COBAS-M: A Proposal Method in Analysis in Financial Corporative Business Architectures

Miguel A. González Serrano[1], Diana Pérez-Marín[2], Miren Idoia Alarcón[1]
[1]Universidad Autónoma de Madrid, Spain
[2]Universidad Rey Juan Carlos, Madrid, Spain
[2]Email: diana.perez {at} urjc.es

***Abstract*—The review of the state-of-the-art and the experience of working in many national and international IT projects reveal that corporative businesses lack of a methodology able to take into account business needs in an efficient process, without relying the success of the development on just technological aspects. In this paper, the focus is to present the analysis phase of a new methodology called COBAS-M (COrporative Business Architecture Software Method) to be used by financial entities. COBAS-M has been devised to allow stakeholders and technical experts to have a meaningful understanding of the organizations' businesses. COBAS-M relies on the definition of new types of business services, which are horizontal to the organization, allowing business users to find the services faster. The analysis phase of COBAS-M has been validated through a case study of real-life projects, in which an improvement of 41% in timing, a cost reduction of 42% and 8 satisfaction level have been registered. These results allow us to recommend appyling the life cycles proposed for the analysis phase to any development in financial corporative applications.**

Keywords-Service Oriented Architecture; Software Engineer; Architecture; Methodology; Analysis.

I. INTRODUCTION

Since Zachman [1] proposed in 1987 his model of business architectures, several authors have developed their own models and methodologies. In the last decade, with the development of web services [2], programming oriented to services has become standard [3]. A new paradigm has been proposed in the treatment of complex distributed applications known as Service Oriented Architecture (SOA), and this paradigm can be applied with a broad range of technologies (e.g. Java, .NET, JSON, HTTP) in addition to web services. Thus, SOA serves to capture principles, guidelines and techniques, this way providing a valid and generic architectural model for an organization in the development of applications.

According to Lago & Razavian [4] a service can be defined as a logical representation of a repeatable (business) activity that has a specific outcome. A service has to fulfill features such as being self-contained, state-less, adhering to a service contract; it may be composed of other services (service composition) and is a black-box to its consumers, between others SOA principles [5].

Business services can be defined as services created in the organization to accomplish a business goal (e.g. to check whether a Service Level Agreement of a client to automatically send a notification about the issue is violated, or to verify the user authentication in a website) independently of their technological implementation.

The review of the state-of-the-art and the experience of working in many national and international IT projects reveal that corporative businesses lack of a methodology able to take into account business needs in an efficient process, without relying the success of the project on just technological aspects.

Therefore, we have worked towards the definition and implementation of a new method for creating SOA for Corporative Business called COBAS-M (COrporative Business Architecture Software Method), which is proposed and validated in this paper for the first time. COBAS-M has been devised to allow stakeholders and technical experts to have a meaningful understanding of the organizations businesses. This is accomplished mainly through the definition of an analysis method, allowing business users to meet their expectations without losing any important detail within their needs. To do this, we propose lifecycles that focus on three important aspects of business analysis:

1. Analysis of business needs.
2. Analysis of existing systems in the organization, which promotes reutilization [5].
3. Analysis of new requirements not covered by legacy systems and considering the creation of new services.

COBAS-M analysis method has been validated using a case study of real-life SOA projects, in which an improvement of 41% in timing, and a cost reduction of 42% have been registered. Moreover, there has also been a significant improvement on the satisfaction levels of the users of COBAS-M analysis methodology. These results allow us to recommend appyling the life cycles proposed for the analysis phase to any development in financial corporative applications.

The paper is organized as follows: Section 2 reviews the state of the art; Section 3 introduces and describes the Analysis phase of COBAS-M; Section 4 presents the case study and the results gathered; and finally, Section 5 ends the paper with the discussion of the main conclusions and some lines of future work.

II. STATE-OF-THE-ART

Traditional Software Engineering methodologies differ one from another depending on the moment in history in which they have been raised, technology status and evolution in each moment of the history. In this paper, we have considered the most relevant approaches up to date. Table 1 gathers the main comparison criteria according to the literature review performed.

Table 2 compares the main methodologies found in the literature [13,14] according to their chronological publication using the criteria described in Table 1.

TABLE 1: COMPARISON CRITERIA FOR SOA METHODOLOGIES

Criterion	Description	Values
Lifecycle coverage	Specifies which of the classical lifecycle phases are covered by the method analyzed.	Analysis (A), Design (D), Implementation (I), Testing (T), Deployment (D), Governance (G)
Process Adaptability	Describes how well the process adapts the new and legacy systems to obtain a complete adaptation into SOA architecture.	Complete, Partial or Not Apply (NA)
Degree of Coverage	Describes if the proposed approach present a complete strategy for moving to SOA, or only a specific art of the modernization	All systems (ALL), New systems (NS) or Legacies (L)
Validation Maturity	Indicates if the proposed approach been applied and validated. We classify the proposed approach as an idea, a method demonstrated by a case study, or a commercially proven technique.	Not Provided (NP), Set of Guidelines (SoG), Case Study (CS) or Commercially Proven Technique (CPT)
Technology	Most of the approaches are just considering Web Services as unique technical solution of SOA services. We consider this as a limitation so this feature measures if the method is applicable to other technologies.	Agnostic (A), Web Services (WS) or Other Technologies (O)
Business Services	Use of the concept of business service as the fundamental unit of the methodology. It measures whether a method is agnostic or independent of the technology and the technical solution applied.	TS: Services Technical perspective BS: Services Business perspective ADS: Application Domain Services GBS: Global Business Services NA: Not Apply
Layers Architecture	Using layers in the architecture design as a grouping of business capabilities.	YES / NO
Services Business Taxonomy	Existence of a taxonomy of services adapted to the organization business where business users can find, use or compose services they need to present the business needs without the help of a technical user rather than support.	YES / NO
Service Provider	This criterion evaluates how deep the method analyzes, designs, and categorizes from the point of view of the service provider.	Single Service (SS) or Application (APP), Architecture(ARCH), Functional Domain (FD)
Service Consumer	This criterion evaluates how deep the method analyzes, designs, and categorizes from the point of view of the service consumer.	Single Service (SS) or Application (APP), Architecture(ARCH), Functional Domain (FD)
Service Functionality	This criterion evaluates how deep the method analyzes, designs, and categorizes from the point of view of the service functionality.	Service Interface Description (SID), Service Functional Description (SFD), Enterprise Business Areas (EBA), Not Apply (NA)
Temporal Efficiency	This criterion considers how the methods perform its phases from a timing point of view	Not Reported (NR) or results description data from study.
Indicator Economic	This criterion considers how the methods perform its phases from an economic (budget and revenue) point of view	Not Reported (NR) or results description data from study.
Product Quality	This criterion considers how the methods perform its phases from a process or final product point of view also taking into account the user opinion	Not Reported (NR) or results description data from study.
Ratio of Incidents	If this reported or not the result of a methodology in relation to the incident management and implementation issues	Not Reported (NR) or results description data from study.

As can be seen, all of the methodologies reviewed are focused on the stages of creating enterprise architectures, analyzing business and taking technology into account. On the other hand, they differ in their phases, some of them including processes based on the different views of the analysis & design of a service [5-7], others based on: traditional approaches to software engineering (top-down, bottom-up or agile) [5, 8, 9]; an enterprise approach [1, 10, 11]; or, layered architecture approaches from a technological point of view [12].

The classification begins by the comparison of the life cycle completeness, where we found cases in which there is a complete description of the life cycle [6, 12] and other cases that focus on the initial stages of the life cycle as [7] or in which we consider the stages most decisive in the creation of a SOA architecture analysis [4, 5].

Process adaptability measures how the methodological process is adapted to the systems that will be part of the new architecture. We consider that a methodology is complete if all the business needs are covered as SOA solution, a good example of a **complete adaptation** is the one presented by Arsanjani [10], which is one of the first authors that considered the need of a **complete and adapted lifecycle** to SOA with an architecture composed of different technical and functional layers. Arsanjani, through his SOMA methodology [10, 11], set out the need for identification of functional and non-functional requirements of the business, advanced concepts that were referenced and reformulated by authors from then on. However, his approach is limited to the evolution of applications made on the OOAD (Object-oriented analysis and design) paradigm with web services as a single technical solution, and the description of the architecture limited to services, business flows and components.

