Influences on Regression Testing Strategies in Agile Software Development Environments

David Parsons[1], Teo Susnjak[2] and Manfred Lange[3]

[1,2]*Massey University, Auckland, New Zealand*

[1]d.p.parsons@massey.ac.nz, telephone +64 9 414 0800, fax +64 9 443 9734

[3]*Agile Utilities New Zealand Ltd, Auckland, New Zealand*

Abstract Regression testing is a well-established practice in software development, but in recent years it has seen a change of status and emphasis with the increasing popularity of agile methods, which stress the central role of regression testing in maintaining software quality. The objectives of this article are to investigate regression testing strategies in agile development teams, and identify the factors that can influence the adoption and implementation of this practice. We have used a mixed methods approach to our research, beginning with an analysis of the literature to identify research themes related to the adoption of regression testing techniques under agile methodologies, from which we developed an analytical framework for the study. This was followed by three exploratory case studies that we used to exercise the main elements of the framework, develop some key themes of interest, and devise a questionnaire for the final stage of the study, an on-line survey to explore the main issues identified in the case studies across different contexts. Within our specific sample, our results suggest that organizational maturity is a key factor in effective regression testing practices and that the adoption of such practices is helped by a coherent testing philosophy and change management processes. We also found that the return on investment in automated regression testing was positive for our respondents, and that adopting these practices in the context of agile methods had been a relatively painless process for the organizations in our survey. We conclude that investing in regression testing tools and processes is likely to be beneficial for organizations. However further work is needed in assessing how organizational culture impacts on the quality process and the financial outcomes for commercial software development organizations.

Keywords

Regression testing strategy; exploratory case study; survey; agile software development.

1 Introduction

Regression testing is a well-established practice in software development, with much pioneering work published in the 1980s and 1990s. However the contexts of software testing, and the tools available to support it, have continued to evolve, changing the processes and practices of

regression testing adoption and execution. In recent years the field of regression testing has seen a change of status and emphasis with the increasing popularity of agile methods, which stress the central role of regression testing in maintaining software quality. Regression testing is no longer an occasional, crisis driven activity undertaken by testers who are isolated from the development team. Rather, it is increasingly an essential component of a continuous integration process. Thus regression testing has been raised in profile from a relatively orthogonal activity to a core practice in agile development strategies. An analysis of the literature shows that much research has been carried out into regression testing activities that might be characterized as operational or organizational, but aspects related to regression testing that might be characterized as strategic (policies for adoption, management and evolution) have been less explored. The available literature suggests that there is no single template for regression testing strategy, which will vary based on external context factors such as types of products and levels of regulatory compliance, and internal context factors such as organizational testing philosophy and maturity of teams and organizations. These context factors impact on aspects of practice such as infrastructure investment and change and risk management. In this article we discuss an exploratory field study and the results of a survey designed to investigate regression testing strategy in agile development teams. Our results suggest that organizational maturity is a key factor in effective practices, but that the pathway to adopting such practices, and thus becoming a mature testing organization, is not particularly difficult, assuming a coherent testing philosophy and change management processes.

In this paper we have analyzed the literature on regression testing, and identified three levels of regression testing practice that have been previously investigated by researchers. We have categorized these as operational (specific regression testing practices), organizational (the day to day management of the test process) and strategy (the broader contexts of regression testing, both within the organization and its external market and regulatory context). We note that the nature and role of regression testing in software development organizations has changed over time, with the increasing popularity of agile methods, with their focus on techniques such as test driven development and continuous integration. We observe that current coverage of regression testing strategy in the literature is less extensive than coverage of operational and organizational testing. Therefore we have chosen to focus our study at the level of strategy. We have identified some core themes from the literature that we believe can be categorized as aspects of strategy and from these have developed some research questions that we have chosen to investigate through three case studies and a subsequent survey. We find that certain contextual factors are worthy of further investigation, including organizational maturity, regulatory compliance, investment, and change and risk management.

The remainder of this paper is organized as follows. In Section 2 we provide a definition of regression testing and explore how that has evolved with the increasing popularity of agile methods. We then define our three levels of regression testing, as derived from an analysis of the literature, identifying core areas of research within each level. Section 3 outlines our research motivation and method, while Section 4 further investigates and defines the concept of regression testing strategy and introduces the proposed analytical framework. Section 5 examines factors

within regression testing strategy through a set of case studies and from these the design of the on-line survey is described. Section 6 describes the survey design and conduct, and section 7 presents the survey results. Section 8 provides our conclusions and some suggestions for future work.

2 Background and Definitions

In this paper we refer both to regression testing in general and also how it is specifically applied in agile environments. This section seeks to clarify our definitions, and also categorizes previous research in the area of regression testing into three categories; operational, organizational and strategy.

2.1 A definition of regression testing

Simply put, a regression test suite is a series of tests run against a modified software component under review, following its maintenance (which may be corrective or adaptive) or enhancement (functional or non-functional). Given that there are different types of maintenance and enhancement, there are also different types of regression testing; progressive regression testing (for changes to the software specification) and corrective regression testing (for other changes) (Korel & Al-Yami 1998). Siegel (1996) notes that regression testing has two objectives; to find failures introduced by changes to a tested software component, and to fail to reproduce old failures claimed to be fixed by maintenance. He also makes the point that regression tests are almost always automated, since in high labor-cost markets it is not cost effective to repeat manual tests many times over. Nevertheless, manual regression testing still takes place, as reported by one of the respondents in our survey (described later). It is important to note that regression "is not an innate quality of the suite itself, which is the same as when you ran it against the original component" (Siegel 1996 p.288). Thus we need to look further to define what we mean by regression testing, since its behavior in practice is actually defined by the nature of the original tests that are being re-run. These same tests may perform different roles at different stages of the development process, for example a test case designed for unit testing may be rerun again at an integration level (Tsai, Poonawala & Suganuma 1998).

Regression testing itself may exclude verifications of the interactions of system functions (Gittens et al. 2002). Thus regression testing is not the creation of a new test suite, nor the arbitrary running of every available test. Rather it is the pragmatic selection of a test suite from tests developed from other parts of the test process. One effect of this is that the regression testing process should follow the same standards and procedures as other aspects of the testing process, and cannot be viewed in isolation (Leung & White 1989).

