

WHITEPAPER

Stabilizing Critical IT Services

Introduction

Today, IT has to not only deal with internal issues and organization, but also juggle with the implications of remote working. With some people working in the office, some remotely, and some in different time zones, IT has to service the internal applications, Enterprise Resource Planning (ERP) software, customer-facing applications, and a host of other applications. They have to be available 24/7 to support a distributed workforce. Additionally, there is a huge amount of uncertainty in business, and that follows through to IT.

At the same time, we are seeing a constant and relentless progress in IT operations and pace. Organizations find themselves grappling to manage IT as they prepare to adapt to changing requests from shifting requirements. They struggle to focus their IT efforts and investments, and find it a challenge to maximize the impact of existing resources and investments in technology.

It is often unrealistic for organizations to expect every single IT change to be rolled out successfully, and the importance of the ability to manage expectations is reflected in the increasing gap between leading and low-performing organizations. However, this gap also puts pressure on IT teams around continuous improvement as stakeholders have higher expectations of the outcomes.

In order for IT to find a viable solution to these challenges, longer-term thinking is needed.
Certain crucial questions need to be addressed before IT can implement agile solutions in a changing world: How can IT visibility be achieved? How should decisions be made? How can complex problems be solved? How can collaboration be fostered? And lastly, what blockers and frictions exist in current IT systems?

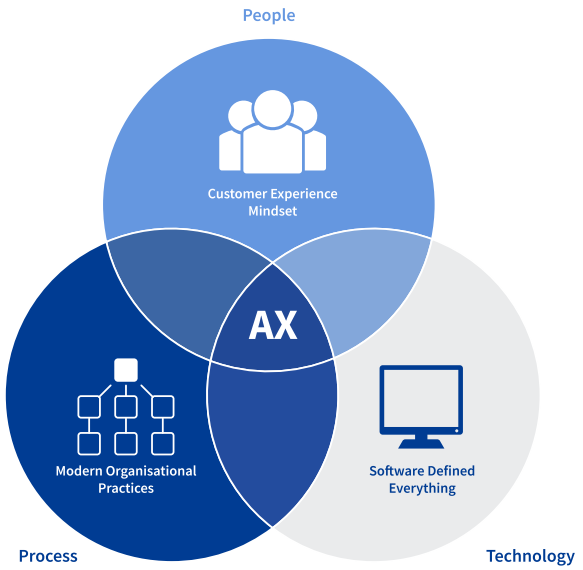
The Application Experience

Thus the goal of modern IT is, in essence, to be part of strategy and to look at what is or is not fit for purpose, which leads to questions about prioritization, limits in resources, limits in budgets, and complexities in trying to execute specific IT goals.

At the center of this is application experience, which is often thought of only in terms of end-user experience, but should instead be viewed from a more holistic perspective.

The Goal for Modern IT

Manage & Enhance your organisation's Application Experience [AX]



Beyond customers and end-users, application experience applies even to teams accountable for infrastructure. Infrastructure teams are responsible for IT services that are consumed by another IT group within the organization, whether it be Virtual Desktop Infrastructure (VDI), Hyper-V platforms, Virtual Machines (VMs), or any

other kind of hypervisor. Although this is very internal-facing, and the "customers" or users are still within IT, these users are still consuming a kind of application. Therefore, the same kind of tooling, thinking and mindset can be applied just as well for infrastructure teams as for application teams.

Other than merely considering how end-users interact with digital systems, organizations should also take into account the experience of operating these applications. Application experience extends to business stakeholders, who want to see what they're getting out of the organization.

All this requires a careful balance of multiple factors, and should be measured in terms of how people, process and technology are brought together to drive impact. For IT teams, key to achieving this is paying close attention to process.

The importance and value of good process should not be underestimated. Whereas IT teams often cannot influence the behavior or size of the workforce, nor determine the budget, establishing a good process is within IT's power.

Although organizations are faced with all kinds of messages about the latest technology, or the need for a different approach or system or upgrade, IT teams tend to overlook the fact that there are many technologies they have already invested in. And many other technologies are well-understood and very commonly available, and can have a huge impact if utilized to their full potential. With the right people, the right technologies, and some simple process enhancements or tweaks, IT teams can get some breathing room to start thinking longer-term and reinvesting resources elsewhere.

