Proceedings of the 5th WSEAS International Conference on Applied Computer Science, Hangzhou, China, April 16-18, 2006 (pp77-79)

SEMF Testing Methodology for an Inter-eBusiness Software development

Víctor Fernández P., Ricardo Chalmeta
Grupo Integración y Re-Ingeniería de Sistemas (IRIS)
Universitat Jaume I. Castellón, 12071. Spain

Abstract: - The change towards an e-business strategy is leading to a strong demand by companies for e-business solutions. However, in spite of companies' interest in this new management model, many e-business implementations fail. One of the main causes is that the existing methodologies to approach a e-business project are not adequate as they do not satisfactorily integrate and complement the strategic and technological aspects of e-business, and this gets more strength when we consider Large Complex eSystems with High Volume Complex Engagements. The purpose of this paper is to provide an experience report of a methodology, called Strategic Engineering Management Facility (SEMF), for the testing of inter-enterprises ebusiness software development in Large Complex eSystems.

Key-Words: - e-Business Strategy, Large Complex Systems, complex engagements, testing, inter-enterprises

1 Introduction

Inside Software Engineering there are different structured approaches to define the project's approach for software development [1]. One of this is the V-Model [2]. It is a well-known variation of the System Development Life Cycle which emphasizes system quality assurance. This model can be extended in order to guide during inter-enterprise e-business software designing, building and testing (Figure 1).

The requirements phases of the development life cycle along the left side of the V-Model are structured so that successive levels of detail may be added to the system's requirements.

The levels of test, depicted along the right-side of the V-Model, are structured to validate the system's requirements defined in the corresponding phase of design. So, this model provides for a structured, top-down decomposition of requirements based on ebusiness needs, and tests that are fundamentally aligned to validate those requirements [3].

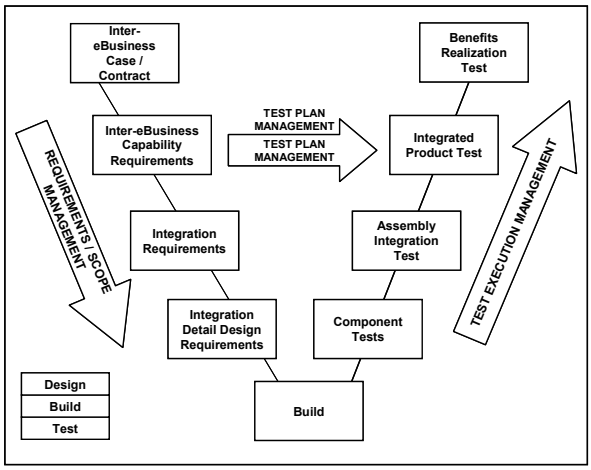

Fig.1 V-Model structure for inter-ebusiness software development

2 SEMF Testing Methodology

However, to improve the efficiency of inter-ebusiness software development and the predictability of the results with the V-Model, a project's methodology should align to the V-Model, where distinct phases of design are defined with corresponding levels of integration test. The methodology should also define the relevant processes and corresponding deliverables at each distinct phase of design and test.

To solve this problem, a testing methodology called Strategic Engineering Management Facility (SEFM) has been developed (table 1). This new testing methodology must be applied to each of the different testing phase levels of the V-model shown in figure 1: component test, assembly integration and integrated product tests.

Requirements Management	Test Plan Management	Test Execution Management	Defect Management
•Centralize requirements repository •Manage changes in scope •Trace a requirement through analysis and and design •Understand the priority and status of a requirement	•Describe how a requirement is tested •Provide analysis of test completeness •Provide testing metrics	•Provide framework for test execution •Provide consistent location for capturing actual results •Provide framework for regression testing	•Relate defects test execution - discovery point •Provide metrics associated with quality, productivity, etc.

Table 1. SEMF Testing Metholodgy phases and objectives

SEMF testing methodology specifies the framework to improve the ability of enterprises 'as a whole' to easily, correctly and safely render their existing (as well as their coming) applications and software communicate and cooperate. This includes well-defined and structured processes, deliverables and tools across the software development life cycle, and is a fundamental part of the Large Complex eSystems, defined by the bullet points highlighted below:

- many people and enterprises collaborating in a common ebusiness (inter-ebusiness)
- multi-year project horizon
- significant ebusiness impact
- high potential value/high potential risk

Following, as an example, we will show how the phases of test plan management and configuration management of the SEMF testing methodology support all test phases of the V-Model (right side of the V-model).

3 SEMF Test Plan Management Phase

Test plan management management specify the structured approach, processes and deliverables to verify the successful implementation of integration requirements and designs into the software components.

The test plan management is defined at each level of test, and describes the test conditions used to validate requirements, and the approach used to exercise the conditions (e.g., using test data and ebusiness process).

