

Future-proofing your identity management system

...

beyond technology

John Jones
December 2011

Abstract

Many IAM systems fail too early or worse, simply never reach their potential; their lifespans are much too short. The factors that determine success or failure are relatively common and (we believe) can be easily identified.

This paper explores the most common reasons for success or failure and provides some advice on how an organisation can identify its own areas of weakness and work towards a healthy and useful system.

The paper explores two key stages in the IAM system's life: establishing the initial system, and evolving the system.

While the industry as a whole has a wealth of information on governance, opportunity and risk management, IAM maturity models, ITIL, CoBIT and CMMI, organisations need to be reasonably mature to take advantage of these.

This article takes a more pragmatic approach and is intended to be useful to those directly involved in the system's day-to-day evolution.

Introduction

About a year ago, we started working on a methodology to assist in Sun-to-Oracle IAM migrations.

This methodology examines the current state of a system, the desired target state and the organisation's capabilities for moving from the one to the other.

This information can then be used to work out the best migration strategy and its associated project plan. The methodology asks a lot of questions, about technology but also about retained knowledge, training requirements, and operational changes. It takes a complete view of the problem.

We observed that organisations that have a very good understanding of the current system, have technical fluency, access to good people and a clear sense of purpose will do well. Hardly surprising!

While the approach was developed for product migration, parts of the capability assessment are clearly applicable to the day-to-day running of the system as a whole.

More specifically, we have observed that the way in which an organisation deals with significant events such as patching, development and hardware upgrades, has a direct bearing on the system's overall health and longevity.

Paradoxically most of these events should move the system into a healthier, more useful state, but they often cause instability, broken functionality and increased frustration.

The message here is clear: deal with these events successfully and the system will go from strength to strength; deal with them badly and the system will weaken and die.

An organisation that is serious about extending the lifespan of their IAM system should examine both their overall organisational approach and the way in which they deal with change events. Both of these areas are discussed below.

Organisational Approach

The following section outlines some of the common problems that arise during the establishment of an IAM system. It does not deal with the obvious issues of funding and management sponsorship - we assume these are well understood and a given.

While many organisations will have already had at least one attempt at building an IAM system, and may feel that they never wish to do so again, it's never too late to change and to push for an approach that does produce a solution that is healthy and which delivers to specifications.

Many of the types of problems that may have been experienced have been seen previously in other integration style projects. They appear to arise for new and complex IT domains for which there is little or no subject matter expertise within the organisation.

Simply put, people don't know what they're doing.

Why are we doing this?

Possibly the most common area of confusion is the system's purpose. What is it for?

Why are we doing this?

Why is this important?

What's wrong with just installing the product, adding HR to AD provisioning, self-service in Phase 2 and then in Phase 3 add the other 200 odd systems?

What's wrong is that people end up with quite different expectations of what the IAM system is for - that is, what it will do for *them*. They become frustrated and disappointed and the system as a whole gets a bad name.

You don't need that and it's easily avoidable.

A true story: an organisation implemented an IDM system, which provisioned users from HR into Active Directory. On talking to the operations team, they voiced their dismay that all this time and effort had been spent to streamline the provisioning of users into AD because, while the on-boarding process did include AD provisioning, it was already the shortest step in getting someone on board. The whole exercise had done nothing to speed up the on-boarding process! What a waste!

The problem here is not that they built the wrong thing - they didn't. But the long-term purpose of the system was not communicated clearly. It was there to gain control over identities in general and that had to be done in steps over time. No one told the operations team that. They hadn't been shown the complete picture.

Building an IAM system is a significant undertaking often fraught with danger. You have to make sure that everyone understands what you are trying to do. Ideally you should have a clear and simple declaration of the overall purpose.

Here's an example:

"We are building an identity and access management system to ensure that the right people get access to the right things at the right time. We also need to be able to prove that this is true.

Why? You know all those contractors; call centres in India; partners in the USA? Well we lost track of who has access to what years ago! Last time we checked we had dead people accessing the Accounts Payable system! That might be right, but we don't know."

