International Journal of Information & Computation Technology.
ISSN 0974-2239 Volume 4, Number 10 (2014), pp. 1011-1016

http://www. irphouse.com

Testability Analysis of Framework Based Software at Requirement Analysis Phase

Noopur Goel
Department of Computer Applications
VBS Purvanchal University, Jaunpur
Uttar Pradesh, India

Abstract

Dependency on automated systems in every sphere of our life has raised a demand for enhanced quality and productivity along with the improved business performance of the software system. Researchers and practitioners in the field of software engineering are striving to achieve the same by applying many techniques. Reuse-oriented software development technique is one of the prevalent techniques, which promises to enhance the quality of the software. Testability is one of the external quality factors of the software, which affects quality of the software. This calls for the need of high testability of the system. Effectiveness of quality of software is increased if the testability analysis of a software system, developed by reuse-oriented approach, is performed in the very early stage- requirement analysis of the reuse-oriented development life-cycle. The effort used to develop software systems using the reuse-oriented approach must be less than the effort used to develop them from scratch. The present paper identifies the factors, which affect the testability of the reuse-based software and presents few testability metrics and a model, which quantifies testability of the application in the requirement analysis phase of the application engineering.
Keywords- Framework-based application, testability, test effort

Introduction

With the increased dependency on automated systems in all sphere of life of human beings, demand for enhancing the quality and productivity of software systems is also increasing to accomplish customers' satisfaction. Researchers and practitioners in the field of software engineering are aspiring by all means to achieve the goal (enhanced quality and productivity with reduced time-to-market).

Reuse technology, like other engineering disciplines, is inherent to software engineering also. Software reuse promises to produce software systems with enhanced quality, productivity, and reduced time to market, if applied systematically. Instead of developing applications every time from scratch, the applications developed using reusable artifacts are significantly in practice from the last three decades, e.g. software product lines. The quality of the software systems developed using the reusable artifacts depends heavily on the quality of the artifacts reused. As the reusable artifacts are time tested one, up to some extent they make sure the quality of the final product also. Object-oriented framework- one of the reusable entities, promises to produce products with enhanced quality, productivity, and reduced time-to-market.
A framework is the reusable (the context) of a system or a subsystem stated by means of a set of abstract classes and the ways the objects of (subclasses of) those classes collaborates (Beck and Johnson, 1994). A framework offers reuse of all the four- architecture, design, code, and test. Being a reusable pre-implemented architecture, a framework is designed "abstract" and "incomplete" and is designed with predefined points of variability, known as hotspots, to be customized later at the time of framework reuse (Jeon et al, 2002). A hotspot contains default and empty interfaces, known as hook methods, to be implemented during customization. While preserving the original design, parts of the framework are extended or customized to build applications using frameworks. A hook is a point in the framework that is meant to be adapted in some way such as by filling in parameters or by creating subclasses (Froehlich et al, 1997). Hook description (Froehlich et al, 1997) is used for many possible implementations of the Framework Interface Classes (FIC), shown in Figure 1, for developing applications in the application engineering stage (Dallal, 2003).

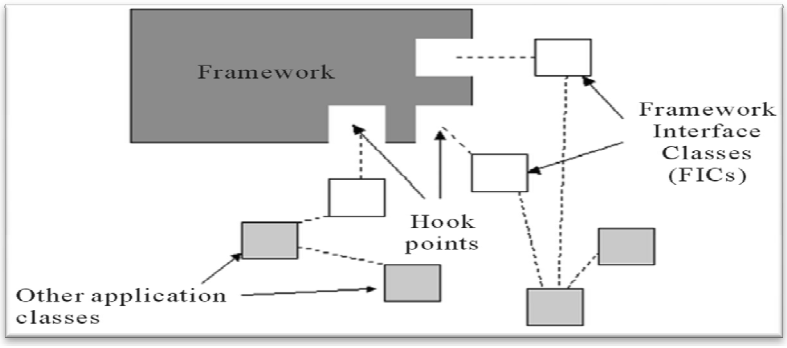

Figure 1. Framework Application Classes (hooks, framework interface classes, other application classes) (Dallal and Sorenson, 2008)

Software testing is performed with the intent of finding faults and is the crucial and effort taking activity of the software development process. It is more effective if performed in the early development life cycle. The resources for testing are limited, so to achieve effective testing software must be designed for testability. The testability of software is an external quality attribute and determines the effectiveness of testing, and hence the probable correctness of the software.

