

Risk and Resilience

The application availability gamble

Dale Vile and Jon Collins, Freeform Dynamics Ltd, June 2008

The modern business is highly dependent on IT. When systems go down, the disruption can be widely felt, and even lead to tangible damage to the business or its brand. Against this background, it doesn't make sense to gamble with systems availability. So why do so many take risks?

KEY FINDINGS

Systems failures occur frequently and impact the business in multiple ways

When more than 1,200 IT professionals were asked about the frequency with which IT systems failures impact their businesses, more than half (57%) alluded to disruptions occurring on at least a monthly basis. The end result is a direct hit on business productivity, increased IT overhead and knock on effects as delays impact processes, schedules and plans. Beyond this general disruption, one in five organisations suffers brand damage or tangible financial loss on at least a quarterly basis.

Application availability hotspots differ by organisation size

Larger enterprises are more inclined to identify core business applications as an availability hotspot, as highly integrated in-house developed systems and heavily customised software packages create a complex landscape with many potential points of failure. Small and medium-sized organisations call out horizontal applications such as email as being particularly troublesome from an availability perspective, as a result of rapid growth in demand and underinvestment in platforms.

Lack of resiliency planning often leads organisations to gamble on availability

Much of the exposure leading to high failure rates comes about because system availability is only considered towards the end of the project lifecycle. This often results in having to choose the lesser of two evils: either slipping delivery times to retrofit resiliency measures, or taking the gamble and putting the system live with vulnerabilities. Even if the will is there to do the right thing, unfortunately the money may not be, as the cost of implementing resiliency will not have been budgeted.

Dealing with the challenges requires a balanced approach

Whether it's poor planning or simply a lack of appreciation of the need to invest, in most organisations, a significant gap exists between the resiliency measures the business requires and those that are actually in place. Issues range from the fundamental such as inadequate controls during the application lifecycle leading to software that isn't 'operations ready', to simple things like the absence of failover solutions for key applications or the lack of effective monitoring to pre-empt potential failures. While the research suggests that addressing such issues individually will pay back significantly, the real aim has to be incorporating resilience and availability into all aspects of IT.

But don't try to boil the ocean, start with the simple stuff

An obvious step to take, if you have not already done so, is to involve IT operations staff early in the project lifecycle. This will highlight resiliency requirements and allow dependencies and conflicts with the existing infrastructure to be understood up front so plans and budgets can be set appropriately. Addressing some of the hotspots identified above is also a good move. Simply stabilising an email or collaboration system, for example, will be a step in the right direction, freeing up resources and getting the business to appreciate the value of uptime, which is a great foundation to lay for the future.

The research on which this report is based was designed, executed and interpreted independently by Freeform Dynamics. Feedback was gathered via an online survey (1223 respondents, predominantly IT professionals from the UK, USA, and other geographies). The study was sponsored by Neverfail.



neverfail
WWW.NEVERFAILGROUP.COM
PREDICT - PROTECT - PERFORM

Introduction

Previous research has told us in no uncertain terms that information technology is generally front of mind when organisations think about risks to their business. Indeed, "downtime of IT systems" is second on the list of factors most frequently considered when organisations are going through formal risk planning, significantly ahead of considerations such as regulatory exposure, criminal activity and terrorism^[1].

This is not surprising. Businesses today, large and small, are highly dependent on IT systems for their day-to-day operations. When these systems go down, users are inconvenienced, the business is often interrupted, and in extreme cases, the organisation can suffer financial loss or tangible damage to its brand and reputation.

Yet despite this acknowledged reliance on IT, high-profile systems failures still hit the headlines, and closer to home, most of us have all too frequently experienced the consequences of downtime first hand. This may be directly in our own working environment, or indirectly when a supplier, for example, is unable to meet our needs because their call centre system is not available, their email system is down or some other IT related problem has occurred.

Given that things are clearly not perfect, the research study reported in this document was conducted to investigate why systems fail and, more importantly, what can be done about it.

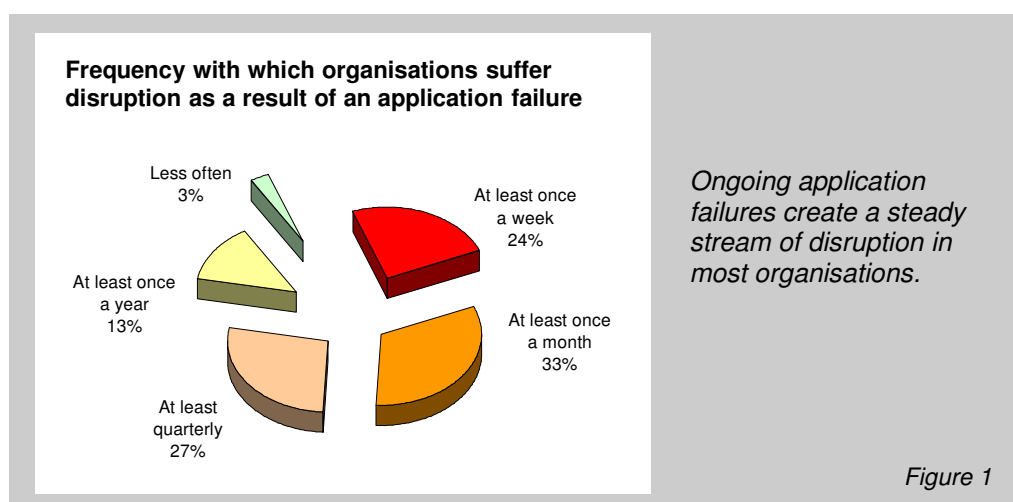
Overview of research methodology

In order to investigate this area to the level of depth required to understand the factors affecting the availability of IT systems, it was necessary to gather input from those most intimately involved in implementing and running those systems: namely, IT professionals working in a mainstream business environment. To this end, an online survey was designed and executed via the Web. This allowed us to gather information regarding the degree to which failures occur within IT, and the factors, both positive and negative, that have the most impact on performance in this area.

