Capgemini

# APPLICATIONS UNLEASHED

## EMERGING TRENDS TO UNLOCK YOUR POTENTIAL

# FOREWORD

Across the public and private sector, one thing remains true: IT is no longer about supporting your business. Instead, it will play a crucial role in defining what your business will become. Whether you want to create a more secure, more sustainable, more Agile or more citizen-orientated environment, IT will be the driving force behind it all.

Applications Unleashed brings you thought leadership from the forefront of application development. Harnessing expert insight across Agile, DevOps, Cloud and more, this report outlines the latest trends and innovations for building the next generation of powerful applications within a modern business context of constant change.

It's no longer enough to say that adopting Agile or DevOps ways of working is the sole responsibility of IT. The truly successful businesses are the ones who embed those principles throughout their entire organization. Leaders in the digital age recognize that Business and IT must come together as a united force for change, underpinned by a cloud native strategy and a keen focus on sustainability.

But it can only work if there is an open dialogue. If this document doesn't provoke a response, then it hasn't achieved its purpose. Therefore, I encourage you to challenge what you read here. Growth is an uncomfortable process, but it is often the difficult conversations that yield the greatest results.

The future of application development is uncharted waters. To navigate the seas of change, everyone must grab an oar.

Ronald van den Berg

Head of Operations
Capgemini Nederland B.V.

# UNLOCK YOUR POTENTIAL

## Applications that can handle the unexpected

### Be a technology business

Breakthroughs in AI, intelligent automation, the Internet of Things, 5G, immersive experiences and distributed platforms (and maybe soon, quantum computing) bring raw, unexplored change potential.

The pandemic necessitated the acceleration of technology-driven transformation plans in many organizations. Online channels were opened, data was shared, cloud migrations carried out, remote working enabled, sensors activated, APIs exchanged, mobile applications unleashed, and virtual collaboration was mastered.

Almost overnight, every Business became a Technology Business. And it demands application services to be built and delivered at high speed and in various incarnations, as close to the business as possible, yet responsive to every demand.

In this year's Technovision report, we advised you to "Be Like Water", capable of adapting with agility to the flow of evolving demands and technology. Now, Applications Unleashed invites you to harness these emerging trends to unlock your business potential.





### StratOps

Application development is an area that has already pioneered capabilities for years. Agile, DevOps and continuous delivery have been paving the way towards the creation of an always up-to- date, flexible portfolio of application services. Even APIs and microservices, which are now evolving towards a distributed "service mesh", are a testament to radical portability and adaptability, both inside and outside the organization.

What if one day, strategy could be as smoothly delivered as a software update? Let's refer to this as "StratOps": applying the DevOps approach enterprise-wide to achieve continuous strategy development and delivery. Just as we have learned from years of applying Agile and DevOps ways of working, StratOps will depend on a powerful, highly automated technology platform, cross-discipline teams and a healthy, culturally established appetite for rapid change. An enterprise that adopts a StratOps approach, that can deal with a shifting business context, disturbing challenges and arising opportunities in equal measure, can literally see itself as renewable – easily morphing to a new state – and portable – smoothly readjusting to a changing external context.

### Dimensions

In order to truly succeed, the applications portfolio of a thriving Technology Business must be lightweight, easy to connect to, and built on the shoulders of typical cloud-native capabilities, predicting and adapting to ever-changing demands and requirements. To unlock the potential of these true Technology Business applications, various dimensions should be considered - and all of them surface in different ways within this publication:

- **Simplify, rationalize and decommission existing applications.** Once perceived as differentiating solutions for business growth, many applications have become a massive, rusty anchor. They are heavy, cumbersome and slow to move, let alone adapt. Faster and more flexible applications enable a speedier response to change, and standard, industry best practice cloud applications are a quick way to deliver exactly that. Hence, sometimes the best way to rebuild an application is to not build it all. Loosely coupled service layers (for example through bots, APIs and Robotic Process Automation) over siloed applications are another option. Whichever route is chosen, it needs a dedicated captain and the decisive mind of a tidying-up guru to actually get it over with.
- **Always be delivering.** Applications should be continuously built and released – DevOps-style. Quick iterations should be delivered by fully merged (not merely "aligned") teams of Business and IT professionals, leveraging microservices, APIs, software containers, serverless computing and radically automated, high-productivity tools. Speaking of the latter: AI augments both specialized software engineers and citizen developers in building next-generation applications without (or almost without) any coding. From there, it is only a short step towards applications services that learn from flaws and new needs, coding or recoding themselves automatically and continuously. So I say again, as we observe the constant evolution of AI capabilities, that the best way to build an application may, in fact, be to not build it ourselves at all.
- **Add a touch of "smart" by augmenting existing or newly developed applications.** AI services for areas including vision, speech, language, knowledge and predictive analytics are routinely available as microservices, all while avoiding the alien realms of deep learning, neural networks, reinforcement learning and computer linguistics. And to ensure a steady flow, easy-to-use, conversational interfaces can be added – such as voice assistants or bots – to make application services much easier and more accessible to users. The ultimate application may be no application. The Application of the future may become so adapted to - and knowledgeable of - its users, that it anticipates their intentions even before they are expressed.

What will always be true is that trends come and go, certainly in the realm of applications and application development. We're never quite there, never established. Thus, we need to continuously embrace what the future brings us. And if there's one thing you can always count on, it's the continuous cycle of revolutionary technology.

Ron Tolido

Chief Technology Officer
Capgemini Insights & Data
Global Business Line

# CONTENTS



01

# BREAK AWAY FROM THE MONOLITHS

1. Need an APP? Build It Yourself.
2. LEGO©-fying your digital Platform is the key to bringing speed and adaptability to your global enterprise
3. Why you should opt for cloud native, even if your competitors don't
4. How Open Source and cloud native technologies are re-inventing the agile enterprise.
5. Domain Driven design is the key to adopting digital transformation
6. Building progressive web applications
7.

# NEED AN APP? BUILD IT YOURSELF

To stay competitive, businesses are increasingly empowering their teams to become citizen developers, developing their own applications on low code/no code platforms.

**IN A NUTSHELL**

- Application development is becoming increasingly accessible to non-developers.
- Low code/no code platforms are fostering wider adoption of "citizen development".
- Citizen development enables business teams to take charge of their own application development, bridging the gap between business and IT, while realising more value, like shorter time to market, reduced lead time, cost savings and better end-user satisfaction.
- The adoption of low code/no code platforms takes the strain off of already overstretched IT departments, enabling technical developers to focus on more complex (IT) challenges.

It's a problem many companies are facing today – how do we meet increasing business demands and stay competitive with limited technology resources? It's impossible to bring new ideas to market if your IT department has an 18-month backlog on existing projects and spends 90% of their budget on legacy. Emerging low code/no code (LCNC) platforms are coming of age and offer an attractive solution to these problems. Citizen development gives this a further boost by empowering business teams to take charge of their own transformation and take the wheel in terms of software development.

According to Gartner, who are credited with coining the phrase "citizen developer" to refer to non-specialist software designers, by 2023 "the number of active citizen developers at large enterprises will be at least four times the number of professional developers"[1]. This DIY mobilization promises to help relieve some of the pressures building in IT, while delivering a variety of business benefits at the same time.

## Tech solutions are becoming business solutions

Companies that adopt citizen development can expect to see a shorter time to market for products, better end-user satisfaction, and more realistic deadlines and budgets. Intended to be a way of empowering people across an organization, citizen development helps bridge the gap between business and IT, ensuring that application development is seen both as a business and technological solution.

Powering the citizen development phenomenon is the rise of LCNC platforms. Marked out from traditional IT development by their simplicity and ease of use, these platforms enable the creation of web applications using standard components that can often be combined by simple drag-and-drop functionality. They either require very little written coding or none at all – hence the term "no code". Another advantage is that they often work "out of the box" and support a variety of features such as one-click deployment to the cloud, solid security, and built-in scalability.

**Forrester already marked the low-code market as "high growth"** back in 2019, and it has been expanding aggressively ever since[2]. Next to all the apparent benefits in boosting productivity and automating workflows, there are major considerations to be made regarding how LCNC environments are implemented at the business level. For one thing, what's good news for business often sounds bad to IT specialists. If we get to the point where even "no code" employees can build apps on top of existing APIs, what is the role left for experienced pro-coders?

## Getting IT right – it's all about oversight

Yet citizen development isn't about replacing IT departments wholesale. By "citizen developer" we mean a non-professional software developer, who makes applications that are in turn used by people from within or even outside the organization. These developers may not report directly to an IT department, but the tools they use are still approved by IT. This is in contrast to the destabilizing effects of "shadow" IT, where unapproved software and applications are used by non-IT specialists in the workplace with the associated risk of data breaches and loss of continuity.

Tech experts are still at the heart of the process with citizen development. Even if, say, 80% of development on a particular app can be developed by citizen development, IT professionals still need to be involved to ensure integrity, through code review, building standard components and integrations, and to perform automated testing. This puts IT in greater control of overall development by giving tech experts oversight of the whole process, from concept to deployment.

**Steve Jones, Capgemini's Chief Data Architect, recently stated:** "To make no-code succeed for less technically adept users, you need more rigor not less in the underlying infrastructure, you need to turn the complex into the simple and make governance and security integral parts yet magically transparent to the end user" [3].

Widely viewed as a DIY model, citizen development still needs support from a clear governance framework. From a privacy and security perspective it's essential to have guidelines and safeguards in place, not least to protect company and customer data. The challenge is making sure these standards are embedded in the business culture and evolve alongside technology changes. In a citizen development environment where new apps are created on a daily basis, a specialist department needs to support and examine ideas and apps as they are built, rather than sending them through an existing approval process that could take weeks.

## Making no code a reality: innovate, optimize, scale

Increased speed at lower cost is one of citizen development's most obvious attractions. Through Microsoft's Power Platform, which is available with most Microsoft licenses, users can simply log into Power Apps, start developing an app and have it live on their phone in minutes. This kind of rapid deployment, however, presents major challenges to governance, many of which are being worked out by market leaders in citizen development such as Betty Blocks, Mendix and OutSystems. As always, it's a balancing act. However, the rewards in terms of accelerated time to market, value and cost reduction are too big to ignore.

At Capgemini, the Innovate-Optimize-Scale framework[4] outlines three phases in a citizen development-type model. An experimental collaboration brings together key players to kick-start the process, which is then structured to improve processes and governance before moving into a scaling phase where the business is essentially running the service on their own. In this kind of deployment, citizen development adds real value through rapid DevOps and close collaboration between different personas, fostering an Enterprise strategy that fuels digital transformation, innovation, and continuous improvement.

As with the rise of any new innovation in technology, it's worth considering the drawbacks alongside the advantages. Demands on existing IT staff could change and there are some clear ethical boundaries – the right platform has to be in place to make sure, say, customer data doesn't end up out in the open. As we understand these issues, we can tackle them by applying the right governance models. Equally, for IT professionals already caught up in the war-for-talent, there's the excitement of rising to meet the new challenges alongside the possibility of taking on more interesting coding work, as the less complex tasks are handled by citizen developers and the platform they use.

Put simply, citizen development doesn't change the fact that IT will always need talented developers.

Citizen Development is already changing the way the Dutch National Police manage their IT operations. Using the citizen development platform Betty Blocks, they have introduced app development that's so easy, police officers can build their own applications. They started with mixed teams of high coders and citizen developers to pave the way and create a centre of expertise. The Betty Blocks platform provides the security measures the police require for software development, even for applications with the highest confidence levels.

**"By empowering our people to build their own applications, we've diminished lead times and operations by 50%, and we are just getting started. In our industry that is literally unheard of."**

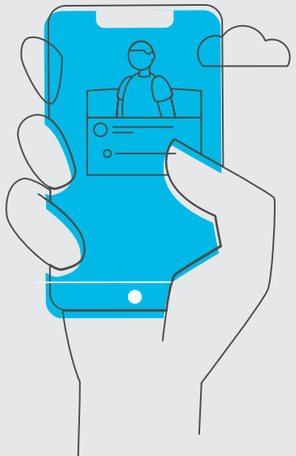
**– Steven Hanekroot, Innovation Manager Dutch National Police**

## Building a secure IT-business partnership

Even though citizen development is still in an early phase, it's easy to see why it looks increasingly attractive to companies who want to handle more of their IT in-house. As the Project Management Institute has pointed out, "With low code/no code platforms, anyone can build applications without software expertise, significantly faster, and at a fraction of the cost"[5]. With the added benefit of self-supporting operations, it looks even harder to find reasons not to consider low code as part of your future business model.

With inquiries coming from manufacturing, retail and (local) government in the Netherlands, as well as globally, Capgemini is putting an increased focus on Competence Centers and Centers of Excellence (CoE). It's all about setting up the right framework to enable and support citizen development as a part of an end-to-end portfolio. And even though there's a drive for DIY solutions, thanks to close relationships with leading LCNC vendors, as well as leading customers and their best-practices, Capgemini can also step directly into a citizen development environment and improve your practices and results. Interesting best practices include combining microservices with low code/no code and BizDevOps, where business teams work within DevOps teams as citizen developers.

Even with all these positives supporting wider adoption, citizen development isn't an answer in itself, and needs to be seen in the context of bigger shifts in business methods and work behaviors - especially given the changes brought on by the COVID-19 pandemic. But the strong appeal of citizen development with its low code/no code ecosystem lies in its adaptability and potential to act as a driver for stronger, deeper partnerships between IT and business operations. A way for companies to further leverage tech innovation and do more with what they already have? This is almost certainly an opportunity you cannot afford to miss.




Marien Krouwel

Expert in Agile, Cloud Services, DevOps, Low Code

LinkedIn


### About the author*

Marien Krouwel is an expert in low code/no code and Cloud Services who helps customers accelerate and scale digital transformation and innovation, by implementing low-code/no-code platforms and setting up development centers for both citizen and tech developers.

*This article was written by an external agency through interviews and in close collaboration with the expert.

marien.krouwel@capgemini.com

### Footnotes

[1] https://www.gartner.com/en/documents/3970067/the-future-of-apps-must-include-citizen-development

[2] http://www.xorlogics.com/2018/09/17/low-code-5-success-factors-for-digital-transformation-projects/

[3] https://www.linkedin.com/pulse/no-code-again-steve-jones

[4] https://www.capgemini.com/service/cloud-services/cloud-native-applications-paas/

[5] https://www.pmi.org/citizen-developer

### Further Reading:


https://searchsoftwarequality.techtarget.com/definition/citizen-development

https://www.uctoday.com/collaboration/team-collaboration/the-rise-of-the-citizen-developer-and-what-it-means-for-business/

https://blog.bettyblocks.com/is-no-code-the-future-of-application-development

https://www.bettyblocks.com/citizen-developer

https://www.mendix.com/citizen-developers/

https://www.mendix.com/blog/citizen-development-is-here-to-stay/

https://www.outsystems.com/blog/posts/citizen-development-readiness/

https://www.mendix.com/citizen-developers/

https://www.yash.com/blog/is-microsoft-powerapps-only-for-citizen-developers/

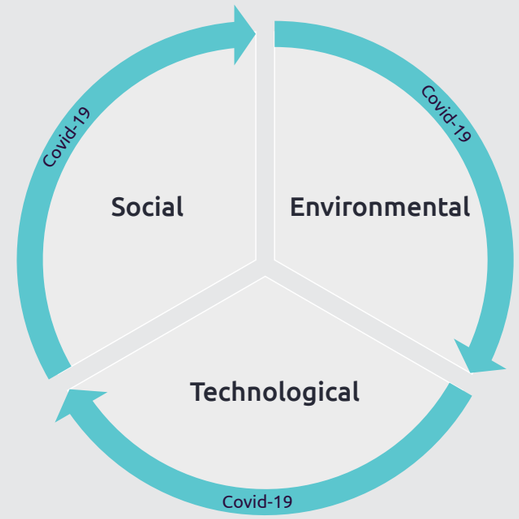
# LEGO©-FYING YOUR DIGITAL PLATFORM IS THE KEY TO BRINGING SPEED AND ADAPTABILITY TO YOUR GLOBAL ENTERPRISE

To survive the accelerating digital revolution, organizations need to adopt new mindsets, structures, processes and systems to stay relevant.

**IN A NUTSHELL**

- Unpredictable change is setting the pace for your business, so continuously and rigorously improving the speed and adaptability of your digital platform is key.
- "Lego-Fy" your digital platforms to ensure both speed and adaptability – move from a monolithic to a MACH (Microservices-based, API-first, Cloud native, and Headless) platform.
- Use the strangler pattern: gradually improve your platform for speed and adaptability to keep up with accelerating change.

**Figure 1:** SET model: Social, Environmental, Technological factors and COVID-19 are accelerating the rate of change in today's business environment.




The need for successful business model innovation to surviving the roaring twenties of the digital age cannot be emphasized enough. However, only focusing on new (digital) business models will not be enough to survive this decade. A lot of organizations are still working with the mindset, organizational structures, processes, and systems stemming from the Industrial Age. Unfortunately, these have all been optimized for efficiency gains in a more predictable business environment, and are not able to cope with the accelerating change we are experiencing today.

## Unpredictable change is setting the pace for your business

Continuously and rigorously improving the speed and adaptability of your digital platform is a key ingredient to surviving the coming decade. It's crucial to be able to respond to the never-ending and accelerating change we are experiencing today. Within this article, I will highlight only some of the most impactful factors determining the current rate of change, and therefore the shift in business models that we are currently seeing.

In essence these factors are both directly and indirectly "SETting" the pace of today's business model innovation - and the current COVID-19 crisis is causing a flywheel effect.

**Social:** With the growing importance of customer centricity, digital, and data driven insights, the way people engage with companies and brands has changed. COVID-19 has accelerated this change even further in B2C and B2B alike, with digital and remote sales taking precedence.

Companies can no longer get away with serving their customers exclusively via face-to-face or phone contact. Nor can they rely on customers placing orders via fax or email. An increasing number of customers expect companies to provide them with self-service capabilities by offering them digital tools like a webstore, a Configure-Price-Quote (CPQ) solution or interactive engagement with both actual service representatives as well as chatbots.

In the end people are social and creative animals, and they need to have interactions with other people. It's natural that they will always look for new ways to foster these interactions, and the current pandemic is leading to new business models.

**Environmental:** Global concerns around the environment and climate change are becoming increasingly important for businesses today.

