Exploring Risk-based Testing and Its Implications[1]

Felix Redmill
Redmill Consultancy

Email: Felix.Redmill@ncl.ac.uk

ABSTRACT

If software cannot be tested exhaustively, it must be tested selectively. But, on what should selection be based in order to maximise test effectiveness? It seems sensible to concentrate on the parts of the software where the risks are greatest, but what risks should be sought and how can they be identified and analysed? 'Risk-based testing' is a term in current use, for example in an accredited test-practitioners' course syllabus, but there is no broadly accepted definition of the phrase and no literature or body of knowledge to underpin the subject implied by it. Moreover, there has so far been no suggestion that it requires an understanding of the subject of risk. This paper examines what is implied by risk-based testing, shows that its practice requires an understanding of risk, and points to the need for research into the topic and the development of a body of knowledge to underpin it.

Keywords: risk-based testing, risk, software testing, test planning.

1 INTRODUCTION

For well known reasons, exhaustive software testing is seldom cost-effective and often not possible in finite time. Planned testing must therefore be selective. But on what basis should selection be made? A starting assumption is that it would make sense to carry out the most demanding testing on the most important aspects of the software, and this is indeed what many testers claim to do. But what criteria do they use for hitting the right spots? How do they derive confidence that they have done it? The current author carried out a small informal survey of a number of testers, asking them what basis they used for planning their testing.

The first replied, 'We test everything'.
The second said, 'We test what is most important.'
When asked how she knew what was most important, she replied, 'It's usually obvious.'
On being asked, 'What indications do you look for?' she frowned but did not reply.
The conversation with the third tester went like this:
'We test where the risks are greatest.'
'What kinds of risks?'
'What do you mean?'
'How do you define risk?'
'Umm …'

Both testers who recognised that exhaustive testing is not feasible postulated criteria for selective testing. But both turned out to be unsure of what they meant. The strategy of basing software test planning on risk is intuitively sensible and, indeed, what many planners think they do, but is it possible to execute it effectively without an understanding of the subject of risk? It will be shown in this paper that, without such an understanding, it

[1] Published in Software Testing, Verification and Reliability, Vol. 14, No. 1, March 2004

is, at best, difficult to know where to look for the risks, what types of risks to be concerned with, and how to determine which ones are 'the greatest'. Confronted by an unfamiliar concept, test planners would probably seek help, but when the basis of testing is risk, with which everyone claims familiarity, most are unaware that they lack the required competence. They don't know that they don't know. Nor is there any body of knowledge to which they can confidently refer. 'Risk-based testing' is neither consistently defined nor supported by literature on either theory or practice.

Yet the UK's Information Systems Examinations Board has included a section on risk-based testing in its practitioner testing syllabus [1]. Without both requisite knowledge and reference material (but see [2]) it is unlikely that lecturers and trainers will consistently, or even correctly, interpret and teach it.

This paper examines the implications of risk-based testing, starting with a brief consideration of the subject of risk (but does not address software constructed by mathematically formal languages). It attempts to show that, whereas risk-based testing has been used intuitively, it is now time for it to be taken seriously as a subject for study, research, and definition. Risk-based testing requires risk-based thinking and this, once in operation, points to the need for improvements not only in testing but also in other parts of the development life cycle.

Focusing testing according to risk implies making judgements about test coverage, the number of tests conducted, the choice of test techniques and types of review, the use of and balance between dynamic testing and static analysis, and other issues. The paper does not address these testing decisions but, rather, explores the use of risk as a basis for them. The paper is concerned with test planning and not test specification.

2 THE SUBJECT OF RISK

It was suggested above that risk, though familiar, is poorly understood. This section offers an introduction to some of its concepts.

