing the level of impact and expose some mitigation/preventive actions that are already in place.
Nomenclature used for "Degree of importance" and "Probability": lowest = 1, highest = 5.

Risk	Concerned exploitation activities	Degree of importance	Probability	Mitigation & Current status
1. Component ownership	Some participants may be reluctant to participate in open source software after the project ends.	1	2	The core component and tools being mostly maintained by institutions or enterprises that only produce open source
2. Difficulty in market penetration/customer reception and acceptance of technology	The R&D tools have long path until these reach the market/products. The incremental innovation takes time to go to products/ or for time-frame for adoption is longer than desired.	3	2	Continue providing demos, showcasing the results and making as much tests as the project partners are able. Testing STAMP tools onto third parties /clients' mission critical software but in a controlled environment.
3. Poor performance of some KER, affecting the early adopters' involvement	The assets presented must be solid enough for external users.	2	3	Focus on optimization and continuous research.
4. Competition appears with same value proposition	As the software market is changing rapidly newcomers may offer same tools.	3	2	After the initial market analysis, we found no similar tools that offer all options/amplification testing services that STAMP offers. Scientific effort is planned to be continued.

5.Key exploitation expertise leaves	Exploitation of a STAMP tool could stop if the person working with that tool in the exploiting organization leaves.	1	4	Automate as much as possible each exploitation mean. Make sure to train at least one other employee in the use of every tool. Improve documentation and tutorials.
6.Key research and development expertise leaves	Continuous development and improvement of STAMP KER could stop if key expertise leaves.	2	3	Ensure a modular design and improve documentation to make tools development more accessible to future developers.
7. A partner is no longer interested in maintaining a KER	Continuous development and improvement of STAMP KER could stop if a partner do not participate.	3	3	Identify replacing partners and cooperation with the help of the sustainability map.

Table 3 Exploitation Risks

9.7. Conclusion

Thanks to the efforts made by the different use case partners to integrate STAMP in the software engineering processes of their principal code base, and thanks to the development of the necessary plugin to facilitate this integration, the STAMP project has achieved TRL 8 for part of its results (namely Descartes) and TRL 6 for other results as committed in the DoA, six months before the end of the project. This level of maturity makes possible the exploitation of the STAMP results directly after the end of the project.

The following table summarizes the ongoing exploitation activities by each member of the consortium:

	Direct	*SaaS /Services*	*R & D*	*Courses*
Activeeon	X			

ATOS	X	X		X
Engineering		X		X
KTH			X	X
Inria	X		X	X
OW2	X	X		
SINTEF			X	
Tellu	X			
TUDelft			X	X
XWiki	X		X	

Table 4 Exploitation Activities

In addition to resources committed by the partners in maintaining STAMP results, the sustainability of the results is stimulated by the open source approach. Indeed, software testing being a developer activity, anyone that uses STAMP results is a candidate to contribute to this toolset by reporting bug or missing feature, by patching errors, by creating new add-ons or plugin, or even contributing to the main code base.

This "open source effect" will be leveraged by service offerings and training and education courses exploitation activities that promote directly the STAMP results to developers. All the subscribers to these services and attendant of these courses who will use the STAMP tools are potential contributors.

10. Preliminary Business Models

Having analyzed the exploitable results/offering, IPR and exploitation options, we now can examine the potential business models for the STAMP results. To conduct this analysis, we use the Business Model Canvas methodology, a strategic management and lean startup template for developing new business models. In this visual chart, we can describe the STAMP value proposition, infrastructure, customers, and revenues. The visual chart is presented in the figure below, where each block is associated with a question that the block must answer:

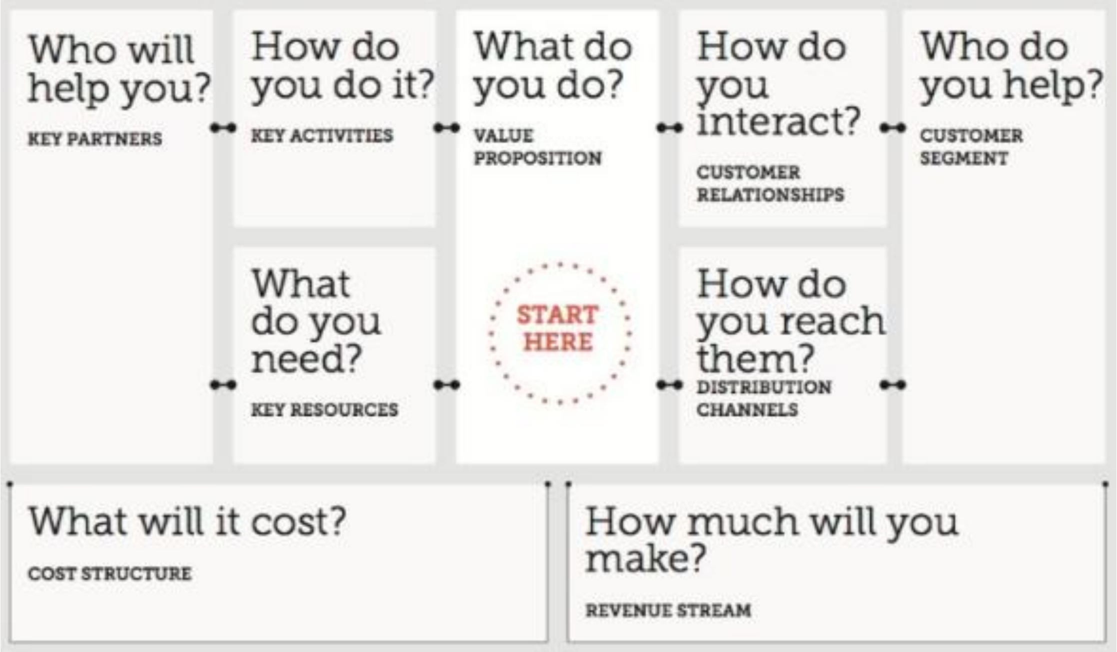


Figure 1 The Business Model Canvas Questions

The canvas shows the different categories:

1. Unique value proposition: the unique value proposition is the differential benefit offered by the STAMP results with respect to other similar solutions in the market. This is the reason why customers are going to be interested in the STAMP solution/services, and not in other solutions.
2. The problem: The problem that STAMP service solves is the reason for being as a “business”, so here we say what problems we are giving based on our solution. Generally, here we add 3 main problems of the target audience related to the field of action.
3. Solution: The solution STAMP gives to these problems is materialized in the offered service. If we properly detect the problems, we can fine-tune the solution we offer. Focus on developing optimally the 3 characteristics of our product or service allow us to give a solution to the problems detected.
4. Customer Segments: includes the problems detected and to whom our service we can provide a solution. Attacking customers is key and those may become early adopters, because they are the most likely to start using our STAMP results.
5. Unique or special advantage: It is what makes one step ahead of our competitors, and that is very difficult or impossible to copy or imitate. It can be an exclusive design, a patent, entry barriers very difficult to penetrate, a specific technology, etc.
6. Channels: Channels are our means of access to customers.
7. Cost structure: Here we must collect everything that will generate expenses in the launch and the business.
8. Revenue flow: In this section, we must define how a business model will make money.