Large organizations cannot permit themselves to neglect these concerns, and need to focus on corporate social responsibility (CSR) and sustainability. As a result, new business models are emerging like re-commerce and circular shopping. Research from the Capgemini Research Institute suggests that "a significant majority of consumers (79%) are changing their purchase preference based on sustainability."[1]

**Technological:** The impact of cloud and digital technology has of course been enormous in the last decade. The number of available digital technology solutions has exponentially grown and is not expected to slow down considering current developments around AI, Machine Learning, Augmented and Virtual Reality, Mixed Reality, and introduction of 5G. Of course, the COVID-19 pandemic has led to a sharp spike in the adoption of these new technologies. For example, we have helped one of our clients in Belgium build a click and collect capability for all their stores at record speed in order to be ready for the second wave of COVID-19.

In short, these three factors are reinforcing each other. The pandemic is adding a flywheel effect on top, accelerating change in the global marketplace.

## Traditional global enterprises are slow to respond to this unpredictable change – and it's putting them at risk

Traditional practices were quite useful in an environment where change was relatively easy to predict. This led to traditional structures that are still dominant in most large enterprises, in which different divisions and business units are organized in functional silos.

This has also influenced the IT systems that these enterprises are implementing. In effect, the current IT systems in place focus on these silos, are not collaborative, and there is no integration with other silos within the organization. This often leads to, for example, a digital commerce system that is not integrated with marketing, service and sales systems and processes, resulting in a highly fragmented and inconsistant customer experience.

To add to this, Global enterprises are typically organized in divisions, business units and regions. Business units have their own GTM strategies, brands and target segments which need to be adapted to the local culture of the countries they are operating in. This leads to even more siloed systems for these various business units and/or brands, and the risk of such a global-local fragmentation is huge.

Consider this - When performing a global website scan at one of our clients, we found few websites were well managed and maintained. This meant an increased risk of being hacked, resulting in the potential theft of customer data, losing the trust of the customers, fines from authorities, and damage to the brand. Hence, siloed systems and a fragmented approach also pose security threats to an entire organization, not to mention the reputational damage that a data breach could cause.

In addition to this, global enterprises are subject to mergers and acquisitions on a regular basis. Acquisitions come with their own culture, processes and systems, which will need to be integrated into the larger entity, causing the enterprise to become even less responsive to change.

As is evident, the prevalent structures within digital & IT teams in large enterprises lead to monolithic, tightly coupled systems which are difficult to manage and change.

### What's the solution?

It is absolutely possible to build a large enterprise that is responsive to change. Amazon is a fitting example of an organization that is very effective in bringing new innovations to life at speed. It has also proven to be capable of effectively integrating an acquisition like Whole Foods,[2] creating real business synergies in the process. One might counter that Amazon was originally born as a pure-play online retailer. Nonetheless it has deliberately evolved its platform to be open and adaptable, as can be illustrated by the Bezos API Mandate memo[3]. Amazon has practically invented building API-first, microservices based, cloud native applications and, in fact, the current AWS platform is a result of the practices employed by Amazon in building their digital commerce platform.

Traditional global enterprises will need to learn from resilient, future-facing companies like Amazon if they are to become more responsive to change.

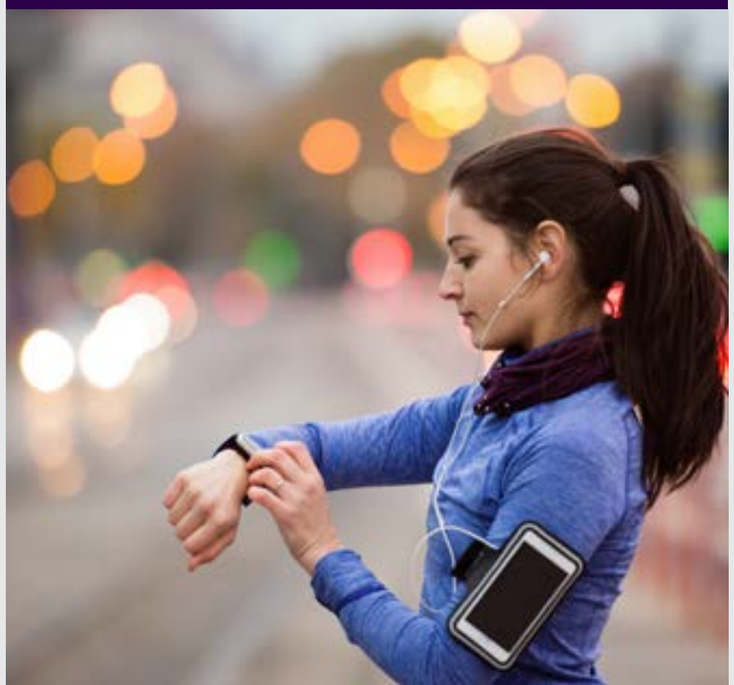
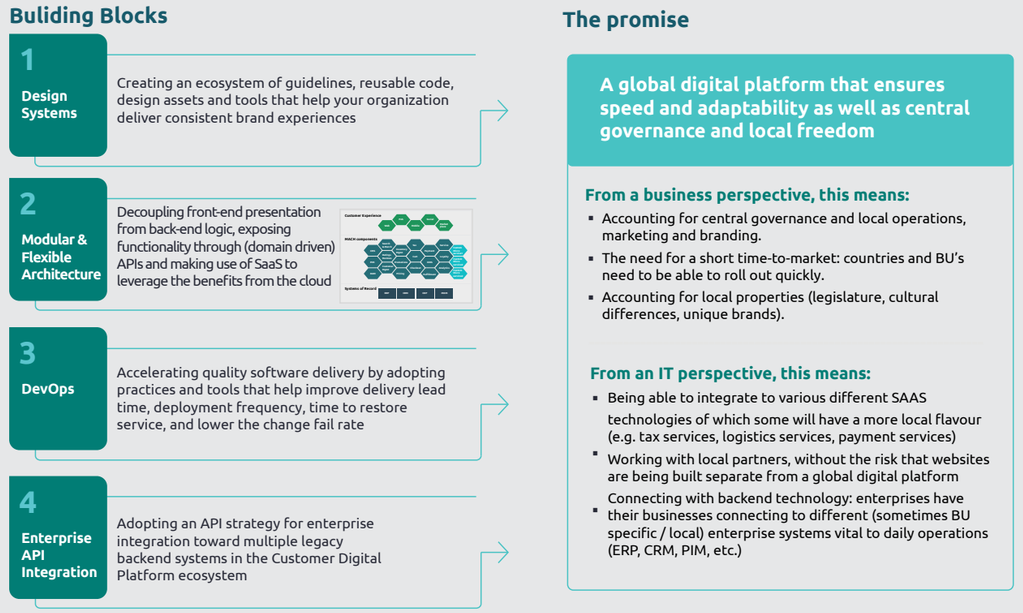


**Figure 2:** "Lego-Fying" digital platforms to ensure both speed and adaptability as well as central governance and local freedom.




## "Lego-Fying" your digital platform

In order to respond quickly to this accelerating change, large global enterprises will need to (gradually) move to a digital platform that is optimized for speed and adaptability. In order to do this, enterprises will need to put the right building blocks in place that can be independently changed or replaced with other small building blocks, thus creating a platform that can be adapted quickly when the situation demands it. In effect these building blocks "Lego-Fy" the global digital platform.

### 1. Design systems

Design systems will help easily build the different user interfaces (heads) to interact with end-customers both online (web, mobile, social, marketplaces) and offline (Point of Sale). It is about creating an ecosystem of guidelines, reusable code, design assets and tools that help your organization deliver consistent brand experiences across all of your different customer interaction channels. More details on design systems can be found in our Idean Expert Book 'Hack the Design System[4].

### 2. Answering monoliths with MACH - A Modular and Flexible architecture

Implement small building blocks that can be changed by independent teams without dependencies on other building blocks in the platform. In practice this can be done using SaaS components or microservices that cover (part of) a business domain. These components and microservices are API-first and API-only. Because these are API-first it also means that they are headless, so different frontends can use these applications to serve a specific need of your end-customers.

These applications are also cloud native, which makes them horizontally scalable. It's worth noting that the industry uses different names for this kind of architecture - composable architecture, composable commerce or a MACH (Microservices-based, API-first, Cloud native, and Headless) platform.

Small building blocks that can be changed by independent teams without dependencies on other building blocks in the platform. In practice this can be done using SaaS components or microservices that cover part of a business domain.

**3.DevOps practices and tools**
Accelerating quality software delivery by adopting practices and tools that help improve delivery lead time, deployment frequency, time to restore service, and lower failure rates.

**4.Enterprise API integration**
Large enterprises typically have multiple backend systems where the master data (product, customer, price, stock availability) that is needed to interact with the end customer is residing. It's not realistic to assume that it is possible to get this data into one single source of truth. Therefore, it is best to manage this data, that comes from multiple systems, in a federative manner by adopting an API strategy for enterprise integration toward multiple legacy backend systems.

## The strangler pattern: gradually improve your platform for speed and adaptability to keep up with accelerating change

The need of the hour is to transform and adapt, but as outlined above, organizations need to change not only their mindset but also their structures and processes, moving away from monolithic platforms in favor of more composable solutions. That said, for large enterprises who already have one or more monolithic digital commerce platforms in place, it may be risky to undertake a big bang re-platforming project. Instead, it may be a better idea to gradually move away from a monolithic platform architecture in favor of a highly modular and flexible architecture by following a "strangler pattern" in small phases and sprints.

**Figure 3:** Applying the strangler pattern to gradually move away from a monolithic toward a composable platform architecture.

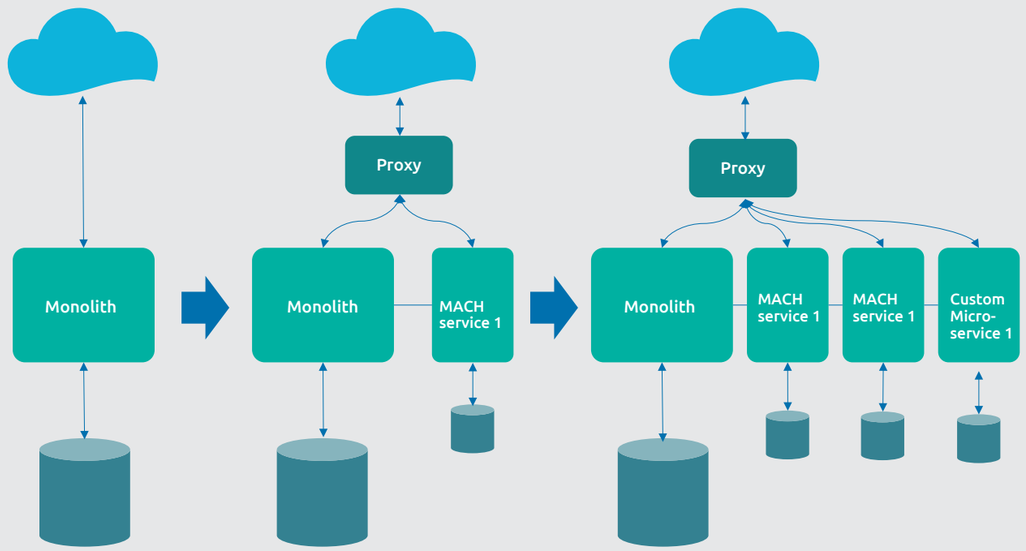
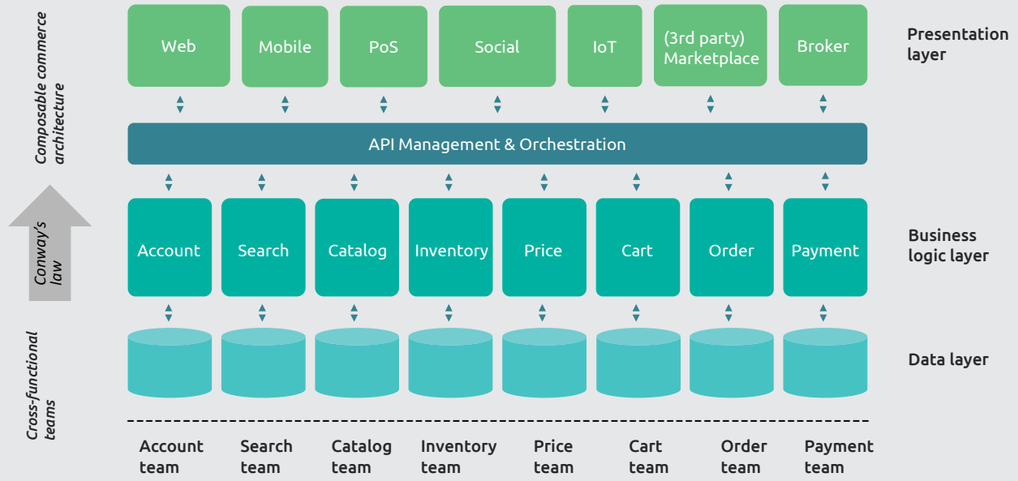



**Figure 4:** Simplified view of the resulting MACH architecture after applying the strangler pattern.




Features that have been built in the monolithic platform will be gradually replaced, either by relatively standard MACH services from specialized SaaS vendors or by bespoke microservices that are needed to differentiate the specific enterprise from its competition.

In the end, the resulting architecture will look more like the figure above, providing a more agile structure.

## Striking the right balance between central governance and local freedom

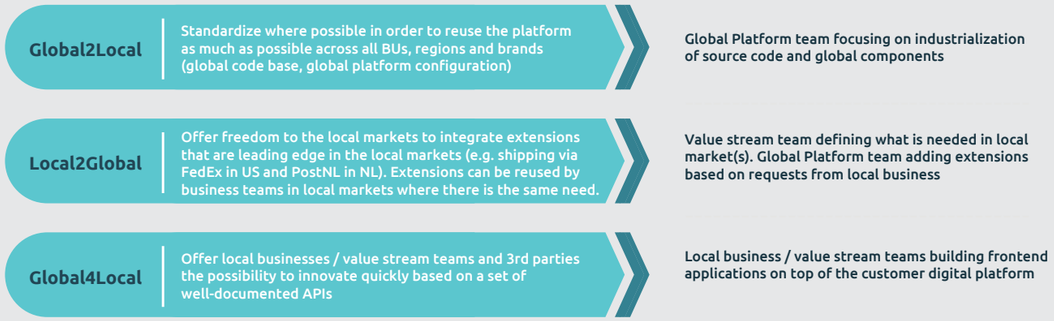
To address the earlier point of the huge risk of a global-local fragmentation in a global enterprise, it's important to question how enterprises will be able to strike the right balance between central governance across all the business units, brands, and countries, while making use of the global platform and, at the same time, offering their local businesses enough freedom to serve their customers in the best possible way.

We faced this question while collaborating with a leading global Paints & Coatings manufacturer to build a truly global digital platform. We started our collaborative MACH journey with two specific commerce use cases for two of their business units using Commercetools as the MACH commerce engine in combination with the Azure platform, GraphQL, and Angular.

This architecture allows the different squads working on the platform to individually design and deploy new APIs when needed. When the platform matures, existing microservices can be easily replaced with third party services. The microservices will be deployed by using serverless infrastructure so that it can be scaled, versioned, secured, and deployed at the API level.

For this purpose, we are also utilizing Lab Digital's MACH composer that will help manage configuration of environments (infrastructure-as-code) across our client's different business units, regions, and brands. This gives us the opportunity to adopt three different approaches in parallel, to build a global digital platform that strikes the right balance between central governance and local freedom.

**Figure 5:** Adopting three approaches in parallel to strike the right balance between global standardization and local freedom.




By following the three approaches outlined above, we are able to iteratively work towards a global unified commerce platform that can serve all of our client's end customers during all the stages of the customer journey. The global enterprise will benefit from global standardization at the same time as offering freedom to the local businesses to service their customers during all the stages of the customer journey in the best possible way.

### Embrace MACH to be future-ready

Bringing speed and adaptability to your global enterprise is key to surviving the coming decade as we experience unpredictable, never-ending and accelerating change in the marketplace, customer expectations, and digital technology. Traditional enterprises are often not well equipped to respond to this change and must, therefore, transform themselves to keep delivering value to their customers. Highly flexible and modular digital platforms like MACH make you future ready and capable of responding to such changes, while also being optimized for both global standardization and local freedom.

Robert kooloos

Customer Experience (CX) experts

LinkedIn

### About the author

One of Capgemini's foremost Customer Experience (CX) experts, Robert supports clients in their journey towards becoming a digital business. He works with clients to help them move toward highly flexible and modular (composable, MACH based) digital platforms and, in parallel, helps them shape their lean/agile delivery and operating models that cover outcome-based organizational alignment, value-driven funding and prioritization, lightweight planning and governance, self-sufficient and autonomous teams.

robert.kooloos@capgemini.com

### Footnotes

[1] https://www.capgemini.com/wp-content/uploads/2020/07/20-06_9880_Sustainability-in-CPR_Final_Web-1.pdf

[2] https://www.digitalcommerce360.com/2019/06/21/how-amazons-whole-foods-acquisition-changed-the-grocery-industry/

[3] https://api-university.com/blog/the-api-mandate/

[4] https://www.idean.com/micro/invent-design-systems

### Further Reading:

https://kooloos.medium.com/is-your-business-getting-scrooged-34ae8b1d6176

Future-Proof Your B2B Commerce Strategy To Meet Customers Where They Buy (hyperlink: https://www.forrester.com/report/Digital+Ups+The+Stakes+For+B2B+Sales+Pros/-/E-RES126861#)

www.businessinsider.in/tech/5g-is-being-used-to-perform-remote-surgery-from-thousands-of-miles-away-and-it-could-transform-the-healthcare-industry/articleshow/70699125.cms

The 2020 State of DevOps Report is here! Hyperlink: https://puppet.com/resources/report/2020-state-of-devops-report/

How sustainability is fundamentally changing consumer preferences? [Hyperlink: https://www.capgemini.com/research/how-sustainability-is-fundamentally-changing-consumer-preferences/]

Dont let technical debt stop innovation [Hyperlink: https://www.linkedin.com/pulse/dont-let-technical-debt-stop-innovation-gunnar-menzel/?published=t]

# WHY YOU SHOULD OPT FOR CLOUD NATIVE, EVEN IF YOUR COMPETITORS DON'T

When it comes to Cloud Native, it's no longer a case of 'if', but rather, 'how soon'.

