Journal of Software Engineering and Applications, 2014, 7, 679-685
Published Online July 2014 in SciRes. http://www.scirp.org/journal/jsea
http://dx.doi.org/10.4236/jsea.2014.78062

Formal Methods for Commercial Applications Issues vs. Solutions

Saiqa Bibi, Saira Mazhar, Nasir Mehmood Minhas, Irfan Ahmed

UIIT-PMAS Arid Agriculture University, Rawalpindi, Pakistan
Email: saiqa_hani@yahoo.com, saira.m.hussain@gmail.com, nasirminhas@uaar.edu.pk, slow59@yahoo.com

Received 28 April 2014; revised 25 May 2014; accepted 20 June 2014

Abstract

It was advocated that in 21st century, most of software will be developed with benefits of formal methods. The benefits include faults found in earlier stage of software development, automating, checking the certain properties and minimizing rework. In spite of their recognition in academic world and these claimed advantages, formal methods are still not widely used by commercial software industry. The purpose of this research is to promote formal methods for commercial software industry. In this paper we have identified issues in use of formal methods for commercial applications and devised strategies to overcome these difficulties which will provide motivations to use formal methods for commercial applications.

Keywords

Formal Methods, Commercial Applications, Issues of Formal Methods

1. Introduction

Formal languages are the languages in which syntax and semantics are properly defined by using mathematical notations. Formal languages are of mathematical nature so; they raise the assurance on the system by reducing uncertainty in the specification of system [1]. Commercial application softwares are designed for vending to serve a commercial need or on the demand of customer. These applications are larger in size.

Formal methods are actually precise techniques; tools support is provided for development of software as well as hardware systems. Mathematical techniques of formal methods enable developer to examine and prove models at any stage of the software development life-cycle: gathering requirements, specification, architecture, design, implementation, testing, maintenance, and development [2].

The purpose behind the promotion of formal approaches was the detonation in software entanglement that began in the 1960s. Around then, software systems were rapidly getting to be more complex, however advance in devices and systems for improvement completed does not keep pace. Accordingly, there was a clear need for new techniques that might permit engineers to understand this complication. Formal methods made this practical by giving a mathematical framework for investigating projects [3].

Formal methods are used in software requirement specification: preparing an accurate report of what the software needs to do, and avoiding conditions on how it is to be attained [2].

Use of formal methods at the stage of formal specification can create very useful documentation [4]. A specification is a practical agreement between vendor and customer to offer them equally with a general acceptance of the software requirement. Absolute system specifications are essential because a design and implementation of system originate their quality in detail from the requirements specification. Modern research is representing the obvious advantages of formal and mathematical techniques for software requirements detain and design. Methods used for such mathematical technique to software requirements capture and design approach are jointly called formal methods for specification of software [5].

At the implementation stage, the use of formal methods is utilized for checking code of software. Each system particular match totally states an accuracy hypothesis that, if most conditions are fulfilled, the project will attain the impact depicted by documentation. Confirmation of code is the endeavour to demonstrate this theorem, or in any event to figure out the reason of theorem ignorance to hold. The inductive statement strategy for project confirmation was imagined by Floyd and Hoare and involves explaining the system with scientific declarations, which are associations that are between the variables of system and the beginning principles; each one time control, achieves a specific focus of the system. Coding can additionally be produced immediately from models provided by formal methods [3].

For many years, it was advocated that applying formal methods in software development would help industry congregate its goals of produce an enhanced software process and increase quality of software. The benefits that have been cited include finding defects in earlier stage of software development, automating checking of certain properties, and minimizing rework; despite these claimed benefits and usability in each phase of software development *i.e.* requirements specification, software architecture, software design, implementation, maintenance, testing, and evolution. In spite of its claimed benefits formal methods are still not widely used by commercial software companies.

In this paper we have described the challenges in the use of formal methods for commercial applications industry, devised strategies to overcome these difficulties which will provide motivations to use formal methods for commercial applications. We have divided our work into two sections. In the first section we have identified barriers of formal methods for commercial applications and devised strategies to overcome these barriers and in the second section we draw motivations for the use of formal methods in commercial applications.

2. Formal Methods: Issues vs. Solutions

Formal methods are still not seen widely in use for commercial applications due to a number of issues:

2.1. Issue-1: Lack of Skilled Persons with Mathematical Background

Formal methods for commercial applications development are not often commonly used or it does not well understood by many of the software engineers because implementation of formal methods demands the unambiguous concepts of discrete mathematics [6]. Formal verification needs mathematical skills not only due to the complex interactions between program subcomponents, but also for the deficiencies in current verification interfaces. These skill barriers economically resist the verification process by avoiding the selection of less skilled persons [7].