Although putting the right processes in place does not guarantee success, having the wrong processes in place does guarantee failure. No matter how good the people or technology, without effective processes that are really focused on removing hurdles to value creation, IT investments will not have the desired impact. Therefore, process is very important and is usually one of the easiest things to change, even on a team level.

The Importance of Strategy

One of the major drivers of success in delivering application experience is a sound strategy that is continually assessed. The more an organization plans, tests, and revisits its strategy to make sure it aligns with the current environment, the greater its chances of success.

Strategy enables organizations to stay abreast of current developments and implement the best solution for the future, rather than what was best in the past. Often, especially in uncertain times such as these, organizations without a well-crafted strategy turn to panic-buying to satisfy immediate needs. However, these buying decisions are frequently not thought through, resulting in the organization having to face a host of complexities around how to install the software, use it, or even whether it does the job. This is not necessarily due to good or bad technology, or even process, but whether the technology and process are appropriate in the current situation.

Before planning a sound IT strategy, organizations need to take two major steps to ensure that long-term goals can be successfully implemented. The two steps are: (1) application assessment and (2) getting control of the worst offenders.

Part 1: Application Assessment

Planning a Strategy

In order to start planning a strategy, IT executives should develop situational awareness. This will enable strategists to identify where the biggest impact can be gained, particularly in view of limited time and resources.

There are two useful models that can help with strategic planning: (a) Wardley Maps and (b) Moore's model for the technology adoption life cycle.

Wardley Maps

A Wardley Map maps the structure of a business or service, along with the components needed to serve the customer or use. It's very external facing, and focuses on where a particular technology lies in its evolution and maturity, all the way from its academic beginnings to its implementation and use. It's fully productized and commoditized, taking into account its utility. Wardley maps lend themselves to making decisions on which components should be built, bought, or outsourced.

Moore's Model

Moore's model is different from Moore's Law, which is well-known in CPU computation. Unlike Wardley Maps, Moore's model focuses on the internal operations of the business or organization. It looks at where different types of applications or technologies fall within an organization's revenue stream.

For example, Moore's model asks questions such as: Is a technology/application sitting in a cash cow that has been there a long time but is probably nearing the end of its life? Or, is it in the middle of that growth phase? Or is it future-looking, in that revenues may not be immediate, but that the technology/application is necessary for future growth?

Asking such questions helps suggest management strategies, which Moore's model looks at in depth. It helps organizations decipher the different types of applications that will benefit an organization or, at least, help to manage risk, and allow it to forge a path for internal transformation and internal innovation.

Making use of the two models

The two models can get complex, but a brief and quick assessment can be done easily by classifying applications according to several criteria, such as whether an application is general-purpose, industry-specific, or organization-specific. Such a classification can be a useful starting-point in determining the strategy for the organization.

For example, if an application is general-purpose, it is likely that it is used by all enterprises. This suggests that it is highly standardized and available as Software as a Service (SaaS), or at least as a commercial off-the-shelf offering.

If an application is industry-specific, it is also likely to have some kind of common solution depending on the size and specialty of the industry. And then, there are organization-specific applications that are so fundamentally tied into how an organization operates differently from any other, whether it's a unique piece of IP or is something that drive differentiation.

After determining the type of application, the next step is to analyze how an application is viewed by the organization or business, including its stakeholders, its management and IT teams. For example, is it internal facing? Is it maintaining current needs, or is it focused on building future growth?

Then, the organization needs to look at different ways of creating, operating and managing applications, and finding stable patterns and combinations. The best possible outcome is for it to be reliable and stable, while taking up minimal resources to operate, and successfully meeting user expectations.

Sometimes, organizations end up getting a good outcome, but are spending more resources than required. However, in the worst-case scenario, an application is unreliable, receiving frequent complaints and outages, even when it's stable. On the other hand, an application, even when meeting user expectations, might have so much technical debt that it is unable to accommodate new features or value-added capabilities to keep up with user expectations.