Three main business models have been defined for STAMP project results, considering their innovation and target TRL 6-8, in order to address the targeted markets. At the current project stage (M29), the STAMP partners have agreed on a common approach which will be used to explain how STAMP results will be offered and exploited in an independent manner. The potential business models selected are: knowledge transfer, SaaS offering and open source business models. The following sections describe these models.

10.1. Knowledge Transfer

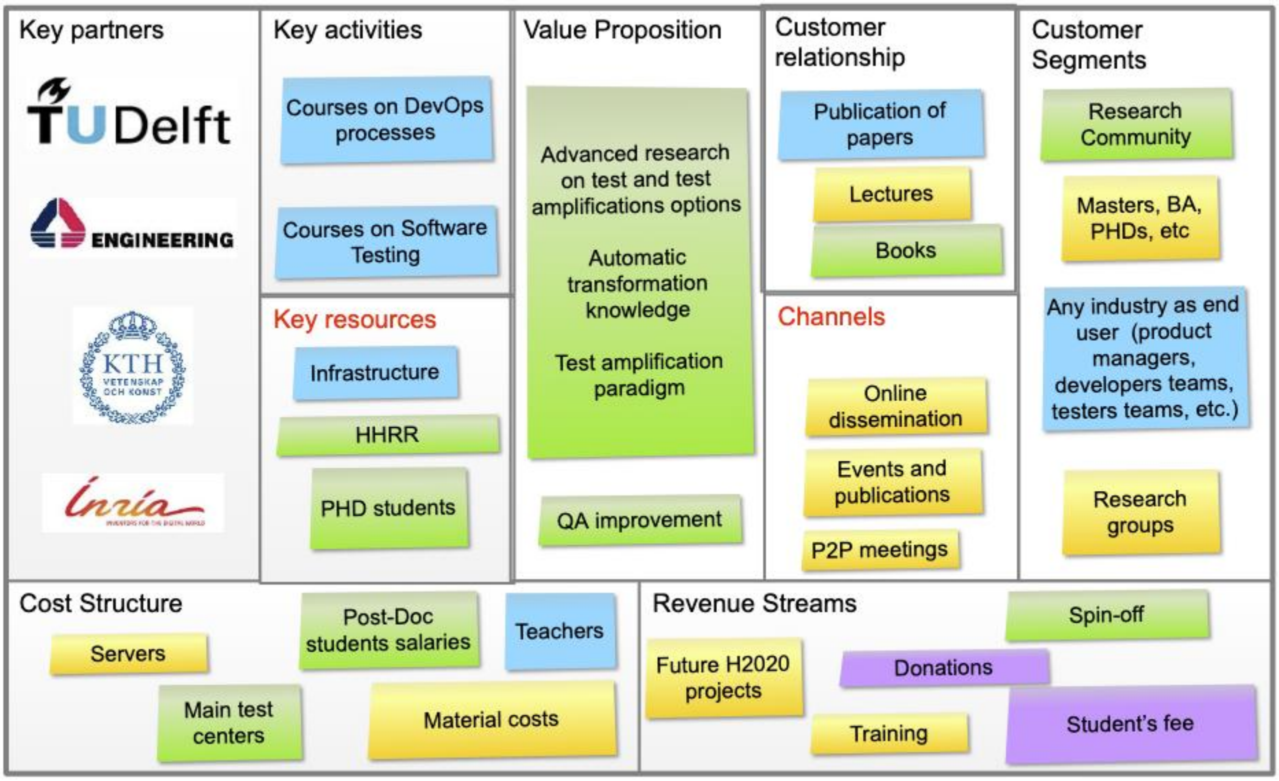
The "knowledge transfer" or training service model means, as its names indicates, running actions like education courses, teaching, and training by charging a fee based on this offering.
The knowledge and tools produced by STAMP as well as advanced research in automatic test generation, automation in DevOps and innovative methods of test amplification will form the content of the courses operated. Different format of courses and ad-hoc trainings, online material and tutorials, will be produced and/or are already produced.
For university partners, the plan is to provide an offering of academic courses in the following topics/areas:

- KTH plans to offer courses based on DevOps and advanced techniques.
- The DiverSE team in Inria/Rennes offers training based on software testing considering the two tools DSpot and Descartes.
- TUDelft aims at offering courses based on advanced software testing (Bosting) and mutation testing (Descartes).

And for the industrial partners:

- Engineering plans to provide courses in their "Engineering Academy". These courses are offered to Engineering's customers.

The STAMP knowledge transfer canvas model is shown in the figure below:



Key partners
TU Delft
ENGINEERING
KTH VETENSKAP OCH KONST
Inria

Key activities
Courses on DevOps processes
Courses on Software Testing

Key resources
Infrastructure
HHRR
PHD students

Value Proposition
Advanced research on test and test amplifications options
Automatic transformation knowledge
Test amplification paradigm
QA improvement

Customer relationship
Publication of papers
Lectures
Books

Channels
Online dissemination
Events and publications
P2P meetings

Customer Segments
Research Community
Masters, BA, PHDs, etc
Any industry as end user (product managers, developers teams, testers teams, etc.)
Research groups

Cost Structure
Servers
Main test centers
Post-Doc students salaries
Material costs
Teachers

Revenue Streams
Future H2020 projects
Donations
Training
Spin-off
Student's fee

Figure 2 Knowledge Transfer Canvas

10.2. STAMP as SaaS

The *Software as a Service* (SaaS) business model means that suppliers deliver on-demand information processing services for users and offer computing utility instead of standalone software. The SaaS model offers connectors and APIs, mostly open source, but the server code is not accessible to the end-user. SaaS vendors offer a bundle of software applications, supported by an IT infrastructure and all necessary services across a network.

When software applications become modulated, open, and standardized, the SaaS business model takes a significant market share [Dan2007]. This is the case in the Java software testing and DevOps domains with such open, modulated, and standardized tools such as JUnit, Jenkins, Git, Docker, etc. upon which the STAMP technologies are built.