Using **business services** as the fundamental unit of the architecture and the orientation to one single **technology** measures whether a method is agnostic or independent of the technology and the technical solution applied. We have found that early methodologies are in most of the cases centered to Web Services as a unique technical solution with the existence of Technical Services (services without orientation to business) [6] and latest methodologies are going closer to the approach of being agnostic to technology and having an architecture with Business Services [4, 23] which represents an evolution towards to this proposal of having business closer to SOA.

Taxonomy of services and an **architecture** composed by layers are other critical factors to evaluate a methodology. A mature organization in terms of business orientation is the one where business users can find, use or compose services to present the business's needs without the help of a technical user rather than support. This is why we consider that methodologies that take into account the use of levels of abstraction [6], technical layers [7] or applications layers [15] in terms of business concepts [17] and also have a service taxonomy [23] for classifying services are closer to a good method.

Some of the methodologies describe the provider and consumer relationship from a traditional point of view, where a **single service** can be a consumer or a provider [10-12], but latest methodologies presents an evolution by considering that these perspectives can be seen as part of an architecture [4] or business domains [3, 19]. Also, **service functionality** can be described as a single service interface [15, 16] or in a modern point of view by using business terms [22].

Some methodologies of this study are focused on legacy systems [17, 18] or consider new functionality and legacy systems to be part of the project scope [5,7]. We consider the degree of coverage as an important criterion to take into account.

We also consider how these methodologies are being tested, not only as case studies but in real projects. This is why our classification includes, into our criteria, the validation maturity concept that classifies the proposed approach as an idea, a method demonstrated by a case study [15, 16], or a commercially proven technique [23].

Other features are to evaluate the improvement in terms of timing, product and process quality, revenue and budget indicators, and incidents post implantation phases. However, in many of the reviewed papers, there is no data published about those indicators.

III. ANALYSIS WITH COBAS-M

COBAS-M (COrporative Business Architecture Software Method) is a new method for creating SOA. It is oriented towards the horizontal coverage of the organization to take into account its business needs. In order to achieve those goals, COBAS-M proposes a modification of the traditional software analysis phase. The lifecycles proposed for the analysis phase in COBAS-M are shown in Figure 1.

In the following sections, each phase will be described in detail. In particular, Section 3.1 focuses on the COBAS-M analysis phase and its three sub-lifecycles: Business Requirements Analysis, Legacy Systems Analysis, and New Systems Analysis.

It is important to notice that iterations between lifecycles are allowed taking into account that each lifecycle could be iterated until there is enough detail for the identified services, and then it can be closed. This condition allows modifications in the main documents and products of each lifecycle.

3.1 COBAS-M: Analysis phase

COBAS-M tries to solve the limitations of technical points of view, implementing approaches such as those proposed in the literature [3, 12, 19] that are focused on technology, by taking into account business's needs. This is the main reason why this phase starts with the Business Requirements Analysis described in Section 3.1.1. Moreover, given that one of the key factors of COBAS-M is the importance of reutilization during the analysis, it is necessary to perform a Legacy Systems Analysis as described in Section 3.1.2. Finally, it is necessary to think of the New Systems Analysis as described in Section 3.1.3.

Figure 2 shows an overview of the COBAS-M analysis phase, and Table 3 provides a list of the documents created in this phase to serve as a guide to improve the reading of the following subsections.

TABLE 2: SUMMARY OF THE PRINCIPAL SOA METHODOLOGIES FOUND IN THE STATE-OF-THE-ART IN TERMS OF METHOD DESCRIPTION.

	Lifecycle coverage	Process Adaptability	Degree of Coverage	Validation Maturity	Technology	Business Services	Layers Architecture	Services Business Taxonomy	Service Provider	Service Consumer	Service Functionality	Temporal Efficiency	Indicator Economic	Product Quality	Ratio of Incidents
Zachman (1987) [1]	Complete	NA	ALL	NP	A	NA	NA	NA	NA	NA	NA	NR	NR	NR	NR
Arsanjani (2003, 2004) [10, 11]	Complete	Complete	ALL	NP	WS	TS	YES	YES	SS	SS	SID	NR	NR	Defined but NR	NR
Zimmermann et al. (2004) [15, 16]	A&D	Complete	ALL	CS	A	BS	YES	NA	ARC H	SS	SID	NR	NR	NR	NR
Huhns et al. (2005) [6]	Complete	Complete	NA	CS	WS	TS	YES	NA	SS	SS	SID	NR	NR	NR	NR
Erl (2005, 2007) [5, 8, 9]	A&D	Complete	ALL	CS & CPT	WS	BS	YES	YES	SS	ARC H	SID	NR	NR	NR	NR
Doddavula (2005) [12]	Complete	Partial	New systems	CPT	WS	NA	YES	NA	SS	SS	SID	NR	NR	NR	NR
Jones (2005) [7]	Initial planning	Complete	ALL	CS	A	BS	YES	NA	NA	SS	SID	NR	NR	NR	NR
Lewis et al. (2005) [17]	Complete	Partial	Legacy	SoG	O	ADS	YES	NA	SS	ARC H	SFD	NR	NR	NR	NR
Sneed (2006) [18]	Complete	Partial	Legacy	CS	O	BS	YES	NA	NA	NA	SFD	NR	NR	NR	NR
Papazoglou (2006, 2007) [3, 19]	Complete	Complete	ALL	NP	WS	ADS	YES	NA	SS	ARC H	SID	NR	NR	NR	NR
López et al. (2007) [20]	Complete	Complete	ALL	NP	WS	NA	NA	NA	NA	NA	NA	Uses ROI and KPI's but there is no data provided	NR	NR	NR
Delgado et al. (2010) [21]	NR	Complete	ALL	NP	WS	BS	NA	NA	NA	NA	NA	NR	NR	NR	NR
Fuhr et al. (2011) [22]	NR	Partial	Legacy	NP	O	BS	NA	NA	SS	SS	SID & SFD	NR	NR	NR	NR
Lago and Razavian (2012) [4]	A&D	Complete	ALL	CS & CPT	A	BS	YES	YES	ARC H	ARC H	SID & SFD	NR	NR	NR	NR
Pohl (2012) [23]	Complete	Complete	ALL	CS	A	BS	YES	YES	SS	ARC H	SID	NR	Defined but NR	NR	Defined but NR

