Grant Agreement N° 215483

Title:	***Results of the initial Empirical Evaluation***
Authors:	*CITY, Lero, SZTAKI, Tilburg, TUW, UniDue, UniHH, USTUTT*
Editor:	*Angela Kounkou (CITY)*
Reviewers:	*Ita Richardson (Lero-UL)*
	George Baryannis (UoC)
Identifier:	*PO-IA-3.2.6*
Type:	*Deliverable*
Version:	*1*
Date:	*11 March 2010*
Status:	*Final*
Class:	*External*

Management Summary

This deliverable reports the results of empirical evaluation activities undertaken by S-Cube partners for the validation of research results. The systematic guidelines and templates used for the documentation of validation results and their related aspects are detailed. The initial set of collected validation results, documented according to these guidelines, is then presented. Finally, planned, upcoming evaluation activities are introduced.

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement n° 215483 (S-Cube).

Members of the S-Cube consortium:

University of Duisburg-Essen (Coordinator)	Germany
Tilburg University	Netherlands
City University London	U.K.
Consiglio Nazionale delle Ricerche	Italy
Center for Scientific and Technological Research	Italy
The French National Institute for Research in Computer Science and Control	France
Lero - The Irish Software Engineering Research Centre	Ireland
Politecnico di Milano	Italy
MTA SZTAKI – Computer and Automation Research Institute	Hungary
Vienna University of Technology	Austria
Université Claude Bernard Lyon	France
University of Crete	Greece
Universidad Politécnica de Madrid	Spain
University of Stuttgart	Germany
University of Hamburg	Germany
Vrije Universiteit Amsterdam	Netherlands

Published S-Cube documents

All public S-Cube deliverables are available from the S-Cube Web Portal at the following URL:

http://www.s-cube-network.eu/results/deliverables/

The S-Cube Deliverable Series

Vision and Objectives of S-Cube

The Software Services and Systems Network (S-Cube) will establish a unified, multidisciplinary, vibrant research community which will enable Europe to lead the software-services revolution, helping shape the software-service based Internet which is the backbone of our future interactive society.

By integrating diverse research communities, S-Cube intends to achieve world-wide scientific excellence in a field that is critical for European competitiveness. S-Cube will accomplish its aims by meeting the following objectives:

- Re-aligning, re-shaping and integrating research agendas of key European players from diverse research areas and by synthesizing and integrating diversified knowledge, thereby establishing a long-lasting foundation for steering research and for achieving innovation at the highest level.
- Inaugurating a Europe-wide common program of education and training for researchers and industry thereby creating a common culture that will have a profound impact on the future of the field.
- Establishing a pro-active mobility plan to enable cross-fertilisation and thereby fostering the integration of research communities and the establishment of a common software services research culture.
- Establishing trust relationships with industry via European Technology Platforms (specifically NESSI) to achieve a catalytic effect in shaping European research, strengthening industrial competitiveness and addressing main societal challenges.
- Defining a broader research vision and perspective that will shape the software-service based Internet of the future and will accelerate economic growth and improve the living conditions of European citizens.

S-Cube will produce an integrated research community of international reputation and acclaim that will help define the future shape of the field of software services which is of critical for European competitiveness. S-Cube will provide service engineering methodologies which facilitate the development, deployment and adjustment of sophisticated hybrid service-based systems that cannot be addressed with today’s limited software engineering approaches. S-Cube will further introduce an advanced training program for researchers and practitioners. Finally, S-Cube intends to bring strategic added value to European industry by using industry best-practice models and by implementing research results into pilot business cases and prototype systems.

S-Cube materials are available from URL: http://www.s-cube-network.eu/

Table of Contents

List of Figures

List of Tables

1 Introduction and Workpackage Vision

In this deliverable, we report the validation results of the Integrated Research Framework (IRF) elements. This section outlines the vision of the workpackage WP-IA-3.2 ; this vision is refined into a strategy in Section 2, which also describes the method used here. Section 3 applies the before-mentioned method to collect and document the validation results of the IRF elements. Section 4 reports on future validation efforts and Section 5 summarizes important findings in this deliverable.

According to the –Cube description of work[1] the workpackage WP-IA-3.2 has three major objectives:

- *Validation of the IRF*: the IRF should be validated as a whole with the help of scenarios, case studies and formal means. The goal of this validation is to improve the IRF.
- *Validation of the IRF building blocks*: research results produced in the joint research activities are the core building blocks of the IRF. These research results should be evaluated using standard research methods such as experiments, case studies, prototypes, demonstrators or formal proofs.
- *Customization of the IRF*: the IRF should be tailored according to the different classes of users defined in workpackage WP-IA-3.1.

These three objectives are presented in more details below.

1.1 Validation of the IRF

The validation goal of the entire IRF is to deliver a *consistent*, *complete*, *useful* and *communicated* IRF at the end of the S-Cube project.

By *consistent* we mean that the integrity of the IRF is ensured between all its elements. It also means that the research results can be integrated in a way to realize a system engineering scenario. Consequently, the inputs and outputs of the research results produced need to be “compatible”.

By *complete* we mean that there are no “disconnected” elements in the IRF. For instance, research challenges should always have related research questions at the end of the S-Cube project. In addition, there should be enough research results to realize a system engineering scenario; e. g. results should not be missing.

By *useful* we mean that the IRF should help the S-Cube partners internally to organize their work towards integration. In particular the IRF should support the six research workpackage leaders to plan, organize and supervise their work. In addition, the IRF should also be useful for conveying the S-Cube vision and its research results within the spreading of excellence activities. In other words the IRF should be useful as a marketing instrument.

By *communicated* we mean that the IRF should have had an impact for instance on academic or industrial research agendas.

1.2 Validation of the IRF Elements

At the end of the S-Cube project all major research results should be validated. This validation should possibly include the use of different validation methods (e. g. a formal proof and an experiment to validate an algorithm). Furthermore, for each research result produced by S-Cube its validation status should be known, e. g. it should be known which results were validated and in the positive case how

[1] Amendment #2; draft from the 5th of November 2009.

strong this validation was. This aspect may be important for upcoming projects when re-using the S-Cube results.

Since validation is a resource-intense activity, validation experiences and validation data gained during the execution of the project should be shared between S-Cube partners.

1.3 Customization of the IRF

Finally, the customization has the goal to implement the user patterns defined in task T-IA-3.1.3. The framework should be enriched with adaptation points, which allows tailoring it for different user types.

2 IRF Element Validation Strategy and Approach

We can derive two concrete tasks from the WP-IA-3.2 workpackage vision regarding the validation of the IRF elements:

- *Validation of the IRF elements*: the IRF elements should be validated. This validation should be organized in close collaboration with the six research workpackages. Since S-Cube involves partners from diverse backgrounds, both traditional empirical research methods such as experiments, field studies, case studies or action research and formal methods such as formal proofs, experimentation with prototypes and proof of concept demonstrators are used to validate research results.

  Given the fact that task T-IA-3.2.3 started in month 15, the work is organized in two phases. In the first phase, the existing validation results are collected. This collection is used in the deliverables containing the validation of the entire IRF to determine the validation status of the IRF. After this initial phase, workpackage WP-IA-3.2 actively triggers relevant validation activities.
- *Documentation of the validation results*: since the validation of research results is costly and largely depends on the experience of the researcher, the results achieved as well as the experiences gained with a particular validation method will be made available to all S-Cube members. To this end, the Integration Committee decided that the validation results become part of the integration framework. This decision implies that the validation results and the experiences gained will be made available on the S-Cube web portal via the IRF tool (cf. *Figure 1*). Because of this availability, experiences with validation methods can be shared among different S-Cube members.