2.2. Solution: Tutorials & Trainings Help to Build Mathematical Knowledge

To build knowledge for formal methods in software development organizations on their own, a high quality tutorials and self-learning materials can help. Self-training materials allow independent developers to become familiar more easily with formal methods on their behalf [3].

2.3. Issue-2: Expensive

Business managers have faith that formal methods can enhance the software quality, but formal methods are not widely used because these methods are considered costly and unfeasible [8].

The most considerable doubt in formal methods, mostly from a perspective of management, is that these methods are expensive because implementing successful formal methods in an organization also need to purchase the tools for supporting these methods, training of engineers and designers, and effort and time to incorporate formal methods in the existing software development process, with other expenditure [5].

2.4. Solution: Ramp-Up Cost for Formal Methods Pay off over Many Projects

The ramp-up cost plays significant role for the implementation of formal methods. The vast number of software development tools and methodologies place their focus on long term cost savings. There is considerable support in the literature that formal methods provide real benefits in this area, increasing system reliability, and thereby decreasing long-term support and maintenance costs, while simultaneously maintaining or even decreasing initial development costs. So while formal methods may not be appropriate or cost effective for one-time use on a particular project, the evidence suggests that the initial investment can pay off over many projects [3].

2.5. Issue-3: The Inadequate Tool Support

In United States a fact for formal methods indicates lack of tool support as a barrier. They also highlighted it the key reason for the lack of appreciation of real world considerations to be the part of formal method community and therefore they are still not used in commercial industry [9] [10].

2.6. Solution: Formal Methods Supported by Variety of Tools

There are many tools available that provide support to formal methods such as Finite State Machines, VDM, Z, and OBJ. These tools are used to increase the productivity and accuracy in all the phases of a system. These tools possess different type of characteristics and used in industry according to the nature or requirement of the systems [11].

In early 80s, tools for Computer-Aided Software Engineering and Computer-Aided Structured Programming were seen as the mean of increasing programmer's productivity and for reducing programming "bugs". Now the Tool support can see as a source of increasing productivity and accuracy for formal developments [12].

It is our expectation that in near future more focus will be paid to Integrated Formal Development Support (IFDS) Environments that will be helpful in support of many of the phases of formal development. These toolkits will provide Integrated Programming Support Environments that will be support in configuration management and version control and facilitate all of the process activities and large scale developments more harmoniously.

2.7. Issue-4: Increase in Development Cycle

Although many established advantages of formal methods, these are badly accepted by industrial professionals. Many causes have been submitted for this situation one of them is that they increase the software development cycle [13].

2.8. Solution: Early Error Detection Helps in Reduce Development Time

It reduces the development time by applying testing techniques in earlier phases of the lifecycle [14]. In the seventies, was reporting that over half the software development time was devoted to testing activities. Formal methods offer new possibilities for verification *i.e.* model checking. It enables us in more effective identification of software defects which allows reducing verification time [7]. The use of a formal methods or model removes ambiguities in specification and thus reduces the chance of errors being introduced during software development. Thus this reduces the testing effort and time [15].

3. Formal Methods: Motivations for Commercial Applications

3.1. Formal Method Maximize Automation with Automated Tools

Automated tools allow producing models for verification promptly and in a convenient way directly from the design of models [16]. Modern progress in analysis tools of formal methods have made it realistic to verify significant properties formally to provide guarantee that design faults are identified and approved correctly in early stage of software development lifecycle [17]. A survey was presented in 2010 for effect of formal methods in software industry. The satisfaction level with automated tools is greater than 80% shown in **Figure 1** [8].

3.2. Automatic Verification Improvement

By using the formal verification approach better verification quality can be achieved with 70 percent less time and effort [18] as compare to other approaches. Only 30% effort is required by using formal techniques. Formal verification results in 2007-2010 survey are shown in the **Table 1**.

3.3. Formal Methods Reduce Cost

From formal specification, we can thoroughly gain effective test cases directly from the requirement. Test cases generation is a cost effective way [19]. Effects of Formal Methods on cost are presented in a survey in 2010 shown in **Figure 2**.

