

Tape Mbo'e: A First Experimental Assessment

Ilse Grau

University of Trento,
Trento, Italy,
ilse.grau@disi.unitn.it

and

Guilherme H. Travassos

Universidade Federal do Rio de Janeiro,
Rio de Janeiro, Brasil, 21941-972,
ght@cos.ufrj.br

and

Luca Cernuzzi

Universidad Católica "Nuestra Señora de la Asunción",
Asunción, Paraguay,
lcernuzz@uca.edu.py

and

Adolfo Villafiorita

FBK Center for Information Technology IRST,
Trento, Italy 38123,
adolfo.villafiorita@fbk.eu

Abstract

The development of software not only needs to consider the construction process, but also other aspects such as cost, human resources and communication among stakeholders. The lack of simplicity into this context becomes explicit when some restrictions, such as service oriented architecture, must be considered as the basic style to build sustainable applications into environments where practitioners are not aware of this software technology. In addition to this, most of the available software processes are not directly applicable nor are they reusable, so learning times becomes risk for the development of the project. Therefore, Tape Mbo'e (TME) has been proposed to support the building of such applications, into development environments like developing countries where we can have economic constraints and scarcity of proficient practitioners. The first application of TME has been to develop a service-based application whose goal is to provide the interoperability among legacy systems of different public agencies in Paraguay. Initial results of this experience indicated the feasibility and simplicity of TME when applied in this field. The evaluation process, its results and conclusions are described in this paper.

Keywords: Sustainability, Interoperability, Evaluation, Service-Based Application

1 Introduction

The interest in Service Oriented Computing has increased due to its observed benefits: support the interoperability among heterogeneous legacy systems, re-usability of functionalities and construction of adaptable and loosely coupled applications.

Building service-based applications requires repeatable software development processes to produce high quality software. On one hand, the existing development processes, like those proposed for object-oriented or component-based paradigms, do not completely satisfy the needs of service oriented application development [1]. On the other hand, the service oriented processes focus on developing services, but not on project management. Project management is critical in the construction of complex systems (e.g. e-government applications) to properly address different aspects such as human resources management, quality assurance management, and outsourcing, among others [2], [3].

For these reasons, Grau et al. have proposed Tape Mbo'e (TME)¹, a service oriented process [4], [5]. It encompasses the whole life cycle of service-based applications (SBAs) that are built considering the sustainability, which means the capability of affording the operation of software in the long term [6]. Besides, an extended version of TME with life cycle, roles, disciplines, and work products is presented in this paper.

TME was not built from scratch; it is based on Open Unified Process (OpenUP)², that has been chosen for being a light version of Rational Unified Process (RUP). RUP is a well-organized methodology and it has been widely used in the industry. However, it includes many artefacts and documents, making its use hard in this context because its adaptation would be time consuming and expensive [7].

TME has adapted OpenUP to support the development of SBAs. TME extends the life cycle and disciplines of OpenUP to cover the maintenance and retirement of applications and to assure the quality of SBAs. Besides, TME improves the project management discipline of OpenUP to consider the needs of development projects of public agencies of the Paraguayan government. The changes in this discipline have been inspired on the standard of project management [8].

We are aware that the changes proposed by TME reduce its agility. However, they support the sustainability, which is more valuable for the type of projects that we are interested in, such as the government projects.

TME is being applied for the first time in the Information Exchange System (IES) that is in charge of interchanging citizens' data among the legacy systems of the whole public sector of Paraguayan government. The validation of TME represents a long-term study. However, the first observations indicated the feasibility and applicability of TME when used by practitioners. It is worth noting, that TME's evaluation has been the first application of rigorous experimental evaluation in the public sector in Paraguay. Therefore, the first results of this experience are going to be described in this paper.

The paper is organized as follows. Section 2 depicts the TME including life cycle, roles, how the disciplines are performed during the life cycle, and differences between TME and OpenUP. Section 3 outlines the followed process to evaluate the application of TME in case studies. Moreover, the process is illustrated with the experience of applying TME in a public agency in Paraguay. Finally, conclusion indicates some future directions.

2 Tape Mbo'e: a Service Oriented Process

TME is a process for developing and maintaining service-based applications (SBAs), and it is based on Open Unified Process (OpenUP) (version: June 1, 2012). OpenUP has been chosen after examining various Agile Methods (AMs) such as Scrum [9], [10], Extreme Programming (XP) [11], [12], Feature Driven Development (FDD) [13], Dynamic Systems Development Method (DSDM) [13]. We have chosen OpenUP because it is the only one among those AMs that considers the project management and the deployment as disciplines. Its work products cover more aspects of the life cycle than the other AMs. Additionally, it is the light version of the Rational Unified Process (RUP). There are differences between TME an OpenUP, however; we will describe them in the next sections.

¹Tape Mbo'e means: "method" in Guaraní Language

²<http://epf.eclipse.org/wikis/openup/>

2.1 Life cycle

TME adopts the four phases of OpenUP and extends the life cycle of software development by adding the new “Operation Phase”.

The life cycle is iterative, incremental, and the phases are executed sequentially. Each iteration produces an incremental value to stakeholders, results of iterations should be defined in the iteration plan. The duration of the iteration might vary depending on project characteristics, but it is suggested to last for one-month.

The increment is a unit of work that can be used to measure project progress, and to make decisions in each iteration.

TME suggests that independent parts of the development can be carried out in parallel iterations.

The next sub-sections describe the phases whose sequences of execution are illustrated in Figure 1.

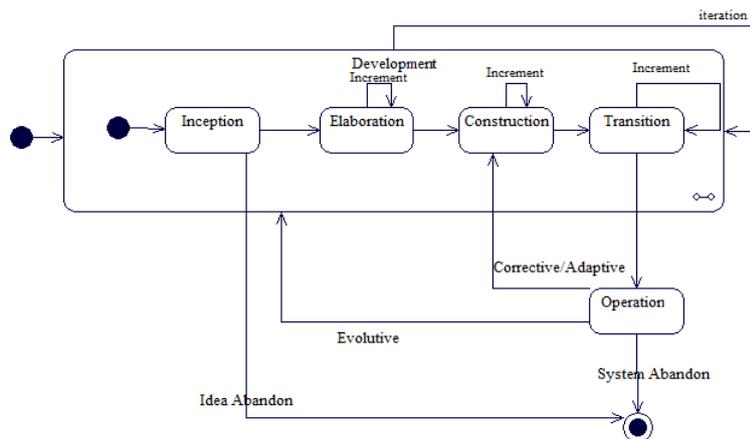


Figure 1: Life Cycle

- **Inception Phase.** The starting point is the need to develop a software that has to meet a set of requirements. The requirements can be refined through the iterations. Thus, the general requirements are collected to estimate cost, human resources, schedule, and to identify stakeholders.

Having determined the requirements, major decisions such as recruiting human resources, acquiring infrastructure, outsource the whole development or a part of it are taken. Also, the quality is planned.

- **Elaboration Phase.** This phase aims at outlining the architecture of the software, described by work products of the disciplines executed during it.

At the beginning, the iteration is organized; this includes the selection of the requirement to be developed. Secondly, the selected requirement is described considering the remark of users, and it is refined incrementally.

Moreover, staff, schedule, and risks are all managed, and the quality of the work products are monitored and controlled. The output of this phase is a set of work products that describe the requirements (see Table 1).

- **Construction Phase.** The effort is focused on building and testing the solution that has to meet the pre-established quality standards. Thus, this phase is composed of increments of building and testing. Once the solution fits the requirements, it is ready to be tested by users in the next phase.

- **Transition Phase.** This phase involves the software release for users testing before its deployment. It can comprehend various increments of testing in which minor adjustments can result from users' feedback.

Moreover, users can request to change some functionality. In that case, a decision on whether accepting the request or not, and on when it will be implemented, has to be taken.

At the end, the software is deployed and the users are trained on how to operate it.

- **Operation Phase.** It tackles software productive life that encompasses corrective, adaptive, perfective, and evolution maintenance as well as the software retirement.