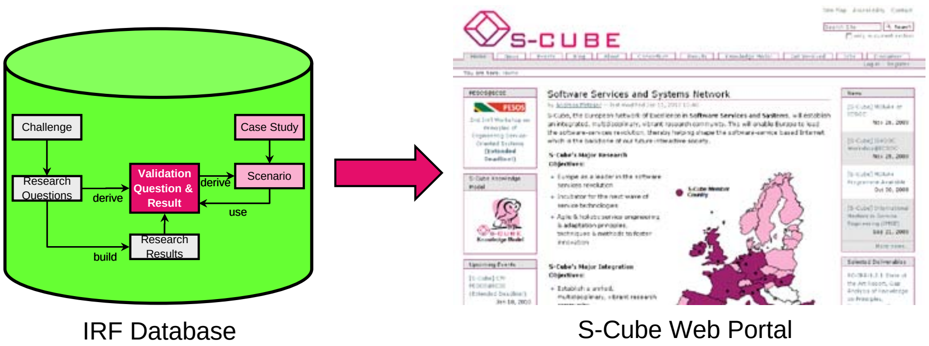


Figure 1: Publication of the Validation Results

The publication of the validation results in the IRF database implies that these results are documented in a uniform way. For this documentation we use the structure depicted in Table 1, which is a minor extension to the structure proposed in [1].

Table 1: Structure for Documenting Validation Results

Validation Set-up & Result	
Name	Name of the validation result.
Synopsis	Brief summary of the validation result (1 – 2 sentences).
Authors	List of authors of the validation result.
Research questions	Reference to research questions specified in the IRF related to the validation result.
Scenario	Scenario used in the validation and any related information (e. g. how the scenario is used in the validation exercise).
Research results	Reference to the research result validated. The research result must be part of the IRF.
Method	The empirical technique used for validation, i.e. one of: experiment, case study, field study, prototype, proof (see below).
Description	Short description of the validation result.
Goal	Description of the goal and objectives associated to the validation.
Set-up	Description of the set-up of the validation (e. g. settings - lab, organisations departments; the tools used - computational platform, technical specifications if appropriate etc.)
Inputs	Description of the materials (e. g., data) used in the validation.
Outputs	Description of the outputs (e. g., results) of the validation.
Outcome	One of: positive (the research result fulfils its goal), negative (the research result does not fulfil its goal).
Experiences	Comments and experiences on the validation (both positive and negative) gained performing the validation and that may inform a replication of the exercise
References	List of link to the paper and/or web page, in which this result was used.
Glossary	References to relevant terms in the knowledge model.
Keywords	List of keywords to facilitate search.

Since the terminology for the different research methods may differ in the various S-Cube research communities we use the following definitions in WP-IA-3.2 (cf. [3], p. 292 and [4]):

- *Controlled Experiment*: a controlled experiment is a research method carried out in a laboratory environment. It aims to test a hypothesis by manipulating its independent variables and measuring its dependent variables.
- *Case Study:* in a case study a single phenomenon is studied in a real-life context (e. g. in an organisation). The researcher doing a case study only observes the real-life context.
- *Field Study*: a field study is the broader version of a case study where multiple phenomena are studied in different real-life contexts (e. g. in different organisations). The researcher doing a field study only observes the real-life context.
- *Action Research:* in action research the researcher attempts to solve a real-world problem while simultaneously studying the experiences gained during the solving process. The researcher doing action research actively participates in the problem solving process.
- *Survey:* a survey uses a structured questionnaire to capture data from different individuals, e. g. by sending the questionnaire via mail to organisations or by using a web questionnaire.
- *Prototyping:* prototyping is used to realise some aspects of an envisioned system (or algorithm) in a demonstrator or prototype to show the feasibility of the approach.
- *Experiments with Prototypes:* existing prototypes may be used to carry out experiments demonstrating the superiority of an algorithm or system—especially in cases where a formal proof is not feasible.
- *Formal Proof:* a formal proof is a mathematical method to formally demonstrate that a (formal) system fulfils certain properties.
- *Proof of concept:* a proof of concept is evidence which demonstrates that the concept being proposed (e.g. an approach, an algorithm) is feasible and viable.

This list is not exhaustive and will be extended as the project proceeds.

2.1 Relation with other Integration Workpackages

For the overall strategy in WP-IA-3.2 it is important to understand the inputs and outputs needed and, therefore, to understand the relations and dependencies with the other integration workpackages. These dependencies are depicted in *Figure 2* and include:

- *WP-IA-3.1 – WP-IA-.3.2*: The most important relationship of WP-IA-3.2 is the one with WP-IA-3.1 since WP-IA-3.1 provides the main inputs to WP-IA-3.2 in form of the IRF and its research questions and research results, which are to be validated. In turn, WP-IA-3.2 provides the relevant materials in terms of validation results, which either become part of the IRF (validation of the IRF elements) or lead to an improvement of the IRF (validation of the entire IRF).
- *WP-IA-2.2 – WP-IA-3.2*: WP-IA-3.2 uses the industrial case studies from WP-IA-2.2 to derive validation scenarios. These validation scenarios are in turn used for extending/refining the industrial case studies and pilot cases (cf. [2]).
- *WP-IA-2.1 – WP-IA-3.2*: Once the first set of validation results is collected and the entire IRF is validated, the results will not only be used to enhance the IRF itself but also to stimulate work in the areas, which are not yet covered. One mean to achieve this coverage is to influence the mobility plan, which is developed in WP-IA-2.1.
- *WP-IA-1.1 – WP-IA-3.2*: The knowledge model provides the relevant glossary terms related to the validation results.

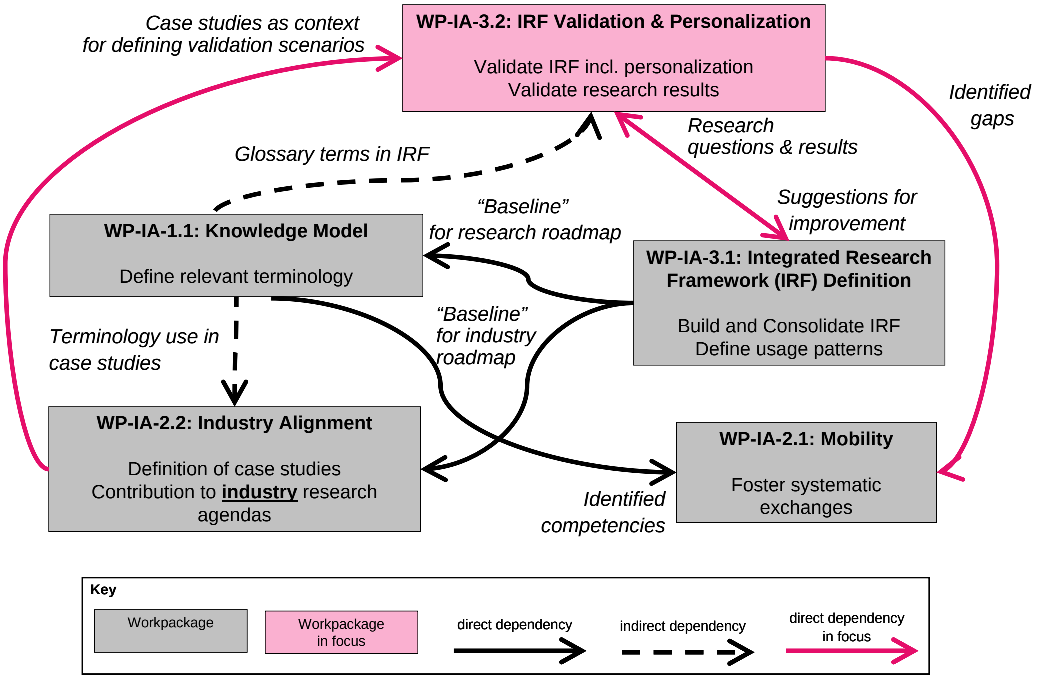


Figure 2: Relation between the Integration Workpackages

2.2 Roadmap and Timeline in IA-3

Since the validation object(s) such as the IRF and its element is produced in WP-IA-3.1 it is important to understand the intertwining with this workpackage as it has a direct impact on the timeline in WP-IA-3.2. This analysis has to consider two tasks in WP-IA-3.1, e. g. T-IA-3.1.2 and T-IA-3.1.3 and three tasks in WP-IA-3.2, e. g. T-IA-3.2.1b, T-IA-3.2.2 and T-IA-3.2.3. Regarding those tasks, we can distinguish between the following three types of dependencies (cf. the vision description in Section 1):

