PaaSage

Model Based Cloud Platform Upperware

Deliverable D1.7.1

State-of-the-art in Evaluation of MDE Methodologies and Platforms:

Specification of the Evaluation Framework

Version: 1.0

D1.7.1

Name, title and organisation of the scientific representative of the project's coordinator:

Mr Philippe Rohou Tel: +33 (0)4 97 15 53 06 Fax: +33 (0)4 92 38 78 22 E-mail: philippe.rohou@ercim.eu

Project website address: http://www.paasage.eu

Project	
Grant Agreement number	317715
Project acronym:	PaaSage
Project title:	Model Based Cloud Platform Upperware
Funding Scheme:	Integrated Project
Date of latest version of Annex I against which the assessment will be made:	8th of July 2014
Document	
Period covered:	
Deliverable number:	D1.7.1
Deliverable title	Analysis and Specification of the Evaluation Framework
Contractual Date of Delivery:	30th of September 2014 (M24)
Actual Date of Delivery:	30th of September 2014 (M24)
Editor (s):	Achilleas Achilleos (UCY)
Author (s):	Achilleas Achilleos (UCY), Christos Mettouris (UCY)
Reviewer (s):	Vasily Bunakov (STFC), Brian Matthews (STFC), Tom Kirkham (STFC), Frode Finnes Larsen (EVRY), Bjarne Gram (EVRY)
Participant(s):	George A. Papadopoulos (UCY), Vasily Bunakov (STFC), Tom Kirkham (STFC), Brian Matthews (STFC)
Work package no.:	1
Work package title:	Technical Foundation
Work package leader:	STFC
Distribution:	R (Report) PU (Public)
Version/Revision:	1.0
Draft/Final:	Final
Total number of pages (including cover):	25

DISCLAIMER

This document contains description of the PaaSage project work and findings.

The authors of this document have taken any available measure in order for its content to be accurate, consistent and lawful. However, neither the project consortium as a whole nor the individual partners that implicitly or explicitly participated in the creation and publication of this document hold any responsibility for actions that might occur as a result of using its content.

This publication has been produced with the assistance of the European Union. The content of this publication is the sole responsibility of the PaaSage consortium and can in no way be taken to reflect the views of the European Union.

The European Union is established in accordance with the Treaty on European Union (Maastricht). There are currently 28 Member States of the Union. It is based on the European Communities and the member states cooperation in the fields of Common Foreign and Security Policy and Justice and Home Affairs. The five main institutions of the European Union are the European Parliament, the Council of Ministers, the European Commission, the Court of Justice and the Court of Auditors. (http://europa.eu)

PaaSage is a project funded in part by the European Union.

Table of Contents

Contents

Table of Figures

List of Tables

EXECUTIVE SUMMARY

This document outlines the specification of the PaaSage evaluation framework at the end of the second year of the project. The deliverable describes the PaaSage evaluation framework that defines two complementary evaluation processes: (i) the core evaluation process and (ii) a secondary evaluation process. The core evaluation process is based on the well-known and widely-used Technology Acceptance Model, which is adopted, modified and extended in this work to address and evaluate the realisation of the project objectives and requirements. The evaluation factors of the extended model are defined and explained, including also the definition of the evaluation criteria. The secondary evaluation process will only be used if it is deemed necessary by the consortium, so as to complement the core process. This can serve as a complementary evaluation method, principally because of the significance of the software tools in the PaaSage platform. Finally, we outline the steps for executing the evaluation, collecting and analysing the results at the end of year 2 and year 3 of the project.

1. The PaaSage Evaluation Framework

1.1. Target

The definition of an evaluation framework aims to provide the capability to assess the adoption and use of MDE methodologies and tools in the current software practices of the use case partners. More importantly, it aims to qualitatively and quantitatively assess the adoption of the PaaSage model-driven methodology and integrated tool-chain. This evaluation will be performed mainly by the PaaSage implementers (i.e., use case partners), based on the realization of the pilots (i.e., use cases demonstrators) that will be performed in the context of the project, but also by the PaaSage developers (i.e., partners developing the components/tools of the platform). In this deliverable, we will be using periodically the terms PaaSage implementers and PaaSage developers as explained above.

The framework comprises two evaluation processes: (i) the core evaluation process and (ii) the secondary evaluation process. The core evaluation process is based on the adoption and extension of the Technology Acceptance Model (TAM). On top of that, a secondary evaluation process is defined mainly because of the importance of software tools in the PaaSage platform. The secondary evaluation process is based on six non-functional dimensions and can be utilised if required to provide an additional qualitative assessment using the radar-chart evaluation method adopted and adapted from the field of organizational development. The core process will be performed via completing surveys/questionnaires and selected software metrics, while the secondary process can be enforced and performed, by completing assessment spreadsheets for each PaaSage tool and use case.

In overall the main objective of the evaluation is twofold: 1) to document use case partners' current practices of MDE and 2) their future intentions in applying MDE based explicitly on the use of the defined PaaSage methodology and developed tooset in the realization of the use case pilots.

