Seventh Framework Programme

CallFP7-ICT-2013-10

Project Acronym:	**S-CASE**
Grant Agreement N^{o}:	**610717**
Project Type:	**COLLABORATIVE PROJECT**
Project Full Title:	**Scaffolding Scalable Software Services**

D6.2 Evaluation framework definition and performance metrics

Nature:	**R**
Dissemination Level:	**PU**
Version #:	**1.0**
Date:	**27 February 2015**
WP number and Title:	**WP 6 – Evaluation (pilot case design and evaluation metrics)**
Deliverable Leader:	**ENG**
Author(s):	**Gabriele Giammatteo (ENG)** **Marina Stamatiadou (DELPHIS)** **Marin Orlic (ENT)**
Revision:	**Kyriakos Chatzidimitriou (AUTH)** **Robert Magnus (ATS)**
Status:	**Submitted** (Draft, Peer-Reviewed, Submitted, Approved)

Document History

Version[1]	Issue Date	Status[2]	Content and changes
0.1	6 February 2015	Draft	ToC
0.2	17 February 2015	Draft	Initial content by ENG
0.3	18 February 2015	Draft	DELPHIS and ENT contributions merged
0.4	19 February 2015	Draft	Internal review by DELPHIS
0.5	19 February 2015	Draft	Internal ENT review
0.6	20 February 2015	Draft	Final draft
0.7	25 February 2015	Peer-reviewed	Reviewers' comments accepted
1.0	26 February 2015	Submitted	Quality Board comments accepted

Peer Review History[3]

Version	Peer Review Date	Reviewed By
0.6	24 February 2015	Kyriakos Chatzidimitriou (AUTH)
0.6	24 February 2015	Robert Magnus (ATS)

This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no 610717.

[1] Please use a new number for each new version of the deliverable. Use "0.#" for Draft and Peer-Reviewed. "x.#" for Submitted and Approved", where x>=1.Add the date when this version was issued and list the items that have been added or changed.

[2] A deliverable can be in one of these stages: Draft, Peer-Reviewed, Submitted and Approved.

[3] Only for deliverables that have to be peer-reviewed

Table of contents

List of Tables

List of Figures

Abbreviations and Acronyms

Abbreviation	***Meaning***
#	Used in unit of measure for dimensionless quantity
CSF	Critical Success Factor
CMM	Capability Maturity Model
DoW	Description of Work
ETICS	E-infrastructure for Testing, Integration and Configuration of Software
FP	Function Point
GQM	Goals Questions Metrics
ITIL	Information Technology Infrastructure Library
KPI	Key Performance Indicator
(K) LOC	(Kilo) Lines of Code
WISE	Watchtower Intelligent Service Ecosystem

Executive Summary

This document defines the evaluation framework for S-CASE pilot cases. The framework has the main objective of assessing the performance of the S-CASE solutions applied to the development of the pilot cases in terms of reducing the development process time and costs.

As first step for the evaluation framework definition, the overall project objective of *minimising the software prototyping effort and required time* has been formalized and analysed in order to derive specific and measurable project goals: a) overall cost and time reduction, b) maximization of the S-CASE outcomes quality and c) maximization of the user acceptance.

The evaluation framework defines a set of twelve Key Performance Indicators (KPIs) linked to the project goals and applicable to all pilot cases plus some addition specific KPIs for each pilot case. The KPIs will provide the mean for evaluating the achievement of project goals against the targets value defined by the framework.

The KPIs will be calculated using formulas - defined by this framework - on the basis of the measures of a total of twenty-six different metrics that will be collected throughout the pilot cases development process.

Finally the evaluation framework also defines the tasks and involved actors of the evaluation procedure that will be executed twice for each pilot case during the project, in correspondence of the two S-CASE platform releases at M22 and M31.

1 Introduction

S-CASE project aims to provide software developers with a toolset for rapid prototyping of software based on user requirements. S-CASE automates the extraction of system specifications and architecture from the user requirements and the discovery and synthesis of workflows in order to implement the requested software functionality. Three pilot cases will be developed within the project scope using S-CASE in order to prove that the platform delivers what it promises and has a bright future in the real world applications environment.

Within the project, the task T6.2 "*Evaluation Framework definition and performance metrics*" has the main objectives of: i) identify and develop evaluation indicators and strategies to measure the effective improvements obtained using S-CASE and ii) provide specific performance metrics that will allow efficient and quantitative measurement of the introduced improvement.

These objectives are achieved with the definition of a formal evaluation framework defined in this document. The framework will be used during and after the realization of S-CASE pilot cases in order to measure the effectiveness and the efficiency of S-CASE toolset in improving the development process. The improvement will be compared against a baseline development process that is not using S-CASE.

For the definition of this framework we adopted an established and acknowledged methodology called Goal-Question-Metric (GQM). Following this methodology we have defined: a) the expected goals to be achieved by the S-CASE platform; b) the metrics that need to be collected in order to measure the development process of the pilots; c) the tools that will be employed for collecting the metrics; d) a set of Key Performance Indicators (KPIs) to assess the goal achievement; e) a procedure and a plan for the evaluation process.

The framework includes a common set of KPIs and metrics for all pilot cases focused on the improvement of the development process, as well as a set of KPIs and metrics specific to each pilot case focused on the requirements of the single pilot cases.

1.1 Intended audience

The document contains the definition of the evaluation framework of the pilots. It is expected to be read and used in the context of following tasks of S-CASE project:

- T6.4 "*Deployment of S-CASE framework for the development of the pilot and Performance monitoring*". The pilot responsible will refer to the document in order to know which metrics needs to be collected during the development process and which measurement tools needs to be set-up;
- T6.5 "*Pilot case evaluation and lessons learnt*". The pilot evaluators will refer to the document in order to known the steps and plan of the evaluation procedure and what and how needs to be evaluated.

1.2 Structure of the document

The remaining of this document is structured as follow:

- Section 2 *Defining a software measurement process* introduces the methodology and main concepts that have been employed in the definition of the evaluation framework;
- Section 3 *S-CASE Pilot Evaluation Framework* describes in detail the evaluation framework objectives, goals and scope. It also contains the application of the GQM method, the metrics and the common set of KPIs;
- Section 4 *Pilot Specific Metrics and Indicators* contains the definition of pilot-specific KPIs and metrics;
- Section 5 *Pilot Evaluation Procedure* describes how the evaluation of the pilots will be performed. The procedure, the actors and the plan are defined in this section;
- Section **Error! Reference source not found.** *Summary and Conclusions*: a conclusive section that summarize the content of the document.

