A Case Study of Metric-based and Scenario-Driven Black Box Testing for SAP Projects

Maya Daneva[1], Olga Ormandjieva[2], Manar Abu Talib[3]

Abstract: This paper makes a first attempt towards improving the testing process in ERP projects by using a metric-based approach [ABU06] based on functional size measurement. The paper reports on how this approach was adapted to a ERP-package-specific project context, how it was applied to five settings in a mid-sized project, and what was learnt of doing it.

1 Introduction

ERP testing is the process of executing configured ERP transactions in a controlled fashion with the objective to validate if the ERP solution behaves as defined in the business requirements document. It can be an expensive and labour-intense process for both systems testers and business users. Industry trend research firms estimate that, for example, when following the AcceleratedSAP methodology for rapid SAP implementation, it can take 20 testers four weeks to manually run a test cycle of an end-to-end business process (or a similar testing need) that consists of 100 transactions. With four testing cycles per year, organizations can expect to spend 12,800 hours manually testing at a cost of $499,200 (based on average salaries of $60,000). Acknowledging that there is a relationship between the quality of the test process and the final quality of the solution being delivered, the improvement of the ERP test process is essential for the project success at any ERP adopting organizations.

In this paper, we make a first step towards systematically improving the ERP testing process by using a metric based approach. We draw on earlier experience [ABU06] gained by two of the authors (Abu Talib and Ormanjieva) in metric-based and scenario-driven black-box testing as applied to projects using UML. In [ABR04, ABU06], Abu Talib et al investigated how test cases and test plans can be built, partitioned, and prioritized early in a software project for which the requirements are documented by means of UML scenario models. Because the typical requirements engineering (RE) process for ERP delivers business requirements in terms of scenario models [DAN04], intuitively we felt that the testing strategy proposed by Abu Talib et al [ABU06] looks like a good candidate to be considered for use in ERP testing. In what follows, we report on how we adapted this approach to ERP project settings and how we applied it to improve ERP testing. Aspects of the testing process were analyzed in the specific context of projects implementing SAP R/3, a leading product in the ERP market.

2 Background

2.1 SAP Testing: Process and Issues

Each ERP vendor provides their client organizations with a package-specific standard testing process that is part of the package implementation methodology the vendors suggest for the project. In SAP project settings, the testing process a client company is likely to adopt is part of the Accelerated SAP methodology for rapid SAP implementation [OVE00]. The SAP testing process is tightly linked with SAP RE process as test cases are derived directly from the elementary logical components of the SAP solution described in the business requirements specification (which is the key deliverable from the RE process). Typically, the SAP business requirement documents specify business scenarios being represented in terms of event-driven

[1] University of Twente, The Netherlands
[2] Concordia University, Quebec, Canada
[3] Concordia University, Quebec, Canada

process chains (EPC) that consist of processes, events, and logical connectors [DAN00]. Thus, within a company, a process describes what an organizational unit does and in the SAP R/3 System, a process means a physical transaction (which is a piece of code). For example, *"Enter Material"* is a user-recognizable piece of business functionality and it also corresponds to transaction *MM01* in the SAP Materials Management module. Similarly to RE, the SAP testing process exclusively relies on the concept of refining these business process scenarios into business processes, then mapping a process into process variants, and, finally, transforming each business function that makes up a process variant into a test case (Figure 1). In other words, the test case models a complete business process, with all its prerequisites for a particular test.

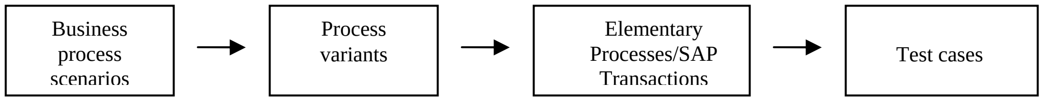
Business process scenarios
Process variants
Elementary Processes/SAP Transactions
Test cases