1.2. Analysis of Existing MDD Evaluation Frameworks

A targeted review and analysis of existing research studies regarding experiences from applying MDE in industry [1], [2], but also from applying and evaluating MDE in research projects such as the MPOWER project [3] and MODELPLEX project [4] revealed that the Technology Acceptance Model [5] is a widely used method of qualitative evaluation. The model provides also the capability to be adapted to the requirements and objectives of the PaaSage project.

1.3. Key Requirements and Evaluation Plan

The PaaSage project key requirements directed the formulation of an evaluation plan that will drive the evaluation framework in this project. These key requirements are:

R1. The application of MDE in different contexts (i.e. the PaaSage use cases demonstrators) needs to be respected, since different organisations have diverse characteristics such as particular processes, tools and expertise, which can significantly affect the adoption and application of MDE.

R2. The evaluation method should permit to collect and disseminate experiences of use case partners on current practices (with and without MDE), and experiences on the use of the PaaSage MDE-based practices. It should also consider that partners might not want to disclose details on their processes. In overall this requirement will aid in dissemination and exploitation of the project results.

Based on the above an evaluation plan was defined based on the following 3-axes:

1. **Definition of Research Questions:** Based on the case studies (i.e., scenarios) of individual partners, generic research questions were defined and listed in the following section, based also on the requirements, goals and intentions in terms of applying the PaaSage MDE methodology and tools (R1).
2. **Collecting PaaSage Implementers Feedback:** An extended version of the Technology Acceptance Model (TAM) will be defined and used as the core of the evaluation process to collect feedback on applying the PaaSage MDE methodology, the usage of PaaSage tools and future adoption intentions (R2).
3. **Evaluation of MDE tools:** PaaSage developers and/or implementers are considered as the main actors to evaluate and assign scores in specific worksheets that can be prepared, so as to collect feedback on the success of accomplishing technical and user requirements (R2).

1.4. Research Questions and the Core Evaluation Method

As aforementioned, the core evaluation method is defined on the basis of the Technology Acceptance Model (TAM) used for MDE-specific evaluation in various research and development projects, such as in the MPOWER project [3] and the MODELPLEX EU project [4].

Figure 1 illustrates the abstract workflow that defines the evaluation method used in the PaaSage project. The generic research questions are initially defined and used to identify and define the necessary TAM factors that are highly related to the project goals. The assessment will be performed by defining a list of concrete evaluation criteria that are extracted from the defined TAM factors. These concrete evaluation criteria serve as the guideline for defining the appropriate survey questions and identifying the proper software metrics that will drive the evaluation method. In fact, the evaluation criteria will steer discussion and aid in interpreting the TAM results, so as to come up with the final results and conclusions that will showcase, to the highest possible degree, if the project has accomplished effectively its goals and objectives.

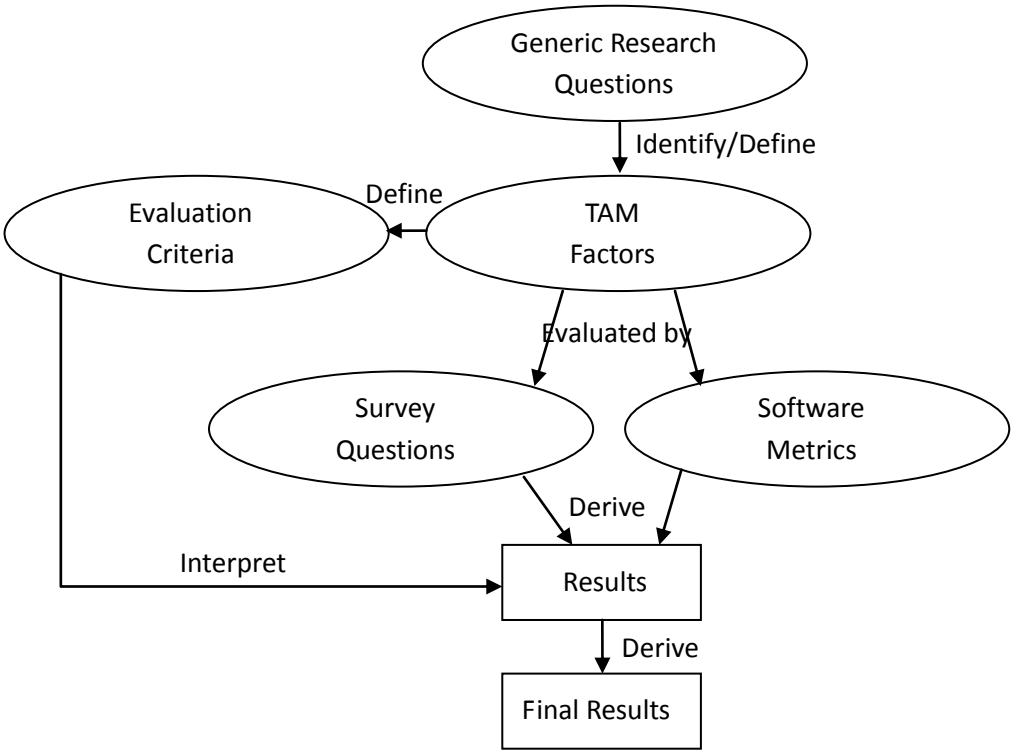

Figure 1: *Abstract Core Evaluation Process Workflow*

Hence, suitable research questions are defined in regards to the needs for adoption of the PaaSage MDE toolset and methodology for capturing the requirements of the use case partners. The following research questions will guide the evaluation method, which includes the feedback received by the use case partners.

