

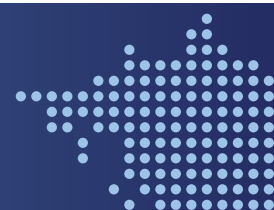
The Keys to Success in Core-Banking Transformations



axxiome
proven global expertise

Contents

1.	Executive Summary	1
1.1.	Purpose of this paper.....	2
1.2	Theme 1: Good preparation	3
1.2.1	Executive support and vision	4
1.2.2	Governance and Roles	5
1.2.3	Project Method	5
1.2.4	Accuracy and simplicity of scope	6
1.2.5	Importance of Planning: know the Critical Path	6
1.3	Theme 2: Project Discipline	7
1.3.1	Architectural 'steal thread'	7
1.3.2	Multi-disciplinary teams	8
1.3.4	Tooling and documentation	8
1.3.5	Team Qualities	9
1.3.6	Testing	9
1.3.7	Migration Planning and Approach	9
1.3.8	Importance of Environment and Release Management	10
2.	Conclusion	11



1. Executive Summary

“Shallow men believe in luck. Strong men believe in cause and effect.”
— Ralph Waldo Emerson

When empires fall, the **effects** are magnified as interdependencies are impacted or demolished. However, it is probable that the **causes** of the fall were building for some time and were suppressed or ignored as a result of inertia, resulting in a disproportionate impact when the status quo is no longer sustainable.

As with empires, so too with countries, industries and companies.

There is a tipping point where the natural resistance to change is overcome by the forces of change. With change inevitable, the question then becomes how will that change manifest – will it be revolution caused by too many ruptures, an evolution to a new state or a slow decline to oblivion?

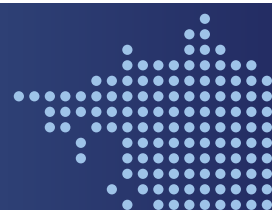
Ideally, an entity will be managed peacefully to a new state of normality, but will the new ‘normal’ be able to incorporate enough flexibility to continue to adapt to changes in the external environment, or will the whole process be doomed to repeat itself in years to come?

On such a tide is the banking industry now afloat. Having coasted on calm waters for decades, the seas are suddenly stormy. With rapidly changing environment in terms of customer expectations, technologies, competitors and regulators, it is a hard time to be captain of the ship.

Facing adversity brings the realisation that the vessel was designed for another era, and the engines are not firing on all cylinders. In some cases, the streamlined modern ship turns out to be a rusty old freighter. As a result, much of the day-to-day focus means working in the engine room, greasing and patching to address the creaking technology, integrating new controls with expensive custom fixes and concentrating disproportionately upon the heavy demands for regulation in the wake of the crisis. While the focus of existing banks is thus engaged, nimble new competitors with better technology arise to present a more streamlined face to customers who are themselves more tech-savvy. Worse again, competition is no longer just coming from the traditional competitors, but is now coming from all directions: from different industries, from niche players and from former suppliers.

Nevertheless, the same technology change that seems to present intractable problems is likely also to lead to a renaissance in retail banking. Standardisation of banking APIs, service offerings, cloud based applications and processes orchestrated by digital platforms give us the ability to develop genuine customer-focussed organisations and become more adaptable to change.

First, though, it must be implemented and this is not always easy. It is an axiom that new technology is only as useful as the business purpose to which it is deployed and a major factor in the success is proper preparation. Within that preparation, a major factor is understanding the business purpose in the first place, articulating it well and using the right people and methods to achieve the desired outcomes. For example, customer focus does not get built in if it is not designed in. In this paper, we set out the experience that we have gained in terms of preparing for success in core bank transformations.



Corebank transformation as the baseline of change

A new wave of finance technology is sweeping the industry, offering more and better engagement with customers. It is our belief that this is triggering a major re-evaluation of the capabilities of financial technology to deliver transformation of business models, and a willingness to invest in real change. It is also our belief that new financial technology cannot truly optimise delivery of products and services until the core software engines that contain the product & services offerings are both stable and transparent enough for regulators and have become highly adaptable to allow for rapid change.

