

A Smart Guide published by



Driving value from IT

Objective decision-making for your
IT infrastructure



With Compliments

Technology needs context

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

An innovative research methodology allows us to gather feedback directly from those involved in IT strategy, planning, procurement and implementation.

Our output is grounded in real-world practicality for use by mainstream business and IT professionals.

For further information or to subscribe to the Freeform Dynamics research service, visit our website or contact us at **info@freeformdynamics.com**



freeformdynamics.com

Driving value from IT

Objective decision-making for your
IT infrastructure

Terms of Use

This document is Copyright 2010 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. Hosting of the document for download on the Web and/or mass distribution of the document by any means is prohibited unless express permission is obtained from Freeform Dynamics Ltd.

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.

Contents

Frontline IT: This is for you	4
What's become so difficult?	6
How did we get here?	9
Efficiency vs. flexibility vs. control	10
Short-termism, lack of responsibility and the illusion of new	11
Working from an incomplete picture	12
Bringing things up-to-date	13
Starting with the end in mind	15
Building a picture of service delivery	16
Winning support for a 'portfolio approach' to IT	19
Taking an inclusive view of existing capacity	20
Making the right decisions	23
What about the mainframe?	26
Rationalizing at the right level	28
Taking things forward	30
References	32

Frontline IT: This is for you

“Have you got a map?” – Rowan Atkinson

“If you want to get there, don’t start from here” – Anon.

Q: What do CIOs, IT directors and IT managers have in common?

A: Most of them don’t get to design their IT infrastructures from scratch.

Imagine being a decision-maker in a well funded start-up firm and being able to dictate your approach to IT right from the outset. Maybe it’s your idea of a dream job. For the rest of us mere mortals working in the IT mainstream, we get what we’re given, which probably amounts to several decades’ worth of infrastructure. Nobody knows what half of it does any more and the other half wouldn’t have been designed that way if we could do it again.

Hopes of building and maintaining a consistent and ‘future proof’ IT infrastructure can be easily waylaid by the constant stream of requirements for new systems and capabilities. Many of them create conflict, and most place additional burdens on the existing infrastructure. Innovation is a great thing, but when it comes to trying to get things right for the future, there never seems to be a good time to do any. The result is that IT decision makers are forced into a mostly reactive mode as they deliver new and improved services to the business as best they can.

No one can erase history and design the optimum infrastructure, but all is not lost. There are opportunities to get things facing in the right direction. Progress can be made if we have the means to revisit our thinking in a couple of areas: getting the most from existing assets; making incremental investments for immediate returns; and introducing future proofing as we go. That's what this guide is all about.

But wait, it would be unfair – and inaccurate – to start off with a downer on IT. Our research has shown time and again that most organizations see IT as being in decent shape. However, most also acknowledge that it could be better. We will start off from this real world view. Working out what we can actually do to make things better involves getting a handle on these basic points:

- Knowing which IT services you want to provide, and how
- Having a clear picture of where you are in terms of existing facilities and capabilities
- Making the right decisions that build on what you have

There are some great, practical lessons to be exploited from those who've already managed to square the circle. The Freeform Dynamics team spends its time finding and evaluating such best practice and we use and reference this throughout this guide.

If you're struggling with your existing IT, if you do not know where to start with a new initiative, or if you just feel that there might be a better way of doing things, then this guide is for you.

What's become so difficult?

“Life is hard, and so is IT” – Anon

“There ain't nothin' ever turns out the way you want it to” – ‘Scratch’ (from the film Crossroads)

[The event: A computer conference somewhere in North America] I was talking to the data center manager who, bizarrely as it turned out, had been proposed to me as a ‘good customer’ of the vendor whose conference it was. The manager told me how frustrated he was, listening to people talking about events management and systems management and service management like they were in some way ‘done’ – while all the time, he knew his role was one of keeping things up and running against all odds. “They have no idea what it's really like to run a data center,” he told me. – Jon Collins, Freeform Dynamics

So, how do things really look inside a data center? Anyone attending conferences a decade ago could rightly expect things to be ‘finished’ by now if they were to believe all the marketing. Blade servers, ‘open’ operating systems, management software, middleware and freeware, etc.; all would play their part to create highly adaptive, ‘on-demand’ data centers able to respond to the needs of their users. All with minimal operator intervention of course! At various times it is likely we have all wondered if our jobs were at risk, given how ‘automated’ everything was going to turn out.

Whether some people drank too much Kool-aid™ or the rest of us knew that nothing ever works quite as planned, is

moot. The fact is that IT service delivery is still much the same as it was a decade ago. In some organizations, technology really is getting in the way of doing good business. But we shouldn't be too downhearted. We meant what we said earlier. Most businesses see what IT does as OK ^[1] for now.



What we're concerned with here is how IT delivers services to help the business address its goals. Few IT organizations can say they deliver services in an optimal manner, but the symptoms are often not recognized as such and are accepted as 'the way things are' or 'that's what IT does' without ever questioning whether they are right or wrong.

The underlying causes of inefficiency in the IT realm can be summed up in a single word: 'fragmentation', which can be propagated in a number of ways and areas throughout the business. You may recognize some of them:

- ‘Organization’ – a lack of coordination between the teams involved in designing, developing and operating IT systems can result in friction, lack of co-operation and occasional conflict. Different environments, such as Windows, UNIX/Linux and mainframe can be treated as if they were different worlds, with the parties involved becoming almost tribal in their attitudes to each other.
- Planning and financing: ‘Too many cooks’ – historical convention or organizational politics can combine to out-rank objectivity when too many different parties are involved in infrastructure decision-making processes. If everybody has to have their say, there is a good chance of inappropriate viewpoints skewing things out of kilter. A common result has been that money has been thrown at, and thereby expanded, areas that aren’t working well.
- ‘Operations’ – coordinating policy, performance, management, support, troubleshooting and so on across technology and departmental domains is fraught with challenges. Technically it is difficult to achieve consistent performance, scalability and security, and the time and effort spent on communicating and co-ordinating between teams adds overhead.
- ‘Skills and resourcing’ – resource and effort can be duplicated in some areas and stove-piped in others. It can be people doing essentially the same things but following different processes and using different tools, or it could be an imbalance of skills in certain areas due to IT staff being locked into specific skill sets, perhaps as a result of some of the influences outlined above.

