CORE®

ANTI-AGING FOR IT SYSTEMS

Evolutionary architecture: taking advantage of technological innovation cycles

Author team
Christian Böhning
Dr. Carsten Wedekind
Dr. Philipp Kleine Jäger
Francisca Di Piazza
Justus Dircks
Karsten Trostmann

February 2021
White Paper

1 Introduction

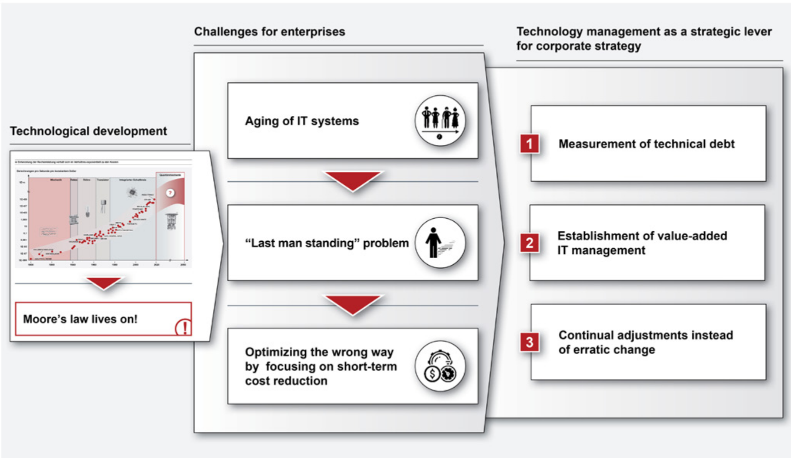
The ability to shape progressive, technological change has developed into an essential, integral part of competitiveness in many industries and sectors that, in turn, determines economic success or failure.

IT systems are subject to natural degeneration over time

The up-to-dateness and adaptability of the IT infrastructure and the application systems within it play a decisive role. In real life, IT systems exhibit a phenomenon that is very similar to the aging process in humankind. With advancing age, they become more cumbersome and lose the ability to adapt quickly enough to changing market conditions. Their efficiency and reliability decline at the same time as the maintenance and operating costs rise.

Generally speaking, the aging process cannot be prevented. It can merely be delayed and its impact limited. The aging process of IT systems is not, in itself, a new phenomenon. Nevertheless, its significance has dramatically risen over the past few years because technological innovation cycles have become increasingly shorter and IT systems are playing a significantly more crucial role in the creation of value with the progressive digitization of processes in companies. In order to limit the effects of the aging process in IT systems, its causes must be understood and appropriate countermeasures introduced in good time.

Technology management as a component of competitiveness



COREresearch 2020

Fig. 1: Technology management as a component of competitiveness

Technology management as an integrated and ongoing architecture process

Today's challenges can no longer be met by traditional enterprise architecture management approaches, which are based on a more planned economy, with development plans and time horizons of 5-10 years. Rather, it is more a question of establishing technology management of IT systems as a continual and evolutionary architectural design process that is pursued in close collaboration with the business units. In order to accomplish this, the deficiencies of current IT systems need to be permanently measured in terms of quality and quantity in relation to the quality and productivity standards established on the market. To mitigate deficiencies already found in the existing IT landscape, it is essential that the management systems are harmonized between the business and IT world. The principle of pursuing project-based budget and portfolio management that is practiced in most companies is primarily geared towards immediate business benefits. The ensuing IT solutions are therefore based on local and short-term optimization and lead to ever-increasing technical costs associated with the IT systems concerned. In order to break through this system, some of the major software corporations have set up joint product, development and business organizations, whereby the adverse effects are mitigated through end-to-end responsibility and a joint P&L steering.

The following chapter provides an insight into the development of past technological eras and suggests ways in which companies can mitigate the aging process of IT systems through sustainable IT management as well as benefit from the potential of cutting-edge technologies at an operational, organizational and strategic level, in order to strengthen their own innovative capability.

2 Why Do IT Systems Age?

2.1 System designs are subject to the conditions of the era in which they are developed

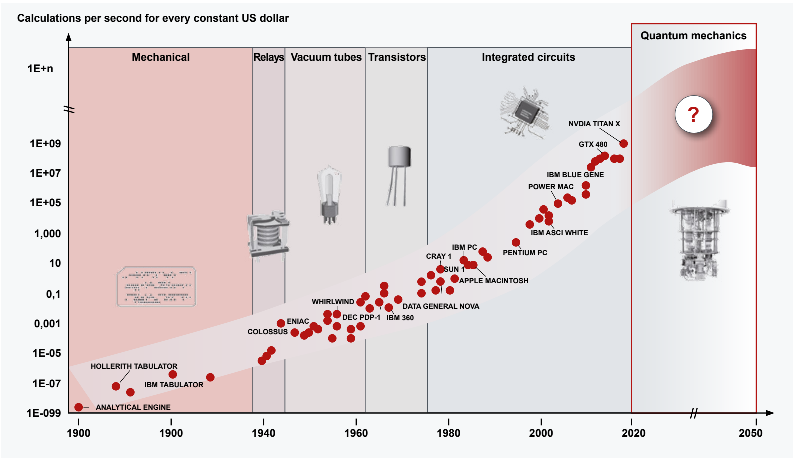
As early as the nineties, computer science was concerned with the aging process of IT systems and software. None other than David L. Parnas, to whom we owe, among other things, the principle of encapsulation and secrecy, presented two major causes in his essay “Software Aging”.

The concept behind IT systems is based on design paradigms of specific periods

The first is that systems are developed according to design paradigms that are geared to the technical, economic and organizational conditions of their era. Nevertheless, these parameters are continually changing, and this change is primarily determined by developments in technology. This considers not only the individual technical innovations themselves, but also the methods and procedures adapted to them.

At the end of the last century, in the eighties and nineties, computing power and storage capacity were expensive. The design paradigms derived from this were the concept of the central computer, the maximum reuse of code and economy as the guiding principle in data modeling. Under current conditions, these paradigms are highly problematic designs, and many companies still suffer from their subsequent effects. What used to be good and appropriate as a solution pattern back then, has now become unsuitable given today’s requirements that IT systems are flexible.

Computing power has been increasing exponentially relative to costs



Sources: Ray Kurzweil, Steve Jurvetson, computerworld.ch

Fig. 2: Moore’s law in the context of past development of cost of computing capacity

Exponential progress in technology without further development of the system leads to aging

If we take a look at the development of the technical conditions by way of the historical development of computing power, then we see that the development of technology is exhibiting evolutionary behavior. As is the case with all evolutionary processes, this development does not follow a linear pattern, it develops exponentially. In 2021, we are all well aware what exponential growth means. Technical innovation cycles are accelerating and the rate of change of both the conditions and the design paradigms is getting faster too. Consequently, IT systems are becoming obsolete quicker as a result of the ever-increasing rate of technological progress.