2 Defining a software measurement process

In order to evaluate the successful application of S-CASE solutions, we are going to define a formal evaluation and measurement process to be followed by all pilot partners. The presence of an established evaluation framework will guarantee that all necessary metrics for measuring the achievement of project objectives will be correctly collected during the pilots' development, analysed and reported in a formal, homogenous and coherent way across the pilot evaluations.

Several process management and improvement standards like ITIL[4], CMM[5] and ISO9000[6] highlight that the definition of a measurement process must be closely linked with the business strategy and the critical success factors (CSF) of the company (or the project, in our case). Indeed, the measurement process is usually established with the main objective of pursuing the business objectives. From the business objectives, it is possible to derive the improvement goals and, in turn, the metrics and the indicators of the measurement process. The introduction of a measurement process also includes the definition of the procedures for collecting and interpreting the data and evaluating the achievement of the goals. Following this structure we can be sure that the measurement process defined will effectively support the realization of the project.

In S-CASE, we started from the overall project objectives deriving the metrics using the Goal-Question-Metric (GQM) method presented in the next section. The application of this method (section 3.2) allowed us to identify a set of metrics and Key Performance Indicators (KPIs) starting from a) a set of "questions" directly derived by the project objectives and b) the S-CASE project indicators.

2.1 Goal Question Metric method

The Goal Question Metric (GQM) [1][2] method has been proposed by Victor Basili of the University of Maryland in 1984 as a business-driven approach to the definition of a software development process improvement. The method contains four main phases described in the following paragraphs and depicted in Figure 1.

The **Planning phase** defines and characterizes the measurement process. The overall plan for the measurement is created in this phase.

The **Definition phase** starts from the identification of business-driven improvement goals. For each goal, a set of questions is generated to characterize the goal in a measurable way. Finally, for each question a set of metrics is identified in order to define what must be

[4] Information Technology Infrastructure Library (ITIL) is a set of practices for IT service management aligning IT service to business needs. TIL describes processes, procedures, tasks, and checklists to keep IT service aligned with the business strategy to deliver value. Since July 2013, ITIL has been owned by AXELOS Ltd [3].

[5] Capability Maturity Model (CMM) is an organization development model. The term "maturity" relates to the degree of formality and optimization of processes, from ad hoc practices, to formally defined steps, to managed result metrics, to active optimization of the processes.

[6] ISO 9000 is part of the family of quality management systems standards designed to help organizations to meet the needs of customers and other stakeholders while meeting statutory and regulatory requirements related to a product.

measured in order to answer that question. For instance, if the goal is to "*Improve the software quality*", a possible question is "*What is the defect density found in the code?*". The possible metrics needed for answering this question are: the *number of defects* and the *lines of code* (since the density of defects can be defined as the ratio between the number of defects and the code size). Finally, during the definition phase, a plan for data collection is created to define how and when data will be collected.

The **Data collection phase** is the phase in which measures for the identified metrics are collected. The outcome of this phase is a dataset containing all measures collected during the software development process.

The **Interpretation phase** takes the outcome of previous phase as input. The data collected is processed in order to answer the questions defined in the definition phase. These answers allow estimating the degree of achievement of the goals.

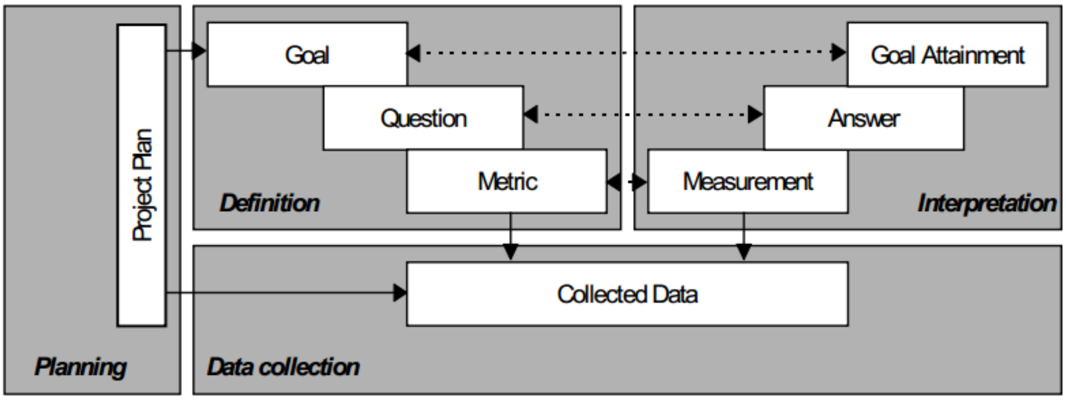

Figure 1 - The four phases of the GQM method

The measurement model defined by GQM is composed by three layers: the **Conceptual level** (goals), the **Operation level** (questions) and the **Quantitative level** (metrics).

The definition phase of the model follows a top-down approach: from the goals at the conceptual level, to the questions at the operation level and finally to the metrics at the quantitative level. Inversely, the interpretation of data collected follows an inverse, bottom-up approach: from measures that have been collected, to the answers to the questions, to the evaluation of the achievement of the goals. In Figure 2, the GQM levels and definition and interpretation approach are depicted.