1) *IRF validation dependency*: the validation of the IRF is implemented in three different cycles. The first validation of the IRF (deliverable CD-IA-3.2.2; delivery date month 24) is based on the definition level of the IRF (deliverable CD-IA-3.1.3; delivery date month 21). The second

validation (deliverables PO-IA-3.2.3 and CD-IA-3.2.4, delivery date: month 36) is based on the consolidated and revised integration framework (deliverable CD-IA-3.1.5, delivery date: month 33). The third and final validation (CD-IA-3.2.5, delivery date: month 48) is based on the final consolidated version of the IRF (deliverable: CD-IA-3.1.7).

Therefore, the planned time for validating the IRF is three months in each cycle. The remaining time is needed to develop the validation method (such as the validation scenarios in PO-IA-3.2.3) and to influence the integration activities according to the outcome of the validation.

2) *IRF element validation dependency*: task T-IA-3.2.3 aims to validate all major research results. Since these validation results become part of the integration framework itself, they will be used for instance to determine the validation status of the IRF. Therefore, the validation results of the deliverables PO-IA-3.2.6 and PO-IA-3.2.7 become part of the IRF.

3) *Personalization dependency*: the second dependency between WP-IA-3.1 and WP-IA-3.2 regards the personalization of the IRF. For WP-IA-3.2 this basically means that the workpackage needs to implement the user patterns developed in T-IA-3.1.3 in the IRF, e. g. by extending the IRF with different user types. No additional deliverables are defined in WP-IA-3.2 and the personalization and customization results are reported in the same deliverables as the other validation results.

   Deliverable PO-IA-3.1.4 (delivery date: month 27) provides the relevant input for validating the defined user patterns and methodologies (documented in CD-IA-3.2.4, delivery date: month 36). The second and final validation of those user patterns (documented in CD-IA-3.2.5, delivery date: month 48) will be based on deliverable CD-IA-3.1.6 (delivery date: month 39).

The dependencies between WP-IA-3.1 and WP-IA-3.2 are visualized in *Figure 3*.

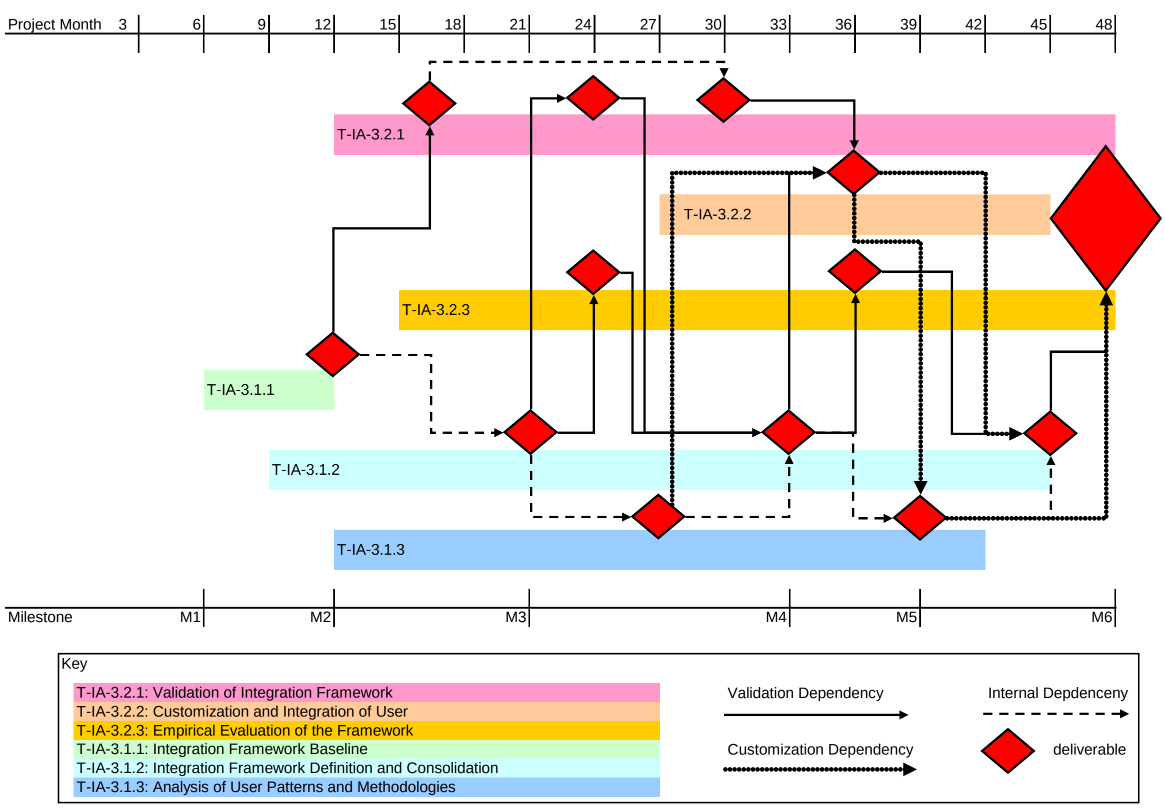


Figure 3: Intertwining between WP-IA-3.1 and WP-IA-3.2

Given the vision and strategy outlined before and the dependencies between the workpackages, the following timeline will be used for WP-IA-3.2 for the years 2–4:

- *Year 2*: Since the first version of the IRF was completed in month 21, the main focus in year 2 is on the consistency check of the IRF. In addition, the validation activities in year 2 will also concentrate on the gap analysis in order to provide input to the mobility program and to the JRAs to coordinate the research in years 3 and 4. Regarding task T-IA-3.2.3, the main focus is on collecting validation results, documenting them in a unique format and making them available via the IRF.
- *Year 3*: In the third year the verification activities will continue and will be extended by the scenario-driven verification. This scenario-driven verification will especially ensure the consistency of the different research results achieved. In addition, validation activities will also start in year 3. In addition, the validation status of the IRF elements will be analysed. This analysis may reveal gaps in the validation, which will in turn trigger validation activities. These validation activities will be executed in close collaboration with the two joint research activities.
- *Year 4*: In the final year the focus will be on external validation in close collaboration with WP-IA-2.2. (industry) and WP-SoE-1.2 (spread of excellence). The internal verification will be limited to those inconsistencies and gaps in the IRF, which will remain after the completion of the S-Cube project since this output cannot influence the S-Cube project anymore. Finally, Task T-IA-3.2.3 will deliver a report regarding the validation status of the IRF elements. This report will contain the elements, which are validated including the validation results. The report will also contain those elements, which were not yet validated during the S-Cube project.

3 Validation Results

The IRF conceptual model defines the elements composing the IRF and the relations among them. A detailed description of the IRF is available in CD-IA-3.1.3 [5]; overall, the IRF comprises eight components clustered in four interrelated blocks as outlined below (see also Figure 4).