2.2 New technologies result in new processes, methods and tools

IT value chains and workflows need to be modernized to make use of innovation potential

Workflow patterns and methods have to be adjusted in order to be able to use the innovation potential of new technologies. This, in turn, enables new opportunities also to be used effectively to improve quality and productivity. This is not only the case with the business processes digitized by IT innovations, but also, most notably, for elements of the IT value chain. Current paradigms for software development that are geared towards agility and efficiency, such as:

- Continuous integration / continuous delivery
- Test-driven development
- Infrastructure as code

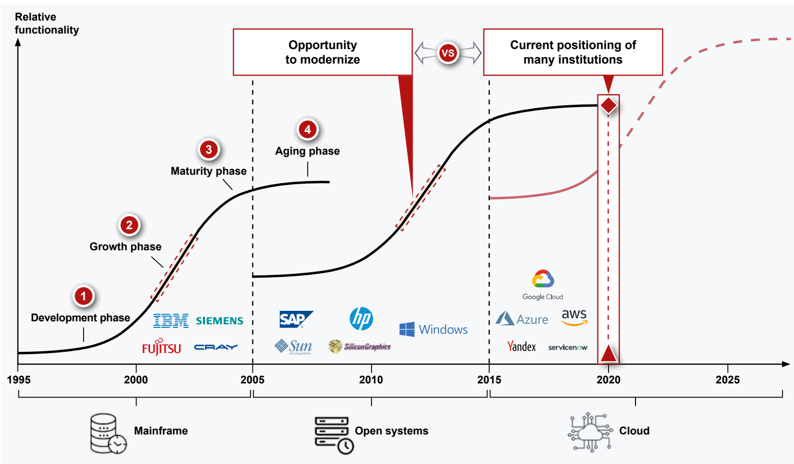
cannot be applied, or only applied to a limited extent, to the technologies of the mainframe era. It is not without reason that the complexity of obsolete IT systems in respect of a loss of transparency and the associated test efforts is cited in computer literature as the second most frequent obstacle for implementing changes. The implementation of technical requirements within the space of 2-3 weeks without any adverse impact on quality is now possible thanks to consistent orientation towards software quality in the development process and extensive test automation. In the first instance, this presupposes the willingness of software developers to continue developing methods and processes aimed at improving the quality. However, this also requires support by means of the technology platform used. Even if the situation is improved thanks to better tool support on the COBOL/mainframe platform, current software development methods used on this platform are more an exception than the rule.

2.3 Market requirements can only be implemented up to the technology barrier

Every technology generation has natural technology barriers

Even well-maintained systems are no longer able to meet requirements competitively after a certain point in time. Unfortunately, taking a look into a crystal ball in order to recognize changes in technology or paradigms in system design is only possible to a limited extent. When the IBM designers conceptualized the IBM System 360 in the sixties, they did not have the option of an online system. Bill Gates is wrongly attributed with saying “640K ought to be enough for anybody“ in connection with memory management of the MS-DOS operating system. Nowadays, every smartphone contains 100-times more RAM. When a technology comes of age and initial applications are created using it, a technological barrier is the automatic result, since the previous technology is inferior in terms of either performance, quality, speed or costs. For instance, online banking at the end of the 1990s was no longer able to be implemented efficiently on the mainframe platform because critical components such as e.g. the web server or frameworks were not available on this platform for web development.

Technology development cycle: genuine innovation requires a change in technology base



Source: COREresearch 2020

Fig. 3: The technology development cycle

2.4 Working and IT value chains are changing

Innovation and evolution now take place in open ecosystems

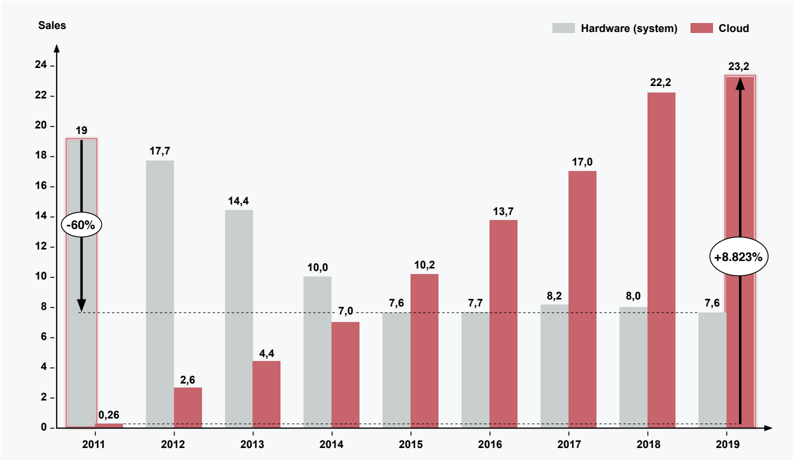
Whether or not a technology platform can adapt to ever-changing requirements through evolutionary means depends ultimately on their ecosystem. Where the system is closed as a result of a restrictive licensing policy or high system-related entry barriers, the ecosystem is determined by a few dominant players, so there is very little competition. Under such conditions, the evolution of a technology platform is very much a “closed shop”.

Nowadays, frameworks required to develop complex software landscapes are generally available as open source technology and are developed further by a huge community. The reason for this is not a utopia-driven patronage, rather the realization that evolution can only take place in an open ecosystem. In addition, the development costs of non-market differentiating components are socialized. All users of the software are able to contribute directly in improving the product and to add new ideas. Parts of the software and tool landscape can be developed more quickly by a broader user basis and reach a higher quality level.

Cloud and SaaS models are making hardware and software licenses obsolete

The IT value chain has therefore been in a turmoil for some time. The successful model established over decades of stable sales from hardware and software licenses is now obsolete. Not only Google, Apple and Amazon, but SAP and IBM too, no longer rely primarily on income from hardware and licenses in their business strategies – they focus on cloud and software ‘as a service’ models. Access to hardware resources is fully democratized thanks to cloud technologies. Program code is often no longer a company secret. Indeed, the model of closed and exclusive ecosystems is obsolete because often it offers barely any competitive advantages.