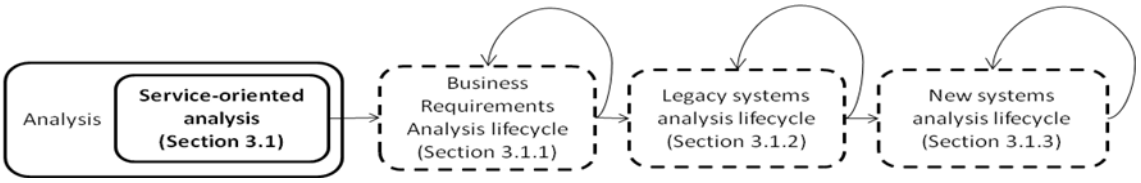


Figure 1: COBAS-M Analysis lifecycles

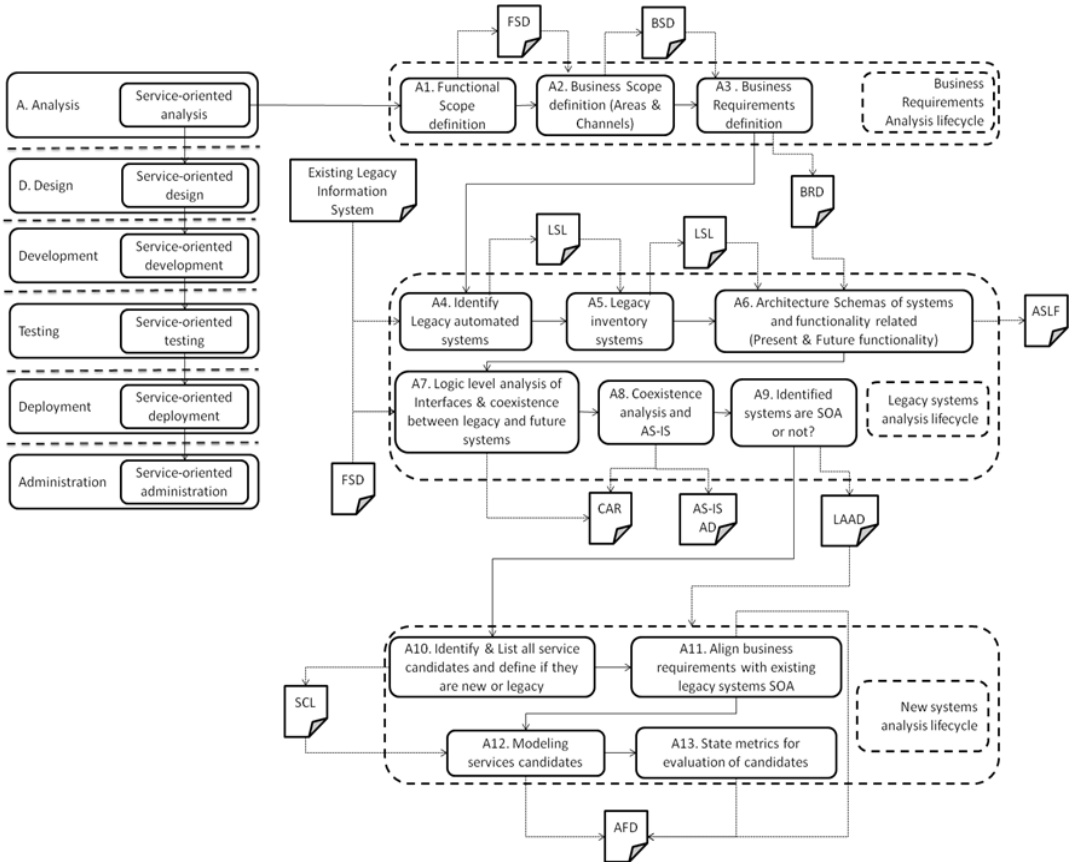


Figure 2: COBAS-M Analysis phase (activities are represented in rectangle shapes, and deliverables, which are all documents, are represented in document shapes)

3.1.1 Business Requirements Analysis

The inputs for this phase are the business needs, the information of the stakeholders and the existing knowledge in the company. The output consists of three documents:

1. **Functional Scope Document (FSD, created in activity A1),** which is created by the business users and is used by the technical Department to start the analysis. This document describes the desired functional behavior from a very high detailed point of view. The business users will describe the business cases they want to implement. The main point is to provide a brief description of the functional problem or opportunity the project is trying to address. Some examples could address issues like changes in legislation, not meeting service level agreements (SLA) or changes in global business requirements.
2. **Business Scope Document (BSD, created in activity A2),** which is created by the stakeholders taking into account all the Business Areas & Channels implicated. This document details all the objectives of each Business Area or Channels in the organization. It should also include the expectations, limits and restrictions of the functional scope (FSD) for each area.

TABLE 3: LIST OF DOCUMENTS CREATED IN THE ANALYSIS PHASE

Document	LifeCycle	Roles
Functional Scope Document (FSD)	Business Requirements Analysis	Created by business users.
Business Scope Document (BSD)	Business Requirements Analysis	Created by business users.
Business Requirements Document (BRD)	Business Requirements Analysis	Created by Technical Architects and approved by stakeholders
Legacy automated System List (LSL)	Legacy Systems Analysis	Created by Technical Architects
Architecture Schemas of Legacy systems and Functionality Document (ASLF)	Legacy Systems Analysis	Created by Technical Architects
Coexistence Analysis Report (CAR)	Legacy Systems Analysis	Created by Technical Architects
AS-IS Architecture Document (AS-IS AD)	Legacy Systems Analysis	Created by Technical Architects
SOA Legacy Availability Analysis Document (LAAD)	Legacy Systems Analysis	Created by Technical Architects
Services Candidates List (SCL)	New Systems Analysis	Created by Architects and approved by business stakeholders
Analysis Final Document (AFD)	New Systems Analysis	Created by Architects and approved by business stakeholders

3. **Business Requirements Document (BRD, created in activity A3),** which is created by the Technical Architects and approved by stakeholders. It describes the essential business requirements for services. BRD includes the first definition of business requirements that will include the service concept as main unit of business representation, but these units will not be considered as service or service candidates, they will be just business requirements that could be services in the Design phase.

Every document of this lifecycle could be modified or completed in each iteration of the lifecycle as shown in Figure 1.

3.1.2 Legacy Systems Analysis

According to [24]: *"a legacy information system is any information system that significantly resists modification and evolution to meet new and constantly changing business requirements"*. This is the main reason why there is a need to improve classical lifecycles to analyze business requirements using legacy systems to reuse the business logic created in the past to gain time and resources and reduce efforts. This lifecycle considers migration strategies [14], replacement, reengineering and wrapping. This decision is included in the analysis task in this lifecycle in activities A7 and A8.

The input to this phase is the output of 3.1.1 and the information of the legacy systems existing in the company. The first activity is the **identification of legacy automated systems (A4).** The **legacy inventory system (A5)** is created by technical SOA Architects in order to relate the requirements with the existing systems. The list is written in the document called **LSL**. LSL contains information about the legacies systems that is used in D1 when the strategy is decided. This legacies information is based on SMART [17], and it main artifact SMIG considering information such as age of the software, complexity or cost and migration effort.

In this phase, it is also necessary to identify the existing **architecture schemas of legacy systems and their related functionality (A6) in the Business Requirements Document (BRD).** The results of this process are written in the document called **ASLF**.

The next activity would be **the logic level analysis of interfaces & coexistence between legacy and future systems (A7)** [25, 26]. This activity has as input the "Existing Legacy Information System" in the organization and the FSD from the previous lifecycle in order to complete all the requirements that need to be kept from existing system and promote the reutilization. It will be mandatory to create the Architecture schemas that show both sides of this functionality. This is an analysis that merges the past activities into a **single schema of information creating the coexistence analysis report (CAR) and the AS-IS Architecture Document (AS-IS AD created in activity A8).** The results are written in the AS-IS Architecture Document created by SOA Architects. This phase will be completed with the production of a Legacy Availability Analysis Document (**LAAD**), a document that will explain how legacies will be impacted by the new requirements. It is important to notice that this document has as input all the documents generated in previous activities. Finally, it will be mandatory to **identify whether the systems analyzed are SOA or not (A9**).

3.1.3 New Systems Analysis

This phase combines the two previous lifecycles *("Business Requirements Analysis Lifecycle" and "Legacy Systems*

Analysis Lifecycle") to get a single picture that completes the analysis phase. The final activities of this analysis, before the design starts, to **identify & list all service candidates and define if they are new or legacy (A10), aligned business requirements with existing SOA legacy systems (A11), service candidates and modeling services candidates (A12) to finally, incorporate business metrics for evaluation of candidates (A13).**

This final lifecycle of the Analysis phase in COBAS-M has as input all the outputs from past lifecycles. This phase is completed with the creation of **the Services Candidates List (SCL),** created by Architects and approved by business stakeholders. Metrics for evaluation of candidates will be an important point to be included into the **Analysis Final Document (AFD).** The metrics will be used to define some aspects of services into the Design Lifecycle in COBAS-M and they have to be specified by business users.

IV. Case Study

This section presents a case study of real-life solutions proposed to be implemented in an existing architecture in a financial enterprise. The goal is to validate the analysis phase of COBAS-M. The design of the case study was done before the projects started. The Section is organized into three subsections: Section 4.1 presents the design details; Section 4.2 the development of the case study; and, Section 4.3 presents the results achieved in terms of time, cost and satistaction of the development team.

4.1 Design of the case study

The case study is based on the evaluation of 2 projects. The projects that have been selected were developed for a major international bank with more than 10,000 employees, a presence in over 20 countries and aimed at retail banking business and private banking. The projects are focused on the investment area, in particular, on the recruitment of financial products (Bonds, Repos, Simultaneous, etc).

All the projects have been chosen because the first author of the paper had the opportunity of participating in them, to test the analysis phase of COBAS-M as Project Manager. Table 4 shows a summary of all the projects in chronological order based on start date.

As can be seen in Table 4, both projects have been developed in an international environment with teams located in different countries. The budget of the analysis phase is 71.520 €. The number of services is between 30 and 35, and they are quite recent, the oldest project started in April 2012.