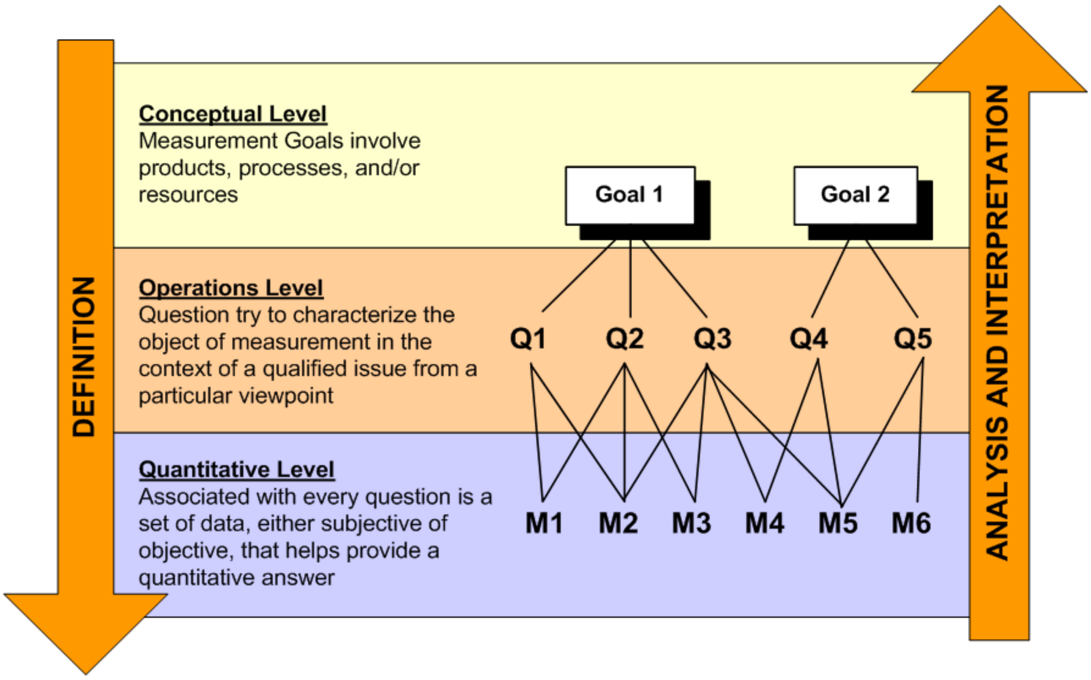
Conceptual Level
Measurement Goals involve products, processes, and/or resources
Goal 1
Goal 2
Operations Level
Question try to characterize the object of measurement in the context of a qualified issue from a particular viewpoint
Q1
Q2
Q3
Q4
Q5
Quantitative Level
Associated with every question is a set of data, either subjective of objective, that helps provide a quantitative answer
M1
M2
M3
M4
M5
M6
DEFINITION
ANALYSIS AND INTERPRETATION

Figure 2 - The GQM levels (Image source [4])

We decided to adopt the GQM methodology in our evaluation framework because it fits our needs and its phases fully cover the evaluation activities in WP6. For further information about the GQM method in literature, please refer to [1][2][5][6].

3 S-CASE Pilot Evaluation Framework

This section describes in details the evaluation framework, its objectives and goals, the metrics and the indicators. We also describe the process and the rationale that brought to its definition.

3.1 Objective, scope and applicability

The objective of the S-CASE Pilot Evaluation Framework defined in this document is the assessment of S-CASE through the evaluation of the performance of S-CASE toolset during the development process of the pilot cases.

The objective of the framework is not to verify and evaluate the quality of individual S-CASE modules. This objective is achieved in Work Package 1, Task T1.2 with the deliverable "D1.2 – S-CASE module verification and testing strategy" [7].

This evaluation framework will be mainly used at the end of the project in Task T6.5 "*Pilot case evaluation and lessons learnt*" when the realization of the pilot cases with the S-CASE toolset will be evaluated and reported. The definition of this framework will guarantee homogeneous, objective and complete evaluations across the various pilot cases.

The main focus of the pilot cases evaluations will be, as detailed in the next sections, in evaluating the improvement of performance in the development process in terms of costs and time when using the S-CASE toolset.

In order to measure such an improvement, a baseline is needed. Ideally, pilot owners should develop their software twice: the first time following their usual development process and the second time introducing S-CASE tools in the development process. The differences in the values of the metrics collected during the two executions indicate the improvement or the worsening of the process.

In S-CASE, only some parts of the pilot cases will be developed twice. For each pilot case, a subset of services required to be implemented by S-CASE will be selected for manual development in the early phases of the project when the S-CASE toolset is still being under development. These services will be used as test-cases to validate the S-CASE behaviour and correctness. The remaining service subset of each pilot will be developed only using S-CASE.

For the second subset, since it is not possible to have direct measures for manual development effort and time, we plan to estimate their value. The estimation will be based on measures collected for the first subset and/or the expertise of the development team and usual values for the development process adopted in the organization. In the evaluation report it will be clearly specified which metrics have been estimated (and not measured) and using which formula/criteria.

3.2 Identification of Goals

As described in section 2, the identification of the goals is the first task of a measurement process definition. Furthermore, the goals should be closely linked and derived from the business strategy and CSF. In our case, we decided to identify the goals from the objectives of the S-CASE project itself, clearly defined and explained in the S-CASE "Description of Work" (DoW) [8]. Of particular interest in our case is section B.1.1.3 *"The S-CASE Scientific* *and Technological Objectives"* that declares the objectives that the project has to realise to be considered successful. Most relevant statements for our purpose contained in that section are:

- "Minimise the cost and the time of software development, […]"
- "[…] software developers will focus only on the essentials complexity of the problem they are trying to solve, by minimizing the overhead that arises many times in software development from the implementation technology, tools and methods applied in a solution."
- "[…] S-CASE will allow for significantly reducing the time required for mapping system requirements to concrete technical design and specifications."
- "Improved and efficient software development process, measured in terms of development times and overall costs, by the use of appropriate indicators."
- "[…] information will be gathered in order to measure developers' acceptance on the introduced paradigm, based on their experience with and without the S-CASE realm."

Similar statements can be also found in section B.1.2.7 *"Measures and Indicators"* that lists the success indicators of S-CASE. Most relevant to our purpose are:

- "Software development process times and overall associated costs"
- "Acceptance by users"
- "Minimum number of extracted SE requirements supported (for each mode)"
- "User acceptance level of the extracted models"
- "User acceptance level of the retrieved workflows"
- "User acceptance level of the S-CASE tools for developers"
- "Software development process times and overall associated costs at each site and participant"
- "Extend to which the initial requirements are met by the provided solutions"
- "Percentage of involved developers with positive feedback"

From the analysis of these statements, we can clearly recognize that the most important goal for S-CASE is to reduce the overall cost and time of the development process and to improve the software project lifecycle.

The improvement of the software development process introduced by S-CASE will be accomplished by automatically transforming multimodal requirements into technical specifications and source code. Therefore, the success of the S-CASE toolset usage in a particular software project will strongly depend on the successful and complete processing of multimodal requirements given as an input to the S-CASE platform.

Finally, the user acceptance and satisfaction regarding the S-CASE outcomes are very important in the project and considered as a success factors in several parts of the DoW [8].

We can summarize the implied objectives of S-CASE in the following three goals:

G1) overall cost and time reduction;

G2) maximization of the S-CASE outcomes quality;

G3) maximization of the users acceptance;

In the next section we will use these three goals for deriving the indicators and the metrics that will be part of the evaluation framework.

3.3 Metrics definition and tools

We identified a set of metrics necessary to prove the successful achievement of the goals defined in section 3.2. Metrics have been identified by applying the GQM method (introduced in section 2.1). The application of the GQM method gave us an initial set of metrics that have thereafter been normalized, de-duplicated and formalized in order to derive the final and appropriate set of metrics for our evaluation framework. We also took into account the project objectives and KPIs as described in "Description of Work" (DoW) [8], section B1.2.7.