- Which MDE processes and tools, if any, does the development team use in current company practices (i.e., cloud development and deployment) and what particular benefits do these processes and tools offer?

- Do the current company's MDE processes and tools offer the capability for efficient cloud migration and re-deployment, as well as multi-cloud deployment and to what extent?

- What are the added benefits of using the advanced MDE tools and processes of PaaSage for effective and efficient cloud migration and re-deployment, as well as multi-cloud deployment?

- Are the provided PaaSage MDE tools and processes consistent with the current company practices and do they assist in improving the effectiveness and efficiency of cloud migration and re-deployment, as well as multi-cloud deployment?

In specific, the TAM model factors adopted and used for empirical data collection and analysis are based on the initial research work performed in the MPOWER project [3] and an extension of this work that added the *Perceived Compatibility* factor in the evaluation method that was followed in the context of the MODELPLEX project [4]. As supported in these research works, but also as emphasised by Creswell [6], qualitative research and evaluation is often exploratory in order to identify the important dependent and independent variables to examine. This is the main reason for adopting and extending these proven research variables proposed in the above variations of TAM, which will provide the capability to answer the generic research questions in regards to the current use and future use intentions of the PaaSage MDE components by project partners.

The PaaSage project evaluation framework extends the aforementioned research works by introducing a new factor, namely the *Use of Software Tools (TOOLS)* and adopting two of its corresponding sub-factors *Tools COVerage (TCOV) and Tools INTegration (TINT).* The TOOLS factor is part of the COCOMO II Software Cost Estimation Model [7], which is disaggregated and calibrated in [8] to include the aforementioned important sub-factors that aid the evaluation of development practices that are highly dependent on software development tools. Consequently, the TOOLS factor and its sub-factors are introduced in the TAM model adopted in this work in order to provide the capability to methodically address and evaluate the PaaSage project by taking into consideration a very important component, which is the MDE tool-chain developed in the project.

It is important to note that the *Tools Performance (TP)* considered in [3], [4] as a separate factor, is logically included in this work as a sub-factor of the TOOLS factor and denotes the main sub-factor that will enable quantitative evaluation using specific software metrics that will be selected when the final PaaSage platform prototype is available. In specific, software metrics of *availability and response time* can be useful respectively for measuring availability and how quickly specific user tasks can be executed. These metrics are used extensively to assess the efficiency and reliability of a software or a service and thus can be used in the context of the evaluation of the PaaSage platform. This is an initial consideration in terms of software metrics to be used for the quantitative evaluation, whereas the final list of software metrics will be decided at a later stage.

The extended and adopted TAM model is illustrated in **Figure 2.** In this model the defined dependent variables are: 1) *Current Use*: refers to the extent partners (e.g., industrial, research organisations, Universities) are currently using MDE and 2) *Future Use Intentions:* refers to the extent of adopting PaaSage MDE in their future work. These refer to the common variables considered in the TAM model, which are dependent on the specific factors selected that are respectively based on the evaluation objectives and the requirements of the PaaSage project. Note that in this project, the term *"solution"* used throughout the deliverable commonly refers to the PaaSage

methodology, the defined modelling languages and the developed MDE tools and software components. In specific, the study of existing works [3], [4] revealed, based on our reasoning and according to our views, that survey and spreadsheet(s) questions should be defined in a way that does not represent generic queries on MDE. These questions need to be tailored to the evaluation of MDE by focusing explicitly though on defining and answering explicit questions in regards to the added-value that the PaaSage MDE tools and methodology offer to use case partners.