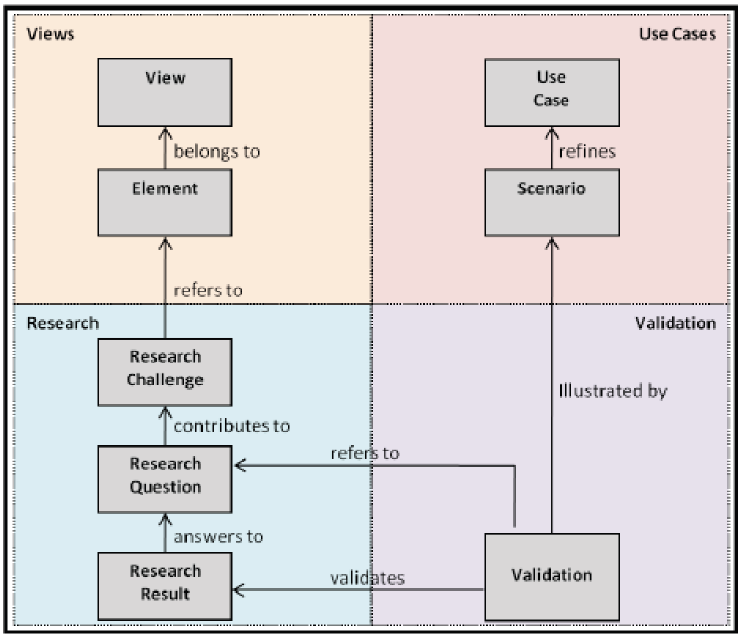


Figure 4: High-level Structure of the IRF Conceptual Model [5]

The IRF provides the basis for the validation results reported here. In particular validation results are reported as part of the IRF (see Section 3.2). Each validation result references a scenario, which in turn is related to industrial case studies. In addition, each validation result is linked to a research question describing the aim of the validation and to a research result describing the object of the validation.

Since the IRF at its definition level is a relatively new product of the S-Cube project, adding elements to the IRF such as missing research questions or missing scenarios was allowed during the preparation of this deliverable.

3.1 Collection Process

The documentation of validation results was performed following agreed guidelines; a template capturing relevant aspects of the validation exercise such as the validation goals, specific procedures carried out, and outcomes was also used (cf. Section 2). Each reported validation referred to at least one research question, one research result, and one scenario drawn from the industrial case studies specified in CD-IA-2.2.2 [2] as prescribed in the IRF conceptual model. Partners using different scenarios from those listed in PO-IA-3.2.1 (“Initial Definition of Validation Scenarios”) had the option to document new scenarios using the relevant templates specified in CD-IA-3.1.3; new research questions and research results could similarly be added to the latest version of the IRF.

3.2 Documented validation results

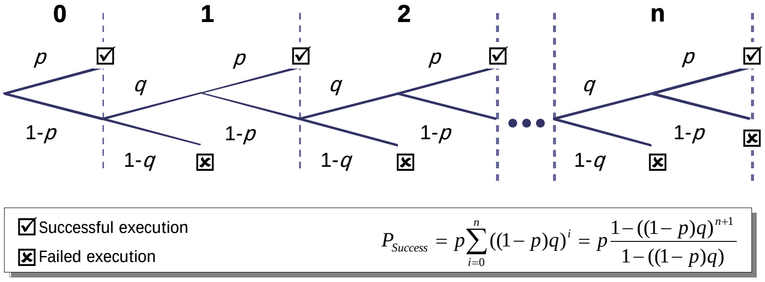
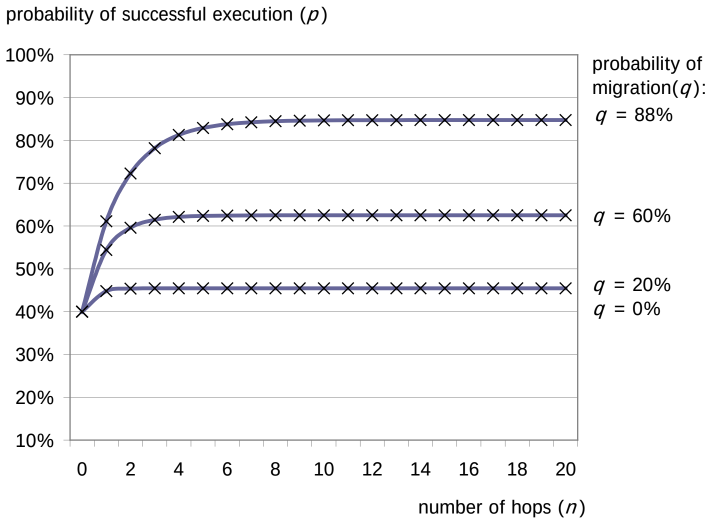
This chapter presents the set of initial empirical evaluations performed by S-Cube partners. The documented validation results are presented in turn.

3.2.1 Validation of process migration

This section documents two approaches to the validation of process migration techniques for the context-aware execution of distributed processes. Process migration is proposed to circumvent device limitations for the execution of complex tasks by allowing sub-tasks to be delegated to other systems; it is presented as a way to logically fragment processes by splitting the responsibilities for the execution of the process into subsets while still preserving the original structure of the process description. The authors report here that the probability of the successful execution of a service-based process can be enhanced by the use of process migration techniques. They use the Collaborative Transport Chain Control (Winery case study) as a validation scenario and use a stochastic model to produce a formal proof based on analytical assumptions (Table 2), which they confirm by practical experiments using a prototype implementation of a generic context model and process management system reported in Table 3.

Table 2: Validation of Process Migration by Formal Proofing

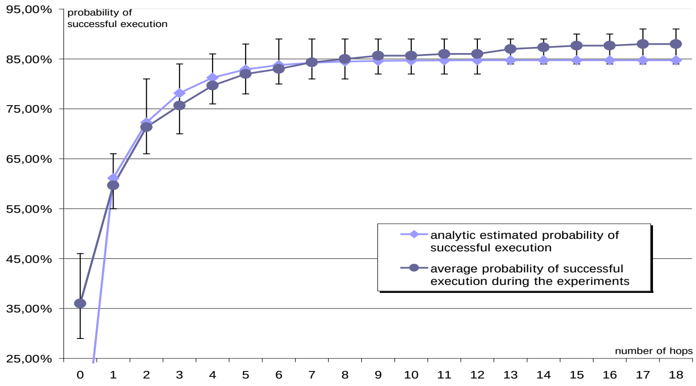
Validation Set-up & Result	
Name	Validation of process migration by formal proofing
Synopsis	The validation shows that the probability of the successful executing of service composition can be enhanced using process migration techniques. The result is derived by formal proofing with analytical assumptions.
Authors	Sonja Zaplata, Christian P. Kunze, Kristof Hamann, Winfried Lamersdorf.
Research questions	Context-Aware Execution of Distributed Processes.
Scenario	Collaborative Transport Chain Control (WINERY-S-1)
Research result	Process Runtime Migration Model.
Method	Formal proof
Description	To discuss the advantage of process migration, we examine a stochastic model. The successful execution probability for a migrating process can be calculated as a converging geometric series of the

	likelihood of successful sub-task execution anywhere in the mobile vicinity (cp. 5).
Goal	The goal of the validation is to show that the probability of the successful execution of a service-based process (~ service composition) in mobile environments can be enhanced by the concept of process migration.
Set-up	Formal proof with analytical assumptions..
Inputs	Stochastic model to calculate the probability of the successful execution of a process; p, the probability of a single device being capable of executing the current task; q, the probability of process migration; n, the number of hops caused by the migration.  Figure 5: Probability Tree of Successful Mobile Process Execution
Outputs	Some exemplary values calculated are presented in Figure 6, showing the probabilities of successful process execution with exemplary migration probabilities of q=0%, q=20% q=60% and q=88%, while p is assumed to be constantly equal to 40%. As one can see, the estimated probability of a successful execution increases considerably already after a few hops, especially if there is a high heterogeneity and thus a high migration probability.  Figure 6: Execution probability of process migration variants (p=40%)
Outcome	Positive.
Experiences	Because the formal proof is based on analytical assumptions, its result had to be confirmed by practical experiments. This is done in the "Validation of process migration by prototyping".
References	Sonja Zaplata, Christian P. Kunze and Winfried Lamersdorf. Context-based Cooperation in Mobile Business Environments: Managing the Distributed Execution of Mobile Processes. In Business and Information Systems Engineering (BISE), Vol. 2009(4), 2009.

