Revitalize your legacy IT

Move to a scalable, agile platform by modernizing your legacy IT landscape

pwc

www.pwc.nl

Contents

Introduction

1. Introduction

Incumbent players, in any industry, have been pushed by competitors and changing client demands to innovate and stay ahead for as long as there have been companies. However, technology is driving change at an unprecedented pace. For example, BigTechs such as Apple or Amazon are entering all types of industries (e.g. Healthcare, Financial Services, etc.) posing a serious threat to established players. These platform-based companies have several distinct advantages: scalability, personalized offerings, fast time to market, and low costs due to economies of scale. Consequently, C-level executives are driving digital transformation agendas to get or stay ahead of the competition.

Driving a digital transformation end to end by re-thinking processes, structures, and technology can be a daunting task and requires additional effort when legacy comes into play.

Gartner predicts that "every dollar invested in digital business innovation through the end of 2020 will require enterprises to spend at least three times that to continuously modernize the legacy"[1]

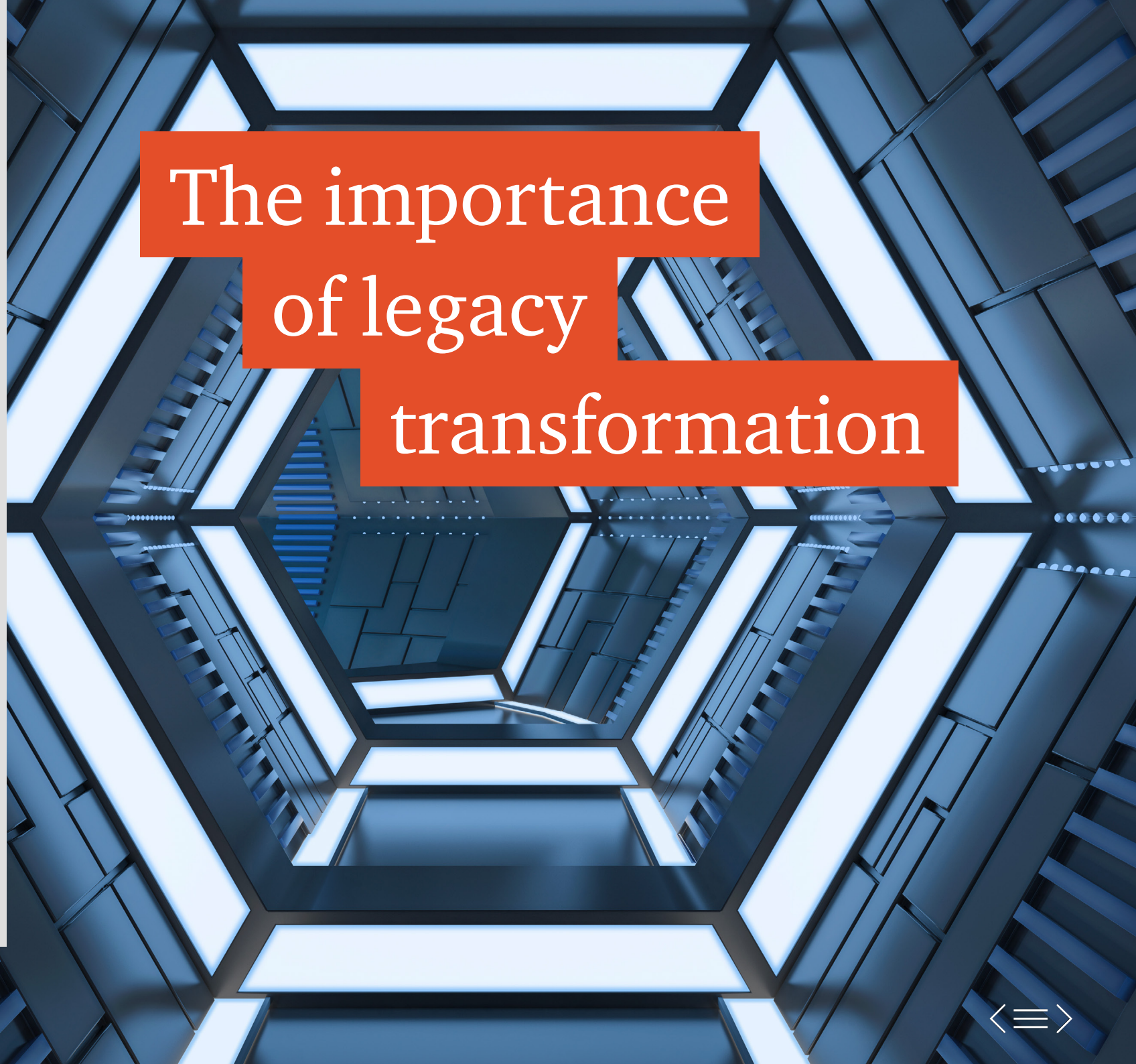
Having an adequate strategy in place for your legacy is therefore inevitable: whether to address your legacy IT or not is no longer the question, the real question is 'how'?

- How can legacy transformation enable your business goals?
- How can companies determine what needs to be done?

This whitepaper will provide:

- Insights into the problems linked to legacy IT and the complexities in coping with these information systems;
- An IT Maturity Model to assess your current IT estate;
- A list of solution patterns for your legacy information systems;
- An approach to start your transformation efforts.

