BUSINESS COMMUNITY
RESEARCH REPORT

Data Governance in the Software Lifecycle

Assuring the security of sensitive information

Martin Atherton, Jon Collins and Dale Vile, May 2008

Many organisations have been driving improvements in information management to gain better control over their information assets. While things in this area are not perfect, awareness of the challenges is now high and action is being taken to enhance capability in the areas of compliance, discovery and, not least, data security. But are all the bases adequately covered?

KEY FINDINGS

Despite higher level initiatives, some important activity is falling under the radar

When feedback was gathered from 240 IT and business professionals on the topic of information governance, it was clear that an important area of activity is frequently overlooked. More than 70% of organisations employ data from live systems during the software development lifecycle for testing purposes. Unlike operational areas of the business that are subject to corporate level guidance and scrutiny, however, information governance in this pre-production environment is left largely to IT.

The risks are significant, and understanding them is important

While those running IT departments and development projects are generally very responsible, the environment is actually more risky than it appears at first sight. The people involved in software development and testing are not always employees, activity is often highly distributed across multiple locations, and the IT landscape used to support the development and test cycle is not always separated from live systems. With the best will in the world, there is inherently a lot of scope for things to go wrong, so effective information governance is critical to assuring ongoing security.

Plugging process and automation gaps is key if risks are to be properly managed

The way in which test data is managed is frequently highlighted as an area for improvement, which points directly to process deficiencies in many organisations. Even where processes are in reasonable shape, though, exposures still exist. The majority of respondents in the research alluded to the need for improvements in automation in areas such as test data management, live data sanitisation, and workflow management during the testing process. This suggests a high degree of reliance on manual procedures, which by definition will be prone to error.

Proactive review of current policy and process is recommended in many cases

If you are responsible for running an IT department or development organisation and haven't yet been challenged on how live data is used during the software lifecycle, it's only matter of time before this happens so it is better to prepare proactively. Rather than thinking of this as a burden, however, there is a real opportunity here to secure the support and funding required for making improvements that will deliver much broader benefits. Whether it is more efficient process or investment in better tools to manage software testing, the result is likely to be a smoother and more efficient operation that is both more pleasant to work in and to manage. From the corporate perspective, however, we cannot lose sight of the real imperative for control, which is effective risk management and preventing that accident which is sitting there waiting to happen.

The research upon which this report is based was designed and interpreted on an independent basis by Freeform Dynamics. Feedback was gathered from 240 IT and business professionals during the study, which was sponsored by IBM.

Introduction

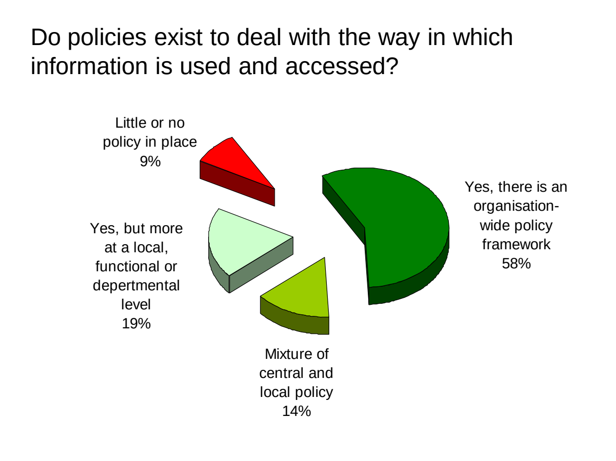
This report is based on the findings of a research study completed in May 2008 to determine how data is used in the software development and testing process, what weaknesses exist, and how these weaknesses can be treated. The study involved in-depth telephone interviews with 240 business and IT decision makers in the three key European economies of France, Germany and the UK. For further information on the research sample please refer to Appendix A.

Background and context

Improving the way business and IT strategy is implemented and executed throughout an organisation in a consistent manner has been a constant topic for discussion since 'governance' became an industry buzzword several years ago. In practical terms, nowhere has this been more important than in improving the way in which business information is treated from an operational, security and overall risk and compliance perspective.

Against this background, previous research [1] has shown that forward-thinking organisations are driving towards implementing information management capabilities to assist them in gaining better control over their information assets. While this is by no means a challenge that every organisation has overcome successfully yet, it is fair to say that most are aware of the limited control they do have and are starting to take action. Information governance, including security and access, is therefore an item that is high on most corporate and IT agendas today.

When we look at the practicalities in this area, one of the considerations is scope of policy, and at first sight, while there is evidence of policy fragmentation in some cases, most tell us that governance frameworks are in place at one level or another (Figure 1).