The net result of fragmentation is inefficiency and waste. The risk of doing the wrong things (or just doing things wrong) is

heightened, time is lost that could be better spent and, worst of all, hard-won budget may be used unnecessarily.

One of the most interesting effects of fragmentation is the 'all or nothing' view of IT platform usage that can prevail because objectivity suffers when the right decision-making ingredients are not present. For example, an application must EITHER run on a mainframe OR in a distributed systems environment. Over time, these views become 'the way things are done', and new decisions are no longer new decisions, they simply replicate whatever was done last time.

This has all crept up on us. How can we think about existing infrastructure without risking falling into the same traps, and in a way that these issues can be dealt with? If we want to beat the 'creep', we have to first understand how we got here. Then we can work out what to do about it.

How did we get here?

In the beginning, the mainframe ran by itself in a tightly controlled environment. Then minicomputers running VMS and other operating systems, and ultimately UNIX-based client/server systems, came along, all of which were perceived as providing more flexibility and local control, at lower cost. The mainframe guys were uncomfortable with the lack of rigor associated with distributed systems, and the distributed systems guys didn't want to be subject to the same constraints as the mainframe group.

UNIX-based systems 'grew up' and became more capable of handling core business requirements. This in turn placed the same operational requirements on them as had existed

in the mainframe world for years. Even though the problems were often similar, however, different tools and processes were developed to solve them in terms of both development and operations. Many wheels were therefore reinvented.

Then the PC was introduced as a stand-alone platform for local/tactical requirements, and languages such as Visual Basic, coupled with macros available in office tools, enthralled a new generation of programmers. PCs became networked, and connected to distributed systems and mainframe back-ends, before taking their own turn in the server rooms. Then the whole cycle was repeated again. Windows-based systems hit the same barriers in terms of service delivery that their more mature counterparts had experienced for years.

Today, we have a whole new series of buzz words, from cloud computing and Software as a Service (SaaS), to scripting languages and Rich Internet Applications (RIAs). These too will grow up and whole new sets of wheels will be re-invented.

What is it about this business that makes it so hard to learn from the lessons of history? Are we really doomed to repeat the mistakes of the past, creating yet more silos which require separate organization, management and skill sets? Hopefully not. To break the cycle we need to first consider its causes.

Efficiency vs. flexibility vs. control

Perhaps the main factor to consider is the balance between running efficient IT systems and the (very human) desire to remain in control. When the mainframe world became too onerous and bureaucratic, individual departments started buying

their own distributed systems to fulfil tactical or local needs. This established an ownership culture. Yes, it was triggered by a desire for flexibility and control but the net effect was an increase in complexity and burden on the IT department.

Today, we see this effect manifest itself as lines of business acquire, for example, their own mobile email ('The BlackBerry Effect') and SaaS applications for CRM and collaboration. Meanwhile, it is common for three worlds to exist separately in the IT department. For example, in development we have .Net/SQL server-centric activity around Windows, Java/Oracle activity around UNIX, and DB2 and various traditional environments on the mainframe. We see the same story when it comes to monitoring, management and security. Separate tools, solutions and suites keep efforts and skills in the IT department disjointed.

Short-termism, lack of responsibility and the illusion of new

Another very human trait is to look for easy answers, and no industry is better than the IT industry at coming up with 'easy solutions'. For example, when rack-mounted servers or blades were first proposed, they were positioned as the ideal way to consolidate existing systems and applications onto a coherent, easy to manage, cost-effective, color-coordinated and sleek platform.

The idea that you can replace what you already have with new hardware, a virtualization layer or an operating system upgrade might indeed appear very attractive. It generates the illusion of starting from scratch. However, such new solutions tend to 'add to the pool' rather than ever fully replacing anything. The

problem of complexity is compounded and a whole layer of resources may languish without ever being fully exploited. Every platform type and operating model has its advantages, but the trick (as we shall see) is to make the most of each.

Working from an incomplete picture

Some inefficiency is inevitable in even the best run IT organizations, but it can build up over time and become a problem when circumstances change. However, one of the key causes of the cycle continuing is the fact that it can be difficult to capture an up-to-date picture of one's own infrastructure. Rife are the stories of 'the server in the corner' running an application which does 'something', but nobody knows what or for whom. Or of spare processor or storage capacity that can't be re-allocated for fear of incurring the wrath of some unknown group of users.

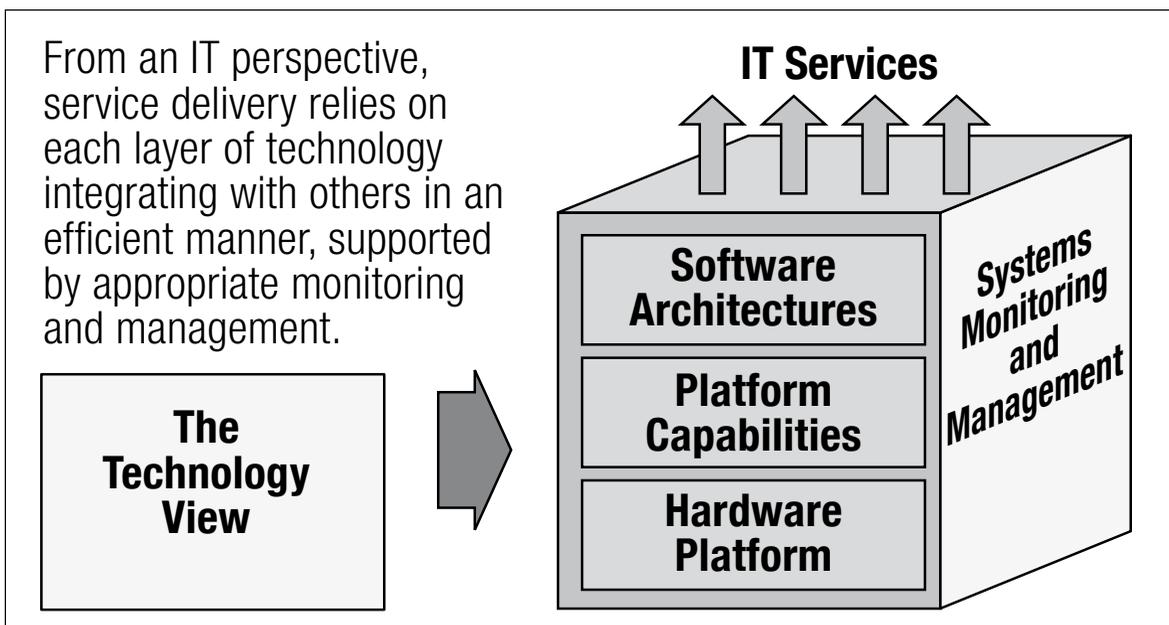
While it is understandable that organizations choose to procure new equipment because it feels like the more straightforward path, it really only compounds the issue. Eventually, new systems are subsumed into the fragmented IT portfolio and become part of the problem.