[1] https://www.gartner.com/smarterwithgartner/7-options-to-modernize-legacy-systems/



The importance of legacy transformation

2. The importance of legacy transformation

2.1. Transforming your legacy is vital to your strategy

Addressing your legacy IT kills multiple birds with one stone: lowered costs, mitigated risks, and increased time-to-volume.

Replacing or modernizing legacy IT is not a new topic; however, it has become more important than ever in today's context of digital technologies, cloud-native development and increasing competitive pressure.

Digital transformations are strenuous, as these require your IT landscape to change with high speed. Customers expect services to be available instantly, anytime and anywhere, which conflicts with the nature of legacy systems that deliver new business functionality in long release cycles (up to several years). Strong customer service is becoming a competitive advantage, but data in legacy systems are often within silos. Moreover, legacy systems including middleware are often built with 100% proprietary code that were not designed to integrate with open source frameworks or 3rd party software components.

On top of this, legacy systems are vital to the business whereas knowledge of these systems has faded over time. Therefore, any modernization effort requires adequate consideration to avoid unnecessary impact and waste of investments.

...Because legacy:

Is expensive to maintain;

Creates data silos and

Is hard to integrate with new technologies; therefore,

Makes it hard for middle and backoffice to support the front, yet it

Is vital infrastructure holding critical data and

Replacement is hard asit temporarily introduces more complexity

Source: PwC analysis

2.2. What do we mean by legacy transformation?

2.2.1. When do we call an information system legacy?

Legacy is often confused with 'old systems'. Gartner's definition of legacy is: "An information system that may be based on outdated technologies, but is critical to day-to-day operations".

However, an old system can be very well developed and kept up-to-date with contemporary standards. Vice versa, you can argue that a relatively new system can be legacy at conception due to wrong decisions.

Furthermore, the issues related to legacy originate from a wider scope than just the technical aspects of an information system. Often the lack of active knowledge with the engineers and administrators is the main driver for labelling an application legacy.

PwC therefore defines legacy as an information system, including management & support processes, which is unable to fulfil one of more of the following criteria in relation to the business demands:

Agility
Stakeholders need a significantly higher degree of agility than what can be offered. Modern day business often requires information systems to provide (functional) updates in short cycles, of months, weeks or even days. Many information systems cannot meet that requirement, sometimes having a release cycle of only one new release every three years. This lack of agility can be a serious threat to business innovations when you fail to take a crucial first mover advantage.

An information system that lacks the required agility is therefore legacy, and should be modernized.

Costs
Operating costs are significantly lower when contemporary technologies and insights are used. Continuous innovation leads to more cost-efficient technologies. These developments often allow modern information systems to operate at a significantly lower cost base than older information systems. This puts the business at risk, as competitors applying these contemporary standards have a lower cost base.

An information system that is expected to have a lower cost base when using contemporary standards is therefore a legacy candidate. If the payback period for a modernization investment is shorter than the expected lifespan of the information system, than we label that information system as legacy.

Risk
Lack of active knowledge of information systems, with the exception of newly introduced technologies, is perceived as a serious risk for business continuity. When an organisation has such a knowledge deficiency, it may lack in one or more of the following domains: a) skilled resources, b) test scripts, c) implemented functional changes or d) documentation. We will provide more insight on this topic later. Insufficient active knowledge is a threat to the business, as this low resilience means the ability to respond adequately to a major incident is seriously hampered.

An operational information system that lacks active knowledge is regarded to be legacy.

2.2.2. The complexities of a legacy transformation endeavour

In many organisations, legacy information systems of course meet more than just one criterion, adding to the complexity of the modernization process. Multiple factors contribute to an information system meeting one or more criteria. The complexity of legacy transformation is underpinned by research which shows that "55% of Financial Services Companies cite Legacy systems as key barrier for digital transformation."[2]

[2] https://www.genpact.com/downloadable-content/hbr-full-report-how-financial-services-views-the-digital-agenda.pdf

There are several dimensions to the complexity of such transformations:

Organisational adaptability

- A different mind-set is required to develop and maintain (re)new(ed) IT capabilities compared to legacy. This may vary from use of different programming languages (e.g., from procedural to object oriented, different syntax, etc.) to different ways of working (e.g. waterfall to agile) and different technical platforms (e.g. mainframe to x86).

- Legacy systems often have decades of work built into millions of lines of code. As such, this code base often carries sentimental value to those who spent years developing these systems.

- Scarcity of knowledge is often an important facet of legacy IT. So, the capacity to change from a current situation is de facto limited.

- End-users need to be trained to use new software and processes which – pending the change – can be time consuming.

Corporate politics

- Depending on the size of your legacy estate, modernizing your IT will require long-term commitment from senior management, as it can affect the business from front to end.

- Legacy transformation programs were tried in the past and often failed due to typical reasons such as scope creep or changing requirements in replacement actions. More often than not, people are unwilling to commit to a long-term project which - in their mind - is high risk.

Technical complexity

- Most of the time, legacy systems in large corporates do not exist in isolation. Therefore, modernization requires an adequate roadmap taking into account best-fit solutions, prioritisation, sequencing, etc. In addition, a phased migration brings its own complexities that need to be sufficiently understood and managed.