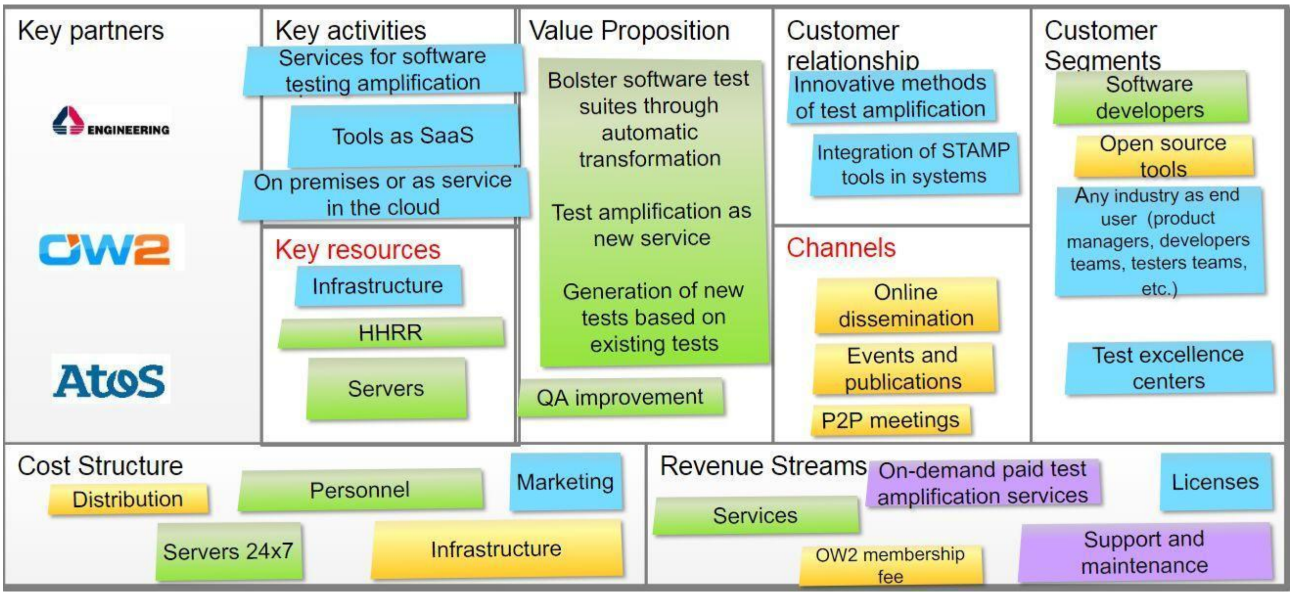
STAMP offers software testing amplification of existing assets such as test cases, API descriptions, dependency models that can be used there to generate more test cases and test configurations each time an application is being updated. Such an iterative usage is well suited for a pay-as-you-go model. At the same time, in the CI scenario we have at least two solutions that make them (limited to Descartes and Botsing) available as SaaS, within the GitHub ecosystem.[16] In addition, with the DSpot, Descartes and Botsing GitHub App, the STAMP consortium demonstrated the feasibility of developing test amplification services available as SaaS in the GitHub ecosystem. This option offers working apps that can also be building blocks for more sophisticated test amplification service solutions.

The option to provide STAMP as SaaS is a good business opportunity that can generate a great value. The high-cost associated to testing software sector are of paramount importance and the fact that STAMP aims to reduce the number and cost of regression bugs at unit level, configuration level and production stage presents a clear advantage in this case because the users can just devote some time to test STAMP services in a pay-as-you-go model. But this business model requires the compromise to maintain the infrastructure and other agreements of usage and services that will be further explored. The SaaS model is a promising business model especially for industrial partners but needs further analysis and in depth considerations to be evaluated.

The most adequate partners to drive such SaaS business model are the service and consulting company (Atos and Engineering) and the OW2 consortium which plans to offers STAMP tools in SaaS mode for its members.[17]An external partner, more specialized, such as Kereval with which Inria is in discussion, could also support such business model.

[16] Currently, only the Botsing GitHub App is actively developed and maintained.

[17] OW2 will not monetize STAMP tools usage directly, but these tools will make the OW2 code base more attractive.



Key partners
ENGINEERING
OW2
Atos

Key activities
Services for software testing amplification
Tools as SaaS
On premises or as service in the cloud

Key resources
Infrastructure
HHRR
Servers

Value Proposition
Bolster software test suites through automatic transformation
Test amplification as new service
Generation of new tests based on existing tests
QA improvement

Customer relationship
Innovative methods of test amplification
Integration of STAMP tools in systems

Channels
Online dissemination
Events and publications
P2P meetings

Customer Segments
Software developers
Open source tools
Any industry as end user (product managers, developers teams, testers teams, etc.)
Test excellence centers

Cost Structure
Distribution
Personnel
Marketing
Servers 24x7
Infrastructure

Revenue Streams
On-demand paid test amplification services
Licenses
Services
OW2 membership fee
Support and maintenance

Figure 3 SaaS Business Model Canvas

10.3. Open Source

There are several open source business models. Per Chesbrough [Che2006], these models include:

- Selling installation, service, and support with the software,
- Versioning the software, with a free version as an entry-level offering and other, more advanced versions as value-added offerings,
- Integrating the software with other parts of the customer's IT infrastructure,
- Providing proprietary complements to open source software (these increase in value as the cost of the open source code falls; one version of this strategy is to create a creative common and then build a proprietary products or services on top of the commons).

In STAMP, the open source model could work by keeping the core components available to everyone and offering premium features with some specific, more sophisticated software components that could have fees. Such a proposition is relevant for innovation oriented products.
Propositions based on integration or service and support are well suited for service and consulting company (Atos and Engineering). They are also of very good added value in the domain of software testing on which several of their customers don't have dedicated specialists or expertise.
Moreover, in relationship with the SaaS business model, a SaaS solution could be articulated around maintenance and scaling, utilizing the partners' know-how in platform management.

The Open Source canvas model is shown below:

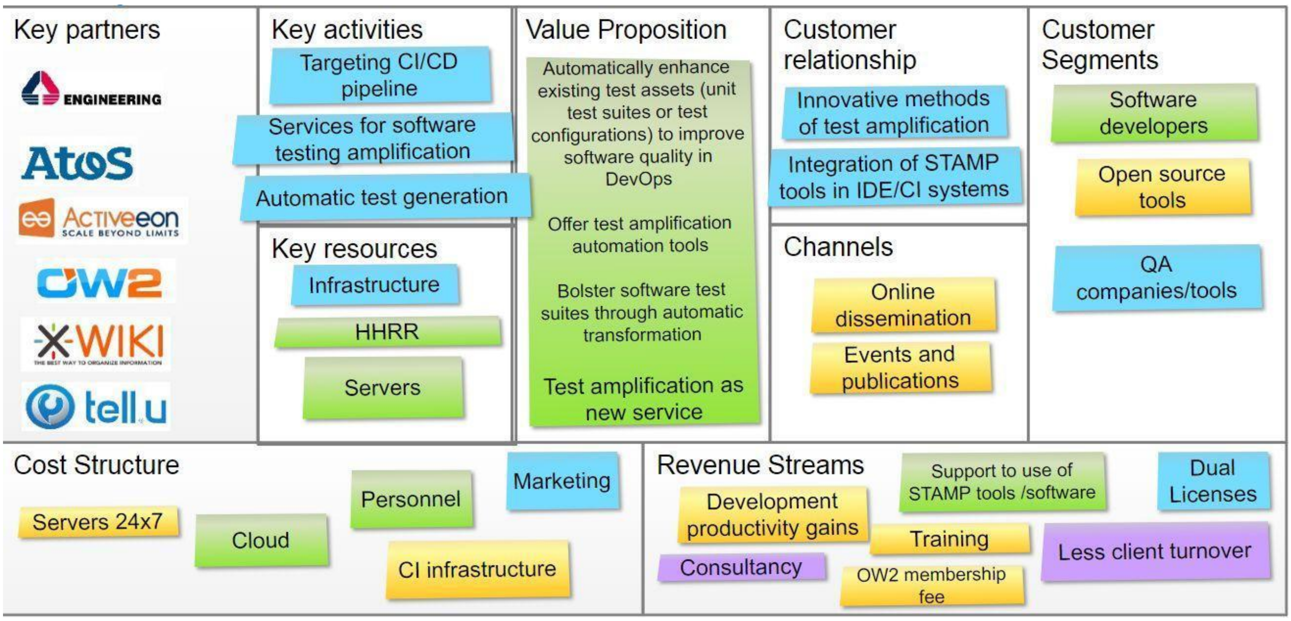

Key partners
ENGINEERING
Atos
Activeeon
OW2
XWIKI
tell.u
Key activities
Targeting CI/CD pipeline
Services for software testing amplification
Automatic test generation
Key resources
Infrastructure
HHRR
Servers
Value Proposition
Automatically enhance existing test assets (unit test suites or test configurations) to improve software quality in DevOps
Offer test amplification automation tools
Bolster software test suites through automatic transformation
Test amplification as new service
Customer relationship
Innovative methods of test amplification
Integration of STAMP tools in IDE/CI systems
Channels
Online dissemination
Events and publications
Customer Segments
Software developers
Open source tools
QA companies/tools
Cost Structure
Servers 24x7
Cloud
Personnel
Marketing
CI infrastructure
Revenue Streams
Development productivity gains
Consultancy
Support to use of STAMP tools /software
Training
OW2 membership fee
Dual Licenses
Less client turnover

Figure 4 Open Source Business Model

10.4. Validation of Business Models

Considering the profile and interests of each STAMP member and to demonstrate the viability of the previously analysed business models a final validation needs to be performed.

To complete the business view/perspective and intentions a questionnaire was provided (See Appendix) to all partners in the context of WP6. The questions aimed at finding the intentions of each partner on issues such as willingness to dedicate resources to the maintenance of software, support actions and promotional work or sustainability aspects. At the same time, in different iterations we have asked to position their intuitions and elect one or more potential business model where they can offer value and provide STAMP-based services.

At the same time, there are more models that could be considered such as "donations," or "Pay-what-you-want (PWYW)"[18] model for generating revenue in different offering. But not sure if donations or this type of revenue present an option, however and truly open source software and its OS business models can be at the heart of a sustainable and profitable business as many companies have successfully survived thanks to the OS business models mentioned before. The exercise needs further analysis in the upcoming months to ensure STAMP sustainability and prosperity.

[18] As defined by Board of innovation, https://www.boardofinnovation.com/guides/50-business-model-examples/

Appendix: Individual Exploitation Plans

In this appendix, we provide a more precise description of individual exploitation roadmaps for STAMP results. Actual and planned exploitation activities are described for each mean of exploitation (see above) in which a partner is engaged, as well as the timeline and sustainability commitment to ensure successful exploitation.

Activeeon

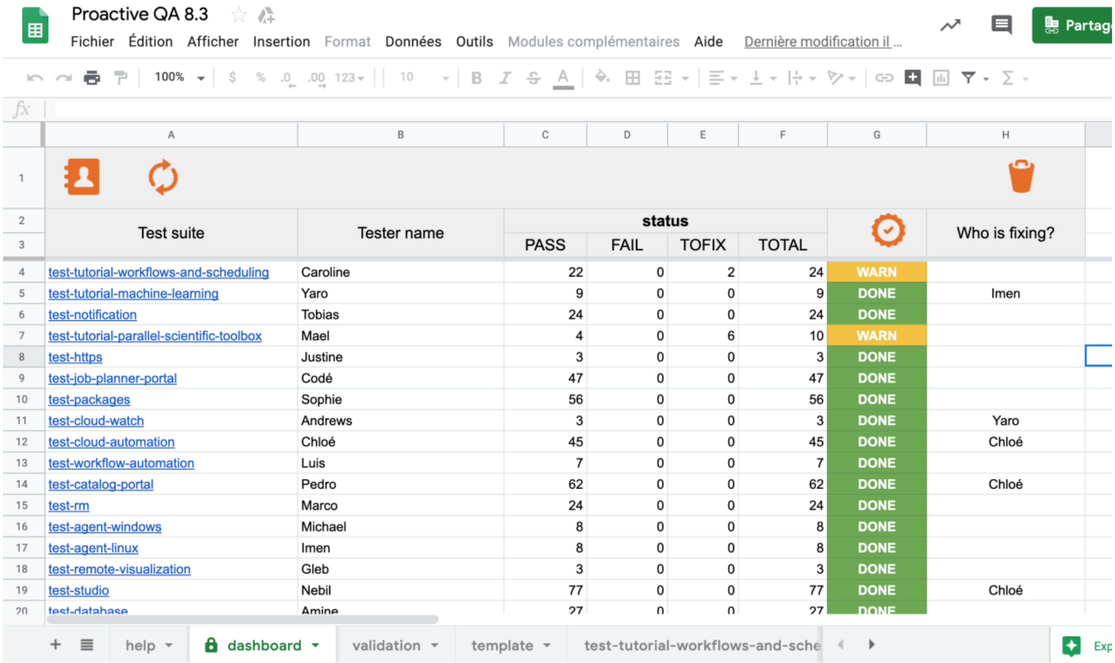
Direct Exploitation

1. Actual exploitation activity

Activeeon has integrated Descartes and DSpot into its CI Jenkins server, allowing to run semi-automatically these tools on relevant subset of its code base. Activeeon refactored the its complex and distributed ProActive Workflow and Scheduling platform in order to allow both Docker containerization to ensure CAMP execution and decentralized logging to improve Botsing usage.
Activeeon is also in charge to develop and maintain the gradle plugins that are useful to execute STAMP tools on its own code base as well as a significative part of Java worldwide development.

2. Planned exploitation activity

Activeeon will integrate CAMP execution in its CI server. Moreover, ensuring that our platform is able to run on different configuration, is important for us and our consumers, so CAMP execution will be integrated into our release management and quality assessment (QA) cycle. Even if the our QA process is not fully automatised it is based on a well-defined procedure to follow that ensure that necessary automatic and manual tests are completed and validated before releasing. See above an instance of an ongoing QA process dashboard.