**IN A NUTSHELL**

- Applications have become very complex and users are demanding more and more of them. Applications need to be able to address the need for speed and agility. And the solution to this is: Cloud Native architecture and technologies.
- Five pillars that provide the foundation for building Cloud Native applications: modern design, microservices, containers, backing Services, automation.
- The Cloud Center of Excellence is an ideal model to accelerate your cloud adoption program.

Cloud Native is a method for software development that harnesses cloud computing to build & run applications in public, private and hybrid cloud environments. The reason you hear so much about Cloud Native applications and systems is because they are highly scalable, resilient and manageable, and are designed to respond quickly to large-scale and rapid change.

For a growing number of organizations, the appeal of Cloud Native adoption is clear: building a nimble, agile enterprise that moves at the velocity of your business's transformation needs. Cloud Native is all about changing the way you think about building and designing your critical business systems.

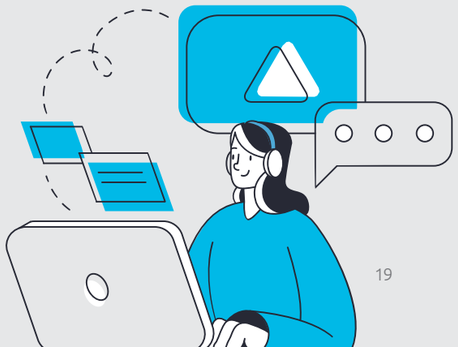
Faster time-to-market, increased cost effectiveness, and scalability of operations can all be considered as the main drivers for adopting a Cloud Native approach.

## The sector agnostic solution

Cloud Native can be applied to any sector. For instance, the retail industry generates substantial quantities of data, all of which can be easily stored and analyzed using cloud technologies and AI solutions that are managed within the cloud. Healthcare can benefit strongly from a single secure place to store information, such as personal records. IoT and Big Data, which also processes a lot of its own data in the cloud, has played a large part in helping to innovate the Agriculture sector. Even the government is embracing the cloud as a solution to help secure their data and services. In this case, you often see cloud providers offering government-specific versions of the cloud.

If we observe trends that we know will dominate in the coming years – IoT, mixed reality (MR, the merging of real and virtual environments), data driven AI in which cloud acts as the backbone – then embracing cloud technology is no longer an option, especially for organizations that are still clinging on to legacy application landscapes.

As organizations and businesses across all sectors become increasingly acclimatized to a "new normal", in which a surgeon can remotely assist in a procedure, a mechanic can check your car engine from a different city or country, and 3D modeling can be used in healthcare to plan surgeries and treatment, the shift to scalable and dynamic options offered by Cloud Native seems to be not only increasingly relevant, but increasingly imminent.



## Cloud Native is nothing without a well architected framework

Cloud Native systems run in modern and dynamic environments, be they public, private, or hybrid clouds. The Cloud Native applications are mostly built using one or more of the following technologies: containers, service meshes, microservices, and declarative APIs, running on immutable infrastructure. The speed and agility of Cloud Native comes from several factors. Of course, cloud infrastructure is key here, but there are five additional pillars that also provide the foundation for building Cloud Native applications[1]:

### Modern Design

A widely accepted methodology for constructing cloud-based applications is the twelve-factor app methodology. It describes a set of 12 principles and practices that developers follow to build as-a-service applications or web applications that are optimized for modern cloud environments. The primary focus is on portability across environments and declarative automation. These principles and practices are considered a solid foundation for building Cloud Native apps. Systems built upon these principles can deploy and scale rapidly and add features to react quickly to market changes.

### Microservices

Cloud Native systems and applications embrace microservices, which is a popular style for building modern applications. The microservice architectural style is an approach to developing a single application as a suite of small services, each operating independently but capable of communicating and exchanging data with other applications. Microservices can scale independently, so instead of scaling the entire application as a single unit, you scale out only those services that require more processing power or network bandwidth. Because the services act as individual components, they can be reused time and time again, which in turn helps to foster a level of standardization.

### Containers

Containers are a great enabler of Cloud Native systems and applications, providing portability and guaranteeing consistency across environments. By packaging everything into a single container image, you isolate the microservice and its dependencies from the underlying infrastructure. This also eliminates the added expense of pre-configuring each environment with frameworks, software libraries, and runtime engines. And by sharing the underlying operating system and host resources, containers have a much smaller footprint than a full, virtual machine. This increases the number of microservices that a given host can run at one time.

### Backing services

Cloud Native applications and services depend upon several different backing services, such as data stores, monitoring, caching, logging services, message brokers and identity services. Cloud Native services typically use backing services from cloud providers, which saves time and reduces costs and operational risks of hosting your own services.

### Automation

The previous pillars specifically focus on achieving speed and agility. But cloud environments also need to be provisioned to be able to deploy and run Cloud Native applications and systems.

So, how can you rapidly deploy your apps and features? A widely accepted practice for this is Infrastructure as Code (IaC)[2]. Using IaC, you can automate platform provisioning and application deployment. DevOps teams that implement IaC can deliver stable environments rapidly and at scale. By adding testing and versioning to the DevOps practices, your infrastructure and deployments are automated, consistent, and repeatable.

## Cloud Native done right

Companies like Netflix, Uber and WeChat have all implemented Cloud Native and achieved speed, agility, and scalability. They have thousands of independent microservices running in production that they deploy between a hundred to a thousand times a day. This architectural style enables them to quickly respond to market demand and conditions. By using a Cloud Native approach, they can instantaneously update small areas of a live, complex application, and individually scale those areas as needed.

Our customer, who helps Global and Dutch retailers and brands conceptualize and deliver fully integrated customer acquisition, engagement and retention policies, wanted to offer new innovative services to their clients, but their existing IT landscape wasn't ready. To offer these new services from their on-premises environment would require a complete renewal of their servers, acquiring new software and licenses, and hiring employees to install and configure them. Not only would this approach have resulted in huge costs, but it would have taken a long time to actually implement.

The cloud offers them immutable infrastructure, with services such as data storage, data analysis, security and identity solutions, and application development solutions, all using a pay to go model.

We helped them implement Cloud Native services to renew their IT landscape, offering their customers a set of services that are specifically designed for performance, security, and redundancy and provides real-time insights on data coming from various sources. This data is partially stored in Azure, but also in on-premise databases.

### How we did it[3]

Together with the client, we decided to take advantage of all the Cloud Native services that Azure has to offer, from a microservices and data analytics and insights perspective. With our offering, we are providing our client with:

**Domain-driven design (DDD):** When implementing a microservices architecture, DDD is a design approach you can benefit from. Where to draw the boundaries is the key task when designing and defining a microservice. DDD patterns help you understand the complexity in the domain.

**Cloud Native design patterns:** To build highly reliable, scalable, secure applications and services, every developer needs to make use of common Cloud Native design patterns. We focus fully on implementing Microsoft best practices and patterns.

**Dev/test optimization:** We bring our own development and test environments to the project. For this, we use container technologies, which have all the commonly used tooling and software pre-deployed. Next, we use automated performance and acceptance tests, fully integrated in Azure DevOps.

**Everything-as-code:** We offer out-of-the-box landing zones, which include security and compliance policies and monitoring rules. These monitoring rules are based on our experiences and best practices that we have developed over the years managing cloud environments for our global customer base. We are implementing zero-touch deployments using Azure DevOps and CI/CD pipelines for automatically building and releasing applications and services.

The cloud offers them immutable infrastructure, with services such as data storage, data analysis, security and identity solutions, and application development solutions, all using a pay to go model.

## How we implemented it

By using landing zones, Cloud Native patterns and Microsoft best practices, and securing it using Azure Policies and Azure Active Directory in our solution, we now have a solid foundation for rapidly building and deploying additional services. At this stage, we have successfully implemented a set of secure microservices for the client, which are automatically deployed across environments, securely connecting to an on-premises SAP environment, and exposed via a single gateway.

Next, we will be implementing the second project, where we will form an additional DevOps team that will implement the solution for storing customer data, and provide real-time insights.

## All organizations need to build a Cloud Native culture

Cloud Native is the only way forward – there is no alternative on the market. And yet, some businesses are still dragging their feet.

The biggest deterrent for Cloud Native laggards is the extra costs associated with moving legacy applications to the cloud. The bad news is, there is no other option. Moving to cloud is the only way to be truly future-ready. Organizations need to embrace Agile and DevOps ways of working in order to be Cloud Native compliant—and this involves a culture change.

For Tata Steel Europe (TSE), we built a Cloud Center of Competence (CCC) in collaboration with Microsoft. The CCC uses a SAFe (Scale Agile Framework) approach to leverage DevOps, and the CCC not only supports application teams in business lines but it also allows for strong cloud governance at TSE.

For Evert de Vos, the then Chief Enterprise Architect at TSE, the business value of the cloud strategy is clear:

> "The biggest driver by far for moving to the cloud is that we are able to apply Advanced Analytics on top of our applications. Innovations in this area gives a big EBITA uplift. We have identified a large amount of use cases, all using data stored in a data lake. As a result, we are already improving operations performance."

## Accelerate your cloud adoption with a Cloud Centre of Excellence

Successfully managing a cloud platform on a large scale requires ownership in the organization. Because cloud environments are managed on a large scale, there is a need for a solid architecture around structuring subscriptions, networking, databases, applications, security, and compliance regulations.

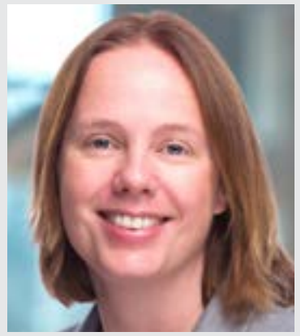
It requires a centralized entity to maintain best practices, onboard customers, and make sure that all services are secure and compliant by default. Implementing these technical aspects on a large scale and embedding them into the organization involves a significant organizational and cultural change. This is where Capgemini's Cloud Center of Excellence (CCoE) comes in.

The primary goal of the CCoE team is to accelerate cloud adoption through Cloud Native[4] and hybrid solutions. A successful CCoE model provides focus on freedom and delegated responsibility. The CCoE equips organizations with a set of guidelines and established and repeatable controls, used by the business. If you want to embark on a CCoE journey, it's important that you set up your team with firm objectives in mind:

- Build a modern IT organization by capturing and implementing business requirements using Agile approaches
- Build reusable deployment packages that fully align with security, compliance, and service management policies
- Maintain a functional cloud platform in alignment with operational procedures
- Review and approve the use of Cloud Native tools
- Over time, standardize and automate commonly needed platform components and solutions

## The future of application development is Cloud Native

The question is no longer "if" you should consider Cloud Native, it's about "when". Every organization needs to embrace the cloud as the future of application development, and it will be those who pave the way that reap the benefits for years to come, while the stragglers are left playing catch-up.



Sjoukje Zaal

Chief Technology Officer – Microsoft, at Capgemini

LinkedIn

### About the author

Sjoukje Zaal is a Chief Technology Officer – Microsoft, at Capgemini, Microsoft Regional Director and Microsoft Azure MVP with over 20 years of experience providing architecture, development, consultancy, and design expertise.

sjoukje.zaal@capgemini.com

### Footnotes

[1] https://www.capgemini.com/2020/09/thinking-cloud-native/

[2] https://docs.microsoft.com/en-us/devops/deliver/what-is-infrastructure-as-code

[3] https://www.capgemini.com/2021/02/microsoft-azure-a-cloud-native-success-story/

[4] https://www.capgemini.com/service/develop-cloud-native-applications/

### Further Reading:

https://www.capgemini.com/us-en/resources/report-embracing-cloud-native-transformation/

https://www.capgemini.com/us-en/wp-content/uploads/sites/4/2020/05/Aerospace-Client.pdf

https://www.capgemini.com/news/capgemini-delivers-logistics-labs-program-to-drive-innovation-at-postnord-using-ibm-cloud/

https://www.capgemini.com/nl-nl/wp-content/uploads/sites/7/2020/09/1B-032.20-Reference-book-Microsoft-Ignite.pdf

https://www2.everestgrp.com/reportaction/EGR-2021-32-R-4202/Marketing

# HOW OPEN SOURCE AND CLOUD NATIVE TECHNOLOGIES ARE RE-INVENTING THE AGILE ENTERPRISE.

True organizational agility has long been every business' goal. Collaboration, Open Source and rigorous software engineering practices provide this in spades.

**IN A NUTSHELL**

- Competition drives ever increasing rates of change – successful organizations optimise adaptability. Open source, Cloud Native, and DevOps enable agility.
- Open source is the world's biggest, most accessible innovation engine .
- A world of choice, with structural skill shortages, means that in-house skills, combined with strategic partnerships is the only way to keep up the pace.
- Integrate DevOps processes with Cloud Native patterns to deliver value to production continuously.

The disruption of almost every sector over the last 10-15 years shows the Darwinian environment that many businesses operate in. Increasingly, one key factor predicts success: the ability to change, and react to change.

In this article, we describe how the collaborative innovation model used in open source, combined with the flexibility offered by the cloud and Cloud Native projects and ways of working that are geared towards delivering end user value, will provide adopters with unparalleled agility and compelling cost efficiencies.

## Open source + Cloud Native = accessible innovation

Open source, and the ethos of sharing innovation for others to build on, has gained massive impetus brought about by contributions from Cloud Native businesses like Google, Amazon, Netflix, Facebook and many other web giants.

These cloud-based organisations (and their supporting technologies) were forged in the fire of previously unimagined scale requirements, including, but not limited to, 365/24/7 availability expectations. In addition, there is an unspoken but very real race to get new features to clients first.

Cloud Native technologies put the power back in the engineers' hands, enabling them to make impactful changes at speed with minimal effort, helping businesses to keep up with changing demands by responding quickly and efficiently. Open source is the key to making this accessible for everyone.

The Cloud Native Computing Foundation [1] offers a list of graduated vendor-neutral open source projects and a trail map every organization can use to implement an Agile Cloud Native platform needed to build, run, manage and observe Cloud Native applications running in a distributed Hybrid Cloud environment. Kubernetes was the first CNCF project and is now the leading container platform in the world. Serverless and Edge computing are examples of new innovations that will become increasingly important in this space over the coming years.

## Open source provides the innovation power of the community

Traditional closed source software vendors use license sales to fund research and development, with the IP created in the R&D phase as the re-saleable asset. The innovation team is as big as the R&D department of the software provider.

Enterprise open source uses donated software and community collaboration to create IP. This means that the innovation team is as big as the community that finds the subject interesting, and willing to donate their time to it. The benefit compared to a closed source vendor is a bigger innovation team delivering more innovations at a faster pace.

This substantially different IP creation model translates to a support-subscription based business model, which are typically cheaper, work on an on-demand basis, and can be cancelled at short notice.

## Closed source versus open source

The procurement implications for licensed software typically involve a lengthy, costly procurement process that needs to be conducted prior to purchase. Once the purchase is made, the cost of switching to any other alternative is prohibitively expensive. This in turn has governance implications that oblige broad adoption of the purchased technology, to the exclusion of other, competing technologies for a number of years. Governance bodies police the software licensing to ensure compliance, but this has a negative impact on your agility, which is stifled in the process.

The procurement process for open source normally follows a free evaluation of the technology with some premium features disabled. If the technology is deemed to be interesting enough, subscriptions acquired on demand provide access to premium features. There is no big bang event, and subscriptions can easily grow with adoption, which in turn has governance implications. Organizations evaluate, elect and discard technologies that suit them, as and when they need them.

In this case, governance bodies act as advisors that evaluate, advise and enable – ultimately helping businesses to foster greater agility.

## Cloud Native and automation = substantial cost savings

The distribution of Cloud Native solutions to provide scalability and reliability has led to a huge advancement in various automation technologies. Spurred on by the need to manage the complexities that are inherent in distributing the solution components, this boom in automation technology often means a substantial reduction in manual effort and subsequently large operational cost savings for businesses.

Well architected distributed solutions emphasize separation of concerns – which brings about two main benefits for businesses:

1. The ability to innovate on individual components – where it will bring the biggest benefit to the business
2. Much needed cost transparency

## Cloud Native provides the next phase of DevOps

In addition to going hand-in-hand with automation, Cloud Native platforms typically provide functionality that simplifies complicated tasks or processes, embodies best practice, and is ideally scalable across the organization.

The different stages in the DevOps lifecycle are often supported by a number of platforms – from a requirements specification platform like Jira, to a CI platform like Gitlab, to a deployment platform like Kubernetes, or a monitoring platform like Prometheus, Elastic or Splunk.

Organizations that are mature in their DevOps journey can integrate these platforms and optimize processes across platforms, which in turn will streamline their DevOps practices. There is a crop of enabling tools/ platforms developing to facilitate this – like Backstage from Spotify.

## The skills gap

Cloud solutions have become the backbone of digital transformation. Every business worth mentioning has a cloud strategy and ambition to become a "cloud-first" organization. And with the rise of cloud there has been an explosion of new technologies, engagement and cost models, and ways of working.

The inherent challenges of changing big, complex systems while they are up and running means it can be hard to find skilled people that can scope, define and implement the changes required. From experience, I believe that sourcing strategies should keep critical skills in house, and make sure that these people are well trained, and supported by communities, partners, collaborators and service providers where necessary.

## Outsourcing to the cloud brings value, but runs risks

The competition to provide compelling solutions and services securely and reliably, combined with a need for cost excellence, poses serious challenges to organizational decision makers.

The public cloud, established by cloud providers like AWS, Azure and GCP, have made concepts like Software-as-a-Service, Infrastructure-as-a-Service, on-demand scalability, and consumption-based pricing pervasive, as well as being generally understood by consumers.

Outsourcing software services and infrastructure management may have positive agility, cost, reliability and scalability effects, but these inevitably come at the cost of agency and autonomy to the service consumer.

The recent pandemic has clearly shown that robustness, and vendor dependency management are as important as cost excellence. In an increasingly distributed and connected landscape the ability to manage and mitigate risks when a provider can or will not perform to expectations is a critical skill. This skill manifests in hybrid, multi, and hybrid-multi cloud strategies for enterprises.

The competition to provide compelling solutions and services securely and reliably, combined with a need for cost excellence, poses serious challenges to organizational decision makers.

Thus, decision makers need to balance speed, cost and reliability, and be sure that they have redundant service providers and viable service migration strategies for crucial processes.