2.2 Regression testing in agile environments

Some of the important published research on regression testing dates from before the widespread adoption of agile methods that took place during the first decade of the 21st century. Agile methods have brought testing in general, and regression testing in particular, much more to the

center of software development, with their emphasis on techniques such as continuous integration and test driven development (Talby, Keren, Hazzan & Dubinsky 2006). These methods and techniques have also changed the nature and emphasis of regression testing. For example, the differences between corrective and progressive regression testing, as outlined in Leung and White (1989), no longer apply in test driven approaches, where the testing and development process is continuous and no longer executed in isolated, discrete stages. Similarly, it is no longer the case that regression testing is usually performed in a crisis situation, nor that regression testing will not be performed by the developers of the product, as was common when Leung & White were writing in 1989. Agile techniques such as test driven development have also changed the role of regression testing as the most likely place to find errors. Hetzel (1984) indicated that the probability of introducing an error during program modification was between 50% and 80%. We would expect this figure to be much lower for regression testing in a test driven environment, since even if the error rate is the same, most of these errors should have been revealed by prior unit testing. Another significant change that occurs in an agile environment is the frequency of testing. Long intervals between testing not only increase the percentage of faults that are undetectable by the original test suite, but can also make it harder to detect some faults due to complex interactions (Kim, Porter & Rothermel 2005). Agile methods encourage frequent build and test cycles, reducing the intervals between tests. In a test driven environment, some form of testing is taking place continuously. It should be noted that this particular study takes place in a context where most of the data has been gathered from organizations applying agile software development methods.

Adopting an agile approach is not, of course, a panacea for all issues in regression testing. Puleio (2006) reports that testing is the greatest of all the challenges faced in migrating to agile software development. In particular, setting up a viable automated test framework is an important step in the move towards an effective agile development process. Even in an environment in which the creation and running of regular tests is an integral part of the build process, effective regression testing faces a number of challenges. Tests are subject to a number of issues in remaining valid and consistent. Depending on the nature of the component being tested (domain model, user interface, data access) and the type of test (functional, non-functional), there are challenges in behavior sensitivity, interface sensitivity, data sensitivity and context sensitivity (Meszaros 2003). Thus a study of regression testing practice in an agile environment may reveal different issues to those in some earlier studies, but nonetheless they are equally challenging.

2.3 Categorizing Regression Testing Research

From the regression testing literature we have identified a number of common research themes that we have categorized into three sets of concerns, with some indicative literature (representative rather than comprehensive - see Table 1.) At the first level are the operational concerns of regression testing. These relate to techniques for appropriate test coverage, prioritization, frequency and selection. Much prior research has been done in these areas, including extensive reviews of the literature such as Engström, Runeson & Skoglund (2010), who analyze previous work on test selection, and Yoo & Harman (2012) who additionally look at prioritization and minimization. The next level is organizational. This research relates more to the day to day

management of regression testing within an organization. This covers areas such as test plans, data, standards and metrics. Again, these have been well explored in the literature. The third level is regression testing strategy, and relates to the impact of context on how an organization adopts, organizes and evolves its regression testing policies and implementation. This includes the external context of market and regulatory forces, the impact of this on the internal context of organizational philosophy and maturity, and the downstream effects of these on infrastructure investment (which may include tools, processes and human resources) and risk management strategies. Unlike the other two categories, this is not about regression testing per se, but about how it fits into the broader software development context. Table 1 shows that these strategic aspects have been considered in isolation in previous studies; however, this is an area of regression testing that has as a whole been less well explored. Therefore in this study we have chosen to focus on multiple factors at the strategic level of analysis.

Level 1: Operational Regression Testing	
Research Focus	*Indicative Literature*
Test Coverage • Ensuring that regression tests adequately exercise the code base	Gittens et al., (2002) Salama (2011)
Test Selection • Selecting which existing tests are currently relevant to exercising the code base	Wong, Horgan, London & Agrawal (1997) Engström, Runeson & Skoglund (2010)
Test Prioritization • Selecting the optimum order in which to run regression tests	Srivastava & Thiagarajan (2002) Orso, Apiwattanapong & Harrold (2003) Yoo & Harman (2012)
Test Frequency • Selecting the optimum frequency with which to run all regression tests	Kim, Porter & Rothermel (2005) Puleio (2006) Zheng, Robinson, Williams & Smiley (2006)
Level 2: Organizational Regression Testing	
Research focus	*Indicative literature*
Test Plans • Contextualizing regression testing within an overall test planning process	Leung & White (1989) Tsai, Poonawala & Suganuma (1998) Engström & Runeson (2010)
Test Data • Selecting the optimum set of test data	Loo & Tsai (1988) Siegel (1996) Orso, Apiwattanapong & Harrold (2003)
Test Standards • Applying rules to the test process to ensure test effectiveness	Leung & White (1989) Damm, Lundberg & Olsson (2005)
Test Metrics • Measuring the effectiveness and efficiency of regression testing	Elbaum, Gable & Rothermel (2001) Gittens et al. (2002)
Level 3: Regression Testing Strategy	
Research focus	*Indicative literature*
External Context • Factors external to the organization that impact on regression testing practice	Fenton & Ohlsson (2000) Svensson & Host (2005)
Internal Context • Factors internal to the organization that impact on regression testing practice	Damm, Lundberg & Olsson (2005) Salama, R. (2011)
Test Infrastructure • Hardware and software systems that are used for regression testing	Do, Elbaum & Rothermel (2004)

Risk Management • Processes and procedures for ensuring quality and mitigating the potential threats to success in regression testing	Persson & Yilmaztürk (2004)

Table 1: The three levels of analysis in regression testing with references to some supporting research.

3 Research Motivation and Method

Our aim in this research study was to clearly define regression testing strategy as distinct from the operational and organizational concerns, a distinction that had emerged as a result of an initial literature review exploring the broad field of regression testing. Our intention was to make use of this definition to subsequently develop an analytical framework with which we could further investigate regression testing practice at the strategic level. This analytical framework was then applied to a pilot field study, from which we developed some themes for exploration in subsequent work, as well as helping us refine our framework and further address our research questions.
The overarching questions that this study attempted to address were the following:

1. How does regression testing strategy vary between organizations?
2. Why do different organizations adopt certain types of regression testing practices?
3. What specific forces influence regression testing strategy?

Our methodology in addressing these research questions was the following:

1. Conduct three exploratory case studies on the topic of strategic regression testing.
2. Analyze these reports according to our framework.
3. Draw out the main themes from the findings to formulate a larger on-line survey on regression testing strategy.

