

# Quality Assurance in the ERP5 Development Process

Rogério Atem de Carvalho<sup>1</sup>, Renato de Campos<sup>2</sup> and Rafael Manhaes Monnerat<sup>3</sup>

<sup>1</sup>Federal Center for Technological Education of Campos (CEFET Campos), R. Dr. Siqueira, 273, Campos/RJ, CEP 28030-130, Brazil [ratem@cefetcampos.br](mailto:ratem@cefetcampos.br)

<sup>2</sup>Sao Paulo State University (UNESP).Av. Eng. Luiz Edmundo C. Coube n 14-01, Bauru, SP, Brazil [rcampos@feb.unesp.br](mailto:rcampos@feb.unesp.br)

<sup>3</sup>Nexedi SARL, Bd. Clémenceau 59700, Marcq-en-Baroeul, France [monnerat@cefetcampos.br](mailto:monnerat@cefetcampos.br)

**Abstract.** The design and implementation of an ERP involves capturing the information necessary for implementing a system that supports integrated enterprise management, starting at the enterprise modeling level and finishing at the coding level. Unfortunately, in both academic and industrial communities, large quantities of papers focus on ERP deployment management, keeping specific development issues aside most of times. Research on specific techniques for developing ERP software – open source or proprietary, is rather deficient. This paper aims to help filling this gap by presenting a development process for the open source ERP5 system, highlighting the Quality Assurance (QA) techniques used, and the tools that support it. The proposed process covers the different abstraction layers involved, and supplies customized Enterprise, Requirements, Analysis, Design, and Implementation workflows. Each of these workflows is accompanied by one or more QA activities to assure the quality of every modeling and implementation artifact delivered.

**Keywords:** *Enterprise engineering, Enterprise Resources Planning (ERP), Software development processes, Software quality assurance, Free/open source software*

## 1. INTRODUCTION

Modeling an ERP software means to deal with the aspects related to the different abstraction layers that must be taken into account in enterprise-integrated management. The ultimate goal of developing an ERP system should be going from the highest abstraction level considered, enterprise modeling, down to code generation, without losing modeling information, guaranteeing that the software is in complete conformity with business requirements. To accomplish this, it is necessary to define a process that can keep modeling information during its execution and that supplies a high-quality final product. The analysis and documentation of business and software requirements by means of models are essential for the system development, making necessary the use of proper techniques and tools [1]. In this sense, a modeling architecture that properly contemplates business processes aspects can facilitate reuse and promote better functionality, better performance, and a better system understanding [2].

On the other hand, Free/Open Source ERP (FOS-ERP) are increasingly gaining acceptance due to their lower costs and the perception that if customization is inevitable, why not adopt a solution that exposes its code to the adopting organization, which can freely adapt the system to its needs [3]? For FOS-ERP, modeling methods have their importance increased, since they can empower the availability of source code by extending and changing it in a way adherent to enterprise models, which can bring more innovation to integrated management.

However, FOS-ERP projects currently don't follow enterprise systems modeling techniques [3]. Moreover, they lack a more including Software Quality Assurance (SQA) approach, like all other open source projects – an exploratory study had show that SQA methods in FOSS projects are limited to testing, bug tracking, and Software Configuration Management [4], in other words, they only use code-related SQA techniques.

This paper aims to help filling these gaps by presenting a development process for the FOS-ERP ERP5, highlighting the SQA techniques used, and the tools that support it – a work initially proposed in [5]. Moreover, since that large quantities of papers focus on ERP deployment project management and deployment techniques [6], and research on specific techniques for developing ERP software, in special in a quality driven fashion, is rather deficient, this article also aims to contribute by discussing aspects of ERP development. The next section introduces briefly ERP5 main concepts, the following describe process' phases, highlighting SQA techniques used, and finally conclusive remarks are presented at the end.