- The millions of lines of code contained in legacy systems are often hard-coded business functionality using underlying logic which is no longer present.

- Legacy applications were created in times when code was developed to exploit the specific platform underneath the application, often making these applications platform dependent.

- Organisations too often default to a full greenfield replacement of a legacy estate. Such endeavours quite often fail (due to lack of knowledge on current system, scope creep, etc.) or lead to a loss of intellectual property (IP) that has been built up over a period of time.

2.3. Drivers for legacy transformation

According to Gartner, every dollar invested in a digital business innovation will require you to spend at least three times that amount to modernize your legacy IT continuously[3]. However, modernizing your legacy IT estate will help you realize certain drivers.

2.3.1. Agility increase

Quickly respond to market and regulatory demands and increase time-to-market and time-to-volume by introducing flexibility and modularity into your IT architecture. As mentioned, business applications that withstood the test of time are often monolithic, consist of 'spaghetti' code, and are updated without the latest development life cycle management techniques such as Continuous Integration/ Continuous Delivery (CI/CD) pipelines for automated release management. Consequently, IT has tremendous issues delivering the agility required by business today. In addition, several industries, like banking, and pension and insurance funds, need to cope with an increasing amount of regulatory pressure (e.g. PSD2, Basel and KYC.).

[3] https://www.gartner.com/smarterwithgartner/7-options-to-modernize-legacy-systems/

With cloud taking the centre stage, modernization efforts have taken a new turn by primarily migrating to cloud-native development environments on the public cloud on top of migrating from proprietary server platforms to commodity hardware and transforming languages like COBOL to Java.

2.3.2. Lower costs

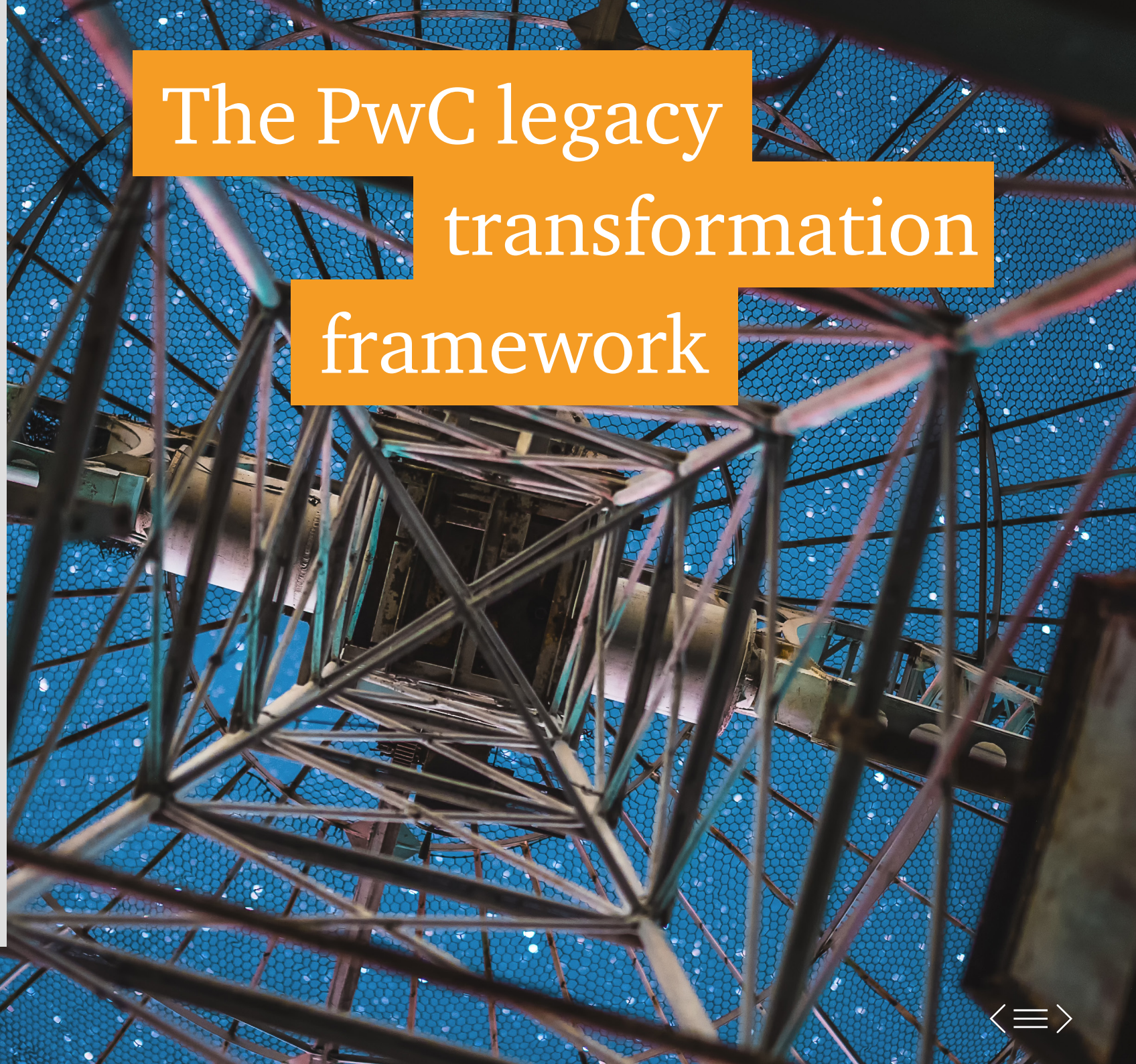
Lower your cost base by eliminating reliance on legacy as old vendor contracts have expensive pricing structures, license and support fees that are no longer in line with the current market or your current IT estate. Moreover, many legacy applications run on closed platforms (e.g. IBM Mainframe, HP NonStop, etc.) with a high fixed-cost component making the transition to a pay-per-usage OpEx business model for the IT organisation difficult. In addition, being so reliant on certain technologies might mean a vendor lock-in, forcing continuous renewal of these expensive contracts. Therefore, moving to commodity hardware or open-source software frees up budgets for investments in value added services. However, just moving legacy applications to cloud does not deliver the promised efficiency. Legacy applications are not designed to use cloud functionality, such as automated upscaling or outscaling. To use these key cloud characteristics, the legacy application requires modernization.