IBM sales from 2011 to 2019 by business segments (in USDbn)



Sources: IBM Annual Reports 2011 - 2019

Fig. 4: Trend in IBM hardware and Cloud sales from 2011 to 2019

2.5 Organizations change at a slower pace than technology

Organizational change as a prerequisite for exploiting technological opportunities

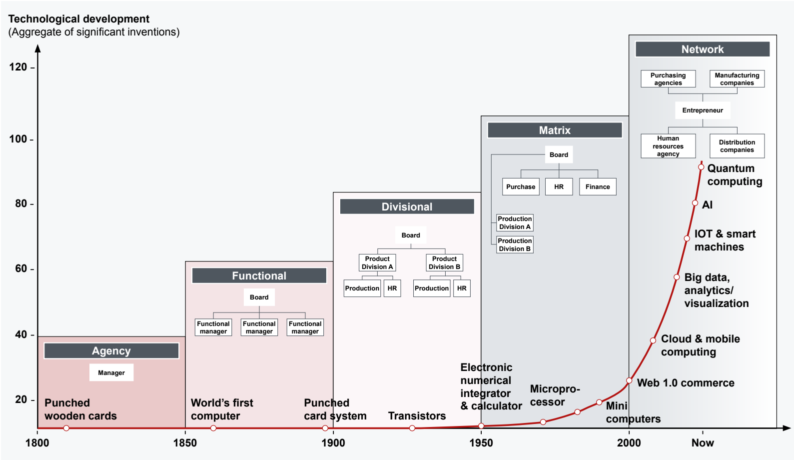
Technology management deals with the challenge of participating in technological developments which are growing exponentially. Extendable, exchangeable and flexible IT systems is only one prerequisite for this. Implementing the necessary structural and organizational changes, in particular, in order to be able to use technological possibilities effectively, is a major challenge since restructuring always has to overcome organizational inertia. Generally speaking, it should be noted that many companies – or rather the people working with them – are overwhelmed by the ever shorter technology development cycle, resulting in necessary changes only being adopted slowly.

2.5.1 Historical growth

Organizational structures are geared towards functional benefits

Most of today's organizational structures in virtually all business sectors are geared towards functional aspects – the skills and know-how of staff are pooled by category and subdivided into different functional business units (so-called 'silos'). Business processes and channels of communication, too, are equally divided into 'function' categories. These organizational models have grown historically and are still managed today according to rules and regulations that stem from the Industrial Age. Examples of these types of organizations range from the organization of the railways of the nineteenth century according to functions, staff-line organizations and divisional organizations to the matrix organization of aviation and aerospace companies of the post-war period.

Changes in organizational structure vs technological development



Sources: COREresearch 2020, World Economic, Miles & Snow 1984

Fig. 5: Changes in organizational structure vs technological development

For a long time, these forms of organizations made sense as their underlying organizational models were geared towards allocating resources to different functional business units, thereby producing reliable output based on constantly recurring processes. In a similar way to the technology platform, these organizational types effectively covered the requirements of "their era".

Organizational silos obstruct technology adoption

Beyond the organizational silos, many business units, particularly in large corporations, were set up in recent years to act as profit centers. This causes fights for resources, especially concerning the budget. The silo and competitive mentality of the business departments as well as the management, organization and process structures that have already been tried in this context are diametrically opposed to the necessary change measures.

2.5.2 IT systems and low acceptance of change measures among staff

The need to replace the underlying technology also raises the problem of a lack of skills in the new technology, approaches and cooperation models and more flexible governance. In particular, the following factors play a dominant role:

Technological change requires the ability to adapt technical skills

- The speed of technological change requires constant renewal of staff skills. According to the CEO of the US telecommunications company AT&T, the skills of 40% of the company's staff will be completely out-of-date after five years.
- Employees stick to old technologies, methods and processes with which they are familiar, and are not particularly motivated to learn new technologies and methods. These historically grown processes, methods and technologies give the people concerned the feeling of security and stability.

- Limited capacity for further education.
- As a result of this lack of skills, staff fear becoming irrelevant and are concerned for their own jobs; thus organizational inertia regularly counteracts the process of renewal.

The dismantling of outdated organizational structures represents a significant challenge for technology management.

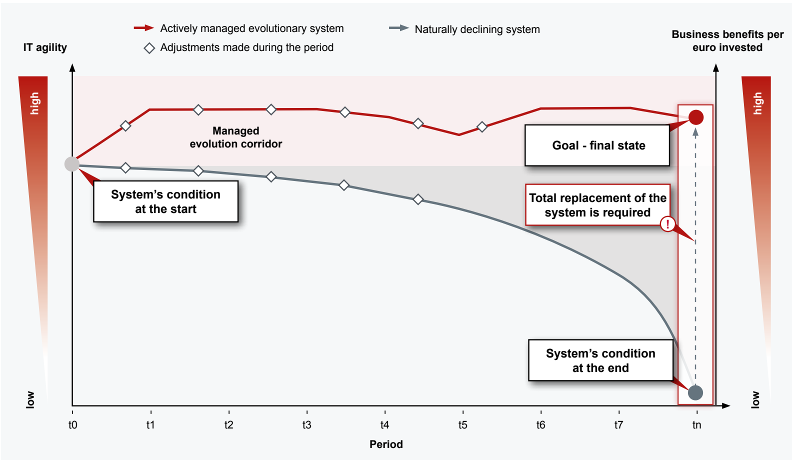
3 What Problems Arise as a Result of Outdated IT Systems?

3.1 Aging means a gradual loss of conceptual integrity and increases the degree of dependency (technically and in terms of skills)

One of the causes for IT systems becoming outdated is the change and adjustment made to IT systems themselves. For the majority of the IT systems used in companies, extensive concepts have been and are being developed that pursue a specific business objective and technical paradigms. When these systems are first installed and commissioned, they correspond to these concepts, so they have a high degree of conceptual integrity and do not exhibit any structural or technical debt. The change process that then sets in generally takes a natural, more or less strong entropic course. The system deviates increasingly from the original concept as a result of an increasing number of changes and implemented functions. As the system ages, the speed with which the original concept erodes accelerates and it becomes increasingly difficult to make adjustments, meaning the quality of the system deteriorates too.

The quality and adaptability of an IT system often decrease with the number of changes made

Long-term and significant effects on IT agility and business benefits



Source: COREresearch 2020

Fig. 6: Long-term and significant impact on IT agility and business benefits

We intend to take a look and explain some of the reasons for the above.

The loss of conceptual integrity is often caused by change management geared exclusively to further functional developments. Changes to IT systems shall be implemented opportunistically, with minimal spend in terms of time and money. The conceptual integrity of the system means nothing to the user, hence it is not a goal nor is it managed regularly. Major changes to monolithic systems with vast areas of functionality and changes to service providers then lead to real surges in erosion.

Incorrectly prioritized change management reduces conceptual integrity

IT systems are implemented in larger projects and in part by external service providers. After initial commissioning, the system software is often handed over to a maintenance team. In most instances, members of the maintenance team only become involved during the latter stages of the project or are not even part of the project team. Consequently, they have to get to grips with the concepts at the heart of the system and become familiar with a large code base, using what is often inadequate documentation. Even if detailed documentation is available, there are still gaps at the transition stage, meaning that there is (still) no 'emotional connection' to the code. Changes are then made pragmatically in ignorance of the original concept behind the system.