## 2. ERP 5

ERP5 aims at offering software for integrated management based on the open source Zope platform, written in the Python scripting language [7]. This platform delivers an object database (ZODB), a workflow engine (DCWorkflow), and rapid GUI scripting based on XML. Additionally, ERP5 incorporates data synchronization among different object databases, through the implementation of the SyncML XML based protocol, and a object-relational mapping scheme that allows much faster object search and retrieval and also analytical processing and reporting. ERP5 is named after the five core business entities that define its Unified Business Model (UBM, Figure 1):

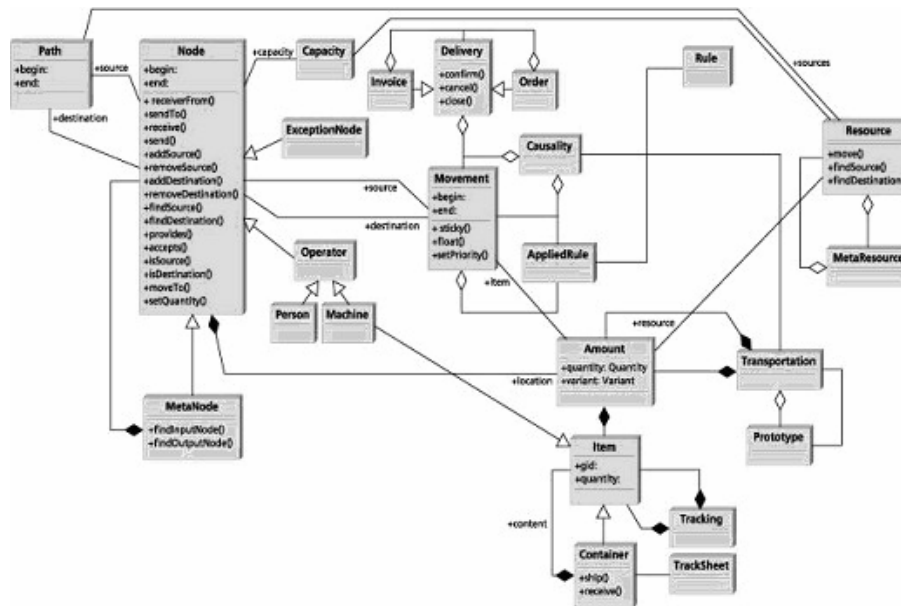
*Resource*: describes an abstract resource in a given business process (such as individual skills, products, machines etc).

*Node*: a business entity that receives and sends resources. They can be related to physical entities (such as industrial facilities) or abstract ones (such as a bank account). Metanodes are nodes containing other nodes, such as companies.

*Path*: describes how a node accesses needful resources.

*Movement*: describes a movement of resources among nodes, in a given moment and for a given period of time.

***Item*: a physical instance of a resource.**



**Figure 1. ERP5 Unified Business Model. The Five Main Classes Form the Basis for Creating New ERP5 Instances**

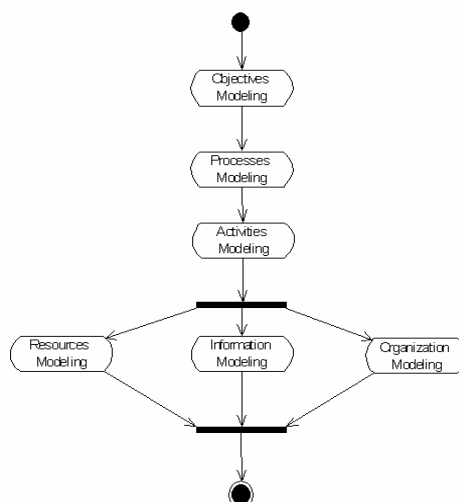
The structure of ERP5 instances is defined through mappings of the particular domain concepts to the five core concepts and supportive classes or, in very rare cases, through the extension of the UBM. This mapping is documented by a proper instance's lexicon. Its behavior is implemented through workflows, which implement the business processes, and consider the concept of Causalities (chains of related events). Very flexible and extensible modules, called Business Templates, are also provided for Accounting, Production Planning, Payroll, Finance, MRP, CRM, Trading, Electronic Commerce, Reporting, and others.