We could say that all this is unsustainable, but some organizations do manage to muddle along for many years, before eventually losing the will to continue and outsourcing the whole lot. In the meantime, estimates vary but generally accepted wisdom is that organizations use, at best, no more than a third of their IT capacity. Or to put it another way, two thirds of the money spent on infrastructure may not have been fully exploited. What an enormous waste of resources.

If we aren't careful, this kind of fragmentation could get even worse in the future. We have already seen the emergence of separate development and operational environments around mobile computing. Cloud-related developments, particularly SaaS and Platform as a Service (PaaS), introduce yet more discrete development and operational domains. However, we know technology can help as well as hinder, as long as it is used in the right way.

Bringing things up-to-date

Despite the fact that IT seems to evolve far more slowly than we might like, things are heading in the right direction. Notably, a number of increasingly mature technologies are of direct relevance to helping us improve how we define, deploy and manage IT resources.



Let's consider **software architectures** first. The 'glue' between application software has become increasingly standards-based through the use of web services and the like: only in exceptional

circumstances is it necessary to hard code interfaces. The maturing of such areas as Service Oriented Architecture (SOA) [2] has been neatly illustrated recently by suggestions of its demise. When the industry stops hyping something, it usually means people are knuckling down and just doing it. Meanwhile, web browser and Rich Internet Applications are becoming more and more the norm, with development of a mashed-up layer 'on top' becoming more straightforward due to underlying standardization.

Within the **systems software platform** arena, perhaps the biggest development has been the popularization of virtualization technology in the x86 server space. The separation of workloads from the physical hardware they run on has resulted in a reduction in dependencies between the two and more freedom to choose a hardware platform with the right characteristics for the job. Meanwhile, standards are also having an impact on the interoperability between computer systems, with Microsoft and others becoming more open and standards-aware and capabilities less siloed.

Hardware platforms have not stood still either. Calls for the demise of the mainframe began in the 1970s and have gone largely ignored as it has been extensively developed to support today's workloads. Meanwhile, distributed systems environments have (through the inclusion of virtualization) once again been suggested as an appropriate place to consolidate workloads. The missing link is orchestration and management to enable workloads to be moved around smartly as needed.

Finally, for **systems monitoring and management**, we see far better support for heterogeneous environments, while

performance management solutions provide the wherewithal to create end-to-end visibility and troubleshooting across domains.

The common vision for all these developments is the ability to take a wide variety of hardware and software components and manage them as a single entity. This image must leave mainframe guys scratching their heads, saying “But isn’t that what we had before?” And they’re right. Given all of these developments, is it possible to take such a mainframe view and marry it with the flexibility and control that the more recent platforms lack?

Above all, such technological developments give us the opportunity to improve our existing infrastructure architecture at the same time as delivering on specific requirements. The challenge becomes: how to make the right decisions for today, building on what is available now but without falling into the traps of the past.

Starting with the end in mind

“It’s important to start with the end in mind”

– Dr Steven R Covey, leadership authority

“It’s not what you do it’s the way that you do it, and that’s what gets results” – Bananarama

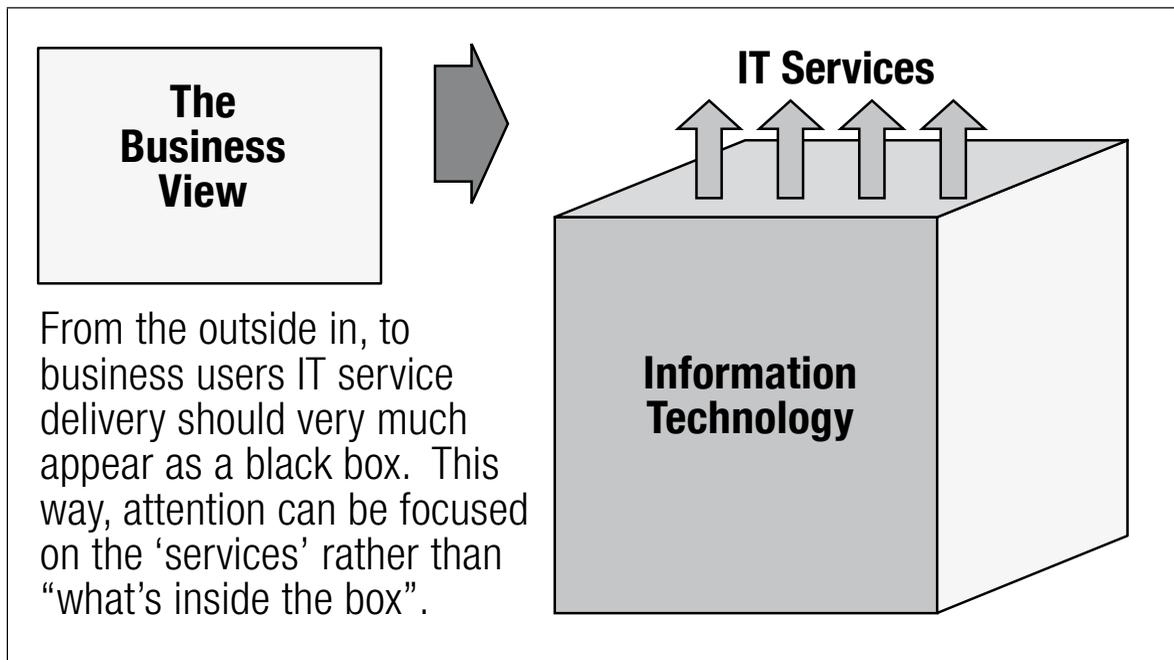
One thing is for sure: if you’re going to get ‘there’, you’re not going to make it happen by yourself. You will need to bring the whole of the organization with you. This may sound ambitious but remember this isn’t about trying to sell some pie-in-the-sky, idealistic vision. Rather, it is about presenting a reasonable and sensible argument for helping the business make better IT decisions than it has done in the past.