The study itself was completed in April 2008 and approximately 1,200 responses were gathered from organisations of all sizes across the UK, USA and other geographies. More details of the research sample are presented in Appendix A.

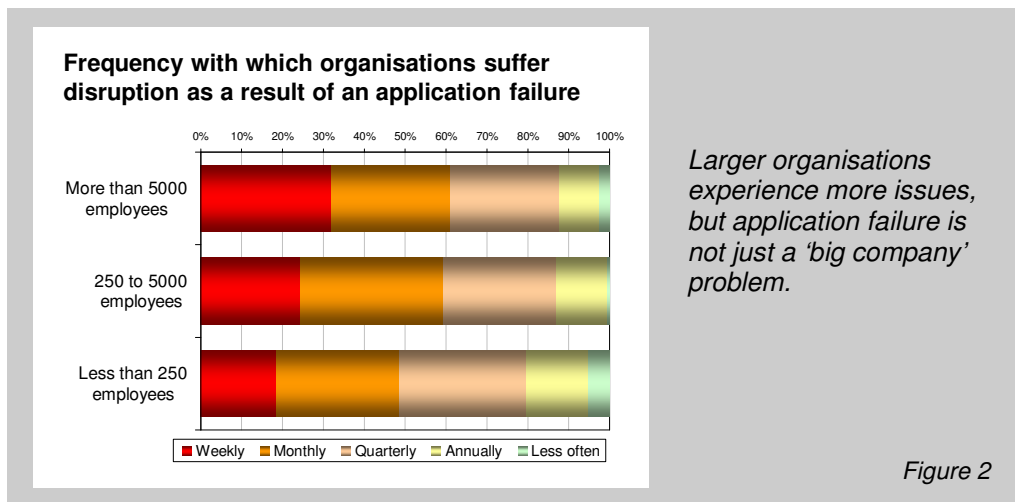
The extent of the problem

Before we get into this topic in depth, it is useful to focus our minds by looking at just how often the business community is affected by application failure (Figure 1).



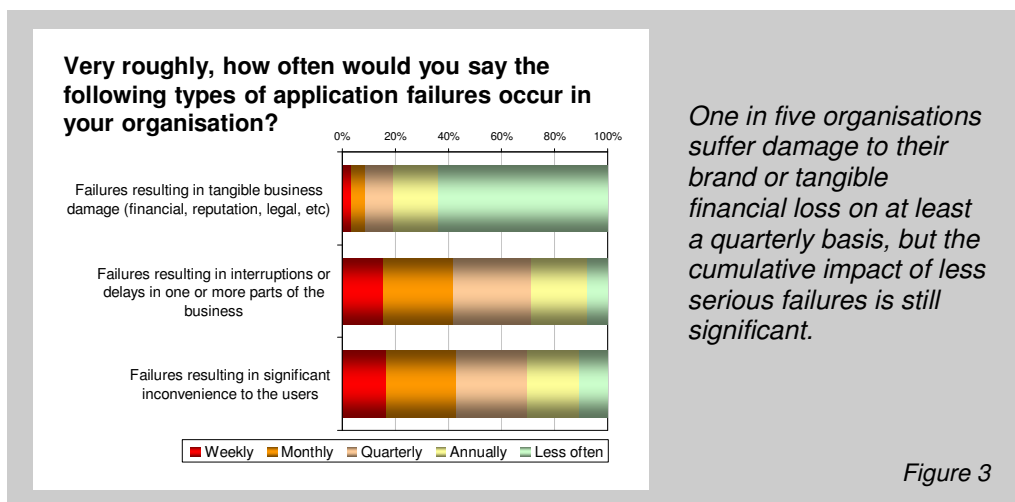
In more than half (57%) of the organisations who participated in the research, we can see from this chart that hardly a month goes by without an IT application failure of some kind causing user inconvenience or disruption within the business.

Larger organisations tend to have a higher incidence of failure on average, no doubt because there is simply more to go wrong in larger more complex environments (Figure 2).



Looking at this chart, though, it is important to note that IT application failure is still a very significant problem for small and medium-sized businesses. It would, therefore, be inappropriate to think of this as a 'big company' issue. The email system going down in a 50-person organisation can be as crippling as a call centre system failure in a large telecommunications company. In fact, smaller enterprises are often more sensitive to business interruptions, where a few hours of downtime can, in some cases, mean the difference between healthy cash flow and the wolf appearing at the door.

Of course not every application failure is catastrophic, so if we want to understand the extent of the problem, it is important to look at the frequency of incidents according to their impact. This kind of analysis tells us that while the majority of failures simply result in a degree of user inconvenience and disruption, incidents with more serious consequences occur more frequently than we might imagine. It is quite an eye-opener, for example, that one in five organisations confess to suffering tangible business damage from IT application failure on at least a quarterly basis (Figure 3).

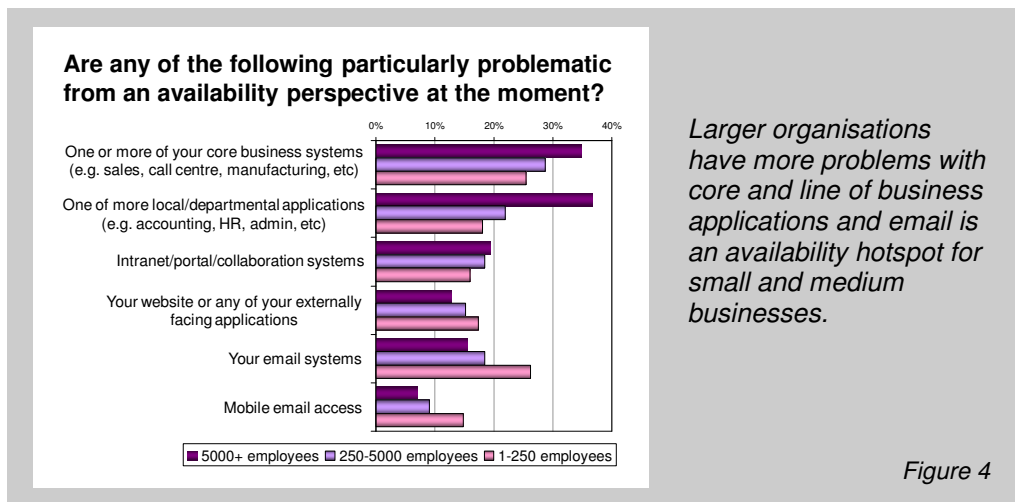


But let's not lose sight of the fact that less critical failures still represent a significant problem. The cumulative impact of frequent downtime incidents include:

- A direct hit on end user and business productivity
- Consumption of a disproportionate amount of IT time and resource to troubleshoot and fix
- Knock on effects in both business and IT as delays disrupt schedules, work patterns and plans
- Undue stress on the relationship between IT and the business

Systems failure hotspots

So far, we have been discussing systems failures in quite an abstract manner, so let's get more specific and look at where the hotspots are in terms of different application types (Figure 4).



Not surprisingly, we see that larger organisations with greater than 5,000 employees are more likely to be experiencing issues with core and departmental applications than with anything else. Custom developed software, heavily customised software packages and a generally more complex environment from an integration perspective introduce more points of failure, and thus more failures occur. On the other hand, larger organisations appear to have fewer issues with horizontal applications such as email and collaboration, which tend to be heavily standardised at a software level, and run on a standard infrastructure that is fit for purpose and maintained via well-defined operational processes by specialist operators. These are, of course, generalisations, but on the whole, the pain in larger enterprises is centred on core and line of business applications.

By contrast, while smaller organisations are still experiencing problems with core business systems, albeit to a lesser degree than their larger counterparts, they encounter dramatically fewer availability issues with departmental systems such as accounting, HR, administration and so on. This is undoubtedly because for such requirements, they are more inclined to make use of software packages with minimal customisation running on discrete dedicated hardware that has been chosen for the job – a configuration that is inherently quite stable.

This begs the question of why smaller businesses appear to be struggling significantly more with the availability of email systems, but the following anecdotes from respondents in the study go some way towards shedding light on this:

"MS Exchange plus shared data plus users' home directories plus print spool all on one server, with disc space issues. I warned that this was a big problem and that we needed to separate onto three servers, but I was the new guy and seen as wanting toys, till a massive print job ran the drive of the server out of space. It took 20 hours of straight work to fix, with no email/data for 24 hours."

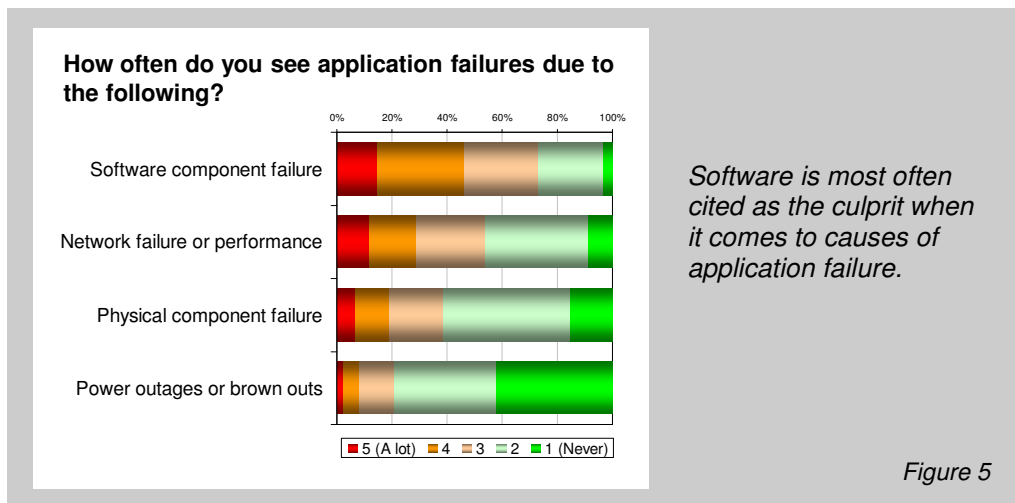
"Storage systems for the email system were running near capacity. Sent an urgent mail up to my superior, was told it would be 'discussed next week'. Two weeks later, the capacity ran out..."

The truth is that email nowadays is typically one of the most dynamic applications in an organisation's IT landscape from a resource and growth perspective, but its importance to a small business is often underestimated from a systems perspective. As a result, it isn't afforded the same status as other applications, so investment in underlying platform technology is often lacking. We'll look at this in more detail later, but for the time being it is worth noting that email is an availability hotspot in the smaller environment.

So, having identified how much systems fail and the kind of applications that create the most issues in different circumstances, let's explore what's behind some of those failures and look at common sources of exposure.

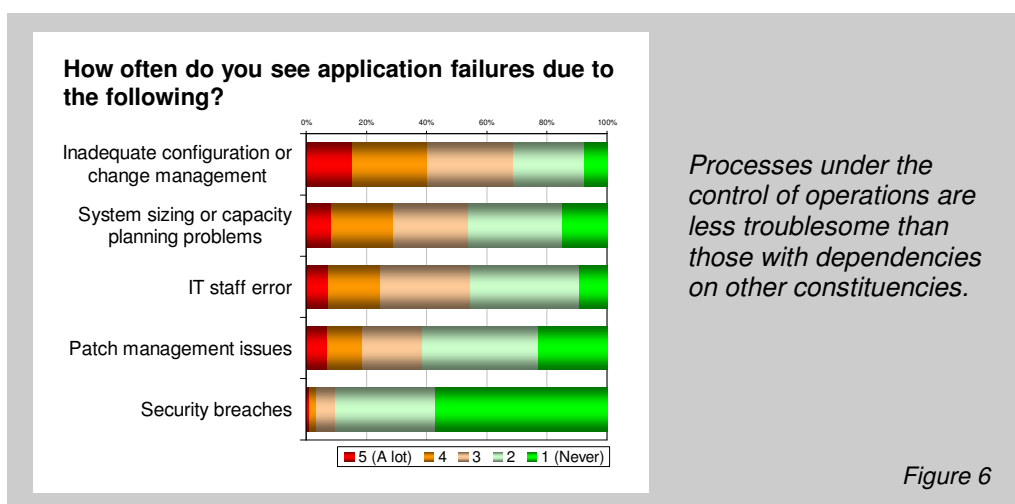
Sources of exposure

If we consider that IT systems are enabled by four core elements – hardware, software, networking and power – it is interesting to see that software is most often cited as the culprit when it comes to causes of application failure (Figure 5).