Our methodological approach has some similarities with that of Engström & Runeson (2010), in that it applied a mixed methods approach, beginning with a qualitative enquiry and subsequently developing an on line survey, informed by the outcomes of the initial enquiry. However our approach differed in that whereas the aforementioned study began with a focus group, we began with an exploratory field study comprising three cases. The case studies undertaken for this research were a convenience sample, but nevertheless provided an opportunity to contrast three very different contexts of agile regression testing, with our concept of regression testing strategy as our frame of reference. Our approach was to apply an observational method to an exploratory field study that we hoped would reveal important insights into aspects of regression testing strategy. To some extent this could be classified as a critical case study, in that it provided the opportunity to develop a critique of regression testing strategy in different contexts (Runeson & Host 2009).

4 Regression Testing Strategy

Regression testing strategy relates to the context of testing, the processes, the infrastructure and the organization's risk management practices. An important aspect of research is how regression testing can be successfully introduced into the broader environment of an organization. Although, as we have previously indicated, this area of regression testing has been less explored in the literature than the operational and organizational levels, there is nevertheless some important related work, as outlined in the following section.

4.1 Related Work

Previous work in this area of research that could be broadly categorized as addressing regression testing strategy includes articles by Tsai, Poonawala & Suganuma (1998), Persson & Yilmaztürk (2004) and Salama (2011). One of the key messages from such studies is that regression testing is not a 'one size fits all' practice. Rather, it is very much context driven, and must be implemented in a way that fits the culture and environment of the host organization. As Fenton & Ohlsson (2000) observe, it is difficult to state static software engineering laws, since results, beliefs and practices will vary with the project.

The complexity of the environment can also have an impact on the effort required to introduce testing processes. For example a system that involves large teams across multiple locations (possibly across cultures), using many programming languages, platforms and dependencies between systems, makes it difficult to create test frameworks for the whole environment. In Svensson & Host's (2005) study of the introduction of XP into a complex software development environment, the issue of continuous testing was seen as the second most positive development practice (after coding standards), however it was also perceived as the most difficult practice to introduce. They noted that it requires much effort to introduce comprehensive testing practices into the evolution and maintenance of software, including legacy systems, and that the practices should be introduced early since they require time to introduce properly.

Damm, Lundberg & Olsson (2005) specify that organizations need a mature development process, with management buy-in. The test automation process needs to be easy to avoid developer resistance. The product needs to be developed in testable way, for example by ensuring that products have a common communication interface that can be simulated for testing.

Persson & Yilmaztürk (2004), utilizing a combination of literature and their own case study data, identified 34 common pitfalls in automated testing, most at a level that may be categorized as strategy (according to our own definition). These may be categorized into a smaller number of core themes in automated test strategies that should be addressed. These are; change management, both in introducing automated testing and its on-going evolution, quality control of the test process (including tools), a coherent testing philosophy communicated to all levels of the organization (including the role of automated versus manual testing, and the integration of the test and development roles), and proper investment in staff, training and infrastructure. Persson & Yilmaztürk (2004) also looked at how these various issues manifest themselves in the different phases of the Automated Testing Life-Cycle Methodology (Dustin, Rashka & Paul 1999); the

decision to automate testing, test tool acquisition, the automated testing introduction process, test planning, design, and development, and execution and management of tests. The authors noted that their original supposition was that pitfalls in introducing an automated testing process could be systematically avoided or eliminated via formal risk management. In fact their work demonstrates a strong relation between the success of the projects and informal risk management.

4.2 An Analytical Framework for Strategic Regression Testing Practice

To provide a conceptual framework from which to further investigate regression testing practice at the level of strategy, we have identified a number of core themes from the literature and categorized them into four main areas; (1) the external context, (2) the internal context, (3) the test infrastructure and (4) the change and risk management concerns. The conceptual framework is presented in Figure 1. In this framework, the external context represents constraints that are imposed by outside forces; the market in which the organization operates and the external bodies that regulate that market. The most important forces at play here are the types of projects being developed, and the levels of regulatory compliance required. The most significant aspect of these constraints is that they are not particularly malleable. While an organization may change or develop their product line, this is not something that is likely to be highly flexible, particularly in the short term. Regulatory compliance is entirely defined by external forces and therefore is not under the control of the development organization. In addition, regulations are likely to change, and apply non-flexible deadlines for compliance. Both of these forces are very strong; the market will dictate the success or failure of a company within that market. External bodies will insist on regulatory compliance. Both of these forces will have a considerable impact on the nature of regression testing in organizations, either directly or indirectly (via the internal context). The second category is the internal context, which may be either supportive of regression testing (for example by fostering a positive testing philosophy) or unsupportive (for example by not recognizing the value of building the maturity of regression testing processes). Unlike the external context, the internal context is more subject to change, though is itself influenced by the constraints of the external context. The most significant areas of organizational change that may directly impact on regression testing are the level of maturity of the organization, which will ultimately impact on the maturity of the development and testing teams, and the organizational testing philosophy, which is subject to strategic changes such as a decision to adopt agile software development methods.

Both the external and internal contexts are likely to have an effect on many aspects of regression testing. In particular we have identified the following aspects from the literature as being important, and likely to be directly influenced by these contexts; the infrastructure within which the testing process operates (including the hardware and software architectures, and investment in resources, including training) and change and risk management (quality control of the test process and test practice evolution). Given the generalized nature of the topics under investigation, a largely qualitative methodology is required to investigate these factors (Runeson, Andersson & Host, 2003).

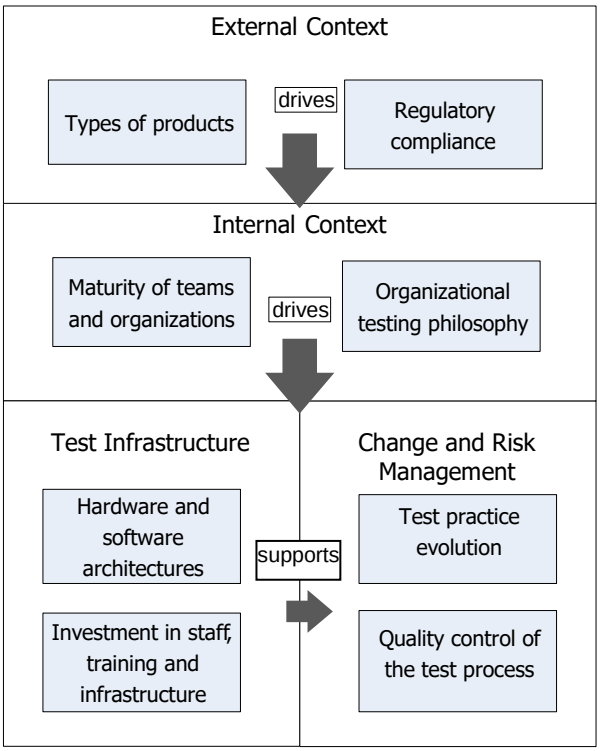

Figure 1: Regression testing strategy and contextual forces

5 Industrial case studies

This section provides some findings from three industrial case studies, exploring some of the key features of how regression testing practices may vary from context to context. In each case we reflect in particular on each of the areas of regression testing strategy previously identified; external and internal contexts, testing infrastructure and change and risk management. These reports come from a range of different software development environments. Each report was provided by one of the authors of the paper as a result of periods of participant observation. The periods of these observations varied from case to case, but ranged from 4 weeks to 4 months. Observations were supported by semi structured interviews.