## What this all means

A word of warning: implementing these technologies successfully requires rare skills that necessitates collaboration with partners. But for those organizations that take the plunge and leverage the immense innovation engine provided by open source, cloud and native technologies, you won't be let down by the benefits it can bring – least of all new features and cost benefits. It's what will make you more agile, more competitive and, ultimately, more future-ready.

### Footnote

[1] https://www.cncf.io/

Jade Eloff

Senior Architect

LinkedIn

### About the author

Jade Eloff is a senior Architect who helps our clients modernize their solution landscapes, and piloting their journeys to open source, Cloud Native technologies, and DevOps practices.

jade.eloff@capgemini.com

René van Leusen

Solution Architect

LinkedIn

### About the author

René van Leusen is Solution Architect who helps our clients in translating business requirements in application and integration architectures with a focus on microservices, event driven, APIs, Cloud Native and open source.

rene.van.leusen@capgemini.com

# DOMAIN DRIVEN DESIGN IS THE KEY TO ADOPTING DIGITAL TRANSFORMATION

For businesses looking to accelerate digital transformation, adopting domain driven design is a crucial step to unlocking new capabilities and services.

**IN A NUTSHELL**

- Domain Driven Design is key aspect in achieving digital transformation.
- With a heavy business focus, DDD marries business and IT together to deliver on business objectives better than ever before.
- Benefits of this approach include faster time to market, reduction in cost, better sprint success rates, higher customer satisfaction and increased profits.

Businesses today are finding themselves operating in an increasingly dynamic world, with an increasing number of external variables that demand them to become more adaptive, more agile and more responsive than ever. One truth remains constant, and that is the requirement to transform their existing business model if they stand any hope of remaining competitive. The pressure is on. In order to stay ahead, organizations need to have innovative processes ready to deploy system changes straight to the market at speed.

Take the restaurant industry as an example[1]. Particularly impacted by the current pandemic, these businesses, especially fast-food chains, are facing increasing requirements to develop the agility needed to operate in these uncertain times, whilst still continuing to meet evolving customer expectations. The leading global restaurant chains are compelled to innovate and adopt digital platforms to achieve future business growth. Digital Wallet, automated inventory management, online ordering, automated kitchens, mobile and digital order placement & loyalty programs are some of the key initiatives which can be built faster if we have clear adoption of the right domains and digital technologies.

This is creating a shift in approach, which is seeing a growing number of businesses adopting digital technologies like microservices & cloud, enabling them to achieve growth and stay ahead of their competitors. In the journey of "Digital Transformation", large enterprises with legacy systems are finding themselves under pressure to deliver new, differentiating digital services to customers that are fast paced, scalable and available.

So what's the solution? As leaders in Digital Transformation, our suggestion is to redesign your applications with microservices architecture using principles of Domain Driven Design. Then, combine this with other digital technologies to dramatically speed up time to market and help your enterprise solve the legacy systems' challenges in an evolving digital economy.

As leaders in Digital Transformation, our suggestion is to redesign your applications with microservices architecture using principles of Domain Driven Design.

## What is Domain Driven Design?

Domain Driven Design bridges the two key aspects of digital transformation: business and technology. The domain concept has a clear, relevant role from a business point-of-view just as much as from a technical implementation standpoint.

At its core, Domain Driven Design is a structured approach to digital application development that centers on programming a domain model with deep business understanding of the processes and rules of said domain. It does this by focusing on three core principles:

1. Focus on the **core domain** and things that are valuable to the business.
2. Explore models in a **creative collaboration** of domain practitioners and software practitioners.
3. Define **bounded context** with clear separation of concern between domains.

Put simply, Domain Driven Design has more of a business focus, ensuring the application reflects the real-world requirements of your domain.

Domain Driven Design has some undeniable benefits when we think of application development and, more widely, Digital Transformation as a whole:

1. The reduction in release cycles to production generates faster time to market.
2. Higher delivery velocity throughput in each sprint leads to higher sprint success rates.
3. Improved platform governance, stability, scalability means optimized cost.
4. Business prioritized services enable higher customer satisfaction and better profits.

## Approach

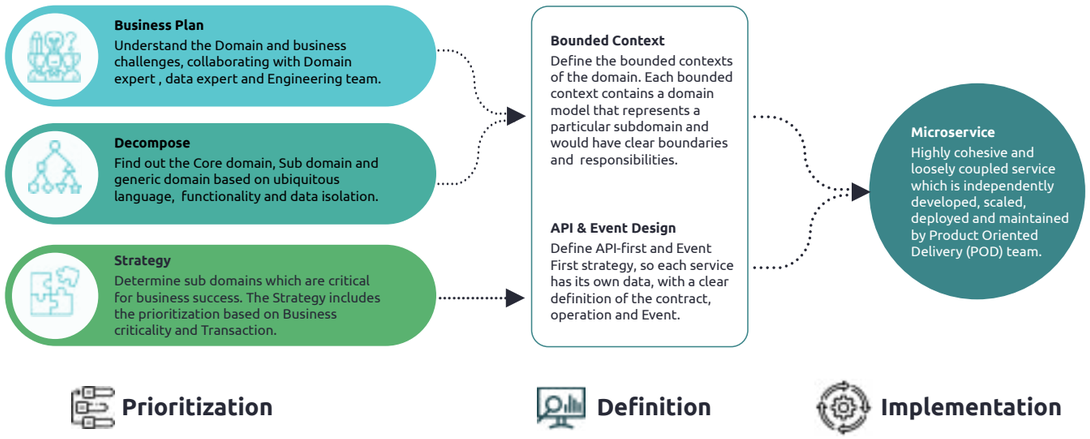
It would be a stretch to say that this is a simple process. In order to succeed with Domain Driven Design, it's important to use a proven approach with clearly articulated phases that ensure end to end coverage right from design to implementation.

This approach relies heavily on successful collaboration between the business domain expert and development team to ensure the product aligns with key business priorities, ultimately helping the business to grow and overcome the challenges in today's market.

The figure below provides a description of our suggested methodology, which is practiced widely for developing distributed system designs when modeling their IT systems.

As you can see, there is a certain methodical fashion that ensures the domain experts and application teams collaborate to develop the To-Be business model. In the future of application development, there will be a lot more emphasis on the merging of business and IT, so getting these two departments to work together efficiently is a crucial step.

**Figure 1:** Key phases to achieve Domain Driven Design.




Along the bottom of the diagram you'll notice that there are three key steps to this particular approach, which we'll dive into below.

## Prioritization

This is an iterative process that is a fundamental step in defining key business domains and understanding business processes. Fundamentally, this phase involves three key activities:

1. **Business function:** The teams need to identify the key business functions of the domain. This can be understood by bringing together the experts of business, data, and engineering teams.
2. **Decompose:** Next, you need to define a hierarchical structure that allows the problem domain to decompose into different areas of the business and sub-domain.
3. **Strategy:** The strategic focus makes sure that your major efforts go towards what's most important to the business right now. This strategic emphasis is crucial in countering the "everything is critical" mindset that has a nasty habit of impeding software development efforts.

To bring this to life, let's go back to the example of fast-food restaurants. Their primary business focus is: Customer, Menu, Offer, and Order. These are their prioritized business services, which not only create a positive customer experience but are also capable of offering more features, like enabling contactless dining and customer rewards.

In addition to the three steps, the prioritization phase has two possible strategies:

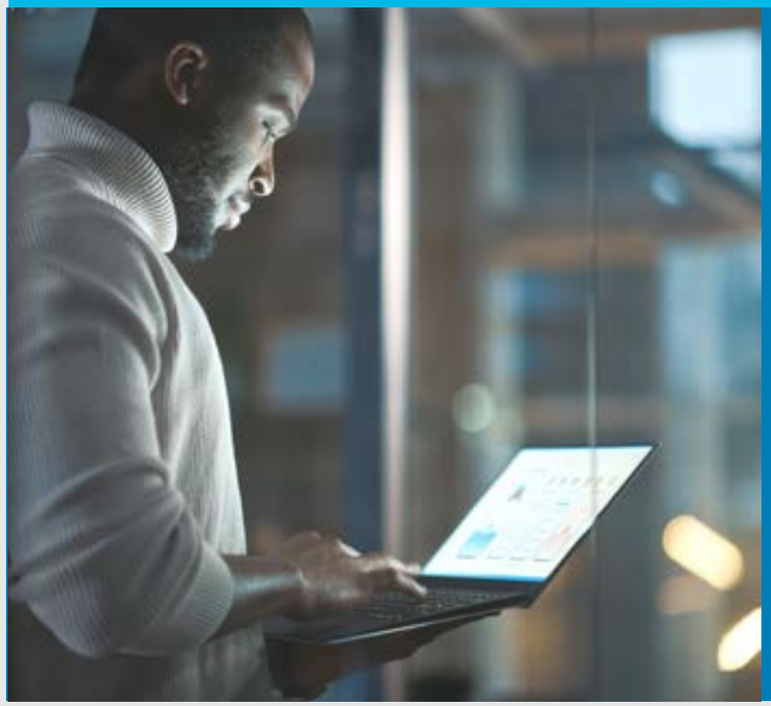
1. Top-down strategy.
2. Bottom-up strategy.

Which one you pick would depend on your domain and existing landscape.

The top-down strategy is typically recommended for greenfield application development, or if you're looking to completely rewrite new legacy systems in modern Cloud Native digital technologies.

The bottom-up strategy is usually practiced for evolutionary reconstruction and is best suited for legacy system business models or brown field development.

Before getting to work on modernizing your legacy system, a few iterations of the prioritization phase will help to uncover essential information needed to create a vision for your target architecture.



## Definition

In this phase, the domain expert, data expert, and engineering team work together to define the bounded contexts of the domain. Each bounded context contains a domain model that represents a particular subdomain of the larger application. As we mentioned previously, a fast-food chain's subdomains based on business priority are: Menu, Order, Customer, Offer, Kitchen and Payment.

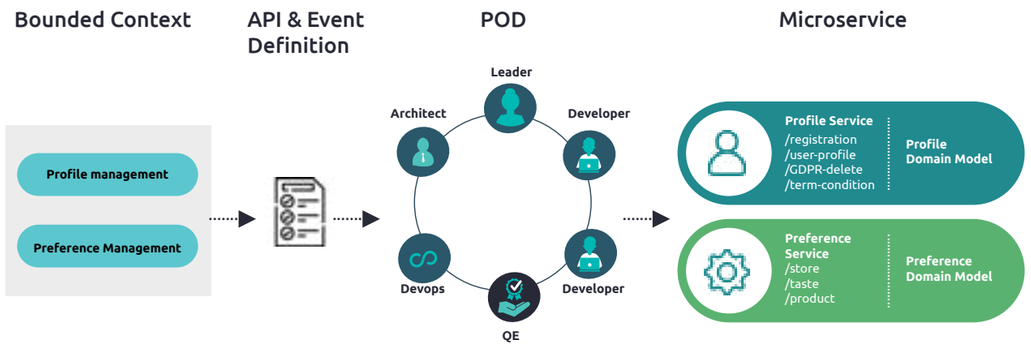
Once the bounded context has been defined, the team will follow the API-first and event-first strategy where they define API and a set of events it publishes and consumes so each service has its own data, with a clear definition of the methods which implement the business logic independently on that data.

## Implementation

In this phase, tactical design of Domain Driven Design has been followed. Each logical group of function or bounded context is assigned to a team called a Product Oriented Delivery team (POD) [2], who are fully responsible for developing business capabilities tied to the bounded context.

These business services have pre-defined boundaries, and each POD team can work independently on one or more microservices. Each microservice can have different databases, different architecture, programming language, and can be deployed independently, as mentioned in our previous article on "Lego-Fying" your digital platform. This way the business services are highly cohesive and independently scalable. This not only provides a clear focus on how to solve the complex problems first, but also helps to deliver the application faster while lowering the iteration cycle for any change.

**Figure 2:** Product Oriented Delivery(POD) Teams.




## What does all of this mean for you?

The biggest reason to practice a Domain Driven Design model-based approach is to facilitate the transition to an API-driven, microservices-based architecture, which will give you an extremely scalable, multi-channel platform with far greater development and integration flexibility. It will also help you grow your business in the long term by supporting the rapid adoption of new offerings that can be launched on the platform in a much shorter time.

Capgemini works with multiple domains to enable digital transformation across the globe. By leveraging accelerators like Digital Cloud Platform[3], our library of pre-built software components, we are typically seeing businesses accelerate their development by up to 50%, drawing on a proven methodology for helping IT organizations embrace POD models for exceptional agility.

In an environment of unprecedented uncertainty, one thing remains clear. In order to not only survive, but thrive, IT needs to align with business.

Domain Driven Design helps us to foster a close working relationship with business stakeholders, gain a comprehensive understanding of their needs, and effectively deconstruct complex business functions to deliver continuous, sustainable value.

Rahul Khandelwal

Enterprise Cloud Architect

LinkedIn

### About the author

Rahul Khandelwal is an Enterprise Cloud Architect with expertise in leading Business Transformation and Application Modernization journeys using cloud native technologies

rahul.khandelwal@capgemini.com

Anand Gupta

Senior Cloud Architect

LinkedIn

### About the author

Anand Gupta is a Senior Cloud Architect with expertise in leading Digital Transformation projects for many large clients by leveraging Microservices, DevOps and serverless.

anand.gupta@capgemini.com

### Footnotes

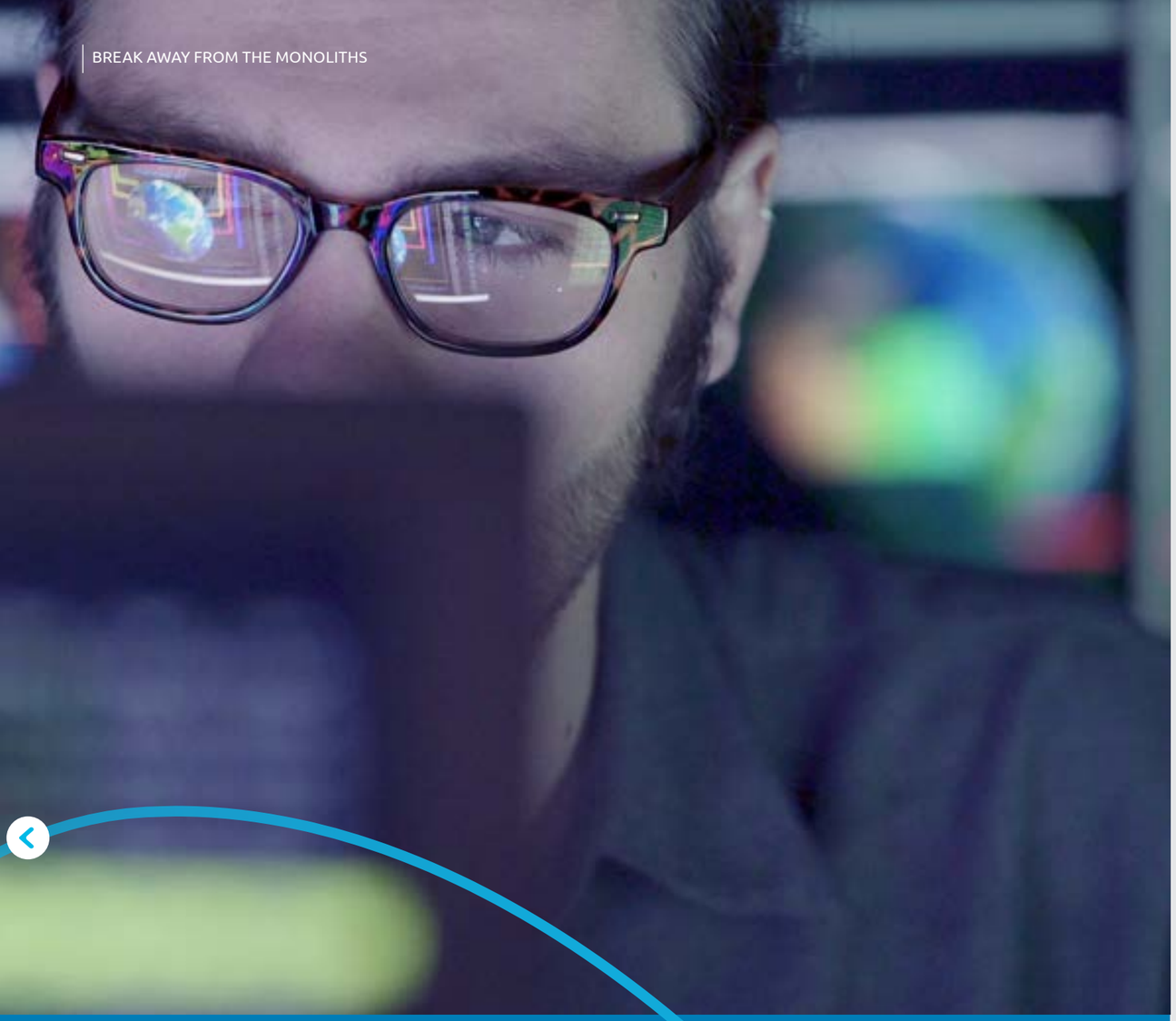
1] https://www.capgemini.com/us-en/2020/09/transform-the-quick-service-restaurant-experience/

[2] https://www.capgemini.com/us-en/2020/11/product-oriented-delivery-enabled-by-pod-models/

[3] https://www.capgemini.com/us-en/service/cloud-services-2/digital-cloud-platform/

### Further Reading:

Approach to Software Development DDD-
https://martinfowler.com/bliki/DomainDrivenDesign.html
Practical learnings on microservices-
https://www.capgemini.com/2019/06/practical-learnings-on-microservices/Marketing



# BUILDING PROGRESSIVE WEB APPLICATIONS

Capable of being deployed at speed, with greater security, increased responsiveness and lower cost, PWAs will play a big role in the future of application development.

**IN A NUTSHELL**

- With an increasing number of customers favoring mobile commerce, organizations need to provide more engaging, mobile-friendly web applications.
- Progressive Web Applications offer seamless, cost-efficient solutions for this growing demand.
- The time to act is now .

**THE BASELINE PWA CHECKLIST:**

- Site is served over HTTPS.
- Responsive pages on mobile and tablet.
- All URLs load while offline.
- Metadata provided for “Add to Home” screen.
- First load is fast even on 3G.
- Site works across browsers.
- Page transitions load quickly even on slow networks.
- Each page has a unique URL.