What do we mean by ‘better decisions’? Ultimately this boils down to striking a balance between old and new. Technologies such as virtualization now make it possible to use existing IT assets more effectively. If you already have a mainframe computer working away in the corner of the data center, for example, we want you to be able to ask, “Why don’t we just use it?” rather than taking a ‘no discussion’ decision to rush off and buy a bunch of new blade servers. We could ask similar questions of other infrastructure platforms.

Using existing equipment to address a new requirement will not always be the best option, but the key is to be in a position to at least consider it as an option. The bottom line, of course, is that any decision-making will need to be justifiable in clear financial terms. What was lacking in the past was a complete view of what IT brings to the table, and the costs involved. We’re going to need to get past this in order to develop a new way of thinking about how best to exploit IT. The first step is to build an appropriate way of articulating what IT actually does for the business.

Building a picture of service delivery

At the very start of this guide, we acknowledged that it’s rare for anybody to enjoy the luxury of building an IT environment from scratch. But what would IT look like from a business service delivery standpoint if you could start with a clean sheet of paper? The answer is that it would be very, very simple. Indeed, as close to a ‘black box’ as you could get.



Nice dream perhaps. But why can't the view to the business be like this? Outside of IT, in general, no-one needs to know what happens inside the box. The key is to be able to talk about what it's all supposed to do for the business.

We are proposing a move from managing IT as ‘a bunch of things’ to managing IT as a series of services. We want to avoid the risk of incomplete views or understanding, caused by fragmentation, influencing decision-making. A services approach acknowledges what is actually being delivered, and thus ‘service management’ becomes driven by ‘what the box does, not what's in the box’.

Putting this into practice may mean pulling a number of different threads together, to create a consistent way of framing, articulating and marshalling the resources at your disposal. Some people call this ‘a portfolio’ view. The important thing to bear in mind is that the information you need probably already exists in different places, and it's a case of pulling it all together, so that you can ‘manage the box’, as opposed to managing each and every component of it separately.

We see the following as being building blocks to let this happen:

- Consistent policies. The goal of the 'black box' is to meet the functional needs of the business in a consistent manner. In practical terms this means delivering against a set of co-ordinated, 'one size fits all' policies. By 'policy' we mean everything from service levels, compliance, risk and security requirements, to support expectations and operational requirements (e.g. backup and recovery requirements etc.).
- An 'architectural' view of IT delivery. Specific technologies may require specific considerations, but the view from the outside should be as seamless as possible. A flexible and open architectural framework is needed to work across platform domains and incorporate new approaches and capabilities such as cloud and mobile services. The goal is to be able to accommodate new developments as extensions of the current environment, rather than create a new one each time.
- Coordination between development and operations. We need to consider design, implementation and management as a joined-up discipline which ensures existing facilities run as smoothly as possible alongside each other. The goal is to create freedom to mix and match resources and platforms in order to exploit the best attributes of each to address different requirements.
- Management of technical knowledge. Taking advantage of the skills and experience that already exist in-house is pivotal to this approach. This leads to better resourcing flexibility, reduced training costs, cross-fertilization / re-use of best practices and better leveraging of knowledge and experience within IT teams. If new skills are subsequently required for a given task, you will know that you made the right choice by starting internally.

- Getting to grips with costs. One of the ‘legacy effects’ of fragmentation is that a true picture of the cost of delivering IT services is hard to come by in many organizations. It also makes the job of deciding the most appropriate course of action more difficult. There are ‘hard’ and ‘soft’ costs to consider. For example, if a system has a certain set of fixed overheads such as software licensing fees, does that mean it’s worth always trying to max out utilization of spare capacity?
- Right tools for the job. This is the net effect of getting the other building blocks in place. The ‘box’ should be capable of delivering appropriate functional, security and service levels as required by the business. This means matching IT capabilities to service requirements in an objective manner, with notional concepts of ownership or organizational politics being eroded over time. This should be as true for new equipment and software as for the reuse of existing IT assets.

It would be a tough job to get all this in place in one go. Rather, they are goals to work towards. The benefits could be significant though: consistent conversations with the business, co-ordination of effort and resource between IT groups, and a better understanding of how to balance quality, cost and risk when designing and delivering services.

Winning support for a ‘portfolio approach’ to IT

All we seek to adopt is a co-ordinated approach to IT decision-making. Who wouldn’t want that? The picture may be a simple one to paint but the biggest challenge to

making anything happen can be getting buy-in. The will to change may be weak. Long established and deep seated differences between groups need to be addressed and worries about 'protecting empires' need to be neutralised.

Senior management needs to accept, and IT leaders need to promote, the fact that all platform domains are equally important to the business. This being said, there will always be conflicts, particularly if any resulting changes fail to take into account the hard realities of running IT systems from day-to-day. While dissolving existing boundaries between platform domains may be appropriate in some cases (e.g. between Windows and UNIX groups), it may not be in others.