ERP5 development process covers the different abstraction layers involved, and supplies customized workflows and SQA techniques. The process is based on the Generalized Enterprise Reference Architecture and Methodology (GERAM), which provides a description of all elements recommended in enterprise engineering and a collection of tools and methods to perform enterprise design with success [8]. Following the classical – but still effective – work of McCall [9], next sections will present SQA techniques and tools used in each workflow.

### 3. ENTERPRISE MODELING

This workflow stands between Concept and Requirements phases of the Unified Process [10], and concentrates on the modeling of function, information, resources, and organization views, according to the GERAM modeling framework. For the sake of addressing enterprise integration [11], models can be built on top of CIMOSA [12]

or Eriksson & Penker approaches [13], depending on the kind of enterprise being modeled and the preferences of modelers. The Enterprise Modeling workflow consists of the activities shown in Figure 2, and can be summarized as follows:



**Figure 2. Enterprise Modeling Workflow**

1. Objectives Modeling: define the strategic objectives of the entity.
2. Process and Activities Modeling: define the behavioral and functional aspects of the organization (Business Modeling Discipline).
3. Resources Modeling: describes the human, informational and technological resources.
4. Information Modeling: describe, using a high abstraction level, the information handled by the organization.
5. Organization Modeling: describe the structure of the organization.

SQA Techniques: model documenting, using both textual and diagrammatic modeling artifacts. Model quality is guaranteed by Formal Technical Reviews.

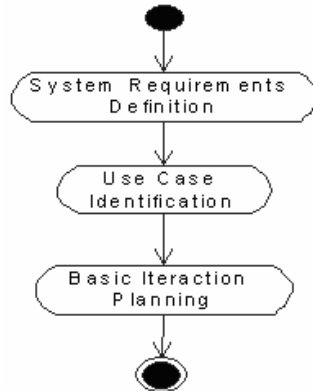
SQA Tools: Any UML CASE tool for Information and Process and Activities Modeling. Text editors for Objectives, Resources, and Organization Modeling.

#### 4. REQUIREMENTS

The information captured by the Enterprise Modeling workflow is detailed and consolidated as requirements for the information system, following the Requirements Workflow, shown on Figure 3. Its activities are:

1. System Requirements Definition: provides a basic requirements document. These requirements are a composition of features identified by the Process and Activities Modeling phases of the Enterprise Modeling workflow with some more detailed system's functionalities that can be identified at this point and are necessary to the consolidation of the business process information needs.

2. Use Case Identification: Use Cases are identified from the activities of an Activity Diagram that represents a specific business process. This activity defines the basic system's architecture, and helps driving requirements detailing.
3. Basic Iteration Planning: establish use case development priorities according to their criticality.



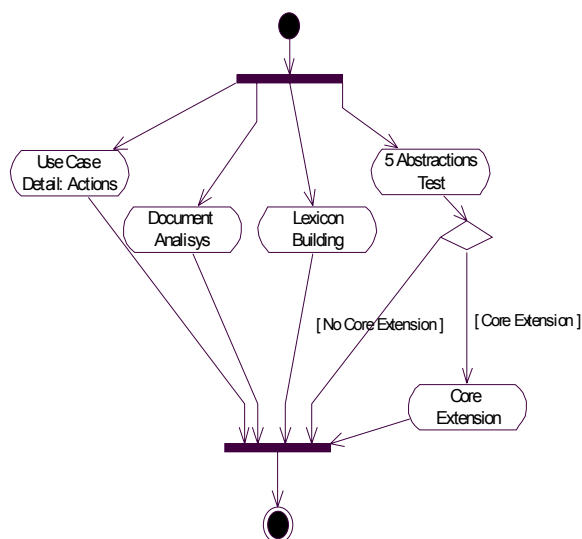
**Figure 3. Requirements Workflow**

**SQA Techniques:** Quality is guaranteed by Formal Technical Reviews, which check if all user requirements were captured and documented.

**SQA Tools:** ERP5 Feature is a tool that aims to help register, control and manage system requirements. This tool is integrated with ERP5 Use Case and ERP5 Project, creating a chain that associates a requirement to one or more use cases (for functional requirements), and then the use cases to project activities. With these tools it is possible to keep track of all requirements implementation and associated resources and costs, in every development phase. Customer inquiries on implementation status are easily answered and change management is facilitated for both the product and the process.