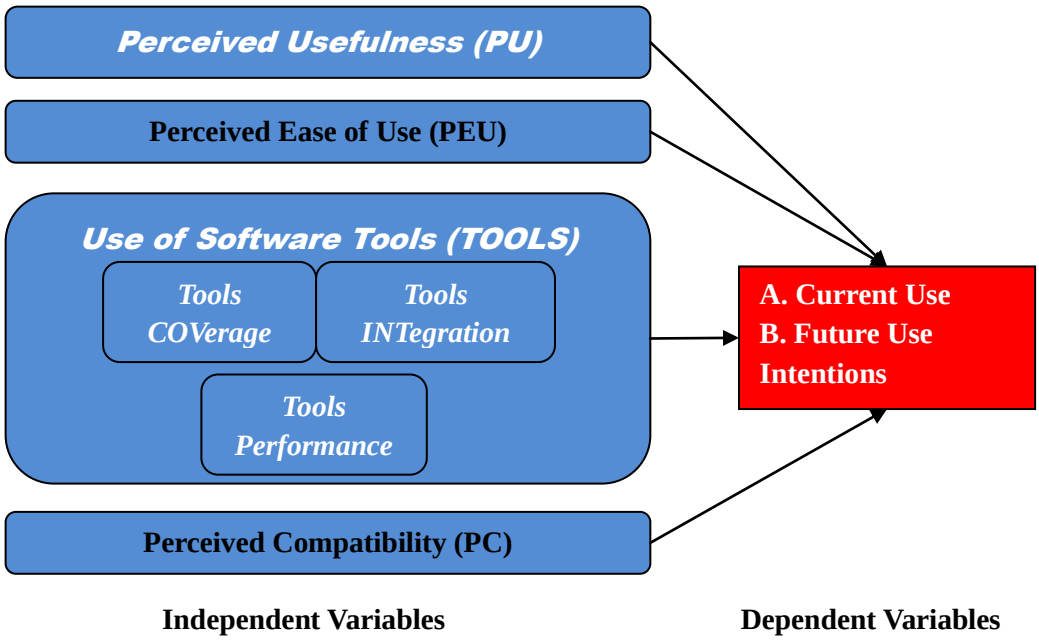

Figure 2: *The conceptual model of PaaSage project evaluation: An extended Technology Acceptance Model (TAM) model*

In terms of the PaaSage platform the main targets of the evaluation framework are reflected and will be assessed by *the TOOLS and the PU factors*. This is due to the fact that PaaSage is a pioneering research project that attempts for the very first time to build and deliver technically challenging tools for, e.g., cloud re-deployment and/or multi-cloud deployment. Hence, the main focus is to build and deliver software tools that are able to successfully accomplish the challenging technical tasks at hand. This can be evaluated with the use of the *TOOLS factor*. Following the same reasoning *the PU factor* will provide the evaluation means for assessing the added-value offered by the PaaSage platform's tools. In terms though of *the PEU and PC factors*, these do not reflect the direct goals of the PaaSage project, but reveal mostly future goals and ambitions. Thus, it is not yet definite if these goals can be met given the technically challenging nature of the software tools to be built, which requires heavy dedication of project's time and budgetary resources on providing capable tools for the tasks at hand, rather than easy to use and tools that are compatible with current development

practices. However, this deliverable captures and defines a comprehensive evaluation framework that considers also imminent goals for such a platform like PaaSage.

The complete list of evaluation factors and the corresponding hypotheses adopted in this work are defined as follows:

F1. Perceived Usefulness (PU) **–** degree by which a user believes that using the PaaSage MDE methodology and tools will enhance job performance.

- ***Hypothesis 1:*** *The perceived usefulness is positively associated with the current use and future use intentions of PaaSage MDE methodology and tools.*

F2. Perceived Ease of Use (PEU) **–** degree by which a person believes that using and applying the PaaSage MDE methodology and tools will be easy.

- ***Hypothesis 2:*** *The perceived ease of use is positively associated with the current use and future use intentions of PaaSage MDE methodology and tools.*

F3. ***Use of Software Tools (TOOLS)*** **–** degree to which PaaSage software tools cover development activities, are highly and consistently integrated, are mature enough to address the necessary tasks and perform to the expected level.

3.1. ***Tools COVerage (TCOV)*** **–** provides the capability to define and evaluate the coverage of activities undertaken in the software development process by the supporting tools (functional).

3.2. ***Tools INTegration (TINT)*** **–** allows defining and evaluating the degree of integration of the tools used throughout the process and the effectiveness in achieving this integration (functional).

3.3. ***Tools Performance (TP)*** **–** the degree to which the PaaSage tools efficiently support the development process and the tasks to be executed (will be assessed using quantitative metrics).

- ***Hypothesis 3:*** *The Use of Software Tools is positively associated with the current use and future use intentions of PaaSage MDE methodology and tools. H3.1: TCOV, H3.2: TINT, H3.3: TP.*

F4. ***Perceived Compatibility (PC)*** **–** degree to which the PaaSage solution is perceived as being consistent with existing values, principles, practices and the past experience of potential adopters.

As aforementioned, the evaluation framework to be used in the project aims to define survey(s) questions that are specific and more closely related to the PaaSage MDE methodology and tools. *However, in [3] and [4] mostly generic questions are defined concerning the current use and future use intentions in regards to evaluating MDE as*

a software development methodology, while additional questions attempt to examine the MDE tools developed in these projects. *According to our views, generic questions add little value to the evaluation.* Thus, *questions should be focusing and be tailored to the current use of PaaSage or similar software tools and processes, as well as the future use of PaaSage MDE methodology and tools*. In this way the extended TAM model will be used for collecting mainly developers but also business people perception and future intentions *concerning usage and adoption of PaaSage MDE methodology and tools, rather than MDE in general*.

1.5. Determining Outcome Indicators based on Research Questions and TAM Factors

Outcome Indicators (i.e., also referred in this project as evaluation criteria) are tools used to gauge the success of the project [4]. In particular, Outcome Indicators are used and enforced in this work to measure how well the project challenges are addressed, whether the objectives are accomplished and whether the intended results are reached. In order to identify and define these evaluation criteria we are adopting and using the approach called Methodology-Practices-Promises-Metrics (MPPM) presented in [9] and used in [4]. The approach defines the following:

1. ***Methodology*** is the subject of evaluation; such as MDE.
2. A ***Practice*** of a software development methodology (or technology or tool) is a new concept or technique or an improvement to established ones that is an essential part of the methodology and differentiates it from other methodologies. We may also call it a core practice.
3. A ***Promise*** is the expected improvement that is given as the main motivation for applying a practice. It is the expected benefit which often comes with a cost.