The challenge is to agree on the principles of a portfolio approach across senior IT management. It need not even require anybody to do things differently – not at first, in any case. However, we have seen improvements in how IT decisions are made from organizations that have taken this route. They are driven to exploit existing IT assets and capabilities to maximize business value, and this process encourages the creation of cross-domain structures, policies, procedures and communication mechanisms.

Taking an inclusive view of existing capacity

So, what exactly is in your IT portfolio? Surprisingly few organizations (perhaps not surprising to those who have worked in IT for a while) have a clear idea of what IT assets they have. A first task then is to build such a view. Recognize from the outset however, that the benefits of specific platforms can be obscured by individual familiarity, 'platform advocacy', out-of-date perceptions, supplier loyalties, and even competition between internal groups.

To overcome these challenges, it is useful to agree and document an impartial view of platform attributes. An important element of this exercise is to identify similarities or equivalences between platforms as well as differences. This then becomes a foundation for making objective deployment/migration decisions, and for understanding the impact of subsequent changes as requirements evolve over time.

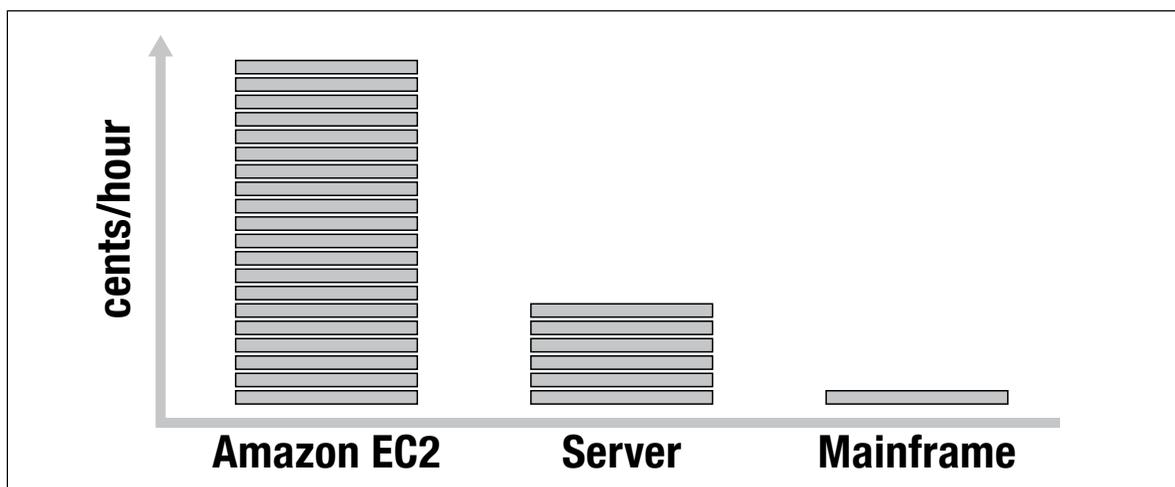
Such attributes include:

- Cost of acquisition/allocation
- Storage cost and flexibility
- Power consumption
- Software licensing costs
- Ease/cost of operational management
- Inherent security
- Ease/cost of security management
- Performance and scalability
- Resilience and recovery
- Application support
- Standards support

By undertaking this evaluation, you will be in a better position to appreciate your existing capabilities in terms of raw processing power and the skills required to harness them, as well as what, if any, assets are consuming resources without delivering any real business benefits. Considering the Pareto principle, you should be able to identify the 20% of your systems that are delivering 80% of business value.

Frankly, it is unlikely that everything you have is used to its full capacity. As well as the configuration management benefits that you can derive from such a review, one of the most useful things that can come out of this exercise is a clearer picture of the spare capacity that remains to be exploited, and the costs of using it. Not only does this give you an indication of current wastage, it also allows you to assess the cost-effectiveness of exploiting existing capacity versus buying, installing and managing new capacity from the outside.

A good example is a US bank which was reviewing whether to run certain workloads using Amazon's EC2 (Elastic Compute Cloud) service. Amazon's price looked attractive, at 10 cents per CPU hour. The company took the time and reviewed how much its distributed server capacity was costing, which came in at 3 cents per CPU hour. For the mainframe platform, the cost was 0.5 cents per CPU hour. It doesn't take a genius to work out which option they went with!



Here's the big question: if, like most organizations, you are only using a third or less of your existing capacity, what hurdles do you need to overcome to gain access to your excess capacity? If it is down to location, then you can consider what access technologies might be appropriate; if you feel you lack skills, then perhaps some

recruitment or training is in order. But if the hurdle is organizational politics or historical decision-making, then the CEO ought to know as your organization is probably throwing money down the drain.

