

Title:	WP4 – D41 – Collaborative Software Engineering Platform
Date:	May 19, 2017
Writer:	OW2
Reviewers:	TU Delft

Table of Contents

1.EXECUTIVE SUMMARY.....	3
2.REVISION HISTORY.....	3
3.OBJECTIVES.....	3
4.INTRODUCTION.....	3
5.REFERENCES.....	4
6.INTERNAL WIKI PLATFORM.....	4
7.PUBLIC KNOWLEDGE BASE.....	5
8.MAILING-LIST MANAGER.....	7
9.SLACK.....	7
10.GITLAB INSTANCE.....	8
11.DEVOPS QUICK START.....	9
12.OSCAR.....	9
13.OW2STACK.....	10
14.APPHUB.....	11
15.COMPUTING AND STORAGE RESOURCE REQUIREMENTS.....	12
16.CONCLUSION.....	13

Illustration Index

Figure 1: An architecture of participation (source: OpenSource.com).....	3
Figure 2: Overview of the Collaborative Engineering Platform.....	4
Figure 3: STAMP private wiki home page.....	5
Figure 4: STAMP Knowledge Base.....	6
Figure 5: Mutation Testing entry in the knowledge base.....	6
Figure 6: Main STAMP mailing-list.....	7
Figure 7: STAMP general Slack channel.....	8
Figure 8: STAMP projects hosted in GitLab.....	9

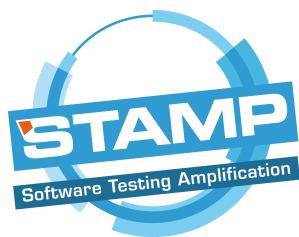
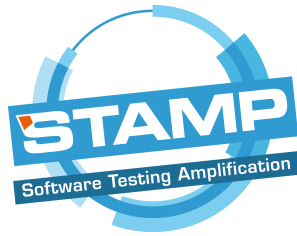


Figure 9: OSCAR Overview.....	10
Figure 10: Ow2Stack.....	11
Figure 11: AppHub Home Page.....	12



1. Executive Summary

This report presents the STAMP collaborative engineering platform and its use within the Consortium. The platform covers all software lifecycle phases from code development to Cloud deployment. It hinges on a wiki, a knowledge base, a mailing-list manager, Slack rooms, a GitLab instance, the OW2 OSCAR framework, an OpenStack instance and AppHub.

2. Revision History

Date	Version	Author	Comments
24-Apr-2017	0.10	Stéphane Laurière (OW2)	Structure, initial content
15-May-2017	0.20	Review by Caroline Landry (Inria)	Review
18-May-2017	0.30	Updates by Stéphane Laurière	Updates taking into account the review.
18-May-2017	0.40	Review by Mozhan Soltani (TU Delft)	Review
19-May-2017	1.00	Updates by Stéphane Laurière	Took Mozan Soltani review into account

3. Objectives

The objective of this report is to explain how the stakeholders of the STAMP project collaborate with a set of tools in order to create, test and deploy software, write documentation and deliverables. The underlying processes are powered by a software engineering platform. Each component of this platform is described in a section of this report.

4. Introduction

STAMP involves 9 partners, each of which contributes to one or several pieces of a global framework that will let developers amplify their tests at three levels: unit testing, configuration testing, runtime testing. This requires an architecture of collaboration so that every partner can bring as easily as possible their input and developments to the global framework. This is made possible by putting at the disposal of the Consortium a set of tools and best practices facilitating collaboration during several phases: requirement gathering, specification definition, code development and testing, integration, packaging and deployment. These phases are supported by an infrastructure hosted and maintained by OW2. It comprises template projects so as to let contributors quickly start new projects.

The STAMP collaborative engineering platform contains a set of components illustrated in figure 1. Each component is described in greater details in the sections of this document.