After applying the method a number of questions along with a corresponding set of initial metrics were produced and are presented in Table 1.

Table 1 - GQM Analysis

Goals	***Questions***	***Metrics***
G1 overall cost and time reduction	**Q1** What is the cost reduction achieved by using S-CASE?	• Overhead cost of using S-CASE toolset • Cost for completing S-CASE prototype • Cost without S-CASE
	Q2 How much is the time-to-market improved for a new or updated feature using S-CASE?	• Time-to-market for satisfying a requirement using S-CASE • Time-to-market for satisfying a requirement not using S-CASE
	Q3 How much effort is saved using S-CASE?	• Effort spent for using S-CASE • Effort spent for completing a software prototype using S-CASE • Total effort spent for completing a software prototype not using S-CASE
	Q4 How the S-CASE usage improves productivity?	• Total effort spent using S-CASE • Total effort spent not using S-CASE • Total size of the software using S-CASE • Total size of the software not using S-CASE

G2 maximization of the S-CASE outcomes quality	**Q5** How many requirements are successfully processed by S-CASE?	• Total number of requirements submitted to the system • Number of requirements successfully processed by S-CASE
	Q6 How many existing software solutions are suggested by S-CASE for a requested functionality?	• Number of existing software solutions processed and suggested
	Q7 What percentage of Commercial solutions is used by S-CASE?	• Number of Open Source software solutions used in the S-CASE registry • Number of Commercial software solutions used in the S-CASE registry
	Q8 What is the code size difference when using S-CASE?	• Size of the software using S-CASE • Size of the software not using S-CASE • Software Function Points[7] (FPs)
G3 maximization of the users acceptance	**Q9** How satisfied are the users of S-CASE with its outcomes (models, workflows, source code)?	• Level of satisfaction with software models generated by S-CASE • Level of satisfaction with relationships and workflows identified by S-CASE • Level of satisfaction with the source code generated by S-CASE
	Q10 How satisfied are the users with S-CASE toolset?	• Level of satisfaction using the S-CASE toolset (ease of use, adequacy, completeness, etc.) • Overall level of satisfaction using the S-CASE platform (suggested SE approach, outcomes, tools, etc.)

[7] Function Point is a unit of measurement of the software size (introduced at IBM in 1979). It is based on the amount of the business functionality that a system provides to the end user. It is the most used software sizing metric along with the Line of Codes (LOC). The Functional Point Analysis is the activity of calculating Function Points of an artefact by categorizing and evaluating the complexity of user requirements.

	Q11 What is the quality of S-CASE outcomes?	• Number of defects found in the models • Success rate of test cases execution

We formalized each metric obtained from the application of the GQM method defining a final set of metrics presented in Table 2.

In Table 2, ***Ref*** and ***Name*** columns are references to Table 1 and ***Sym*** is a unique symbol assigned to identify the metric (mainly used in formulas).

The ***Unit*** is one (or more) suggested units of measurement for the metric. When measures are collected, depending on the tool or the process, different units could be used. For instance for economic metrics we can use €, $, or £, while for code size we can use Kilo Lines of Code (KLOC) or Function Point (FP). The important aspect is that the units are always clearly indicated in the data collected and that all computations based on the values of the metrics have coherent units (i.e. dimensional analysis).

The ***Calculation*** field lists different possible methods for calculating metric values. If "*Direct measurement*" is specified, then the metric is a direct metric and can be measured directly (e.g. using a tool). If one (or more) formulas are specified, then it is an indirect metric and can be derived by applying the suggested formula on other direct or indirect metrics (formulas can contains references to other metrics defined in the same table). Finally, when "estimation" is specified, an estimation (according to section 3.1) is accepted, as the metric might not be available or cannot be measured.

Table 2 – Framework metrics

Ref	***Name***	***Sym***	***Unit***	***Description***
G1.Q1	Overhead cost of using S-CASE toolset	**Cs**	€	The economic cost of using the S-CASE toolset. Includes both the effort spent for getting engaged with the tools and other costs related to the software development using S-CASE (e.g. effort rates, 3rd party services). ***Calculation***: • Es * *effort using cost*[8] + *fixed costs*[9]*;*
	Cost for completing a software prototype using S-CASE	**Cd**	€	The economic cost for completing a software prototype generated by S-CASE. ***Calculation***:

[8] This constant represents the cost for each unit of effort (e.g. 1MM). Used to transform a given effort to cost.

[9] Additional costs that may occur that are not proportionally related to the effort (e.g. a fee to pay for using a tool/service).

				• Ed + *effort using cost*[8] *+ fixed costs*[9]*;*
	Total cost for developing a software prototype not using S-CASE	**Ce**	€	The economic cost of software development not using *S-CASE.* ***Calculation***: • Ee * *effort using cost*[8] *+ fixed costs*[9]*;*
	Total effort spent for developing a software prototype using S-CASE	**C**	€	The economic cost of software development not using *S-CASE.* ***Calculation***: • Cs + Cd
G1.Q2	Time-to-market not using S-CASE	**Te**	days	Workdays for completing a software prototype not using S-CASE ***Calculation***: • Direct measurement; • Estimation
	Time-to-market using S-CASE	**T**	days	Workdays for completing a software prototype using S-CASE. ***Calculation***: • Direct measurement; • Estimation;
G1.Q3	Effort spent for using S-CASE	**Es**	MM	The effort needed for getting engaged with the S-CASE toolset. ***Calculation***: • Direct measurement;
	Effort spent for completing a software prototype using S-CASE	**Ed**	MM	The development effort needed to complete the prototype generated by S-CASE. ***Calculation***: • Direct measurement;
	Total effort spent for developing a software prototype not using S-CASE	**Ee**	MM	The effort needed for developing the pilot without S-CASE in the development process. ***Calculation***: • Direct measurement;

				• Estimation;
	Total effort spent for developing a software prototype using S-CASE	**E**	MM	The total effort needed for developing the pilot using S-CASE in the development process. ***Calculation***: • Es + Ed
G1.Q4, G2.Q8	Total size of software not using S-CASE	**Se**	LOC	The total size of a software prototype developed not using S-CASE ***Calculation***: • Direct measurement; • Estimation
	Total size of software using S-CASE	**S**	LOC	The total size of a software prototype developed using S-CASE ***Calculation***: • Direct measurement;
G2.Q5	Total number of requirements submitted to the system	**R**	#	***Calculation***: • Direct measurement;
	Number of requirements processed by S-CASE	**Rs**	#	***Calculation***: • Direct measurement;
G2.Q6	Number of existing software solutions suggested to the user	**N**	#	The number of the existing solutions that S-CASE processes in order to provide a list of suggested ones to fulfil a given requirement. ***Calculation***: • Direct measurement;
G2.Q7	Number of Open Source software solutions in the S-CASE registry	**No**	#	***Calculation***: • Direct measurement;
	Number of Commercial software solutions in the S-CASE registry	**Nc**	#	***Calculation***: • Direct measurement;