In many ways, this chart is quite encouraging. It tells us that through a combination of hardware today being inherently more robust than it was a few years ago, and organisations generally having taken sensible measures to protect against obvious vulnerabilities such as power outages, failures at a platform level are less of an issue. Networking is more troublesome, and with the dependency of modern software applications on distributed and remote processing, and the increased use of multi-media, whether in the form of email attachments or real-time voice and video traffic, the relentless increase in pressure on the network infrastructure can be difficult to keep up with.

However, we have to be careful about considering the elements that make up an IT system in isolation, as the way a system is designed, constructed, evolved and operated at an overall level has at least as much bearing on the resulting stability, if not more. In fact, we can see some examples of this if we look at causes of application failure from more of a process perspective (Figure 6).



The most interesting observation from this chart is how processes that are under the control of operations staff, such as security and patch management, are far less troublesome than processes such as change management and system sizing where operations professionals are dependent on other constituencies. There is little that can be done, by even the most competent operations team, to ensure good systems availability if the software they are expected to run is inherently unstable, for example. And if the business analysts have forecast demand at one level, and the architects have

designed the platform to scale in line with this, there is again little that operations can do if the actual load on the system is several times more than it was constructed to cope with.

In line with this observation, we received a lot of freeform feedback from operations staff, in particular, articulating a degree of frustration. Whereas they are the ones in the firing line when systems fail, and the ones dragged out of bed in the middle of the night to put things right, the truth is that the issues and risks to do with systems failure are more often than not created as a result of factors and constraints over which they have little or no control. Here are some representative comments on the topic that provide warnings at a general level:

"Management have to come to understand that there are more important things than how quickly the code can be got out the door. Code that is rushed through development and rollout is quite often of a lower standard, and hence the system will inevitably be less reliable in the long-term."

"I have seen endless long-term problems caused by Ops being forced to shoehorn a badly-fitting application into existing infrastructure, simply because it is 'too late' in the project to properly fix it."

"Managers too often don't properly understand the technology and methods available to improve availability, but by the time any technical staff are involved, the budget is already allocated and the equipment / software bought."

And here are a couple of anecdotes that illustrate the consequences if such warnings are ignored in relation to new systems:

"A major new application was being rolled out nationwide (~5000 users). The Head of IT permitted the project to skip load/stress testing, based on the project manager's personal belief that it would be fine, and against the strong objections of the Ops staff who could see that it was a pig. Twenty minutes after launch, it ground to a halt, and stayed that way. Performance testing suddenly became a high priority :). Three months later, after all the problems stress testing found were fixed, the app re-launched, with virtually no issues."

*"A new call centre customer service system was being implemented. This was *Very Important* (according to management) yet our (IT) requests for adequate infrastructure and stress testing were brushed aside. We warned people the system was unproven in a real situation and could either be very slow or even fail. 'Bah!' was the typical response. Sure enough, Day One arrived, everyone logged on and started working at once, and the network infrastructure promptly went into meltdown. Of course, IT was blamed but happily all our warnings were meticulously documented. It was a case of 'we told you so' but you can guess who had to clean up the mess anyway."*

But it is not just new systems we need to think about in terms of resiliency. Here are some anecdotes illustrating how existing systems can fail if growth and change are not taken care of properly:

"For three years we submitted a hardware refresh project for a collaboration system. Each year, the project got pushed off due to 'new functionality' projects taking priority. The refresh only became critical once the environment completely and totally ran out of storage space and impacted supplier collaboration."

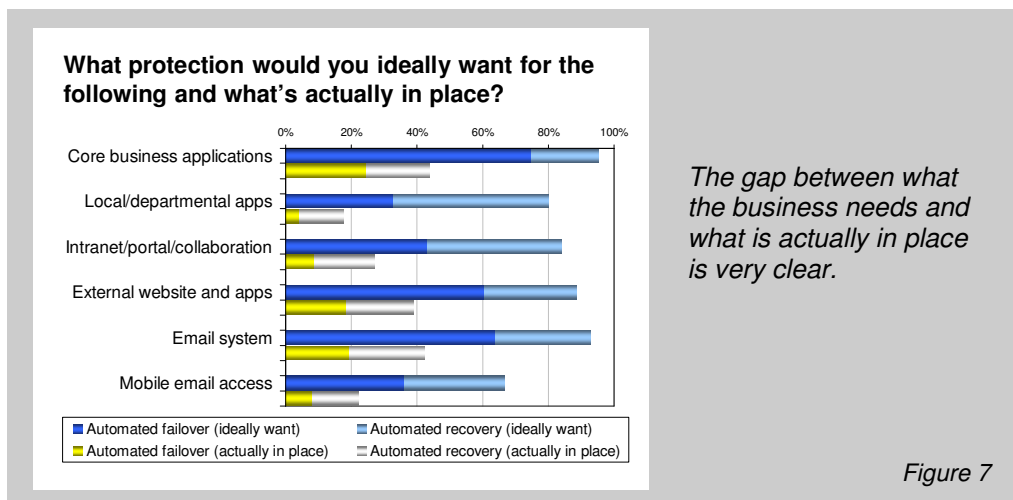
"Following the takeover of a competitor, I advised that one of our SAN's was going to be full in 1 month, 3 weeks, 2 weeks, 1 week, 3 days, 2 days, 1 day. It's going to fail tod... Oh look. It's knackered."