The corrective maintenance focuses on repairing faults and errors detected during the software operation while the perfective aims at improving code. The adaptive maintenance adjusts the application to support changes such as in the infrastructure layer; in the application context and location; of the user types, preferences, and constraints; in the functionalities provided by the composed services; in the way the service is being used and managed by its consumers. Furthermore, in this phase it is possible to adapt or substitute the component services to improve the performance of the software.

The evolution implies starting a new development cycle.

Finally, software retirement entails uninstalling the whole application or services from production environment. Due to its critic importance for service consumers, we have to assure the remaining services continue to work correctly.

2.2 Roles

Tape Mbo'e (TME) defines roles, each in charge of tasks needed to meet objectives. Its roles are like the three types proposed by OpenUP³: Basic, Deployment, and Environment.

2.2.1 Basic Roles

This type of roles encompasses those who participate in the development and the stakeholders.

TME introduces two changes in the basic roles of OpenUP, it removes the Any Role and add one of Quality Analyst. The former has been exempted because it does not participate in a specific discipline, while the latter is included to be in charge of the Quality Assurance discipline.

The roles are the following:

- **Stakeholder.** The role represents all the people interested on the execution of the project or affected by it. In this sense, all other roles can be considered as a stakeholder, and besides a stakeholder is called user when it is the person who asks for the development and/or will use the application.
- **Project Manager.** The role is responsible for the Project Management discipline. Hence, the project manager is responsible for coordinating and controlling the team and the project execution.
- **Analyst.** The role is responsible for the Requirements discipline. The analyst meets with stakeholders to gather the requirements. Having collected these, the analyst describes them with use cases and mock-ups.
- **Architect.** The role is responsible for the Architecture discipline. The architect examines the requirements to outline architecture of the application through models and documents. Moreover, is in charge of the technical decision-making that can affect the whole development.
- **Developer.** The role is responsible for the Development Discipline. Developers constructs the software according to the architecture and perform verification tasks once the development is done.
- **Tester.** The role is responsible for the Testing Sub-discipline. The tester verifies and validates the software. The tester supports the revision of the software when it is reviewed by an user.
- **Quality Analyst.** The role is responsible for the Quality Management sub-discipline. The role is in charge of ensuring the quality of the process and deliverables.

At the end of a phase or iteration, the Quality Analyst controls and evaluates the quality of the work products of the disciplines to suggest adjustments or improvements if necessary.

³<http://epf.eclipse.org/wikis/openup/>

2.2.2 Deployment Roles

These roles are in charge of the tasks that are carried out after submitting the application. TME does not include the roles of “Course Developer”, “Product Owner”, “Technical Writer” of OpenUP.

TME assigns the responsibilities of the “Course Developer” to the “Trainer”, and it hands the responsibilities of the “Technical Writer” to the “Quality Analyst”. Also, it calls “User” to the “Product Owner”. The roles are the following:

- **Deployment Engineer.** The role is in charge of deploying software into production environment as well as publishing services into service repository. After deployment it verifies the success of the procedure, as well as the implementation of the roll-back plan if needed.
- **Deployment Manager.** The role is responsible for coordinating the deployment into production environment and service repository as well as monitoring and controlling the deployment engineers.
- **Trainer.** The role is responsible for training the team members and users. Team members are trained on skills needed to accomplish their jobs, while users on software usage. Additionally, the trainer prepares the material for the training course.

2.2.3 Environment Roles

These roles include people in charge of tools, configuration, and assets. TME considers the role of “Configuration Analyst” to be in charge of the Configuration and Change Management. Moreover, it excludes the role of “Process Engineer” proposed by OpenUP because its responsibilities are carried out by the Quality Analyst.

The roles are described as follows:

- **Tool Specialist.** The role is responsible for the Configuration and Change Management sub-discipline. It takes care of providing technical assistance about the tools used in the project including purchasing, installation, configuring, and maintenance.
- **Configuration Analyst.** This role is in charge of tracking and controlling the changes in the assets of the project, as well as keeping the assets.

2.3 Disciplines execution during the life cycle

Tape Mbo'e (TME) follows an iterative, incremental, architecture-centric, and risk driven process. It is represented by analysing two dimensions: temporal evolution (life cycle) and disciplines as it is illustrated in Figure 2. The former shows the software life cycle, while the latter mentions the disciplines involved to produce software [4].

A discipline is a set of tasks that have a common goal, and produce work products. The work products can be documents (manual, planes, schedule, contract, etc.), models, code, templates, mock-ups, service level agreement (SLA). The disciplines are executed during the life cycle, and each of them is under the responsibility of one of the aforementioned roles.

TME proposes six disciplines whose relationships are shown in Figure 3. The vertical disciplines (i.e., Project Management and Quality Assurance) are in charge of organizing, controlling, and suggesting tasks to the horizontal disciplines. The disciplines, their sub-disciplines, work products, and the phases in which they are carried out are presented in Table 1 and described bellow.

2.3.1 Inception

The major goals are the definition and delimitation of the development project. These are accomplished through the following disciplines:

Table 1: Disciplines, Sub-disciplines, Work Products, Phases, and Roles

	Sub-Disciplines	Work Products	Phases	Roles
Project Management	Communication Management	<ul style="list-style-type: none"> • Meeting Minute • User Training 		
	Development Management	<ul style="list-style-type: none"> • Project Plan • Iteration Plan • Interchange Agreements 	<ul style="list-style-type: none"> • Inception 	
	Human Resource Management	<ul style="list-style-type: none"> • Task Assignment • Progress Report 	<ul style="list-style-type: none"> • Elaboration • Construction 	Project Manager
	Outsourcing Management	<ul style="list-style-type: none"> • Technical Specification • Suppliers Evaluation • Contract • Monitoring Suppliers • Guarantee 	<ul style="list-style-type: none"> • Transition • Operation 	
Quality Assurance	Quality Management	<ul style="list-style-type: none"> • Service Level Agreement (SLA) • Standards • User Documentation • Support Documentation • Training Materials 	<ul style="list-style-type: none"> • Inception • Elaboration • Construction 	<ul style="list-style-type: none"> • Quality Analyst • Configuration Analyst • Tool Specialist • Tester • Trainer
	Configuration and Change Management	<ul style="list-style-type: none"> • Change Request • Version Control 	<ul style="list-style-type: none"> • Transition • Operation 	
	Testing	<ul style="list-style-type: none"> • Test Cases • User Acceptance 		
Requirements	—	<ul style="list-style-type: none"> • Requirements List • Use Case Diagram • Mock-ups • Glossary 	<ul style="list-style-type: none"> • Inception • Elaboration • Construction • Transition • Operation 	Analyst
Architecture	Business Modelling	<ul style="list-style-type: none"> • Business Model 	<ul style="list-style-type: none"> • Elaboration • Construction • Operation 	
	Integration	<ul style="list-style-type: none"> • Domain Model • Interchange Standard • Structure Model • Deployment Model 		Architect
	Components Specification	<ul style="list-style-type: none"> • Data Model • Service Model 		
Development	—	<ul style="list-style-type: none"> • Source Code • Service Composition 	<ul style="list-style-type: none"> • Construction • Transition • Operation 	<ul style="list-style-type: none"> • Developer • Deployment Engineer • Deployment Manager
Deployment	—	<ul style="list-style-type: none"> • Service Repository • Deployment Plan • Back up Plan 	<ul style="list-style-type: none"> • Construction • Transition • Operation 	<ul style="list-style-type: none"> • Deployment Engineer • Deployment Manager