Many businesses deliver front-end experiences through eCommerce platforms or content management systems (CMS). With the rapid proliferation of mobile commerce, however, customers expect the speed, functionality, and convenience of a mobile application throughout their shopping journey.

How, then, can organizations address these needs quickly and cost-effectively? Enter Progressive Web Applications (PWA). In this article, I will not only provide a quick overview of the underlying technology that makes PWAs so engaging and useful, but I will also lay out a step-by-step roadmap that will enable readers to prepare their organizations to easily launch a successful PWA experience.

## What is a PWA?

PWAs are applications based on web browsers[1]: they run in the browser itself and do not require installation like native applications do. This makes them quick to build and deploy, safer than native mobile apps, and functional on all kinds of operating systems and browsers.

One PWA that you may be familiar with is the ordering system built by Starbucks on the web. While it delivers a similar user experience to their native app, like browsing the menu, customizing orders and adding them to your cart, it’s actually 99.84% smaller than the iOS app, and all of these actions can be done offline. Then, once you’re connected to a network, you can place your order. By launching this new PWA, Starbucks have managed to double the number of daily web users, reportedly receiving an equal amount of desktop and mobile orders every day.

At a high level, PWAs need to be **reliable, fast, and engaging:**

**Reliable** –PWAs instantly load and are never down, even in uncertain network conditions.

**Fast** – PWAs quickly respond to user interactions with smooth animations and no delayed scrolling.

**Engaging** – PWAs feel like a natural app on the device – mobile, tablet, or desktop – and have an immersive user experience.

## PWAs today

Since being introduced by Google in 2015, PWA adoption continues to grow exponentially. Throughout the industry, PWAs are being looked at as the future of digital experience, and therefore play a large role in the future of application development.

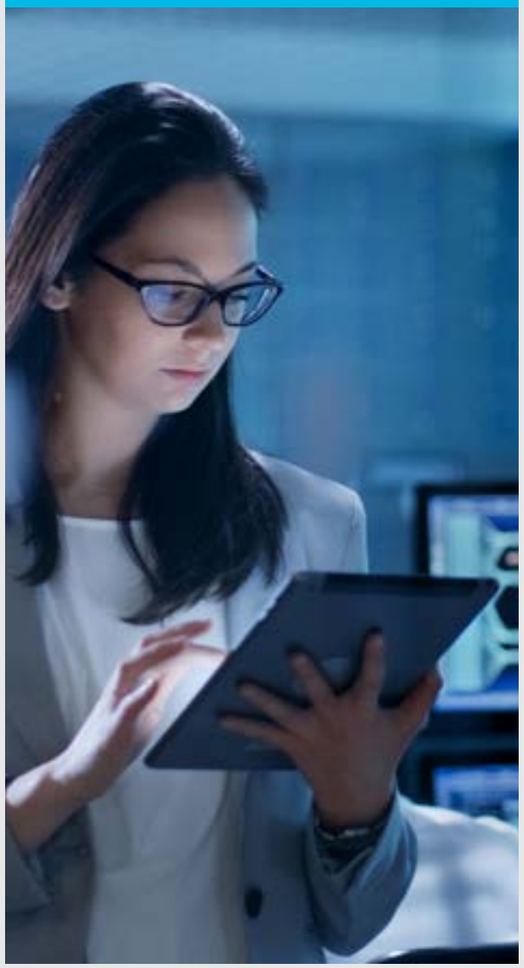
## So, why are PWAs so popular?

These fast, app-like experiences are built on HTTPS and modern web technologies, boasting enhanced site performance, search visibility, and security - all massive differentiators. The enhanced page load speed alone is shown to improve search rankings and enhance customer engagement. Add in the other benefits (no updates, seamless offline operation, lower cost, quick installation, increased conversion), and it becomes obvious that PWAs greatly enhance the customer experience.

This is not to downplay the capabilities of native apps – the kind you download from the App Store or Google Play – which provide unmatched opportunities for customization and engagement. They do, however, require storage space on a user’s device and are often extremely expensive and time-consuming to build.

Furthermore, mobile web traffic is by far outpacing the growth of native applications, meaning consumers are increasingly likely to utilize a mobile browser than hunt for a specific application to download. Because of their ability to provide an enhanced customer experience without requiring users to add, update, and store applications locally on their devices, in 2019 Gartner predicted that PWAs would represent 50% of web applications by the end of 2020[2].

> "With the advent of PWAs, web apps have become just as capable as native, supporting features such as push notifications and Add to Homescreen, without giving up the reach of the web."
>
> - Google



## Strategically Launching a PWA

The value of PWAs is obvious, and the concept is not entirely new to the market. But from what we've seen, the challenge lies in understanding how to build, implement, and optimize a PWA to continually deliver a differentiated, more seamless and engaging customer experience.

In order to create a PWA that not only delights customers but is also cost-efficient, organizations need to follow a strategic roadmap. From initial ideation through final iterations and optimizations, following a set path and adapting along the way will position your organization to quickly release a winning product, improve it based on customer feedback, and stay ahead of the competition.

## The PWA Roadmap

### Step 1: Define the end goal

As with any digital transformation objective, every move you make needs to be goal oriented. Building a PWA just to build a PWA is not a strategic way to go about creating a successful digital experience.

In this brainstorming stage, make sure to pull in resources, competencies and skillsets from across the organization. Successful PWAs require data and input from just about every aspect of the business. Technical teams, marketing teams, and sales teams will all have different ideas and visions for what they believe the user experience needs to look like. Only when all the cards are on the table can you effectively prioritize needs and address what matters most first.

**Key Questions to Help Define Your End Goal**

- What are the desired business outcomes?
- What are the functionalities your users are looking for?
- How much do you want your PWA to feel like a native app?
- How important is installability and discoverability?
- How would offline access benefit users?
- How can you use push notifications to re-engage users?
- What additional features would your users love to access?

### Step 2: Address technical debt

We've already touched on the value inherent to a PWA, but this value can only be realized if the current technical ecosystem is able to effectively perform. Whether you are planning on leveraging a commerce platform, CMS, or a headless, API-based solution, the underlying technology stack will determine whether your PWA can perform in a way that is beneficial to your visitors.

Trying to launch a PWA on outdated technology will not only result in a poor product that deters customers, but it will also move you even further behind your competition. Addressing existing technical debt up front is mandatory for organizations looking to build a new, successful PWA.

### Step 3: Quickly launch a Minimal Viable Product (MVP)

The beauty of PWAs is their speed and simplicity. So, if your technology is able to support the needs of a modern PWA experience, it is time to get an MVP launched.

A word of warning. It can be easy to get so bogged down in the granularity of the front-end options inherent to PWAs – navigation, color schemes, push notifications, etc. – that you fall behind deadlines, and the competition, by trying to get every pixel perfect before launch.

Instead, rolling out a functional PWA as an MVP on an expedited timeline is the way to go. It gets your PWA into the market, drives engagement, and also gives you the opportunity to continually wow visitors with each new optimization.

**What you need to launch a PWA –**
(According to Jason Grigsby in his book, "Progressive Web Apps")[3]:

- Secure site using HTTPS
- Build service workers for performance improvements

  Write scripts (service workers) that allow for data to be pushed from the server to the app when the app is not active.

  Develop a baseline caching strategy to cache all previously viewed pages. This reduces server calls and enables users to browse and load pages seamlessly.

- Build an offline fallback page using file manifest to support offline traffic

  Create a short manifest file with a display mode defined as well as some simple icons. These will enable the PWA to respond quickly even though users are accessing the experience offline.

### Step 4: Add front-end functionality

Once the underlying nuts and bolts are in place, it's time to dress up the front-end experience. A lot of this stage in the roadmap will revolve around caching – or storing commonly used files outside the host server to increase page performance.

Caching recently viewed pages as well as pre-caching homepage assets, popular pages, and other critical content will speed up the user experience and make it more engaging.

At this stage, organizations also need to look at unique PWA functionality such as notifications and iconography. Defining the content, promotions, and messaging along the customer journey is critical to a sound notification strategy. Here are some options:

- Sale on popular items
- Abandoned cart
- Abandoned session
- Relevant follow-up content to site behavior
- Order confirmation
- Shipping confirmation

### Step 5: Plan for the future

The value of launching a functional, but not full-furnished, PWA is that it allows you to be Agile in future iterations of the experience. Customer demands and expectations are changing faster than most organizations can react, and this approach prevents you from being tied down to a complex PWA that misses the mark on future trends.

As you further understand your customers, market, and their behaviors – especially on your PWA – you can use that data to prioritize future optimizations.

EXAMPLES OF FUTURE OPTIMIZATIONS:

- Payment options.
- Integrations.
- Notification strategy.
- New tech (VR/AR).

## It’s now or never

The benefits of PWAs are clear: enhanced performance, increased security, and engaging content boost engagement and conversion in profound ways. So, when is the best time to build a PWA of your own?

**Now.**

Advancing your digital framework to be on the cutting-edge of customer demands can take a very long time. The tricky part is that consumers want high-end experiences now. Improving your digital experience with a high-performance PWA can not only address these market demands, but also drive revenue and success for your business.

The strategic roadmap approach to PWA development outlined in this article enables you to begin your journey towards an enhanced customer experience, while staying ahead of the curve in a rapidly evolving market.

Rich Minns

Enterprise Architect Director

LinkedIn

### About the author

Rich Minns is the Digital Customer Sector Leader for CPRD, with a focus on helping clients build better connections with their customers through Capgemini’s Digital Customer Experience Initiative.

rich.minns@capgemini.com

### Footnotes

1] https://techwireasia.com/2019/04/why-demand-for-progressive-web-apps-is-growing-rapidly/

[2] https://www.cio.com/article/3395157/make-the-move-from-mobile-apps-to-next-generation-progressive-web-applications.html

[3] Jason Grigsby, Progressive Web Apps



02

# THE PROMISE OF SAP RENEWABLE ENTERPRISE

1. Are you really getting the most out of your SAP system?
2. Mendix and SAP: Choose or combine?
3. SAP And DevOps: Hot? Or not?
4. It’s time to unlock the value of sustainability

# ARE YOU REALLY GETTING THE MOST OUT OF YOUR SAP SYSTEM?

As intelligence-based applications and web services surge in popularity, is it fair to expect the same level of functionality from our SAP systems?

**IN A NUTSHELL**

- An increasing number of businesses are making the decision to get more out of their SAP systems by harnessing more advanced intelligence-based technologies.
- This move to the "Renewable Enterprise" means not only looking at your technology and IT, but at your business as a whole.
- By harnessing cloud technologies and SAP products like BTP, companies can use their SAP landscape to become more scalable, sustainable, secure, compliant and cost efficient.

Increasing exposure to applications and web-based services with built-in intelligence and intuitive user experiences (think of navigation or translation apps and websites) means that business users have grown to expect more from technologies for personal use.

With this in mind, it's no wonder that organizations are starting to expect more from their SAP software. How can you rationalize the fact that your phone has a bunch of apps that combine augmented reality, machine learning and more – while your SAP suite of applications work with a number of transactional applications that mostly read and write to a database?

If this is something that's crossed your mind, you're not alone. In fact, it's one of the most common reasons for businesses making the decision to get more intelligence and usability out of their SAP landscape. And if you think about it, why wouldn't you want to "app-ify" your SAP experience?

## The rise of the Renewable Enterprise

The answer just below the surface is as simple as it is true: it's not easy to make this move and neither is it a one-time change. It's a transformation that cannot be achieved by simply assessing your existing technology and IT. Rather, you need to be looking at your business as a whole to start growing as a Renewable Enterprise.

Becoming a Renewable Enterprise that is agile, data driven, innovative and connected requires both a technical and an organizational component.

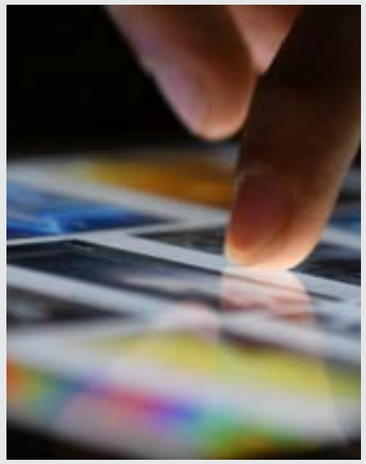
While we share a strong focus on the technical aspect within this "Applications Unleashed" report, articles such as "SAP and DevOps: Hot? Or Not?" look closely at how to build a more rapid and Agile mindset into your organization.

## Why you should consider moving from an ERP-central to a Cloud-central world

For a long time, many SAP clients have been using their SAP system as the center of their Financial and Operational IT, due in no small part to its integrated capabilities. Making SAP the heart of the operation means integrating it with any other required "satellite" systems using either standardized approaches such as Electronic Data Interchange (EDI) or making new Advanced Business Application Programming (ABAP) custom code to handle incoming and outgoing data.

But the benefits of a cloud-centric SAP environment, like better options for sustainability, more cost-effective test environments, end-to-end process and data integration capabilities, availability of innovative new technologies such as Blockchain, and additional security and compliance, are becoming hard to ignore.

In the past, the most common way to create new apps was to use ABAP to make any custom apps that acted on your SAP data and processes. While almost infinitely flexible, over time every SAP client has found themselves dealing with a mass of custom ABAP code that had been developed over the years by different system



Only sustainable as long as there are enough specialists to assess the current custom coding and then create new solutions despite the existing complexity.

integrators that each followed the SAP programming paradigm of the age – evolving from Dynpro's, BSP pages, Web Dynpro's, ABAP Objects and a vast array of different steps in between.

SAP's iron-clad effort in making any solution backwards-compatible made sure that any code written in an older ABAP paradigm would run as well as code created with the newest. This is exactly why it was always an attractive proposition to just keep making custom ABAP solutions to exactly fit new requirements. In the meantime, it was getting so complex and inflexible to maintain current code that for any new requirement, a thorough assessment had to be done on the impact it would have on all the other custom solutions.

This way of working is only sustainable as long as there are enough specialists to assess the current custom coding and then create new solutions despite the existing complexity. While manageable with the right expertise, costs of maintenance can only increase over time. And in today's market specialists are in short supply.

With this in mind, for many SAP ECC clients making the move to S/4HANA, the migration could present an opportunity to re-assess, start over and do things differently.

## Making your extensions on SAP BTP

One of the big use cases for the SAP Business Technology Platform (BTP) is for it to become the place where your custom "bespoke" applications would run, giving you the possibility of easily retrieving and connecting the apps with your SAP backend, other cloud applications and the available intelligent technologies. SAP BTP also imbeds intelligent technology like AI and Machine Learning to enhance the overall user experience and deliver additional customer-centric value to businesses.

Of course, you have to start somewhere. The big question to ask yourself is: which functionality do you want to move to your cloud system?

This can feel like a counterintuitive decision. Why would you want to re-design or re-implement a custom ABAP application that is already giving you value? From an IT-perspective, it could be easy to enumerate the benefits – but from a business standpoint, why invest in something you already have?

Often, we see that companies would rather choose to implement wholly new functionality in the SAP BTP to begin with, leaving the existing custom code running as it is.

While this approach makes initial financial sense, it's not without its risks. For example, you could find that your developers are making apps using the best technology possible available, while older, bespoke applications would have been developed in ABAP (kept up to date between SAP releases). Meanwhile, newer functionality might be made first in Fiori on SAP BTP NEO and later using something like SAP CAP, Fiori Elements or Mendix on Cloud Foundry. And future apps would be made on whatever future system you would choose.

As time passes, this will lead to a complex infrastructure of apps with different codebases and user-interfaces. And this poses a problem. On the one hand, maintenance will become increasingly costly and difficult since you need to find people with different technical skill sets to build and maintain the apps, and on the other hand your business users might be frustrated as the "look and feel" of each app is different, depending on the underlying technology.

## Investing in your future

Coming from a technology background, it's easy for me to only focus on the obvious technological benefits of new technologies vs old. But if we look at this through a more pragmatic, business-minded lens, the real question is around Total-Cost-of-Ownership.

On an app-per-app basis, this calculation should include the individual investment, as well as the total amount of time spent on re-work for this application. How much does it cost you to keep the functionality running as you would like? Not only due to new changes in the functionality, but also all the rework needed to keep it up to date with smaller and bigger SAP upgrades, like HANA and S/4HANA, as well as the changes needed to keep up with more future-facing requirements.

At the end of the day, the further along you decide to take your existing ABAP code, the more money you will need to invest to keep it working. By this metric alone, you could already make a case to re-implement it as a "loosely coupled" app on your SAP BTP. Yet, to get a true picture of your TCO – you will need to look at both IT and organizational aspects. It's not called the SAP Business Technology Platform for nothing!

While the value of making apps on BTP is undeniable, it's never just about technology. The true power of the platform comes out in a company-wide vision on how to use it to develop apps and bring value in other ways. So, ask yourself this: when it comes to evolving new ways to develop future applications, can you really afford not to put your SAP system to work for you?

Maarten Stam

SAP technology consultant

LinkedIn

### About the author

Maarten Stam is an experienced SAP technology consultant working in Consumer Products, Retail and Logistics.

maarten.stam@capgemini.com

# MENDIX AND SAP: CHOOSE OR COMBINE?

When it comes to bespoke customizations, our advice is to keep the core clean and innovate using low code

**IN A NUTSHELL**

- Keeping the (SAP) core clean is key to enable innovation and adaptability, for which you will need to use additional tools.
- Choosing between Mendix and 'standard' SAP (Fiori) depends on several dimensions, including the use case you want to tackle.
- Mendix has a strong business case for creating innovative and differentiating apps on top of your existing SAP landscape.
- Mendix can bring software development towards the business, enabling a new type of developers (citizen development), overcoming the shortage of tech talent.
- efficient.

With the launch of S/4HANA, SAP coined the principle "keep the core clean[1]". This came as a response to the fact that there are too many bespoke developments in our current ERP systems, which block us from easily transforming our system and getting new technologies and applications into production to help businesses accelerate. Meanwhile, there is still a need for customizations and innovation.