"I presented a proposal to upper management regarding our existing email server, how it was old (6+ years) and already unstable, and how it required immediate replacement to prevent an 'email disaster.' The server died in the middle of the meeting."

The above are representative of literally hundreds of comments that were provided in the same vein. We are hearing time and time again that many organisations are routinely taking risks when it comes to IT systems resilience.

Beyond such anecdotal feedback, the research provides other evidence of risk taking. The reality is that no matter how robust hardware and software components are, and how well they are put together and managed, the nature of the IT beast is such that failures will always occur at one level or another. The question then becomes how to keep the application (the piece that really matters) up and running or at least to minimise its downtime. Techniques such as automatic failover to a standby or parallel system for more critical applications, and automated recovery where the business can live with a

certain amount of downtime, then come into play, and solutions to implement such measures are available and well proven. However, when we look at the kind of protection organisations would ideally want in place and compare this to what is actually there, we see a pretty big gap (Figure 7).



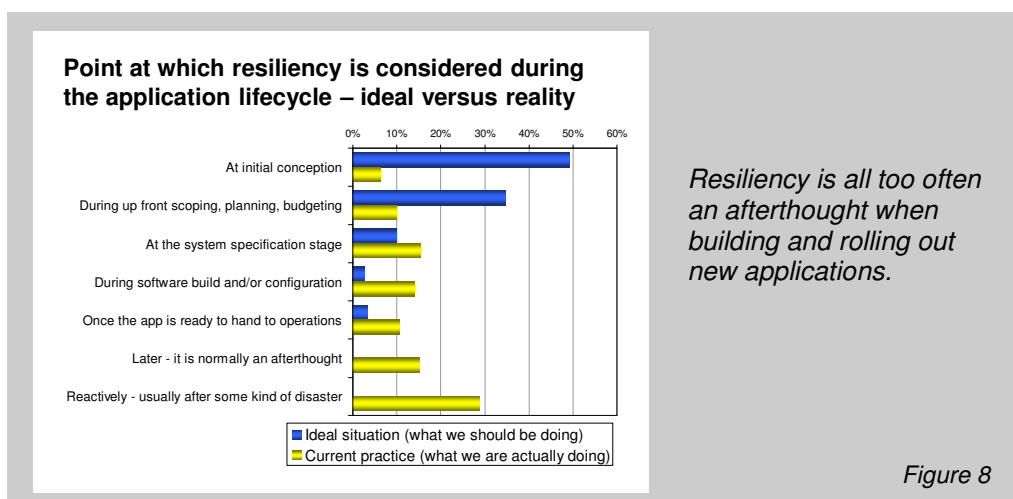
The absence of 'safety net' measures, such as these, that the majority tell us they would like to have in place leads to application downtime that could in many cases be prevented.

Zooming out a little from the above, it is clear that while the exposures are often being flagged up, particularly by operations staff, on too many occasions IT and business management are opting to accept a higher level of risk in return for faster rollouts and lower direct costs. It is, therefore, not surprising that application downtime is so much of an issue.

But what lies behind this apparent gamble?

Confessions of resilience gamblers

The truth is that many business and IT managers are forced into gambling because systems resiliency hasn't been considered early enough in the application lifecycle. It is not that organisations don't know that up-front consideration of application availability requirements is necessary, quite the opposite in fact, it's just that they have not taken steps to make sure it's part of the process (Figure 8).

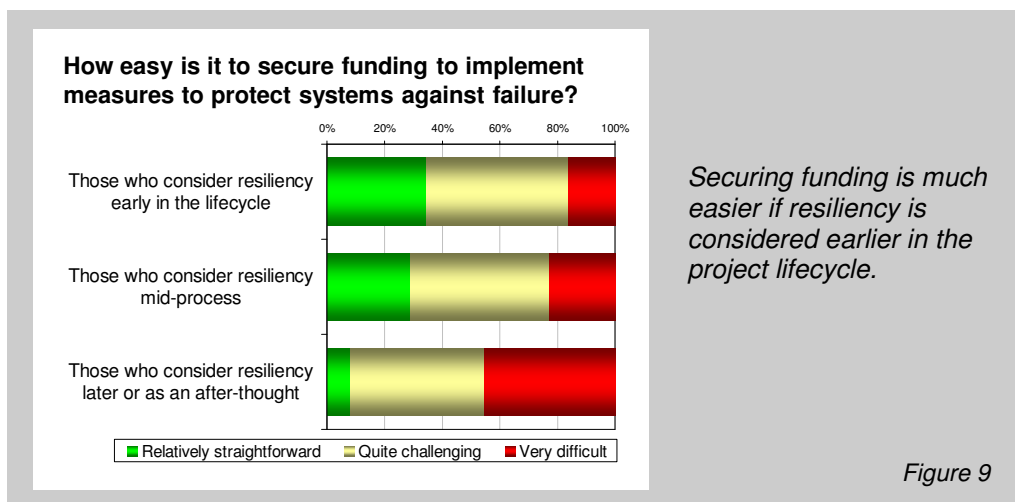


Some of the consequences of this have already surfaced from previous comments, such as architects not paying enough attention to resiliency at the design stage, and the lack of time allocated to test and prepare the application infrastructure for safe live deployment. Such factors often lead to a choice between delivering on time by taking the resiliency gamble, or delaying the rollout to strengthen the application, which risks letting stakeholders down on delivery expectations.

Apart from the impact on schedules and resources, however, when resiliency is only considered as an afterthought we run into the simple problem of limited or no budget being left to deal with it. The following comment from one respondent in the research sums up the issue very well:

"We try to insist that projects are properly specified with required availability/resilience as a primary focus before the costs are considered, but too often the features are specified and the amount of availability/resilience funding is reverse-engineered based on the original budget."