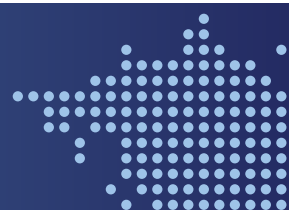
As the industry moves towards digitization, a great many innovations will turn banks (and non-banks) towards a new and different types of banking business.

Within this set of innovations and renewals, the renewal of a core bank engine will not set hearts racing, nor does it create a new paradigm for a financial services organisation. Nevertheless, it is not possible to create a flexible, digital organisation without a flexible back-end powering it. Or put another way, an elegant yacht cannot put to sea without a high performing engine. As with building a new house, the architecture may be visually stunning and the space can be user friendly, but the house always must rest upon strong foundations.

1.1. Purpose of this paper

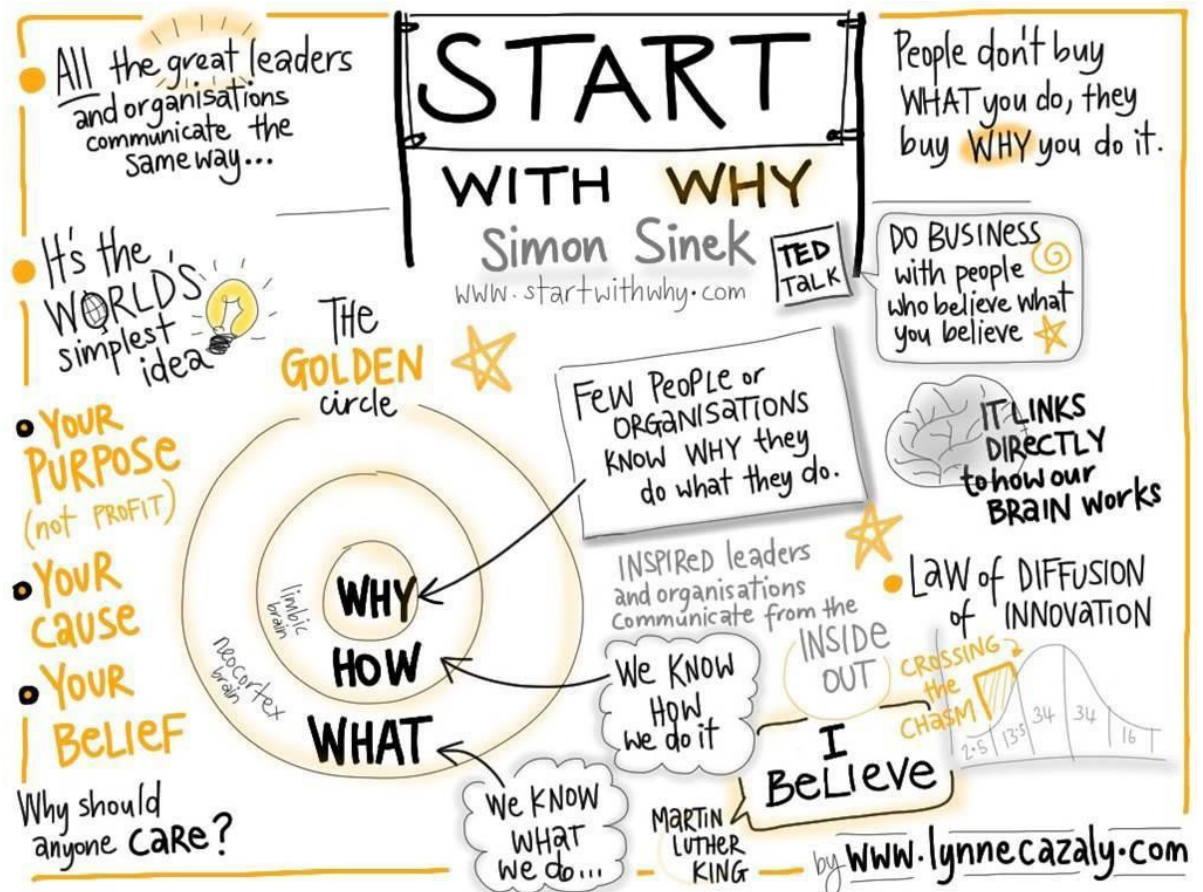
Armed with the right information an organisation can make informed decisions. We aim to provide some pointers on how to undertake an implementation of core banking applications, to improve return on investment whilst reducing complexity, risk and delivery timescales. Of the number of key areas which will require particular attention in this section, we will address those that could attract the highest impact, and we trust that this information here will help to prevent flawed decision making. . We see the same factors emerge time and again, regardless of geography, and regardless of software and platform technology. We can categorise these factors into two themes:

- Good Preparation
- Good Programme Discipline



1.2. Theme 1: Good preparation

In this section, we will explore many of the factors that are required (or at the very least, strongly recommended) before a core banking implementation can even take place.