Clearly the devil is in the detail. Working out who the right people are, what they should have access to and being able to prove it is where the interesting work is. However, until you can do that, you don't really have anything like a complete IAM system.

What is "the system"?

Another common source of problems for many organisations is the wide variety of ideas about what "the system" should be.

For the CIO it's an IDM product, for a developer it's something that must be rewritten from scratch, for the project manager it's what they have to deliver, and for the help-desk it's a way of getting users into AD.

Why is this a problem? It's a problem because chaos is expensive and time consuming: the CIO will buy a product over lunch, the developer will re-write it, the project manager will deliver it a year late and the help desk persists in using the old "back door" method because they prefer it!

The result is a bunch of people being busy buying products, delivering technology and making life difficult for the people at the help-desk. Then, after a year of pain and suffering the vendor shows up and tries to sell you an upgrade!

The reason this confusion happens is because no one provided a clear idea of what needs to be done to achieve your IAM aims.

Again this is an easily avoidable problem. You must be clear about what the complete set of people, processes and technology should be to support your organisation's agreed purpose for IAM.

What we see is that almost all organisations start by buying a product. This technology focus leads to a belief that once it's in and running, you'll have a completed IAM system! Not true: you will have an expensive piece of technology generating heat in a data centre somewhere, not a solution that can realise your IAM aims.

Creating "the system" is a large undertaking. It must include all parties involved in managing the life cycle of the users such as HR, the business system owners, and it must involve the users themselves. You will need to become familiar with strange new terminology such as Attestation; you will come to understand the difference between roles and entitlements.

All this will take time, but it must be done.

How do I know my "system" is working?

By working I don't mean: "are the little red lights flashing on the server". I mean, is the system delivering on your IAM objectives?

Truthfully, very few organisations have a clue as to whether or not their IAM system is making things better or worse. Most don't have a good feel for what "better" or "worse" would mean for them. Is that the number of unidentified IDs in the system? Is it the number of IDs we are sure have the right set of entitlements? What is it?

Why is this important? Because without some clear idea of how things are going, it is difficult to tell if all the time and money being spent on building and running the system has been worthwhile. Without this, more esoteric aims such as IAM governance are simply not possible - you cannot tell if the "right people and doing the right things at the right time".

This is such a major problem that there is whole category of products sold to address these needs. These are the GRC and Identity Analytics products, essentially products that shine a bright light under the kitchen sink of your IT systems so you can see the cockroaches scurrying to and fro.

These systems can take a snapshot of all the important systems and their identities, associate them with real people and systems, determine if they should be there or not, who their managers are, who owns the systems they are accessing, and if they should have the entitlements they do. What's left over is a sticky residue of identities and entitlements that needs to be scrubbed from the system. For many organisations, success means reducing this sticky residue to a minimum. That's how it gets measured.

It's worth mentioning that many organisations today are starting their IAM journey with the implementation of these systems first. Some organisations we know are starting afresh with this approach and abandoning their initial attempts altogether.

Who owns this system anyway?

Because of the integration nature of the system, ownership and responsibility take many different forms. There are operational owners, target system owners, Identity Manager (as an application) owners, non-production environment owners, directory server owners, and the list goes on.

Unsurprisingly this is a common area of frustration for many organisations. As a result, issues arise when something needs to be done. For example, the vendor releases a service pack. You determine that it should be applied but it includes some functional changes - some old functionality has been deprecated. An impact analysis of the code changes needs to be conducted and regression testing will need to take place. Is this all covered by the operations team? Will they want to pay for the development work and functional regression testing? Probably not.

A lack of clarity around ownership is often made worse when the organisation has an outsourced model. You would think that it would help as surely as the contracts and SLAs will make things clearer! Not so. Not so at all! The problem expands rapidly with the number of outsourced parties. It adds substantially to the cost of any integration project and has a habit of taking much longer to get to production.

This isn't really an IAM specific problem but is a more general integration delivery problem. The difference in the IAM world is that if you are relatively new to this field, the various scenarios will not have been thought out or worked through in advance.