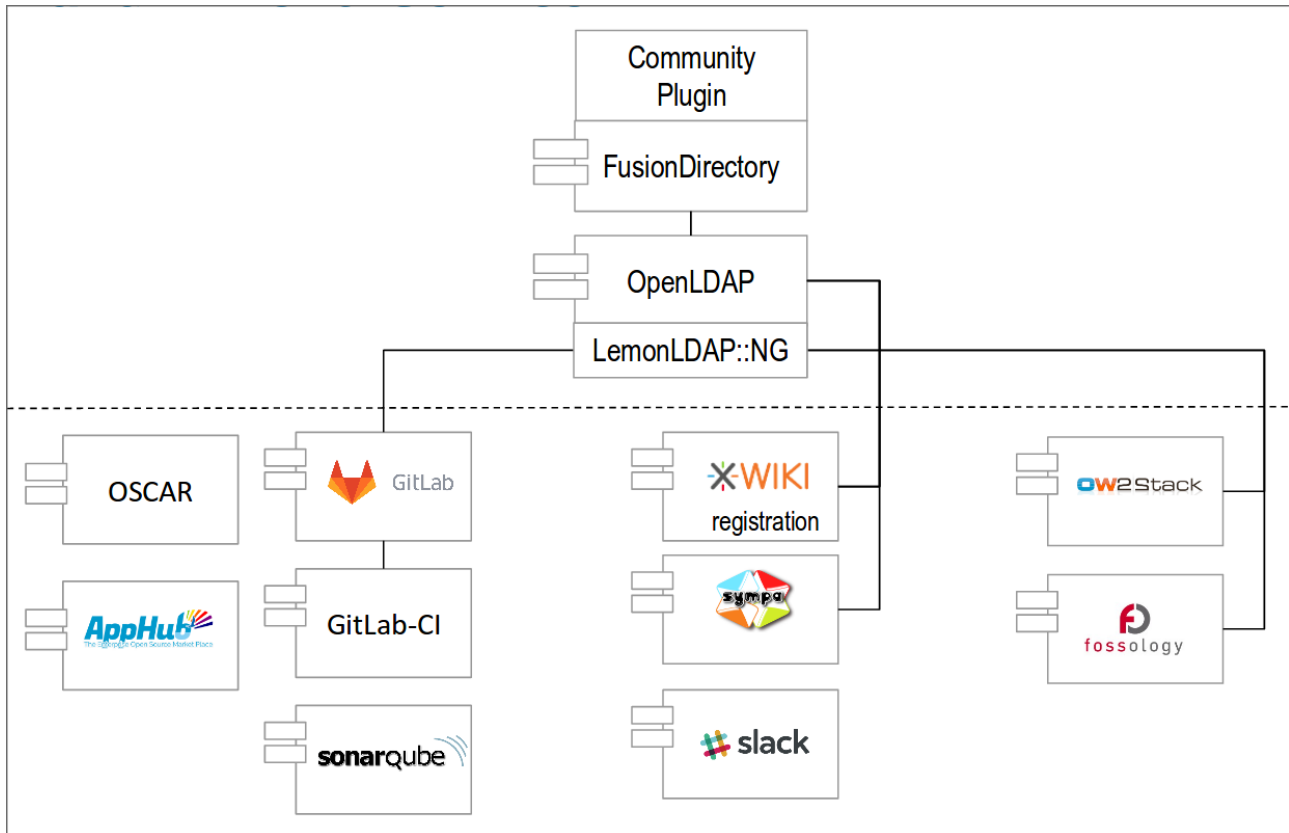


Figure 1: Overview of the Collaborative Engineering Platform

5. References

- [1] STAMP quality plan: [d71_stamp_quality_plan.pdf](#)
 [2] D41 – Collaborative Software Engineering Platform: [d41_cse_platform.pdf](#)

6. Internal Wiki Platform

A wiki restricted to the members of the Consortium has been set up. It can be accessed from the same domain as the main STAMP web site, in a private space: <http://www.stamp-project.eu/wiki/>. The figure below illustrates the wiki home page. Authentication and access control are managed by an LDAP directory with an interface based on FusionDirectory, a LDAP user interface.