The core practices and promises of MDE are identified as follows [9]:

Practice	Promises and Costs
Models Everywhere: Models are primary software artefacts in all or most stages of software development. More effort will be spent in MDE on modelling and activities related to modelling such as defining modelling languages and quality verification of models than in traditional software development based on source code.	**Promises:** • Improved Communication amongst Stakeholders. • Improved Software Quality by using models for early analysis and testing. **Cost:** • Modelling tools must be integrated with other tools (e.g., configuration management tools).

Multiple Abstraction Levels and Separation of Concerns in Models: Abstraction and Separation of Concerns are the main techniques to handle complexity of software development.	**Promises:** • Improved Communication due to the separation of concerns. • Improved Software Quality since developers focus on one aspect of development at a time. • Portability of Solutions if models are defined as platform-independent. **Cost:** • Keeping models consistent with one another [10].
Generating Artefacts from Models: Generation of artefacts from models is the key technology to achieve automation and reduce manual work. Generation is done through transformations; either Model-to-Model (M2M) or Model-to-Text (M2T). During transformations, output models are supplied with information not present in the input models. An example of such information is the platform concept. Generation actually supports separation of concerns and adding details later; not by manual work but by applying transformations.	**Promises:** • Less manual work. • Consistency and traceability between artefacts. • Improving the quality of models and other artefacts such as their syntactic correctness and completeness. **Cost:** • Developing transformations.
Metamodeling: The concepts of metadata, OMG's Meta Object Facility (MOF) and the MOF-like Eclipse's metamodel (Ecore) allow definition of new modelling languages or extending the existing ones; for example as Domain Specific Languages (DSLs) or UML profiles.	**Promises:** • Sharing the same language between domain and IT experts and narrowing the gap between them. • Involving domain experts in all stages of design. • Defining relations between metamodels or instances of them. • Exchanging models between tools; thus achieving interoperability

	between tools. **Cost:** • Defining metamodels and supporting tools requires high initial investment and needs language and tool expertise.

Table 1: *MDE methodology: Practices, Promises and Costs*

The above view of the MDE as a methodology and the identified/defined practices and promises related to PaaSage, allow identifying and defining the PaaSage list of evaluation criteria. Studying the criteria defined in [9] we have concluded, adopted and defined also additional evaluation criteria that are in-line with the requirements and objectives of the project. The following tables present the complete list of evaluation criteria and the classification of them in accordance to the TAM factors, which enables better interpretation of the results obtained from the survey questions.

Evaluation Criteria	Description
Models Completeness	All information (e.g., configuration, deployment) gathered into the models (i.e., CloudML, Saloon, WS-Agreement and Scalability Rules Language) is complete and capture the details required to drive the PaaSage workflow.
Models Quality	The solution improves the quality of design (i.e., CloudML, Saloon, WS-Agreement, and Scalability Rules Language) by identifying and pinpointing to the modeller poor design or design errors.
Quality of the generated artefacts	The quality of configuration information, deployment information, code, documentation, etc. that are generated from models is acceptable (e.g., usable by PaaSage components).
Effort spent on software development	Development effort spent reflects design effort, coding effort, testing effort in comparison to using a non-MDE approach.
Appropriateness of Solution	The PaaSage solution is appropriate and suitable to solve the problem at hand (e.g., multi-cloud deployment).

Table 2: *Identified Evaluation Criteria for Perceived Usefulness (PU)*

Evaluation Criteria	Description
Learning Curve	The adequacy of resources provided (e.g., documentation completeness, availability of tutorials and/or training materials) and the difficulty faced in achieving an adequate proficiency level with a solution.
Models Understandability	The models are easy to understand by different stakeholders.
Effectiveness	Effort required by users to solve a task after learning how to use a solution.

Table 3: *Identified Evaluation Criteria for Perceived Ease of Use (PEU)*

Evaluation Criteria	Description
Provision of required modelling tools	The required modelling tools are developed to support fully the related design activities.
Provision of required validation tools	The required design validation tools are developed to support fully the related models validation activities.
Provision of required transformation tools	The required transformation tools are developed to support fully the related generation activities.
Integrated toolset	The various tools and components are comprehensively and effectively integrated to support the developer's tasks. For instance, no issues arise when using the generated output of a transformation tool by the corresponding component.
Integrated Toolset Maturity	The maturity of the integrated PaaSage toolset based on the completeness of the documentation and the technical support provided by an established community.
Extensibility of the developed toolset	The integrated software toolset can be easily extended by adding additional capabilities (e.g., MDE tools).

Table 4: *Identified Evaluation Criteria for Use of Software Tools (TOOLS)*

Evaluation Criteria	Description
Cost of adoption of the solution	The degree of complexity involved in setting up, configuring and customising the solution to apply the methodology.
Integration with other solutions	The degree of complexity involved in integrating the toolset with already available tools and development practices.
Standards Compliance	Conformance of the solution to the appropriate standards.

Table 5: *Identified Evaluation Criteria for Perceived Compatibility (PC)*

2. Evaluation Context

2.1. Methodology and Tools

The evaluation framework will be applied for examining the methodology and tools that will be defined and developed in the project. Hence, the defined survey questions will focus mainly on the evaluation of the key Cloud lifecycle phases adopted in the PaaSage model-based methodology and the developed PaaSage components/tools.