Table 5 provides a list of selection criteria identified for the projects of Table 4 so that these projects are similar and thus, both projects can be compared.

TABLE 4: PROJECTS SUMMARY

A	**Project Budget**	446.520 €	**International environment**	YES
	Analysis Phase Budget	71.520 €	**Start Date**	June 2012
	Global Business Scope			
	Product Analysis Department of Back Office should be able to check that all operations are input according to Trading are correct in systems and the following steps have the correct control in terms of events, terminations & fixings.			
B	**Project Budget**	446.520 €	**International environment**	YES
	Analysis Phase Budget	71.520 €	**Start Date**	April 2012
	Global Business Scope			
	The Department of "Internal Reconciliation" must establish alarms in relation to operating and market risk by finding discrepancies between the FO and BO systems and making internal reconciliation of operations.			

TABLE 5: PROJECT SELECTION CRITERIA LIST

Item	Criteria
1	Timing estimated by the team in the analysis phase.
2	Same analysis phase budget.
3	Same global project budget (analysis, design, development, test and delivery phases).
4	Number services candidates (variability +/- 5 services from one project to another).
5	Global Business Scope: same functional scope and similar business requirements.
6	International environment: several countries involved with project team members located in different places.
7	Project Kick Off. Each of the projects should start within a 6 month interval from one another.
8	Team members in each project should be the same or have at least similar profiles

TABLE 6: COMPARISON OF PROJECTS BY PAIRS ACCORDING TO TABLE 5

	Analysis Estimation (days)	Analysis Phase Budget	Global Budget	Services Candidates	Global Business Scope	International environment	Project Kick Off	Team Members
A	120	71.520 €	446.520 €	30	Back Office Area in Investment Banking	YES	June 2012	Same as B
	Services should be created from an existing J2EE application layer. Web Services should be the interface between .NET existing applications and J2EE developments.							
B	120	71.520 €	446.520 €	35	Back Office Area in Investment Banking	YES	April 2012	Same as A
	Services should be created from an existing J2EE application layer (15 migrated and 20 new functionalities). Web Services should be the interface between .NET existing applications and J2EE developments.							

Project A was developed according to COBAS-M, and project B was developed according to another methodolody (no COBAS-M). Both projects were developed in the same conditions, and their results were compared to evaluate the possible benefits of applying COBAS-M analysis phase.

4.2 Development of the case study

Following the indicators gathered in Table 5, projects A & B will be compared as shown in Table 6.

Projects A & B had to work with a bigger set of services in this case study, from 30 to 35 new and migrated services candidates in each project. As can be seen, in Table 6, both projects were carried out in an international environment, the team, and the functional and technical requirements where quite close. The project team was in both projects composed by a Project Manager, an Architect and a Functional Analyst.

The following paragraphs describe step by step the products created in project B, which was the one developed according to COBAS-M. Note that some of the data shown have been completed detailing high functional level for not incurring a breach of the privacy policy details of the entity where the project was performed.

The main input of the Business Requirements Analysis lifecycle that starts COBAS-M (just in project B) is the Functional Scope Document (FSD, created in activity A1, see Figure 2) which is a document in a way of Project Charter. This document establishes the needs of the main business that the project sponsor has received to justify the project. These reasons are specified so project B has as a summary of the document to the following situation:

"Due to the current economic situation which is characterized by great market volatility, the strategic guideline of the Bank as regards the recruitment of investment products has changed. Hiring of this type of products strategy now has the aim of procurement operations with fewer amounts, less risk and the requirement to obtain the same benefit but distributed in many engagements. To this end, the Bank technology strategy aims to modify current Midldle, Front and Back Office systems. This initiative is composed of more than 20 projects developed in parallel by several groups of business and architecture. Some of them use COBAS-M as the implementation method and others not. In our case we have grouped the two following projects since they are very similar according to the criteria in Table 5."

Table 7 below shows a summary of the Business Scope Document (BSD, created in activity A2, see Figure 2) detailing the outcome of the meetings of analysis of business requirements which examines the metrics and KPI's (Key Performance Indicator) necessary for business as well as major business requirements. The techniques used in this phase of the project are BrainStorming, Sensitive Analysis, Decision Tree Analysis; this is why at this point the business requirements are still in a high level point of view.

Once global business targets have been described, it is important to obtain the concrete business requirements. Each business requirement should be categorized with a status that describes its stage at this COBAS-M phase. Table 8 shows parts of one of the BRD documents that have been generated in this phase. It is important to note that the State field of the requirement passes between the following States:

1. Original: Identified requirement.
2. Accepted: Requirement that will be included in a version of a product.
3. Cancelled: Requirement that is not going to be included in a version of a product.
4. Deferred: Requirement whose inclusion is postponed to a later version of the product.
5. Detailed: Requirement specified at the level of detail sufficient for the customer.
6. Ongoing: requirement where he is working (design, development, testing).
7. Completed: Requirement whose development has finished.
8. Implemented: Now available for customer requirement.

There should a unique way of identifying each document, the following pattern of unique requirement identification is used: *"PROJECT ID – FUNCTIONAL MOD – TEC/FUN - <Unique ID>"*

TABLE 7: BUSINESS SCOPE DOCUMENT (BSD, CREATED IN ACTIVITY A2) CONTENT SUMMARY OF PROJECT B

Functional Module	Business Targets	KPI's & Business Metrics
Product Management	Dealings are double checked according to Trading specifications, deal's life cycle in terms of fixings, events and settlements are correctly made. Compare Trading specifications in deals.	Follow up operations in progress. Number of issues with a particular product. Total of operations with a particular product.
Internal Reconciliation	Check and report breaks and issues within the internal interfaces intervening in the deal's life cycle. Manage and fix operation with issues. Report issues. Rollback with Issues or problems detected by the systems. Notify issues in deals, once detected they must be reported and logged.	Number of control issues between FO & BO. Number of issues in BO Operations Number of incidents in FO operations.
Risk Management	Calculate Operative risk index, Accounting risk index, and Collateral risk index. Notify business users about the indexes calculated. In relation to some intervals of each index set up some alarms.	Operative risk index. Accounting risk index. Collateral risk index. Number of alarms of each index. Percentage of operative risk exceeded according to the range of market confidence.
Account Reconciliation	Match incoming and outgoing currency flows with the expected settlement information. Obtain the number and rate of control settlement issues. Detect currency flow problems in operations. Calculate the result of incoming / outgoing flows.	Number and rate of control incidents in liquidations. General view of critical view of the whole set of incoming/outgoing operations. Rate of currency flow problems in operations. Final result of incoming / outgoing flows.
Account & Management Control	Control and Report account status. Prevent, Detect & Control accounting issues. Notify results to business users by mail once a control issue has detected. Notify Spain Central Bank about some issue detected.	Rates in accounting issue. Number of issues reported to Spain Central Bank in a period of time. Number of Control accounting issues reported, detected and fixed. Fix accountable issues. Edit accountable flows.

TABLE 8: BUSINESS REQUIREMENT DOCUMENT (BRD, CREATED IN ACTIVITY A3, SEE FIGURE 2) CONTENT SUMMARY OF PROJECT B

Id. Req	Requirement Status	Functional Area / Channel	Functional Module or System	Functional Description	Risks
N/A	N/A	Investment Banking (Back Office)	Product Management	The system will have to be able to double check dealings according to Trading specifications.	N/A
N/A	N/A	Investment Banking (Back Office)	Product Management	The system needs to deal's life cycle in terms of fixings, events and settlements are correctly made. Compare Trading specifications in deals.	N/A
N/A	N/A	Investment Banking (Back Office)	Internal Reconciliation	It is important for business users that set of systems checks and reports breaks and issues within the internal interfaces intervening in the deal's life cycle.	N/A
N/A	N/A	Investment Banking (Back Office)	Internal Reconciliation	Manage and fix operations with issues.	N/A
N/A	N/A	Investment Banking (Back Office)	Internal Reconciliation	Report issues to business users by mail or other method of notification.	N/A
N/A	N/A	Investment Banking (Back Office)	Internal Reconciliation	Rollback and 2phase commit with problems detected by the systems in operations.	N/A
N/A	N/A	Investment Banking (Back Office)	Internal Reconciliation	Detect and notify issues located in in deals. Once detected, issues must be reported and logged into the audit global system through Web Service call.	N/A

