

A RESPONSE TO QUESTIONS RAISED

Exploring the demands of a Best Practice Method to mitigate risks associated with the electronic trading of Insurance.

INTRODUCTION

The primary objective of this document is to set out the various requirements of a system of best practice related to the mitigation of risk associated with the electronic trading of Insurance Products. It has been brought about as a response to the various questions that abound related to the potential requirements of the most recent output from FSA.

Before any attempt can be made to suggest a form of best practice, there are a number of definitions and introductory explanations required. Additionally, it is only possible to set out the building blocks of a suitable form of best practice, as this document is also intended to stimulate discussion, rather than set out to be an autocratic standard.

The following paragraphs set out, first, the arguments pertaining to “context”, then arguments pertaining to “issues” and finally set out the fundamental requirements of an adequate process. Thus, this document is divided into three main sections.

Credentials:

The writer has spent the majority of his career (over 20 years) in System Testing, enjoys an international reputation and is regarded as a leader in his field. Additionally, he has contributed to many standard works on the subject and spent many years in close association with testing and quality issues related to the electronic trading of insurance. Collectively, with his colleagues at Ingenuity and with the experiences of many Insurer clients, the writer is able to set out for discussion an authoritative and substantive document based upon experience.

CONTEXT

Without setting out the words of the FSA and attempting to provide an intricate analysis of their meaning, it is possible to suggest the basic tenet of what they will be seeking to have satisfied by Insurers.

There will be a requirement upon Insurance companies to provide demonstrable proof that they have taken reasonable steps to mitigate any risk; that products traded electronically are neither misrepresented nor making a detrimental impact upon the book of business.

Misrepresentation is a reasonably straightforward concept, suggesting that the cover indicated and terms required must be accurately portrayed.

A detrimental impact on a book of business would be construed as collecting insufficient premium and rendering the book of business at risk of not being able to meet its potential commitments i.e. mismanagement.

These basic arguments are related to the Insurance companies proving that they have taken all reasonable steps to mitigate risks, where third parties are involved in representing them for the introduction of new business, adjustments to existing business and the processing of renewal invitations.

Specifically, the third parties referred to are using computerised methods to obtain introductions to an insurer and, thereafter, to conduct business.

These third parties are either Insurance intermediaries or more “direct” but all have in common the principal means of representing an insurer by way of one electronic means or another. The reader would anticipate these to be Quotation Systems, possibly involving the full business cycle.

The types of system currently in use vary from business systems supplied to Intermediaries, such as that produced by Misys Financial Systems, to bespoke systems, typified by Coversure or Hill House Hammond or World Wide Web “dot.com” providers.

Where they may differ is in the nature of the “physical intermediary” between the software and the “member of the public”. Where they do not differ is that they are, generally speaking, the products of software developments outside of the direct control of the Insurer that they are representing.

The requirement of the FSA is that the Insurers conduct themselves in such a way that they exercise all reasonable control, over those to whom such works have been out-sourced, to prevent possible detriment and misrepresentation.

THE ISSUES

As stated above the issues are primarily related to:

- Software development
- Out sourcing
- Maintenance of “quality”.

Inevitably, the representation on computerised systems requires a software development to be involved.

Invariably, the greater part of all works is undertaken by organisations acting on behalf of a number of Insurers and Intermediaries.

When distilled, the variance between what an Insurer expects of an electronic trading method and what they actually receive is perceived as a “quality” issue.

The subject of risk mitigation cannot ignore these issues but they are not the only ones, as the following paragraphs explore.

The prudent manager responsible for electronically traded products will recognise that there are many differential dynamics which are the province of software development, rather than the day-to-day exploits of marketing insurance products.

The significant arguments related to the software, when viewed from the perspective of the prudent manager, are as follows:

- Who is responsible for the software quality?
- What is an acceptable level of quality?
- How is that quality maintained?
- How is that quality controlled?
- Why does the EDI element of the business system not accord with the business practice of my Insurance Company?

Fundamentally, as the Insurer is responsible for the requirements, which are subsequently programmed by the software provider, the Insurer MUST carry a significant responsibility for ensuring the quality of its products.

Once the subject has migrated to that of “Software Development”, the rules relating to software development apply.

The single cause of the greatest number of variances between what was wanted of a computer system and what it actually delivered, is that of “Requirements” or “The System Specification”.

There is a small complexity in this particular context, in that the provider of the software is generally providing a system upon which a number of Insurer’s products are represented. However, in general, the overall system breaks down into a number of basic components:

- A back office system, to handle the needs of the intermediary and accounting
- A quotation process where details of a proposed risk are gathered
- The quotation “engine”, which handles the requirements of an individual Insurer-product
- The Document production and EDI (Business Process) elements {where applicable}.