Figure 2 shows the integration test plan structure, and the corresponding relationships to other entities.

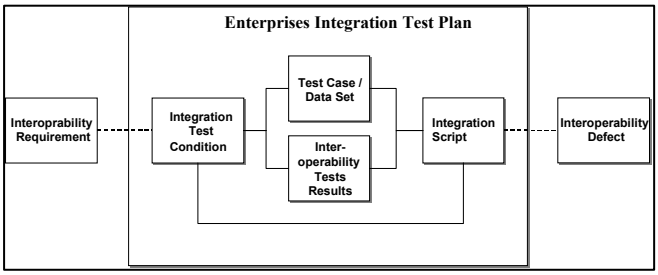

Fig.2 Test plan management structure

The key components of the test plan are highlighted below:

Integration Test conditions: identify which enterprise and process is being tested within the given level of test. One to many conditions are typically 'linked' to an interoperability requirement. Conditions are structured to 'prove' all facets of a requirement are tested, in some cases this might include negative or alternative conditions for a given requirement (e.g., confirm customer found / confirm customer not found). Test conditions may be organized around inter-business process or function area for ease of definition and management.

Test case/data: includes the data entities, like customers or orders, used to validate a condition. Typically, groups of mutually exclusive test conditions make up the test case and the detailed data defined for the test case satisfies the test condition/requirement. Where possible, engagement typically rely on production-type data for use in testing to reduce time spend in manufacturing data and to improve the realism of the test.

Interoperability Results: define the expected and actual results of the integration test execution. Generally, expected and actual results are associated with steps within the test script to associate the expected / actual outcome with the action specified in the script.

Integration Test scripts: define how the test is conducted within the context of the inter-ebusiness process and supporting applications. Integration test scripts typically contain appropriate level of detail required to provide instruction on using the application to exercise the integration test conditions.The integration test script should provide cross-reference to conditions exercised within the script, as well as references to appropriate test cases to be used in the context of the script. Once the integration test script is executed, this one provides the necessary proof that a interoperability condition is satisfied. Generally integration test scripts are organized into sub-cycles and cycles, which reflect the order and precedence of script execution according to inter-business process rules and dependencies. Organization of cycles and sub-cycles is then used derive the test schedule structure and dependencies.

The custom components of the SEMF framework provide the essential linkage between requirements and the associated test conditions within a given level of test. SEMF provides the overall blueprint to establish and map requirement types into an appropriate level of test. Requirements of a given type (e.g. integration capability design - requirements) are then mapped into an interoperability test condition 'repository' so conditions may be subsequently associated with that requirement.

4 SEMF Integration Configuration Management Phase

Integration configuration management address the need to migrate software components between development and test environments to a centralized real environment through which the differents enterprises could interact and manage its inter-ebusiness. Integration configuration management provides the logical framework to group like remote software module components, as well as the outline for the physical environment and tools to promote/demote software between development and test environments.

The SEMF testing methodology provides the ability to link one to many requirements to a software component/package. SEMF enabled links between requirements and software components provide the ability to understand scope of design changes to a given software component. Additionally, as interoperability defects are discovered during integation test execution, impacted modules can be linked to impacted requirements by analyzing the script-to-condition relationships.

5 Conclusions

The SEMF testing methodology showed in this paper has been used and validated in fifteen inter-enterprises e-business projects. As a result of this, the following conclusions of the use of SEFM have been obtaining:

Effective testing of the system's requirements relies on a structured test plan and execution approach, which describes what is being tested (e.g., conditions based on design requirements) and how the requirements are being tested (e.g., using interoperability process and data).

A structured interoperability test planning and execution, along with integration of requirement and defect entities, improve the integration project's ability to verify requirements.

SEMF ensure completeness of test and ability to describe 'how' requirement is satisfied through structured test plan model

SEMF reduce time and effort required for project to establish the integration test planning processes, deliverables and tools

SEMF support definition and reporting of standard test planning and execution metrics (e.g., productivity, quality, etc.)

SEMF reduce test execution times by integrating test automation tools with established test plan structure.

Acknowledgements:
This project has been founded by different enterprises, Accenture and CICYT DPI2003-02515.

References:

[1] Sánchez J.S., Chalmeta R., et al.," *Ingeniería de Proyectos Informáticos: Actividades y Procedimientos*", Colección "Universitas", Publicacions de la Universitat Jaume I, 2003.

[2] Plogert, K., "*The tailoring process in the German V-Model*", Journal of Systems Architecture, 1996, 42:8, pp.601-609.

[3] Knight L, "*System Development Methodologies for Web Enabled E-Business: A Customization Paradigm*", http://www.kellen.net/SysDev.htm, 2005