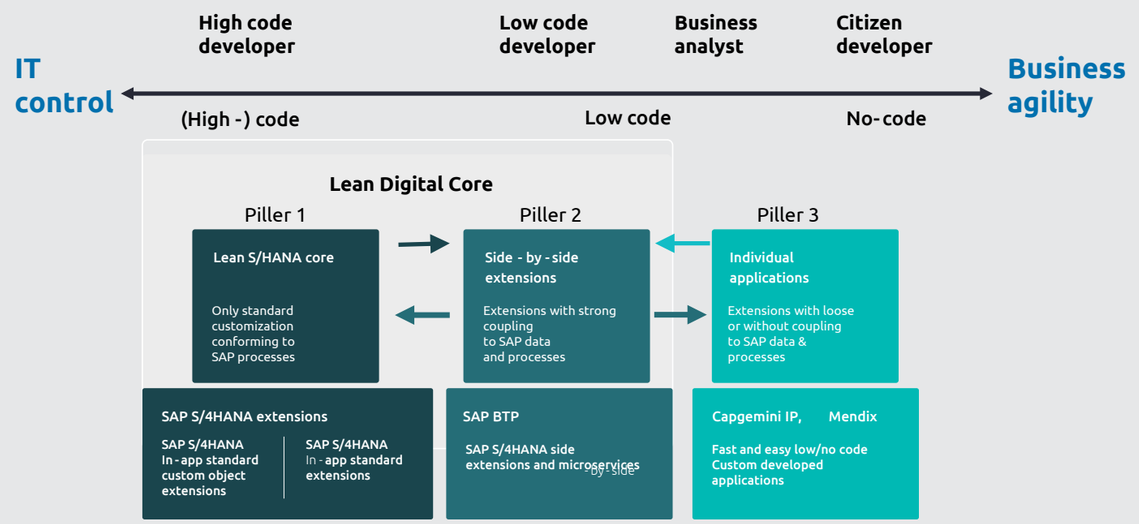
In 2017, the low code platform Mendix became part of the SAP portfolio[2]. Since then, low code has taken a huge leap forward, with growth numbers of over 25% annually[4], a figure that is supported by Gartner, who states that by 2024 65% of application development will be performed using a low code platform[4], while 66% of big companies are already using at least four low code platforms[5]. It is therefore no surprise that SAP is also embracing low code, resulting in a partnership with the citizen development platform Betty Blocks in 2021[6], and the announcement from SAP that it will incorporate the no code platform AppGyver into their portfolio[7].

As a large global SI, Capgemini is often involved in numerous digital transformation projects, responsible for bringing down IT costs while driving innovation. As a result, over the past few years we have received many questions around what to consider when choosing between Mendix – the only low code platform that runs directly on BTP – or 'standard' SAP building blocks such as Fiori and BTP.

In this article, we will share some of our thoughts and approach this question from various angles, taking into account technical abilities and costs, along with some example use cases.

Our multi-pillar solution architecture (Figure 1) helps businesses choose the right level of code to create that lean digital core, so you can remain as agile as possible.

**Figure 1:** Multi-pillar architecture



The first step to choosing a technology is identifying the use case you want to tackle. Think of it this way – you could hammer a nail with a screwdriver, but why would you not use a hammer? In short, pick the technology that suits your needs. Generally, one can state that if standard SAP Fiori fits your use case, that's what you should choose. If, however, there is a need for more complex customizations, for example, you need to add data, process support, or future-proof flexibility to accommodate changing requirements, Mendix becomes the better tool to use. This is because it enables you to easily implement custom logic or workflow at lower costs.

To put things into perspective, we gathered some example use cases and our view on the 'best solution' from a functional perspective. Note that there are other criteria to take into consideration, and the business case for Mendix will be harder when considering a single use case. Instead, pick several use cases and try to find the best match.

## Typical use cases

### Business Partner Management

The viewing and changing of business partner data – data about suppliers and customers – by both internal and external users, uses standard and simple API's. For these uses cases it's most efficient to make a Fiori Element app with standard building blocks and, if needed, some small extensions. This will make the solution easy to maintain, without the costs of additional Mendix licenses.

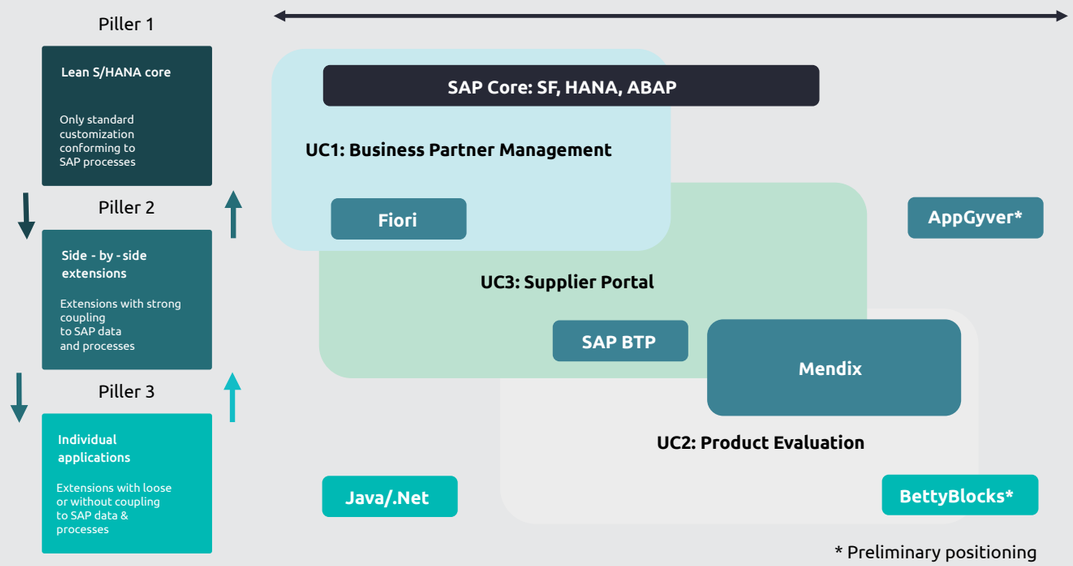
### Product Evaluation

In order to get customer feedback on your products, a short evaluation can be put out. Since there is little to no interaction needed with existing SAP components, this app is easily built with a platform like Mendix, leveraging its capabilities for (native) mobile development.

### Supplier Portal

Imagine a complete Supplier Portal, where suppliers can log on and perform various actions, including managing their contracts, uploading documents, uploading article data and requesting changes to contracts. These are all separate solutions that will likely be available through a single SAP BTP-based portal. However, the single apps are best built with a mix of available solutions – including but not limited to Fiori Elements, Freestyle Fiori and Mendix. Specifically, the 'document management' solution is an example of a 'hybrid' solution. Part of the solution will likely be the standard 'Document Management' service on the SAP BTP, but other requirements can be implemented by both SAP Workflow and Mendix, where Mendix seems to be the better option for frontend handling of documents. It all depends on the specific criteria, not only for your current situation, but also in the immediate and long-term future.

**Figure 2:** Overview of the typical solutions and use cases




* Preliminary positioning

## Fiori vs Mendix – technical capabilities

For the technical aspects, both platforms offer a rich User Experience, including on mobile, and apps can be deployed anywhere. However, Mendix is moving from hybrid to native and PWA, while Fiori is focusing on hybrid browser apps[8].

| | Fiori | Mendix |
|---|---|---|
| UX | Everything is possible within Fiori guidelines | Everything is possible |
| Mobile | Hybrid browser apps only, possibility to use SDK for native (IOS and Android) apps | Offline-first approach with native mobile, moving towards PWA |
| Deployments | Any SAP environment | Anywhere, BTP is preferred for SAP integrations |

## Integrating with existing SAP landscapes

Both Fiori and Mendix prefer to use the OData (Open Data) protocol to integrate with existing SAP landscape, as supported by EEC6+ etc. However, creating the OData services can require a lot of manual work on older SAP environments, while S/4HANA delivers a lot of standard OData services to connect to. If your landscape still runs on an older version, consider upgrading while migrating your SAP customizations to future-proof technologies.

## Cost efficiency

Of course, cost is an important aspect to consider when choosing your tool. While Fiori comes with most SAP standard licenses, Mendix requires an additional license. However, when comparing one to the other, you should not only consider the annual costs, but also the development and maintenance costs. While Mendix comes with additional annual costs (OPEX), it typically lowers project costs (CAPEX). As Mendix licenses become cheaper by size, it is easier to build a business case for a range of use cases, rather than a single use case.

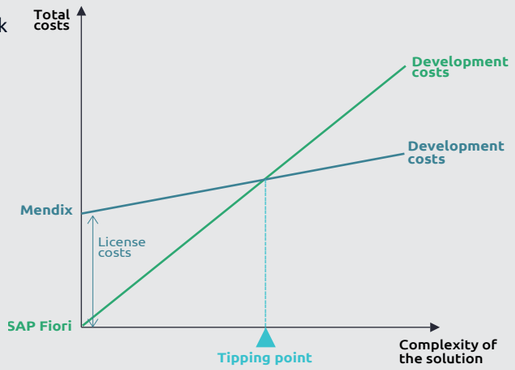



**Figure 3:** Licensing and costs (Note: estimates based on 200 users cf. list price)

| | AP Fiori | | Mendix | |
|---|---|---|---|---|
| **Project costs** | | | | |
| **Development** | Medium to High | 200K | Low to Medium | 100K |
| **Integrations** | Medium to High | 100K | Low to Medium | 75K |
| **Annual costs** | | | | |
| **Licenses** | Low | 0 | Medium to High | 60K |
| **Maintenance and support** | Medium to High | 100K | Low to Medium | 60K |
| **Total costs over 5 years** | | **800K** | | **775K** |

Put simply, Mendix will be profitable when you need lots of new or changed applications. Particularly when developing several applications, the business case is hard to ignore.

## Accessibility for developers

Low code platforms like Mendix address different types of developers.
While Fiori requires deep technical knowledge, Mendix users don't need to have deep technical skills, leaving more room for business knowledge. Therefore, which platform you choose really depends on the type of developers you have available or that you want to attract. As Mendix equips business end users with their citizen development tool, you will have a broader range of possible 'developers' available to create applications, helping to overcome the challenges presented by the tech talent gap[9].
If your business is struggling with tech availability, and you want to embrace a new breed of developers, low code and no code is the way to go.

## The future of application development on SAP

While no-one can know for sure what the future will bring, it is certain that there will be a plethora of low code platforms available to the market over the coming years, and SAP will likely have a part to play[10]. But it's important to note that while the low code market is growing fast, and is able to cover an increasing number of use cases – from simple front-end facing applications to complex back-end systems – low code is not always the best choice.

Fiori is being constantly enhanced, so it stands to reason that there will always be use cases that can be implemented easily against low costs with your standard SAP license. For the more complex use cases however, a business case can be made in favor of low code, where the lower development and maintenance costs easily beat the cost of annual licensing. This is particularly true where more applications are being built with low code, enabling businesses to drive costs down and boost innovation at speed and scale.

In the end, there's no silver bullet or one-size-fits-all solution. It all comes down to your requirements, access to technical skills, and what your existing licenses are capable of. The good news is that with so many options on the market, you're guaranteed to find something that fits your vision for the future.

### Footnotes

[1] https://www.sapinsideronline.com/are-you-ready-for-sap-s-4hana-the-time-is-now/

[2] https://www.mendix.com/press/mendix-announces-global-reseller-agreement-with-sap-providing-enterprises-low-code-development-for-building-web-and-mobile-apps-at-speed-and-scale/

[3] https://www.gartner.com/en/newsroom/press-releases/2021-02-15-gartner-forecasts-worldwide-low-code-development-technologies-market-to-grow-23-percent-in-2021

[4] https://www.statista.com/statistics/1226179/low-code-development-platform-market-revenue-global/

[5] https://www.forbes.com/sites/ilkerkoksal/2020/04/29/the-rise-of-low-code-app-development/?sh=4d8966c26807

[6] https://blog.bettyblocks.com/sap-partneredge-marketplace-embraces-no-code-with-betty-blocks

[7] https://news.sap.com/2021/02/sap-acquire-appgyver/

[8] https://clearbridgemobile.com/mobile-app-development-native-vs-web-vs-hybrid/

[9] https://www.monster.com/career-advice/article/tech-talent-gap-survey-0816

[10] https://www.nolanramsey.comtheradreport/2021/3/19/does-saps-acquisition-of-no-code-developer-appgyver-hurt-their-solex-partner-siemens-mendix


Marien Krouwel

Expert in Agile, Cloud Services, DevOps, Low Code

LinkedIn


### About the author

Marien Krouwel is an expert in low code/no code and Cloud Services who helps customers accelerate and scale digital transformation and innovation, by implementing low-code/no-code platforms and setting up development centers for both citizen and tech developers.


marien.krouwel@capgemini.com



Maarten Stam

SAP technology consultant

LinkedIn


### About the author

Maarten Stam is an experienced SAP technology consultant working in Consumer Products, Retail and Logistics.


maarten.stam@capgemini.com



Ruud de Wit

Senior Solutions Architect

LinkedIn


### About the author

Ruud de Wit is a Senior Solutions Architect in the Retail and Agri business domains. He supports customers in their digital journey by creating the right fusion of packages solutions, low/no code platforms and custom coding.


ruud.de.wit@capgemini.com



Alex Bulat

CTO Global SAP business

LinkedIn


### About the author

Alex Bulat van den Wildenberg is the Chief Technology & Innovation officer Global SAP Business. He truly believes "Changing the world" starts with curiosity. It is the force that drives us to new discoveries, to exploration and new horizons. As the CTIO his responsibility is to encourage this curiosity and exploration.


alex.bulat@capgemini.com

# SAP AND DEVOPS: HOT? OR NOT?

Switching to DevOps can help to speed up processes across business – even in areas previously considered out of bounds, such as SAP.

**IN A NUTSHELL**

- Although they seem like an odd match, combining DevOps with SAP can actually speed up product lifecycle and improve overall process quality.
- Benefits of DevOps for SAP include minimized risk and more availability for product teams to focus on value-add tasks.
- In order to integrate SAP and DevOps successfully, a solid roadmap is key.
- SAP SE has already made the step to offer DevOps learning to help businesses realize the benefit of this approach.

SAP systems drive the business processes behind some of the world's biggest companies. But for many it has been viewed as ill-suited to the continuous deployment methods being adopted in other areas as part of a shift to a more Agile, DevOps-ready culture. SAP already offers a stable, configuration-based platform that cuts down on the need for development. Does bringing an accelerator such as DevOps into that environment make sense? After all, if something isn't broken, why try to fix it?

The problem is that, especially in development and testing, SAP has tended to follow a rigid waterfall model that's increasingly unsuited to modern business needs. This model is common in manufacturing where deployments are highly structured and lead to a "big bang" release which has all its requirements established in advance. But in a DevOps environment, top-down planning is replaced by a more fluid approach where Agile teams are encouraged to combine skills, work collaboratively and incorporate automation.

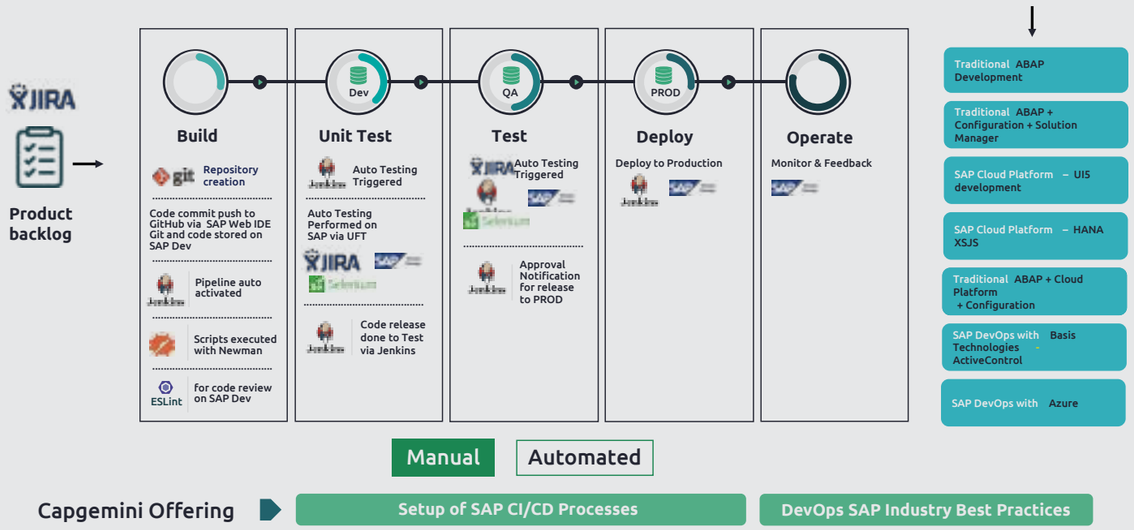
For this reason, DevOps and SAP have not been an obvious pairing, and for a lot of companies putting the two together would be a bold move. Yet it's one they might be persuaded to make if they understood the advantages DevOps offers in dramatically speeding up product lifecycle within SAP, opening up possibilities for application delivery in shorter and more frequent cycles, and delivering improvements in overall process quality with reduced risk.

## Speed up production with Continuous Integration/ Continuous Deployment (CI/CD)

Across industry the move to cloud and S/4HANA has put increased pressure on businesses to innovate and become more responsive to change. The move to DevOps can help make that happen, especially with its focus on business outcomes, and SAP teams need to be part of it. Of course, SAP brings specific challenges. But speeding up the production lifecycle is still possible, especially in the CI/CD pipeline where there are significant opportunities for increased automation.

It's true that at present many of the necessary automation tools are deployed from outside SAP. For instance a company facing a product backlog might use Jira to give an overview of the work needed to realize a project, including development and operations, and identify which areas are suitable for automation as well as those that still require manual intervention. Yet even within the pipeline it's possible to deploy some level of auto-activation, say with scripts that are automatically executed by a task manager such as Newman.

**Figure 1:** DevOps Practices : SAP DevOps Ecosystem.




In this way, SAP can integrate a variety of tools capable of automating the SAP lifecycle, while with ABAP it's also possible to integrate with Azure DevOps and Eclipse to access a complete toolchain that runs from code quality checks to automated configuration and source code management. This approach is also applicable for SAP configuration as well, and includes the use of a platform, like the SAP Business Technology Platform (BTP).

## Putting DevOps to work with agile

For some companies this kind of change is already happening. In response to the COVID-19 pandemic, a large retail company which had only made tentative steps towards Agile in its SAP division found that introducing a new DevOps working environment immediately accelerated cooperation and productivity. Within six months, new digital solutions were put into production as the ABAP-trained ops team shared ideation and responsibility with their associated SAP dev team – though the biggest win was the enhanced cooperation with business operations thanks to DevOps and its emphasis on validation. By asking, "Are we on the "right" track?" the risk of failed delivery for teams was reduced and there was an accompanying reduction in the need for additional work post-delivery. It's worth noting here that functional consultancy carrying out potential system configurations are considered to be part of the same SAP Dev team.