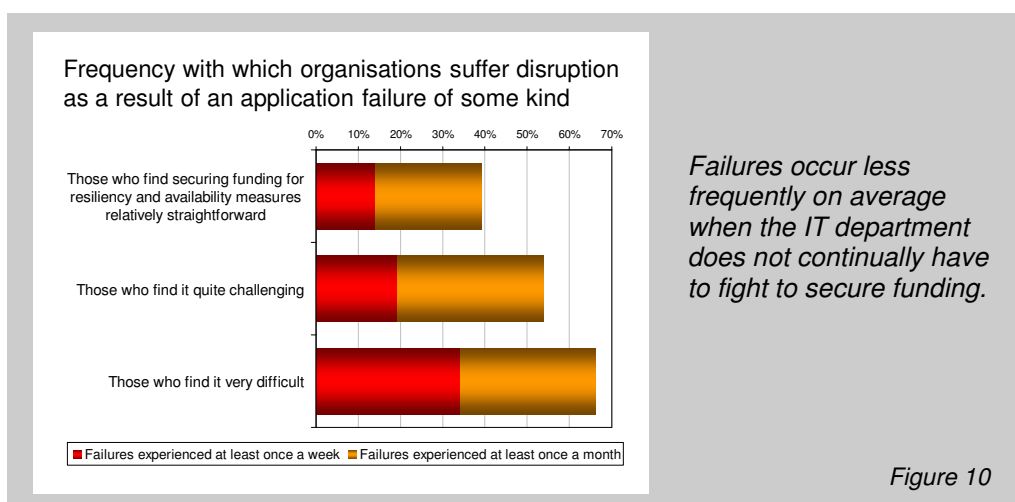
The statistical data bears this out. Those who leave the consideration of resiliency until late in the application lifecycle are significantly more likely to run into funding issues (Figure 9).



The other observation we can make from this chart, however, is that even where resiliency is considered early in the application lifecycle, many still run into challenges securing the funding necessary to implement systems robustly. Unfortunately, it can often be quite difficult to persuade budget holders of the need for resiliency if they have never suffered the pain of a serious systems failure firsthand. As one respondent put it:

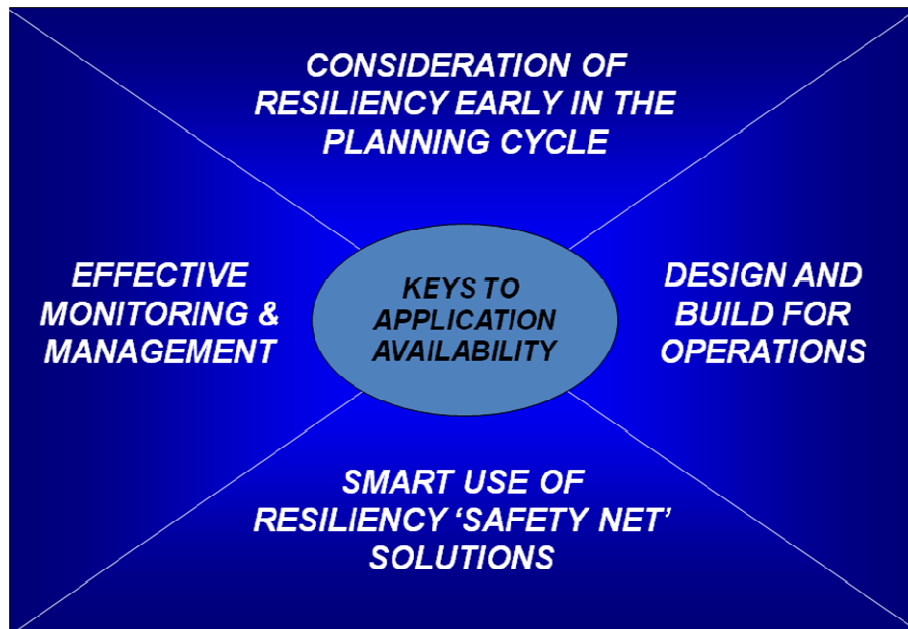
"The trick is to work with management who have experienced significant failures. They are much more open to the needs (and costs) of resilient systems. Those managers that haven't experienced a failure are much harder to convince ... till the server dies."

But how much does the appreciation of the need for resiliency and the willingness to invest in it really matter? Well as a rough indicator of this, we see from statistics that failures occur less frequently on average when the IT department does not continually have to fight to secure funding (Figure 10).



Stacking the odds in your favour

So, standing back and considering the research overall, what can we learn from the responses? First and foremost, that despite systems failures being a very real issue for most organisations, there are some very clear areas that can be treated to visibly reduce the level of failure. At a high level, this boils down to focusing on four areas, from the very inception of a new application through to its deployment and operations, as shown in Figure 11.



The 'magic formula'

Figure 11

The impact of better resiliency practice can be profound: for example, we know from respondents that those building resiliency in at the planning stage suffer only a third as many weekly disruptions from application failure as those looking at availability as an afterthought. When we home in on the more serious category of failures that cause tangible business damage, we see an even more dramatic impact, with more than a seven-fold difference at this weekly level.

As well as such general areas of focus, the research gives us a number of tangible, practical guidelines that can be taken by those organisations wishing to improve the resilience of their core systems. We have identified these guidelines from a combination of statistical analysis and anecdotal feedback, as having a significant positive impact on overall application availability.

The guidelines are summarised below, but are presented in more detail together with their rationale and key actions in Appendix B.

- Determine the availability requirements for new applications as part of the planning process.
- Involve operations professionals up front during the project planning process to discuss infrastructure requirements for new applications.
- Be mindful of the impact of software quality on ultimate application resilience. For in-house developed applications, 'design for operations' is a good starting principle.
- Ensure adequate change and configuration management processes are in place for both new and existing applications.
- Ensure that the appropriate safety nets are in place for key applications, permitting automatic failover or at least, automated recovery.
- Proactively monitor the applications in your IT environment and the platforms upon which they run, for current and future health.

It is worth saying a few more words on this last recommendation to do with monitoring, which is often thought of in the context of service levels and service level agreements (SLAs). One of the somewhat counterintuitive findings from the research was that the presence of service level agreements *per se*, and even complete best practice frameworks such as ITIL and COBIT, makes little difference to systems availability. The fact is that it is action not promises that count, and while SLAs can be useful for setting expectations, it is the proactive monitoring of systems and performance that makes the biggest difference, which may or may not go hand in hand with SLAs.