It has become common for information use and access policy to be defined centrally, underlining how seriously this area is being taken.

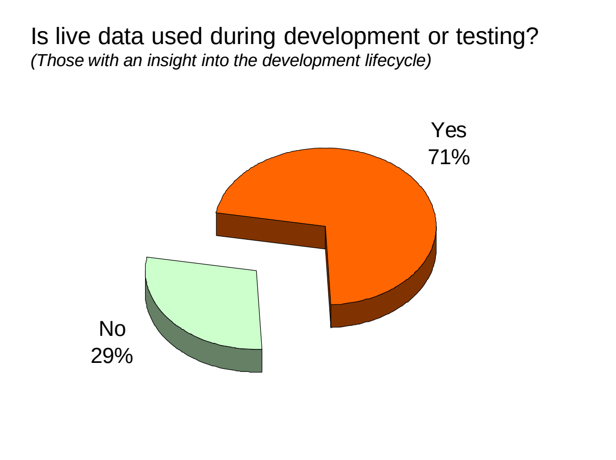
Figure 1

The presence of such policy is clearly important, as it is hard to expect people to treat information appropriately and consistently if they are not given the necessary guidance. Where there is a reliance purely on locally defined policy, however, or when certain areas within the organisation fall outside of 'company-wide' frameworks because they don't immediately seem relevant when considering requirements from an information policy and risk perspective, exposure can occur. This sets the scene for the core theme of this report, which is information governance within the software development lifecycle.

Spotlight on software development

Software development is an area that does not always come under as much scrutiny as elsewhere concerning information management and risk. This is understandable as pre-deployment activities taking place here are somewhat 'hidden from view' in an operational sense, and because there is seemingly little to worry about until a project enters the operational side of the business by 'going live'.

However, the fact that over 70% of those with an insight into the development lifecycle tell us they are using live data in their testing environments suggests a need for many of us to think again about some of our prior assumptions in this area (Figure 2).



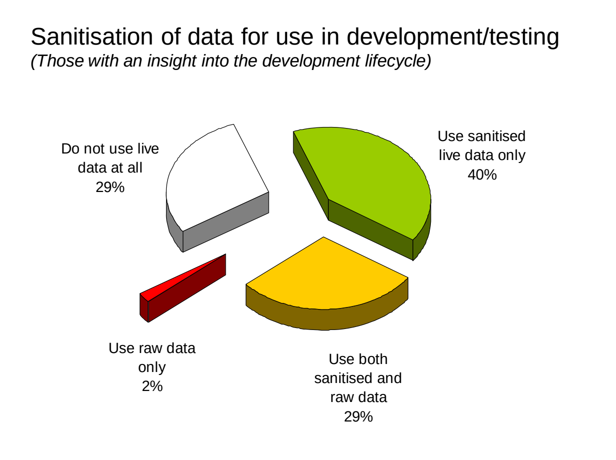
Of the respondents with an insight into the development lifecycle, the majority indicated use of live data during software testing.

Figure 2

The use of live data in this way should not come as a surprise, however. During the research, respondents highlighted a range of reasons to explain why it is often necessary to use data extracts from live systems during the development and testing process. In order of importance, these were:

1. We need data in sufficient quantity to model workloads and performance levels accurately
2. It would not be possible to replicate the live environment without using real data
3. We cannot fulfil our testing obligations without demonstrating success with real data
4. Use of live data saves us a lot of work creating test data from scratch
5. Live data ensures that specific situations are modelled accurately in the data

In terms of data preparation and control, while many take steps to 'sanitise' or 'anonymise' live data before use in development and testing, about a third of respondents indicated use of data straight out of live systems in raw form (Figure 3).

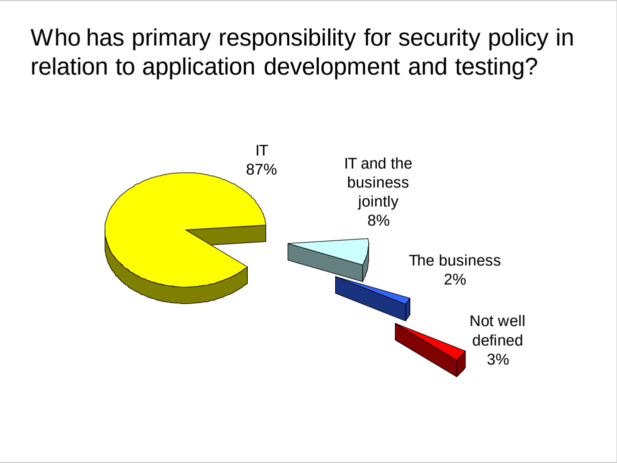


Just under a third of respondents indicated use of data straight out of live systems in raw form, with another 40% routinely sanitising data before use.