## 5. ANALYSIS

After the enterprise modeling stage it is necessary to define the activities that will transform structural and behavioral models into source code that reflects integrated business requirements. The workflow for this phase, presented in Figure 4, is executed for every Use Case:



**Figure 4. Analysis Workflow**

1. Use Case Detail – Actions: it is used two-column Use Cases, one for describing actors' actions and other for describing system's responsibilities or reactions [14]. For this incremental process, in the Analysis activities only the actions and some basic reactions are described, since detailed reactions are identified only during the Design activities. Security issues can also be addressed in this activity.
2. Documents Analysis: ERP5 is a document oriented ERP based on document workflows, since documents are considered a common language understood by all personnel in any organization. This activity consists of identifying the documents that support a given Use Case, starting by ERP5 default document templates that provide a basis for customization.
3. 5 Abstractions Test: the goal of this test is to find out if the UBM can support the Use Case, or in other words, if the resources flows described in a given business process can be represented by ERP5 core model. If not, Core Extensions are implemented.
4. Lexicon Building: maps concepts from the business world of the client to ERP5. This is necessary because, to support reuse, ERP5 names are quite general.
 

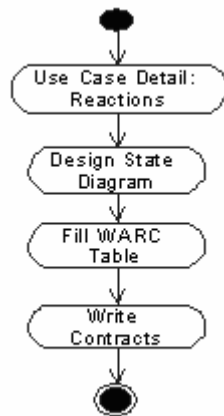
SQA Techniques: Quality is guaranteed by Formal Technical Reviews, which check requirement covering by use cases. Additionally, abstraction tests highly promote reuse, also facilitated by the presence of a lexicon.

SQA Tools: ERP5 Use Case module allows the definition of Use Cases, including their actors and scenarios. ERP5 Document Analysis module helps the identification and naming of documents and their items. ERP5 Lexicon module helps mapping

domain terms into ERP5 terms. Any XMI compatible [15] CASE Tool can be used to create UML models.

## 6. DESIGN

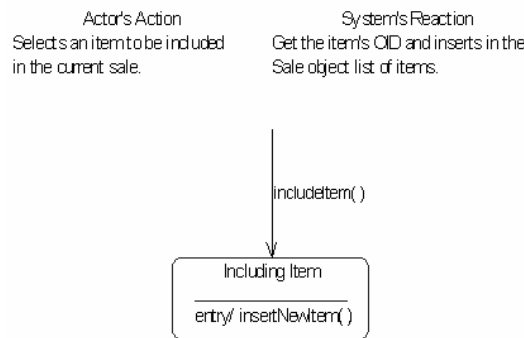
The Design workflow is based on an adapted version of the Workflow, Object Oriented Method (WOOM) [16]. This method focuses on tying structure (classes) to behavior, modeling the second as state machines. The activities of this phase are represented by Figure 5, and described as follows:



**Figure 5. Design Workflow**

1. Use Case Detail – Reactions: the reactions correspond to the second column of the UC, they define what the system is suppose to do according to an actor’s action.
2. Design Statechart Diagram: states names correspond to the state of the system in a particular moment. From the UC, verbs in the actions column identify state transitions; in the reaction column verbs identify states internal activities. Figure 6 shows an example of a single UC row with a correspondingly transition in a statechart diagram.
3. Fill WARC Table: a new modeling artifact, named WARC Table (Workflow – Action/Reaction – Responsible – Collaborators), is used to associate structure to behavior, guaranteeing encapsulation in object-oriented design. For the process here proposed, a different use of the WARC table is considered: a state transition is associated for each action, and a state internal action to each reaction – forming the Responsible column of the table. The objects that are manipulated by the transition or internal action are listed in the Collaborators column (the objects that participates on the UC were already identified in the Analysis phase). Table 2 shows the rows that represent in the WARC Table the UC step exemplified on Figure 6.
4. Write Contracts: This final step takes care of writing a contract [17] for each action and reaction. Contracts will determine what each transition/internal activity must

do to collaborate to the workflow correct realization. For describing operation's responsibilities, pseudo-code, proto-code, plain text, Object Constraint Language (OCL) and Activity Diagrams can be used.