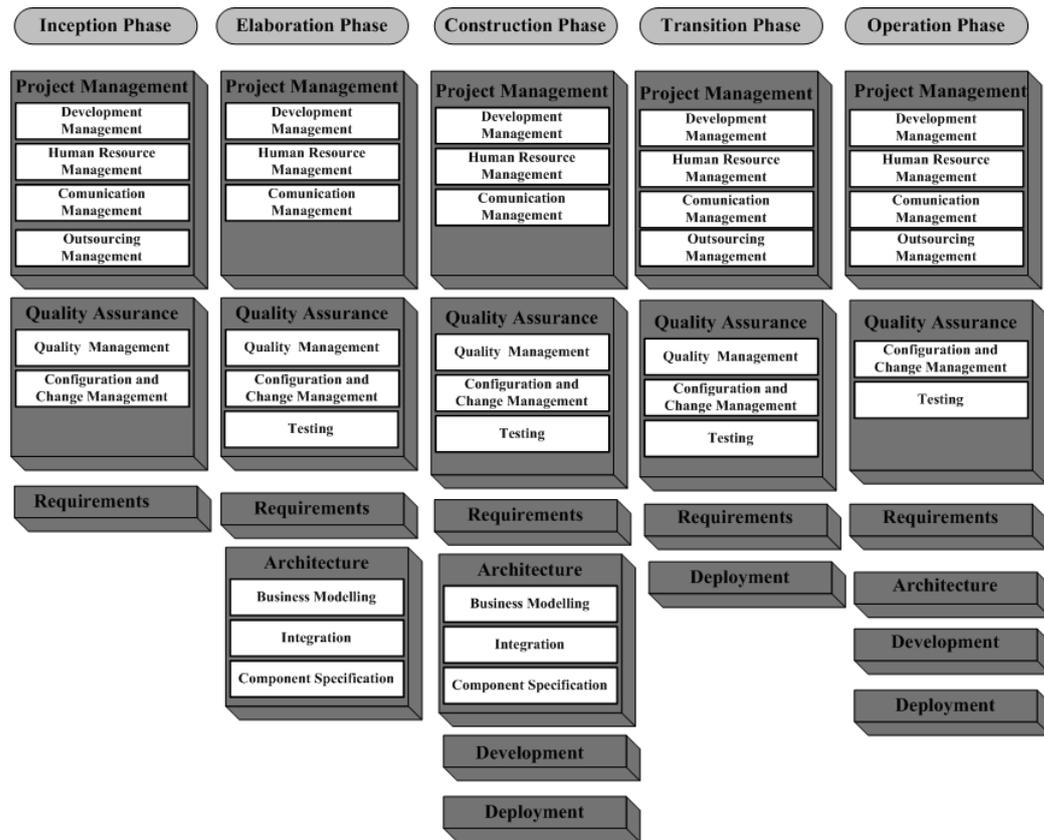


Figure 2: Life Cycle and Disciplines

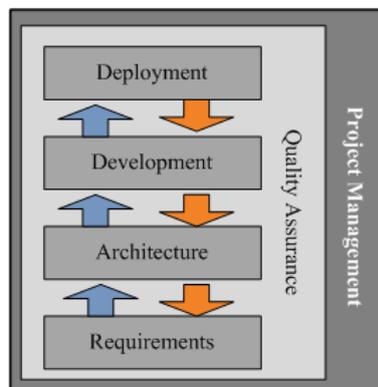


Figure 3: TME Disciplines

Project Management. The project is established applying the next sub-disciplines:

- **Development Management.** A Feasibility study is carried out to determine the project’s viability. The project scope is delimited, and first estimations of cost, time, infrastructure, and human resources needed are done. The risks are identified, and suitable responses to them are planned. Moreover, the work breakdown structure (WBS), schedule, and budget are elaborated as part of the project plan.

- **Human Resources Management.** Recruitment can be needed in which case the candidates are interviewed, and hired thereafter.

The team is organized, and its members are notified about their responsibilities and functions.

In addition, training courses for the staff of the project are planned.

- **Communication Management.** The stakeholders are identified and the information flow among them is prepared/organized.

Furthermore, the meeting frequency is established. Meetings with users are arranged and performed, and during those meetings, the major decision of the project are taken.

Meeting minutes are elaborated to report on the topics discussed in these meetings

- **Outsourcing Management.** The decision about outsourcing development is made, and as a results technical specification of purchase or engagement is created.

The suppliers are called, and their proposals are assessed in order to carry out the selection process. Finally, negotiating, awarding, and the signing of the contract with the selected supplier is done.

Quality Assurance. It aims at planning how the quality during the whole project will be managed. The following sub-disciplines are considered.

- **Quality Management.** The definition and standardization of process, documents, and work product products are set. Templates, as well as tools (check-lists, questionnaires) are created in order to evaluate the quality of work products. Also, good practices are suggested to be employed during the whole life cycle.
- **Configuration and Changes Management.** The identification, collection, classification, labelling, and maintenance of project assets version are done. Moreover, process of change request is established.

Requirements. The major project requirements are identified with users, and these are ordered by priority.

2.3.2 Elaboration

This phase is characterized by the activities of analysis and design. Therefore, it involves the following disciplines:

Project Management. It is performed through the next sub-disciplines.

- **Development Management.** At the beginning of the phase, the iteration is planned and arranged. Also the requirements to be developed during the iteration are negotiated with users.
The tasks of monitoring, controlling, and updating scope, risks, schedule, and cost are carried out.
- **Human Resources Management.** The tasks are assigned to human resources who are monitored to ensure the tasks completion.
- **Communication Management.** At the beginning of the phase, the analysts meet with users to determine the requirement to develop in the iteration and receive requests for changes.

Moreover, team meetings are held according to the settled frequency, as to update about progress, pitfalls, and support decision-making.

Quality Assurance. It focuses on the quality of design and keeping assets version. These following sub-disciplines are involved.

- **Quality Management.** At the beginning of the iteration, the Service License Agreement (SLA) is negotiated.
The quality of the design is monitored, and corrective actions can be suggested.
- **Configuration and Changes Management.** All the assets affected by changes are identified and, they must be updated accordingly All the work products have to be maintained and their versions are tracked. The tools to be used for designing are installed and customized as well.

Requirements. The requirements are gathered from users, documents, and forms. First, the use cases and then the mock-ups are created. The mock-ups helps identifying inputs and outputs as well as understanding how the interface has to be.

A glossary is created to unify concepts that can be either incompatible or different across diverse legacy systems.

Finally, use cases, mock-ups, and glossary can be refined, and the requirements list can be updated.

Architecture. It focuses on defining, depicting, and updating the application architecture. It is performed through the following sub-disciplines.

- **Business Modelling.** The organizational processes that can be reused and/or affected during development are identified.
The process description is made using Business Process Modelling Notation (BPMN) that has been chosen for being a standard notation⁴. BPMN involves process, sub-processes, activities, services, flows, and entities.
TME calls **Business Model** to each BPMN diagram, and suggests describing with BPMN each use case identified by the Requirements Discipline.
- **Integration.** The top-down description of the application and its environment is provided including services and legacy systems. This is represented with the **Domain Model** that is elaborated with UML Component diagram according to the meta-model shown in Figure 4. The meta-model represents

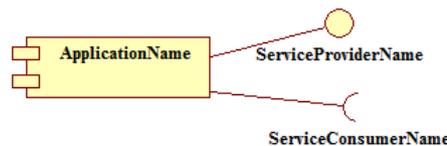


Figure 4: Domain Meta-model

the applications as components, and the services as interfaces. Relation of dependency connects the service consumed with its component, and relation of realization links up the service provider with its component.

As an example, the Domain Model of Information and Interchange System (IIS) shown in Figure 5 is proposed. The model includes the systems of the following agencies: Civil Registration Office (CROS), Ministry of Education (MES), Ministry of Health (MHS), Secretary of Information and Communication (IIS), National Police (NPS), and Ministry of Treasury (MTS). CROS provides the *birthService* and *deathService* while NPS supplies *citizenPoliceRecordService*. IIS consumes these services, and composes them into the *citizenDataService* consumed in turn by MES, MHS, and MTS. We can observe how the domain encompasses legacy systems and also produced and consumed services for them. This supports the re-usability of functionalities and services.

⁴<http://www.omg.org/spec/BPMN/2.0/>

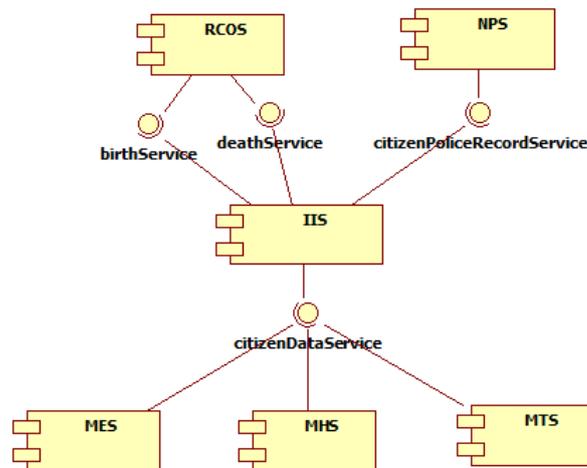


Figure 5: Example of Domain Model

The application is decomposed into modules, which are held in the **Structure Model**.