Another but very crucial aspect is that decreasing conceptual integrity goes hand-in-hand with a loss in value of the IT system. This is not recorded anywhere but weighs on the system like an invisible and steadily growing mortgage. Technical debt is only systematically recorded and known to companies in very few cases. Consequently, the extent of the problem is not fully appreciated.

Failure to record technical debt results in ignorance of the actual state of the systems

3.2 Last man standing

In most companies, IT systems and applications have grown over time. They are optimized for different platforms, sometimes programmed in other languages and interlinked. These old IT systems are used jointly by different business units or departments within a company, with the costs for running the systems and platforms, as well as the processes and organizational landscape, being split between them. Hence the costs appear to be relatively affordable due to the number of participants using the system concerned. Apart from the aspects of cost reduction, it is often a question of exploiting the options for reuse and the full capabilities of the current systems, which is why companies frequently invest more than 70-80% of their IT budget on maintaining the status quo of their IT systems. Other reasons for retaining old systems include:

Cost-savings and reusability of old IT systems often take up the majority of the IT budget

- The system works so there is no need to replace it, or to put it more simply – "if it ain't broke, don't fix it."
- The system is extremely complicated and the technical documentation is often not ideal, so a simple definition of scope is difficult.
- Redesigning the system is exceedingly costly as a result of the closed and extensive architecture.
- The purchase of a new system is postponed because of a lack of capacity (staff, budget, etc.).

Due to the dynamic changes that take place in IT and general market pressure, a transition to a newer and better system is not only necessary, it is inevitable.

Consequently, the advantages and business benefits of switching to a new system must be thought out thoroughly and comprehensively calculated.

A cost-benefit analysis (or a full cost analysis) serves as the basis for substantiating the business case. This also includes taking factors such as compliance, data integration and security into account as well as clarifying the costs and efforts involved in implementing planned innovations and necessary measures. Apart from the actual running costs, the calculation also takes into consideration the cost or damage to internal and/or external reputation caused by old IT systems and the extent to which market opportunities are constrained as a result.

Often the first two major hurdles are the lack of a full cost analysis and the lack of modularity, as the actual costs of the system are not transparent, which gives the impression that even limited renewals de facto equate to an overall transformation.

Departments that miss out on a collective switchover to new systems, should have to bear high operating costs by themselves (polluter pays principle)

If, for most of the participants (business units), the benefit/business value of switching to a new system outweighs that of remaining on the old system, they will switch at some stage. The department that decides not to move but to remain on the old system will ultimately have to bear the full cost of operating the system, which was previously shared among several participants. This leads to the so-called ‘last man standing’ problem.

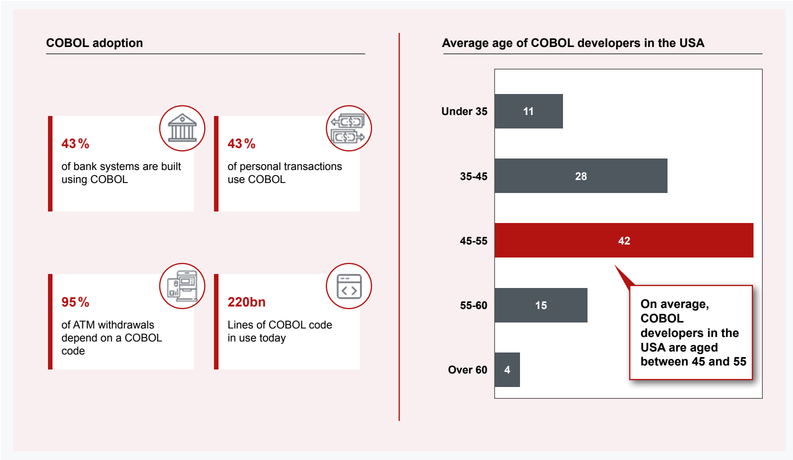
This does not describe a technical problem, rather it is more a question of the proper and timely evaluation of the costs and benefits associated with the change as well as a consideration of the system life cycle, which also takes economic aspects into account (forgone/lost turnover, etc.). It is basically recognizing the right time to replace aging systems, in order to avoid having to bear the entire costs of operating a legacy system as the last remaining department/business unit.

3.3 Failure to consider the mortgage leads to false optimizations

Staff and skills for outdated technology are costly and scarce

If the ideal time to implement change has been missed, there is a risk that the entire system will be caught up in a self-perpetuating downward spiral: the human resources as well as the technological skills required become less relevant for the labor market, leading strangely enough to higher costs for the resources. This is exactly what has happened with specialized service providers for COBOL or Fortran programming. Moreover, with outdated systems, there is also the risk of having to invest heavily in the further development and safeguarding of rudimentary operational capability due to the end-of-life cycles of hardware and software.

COBOL core applications in the finance industry have a next-generation problem



Source: Reuters Graphics84

Fig. 7: The technically outdated and generally unusual programming language COBOL continues to be widespread in the financial sector.

The basis for decision-making for CapEx/OpEx expenditure is often inadequate compared to taking a holistic total cost of ownership approach

Failure to take technical debt into account when making strategic and architectural decisions leads to a distorted view of the total cost of ownership for continued operation or replacement, since an outdated system often uses an unexpectedly disproportionate amount of resources – an effect caused by a high run-the-company change-the-company ratio, that can be measured. This incomplete time view with significantly above-average costs towards the end of the service life is further exacerbated by optimization decisions taken at a local level. Consequently, business domains with a need for change on a long-term basis only can thus become millstones for an entire architecture and organization.

Even well-designed systems and architectural designs suffer from this effect, as operating costs must always be considered in relation to the technology used by competitors. In IT, in particular, with its exponential technology development (see Moore‘s Law in Fig. 2), significant differences in efficiency can be the result of several years of technological difference in age. As the software industry is evolving more into an ecosystem and modern and efficient technologies and frameworks currently show high interdependencies in 2021, network effects frequently make this dependency even more acute. Service meshes, modern API gateways, containers and software for infrastructure management and automation often cannot be used without Kubernetes.

A delayed switchover worsens the competitive situation and decision-making options

Due to the intertwining of many legacy architectures, the decision to switch is often delayed until the last possible point in time. This, plus competitive pressure result in there being no time for proper, prudent and long-term planning and transition. By investing in obsolete systems, the opportunity to invest in modern technology is lost. As a result, the new technical platform often suffers from birth defects right from the start.