Next to this, the legacy technology often demands specialised skilled resources. At the same time, the number of available people decreases, resulting in higher wages and thus higher costs.

2.3.3. Risk mitigation

Mitigate your key man-dependency and other risks by modernizing your IT architecture. Many of the long-serving applications in an organisation are tightly-coupled (monolithic) applications with decades of critical business logic built into them. The knowledge of these applications is limited, tacit, or might even have left the company years ago. Furthermore, the knowledge present is usually not 'active' leading to potentially disastrous impact during critical incidents. In addition, these systems were developed using programming languages that are no longer common ground for modern engineers, such as COBOL, ALGOL or PL/1. Therefore, companies are dependent on a limited number of key resources, often sourced externally, to maintain applications crucial to business operations.

The PwC legacy transformation framework



3. The PwC legacy transformation framework

Now that the difficulties in legacy transformation have been established, in this Chapter we present the solution options. There is no one-size-fits-all solution for legacy modernization. We have developed a standardised framework consisting of an IT maturity model, a set of reusable solution patterns, and best practices for program delivery. To provide a best-fit solution to your problem it is important to understand

- the context of your problem: its most important drivers and state of your IT landscape;
- the intricacies of the various options available;
- the suitability of each option given the particular context;
- best practices for fail-safe execution.

To facilitate the process, the framework has standardised questionnaires, models, dashboards etc. that can be used by the project team.

3.1. IT Maturity Model

To support this endeavour, PwC has developed an IT Maturity Model to assess your current IT estate, and select the most appropriate measures for modernization.

The IT Maturity Model (Figure 1) quickly pinpoints the compelling blocking issues - that render an information system 'legacy' - within four focus areas: Active Knowledge, Application Life Cycle Management, Application Design, and Technology Configuration.

Figure 1: Maturity levels per focus area

Focus area	Level 1	Level 2	Level 3	Level 4	Level 5
Active Knowledge	No skilled resources	Skilled resources	Skilled resources + 1 additional	Skilled resources + 2 additional 4	Skilled resources + 3 additional 4
Application Life Cycle Management	Waterfall	Iterative	Agile	DevOps	SecDevOps
Application Design	Monolith	Decoupled	Loosely Coupled	Containers	Microservices
Technology Configuration	Closed System	Open System	Infrastructure as a Service	Platform as a Service	Serverless / Software as a Service

An organisation should always aim for maturity level 5 on Active Knowledge. However, for the other three focus areas, the maturity levels should be equal with a maximum difference of 1 level to prevent a mismatch between the organisation and its IT architecture.

These focus areas in the IT Maturity Model are based on the drivers for legacy modernization.

[4] See Active Knowledge detailed description

Figure 2: Focus areas per legacy modernisation driver

Driver	Focus area
Risk reduction	• Active Knowledge (AK)
Agility improvement	• Application Life Cycle Management (ALCM) • Application Design (AD)
Cost lowering	• Technology Configuration (TC)

For example, if the main driver is to increase the agility, the main areas to focus on are improving the Application Life Cycle Management, the Application Design or both. However, more often than not, an organisation will cite all three as drivers for modernization. This will result in a large overhaul of the information system, as multiple activities have to be undertaken in order to reach higher maturity on all four focus areas. If the organisation focuses on one driver at the time, the complexity of the project can be significantly reduced.

Active Knowledge

PwC uses the term Active Knowledge to indicate keeping knowledge of an information system on a sufficient level is a continuous process. Too often the focus is just on keeping documentation up to date, or having a specified number of skilled resources. To have a mature and sufficient Active Knowledge, an information system should also have sufficient test scripts of adequate quality in place. Something that is often overlooked is that the information system should have about the same number of functional changes implemented as other non-legacy applications in order to continuously sharpen and update the skills and test scripts. It is imperative that engineers not only implement low-level changes, such as a change in a configuration file. They should also design and implement high-level changes, such as functional changes, in order to build and maintain a good understanding of the functionality and the source code, and update and improve the required test scripts and documentation.

For Active Knowledge we've defined the following maturity levels.