Furthermore, this sub-discipline defines the interchange standards including data format and processes. An example of standard data format is presented in Figure 6. The example is in XML Schema Definition (XSD), a format in which the service providers present the information needed to consume their services.

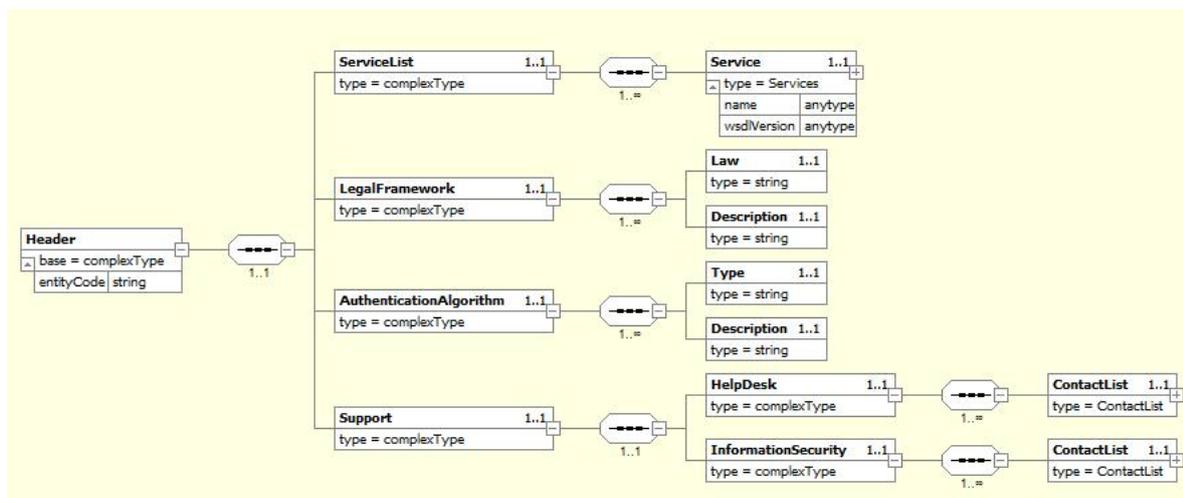


Figure 6: Example of Standard Data Format

The identification of services to be re-used is another important activity. It is expected to reduce redundancy, development time, and save costs for both maintenance and operation. TME suggests the application of techniques for service identification, but does not indicate a specific one. However, some suggestions can be found in the work of Arsanjani et al [14] which are Domain decomposition, Goal-service modelling, and Existing Asset Analysis [15], [16], [17], [18]. The techniques focus on searching a set of candidate services which might accomplish the development requirements. The search can embrace business processes, existing assets which include legacy systems, functionalities, procedures, existing services into service repository, rules and others [15]. After discovering the candidate services, they should be filtered according to pre settled criteria such as Service Litmus Test (SLT) [14], [16].

- **Testing.** The types of tests to be carried out are chosen, and the test cases are elaborated. Then, the verification of functionality, usability, supportability, reliability, and performance of software is carried out by developers and testers.

Requirements. The requirements list, use cases, mock-ups, and glossary can be updated.

Architecture. It is performed through the following sub-disciplines:

- **Business Modelling.** The business model can be updated and refined.
- **Integration.** The run-time configuration of processing nodes and its components are described in the **Deployment Model** that is elaborated using UML deployment diagram.
Besides, the domain model, the structure model, the interchange standards, and the deployment model can be refined and updated.
- **Components Specification.** The service model and the data model can be updated.

Development. It is carried out according to the design and the quality standards. The service composition is performed using an orchestration language.

Deployment. The plans of deployment and back-up are elaborated.

Once the application is finished, it is deployed in test environment.

2.3.4 Transition

Once the software is ready to be delivered, its verification, validation, and acceptance are carried out by users. Finally, it is deployed and its productive life starts.

In this phase the following disciplines are involved:

Project Management. It is performed through the following sub-disciplines.

- **Development Management.** The project plan and chart can be updated. Additionally, the process to authorize the application deployment is followed.
- **Human Resources Management.** The human resources are monitored and controlled.
- **Communication Management.** The team meeting is held to discuss about the iteration issues and lessons learned.
The analysts meet with users to verify and validate the application. Furthermore, users are trained to use the software.
- **Outsourcing Management.** The provider delivers the application that has to be verified and validated before accepting.

Quality Assurance. It is performed through the following sub-disciplines.

- **Quality Management.** The quality of deliverables is monitored and controlled, and improvements can be suggested.
Furthermore, the deliverable documentation and manual are elaborated to be given to final users.
- **Configuration and Changes Management.** The deliverable version is maintained and tracked.
The change request can be received and analysed. Finally, decisions are made on whether to apply the changes or not.
- **Testing.** The verification and validation of the software is performed by users who have to accept it before deployment.

Requirements. The requirements list is updated.

Deployment. The infrastructure where the application will be installed is verified to ensure its adequacy, and corrective actions can be suggested. Then, the software is installed in production environment, and the services are published in the service repository. The services repository provides information about business entities, the type of services, and mechanisms for accessing to them. The industrial standards of service registry are the Universal Description, Discovery and Integration (UDDI) and Electronic Business Markup Language (ebXML) though the most widely spread is UDDI [20].

In addition, the deployment can be reversed if necessary.

2.3.5 Operation

The activities focus on ensuring the effective and efficient performance of the software. Moreover, the software or part of it can be retired of the production environment; in this case, the operation of related applications has to be assured.

These activities are achieved with the following disciplines:

Project Management. It is performed through the following sub-disciplines.

- **Development Management.** It deals with warranties management as well as notification of software retirement. However, a decision to start the new development iteration can be taken to evolve the software.
- **Human Resources Management.** The staff in charge of maintenance is monitored to ensure the tasks completion.
- **Communication Management.** The team meetings can be held.
- **Outsourcing Management.** The warranty fulfilment can be requested to the provider.

Quality Assurance. It is performed through the following sub-disciplines.

- **Configuration and Changes Management.** Changes on project assets are monitored, and the version is updated if necessary.
- **Testing.** The verification and validation of the software, that have been maintained, are carried out by developers and users. Users have to accept the software before deployment.

Requirements. The glossary can be updated.

Architecture. The business model, the domain model, the interchange standards, the structure model, and the deployment model can be updated.

Development. The corrective, adaptive, and perfective maintenance are performed.

As we stated in Section 2, it is possible to adapt and substitute the constituent services to improve software performance. TME considers mechanisms to monitor the needs for adaptation. Once adaptation requirements are detected, it is possible to choose the most suitable mechanisms to carry out the adaptation. The mechanism to be implemented depends on the change that promotes these needs. Some of such mechanisms could eventually support self-adapting service based applications.

Before retiring software from the production environment, all the related legacy systems have to be examined and adjusted if necessary. Thereafter, the retirement takes place and the environment is tested again.

Deployment. Updates are deployed into production environment, and / or into the service repository. The software can be returned to a previous state if necessary.

2.4 TME and OpenUP: extensions and differences

TME has introduced various changes into OpenUP, mainly, in the life cycle, roles, and disciplines. The goal of these changes has been to provide to OpenUP with the characteristics required to support the development of SBAs. Besides, they intend to overcome some weaknesses of OpenUP such as the lack of quality assurance; a life cycle disregards maintenance and retirement; project management that does not consider outsourcing, human resource, and so on.

Considering this, TME has incorporated a new phase into its life cycle “Operation Phase”, in order to cover the gap between deployment and retirement of the application in OpenUP’s life cycle.

On one hand, TME has added the new discipline “Quality Assurance”, to assure the quality of the process and work products; and has included the Test Discipline of OpenUP into this new discipline, as a sub-discipline. On the other hand, it has given up the Environment Discipline of OpenUP, because its functions are covered by the Configuration and Change Management sub-discipline.