G2.Q8	Software Function Points	**FP**	FP	***Calculation***: • Direct measurement;
G3.Q9	Total number of answers to the User Survey	**UT**	#	***Calculation***: • Direct measurement;
	Level of satisfaction with software models generated by S-CASE	**U1**	#	***Calculation***: • Direct measurement using User Survey (presented in Annex A): average answers to Q1;
	Level of satisfaction with relationships and workflows identified by S-CASE	**U2**	#	***Calculation***: • Direct measurement using User Survey (presented in Annex A): average answers to Q2;
	Level of satisfaction with source code generated by S-CASE	**U3**	#	***Calculation***: • Direct measurement using User Survey (presented in Annex A): average answers to Q3;
G3.Q10	Level of satisfaction using the S-CASE toolset (ease of use, adequacy, completeness, etc)	**U4**	#	***Calculation***: • Direct measurement using User Survey (presented in Annex A): average answers to Q4 and Q5;
	General level of satisfaction using the S-CASE platform (suggested SE approach, outcomes, tools, etc)	**U5**	#	***Calculation***: • Direct measurement using User Survey (presented in Annex A): average answers to Q6;
G3.Q11	Number of defects found in the models	**Dm**	#	The number of manual corrections in the generated .yaml file(s) made by the user due to erroneous output of S-CASE (applicable only for requirement-related fields) ***Calculation***: • Direct measurement;

	Success rate of test cases execution	**TCs**	#	The number of test cases (provided by the user) that pass against the total number of test cases ran for a given functionality. ***Calculation***: • Direct measurement;

Most of the metrics defined in Table 2 are defined as *direct metrics* and they will be measured during the development of pilot cases using appropriate tools. With regards to the tools for collecting metrics, we can recognize three different groups of metrics:

- *Source code* metrics (e.g. Cyclomatic Complexity, Software size): several tools exists for static analysis of the source code depending on the programming language. These tools are often integrated in automated continuous integration/testing tools (e.g. Jenkins, Bamboo) and/or development environments (e.g. Eclipse, Visual Studio). Some of the pilot owners could already have such tools established in their organization and, in this case, those tools could be used to collect the metrics;
- *User acceptance* metrics (e.g. Satisfaction with S-CASE workflow): a user survey will be designed, realized and distributed within the project members in order to collect user acceptance related metrics. These metrics are subjective from person to person and can not be collected using automated tools. A first version of the User Survey that will be used during the evaluations is presented in this document in Annex A;
- *Cost, time and effort* (e.g. effort spent) metrics: the measurement of these aspects is often done using managerial tools fed with data from developers (e.g. worked hours) and project managers (e.g. financial values). Also in this case, pilot case owners will probably have these tools already established in their organization;

The exact set of tools chosen for each evaluation will be decided during the development environment set-up phase (section 5.1) and reported in the final evaluation report.

3.4 Common Key Performance Indicators

KPIs are of essential importance in the pilot evaluation framework. In fact, the KPIs will be the quantitative indicators of the given goals' achievement.

The KPIs defined in this section are common for all pilot cases and will be used in the early and final evaluation. In addition to these KPIs, each pilot has defined a set of pilot-specific KPIs (presented in section 4) that will be used only for the pilot case to which they belong.

It is worth noting the one-to-one relationship of the KPIs defined here with the questions obtained from the GQM application. Indeed, the KPIs have been defined with the aim of providing quantitative answers to the questions given in the previous tables. This way, it will be possible to answer the questions and, therefore, evaluate the achievement of the goals in a formal and quantitative way. A similar relationship exists also between the KPIs and the project success indicators defined in section B1.2.7 of the S-CASE DoW [8]. When possible, these relationships have been exploited to set coherent targets for the KPIs.

It is also interesting to note that most of the KPIs are defined as a variation of a metric (e.g. cost, effort, productivity). This way we are able to capture the improvement or worsening of the quantity between the development process without and with S-CASE. All KPIs are summarized in Table 3.

Table 3 - Common KPIs summary

Id	Name	Unit	Target	Pilot
P0K01	Cost variation	%	<= -20%	All
P0K02	Time-to-market variation	%	<= -20%	All
P0K03	Effort variation	%	<= -20%	All
P0K04	Productivity variation	%	>= 10%	All
P0K05	Requirements coverage	%	>= 65%	All
P0K06	Existing solutions usage	%	>= 75%	All
P0K07	Commercial solutions usage	%	>= 30%	All
P0K08	Software size per FP variation	%	< 0%	All
P0K09	User satisfaction with S-CASE outcomes	%	>= 85%	All
P0K10	User satisfaction with S-CASE	%	>= 85%	All
P0K11	Defect density in S-CASE outcomes	#/KLOC	5/KLOC	All
P0K12	Testcases success rate	%	80%	All

The remaining of this section contains the definition of the KPIs which are common to all the pilot cases. For reading and layout convenience, each KPI is defined in a separate box containing the following fields:

- **Id**: a unique identifier of the KPI which will be used for reference. As convention the format is *P<pilot-sequential-number>K<kpi-sequential-number>* with "0" for KPIs common to all pilots;
- **Name**: human readable name of the KPI;
- **Formula:** the formula used to calculate the KPI. The formula normally contains one or more metrics and/or one or more constants;
- **Unit:** the unit in which the value of the KPI is expressed. It is worth noting that it must be coherent with the units used to measure the metrics and the formula;
- **Target:** it is the target value (or range of values) of the KPI to be able to achieve the goal;

- **Prj. ref:** a reference to the project indicators of interest/covered by the KPI. The numbers refers to the objective indicators as defined in section B1.2.7 of the project DoW [8];
- **GQM ref:** a reference to the questions of interest/covered by the KPI as defined in Table 1;
- **Notes:** optional. Description or notes;

Id	POK01
Name	**Cost variation**
Formula	$\frac{((Cs+Cd)-Ce)}{Ce}\cdot 100$
Unit	Percentage (%)
Target	Less than or equal to -20%
Prj. Ref	Objective 1
GQM Ref	G1.Q1
Notes	n/a