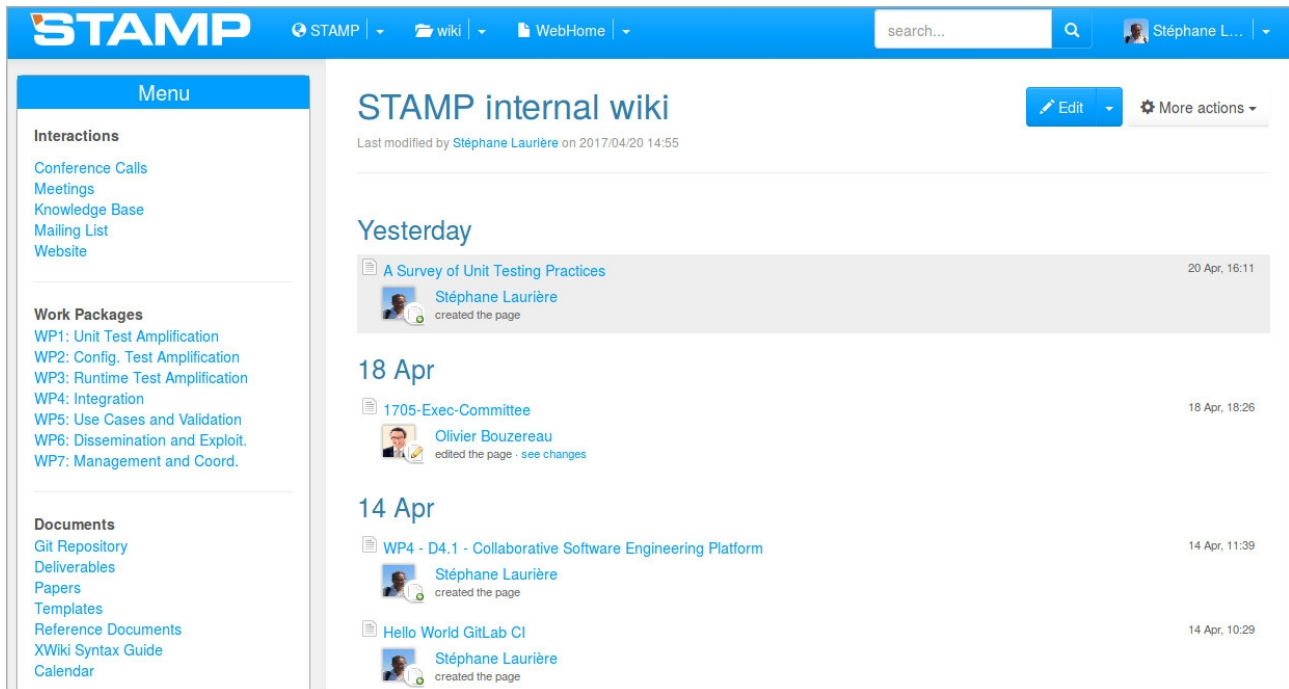


Figure 2: STAMP private wiki home page

The home page is organized as follows:

- The main column displays an activity stream showing the most recent additions or updates brought to the wiki.
- A panel on the left contains links to key pages of the wiki, organized in three main sections:
 - Interactions / Communication
 - Work packages
 - Documents (templates, reference documents, etc.).
- A toolbar at the top gives quick access to the search feature and to the user profile

The private wiki is mainly used for the following tasks:

- Meeting preparation and follow-up (availability of all meeting slides)
- Work-in-progress documents aimed at gathering input from the partners
- Publication of intermediary versions of the deliverables

7. Public Knowledge Base

A public knowledge base has been put in place at <https://www.stamp-project.eu/bin/view/kb/> in order to gather in a common place all the scientific publications, articles, presentations, events or software that relate to the topics covered by STAMP. It allows to share knowledge easily among members. It is also a way to make the STAMP site a reference in the field of testing in general and test amplification in particular. The illustration below shows a list of definitions relating to the STAMP domain: flaky test, mutation testing parameterized test.

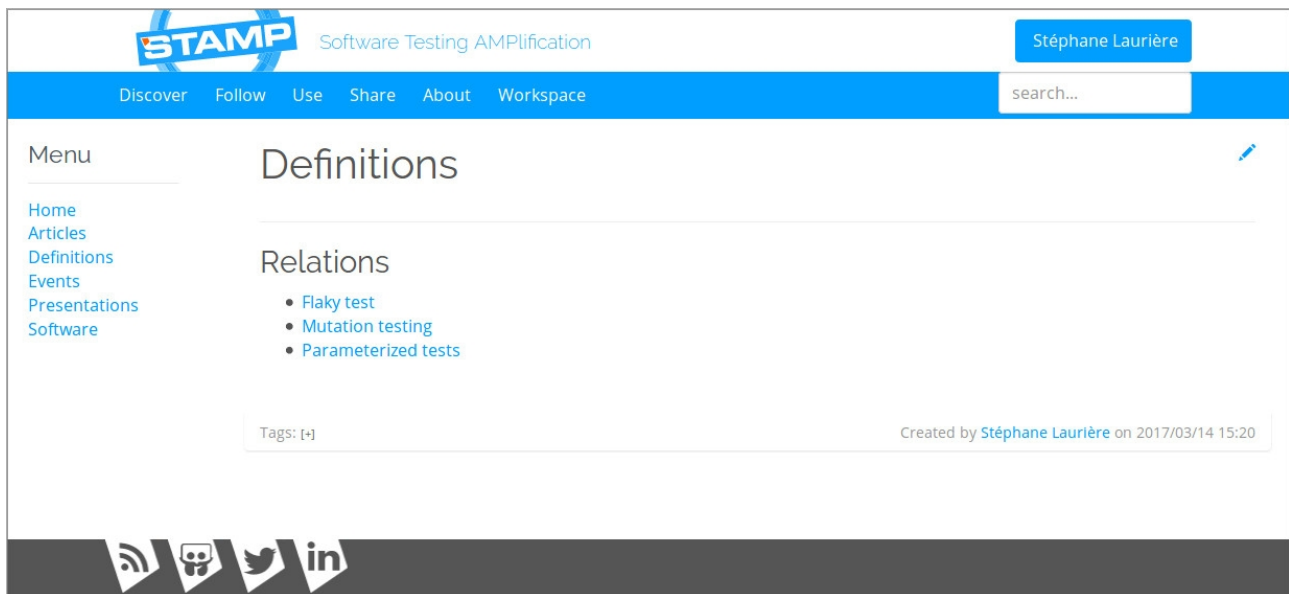


Figure 3: STAMP Knowledge Base

When hitting the “Mutation testing” link, the user heads to a page giving the definition of the concept, together with links to software that relate to it, in this case PIT, as exhibited by the figure below. At the time of the writing of this document, the knowledge base contains approximately 20 pages.