2.1 Risk Perception

The concept of risk is familiar, intuitively understood, and universally perceived to imply uncertainty and undesirable outcome. When the word is mentioned, people think they know what it means. Everyone practices risk management, making judgements about avoiding, reducing, accepting, and sharing risks in their lives, and there is often an implicit assumption that proven competence in simple situations is adequate in complex ones. But, as considerable empirical evidence from the field of psychology (e.g. [3]) demonstrates, this is not the case. Not only do the mental heuristics used by humans to solve complex problems not lead to 'rational' results [4], but also perceptions of risk vary from person to person, depending on many subjective variables (e.g. [5]). While the cited research is based on a normative perspective, there is also work that does not assume the independence of decisions from other knowledge held by the decision-maker (e.g. [6]). However, all research concludes that decisions about risk are influenced by the perceptions and biases of the decision-makers.

Moreover, Henrion and Fischhoff [7] have shown that this is so in experts as well as in the lay public. Those untrained in the subject of risk may not be aware that their risk estimate or assessment is not the only one possible, or even the most appropriate in the circumstances, and they are unlikely to be aware that it is biased.

Traditional risk analysis in science and engineering appears to avoid the complexities of perception by taking risk to be a function of two variables, probability and consequence. Yet, at all stages of the analysis process there is subjectivity - in defining the scope of the analysis, selecting the techniques to be employed and the ways in which they are used, choosing tolerability criteria, and making other decisions [8, 9].

2.2 Defining Risk

Definitions of risk differ in detail. The Royal Society [10] defines it as 'the probability that a particular adverse event occurs during a stated period of time, or results from a particular challenge', and the International Electrotechnical Commission as the 'combination of the probability of occurrence of harm and the severity of that harm' [11]. Both definitions include probability and, although the first does not explicitly mention consequence, 'adverse event' implies it. The Royal Society's definition suggests quantitative analysis and says exactly how to arrive at a result - by determining a probability (though it gives no hint of the difficulty in doing so). The IEC's 'definition' is imprecise and not a definition at all, but it allows for qualitative as well as quantitative analysis.

In the Royal Society's definition, the word 'particular' is a reminder that the adverse event needs to be defined precisely - otherwise the derived probability might refer to something else. To exemplify this, consider three adverse events, loss of control by a car driver, a resulting accident, and consequent death or injury. Although the three may occur consecutively in a train of events, it is possible to lose control without having an accident and to have an accident without death, injury, or even damage to the vehicle. Thus, the probability of loss of control is greater than that of accident, which, in turn, is greater than that of damage. Vehicle and component manufacturers are particularly interested in loss of control, for their failures could cause it, whereas they have no control over the events to which it might lead. The probability of accident is of most concern to insurance companies, and the probability of death or injury is of interest to the health service. What is of concern, and thus what is defined as being the undesirable event, is a function of the perspective taken. The probability derived in risk estimation depends on what the event is, how it is defined, and from whose perspective it is perceived. It is therefore important to be precise about these factors, which a purely intuitive consideration of risk is likely to overlook.

2.3 Risk Analysis

Risk analysis has traditionally been quantitative, based on probability theory. But two assumptions on which probability theory hinges - randomness and large populations - do not hold for bespoke software (software failures are systematic and not random). Nor does historic data exist on which to base predictions of future performance. Of course, a fault history may exist after testing, but as the purpose of testing is to find bugs for correction and every correction changes the software, this is not predictive. When the use of probabilistic methods cannot be justified, and when, for whatever reason, confidence in numbers is low, qualitative risk analysis is likely to be more appropriate than quantitative analysis. A means of combining the two in dependability and risk assessments is provided by Bayesian Belief Networks, an introduction to which is given by Neil, Littlewood and Fenton [12], and research into how this may be applied to risk-based testing may be worthwhile.

3 USING THE COMPONENTS OF RISK TO INFORM TEST PLANNING

A number of general points about risk were made in the previous section, and now the use of its two components - consequence and probability - will be considered. Note that the term 'likelihood' will, for reasons to be explained, be employed instead of 'probability'. Use of the combination of the two components will also be considered.