Level 1	There are not enough skilled resources
Level 2	There are sufficient skilled resources
Level 3	There are sufficient skilled resources, plus one of the following: Sufficient Test Scripts, Sufficient number of functional changes, Code is continuously (peer) reviewed and up to company standards
Level 4	There are sufficient skilled resources, plus two of the following: Sufficient Test Scripts, Sufficient number of functional changes, Code is continuously (peer) reviewed and up to company standards
Level 5	There are sufficient skilled resources, plus all of the following: Sufficient Test Scripts, Sufficient number of functional changes, Code is continuously (peer) reviewed and up to company standards

Example:
For an information system, sufficient skilled resources are available. Also, the documentation is in order. Yet, the test scripts are out of date and cannot be used, and over the past five years hardly any changes have been implemented.

The maturity level of Active Knowledge for this information system is 2.

Application Life Cycle Management (ALCM)

Application Life Cycle Management is the product life cycle management (governance, development, and maintenance) of computer programs. It encompasses requirements management, software architecture, computer programming, software testing,

software maintenance, change management, continuous integration, project management, and release management.

The ALCM has evolved from a Waterfall approach, where everything is designed and planned upfront and building and deploying is done in carefully managed sequential phases, to a SecDevOps way of working, where design, planning, building and deploying is done in sprint sessions. Where the Waterfall approach usually only results in a new large, release every two or three years, in SecDevOps a mini release can be deployed every day. It is therefore generally accepted that the Waterfall approach is suited best for information systems that are monolithic, while SecDevOps is best used in a Microservice Architecture.

When modernizing an information system, it is important not to focus solely on the technical aspects such as the underlying technologies and the architecture. Modernization of the ALCM is also required in order to provide the required business agility, and to be able to deliver new requirements fast.

Application Design (AD)

This parameter in the IT Maturity Model shows how flexible an application is. For this, we look at the architectural design of the application. The general opinion is that the more modular the application design, with the use of standard communication protocols, the easier, and thus faster, the integration and delivery of functional changes can be realised.

Microservices are currently seen as the most modular approach for Application Design. Although it comes with its own challenges, this type of Application Design provides the most flexible and agile application design to date. In contrast, the monolithic application has tightly coupled functional components, which can only be changed with a lot of design and planning upfront. This of course has a negative impact on the agility.

The coding language in which the application is developed is not part of this parameter. It is not relevant within this parameter, as much of the application design can be realised in any coding language. Coding language is mainly an issue when skilled resources proficient in the coding language are hard to find, or when the language is deemed no longer a strategic capability.

An application in COBOL, with a decoupled application design, sufficient Active Knowledge, and adequate cost base, is not legacy even though a relatively old programming language is used.

Technology Configuration (TC)

This model parameter shows the maturity of the used technology configuration. The general assumption is that the more open and cloud-based the technology is, the more flexibly it can be used, with a lower cost base.

Closed system architectures use hardware that is not shared or open to other manufacturers, making it incompatible with other operating software. For example, the IBM Mainframe only works with z/OS. An Open System Architecture uses hardware that is open to other manufacturers. This hardware can run different Operating Systems like Linux or Windows. Infrastructure as a Service (IaaS) is when a server instance is available in cloud, on which an Operating System is or can be installed. Platform as a Service (PaaS) provides a runtime environment to the user, e.g. a Database as a Service, or Tomcat as a Service. The scaling of the platform is still the responsibility of the user. With serverless computing, the user can deploy functionality in the cloud, without even having to worry about the underlying platform or required scale. With serverless computing, the scaling is completely managed by the service and based on the actual, currently required capacity.

Closed systems require huge upfront and recurring capital investments, with often high costs for licenses and support contracts. Serverless computing is driven by operational expenses and is almost always significantly cheaper.

3.2. Transforming the legacy information system(s):

The IT Maturity Model helps to visualise the current and the required state of the information system. Using the IT maturity assessment, we can plot the current state of the information system(s) into the model.

Example:
Based on the assessment of the current state, the current maturity of an information system may be plotted as follows:

Focus area	Current	Required
Active Knowledge	Level 5	
Application Life Cycle Management	Waterfall	
Application Design	Monolith	
Technology Configuration	Closed System	

After the assessment of the current state, the required target state is assessed. This assessment addresses the main driver for the modernization, as well as the required target maturity levels for Active Knowledge, ALCM, Application Design, and Technology Configuration. These results are then plotted in the same diagram as the current state.

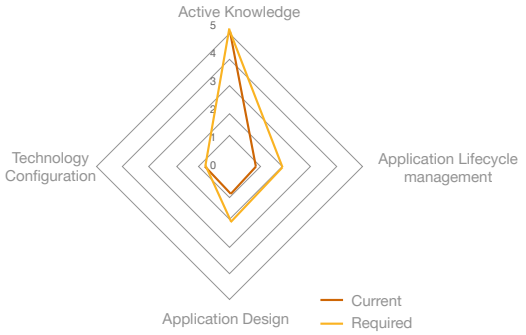
Example:
After the assessment of the required target state, it is found that the key driver for modernization of the information system is a higher required agility. Using the assessment, it is concluded that the current release cycle of once every three years has to be sped up to once every six months. Looking at Figure 2, we learn that most likely the ALCM and the Application Design require improvement to reach a next maturity level. As Active Knowledge is already at maturity level 5, this does not require any improvement.