Proactive QA 8.3
Fichier Édition Afficher Insertion Format Données Outils Modules complémentaires Aide Dernière modification il ...
Partag
100%
help
dashboard
validation
template
test-tutorial-workflows-and-sche
Exp

Test suite	Tester name	PASS	FAIL	TOFIX	TOTAL		Who is fixing?
test-tutorial-workflows-and-scheduling	Caroline	22	0	2	24	WARN	
test-tutorial-machine-learning	Yaro	9	0	0	9	DONE	Imen
test-notification	Tobias	24	0	0	24	DONE	
test-tutorial-parallel-scientific-toolbox	Mael	4	0	6	10	WARN	
test-https	Justine	3	0	0	3	DONE	
test-job-planner-portal	Codé	47	0	0	47	DONE	
test-packages	Sophie	56	0	0	56	DONE	
test-cloud-watch	Andrews	3	0	0	3	DONE	Yaro
test-cloud-automation	Chloé	45	0	0	45	DONE	Chloé
test-workflow-automation	Luis	7	0	0	7	DONE	
test-catalog-portal	Pedro	62	0	0	62	DONE	Chloé
test-rm	Marco	24	0	0	24	DONE	
test-agent-windows	Michael	8	0	0	8	DONE	
test-agent-linux	Imen	8	0	0	8	DONE	
test-remote-visualization	Gleb	3	0	0	3	DONE	
test-studio	Nebil	77	0	0	77	DONE	Chloé
test-database	Amine	27	0	0	27	DONE	

CAMP tests execution will be included in this QA procedure. Other STAMP tools will still be made available for Activeeon developers, for an on-demand execution.

3. Timeline and sustainability

CAMP tests execution integration will be part of the ProActive Workflow and Scheduling quality assessment procedure before the end of the STAMP project. Gradle plugin will be maintained after the end of the project as long as the different STAMP results continue to have a positive impact on Activeeon code quality.

Atos

Direct Exploitation

Actual exploitation activity

In Atos and considering the different areas of the company testing presents many challenges that we want to overcome. We aim to exploit STAMP results in the following areas:

For commercial products/Atos offering

- To improve internal testing processes such as own software factories,
- For providing innovative testing tools for our Testing Factory in Seville, Valladolid
- To provide and investigate, open path to further R&D in testing.

- To provide consultancy services for other Atos units involved in testing tasks. And to improve third clients testing processes
- Canopy Compose (our cloud offering)

For R&D

- To present the assets in the innovation board for technology transference, to promote R&D results to the market.
- For our own internal Atos Research methodology and approaches to test amplification
- To use outcomes of the STAMP for enriching the Software Development Methodology for Atos Research Unit (the testing chapter).

Own tools/developments

- We have developed an Eclipse plug-in
- Improvement the Dspot performance
- Investigate R&D in testing.

Research Results & Innovation Management

In the R&D department, Atos Research & Innovation, there's an internal initiative to standardize all the process for software development and delivery. In this context, Atos Research & Innovation has a methodology for software development with certain guidelines and rules on how to develop software -to design, integrate, develop and store-, it aims to incorporate testing methodology (based on STAMP) in this methodology for their own software development. In terms of quality assurance, there's a group now in ARI that is dealing with QA (Quality Assurance) and it is expected to reinforce the QA process thanks to STAMP outcomes/guidelines on testing. Thus, we aim to use the testing knowledge to serve the part about "How to make tests" and amplification within the "Methodology for development in the Atos Research Unit". The STAMP outcomes will serve the common methodology to develop, store and share software in ARI to foster adopting DevOps philosophy within ARI.

In addition, the Atos Research and Innovation Group has its own methodology for innovation management. The "Innovation board" is an internal decision-making group composed by a mix of technical and business-oriented profiles, to ensure exploitation of R&D are transferred to the market, and the developed STAMP services will be presented to this board. Also, the testing specifications and knowledge to automatize certain tasks thanks to the use of tools will serve to the common development methodology developed in ARI, which is the next step towards adopting DevOps philosophy within ARI. The outcomes of STAMP will be used as well as for serving the testing knowledge to the methodology Atos has for easing the gap between the research and innovation. Thanks to a methodology for incubating assets and present ATOS assets to the innovation board (the Innovation management process defined in ARI with own methodology).

Timeline and sustainability:

The exploitable assets will be presented to the innovation board. We plan to submit our asset during Q3 2019 and even beyond the project finishes. To maintain those STAMP tools where there's co-ownership, Atos will look for agreement with partners/components owners.

We plan to incorporate the STAMP knowledge in the software methodology during Q3 2019.

We plan to contact other company units to see how the results may be reused in our portfolio from Q1 2019 onwards, and continue making demos, showcasing STAMP amplification tools in the context of the project until Q4 2019.

Service Offering

Planned exploitation activity

We plan to use STAMP testing amplification tools to enrich the following Atos products:

- **Canopy Compose (Atos cloud offering)**

Canopy Compose offering contains a catalog of rapidly deployable, standardized application infrastructure environments called blueprints that simplify and standardize application deployments to various target cloud platforms, both public and private. Our ambition is to enhance it through DevOps tools and amplification testing methodologies developed in STAMP.

Canopy provides end-to-end services to a global client base, transforming the infrastructure and applications that power digital business and delivering improved business agility and faster time to market. STAMP results are fully aligned with the strategy defined for Canopy Compose so that we might include STAMP outcomes/plug-ins and tools as part of its offer.

- **STAMP CAMP tool**

The cloud offer in Atos is mostly focused to adapt software and for delivery, not for development of software. So, we would like to test if CAMP tool is suitable for the part of the configuration management and to be used in the optimization of the delivery. In the process of validation of an application in the moment where the deployment is made, the tool would allow to configure according to certain tests and check the functional/sanity checks and performance tests. For example, we want to try CAMP, which can be used for the different configurations in the Canopy suite. in this way, we can get the correct configuration, or the optimal configuration based on the requirement or services that the software needs. Also, we can simulate peaks of use, or saturation for example to satisfy an Nº of users demands, or workloads etc., and CAMP may help us to automatize this process resulting in faster and cheaper delivery of our cloud services.

Timeline and committed resources

We plan to contact canopy sales managers' units to see how the results may be reused in our portfolio from Q1 2019 onwards.

Transfer

Planned exploitation activities

- For providing innovative testing tools for our Testing Factory in Seville and Valladolid in Q2 and Q3 2019.
- For our own internal methodology and approaches to test amplification.
- To provide consultancy services for other Atos units involved in testing tasks
- To investigate and of course open path to further R&D in testing.

ATOS Software Testing Factory

Therefore, for each we plan:

- Atos has a well-known and appreciated offering in testing service developed from a dedicated ATOS Software Testing Factory located in the south of Spain, in Seville. The factory provides testing services to many customers in public administrations and in the private sector, employing different teams and provides all kinds of testing techniques. STAMP will be of great value for this factory by allowing them to facilitate and amplify as well as to automate some of the testing tasks the Devops team performs, and for automation of the deployment CI-CD. The best practices for testing will be used for the ARI common software development methodology developed in ARI. The testing is of paramount importance and will serve for adopting the testing best practices into it.
- In addition, we aim to us the tools CAMP y BOTSING to perform quality assurance testing in third-party applications. The tools would allow Atos to check the assurance analysis and to improve performance and quality. In fact, the Seville factory will test STAMP tools during Q2, and it is foreseen to use these tools to test a software of an Atos client -(a company that belongs to the transportation sector and part of the Ibex-35).
- Also, as we must ensure the best quality for that mission critical software to our clients, the CAMP tool may be used in the factory to test applications. So, to get the optimal configuration and maximise the performance of software configuration as defined in the stress tests.