N/A	N/A	Investment Banking (Back Office)	Risk Management	The system must be able to calculate the operative risk index, Accounting risk index, and Collateral risk index.	N/A
N/A	N/A	Investment Banking (Back Office)	Risk Management	Once the system has calculated the below indexes the system should notify automatically to business users.	N/A
N/A	N/A	Investment Banking (Back Office)	Risk Management	The system should have the way of configuring and establish alarms to some intervals of each index set up some alarms.	N/A
N/A	N/A	Investment Banking (Back Office)	Account Reconciliation	Match incoming and outgoing currency flows with the expected settlement information.	N/A
N/A	N/A	Investment Banking (Back Office)	Account Reconciliation	Calculate the number and rate of control settlement issues.	N/A
N/A	N/A	Investment Banking (Back Office)	Account Reconciliation	Automatic detection of currency flow problems in operations online.	N/A
N/A	N/A	Investment Banking (Back Office)	Account Reconciliation	The new functionality should be able to calculate the result of incoming and outgoing flows according to the product types considered in this project (REPOS, Bonds and Simultaneous)	N/A
N/A	N/A	Investment Banking (Back Office)	Account & Management Control	Create reports of control and Report account status.	N/A
N/A	N/A	Investment Banking (Back Office)	Account & Management Control	Prevent, Detect & Control accounting issues.	N/A
N/A	N/A	Investment Banking (Back Office)	Account & Management Control	Notification by mail of results to business users by mail once a control issue has detected.	N/A
N/A	N/A	Investment Banking (Back Office)	Account & Management Control	Notify Spain Central Bank about some issue detected by mail.	N/A

It should be noted that there is more specific information (risk, etc) not listed in the above table since it is sensitive to that it can be published, such as head of the business requirement, non-functional requirements of each requirement such as response times and information operations of the Bank in relation to internal business rules.

Once the business requirements have been listed and described, technical architects starts the Legacy Systems Analysis lifecycle, that has as principal input the FSD (created in activity A1, see Figure 2) and the BRD (created in activity A3, see Figure 2) and whose main is create the LAAD (created in activity A9, see Figure 2) which contains the CAR (created in activity A7, see Figure 2). This lifecycle starts with LSL (created in activity A4 and completed in A5, see Figure 2) and ASLF (created in A6, see Figure 2) creation, the following image shows all the functional modules and systems that will be affected with new and modifications in their functionalities due to the projects of this case study (included into the ASLF, created in A6, see Figure 2).

To finally complete the LAAD (created in activity A9, see Figure 2) is necessary detect connections between different functional modules affected legacies. So, it is necessary to include in the final documentation a diagram like that we present below (see Figure 6), showing calls between modules and their relationship. Many of these calls are made to this day and many others are functionality that was included in the architecture, noted that some of them have needs to be connections, synchronous and others simply asynchronous.

The final document LAAD (created in activity A9, see Figure 2) includes a list of functionalities that are supposed to be modified and some others functionalities that are supposed to be new. This document is the main entry point of the following lifecycle, New System Analysis, where Technical Architects will create the AFD (created in activity A11 and completed in A12 & A13, see Figure 2) with all the information of previous lifecycles and with the SCL, created in activity A9 (see Figure 6).

To complete the LSL (created in activity A4 and completed in A5, see Figure 2) it was needed to know which systems where affected by each functional module, this is why the following image shows all the real systems involved in the functional architecture. In the image below, there are systems such as Murex©, RiskArt©, Calypso© or Alfresco©, commercial IT systems that are part of the functional operative in the bank and that will be affected in the new business requirements.

Once the real IT systems are detected, technical architects should determinate which of these systems functionality will be modified or considered to create a new module inside that covers this functionality (see Figure 5).

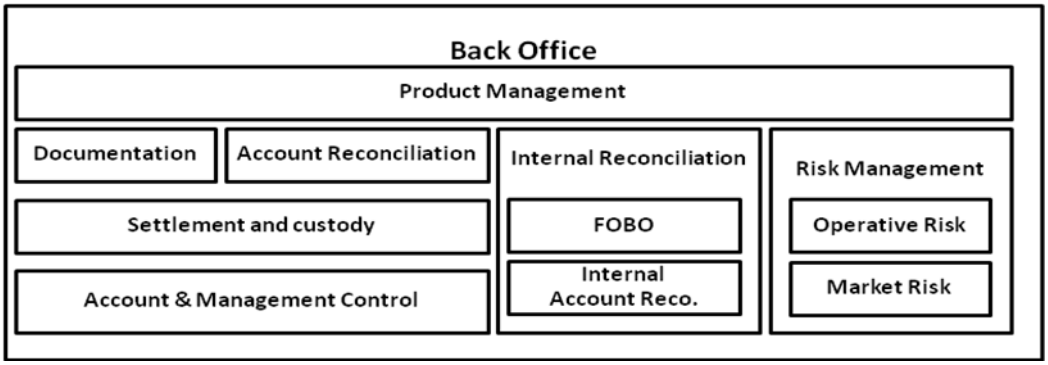


Figure 3: Functional Modules affected in Projects A & B (included in ASLF, created in A6)

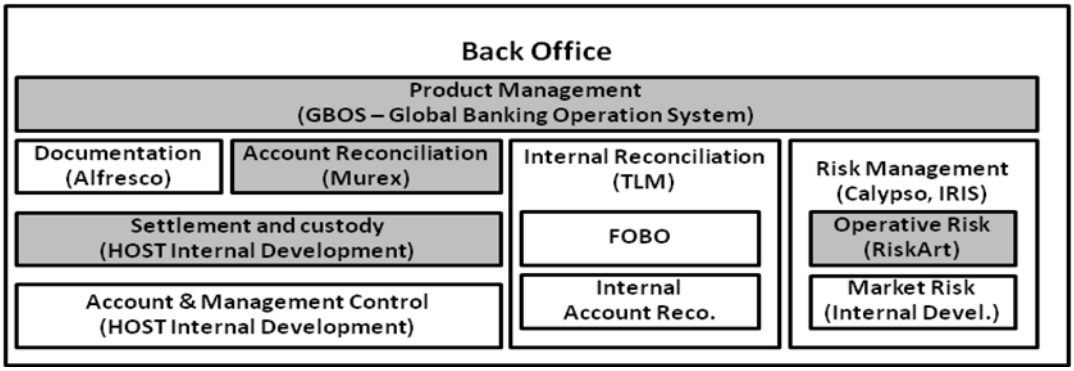


Figure 4: Functional Modules affected in Projects A & B described with their IT real systems

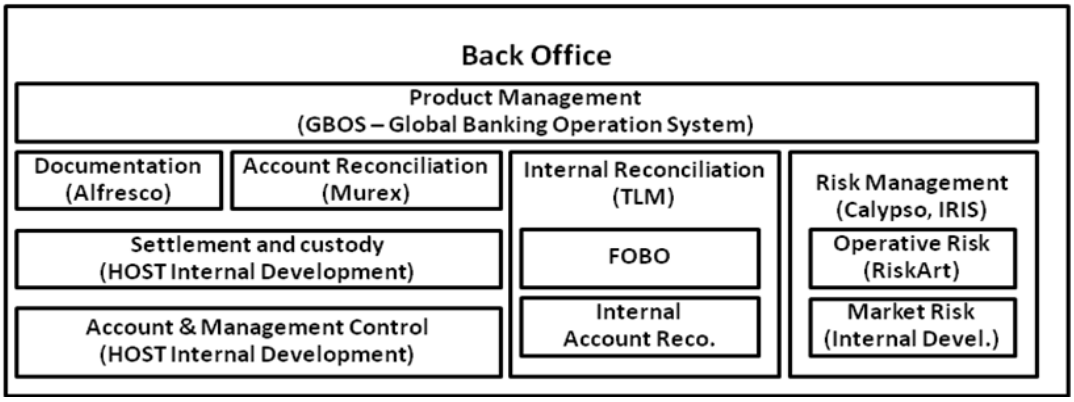


Figure 5: Functional Modules affected in Projects A & B, grey modules will be modified