Starting an assessment

Application Assessment Worksheet

1. TYPE	General purpose enterprise application (eg. Payroll)	Industry specific application (eg. Hotel reservation)	Organisational specific application (eg. Proprietary in-house trading platform)	
2. FOCUS & PURPOSE	Focused on the future of our business and meeting future customer needs	Focused on maintaining our existing customer base and meeting current customer expectations	Focused on internal users and efficiencies with no direct Customer visibility or value	
3. ORIGIN & LOCATION	In-House Custom Development Application	Outsourced Custom Application	Self-Hosted COTS Application	Outsourced or SaaS COTS Application
4. EFFECTIVENESS	Reliable & Stable using minimal resources, meets User Expectations	Reliable & Stable but consumes a lot of resources to maintain, meets User Expectations	Unreliable with frequent complaints, escalations and incidents, doesn't meet User Expectations	

An application can be reviewed against each row to see which description fits it best.

Some applications have stable patterns that are likely to lead to the best outcome. And some are probably unstable, even when they are not causing problems at the moment, but might in the future.

Take the example of certain kinds of general purpose enterprise applications, such as payroll. Although payroll could differ according to geography, different tax rules, HR rules and so on, it is generally something well-solved. It is not something most organizations would be creating in-house, and is generally an SaaS or commercial off-the-shelf solution, even if it is an on-premise type deployment. Other than a vendor-management kind of relationship, such applications require no overhead or resources on the part of the organization.

In such a scenario, user expectation tends to be very stable. Expectations do not change very often and are well defined. This is an example of a stable pattern. However, if something does not fit this pattern, it does not mean that something is fundamentally wrong. However, it does mean that it should be looked at and questioned, and that the reasons for not following the pattern are well-understood.

Side note: Moving to the cloud

Even though an application might be running on cloud infrastructure, it is still self-hosted and self-operated, which is different from SaaS or Platform as a Service. In the case of cloud hosting, the infrastructure is outsourced, but not the operations of the application itself. For general purpose applications, except when there are clear and explicit constraints, an SaaS is recommended because it will handle both the infrastructure and operations.

Cloud is not the solution for everyone. Although cloud providers allow for flexibility to scale up or down, scaling up or out increases cost, and the exact cost increase is sometimes unforeseen. When moving to the cloud, it is important to conduct a trial of the application and staging environment, to see if it can handle the load. If not, an organization should calculate the amount of time, effort and resources needed to get to an acceptable point. Then, a cost-benefit analysis should be made, in terms of hosting an application on a private versus public cloud.

Huge cost, huge risk

General purpose apps, and even some industry-specific ones, can put organizations in precarious situations. When such an app is built and operated in-house, the best case outcome is probably that it is reliable and stable and that it meets user expectations. However, it can consume a huge amount of resources to maintain, although it brings no real customer value or visibility. In the worst case, organizations have the resources needed to keep it stable, and massive chaos and disarray ensues when something goes wrong. And because these apps entail huge cost, huge risk, and very little benefit in terms of business growth, the outcome of these apps is the opposite of the ideal scenario in terms of the balance between stability, value, cost and risk. This category of apps is what is termed anti-pattern.

A classic example of such a general purpose app is, again, the payroll app discussed above. However, many industry-specific apps also follow this trend, such as shopping carts, hotel reservation software, and the like. For example, if a hotel reservation software stops working, a business will be in a lot of trouble. At the same time, a hotel reservation software is focused on maintaining current existing customers and does not draw new ones. In this case, outsourcing or using SaaS versions are ideal. Even though it might seem like the upfront dollar value is larger, such services enable organizations to offload the resourcing, energy, time, planning and skills.

Sometimes, organizations contemplate outsourcing such apps to custom development firms or self-hosting off-the-shelf versions. Again, the best case scenario is probably achieving reliability, stability and user experience, but the amount of resources in terms of vendor management, operations and cost could exceed the value of the app. Custom apps are in danger of accruing a huge change cost, or a very long lead time on changes with an outsourced provider. Custom apps or self-hosting are not always wrong, but should be justified to ensure benefits and payoffs that are in line with business value.

Managing Trade-offs

Exceptions can always be found for patterns and anti-patterns, but if an organization has limited resources, they need to be put into very strategic, high-value, essential customer-facing applications. The only way to do this without extra people, money, or technology is to make some trade-offs. Models like Wardley maps or Moore's model can help show where trade-offs should be made in order to best manage the risk because, realistically, there is never going to be zero-risk. What organizations should be focusing on are applications that are going to directly impact top line and bottom-line revenue, and resources should go into

making them excellent. For such cases, there is a big red flag around outsourcing.