The findings from the case studies were subsequently analyzed using constructs from our analytical framework in order to uncover themes and formulate more informed questions for a larger online survey.

5.1 Case study descriptions

In this section we briefly outline each of the case studies. For each study we identify factors specifically related to infrastructure, test practice evolution and quality control of the test process. We follow this with a cross–case analysis that collates our findings within the internal and external analysis contexts outlined in Figure 1.

5.1.1 Experience report A – immature process

This experience report comes from a small in-house development team in a public sector organization whose main business was not software development. The team was developing a web based system intended for in-house use but potentially marketable to third party clients. They were subject to a high level of regulatory compliance, but the regression testing process had a low level of maturity. An automated regression testing system had been put in place by an external consultant in the initial stages of an overall agile transformation project. Unfortunately there was limited buy-in to the regression testing system by the team for reasons of trust due to problems with the software on the test servers. In addition, bugs were being picked up in manual testing that should have been found in the automated tests. On reflection, there were several issues that led to, or compounded, the problem.

1. In terms of test practice evolution, further development was needed. The automated regression test suite did not include tests of the web layer, so only manual testing was able to pick up errors in the web components. Many of the team's unit tests were in fact integration tests. Few of the tests were able to isolate errors to specific parts of the system as no mocking was being used.
2. For quality control of the test process, the way that the development and testing approach had been introduced was top down, with inadequate buy-in from the team. Thus quality control was poor.
3. For test infrastructure, the team was working with inadequate investment; they had limited resources, poor quality software and a lack of experience.

5.1.2 Experience report B – building maturity

This experience report comes from an organization producing systems for fruit sorting, combining both hardware and software. The organization had a currently immature (but maturing) regression testing process and was not subject to any regulatory compliance. On-going training was being provided, including design principles with the aim of achieving cleaner architectures, greater coverage in unit tests and a better underlying code base. These design principles included the SOLID concept; Single area of responsibility, open for extension and closed for modification, Liskov substitution, interface segregation and dependency injection (Martin 2011). Key aspects of the testing strategy were:

1. Test practice evolution was well developed, but still maturing. A dedicated tester tested all user stories after assigning them acceptance criteria, using an acceptance test tool. Customers performed site trials of the software.
2. For quality control, the testing philosophy was based around independent code review, Test Driven Development (TDD), mocking, and refactoring legacy code to remove technical debt.
3. Investment was high in tool support for code test coverage (>80% target) and refactoring, with further tool investment being considered. Test infrastructure included both computing resources and fruit sorting machinery.

5.1.3 Experience report C – mature process

The third and final experience report comes from the printing industry, involving the testing of application servers, web servers and native clients for off-the-shelf and custom systems. External forces included regulatory compliance for accounting aspects of the system, some ergonomic requirements and a complex product line caused by acquisitions. Internally the team's process was mature, and had a well-developed infrastructure, that included service oriented architecture and templates for virtual machines (instantiating a template typically took only a few minutes while setting up the VM each time could potentially take hours and require manual intervention.)
Key aspects of the testing strategy were:

1. Test practice evolution was fully developed. Quality controls were based on test coverage, build time and number of code duplicates found in the system. In addition all automated tests had to pass in all test beds, which modeled the most popular and critical deployment scenarios. The defect arrival rate was measured and tracked over time.
2. Quality control was based on a rigorous application of TDD across all layers of the architecture.
3. In infrastructure investment, the organization had invested heavily in hardware and software, including a complete set of refactoring and coverage tools for all team members and a home-grown test bed controller (as no off-the-shelf tool was available.) Network bandwidth was increased to speed up overall network performance. All team members had customized screens to give immediate visibility of the status of all test beds and builds. As a result visibility of the current status of the regression test was increased. At the daily scrum, an additional screen was available to the team to see the status of all test beds.

5.2 Combined analysis of the experience reports

For the first case study, issues relating to shortcomings in the regression test suite related directly to the testing philosophy of the team. Lack of team buy-in related to change management, and lack of investment was also a key issue. External forces were relatively strong (high levels of compliance to data security requirements and possible third party clients). This put pressure on the internal context, given the immaturity of the team's processes and the lack of robustness of the infrastructure. The consequences for regression testing practice were that these practices were less effective than they should have been.

In the second case, the organization had a positive testing philosophy with a strategy of proactive change support by investment in training and tools, much of it targeted at the quality of the process. The test infrastructure was appropriate for the environment. External forces were not excessively pressurizing. The increasing organizational maturity was reflected in increasingly effective regression testing practices.

The third case shows a team with a mature process, supported by a highly developed infrastructure. The regression test practices were based on a strong philosophy, support by

substantial investment and a focus on quality. In a context of significant external forces, the regression testing process proved effective.

Table 2 summarizes the findings from these three cases. From our observations in the field, we found that the level of regulatory compliance did not necessarily guarantee that this would translate to meaningful responses either at the internal context or in the level of regression testing practices. Rather, our experience of the case organizations being studied indicated that the primary external drivers of developing a mature regression testing philosophy were the competitive and dynamic business operating environment that demanded high product agility as well as complex product types. This was complemented by high investment levels into training and both software tools and infrastructure to develop a robust regression testing philosophy.