It is the quotation process and the EDI transmission elements where there is often a difference between what was hoped for and what was achieved.

In both respects, whilst there are fixed elements, such as question-sets and EDI message formats, there are significant elements of Insurer-product specific requirement involved.

The electronic trading systems are essentially a collection of software “capable of performing the business process”, at least in it’s widest sense. The specifics are the subject of Insurer specification and there are compelling reasons why the Insurer must take responsibility for the outcome. In highlight these are:

- The software creators are generally able to demonstrate their overall ability to perform the task intended. If not, it would be germane for the Insurer to be most cautious about entering into any representation of their products on such a system.
- The Insurer is responsible for providing the specific requirements.
- The majority of problems are demonstrable as having their proximate cause in the subject of inadequate specification material, rather than an inability, on the part of a software developer, to programme.

Institutionally, the writer has found most requirements to be articulated in a way that is most straightforward to an Insurance Underwriter and capable of being interpreted by a developer. Therein lies the main difficulty. Requirements must never be articulated in a way that demands “interpretation”. They must be clear and precise and voiced in such a way as a developer can programme.

This demands that Product Requirements are created and maintained by those trained in and adept at producing software specifications, following an analysis of business requirements.

The subject of Polaris and Productwriter code will be cited, by some, as a means of avoiding such pitfalls and is patently a complete misnomer. Productwriter code is, by any reasonable definition, a piece of software. Thus, the delivery of a piece of Productwriter code to a Software integrator suffers the risk of making matters worse rather than better.

Why is this?

First, there is still ample opportunity for the Productwriter code to reflect poor requirements, unless these have been made abundantly clear.

Second, there are vast differences between the data capture models of the different integrators, providing many opportunities for a miss-fit between seemingly sound Productwriter code and the Software it is to be integrated with.

Therefore from a strictly software quality management viewpoint, the introduction of Productwriter code into the “quality equation” provides at least two areas of failure where there used to be one (Product Requirements).

A BEST PRACTICE MODEL

From the foregoing, any attempt to mitigate risk is, clearly, having to take into account the dynamics of software development, product requirement’s specification and the myriad of different software providers and their methods.

It must be stated at this point, that all of the foregoing are matters of “public and available” knowledge, albeit in the province of Software Development, Quality Management and System Testing. The information and it’s relevance to recognising and mitigating risks associated with electronic trading are unquestionable.

The one common factor is that this subject is one of adopting a pertinent Quality Management Model, even if it is not fully appreciated as such. The salient features of such a model will need to address the dynamics considered above but will also need to be realistic.

First, the Quality Management Model will need to:

- Address the measurement of “risk”.
- Be resilient enough to cope with many and varied representations of products
- Cope with frequent product changes.
- Recognise the Dynamics of Software Development Issues in an environment which generates frequent software updates (monthly, at least, in most circumstances).
- Have a mechanism to deal with issues raised.
- Be demonstrable in its design, execution and effectiveness.

In more detail:

Any quality management model requires some form of measurement to take place, otherwise there can be no grounds for suggesting that any form of “management” is in place.

The tenet of the FSA requirement is that the Insurer or, by way of a Service Level Agreement, their “supplier” is under an onus to demonstrate that appropriate steps have been taken.

This would suggest that the management of the risk requires the insurer to understand the dynamics and respond to them accordingly.

The measurement of risk must have boundaries of expectation, have a defined approach and a method of dealing with and responding to the outcomes of the management process.

How do we measure the risk? We test.

How do we test? We ensure that we seek to prove that the most important indicators to unacceptable risk have been identified and attended to.

How is this achieved? By creating a cohesive Test Strategy.

This demands that a Test Strategy of some form is created to respond to the demands of a Quality Management Model, that the Quality Management Model addresses risk and can show that all appropriate preventative measures have been enacted.

The central element would appear to be measurement by way of testing and by having a mechanism to regulate the testing pursuit, together with some suitable system to deal with the outcomes.

This leads us to the manner in which the demand to mitigate risk can be approached in practical terms.

A PRAGMATIC APPROACH

How to Measure?

This is the province of the System Tester and must be covered in a Test Strategy Document.

What to Measure?

This has to depend not upon the ability to ignore risk by way of economic blinkers but to seek to check that all important indicators have been tested. Again, the subject of a Test Strategy.

How Much to Measure?

This has to depend upon a different type of risk, associated with exposure, again part of a pertinent Test Strategy.

A suitable Test Strategy would recognise that there has to be a reasonable limit to testing, which is incapable of being quantified outside of the context of a specific product.

That limit has to take into account the most important features of validation, at the quotation stage, together with calculation of premium and any statements related to policy terms and any other limitations.

