



ERUDINE[®]
replaces legacy with flexibility



REGULATORY COMPLIANCE IN IT SYSTEMS

Overcoming legacy to implement a sustainable regime

WHITE PAPER BY HUGH BEEVER

Contents

Introduction	1
Demonstrating compliance	1
1. Say what you do	3
2. Do what you say	3
3. Prove it	4
Traceability	4
Implementing a sustainable regime	5
Conclusion	6

© Remote Operations Limited (trading as Erudine) 2009

No part of this publication can be reproduced, stored in a retrieval system, transmitted or made available to the public in electronic form or by any other means (electronic, mechanical, photocopying, recording or otherwise) without the written permission of the publisher. Whilst every care has been taken to ensure the accuracy of the editorial content the publisher makes no representation and gives no warranty as to its accuracy and cannot accept any liability for any direct, indirect or consequential damage or loss howsoever caused arising out of or in connection with the content of this publication.

Erudine, Erudine Behaviour Engine, the Erudine logo and erudine.com are trademarks or registered trademarks of Remote Operations Limited (trading as Erudine) in the United Kingdom, other countries, or both. These and other Erudine trademarked terms are marked on their first occurrence in the information with the appropriate symbol (® or ™), indicating UK registered or common law trademarks owned by Erudine at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries.

The trademarks of other companies are marked on their first occurrence with the appropriate symbol (® or ™). Other company, product or service names may be trademarks or service marks of others.

In line with Erudine's environmental policy, this document has been optimised for double-sided printing.

Introduction

Corporate governance and risk management is moving rapidly up the C-level agenda. Even when the economic landscape is settled, governance and risk management play a large part in running a successful economy and maintaining a level playing field for all. The OECD describes good corporate governance as central to the health of our economies and their stability and, in times of crisis such as the recent recession, poor governance is often considered a major contributor to the loss of confidence in key markets.

It is easy to see how compliance plays an important role in maintaining an effective governance regime. Indeed many organisations now adopt a unified approach to governance, risk and compliance (GRC) and a whole industry is developing around this discipline. However, the unified scope is too wide for a single paper to cover and here we focus on how organisations deal with compliance within the IT systems that support their business.

Good corporate governance is central to the health of our economies.

There are several reasons why IT system compliance is important, including:

- Adherence to existing legislation: organisations must demonstrate how their processes meet or exceed the standards set by relevant legislation for their current markets. This includes maintaining compliance through system and legislative evolution.
- Entry into new markets: financial markets in particular are heavily regulated and, in order to participate, organisations must demonstrate how their systems comply with the relevant framework.
- Globalisation: the regulatory regimes in different geographic regions may have a common kernel but with local variations. These variations may apply to the initiation of transactions, the participation in transactions, the ability to see certain information or the way the information is presented.
- Rationalisation following mergers and acquisitions: there are strong business drivers to achieve cost savings by rationalising systems where two or more applications support the similar business process. Care must be taken to ensure the rationalised system meets all appropriate regulatory requirements.

Whatever the business driver, a dynamic and sustainable approach to compliance will significantly reduce the risk faced by organisations as well as the costs of carrying out audits and meeting reporting standards. This paper outlines the challenges faced by many organisations in demonstrating compliance within legacy IT systems. It goes on to explore how technology can be used to address these issues and implement a sustainable compliance regime.

Demonstrating compliance in IT systems should be trivial ... if only life were so easy!

Demonstrating compliance

The essence of compliance can be summarised as:

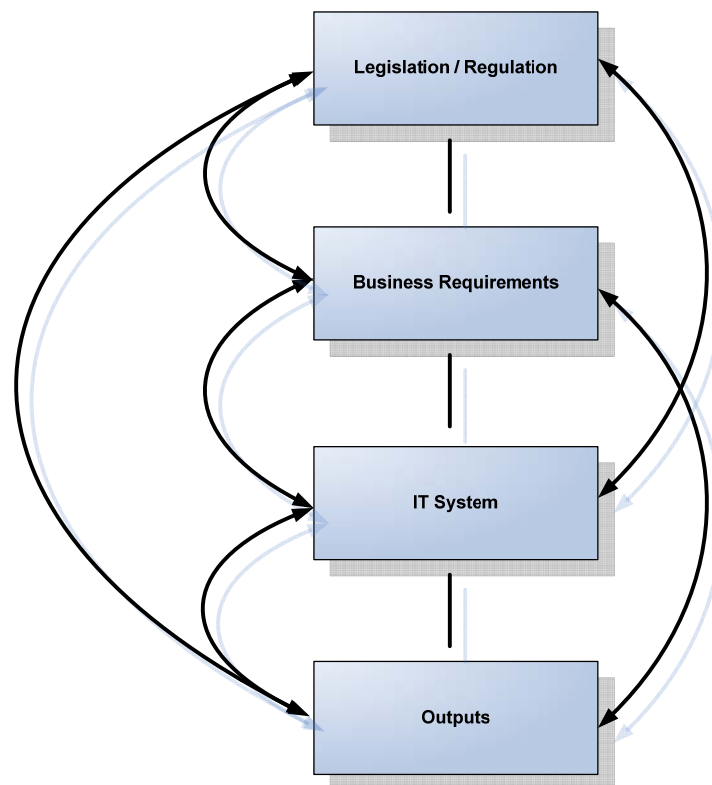
1. Say what you do
2. Do what you say
3. Prove it

With this in mind, demonstrating compliance in Information Technology (IT) should be trivial. After all, a computer system is simply an electro-mechanical device doing what it has been programmed to do in a repeatable and deterministic way. If only life were so easy!

Inputs and outputs of a system are insufficient to demonstrate compliance – there must be understanding of how the output was achieved.

A typical IT system will likely have been developed to a formal specification written several years ago, derived in part from the prevailing legislation and documented using software tools that are no longer supported. The code produced by the developers will have been subjected to extensive testing, hopefully using data and test cases independently derived from the requirements, and put into live operation where it will probably have been maintained by a separate support team. These boundaries between different disciplines and technologies, and the different personnel, can make it difficult to establish a clear link between specification and behaviour.

Furthermore, whilst an output from a system may appear to comply with regulations, this is meaningless unless the process by which it was derived from the inputs can be demonstrated. For example, the MiFID (Markets in Financial Instruments Directive) regulations require that financial transaction reporting to be complete and accurate. Showing a date and time at which the transaction was executed may appear to fulfil this brief, however, this is not sufficient to show IT system compliance. It is also necessary for the organisation to show how the application has generated the information, how this relates to the documentation and, ultimately, how the legislation has been interpreted and implemented.



Providing traceability from the outputs of a system back to the legislation is extremely problematic with legacy systems. This is particularly true where a system has been subject to considerable change, perhaps in response to a competitive market, and the links between behaviour, documentation and prevailing legislation have not been maintained. Often the documentation will remain unchanged or will be complemented by reams of change notes and addenda that try to explain how the system has changed. In some cases, the resulting outputs may not be fully

validated against the prevailing legislation leading to an inadvertent breach of compliance.

The main challenges of traceability can be summarised as:

- The business requirements may not accurately record the applicable regulatory clause(s)
- The link between the IT system and the requirements may be out of date, inaccurate or uni-directional. Further, some of the business requirements may be being met outside of the IT system.
- The outputs produced may be anomalous due to the complex interaction of multiple application modules.

Increasingly, traditional approaches to IT system development and change management are proving incapable of keeping pace with regulatory change. This was evident in a recent case where the FSA fined a major financial institution, in part for failing to implement fully the regulatory changes introduced by MiFID in 2007. So what approaches are organisations taking in endeavouring to meet their compliance obligations?

1. Say what you do

If we follow the mantra from the introduction, the first stage is to say what the system does. This involves a detailed review of the system documentation describing how, both operationally and functionally, the system adheres to the regulatory framework.

Many legacy systems fail at this very first hurdle because the system documentation is either inadequate, out of date or, in some cases, non-existent.

It is common to have a comprehensive library of technical documentation describing how to run the system, what the architectural dependencies are and how to resolve the most common issues. However, the documentation describing the business processes seldom keeps pace with changes applied to the operational system.

In order to demonstrate compliance it must be possible to select a specific piece of legislation or a regulatory requirement and identify the business requirements through which it is implemented. This requires the documentation to be maintained in a repository or management tool that can record often complex links between different textual requirements. These links need to be preserved and extended as changes are made to the system.