3.1 Consequence

Different consequences arise from different undesirable events. So, what risk - or, what undesirable event - is likely to be of interest as a basis for test planning? The immediate, and often the only, answer for many is that of system failure, and this gives rise to two significant related questions. First, what is meant by 'failure'? And, second, from whose perspective is the risk to be considered?

Failure is usually taken to mean a deviation of system behaviour from that which is 'expected'. The basis of expectation is not always stated, but confidence in judgements is likely to be increased if it is a documented specification. Most systems perform a number of different functions and provide a number of different types of output, and deviation from any expected (specified) functional performance or output provision would constitute a failure. Thus, basing test planning on the risk of failure requires the identification and exploration of the risks arising out of the various types of failure and of the potential locations of those faults that would result in the most consequential failures. Each system has a number of stakeholders, for example, its users, its owner company, and the clients of the company, and each failure is likely to have a different consequence for each stakeholder. Thus, understanding the consequences requires identification of the stakeholders' viewpoints and determination of the effects of different types of failure on them.

The different types of failure can be identified in a number of ways, for example, from past experience, and by expert judgement, brainstorming sessions, and more formal enquiries. Then, the potential consequences may be determined by analysis. The parts of the software whose faults would give rise to the failures of greatest consequence would be the parts on which to focus testing, but it is not likely to be easy to identify those parts, in which respect two points are relevant. The first is that many failures are caused not by individual items of software but by interactions between items, the relevant faults being found in integration testing. The second point is that, whereas a single subsystem may perform a given function or control a certain type of output, the function or output is usually derived from many items of software that collect, sort, store, analyse and retrieve data. Thus, all the relevant items of software need to be identified - perhaps from a top-down design of the system, or perhaps from a cause-and-effect tree - and included (with their modes of interaction) in the 'part' of the software under consideration. There is scope for research into the processes of identification (such as a modification of fault tree analysis), but in this paper it is assumed that a relevant 'part' of the software can be identified.

So how do consequences point to the part of the software that should be of concern? The answer depends on whose perspective is taken. Consider the example of a conceptually simple bank account system. This may have a subsystem for use by bank operators and another for use by bank clients via the internet, and, as far as the test planners are concerned, either could give rise to the greater adverse consequences (potential losses), depending on the information available to them. In many organisations, test planners' information is limited to that gleaned from talking to programmers, reading parts of the system requirements specification, and perhaps having discussions with intended system users. Mostly these sources take the users' perspective, considering such matters as system down-time, lost transactions, the loss of production by operations staff, corrupted data, re-work, and maintenance costs. But the significance of these may be dwarfed by that of other potential losses not suggested by the information available to the test planners, such as those arising if the clients' subsystem went wrong - for example, if customers were denied access to their accounts, if their statements were incorrect, or if they were given access to the accounts of others. From the perspectives of bank operators and maintenance staff, such failures may cause small inconvenience and low costs, or may not even be imagined. In fact, if publicised in the press, they have the potential to create huge losses in terms of diminished goodwill, customer defections, decline of market share, and marketing costs to regain the company's former position. But to recognise these possibilities, the test planners would need to take a business rather than a user perspective, to know the objectives for the system as well as its required functions, and to understand the implications of the customer functions. For this, they would need information from the customer's senior management and strategists as well as from the functional specification and system operators - and, in practice, they are seldom party to such information.

Two points are therefore worth noting. One is that, prior to the estimation of consequences, there must be a determination of which perspective they should be considered from. In some cases all consequences may need to be estimated so as to compare, prioritise, or sum them, and in others it may be sufficient only to determine the consequences from a particular viewpoint.

It should be remembered that there are more viewpoints than those of the system owners and system users. The developers, testers, maintenance team, and development organisation, among others, are all at risk from various events. For example, system failure within a certain time of delivery, or a number of system failures, could result in diminished reputation of the developers, or, because of the need to occupy staff with re-work, the loss of an opportunity to move on to a new project. Unacceptable interaction of the system with users or other systems could have the same effect. And system failure or poor performance could result in huge warranty costs in the form of maintenance effort, re-work by developers, and re-testing. Taking a risk-based view leads to the recognition that the greatest risks may not lie where they are first assumed to be.