Figure 3

Now before we get carried away with the notion that some IT departments are being reckless by using live raw data in this way, they are not necessarily taking risks, as it will depend very much on the data in question. The use of live data will only become an issue if the act of using it contravenes internal or external (industry) guidelines, or creates opportunities for theft or loss.

This, and indeed the question of how the sanitisation of live data takes place from a procedural perspective, brings us to the question of how data security in particular is managed within IT. When we look at this, we find that the IT function is largely left to itself when it comes to defining and implementing security policy in the development and test environment (Figure 4).



The IT function is largely left to figure out and implement its own security policies in relation to application development and testing.

Figure 4

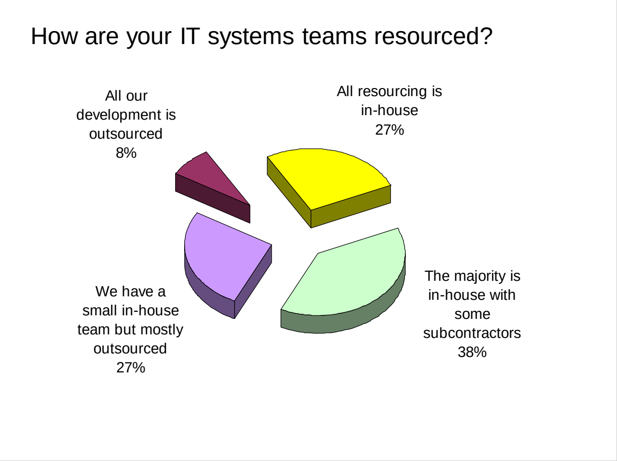
Some would argue that this in itself is a cause for concern given that there is a business risk associated with using live data in any context, so accountability is important. Beyond accountability, however, there are also other factors which need to be considered when assessing the overall risk.

Understanding the risks

There are a number of areas specific to the software development cycle which contribute towards the risk of information governance related policies and standards being breached:

- Who is involved in software development and testing?
- How is testing activity spread geographically?
- How well separated are test and live environments?

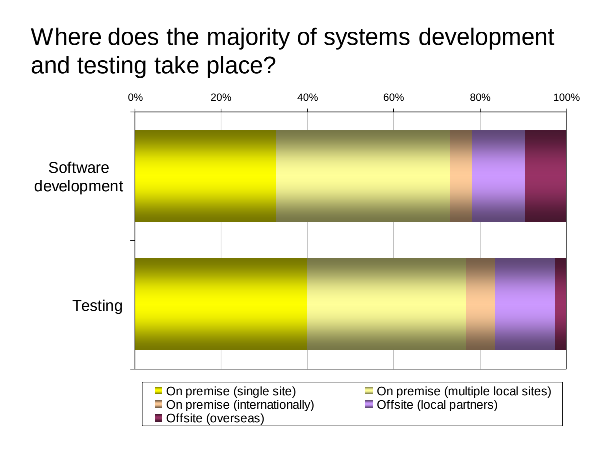
When it comes to resourcing IT teams, it is clear that consideration must be given to external as well as internal staff (Figure 5).



The majority of IT teams in modern business are comprised of a mixture of internal and external staff.

Figure 5

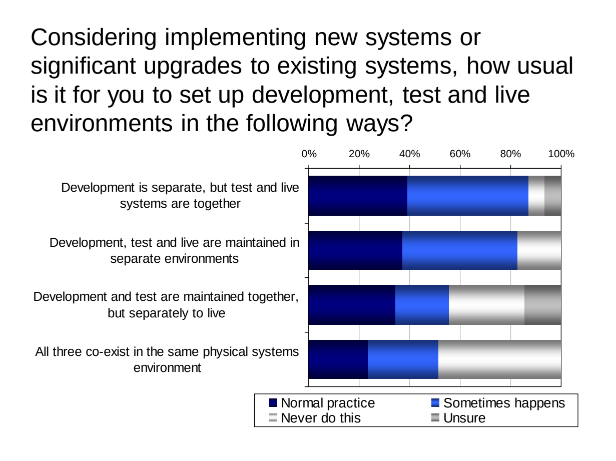
The physical location and distribution of development and testing activity adds another interesting dimension to the consideration of lifecycle security (Figure 6), especially if a project leader has less control over the policies and procedures employed outside their own physical location.