3.4 The problem of management based on the aspect of costs

From a historical point of view, IT is positioned as an internal cost center with its own structures in many companies, especially those in the finance sector. Consequently, investment in IT is assessed mainly in terms of cost. The socialization of costs that often occurs due to the inability to allocate costs in line with expenditure exacerbates this mechanism further.

By contrast, there are alternative types of companies whose IT is integrated into the respective profit centers. Even if there is also a central service area in such companies, this setup has significant advantages, as the flexibility for investment is higher and the control processes for overall optimization are frequently not required as much.

The need to have a strategy-oriented company is widely understood as a management paradigm. There is research and evidence (Conway's Law) that this same mechanism also works for IT systems – and even more for the interdependencies between systems and companies. Nevertheless, it is still not widespread in practice.

Technology management focusing on processes, technology, compliance and organization

Consequently, it is now imperative to develop adequate technology management approaches for the combination of organization, technology, processes and compliance as well as, last but not least, culture:

Individual profit & cost centers require comprehensive technical and architectural management in the IT organization that carries out optimization processes at the overall company level and, in particular, creates transparency regarding the financial impacts of functional and non-functional requirements. Apart from the technological aspects and ensuring the ability to act by means of decoupling and using modern technology, a full cost calculation and business domain mapping are necessary, in order to continue making decisions regarding technology and systems. To make sound decisions around technology, it is necessary to keep the total cost of ownership of the current system comparable with possible alternatives, in order not only not to miss the ideal time for optimal change, but also to be able to provide well-founded financial justification.

Integrated profit & cost centers have long been common, especially in large technology corporations. However, IT cost centers can also exist in these corporations such as, for instance, in infrastructure operations. However, technology industry leaders have often positioned these cost centers on the market as independent companies to solve the problem of sufficient incentives in terms of efficiency. Examples of formerly internal cost centers include hyperscalers such as Amazon AWS and the Google Cloud platform. The organization according to products and extensive decoupling has the advantage of being able to discontinue unprofitable products and areas at any time without a far-reaching impact on the overall company and IT landscape and thus be able to avoid all associated costs.

A short-term perspective based solely on financial aspects of a single requirement ignores important interdependencies such as the accumulation of technical debt and follow-up costs.

4 Proactive Technology Management as an Anti-Aging Program for IT

How can companies effectively meet the challenges of progressive aging and degeneration of IT systems caused by changing market conditions and a faster technological change?

Technology management focusing on processes, technology, compliance and organization

4.1 Monitoring the technical state of IT systems

To be able to make sound management decisions on how systems are to be architecturally aligned, transparency of the current status and an understanding of the development of the (software) market, the company's own business model and possible business strategies must first be established.

Possible courses of action to slow down the aging process

This is done by means of a review framework, which allows for technical due diligence of specific services and entire systems, assessment of the enterprise architecture and organizational structure as well as supporting measures such as control, governance, compliance and reporting. This results in findings in the form of technical debt and deficits that arise over time. These are not technical errors, but are various aspects of inadequate software quality that can be classified as follows:

- Design or architectural weaknesses
- Shortcomings during the implementation stage (code smell)
- Testing debt as a result of insufficient test coverage or too little automation
- Inadequate documentation

Approaches for assessing systems need to be based, first and foremost, on the architecture and the operating model: an in-house system will be assessed differently compared to one based on COTS or a turnkey system. An important indicator to determine the maintainability of systems is the complexity, which can also make the loss of conceptual integrity described above transparent:

Assessment of the complexity of the entire system architecture

Architectural sedimentation: an increasing number of layers, such as anti-corruption layers or integration systems, not only results in a deterioration in the system's performance, but also massively increases the connectivity and thus the complexity of the entire system due to the often necessary networking and combination of different functions at different levels. Host/mainframe-based core systems are often found in the finance sector, which have been abstracted by function services, networked by means of integration and process automation systems and then connected by means of process services using channel-specific interfaces. These interfaces are often encapsulated again by additional business logic – and then there are various technical modules such as API gateways, service repositories and assorted access control systems, whereby the complexity of the underlying network infrastructure with various virtual and physical networks, routers, switches and firewalls is not yet taken into account. Measuring the degree of sedimentation and/or the degree of abstraction help to assess this complexity. Modern architectures are based on abstracted infrastructure with self-contained services and, where necessary, an API gateway. Here, business logic dictates that the number of layers should tend towards 1.

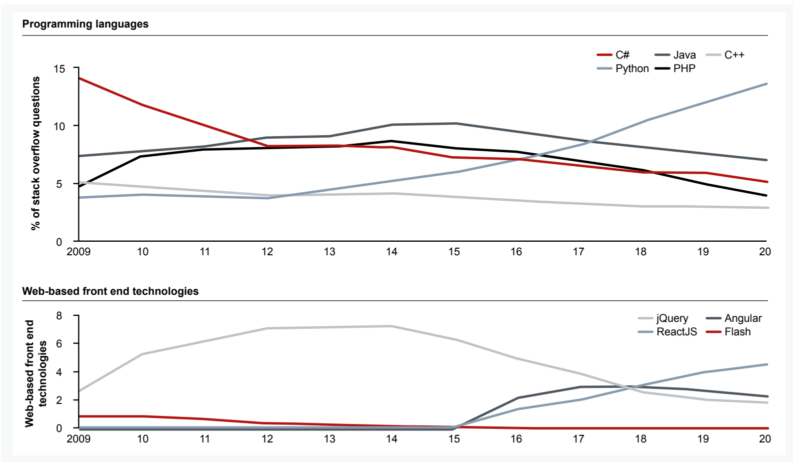
The degree of system/augmentation: The relationship between systems and their augmentation can be used as a further gauge: the more that support systems and additional functional modules have to support the original system over time due to new requirements, the lower the conceptual integrity and the higher the complexity – an indicator of inefficiency. This phenomenon, too, is frequently found in finance systems. For decades, additional regulatory requirements and the need to connect various ATM and card systems, as well as online banking and other channels, have driven the complexity of the systems. Consideration of the necessary applications in order to fully map a function, domain or capability can provide clarity. Here, too, the key figure should tend towards 1 in an ideal system.

Measurement of conceptual integrity through interdependency of support systems and modules

Aging indicators: The creation of a technology index consisting of several key indicators helps assess where a component or framework is in technology development and whether a replacement should be considered. Possible parameters that could be included in the technology index are, for example, the number of inquiries about certain frameworks on stack overflow, for open source components the number of contributors and commits on Github or similar, programming language indices and supply/demand on platforms such as GULP. For COTS software, indicators can be market analyses such as Gartner Magic Quadrants, industry trend analyses and manufacturer life-cycle notices as well as the number of customers and their development.