	Experience Report A	**Experience Report B**	**Experience Report C**
Organization type	Public sector	Private sector - engineering	Private sector – printing industry
Software type	Web based system for in-house use	Automation and machine control for third party clients	Off-the-shelf and customized application and web servers and native clients for in-house use
Required level of regulatory compliance	high	low	medium
Regression testing process maturity	immature	maturing	mature
Regression testing process development team buy-in	low	high	high
Investment level into regression testing: training, staff, infrastructure	low	medium	high
Maturity level of the organizational testing philosophy	initial	defined	optimizing
General development and testing philosophy of the team	Use of manual and integration testing, but poor/partial implementation of unit tests.	TDD	TDD
Company market share and competition	Not a commercial market	High market share and tough competition	High market share
What were the primary external factors influencing the overall strategic level of regression testing practices?	High compliance requirements.	The type of product and the industry that is fast changing and competitive, requiring agility.	Accounting compliance and complex production lines.
In what ways have the external forces affected the internal context of regression testing practices?	Adopting a philosophy of outsourcing the development and maintenance of regression testing practices.	Development of a robust testing philosophy manifesting itself through a focus on extensively refactoring legacy code in order to remove technical debt.	Development of a sophisticated testing philosophy and a mature development team.
In what ways have the internal forces Test Infrastructure	Investment through the outsourcing of the initial setup of an automatic regression testing system to a consultant and	Investment into both staff training as well as a dedicated tester.	High investment in software tools and infrastructure to support regression testing.

influenced the responses in the evolution of the test infrastructure and the change and risk management?		seeking periodic consultancy at critical points thereafter.		
	Change and risk management	Quality control centered on a top-down test process, due to inadequate unit testing, most of the errors had to be identified manually. Change and risk management remained inadequate.	The refactoring of the legacy code enabled the development of more robust unit and integration tests that support the new regression testing practices and better manage change and risk.	A high integration of testing process at all stages of the development cycle, well defined quality assurance and quality control plans.

Table 2: Summary of the three cases.

6 Survey Design and Conduct

Having partly exercised our analytical framework from a small field study, we applied the framework to a larger data set in order to try to gain further insights into how regression testing strategy is influenced by various aspects of context. We distributed an online survey, developed from the key themes identified in the literature regarding contextual factors in regression testing, reinforced by our case study analysis. The main issues that the survey was intended to identify were:

- What were the specific external contexts experienced by developers (in terms of products and compliance)?
- What were the specific internal contexts experienced by the developers (i.e. what were the maturity levels and testing philosophies of their organizations)?
- How did they operate their regression testing process within these contexts (i.e. what were their hardware and software architectures, investment in staff training and infrastructure, quality control of the test process, and change and risk management strategies)?

The intention of the survey was to see if we could identify any further relationships between these various factors and regression testing practice.

The first part of the survey (external context) was addressed by multiple choice questions. We asked what type of software was being developed, for example system software, middleware / infrastructure, application software (in-house use, commercial off-the-shelf, for third party clients) or something else. Multiple selections were possible. When asking about compliance, a single option could be selected from ‘minimal’, ‘limited’, ‘significant’ or ‘major’ (descriptions of these were provided).

For the internal context, we asked a combination of multiple choice and open questions. In asking about the maturity of the development team or organization, we based our options on the 5 CMM levels of maturity from ‘initial’ (lowest) to ‘optimizing’ (highest). Therefore the possible set of values for external forces was within a specified set of options. However for testing philosophy we asked an open ended (free text) question; ‘How would you characterize the organizational testing philosophy?’

All of the remaining questions in the survey were free form text responses. For infrastructure we asked two questions, asking the respondents to describe the hardware and

software architectures they used for regression testing, and what types of investment had been made by the organization in staff, training and infrastructure to support regression testing. In asking about change and risk management we split the question into two parts, one addressing the introduction of automated regression testing, the other addressing its on-going evolution. The final question asked what quality controls were in place for the test process within the organization.

At the end of the survey, we asked a general question about the themes of our study; "Please provide any general thoughts that you have relating to how regression testing can be successfully introduced and maintained, in the context of different types organizations, teams and products."

The survey was created using the Qualtrics tool (licensed by the university) a low risk notification was made to the university ethics committee prior to distribution. Invitations to participate were distributed via the IS World mailing list, the Yahoo Test Driven Development mailing list, the British Computer Society special interest group in software testing and a number of social media channels such as Twitter and LinkedIn. In addition, direct contacts in the software industry were approached and requested to respond to the survey. The survey was university branded, to make its origins clear, and the first page explained its low risk ethics status. The survey was completed to a useable level by 37 respondents, of whom 23 completed the whole survey. This relatively low level of completion is unsurprising given the number of free form text responses. None of these were compulsory so that respondents who were unable or unwilling answer any questions were not prevented from completing the rest of the survey.

7 Survey Results

In this section we present the results of our survey, beginning with the quantitative multiple choice questions, and then exploring the qualitative free text responses.

7.1 External context

The results of the survey revealed that the external context (types of product and regulatory compliance) showed a broad spread of responses. In terms of types of product, all types of software development were represented, though application software for in house or third party use accounted for more than half the sample (Figure 2).

Type of software product	Responses
System software	12
Middleware / Infrastructure	12
Application software (for in-house use)	21
Application software (commercial off-the-shelf)	10
Application software (for third party clients)	18
Other (please specify)	3
Total	76

Figure 2: Types of software products (respondents could answer in multiple categories)

Level of required compliance also varied broadly across the respondents. Figure 3 shows that all levels were represented in the survey.

Level of Compliance	Responses
Minimal or None (only general legal compliance is required)	6
Limited (some external regulations but the overhead is small)	11
Significant (there is significant regulatory compliance required that impacts on the development process)	8
Major (regulatory compliance is a major concern and an essential aspect of the software)	3
Total	28

Figure 3: levels of regulatory compliance impacting on software development

Figure 4 shows a cross tabulation of product type and level of regulatory compliance. Although the figures do not show any strong patterns, they suggest that system software had fewer compliance requirements than other types of software for the respondents in our sample.

		Regulatory Compliance			
		Minimal or None	Limited	Significant	Major
Product	System software	2	3	3	1
	Middleware / Infrastructure	1	2	4	1
	Application software (for in-house use)	1	5	6	3
	Application software (commercial off-the-shelf)	1	1	3	1
	Application software (for third party clients)	2	5	6	2
	Other (please specify)	2	1	0	0

Figure 4: cross tabulation of types of products and levels of regulatory compliance (actual frequencies)

7.2 Internal Context

Looking at the internal context, again we found a broad spread of responses in terms of the 5 levels of organizational maturity, as defined broadly by the descriptors of the CMM (Figure 5).