3.4. Formal Methods Reduces Defects at Early Stage

Formal specification produces accurate requirements and designs so that it reduces the chances of unintentional fault injections. Correctness of software system is also proved by formal verifications. Axiomatic correctness is one of verification methods [20]. Formal description forces the writer to ask all sorts of questions that would be delayed until phase of coding. This helps to decrease the errors that may occur in coding phase [16]. The results presented in 2011 indicate that applying ASD as a formal technique for developing controls software could re-

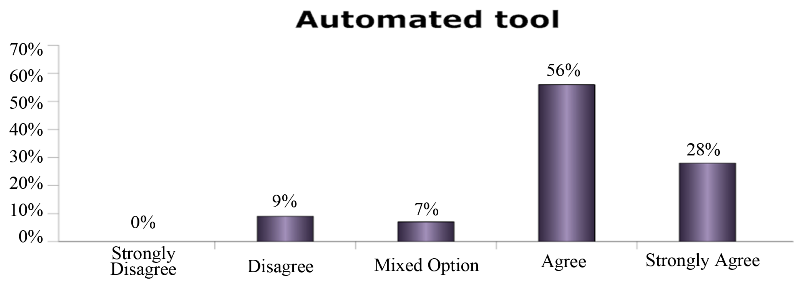

Figure 1. Satisfaction level with automated tools [8].

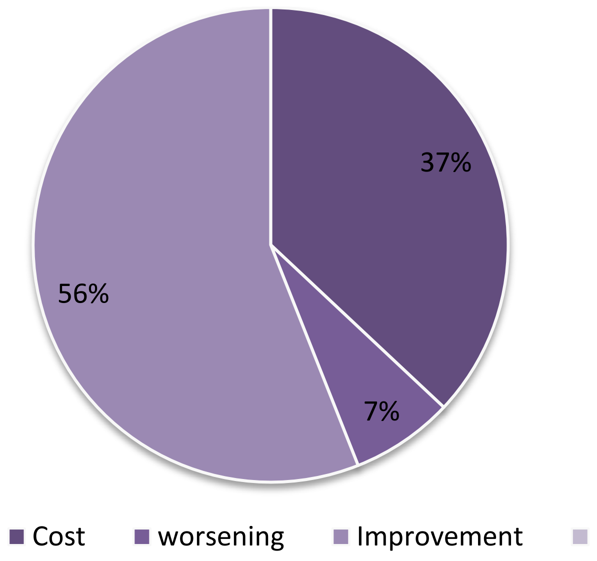

Figure 2. Formal approaches' effect on cost [8].

sults in fewer defects [21]. 63% defects are reduced by using formal techniques. The **Table 2** given below shows that the defects are reduced where Formal techniques are applied:

Figure 3 also present the effectiveness of formal methods as compare to traditional approach. Formal speci-

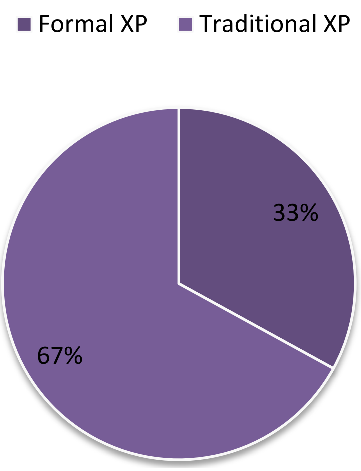

Figure 3. Less error rate during development process using formal approach [22].

Table 1. Formal verification vs. simulation [18].

Subtasks	Simulation	Formal verification
Preparation	Simulation script is generated by register tool	Properties are generated by register tool
Execution	3 days of simulation time	1.5 days for automatic set-up of 31 register block set-up and exhaustive verification of 12,600 properties
Analysis effort	60,000 entries to be analysed	No additional effort
Quality of analysis	Not-exhaustive, semi-automatic, error prone	Exhaustive, automatic, fail-safe
Total effort	3 days compute time + 2 days manual effort	1.5 days compute time (70% less than simulation)

Table 2. Defects are reduced with formal approaches [21].