Ofcourse there are many challenges to be overcome in this kind of DevOps transformation. Often, it's possible to make a strong business case but still face questions such as: which solution should we choose? How does it connect with our move to cloud? Will our future move to S/4HANA be affected? And in addition to the lack of specialized DevOps tooling in SAP, companies often have to work with the complications brought about by multiple code sets or competing cloud platforms.

Solving these issues requires a solid roadmap and the right toolset, not least putting Agile development front and center. As was pointed out by James Roberts on an SAP blog in 2016, "You can adopt Agile for SAP without making the step to DevOps, but DevOps won't work without Agile[1]." Linking development and testing with business outcomes in an Agile way is a good first step towards DevOps only if it's backed up by automation across the SAP pipeline.

## Building an SAP value chain

For customers who are interested in combining DevOps and SAP, Capgemini is ideally positioned as a delivery partner rather than just a vendor. We also align on issues such as sustainability, not least in support of SAP's new Climate 21 initiative, and see the importance of building cross-functional, multi-skilled teams that stand ready to foster a collaborative culture supported by intelligent use of automation.

Much has been written about DevOps in recent years, yet a lot of companies remain reluctant or unable to make the move to Lean and Agile even as they want to develop more product- or customer-centric operations. In this context, guidance and shared responsibility is key – no company wants to break their old system in going after a new one. The shift towards DevOps is as much cultural as it is technical, freeing teams to both share their expertise and step out and learn from others. Often this exchange can be awkward to set in motion, but quickly reaches a "tipping point" after which it becomes the default way of working.

In this context it's important to think about SAP in terms of a value chain that has teams built around it to continuously deliver features for specific parts of the customer journey. With production happening in smaller time frames product teams will have more focus, especially if more automation is built into the delivery cycle. At the same time, risks are minimized since, should something go wrong during a production cycle, it can be quickly reversed and the source of friction accurately identified.

Another key area to consider when weighing up DevOps options is oversight. SAP processes have long benefitted from tools to enforce governance, audit, workflows and approvals. But DevOps for SAP is different due to its underlying architecture. Agile teams need autonomy and boundaries, so it's important that team members know who to reach out to on security and governance issues. There also needs to be more focus on program portfolio management in SAP since the high-level focus is often 2-3 years ahead.

For instance, if a company is moving towards an S/4HANA environment there might be things they need to implement in advance even while maintaining a shorter rolling production environment with DevOps.

Typically in an SAP environment it's important to think about shared surfaces and the interaction between enterprise architecture and standards, security, compliance, quality, and so on. Providing this type of flexible support is one of Agile's core strengths, since different teams can be accessed on an as-needed basis, something which could be critical when managing platform services that are designed to be scaled up and down as needed.

SAP environment it's important to think about shared surfaces and the interaction between enterprise architecture and standards, security, compliance, quality, and so on.

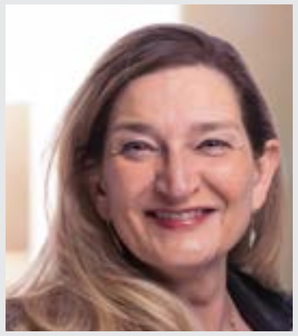
## A big boost for business

With SAP themselves now offering learning on DevOps[2] it's clear in which direction we're moving. In fact, if DevOps can work when applied to a foundational, business-critical platform such as SAP then it could also be a driver for wider adoption across an enterprise. An important first step towards making business processes fully Agile.

Capgemini's own monitoring shows the benefits are clearly visible. When combined with DevOps, an SAP deployment can lead to 40% more applications in production even as 76% less time is spent on deployment. We also see fewer outages and a 15% increase in productivity. Such results suggest that DevOps combined with SAP is most definitely "hot", even in the context of rapidly shifting, post-pandemic economic conditions.

That said, there's some way to go before SAP systems can be fully automated – and so realize the full rewards DevOps is ready to deliver. But the steps companies can take today have the potential to be very significant in the not-so-distant future, for example in reducing lifecycle and increasing efficiency in testing.

The question once was whether DevOps and SAP could work together. The answer is they already are, even if the biggest boost for business is still to come.



Fina Piazza

DevOps Expert

LinkedIn

### About the author*

Fina Piazza is an expert in Digital Transformation, with a focus on helping clients transition to an Agile way of working to remain on the cutting edge of Business and IT.

*This article was written by an external agency through interviews and in close collaboration with the expert.

fina.piazza@capgemini.com

### Footnotes

[1] https://blogs.sap.com/2016/10/14/devops-for-sap-laying-the-foundations-for-success/

[2] https://open.sap.com/courses/devops1





# IT'S TIME TO UNLOCK THE VALUE OF SUSTAINABILITY

Setting sustainability targets is critical for all organizations, but it means nothing if you don't have the means to measure your progress.

**IN A NUTSHELL**

- For companies that want to get serious about sustainability, accessing insights into the effectiveness of your initiatives should be step 1.
- Studies show just 15% of Fortune 1000 employees are satisfied with their companies' sustainability efforts).
- Sustainability needs to be a top priority regardless of how big or small your business is.
- With 75% of global transactions going via an SAP system, Capgemini and SAP are helping businesses to effectively monitor their environmental impact across their product and business processes.

With ever-stricter standards being set on climate goals, an increasing number of companies are having to integrate Environmental, Social & Governance policies directly into their investment decisions. It's partly a response to changes in consumer and employee attitudes – a recent survey showed that only 15% of employees at Fortune 1000 companies were satisfied with their companies' action on the environment and climate[1]. But across industries there is growing consensus that action is needed now to ensure a sustainable future for all.

To help make realizing these targets a reality, Capgemini has developed an integrated approach that leverages new and legacy technologies on ERP, while creating a holistic view on the sustainability in your business processes and value chain.

## Green core in action

Given Capgemini's announcement on plans to go carbon neutral by 2030, it should be no surprise the group has committed to embedding sustainability in its own solution portfolio. Capgemini looks to provide increased visibility into sustainability impacts across the value chain as well as contributing to and advocating for the sustainability of our customers.

Changes to the business ecosystem such as this will require top level leadership alongside the kind of collaborative partnership that can support companies that already have an ERP system. Fortunately, with the applications now coming to market, data and insights are set to be easier to manage than ever. Capgemini's approach will look to aggregate data from SAP transactions and calculate sustainability metrics through analytic apps and dashboards, providing actionable insights on how to improve sustainability.

Of course, promoting change at a global scale means this isn't just about ERP. But given that 75% of the world's transactions touch an SAP system, they are set to play a central role. SAP are already seeing change in attitudes to sustainability happening with many of their major clients – Allianz no longer insure coal-fired power stations, Maersk are moving to zero-emission vessels and Walmart have committed to taking CO2 out of their supply chain.

Such moves reinforce the feeling that every business, no matter their size, needs to be concerned about sustainability performance – and the change of mindset being promoted by SAP is part of that. The lesson is that traditional GHG reporting is no longer enough. We need more insights into issues such as carbon footprint, but also the ability to understand how they relate to productivity. At the same time there's evidence of how increased transparency has the potential to create a "domino effect" across supply chains as insights are shared and common goals can be established.

## Drawing the "green line"

Companies are increasingly being held to account for their environmental impact, with the expectation being that all businesses should know things like the carbon footprint of their products and business processes. What if it was possible to break down the total footprint and focus on where emissions were produced at each stage of production, from raw materials to manufacturing to shipping?

Understanding the end-to-end effect that production, manufacturing, logistics etc. are having on our sustainability initiatives requires a new approach of the kind outlined by SAP Climate 21. It's a key driver in the vision of SAP to link enterprise software with real-world sustainability benefits. By using analytics as an extension to existing ERP solutions, SAP becomes an enabler for sustainability across all business functions, helping to provide insight into how close a business is to becoming carbon neutral.

Long known as an out of the box ERP solution tailored for business processes, SAP already provides many organizations with oversight across their work environment in areas such as health and safety. Yet they also see how the emerging risks associated with climate change, from changing markets to increased regulation, are often not being addressed in a timely way. So with decisive action fast becoming a priority, they've developed a platform based around data collection and analysis that can deliver extra insights into businesses' sustainability footprint, whether in IT, supply chain or manufacturing.

In launching Climate 21, SAP executive Thomas Saueressig has referred to sustainability as the "green line", a third dimension to driving business success[2]. "SAP helps customers manage the top line and bottom line of their business," he says. "So we are in the best position to now help companies manage their green line by minimizing the carbon footprint and negative environmental impact of their product."

With Climate 21, SAP is providing businesses with the necessary tools to assess how close they are to carbon neutrality. The insights gained can then be applied not only to their own processes but those of other companies, such as suppliers. For example, a business might want to check whether the suppliers in their network are performing well or not, how that performance impacts their own sustainability and whether they need to build more sustainable partnerships or hold existing partners more accountable for their environmental efforts.

In practice, this means building an Emission Maintenance system that can support the work of a Production Sustainability Expert. They can then move from an overview of multiple processes to a highly focused level of detail through dashboard monitoring, examining the carbon emissions generated by the supply of raw materials and energy purchased for production processes alongside those created by outbound transport partners[3].

## Bring sustainability to the core

It's a fresh approach to an emerging problem, but in many ways the Climate 21 program builds on the existing SAP functionalities already in place at many companies around the world. SAP Environment, Health and Safety (EHS) management software already contributes to social sustainability of the workforce, SAP Ariba can boost ethical procurement, SAP Human Resource (HR) promotes diversity and inclusion, and SAP Concur provides insights on travel that can be used to offset CO2. Other SAP modules, meanwhile, can assist with the financial processes associated with recycling, as when empty bottles are returned to a supermarket, or the tracking of dangerous goods and chemicals.

On a wider scale, we know how technological development can be a multiplier for positive impact in the world. So while in business terms it often feels as if finance comes first and sustainability second, looking beyond "greenwash" marketing makes it clear that the two go hand in hand. Increased efficiency at the application level, say, can help deliver reductions in operating costs and greenhouse gas emissions. We can see this happening in agriculture, where digital enablers are actively shrinking the environmental impact of production and distribution, a key factor for consumers who are increasingly aware of how their food is grown and transported[4]

With a long existing knowledge base of industry processes, Capgemini recognizes that the development of ERP add-ons is a great step towards a more sustainable future, but the real win is in incorporating sustainability in your core business processes. Technology is a great enabler, but we need to use it in the right way to unlock its real value for the industries. A typical use case for industry might be an app which can display carbon emission footprints in real time for freight orders, with insights on carrier performance, route and method of transport. Capgemini has developed not only a set of SAP Accelerators, but a matching holistic approach for business transformation with a focus on the sustainability targets that are set by businesses today.

In launching Climate 21, SAP executive Thomas Saueressig has referred to sustainability as the **"green line"**, a third dimension to driving business success.

## Unlocking a sustainable future

Understandably, costs are still a major concern for many companies that might otherwise look to invest in more sustainable practices. But not only are the potential benefits long-term, it's increasingly obvious that sustainability is actually growing markets rather than shrinking them. A recent article by the World Economic Forum showed rapid growth in energy, food and finance markets, and quoted research suggesting the retail nutrition and wellness market alone could grow to US$50 billion by 2025[5].

If businesses are to take advantage of this expansion and unlock its future value, it's critical to put the right systems and approaches in place today. For Capgemini, it's evident that there is a growing need for business transformation because of the sustainability targets that are set by businesses. Having add-ons on your ERP system is a good first step towards transparency in your performance when it comes to sustainability, but it doesn't give you real time visibility into your entire landscape. Therefore, only business transformation with a focus on sustainability can unlock its future value.

Where should businesses be in a year's time on sustainability? What about 10 years? Developing a vision which looks ahead like this is obviously a formidable challenge. We need to look at the issue holistically and be realistic about outcomes. But thinking beyond where we are now is a way of ensuring that targets don't just stay targets. It also recognizes that there's no "magic bullet" or one size fits all approach. Gathering the right insights and applying the right analysis, though, is the first step towards securing a sustainable future for us all – and it's one we can take today.

Claire Knoester

SAP Sustainability Lead

LinkedIn

### About the author*

Claire Knoester is an expert consultant in SAP, specializing in transformation to the digital supply chain, with a focus on leveraging the latest in SAP innovation and technology.

*This article was written by an external agency through interviews and in close collaboration with the expert

claire.knoester@capgemini.com

### Footnotes

[1] https://www.sap.com/documents/2020/06/ec443a74-9d7d-0010-87a3-c30de2ffd8ff.html

[2] https://news.sap.com/2020/06/climate-21-track-greenhouse-gas-emissions/

[3] https://www.sap.com/documents/2020/06/76673d74-9d7d-0010-87a3-c30de2ffd8ff.html

[4] https://www.capgemini.com/wp-content/uploads/2018/06/Digital-Agriculture-Bytes-Against-Hunger-Trust-in-the-Agri-food-Chai....pdf

[5] https://www.weforum.org/agenda/2021/01/davos-agenda-sustainability-digital-revolution/





03

# THE AGILE DEVOPS WAY OF WORKING

1. Is Agile dead?
2. Only the best leaders have what it takes to create high performing Agile teams
3. DevOps: Why we aren't going faster
4. Security and Agile development: Striking the right balance

# IS AGILE DEAD?

Becoming an Agile organization has never been more crucial. And yet, there is a growing debate within application development as to whether Agile ways of working are helping, or hindering.

**IN A NUTSHELL**

- The problem with Agile frameworks isn't the rules, it's the reluctance to stray from them.
- If you don't need to scale – don't force it. Consider reducing complexity instead.
- Multinational teams are on the rise, with an increased focus on hybrid working models made up of partially remote workforces.
- Scrum Masters and Agile Coaches need to equip themselves with the knowledge to help their teams with their software development skills.
- Autonomous teams will accelerate your initiatives and enable you to be truly Agile.

For years, many organizations have successfully applied Agile ways of working in the development of their systems and applications. However, there are plenty that are still struggling to achieve the business Agility they were hoping for. It seems that while some of the "Agile community" are happy to adopt Agile ways of working, this does not apply to everybody.

There is a growing population on social media and across the developer circle that are expressing their discontent about Agile frameworks. They are too rigid. Sprints feel like running on a hamster wheel. And story points and velocity feel like a method for management to control development teams.

## Rules, rules, rules

The problem is, once a new framework is adopted, such as Scrum or SAFe, a certification system is set up and people must get accredited to be able to apply for jobs. And if you're asking people to pass an exam, it's almost guaranteed that they will only learn what they need to in order to get by, without giving much thought to the practical application.

Then the framework becomes the rule. These frameworks are a great place to start when you rely on them for guidance, as you would rely on a recipe when cooking a meal for the first time, but occasionally it's important to think outside the box to create genuinely great and innovative products.

The issue many organizations encounter is that they fear any deviation from "the rules", and as such, people have a hard time looking beyond them. The solution? Don't do the Scrum events because it says so in the Scrum guide. Only do them if you truly believe it can help your team be more successful. If not, look for other ways. The absolute key to becoming truly Agile is to adopt an Agile mindset. This includes being open to using the frameworks in new and Agile ways.

## How to scale

Most organizations don't have trouble setting up a few successful scrum teams, but the challenge really starts when organizations want to go full scale Agile. Methods for scaling Agile frameworks are often overly complex and misunderstood. Personally, I think it's a common misconception that one can scale Agile by just reorganizing into more Agile teams, providing a framework to coordinate these teams and having them participate in joint planning sessions. Communication between teams should be considered as a type of waste that should be avoided. That's not to say that collaboration is bad, but rather that if you're looking to increase flow and speed, having lots of dependencies between your teams doesn't help to foster that.

Instead of a scale-up framework, consider a scale-down approach. Devise a plan to scale down the complexity in your organization. If you can change your architecture to enable autonomous, self-organizing teams, then do so. Then, redesign the organization into product-oriented teams.

The scaling frameworks contain useful ideas in essence, but the drive for certifications systems have made them grow in complexity. If you don't need a scaling method, don't use it. In time, the most successful organizations will learn to use frameworks wisely and tailor them to their own needs.

## The future is global

For a long time, co-location was seen as a requirement for good teamwork and high-quality outputs. But for many developers, this is not the most productive way of working. A lot of people in the developer community are introverts, and flourish when they get sufficient alone-time. In the book Agile2[1], this is called "The introvert versus. extrovert problem". Agile ways of working often "prescribe" extensive face-to-face communication, which has long been begrudgingly accommodated by introverted members of the developer field. On the opposite side of the spectrum, software development requires a great deal of concentration, which is hard to do when you're working in a busy office – and you're also an introvert.

As has been widely acknowledged over the last 18 months, the end of the pandemic will see many of us continuing to work at least partly from home. In fact, our report on The Future of Work[2] highlighted that 6 out of 10 organisations reported productivity gains of between 13-24% due to remote working. And it turns out that many software developers are among this group. This, of course, varies from person to person, and the future of application development in this area looks to be a hybrid working model, with some developers in the office and others located at home or abroad.

A lot of people in the developer community are introverts, and flourish when they get sufficient alone-time. In the book Agile2, this is called:

**"The introvert versus extrovert problem."**

Which brings us to the next point. An increase in home working means companies can widen their geographic talent pool, employing the right people for the right job, without the restrictions of location. This means the multi-nationality of teams will also increase.

How will this help? To become more Agile, we need to be able to scale up quickly and add competent developers to our teams, no matter where they are. This implies that team members will be based in different countries. According to Heidi Helfland, teams will become more dynamic, and will change composition more frequently[3]. She challenges the conventional wisdom that in Agile we should have long-standing teams in order to be successful. We will find new ways to cooperate and "gel" as a true team. If teams are in different time zones, we will have to do more asynchronous development. In this case we discuss working with online tools, in which everyone can comment and edit in their own time, thereby reducing the number of meetings required.

That's not to say that the new hybrid working model won't present challenges. Team members might feel more disconnected from their organization and there may be less opportunity to share ideas or experience serendipitous moments. We need to find the right balance between time spent together exchanging ideas in discussions and time concentrating without distractions. Other areas to address are avoiding the classic trap of "video conference overload" to subsidize human interaction and ensuring that, while somewhat disconnected, teams still share a sense of having one common purpose. Semi-regular alignment meetings and, conversely, "no meeting" periods are one way to address these.