Maturity Level	Responses
Initial (ad hoc, chaotic)	6
Managed (processes are planned and controlled)	8
Defined (practices standardized and embedded across the organization)	9
Quantitatively Managed (performance data is gathered and analyzed)	4
Optimizing (culture of continuous improvement)	5
Total	32

Figure 5: Maturity levels indicated by the respondents, using CMM descriptors

In analyzing the quantitative relationships between the internal and external contexts, we found no particular pattern related to the types of product and the level of organizational maturity, with the exception of an apparent tendency, within our sample, for software for third party clients to be developed by organizations with lower overall levels of maturity (Figure 6).

		Maturity				
		Initial	Managed	Defined	Quantitatively Managed	Optimizing
Product	System software	2	4	0	1	3
	Middleware / Infrastructure	1	2	1	3	2
	Application software (for in-house use)	1	5	6	2	4
	Application software (commercial off-the-shelf)	1	2	2	1	2
	Application software (for third party clients)	2	4	6	3	1
	Other (please specify)	0	1	1	0	1

Figure 6: cross tabulation of types of products and levels of organizational maturity (actual frequencies)

In looking at regulatory compliance and organizational maturity in our sample, the level of maturity tended to increase with the degree of compliance (Figure 7).

		Maturity				
		Initial	Managed	Defined	Quantitatively Managed	Optimizing
Regulatory Compliance	Minimal or None	3	1	1	0	1
	Limited	1	3	4	1	2
	Significant	0	3	3	2	0
	Major	0	1	0	0	2

Figure 7: cross tabulation of organizational maturity and levels of regulatory compliance (actual frequencies)

Despite the apparent patterns in the data described above, the small sample size means that any relationships must remain speculative since the sample size renders these results statistically insignificant. However we also gathered a large quantity of qualitative data in the form of free-text answers, enabling us to investigate the other themes of our study in more detail, as described in the following sections.

7.3 Qualitative Data Analysis

In this section, the qualitative data gathered from the free text questions is analyzed using the same themes as the preceding analyses. Twelve respondents gave extensive detail in these questions. A further five answered some questions but not all. In one case, confidentiality was given as the reason for not providing answers, in other cases the respondent was unable to answer particular questions, for example because they joined the organization after relevant events had taken place.

7.3.1 Testing philosophy

In terms of testing philosophy, a number of respondents based their responses around test driven development. There were some other comments that reinforced some of the core practices of agile methods in testing, for example "customers dictate the quality assurance, we have the full support of management for test automation" and "testers are part of the development team rather than being an independent unit working after items are developed." However there were other less expected responses, in particular "there are also cultural issues at play within a global organization like ours. Development teams in the far east do not necessarily understand automated testing in the same way as western teams." As indicated earlier in this paper, the vast majority of our respondents have an agile approach to software development. However, one of the respondents stated that "the organization as a whole is relatively poor and waterfall oriented in their approach to quality control…there is a feeling that regression can only be achieved on some occasions because it is too cumbersome. This excuse mentality stops development from actually having it as a goal to achieve full regression testing." This response usefully contextualizes the role of regression testing in agile development, in comparison with more traditional methods, and underlines the changes in the nature of regression testing that have taken place with the adoption of agile methods.

7.3.2 Test Infrastructure

Responses to the first infrastructure question about software and hardware architectures revealed a very wide range of tools and configurations in use, reinforcing the idea that each regression testing context is different and operating under unique requirements. Internal test hardware included Windows and Mac PCs, HP blades, Unix and Linux machines. A number of respondents indicated that their test systems are virtualized but only one referred to this being hosted in the cloud. While test infrastructure frequently mirrored the real world deployment (external context) as closely as possible, one respondent stated that "all attempts are made to replicate production like hardware to simulate 'real life' but some hardware is 'below spec' following time, cost and quality assessments and ROI considerations." Similarly, another respondent stated "All applications are present but not copies of production data and not scaled to production."

Use of software tools included a range of automated processes including application lifecycle management, source code analysis and code coverage tools. However one respondent said their regression testing was still being done manually. Some teams are using commercial testing tools (HP QC and QTP, and Microsoft TFS), others are using open source tools (Hudson / Jenkins, Ant). In many cases the tools are specific to what is being tested (e.g. dedicated tools for testing Rails and JavaScript).

In terms of both hardware and software infrastructure, the influence of the external context of software development can clearly be seen. The type(s) of application(s) that are supported by the development organizations influenced the nature of the deployed software in terms of both application types and scale of deployment. Both of these factors influenced the test hardware and software, including whether the tests tools were commercial or open source.

When asked what types of investment have been made by organizations in staff, training and infrastructure to support regression testing, most respondents said that investment had covered both systems and training. However this varied between respondents, with some suggesting that infrastructure investment was greater than training, others saying the opposite. It was noticeable that only a few respondents noted large investments being made. Several stated that investment had been minimal. Nevertheless one respondent noted that such investment does have a payoff; "we have actively gone out of our way to implement automated regression wherever possible. A major benefit is to reduce time spent by developers on manually repeated testing."

7.3.3 Change and risk management

Most respondents seemed to find difficulty answering the question about how the organization handled change and risk management when it introduced automated regression testing. In some cases this was because the respondent was not working for the organization when this took place. In other cases there was no previous system because automated regression testing was there from the start. Where the question was answered in detail, these organizations had moved in a planned and managed way from manual to automated testing. Perhaps one reason that respondents found this question difficult to answer was they that did not see this change as being high risk. As one respondent stated, "there is only risk in not implementing it, why would you not do it?" In addition, the change appeared to be seamless, as outlined by another respondent; "There was no real change, the automated tests just replaced the manual ones being run. Over time the extent of automated testing on some applications has grown tremendously but that has no effect on the change or risk management process."

In addressing the question about how organizations handle change and risk management in the on-going evolution of automated regression testing, a number of respondents commented on the continually evolving technology and tool set, and ways of upgrading their processes to incorporate these. For example; "the project team is open to trying new technologies and techniques. By using the release branch isolation and version labeling, this allows for new practices to be tried without any major impact if the outcome is not desirable." Trying new approaches was generally seen as desirable; "We openly allow developers to try new technology and techniques - you can make mistakes, just don't repeat them!" Another respondent commented; "ongoing monitoring of system performance in both test and production environments is also used to reduce risk." Drivers for change included a desire to increase the speed of the testing process (i.e. internal context) and changes to the tool set available (i.e. infrastructure driven by the external context.)