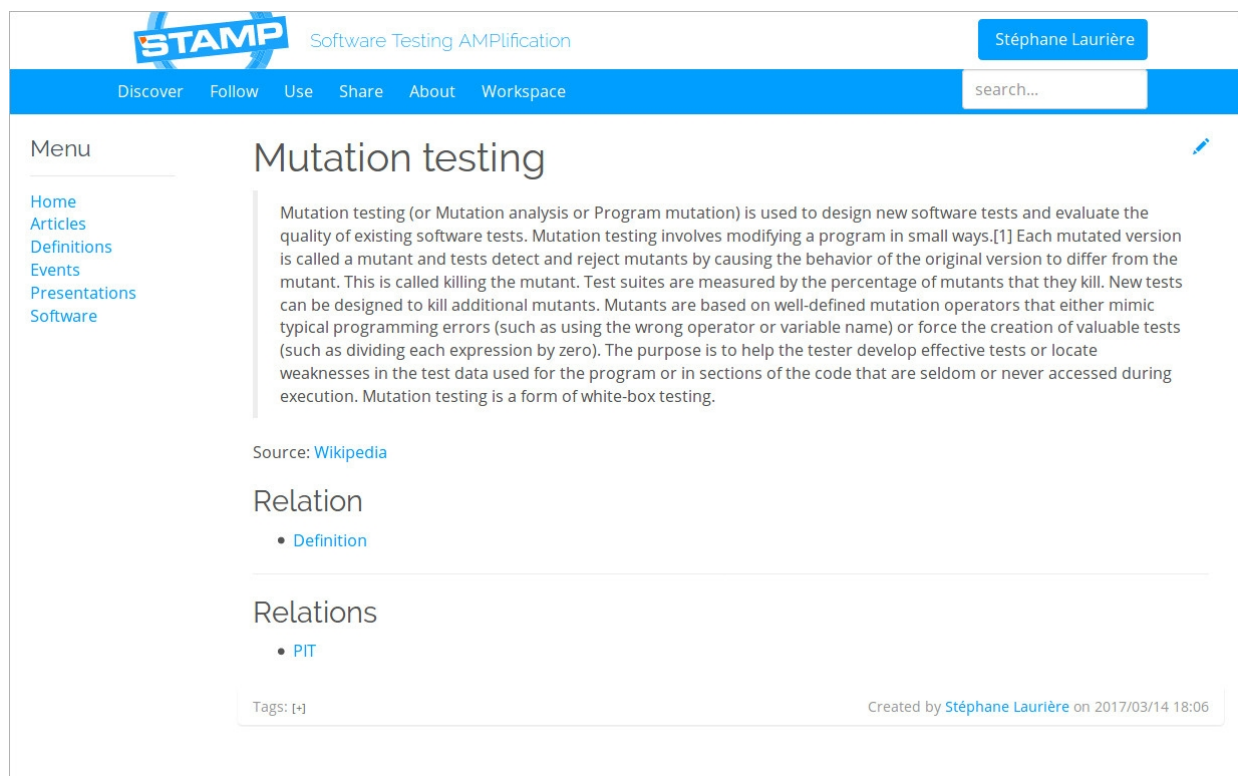
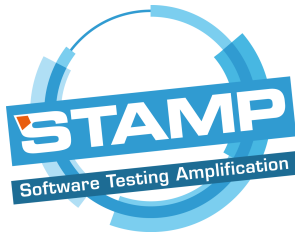


Figure 4: Mutation Testing entry in the knowledge base



8. Mailing-List Manager

The project uses the OW2 infrastructure for managing mailing-lists, based on SYMPA. Two mailing-lists are used: a general one covering most of the topics discussed in STAMP so as to encourage information sharing across all work packages, and one dedicated to the use cases workpackage, WP5. The strategy is to avoid the proliferation of mailing lists. However other lists may be set up only when the need has been duly qualified. The main list can be administered by the INRIA team representatives and by the OW2 team. The archive of the main STAMP list is illustrated by the figure below. The list has 43 subscribers.