Using the findings, it can be determined that while on the one hand the current waterfall approach is not sufficient, on the other hand, the Agile approach offers too much agility, with release cycles of three to four weeks. The best ALCM maturity level is therefore Iterative, one step up in maturity level.

In order to assure the required agility, the current Application Design, Monolithic, has to improve to reach the next maturity level as well: Loosely Coupled. Without this improvement, the release cycle would still require a lot of design and planning upfront, jeopardizing the required release cycle.

As a lower cost base is not required, the current Closed System does not have to improve immediately to reach a higher maturity level. In time, when the ALCM and the Application Design have matured to level 2, the improvement of the Technology Configuration to level 2 'Open Systems' can be considered.

Focus area	Current	Required
Active Knowledge	Level 5	Level 5
Application Life Cycle Management	Waterfall	Iterative
Application Design	Monolith	Loosely Coupled
Technology Configuration	Closed System	Closed System



3.3. Transforming the legacy information system(s)

After identifying the current and required maturity levels for an information system, it is time to assess how to reach the target maturity level. In doing this assessment, standard solution patterns can be used.

3.3.1. Solution patterns

A solution pattern is a general, reusable solution to a challenge in legacy modernization. It describes the building blocks of the solution. It also describes the required project results, and the order in which to achieve these results.

Solution patterns have three major benefits. First, solution patterns have proven to work. All Key Design Requirements and Key Results are part of the pattern. Basically, if you have an agreed solution for all building blocks in the pattern, the solution design will work. This is a major benefit in projects.

Second, patterns make communication between your and our team members more efficient. All stakeholders involved can use the patterns to discuss the solution building blocks, the available options, determine what work is required and what impact the pattern will likely have on the IT landscape and organisation.

Third, the patterns provide the scaffolding needed to start up the solution design and delivery quickly. By using the standardised documents and work packages, teams can get into an active mode quickly.

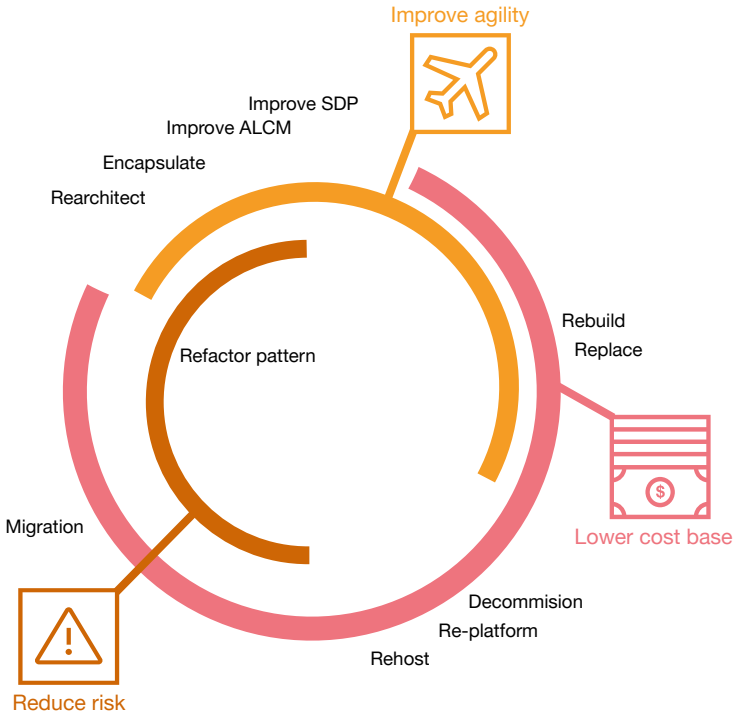
Figure 3: List of solution patterns (non-exhaustive)

Pattern	Description
Improve Application Lifecycle Management (ALCM)	To improve the ALCM, from e.g. Waterfall to DevOps
Improve Software Deployment Pipeline (SDP)	To improve the used tooling in the ALCM processes
Decommissioning	To decommission an information system
Migrate	To migrate the functionality of an information system to another existing information system
Encapsulate	To encapsulate data and functions in the application and make them available as services via an API
Refactor	To restructure and optimize existing code without changing its external behaviour to improve the features and structure
Re-architect	To alter the application code so it can be shifted to a new application architecture
Rebuild	To rebuild or rewrite the application from scratch, while preserving its scope and specifications
Replace	To replace with a new application taking new requirements into account
Re-platform	To migrate an application to a new runtime platform
Re-host	To redeploy the application to another infrastructure
Shadow Production	To implement a shadow production test environment, often useful when the active knowledge is not mature

3.3.2. Linking these patterns back to the original drivers

Figure 4: Focus areas per legacy modernization driver

Driver	Focus Area
Agility improvement	• Improve ALCM • Improve SDP • Encapsulate • Re-architect • Rebuild • Replace
Cost lowering	• Decommissioning • Re-platform • Migration • Rebuild • Replace • Re-host
Risk reduction	• Improve ALCM • Improve SDP • Refactor • Encapsulate • Re-architect • Migration



The default is cloud

4. The default is cloud

4.1. Reasons for migration to cloud

Cloud computing has become the focus of current and future technology needs for the enterprise, driven by competition from 'born-in-the-cloud' innovative companies, as well as by consumer demand for more efficient and customer-centric services in businesses today, across all market segments.