Creating an index helps with the assessment of current technologies

Development of major programming languages and technologies



Sources: Stack overflow

Fig. 8: Development of major programming languages and technologies

Pattern violations: Consistently applied design patterns indicate an architecture design with basic concepts that are proven adequate for the intended purpose. Compromises and deviations from the conceptual paradigms (especially when frequent) point to shortcomings in the architecture design. Where available, documentation relating to the architectural deviations or the technical debt, or similar documentation, provide a point of reference. An architecture review on different levels may be useful in the absence of such structured records.

Comprehensive test management to measure the code quality

Code quality: To achieve a high degree of code quality, formal design requirements such as conformity, comprehensibility, analysis, modifiability, adaptability, verifiability (for instance named and defined according to ISO/IEC 9126) are regular tests which cannot be avoided in the event of any adjustments to the source code. This should not only focus on static code quality tests, rather complete (happy & sad path) test coverage should be ensured, but in order fully to weigh up the positive and negative impact of changes, there should also be extensive linting, fuzz testing, mutation testing, load & performance measurements, penetration testing, integration testing, smoke and end-to-end testing as well as chaos testing. Similarly, in situ monitoring of KPIs should be included for application stability such as uptime, latency, error rate, etc.

Deviation from the market standard: Certain de facto market standards emerge over time in the software industry as well as in individual industrial sectors. Deviating from these standards requires a well-founded strategic reason, otherwise it is likely to end up, in hindsight, being an incorrect decision (Betamax syndrome).

Technical debt expresses the effort of the measures required to raise the IT system back to the quality level to be achieved. Implementation and test debt can be recorded quite easily using tools. The introduction of formal and tool-based track recording for architectural decisions is helpful for documentation as it can be used to understand decisions and concepts. They also serve to document the reduction of structural deficits.

IT integrity is validated by regular architecture fitness functions

Detection of structural weaknesses should be done using methods of an evolutionary architectural approach. At the heart of this method are the so-called architectural fitness functions that are formulated as architectural scenarios and validate the structural integrity of IT systems in regular reviews (every 3-6 months).

4.2 Moving away from short-term cost-optimized to value-added management

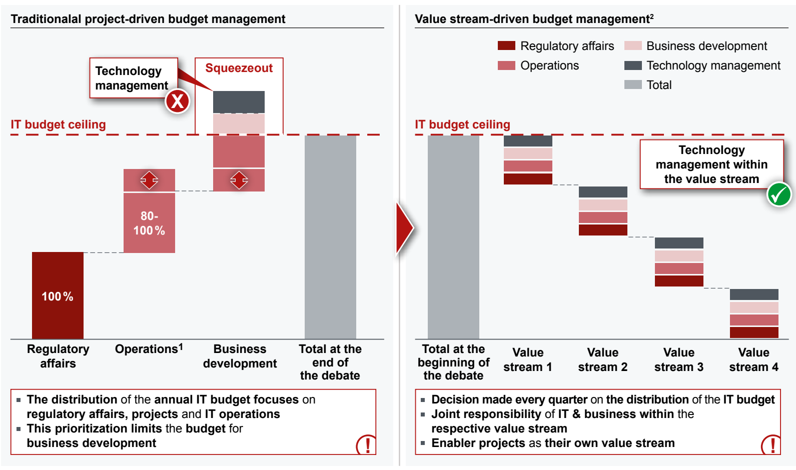
IT organized as a cost center, where there is no management that safeguards the conceptual integrity of the architecture with the necessary processes and no creation of transparency, for example through technology assessments and the analysis of the effects of individual architectural decisions, both the systems and the organizations operating them end up in a dilemma that can only be resolved by means of major transformation projects. Typical signs include an excessive run (operations) budget compared to the change, often in the order of 80% to 20%. As long as the replacement of the old platform has to be financed from the change budget in accordance with this logic and technical debts are not priced in, this is an irresolvable dilemma.

If plans involve keeping these structures for the time being, technology and architecture management processes must be established which also integrate the measurement and management of technical debt and can develop a control effect. In addition, technology life cycle management processes should control long-term changes in the technological platform. These specialized processes must, in turn, be integrated into the strategy and budget processes.

Managing overarching infrastructure requirements using a dedicated IT value stream

One way of doing this is to consider the entire IT costs for change and run over the planned term and to pair them with the anticipated benefits of IT use (so-called value streams). The value stream approach involves taking an overall view of the benefits and costs over the entire planned term of the IT solution and is the responsibility of the respective business unit. The business unit is in charge of the entire budget for the run and change of the IT solution, including the implementation of regulatory requirements and life cycle measures. Consequently, costs for omitted life cycle measures or regulatory requirements associated with the value stream can no longer be externalized and do not affect the budgets of other value streams. This means that every value stream is incentivized to implement corresponding life cycle measures in such a way that a favorable cost base is achieved in the long term and that regulatory measures are appropriately designed.

Overall view and distribution of the IT budget across value streams



[1] Includes measures to ensure ongoing operations as well as explicit change topics in operations | [2] Sample budget distribution | Source: COREinstitute 2020

Fig. 9: More flexibility with portfolio management in the shape of value streams

One disadvantage of value streams is that cross-divisional optimization is not implemented, and each business unit 'does its own thing'. IT solutions above a relevant size, where, from a comprehensive point of view, it would make sense to have joint use by several value streams, should be run as independent and possibly IT-motivated value streams.

IT budget allocation to independent business units

As for overarching technical platforms that are used by many other units within the context of infrastructure, this model also establishes a value stream, for which IT is generally responsible. This platform's requirements are then generally prioritized by IT within the budget framework (including life cycle measures). A platform should aim to achieve either a similar level of cost comparable to others in the market or to provide a higher quality of services. IT services that do not serve to differentiate the competition should consistently be sourced from the market.

A technical prerequisite for the successful implementation of budgeting geared towards value streams is a clear and functional domain delimitation via well-defined interfaces, e.g. by means of domain mapping and strategic domain-driven design, in order to decouple individual areas and to keep responsibilities and autonomy clearly defined.

Nevertheless, even in these types of organization, it makes sense to set up an overarching engineering culture or a certain overall orientation of the architecture, since the mobility of developers, and thus the flexibility of domains, is constrained by small-scale engineering.