As discussed, quality of the application developed using framework heavily depends on the quality of framework itself. Before developing the application, the software engineers must ensure that the effort of developing, testing, and maintaining the application using reusable artifacts is lower than the effort of developing, testing and maintaining the application from scratch. The test artifacts developed, during the domain engineering stage, to test the framework is also reusable during the testing of completed application during the application engineering stage.

In IEEE glossary (IEEE 1990), testability is defined as: 1) the degree to which the system or component facilitates the establishment of test criteria and performance of tests to determine whether those criteria have been met; 2) The degree to which a requirement is stated in terms that permit establishment of test criteria and performance of tests to determine whether those criteria have been met. ISO defines testability as the attributes of software that bear on the effort needed to validate the software product (ISO/IEC 9126, 1991). (Bache and Mullerburg, 1990), define testability as effort needed for testing. Effort is the vital quality characteristic of the software and claims that the most important software characteristic contributing to testability is the number of test cases needed for satisfying a given test strategy.
Requirements are the unambiguous representation of test cases. During the framework instantiation phase, requirements of the framework are analyzed for the selection and adaptation of the framework. The framework specifications and requirement specifications specified at the hooks of the framework form the basis of our work presented in this paper. This paper attempts to identify the factors and quantify the effort needed for testing the reuse-based application during the application engineering stage.

The paper is organized in four sections. Although, testability of framework-based applications are very important before the development of reuse-based applications, very few work is performed earlier, as per our literature survey and is discussed in Section 2. In Section 3, our proposed work is detailed. Conclusion and future work is specified in Section 4.

2. Related Work

(Goel and Gupta, 2012a) identifies flexibility provided at the hooks of the framework as the factor influencing the testability of applications at all levels of testing and proposes the testability model. In their next work, (Goel and Gupta, 2012b) performs an empirical evaluation to prove the testability model. (Goel et al, 2013) also proposes a framework adaption methodology, HDTFD, which enhances the testability of the framework-based application.

3. Proposed Work

For effective testing, all applications must be designed for testability. Like single application system development, framework-based application must also be designed for testability. Flexibility defined at the hotspots of the framework, influences the testability of the framework-based applications [9]. Early estimation of testability of

framework-based application helps to reduce the overall development cost of the application. Requirements are the unambiguous representation of test cases. Requirements of the framework and requirements specified at the hooks of the framework form the basis of our approach. The commonalities (i.e., the common requirements identified during domain analysis of the framework engineering stage) of the applications within a particular domain are analyzed and encoded within the core of the framework. These requirements are reused during application engineering stage. The variations (i.e., variations in the requirements of various applications identified during domain analysis of the framework engineering stage) among the applications within that particular domain are abstracted and defined incomplete at the hotspots of the framework. These abstracted and incomplete requirements may be adapted and implemented during application engineering stage in three ways:

a. By using them as defined i.e. as-is.
b. By ignoring the requirements specified for the behaviors that are not needed in implementing the application.
c. By specifying new requirements for the added behaviors to meet application requirements.

3.1 Testability Metrics for Estimating the Testability of Framework-Based Applications

Few testability metrics for framework-based application are proposed, which may help in analyzing the testability of the framework-based application during the requirement analysis phase of the application engineering stage.

1. Number of requirements, which are common among all applications within the domain and implemented in the framework = Reqcom
2. Number of prospective requirements, which provides variations among the applications within the particular domain and specified at the hooks of the framework = Reqvar
3. Number of requirements, which are specified at the hooks of the framework and used as-is during application engineering stage = Reqvar_as-¬is
4. Number of requirements, which are specified at the hooks of the framework and customized as per user's requirements during application engineering stage = Reqvar_cust
5. Number of requirements which are newly specified and implemented at the hooks of the framework during application engineering stage = Reqvar_new
6. These metrics can further form the basis for the estimation of test effort of the framework-based application.

3.2 Testability Model Considering the Common/Variable Requirements of the Applications within the Domain

Object-oriented frameworks represent a group of applications that share some common requirements within the defined domain. The common requirements are defined and implemented concretely in the framework and the prospective variable

requirements are abstracted and left incomplete and are known as hotspots of the framework. Although, it is not very easy to find to find the exact variable requirements within the defined domain, a rough measurement may be performed to make estimation of the testability of framework-based application.