If something is strategically crucial to line of business or to an organization as a whole, to be dependent on an external outsourcing firm is very risky, and vendor management and due diligence has to be water-tight. Organizations also need to have enough clout to have the dominant positon in that relationship, in order to minimize risk and be in a stable position. If deciding to outsource, decision-makers should make sure to justify why outsourcing is necessary, and how risks can be mitigated if the vendor is not performing.

Part 2: Getting control of the worst offenders

The Key to Rapid Recovery

In essence, an organization is trying to reach higher levels of velocity in terms of the amount of impact and the amount of change it can deliver. This cannot be reached without stability, especially in the long-term. In the short term, an unstable application may be able to deliver, but it is likely to collapse when there is a crisis, and any form of progress or velocity is at best an illusion.

Stability, meanwhile, cannot be ensured without availability – in the broadest sense of the work. Imagine a cyberattack, where it is DDOS, a data breach, or ransomware. These attacks might be classified as security issues, but the impact of them is lack of availability of that application. Therefore, availability needs to be an organization's primary focus, because operations are often tracked by mean time to failure and mean time to recover.

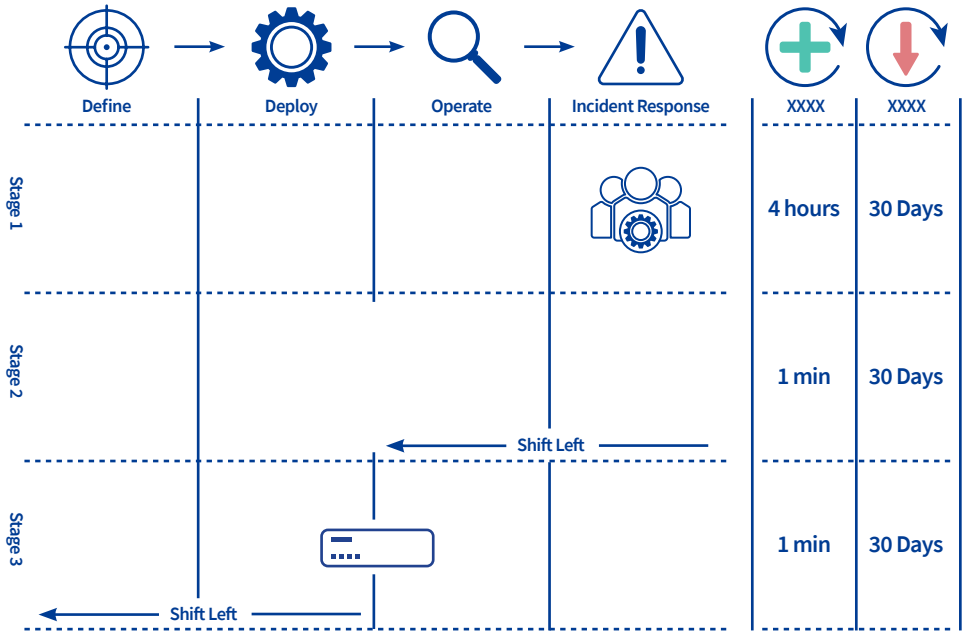
If some simple changes are made, in terms of how applications are deployed and operated, mean time to recover can be dramatically reduced. This means that, even when inevitable problems crop up, they can be mitigated very, very quickly from a user perspective.