Software development is a cross discipline, cross firewall and sometimes even a cross continental affair.

Figure 6

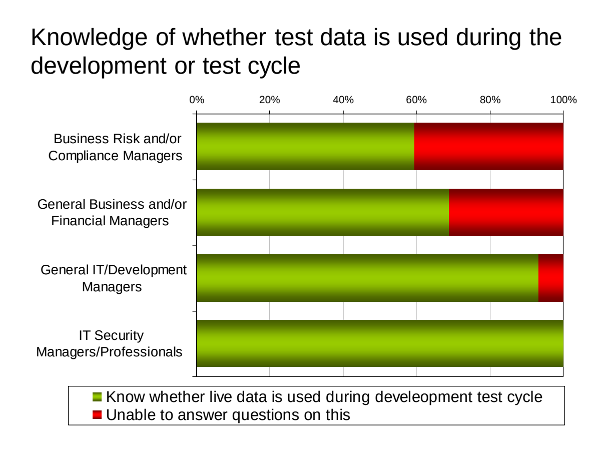
Then there is the question of how much development, test and live environments are separated during the lifecycle. Overall, while around 50% of organisations never mix live and test environments, there is a fair degree of mixing going on in the other 50% (Figure 7).



There is no hard and fast rule as to how live and test environments are treated – and it can vary from project to project.

Figure 7

And highlighting the issue of visibility and accountability, it is telling how often those responsible for overall business risk and/or compliance are not even aware of the fact that live data is used in the testing environment, let alone how it is being managed and access controlled (Figure 8).



Business respondents are far less likely to know when live data is used in development or test, than IT respondents.

Figure 8

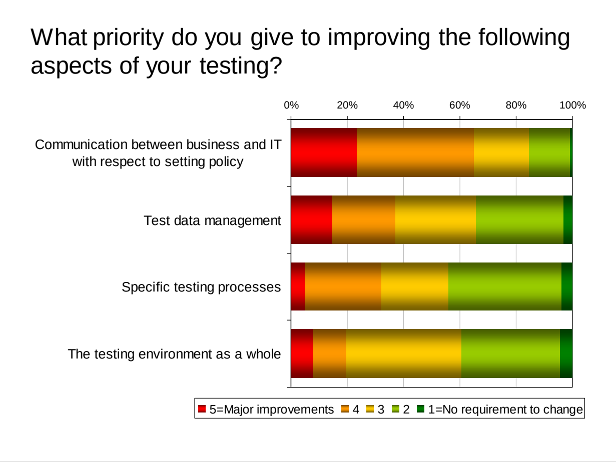
So, what we see is a situation in which IT departments could be in the position of having to use live data in the development lifecycle, while controlling access by both internal and external staff, with project teams spread across multiple sites, working on development and test systems that may or may not be co-located with live systems. Even if only one or two of these risk factors apply in any given instance, the scope for mistakes and oversights alone, without even considering deliberate or malicious breaches, creates a significant potential exposure from a data security perspective. And yet IT is rarely challenged on how this exposure is managed.

So how real is this risk?

To determine the extent of the problem we can look at the level of confidence that exists in how things are currently controlled, paying particular attention to where the need for improvement is highlighted.

Assessing the level of exposure

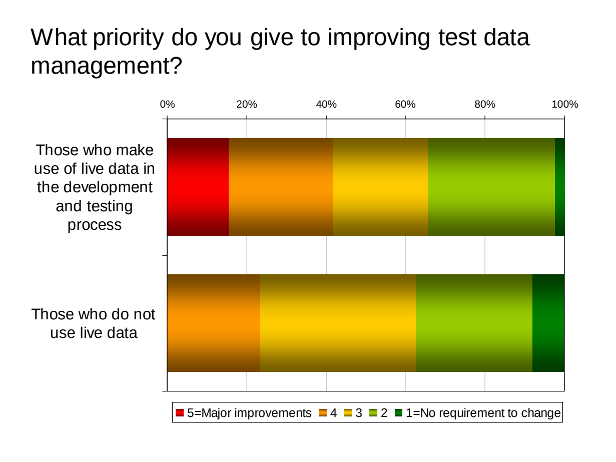
Focusing in on improvements that organisations seek in their testing environments, we can see two very interesting 'headline' findings. The majority of organisations see communication between IT and the business as a major area for improvement, and also acknowledge that their test data management needs improving (Figure 9).