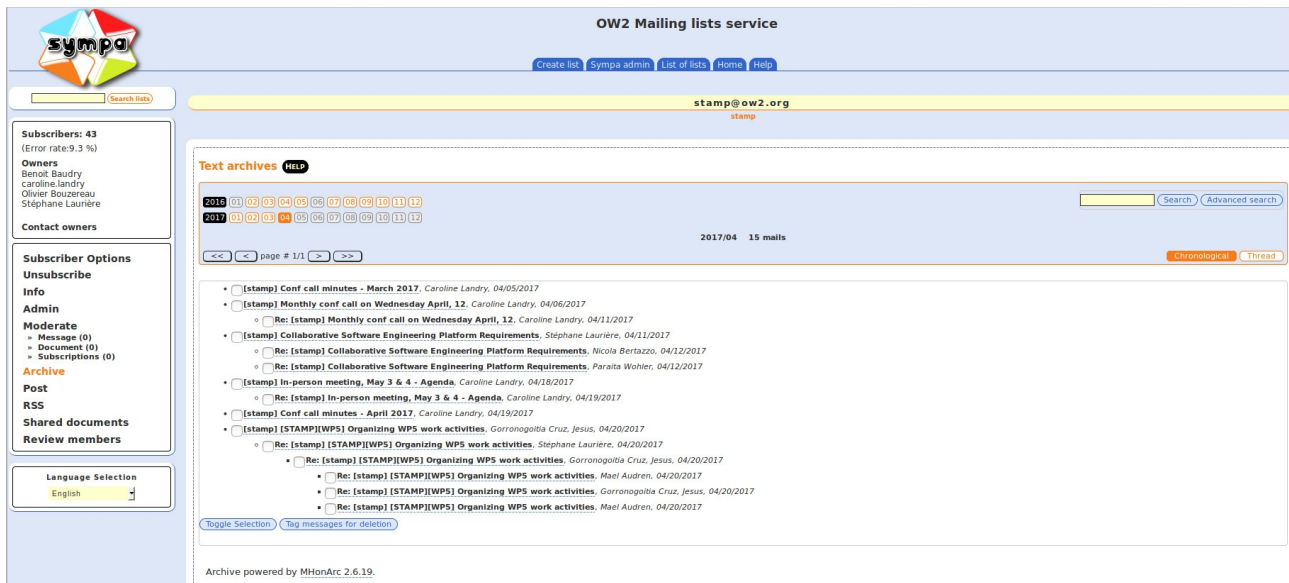


Figure 5: Main STAMP mailing-list

9. Slack

As a complement to the mailing-lists, several Slack channels have been created for real-time communication. Slack is a cloud based communication tool allowing team members to communicate without the use of email. One Slack room has been created for general discussions about STAMP, and one room per workpackage is available for discussing more specific topics.