Figure 1: SAP Testing with eCATT

The SAP testing process requires a testing environment be set up separately from the configured solution and be ready-to-use prior to testing. This environment is a replication of the productive system (including all its stored data) and serves to simulate customized developments and evaluate them against real data. Experiences [OBE00] indicate that replicating the productive system is a time-consuming task requiring a lot of storage and creating ongoing hardware costs. Moreover, test manager will need to ensure that all the necessary data (master, transactional, test data) is properly loaded as part of the assessment for the test readiness review. To support clients in testing activities, SAP offers (as part of the SAP package) a suite of testing tools, the so-called SAP Test Workbench. It is used to create, analyze, monitor, track progress on, and systematically archive manual and automatic test cases. It also allows clients to integrate cases and test scripts from non-SAP providers in test plans. It consists of (i) Test Data Migration software, a data transfer tool enabling ERP adopters to extract representative application data from a productive system and standardize the set-up of non-productive systems, and (ii) the extended Computer-Aided Testing Tool (eCATT) for generating, recording, managing, and executing test cases. Despite of the supporting SAP standards and tools, the SAP testers get very little guidance on what to do in order to test fragments of the SAP solution in a cost-effective way or on how to prioritize test cases in the face of shrinking testing budgets. The quality of the decision-making processes related to these two questions heavily depends on the tester's judgments based on his/her personal past testing experiences with SAP [OBE00].

2.2 The Testing Strategy

For the purpose of our case study research, we have chosen to use the COSMIC-FFP-based black-box use-case-driven testing [ABU06]. Its basic idea is first to cont the COSMIC Full Function Points (FFP) of an application whose requirements are specified by UML use cases. As a by-product of the counting process, an exact inventory of the application usage scenarios and its elementary building blocks will be done. Taking inventory of what is in the use cases, we then get the use cases divided into sets and then those sets that include more use case information than other sets are used for testing purposes. This testing strategy includes the following steps: (i) generation of test cases through mapping the scenarios to a sequence of events in time (or data movements in COSMIC-FFP); (ii) partitioning test cases into equivalence classes by using a metric-based partitioning algorithm; (iii) prioritization of test cases based on functional complexity measurement in a COSMIC-FFP context; and (iv) selection of optimal subset of test cases by using a budget related algorithm. We decided on using this testing strategy, because (i) it rests on a model-driven RE process as the SAP project context does, (ii) its proof-of-concept has been demonstrated [ABU06], and (iii) it relies on an inventory of the elementary functional components of the system under development done by a standardized functional size measurement technique, namely COSMIC FFP.

2.3 Research Method

Our paper has the objective to answer two research questions: First, how to adapt the testing strategy to the SAP testing process, and second, where in the testing process the newly introduced strategy can create benefits to the testing team. Our research method consists on two parts: first, we map the logical components of the testing strategy from UML settings to the SAP settings, then we carry out a case study to try out our approach to improve ERP test-case partitioning, selection, and prioritization.

The unit of analysis in out case study is the testing strategy. Its application is observed in five SAP sub-project settings [DAN04] that implemented five business processes based on the packages SAP functionality. These are: External Service Procurement, Maintenance of Service Master Record, Procurement of Stock Materials, Maintenance of Material Master Record, and Procurement of Consumable Materials.

3 The Case Study

3.1 The SAP Testing Strategy

The adaptation of the approach in [ABU06] to the SAP settings included (i) the replacement of UML scenarios by means of SAP scenarios that are the building blocks of the SAP requirement document, (ii) the use of the IFPUG FPA method [DAN00] instead of the COSMIC FFP method (this was justified by the fact that FP counts already existed in the organization prior to carrying out this case study and that noone on board was familiar with the COSMIC FFP method), and (iii) mapping the steps of the approach by Abu Talib et al [ABU06] to the steps being carried out by using the SAP standard tools. The results of this exercise are sets of rules for practitioners that help him/her translate the testing strategy in the terms of SAP project artifacts. The adapted approach included these steps: First, the set of SAP scenarios that served as an input to FPA represents the set of test cases required to form the test set to be generated. Second, that set of test cases is partitioned into equivalence classes. Third, the test selection algorithm is run on the set of non-empty equivalence classes. The result is a set of selected test cases that ensures the best possible coverage [ABU06]. However, as Abu Talib et al already indicated, we can not claim the exhaustive fault coverage since this algorithm is maximizing the test coverage within the limits of a given budget.

3.2 Application and Findings

The adapted approach was applied to a case study setting with five subprojects in a medium-size project upgrading the SAP Material Management module in a telecommunication company. The business process requirements for this project were documented in the forms of EPC models by using the LiveModel tool which assisted in automatically mapping the company-specific process diagrams to the transactions of the SAP software package. By using this tool, a user can navigate from a business scenario process, to an elementary process, to a transaction, to a screen, and ultimately to a test case. The use of the LiveModel tool ensured that the inventory of the transactions to be tested was done quickly. For each scenario, the LiveModel tool identified the elementary processes components that served as inputs to our testing approach. For example, the External Services Procurement scenario involves seven elementary processes: create purchase requisition for service, determine possible service providers, create purchase order, monitor purchase order, enter services actually performed, accept services performed, verify invoices for services. Once these process components were identified, we applied the approach as described in [ABU06].