The overall testing environment is certainly not seen as perfect, but it is the areas of test data management and communication between business and IT which give most cause for concern.

Figure 9

For a clearer indication of the dangers that have arisen in some organisations, we can consider how the responses stack up between organisations that are using live data in development and testing, compared to those who are not (Figure 10).

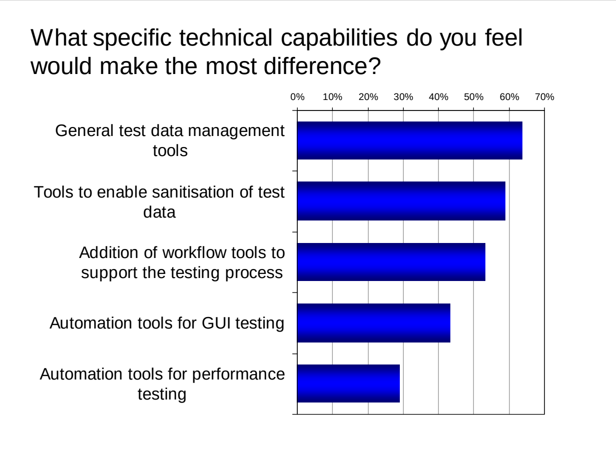


Those organisations using live data in their test environments are more sensitive to the need for improving their test data management capabilities – this is an appropriate reflection of awareness.

Figure 10

It is those with ‘more to lose’ (i.e. those using live data) that acknowledge the need for most improvement. This is no doubt in part due to a heightened awareness of the issues surrounding the appropriate use and protection of live data. However, if an organisation uses live data in its testing activities and considers that it should make significant improvements to its testing in general and improving the management of test data specifically, this would suggest current provisions are inadequate, i.e. highlights a significant real exposure from a process perspective.

Beyond the adequacy of processes, we have the question of technical capability, and it is really quite telling that improvements in some pretty fundamental areas are highlighted, most notably tools support in relation to general test data management (Figure 11).


What specific technical capabilities do you feel would make the most difference?

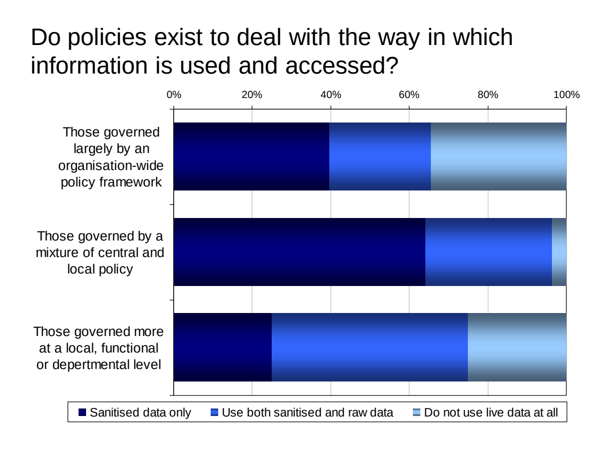
There is strong acknowledgement that the testing environment can benefit from improvements in technical capabilities

Figure 11

Also noteworthy are the gaps highlighted in relation to tools to assist with test data sanitisation and the management of workflow during testing. This overall picture, in fact, suggests that IT departments are generally underserved in terms of automation, which in turn means a heavy reliance on error-prone manual procedures.

Managing the risks

Of course one way of managing the information related risk in the software development lifecycle is to cease using data from live systems. As we have seen, however, live data extracts are used for very good reasons. In fact this is to be encouraged as part of testing best practice to ensure that system performance and behaviour is assessed running against information that mimics the real world as much as possible. With this in mind, it is interesting to see that organisations where good local policy and process is implemented in conjunction with higher level frameworks appear to be making broader and safer use of live data for testing (Figure 12).


Do policies exist to deal with the way in which information is used and accessed?

A combination of central and local policy guidance drives up the safest way of using live data.

Figure 12

This picture makes absolute sense, as a combination of central and local policy provides the flexibility required to deal with the needs of individual projects, while preserving the principle of accountability. A big advantage with this approach is that decisions may be taken on an informed basis at a local level, rather than leaving things to some distant, central body that may apply corporate policy blindly and create unnecessary constraints.

In contrast to this, as we can see, where corporate frameworks predominate and local controls are absent or weak, IT departments can sometimes be inhibited from taking advantage of live extracts, which is not in the best interests of the organisation. At the other extreme, where activity is governed purely by local policy, the picture is very mixed, and the outcome less predictable.

