

Summary of “Balancing Practices: Inspections, Testing, and Others”

Session at ISERN 2005, Noosa Heads, Australia

Session leads: Marcus Ciolkowski, Forrest Shull, Tony Cowling, Masa Katahira, Lesley Land, Jose’ Carlos Maldonado, Bernard Wong

This session had as its goal an exploration of the strengths and weaknesses of different quality assurance (QA) techniques for safety critical software development. The objective was to leverage the expertise of the ISERN group in this area, to help develop a list of important criteria that would be used when trying to make decisions in industry about choosing a set of QA techniques or some combination thereof. This information could be useful for:

- 1) Breaking down “stove pipes” in research – Helping researchers to think about along which criteria different QA techniques provide the same benefits, and in which areas they would be considered more complementary.
- 2) Helping researchers who are investigating these areas to focus on convincing criteria for industry when attempting to demonstrate the effectiveness of one technique over another.

To do this, a short scenario describing end product goals and development constraints at the Japanese Space Agency was presented to provide some inspiration to participants. At the end of the session, the suggestions from the group were compared with empirical data that had been collected in that environment. To focus the discussion participants were asked to think about QA techniques consisting of various types of inspection, formal methods, and testing.

As was pointed out by a participant, these results are not intended to replace a rigorous literature review concerning the measured effects of the various techniques. [Especially in the safety critical domain many of the necessary results are probably not even published anyway, thus unavailable for integrating into a structured review.] These results should be seen only as tentative hypotheses based on the experience of experts in the area who also happen to be familiar with the literature in the area.

Suggestions from participants included the following:

- Focus on early defect detection
 - can afford expensive QA early in process (assumption: manned spacecraft)
 - Use of static models
 - Formal models to address timing issues
 - Domain-specific requirements language
 - domain-specific inspection technique
- Need to know for each technique how the costs are distributed across phases
- Risk is a central determinant
 - However the path from identifying risks to choosing QA techniques is not straightforward. It may not be worth formulating heuristics such as “When Risk X is identified, make sure Technique Y is in place.”
- To avoid overwhelming costs, need to segment according to the most risky/most critical functionality.
- Formal methods have hidden costs
 - Need to have a detailed design document available

- However for safety critical functions this could be unavailable or even classified. The most used formal methods are FSM – Finite State Machine and EFSM – Extended FSM.
 - Satisfactory documents might not even have been produced in the first place – have to pay to reverse engineer
 - Implementing formal methods can require re-engineering the entire development process to be compatible
- Formal methods: If actual environment doesn't match assumptions, QA effort could have been wasted. Hypothesis: Formal methods are best applicable when you have an accurate perception of the likely conditions of use. [In the safety critical application FSM and EFSM at least are usually applied. These are a type of formal method. There is also lots of solid work on testing these formal notations.]
- Applying formal methods [and model checking] doesn't mean that inspections and/or testing not needed.
 - Formal methods can help validating the system based on specs and model checking but inspections and testing are still needed to ensure that the specs are correct, since there are many drawbacks associated to model checking, e.g. the state explosion problem.
 - Even if formal methods are applied, couldn't do without testing. Testing and Inspections are still needed to get immediate feedback during development.
- Evidence from Parastoo & Conradi paper: Inspections capture more serious defects IF you have access to design and modeling guidelines about how software is developed
- Defect taxonomies such as ODC can help identify which QA techniques are helpful for which subclasses

Some highlights of the empirical results collected at JAXA included:

- The cost associated with model checking is a significant barrier to its use. There are projects which simply have time and budget constraints that are too tight, in which case model checking is replaced largely by checklist-based review. (On these projects, checklist-based review was responsible for finding at least 40% of the significant defects.)
- Checklist-based review is considered especially useful for verifying data correctness and consistency of the data handling systems.
- When there is sufficient time to employ model checking, it does find a large percentage of the significant issues (~40%).
- Review (without checklist) is quite useful for finding requirements defects.
- However, results do show that the review (without checklist) will miss defects that can be found by checklist-based reviews and by model checking.

A summary of the strengths and weakness of checklist-based review and model checking, which may serve as testable hypotheses to be confirmed in other contexts, is presented in Table 1.

Method	Advantage	Disadvantage
Formal Model/ Model Checking	<ul style="list-style-type: none"> ● It is useful to find problems concerning complicated state/mode transitions and processing timing issues which is hard to find manually. ● More effective than normal review for discovering erroneous description in the spec.. 	<ul style="list-style-type: none"> ● A certain amount of time are necessary for modeling and model checking. ● Need to have modeling and model checking knowledge. ● Low cost effectiveness for software which does not have complicated logic such as data handling, or transformation
Review with Checklist (Inspection)	<ul style="list-style-type: none"> ● High cost effectiveness even if there is very short time to access it. ● Erroneous descriptions are covered by check items in the list. ● It does not depend on the skill of evaluators in contrast to modeling methods. 	<ul style="list-style-type: none"> ● Checks limited to the items in the checklist. ● It is hard to check the detailed behavior in complex systems and to cover all possible combinations.

Table 1: Lessons learned from JAXA case study

Ideas for future work:

It was suggested that since the ISERN community has been involved in so many independent studies of QA techniques, it would be worthwhile to see if the community could agree on a set of standard metrics that would allow meaningful comparisons of the different techniques to be facilitated.

Based on this we think we need a framework for organizing studies in the area of VV&T (including at least model checking, inspection, and testing techniques) aimed at identifying the complementary aspects of these techniques, based on critical defect classification. At first all the available QA techniques should be applied for safety critical application. These studies should be organized in an experimental framework. Other related issues (like safety, reliability, ...) can be addressed in the same framework.