IVVES

Industrial-grade Verification and Validation of Evolving Systems

Labeled in ITEA3, a EUREKA cluster, Call 5

ITEA3 Project Number 18022

D5.1 – Requirements for the IVVES experimentation framework

Due date of deliverable: Sept 30, 2020
Actual date of submission: Oct 8, 2020

Start date of project: 1 October 2019 **Duration:** 39 months

Organisation name of lead contractor for this deliverable: **OU**

Author(s):	Tanja E.J. Vos
Status:	Draft (living document)
Version number:	1.1
Submission Date:	08-10-2010
Doc reference:	IVVES_Deliverable_D5.1_V1.0.docx
Work Pack./ Task:	WP5.1
Description: ***(max 5 lines)***	This deliverable describes the requierements for the IVVES framework, an online platform where all IVVES artefacts to the people outside of the consortium.

Nature:	<Use one of these codes: **R**=Report, **P**=Prototype, **D**=Demonstrator, **O**=Other>		
Dissemination Level:	**PU**	Public	**X**
	PP	Restricted to other programme participants	
	RE	Restricted to a group specified by the consortium	
	CO	Confidential, only for members of the consortium	

DOCUMENT HISTORY

Release	Date	Reason of change	Status	Distribution
V0.1	02/06/2020	First draft	Draft	WP5
V0.2	10/06/2020	Added more requirements	Draft	WP5
V0.3	16/07/2020	Added more requirements	Draft	WP5
V0.4	09/09/2020	Changes some requirements	Draft	Whole project
V1.0	17/09/2020	Send to PMT	Concept	Whole project
V1.1	08/10/2020	Approved by PMT, to be submitted to ITEA3	Final	ITEA

Table of Contents

Glossary

Abbreviation / acronym	Description
IVVES artefact	IVVES outcomes. Types of artefacts we will have are: tools, data, courses, tutorials.

1. Executive Summary (OU)

The IVVES framework will be an online platform where we present the IVVES artefacts to the people outside of the consortium. By making sure that all the IVVES outcomes come with a well-defined and clear package of training and possibilities to try them out (experiment with them), we hope to give dissemination of the IVVES outcomes a real boost.

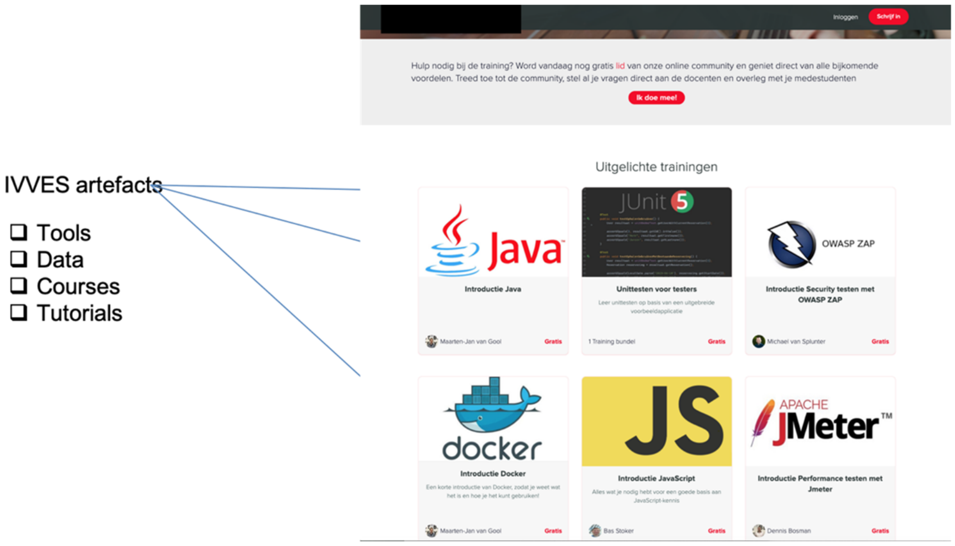
This document describes the requirements of the platform.

2. Requirements for the IVVES experimentation framework

During a first round of brainstorming, we came to agree that a platform like the PRAEGUS online academy would be a perfect match for what we have in mind for the IVVES framework. This means that we have already agreed upon the following general requirements:

1. The IVVES framework will consists of a platform that links to the shared IVVES outcomes, or artefacts. Types of artefacts we will have are:
    1.1. tools
    1.2. data
    1.3. courses
    1.4. tutorials

2. On the homepage, for each IVVES artefact, some sort of "tiles" are present that, when clicked, lead the user to more details on that specific IVVES artefact



Hulp nodig bij de training? Word vandaag nog gratis lid van onze online community en geniet direct van alle bijkomende voordelen. Treed toe tot de community, stel al je vragen direct aan de docenten en overleg met je medestudenten

Introductie Java
Maarten-Jan van Gool Gratis

Unittesten voor testers
Leer unittesten op basis van een uitgebreide voorbeeldapplicatie
1 Training bundel Gratis