Breaking the Vicious Cycle

There is no **Velocity** without **Stability**

There is no **Stability** without **Velocity**

Failures can be compex, but **Recovery** is simple

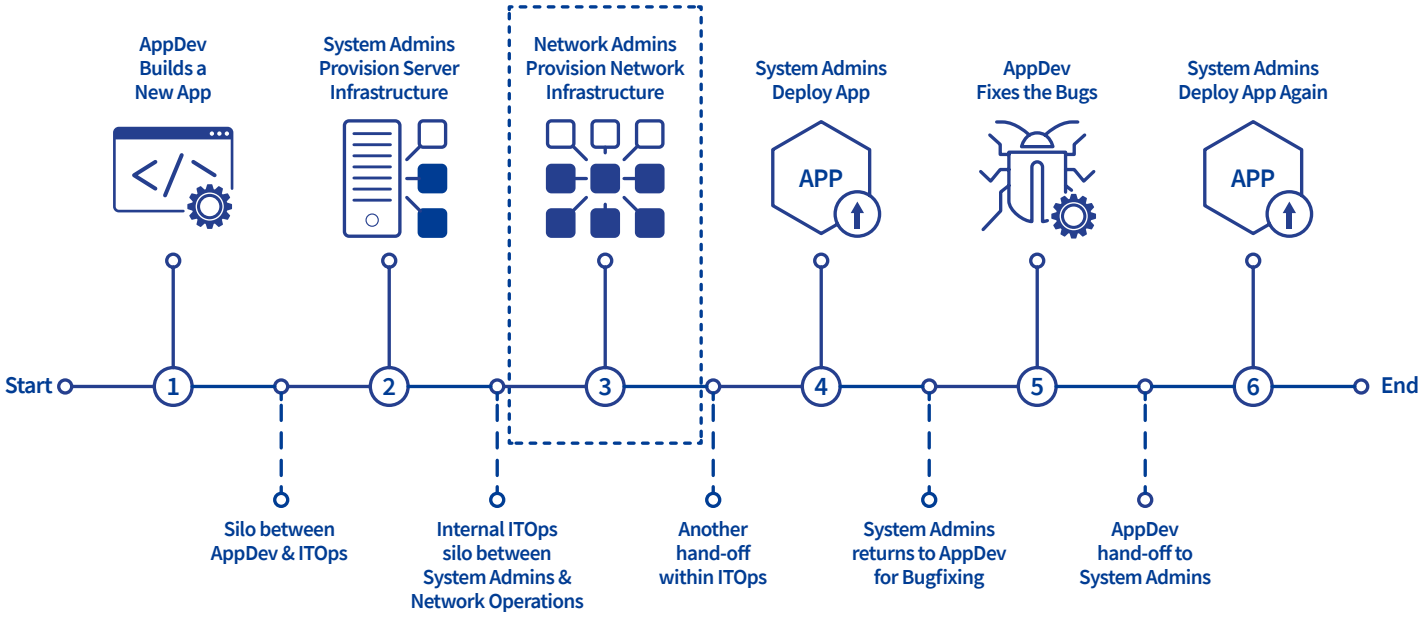


Controlled Deployments

A large percentage of outages within an organization are caused by bad changes, or changes that have a negative impact or an unexpected consequence. In light of this, controlling deployments and changes is key. In the case of an in-house application, everything from design, development to product, and getting it to customers requires different types of teams.

In this case, lean operations such as DevOps aim to break down frictions and barriers. But often, network and network infrastructure are ignored because they are seen as purely there to provision an IP, put a load balance in front of it, and open the file or port.

The Key to Rapid Recovery



When an application team is actually doing a deployment or a change, the network team is not necessarily involved or informed, and they are not utilized proactively to help mitigate the risk of the application team making an error. Network infrastructure needs to be made part of the change process and the deployment process, using technologies like load balancing or global-site load balancing that are completely vendor agnostic because they all have some fundamental capabilities.

Every time a change is made, is should be treated as a deployment, even when the application is not being upgraded to a new version. It could just be a setting change on the application or the underlying infrastructure, but each change risks availability. If that goes wrong, a risk needs to be mitigated, and there are some controlled deployment patterns that can be used to do that.

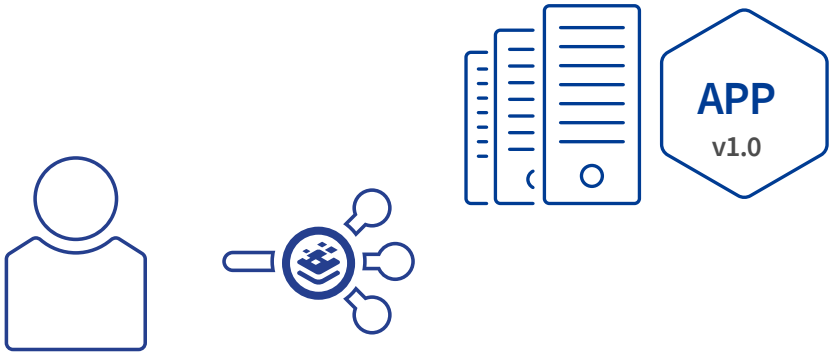
Deployment patterns

Controlled Deployments

All **Progress** requires **Change**

Every **Change** is a **Deployment**

Every **Deployment** risk **Availability**



Blue/Green or Red/Black Deployment Pattern

One approach is to put server load balancing in front of an application pool. Rather than doing in-place changes to a servers, building a whole new server pool, the only difference is the version of the application, or the version of the change or the configuration, and then, at load-balance level, switching between those pools. Users are redirected to one of the other. This gives an essentially instantaneous ability to flick users away from a failing version or configuration back to the pre-change application.