	http://bibadmin.s-cube-network.eu/show.php?id=72
Glossary	Service Orchestration, Process Fragmentation, Context, Context-Awareness, Migration, Runtime Process Migration.
Keywords	transport logistics

Table 3: Validation of process migration by prototyping

Validation Set-up & Result	
Name	Validation of process migration by prototyping
Synopsis	The validation shows that the probability of the successful execution of service composition can be enhanced using process migration techniques. The result is derived by experimentation with a prototype on the DEMAC platform using a process with one activity and six devices.
Authors	Sonja Zaplata, Christian P. Kunze, Kristof Hamann, Winfried Lamersdorf.
Research questions	Context-Aware Execution of Distributed Processes.
Scenario	Collaborative Transport Chain Control (WINERY-S-1)
Research result	Process Runtime Migration Model.
Method	Prototyping and experiments with the Prototype
Description	We evaluated the applicability of the generic context model and process management system with a prototype implementation realized in the DEMAC (Distributed Environment for Mobility-Aware Computing) project (cp. Kunze et al. 2008, pp. 467-469). The evaluation includes an experiment to determine the probability of successful execution. To test the behaviour of the prototype under load, several test runs have to be carried out, each including 100 processes.
Goal	The goal of the validation is to show that the probability of the successful execution of a service-based process (~ service composition) in mobile environments can be enhanced by the concept of process migration.
Set-up	The prototypical evaluation was executed with a prototype of DEMAC (Distributed Environment for Mobility-Aware Computing) mobile process engine and context management system, a middleware which realizes the concept of process migration.
Inputs	DEMAC (Distributed Environment for Mobility-Aware Computing) platform Simple process with one single activity, six heterogeneous devices with two devices having the capability to execute the processes' activity, and four devices unable to do so.
Outputs	**Fehler! Verweisquelle konnte nicht gefunden werden.** shows the average number of hops resulting from migrations necessary to execute the process successfully compared to the expected analytical value. The analysis of the experiments further shows that only a few hops suffice to increase the probability of successful execution to levels more than twice as high. The estimated probability and the applicability of the presented concept can therefore also be confirmed by practical experimentation.

	 Figure 7: Results of the Experimental Evaluation (Kunze et al. 2008, p. 469)
Outcome	Positive.
Experiences	These results confirm the analytical assumptions made in the “Validation of process migration by formal proofing”.
References	Sonja Zaplata, Christian P. Kunze and Winfried Lamersdorf. Context-based Cooperation in Mobile Business Environments: Managing the Distributed Execution of Mobile Processes. In Business and Information Systems Engineering (BISE), Vol. 2009(4), 2009. http://bibadmin.s-cube-network.eu/show.php?id=72
Glossary	Service Orchestration, Process Fragmentation, Context, Context-Awareness, Migration, Runtime Process Migration
Keywords	DEMAC, mobile computing, transport logistics

3.2.2 Validation of a service virtualization architecture

Cloud-based services provide non-functional guarantees in the form of Service Level Agreements (SLAs), such as guarantees on execution time or price. However, due to system malfunctions, changing workload conditions and failures (hardware and software), established SLAs can be violated. In order to avoid costly SLA violations, flexible and adaptive SLA attainment strategies are needed. The authors propose an architecture for SLA-based service virtualization that provides an extensive solution for executing user applications in Clouds. The architecture is axed around three basic components: agreement negotiation, service brokering and deployment using virtualization, and proposed solution their incorporates enhancements of a meta-negotiation component for generic SLA management, a meta-brokering component for diverse broker management, and an automatic service deployment for service virtualization on the Cloud. This section documents the authors’ use of an automotive case study to demonstrate the feasibility and advantages of their approach for achieving reliable service operations.

Table 4: SLA based autonomous and adaptive service virtualization

Validation Set-up & Result	
Name	SLA based autonomous and adaptive service virtualization
Synopsis	We validate service virtualization architecture through an industrial car-manufacturing scenario, highlighting the cases where the autonomous properties of the components of the architecture were activated in order to cope with failures during service executions.
Authors	Attila Kertesz, Gabor Kecskemeti, Ivona Brandic
Research questions	On-demand, dynamic service provisioning Self-optimization and self-healing of a single service Monitoring and adaptation approaches that support the creation and sustainable usage of autonomous components covering the full

	lifecycle of a SBA
Scenario	Autonomic arrangements of car assembly testing services with virtualized service execution environments (SZTAKI_AUTONOMIC_CAR)
Research result	Autonomic Resource Virtualization in Cloud-like Environments SLA-based resource virtualization approach for on-demand service provision
Method	Proof of concept
Description	We argue and demonstrate that the combination of negotiation, brokering and deployment using SLA-aware extensions and autonomic computing principles are required for achieving reliable service operation in Cloud-like environments.
Goal	To present the feasibility of the service virtualization architecture.
Set-up	Alignment with the case studies and scenarios used.
Inputs	Automotive industrial case study, SZTAKI_AUTONOMIC_CAR scenario, the service virtualization architecture proposed by previously mentioned research results.
Outputs	Highlights the advantages of using the proposed architecture in the Automotive industrial case study. Offers initial answers to the related research questions.
Outcome	Positive
Experiences	N/A
References	Attila Kertesz, Gabor Kecskemeti, and Ivona Brandic, Managing the Clouds: A Self-adaptive approach for SLA-based Service Virtualization, Submitted to Journal of IEEE TSE, Nov. 2009.
Glossary	Service Deployment, Service Level Agreement Negotiation, Self-Adaptation, Service Orchestration
Keywords	Virtualization, Cloud computing, SLA, negotiation, service, brokering, deployment, autonomous service

3.2.3 Validation of the influential factor analysis of business process performance

Business activity monitoring enables the observation of key performance indicators (KPIs), and provides information on their achievement; currently, it does not however support a deep analysis of the factors contributing to the violation of KPI target values. This section documents the validation of a machine learning technique (decision tree learning) for the analysis of influential factors of KPIs and SLA violations. Decision trees are used here to construct a tree of decision nodes each consisting of a test, with the leaf nodes representing a classification to a category (i.e. whether a KPI has been violated or not). The authors implemented the Purchase Order Processing scenario to run a validation experiment during which influential metrics were evaluated and dependency trees were generated for analysis. They reported a positive outcome of the validation exercise indicating that the generated decision trees were consistent with the expected influential factors (Table 5). In Table 6, the authors report on another prototype-based experiment demonstrating that the effort spend for the run time prediction SLA violations based on machine learning techniques is acceptable for large numbers of monitored services.