Our suggestion here is to run through the various scenarios that will affect your IAM system and act them out. Each of these scenarios will require people to do work, approve changes, find people and pay money. You should be able to identify those responsible for these things in advance. It sounds easy, but it's not. Sketch out the scenarios and fill in the gaps then get the various parties to agree.

How? Why do I need a rocket scientist to install my IDM products?

Many organisations spend almost all their "IDM project time" trying to build a technology solution. Many have done this more than once. If you're new to this field, these products appear complex and strange, are almost impossible to get working, appear to have very little of what you want, and a lot of stuff you've never heard of.

Why is that? Regardless of the vendor, none of these products is perfect - they are inherently complex integration solutions that are constantly evolving to accommodate new IAM ideas and technologies. If you're struggling it's most likely that you don't have the right people doing the work. That's not to say you don't have smart people - you probably do, but if they don't do this stuff for a living, it will be a demoralising uphill struggle.

(To be honest, it's an uphill struggle for many people who do do this for a living, but they stand a much better chance of knowing where the potholes are and what the product is good and bad at. They are also likely to be much more realistic about project estimates and better at getting sense out of the vendor. All these things should add up to significant advantage.)

The other common problem that organisations face is that they discover late on that the product does not support their requirements. This is a really bad place to be and there is rarely any way back! The brave will rework their requirements but many battle on regardless, spending more and more money until the project comes to a sad and disappointing end. This is frighteningly common for organisations that are on their first iteration of IAM and try to go it alone.

Again the message here is that you don't need a rocket scientist but you *do* need people who have done this before. They will need to be brave enough to tell you that you're mad - in a tactful, consultative manner.

So you did use an IAM systems integrator and that wasn't great either?

Unfortunately this experience is common too. The simple truth is that the technology design and deployment should be a relatively straightforward exercise for most common usage models.

So choose an organisation that is really good at delivering IAM integration solutions. Seek out other organisations like yourselves who have been through this and find an IAM delivery organisation that they rave about.

Don't settle for mediocre - it will come back to bite you. Don't make the technology deployment your main IAM problem - there is so much more your organisation should be doing to build "the system".

As a simple rule, 80% of the time should be spent on designing process, workflows, the information architecture, approval processes, information custodian models and all those things that are required for the system as a whole to work. The other 20% is for the technology deployment.

Handling life-cycle events

We've stated that the system really needs to include the people, process and technology that will evolve over time to support your IAM objectives.

The IAM system will grow, take on more systems and more identities; it will need patching and product upgrades; it will change its functionality through the extension of workflows and the addition of systems; it will migrate onto new hardware - it lives!

Each change event is a challenge that causes stress to the system but also represents an opportunity to gain strength and resilience. These events are often an opportunity for the system to grow and provide more value to the business.

Understanding these events and your ability to cope with them is absolutely crucial to the long-term success of your IAM system. The difference between organisations who do this well and those that don't is so clearly obvious that

you wonder why the difference exists at all. But much like a dysfunctional family, it's not so easy to fix.

We'll explore some of the key capabilities needed to deal with common change events. The capabilities will look obvious to most, and that's because they are. The real challenge is working out how you gain those capabilities - this is rarely a question of money.

Capability: Understanding

If your organisation has a good understanding of its current system it will be well placed to assess the effects of various events as they arise.

This capability is one of the most important determinants of long-term success. A loss of overall understanding through staff or partner loss will put the organisation as a whole in a very poor position to tackle almost any change event that might arise thereafter.

Here's a candidate set of questions you can use to get a feel for your capability in this area. Get your team in a room and get them to:

- Draw the system on a whiteboard and describe how it works
- Describe the overall end-to-end business process flow and the corresponding information architecture
- Explain the technical mechanisms such as load balancing and fail-over, replication, back up and restore
- Create some simple scenarios such as a patch release and work-flow change and see how quickly and easily they are able to determine what the impact will be, what changes need to be made and how long they will take.

If your questions are met with stunned silence, start to panic!

Capability: Information quality and sharing

If your organisation has good quality documentation that is up-to-date, along with a strong information sharing culture, you will be well placed to take on new staff, engage external parties for major work, move the system to new platforms, and extend the system in the future. Good, clear documentation is one of the most important and long-lasting aspects of the IAM system and should outlive almost all major events such as product cross-grades or the engagement of a new support vendor.