1.2.1. Executive support and vision

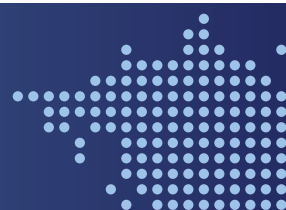
Preparing for success is everything, the alpha and omega of programmes. Priority one in preparation is the support of the bank's board and their alignment with the vision of the transformation programme. This will enable it to be successful. Working out why the transformation is necessary provides the rationale that backs the vision. With the right executive support, communications are improved, issues are speedily remedied, decision cycles shortened and red tape is cut. This helps to bring design closer to the corporate vision.

One of the major issues on every transformation programme is articulation of scope derived from the vision of the transformation programme, which must then be turned into more specific business requirements. When selecting and implementing a new application suite, the financial institution needs to understand what they are trying to achieve in the long term and have that vision articulated and agreed amongst executives. Later, that same vision requires continuous promulgation amongst programme management and members and their interests too must align with the vision.

Thereafter, when it comes to selecting an application suite, the organisation needs to understand how it works and what its capabilities are. These will be different to the existing system and therefore the design of the future system must focus upon creating the required business outputs. Without realising it, programme members will try to rebuild the existing system's look and feel along with current procedures. It is natural, but it is bad business practice, and can only be countered by thorough design bringing together vision and application capabilities.

An organisation needs to take time to see exactly what the application suite has to offer and how to gain the maximum benefit from implementing it. When selecting the platform and applications required, there are some rules of thumb that we have developed over the years

- Don't rush into it -investing some time in Proof of Concept exercises will pay back significantly when the program begins.
- Spend time to analyse existing processes and question if there is a better way to achieve results will assist in the success of the program.
- Spend time to invest in training programs for key personnel, who in turn will drive from the organisation's side.
- Spend time to consider the elements around integration – for example, rather than considering it as a technical solution, determine how the business processes will be modified and orchestrated in order to achieve maximum automation and therefore efficiencies.
- Spend time to ensuring continuity between selection and implementation. This highly recommended to ensure that the rationale for selection and the required outcomes permeate the programme. We have seen some instances where these activities have been kept separate which does not help to align interests between parties.



1.2.2. Governance and Roles

At every level, the governance of a programme is the difference between success and failure (or more likely, a very expensive compromise in the middle of those two extremes). Preparing the structure of the programme to take account of good project governance and solution governance is a critical success factor.

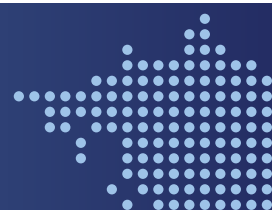
- Programme Governance is a science in itself, and usually results in a fine line requiring a balance between too much and too little governance. Applying too much governance can mean slowing down a programme with reporting overheads or decision-avoiding committees, boards and actions that will gradually drain the energy of the project. Too little governance and the project can lose cohesion and splinter into self-protecting cells of subject matter experts working to their own agenda.
- Solution Governance: In our experience, there is a very real likelihood that the holistic view of the solution is not considered – this is particularly likely in a big programme, where a core banking application is implemented by the consultancy arm of the software company. It is therefore easy for the application to become a ‘black box’, neither understood nor interpreted by the rest of the programme. As a result, impact assessments are not done at the right time or there is incomplete participation in solution design detail. This inevitably leads to multiple change requests later.
- Roles and Responsibilities: A well-defined RACI matrix can work wonders. The real value of this type of document is that it forces all parties to sit down at inception, and work out precisely what their programme roles are and how they are aligned to the organisation’s expectations. Furthermore, taking account of the resources on the programme through well-defined roles and responsibilities is another critical success factor. Here, we mean going a little deeper than giving people the right titles: defining the individual accountabilities and responsibilities helps to support those individuals to be comfortable about their decision making, and to be comfortable knowing when they should escalate an issue or a decision.