The cloud offers compelling economics, the latest and greatest in technologies and platforms, and the required agility, leading to an increasing number of organisations to choose cloud.

We see the following main reasons for cloud adoption by companies:

Flexibility in resource utilization (Scale up and Scale out)

Cloud computing offers businesses with unmatched flexibility in terms of usage policies and scalability. Data storage is one of the major resources that companies want to move to the cloud

Cost effectiveness

Flexibility in resource utilization (and management) leads to reduced operational costs as well as controlled financial spending

Availability of talent

The cloud leverages modern technologies which are comparatively easier to learn through multiple, easily accessible platforms. Also, most of the modern stacks are supported on cloud, which makes it easier to attract the right talent, reducing dependency on knowledge of reducing the talent pool of legacy languages

Agility and latest technologies

The cloud offers an Open Systems environment in which high productivity and rapid innovation take place at a tremendous rate. A well-designed strategy based on implementation of a cloud infrastructure scales easily and quickly, both scaling up and down, to synchronize with business demand. Non-functional requirements such as back-up, redundancy and disaster recovery is seamless. Enterprise-wide database sharing is achievable, at the same time managing high performance. Additionally, multiple devices and end-user platforms are supported, owing to cloud supporting a variety of integration patterns.

4.2. Legacy transformation cannot be seen in isolation from IT 'commoditization' through cloud

With the increasing adoption of cloud, the imperative question also involves the attractiveness of moving legacy applications and workload to cloud. Organisations relying on legacy systems (such as mainframes) face an increasing urgency to transition workloads to the cloud. Confronted with a rapidly shrinking talent pool of expertise amid escalating costs, organisations stand to save substantial expenditures while being agile at the same time —if they can migrate equivalent functionality to cloud.

With the rapid adoption of cloud, enterprise cloud migration has re-fuelled/revived the legacy modernization services. Here we clearly see two patterns emerging:

Lift and shift

Many organisations are on a path to transition workloads to the cloud using the lift and shift method. This may not contribute to maturity enhancement of Application Design (AD), but does enable a higher maturity for Technology Configuration (moving from Closed System to IaaS), thus enabling benefits. The primary driver we see for Lift and shift is Cost (reducing cost of license or usage) and to a limited extent Risk (moving to a new platform to reduce the risk of Life Cycle Management (LCM) on old platform). Lift and shift can have two approaches:

- A combination of re-hosting and re-platforming: re-hosting existing mainframe applications on a cloud platform using a mainframe emulator (i.e. Micro Focus Enterprise Server etc.) with

a limited impact on the end user and application landscape. This enables reuse and requires less changes to standard mainframe technologies such as Cobol, JCL, DB2 etc., while re-platforming to new stack for the required component (such as Databases from DB2 on mainframe to SQL server).

- Migration of workload to cloud, such as moving or the execution of batch jobs, which often are a large portion of the mainframe application portfolio and frequently consume a large amount of processing capacity. Additionally, online transaction processing workload, which includes heavy transaction processing, while consuming a large amount of CPU, is good candidate for moving to IaaS. Additionally, the workload of online transaction processing is a good candidate for a migration to IaaS, due to its high volume and required processing power. This could be done as a mixed of (partial) Re-hosting, re-hosting and/or re-architecting.

Re-engineering

Some workloads and applications are not meant to or should not be migrated to cloud-based environments through the lift and shift model. Often, only some changes are required for applications to make them benefit the most from a cloud-based environment.

The re-engineering approach is recommended when the existing mainframe application is no longer able to meet future-state business requirements or an agile target architecture. This approach will create a new application or set of applications, with comparative performance and equal or enhanced functionality (depending on scope of re-engineering). Typically, this is done using cloud-native techniques, leveraging micro-services, Containers etc. along with using additional features on cloud platform such as AI, Data analytics etc.

Based on the patterns we discussed in this document, the patterns of Re-architect, Re-build, Replace and Re-factor would be possible patterns for this option/scenario.

The Re-engineering approach enables maturity across Active Knowledge (from level 1 to possibly level 5), Technology configuration (from level 1 to level 3-5, depending on choice of pattern) and application design (from level 1-3 to level 3-5, depending on as-is situation and choice of target pattern)

Regardless of the selected approach, it is important for companies to understand that legacy workloads and applications should be considered in their cloud migration strategy. This will often result in significant cost savings, increased agility and a future-proofed architecture.

PwC specific approach

5. PwC specific approach

5.1. A simple 5-step approach to realise full end-to-end transformation

The 5-step approach is based on best practices and validated at clients that are market leaders within their industry. Furthermore, the approach can be successfully applied, regardless of the way of working at a client (waterfall or iteratively in a full agile manner).

Step 1. Understand the business drivers, gain full insight into the IT landscape, and set up a high-level Business Case (BuCa)

- Key to any successful project/program/initiative is understanding the business drivers. Legacy transformations are no exception. IT being a means to an end, it is crucial to understand the end goal and where the IT capability does not perform at the required level. If cost reduction is the main driver, this may have other implications than when the problem at hand is de-risking.