Table 5: Validation of the Influential Factor Analysis of Business Process Performance

Validation Set-up & Result	
Name	Validation of the Influential Factor Analysis of Business Process Performance
Synopsis	Experiments show that the generated decision trees present the expected influential metrics in a satisfactory manner. As expected, the decision tree algorithms make dependency analysis produce suitable

	results "out of the box" making the approach suitable for non-IT personnel.
Authors	Branimir Wetzstein, Philipp Leitner
Research questions	Analysis of Influential Factors of KPIs and SLA Violations Based on Machine Learning techniques
Scenario	Automotive Purchase Order Processing Scenario (Purchase_Order_Processing_BPM)
Research result	Monitoring and Analyzing Influential Factors of Business Process Performance
Method	Experiment (based on a prototype)
Description	See set-up and inputs
Goal	The goal of the experiment is to analyze whether the generated decision trees show the expected influential factors.
Set-up	The purchase order processing scenario has been implemented in WS-BPEL (Apache ODE). A set of Java-based Web services (used in the process) are configurable to simulate certain behaviour of influential factors. One can, for example, configure the response time, availability, and outputs of services over time and dependant on business process data. The process is then run a certain number of times (100, 400, and 1000 times) using a test client (which triggers the execution of process instances). During process execution, the metrics are evaluated and the dependency trees are generated. The results (the influential metrics shown in the tree) are compared with the expected configured outcome.
Inputs	We have performed two experimental runs (see Reference below for more details). The configuration consists of the KPI Order Fulfillment Lead Time and a set of 31 potential influential factors. For the first run, we have created a configuration from which we expect the KPI to be mainly influenced by order in stock, product type, supplier 1 delivery time, and shipment delivery time. Other metrics (in particular response times of services) also influence the KPI value, but in a marginal way and should not be shown in the tree.
Outputs	The generated decision tree shown in the next Figure below has been generated using J48 (implementation of C4.5) based on 100 process instances. The most influential factor is the shipment delivery time; if it is above 95 time units all process instances lead to KPI violations ("red"), otherwise they depend further on the order in stock metric and supplier 1 delivery time. The leaves of the tree show the number of instances which are classified as "red" or "green".

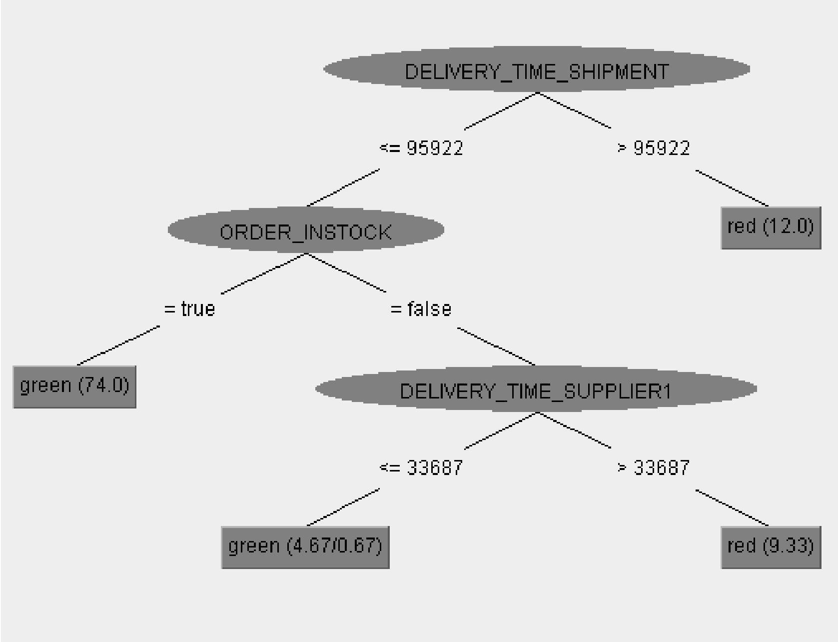
<table>
<tr><td></td><td>


Figure 8: Generated Dependency Tree

The dependency tree shows three of the four influential factors we have configured. Interestingly, the fourth factor, the product type, is not shown. The reason for this is that product type directly influences order in stock, which again influences the KPI value which is shown in the tree; as both metrics influence the KPI value in the same way, only one of them is shown in the tree. This problem can be dealt with by either drilling-down or by removing metrics from the potential influential metric set. For more details see Experiences below or the corresponding Reference.
More details on the size of the tree and performance of different algorithms can be also found in the Reference below.
</td></tr>
<tr><td>Outcome</td><td>Positive</td></tr>
<tr><td>Experiences</td><td>Concerning the influential factors displayed in the tree, we have identified two problems: (i) as the tree gets bigger it contains often more metrics than expected, i.e. metrics which have only marginal influence and thus only “blur the picture”; in that case one can try to tune the algorithm by using, for example, reduced error pruning, or one can simply remove those metrics from the analyzed metric set and repeat the analysis; both techniques lead to more satisfactory results; (ii) the tree does not show some of the expected metrics: we have shown that this is often the case when there are “multi-level” dependencies between metrics; in that case further analysis (drill down) of lower-level metrics may help to find further influential factors.</td></tr>
<tr><td>References</td><td>Branimir Wetzstein, Philipp Leitner, Florian Rosenberg, Ivona Brandic, Frank Leymann and Schahram Dustdar. Monitoring and Analyzing Influential Factors of Business Process Performance. In Proceedings of the 13th IEEE International Enterprise Distributed Object Computing Conference (EDOC'09), Auckland, New Zealand, 2009.</td></tr>
<tr><td>Glossary</td><td>Key Performance Indicator, Business Process, Service Level Agreement</td></tr>
<tr><td>Keywords</td><td>Key Performance Indicator, Service Level Agreement</td></tr>
</table>

Table 6: Runtime Prediction of Service Level Agreement Violations for Composite Services

Validation Set-up & Result	
Name	Runtime Prediction of Service Level Agreement Violations for Composite Services
Synopsis	Experiments show that the prediction of SLA violations at different checkpoints can be integrated successfully in a process including composite services. The effort for the teaching process of the prediction logic is tolerable on a current system setup. Prediction errors decrease rapidly towards the end of a process.
Authors	Philipp Leitner, Branimir Wetzstein, Florian Rosenberg, Anton Michlmayr, Schahram Dustdar, Frank Leymann
Research questions	Runtime Prediction of KPIs and SLA Violations Based on Machine Learning Techniques
Scenario	Automotive Purchase Order Processing Scenario (Purchase_Order_Processing_BPM)
Research result	Monitoring and Analyzing Influential Factors of Business Process Performance
Method	Experiment (based on a prototype)
Description	See set-up and input
Goal	The goal of the experiment is to show that the effort spend for runtime prediction of SLAs is acceptable even for large numbers of monitored services.
Set-up	We built a prototype prediction tool in the Java programming language. Our core implementation is based on our earlier work on event-based monitoring and analysis (see Validation Set-up & Result of Validation of the Influential Factor Analysis of Business Process Performance). Data persistence is provided using a dedicated database (MySQL database and Hibernate combination). QoS values are gathered by events (Apache ActiveMQ9 as JMS middleware), and client and server-side (VRESCo) monitoring. Finally, the open-source machine learning toolkit WEKA is used to build prediction models. The valuation process is run by the WS-BPEL engine Apache ODE, with support for BPEL lifecycle events which are essential for the experiment. Apache CXF10 hosts the necessary base services which are used in the experiment (e.g., supplier services, banking service, stock service) At first, the prediction toolkit is taught with historical instances of the process. Then 100 new random executions are set-up and at the end the results are evaluated. This includes cost for teaching activity and prediction overhead as well as Mean Prediction Error (MPE) and Error Standard Derivation (ESD).
Inputs	The inputs for the prediction toolkit require historical data from previous processes. The execution sequence of the newly triggered processes is random. That is, current QoS values and checkpoints (denoted by C) are known to the evaluator to calculate the occurring deviations.
Outputs	For the teaching process the output is the time spent for a training session and additional computing cost in comparison to the number of historical instances. The results show that both training measurements depend on the number of training instances available. The time necessary for building the knowledge model depends linearly on the number of historical instances available. However measurements demonstrate, even for e.g.,