The second key point is that in order to estimate consequences, information must be obtained from appropriate sources - and from sources in which justifiable confidence is held. Appropriateness implies, for example, that, if the consequences of failure on system owners is to be understood, information should be derived from those who have defined the system's objectives, such as senior managers and system strategists - and subsequent review is always a requirement, for business objectives change with time. The need for confidence in the source is a reminder that all members of a category of stakeholder (e.g. users, senior managers) do not possess the same knowledge or experience, and do not hold equally valid opinions. In seeking information it is important to select a source that is not only nominally appropriate but also judged to be of high pedigree.

3.2 Likelihood

As already pointed out, the necessary conditions for a probabilistic analysis of software failure are unlikely to exist, and certainly not for newly written software, which is the subject of this discussion. The more general word, 'likelihood', will therefore be used so as not to seem to imply confidence in numbers. In this section, ways in which likelihood may be used to inform test planning are considered.

First, it may be assumed that the likelihood of failure of a given subsystem is not known, but that the potential consequences, if it did fail, can be estimated. There are two ways of using this information. One is in relative terms, by prioritising the potential consequences of the failures of all subsystems and focusing testing accordingly. The other is to use the knowledge in absolute terms, by relating consequence to the importance of the subsystem not failing, i.e., the greater the consequence of failure, the lower the likelihood of failure that should be achieved. On this basis, the requirement is to make the risk of failure low (or below a certain threshold), and the assumption is that increasing the concentration or rigour of testing, and then fixing and testing again, would achieve this.

If there were an understanding of what each of a number of types of testing would achieve, it might be possible to equate a desired risk tolerability level to a particular test programme. Indeed, this is proposed in the Motor Industry Software Reliability Association's software development standard [13], in which four levels of testing are proposed. In this way, test plans and specifications can be devised, not merely to focus more on the most important parts of the software, but to achieve, with some confidence, a risk reduction appropriate to what is desired. The types, coverage, and volume of testing are determined from the desired final level of risk.

But there is a another way in which likelihood may be used. Suppose something about the risk of failure of the software to start with could be determined. Suppose it was estimated to be, say, medium; perhaps the testing should be more onerous than if it had been deduced to be low. In order for the assessment to inform test planning, it would have to be done prior to testing and would therefore need to be based on knowledge of the design and construction of the software, which could be gained in a number of ways:

- Observation of the quality of the structure and documentation of the code;
- The quality of previous work by the software's designer and programmers;
- The complexity of the software, derived from a pre-test automated measurement.

Such a scheme is a valid use of risk as a basis for test planning. Indeed, it gets to the heart of focusing on risk, for it goes to the source of the risks. Just as, in the previous section, testing was based not on the combination of the two components of risk but on one factor only (consequence), so here there is single-factor analysis (of likelihood) for the purpose of informing test planning. But for confidence to be placed in the correlation between development-related factors and the risk of system failure, records of previous work of the development team in question would be required, and maintaining such data is not the norm in most organisations. Further, 'bad' developers subjected to training and improved discipline are expected to improve, thus negating their existing assessments, so relevant data would need to be kept up-to-date and verified prior to each usage. Yet, it is quite possible for test managers to maintain records of the quality of the work of software designers and coders for use in estimating the risks attached to their current products.

3.3 Combining the Two Components

In certain circumstances, single-factor analysis can be turned into a full risk analysis, because means have now been identified for estimating both the likelihood and the consequence of failure (of a 'part' of the software). It would be feasible to combine the two and thus derive risk values, and this could be done qualitatively by use of a risk matrix, each cell of which represents a risk. Different testing programmes could then be specified for the software in the various risk categories, with the most rigorous being applied to those parts that carry the highest risks.