Responses to the question about quality control of the test process fell into three categories; general aspects of process management, application of specific quality criteria, and tools used to support quality. Process management included formal code reviews, business sign-off on all changes, regression matrices, clear demarcation on system access and release procedures within a controlled environment. Explicit quality criteria included process, performance and integrity criteria including minimum code coverage, build time, metrics, and only merging tested, unduplicated code. Much of this was automated. Separation of the test and production contexts

was one important factor: “No access whatsoever to production environments.” The responses to this question suggested that development teams take this aspect very seriously and have rigorous procedures in place.

7.3.4 General responses

At the end of the survey, respondents were asked to provide any general thoughts they had relating to how regression testing can be successfully introduced and maintained, in the context of different types of organizations, teams and products. Responses stressed the importance of commitment to the process from all stakeholders; management, customers and developers, as well as testers. Dedicated resources are required to organize and maintain test environments, and one person must have overall responsibility for the integrity of those environments. Tool support was also seen as central. It was regarded as essential in automating testing procedures that provide comprehensive coverage and utilities, with tools to help developers ensure test quality. Automation of test generation was proposed as an important progression from manual test generation. It was also indicated that certain development tools encourage testing more than others. A process of continuous improvement is required that is based on concrete measures (benchmarks, metrics and service levels). One respondent sounded a warning about not having a quality automated regression testing process in place; “There are many projects that we are not directly involved in that have no test automation. The result is a haphazard out of control environment, no real controls in place…Quality costs. Most managers just do not seem to get it.” Looking ahead, one respondent stated; “Automation of test generation is perhaps the next step, where software monitors inputs and outputs from various elements of code, and then checks what happens when these inputs are exceeded.”

7.3.5 Relationships between external context and regression testing practices

From the qualitative responses it did not appear that the types of software application had any close relationship with regression testing practices. This may be because the main differences in infrastructure were driven by the size of the software development operation, rather than by the type of software being written. Since several of the respondents referred to their test environments being virtualized they did not provide details of, for example, how many machines were used for testing, since this would be variable. There seemed to be a tendency for teams working on commercial off-the-shelf software (COTS) or third party software, particularly where customization was included, to have a testing philosophy that emphasized the use of TDD. Interestingly a high level of regulatory compliance did not seem to suggest that organizations would necessarily have a greater focus on risk management or investment, since some organizations working with low level of compliance often had equal commitment to these.

7.3.6 Relationships between internal factors and regression testing practices

In general the higher the level of organizational maturity, the more processes had been put in place for risk management, and the higher the level of investment had been in supporting regression testing. While most respondents referred to investment in infrastructure, the more mature organizations also reported investment in training. Organizations with a specific testing philosophy (TDD) tended to have better processes in place and a more coherent set of testing practices and infrastructure.

7.4 Answering the Research Questions

In section 3 we proposed three research questions. First, we asked 'How does regression testing strategy vary between organizations?' The results of our study suggest that it varies greatly, dependent upon context. This reinforces, from a broader perspective, Engström, Runeson & Skoglund's (2010) findings that test selection techniques varied widely between organizational contexts. Our three case studies indicated that strategy was reflected in the level of organizational maturity. However there was also a practical disjoint between the intention of strategy and actual practice. For example in the first case study, the policy to use continuous integration was present but in practice this was not being followed through. Perhaps the most obvious variation in strategy discovered was the wide range of test infrastructure being used, tailored to the types of application context (Our survey also revealed that these application contexts covered many different types of software development). Given that strategy encompasses the set of practices adopted for regression testing, our second question was 'Why do different organizations adopt certain types of regression testing practices?' Factors that emerged from our study include; the software development method adopted by the organization (e.g. waterfall v. agile), the distribution of the development team (different philosophy on automated testing in offshore teams), the type of software being developed (e.g. COTS teams appeared to favor TDD) and the level of management buy-in for investing in test automation. Our final question was 'What specific forces influence regression testing strategy?' This question is related to question 2, but sought to identify any common factors that applied across organizations. From our data set we are unable to generalize, however from our respondents we note that the type of software being developed can impact on the level of regulatory compliance, the level of maturity of an organization seemed to be less critical when developing software for third parties, and that regulatory compliance may be one driver for organizational maturity, but that it is not the only factor.

8 Conclusions and Future Work

We undertook this research in order to try to identify the possible relationships between external context factors, internal context actors and the infrastructure and management of regression testing practices. The results of our study indicate that this is a complex area, with many factors at play. However we can recognize some themes from our data that may give some insights in to how

regression testing strategy may be influenced by various factors. In particular we have seen that while organizational maturity appears to have some kind of relationship with the levels of regulatory compliance, which we might intuitively expect, these relationships do not necessarily follow through to specific practices in regression testing. Maturity tends to be more significant than compliance, not least because many mature organizations do not claim to be working with high levels of compliance. While investment seems to be variable between different organizations, there is little doubt that investment in automated regression testing had paid off for many of our respondents. It also appears that adopting it in the context of agile methods is a relatively painless path for an organization to take and, given the comments regarding the problem of organizations not taking this path, it seems highly likely that such changes would see a return on investment.

Our work so far has been based on the development of an analytical framework from the literature, with a focus on strategic aspects of regression testing. We have applied this model to the analysis of an exploratory field study and a small scale survey. In future work we need to refine the analytical framework in the light of our previous findings, and gather more extensive data to further explore the themes that have begun to emerge from this study. Analyzing the responses from the case studies and the survey have resulted in identification of options for future research, for example the impact of organizational culture on the quality process, in particular assessing how it can influence the financial results of commercial companies, given the return on investment issues raised by a number of our respondents..

References

Damm, L-O., Lundberg, L., & Olsson, D. (2005). Introducing Test Automation and Test-Driven Development: An Experience Report. *Electronic Notes in Theoretical Computer Science 116*, 3–15.

Do, H., Elbaum, S. & Rothermel, G. (2004). Infrastructure support for controlled experimentation with software testing and regression testing techniques. *Proceedings of 2004 International Symposium on Empirical Software Engineering (ISESE '04)* 60 – 70.

Dustin, E., Rashka, J., & Paul, J. (1999). Automated Software Testing, Addison-Wesley.

Elbaum, S., Gable, D. & Rothermel, G. (2001). Understanding and measuring the sources of variation in the prioritization of regression test suites. *Proceedings. of Seventh International Software Metrics Symposium (METRICS 2001)*, 169 – 179.

Engström, E., Runeson, P. & Skoglund, M. (2010). A systematic review on regression test selection techniques. *Information and Software Technology*, *52*, 14-30.