Based on the following methodology phases, survey questions will be defined accordingly, capturing the needs for evaluating each phase in terms of the selected Perceived Usefulness, Perceived Ease of Use, Use of Software Tools and Perceived Compatibility factors. Hence, it is currently considered by the consortium and if possible we will define specific survey questions that correspond to each PaaSage lifecycle phase, in order for the evaluation to be executed and the results to be obtained and analysed based on all phases of the whole PaaSage process.

1. **Configuration Phase:** Configuration is concerned with modelling the deployment of applications, profiling platforms and infrastructures, and specifying Quality of Service (QoS) requirements and data management policies.

2. **Deployment Phase:** Deployment is concerned with matching the Deployment Models of applications with the profiles of platforms and infrastructures based on negotiated SLAs and policies, and selecting one or more suitable Deployment Models.

3. **Execution Phase:** Execution is concerned with the management of the run-time execution of applications and monitoring / recording of KPIs based on SLAs and policies.

As aforesaid, the survey questionnaires will be defined to serve the core evaluation process against the two core TAM model factors (i.e., TOOLS, PU) and if possible consider and address also the two other TAM model factors (i.e., PEU, PC). In the case is deemed necessary by the consortium, the evaluation framework will define and provide additional "PaaSage Tool Assessment" spreadsheets (see Annex 1), which will serve and support the assessment of PaaSage tools/components when used by PaaSage developers and/or PaaSage implementers. Note also that, the selection of components to be evaluated is still under examination by the consortium partners and thus the complementary evaluation process is still to be approved and defined based on the final PaaSage platform prototype. At the moment it is considered that PaaSage implementers can evaluate the PaaSage components with which they interact during the realisation of a use case and/or PaaSage developers can evaluate all components of the PaaSage platform. This decision is still to be examined and confirmed.

In specific, the PaaSage components that can be assessed are defined as follows:

- **IDE/Cloud Modeller:** The IDE component is the start point at which the user/application designer engages with PaaSage. The IDE will support a variety of modelling languages, as explained below, for creation of applications for the PaaSage platform.

    - **CAMEL:** A set of domain-specific modelling languages that will form the CAMEL standard, which will focus on defining more precise constraints in the application design process. It is expected to be a group of DSLs including CloudML, Saloon, WS-Agreement and the Scalability Rules Language for constraint expression and management.

- **Profiler:** This component examines a list of user defined goals and preferences to come up with a list of probable cloud providers that satisfy the aforementioned inputs and other additional constraints like SLA and elasticity rules.

- **Reasoner:** The Reasoner receives application and context models (from the Profiler) in CAMEL format and outputs Deployment Models in CAMEL. This process relies on the Reasoner extracting requirements from the CAMEL and using the current state of Cloud Infrastructure and knowledge from the metadata data base to conduct reasoning.

- **Adapter:** The adapter has two main responsibilities. First, it is responsible for transforming the currently running application configuration into the target configuration in an efficient and consistent way. Second, it is responsible for performing high-level application management, which involves monitoring and adapting components deployed on multiple cloud providers.

- **Metadata Database (MDDB):** The MDDB is meant for long-term preservation of information on Cloud deployments. It is designed to associate mutations with a wall-clock timestamp and to trace the identity of the sources of mutations. It thus shares principles with archival systems, temporal databases, and provenance systems.

- **Executionware:** The main purpose of the modules and artefacts provided by the Executionware are to enable the execution of the individual components (services) of the PaaSage application in a fashion that the goals and constraints are met.

In addition to the PaaSage tools assessment, "PaaSage Pilot Assessment" spreadsheets (see Annex 2) will be defined if deemed necessary, which will serve and support the assessment of the PaaSage methodology and the MDE environment by PaaSage implementers in regards to the execution of each business use case pilot.

This complementary evaluation of the tools and the pilots will be based on weighing statements by means of a Likert scale, so as to execute the evaluation on the basis of the following non-functional dimensions:

1. **Dimension A:** Effectiveness – Usefulness of the service/component to the business.
2. **Dimension B:** Efficiency – Performance of the service/component.
3. **Dimension C:** Understandability/simplicity of the service/component.
4. **Dimension D:** Satisfaction & Attractiveness of the service/component.
5. **Dimension E:** Learnability, memorability of the service/component.
6. **Dimension F:** Use preparation & maintenance of the service/component.

The terms "service/component" refer to either the whole PaaSage process being offered as a service for the execution of a pilot or to a specific PaaSage tool being evaluated. Two example spreadsheets are presented in ***Annex 1: "PaaSage Tool Assesment"*** and ***Annex 2: "PaaSage Pilot Assesment"***. Different spreadsheets can be defined, based on these sample spreadsheets, for the different PaaSage tools and PaaSage pilots. The statements/assertions will be adapted and defined accordingly based on the PaaSage tool (e.g., adapter, profiler) or the PaaSage pilot (e.g., financial, e-Science) being assessed using a Likert scale (e.g., Disagree 1 – Agree 4). As aforesaid, the consortium examines the tools that will be assessed on the basis of the final PaaSage platform prototype and the way the user will interact with the platform (e.g., which tools will be used). There could be the case that the platform will be evaluated as a whole using this secondary approach.