Timeline and committed resources

- We have already being in contact with the leaders of the Atos Seville factory, in fact they have attended the STAMP workshop organized in Madrid Dec 2017. This year, have done a demo with all the results during May 2019 and we foresee that during July 2019 the tools are used in an Atos client. All activities take place during Q2-Q4 2019.

Further Research

Planned exploitation activity

Future projects, proposal and CITYGO

We plan to include STAMP results in future research, all related to test amplification. Also, about the use case CityGO, many commercial actions have taken place to further invest in the asset. The STAMP tools ensure that the app is robust enough to continue with improvements and even add more complexity to the predictive part of the application.

Timeline and resources:

We already included STAMP research findings, knowledge, etc, and CityGO in different proposal in 2018, and we plan to continue market actions from Q1 2019 onwards.

Academic Courses and Professional Training

Planned exploitation activity

Atos does not provide training as such, however we'll focus on innovation management/technology transference. This may be considered as a professional training activity that we want to (naturally perform in the context of knowledge transference).

Timeline and resources

We plan to present the asset in Q3 2019 and continue R&D research/training beyond the project lifetime in 2020-2024.

Engineering

Direct Exploitation

Actual exploitation activity

STAMP's tools have been described and proposed to the Engineering's infrastructure team. Since Descartes evaluates the *mutation score*, which is a relevant quality indicator for existing test suites, Engineering's Infrastructure team included Descartes in the set of solutions used to enforce the quality of produced software.
Specifically, several Java projects using Maven as build system, have been selected to introduce the mutation score has a quality indicator, along with SonarQube profiles and quality gates, test coverage, etc. This activity aims at introducing Descartes in a consolidated process of ALM analysis adopted by Engineering Software Laboratories (ESL) and focused on a Jenkins CI server. It includes also static analysis performed by a SonarQube instance, the execution of unit, integration and system tests, deployment on target systems (in form of virtual machines or Docker containers).

Currently the team managing the infrastructure is configuring several freestyle jobs and pipelines (not all Engineering production projects are using the more recent concept of Jenkins pipelines) and a first set of results is expected before summer holidays.

Planned exploitation activity

- Collection of the results from Descartes execution
- Providing feedback to developers, helping them to analyze Descartes reports available within Jenkins dashboard
- Supporting developers to increase mutation coverage in their projects
- Introduction in selected projects DSpot to increase mutation coverage
- Proposing the usage of CAMP to projects which started to adopts container technology within their solutions
- A Jira instance is available within Engineering corporate infrastructure: the adoption of a Botsing will be proposed to the team who manages the infrastructure

Timeline and sustainability

STAMP's project team will cooperate with Engineering's business units (in particular Infrastructure and ESL) to perform the described activities. A first set of results is expected by July and will concern the introduction of Descartes in the ALM processes.
The introduction of DSpot is planned by October and the first results expected by the end of 2019.
In January CAMP will be proposed for docker-based projects and the results are expected a couple of months later.
In parallel Botsing will be proposed to the Infrastructure team.

Service Offering

Actual exploitation activity

STAMP technology has been described since the beginning of the projects in internal innovation channel (at Engineering Group several channels are available: two web magazines, an Office 365-based communication infrastructure – Yammer, Teams, etc - and several other channels). As soon as the first results of the research activities have been made available, the Tender Engineering organizational unit decided to propose STAMP technology as a qualifying asset among its professional test services.

Planned exploitation activity

Keep on proposing STAMP technology to Engineering customers, providing them with consultancy, support on adopting STAMP tools in their CI processes, having the implementation within Engineering corporate infrastructure as a reference implementation.

Timeline and committed resources

As a qualifying asset, STAMP will be proposed whenever it will be considered useful. Engineering's STAMP team will be available for internal consulting to provide internal and external customers with requested information on the technology. At this stage no time-resource limits can be established: in particular the resources will be provided by the project/customer which will require STAMP technology.

Academic Courses and Professional Training

Actual exploitation activity

Currently, STAMP tools usage is promoted, since first year of the project (when DSpot and Descartes tools were already available and usable) at the Engineering School of ICT & Management within the "Test Process" course. During these years also CAMP and Botsing were presented during the course lessons. Attendants for "Test process" course are developers, analysts, solution architects, testers and test managers. On average, every year almost 50~60 attendants, coming from all the organizational units, participate to this course, distributed in three editions. In the last two years the offering of Engineering School of ICT & Management was extended also to external customers. This course is considered critical for the curriculum of every software specialist.

Planned exploitation activity

- Expand the CAMP adoption and exploitation within Docker-based projects
- Keep on promoting usage of Descartes as it has been made officially available within the production infrastructure

Timeline and resources

This course is available in the catalog of Engineering ICT & Management school since more than 10 years, and it is one of the bigger classes. For the current year 3 editions are already planned (the first will be in June, the last one will be in October), and it is expected that the same will happen next year.

Inria

Direct Exploitation

Actual exploitation activity

- Research activities
  - DSpot is an essential contribution of the PhD thesis of Benjamin Danglot
  - Descartes is an essential contribution of the PhD thesis of Oscar Luis Vera Pérez
  - PhD thesis in collaboration with Kereval (CIFRE)
- Teaching

    - Descartes and DSpot are used in our Master 2 course on “Validation & Verification”
- Industry talks (POSS, EclipseCon, BreizhCamp, Devoxx) and meetups (Kereval, les pages jaunes, Orange Labs, EC DGIT). The tools have been also presented in sessions of the Paris Java User Group and Breizh Java User Group
- The tools have been in scientific experiments targeting datasets of industry-level open source projects.
- Scientific talks and papers:
    - Danglot, B., Vera-Pérez, O. L., Baudry, B., & Monperrus, M. (2018*). Automatic Test Improvement with DSpot: a Study with Ten Mature Open-Source Projects*. DOI: 10.1007/s10664-019-09692-y. Empirical Software Engineering, Springer Verlag, 2019. Also presented in the Journal-First Track at the 41st ACM/IEEE International Conference on Software Engineering, ICSE 2019
    - Vera-Pérez, O. L., Monperrus, M., & Baudry, B. (2018). *Descartes: a PITest engine to detect pseudo-tested methods-tool demonstration.* DOI: 10.1145/3238147.3240474. Presented at the ACM/IEEE International Conference on Automated Software Engineering.
    - Vera-Pérez, O. L., Danglot, B., Monperrus, M., & Baudry, B. (2018). *A comprehensive study of pseudo-tested methods.* Empirical Software Engineering, 1-31. DOI: 10.1007/s10664-018-9653-2. Also presented in the Journal-First Track at the 41st ACM/IEEE International Conference on Software Engineering, ICSE 2019

Planned exploitation activity

- Research activities
    - DiverSE Team will continue research and development of Descartes and Software Testing Amplification tools
- Teaching
    - Still using Descartes and DSpot in our Master 2 course on “Validation & Verification”
- Integrate Descartes in INRIA development platforms (GitHub/GitLab and Jenkins) to be used by other INRIA projects

Timeline and sustainability

- Oscar Luis Vera-Pérez will have a permanent position in the DiverSE Team. He is in charge of the Master 2 course “Validation & Verification”, and of the research and development on Descartes and DSpot topics.