Discussion

There is one last question that needs to be answered. So far in this report, we have looked at how systems are failing, what the impacts of those failures are and what can be done to reduce the likelihood of failure. As we all know, no organisation is acting in a vacuum. Not only are there numerous projects already underway, each of which needs to be prioritised and deployed at some point, but also the failure rate is draining valuable resources required to prop up unreliable older systems. Sweeping changes are never a good idea – even if they were achievable, which given the current landscape they most clearly are not. So the question becomes, “What can I do, right now that will start to make a difference?”

Having gleaned a wealth of experience from this and numerous other studies, our best advice would be to keep things firmly grounded in reality. In the case of systems resilience, this boils down to identifying those places where efforts will make the most difference – considering those systems which are most critical to the business, either in development or operations, and tackling these as a priority. Of highest concern are systems that have already been deployed, particularly if they are already identified as ‘unreliable’; meanwhile, this should not distract from ensuring that appropriate measures are in place for those systems in development.

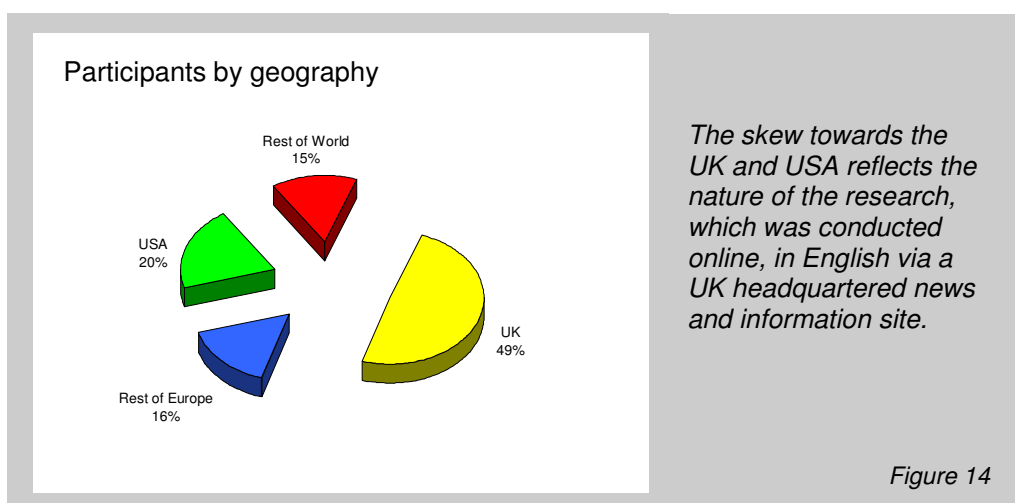
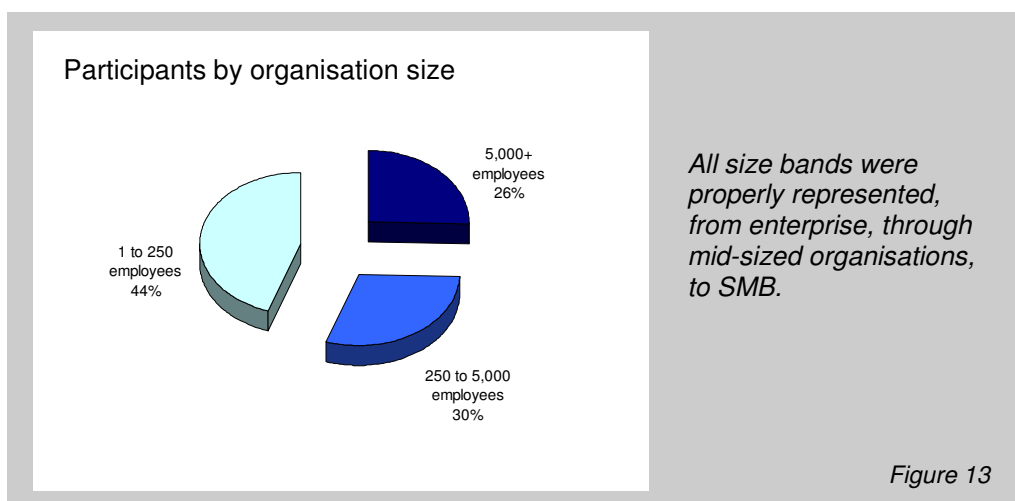
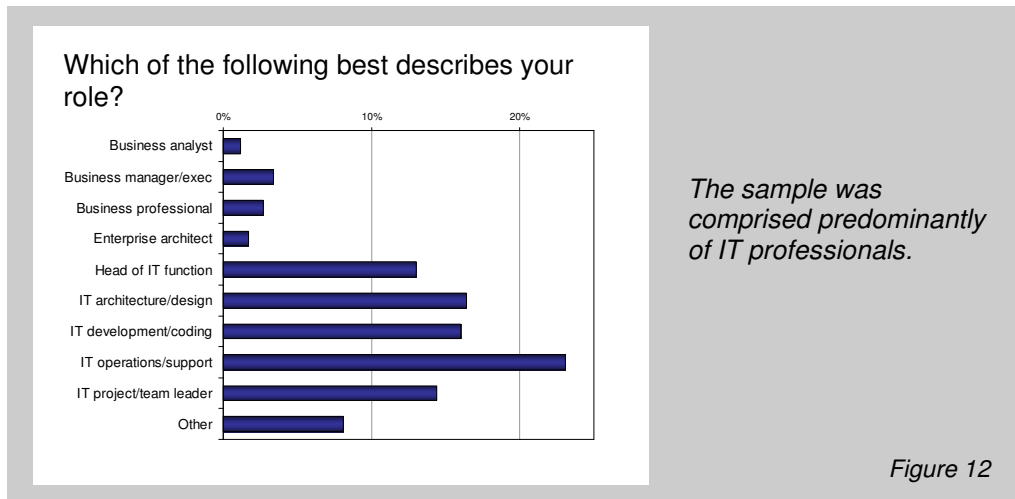
To move forward, we should not ignore the structural and political issues. Much of the anecdotal feedback alluded to the gulf in communication and understanding that typically exists between different groups – business and IT, development and operations, management and staff. To all the IT professionals reading this we would say, come on guys, you need to play more nicely together. We know that you project managers and developers think the ops guys tend to overcomplicate things, and that you in IT operations think the developers have no idea what it takes to run a system. Often you might be right, but at the end of the day, let’s bury those hatchets and start building stuff that works well out of the starting gate.