**Figure 6. Example Transformation from a Use Case Row to a Statechart Diagram's Transition and State in WOOM**

SQA Techniques: use of a well-defined modeling method (WOOM), formal technical reviews for checking the quality of models developed under this method, and Model Checking for checking the consistency of workflows.

SQA Tools: Any XMI compatible CASE Tool can be used to create UML models. For supporting WOOM, currently is in development ERP5 *Deployer*, which will provide integration between use cases and WARC tables. Additionally, formal methods based on Model Checking [18] are under investigation to make *Deployer* check workflow consistency automatically, reducing the necessity for testing code. A plug-in for the Use Case Module will implement WARC tables with some basic features such as selection lists of available classes, transitions, and state activities and automatic updating of class diagrams. These features will avoid ordinary modeling mistakes and accelerate code transformation.

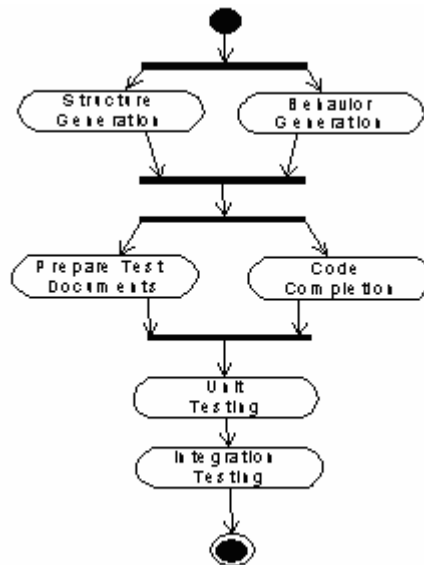
**Table 1. Example WARC Table Row for Figure 6**

Action/Reaction	Responsible	Collaborators
Select item	includeItem()	Product
Insert item on list	InsertNewItem()	Purchase, Item



## 7. IMPLEMENTATION

Implementation in ERP5 Process consists of generating code from UML diagrams, writing algorithms for completing this code, and testing. Implementation workflow is executed for every use case as shown in Figure 7, and described as follows:



**Figure 7. Implementation Workflow**

1. Structure and Behavior Generation: using a code generation tool – ERP5 Generator, the new type and related workflow are automatically generated from a XMI file.
2. Prepare Test Documents: these documents are instances of each scenario of each Use Case with specific values.
3. Code Completion: represents the implementation in source code of the algorithms of the workflow's transactions and internal activities.
4. Unit Testing: is run by a testing script, which automates the steps described in the Test Documents. This activity is supported by the ERP5 testing framework.
5. Integration Testing: the Use Case is tested in conjunction with others to check consistency among functionalities that must work integrated.

SQA Techniques: Black and White box testing is used at this stage, since it comprises of both coding and module integration activities. Software Configuration Management is accomplished through the use of a proper tool. Code generation avoids common programming mistakes, reducing testing activities to the code manually written during the code completion activity.

SQA Tools: ERP5 Generator is a tool that generates structural, behavioral, and GUI elements from specific artifacts. From Class Diagrams, Python classes, their

relational mapping and basic GUI for object maintenance (create, destroy, getters and setters) are generated. From Statechart Diagrams workflows are generated. ERP5 Generator parses XMI files exported by a compatible CASE tool, check it against a WARC Table, and creates the portal type and associated workflow. Complementing Generator, ERP5 Subversion integrates version control with testing and project management. Finally, ERP5 Test Case provides template testing scripts that automate most of Unit and Integration tests, and Zelenium, a Zope GUI test tool provides user interface testing.