The involvement of the bank’s key users and business analysts is a must for the success of the program but also for the ongoing operation of the new solution going forwards.

1.2.3. Project Method

One of the first objectives of the programme must be the definition and sign off of a means of co-ordinating the activities and tasks to come. This will come to form the Project Method and an agreed Method will drive the work execution and documentation in a way that should make life easier for the programme. It is imperative to avoid unnecessarily complex methodologies and redundant documentation. There should be a clear critical path along the method and, in most cases, deliverables are only created when there is a subsequent consumer which creates efficiencies through re-use and helps focus on why the deliverables are being created. For example, business processes are designed so that functional, technical and integration resources can use them as an input to their deliverables and, ultimately, these inputs will be available to end users for operational documentation.

Furthermore, the method should be understood by all project members and should be the foundation for the creation of detailed project plans.



1.2.4. Accuracy and simplicity of scope

The more accurate the scope, the easier it is to avoid costly and unnecessary changes. Our advice is to spend enough time to have a very clear scope before engaging project teams. Furthermore, divide scope into multiple releases that keep each phase of the project 'clean'. Use each phase to build upon previous phases whilst delivering realistic and tangible business outcomes at each phase that make the programme progress more visible. Better still if each phase can deliver business value.

We have seen more than one organisation selecting a replacement platform where those involved in the initial scope study had no knowledge of the application they were selecting. Each time, this approach led to complex requirements being described, other requirements seen as gaps (when they weren't), and in some cases a complete omission of important requirements.

By signing these requirements off without robust challenges (and in some cases without reference to the overall desired outcomes), a large number of change requests subsequently appear. In every case that results in seriously impeded progress by adding to complexity, increased build time and higher costs. In the end, no matter what the contractual situation, it is the organisation that carries the risks and costs.

At the risk of repetition, the important principle is to understand and align the business vision with an understanding of the target system. It is good practice, and never a wasted investment, to educate those involved before they embark on design and detailed requirements definition.

1.2.5. Importance of Planning: know the Critical Path

For a core banking transformation, integrated plans are key - nothing should be undertaken that does not have a plan. We would also underscore that all plans must be understood and signed off by all parties involved, to ensure everybody is committed towards their realisation.

It is important to ensure that there are regular and accurate updates, especially after change control approvals, and it is equally important to have a strong and clear Change Control process that measures impacts of Change Requests in project plan and business case terms.

Also key when planning in a complex architectural environment (as it may be any bank's landscape) is to consider the projects that will be running or going live on other areas of the landscape during the duration of the transformation program. These could impact the programme in many ways and they must be considered in the overall plan.

Note: The services layer integration and the necessary changes to existing systems and applications are often underestimated and take longer than the configuration of the new application suite. This impacts the key sub-phases that make up the presumed critical path and the actual Go-Live date. An important lesson from projects is that the programme team needs to ensure that the critical path is understood and all parties are aligned to it.



1.3. Theme 2: Project Discipline

1.3.1. Architectural 'steel thread'

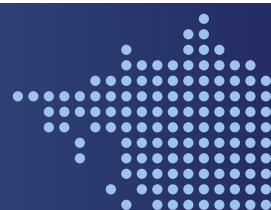
Spend as much time as possible creating clarity about the overall solution.

A full understanding of the architectural scope of the end to end solution, (e.g. business process, dependent applications, integration points and data objects) is vital, so that it is known where functions will be 'realised' and how. This also helps with understanding the application and data rationalisation needs. It also helps to counter the package-centric approach, where some applications are treated as a black box without understanding the wider context.

An important area of architectural focus is of course the integration with an existing application landscape. In previous projects, we have seen clients, system integrators and application areas address the integration needs to their own application areas only. Without central co-ordination, this leads to huge duplication and confusion. In the end, it is the organisation that pays for the significant rework required, with additional costs and delays to Go-Live.