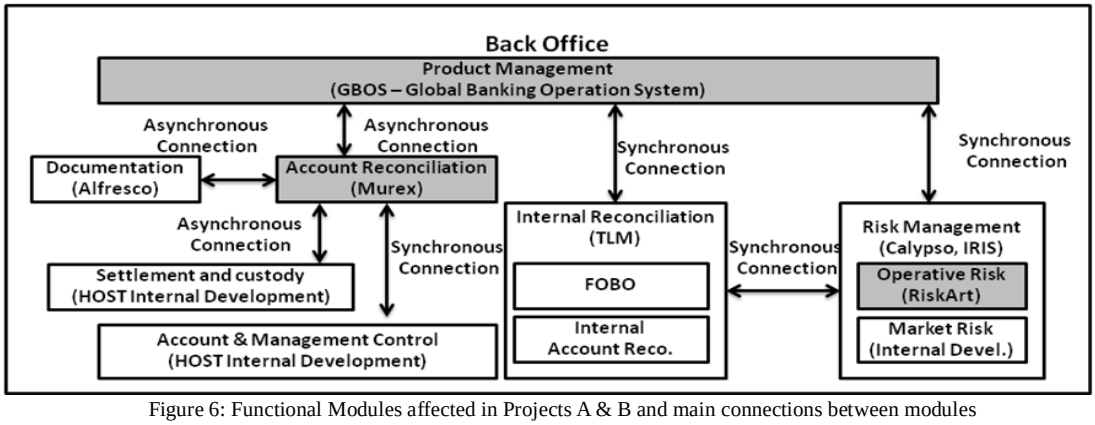


Figure 6: Functional Modules affected in Projects A & B and main connections between modules

TABLE 9: SERVICES CANDIDATE LIST (SCL, CREATED IN ACTIVITY A9) CONTENT SUMMARY OF PROJECT B

Id. Req	Functional Area / Channel	Functional Module or System	Service Candidate
N/A	Investment Banking (Back Office)	Product Management	Set/Get Operation Status
N/A	Investment Banking (Back Office)	Product Management	Set/Get Alarm In Operation
N/A	Investment Banking (Back Office)	Product Management	Change Operation Path
N/A	Investment Banking (Back Office)	Product Management	Compare Set of Operations
N/A	Investment Banking (Back Office)	Product Management	Get Breaks of a product
N/A	Investment Banking (Back Office)	Product Management	Show Operation Path
N/A	Investment Banking (Back Office)	Product Management	Calculate Operation Result
N/A	Investment Banking (Back Office)	Product Management	Get Operation Status at a concrete Time
N/A	Investment Banking (Back Office)	Internal Reconciliation	Manage Alarms In Operation
N/A	Investment Banking (Back Office)	Internal Reconciliation	Obtain number of issues in BO/FO.
N/A	Investment Banking (Back Office)	Internal Reconciliation	Manage and fix operation with issues.
N/A	Investment Banking (Back Office)	Internal Reconciliation	Report issues.
N/A	Investment Banking (Back Office)	Internal Reconciliation	Rollback operation with Issues.
N/A	Investment Banking (Back Office)	Internal Reconciliation	Notify issues in an operation
N/A	Investment Banking (Back Office)	Risk Management	Manage Operative Risk.
N/A	Investment Banking (Back Office)	Risk Management	Manage Account Risk.
N/A	Investment Banking (Back Office)	Risk Management	Manage Collateral.
N/A	Investment Banking (Back Office)	Risk Management	Calculate Index of a risk in a future/past date.
N/A	Investment Banking (Back Office)	Risk Management	Manage alarms and intervals of confidence
N/A	Investment Banking (Back Office)	Account Reconciliation	Obtain the Number of control issues in liquidations
N/A	Investment Banking (Back Office)	Account Reconciliation	Get the rate of control incidents in liquidations (% index in a period of time)
N/A	Investment Banking (Back Office)	Account Reconciliation	Calculate the result of a set of incoming/outgoing operations.
N/A	Investment Banking (Back Office)	Account Reconciliation	Obtain the Rate of currency flow problems in operations.
N/A	Investment Banking (Back Office)	Account Reconciliation	Get Final result of incoming / outgoing flows.
N/A	Investment Banking (Back Office)	Account Reconciliation	Notify results to business users by mail.
N/A	Investment Banking (Back Office)	Account & Management Control	Obtain rate in accounting issue.
N/A	Investment Banking (Back Office)	Account & Management Control	Notify Spain Central Bank about some issue detected.
N/A	Investment Banking (Back Office)	Account & Management Control	Notify about some issue detected to other financial Entities.
N/A	Investment Banking (Back Office)	Account & Management Control	Notify results to business users by mail.

4.3 Results

This section provides the final results of the case study. The case study has been evaluated in terms of:

1. **Time** in the development of analysis phase on each project. The goal is to reduce the timing. It is measured in number of days. Equation 1 is the one used to calculate the results:

$$TimeImpr = 100 - \frac{Analysis\ Real\ Timming\ (COBAS - M\ Project) * 100}{Analysis\ Real\ Timming\ (NON\ COBAS - M\ Project)} \quad (1)$$

2. **Cost analysis** related to factors such as budget planned, revenue expected and real cost and real revenue has been included. The goal is to reduce the cost. It is measured in euros. Equation 2 is the one used to calculate the results:

$$CostImpr = 1 - \frac{REAL\ Cost\ of\ the\ Analysis\ Phase\ (€)}{ESTIMATION\ Cost\ of\ the\ Analysis\ Phase(€)} \quad (2)$$

3. **User's satisfaction**, all activities of COBAS-M were evaluated in a questionnaire in order to get feedback from all the team members taking into account factors such as usability, easiness to use, easiness to learn, and general satisfaction. The goal is to increase the easiness of use and learning of COBAS-M, and the general satisfaction of the team using COBAS-M. It is measured in numerical scales from 0 (minimum level of the value) to 10 (maximum level of the value). Equation 3 is the one to used to calculate the results (N is the number of users of the methodology surveyed):

$$SatisfImpr = \frac{\sum satisfValue}{N} \quad (3)$$

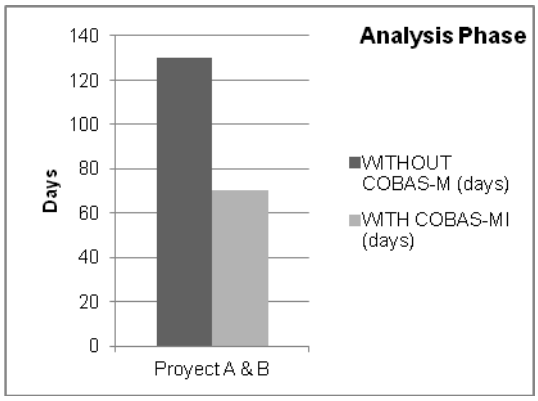
4.3.1 Timing Results

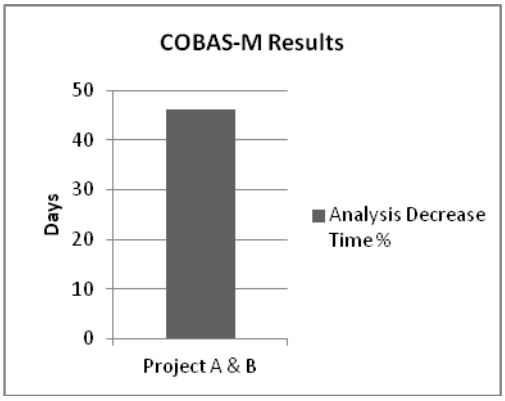
The timing results taken from each of the projects analyzed are shown. It can be seen how COBAS-M as an Analysis method can generate an improvement of 46% comparing projects that use COBAS-M (project B) with the other that does not used COBAS-M (project A).

Table 10 describes the values in days registered in the Analysis phase. Project A and B were estimated to be accomplished in 120 days, but eventually project B was finished in 130 days and B, in which COBAS-M was used, was finished in 70 days. This means an improvement of 41% (i.e. 50 days less than the estimation).