Id	POK02
Name	**Time-to-market variation**
Formula	$\frac{(T-Te)}{Te}\cdot 100$
Unit	Percentage (%)
Target	Less than or equal to -20%
Prj. Ref	Objective 1
GQM Ref	G1.Q2
Notes	n/a

Id	POK03

Name	**Effort variation**
Formula	$\frac{((Es+Ed)-Ee)}{Ee}\cdot 100$
Unit	Percentage (%)
Target	Less than or equal to -20%
Prj. Ref	Objective 1
GQM Ref	G1.Q3
Notes	n/a

Id	P0K04
Name	**Productivity variation**
Formula	$\frac{\left(\frac{S}{Es+Ed}-\frac{Se}{Ee}\right)}{\frac{Se}{Ee}}\cdot 100$
Unit	Percentage (%)
Target	Greater than or equal to 10%
Prj. Ref	Objective 1
GQM Ref	G1.Q4
Notes	n/a

Id	P0K05
Name	**Requirements coverage**
Formula	$\frac{Rs}{R}\cdot 100$
Unit	Percentage (%)
Target	Greater than or equals to 65%

Prj. Ref	Objective 6
GQM Ref	G2.Q5
Notes	n/a

Id	P0K06
Name	**Existing solutions usage**
Formula	$\frac{N}{Ntot}$
Unit	Percentage (%)
Target	Greater than or equal to 75%
Prj. Ref	Objective 5
GQM Ref	G2.Q6
Notes	*Ntot* is the total number of existing solutions requested by each of the pilot cases during the initial phase of the project

Id	P0K07
Name	**Commercial solutions usage**
Formula	$\frac{Nc}{Nc+No}\cdot 100$
Unit	Percentage (%)
Target	Greater than or equals to 30%
Prj. Ref	Objective 5
GQM Ref	G2.Q7
Notes	n/a

Id	P0K08
Name	**Software size per FP variation**

Formula	$\frac{\left(\frac{S}{FP}-\frac{Se}{FP}\right)}{\frac{Se}{FP}}\cdot 100$
Unit	Percentage (%)
Target	Less than < 0%
Prj. Ref	n/d
GQM Ref	G2.Q8
Notes	Generated software has a lower value of lines of codes required per Function Point than manually wrote code [9](pag. 673-675)[10]. We expect that code generated by S-CASE follows this pattern.

Id	P0K09
Name	**User satisfaction with S-CASE outcomes**
Formula	$\frac{U1+U2+U3}{3*mScore}\cdot 100$
Unit	Percentage (%)
Target	Greater than or equal to 85%
Prj. Ref	Objective 3, Objective 4
GQM Ref	G3.Q9
Notes	With reference to the User Survey (presented in Annex A), *mScore* is the maximum score that is possible to assign to an answer (e.g. 5).

Id	P0K10
Name	**User satisfaction with S-CASE**
Formula	$\frac{U4+U5}{2*mScore}\cdot 100$
Unit	Percentage (%)
Target	Greater than or equal to 85%

Prj. Ref	Objective 1, Objective 5, Objective 6
GQM Ref	G3.Q10
Notes	With reference to the User Survey (presented in Annex A), *mScore* is the maximum score that is possible to assign to an answer (e.g. 5).

Id	P0K11
Name	**Defect density in S-CASE outcomes**
Formula	$\frac{Dm}{S}$
Unit	#/KLOC
Target	Less than or equal to 5/KLOC
Prj. Ref	n/d
GQM Ref	G3.Q11
Notes	This KPI counts all defects in the S-CASE models and workflows normalized by the size of the project (in KLOC)

Id	P0K12
Name	**Test cases success rate**
Formula	$\frac{TCs}{Tot} \cdot 100$
Unit	Percentage (%)
Target	Greater than or equal to 80%
Prj. Ref	n/d
GQM Ref	G3.Q11
Notes	*Tot* is the total number of test cases executed

4 Pilot Specific Metrics and Indicators

The definition of metrics and indicators described in section 3.3 and 3.4 are common to all pilot cases and will be applied to all the evaluations. However, in order to capture the impact of the usage of S-CASE on some aspects specific of each pilot case, we decided to define some pilot-specific metrics and KPIs to be used in addition to the common ones.

Table 4 lists all the KPIs defined for all pilot cases, while a detailed definition is given in the following sub-sections.

Table 4 - Pilot-specific KPIs summary

Id	Name	Unit	Target	Pilot
PEK01	Per-operation integration effort	MM	<= 0.1 MM	ETICS
PEK02	Complexity of the integration code	#	<= 10	ETICS
PEK03	Performance overhead of the generated service	%	< 10%	ETICS
PWK01	User adoption	#	> 10	WISE
PWK02	End-user service complexity	%	<= 20%	WISE
PGK01	Business logic integration effort	MM	<= 0.05 MM	GiftCase
PGK02	Data access code size	LOC	<= 10	GiftCase
PGK02	Data access code complexity	#	<= 5	GiftCase

4.1 ETICS

The ETICS pilot case will implement a RESTful web service for managing virtual IaaS lifecycle. This service will be exploited by the ETICS system in order to manage on-demand virtual development and testing environments.

There are two main aspects of interest in this pilot not covered by the common KPIs: a) the effort for the integrating the new services developed by S-CASE with the rest of the ETICS platform and b) the variation in performance using the S-CASE generated services.

Controlling both aspects will guarantee that the benefits gained by the usage of S-CASE toolset during the development process, will not be nullified by the integration costs and/or the degradation of performance.

In the following three boxes, three specific KPIs for ETICS are defined. At the end of this section, the definition of the metrics used is provided.