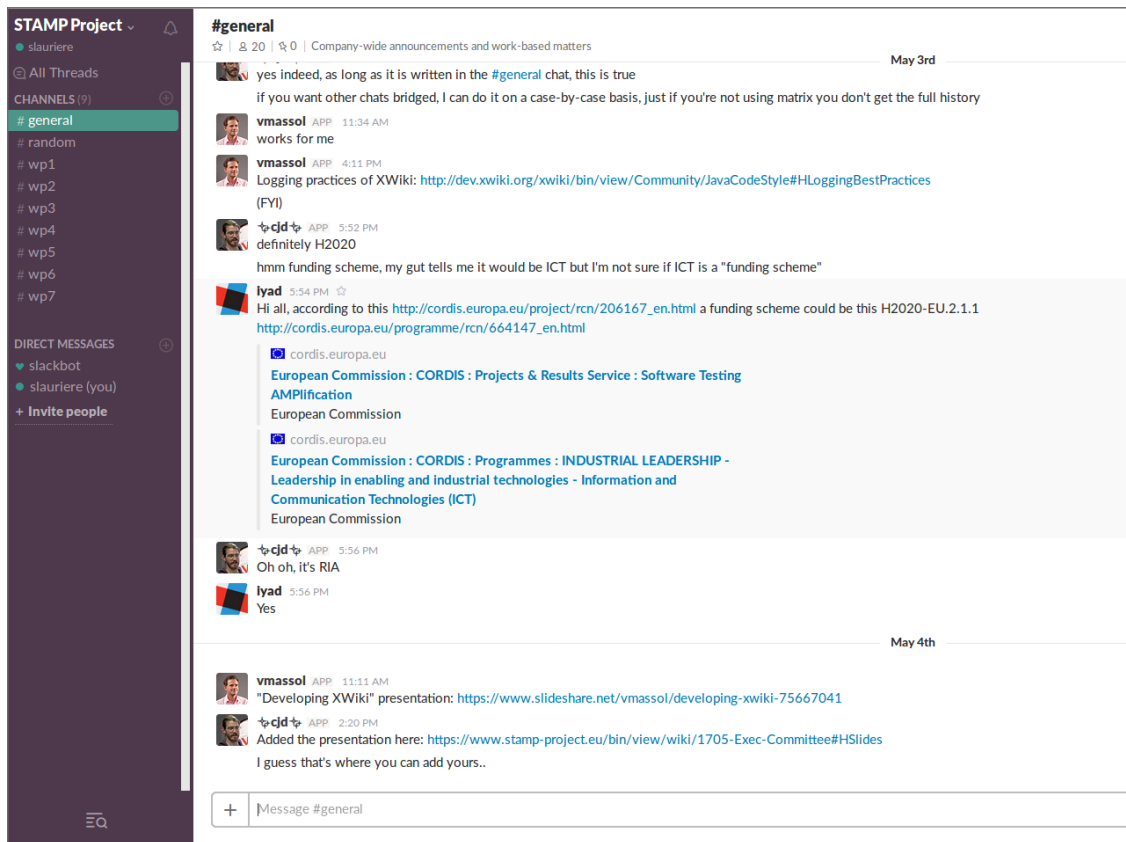


Figure 6: STAMP general Slack channel

10. GitLab Instance

[GitLab](#) is a web-based Git repository manager with wiki and issue tracking features, using an open source license. It has been selected by OW2 for its open source license and for its wide list of features. GitLab is running on OW2 premise. It brings to STAMP powerful tools for continuous integration and Docker images management. The installation includes a Docker registry, making it easy for developers to code, test, and deploy Docker container images using GitLab CI, the continuous integration server included in GitLab. An image containing Maven artifacts has been set up, in order for a project to speed up the set up of a build environment without the need to download all Java dependencies at each build. A STAMP group has been created in the OW2 GitLab instance. As of May 2017, the STAMP group contains 7 distinct projects, exhibited by the figure below. Six projects are public, one is private as it relates to internal documents.

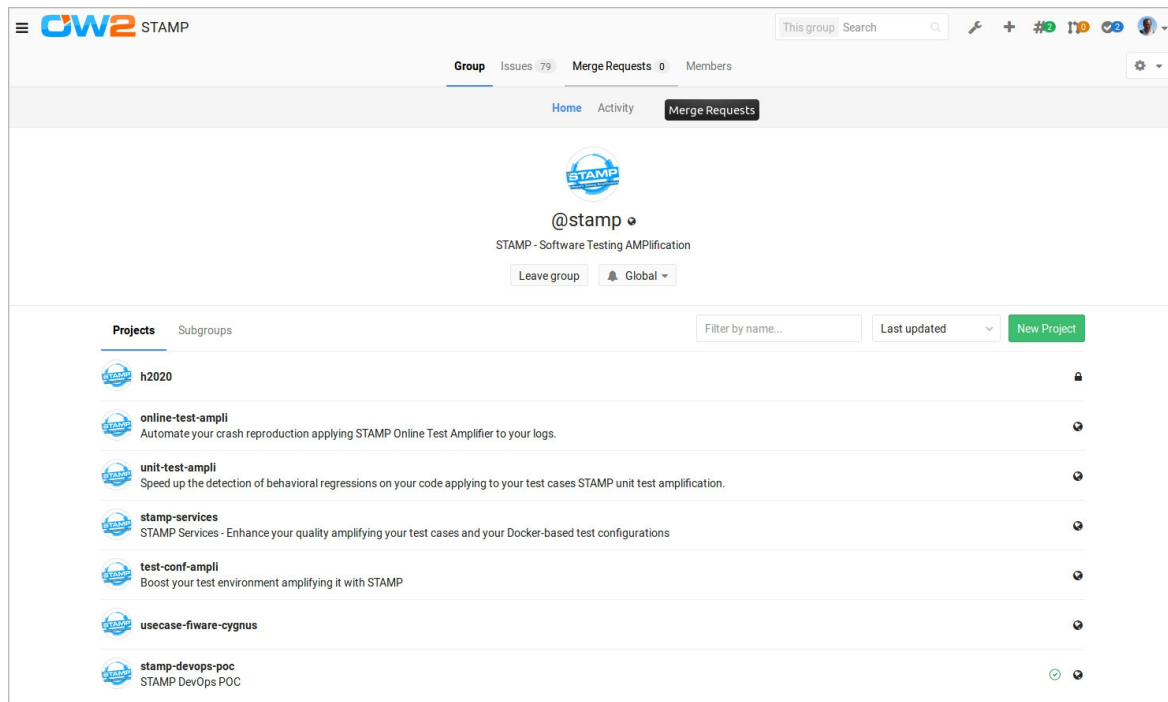
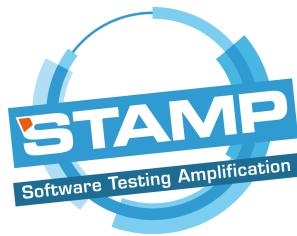


Figure 7: STAMP projects hosted in GitLab

11. DevOps Quick Start

An example application illustrating how to use build pipelines has been developed by Engineering. It is named “DropWizard Demo Application”. It consists of the following:

- Java code of a HelloWorld application,
- Instructions on how to build the code and to run tests,
- A YAML file to configure the continuous integration pipeline within GitLab-CI, the GitLab continuous integration server,
- Instructions on how to run the application in “bare-metal” using the DropWizard framework¹ which turns the built artifact into a standalone application,
- A Docker file in order to run the application in a Docker image.