It would also be feasible to go beyond this. If it is possible for the consequence of failure of a part of the software to be estimated before it is tested, this could be done even earlier - prior to its design and construction. So why not do it then, and assign the better designers and programmers to those parts of the software whose failures carry the greatest consequences? This would reduce risks in advance, rather than building in the problems that exacerbate them and then hoping for them to be found in testing. This is within the control of project and development managers and not the test planners, but it is certainly within the scope of test planners and managers to broach the subject.

4 SUMMARY OF HOW RISK MAY BE APPLIED TO SYSTEM FAILURE

The previous section offered insights into how the elements of risk may be used as the basis of test planning. Here the themes already raised are summarised and added to.

First, the consequences of failure of parts of the software might be compared, and the parts with the highest consequences selected for the most rigorous testing. But other judgements need to be made. Should the objective be to reduce the risk of failure to some given level - and, if so, what level, and how can it be known when it has been achieved? Or should the objective be to minimise the risk of failure - and, if so, within what constraints? Or should the testing of each part of the software be defined by budget or time constraints determined by the risks involved? This relative use of consequence could result in some parts of the software not being tested, or being under-tested, due to time or budget limitations. Importantly, it should be recognised that the use of risk as a basis for test planning does not provide a formula for perfection; it requires the formulation of questions, the search for answers, and the taking of decisions.

Second, the consequence of failure might be used in absolute terms. That is, for each part of the software, the potential consequences provide the basis for determining a tolerable likelihood of failure and a test programme considered appropriate to achieving it. As it cannot be proved by measurement that the risk of failure has been reduced to the desired level, there is an implicit assumption that the chosen test programme will achieve the desired level of reliability. Test planners should therefore consider how confidence in this might be gained. An alternative process is to ignore the notion of determining a likelihood of failure (given that it is not measurable) and to equate the consequence of failure directly with a test programme. Again, it cannot be proved that the test programme will lead to the probability-of-failure target being met.

Third, an estimate of the likelihood of failure of an item of software may be used as the basis for planning the item's testing. Such an assessment would be based not only on the software's structure, complexity, and documentation quality, but also on the quality of the previous work of its designers and programmers - as recorded on the current and other projects. This would to some be controversial, but in dealing with risk there is no better course of action than to get to the source - and the designers and programmers are very much connected to the likelihood of failure of their software.

Fourth, a risk value for an item of software can be derived by combining the likelihood assessment with the determined consequence of its failure. As the combination would need to be qualitative, the result would be a risk category, on which test planning could be based. Necessarily, the information from which the two risk components (likelihood and consequence) are estimated will be derived from different sources, the one part from users and business strategists and the other from technical sources close to the developers.

5 RISKS OTHER THAN THOSE OF SYSTEM FAILURE

The above points are based on the risk of software failure. But to restrict risk-based testing to this would be to place an arbitrary and unnecessary limit on the method. There are other relevant criteria, and care needs to be taken in choosing and defining the viewpoint from which to make evaluations of consequence or risk. Though the system user's viewpoint may at first seem most appropriate, the system owner's (the business perspective) is often the most significant. For example, when it is critical to bring a product to market within a defined time, the risks of not doing so may be considered more significant than those attached to system failure.

In systems for which availability is more critical than reliability, it may be appropriate to focus testing less on where component failure might occur than on where it should be detected and compensated for, such as in redundancy mechanisms and diagnostic and recovery software. (This is not a judgement on what is best, merely an example of the fact that decisions need to be made.) The maintainer's perspective may also be relevant, for maintenance costs and the need for extra call-out staff can in some cases be a significant factor in risk judgements.

There are also other factors that motivate testers and programmers. Their concern may be in the local causes of failure, such as the number of bugs in an item of software, or the types of bugs, and it is in these areas that risk may be of interest, for example during the planning of unit testing.

Each stakeholder's viewpoint carries a risk that is different from all others, and taking a risk-based approach implies the need to identify, define, and understand the relevant viewpoints, examples of which are offered in the following list.