TABLE 10: TABLE DATA FROM ANALYSIS PHASE TIMING CONSUMING

Proyect ID	Estimation	Real Timming	% of improvement (Estimation vs Real)
A (WITHOUT COBAS-M)	120	130	-8,33%
B (WITH COBAS-M)	120	70	41%





Figures 7 & 8: Timming results for COBAS-M on project B

TABLE 11: COST ANALYSIS TABLE

Project	Budget Planned	Revenue Expected (%)	Revenue Expected (€)	Real Revenue (%)	Real Cost of Analysis Phase
A	71.520 €	25%	17.880 €	-6,66%	76.280,00 €
B	71.520 €	25%	17.880 €	41,67%	41.720,00€

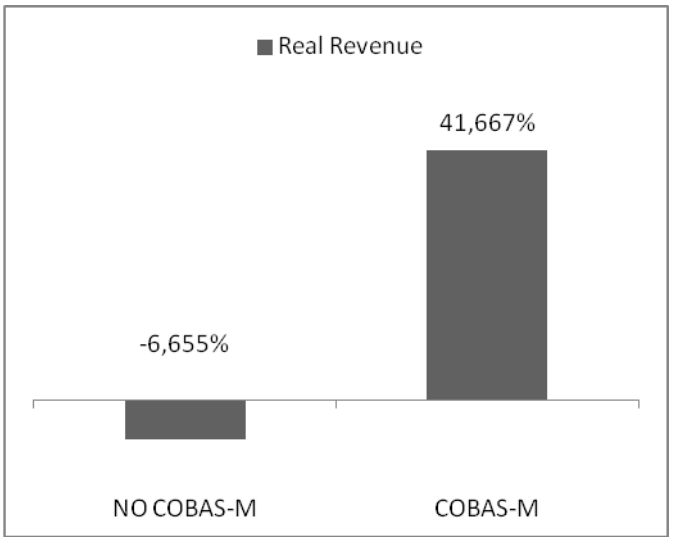


Figure 9: Total results in cost analysis

As can be seen, there is not only possible to successfully apply COBAS-M, but there is an improvement of 41% in the timing of the analysis phase. Moreover, it is important to note how it has been possible to improve the initial estimation made at the beginning of the project given that whenever COBAS-M is used, the estimated times are improved compared to the real timing.

4.3.2 Cost Analysis

The cost results taken from each of the projects analyzed are shown in this section. It can be seen how the analysis phase of COBAS-M can generate an improvement of 10% to 27,04% in each of this lifecycle phases. Table 11 describes the values registered by the pair of projects under study in terms of budget planned, revenue expected and real cost.

As can be seen, using COBAS-M increased the revenue expected due to the number of days estimated to complete each phase was decreased. Projects A and B were estimated to gain 25% each. Project A did not use COBAS-M and its cost was 4.760,00 euros more that the expected cost, but on the other hand, project B that used COBAS-M registered a gain revenue

of 29.800,00 euros, which is an improvement of 41,67%.

Figure 9 shows a graphical comparison of the total average revenue expected for analysis phase for the pair of projects of the study.

As can be seen, not only it is possible to successfully apply COBAS-M, but there is an average improvement in cost reduction of 41,67%. Moreover, these results are highly related to revenue expected, it could be hypothesized that COBAS-M shows a better revenue result when applied to high estimation projects. However, more research should be carried out to validate that hypothesis.

4.3.3 User's satisfaction

User's satisfaction has been evaluated in the whole COBAS-M taking into account all the activities of the method by giving a questionnaire to all team members of each project in which COBAS-M was used. Any interested reader can contact us for the whole questionnaire and results gathered. They are not published here due to their length. It is important to note that the results of the questionnaire are the promedio of all the responses of stakeholders and team members of each project.

This analysis was evaluated by each team member by rating each activity from 0 to 10 individually. Finally, all the team members talked about COBAS-M giving us interesting feedback about the methodology with comments such as the following:

1. *"COBAS-M is a very organized method that guides development teams using SOA in an intelligent way."* (SOA architect in project B).
2. *"I like the way COBAS-M reuses legacy systems and combine them with new functionalities."* (Senior Functional Analyst in project B).
3. *"The use of separate life cycles to distinguish between legacies, new systems and business requirements greatly helps business users to find the services they need."* (Business User in project B).

Figures 10 and 11 show the point of view of 9 COBAS-M business users and 8 COBAS-M technology users.

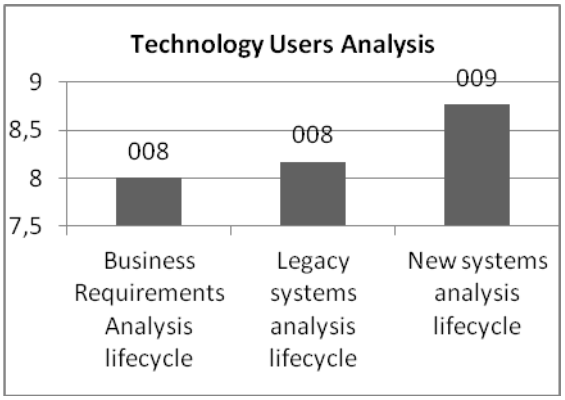


Figure 10: Technology users analysis

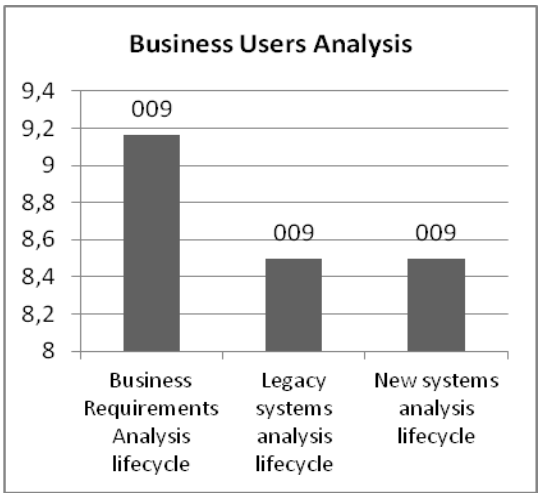


Figure 11: Business users analysis

The following topics were considered given their relevancy as main conclusions of users satisfaction with COBAS-M:

1. Business users consider business requirements analysis lifecycle the most important phase (value=9,17).
2. New systems analysis is the most important lifecycle for technology users (value=8,77) being the business requiremente analysis lifecycle the less important for them. This is probably because technology users does not play the main role at this stage of the methodology.
3. From both points of view legacy systems analysis lifecycle can be erased if there is not a migration of services or functionality inherit.
4. Even new systems and business requirement analysis lifecycles are the most important parts of COBAS-M for business users and technology users, there is no significant difference in terms of rating with legacy system analysis lifecycle, the rate difference is quite irrelevant (see Figure 12).

Other important target of this analysis was to get feedback from all the team members of each project in topics such as usability, easiness of use, easiness of learning, and general satisfaction. Figure 13 shows the results gathered.

As can be seen in Figure 13, regarding user satisfaction with COBAS-M: for all team members, the general satisfaction is near to rate of 9 in the scale 0 (minimum value) to 10 (maximum value). Usability is the second important feature for every user, which tells us those users, finds COBAS-M as useful in their corporations.

The easiness of use of COBAS-M is maintained independently of the type of users; thanks to this we can say that COBAS-M is adapted to technical and non-technical users. Easiness of learning is rated with a 7 I the scale 0 (minimum value) to 10 (maximum value).

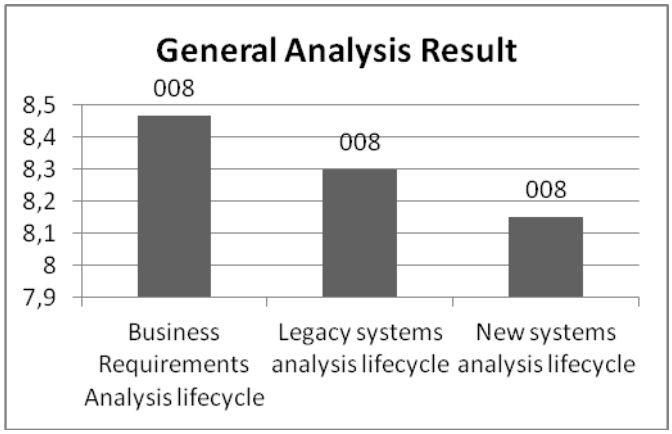


Figure 12: Business users analysis

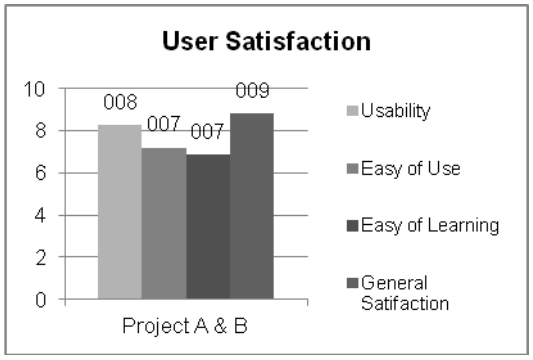


Figure 13: User satisfacion

V. CONCLUSIONS AND FUTURE WORK

The review of the state-of-the-art and the experience of working in several national and international projects reveal **that corporative businesses lack of a methodology able to take into account business needs** in an efficient process, without relying the success of the project on just technological aspects.