This quick start application is meant to provide a framework to the STAMP contributors to test, build and deploy their applications into a Docker container.

12. OSCAR

In addition to the example pipeline and guidelines provided by the DevOps quick start application described above, the STAMP collaborative software engineering platform includes the OSCAR tools. OSCAR stands for Open source Capability Assessment Radar. It is a general framework developed by OW2 and providing guidelines and measurement tools aiming at improving the quality of open source products.

OSCAR is organized around two pillars exhibited by the figure below: Governance and Engineering. Each pillar is organized in chapters, each of which is instrumented by one or several tools.

¹ <http://www.dropwizard.io/1.0.0/docs/>

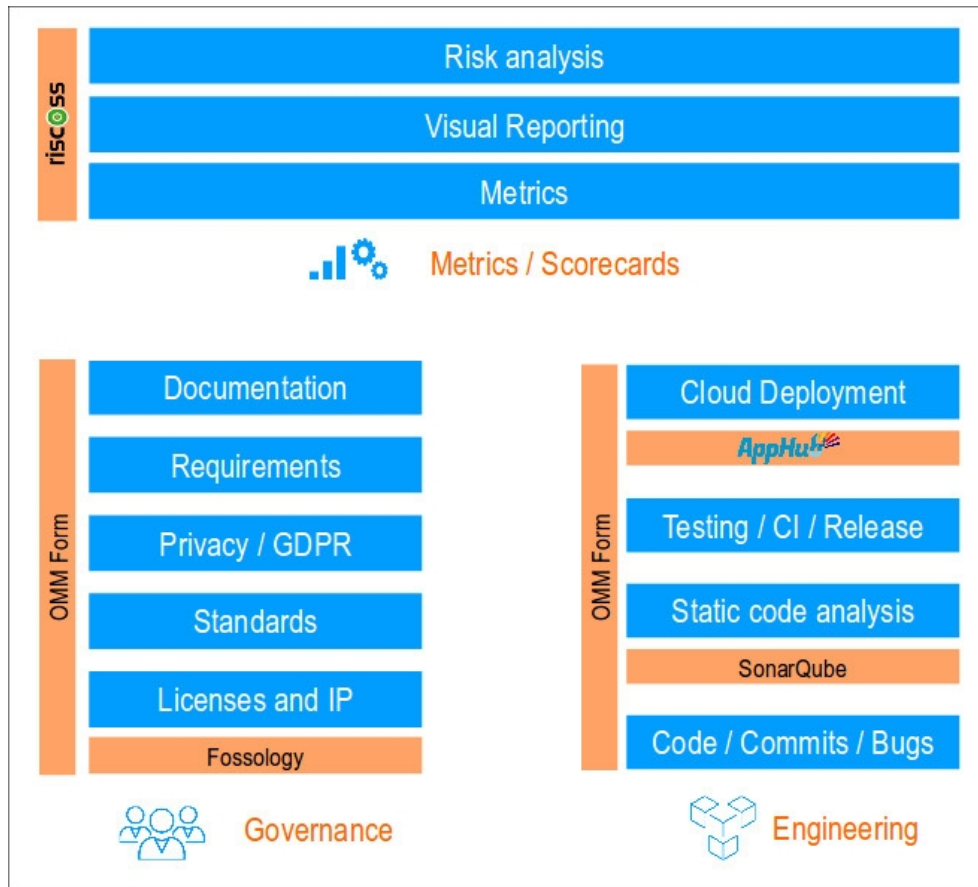


Figure 8: OSCAR Overview

Some of the tools provided by OSCAR result from previous European projects:

- RISCOSS to compute and display Market Readiness Levels
- AppHub to ease the deployment of projects into the Cloud

The projects developed during the course of STAMP will benefit from the OSCAR programme, in particular SonarQube for static code analysis, and FOSSology for licenses and IP check. Conversely, the outcome of STAMP will be integrated into the OSCAR testing chapter.

13. Ow2Stack

Beside offering the possibility to run STAMP applications within a Docker container, the STAMP collaborative engineering platform includes an OpenStack instance into which the applications can be deployed. This instance is named Ow2Stack. The figure below illustrates a set of OW2 projects images deployable within Ow2Stack.