Id	PEK01
Name	**Per-operation integration effort**
Formula	$\frac{EEi}{ops}$
Unit	MM
Target	Less than or equal to 0.1
Prj. Ref	n/a
GQM Ref	n/a
Notes	*For integrated operations, it is meant code level integration, with passing functional tests*

Id	PEK02
Name	**Complexity of the integration code**
Formula	ECC
Unit	#
Target	Less than or equal to 10
Prj. Ref	n/a
GQM Ref	n/a
Notes	Represents the Cyclomatic Complexity of the ETICS code written to interact with the generated services

Id	PEK03
Name	**Performance overhead of the generated service**
Formula	$\frac{ERTs - ERT}{ERT} \cdot 100$
Unit	Percentage (%)
Target	Less than or equal to 10%

Prj. Ref	n/a
GQM Ref	n/a
Notes	Measures the worsening of the performance when the functionality is realized with a service generated by S-CASE instead of manually integrated in the ETICS source code

Table 5 - ETICS-specific metrics

Name	***Sym***	***Unit***	***Description***
Integration effort	**EEi**	MM	Measures the effort spent for integrating the service generated by S-CASE in the ETICS platform ***Calculation***: • Direct measurement;
Number of operations in the service	**ops**	#	The number of operations in the service. ***Calculation***: • Direct measurement;
Complexity of the integration code	**ECC**	#	Measures the Cyclomatic Complexity[10] of the code written to integrate the service generated by S-CASE in the ETICS platform ***Calculation***: • Direct measurement;
Response time using S-CASE services	**ERTs**	s	Measures the response time of using the S-CASE generated services ***Calculation***: • Direct measurement (average should be applied);
Response time not using S-CASE	**ERT**	s	Measure the response time of requests integrating the functionality directly in the ETICS

[10] Cyclomatic complexity is software metric developed by Thomas McCabe in 1976 and is used to indicate the complexity of a program. It directly measures the number of linearly independent paths through a program's source code. Further information at http://en.wikipedia.org/wiki/Cyclomatic_complexity

service			code ***Calculation***: • Direct measurement (average should be applied);

4.2 GiftCase

GiftCase pilot case is a service for content personalization and purchasing gifts for other users of the service. The service uses user's data available at telecommunication operators is merged with publicly available third-party social network data (e.g. Facebook) in order to enable high personalization accuracy and allow users to purchase personalized gifts to their acquaintances. The coding effort and the code complexity required for accessing the user's data has an impact on the overall development process and for this reason, they become a KPI for this pilot case.

In the following three boxes, three specific KPIs for GiftCase are defined. At the end of this section, the definition of the metrics used is provided.

Id	PGK01
Name	**Business logic integration effort**
Formula	$\frac{GEi}{Nt}$
Unit	MM
Target	Lesser than or equal to 0.05
Prj. Ref	n/a
GQM Ref	n/a
Notes	The overall effort for getting required data items from other resources/services (e.g., social network, content providers, and telco services) when implementing a recommendation algorithm (or other business logic).

Id	PGK02
Name	**Data access code size**

Formula	$\frac{GS}{Nt}$
Unit	LOC
Target	Lesser than or equal to 10
Prj. Ref	n/a
GQM Ref	n/a
Notes	The average size of the code that needs to be written to fetch a data item from external web service resources.

Id	PGK03
Name	**Data access code complexity**
Formula	GCC
Unit	#
Target	Lesser than or equal to 5
Prj. Ref	n/a
GQM Ref	n/a
Notes	The average Cyclomatic Complexity of data access code that needs to be written to fetch a data item from external web service resources (measuring the related testing effort).

Table 6 - GiftCase-specific metrics

Name	***Sym***	***Unit***	***Description***
Integration effort	**GEi**	MM	Integration effort ***Calculation***: • Measurement;
Number of operations in the service	**Nt**	#	Number of the external resources ***Calculation***: • Measurement;

Code size	**GS**		Size of the data acquisition item code ***Calculation***: ● Measurement;
Complexity of the integration code	**GCC**	#	The Cyclomatic Complexity of the data access code ***Calculation***: ● Measurement;

4.3 WISE

Watchtower is a software platform hosting building intelligence algorithms, each for a very specific task. Its goal is to increase energy efficiency by detecting faults and anomalies, diagnosing their causes, notifying responsible parties and managing the notification tickets.

Similarly to the ETICS pilot, since the system is a software platform it is of great impact in the development process the complexity of integrating the service produced by S-CASE with the rest of the platform. The success (i.e. number of users) of the final end-user application is in this case a key indicator for the successfulness of the pilot.

In the following two boxes, two specific KPIs for WISE are defined. At the end of this section, the definition of the metrics used is provided.

Id	PWK01
Name	**User adoption**
Formula	$NUwise$
Unit	#
Target	Greater than or equal 10
Prj. Ref	n/a
GQM Ref	n/a
Notes	*NUwise* is the total number of developers that interact with the WISE PaaS to consume its services and integrate them into their own applications.

Id	PWK02
Name	**End-user service complexity density**
Formula	$\frac{Ns}{S} \cdot 100$
Unit	Percentage (%)
Target	Lesser than or equal to 20%
Prj. Ref	n/a
GQM Ref	n/a
Notes	The Cyclomatic Complexity density will be measured by using the appropriate tools. *S* is the service's LOC

Table 7 - WISE-specific metrics

Name	***Sym***	***Unit***	***Description***
WISE application users	**NUwise**	#	***Calculation***: • Direct measurement;
End-user service complexity	**Ns**	#	The Cyclomatic Complexity of the end-user service after the S-CASE outcomes have been integrated ***Calculation***: • Direct measurement;

5 Pilot Evaluation Procedure

In this section we will define all the steps that will be followed within the S-CASE project in order to collect the required data, analyse it and produce the final evaluation report.

For each pilot the procedure will be executed twice, corresponding to the two S-CASE toolset planned releases, accordingly to the plan presented in section 5.3. Each validation procedure strictly follows the steps described in section 5.1 by the actors described in section 5.2.

5.1 Procedure

The procedure hereby described applies to the evaluation of a single pilot case. It will be replicated for each pilot case. Furthermore, since some of the metrics are expected to be collected during the manual development of the pilot case (in order to have comparison terms), this procedure needs to be executed also during the first phase of manual development.

The procedure begins when the pilot case development starts (M19). In fact, while some of the metrics can be measured analysing the code after the end of the pilot development, other metrics should be collected and stored throughout the development process. For this reason what and how it needs to be collected must be clear before the pilot development starts.

The steps to be executed in the procedure are:

1. Involved actors read the most updated version of the evaluation framework (presented in this document). This is a required step in order to know which metrics need to be collected;
2. For each metric that needs to be directly measured, the development team sets-up and configures the required measurement tool(s) to measure the necessary attributes;
3. Once all the measurement tools are in place, the pilot development can start. No further actions are required until the end of the development;
4. At the end of the development process, final measurements are documented. The User Survey is circulated between the development team and answers are collected;
5. Data collected during the process is analysed by the evaluation team and calculation of all the formulas of the applicable KPIs takes place;
6. An evaluation report is produced by the evaluation team to report all the relevant data, including all values of the KPIs. The report will also contain an interpretation of the values from the project objectives point of view.