Estimation of Test effort, TE, for testing the framework-based application at the requirement analysis phase during application engineering stage is:

$$\text{Test effort, } \mathbf{TE} \; \alpha \; \frac{\mathbf{Req_{var}}}{(\mathbf{Req_{com}} + \mathbf{Req_{var}})} \qquad (1)$$

where,

$$\text{Reqvar} = (\text{Reqvar\_as-¬is} + \text{Reqvar\_cust} + \text{Reqvar\_new}) \qquad (2)$$

Testability, Tb, of application software is inversely proportional to test effort to perform testing of the application i.e.

$$\text{Tb} \; \alpha \; 1 / \text{TE} \qquad (3)$$

Hence, testability of framework-based application is

$$\boxed{\mathbf{Tb} \; \alpha \; \frac{(\mathbf{Req_{com}} + \mathbf{Req_{var}})}{(\mathbf{Req_{var\_as\text{-}is}} + \mathbf{Req_{var\_cust}} + \mathbf{Req_{var\_new}})}} \qquad (4)$$

This fraction may be calculated before the design of the framework-based application to have an idea that whether the framework-based application is testable or not. This model provides an idea of testability of framework-based application before the design of the framework-based application.

4. Conclusion and Future Work

Framework-based applications (e.g. software product lines) are currently in practice, promising to enhance quality, productivity and reduce time-to-market. Software systems must be designed for testability for effective testing and reducing effort, which in turn helps in achieving the goal- enhanced quality, productivity and reduced time-to-market.

This paper identifies “Requirements Specifications” specified during the framework engineering, as the influencing factor for testability of the framework-based application and proposes few testability metrics and a model. In future, few more factors would be identified, which may influence the testability of framework-based application. It is also supposed to perform a case study and empirical analysis of the work presented in this paper.

References

[1] K Beck and R Johnson (1994), “Patterns Generate Architectures,” Proceedings of 8th European Conference on Object Oriented Programming, Bologna, pp. 139-149.

[2] T Jeon, S Lee and H Seung (2002), “Increasing the Testability of Object-Oriented Frameworks with Built-In Test,” Lecture Notes in Computer Science, Vol. 2402, pp. 873-881.

[3] G Froehlich, H J Hoover, L Liu and P Sorenson (1997), "Hooking into Object-Oriented Application Frame- works," Proceedings of the 19th International Conference on Software Engineering, Boston, May pp. 491- 501.

[4] J Al Dallal (2003), "Class-Based Testing of Object-Oriented Framework Interface Classes," Ph.D. Thesis, Department of Computing Science, University of Alberta.

[5] J Al Dallal and P Sorenson (2008), "Estimating the Coverage of the Framework Application Reusable Cluster-Based Test Cases," Information and Software Technology, Vol. 50, No. 6, pp. 595-604. doi:10.1016/j.infsof. 2007.07.006

[6] IEEE (1990), "IEEE Standard Glossary of Software Engineering Terminology", IEEE CSP, New York.

[7] ISO/IEC 9126 (1991), "Software Engineering Product Quality".

[8] R Bache and M Mullerburg (1990), "Measure of Testability as a Basis for Quality Assurance," Software Engineering Journal, Vol. 5, No. 2, pp. 86-92. doi:10.1049/sej.1990.0011

[9] N Goel and M Gupta (2012a), "Testability Estimation of Framework Based Applications", Journal of Software Engineering and Applications, Vol. 5, No.11, pp. 841-849. doi:10.4236/jsea.2012.511097

[10] N Goel and M Gupta (2012b), "Empirical Evaluation of the Effect of Flexibility upon Testability of Framework Based Application", ACM SIGSOFT Software Engineering Notes, November 2012, Vol. 37, Number 6, pp.1-6. doi>10.1145/2382756.2382763

[11] N Goel, A K Tripathi and M Gupta (2013), "Hook_Test: An Aid to the Hook-Driven Test-First Development of Framework Based Applications", International Journal of Computer Applications, Volume 65, Number 16, 10.5120/11005-6299, ISSN online: 0975-8887.