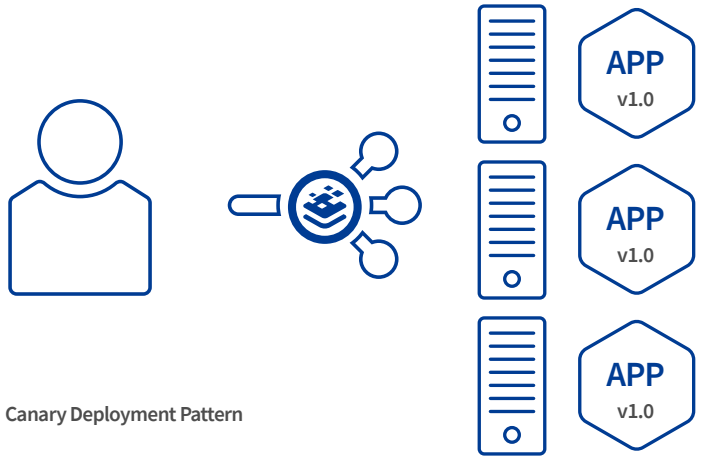
The other pattern is canary deployments, and this is requires less state-based applications. It needs to be modern applications, particularly, modern web applications and in-house applications an organization directly controls. This uses a pattern where changes are made one by one within a pool, with a certain percentage of users using the new version and others, the older version. And this is particularly good for finding performance issues, because it can slowly be staggered up to see how well it does on the new configuration scale.

Controlled Deployments

All **Progress** requires **Change**

Every **Change** is a **Deployment**

Every **Deployment** risk **Availability**



Canary Deployment Pattern

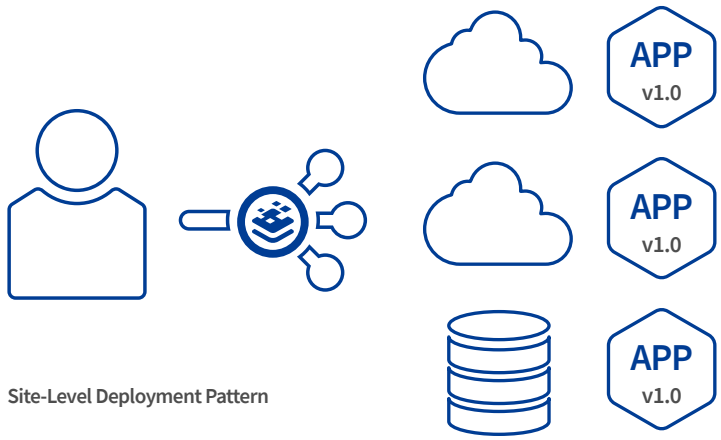
That pattern can be repeated with global-site load balancing which operates at a DNS level. An entire data center or cloud region can be upgraded, and its performance measured in terms of smooth operations and user experience, before rolling it out to other sites. Again, if there are problems at any point, users can be redirected to the known good version. What this means is that rather than having in place risky, stressful, mitigation strategies that try to resolve issues while users are complaining or applications are down, it allows an organization to quickly return service to users. Then, in a planned methodical way, the organization can internally find out what went wrong, but without pressure or escalations, which frees up time and energy to bring more stability and reduce failures in the systems.

Controlled Deployments

All **Progress** requires **Change**

Every **Change** is a **Deployment**

Every **Deployment** risk **Availability**



Site-Level Deployment Pattern

Measure the Progress

When changes are made, an organization should try it in one team or one application, and measure the change in outages, in availability and other metrics. This proves the value of the change, because change is something many people resist.

Communicating the Impact

Different silos and different functional groups should be made to talk to each other, and the value of the change should be communicated to business stakeholders to garner support for broader changes and re-investments. Such communication can really help bring that stakeholder engagement to another level, through showing how IT has allowed the organization to save money, save down time, and increase revenue.

CONTACT US



More Kemp Resources at **kemp.ax**