In addition, TME has modified various disciplines of OpenUP; the major changes have been into the Architecture and Project Management disciplines. They have been separated into sub-disciplines to specialize their management, and their work products have been incremented as well (see Table 3).

TME has introduced the new work product “mock-up” to the Requirements discipline. Since the service oriented computing involves a new programming paradigm, the Development discipline has been adjusted, including the service composition and the Deployment discipline to consider service publication into the service repository.

All these changes have affected the definition of roles, consequently. Two roles: Quality Analyst and Configuration Analyst have been added. Moreover, TME has excluded roles whose functions are covered by others, namely the: “Any”, “Course Developer”, “Product Owner”, “Technical Writer”, and “Process Engineer” roles.

The major changes incorporate in TME are shown in bold face in Tables 2 and 3.

To summarize, we highlight the improvements that TME has proposed, as follows:

- TME manages the characteristics of service-oriented computing while OpenUP does not consider these aspects. This is noted on, OpenUP’s Deployment discipline missing the services deployment and publication.
- From the software engineering perspective, TME covers the whole life cycle. During its elaboration phase various work products are created to describe the application’s architecture in more detail than the Architecture Notebook. The Architecture Notebook is the only work product of OpenUP that depicts the architecture (see Table 3).

Moreover, TME covers a larger domain by including operations outside the scope of OpenUP

- From a quality assurance viewpoint, TME manages quality through standards, definition of process for managing configuration, controlling changes, etc. However, OpenUP does not consider these aspects. OpenUP involves the test practices, but does not envisage the user acceptance.
- In the project management dimension, TME proposes to manage human resource, communication, outsourcing and development as part of the Project Management discipline. By being loyal to agile philosophy, OpenUP manages the project informally, and as a result being loyal with agile philosophy. and as a result, the activities of controlling schedule compliance, cost, scope, accomplished tasks, human resources, outsourcing, and quality are not considered.

For all of these reasons, TME seems to be an interesting process in order to produce and maintain high quality service-based applications whose life cycle is supported by a well-defined project management.

3 Evaluation Plan

The assessment is inspired in the evaluation framework proposed by Grau et al [3] whose baseline is the work of Cernuzzi and Rossi [21]. The evaluation plan is composed of three phases: Planning, Execution and Analysis.

Table 2: Differences between TME and OpenUP

Item	OpenUP	TME
Life cycle	<ul style="list-style-type: none"> • Inception • Elaboration • Construction • Transition 	<ul style="list-style-type: none"> • Inception • Elaboration • Construction • Transition • Operation
Disciplines	<p>Project Management</p> <p>Requirements Architecture</p> <p>Development Deployment</p>	<p>Project Management</p> <ul style="list-style-type: none"> • Development Management • Human Resources Management • Communication Management • Outsourcing Management <p>Requirements Architecture</p> <ul style="list-style-type: none"> • Business Modelling • Integration • Components Specification <p>Development Deployment Quality Assurance</p> <ul style="list-style-type: none"> • Quality Management • Configuration and Change Management • Testing
Roles	<ul style="list-style-type: none"> • Basic Roles <ul style="list-style-type: none"> – Stakeholder – Project Manager – Analyst – Architect – Developer – Tester – Any • Deployment Roles <ul style="list-style-type: none"> – Course Developer – Deployment Engineer – Deployment Manager – Product Owner – Technical Writer – Trainer • Environment Roles <ul style="list-style-type: none"> – Process Engineer – Tool Specialist 	<ul style="list-style-type: none"> • Basic Roles <ul style="list-style-type: none"> – Stakeholder – Project Manager – Analyst – Architect – Developer – Tester – Quality Analyst • Deployment Roles <ul style="list-style-type: none"> – Deployment Engineer – Deployment Manager – Trainer • Environment Roles <ul style="list-style-type: none"> – Tool Specialist – Configuration Analyst

3.1 Planning Phase

The Planning Phase describes how TME evaluation will be done and includes its goals, measures, scales, formulas, and steps. More specifically, it encompasses the following steps:

3.1.1 Definition of Research Questions

The goal of this evaluation step is to identify the relevant questions that will be answered.

Table 3: Work Products of TME and OpenUP

OpenUP	TME
Project Management <ul style="list-style-type: none"> • Risk List • Work Items List • Iteration Plan • Project Plan 	Project Management <ul style="list-style-type: none"> • Project Plan • Iteration Plan • Interchange Agreements • Task Assignment • Progress Report • Meeting Minute • User Training • Technical Specification • Suppliers Evaluation • Contract • Monitoring Suppliers • Guarantee
Requirement <ul style="list-style-type: none"> • Glossary • Use Case Model • Use Case • System Wide Requirement 	Requirements <ul style="list-style-type: none"> • Requirements List • Use Case Diagram • Mock-ups • Glossary
Architecture <ul style="list-style-type: none"> • Architecture Notebook 	Architecture <ul style="list-style-type: none"> • Business Model • Domain Model • Interchange Standard • Structure Model • Deployment Model • Data Model • Service Model
Environment <ul style="list-style-type: none"> • Project Defined Process • Tools 	
Development <ul style="list-style-type: none"> • Implementation • Build • Developer Test • Design 	Development <ul style="list-style-type: none"> • Source Code • Service Composition
Deployment <ul style="list-style-type: none"> • Product Documentation • Support Documentation • User Documentation • Training Materials • Back out Plan • Deployment Plan • Release Communication • Release Control 	Deployment <ul style="list-style-type: none"> • Service Repository • Deployment Plan • Back up Plan
Test <ul style="list-style-type: none"> • Test Case • Test Script • Test Log 	Quality Assurance <ul style="list-style-type: none"> • Service Level Agreement (SLA) • Standards • User Documentation • Support Documentation • Training Materials • Change Request • Version Control • Test Cases • User Acceptance

To do so, we focus on answering the following questions:

Q.1 Does TME cover the intrinsic features of the service oriented computing better than the previous

organization process?.

Q.2 Does TME address the major characteristics of a software engineering process better than the previous organization process?.

Q.3 Does TME address activities for project management better than the previous organization process?.

Q.4 Does TME support the sustainability of applications better than the previous organization process?.

3.1.2 Definition of Measuring Goals

With the research questions defined, for each of these questions we proceed to define a set of goals using GQM technique [22]. GQM separates goals in sub-goals hierarchically, as a tree where the leaves are the metric-associated questions used to measure the goals' achievement. The major goals with their sub-goals are described as follows (see Table 4).

Table 4: Tree of Goals

SO Service Oriented

- SO.1 Service Description
- SO.2 Service Implementation
- SO.3 Interchange Information

SE Software Engineering

- SE.1 Life Cycle
 - SE.1.1 Requirement
 - SE.1.2 Analysis and Design
 - SE.1.3 Development and Deployment
- SE.2 Quality Assurance
 - SE.2.1 Quality Management
 - SE.2.2 Configuration and Change Management
 - SE.2.3 Test

PM Project Management

- PM.1 Development Management
- PM.2 Communication Management
- PM.3 Human Resource Management
 - PM.3.1 Recruitment of Human Resources
 - PM.3.2 Role Assignment
 - PM.3.3 Monitoring and Controlling Human Resource
- PM.4 Procurement management

SU Sustainability

- SU.1 Documentation
- SU.2 Low Cost of Implementation

SO Service Oriented. Is related to intrinsic characteristics of service computing and data interchange issues. Its major sub-goals are the following:

SO.1 Service Description. Examines how services, relationships among services, service roles, parameters, messages, service level agreements (SLA), and service composition are described.

SO.2 Service Implementation. Finds out how service deployment, service publication, and enterprise service bus (ESB) are carried out.

SO.3 Interchange Information. Inquires about the processes and data format defined to support the interoperability among legacy systems.

SE Software Engineering. Involves the qualities that are required to be present for any software engineering process. Its major sub-goals are the following:

SE.1 Life Cycle. Covers the software life cycle from the conception of its original ideas until the software is abandoned. The life cycle encompasses the following sub-goals:

SE.1.1 Requirement. Inquires on how the requirements are managed.