However the above picture is interpreted, though, the lesson is that one of the ways in which the information related risk in the development and test cycle can be mitigated is by ensuring that the policies and processes used to control things in the IT department are consistent with, and supportive of, any broader information governance framework that is already in place. This means IT management understanding the higher level objectives and imperatives as they relate to information risk, but also making sure that those in the business with responsibility for information related governance and compliance have an awareness and appreciation of how live data is used pre-production.

At the next level down, drilling into what goes on in the development process itself, indications are that improved testing processes also reduce the risk, as measured by the amount of data sanitisation carried out. This is consistent with other findings suggesting that organisations using sanitised live data during testing (the optimum situation) are more likely to have a single set of testing processes that are stringently applied.

The lesson here is to make sure procedures are documented and enforced at the appropriate level of detail to cover how live information is used and managed during the development process. While doing this, particular attention needs to be paid to how the sensitivity of data from live extracts is assessed (live data is sensitive, some is not) and therefore when sanitisation needs to take place before data is make accessible to developers and testers. An implicit part of this is dealing with the question of who is responsible for ensuring proper sanitisation when necessary and how such sanitisation is carried out and checked for effectiveness.

This brings us to the last major imperative from a risk management perspective, which is to minimise the reliance on error prone manual processes wherever possible, and given the nature of the automation gap we have seen, the importance of this should not be underestimated. If development staff themselves are responsible for sanitisation using relatively crude techniques such as tracking down sensitive data in databases and ‘blanking’ it using interactive SQL or ad hoc developed scripts, then things will be overlooked and security exposures will result. Use of appropriate tooling for defining, planning and executing the sanitisation process in a structured and automated manner will reduce the risks considerably, as will broader test data management and workflow solutions that lead to more visibility and predictability.

In a companion paper [2], we provide a number of more detailed actionable guidelines. These start with building an understanding of what information is in use in test and development, then move forward from there.

Conclusion

Over the past decade, we have seen a sea change in how risk is considered. Organisations have moved from thinking about data security to information governance. However good we get, there will always be room for improvement: after all, this is about a journey, not a destination. Equally, there will always be more to do: as risks are better understood and mitigated, new areas of risk are uncovered.

One such area concerns how data is used in software development and testing. From this study we have seen a number of factors conspiring to create unnecessary levels of risk, not only in how the IT department shoulders the responsibility, with little business input, but also the fundamental challenges of how applications are tested and made ready for live deployment.

The question of whether or not there are specific problems associated with using live data extracts in the test environment does not generate a black and white answer. It is a highly subjective area, but the research findings suggest that there is significant scope for mis-hap, and therefore an opportunity for improvement and risk mitigation.

To deal with the risks, there are a number of points to consider, and a number of areas that any organisation may be keen to explore. These include aspects such as policy, process and tooling, all of which need to be considered as a whole in order to respond to the risks that clearly exist.

As a final comment, addressing this space has highlighted that using live data in a test environment might not automatically mean danger. However, for legislative as well as general security reasons, organisations would be wise to ensure they know exactly where they stand. Ignorance is not an option.

References

[1] “Information Governance: The keystone of a sustainable business and IT strategy“, Martin Atherton and Jon Collins, February 2008

[2] “Data Governance in Software Testing: A best practice primer”, Jon Collins, May 2008

Appendix A

RESEARCH SAMPLE

The research sample was comprised of 240 respondents, distributed as shown in the figures below.

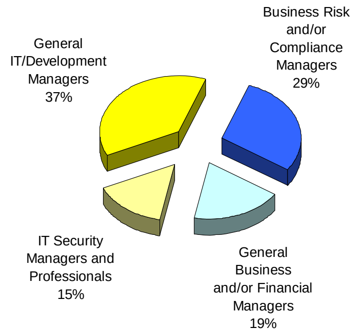

Participants by Role

Figure 13

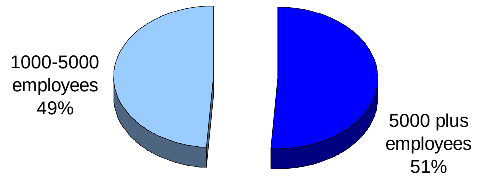

Participants by Organisation Size

Figure 14

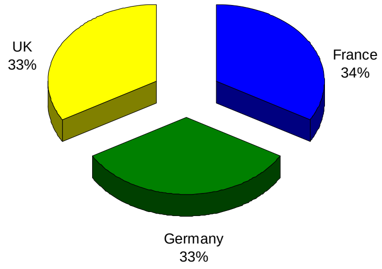

Participants by Country

Figure 15

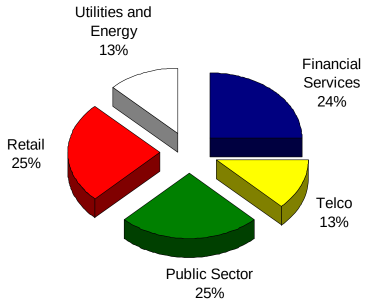

Participants by Industry Sector

Figure 16

About Freeform Dynamics

Freeform Dynamics is a research and analysis firm. We track and report on the business impact of developments in the IT and communications sectors.