Figure 3 presents part of the example spreadsheets. Note that the current statements on the spreadsheets presented in Annex 1 and 2, as well as on **Figure 3** are exemplary and need to be altered and defined accordingly for evaluating each PaaSage pilot and tool/component. Currently, they serve merely for demonstration purposes as part of this deliverable.

Evaluation of PaaSage Tool

Tool/service:

Persons/Organisations -date:

Give Assessment values for each statement in column C: 1: disagree, 2: slightly disagree, 3: slightly agree, 4: agree. If you are not able to give an assessment, sign "X " in column D. For each dimension, please,give your propositions for improvement.

Dimension A		Assessment Disagree 1 - 4 Agree	don't know X
Effectiveness -- usefullness to the network/company			
A1	It is easy to **understand** the objective and benefit of the service to my organization/network.	4	
A2	The **outcome of the service** is important / useful for the company/network. The service creates **value** for my company &network, for example by - saving costs - increasing income - saving time, accelerating processes - decreasing risks - improving quality	4	
A3	It is easy to **achieve** the planned business objectives / perform the tasks with the service.	3	
	Effectiveness	**3,666666667**	
A4	**What could be improved to make more value of the tool/service?**		

PaaSage Industrial Use Case (Public Sector) Pilot Assessment

Demo Scenario References

Demo Scenario		Actors involved:		Components involved:	
Contact Point		Supporting Material			
Short Description	e.g. Deploy application to multiple clouds				

Test Script

GOAL: e.g. Deploy the same application on two Cloud platforms

Non-Functional Evaluation

Dim.A (Effectiveness -- usefullness to the network/company)		Total	3,0
Ref #	Question		Answer (0-4)
A1	It is easy to **understand** the objective and benefit of the service to my organization/network.		3
A2	The **outcome of the service** is important / useful for the company/network. The service creates **value** for my company &network, for example by - saving costs - increasing income - saving time, accelerating processes - decreasing risks - improving quality		3
A3	It is easy to **achieve** the planned business objectives / perform the tasks with the service.		3
A4	**What could be improved to make more value of the tool/service?**		

Figure 3: *Part of the PaaSage Tool and Pilot Assessment spreadsheets (with mock values)*

The same process can be followed for creating all the assessment spreadsheets and adapting the statements/assertions based on the tool or pilot being evaluated, and in accordance to the assessed dimension. The results of the evaluation will be extracted and plotted in the form of a radar chart, as showcased in the example chart presented in **Figure 4**. A radar chart is a graphical evaluation method of displaying multivariate data in the form of a two-dimensional chart of three or more variables (i.e., dimensions) represented on axes starting from the same point. Also commonly known as spider charts, these charts are particularly useful when examining several factors that are related to one item [11]. The key benefits are the following [11]:

1. Provides insight to potential improvement opportunities.
2. Helps to understand what variables are dominant for a given process.
3. Helps to understand which observations are most similar (are there clusters of observations?)
4. Helps to understand if there are outliers.
5. Displays the performance metrics of any process and allows the viewer to see opportunities quickly.

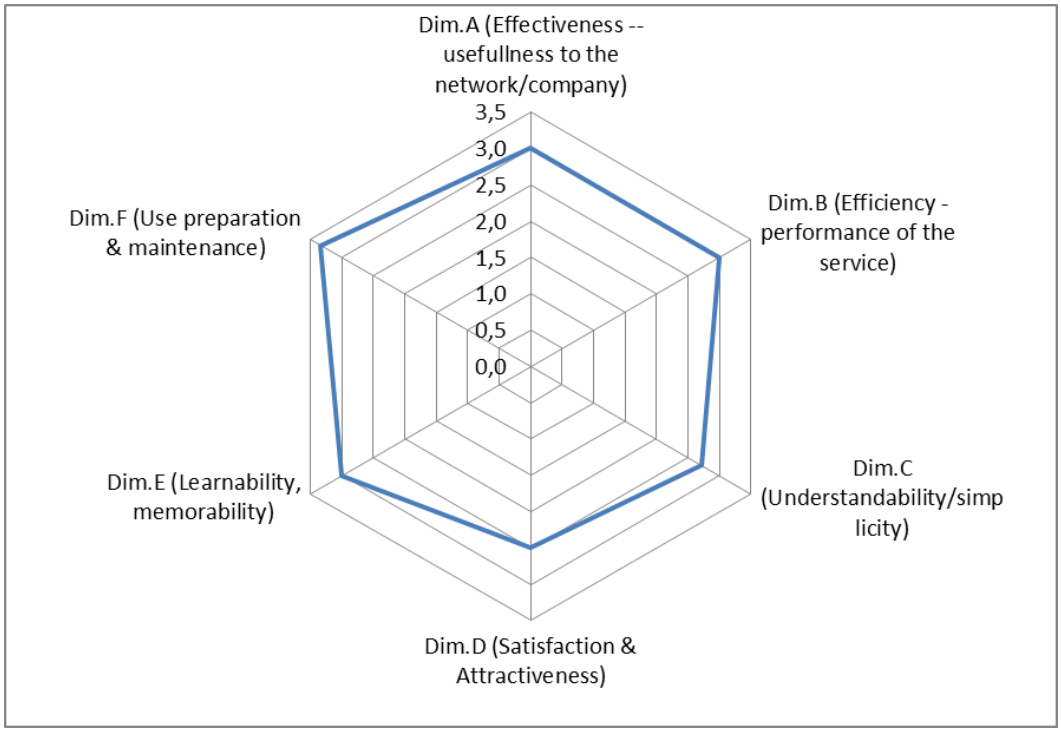

Figure 4: *An example Radar Chart of the PaaSage Tool Assesment results (with mock/example values)*