5.2 Actors

Two main actors are involved in the evaluation procedure:

- *Development team***:** this team is composed by all developers that will take part in the pilot case development. The team belongs to the pilot owner organization. The activities are:

  - Set-up and configure the measurement tools;
  - Develop the pilot case;
  - Answer the User Survey;
  - Send collected data to the evaluation team;
- *Evaluation team*: this team is in charge of all activities concerning the data analysis and reporting. While the knowledge of the development process and the software under analysis is a plus, it is not required that the evaluation team belongs to the pilot case owner organization. Main activities are:
  - distribute the User Survey;
  - receive collected data from the development team;
  - analysis data and calculate KPIs value;
  - write the evaluation report;

Within the S-CASE project, almost all Actors are considered to be the responsible pilot partners from DELPHIS, ENG and ENT, who are going to apply the proposed evaluation framework in their business context.

5.3 Plan and reporting

S-CASE project follows an iterative development process with two releases of S-CASE toolset: an early version at M22 (August 2015) and the final version at M31 (May 2016).

Concerning the pilot cases, after the full specification will be ready (M18) the pilot cases will be developed while the S-CASE platform is being developed also in order to have a quick feedback on tools correctness and functionality. The first development phase will end at M22. During the first development phase an early evaluation procedure will be executed. The results of the evaluation will be reported in an internal document that will be included in the final evaluation report.

The second pilot cases development phase will go from M31 to M32 using the final platform release (delivered at M31). A second complete evaluation procedure will run from M32 to M36. The final evaluation report will be included in D6.5 deliverable at M36.

Table 8 - Evaluation Plan

Task	***Start***	***End***
Evaluation Framework definition	M13	M16
Pilot Cases Specifications ready		*M18*
Early pilot cases development and deployment	M119	M22
Early platform release		*M22*
Early User Survey distribution and answers collection	M22	M24

Early evaluation reporting (internal document)	M22	M24
Final platform release		*M31*
Final pilot cases development and deployment	M31	M32
Final User Survey distribution and answers collection	M32	M34
Final evaluation reporting (D6.5)	M32	M36

Concerning the reporting format, since results from the application of the proposed evaluation framework are going to be documented in detail in D6.5, a report template was not considered as necessary within the context of Task 6.2.

6 Summary and Conclusions

In this document we defined an evaluation framework for the S-CASE project pilot cases development. This framework will be used during the development of S-CASE pilot cases in order to measure the effectiveness and the efficiency of S-CASE in improving the development process.

The definition of the framework followed the GQM methodology to guarantee that the performance metrics and KPIs defined are aligned with the overall project vision and objectives.

The resulting evaluation framework is composed by:

- a set of three clearly defined and measurable goals that the usage of S-CASE toolset aims to achieve;
- a set of twelve common KPIs that will be used to assess the improvement of the development process using the S-CASE toolset. The common set of KPIs is focused on a) the improvement of time and cost of the development process, b) the correctness and completeness of S-CASE toolset outcomes and c) the user satisfaction with the S-CASE toolset. For each KPI defined a target has been set based on the overall project performance indicators and/or S-CASE toolset specifications;
- a set of twenty-six common metrics that need to be collected during the development of the pilot cases in order to calculate the values for the KPIs. While some of the metrics need to be measured directly, other metrics can be indirectly calculated on the basis of other metrics and/or known constraints;
- a set of three pilot-specific KPIs (and five metrics) for the ETICS (owned by ENG) pilot case that will measure specific aspects of the pilot;
- a set of three specific KPIs (and four metrics) for the GiftCase (owned by ENT) pilot case that will measure specific aspects of the pilot;
- a set of two pilot-specific KPIs (and two metrics) for the WISE (owned by DELPHIS) pilot case that will measure specific aspects of the pilot;
- a procedure for the evaluation activity that details the activities to collect, analyse and report the data for the evaluation. The procedure also specifies the roles and the reporting format;
- an evaluation plan that will be followed to run the evaluation procedure;

The evaluation framework is based on S-CASE toolset specifications and the pilot cases descriptions as known at the moment of writing. Changes and refinement of this documentation might result in an adaption of the evaluation framework to the new specifications. Any change to the framework will be included in successive project deliverables.

References

[1] Basili, Victor R., and David M. Weiss. "*A methodology for collecting valid software engineering data.*" Software Engineering, IEEE Transactions on 6 (1984): 728-738

[2] Caldiera, Victor R. Basili-Gianluigi, and H. Dieter Rombach. *"Goal question metric paradigm."* Encyclopedia of Software Engineering 1 (1994): 528-532

[3] Axelos Global Best Practice *"ITIL® - IT Service Management."* https://www.axelos.com/itil

[4] Derek Huether "*How do you know your metrics are worth it.*" http://thecriticalpath.info/2010/02/11/how-do-you-know-your-metrics-are-worth-it/

[5] Van Solingen, Rini, and Egon Berghout. "*The Goal/Question/Metric Method: a practical guide for quality improvement of software development.*" Vol. 40. London: McGraw-Hill, 1999

[6] Koziolek, Heiko. "*Goal, question, metric.*" Dependability metrics. Springer Berlin Heidelberg, 2008. 39-42.

[7] S-CASE Project *"D1.2 – S-CASE module verification and testing strategy."*

[8] S-CASE Project "*S-CASE Annex I – Description of Work.*" Part B, 24 July 2013

[9] Pressman, Roger S. "*Software engineering: a practitioner's approach*." Palgrave Macmillan, 2005

[10] Quantitative Software Management (QSM) "*Function Points Languages Table*." http://www.qsm.com/resources/function-point-languages-table

A. Annex - User Survey Draft

This section contains a first draft of the User Survey that will be filled-in by S-CASE users in order to collect data concerning the subjective satisfaction of users of S-CASE toolset during the pilot development. It will be distributed among the S-CASE users involved in the pilot cases development.

Possibly, it will be implemented using web-based tools in order to facilitate the distribution of the survey, the collection of answers and the analysis of the data.

Finally, the following survey could be integrated in its final version with additional questions without the objective of collecting data for the evaluation framework, but of interest to measure other achievements of the project.

For each question answer with a level of satisfaction from 1 (low) to 5 (very high)

N°	**Question**	**1**	**2**	**3**	**4**	**5**
Q1	How are you satisfied with the models generated by S-CASE?					
Q2	How are you satisfied with the workflow generated by S-CASE?					
Q3	How are you satisfied with the source code generated by S-CASE?					
Q4	How are you satisfied with the S-CASE tools functionality?					
Q5	How are you satisfied with the S-CASE tools usability?					
Q6	What is the level of satisfaction with the S-CASE approach?					