1000 instances the absolute rebuilding time is below 32 seconds, which seems acceptable for practice, considering that model rebuilding can be done sporadically and online. The overhead necessary for actual online prediction is constant and rather small (well below 1 second), which seems very acceptable for prediction at run-time.
Deviations are calculated with the prediction output and the correct status value at the different checkpoints. The next figure shows the process with checkpoints (C) and derived results. C1 is located directly after the order is received, C2 after the internal warehouse is checked, C3 after eventual orders from external suppliers have been carried out, C4 during the payment and shipment process, and finally C5 when the execution is already finished

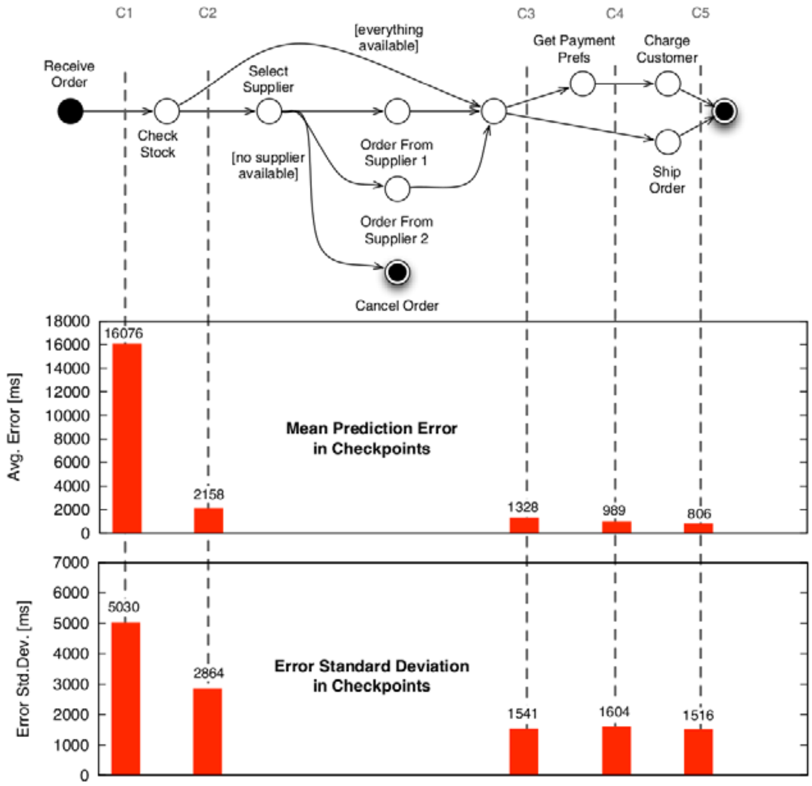


Figure 9: Average Error Rate

The results below the process show the output of the experiment. In C1, the prediction is mostly useless, since no real data except the user input is available. However, in C2 the prediction is already rather good. This is mostly due to the fact that in C2 the information whether the order can be delivered directly from stock is already available. In C3, C4 and C5 the prediction is continually improving, since more actual QoS facts are available, and fewer estimates are necessary. Speaking in absolute values, MPE in e.g., C3 is 1328 ms. Since the average SLO value in our illustrative example was about 16000 ms, the error represents only about 8% of the actual SLO value, which seems satisfactory. Similar to MPE, ESD is also decreasing; however, we can see that the variance is still rather high even in C3, C4 and C5. This is mostly due to our experimentation setup, which included the (realistic) simulation of occasional outliers, which are generally unpredictable.

Outcome	Positive
Experiences	See explanations in output
References	Philipp Leitner, Branimir Wetzstein, Florian Rosenberg, Anton Michlmayr, Schahram Dustdar and Frank Leymann. Runtime

	Prediction of Service Level Agreement Violations for Composite Services. In *3rd Workshop on Non-Functional Properties and SLA Management in Service-Oriented Computing, co- located with ICSOC 2009*, 2009.
Glossary	Key Performance Indicator, Business Process, Service Level Agreement
Keywords	Prediction, Monitoring, SLA

3.2.4 Validation of a compatible evolution framework

Software services are subject to change due to the introduction of new functionality, the modification of existing functionality (to improve performance for instance), or the inclusion of new policy constraints requiring an alteration of the services' behaviour. These changes can occur along different dimensions (e.g. structural changes or business protocol changes) and in order to control service development, an application developer needs to know about these changes – their motivation, implications, effects, and whether the change is complete. This history of service evolution can be expressed through the creation and decommissioning of different versions of the service during its lifetime. The authors investigate service versioning and propose an evolution framework for analysing, evaluating and constraining the evolution of services using their proposed formal definition of compatibility for services and a novel abstract service description model. This section documents the validation of the compatible evolution framework using an automotive purchase order process scenario to demonstrate its correctness and applicability. The application of the evolution framework to the scenario is found to be efficient, founded on solid theoretical foundations, and extensible to other technologies.

Table 7: On the Evolution of Services

Validation Set-up & Result	
Name	On the Evolution of Services
Synopsis	(see description below)
Authors	Vasilios Andrikopoulos, Salima Benbernou, Mike P. Papazoglou
Research questions	Control of the evolution of services
Scenario	Automotive Purchase Order Processing Scenario (Purchase_Order_Processing_BPM)
Research result	Compatible evolution framework
Method	Proof of concept
Description	For the purpose of validating our compatible evolution framework we used the Purchase_Order_Processing_BPM scenario that is being developed and used as one of the validation scenarios in S-Cube IA-3.2.1. The scenario is based on the Supply Chain Operations Reference model (SCOR) that provides abstract guidelines for building Supply Chains. This scenario is an example of how to realize SCOR level 3 activities using SOA-based processes for an enterprise in the automobile industry called Automobile Incorporation (aka AutoInc). AutoInc contains different business units, e.g. Sales, Logistics, Manufacturing, etc, and collaborates also with other partners like suppliers, banks, carriers, etc. In this scenario we assumed that the various process activities are implemented as services.
Goal	To demonstrate the correctness and applicability of the developed method
Set-up	Use of scenario as a demonstrator of the developed method
Inputs	Evolution framework and scenario description
Outputs	Confirmation of the theory through the use of the method
Outcome	Positive.
Experiences	We defined the concept of T-shaped changes, i.e. changes that respect the compatibility of a service, and we demonstrated how we can reason on

	them using the case study. This approach was shown to be more efficient than the existing approaches on compatible evolution, founded on a solid theoretical foundation and more importantly, extensible to other technologies. In future we may leverage the scenario for validating the evolution framework for service contracts between providers and consumers that allows for greater flexibility in evolving both parties in a compatible manner.
References	V. Andrikopoulos, S. Benbernou, and M. P. Papazoglou, "On The Evolution of Services", IEEE Transactions on Software Engineering, special issue on Service Engineering (submitted).
Glossary	Life cycle model, Service Based Application Construction, Service Composition, Evolution
Keywords	Service Evolution, Services Engineering, Versioning, Service Compatibility

4 Planned validation activities

Validation activities planned for execution within the 2 months subsequent to the time of writing were also reported by S-Cube partners; these are described below in this section.