The later business process testing has to ensure that the terms and conditions are accurately conveyed to the insurer and that their internal systems cope with them, with an equivalent precision.

At the quotation stage, even for a simple motor product, there are Millions of test permutations. Household carries at least five times the complexity and Commercial Package business, at least 7 times the complexity.

It is not possible to test all permutations and act upon the results of these tests on all electronic trading solutions for every release of software, there simply is not enough time and the cost of so doing would make the form of business entirely cost-prohibitive.

Conversely, it is possible to identify the most important indicators, create tests based upon them and execute these in different depths and breadths of coverage commensurate with the level of exposure to risk, as presented by the different trading methods.

The same may not be true of EDI, Point of Sale Documents and Business Process but the permutations of test are greatly reduced by being limited to acceptable business.

There are methods for setting out Test Strategies and Methods and the minimum requirement of a test strategy to address the situation in this context is as follows:

1. To create a set of Test Conditions which exercise all important aspects of underwriting validation and rating requirements.
2. Devise a method, which ensures that the adequacy of the tests conforms to both the expectations of the business, AND have been devised by way of formal structured methods.
3. All Test Conditions are capable of being shown to be both independently devised and conformant to a formal method such as “MIT’s” (Most Important Tests), thus allowing the management of the coverage adequacy.
4. That a method be in place to allow for a precis of the tests to be identified as the “must be executed” tests, in the case of small systems or “rushed” releases.
5. That the Tester and “the business” have a suitable means of reviewing the pertinence of any risk managed decisions in concert with the relevant competent authority within the business.
6. That measures are in place to report all problems and identify their severity in relation to business risk, as viewed by a competent authority.
7. That the proximate causes of problems are resolved BEFORE engaging upon their correction at a software level.
8. That the test effort is able to assist in the resolution of problem causes and their correction.
9. That Retests are performed and, where feasible, completed prior to a corrective release.
10. That the needs of the developer, in terms of programme change, unit and integration testing are taken into account in scheduling problem alleviation programming.
11. That all corrections are scheduled in such a way as to address known issues related to too rapid a fix-on-fail policy.
12. That pre-release testing is adopted wherever possible, without detriment to an impending release.
13. To ensure that the necessary tests are conducted with each release of software and that a suitable amount of coverage follows any fix-on-fail test requirement.
14. That evidence is retained related to all aspects of test condition design, test case creation and test execution.
15. The Test process must provide timely reports to assist in steering future testing and requirements production, so as to engage in a true “Process of continuing improvement”.
16. That Test Metrics are maintained to demonstrate that problem yields are under control and to show that the incidence of live errors is being used to refine the test process.
17. That periodic reviews are undertaken to demonstrate that the testing methods, test cases and coverage are maintaining their pertinence to the overall goals.

CONCLUSION

The above explanations and assertions are influenced by significant experience in Quality Management and Testing Process generally and in the electronic trading of insurance specifically.

Where the notion of “independent” has been suggested, this does not mean an “independent contractor”. It requires that the processes have been the subject of independent assessment.

Let there be no doubt that sufficient formal methods exist to support the above assertions and that it is entirely reasonable to be expected to follow such formality, with proofs, as being the only demonstration that effective means have been deployed to mitigate risk.

This does not mean to say that hundreds of thousands of tests have to be executed or that inappropriately large quantities of testing need to be conducted when the risk provided by exposure is slight. However, there is a need to demonstrate that formal methods have been adopted to measure, address and mitigate risk. Anything less may well be construed as not attending to managing risk.

To state it clearly: To “tick a box” and say “it has been tested”, without proof and demonstration or to assume that the software developer carries the entire responsibility for the correctness of an Insurance Products’ representation is imprudent.

Whilst the foregoing is pragmatic, it does not constitute any more than an ordering of the various considerations. A formal Test-Strategy would be the subject of a more specific document.

The writer has attempted only to set out the arguments from an informed and expert standpoint. There is no desire to anticipate that this document defines the only way forward. It does address the detail behind the requirement and it can be assumed that clients of Ingenuity will expect to have a refined and detailed version of the test-strategy document, once all discussions have been completed.

© Simon Mills FRSA 2003 ~ All Rights Reserved,
Managing Director, Ingenuity System Testing Services Ltd

References

Software System Testing and Quality Assurance, Beizer 1996
Software Testing Techniques, Beizer 1990
Software Test Automation, Fewster & Graham [Chap 17. Mills] 1999
Introducing Software Testing, Tamres 2002
Testing Insurance Software Properly, Mills 2001
The Mythical Man Month, Frederick P Brooks, 1975 & 1995
The MIT's (Most Important Tests) Method, Hutcheson 1997