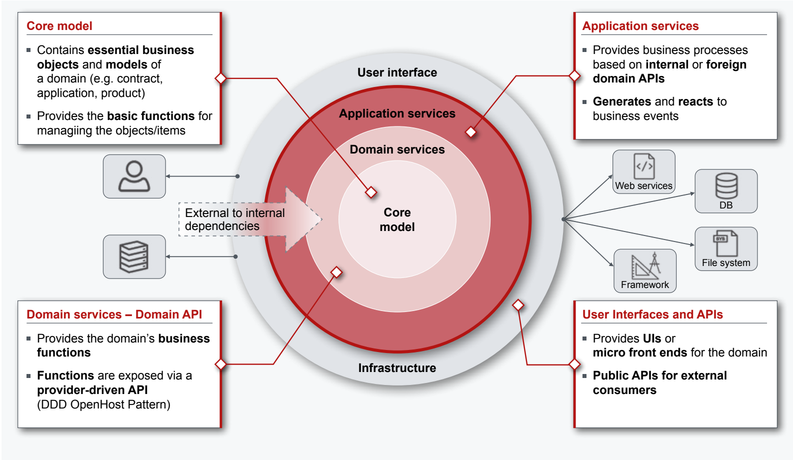
4.3 Evolutionary development instead of large-scale transformations

Modern-day approaches to architectural and technology management still make extensive use of the principles of a planned economy. The practical benefit of detailed development plans with time horizons of more than 10 years is now being called into question by ever shorter technology cycles, changes in business strategies and shifting business priorities. It is now much more a matter of developing IT systems in terms of permanent renewal and modernization. Two opposing principles can be applied here:

Ongoing adaptability by applying antagonistic principles

Disposable architectures: A definite 'end of life' (expiration date) is defined when the system is first built and a new implementation is priced in as technical debt during the budgeting stage. This approach works well for IT systems and components that are particularly prone to update cycles, including, for instance, customer front ends.
Design-to-Change: The system's architecture is geared towards maximal modifiability. Here, the principle of clean architecture is combined with modularization based exclusively on the technical function using domain-driven design. These approaches are more complex in their implementation but more robust against changes, thanks to strong modularization and technical decoupling.

Structure of domains according to clean architecture principles



Source: COREresearch 2020

Fig. 10: Principles of clean architecture

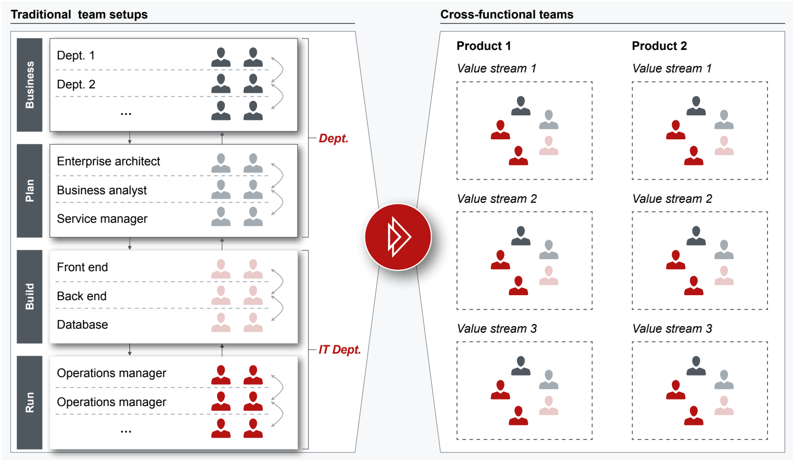
4.4 Organization must also change

New technologies, methods and changed value chains in IT also require adjustments to organizational structures. The practice of a "two-speed IT" as a combination of agile forms and digital speedboats, on the one hand, and a classic organization on the other, is usually not expedient in practice because business requirements do not just extend to the customer front end, but to the entire product cycle. Designing digital products in an agile manner and bringing them to market is only successful in the long term if the provision of services, operations and product support are able to keep pace with the further development of the products.

A high level of vertical integration requires consolidation of know-how and slows down the implementation

Even today, many companies still operate their own data centers and IT infrastructures. Consequently, many IT organizations are now more focused on the running of and less on the further development of IT systems. A very wide range of specialist knowledge needs to be built up and maintained as a result of the high level of vertical integration. To be able to map this efficiently within an organizational structure, the necessary know-how for development and operation is generally bundled in layers in specialized units. Each overarching change needs approval and coordination between the units responsible for a particular layer, making the management of changes complex and their implementation cumbersome and slow.

Adapting team structures



Source: COREresearch 2020

Fig. 11: Adapting team structures

We feel that the approach of reflecting shared responsibility between business and IT side at a product level for larger sections of the value chain is the more effective approach.

Assimilation of business and IT at a product level

However, closer integration of the business and IT side is only possible through new labor-sharing and value chains in IT and a reduction of one's own manufacturing depth. Due to the strong standardization of IT infrastructure services, these services can be sourced from specialized providers and managed by automated means using machine interfaces. The need for dedicated operational interfaces is eliminated due to the extensive automation of classic IT operations tasks. Development and operation can merge into one entity. Another advantage of the approach of joint responsibility, otherwise known as the 'Dev-Ops' model, is that it is easier to prevent short-term optimization of the development at the expense of operational reliability.

In-house manufacturing depth is reduced by sourcing standardized IT infrastructure services

If this approach is rigorously pursued, there is no longer any reason to split IT and the business side any more. The methods currently used to plan and carry out digital processes presuppose a close interaction between IT developers, Dev-Ops and those responsible. Nowadays, many companies rely on a matrix structure, which, however, is determined in the first order of IT by technology layers. This matrix structure should be transposed so that the business unit converges on IT and IT converges on the business unit. Instead of the previous layers consisting of specialists, permanent vertical structures comprising mixed teams are formed so that a requirement can be mapped end-to-end by a team. The specialist domain structure thus becomes the primary structural order. Further development of the necessary technical know-how can take place in a second organizational structure, e.g. within a guild structure as a community.

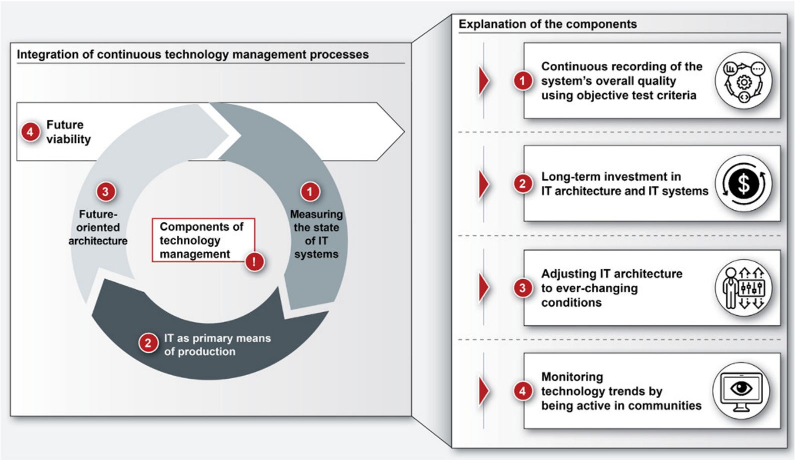
According to Conway's observation that system structures always also reflect the communication structures of an organization, changing framework conditions and division of labor and methods also require changes to the organization. Otherwise design-thinking courses, certified scrum masters, product owners and kanban boards are nothing more than a kind of agile smokescreen.