- The business whose objectives the system is intended to fulfil;
- Customers of that business;
- System operators (users);
- The maintenance team;
- System developers, who may need to budget and plan for the future investment of time and effort, particularly if there is a warranty on the system;
- The test team.

Moreover, it is not only financial loss that might be consequential. From all of the viewpoints there are other considerations, for example:

- A loss of reputation, which could affect not only current relationships with respect to the system under consideration, but also future goodwill and business possibilities;
- Trades-off between resources, human and otherwise, which may need to be diverted from elsewhere in the event of system or project failure;
- Compromises in the quality of service provided, or between levels of service;
- The need for and management of maintenance teams;
- Compensation and legal costs, and the price of transferring risk via insurance;
- Security of the system or of entities protected by the system;
- Safety of people, property or the environment.

Some of these consequences may be quantified (though, not necessarily accurately) and reduced to common, comparable, units such as money. But this is not the case with some others, such as human life, environmental loss or degradation, and reputation. This inability to cost accurately may be a problem if comparisons are to be made. In some risk analyses, cost benefit analysis is employed, but, though it is useful when marketed goods (with commonly agreed prices) are concerned, it meets an obstacle in the case of non-marketed goods for which costs are open to the subjective judgement of the decision maker. Similarly, compensation and legal costs may not easily be determined in advance and may require actuarial services, which themselves can be costly. Before risk analysis is carried out, its objectives need to be determined and defined [8].

Emphasis has been laid on identifying the risks from various viewpoints, and in distinguishing between different types of risk, but it should be pointed out that the total risk from a given viewpoint may require the summation of a number of risks, for example of system failure, the loss of reputation, and the loss of market share.

6 NOTES ON RISK IDENTIFICATION AND ANALYSIS

The risks that are identified depend on where the search for them is conducted. Looking for the risks that are most significant in the given circumstances depends on determining what is of significance, and what is significant depends on viewpoint and on circumstances. Basing test planning on risk requires a risk-based way of thinking. Identifying one decision and making it, but ignoring all others, is not handling risk effectively. Risk-based testing is not throwing all possible effort at one intuitively identified risk, but making judgements about and between risks and the factors that influence them - and this demands that the risks be identified and understood in the first place.

Risk identification and analysis may be carried out in many ways. The transfer of techniques from the safety-critical-systems domain offers an opportunity for research, but risk analysis may also be informed by expert judgement and by experience from past projects and systems of the same type as the one under consideration. Local knowledge may form the basis for assessing which subsystems are exceptionally complex or technically novel, and well managed brainstorming sessions can be particularly valuable. The principal constituents of practical risk analysis are the right information - with high confidence that it is complete and correct - and a good understanding of risk. Only then can techniques be applied appropriately - but it is not the purpose of this paper to offer an exposition on techniques.

7 BACKWARD RISKS

Risk-based testing may seem to address only the 'forward risks' (from the testers' perspective - see Figure 1) - i.e. those affecting system use. But risk-based thinking immediately directs attention also to the 'backward risks' - those imposed on the testers by the developers. It also leads to the recognition that the developers too could look forward one step and identify, analyse, and manage the risks that they pose, such as:

- A delay of more than a given time to the delivery of an item of software;
- The software items not being delivered in the planned order;
- Significant bugs not being found in testing because they are masked by numerous other, less significant, bugs ('white noise') that should have been found and rectified by the coders in unit testing;
- The need to retest more than once because of hasty and unchecked bug fixing.