This process was repeated with four more SAP scenarios. Our early results gave initial indication that partitioning of test cases and giving priorities to each test case based on the algorithms by Abu Talib et al fits with the SAP testing process. Our study suggests that this testing strategy can be adapted to package implementations at no substantial costs incurred to the project. We also identified a few prerequisites that significantly impacted the straightforward application of our testing approach: First, we did not allow testing to span over several SAP system instances each of which implementing the same SAP-supported business process. Second, the business processes documentation was up-to-date and valid, so that testers were sure that they were testing the most recently required functionality. Third, the transactions were automatically mapped to the business process scenarios by using a tool. Thus, any subjective influence on our making inventory of SAP functionality was eliminated.

However to assess to what extent exactly our testing strategy improves testing decision-making in a face of budget constrains, further case studies need to be carried out.

3.3 Evaluation of Validity Concerns

Following Yin [YIN03], we considered two types of validity concerns regarding this case study. The threat to external validity means that the case study project may not be representative for all ERP projects. We decided for a setting that (i) had an up-to-date business requirement models in form of scenarios and (ii) was typical for the telecommunication companies in North America. It is our experience that the method works best when the company has modeled its ERP-supported processes and mapped them onto the set of configurable transactions in the package. We considered as the internal validity threat to our study the risk which while the testing strategy delivered meaningful results, this might be due to factors that we are unaware of and over which we do not have any control. To ensure internal validity, each process scenario was first taken and studied from the public SAP HELP library available in the net. Then, we re-examined the implementation of each process in the company-specific setting. We made sure that each business requirements document was created using the same RE process, standards and tools and that transactions were mapped to scenarios by using the LiveModel tool. When adapting the approach by Abu Talib et all to the SAP settings, the mappings between the UML concepts and the EPC modeling notation were borrowed from published literature sources [OBE00].

4 Conclusions

This paper explored the adaptation and the application of the metric-based scenario-driven black-box testing method by Abu Talib et al [ABU06] to a specific class of projects, namely ERP. The reported study revealed that this method has the potential to complement traditional ERP testing approaches, as the ones built-in and assumed in the ERP packages. Our study also revealed four issues: testing should not span over several SAP instances; the business processes documentation should be up-to-date and valid, so that testers are sure that they are testing the most recently required functionality; the transactions should be mapped to the scenario processes from the business requirements.

We recommend, however, a replicated follow-up case study be carried out do a deeper investigation on the validity threats with respect to our testing method. More case studies will also help promote the use of our method.

Acknowledgements
We would like to thank Tina James, Mac Kutty, Catherine Lee, and Marry-Anne van Alpen for share thoughts on aspects of this topic.

References

[ABR04] Abran, A., Ormandjieva, O. and Abu Talib, M., A Functional Size and Information Theory-Based Functional Complexity Measures: Exploratory study of related concepts using COSMIC-FFP measurement method as a case study, 14th Int.Workshop on Software Measurement (IWSM-MetriKon 2004), Shaker-Verlag, 2004, pp. 457-471.

[ABU06] Abu Talib, M., Ormandjieva, O., Abran, A., Khelifi A., Buglione, L. Scenario-based Black-Box Testing in COSMIC-FFP: a Case Study, Software Quality Professional, 2006, accepted for publication.

[ALA06] Alagar, V.S., Chen, M., Ormandjieva, O. and Zheng, M., Automated Generation of Test Suits from Formal Specifications of Real Time Reactive Systems. Submitted to the IEEE Trans. on Software Engineering Journal in 2006.

[DAN00] Daneva M, Practical Reuse Measurement in ERP Requirements Engineering, Proc. of the 12th Intl Conf. on Advanced Information Systems Engineering (CaiSE'00), Stockholm, Sweden, June 5-9, 2000, LNCS, Springer Verlag, pp. 309-324.

[DAN04] Daneva, M., ERP Requirements Engineering Practice: Lessons Learnt, IEEE Software, 21(2), pp. 26-33.

[OBE00] Oberneidermaier, G., M. Geiss, Testing SAP R/3 Systems, Addison-Wesley, Paperback, Published October 2000

[Yin03] Yin, R.: Case Study research: Design and Methods. Sage Publications, 2003.