5 Conclusion

Market differentiation and competitiveness achieved through IT system efficiency

The increasing importance of information technology for business models is ever-present in virtually all industries. The quality and productivity of IT systems as the primary means of production now play a major role in a company's competitiveness. From the perspective of a business strategy, information technology systems should therefore be seen as important production assets and not exclusively as a cost factor that requires optimization. IT systems are faced with a constant aging process that is driven, on the one hand, by ongoing technological development and, on the other hand, by changing business conditions. To maintain and extend the value of IT systems, permanent investment in these important means of production is essential. This, in turn, requires comprehensive knowledge regarding the qualitative condition of the IT systems, which needs to be recorded and evaluated using objective and verifiable criteria. Not only do the 'external' operational qualities that can be directly experienced by the user play a role, but so do 'internal' values of the systems, such as structural integrity, internal and external dependencies and their degree of technical maturity.

Technology management as a prerequisite for economic success



COREresearch 2020

Fig. 12: Technology management as a prerequisite for economic success

Systematic recording of system quality is a core element of technology management

Therefore, regular and systematic recording of the qualities and deficits of the system as management variables is a significant component of modern technology management. In this respect, technology management should not be something that acts restrictively as a supervisory authority, but should try to flexibly control the system quality within a partnership. Here, methods of evolutionary architecture management can support this, the core idea of which is the ongoing adaptation of IT architectures to changing framework conditions, and thus offer better options for adapting to accelerated innovation and technology cycles than rigid planned-economy master plan models and the associated large-scale projects and transformations.

Planning and design of IT systems based on expiration dates and modularity

Nowadays, new IT architectures should always be designed with the view that they will have to be replaced, either partly or fully, in the future. For IT systems with long-term usage outlook, particular attention should be paid to modularity and loose technical coupling at the design stage. In areas that are subject to a particularly high pressure to innovate and change (e.g. customer interfaces), it may be more effective to give IT systems a kind of 'best-before' date, and to re-implement these systems on a new technology platform after the date has expired.

Technology innovation in IT along with the related productivity boosts are, to a large extent, already taking place in the global community as an evolutionary process. It is practically impossible nowadays to separate a self-developed technology platform from the rest of the world as a result of the strong working division of technology development. Consequently, active tracking of current technology trends and cycles is part and parcel of technology management. In this respect, a regular exchange of in-house software developers and architects with globally networked developer communities, combined with targeted formation of mixed teams of internal and external developers, can provide the necessary impetus. This will mean higher expenditure and does not directly pay off with a desired new function or change. However, it does ensure improved competitiveness of in-house IT systems in the long run.

Authors

Christian Böhning, Managing Partner at CORE, is in charge of planning and managing complex IT transformations, in particular as part of the modernization and realignment of IT organizational architecture. He ensures our clients' success by combining technical IT and management skills. Christian has over 20 years of professional experience in dealing with technology-driven change in the finance industry and biotechnology sector.

Christian Böhning
christian.boehning@core.se

Dr. Carsten Wedekind, Expert Director at CORE, focuses s on the strategic alignment of IT architectures. He is co-chairman of the Lending Working Group of the Banking Industry Architecture Network (BIAN). Carsten is a doctor of physics and has extensive experience in the implementing of IT projects using agile and classic methods. Prior to joining CORE, he supported well-known international companies with their adoption of standard software packages as well as with individual developments.

Dr. Carsten Wedekind
carsten.wedekind@core.se

Dr. Philipp Kleine Jäger, Managing Partner at CORE, hass vast consultancy experience in IT strategy and organizational development as well as sourcing, technology and innovation management. Over roughly 20 years, he has acquired extensive knowledge of complex financial products and systems through his work as lead architect for an international commercial bank for structured credit business and developer of portfolio management systems. In addition to classic transformation projects, Philipp has planned and implemented cloud architectures for clients.

Dr. Philipp Kleine Jäger
philipp.kleinejaeger@core.se

Francisca Di Piazza, Transformation Associate at CORE, graduated from the University of Potsdam with a masters degree in Business Administration. Her focus areas include innovation and technology management, technology foresight, identity management, data science, social media research and digital platform design. She already has experience in conducting market analyses, agile product management as well as the product design and evaluation of feasibility and implementation of a product.

Francisca Di Piazza
francisca.dipiazza@core.se

Justus Dircks, Transformation Associate at CORE, graduated from the University of Lund in Sweden with a Masters in Entrepreneurship and Innovation Management. His main topics comprise agile project management and start-ups. He has already gained experience in managing developer teams in the further development of mobile applications and various transformation projects in the banking sector.

Justus Dircks
justus.dircks@core.se

Karsten Trostmann, Expert Director at CORE, is a computer scientist and focuses on the following: IT strategy, evolutionary IT architecture, domain-driven design, agile software development and cloud infrastructures. His experience includes, among other things, the evaluation of the platform strategy for a digital insurer, the development of cloud operations for an identity provider, specialist architecture and implementation management for an online payment process and carrying out architecture reviews in various project situations.

Karsten Trostmann
karsten.trostmann@core.se

About COREresearch

As an independent technology think tank, we research the systematic technologically driven transformations in industries with a high degree of IT in the value creation process. As part of our research activities, we analyze markets and technologies, address structures, causes and mechanisms of technological change and curate results for clients and the public. Furthermore, we make selected results of our interdisciplinary research available to a broader section of the public in the form of comprehensive publications, individual studies as well as lectures.

https://core.se

Disclaimer

CORE SE
Am Sandwerder 21–23
14109 Berlin | Germany
https://core.se/
Phone: +49 30 263 440 20
office@core.se

COREtransform GmbH
Am Sandwerder 21–23
14109 Berlin | Germany
https://core.se/
Phone: +49 30 263 440 20
office@core.se

COREtransform GmbH
Limmatquai 1
8001 Zürich | Helvetia
https://core.se/
Phone: +41 44 261 0143
office@core.se

COREtransform Ltd.
Canary Wharf, One Canada Square
London E14 5DY | Great Britain
https://core.se/
Phone: +44 20 328 563 61
office@core.se

COREtransform Consulting MEA Ltd.
DIFC – 105, Currency House, Tower 1
P.O. Box 506656
Dubai I VAE Emirates
https://core.se/
Phone: +97 14 323 0633
office@core.se