## 8. CONCLUSIONS

This paper presented ERP5 architecture, its proposed development process and associated SQA techniques quite briefly, given space limitations. It is believed that ERP5 framework addresses all the eleven McCall's quality factors, being highly reusable, easy to maintain, strongly secure, and very usable. Also, ERP5 Process defines a clear flow of model transformations, with consistency checks supported by proper techniques and tools in each transformation. Aiming to enhance even more the use of tools during the development, ERP5 *Deployer*, a tool fully adherent to the proposed development process and integrated with all others cited on this work, is in development. This tool will automate the development workflows, provide template documents for managers – based on the Project Management Body of Knowledge (PMBOK), and improve consistency checks among the successive model transformations that occur during the process, automating it even more.

It is important to note that the goal of the presented process is to supply ERP5 adopters with the option of a model-driven development method based on proper practices and tools, but they are not obliged to follow it entirely.

## REFERENCES

1. M. Odeh and R. Kamm, Bridging the Gap Between Business Models and System Models, *Information and Software Technology*. Volume 45, pp.1053-1060, (2003).
2. R.D. Campos, R.A.D. Carvalho, and J.S. Rodrigues, Enterprise Modeling for Development Processes of Open Source ERP, in *Proc. 18<sup>th</sup> Production and Operation Management Society Conference* (Dallas, USA, 2007).
3. R.A.D. Carvalho, Issues on Evaluating Free/Open Source ERP Systems, in *Proc. of Research and Practical Issues of Enterprise Information Systems (IFIP Series)* (Springer-Verlag: New York, 2006), pp.667-676.
4. L. Zhao and S. Elbaum, Quality assurance under the open source development model, *The Journal of Systems and Software*. Volume 66, pp.65-75, (2003).
5. R.A.D. Carvalho and R.D. Campos, A Development Process Proposal for the ERP5 System, in *Proc. of 2006 IEEE International Conference on Systems, Man, and Cybernetics* (Taipei, Taiwan, 2006).
6. V. Botta-Genoulaz, P.A. Millet, and B. Grabot, A Survey on the recent research literature on ERP systems, *Computers in Industry*. Volume 56, pp.510-522, (2005).

7. J.P.S. Solanes and R.A.D. Carvalho, ERP5: A Next-Generation, Open-Source ERP Architecture, *IEEE IT Professional*. Volume 5, pp.38-44, (2003).
8. IFIP – IFAC GERAM: *Generalized Enterprise Reference Architecture and Methodology*, IFIP – IFAC Task Force on Architectures for Enterprise Integration (1999).
9. J.A. McCall, P.K. Richards, and G.F. Walters, *Factors in Software Quality*. Volumes. 1, 2, 3 - AD/A-049-015/055 (Springfield, 1977).
10. J. Arlow and I. Neustadt, *UML and the Unified Process – Practical Object-Oriented Analysis & Design* (Addison Wesley: London, 2002).
11. V. Botta-Genoulaz, P.-A. Millet and B. Grabot, A Survey on the recent research literature on ERP systems, *Computers in Industry*. Volume 56, pp.510-522, (2005).
12. F.B. Vernadat, Enterprise Modeling and Integration (EMI): Current Status and Research Perspectives, *Annual Reviews in Control*. Volume 26, pp.15-25, (2002).
13. K. Kosanke, F. Vernadat, and M. Zelm, CIMOSA: Enterprise Engineering and Integration, *Computers in Industry*. Volume 40, Number 2, pp.83-97, (1999).
14. H. E. Eriksson and M. Penker, *Business Modeling with UML* (John Wiley & Sons: New York, 2000).
15. R. Wirfs-Brock, *Designing Scenarios: Making the Case for a Use Case Framework*, Smalltalk Report (SIGS Publications: NY, Nov-Dec 1993).
16. Object Management Group, *MOF 2.0/XMI Mapping Specification*, v2.1 (2005).
17. R.A.D. Carvalho, Device and Method for Information Systems Modeling, *Brazilian Patent PI0501998-2* (June 09, 2005).
18. B. Meyer, Applying Design by Contracts, *IEEE Computer*. Volume 25, Number 10, (1992).
19. K.S. Merz, Model Checking and Code Generation for UML State Machines and Collaborations, in *Proc. 5<sup>th</sup> Workshop on Tools for System Design and Verification* (Augsburg, 2002).