As part of this, we use an innovative research methodology to gather feedback directly from those involved in ITC strategy, planning, procurement and implementation. Our output is therefore grounded in real-world practicality for use by mainstream IT professionals.

For further information or to subscribe to the Freeform Dynamics free research service, please visit www.freeformdynamics.com or contact us via info@freeformdynamics.com.

About IBM

At IBM, we strive to lead in the invention, development and manufacture of the industry's most advanced information technologies, including computer systems, software, storage systems and microelectronics.

We translate these advanced technologies into value for our customers through our professional solutions, services and consulting businesses worldwide.

For more information on IBM, please visit www.ibm.com.

Terms of Use

This report is provided for your general information and use only. Neither Freeform Dynamics Ltd nor any third parties provide any warranty or guarantee as to the suitability of the information provided within it for any particular purpose.

Data Governance in Software Testing

A Best Practice Primer

By Jon Collins, May 2008

In association with:

Introduction

In the research report "Data Governance in the Software Lifecycle" [1] we consider a number of areas of risk that can impact software development in general, and software testing in particular. The areas cover:

- Sourcing – how development teams are resourced, and with whom
- Geography – where developers reside, and the level of distribution within teams
- Environment – how the development, test and live systems environment is structured
- Data – what information is used in development and how it is managed

Each of these areas brings with it a set of specific risks which may need to be treated in some way, depending on the context. In this short guide we concern ourselves specifically with risks to do with the data used in software testing, and how they can be mitigated.

Much of what we lay out here may be considered to be basic common sense, but we make no apologies for this as our research indicates that some may not be fully aware of the issues and others need reminding. However much you feel you have this area under control, we encourage you to run through the guidelines and actions section on page 3 which has been designed to be used as a checklist.

Assessing the data related risk

There are several kinds of data that may be involved in the software testing process, including:

- Test scripts and guidelines
- Test results and metrics
- Configuration data
- Application data

When we look at this list, we may wonder what some of these have to do with security, so it is worth reviewing the potential areas of exposure.

The application being developed may give the business a competitive edge, for example, and test information taken as a whole may provide an indication of where that edge lies. In such cases, it may therefore be necessary to keep the very existence of the application under wraps, and therefore the entire test environment should be secured. Whether the application being developed is to support a new service or product line, or to enable entry into a new geographic market, the chances are that alerting competitors in advance of a launch via leaks or indiscretion within the internal development organisation or a systems integrator can have a tangible negative impact on the business.

Beyond this, our research has shown [1] that the majority of organisations use data extracts from live systems in one form or another for testing purposes. While some of this data may be quite innocuous, other information might be extremely commercially sensitive and/or subject to legal or other regulatory constraints. Ensuring that such data is either appropriately sanitised or only accessible to those with the clearance to view it, is a fundamental part of security assurance within the software development and testing lifecycle.

Given the wide range of sensitivity that exists, a good first step is to classify each type of data according to its value and level of risk. There are many ways of doing this in practice, and classification systems in themselves can get quite complex. In order to facilitate effective communication across teams, however, we tend to recommend keeping things as simple as possible. Even a simple traffic light system may be adequate in many cases, such as the following:

- RED: Company proprietary or confidential, to be treated stringently
- AMBER: There are specific constraints on the data that need to be considered
- GREEN: There are no specific requirements to protect this data

Coming back to the nature of the risks themselves, it is then useful to firm up on the question of what exactly we are looking to protect data from. While the list of threats and exposures and the weightings given to each will vary by organisation (and even by project), here are some things to consider:

- Abuse of intellectual property
- Knowledge of system/process vulnerabilities falling into the wrong hands
- Engineering of vulnerabilities into applications for later exploitation
- Sabotage for personal or political reasons, and/or financial gain
- Unauthorised access to live systems or data
- Legal and compliance implications of live data use

This list is by no means exhaustive, but it is a useful starting point to identify where time and effort should be spent when considering the nature and level of exposure. Together with identifying the risk, of course, it is good practice to document the likelihood and potential impact of each.