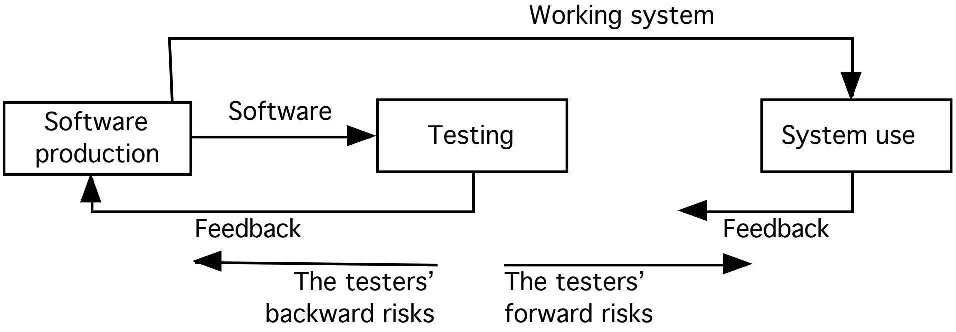

Figure 1: Testing as the second stage of a three-staged process

This is not an exhaustive list of testers' backward risks (the developers' forward risks). Examples of well known causes are:

- Poor software design;
- Casual programming;
- Use of untrained programmers;Failure of programmers to take unit testing seriously;
- Lack of inspection or peer review of software by developers;
- Importantly and frequently, ineffective management.

Utopian assumptions of software development may not allow for such causes, but so frequent are they that most test teams take them for granted and do not even complain about them. Yet, 'you can't make a silk purse out of a sow's ear', and testing, however meticulous, does not build quality into the software. It merely identifies some of the incidences of bad quality. Not only would software quality be improved, but so would the testers' lot, if testers identified their backward risks and called on development management to put their houses in order. Indeed, if the development management adopted a risk-based way of thinking, and addressed their forward risks, huge improvements and considerable financial savings in projects could ensue. The risk-based way is not restricted to testing, or to the risk of system failure.

If it is worth carrying out a risk analysis, it is worth doing so early in a project. If the consequences of the failure of a subsystem are high, it is not enough for the testers to know this; it should have been known to the designers and programmers and, indeed, to the development manager. Then reduction of the risk of failure would not be left to the testers but would be considered in the design and construction of the software. The higher the required integrity of the software, the more important it is to introduce appropriate risk-reduction features such as redundancy, diagnostic software, and protection systems at the design stage, rather than to rely on testing to make it fault-free. Then, given an architecture that distinguishes the modules or subsystems of the highest criticality, the development manager can further reduce risk where it matters by allocating the best programmers and module testers to them. Finally, testing can be planned to focus on the software according to its required integrity, but as the last, rather than the main, barrier against risk.

8 DISCUSSION

The term 'risk-based testing' has been in currency for some time, but, in the absence of an agreed definition underpinned by an accepted body of knowledge, and supported by an associated literature, it is applied to almost any pragmatic approach to testing. This paper interprets the term, 'risk-based testing', literally, i.e. as the use of risk to inform test planning, and explores how risk can be used in this way. In doing this, the paper has shown that there are several approaches to applying risk to test planning and that risk varies according to the viewpoint from which it is perceived.

The use of risk not only increases the effectiveness of testing but also reveals problems in development that create or increase the risks. It is, or can be, a powerful tool and is therefore worth developing and employing. But for this, the hitherto intuitive appreciation of the subject is not sufficient. Although everyone has an intuitive understanding of it, risk is a difficult and elusive subject. How it is perceived depends on a person's experience, biases, and personality. Moreover, because the subject is so intuitively familiar, few are aware that their intuition is inadequate in non-simple situations. Effective use of risk as a basis for testing demands the identification and analysis of risks and the making of judgements between them. Further, if this application of risk turns out to be a good thing, it needs to be documented, repeated, and taught, which requires a clear understanding of the subject and its application. Risk therefore needs to be included in the education and training not only of testers but, indeed, of all software-development practitioners and managers.

Employing risk as the basis for test planning does not provide a formula for perfection. It leads the planner to the formulation of questions, the quest for information, and the taking of decisions, so it is not a tool to which responsibility can be abdicated. Test planners must retain responsibility, but informed use of risk can provide illuminating guidance. Risk implies uncertainty, and where there is uncertainty it is not proof of correctness that must be sought but increased confidence. But confidence in what? Confidence that execution of the test plan (followed by correction) will reduce the significant risks to tolerable levels. And a quest to understand where confidence needs to be increased and to take decisions that increases it directs the test planners to pursue a methodical questioning approach, with the result that testing is effective as well as merely efficiently executed.