- Once the business context is clear, understood, and agreed upon by the relevant stakeholders it is time to gather insights into (parts of) the IT landscape. For this, we require a standard set of artefacts - that are needed within the assessment to provide insight - such as architectural design, functional and technical documentation, data model, etc. These artefacts, in combination with the previously determined drivers, and our IT Maturity Model, will form the fundamental functional and technical analysis to understand your IT landscape and problems at hand.

Step 2. Identify solution patterns, create a high-level roadmap with scenario planning and refined BuCa

- Using our repository of solution patterns, we can take the derived insights even further and provide you with scenarios for moving forward with the (set of) application(s) and platforms in scope. Moreover, we will substantiate these scenarios with a sound business case to support the decision-making process, and choose the best-fit solution.

Step 3. Crosscheck scope and 'handshakes' with other strategic initiatives

- Most likely, the legacy transformation initiative is not your only undertaking. In fact, it is very likely that the same applications are targeted by several programs within your organisation. It is therefore crucial to validate the scope against other initiatives, have agreements with relevant stakeholders in place, jointly determine prioritisation and engineer a holistic plan of approach.

Step 4. Deliver quick wins, create solutions for remaining dossiers and refine the roadmap

- As soon as the prerequisites are in place, we will immediately address your legacy by picking up any quick wins that are there, to fire start the initiative, create momentum and drive direct business value.

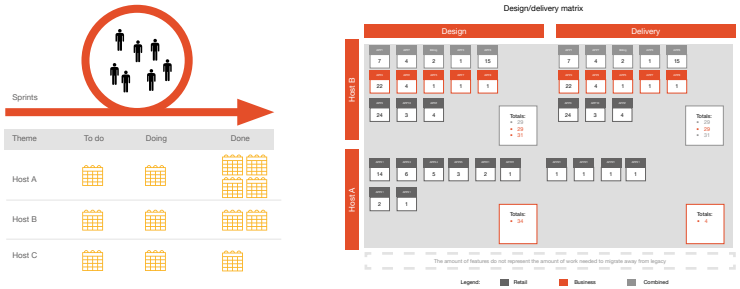
- In parallel, we will start slicing the more complex dossiers into manageable chunks and create detailed solutions using, again, several scenarios for decision making. Consequently, this will lead to a refined roadmap with a more detailed sequencing of activities and firmed planning.

Step 5. Execution

- We will assume responsibility for our solutions and roadmap, and will take on the execution in line with this roadmap. For knowledge-sharing purposes, this will be carried out in embedded teams. In this way, we can quickly resolve impediments and maintain the momentum that is so crucial for critical initiatives.

- We will leverage our 'Digital PMO' to fire start delivery and quickly turn data into insights for management to effectively steer the initiative. The Digital PMO is a digital progress dashboard developed by PwC.

Figure 5: Examples of PwC Dashboard



5.2. PwC Way-of-Working

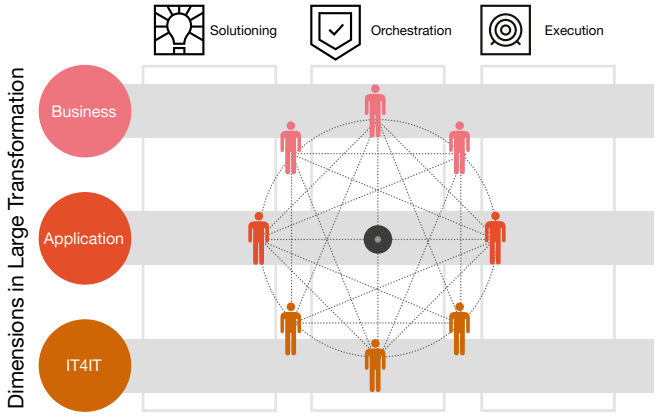
It is our firm belief that to deal with legacy transformations successfully from beginning to end, you need a team with a diverse skill set. We therefore work in squads covering an array of skills:

Solutioning
With a thorough understanding of our clients and the ability to solve complex problems; creating (technical) solutions; re-imagining the possible.

Orchestration
Able to orchestrate decision making by navigating the corporate politics throughout the full spectrum of stakeholders (C-Suite to engineer); also creating business cases and roadmaps.

Execution
Driving execution of initiatives through strong (agile) program management skills with a strong focus on outcomes: having a solid vision, but striking a delicate balance with being agile. Although we will not be coding, we will fire start and drive your initiative.

In addition to these capabilities, it is important to cover all aspects from strategy to infrastructure, in order to fully support these legacy transformations. Next to aligning with the more obvious IT skills in the application layer and understanding your IT4IT, it is also crucial to align these IT capabilities with to the businesses these capabilities are designed to support. In quite a lot of situations you will need to have a fully functional roadmap, with an alternative platform for every single piece of functionality, and put this on the backlog of business.



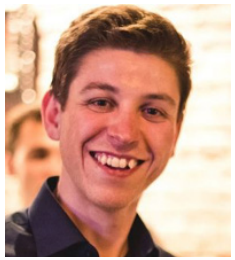
Agile Transformation Squads

6. Authors

Remi Hesterman
Director Technology Consulting

Omit Singh
Senior Manager Technology Consulting

Martin Borman
Senior Manager Technology Consulting



Brian van der Wyck
Manager Technology Consulting