Transfer

- Continue the dissemination of Descartes in industry by giving talks in industry conferences (POSS, BreizhCamp, Devoxx) and by making specific presentations in organizations. Already done with Kereval, Les pages jaunes, OrangeLabs and EC/DGIT, and to be planned with SNCF and Boursorama.

- Develop collaborations (Thesis, collaborative projects) with companies, already done with Kereval with a joint PhD thesis (CIFRE) started in February 2019
- Using Descartes and DSpot in the course "Validation & Verification"

Further Research

Planned exploitation activity

INRIA will continue the research activities on Descartes and DSpot topics. The work done on these topics and the exchanges with industry have raised new research questions and identified some industry's needs which are in the scope of Descartes and DSpot topics, e.g. correcting automatically the weaknesses identified by Descartes by applying test amplification on targeted tests, or optimizing tests suites, or customizing the mutation engine according to individual practices.

Academic Courses and Professional Training

Actual exploitation activity

Descartes and DSpot are already used in our Master 2 course on "Validation & Verification", a module on advanced testing technics. This course targets students from ISTIC at the Master 2 level. The course introduces the students to advanced testing practices and techniques, from fuzzing, mutation testing, automated test generation and even their use on typical development workflows. Descartes and DSpot fit well in the scope of the course and are example of state-of-the-art tools. Students have the possibility of learning, not just about testing problems, but they get to see actual solutions in practice and how they have been implemented and applied.

Planned exploitation activity

Descartes and DSpot will still be used in our Master 2 course on "Validation & Verification"

KTH

Further Research

Actual exploitation activity

- Higher education and teaching
  - DSpot is an essential contribution of the PhD thesis of Benjamin Danglot
  - Descartes is an essential contribution of the PhD thesis of Oscar Luis Vera Perez
  - We have used Descartes and DSpot in our course on testing and DevOps.
- Industry talks and meetups
- Scientific experiments

Planned exploitation activity

- Teaching
  - Descartes and DSpot
  - We use these tools in our automatic testing and DevOps course
  - Both tools are relevant because
    - they operate on Java code, the most popular language for backend microservices,
    - they can be integrated in automatic build tool chain, an essential aspect of modern software engineering that needs to be taught
    - they are the frontline of automatic testing
- Research projects
  - There is a strong interest from our industry partners because they address an important need (test assessment and generation)
  - KTH will continue the research activities related to Descartes and DSpot

For Descartes, we have 2 objectives: automatic fixing of pseudo-tested methods; integration in the CI workflow

For DSpot, we have three objectives: automatically generate comments that explain the amplification to developers; integration in the CI to amplify in the scope of the change; build a second version of DSpot that can amplify system tests (e.g. cucumber).

These works are relevant for our research agenda that aims at increasing the resilience of Internet-based software applications.

Timeline and sustainability

In the last 6 months of the project KTH has recruited 3 Masters interns to develop new features and experiments about DSpot (choice of amplifiers, generation of messages for amplified tests pull requests)

the following 9 months, KTH continues research and development of DSpot with the recruitment of a PhD student funded by the ITEA project Internet of DevOps.

OW2

Direct Exploitation

Actual exploitation activity

OW2 through its use-case already exploits STAMP tools in some of its community projects. Descartes and DSpot are the main STAMP results that OW2 projects use. Descartes is important for the OW2 quality assessment model. The mutation score provided by this tool is seen as an indicator of the project quality.

Planned exploitation activity

OW2 will exploit the STAMP tools for all its community projects and not only those cited in the use-case. The mutation score will be provided in the project quality dashboard.

Timeline and sustainability

OW2 hired an engineer whose role is to integrate STAMP tools within the OW2 infrastructure. We have already started the development of the glues between the tools and the OW2 infrastructure. As soon as the project ends, the services will be available for the OW2 projects. They will be maintained by OW2. The service will be deployed with additional glues if necessary. Developing the extra glues will be mainly held during the first year of deployment which means up to 12 months after the project ends.

Service Offering

Actual exploitation activity

The actual service is only for the projects that are selected in the scope of the use-case. The STAMP tool is Descartes. There is about 186K lines of code of the four tested projects. The integration of the STAMP tools in the OW2 infrastructure will help our community to evaluate the quality of their project. Moreover, the quality self-assessment form will use STAMP to validate test indicators and to give quality scores about the project.

Planned exploitation activity

The service will be running on the OW2 infrastructure. It will be offered free of charge to OW2 projects. Online services, that can be activated on-demand by hosted project owners. Just like existing ci/cd tools, including Jenkins or Travis. The service is provided to the OW2 community to give a quality and maturity indicator of the OW2 projects. The projects hosted at OW2 that want to obtain higher quality score, they can use STAMP tools which means they will be able to integrate these tools into their development environment. OW2 will provide a support for the usage of these tools to their project community.

Timeline and committed resources

OW2 has already 2 software engineer and system engineer. They will contribute to the support given by OW2 around STAMP tools. The last 6 months of the project OW2 will develop the necessary tools to be able to provide the STAMP tools as a service for their community projects. After the project ends, the main activity will be the support around these tools and it lasts at least 3 years.

SINTEF

Direct Exploitation

Actual exploitation activity

The main exploitation activity of SINTEF is focused around CAMP, developed in WP2. This artefact complements a number of software tools developed at SINTEF. CAMP enables empirical research on fundamental software engineering issues including testing, coupling, and resiliency. CAMP is particularly relevant for SINTEF, as the number of projects around micro-services and docker-based technologies has been growing the past few years.

Planned exploitation activity

Most of the planned exploitation activity will be focused on consolidating CAMP, through the STAMP use cases, but also by applying it to a number of open-source projects beyond STAMP, where we aim at collecting empirical evidence about forward/backward compatibility issues. A few proposals are currently being elaborated at SINTEF where CAMP will be SINTEF's baseline technology.
As part of a recently-started strategic internal project aiming at improving the quality of the software tools being developed, SINTEF will use the advanced knowledge and know-how acquired during STAMP on DevOps and testing in particular.

Timeline and sustainability

Most of SINTEF's effort until the end of the project will be put on CAMP, to improve and consolidate it together with STAMP's use case partners. In addition, we will make sure CAMP can generate useful insight beyond the STAMP use cases, by applying it to a number of popular open-source projects and seek to submit pull requests based on results generated by CAMP on those projects. After the end of STAMP, we plan to continue using and developing CAMP in sub-sequent projects.

Transfer

Planned exploitation activity

Nowadays, SINTEF does not foresee the commercialisation of CAMP, which is available under the MIT license. SINTEF's strategy aims at a self-sustained open-source community taking over the development and maintenance in the long run.