Engström, E. & Runeson, P. (2010). A Qualitative Survey of Regression Testing Practices. In: *Proceedings 11th International Conference on Product-Focused Software Process Improvement (PROFES).* Ed. by M. Ali Babar, M. Vierimaa, and M. Oivo. Vol. 6156. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, pp.3-16.

Fenton, N.E. & Ohlsson, N. (2000). Quantitative analysis of faults and failures in a complex software system. *IEEE Transactions on Software Engineering*, *26*(8).

Gittens, M., Lutfiyya, H., Bauer, M., Godwin, D., Kim, Y.W., & Gupta, P. (2002). An empirical evaluation of system and regression testing. *Proceedings of the 2002 conference of the Centre for Advanced Studies on Collaborative research (CASCON '02)*, 3.

Hetzel, W. (1984). *The Complete Guide to Software Testing*, QED Information Sciences, Wellesley, Mass.

Kim, J-M, Porter, A. & Rothermel, G. (2005). An Empirical Study of Regression Test Application Frequency. *Software Testing, Verification and Reliability, 15*(4), 257–279.

Korel, B., & AI-Yami, A. (1998) Automated Regression Test Generation. *Proceedings of ISSTA 98* Clearwater Beach Florida USA

Leung, H.K.N., & White, L. (1989). Insights into regression testing. *Proceedings of the Conference on Software Maintenance*, 60 – 69

Loo, P.S., & Tsai, W.K. (1988). Random testing revisited. *Information and Software Technology. 30*(7), 402–417

McKeeman, W.M. (1998). Differential Testing for Software. *Digital Technical Journal, 10*(1), 100-107.

Martin, R. (2011). *Agile Software Development, Principles, Patterns, and Practices*, Pearson.

Meszaros, G. (2003). Agile Regression Testing Using Record & Playback. *Proceedings of the 18th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications (OOPSLA '03)*. 353-360. ACM.

Orso, A., Apiwattanapong, T, & Harrold, M.J. (2003). Leveraging field data for impact analysis and regression testing. *Proceedings of the 9th European software engineering conference*, 128 - 137 ACM.

Persson, C., & Yilmaztürk, N. (2004). Establishment of Automated Regression Testing at ABB: Industrial Experience Report on 'Avoiding the Pitfalls'. *Proceedings of the 19th IEEE International Conference on Automated Software Engineering.*

Puleio, M. (2006). How Not to do Agile Testing. *Proceedings of Agile Conference, 2006*, 307-314

Runeson, P., Andersson, C. & Host, M. (2003). Test processes in software product evolution - a qualitative survey on the state of practice. *Journal of Software Maintenance and Evolution: Research and Practice, 15*, 41-59.

Runeson, P. & Höst, M. (2009). Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering* 14, 131-164

Saff, D., & Ernst, M.D. (2003). Reducing wasted development time via continuous testing. *Proceedings of the 14th International Symposium on Software Reliability Engineering*, 281 – 292

Salama, R. (2011). A Regression Testing Framework for Financial Time-Series Databases. An Effective Combination of FitNesse, Scala, and KDB/Q. *Proceedings of the ACM international conference on Object oriented programming systems languages and applications (SPLASH '11)*, 149-154

Siegel, S. (1996). *Object Oriented Software Testing: A Hierarchical Approach*. New York: Wiley.

Srivastava, A. & Thiagarajan, J. (2002). Effectively prioritizing tests in development environment. *Proceedings of the 2002 ACM SIGSOFT international symposium on Software testing and analysis (ISSTA '02)*, 97 - 106

Svensson, H., & Host, M. (2005). Introducing an Agile Process in a Software Maintenance and Evolution Organization. *Proceedings of the 9th European Conference on Software Maintenance and Reengineering*. 256 – 264.

Talby, D, Keren, A, Hazzan, O, & Dubinsky, Y. (2006). Agile software testing in a large-scale project. *IEEE Software, 23*(4), 30 – 37

Tsai, W-T., Poonawala, M., & Suganuma, H. (1998). Regression testing in an industrial environment. *Communications of the ACM, 41*(5), 81–86.

Wong, W.E., Horgan, J.R., London, S., & Agrawal, H. (1997). A Study of Effective Regression Testing in Practice, *Proceedings of the 8th International Symposium On Software Reliability Engineering*, 264 – 274

Yoo, S., & Harman, M. (2012). Regression testing minimization, selection and prioritization: a survey. *Software Testing, Verification and Reliability, 22*(2), 67–120.

Zheng, J., Robinson, B., Williams, L. & Smiley, K. (2006). Applying regression test selection for COTS-based applications. *Proceedings of the 28th international conference on Software engineering (ICSE '06)*, 512-522.

Appendix – Survey Questions

Question1: What type(s) of software product(s) are being regression tested in your organization? (select all that apply)

A. System software
B. Middleware / Infrastructure
C. Application software (for in-house use)
D. Application software (commercial off-the-shelf)
E. Application software (for third party clients)
F. Other (please specify)

Question 2: What is the maturity of the development team, organization, or other relevant context within which regression testing takes place? These options are based on the 5 CMM levels of maturity from Initial (lowest) to Optimizing (highest). Select the highest level that is appropriate for your organization / team.

A. Initial (ad hoc, chaotic)
B. Managed (processes are planned and controlled)
C. Defined (practices are standardized and embedded across the organization
D. Quantitatively Managed (performance data is gathered and analyzed)
E. Optimizing (culture of continuous improvement)

Question 3: What level of regulatory compliance is required for your development context?

A. Minimal or None (only general legal compliance is required)
B. Limited (some external regulations have to be complied with but the overhead is small)
C. Significant (there is significant regulatory compliance required that impacts on the development process)
D. Major (regulatory compliance is a major concern and an essential aspect of the software)

The following questions were all free text responses:

Question 4: What hardware configuration is used for regression testing in your organization?

Question 5: What software architecture is used for regression testing in your organization?

Question 6: How did your organization handle change and risk management when it introduced automated regression testing?

Question 7: How does your organization handle change and risk management in the on-going evolution of automated regression testing?

Question 8: What quality controls are in place for the test process within your organization?

Question 9: How would you characterize the organizational testing philosophy?

Question 10: What types of investment have been made by your organization in staff, training and infrastructure to support regression testing?

Question 11: Please provide any general thoughts that you have relating to how regression testing can be successfully introduced and maintained, in the context of different types organizations, teams and products.