2.2. Data Collection and Analysis

The required feedback from the partners will be received with the help of online survey(s), which will be defined, distributed and completed by the project partners. The number of use case partners and developer partners that will be involved in the evaluation will be decided following the delivery of the PaaSage platform prototype. Note that, the involvement of evaluators outside the PaaSage consortium is also highly considered as part of the social network that will be formed via the MDDB tool and the selected community that will adopt PaaSage as an open-source solution. The involvement of external evaluators and the completion of the survey(s) by them can provide a neutral evaluation of the PaaSage solution.

The survey(s) will be defined using the *LimeSurvey open-source tool*[1] that will be setup and deployed on a dedicated web server for the needs of the project. The requests for survey participation will be sent via email or using the project's online collaboration platform and the results from the questionnaires will be *exported from LimeSurvey and imported into SPSS, Minitab or any other software tool for statistical analysis*. Besides, in case a final decision is made to execute the secondary evaluation process, the necessary *"PaaSage Tool Assesment" (Annex 1)* and the *"PaaSage Pilot Assesment" (Annex 2)* spreadsheets can be defined, distributed and completed by the project partners. These spreadsheets can be returned via the collaboration platform or via email, so as to perform the analysis of the results.

[1] http://www.limesurvey.org/en/

3. Conclusions

The evaluation framework will be applied for examining and evaluating the PaaSage methodology and tools defined in the project. In specific, the framework will follow the core evaluation process that relies on the extension and adaptation of the TAM model with the four key factors refined in this work. The core evaluation process will be performed in the form of questionnaires delivered and answered by the two main PaaSage partner roles: the components/tools developers and use case implementers, whereas the involvement of external evaluators outside the consortium is highly considered. A supplementary evaluation can be also performed for the appropriate components and the use case pilots via the use of assessment spreadsheets, which can be completed by components/tools developers and use case implementers. This can provide if needed an additional analysis and evaluation method that will focus only on the non-functional requirements. The analysis of the results will be performed using statistical analysis software such as SPSS or Minitab.

References

[1] P., Mohagheghi, V., Dehlen, "Where is the Proof? - A Review of Experiences from Applying MDE in Industry", In: European Conference on Model Driven Architecture Foundations and Applications (ECMDA 2008), LNCS vol. 5095, pp. 432—443, Springer (2008).

[2] E Rios, P H Meland, S Ardi, A Bagnato, J Jensen, W Mallouli, F Raiteri, T Sanchez, I A Tøndel, B Wehbi, "A qualitative evaluation of model-based security activities for software development", Security in Model-Driven Architecture 01/2009.

[3] S. Walderhaug, M. Mikalsen, I. Benc and S. Erlend, "Factors Affecting Developers' Use of MDSD in the Healthcare Domain: Evaluation from the MPOWER Project", In: From Code Centric to Model Centric Software Engineering: Practices, Implications and ROI. Workshop at European Conference on Model-Driven Architecture (2008).

[4] P. Mohagheghi, "An Approach for Empirical Evaluation of Model-Driven Engineering in Multiple Dimensions", SINTEF, Forskningsveien 1, Oslo, Norway, (2010).

[5] A, Dillon, "User Acceptance of Information Technology". In W. Karwowski (ed). Encyclopedia of Human Factors and Ergonomics. London: Taylor and Francis, 2001.

[6] JW Creswell, "Research design – qualitative, quantitative, and mixed method approaches", Sage Publications, 2002.

[7] Boehm, B., Abts, C., Clark, B., Devnani-Chulani, S., Horowitz, E., & Madachy, R. (2000). COCOMO 2 model definition manual, version 2.1. Center for Systems and Software Engineering, University of Southern California.

[8] Baik, J., Boehm, B., & Stecee, B. M. (2002), "Disaggregating and calibrating the CASE tool variable in COCOMO 2", IEEE Transactions on Software Engineering, 28(11), 1009–1022. doi:10.1109/TSE.2002.1049401.

[9] Mohagheghi, P., "Evaluating Software Development Methodologies Based on their Practices and Promises", In: New Trends in Software Methodologies, Tools and Techniques - Proceedings of the Seventh SoMeT, pp. 14—35 (2008).

[10] Mellor, S.J., Balcer, M.J., "Executable UML: A Foundation for Model-Driven Architecture", Addison-Wesley (2002).

[11] ASQ (American Society for Quality), Radar Chart, Available online: http://asq.org/service/body-of-knowledge/tools-radar-chart.