SE.1.2 Analysis and Design. Examines how the architecture of the software is designed and documented.

SE.1.3 Development and Deployment. Ascertain how the construction and deployment of the software are done.

SE.2 Quality Assurance. Focuses on the quality of development process and work products, and it is composed of the following sub-goals:

SE.2.1 Quality Management. Examines the quality of the development process; if the work products have templates; if good practices of development are implemented.

SE.2.2 Configuration and Change Management. Inquires how the changes are managed; if the versions of the project assets are kept up, and they are updated when necessary.

SE.2.3 Test. Inquires about the verification and validation of the software.

PM Project Management. Involves the fundamental activities necessary to manage any type of project according to project management standard [8]. Its major sub-goals are the following:

PM.1 Development Management. Encompasses the Integration Management, Scope Management, Time Management, and Cost Management of the standard of project management [8]. Consequently, it verifies how these areas of processes are managed.

PM.2 Communication Management. Ascertain how the communication is carried out among the stakeholders, and how the information is distributed.

PM.3 Human Resource Management. Verifies how staff is managed, and it includes the following sub-goals.

PM.3.1 Recruitment of Human Resources. Verifies how the processes of recruitment, selection, and hiring of human resources are done.

PM.3.2 Role Assignment. Ascertain whether the human resources know their responsibilities, and the tasks that they have to carry out.

PM.3.3 Monitoring and Controlling Human Resource. Verifies if human resources are monitored, and evaluated.

PM.4 Procurement Management. Encompasses identifying, and selecting sellers as well as awarding contracts. It also manages the relationship with sellers, while monitoring contract execution. It is referred not only to infrastructure purchases, but also to outsourcing.

SU Sustainability is the capability of organizations to support long-term software maintenance [6]. Its major sub-goals are the following:

SU.1 Documentation. Examines if TME proposes models and documents to cover the different aspects of the software development.

SU.2 Low Cost of Implementation. Inquires about the strategies implemented to save development costs.

Table 5: Measures

Measures		
Attribute	Frequency	Aggregation
Code	FR	AG
Description	Used to measure the questions that are the leaves of the tree	Applied to non-leaves and it is the arithmetic mean of the child nodes
Type	Direct	Indirect
Property Measured	Occurrence	Occurrence
Type Scale	Likert	Likert
Scale Value	Positive Numbers	Positive Numbers
Interval Value	[0,4]	[0,4]

3.1.3 Definition of Measures and Scales

Two types of measurements are done: direct and indirect. The former is used for leaves while the latter for the others. For each of these types the following measures have been established: Frequency and Aggregation. These measures are described in Table 5.

The scale of the Aggregation Measure encompasses the range of values that are interpreted as the degree of satisfaction for goal achievement. The interpretation of the Frequency and Aggregation Measures is shown in Table 6.

Table 6: Interpretation of Frequency and Aggregation Measures

Measures			
Scale	Frequency	Aggregation	Description
No Apply	0	$0 \leq \bar{x} < 1; 0 \leq G < 1$	The evaluation can not performed.
Never	1	$\bar{x} = 1; G = 1$	It has never been achieved.
Rarely	2	$1 < \bar{x} \leq 2,5; 1 < G \leq 2,5$	It has been achieved less than 50% or equal of 50% of the time.
Normally	3	$2,5 < \bar{x} < 4; 2,5 < G < 4$	It has been achieved more than 50% of the time.
Always	4	$\bar{x} = 4; G = 4$	It has been achieved the 100% of the time.

3.1.4 Definition of Measurement Process

The leaves are the questions answered by practitioners. These are measured by giving it a value found in the Frequency Scale (see Table 6). Once all the questions are answered, the value of non-leaves (goals and sub-goals) is calculated by finding out the arithmetic mean (\bar{X}) obtained with Formula 1. Thus, the value of a non-leaf is equal to the arithmetic mean calculated from its child nodes.

$$\bar{X} = \frac{\sum_{k=0}^n a_i}{n} \quad (1)$$

a is the child node
i is the depth where the child node is located
n is the number of child nodes

Different practitioners can be interviewed for a given case study. Each of these practitioners responds a questionnaire that contains a set of questions that are grouped according to the goals of the tree of goals, and they aim to measure its achievement (see Table 4). The possible responses to each question is in a frequency scale (see Table 6), as the example shown in Figure 8 shows.

The questions chosen are matched with their numerical value in a measurement template. Figure 9 shows the measurement template for the example of Figure 8 that corresponds to the questionnaire E_2 .

SO Service Oriented	
SO.1 Service Description	
<input checked="" type="checkbox"/> SO.1-Q.1	Are the services relationship shown?
<input type="checkbox"/>	No Apply
<input type="checkbox"/>	Never
<input type="checkbox"/>	Rarely
<input type="checkbox"/>	Normally
<input checked="" type="checkbox"/>	Always
<input type="checkbox"/> SO.1-Q.2	Are the services role shown?
<input type="checkbox"/>	No Apply
<input type="checkbox"/>	Never
<input type="checkbox"/>	Rarely
<input type="checkbox"/>	Normally
<input checked="" type="checkbox"/>	Always
<input type="checkbox"/> SO.1-Q.3	Are the services input shown?
<input type="checkbox"/>	No Apply

Figure 8: Questionnaire Example

Code	Goals		CC	E ₁	E ₂
	Total	AG	1,4	2,89	2,96
SO	Service Oriented	AG	1	2,56	2,33
SO.1	Service Description	AG	1	2,67	3
SO.1-Q.1	Are the services relationships shown?	FR	1	3	4
SO.1-Q.2	Are the services roles shown?	FR	1	2	4
SO.1-Q.3	Are the services inputs shown?	FR	1	4	4
SO.1-Q.4	Are the services outputs shown?	FR	1	4	4
SO.1-Q.5	Is the SLA considered?	FR	1	1	1
SO.1-Q.6	Is the services composition described?	FR	1	2	1
SO.2	Service Implementation	AG	1	2	1

Figure 9: Template of Measurement Example

Using this the value of sub-goals, goals, dimensions, and total value are calculated. Finally, the total value of the case study is the upshot of combining the dimensions and the total value of all the questionnaires. The combination is done through obtaining the geometric mean (G) with Formula 2.

$$G = \sqrt[M]{R_1 * R_2 * \dots * R_M} \quad (2)$$

G is the geometric mean
 R can be the value of the dimension or the total value
 M is the number of individual questionnaires

The results of Formulas 1 and 2 are given using the Aggregation Scale (see Table 6).

This linear model is very simple and will probably be replaced by a more powerful aggregation mechanism, like the Logic Scoring of Preference [23]. Moreover, according to the objective of the stakeholder it is possible to associate a priority or different weights to specific goal and /or any goal of the tree to better cover significant aspects for the assessment. Such differentiation accommodates the needs of assigning priorities to a specific aspects and a more sophisticated comparative analysis. However, we leave this consideration for future works.

3.2 Execution Phase

The goal of this phase is the execution of the evaluation plan by running case studies. From the set of evaluation studies already performed for our case study, we will described the first one in the following section.

3.2.1 Case Study: Information Exchange System (IES)

TME was first evaluated in a case study regarding an Information Exchange System (IES) developed in a public institution in Paraguay (PIP). The PIP was set up in April 2012 and, before using TME its development process used to be informal, according to the information collected from its quality analyst in October 2012.

The PIP used TME to manage the development of the IES, which had as its main goal the interoperability of legacy systems in public agency through the use of service-oriented computing. The operation of the mentioned IES is as follows: first, it consumes services from the providers; second, it integrates these services into a new single service; and thirdly, it provides this new service (see Figure 6). It will take some years to complete the full development of the IES. Therefore, this first evaluation encompasses the period from October 2012 to March 2013.

During this period, four services were developed that exchanged citizens' data from five legacy systems that belonged to different public agencies. Each agency has a development team that is represented by a leader. The leader of the PIP is in charge of managing the communication among all leaders, and the PIP is responsible for defining the interoperability standards such as data format (see Figure 6), request process and to access to a service etc. The development team is in charge of services development for the owner of the legacy systems (provider or consumer). At the moment, there are three consumers, but this number is planned to increase in the future.