However, if the CAMP tool encounters a particularly wide and successful adoption, a startup could be set up in the context of SINTEF-TTO.

Further Research

Planned exploitation activity

We plan to extend STAMP results, and CAMP in particular, beyond Java projects and apply them to more heterogeneous microservices systems. In particular, we plan to exploit CAMP to collect empirical evidence about forward/backward compatibility in open-source projects that use semantic versioning. This may help avoid unexpected bugs due to maintenance operations such as dependency upgrade for instance.

Timeline and resources

We are currently working on several proposals related to customization and configuration testing of microservices system where CAMP is our baseline input to the projects.

Academic Courses and Professional Training

Planned exploitation activity

We foresee that the knowledge and knowhow acquired on DevOps and microservices is extremely relevant for the Norwegian industry, which SINTEF will provide them through consulting activities or through joint participation into research projects.

TUDelft

Further Research

Planned exploitation activity

TU Delft will continue to develop new approaches and new algorithms for search-based software testing. In this context, Botsing will be maintained, improve and extended to experiment with crash reproduction, model seeding, common and uncommon behavior reproduction. This research takes place in a broader context dedicated to the improvement of software testing using artificial intelligence (including search-based) approaches.

Timeline and resources

TU Delft will provide support for Botsing and other STAMP technologies developed by TU Delft during the 9 months following the project. Botsing is open source so if the tool remains popular, we will draw additional person power from the open source community. One PhD. student will work and finish his PhD. until August 2021. One postdoc is employed by TU Delft to support and further develop STAMP technologies until October 2020.

Academic Courses and Professional Training

Actual exploitation activity

It has been covered in a bachelor course on software testing, a master course on advanced software testing.

Planned exploitation activity

It will feature in a new course on search-based software engineering.

Tellu

Direct Exploitation

Actual exploitation activity

Tellu develops and operates a 24/7 cloud IoT-platform. It is used in different domains, but a current focus is on e-health. It is used by health service providers and large enterprises. It provides data gathering, storage and processing, with configurable business logic. Exchange of data with third-party systems, with conversions to other formats, is also important.

Tellu is using Descartes, both when developing new unit tests and to compute mutation score of the test suites. Descartes provides valuable insights into the quality of our unit tests. It finds pseudo-tested and partially-tested methods, pin-pointing what to improve to increase test quality. If a developer writes a test which invokes a method, but fails to catch errors in this method, this can be a serious omission in the test, causing false confidence in the tests. Descartes finds these omissions in an efficient way. We therefore use Descartes as part of our DevOps process, having the test developer manually run the tool to check new tests and fix issues right away. We also use Descartes to compute mutation scores for whole test suites, as it does this faster than comparable tools. Descartes is used to ensure the test quality of all Tellu Java code, which is around 100,000 lines. It is used when writing a new test. Currently we are only able to write a few new unit tests per month, but the ambition is to do much more comprehensive testing, and Descartes is an important tool in building high-quality test suites.

Planned exploitation activity

Tellu's plan and ambition is to move towards DevOps, with continuous integration and testing, and frequent releases of the micro-services of TelluCloud. We depend on strong automation of testing both on the unit, service and system levels, to ensures the quality and reliability of the code is maintained and no regressions are introduced. We plan to operationalize more of the tools and technologies of STAMP as they become mature enough to be used in our DevOps process.

Specifically, we plan to exploit DSpot to automatically generate unit tests, increasing test coverage for our TelluCloud production code base. This will help to ensure its quality and speeding up the process of releasing new versions for our customers. We plan to integrate DSpot into the build process with a Jenkins plugin. DSpot will be used on the Java code of TelluCloud, which is around 100,000 lines.

Tellu is especially interested in configurability test amplification, with its aspects of managing diverse configurations and deploying a complex system in various configurations. For deployment and test, it is important to be able to deploy both micro-services and full systems in a quick and easy manner. We plan to exploit CAMP to facilitate micro-service and system testing of our TelluCloud system. CAMP will be used to generate new system configurations, deploying these and testing against them, making it feasible to do more extensive system testing and thereby raising the quality of the system and enabling more frequent releases of new versions. This will be used to test the whole system as changes are made, such as running system tests once per day.

Timeline and sustainability

For the last 6 months of the project we continue to test and validate CAMP tools, and to work with the tool developers to adapt tools to the Tellu usecase. The committed resources in person months are estimated below, and based on involvement in the STAMP work packages:

- DSpot (2 PM): Testing and validation on Tellu use case, interaction with developer to resolve issues, integration in Tellu DevOps process.
- CAMP (2 PM): Working with tool developer SINTEF to adopt CAMP to work with Kubernetes files and deploying TelluCloud with Kubernetes. Finding suitable configuration parameters for amplification and testing. Test and validation of CAMP for use in Tellu toolchain.
- Botsing/EvoSuite (2.5 PM): Work with EvoSuite behavioral model to evaluate its usefulness on the TelluCloud case.

Further work after the end of the project will consist of any remaining adaptation of CAMP tools for use in Tellu's toolchain, deploying and maintaining the tools, and internal training in their use. It is still too early to give a detailed roadmap for this activity, but we anticipate a total of 4 PM over the first nine months.

XWiki

Direct Exploitation

Actual exploitation activity

Currently, the following STAMP results are used:

- Descartes is used in production on the development of the XWiki software. It's used inside the XWiki Maven build and whenever a developer commits some code that lowers a module's mutation score, the build fails. Verified automatically in XWiki's CI, at each commit.
- CAMP/TestContainers is also used in production on the development of the XWiki software to execute the functional tests under various configurations, and discover configuration-related issues.
- Flaky test detection is also used In production and included in XWiki's Jenkins pipelines.
- Global coverage computation and comparison with previous reports is also used in production. Whenever the global TPC is lowered, XWiki's CI fails and developers are warned and need to increase it again before a version of XWiki can be released.

These are relevant to XWiki SAS since they all increase the quality of the XWiki software. They are all the more important that they are used on the full XWiki codebase, which is sizeable (480K lines of code, excluding comments and blank lines).

They are also interesting when recruiting developers since it shows XWiki's participation and state-of-the-art in software testing and thus helps attract talents.

Planned exploitation activity

- Try to use DSpot at the CI level: whenever a commit is pushed, execute DSpot in the CI to try to generate new tests for the pushed tests. This is because executing DSpot on the full code base would not be possible (too slow on such a scale).
- Try to use Botsing in the Jira issue tracker: whenever a Jira issue containing a Java stack trace is created, automatically execute Botsing on it to try to generate a test to reproduce that stack trace and if successful, attach that test to the Jira issue, in a comment. The goal is to help developers fix such issues by having a test to start with their investigation, and thus be more productive.

Timeline and sustainability

XWiki will maintain the CAMP/TestContainers framework on the long run since XWiki has converted its functional tests to use this framework. It'll be maintained at least by the XWiki committers of the open source project working for XWiki SAS. XWiki will also continue to work on the CI pipeline scripts we use for global coverage computation and differences between reports, in addition to the flaky test script handling.