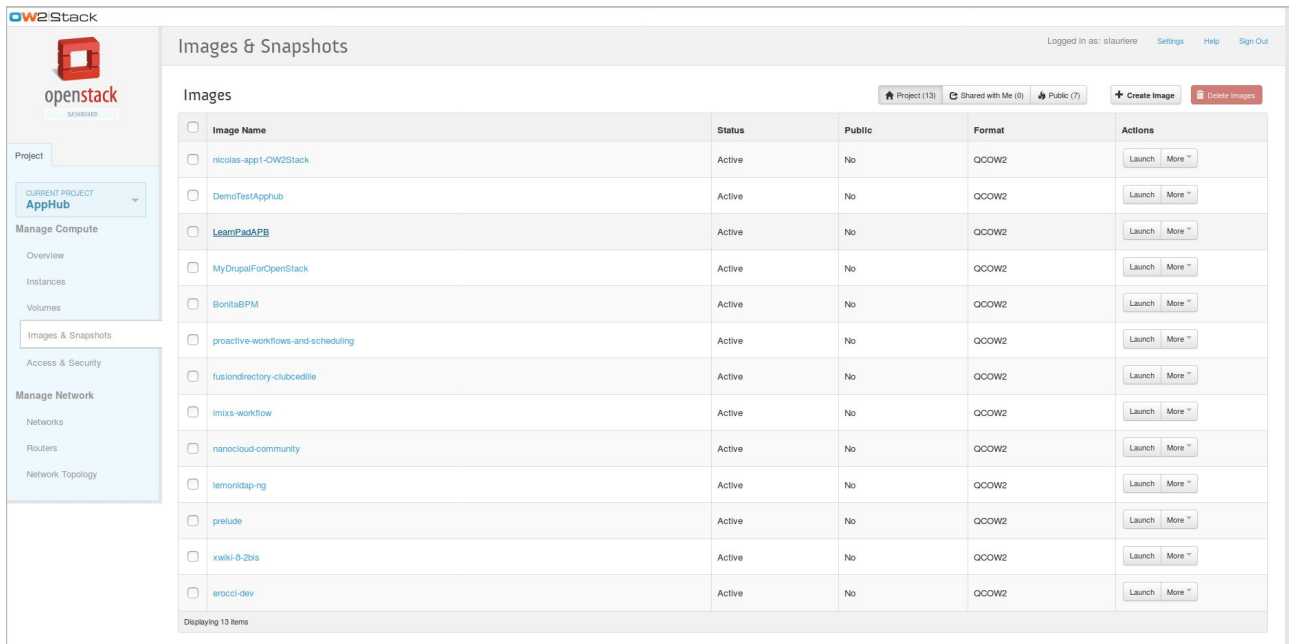
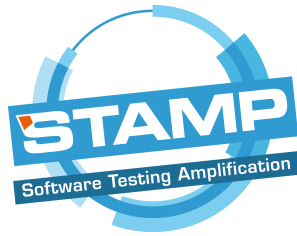


Figure 9: Ow2Stack

14. AppHub

AppHub – The European Open Source Software Marketplace – is a service platform that brings open source projects closer to the market. It was developed in the frame of an EU project in 2015-2016. AppHub helps the market to seamlessly identify, position and implement the software outcomes of open source projects. The platform is run by OW2 and UshareSoft.

A key objective of AppHub is to foster adoption of open source projects by making them trustworthy, easy to find, and easy to download and run. AppHub provides open source project software as cloud-ready packages that can be executed by a broad range of cloud service providers. The open source software great shopping mall will also help accelerate open source adoption.

STAMP will benefit from the AppHub marketplace by taking advantage of the AppHub Factory which will let STAMP project owners package their software assets as images which can be deployed in a large variety of Clouds: OpenStack, AWS, Google Compute Engine, Azure and more.

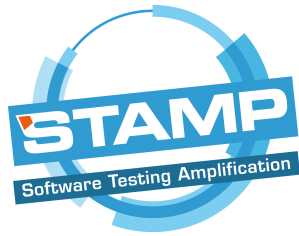


Figure 10: AppHub Home Page

15. Computing and Storage Resource Requirements

A survey has been conducted to evaluate the resources that will be needed by the partners during the course of the project. At this stage of the project, the expressed requirements are gathered in the table below. They will evolve during the next steps of the project.

Organization	VM number	Specifications	Other resources
Engineering	3	<ul style="list-style-type: none"> CPU: 4 RAM: 8GB Storage: 200GB 	<ul style="list-style-type: none"> Container registry installed in the GitLab infrastructure GitLab runners
ActiveEon	1	<ul style="list-style-type: none"> CPU: 4 RAM: 4GB Storage: 16GB 	<ul style="list-style-type: none"> 1 public IP Direct access to the VM



16. Conclusion

A set of powerful components has been set up in order to provide the Consortium with an efficient collaborative software engineering platform. The platform offers all the main services covering software lifecycle: requirement gathering, communication, code development, build, continuous integration, testing, deployment, quality assessment and documentation writing. The infrastructure will continue to evolve along the course of the project in accordance with the needs expressed by the partners.