Making the right decisions

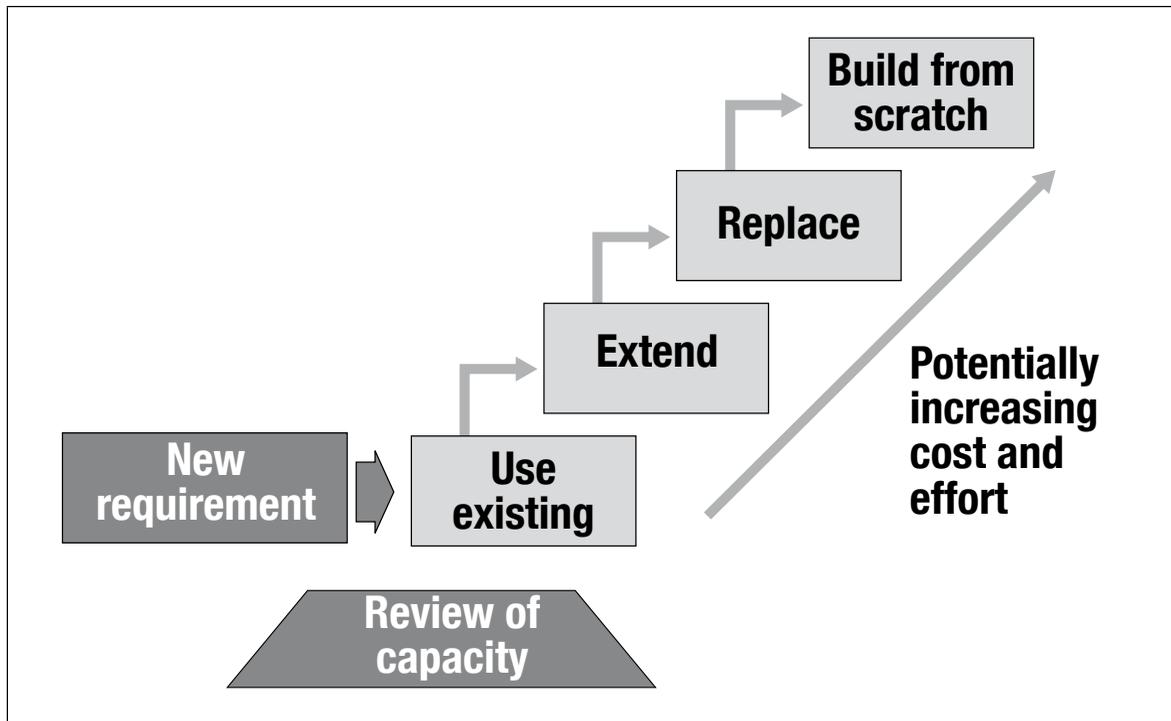
Given the above, how can we move forward? The key is an approach which cuts across domain-specific teams to allow platform-related decisions to be made on a dispassionate basis according to objective business and technical criteria. Whether this manifests itself as a formal body of people or is simply a mechanism to pull together the relevant people 'on demand' is less important than the participation of people from all domains. In some circles this is simply referred to as 'good IT governance'.

Faced with a demand for a new IT service, a decision-making flow which builds upon everything we have seen so far in this guide is needed. It goes something like this:

- Given your knowledge of existing capabilities and spare capacity, can you make better use of what you already have?
- If not, does the opportunity exist to extend or upgrade existing facilities?
- If not, can they in some way be replaced in all or in part?

If the answer to all of the above is 'no', then you may well need to build something from scratch or buy it. You can be quite sanguine about this. There will be occasions when it is not possible to build on what you have. But it is better to make an active decision to do so, rather than defaulting to the behavior of buying new. This

'eyes wide open' process needs to be fed with all costs, including integration and skills development, as well as pure acquisition costs.



Let's look at each of these areas in turn:

Use existing. Common sense suggests that the starting point for any investment decision should be, "Why can't I use what I already have?" For example, if you have determined that you do indeed have excess mainframe capacity lying unused, the question becomes, "Why not the mainframe?" There may be very good reasons why not – for example there may be Windows dependencies in this case. At which point, the question might become, "Why not the under-used blade servers?"

Which brings us to:

...Extend existing. Another opportunity may be to make an incremental investment in existing infrastructure, either to add capacity or to adapt it to suit the new workloads. For example, the use of virtualization in either the mainframe or the distributed

worlds allows workloads to be consolidated down to a smaller hardware footprint, freeing up resources for new workloads. Indeed, given everything we have said about ownership culture and politics, virtualization may well offer a half-way house. "Want to keep your own servers? Fine – but they will be virtual servers." If this isn't possible, you may look to...

...Replace existing. In some cases, it may be most appropriate to buy a consolidation platform which adds flexibility to support existing and new workloads. Our research has shown that the mainframe is seen as just as attractive as a server-based platform in these scenarios. You may also find that a reuse opportunity still emerges in the software layer. While less attractive from a hardware cost perspective, an organization may ultimately decide that its existing environment is not appropriate to be built on. So you may be forced to...

Build from scratch. This has to be the least attractive option from both a cost and effort perspective, but sometimes there will be no choice other than to rip up the current environment – or part of it – and start again. In the worst case from an IT economics perspective, this will involve creating 'another silo'. However, valid reasons may exist, for example, to meet specific security requirements or run a highly customized application.

Sometimes, spare capacity may come from unexpected quarters. A UK bank we talked to said "We would never use cloud-based services for our core business". However, when the conversation turned to spare capacity, the same organization proudly described how it was using the unused processor power at its disaster recovery site for offline analytics. Resource is resource, wherever it is being hosted!

What about the mainframe?