4.1 Codified Knowledge about User Task Modelling Applied to Service Discovery

The application of user task models is proposed as part of a new requirements-based service discovery approach. Task models here describe structured sets of activities, performed in interaction with a system and influenced by its contextual environment, that a user has to perform to attain goals. It is suggested that user task models can contribute to the integration of human actors in the discovery and selection of SBAs by providing richer, more contextual descriptions and models than those currently available. Task models' specific, rich semantics may provide more context-specific information to inform SBAs' modeling, enabling for instance a mapping to service types and capabilities for finer-grain service discovery and selection, and more effective service composition. As part of the validation of the approach, an identification of task knowledge that is domain-specific is planned, followed by the extraction of domain-independent task knowledge that can be reused, this for the population of a prototype Task Knowledge Base being built. The first stage will elicit knowledge for known tasks in the navigation domain based on the S-Cube E-Government case study scenarios. Domain-independent task knowledge that can be reused will then be extracted and codified for matching to service requests using the SeCSE service discovery algorithm.

4.2 Measuring the Contributory Value of Service Science Networks

The growth in service science has underscored the need to investigate the contributory value of business processes and the influence on which both people and technology (service system) affects the delivery of a service and organisational performance. Service science explores the value co-creation of interactions between service systems. As service networks continue to grow, understanding the dynamic exchange of information and the value of the relationships between service systems is of critical importance. The contributory value is often referred to as the "application of competences (resources) to benefit another" [6]. Thus, this focuses on the information exchange (or value-exchange) of one resource to another which presents both parties with some benefit (tangible and intangible). A service is often referred to as "protocols plus behaviour" [6]. Failing to measure the value-exchange of service networks inhibits our capability to discover and monitor business process performance. We can model and validate the relational interaction of a service networks and measure its operational value (functional, behavioural, and structural) through the use of social network analysis (SNA). SNA is an approach and set of techniques which studies the exchange of resources among actors. It focuses on patterns of relations among nodes such as people, groups, organisations, or information systems and allows us to validate the value of ties and relationships between each node to provide a visual and mathematical representation of interaction and exchanges which influence behaviour. This will enable us to map service business process patterns and allows develop a framework to prescribe users how to efficiently redesign the network infrastructure through a multi-phase validation cycle. This validation technique will afford us the opportunity to apply the process pattern improvements to various other domains to determine whether we can match service science key performance indicators in other sectors. The first case study will be applied in a third level educational setting.

4.3 Software Process Model for the Adaptation of Service-Based Applications

It is generally accepted that an optimised software development process reduces development costs and ensures that a high quality software product is produced [7]. There are many mature software development life-cycles that contain development process details for all aspects of software development. They vary greatly in their level of detail, for example the popular waterfall model [8] provides only high level process descriptions while the ISO 15504 reference life-cycle [9] provides in dept details of each process in the development life-cycle. Many of these development process reference models were designed with development paradigms such as Object-Oriented Analysis and Design (OOAD) or Component Based Development (CDB) in mind. Unfortunately when service-based applications are considered many of these development process models are not directly applicable. This is because service-based applications are loosely coupled systems composed of component services that are accessed over intranets or the internet. The service-oriented paradigm promotes adaptation as a key feature which allows service-based applications to adapt to real-time conditions. Service-based applications adaptation is a novel process that does not exist in traditional process models. There are many reasons why service-based applications might want to adapt, for example to use more reliable component services, to meet business requirements or to facilitate interoperability between service providers and client applications. There are two ways that service-based applications can be adapted: static adaptation and dynamic adaptation. Static adaptation occurs when the application is adapted manually while dynamic adaptation occurs when the adaptation logic is built in to the application in advance and is executed automatically when the application runs. Adaptation is different to traditional software maintenance in that it is a re-configuration of component services rather than a costly modification to the core application.

In order to develop a suitable process model for the adaptation of service-based applications a four step program will be followed. The first step will be to illicit suitable practices for the adaptation process form existing process models. The second step will be to collect adaptation practices from industrial case studies. The third step will be to formulate an adaptation process model from the practices collected from the literature and industry. The final step will be to validate the process model using an industrial case study.

4.4 Formal Framework for concepts of business transaction

Activities are the smallest units in business process execution and consequently carry the notion of transaction more than any other element in an end-to-end process. An end-to-end long running transaction process embodies various activities that can be failed at any point during the transaction; however not all activities hold a same weight - any activity in the process can be more vital or less vital than another activity. A failure of a more vital activity in a long running business transaction may cause a failure of the entire transaction; in contrast the failure of a less vital activity may only degrade the overall performance of an end-to-end process and pragmatically, this type of failure may be tackled by either ignoring the failure or making a suggestive solution. Studies however show that currently, vital (or more vital) and non-vital (less-vital) activities are not well distinguished despite their high significance for long-running business transactions. This research intends to address this challenge and to provide a formal solution to understand vitality and non-conventional atomicity, which deals with vital activities that are performed as a single unit of work. Preliminary models of temporal logic specification of business transactions are proposed: a weak model corresponding to the flexible intuition of business transaction atomicity, and a strong model corresponding to a strict business transaction atomicity.

The validation approach planned for this research will entail the use of a Purchase_Order_Processing_BPM scenario based on the Supply Chain Operations Reference model (SCOR) that provides abstract guidelines for building Supply Chains. In this scenario it is assumed that the various process activities are implemented as services tied with promises to be satisfied. These promises are represented by agreement parameters composed of KPIs and policies that embody business rules, constraints such as temporal constraints, and QoS constraints that assist inferring the significance of an activity for a transaction. These will be regarded as the input for deciding the distinction between a vital and a non-vital activity; the correctness and applicability of the proposed method will be evaluated using the case study.

5 Conclusion

This deliverable reported on the empirical evaluations carried by S-Cube partners to validate S-Cube research results. The validation documentation template and guidelines were outlined, followed by documented validation results linked to S-Cube case studies and scenarios contributed by S-Cube partners. Planned validation activities for the upcoming months were then presented.

Considering the validation activities are ongoing and will be carried out throughout S-Cube’s lifespan, the initial set presented here will be extended as more results are collected and documented. PO-IA-3.2.7 (“Results of the Second Empirical Evaluation”) notably will present validation activities carried during the third year of the project. Finally, the evolving set of documented validation results will also be made available on the project portal for partners’ reference and to inform activities across work packages.

References

[1] Kazhamiakin, R. (Ed.): *Initial Definition of Validation Scenarios*. S-Cube Deliverable PO-IA-3.2.1, 2009
[2] Mazza, V.; Pernici, B.: Report on Common Pilot Cases. S-Cube Deliverable CD-IA-2.2.4, 2009
[3] Palvia, P.; Mao, E.; Salam, A.; Solman, K.: *Management Information Systems Research: What's There in a Methodology*? In: Communications of the Association for Information Systems, 11 (2003), p. 289–309
[4] Easterbrook, S.; Singer, J.; Storey, M.; Damian, D.: *Selecting Empirical Methods for Software Engineering Research*. In: Shull, F.; Singer, J.; Sjøberg, D. I. K. (Eds.): Guide to Advanced Empirical Software Engineering. Springer, London, 2007, p. 285–311
[5] Gehlert, A.; Pistore, M.; Plebani, P.; Versienti, L. (Eds.): First Version of Integration Framework. S-Cube Deliverable CD-IA-3.1.3, 2009
[6] Spohrer, J., Maglio, P. P., Bailey, J., & Gruhl, D.: *Steps toward a science of service systems*. In: IEEE Computer Society, 40, 71–77, 2007 (January).
[7] W. S. Humphrey. Managing the Software Process. Addison-Wesley Professional, 1989.
[8] W. Royce, “Managing the development of large software systems,” in Proceedings of IEEE Wescon, vol. 26, no. 1, 1970, p. 9.
[9] ISO/IEC/JTC 1/SC, ISO/IEC 15504-5:2006, Information technology - Process Assessment - Part 5: An exemplar Process Assessment Model, International Organization for Standardization, 2007.