Therefore, in this paper, we have proposed and validated the analysis phase of **a new method for creating software for Corporative Business called COBAS-M** (COrporative Business Architecture Software Method), which takes into account business needs.

COBAS-M has been devised to allow stakeholders and technical experts to have a meaningful understanding of the organizations businesses. This is mainly achieved by defining new types of business services horizontal to the organization, allowing users to find the services faster.

COBAS-M has been validated with a case study of real-life projects. The results gathered show evidence **of improvement in terms of:**

1. **Time** required in the development of the analysis phase. The goal is to reduce this factor. It is measured in number of days. A reduction of the time needed for the analysis phase when COBAS-M is applied has been found of 41%, i.e. companies users of COBAS-M are able to find the services faster and complete the analysis and design faster.
2. **Cost analysis**, which is related to factors such as budget planned, revenue expected and real cost and real revenue have been included. The goal is to reduce the cost. It is measured in euros. There is a reduction of cost of 41,67% in the analysis phase when COBAS-M is applied, i.e. companies users of COBAS-M are able to reuse better their resources.
3. **User's satisfaction**, all activities of COBAS-M have been evaluated in a questionnaire in order to get feedback from all the team members of each projects, taking into account usability, easiness of use, easiness of learning, and general satisfaction. The goal is to increase the easiness of use and learning of COBAS-M, and the general satisfaction of the team using COBAS-M. It is measured in numerical scales from 0 (minimum level of the value) to 10 (maximum level of the value). Architects rated COBAS-M as easy to learn, although learning COBAS-M in bigger projects is harder. In general, using COBAS-M is easy independently of the size of the project. For all team members in all the projects using COBAS-M the general satisfaction is near to 8 (see Figure 13).

As future work, we would like to keep validating the COBAS-M methodology, with the design and testing, once the analysis phase has shown promising results, which allows us to recommend its use to any other team working on the development of software projects for financial corporations.

REFERENCES

[1] J.A. Zachman, "A framework for information systems architecture", IBM SYSTEMS JOURNAL, VOL 26. NO 3, 1987

[2] E. Newcomer, G. Lomow, "Understanding SOA with Web Services (Independent Technology Guides)". Addison-Wesley Professional. ISBN 0321180860. December 2004.

[3] M. P. Papazoglou, M. P.; W. J. van den Heuvel, "Service-oriented design and development methodology", International Journal of Web Engineering and Technology (IJWET), 2006.

[4] P. Lago and M. Razavian, "A Pragmatic Approach for Analysis and Design of Service Inventories", Department of Computer Science, VU University Amsterdam, The Netherlands, 2012

[5] T. Erl, "Service-Oriented Architecture (SOA): Concepts, Technology, and Design", Prentice Hall, August de 2005, ISBN: 978-0131858589

[6] M. Huhns, M. Singh, "Service-Oriented Computing: Key Concepts and Principles", IEEE Computer Society, Febrero 2005

[7] S. Jones, "A Methodology for Service Architectures", Capgemini UK plc, August 2005.

[8] T. Erl, "Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services", Prentice Hall, 26 de April de 2004, ISBN: 978-0131428980

[9] T. Erl, "SOA: Principles of Service Design", Prentice Hall, 28 de Julio de 2007, ISBN: 978-0132344821

[10] A. Arsanjani, "Service-Oriented Modelling and Architecture (SOMA)", IBM developer-Works 2004

[11] A. Arsanjani, "Explicit Representation of Service Semantics: Towards Automated Composition Through a Dynamically Reconfigurable Architectural Style for On Demand Computing", International Conference on Web Services; pp 34 - 37; 2003

[12] S. K. Doddavula, "Designing an Enterprise Application Framework for Service-Oriented Architecture", Java.Net, April 2005.

[13] E. Ramollary, D. Dranidis, A. J. H. Simons, "A Survey of Service Oriented Development Methodologies", South East European Research Centre, 2007

[14] A. A. Almoadies, J. R. Cordy, T. R. Dean, "Legacy System Evolution towards Service-Oriented Architecture, School of Computing", Queens University Kingston, Ontario, Canada, 2010

[15] O. Zimmermann, Krogdahl & Gee, "Elements of Service-Oriented Analysis and Design", 2004, IBM Zurich Research Group

[16] O. Zimmermann, N. Schlimm, G. Waller & M. Pestel, "Analysis and Design Techniques for Service-Oriented Development and Integration", IBM Deutschland, 2004

[17] G. Lewis, E. Morris, L. O'Brian, D. Smith and L. Wrage, "SMART: The Service-Oriented Migration and Reuse Technique", Integration of Software-Intensive Systems Initiative, September 2005

[18] H. M. Sneed, "Integrating legacy Software into a Service oriented Architecture", AneCon GmbH, Vienna, Austria, 2006

[19] M. P. Papazoglou, P. Traverso, S. Dustdar, F. Leymann, "Service-Oriented Computing: State of the Art and Research Challenges", IEEE Computer Society, 2007

[20] G. López, A. Echeverría, P. Fierro (PhD.); I. Jeder. "Una propuesta de modelos de ciclo de vida (mcvs) para la integración de los procesos de negocio utilizando service oriented architecture (SOA)", Laboratorio de Informática de Gestión Facultad Ingeniería - Universidad de Buenos Aires, 2007

[21] A. Delgado, F. Rui, I. García-Rodríguez de Guzmán, M. Piattini, "A Model-driven and Service-oriented framework for the business

process improvement", Computer Science Institute, Faculty of Engineering, University of the Republica, Technologies and IS Depto.,Faculty of Computer Science,University of Castilla-La Mancha, Journal of Systems Integration, 2010

[22] A. Fuhr, T. Horn, V. Riediger, A. Winter, 2013. "Model-driven software migration into service-oriented architectures", Comput Sci Res Dev 28:65–84.

[23] K. Pohl, "S-Cube - The European Network of Excellence in Software Services and Systems", 2012

[24] R. Pérez-Castillo, I. García-Rodríguez de Guzmán, M. Piattini Alarcos "Business process archeology using MARBLE", Research Group, University of Castilla-La Mancha, Ciudad Real, Spain, Information and Software Technology

[25] ITIL V3 Glossary, 30 May 2007

[26] N. Joachim, D. Beimborn, T. Weitzel, "Investigating Adoption Determinants of Service-Oriented Architectures (SOA)", Proceedings > Proceedings of SIGSVC Workshop. Sprouts: Working Papers on Information Systems, 10(126), 2010.

[27] P. Allen, "The service oriented process", in CBDi Journal, February 2007

[28] L. Bass, P. Clements, R. Kazman, "Software Architecture in Practice", Addison-Wesley Professional; 2sd edition, 19th April 2003, ISBN: 978-0321154958

[29] G. Booch, InfoWorld interview 2004

[30] M. A. González Serrano, D. Pérez-Marín, M. I. Alarcón. "Clasificación de los Servicios Web de Negocio Corporativos basada en la Funcionalidad, Horizontal de las Organizaciones". M.V. de Castro, J.M. Gómez, I. Iribarne (Eds). JCIS'2012, pp. 81-87, ISBN 978-84-15487-26-5. Jornadas Sistedes'2012, 17-19 septiembre, Universidad de Almería.

[31] Jacobson, G. Booch; J. Rumbaugh, "The Unified Software Development Process", Addison Wesley Longman, 1998.