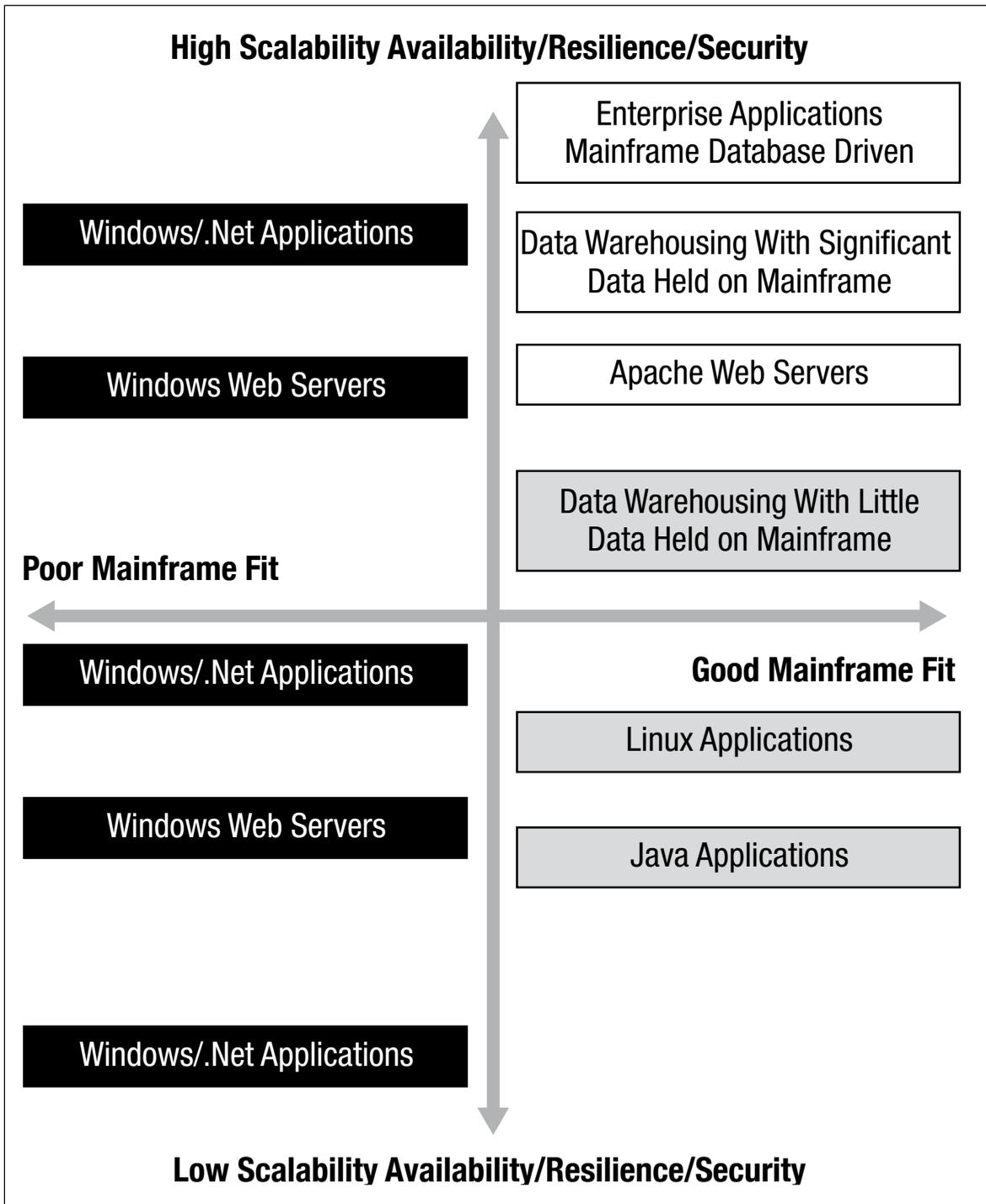
While the mainframe platform may have a reputation as ‘old technology’ for some, it is equally accepted that it offers low incremental management costs as workloads increase; it is also accepted as providing the ‘gold standard’ for security, assuming of course that it is administered correctly. Yet many organizations are not making full use of their mainframe assets, which essentially means they are paying for something and not exploiting the maximum value they could be getting from it. In other words, they are wasting money. A good question to pose is, “If I have spare capacity available, why not use my most efficient, least expensive and most secure infrastructure more?”

While the mainframe is not appropriate for every workload, it does offer a low incremental TCO compared to other platforms. In other words, if you have it, you might as well use it. In addition to DB2, Oracle, SAP and WebSphere, there are over 600 applications that now run on the mainframe so examination of it as an option for new or consolidated workloads is warranted.

It is also worth mentioning that ‘speciality engines’ are now available on modern mainframes, which enable lower-cost processing of specific workloads, such as data-intensive applications, Linux virtual machines and Java applications. With these capabilities and costs in mind, the figure alongside gives an overview of which workloads are appropriate to consider for a mainframe system:

- ‘Clear’ boxes are most suitable for the mainframe, especially where it has speciality engines in place and workload capacity available

- ‘Shaded’ boxes cover workloads that may well be good candidates for running on a mainframe using an offload engine. These will even shade towards ‘clear’ as scalability increases or availability becomes an important factor
- ‘Solid’ boxes identify workloads that are today unsuitable candidates for running on a mainframe.



We recognize that rationalizing the use of IT in general is an evolutionary process, so it is important to make each decision count towards the vision. A single decision to deploy a new core application on an existing mainframe, that would otherwise require the acquisition of a new distributed x86 platform, could save millions in terms of total cost of ownership. By the same token, replacing a rigid and expensive-to-maintain legacy mainframe system with an Oracle-based package on Linux or UNIX could reduce costs and boost service levels dramatically. The choice will be unique to your business, the resources available and the requirements at hand. It is being free to take an objective approach to choosing and mixing platforms based on the genuine requirements that really can pay dividends for both IT and the business.

Rationalizing at the right level

Improving decision-making is not just about addressing new requirements. We also need to consider incremental improvement of the infrastructure as a whole. ‘Evolution, not revolution’ is the key phrase here, and a broader view than ‘just infrastructure’ is required. The service management view of the world works as a pivot for activities within the IT department, and can be used in a consistent manner across all domains and teams when thinking about the bigger picture too.

With this in mind, a number of areas can be rationalized over time:

Development. While attention is often more on rationalizing and consolidating hardware, it is worth trying to rationalize what’s used in the development realm too, particularly for

analysis, design and documentation. Dropping platform-specific languages and tools for new developments is unlikely to make sense, but introducing portable languages and tools that can run across multiple platform domains, e.g. J2EE, web scripting languages, etc., is useful wherever possible without compromising performance, productivity, etc.

Operations. From an operational perspective, one of the first priorities is to gain visibility across platform domain boundaries. While it might be difficult to justify replacing your domain-specific monitoring and management solutions overnight, at least look at how basic integration can be achieved and/or information can be exchanged in terms of dependencies, performance, status, events and so on. As time goes by, it is worth focusing investments on those toolsets that genuinely embrace multiple domains, and phase out those that are constrained to supporting just one environment.

Communications. The requirement here is two-fold: firstly to break down prejudices and build trust and working relationships across IT and business domains; and secondly to raise the awareness, knowledge and competence of staff with regard to activity taking place in related domains. Establishing cross-domain teams to work together on either policy and process or specific project requirements could be a positive and constructive place to start. Where necessary, briefing sessions will broaden people's thinking and facilitate cross-fertilization of ideas and best practices.