ASD used	Unit	Lines of code				Defects			
		Manual LOC	**ASD LOC**	**Total LOC**	**ASD%**	**Manual defects**	**ASD defects**	**Total defects**	**Defects/ KLOC**
No	Acquisition	6140	0	6140	00.00%	33	0	33	5375
No	BEC	7007	0	7007	00.00%	44	0	44	6279
No	EPX	7138	0	7138	00.00%	7	0	7	0.981
No	FEAdapter	13190	0	13190	00.00%	18	0	18	1.365
YES	**FEClient**	**15462**	**12153**	**27615**	**44.01%**	**9**	**2**	**11**	**0.398**
YES	**Orchestration**	**3970**	**8892**	**12862**	**69.13%**	**3**	**4**	**7**	**0.544**
No	QA	23303	0	23303	00.00%	90	0	90	3.862
No	Status Area	8969	0	8969	00.00%	52	0	52	5.798
No	TSM	6681	0	6681	00.00%	7	0	7	1.048
No	UIGuidance	20458	0	20458	00.00%	23	0	23	1.124
No	Viewing	19684	0	19684	00.00%	294	0	294	14.936
YES	**XRayIP**	**14270**	**2188**	**16458**	**13.29%**	**27**	**0**	**27**	**1.641**

fication, formal verification techniques can lead to a higher quality product with reduced error rate and improved time efficiency [22].

3.5. Formal Methods Improves Quality

A survey presented in 2010 for effect of formal methods in software industry. It is presented that use of formal techniques improves quality of software in industry, 92% cases reported that quality is increased against the other approaches, and there is no single case that reported a decline in software quality. **Figure 4** shows the effect of formal methods on quality of software [8].

Overall effect of formal methods in software industry is shown in **Figure 5**. By applying formal methods in commercial software industry batter results can be achieved as compare to other approached as shown by survey results. Researchers are hopeful about the flourishing use of formal approaches for commercial software industry in future.

4. Conclusion

For many years, it was advocated that applying formal methods in software development would help industry congregate its goals of producing an enhanced software process and increasing quality of software. De-

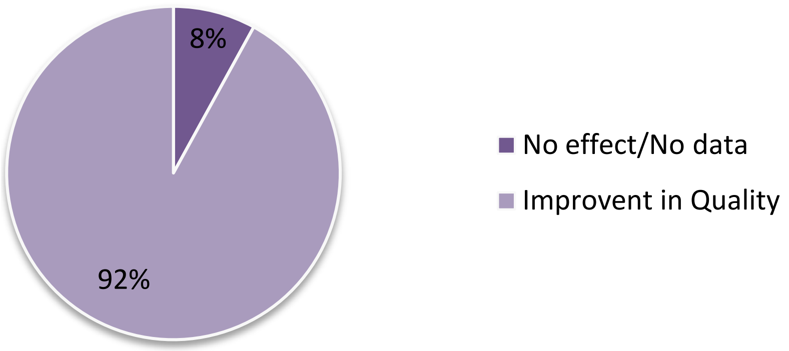

Figure 4. Formal approaches' effect on quality [8].

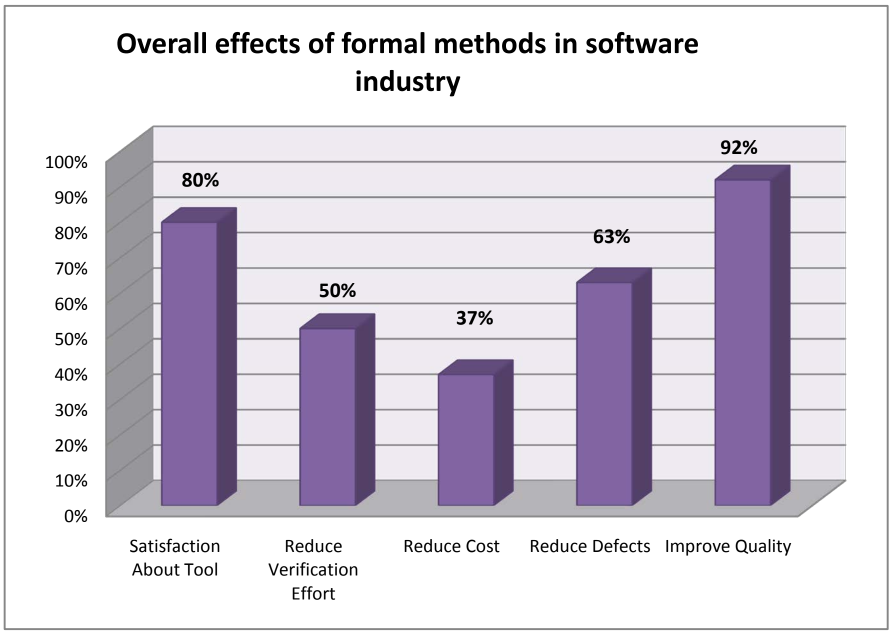

Figure 5. Overall effects of formal methods in software industry.

spite claimed benefits and usability in each phase of software development, formal methods are still not widely used by commercial software companies. Formal methods have not been widely used in industry due to a number of barriers. We have identified barriers of formal methods for commercial applications and then provide their solution. Formal methods offer several advantages *i.e.* maximize automation with automated tools, automatic verification improvement cost saving, defect reduction and quality improvement. These benefits are the stimulus to use formal methods in commercial software industry. By applying formal methods in commercial software industry batter results can be achieved as compared to other approaches shown by survey results. The purpose of this research is to promote formal methods for commercial application software in industry.