2. Do what you say

From a robust set of business requirements that accurately reflect the prevailing regulatory framework the next step is to demonstrate how the application implements those requirements. There is a psychological boundary to be crossed here; between the disciplines and technologies associated with business analysis and those associated with development.

Tracing from written requirements to the software code base often requires 'backward chaining'; that is, searching the code repository for references to the specific business requirement. This is because the written documentation seldom holds a reference to the code through which it is implemented. Some code repositories use tags or labels to record the requirement identifier but, in older systems, it can be a case of scanning the code for comments.

Knowing what the system does is the first step in demonstrating compliance. Many legacy systems fail at this first hurdle because the documentation is inadequate.

Recording everything is a sledgehammer approach to retaining evidence. However, the effort required to extract meaning from the resulting digital landfill can be huge.

Once a link has been established it is possible to review the unit tests associated with the software module(s) and assess the degree to which these have been tested against the requirements and underlying legislation.

There is a significant weakness here in that the unit tests do not adequately represent the behaviour of the application as a whole. Unit tests make assumptions about the way the module will be invoked and the way the data will be presented. A software module may have 100% coverage (meaning every functional point has a corresponding unit test) but the complex interaction of software modules means that the actual operational data may not conform to the expected pattern. Thus, the behaviour of the overall application may be unpredictable and untested.

3. Prove it

Having reviewed how the requirements reflect the regulatory framework and how the code and unit tests implement these business processes, the next step is to collect the evidence that the delivered application performs as expected. This can be achieved in one of two ways:

- Review the inputs, outputs and corresponding reference data from the live system
- Run the application in a controlled environment using a known set of scenarios

Many organisations have taken a sledgehammer approach to this and amassed significant hoards of log data, transactional outputs and related reference data from operational systems with the view that, by recording everything, they will have the evidence they need. But the sheer scale of challenge associated with extracting meaning from this digital landfill can be huge. Specialised software is available to support this analysis but this further increases the cost and complexity of demonstrating compliance.

Where an organisation chooses to run the application in an isolated environment using known data and reference outputs, the identification of anomalies is usually much simpler. Nonetheless, considerable effort is required to keep the test pack and corresponding outputs up to date, establishing the start conditions, running the tests and analysing the results. On top of this, the whole process must be documented.

Even those organisations with mature business processes that have achieved, or aspire to achieve, CMMI level 5¹ or equivalent will find they still have heavyweight processes in place to meet the reporting standards. The cost of maintaining the integrity of these processes in a fast-changing environment is likely to be high.

Traceability

In the event that the application generates anomalous outputs that breach regulatory guidelines, how do we trace back from these outputs through the system itself to the prevailing legislation? It is important to understand where the translation of legislation to result has broken down so that we can be confident of a full and proper impact assessment. The issues highlighted so far apply equally when the analysis is carried out 'bottom up' rather than 'top down'.

¹ Capability Maturity Model Integration provides a mechanism for appraising an organisation's process maturity and guiding process improvement. Process maturity is mapped on a continuum from 1 (low) to 5 (high), with 5 representing a nominal ideal state

Crises are often precipitated by high-impact, low-probability events and these might use code paths that have never previously been exercised. If analysis of the event identifies that the business requirements have not kept pace with changes in the regulatory framework, the impact could be widespread throughout the application. If it is simply the technical implementation of a code change then the weakness could be limited to a small area of functionality and its associated tests.

In order to have confidence in an IT system it must be possible to trace from the outputs produced all the way back to the specific regulation that applies and vice versa. In most cases, this process requires significant human effort and often relies on a sole expert within the organisation who maintains a mental model of the application within his or her head. So, if traditional approaches do not work, what can we do differently that will give us the result we need?

So if traditional approaches do not work, what can we do differently?

Implementing a sustainable regime

There is no single technical solution to the complex and wide-reaching challenges brought about by regulatory compliance. However, use of the right technology can make a significant difference to the effort required to be confident of demonstrating compliance and meeting reporting standards.