1. The lesson drawn from previous projects sound obvious:
2. Ensure that one team or one manager is in charge of the integration effort
3. Ensure production of a single interface/enterprise services catalogue as part of the design linked to other key deliverables.
4. Ensure functional specifications for Source, Conversion and Target are signed off together

The 'steel thread' comes from being able to act as the guardians of the design, to ensure that all the participants can be



brought back into line when they veer off course. In this, the architects must act as though they are the content side of the project management team.

Another important lesson, validated through experience, is that resistance to change may also come from legacy system owners/experts assigned to the programme, as they may see some decision as threatening (whether to careers, internal empires or their position as 'experts'). This resistance often manifests at more senior levels where it may negatively influence the decision making process. It is therefore one of the key threats that the architectural 'steel thread' will be required to counter.

1.3.2. Multi-disciplinary teams

It is our strong belief that integrated teams (by which we mean multi-disciplinary roles inclusive of customer business representatives) bridge the traditional divide between business and technology resources. The reasons are manifold. Driving design and documentation through the prism of business process allows the customer to understand what is being discussed and delivered.

Having the business in close proximity allows application experts to define, prototype and deliver application configuration against the same understanding.

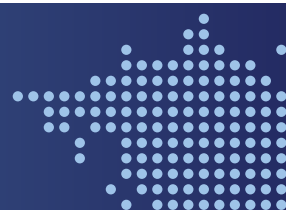
In addition having this close relationship promotes transparency, accuracy and knowledge to the customer community. To reiterate a point expanded elsewhere, there is a major programme and operational benefit to involving key users as soon as possible. This extends to providing them with training in the new platform as early as is possible in the programme – training is rarely wasted, whereas the hidden costs of inefficiencies through poor knowledge or staggered decision taking are insidious and project-threatening.

1.3.3. Requirements Traceability

Having a multidisciplinary team promotes the capture of reasonable requirements. Automating the management of those requirements is priceless. Capturing and linking requirements to subsequent deliverables such as process flows, system flows, configuration rationale and test cases means that an item captured once can evolve, can undergo a managed change and can be tested against its original articulation.

In practical terms, controlling and automating the requirements thread can be one of the biggest time and costs savers on a programme.

This requirements thread supports the 'steel thread' discussed in the architecture session.



1.3.4. Tooling and documentation

Tooling has often proved to be an issue when documenting programmes. The toolset should be determined at the outset and have strong standards and quality control imposed rigorously (possibly by a strong PMO).

The documentation generated during the build process will be there for a long time. Documentation created and captured during a programme is essentially the basis for the customer's knowledge retention and can be crucial for knowledge transition during and after the program. There exists any number of solutions to enhance good documentation but when working with a new application suite, it makes sense to understand its inherent capabilities, and adapt accordingly. Re-use of the same base set of documents for different purposes prevents duplication, re-creation or worse.

1.3.5. Team Qualities

A programme consists of a large team, within which there are multiple teams. A preferred approach is to constitute teams of resources that are known and trusted and hold both the experience required along with the desire for success. In this regard, when a team is running well and trusted, it does not have to be onsite any longer. Savings can be made without sacrificing quality.

For example, when an onsite team is constituted by strong and experienced blueprint designers, it can often be more economic and more productive to site the development staff offshore in development hubs where services and outputs can be shared (or re-used) amongst experienced staff. It is also worth checking that the experienced staff offered by suppliers are genuinely experienced and can be verified to be so.

Nevertheless, it is worth considering that some staff will also form an ideal liaison role between the client site and an offshore development hub. When considering off shore teams, it is advisable to define a clear Lead Role for each off shore hub. This person will be responsible for the hub's developments and will also often travel to the programme's onsite location to ensure the latest developments are clearly understood on their side. It's also proven to be a good practice to plan one trip for each of the hub's member to project onsite as this allows them to put faces to names and also experience the working atmosphere in order to replicate it in the hubs.

1.3.6. Testing

The key to testing is to have a good, clear strategy as soon as possible in the program. This helps the team understand what will be done on this, the Sign-Off pre-requisites (and what is the Sign-Off process) and what is the likely resourcing. There is one main principle that is often overlooked when testing: Testing is to FIND issues, not merely to go through an allotted set of test scripts as soon as possible. While this understanding may be embedded in an experienced testing team, it also needs to be normalised with additional resources drafted in to help (such as key users and business analysts). Based on this principle, any tester should explore deeper if he thinks there may be something wrong even when a script has been successful. Overall testing is to prevent issues in production where things are much more expensive and critical to fix, so a well planned, well-resourced Testing Phase is a good investment.