There are no easy answers, so we shall not try to round off with any glib statements. We do know however, that reducing the failure rate can start to introduce some slack into what is often an over-stretched IT department. The ultimate goal is to get IT onto the front foot, so it can better deliver the services it so wants to provide to the business. We hope that the insights and practical advice offered in this report can act as both a tool to help convince stakeholders of the importance of implementing pre-emptive resilience, as well as guidance to reduce failures in deployed systems.

Appendix A – Study Sample

The study, which was completed in April 2008, was designed, executed and interpreted by Freeform Dynamics. 1,223 responses were collected via an online survey, predominantly from IT professionals.

The demographics for the interview sample are shown in graphical format in figures 12-14.



Appendix B – Guidelines and Actions for Application Resiliency

Guideline	Rationale	Key Actions
Determine the availability requirements for new applications as part of the planning process.	This is a business, rather than a technical discussion. Availability requirements will vary by application: some need to run 24x7 with no downtime, others are less critical. By assessing availability needs up front, budget and resource may be better focused.	<ul style="list-style-type: none"> Look at “what if?” failure scenarios based on known and potential usage models. Research the common causes of failure for similar applications and architectures. Conduct a resilience workshop, bringing in external expertise if necessary. Feed availability and performance criteria into the development process as part of the requirements specification.
Involve operations professionals up front during the project planning process to discuss infrastructure requirements for new applications.	This will ensure that the necessary funding is incorporated into the overall project budget. It will also make sure that operations level preparation and testing is built into the plan, avoiding the ‘slip or gamble’ scenario discussed in the report.	<ul style="list-style-type: none"> Start talking! There may be historical baggage that needs to be worked through before meaningful dialogue starts. Identify the experts. Ensure you are talking to the right people about each topic. Collaborate on impact analysis. Conduct joint activities to assess the impact on existing infrastructure.
Be mindful of the impact of software quality on ultimate application resilience. For in-house developed applications, ‘design for operations’ is a good starting principle.	If an application is inherently unstable or intolerant of conditions or usage patterns that will be encountered in the real world, even the best operations team managing it on the most appropriate infrastructure will have difficulty ensuring its availability.	<ul style="list-style-type: none"> Ensure code reviews and unit tests include working through potential failure scenarios. Incorporate operational level testing (e.g. stress testing) as well as functional testing as part of the application acceptance process. Document built-in resiliency capabilities of the software, for example how database rollback or warm restart is achieved.
Ensure adequate change and configuration management processes are in place for both new and existing applications.	It is all too easy for an apparently trivial change to an application, its configuration or the platform upon which it runs to have unexpected effects that lead to failure.	<ul style="list-style-type: none"> Document and communicate the mapping between development and operational change management. Implement a clear separation between development/test and live systems. Ensure sign-off responsibilities are in place and being followed for new software releases.
Ensure that the appropriate safety nets are in place for key applications, permitting automatic failover or at least, automated recovery.	Even when systems are sized and designed with stability and resilience in mind, and robust change management is in place, components can fail and accidents can still happen.	<ul style="list-style-type: none"> Document ‘key’ existing applications and review whether failover/recovery mechanisms are in place. Review design criteria for systems in development to check for hardware and software redundancy, failover and recovery.
Proactively monitor the applications in your IT environment and the platforms upon which they run, for current and future health.	Effective monitoring, coupled with the appropriate alerting mechanisms, not only allows problems to be anticipated and prevented, but also generates logs that can help to troubleshoot issues when they occur.	<ul style="list-style-type: none"> Such monitoring may include throughput, memory, disk space, network traffic and other relevant parameters. Define thresholds beyond which system behaviour may become erratic and monitor against these. Monitor the maximum of useful information, even if only a subset is reflected in the SLA.

Appendix C – Referenced Work and Further Reading

[1] IT Risk in Context Freeform Dynamics Dec 2006

Suggested Further Reading

Delivering Effective IT Support Freeform Dynamics Aug 2006

Managing Information Risk Freeform Dynamics Dec 2006

IT Management Checkpoint Freeform Dynamics Jan 2008

Relieving the Systems Management Burden Freeform Dynamics Jan 2008

IT on the Front Foot Freeform Dynamics Apr 2008

All these titles are freely downloadable from: www.freeformdynamics.com

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.

About Neverfail



Neverfail is a leading global software company providing affordable data protection, high availability, and disaster recovery solutions focused on keeping users productive. Neverfail's software solutions enable users to remain continuously connected to the live software application irrespective of hardware, software, operating system, or network failures. Neverfail's mission of eliminating application downtime for the end user delivers the assurance of business continuity, removes the commercial and IT management costs associated with system downtime and enables the more productive use of IT resources. More information can be found at www.neverfailgroup.com.

Terms of Use

This report is Copyright 2008 Freeform Dynamics Ltd. It may be freely duplicated and distributed in its entirety on an individual one to one basis, either electronically or in hard copy form. It may not, however, be disassembled or modified in any way as part of the duplication process.

The contents of the front page of this report may be reproduced and published on any website as a management summary, so long as it is attributed to Freeform Dynamics Ltd and is accompanied by a link to the relevant request page on www.freeformdynamics.com. Hosting of the entire report for download and/or mass distribution of the report by any means is prohibited unless express permission is obtained from Freeform Dynamics Ltd.

This report is provided for your general information and use only. No warranty or guarantee is provided by Freeform Dynamics Ltd with respect to the suitability of the information provided within this document for any particular purpose.