There have been six teams in charge of developing the IES. However, our analysis only involves the PIP team that was composed of a project manager, a leader, an architect, a quality analyst, and three developers. The project manager and one of the developers have a degree in computer sciences; the leader, the architect, the quality analyst, and one of the developers are system engineers; one developer is a student; and finally, the quality analyst is a Ph.D. student.

As far as professional experience is concerned, the project manager, the leader, the architect, the quality analyst, and one of the developer have been working in diverse projects, for more than ten years. One developer worked in a project before the IES, while for the other developer this is its first work.

The first part of the IES development was financed by a sponsor who monitored the development and demanded that his own quality standards to be followed. The sponsor support lasted until March of 2013.

The IES construction was delayed several times before due to some external factors such as political reasons, lack of a law that regulates exchange of sensitive data, and staff changes. The regulating of exchange of sensitive data was only passed on January 2013, which facilitated the decision making regarding the system.

Definition of the Assessment Scope. The evaluation encompassed from the Inception Phase to the Transition Phase, over the course of six months. Focusing above all in the interoperability of legacy systems that can belong to different public agencies, the project management, and the sustainability of the project.

Since we were able to implement TME only in the PIP, the assessment considers only this agency.

The interviewees were selected according to their knowledge on the whole process proposed by TME. Furthermore, it was required that they were part of the IES team of during the period from October 2012 to March 2013. Only the team leader and the quality analyst fulfilled these requirements.

Data Collection. The questions associated to each goal or sub-goal were presented in a questionnaire that was answered by the case study participants.

The questionnaires were then sorted out in Control or Experimental categories, according to the process that they report on. The control questionnaires were answered by the control group, these inform about the process of development before applying TME in the PIP. The experimental questionnaires were answered by the experimental group (i.e the team leader and quality analyst) and they measure TME's performance.

The questionnaire used for this case study is composed of 83 questions that measure 20 sub-goals. The questionnaire is written in the local language (Spanish), and it takes 20 minutes approximately to be filled by the practitioners. An example of the format of the questionnaires is shown in Figure 8. We assign a numeric value in the Frequency scale to each response of the questionnaire. Having completed the questionnaire, the numeric value of each selected response is registered in a template to calculate the results (see Figure 9). The control questionnaire (CC) was answered by the quality analyst before applying TME. After six

months of TME implementation, its performance was examined from the viewpoint of team leader (E_1) and the quality analyst (E_2) who filled in the experimental questionnaires.

It is worth noting that for the purpose of this study all the goals and sub-goals have the same weight. However, according to the stakeholder's objectives of the TME allows to assign different weight to specific goal and/or any goal of the tree.

Measuring Process. Once the questionnaires were completed, their responses were matched to the values of the Frequency Measure (see Table 6). Subsequently, the values of goals and sub-goals were obtained according to the process defined in Section 3.1.4.

As the questionnaires are clustered by type, next step was to follow the process defined in Section 3.1.4.

According to the range of values in which the numerical results are to be interpreted, these were matched to the scale of Aggregation Measure. The tree of goals with the results of the three questionnaires (i.e, 1 before TME and 2 after adopting TME) is shown in Table 7. Note that the results are in aggregation measure, and the questions (which are measured in frequency measure) are not presented.

The results for each dimension in Aggregation Measure are shown in Table 8, and the final result is "Normally".

Table 7: The tree of goals with the results of the three questionnaires

Code	Goals	CC	E_1	E_2
	Root	1,4	2,89	2,96
SO	Service Oriented	1	2,56	2,33
SO.1	Service Description	1	2,67	3
SO.2	Service Implementation	1	2	1
SO.3	Interchange Standard	1	3	3
SE	Software Engineering	1,42	2,78	2,75
SE.1	Life Cycle	1,27	3,01	3,03
SE.1.1	Requirement	1,5	2,83	4
SE.1.2	Analysis and Design	1,33	3	3,5
SE.1.3	Development and Deployment	1	3,2	1,6
SE.2	Quality Assurance	1,62	2,54	2,46
SE.2.1	Quality Management	1,2	2,8	3,4
SE.2.2	Configuration and Change Management	1,83	2,33	4
SE.2.3	Test	1,66	2,5	0
PM	Project Management	2,2	2,48	2,57
PM.1	Development Management	3	3,57	3,86
PM.2	Communication Management	1	1,25	1,5
PM.3	Human Resource Management	3,3	3,17	3,33
PM.3.1	Recruitment of Human Resources	4	4	4
PM.3.2	Role Assignment	2,5	2,5	2,5
PM.3.3	Monitoring and Controlling Human Resource	3,5	3	3,5
PM.4	Procurement Management	1,5	1,63	3,25
SU	Sustainability	1	3,75	3,83
SU.1	Documentation	1	3,5	3,67
SU.2	Implementation Cost	0	3	4

3.3 Assessment Validity

The IES mentioned in this study is the first service-based application implemented in the public sector in Paraguay. For this reason, we could not compare TME with another development process for service-based application in the same context.

Table 8: Upshot of Dimensions

Code	Goals	Frequency Scale				Aggregation Scale			
		CC	E_1	E_2	E_1, E_2	CC	E_1	E_2	E_1, E_2
	Root	1,4	2,89	2,96	2,92	Rarely	Normally	Normally	Normally
SO	Service Oriented	1	2,56	2,33	2,44	Never	Normally	Rarely	Rarely
SE	Software Engineering	1,42	2,78	2,75	2,76	Rarely	Normally	Normally	Normally
PM	Project Management	2,2	2,48	2,57	2,52	Rarely	Rarely	Normally	Normally
SU	Sustainability	1	3,75	3,83	3,79	Never	Normally	Normally	Normally

Since the questionnaires are the tool used to measure the achievement of the goals, their reliability affect the validity of the evaluation. For this reason, they have been elaborated to embrace all the aspects related with the goals and their questions strive to be accurate and easy to understand. It is also important to note that the validity is constrained by the skills and perception of the interviewees.

Another limitation is that only two persons were interviewed. Even though these two persons are considered proficient professional, and they have been in charge of important projects in public and private sector in Paraguay.

3.4 Analysis Phase

Even though one study is not enough to support a strong claim, independent observers have managed to observe improvements in their processes due the use of TME.

It is worth mentioning that almost all the dimensions achieved the value “Normally”, and particularly the Sustainability goal achieved values very close to “Always” (see Table 8):

Service Oriented Dimension. The score given to this dimension is “Rarely”, this means that less than the 50% of the time, the suggestions of TME have not been observed (see Table 8). Nevertheless, if we consider its the numerical value of “2,44” we can appreciate that it is quite close to “2,5” that represents “Normally”.

Even though TME suggests signing the Service Level Agreement (SLA), having a service repository (UDDI), and implementing techniques for concepts unification; they have not been implemented by the PIP (see Table 1). Table 7 shows that the “Service Implementation” is the weakest goal of this dimension. Moreover, the quality analyst did not give an answer on the “Service Implementation”, because she did not participate in the development.

Previous processes of the PIP have obtained a “Never” mark, so when comparing this with the “Rarely” mark obtained through TME, we can observe that TME has improved the development process of the PIP in response to the question research “Q.1”.

Software Engineering Dimension. There are different responses in Requirement and Configuration, and Change Management due to practices and documents proposed to them; but they have not always been considered by the developers.

The quality analyst suggested to follow certain standards during the development, but it was not her responsibility to guarantee that the developers really followed the suggested standards.

Testing has not always been carried out by other people besides developers and there was not a responsible person for accepting the development before deploying. Besides, verification and validation practices were not suggested at that moment.

This dimension has obtained the mark “Normally”, which means that more than the 50% of the time the suggestions of TME in the “Software Engineering” area have been observed. This result suggests that TME has improved the previous process of the PIP, considering the question research “Q.2” (see Table 8).

Project Management Dimension. Various processes and documents were not followed and completed because they were unknown by the majority of the staff due to a fault in Communication Management. This situation is reflected in the very low score in “Communication Management”, and affected the Role Assignment because some members of the team were not notified about their responsibilities.

Part of the IES development was outsourced, and the contract with the provider was signed before applying TME. For this reason, several of TME's suggestions were not been considered.