To get an idea of how well you fare in this area, see if you can lay your hands on documents that describe the following:

- A (business architecture) document that states what your IAM system is for, who owns it, describes the management structure and outlines a plan for the future
- A (derived requirements) document that captures all the use-cases and the business process
- A document that describes the identity life cycle for each group of identities, the approvals required, their default entitlements and any other basic information that described the life cycle of identities through the system

- The information, technical and application architectures. *Note that high level documents that clearly describe how the system works are much more useful than detailed design and deployment guides.* They remain relevant for much longer and provide a more complete picture of the system as a whole.

Capability: Delivery Team

If your organisation has a clear ability to deliver IAM projects with confidence and certainty it will be well placed to extend the system over time. If this capability is not available, the system's growth is likely to be very limited indeed.

To get an idea of how well you fare in this area, see if you can put names to the following team members and be honest about their abilities:

- A person (architect) who is able to design extensions to the system while maintaining the overall system integrity
- A person (architect) who understands what functionality is available within your products and knows how best to use it
- A developer(s) who has a good working knowledge of the current system and the products they use.

You will also need a software development life-cycle that has been tried and tested in the delivery of similar projects on your system.

Capability: Functional Testing

If your organisation can easily conduct functional testing it will be well placed to extend and change the system with confidence and certainty. An organisation that is not able to do this is likely to suffer production instabilities, rollbacks and the expensive activity of debugging in production.

Remarkably, many organisations get away with almost no capability in this area. So while we would regard this as best practice and certainly something to strive for, a company with a good understanding of its current system and a strong delivery team can often get away with a relatively weak testing capability.

The following should be regarded as a bare minimum - do you have these?

- A complete and up-to-date set of functional test cases
- A complete set of unit tests for system testing
- Automated regression tests.

Capability: Integration Testing

Due to the integration nature of IAM deployments, integration testing plays a significant role in an organisation's ability to push past more than a very small handful of target systems.

Do you have these in place?

- An SIT environment that includes the complete application ecosystem?

- A complete set of integration tests?

Capability: Non-functional Testing

If you have a large, highly available, highly secure system then you simply have to be able to conduct non-functional testing.

You will need this if you want to grow the system knowing that the underlying infrastructure has sufficient capacity, supports fail-over, and is secure and reliable. Without this, you are likely to struggle to maintain a reliable and available production environment. Poor performance, capacity issues and brittle production environments are a common result.

Do you have the following?

- A production support environment where production issues can be faithfully reproduced and where pre-production releases can be faithfully tested
- The ability to conduct non-functional testing of load and performance, high availability tests such as fail-over, back-up and restore procedures, etc.

If you only have a small in-house system then it's not a major determinant of your long-term success.

Capability: Production Manageability

How easy is it to manage the production system? Do you know what's happening under the covers? Are you confident that the DR system works? That the backups can be restored?

If you have good control of the production environment and can manage the day-to-day activities without being worried about causing unplanned outages then you're in a strong position to tackle change events and expand the system. If your capability in this area is weak, you'll eventually experience an inability to deal with production issues in a timely manner, and you'll see a drift away from reliability that is accompanied by a lack of faith in the system as a whole and a general reluctance to extend the system.

How do you fair with the following questions?

- Are you able to conduct most production changes without a business outage?
- Are you confident that DR fail-over works?
- Are you confident that backups can be restored?
- Do you know what's going on? Are all the systems monitored for faults: hardware, applications, network, etc. Are the logs monitored for errors and warning?

Clearly you could extend this list - the intent here however is to provide relatively simple sanity check on your current ability to grow your system over time and continue to provide business benefit.

END

To find out more about our IAM capability and experience, contact us:
email, at info@qubitconsulting.com;
or call me on +61 404 815 874
or at www.qubitconsulting.com.

Copyright of Qubit Consulting Pty Limited, PO Box 1157, Newport Beach,
NSW 2106, Australia. This paper was written in December 2011.