In terms of the team, we recommend that the Business should be in charge of producing the business test scenarios, and should also be responsible for the ultimate sign off of the solution and test phase. Test execution can be run by less experienced resources (albeit with a good working knowledge of the solution implemented) but test support is key to the success. There needs to be a very experienced team fixing any issues to ensure a quick turnaround and excellent quality. Testing must also verify the correctness of each Business Requirement, taking advantage of the traceability put in place. Additional effort may also be required by dependent areas to run multiple test cycles and scripts, defect fixes; and this should all be co-ordinated centrally - disparate ways of reporting and managing progress don't allow complete view across all components.

1.3.7. Migration Planning and Approach

If the programme encompasses a Migration effort, this has to be tackled as early as possible (albeit with a lean and experienced team) in the overall timeline:

- to help recognise issues upfront,
- to determine the strategy, approach and tool support required
- to drive the overall designing of the solution in a way that Migration can support,
- to identify contact points with the Solution team.
- to gather the source data from legacy platforms

It can also be very useful (if started early) to perform a data cleansing operation upon the legacy platforms before there is a migration effort and where there are sufficient legacy operatives available.

In terms of execution, however, there is usually a hard start required such as having the detailed design signed off for the new system, which will give a stable platform for designing the target datasets and business rules required for transformation.

1.3.8. Importance of Environment and Release Management

This is a function of the platform choice but must be considered to some extent. A cloud supplier can be expected to provide a greater degree of service provision 'out of the box' as part of the contract than an organisation planning to host in-house will have a greater degree of autonomy in managing this element.

Regardless of service provider, it is probable that multiple environments are required across all major systems in scope for Integration, cutover testing, data proving and Operational Acceptance testing. Our experience is that on most projects, there are not enough environments to allow for multiple testing strands.

Furthermore, we also recommend co-ordinating multiple technical fixes/patches for all defects and changes required across both SAP and non-SAP system, whilst maintaining code and configuration integrity across Verification and Production environments



It is worth investing early in tools that maintain consistency across a complex environment which can be implemented (and learnt) during a project, and then used to reduce the transition risk into operational running. For this, it is worth taking an early position on how the landscape will be run after the program, in terms of release strategy, continuous enhancements and application functionality. This brings us full circle to what the organisation is trying to achieve in terms of the business vision of the executive. Rapid deployment of new products and innovative functions will help to determine the logistics of environment and release management required to support this vision and make it real.

2. Conclusion

Before setting out to sea, a lot of preparation is required to ready the ship, test the engines, check the weather forecast and bring aboard sufficient provisions and experienced sailors who 'know the ropes' and are not learning to sail on the job. As well as running a tight ship during the voyage, a successful voyage is to a large degree a function of the preparation.

The same principles apply to ensuring that a core banking transformation will be delivered on time and on budget. There needs to be one captain, and that captain needs to know where the destination is, as well as having a good idea about how to get there. The members of the team need to understand and have enough experience to ensure that they can fulfil their tasks with the minimum outlay of effort. They need to have a common way of working, and that needs to be supported by the right toolset. If there are surprises along the way, then there needs to be mechanisms to handle those surprises with a minimum of disruption.

Our advice is, like a ship approaching harbour, use experienced navigators and pilots to plot the course, and ensure that in-house staff are part of the team. This will drastically decrease the risks intrinsically connected with a transformation initiative.

* * *

ABOUT AXXIOME

Axxiome is a global solutions provider for the financial services industry focusing on advisory in banking and insurance complexity reduction and analytics. Axxiome provides advanced SAP Banking and Insurance expertise for accelerating the implementation and integration of transformation initiatives. With these methods, products, and expertise, Axxiome helps financial institutions reach clarity from a business, technology, integration and portfolio perspective. Axxiome is committed to supporting the achievement of tangible business outcomes including: new business operating models, regulatory agility, increased efficiency, the transformation of customer channels, and reducing total cost of ownership for its clients.

To find out more about Axxiome visit:
www.axxiome.com