As an approach, we would suggest using a format such as the following, though again, this is for illustration only as there may well be mechanisms already in place, either within IT or in more corporate level risk management systems that could be utilised:

Data Type	Classification	Instance and location	Risk and impact

Armed with this type of information, however it is captured, we can move on to how some of the risks can be treated or mitigated. In some situations, it might be appropriate to start with the question of whether the data in use is strictly necessary, but assuming it is, the principles of classification, risk assessment and access control will apply.

It is beyond the scope of this document to provide exhaustive advice on the appropriate action to be taken in each individual instance, but please see the guide on page 3 for a high level steer in the right direction.

References

[1] “Data Governance in the Software Lifecycle”, Atherton, Collins and Vile, May 2008

Guidelines and Actions for Data Governance in Software Testing

Here we consider some general guidelines to mitigate data risks in software testing, derived from the report mentioned above [1], together with the specific actions that result.

Guideline	Rationale	Key Actions
Get up to speed on relevant legislation and compliance criteria.	Information about the test environment and live data used in the testing process may be subject to a number of standards and norms, which may in turn be legally binding. As a spin-off benefit, the compliance posture of the organisation may be enhanced.	• Build a picture of what regulations and standards apply to the data in use in the test environment. • Where snapshots of live application data are being used in testing, check the applicable regulations. • Use the checklist of test information (below) to cross-reference whether the list of applicable regulations and standards is complete.
Build a picture of all data at use in the test environment, both current and historical	When identifying test-related data in play, it is too easy to look only at material currently in use. However, some historical information may be still accessible, either in an archive or potentially just sitting in the file system (or indeed a filing cabinet).	• Create an inventory of the different types of test-related information in use, and assess it using the traffic light system described above. • Decide on the risks, likelihoods and impacts for each element of test information, taking into account the regulations and standards.
Employ a mix of centralised and local policies for data handling.	Where only central policy is involved, it can be quite separate from the 'coal face' and hence is sometimes seen as irrelevant or incomplete. Local policies should complement central policies to ensure completeness of coverage.	• Review central policies to ensure they cover the needs of the testing environment • Create a local testing policy document to cover the specific requirements of local testing. • Assign responsibility for local policy implementation and review. • Gain buy-in from both testers and management to maximise the effectiveness of policy implementation.
Review test processes for risk mitigation and consistency with standards/regulations.	Test processes should take into account not just testing completeness and consistency, but also the security of test information. A weak process puts data at risk.	• Collate a complete set of test processes for the testing at hand. • Verify that testing processes are being followed consistently (if not, that's a security risk in itself). • Recommend and apply improvements to test processes where they prove to be inadequate.
Define appropriate roles to oversee the testing process from a security perspective.	Security implementation in general, and in the test environment in particular, works best when it is being treated as a priority by the right level of seniority within the organisation. Business involvement is particularly effective.	• Review existing security roles and responsibilities and identify where the role for test data management would best fit. • Check compliance and standards for specific requirements to incorporate responsibilities into other roles in the organisation as necessary.
Deploy appropriate processes and tools for mitigation of test data risks.	Certain types of data, for example customer data, may be at risk (or even illegal) if used in their raw state in the test environment.	• Identify if and where live application data is being used that may pose a risk when used in the test environment • Review test data management processes (where these exist) to minimise the use of such data and ensure it is correctly managed. • Consider tools to assist with the sanitisation of such data to further reduce live data risks.
Destroy unnecessary or obsolete data.	Data may be retained unnecessarily, creating a management overhead and an ongoing risk.	• Take a careful look at the data being stored. Check it is still applicable, and indeed usable. If not, then delete or otherwise destroy it. • If in doubt, quarantine it – i.e. collate it and store it in a secure place.

About Freeform Dynamics

Freeform Dynamics is a research and analysis firm. We track and report on the business impact of developments in the IT and communications sectors.

As part of this, we use an innovative research methodology to gather feedback directly from those involved in IT strategy, planning, procurement and implementation. Our output is therefore grounded in real-world practicality for use by mainstream business and IT professionals.

For further information or to subscribe to the Freeform Dynamics free research service, please visit www.freeformdynamics.com or contact us via info@freeformdynamics.com.

Terms of Use

This document is provided for your general information and use only. Neither Freeform Dynamics Ltd nor any third parties provide any warranty or guarantee as to the suitability of the information provided within it for any particular purpose.