References

[1] Sammi, R., Rubab, I. and Qureshi, M.A. (2010) Formal Specification Languages for Real-Time Systems. 2010 *International Symposium in Information Technology* (*ITSim*), Kuala Lumpur, 15-17 June 2010, 1642-1647. http://dx.doi.org/10.1109/ITSIM.2010.5561643

[2] Woodcock, J.I.M. and Bicarregui, J. (2009) Formal Methods : Practice and Experience Engineering College of Aarhus. *ACM Computing Surveys*, **41**, 1-40.

[3] Geer, P.A. (2011) Formal Methods in Practice: Analysis and Application of Formal Modeling to Information System.

[4] Bowen, J.P. and Hinchey, M.G. (2006) Ten Commandments of Formal Methods... Ten Years Later. *Computer*, **39**, 40-48.

[5] Sommerville, L. (2009) Chapter 27 Formal Specification.

[6] Stidolph, D.C. and Whitehead, J. (2003) Managerial Issues for the Consideration and Use of Formal Methods. *Lecture Notes in Computer Science*, **2805**, 170-186.

[7] Schiller, T.W. and Ernst, M.D. (2012) Reducing the Barriers to Writing Verified Specifications. *ACM SIGPLAN Notices*, **47**, 95-112.

[8] Fulara, J. and Jakubczyk, K. (2010) Practically Applicable Formal Methods. *Lecture Notes in Computer Science*, **5901**, 407-418.

[9] Stidolph, D.C. (2003) When Should Formal Methods Be Used?

[10] Jhala, R. and Majumdar, R. (2009) Software Model Checking. *ACM Computing Surveys*, **41**, 1-54. http://dx.doi.org/10.1145/1592434.1592438

[11] Kefalas, P., Eleftherakis, G. and Sotiriadou, A. (2003) Developing Tools for Formal Methods.

[12] Bowen, J.P. and Hinchey, M.G. (1994) Seven More Myths of Formal Methods : Dispelling Industrial Prejudices. *Lecture Notes in Computer Science*, **873**, 105-117.

[13] Knight, J.C., Dejong, C.L., Gibble, M.S. and Nakano, L.G. (1998) Why Are Formal Methods Not Used More Widely?

[14] Hierons, R.M., Bogdanov, K., Bowen, J.P., Cleaveland, R., Derrick, J., Dick, J., *et al.* (2002) Using Formal Specifications to Support Testing. *ACM Computing Surveys*, **41**, Article No. 9.

[15] Singh, M. (2013) Formal Methods: A Complementary Support for Testing. *International Journal of Advanced Research in Computer Science and Software Engineering*, **3**, 320-322.

[16] Cofer, D., Whalen, M. and Miller, S. (2008) Model-Based Development. 1-8.

[17] Whalen, M., Cofer, D., Miller, S., Krogh, B.H. and Storm, W. (2008) Integration of Formal Analysis into a Model-Based Software Development Process. *Lecture Notes in Computer Science*, **4916**, 68-84.

[18] Knäblein, B.J. and Sahm, H. (2010) Contributed Article Automated Formal Method Verifies Highly-Configurable HW /SW Interface. 1-7.

[19] Batra, M., Malik, A. and Dave, M. (2013) Formal Methods : Benefits, Challenges and Future Direction. *Journal of Global Research in Computer Science*, **4**, 2-6.

[20] Alves, M.C.B., Dantas, C.C. and Silva, R.B. (2007) A Topological Formal Treatment for Scenario-Based Software Specification of Concurrent Real-Time Systems. 1-7.

[21] Groote, J.F., Osaiweran, A.A.H. and Wesselius, J.H. (2011) Benefits of Applying Formal Methods to Industrial Control Software. 1-10.

[22] Shafiq, S. and Minhas, N.M. (2014) Integrating Formal Methods in XP—A Conceptual Solution. *Journal of Software Engineering and Applications*, **7**, 299-310.