Sustainability Dimension. Even though the models were not easily understandable to some of the staff, the documentation was usually updated.

The PIP determined that all the tools used for developing IES should be open source. These tools include database, programming language, modelling language, project management software, version control software, etc. Moreover, TME includes standard notation such as UML and BPMN.

However, the lack of documentation of the applications developed at the PIP before applying TME has hindered their maintenance.

Since the previous process of the PIP obtained the "Never" mark, when comparing it to the "Normally" mark obtained with TME we can consider that it has become more sustainable than the previous process in response to research question "Q.4" (see Table 8).

4 Ongoing works and Conclusion

Among the various Agile Methods, OpenUP has been chosen to be the baseline of Tape Mbo'e (TME). Furthermore, TME is a service oriented process that modified the life cycle and the disciplines of OpenUP. The former covers the maintenance and retirement of systems while the latter includes the service oriented computing features, the quality assurance, and improvements to the project management. These changes have affected the definition of roles and work products as well. Consequently, TME introduces new roles, gives up some existent roles, and includes new work products.

On the one hand, TME aims at proposing a general process. For this reason, the processes for addressing more specific aspects such as the publication of services on the service repositories have not been included by this version of TME. It is worth noting that TME is not a methodology, consequently, specific aspects like techniques for service identification, criteria for reusing functionalities, metrics to measure the complexity of the development, etc. are not presented here. However, as mentioned in section 2.3 for service identification, TME consider the adoption of other existing proposal.

To evaluate the TME, we prepared an evaluation plan that is described in Section 3. This plan includes goals, measures, and scales. In addition, it applies Goal-Question Metric to organize the overall observation structure and guide the case studies execution.

This paper has presented the results of the first case study evaluated with the TME plan, which is also the first application of rigorous experimental evaluation in the public sector in Paraguay. The case study is the Information Exchange System (IES) which belongs to a public Paraguayan agency. The development of IES had been addressed using TME and its performance was assessed. In the evaluation TME has demonstrated to contribute to the agency's improvement when compared to the previous process as it is shown in Section 3.4.

The assessment of other case studies is still on going. These represent the experiences of using TME in different public and private organizations in Paraguay. The results of those case studies are currently being integrated by using meta-analysis (non-parametric aggregation) through Vote Counting [24]. Since this paper intended to discuss just one of these case studies, the use of vote counting is not feasible.

As future work we are going to present the results of assessing TME with vote counting. In addition, in order to outfit TME with a set of suggestions about existing techniques, we have researched about techniques for service identification and criteria for their selection, techniques to estimate the project size, criteria to reuse existing assets, etc.

The contributions of this paper are the description of the evaluation plan and its application to the first assessment of TME's performance in a real case study in Paraguay.

References

- [1] V. Andrikopoulos, A. Bucchiarone, E. D. Nitto, R. Kazhamiakin, S. Lane, V. Mazza, and I. Richardson, "Service engineering," in *S-CUBE Book*, 2010, pp. 271–337.

- [2] S. Lane and I. Richardson, "Process models for service-based applications: A systematic literature review," *Inf. Softw. Technol.*, vol. 53, no. 5, pp. 424–439, May 2011. [Online]. Available: <http://dx.doi.org/10.1016/j.infsof.2010.12.005>
- [3] I. Grau, L. Cernuzzi, and A. Villafiorita, "Analysing service oriented methodologies for sustainable applications," in *The 32nd International Conference of the Chilean Computer Science Society (SCCC)*, November 2013.
- [4] I. M. Grau, L. Cernuzzi, and A. Villafiorita, "Tape Mbo'e TME: A service oriented process," in *The 32nd International Conference of the Chilean Computer Science Society (SCCC)*, November 2013.
- [5] I. Grau, G. Travassos, L. Cernuzzi, and A. Villafiorita, "Tape Mbo'e: A first experimental assessment," in *Actas del XVII Congreso Iberoamericano en Software Engineering, CIbSE 2014, (Experimental Software Engineering Latin American Workshop - ESELAW)*, April 2014.
- [6] B. Eshete, A. Mattioli, A. Villafiorita, and K. Weldemariam, "ICT for Good: Opportunities, Challenges and the Way Forward," in *Fourth International Conference on Digital Society, 2010. ICDS '10.*, feb. 2010, pp. 14–19.
- [7] I. Christou, S. Ponis, and E. Palaiologou, "Using the agile unified process in banking," *IEEE Software*, vol. 27, pp. 72–79, may-june 2010.
- [8] *A Guide to the project management body of knowledge*. Project Management Institute, Inc, 2008.
- [9] K. Schwaber, *Agile Project Management With Scrum*. Redmond, WA, USA: Microsoft Press, 2004.
- [10] P. Deemer and G. Benefield, "The scrum primer. An introduction to agile project management with Scrum," 2007. [Online]. Available: <http://www.rallydev.com/documents/scrumprimer.pdf>
- [11] S. Ambler, *Agile Modeling: Effective Practices for eXtreme Programming and the Unified Process*. New York, NY, USA: John Wiley & Sons, Inc., 2002.
- [12] K. Beck and C. Andres, *Extreme Programming Explained: Embrace Change*. Addison-Wesley Professional, 2004.
- [13] P. Abrahamsson, O. Salo, J. Ronkainen, and J. Warsta, "Agile software development methods - review and analysis," VTT PUBLICATIONS, Tech. Rep. 478, 2002.
- [14] A. Arsanjani, S. Ghosh, A. Allam, T. Abdollah, S. Gariapathy, and K. Holley, "SOMA: a method for developing service-oriented solutions," *IBM Systems Journal*, vol. 47, pp. 377–396, July 2008. [Online]. Available: <http://dx.doi.org/10.1147/sj.473.0377>
- [15] Y. Tian, Y. Su, and X. Zhuang, "Research on service identification methods based on SOA," in *2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)*, vol. 6, Aug. 2010, pp. V6–27 – V6–31.
- [16] N. Zhou, L. Zhang, Y. Chee, and L. Chen, "Legacy asset analysis and integration in model-driven soa solution," in *2010 IEEE International Conference on Services Computing (SCC)*, ser. SCC '10, Washington, DC, USA, July 2010, pp. 554–561.
- [17] L. Zhang, A. Arsanjani, A. Allam, D. Lu, and Y. Chee, "Variation oriented analysis for SOA solution design," in *IEEE SCC*. IEEE Computer Society, 2007, pp. 560–568.
- [18] K. Levi and A. Arsanjani, "A goal-driven approach to enterprise component identification and specification," *Commun. ACM*, vol. 45, no. 10, pp. 45–52, Oct 2002. [Online]. Available: <http://doi.acm.org/10.1145/570907.570930>
- [19] V. de Castro, E. Marcos, and B. Vela, "Representing WSDL with extended UML," *Revista Colombiana de Computación - RCC*, 2004.

- [20] W. T. Tsai, X. Zhou, Y. Chen, B. Xiao, R. Paul, and W. Chu, "Roadmap to a full service broker in service-oriented architecture," in *Proceedings of the IEEE International Conference on e-Business Engineering*, ser. ICEBE '07. Washington, DC, USA: IEEE Computer Society, 2007, pp. 657–660. [Online]. Available: <http://dx.doi.org/10.1109/ICEBE.2007.95>
- [21] L. Cernuzzi and G. Rossi, "On the evaluation of agent oriented modeling methods," in *Proceedings of Agent Oriented Methodology Workshop*, 2002, pp. 21–30.
- [22] V. Basili and D. Weiss, "A methodology for collecting valid software engineering data," *Software Engineering, IEEE Transactions on*, vol. SE-10, no. 6, pp. 728–738, 1984.
- [23] J. J. Dujmovi'c, "The LSP method for evaluation and selection of computer and communication equipment," in *Proceedings of MELECON'87, Mediterranean Electro technical Conference and 34th Conference on Electronics, IEEE/RIENA*, 1987, pp. 251–254.
- [24] G. V. Glass, "Primary, secondary, and meta-analysis of research," *Educational Researcher*, vol. 5, no. 10, pp. 3–8, 1976. [Online]. Available: <http://dx.doi.org/10.2307/1174772>