Skills. A 'hybrid' IT environment also requires 'hybrid IT managers and staff'. We have learned from a number of organizations the importance of 'sticking to the knitting': deep technical knowledge of specific areas is an absolute requirement. However, we have also learned that organizations employing multi-skilled

technical staff are better able to deal with new requirements and deliver flexible solutions than those who employ only single-skilled staff. In many cases, dual-skilling is the key, for example, UNIX/Windows, Mainframe/Linux, or Storage/Security.

Architecture. Specific areas to consider include capturing examples of standard usage patterns, which can be used as templates for other initiatives, or designing new ones for meeting commonly occurring requirements such as mixing and matching platforms according to their attributes and the ease with which they work together. Examples include:

- Mainframe central processing with local user empowerment via Windows-based solutions
- Extension of mainframe packages into the distributed world and vice versa
- Local business intelligence requirements and situational application requirements drawing on information from back-end systems.

As well as producing useful practical outputs, such activity will start to dispel myths, bring historical perceptions up-to-date, and generally build trust across and between teams. Note that the goal of all rationalization exercises is to reduce waste and increase flexibility and they should be judged accordingly. If an initiative is not going to save you money or make you money then why do it?

Taking things forward

We have highlighted a number of areas which could form the basis of a strategy to help improve how you use your IT

infrastructure to deliver services to the business. Here is a brief reminder of how they can translate into action in your business:

Work out where you are today: A dispassionate, ‘matter of fact’ analysis of your IT infrastructure means you get all the cards on the table. This isn’t a blame game, it’s the first step towards being able to make better decisions.

Get up to speed with what’s new: Software architecture, system software, hardware platforms and monitoring and management capabilities have all changed – perhaps significantly – since the last time your organization took a proper look. There is more choice than ever before and your new, more open approach means your options may have just expanded!

Arm yourself for *real* decision-making: ‘Nobody knows what they don’t know’. Equally, decisions you take in isolation could be massively skewed without you even realising it. Instead, ‘starting with the end in mind’, build a picture of IT service delivery using the building blocks we have outlined, push the portfolio approach and really get to grips with the attributes and capabilities of your existing capabilities.

Apply what you’ve learned: Practice this new approach to IT decision-making. Take an existing IT investment or project and work through the steps outlined in this guide. Would you have done anything differently? Should you have ‘used’, ‘extended’, ‘replaced’ or ‘built from scratch’? Gain confidence, gain support across your teams and then let it go for real next time there is a decision to be taken.

We do not advocate any one technology or approach. However, we are slightly evangelical about the notion of making the most

of what's there, and breaking the cycle of throwing away the goodness in past investments and starting from scratch every time.

If you are setting out to make some improvements or investigating a potential new investment, you should treat it as any other change programme. The ultimate enemy is the status quo, so consider using incentives and recognition programs to encourage the right behavior. The aim is to neutralize any adversarial or competitive mindsets between groups and to reward or recognize achievement that relates to the greater good.

As a last thought, we advise you to take things steadily. Many of the changes we talk about here involve tackling years of ingrained views and behavior. The areas we are exploring depend as much on people as they do on technology and process. Getting it right, or starting to get it right, however, means huge benefits can be unlocked.

We hope you have found this guide useful. If you have any feedback please do not hesitate to contact us at info@freeformdynamics.com.

Cheers!

References

[1] "IT Management Checkpoint – The Next 5 Years" – Freeform Dynamics

<http://www.freeformdynamics.com/fullarticle.asp?aid=129>

[2] "Service Orientation in Business – Harnessing change from the board room to the data center" – Freeform Dynamics

<http://www.freeformdynamics.com/fullarticle.asp?aid=632>

SUCCESSSES TO CONSIDER AND LEARN FROM

Bank of Russia saves US\$400 million per year by consolidating to IBM System z9

With a variety of local payment processing systems running on more than 200 distributed servers in 74 data centers across 11 time zones, Bank of Russia faced significant challenges in terms of operational efficiency, technical support, and security.

To resolve this, the bank consolidated its 200-server infrastructure to 4 IBM System z Enterprise Class mainframes in two data centers. The full-time staff requirement was reduced from 1,500 for the old systems to 200 full-time staff to run the new environment.

Payment processing costs have been reduced by 95%, saving \$400 million per year. Server and data center consolidation creates further savings on hardware and software licensing, maintenance and electricity, and increases security. Workload for technical staff has been reduced by 85%.

For further information see

<http://www.ibm.com/software/success/cssdb.nsf/CS/STRD-7KNCM7>

Baldor Electric drives down IT costs through standardization

Baldor Electric is a worldwide manufacturer of electric motors, mechanical power transmission products, drives, and generators. As its business grew, the company found itself in a vicious circle where they had to constantly add more servers and operations staff to scale the computing environment. As corporate acquisitions took place, IT inherited both more distributed servers and more people to manage them.

To break the cycle, Baldor standardized its business processes and consequently its IT environment, from various applications running on numerous UNIX servers, to SAP running on mainframe Linux. As a result, the company has driven down its IT costs as a percentage of the company's sales to 1% of sales! In addition:

- Electric costs have dropped 40%.
- Air-conditioning costs have been cut 50%
- Floor space requirements have dropped from 6,000 square feet to only 1,000 square feet
- All while greatly improving systems availability, reliability and security

For further information see

<http://zjournal.com/index.cfm?section=article&aid=1214>



A Smart Guide published by



While no one has the luxury of wiping out history and designing the optimum infrastructure from scratch, there is a lot that can be done by rethinking how to get the most from your existing IT assets. Addressed in the right way, incremental investments can not only boost immediate returns, but can also introduce more future proofing over time.

This guide is for IT leaders and professionals who are interested in evolving their IT infrastructure and how it is managed with a view to unlocking both shorter and longer term value, but who don't necessarily (who does?) have the luxury of embarking on highly disruptive and expensive transformational initiatives.