Risk is a broad subject. As well as being employed in engineering and scientific risk analysis, it is studied and researched in the fields of psychology, sociology, and anthropology. A mere requirement for the study of risk could lead to irrelevant or unfocussed syllabuses on the one hand, and their inconsistent interpretation by lecturers and trainers on the other. What is now required is research into the field of risk-based testing, leading to a carefully defined body of knowledge that first explains the essential principles of risk, risk analysis, and risk management, and then describes their application to software test planning - and project planning too. It is suggested that the themes discussed in this paper are relevant to such research. In addition, the transfer of techniques such as fault tree analysis [14] from the safety-critical systems domain to the field of risk-based testing is worth exploration.

There is also a need for literature that reports on the use - successful or otherwise - of the application of risk as described in this paper. Empirical results of controlled trials are required, and this need offers opportunities to meticulous and observant practitioners as well as to researchers.

The purpose of this paper is not to teach either testing or risk, but to say enough about risk to show that, if it were better understood by test planners, if could be used as an effective basis of test planning. Perhaps there has always been an implicit assumption that testing is based on risk, and perhaps it is true that testing - together with fixing and re-testing - is likely to reduce risk, but it is not true that risk has been used effectively as the basis of test planning. But it could be, if there were a fuller understanding of the subject.

REFERENCES

[1] Information Systems Examinations Board. *Practitioner Certificate in Software Testing - Guidelines and Syllabus. Version 1.1*. British Computer Society, Swindon, 2001

[2] Gerrard P and Thompson N. *Risk-Based E-Business Testing*. Artech House, Norwood, 2002

[3] Kahneman D and Tversky A. Prospect Theory: An Analysis of Decision Under Risk. *Econometrica*, 47, 263-291, 1979

[4] Tversky A and Kahneman D. The Framing of Decisions and the Psychology of Choice. *Science*, Vol 211, 30 January 1981, 453-458, 1981

[5] Slovic P. Perceptions of Risk: Reflections on the Psychometric Paradigm. In Krimsky S and Golding D (eds): *Social Theories of Risk*. Praeger, Westport, 1992

[6] Ayton P and Hardman D K. The StAR Risk Advisor: Psychological Arguments for Qualitative Risk Assessment. In: Redmill F and Anderson T (Eds.), *Safer Systems - Proceedings of the Fifth Safety-critical Systems Symposium, Brighton UK*. Springer-Verlag, London, 136-159, 1997

[7] Henrion M and Fischhoff B. Assessing Uncertainty in Physical Constants. *American Journal of Physics*, 54, 791-798, 1986

[8] Redmill. Risk Analysis - a subjective process. *Engineering Management Journal*, 12, 2, 91-96, 2002

[9] Redmill. Exploring Subjectivity in Hazard Analysis. *Engineering Management Journal*, 12, 3, 139-144, 2002

[10] Royal Society. *Risk: Analysis, Perception and Management*. Royal Society, London, 1992

[11] IEC. *International Standard IEC 61508: Functional Safety of Electrical/Electronic/Programmable Electronic Systems*. International Electrotechnical Commission, Geneva, 2000

[12] Neil M, Littlewood B and Fenton N. Applying Bayesian Belief Networks to System Dependability Assessment. In: Redmill F and Anderson T (Eds.), *Safety-critical Systems: The Convergence of High Tech and Human Factors - Proceedings of the Fourth Safety-critical Systems Symposium, Leeds UK*. Springer-Verlag, London, 71-94, 1996

[13] MISRA. *Development Guidelines for Vehicle Based Software*. The Motor Industry Software Reliability Association, UK, 1994

[14] Vesely W E, Goldberg F F, Roberts N H and Haasl D F. *Fault Tree Handbook*. U.S. Nuclear Regulatory Commission, Washington D.C., 1981