Introductie Security testen met OWASP ZAP
Michael van Splunter Gratis

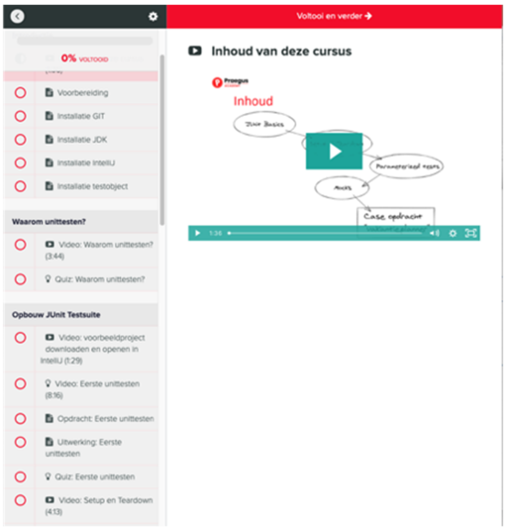
Introductie Docker
Een korte introductie van Docker, zodat je weet wat het is en hoe je het kunt gebruiken!
Maarten-Jan van Gool Gratis

Introductie JavaScript
Alles wat je nodig hebt voor een goede basis aan JavaScript-kennis
Bas Stoker Gratis

Introductie Performance testen met Jmeter
Dennis Bosman Gratis

3. The artefacts of IVVES can then be presented in different (overlapping) ways and will have contents like:
    - Video material (tool demos, online lectures, presentations)
    - Links to SaaS tools (tools will not be hosted by the framework)
    - Course material
    - Presentations
    - Downloadable material (tools, data)

This document and the information contained are the property of the IVVES Consortium and shall not be copied in any form or disclosed to any party outside the Consortium without the written permission of the Project Coordination Committee, as regulated by the IVVES Consortium Agreement and the AENEAS Articles of Association and Internal Regulations.



The **contents** of the IVVES artefacts unfolds

Other high-level requirements for the framework are:

4. The framework should be IVVES branded, so that it is recognizable. This means we cannot for example build upon or re-use the PRAEGUS online academy, instead we need to find another, but similar option.

5. The solution should be an open source platform

6. Possibility to install the solution on a server that will be maintained by IVVES partners at least 5 years after the project.

This document and the information contained are the property of the IVVES Consortium and shall not be copied in any form or disclosed to any party outside the Consortium without the written permission of the Project Coordination Committee, as regulated by the IVVES Consortium Agreement and the AENEAS Articles of Association and Internal Regulations.

3. First inventory of the IVVES artefacts

This inventory will grow during the execution of the project. This deliverable is a living document.

- **TESTAR tool**
    - TESTAR is a tool for automated testing at the GUI level
    - Experiment with it online (having for example online access to a virtual desktop hosted outside the platform)
    - Links to open source code (github)
    - Courses
        - In depth course on scriptless testing with TESTAR with:
            - Videos
            - Exercises
            - Experience papers
    - Tutorials:
        - How to get started
        - How to install it
        - How to build it from scratch
        - How to become a contributor

- **SaFReL (RISE)**

    SaFReL is a self-adaptive fuzzy reinforcement learning-based performance testing framework which makes the tester agent learns the optimal policy for generating (platform-based) performance test conditions meeting the testing objective without access to a performance model of the system and replays the learnt policy on further testing cases. Finding the performance breaking point of the software under test (SUT), at which the system functionality breaks, or the performance requirements are not satisfied anymore, is the main objective of the performance testing in this framework. The current prototype mainly focuses on stress testing regarding manipulating the resource availability.

    It assumes two learning phases, i.e., initial and transfer learning. First, it learns the optimal policy through the initial learning and then reuses the learnt policy for observed software systems with performance sensitivity analogous to already observed ones while still keeping the learning running in the long-term. SaFReL uses Q-learning augmented by fuzzy logic and has the capability of transfer learning and reusing knowledge in similar situations. It benefits an adaptive action selection strategy that adapts the learning to various testing situations and subsequently makes the agent able to act efficiently on various SUTs. The current prototype uses a performance simulation module to estimate the effects of the applied actions. This framework could be deployed on a cloud-based testing setup with the capability of resource scaling.

- **RELOAD (RISE)**

  RELOAD is an intelligent reinforcement learning-driven load generation tool which generates efficient test load and executes it through a load runner such as Apache JMeter on SUT. RELOAD learns the optimal policy to generate an efficient test workload which meets testing objective, e.g. reaching an intended error rate, without access to a model or source code of SUT. It effectively learns the effects of different transactions involved in the workload and how to tune the load of transactions to approach the intended error rate.
  The intelligent tester agent can reuse the learned policy in further similar testing scenarios, for example, reaching different target values of response time or error rate. The learning-based load testing can reach the testing objective with lower cost in terms of workload size (number of users), i.e. smaller workload, compared to a typical load testing process. It is generally beneficial to the continuous testing activities such as varying scenarios and performance regression testing.

- **VARA (RISE)**
  - Variability-Aware requirements Reuse Analysis (VARA) is a tooled method that aims to automate the requirements reuse analysis and thus helps teams achieve quick and quality delivery of software systems. VARA takes customer requirements as input and uses state-of-the-art natural language processing and machine learning algorithms to predict already implemented features that can be reused to realize customer requirements.
  - In IVVES platform, we will offer tutorials and demos on VARA to the partners, and a course on how to automate reuse analysis with NLP and machine learning.

- **Tests selection python module (University of Helsinki, F-secure)**
  - The module is a plugin to pytest framework. It analyses tests coverage data and selects for execution only tests related to recent changes in the project basing on VCS (git). Thus, it speeds up both local testing and testing on CI build server.
  - Links to open source code (github)

4. Follow up and Conclusions

This is a living document where artefacts will be added throughout the course of the project. The next steps are:

- Setting up a first version of the infrastructure for the platform respecting the requirements from Section 1.
- Adding the tiles from the first inventory.
- Keep adding more tiles for artefacts that are being engineered throughout the course of the project.