To be fully confident that an IT application is, and remains, compliant with the prevailing regulatory framework, the business must be able to:

- Demonstrate the actual behaviour of an application in all foreseeable situations, including the low-probability, high-impact events that have shaped the recent recession.
- Accommodate functional changes quickly and easily whilst ensuring existing functionality remains unaffected.
- Show a direct link between regulatory frameworks and the application processes designed to implement them.
- Trace requirements down to the implementation and vice versa.
- Understand how the application is being used in day-to-day operation and easily extend the test coverage in little used areas.

An IT system that allowed an organisation to meet these obligations, within the system itself and without depending on the external documents, document management systems or experts discussed above would greatly benefit any organisation concerned with demonstrating regulatory or legislative compliance.

The Erudine Behaviour Engine[®] (EBE[®]) is a technology designed to deliver business applications that automatically maintain a direct and tangible link between requirements and functionality throughout the life of the application. The tight coupling of textual requirements, often drawn directly from legislation, to the application behaviour provides a robust and reliable means of demonstrating compliance with the relevant regulations or business requirements. Behaviour is also supported by specific test cases that can be used to exercise the application, providing reassurance to the business that the application will behave in a consistent and predictable way in all foreseen events.

EBE can even be used to model manual processes that complement the IT system. In many cases, the business requirement is only partially satisfied by the IT system and the accompanying human processes are an integral part of the solution. As these processes are typically deterministic (a given situation will always generate the same output), they can be captured in the same model as the core system.

The Erudine Behaviour Engine will not allow a change to be made that will break existing behaviour, thereby protecting the application from inadvertent compliance breaches.

By using case-based reasoning, the Erudine approach guarantees that every aspect of the application will have an associated specific test case that demonstrates the application's behaviour. Put simply, this means the application is built by presenting specific business scenarios and 'teaching' the application how to deal with them. There is no software code generated and no complex language to learn; the business logic is captured as behaviour and presented graphically.

Further, when new functionality is added to the application, the Erudine Behaviour Engine automatically tests the changes against all existing behaviour and will not allow a change to be made that will inadvertently break existing behaviour. This virtually eliminates the need to carry out expensive and time-consuming regression tests of the application. It also protects the application from the inadvertent introduction of behaviour that undermines the application's compliance with the regulatory framework.

Full diagnostic traceability is provided through EBE allowing both developers and auditors to select a specific conclusion and, with a mouse-click, see the business requirement to which it relates.

Conclusion

Regulatory compliance is fundamental to the effective running of a successful business. Regardless of the size of the organisation, failure to demonstrate compliance can lead to punitive fines or even disqualification from their chosen markets. The key here is that compliance must be demonstrable.

Companies often approach this subject from a purely procedural angle, creating and maintaining large volumes of documentation that describe how the outputs correspond to the business requirements and how these relate to the regulatory framework. This approach is expensive, time-consuming and vulnerable to subversion through expedience. Most significantly, this does not effectively record the actual behaviour of the IT applications used to run the business.

Revisiting the earlier question 'what can we do differently', we can take advantage of technologies now available that address many of the weaknesses inherent in code-based solutions. This approach will reduce the total cost of ownership for applications as well as reducing the cost and complexity associated with compliance reporting.

Applications developed using the Erudine Behaviour Engine have significant advantages over those developed in traditional languages, including Java and .NET, because of the strong association between requirements and behaviour. The traceability provided by this tight coupling provides direct and incontrovertible evidence of how the application implements the regulatory framework.

The IT Compliance Survey² likens IT GRC to driving a car, where increased speed of change increases risk to the organisation; a risk that is exacerbated by the organisation's process maturity. To continue this analogy, the Erudine Behaviour Engine provides active safety features such as traction control, ABS and adaptive cruise control to allow the driver to concentrate on direction while reducing the risks associated with high-speed business.

² (1) IT Policy Compliance Group – 2008 Annual Report – IT Governance, Risk and Compliance

http://www.itpolicycompliance.com/research_reports/it_governance/

The traceability provided by this tight coupling provides direct and incontrovertible evidence of how the application implements the regulatory framework.



Hugh Beever
info@erudine.com
+44 (0)8456 123 862
www.erudine.com

Central House
Beckwith Knowle
Otley Road
Harrogate
North Yorkshire
HG3 1UF