## Self-directing or self-organizing?

Agile teams are expected to become self-organizing, and it has become a misconception that self-organization also means self-directing. Some organizations have taken the misconception too far, diminishing the role of leadership as a purely facilitative and inspirational role. Successful organizations have strong leaders, who are often right. And they're successful because they have good instincts.

Ultimately, a successful Agile organization is defined by smart decisions. You can have hundreds of great ideas for new products or business models, but there is no cheap way to try them all. You need people with a real understanding of customers and trends, and who are able to determine what will have the best chance of success to generate value that benefits the company. Great leaders are visionaries who inspire others, and who are not ruled by their egos. That said, they are not afraid to be perceived as unreasonably strict when needed. In the future, the focus for Agile development will be on finding the right balance between authority and autonomy, in which a "one team" culture is adopted, and decisions are made in line with common goals.

Agile teams are expected to become self-organizing, and it has become a misconception that self-organization also means self-directing.

## The role of the Agility master

The wider adoption of Agile frameworks and ways of working seems to have caused a diversion of focus away from software development skills and software development methods such as trunk-based development, automated testing, automated deployment and CICD.

The role of the Scrum Master and Agile coach now leans towards team process coaching, and increasingly, Scrum Masters and Agile coaches have little understanding of the technology and skills required for successful software delivery. For a business to become truly productive, software developers also need coaching on their software development skills.

The person who assists the team in optimizing their collaboration efforts should not be a master of the Scrum framework alone but should also have knowledge about other Agile methods with an understanding of ways to optimize the software development process. These skills include DevOps, CICD and testing. Ultimately, the person leading the charge should not only be an "Agility Master", but also a "Technology Master".

## Be like water[4]

Agility is and will continue to be one of the most important skills within an organization. If you cannot move fast and deliver value, you are lost. Of course, Agile is not dead.  But it has become a misunderstood hype word that refers to a mix of several methods and frameworks and has perhaps been misconstrued as law, instead of the guidelines it was intended to be.

Successful companies and development teams of the future will benefit from allowing some flexibility with their Agile frameworks, as opposed to following them to the letter. This will form the basis of an adaptive organization that can make adjustments based on experience and lessons learned, and in which autonomous teams are empowered to deliver value. As we move to a model that is made up of increasingly multi-national, distributed groups, developers will be able to define their method for cross-team collaboration to find a solution that fits for their team, leading to greater overall satisfaction.

To create these autonomous team setups that will ultimately accelerate initiatives, organizations should adopt methods of continuous integration, continuous development and Agile architectures, to become truly Agile, fluid, rapid and dynamic – like water.

Rik Pennartz

Agile Coach

Linked in

### About the author

Rik Pennartz is an Agile coach,  whose core expertise lies in building high-performing, Agile teams and organizations.

rik.pennartz@capgemini.com

### Footnotes

[1] Agile 2: The Next Iteration of Agile; March 2021; Cliff Berg, Kurt Cagle, Lisa Cooney, Philippa Fewell, Adrian Lander, Raj Nagappan, Murray Robinson

[2] https://www.capgemini.com/wp-content/uploads/2020/12/Report-The-Future-of-Work.pdf

[3] Dynamic Reteaming: The Art and Wisdom of Changing Teams; July 2019; Heidi Helfand

[4] https://www.capgemini.com/nl-nl/technovision-2021-technologische-trends-in-het-bedrijfsleven/



# ONLY THE BEST LEADERS HAVE WHAT IT TAKES TO CREATE HIGH PERFORMING AGILE TEAMS

To successfully lead your Agile teams to high performance, there are 5 areas you should consider.

**IN A NUTSHELL**

- Teams simply cannot perform well in a toxic environment.
- Develop your mission, then find the people who are invested in making it happen.
- Truly great leaders put the needs of their team first – an empowered team yields better results.
- It's the leader's role to ensure teams aren't bogged down in things they can't control.
- Once your team reaches high performance status – it's up to you to make it sustainable.

In the last decade alone, a handful of giants and a growing horde of hungry startups are consistently overpowering and even replacing top traditional industry players. Do they have better people, more audacious leaders, or better technology? If you are in a leadership position, you know that these are not reasons, but excuses that you tell yourself. After all, who is accountable for hiring, leading, and purchasing?

Instead, it's more likely that they have fully embraced today's volatile, uncertain, complex, and ambiguous (VUCA) world, and understood the key role that team performance plays in it. Despite their stark differences in culture, companies such as Apple, Google, Amazon, Uber, Facebook and the successful startups each seem to have found a way to get the best out of their teams. Their way of achieving this, and the underlying belief systems, are so different from traditional management approaches that it would be unwise to ignore them.

In this article, we'll look at what it takes to build successful Agile teams from an executive perspective, breaking down this huge task into five logical steps.

## Step 1: People – Develop a candor antenna

In order to build a good team, we need to cast good people. While hiring for competency is pretty obvious, it is not sufficient. Candidates not only need to know their stuff (competence), they should also value team success over personal success (behavior) and they should be triggered by the challenge at hand (motivation). All three conditions should be met in each individual as a basic starting point for high performance as a team. It is relatively difficult to assess someone's team player attitude, but keep in mind that it's tougher to help them catch up on their soft skills than it is to develop their hard skills.

Specific vigilance is required towards toxic behavior. A brilliant engineer who belittles people around him, a controlling Business Analyst who undermines ownership of others by micromanaging, a self-doubting or competitive person who confuses a debate on ideas with personal judgement, are just a few of the behaviors that we sometimes accept to a certain degree. It's important to make decisions based on what's best for the team. Letting toxic behavior slip by because you need that person's specific skill set only serves to undermine every effort you put towards building a culture of trust. The truth is that a few false notes easily bring down a hard-working orchestra. Try to develop an ultra-sensitive antenna for behavior that suppresses candid conversations. People will naturally self-preserve by keeping opinions to themselves when there is a lack of psychological safety, but this very understandable reflex is the single most crippling element for true performance in teams.

The motivation side of the casting process is often the most overlooked, as many teams are composed via an existing pool of candidates. Look at the team and its mission from the perspective of the candidate: does the mission mean something to them? Do the technological or innovative challenges align with their personal learning objectives? Who will they be working with, not only team members but also the stakeholders and customers? In other words, how does this mission tap into the candidate's career path and personal motivators? Asking yourself these pertinent questions will help you to understand who will be truly invested in your goals.

## Step 2: Teamwork – use the Pygmalion effect to your advantage

As studies have demonstrated repeatedly, it is less about who is on your team, and more about how well they work together.

Team effectiveness models go back several decades, from Tuckman in the 1960s and GPRI in the 1970's (Goals, Roles, Procedures & Interpersonal Relationships) to Lencioni with the 5 dysfunctions of a team early this century[1].

Studying and applying these models is a great starting point. As an executive, your real challenge lies somewhere else: what is the fundamental mission for which you bring this group of people together? The traditional expectation is for a team to execute a series of activities – comparable to a machine. We focus on giving them the right input & steering, and we expect efficient processing and quality output. When managed well this can lead to good results. Unfortunately, it is a weak setup for VUCA. Frequent changes, decisions in ambiguity, experimentation and innovation are often not part of the game and therefore undermine the mission.

Compare this with a team that is given a purposeful problem and then receives enough context and support to come up with a solution. Context is critical: clear boundaries of what we are trying to solve, a good insight and access to customers, deep understanding of limitations and concerns, and so much more.

Simply printing out and displaying a vision or mission statement on the walls will not be enough. Also, more than steering committees, your team needs prompt support in any way imaginable for impediments that they are not able to overcome themselves – including motivation when an experiment did not go well.

In most cases, the measure of success tells a lot about the expectations that they are entrusted with. Are your teams measured through the impact of their solution (the outcome), or rather by quality and quantity (the output)? Through the nature of these expectations, the Pygmalion effect kicks in. By defining the work in requirements, we effectively reduce accountability – only to be surprised afterwards about a team's "lack of commitment".

Which of the two scenarios is predominant in your organization? If most of your teams exist to crunch requirements, many of this article's recommendations are merely plasters on a wooden leg.

## Step 3: Agile – Conway is our ally

In the 20 years since the Agile Manifesto, new ways of working have been uncovered and developed at breakneck speed. Applying Agile as a mindset more than as a methodology is, understandably, more challenging in traditional organizations than in emerging ones. Fake or veneer Agile is not uncommon – in which organizations apply new terminology and practices with little or no cultural or structural changes. From an executive perspective, you will have to lead the dance, changing your own leadership style and fundamentally transforming your middle management.

As an example, let's zoom in on how good we are at empowering our teams. On this aspect, the schism between traditional organizations and front runners is deep and has even become a reason for dynamic talent to no longer consider working for traditional organizations. Empowerment starts with the aforementioned "give them a meaningful problem to solve" and allowing the team (or team of teams) to discover solutions autonomously. This sounds easy enough until we start thinking about the implications and how difficult it will be to transform towards teams who are:

- Working in funded value streams (products) rather than in funded time blocks (projects)
- Handling the end-to-end story: from addressing the market and the customer need to prioritizing, developing, and maintaining the solution
- Functionally and operationally loosely coupled and decide autonomously on tooling, way of working, metrics, releasing, etc.

When we apply such changes, we are effectively turning the constraint of Conway's law into an opportunity:

**"Any organization (…) will inevitably produce a design whose structure is a copy of the organization's communication structure." Melvin E. Conway, 1967[2]**

At the same time this approach of multiple mini enterprises can't work well without introducing solid mechanisms of alignment and transparency. OKR (Objectives and Key Results) is one of the most popular solutions for this challenge, applied at Google, Intel, Uber, Snapshot, Sears, Zynga and countless others. Alternatively, each of the dominant Agile Frameworks (SAFe, DA, LeSS, Nexus, etc.) will embed its own answer.

### Step 4: Performant – where's the cognitive load?

Now that we have these autonomous Agile teams of great people, let's talk about performance. The lagging measure of performance should always be business related. However, in one of the most interesting performance research studies of recent years (Forsgren & co - Accelerate)[3], the combination of speed and quality proved to be the best predictor of business success. Teams that produce stable products (better Mean Time to Recover and lower Change-Failure Rate) will also be able to reach higher speed (higher Deployment Frequency and shorter Lead Time for Changes).

Please note that productivity, in the traditional sense of churning out estimated requirements (or even worse: lines of code, amount of test runs etc.), is a fundamentally different perspective.

Let's make sure that the teams worry about the right things. When observing a team in action, it is easy to detect the type of things that keep them awake at night. About what do they raise issues, complain, ask for help, or get excited?

- Wrong or foundational things: onboarding, tensions, uncertainties, unavailability, unreliability, asking unnecessary permission, hiding mistakes, politics, self-promotion, commitment, recognition, etc.
- Skill related things: professional shortcomings, gaps in the skill matrix, training, shore-related unbalances, knowledge hoarding, regressing defects, etc.
- Technical things: excessive complexity, external dependencies, security, deployment, technology, system performance, etc.
- Business things: customer behavior and loyalty, business decisions, product monetization, usage statistics, prioritization, user experience, innovations, etc.

As a general rule, you need to ensure that they grow out of the foundational issues, that you provide the right level of training, coaching and hiring on the Skill issues so that their excitement is all about Business and Technical challenges.

The cognitive load of teams has its limitations. The more they need to sort out basic stuff, the less they can participate in what they were actually hired for. A specific mention should be made about the technical challenges, where many organizations have not yet moved away from overly complicated, heavily interdependent monolith systems and processes. These will consume the valuable time and creativity of your engineers, slowing down change and reducing their involvement in business challenges. When we talk about the future of application development – simplification of environments and processes, and a move from monolithic to MACH architecture is key.

Handling the end-to-end story from addressing the market and the customer need to prioritizing, developing, and maintaining the solution.

### Step 5: High Performance – The runner's high

With the previous steps in place, it is likely that some of your teams will reach a high-performance state. They reach an incredible flow of output combined with delighted customers. Business needs are continuously met or surpassed, unexpected advances in technology and solutioning are reached. Intrinsic motivation rules and an atmosphere of comradery blossoms within the team.

While this is a very rewarding place to be, it is never a comfortable one. They are on a runner's high, where everything seems possible, but running is an energy-consuming activity. Make sure to keep the pace of work sustainable. In fact, it is sometimes needed to protect them against themselves. Coach them to avoid a work regime that takes over their personal life. While this may be required in situations of exponential growth or crisis, it should not become the norm as it will undoubtedly break people, commitment, and loyalty in unpredictable ways.

Another challenge may be alignment with the rest of the organization. High performing teams may start to develop feelings of greater authority, neglecting or belittling other teams, company rules, leadership, etc. While it is fantastic that they challenge the existing state, this should always be done in a constructive manner, and your behavior and (lack of) interventions will make or break the bigger picture.

At the individual level, a surprising new phenomenon is visible. Because the team is so tight and inter-challenging, the engineers rediscover the virtues of solitude. Especially in today's new normal, they are applying the advantages of being un-distractible: sit down, turn off Teams, Slack and Facebook, and just code for most of the morning. Alignment is done asynchronously. As well, when observing an engineer "in the zone", there's a close collaboration between man and machine.

There are two more tasks on the executive's plate here. First: get out of the way. Don't interrupt them with meetings unless really needed and make sure their day has long periods of work time. Second: be very attentive to complaints about the hardware and software that they are using. Frustrations of processing power, screen quantity or quality, not having the latest AI tool in the IDE, missing administration rights and so on are the thousand paper cuts that will kill the team performance.

As a leader, you are in that fortunate role to be able to provide your teams with the context for success, and to grow them from merely being competent, to being performers. It's a tough job, but a very rewarding one.

The cloud offers them immutable infrastructure, with services such as data storage, data analysis, security and identity solutions, and application development solutions, all using a pay to go model.

Stijn Follet

Global Leader Agile

LinkedIn

## About the author

With more than two decades of experience in Leading and Coaching Agile teams and organizations, Stijn Follet is Capgemini's Global Leader of the Agile & DevOps Movement, co-creator of the Agile Contract Manifesto, and global content owner of the Agile training curriculum.

stijn.follet@capgemini.com

## Footnotes

[1] Patrick Lencioni, 5 Dysfunctions of a Team

[2] http://www.melconway.com/research/committees.html

[3] Forsgren, Humble & Kim, Accelerate

## Further Reading:

Marty Cagan, Empowered

Amy Edmondson, The Fearless Organization

https://www.jstor.org/stable/2666999?seq=1

Douglas Squirrel & Jeffrey Frederick, Agile Conversations

https://www.nirandfar.com/skill-of-the-future/

https://www.ics.uci.edu/~gmark/chi08-mark.pdf

Google's Project Aristotle:
https://rework.withgoogle.com/print/guides/5721312655835136/

Simon Sinek, Leaders Eat Last





# DEVOPS: WHY WE AREN'T GOING FASTER

Our focus on technology and automation has caused us to neglect the real engine behind the tools: People.

**IN A NUTSHELL**

- The key to a successful DevOps transformation is investing in your people, not the latest tools.
- Practice running, not buying better sneakers.
- Your “lead time” is the metric to watch.
- You are not Amazon (and that’s okay).

People are agile, not tools. While the tools are a crucial part of a DevOps transformation strategy, it is the strategy around the people and skills required to use these tools effectively that is usually under-developed.

DevOps practices are often touted as an escape from the conundrum of having to choose between speed and quality. Whereas, in more traditional software delivery cycles, one must often choose between how quickly something can be delivered and the anticipated quality thereof, with the correct practices in place it has been shown repeatedly that it is possible to deliver faster and with higher quality.

As part of the research carried out by Dr Nicole Forsgren, Jez Humble and Gene Kim for their book Accelerate[1], they discovered that “…when compared to low performers, the high performers had:

- 46 times more frequent code deployments.
- 440 times faster lead time from commit to deploy.
- 170 times faster mean time to recover from downtime.
- 5 times lower change failure rate (1/5 as likely for a change to fail).

This trend has only accelerated since then.

While Accelerate outlines 24 capabilities that can help organizations become high performing, this article focuses on just one, Continuous Integration (CI), as an example of how a practice can impact an organization’s performance. That said, I will provide a few suggestions at the end of the article on where to begin laying the groundwork for successful practice adoption.

We observe that organizations moving to a “DevOps” way of working will adopt a “build it and he will come” mentality, rushing to adopt various automation and technology in the hopes that these innovations will lead to an increase in performance. They make investments in cloud migrations, set up a team of DevOps automation engineers and invest in setting up pipelines that they expect will speed up delivery. Sadly, they are often only frustrated in their ambitions.

Focusing on automation exclusively is like building a 4-lane highway without having any drivers to use it: while the infrastructure improvements are crucial and the teams will depend on them enormously, they are all but useless unless teams have been trained in how to adapt their way of working.

In a recent presentation about fulfilling the promise of CI/CD, Charity Majors (CTO of honeycomb.io) raises the point that a good CI/CD pipeline should not take longer than 15 minutes to run from the moment an engineer merges a change in version control to when it is deployed on production[2]. This is the benchmark which our DevOps software delivery cycle is going to be measured against.

It is a seemingly unattainable speed for many organizations still struggling with the basic skills and tooling required to improve their delivery. It may not even be a worthwhile goal to pursue, depending on the business needs of the organization and the customers, but it is important to know what “high-performing” really looks like in order to challenge ourselves to do better.

The time from first merge of code to deployment on production is typically called lead time, one of the metrics mentioned prominently in Accelerate, an indicator of how well your organization is doing in terms of technology and practices. So, if a fast lead time is important, and automation is not the complete answer, then how can practices help improve lead time?

46
times more frequent code deployments

440
times faster lead time from commit to deploy

170
times faster mean time to recover from downtime

5
times lower change failure rate (1/5 as likely for a change to fail)

## Continuous Integration

When thinking about Continuous Integration and Continuous Deployment (CI/CD), many leaders skip over the CI bit and focus solely on the wonderful potential of CD. This is, after all, the goal of quick time to market (with higher quality) that they dream of.

However, Continuous Integration is not a foregone conclusion in technical teams. Agile practices in software development are not the same as the skillset that is required for more traditional approaches. The focus on iterative development and fast feedback cycles, as well as the focus on rapid deployment, requires a team to adapt their ways of working, their tooling and, most importantly, their mindset in order to be able to fit into this new world. It can be a steep learning curve and in many digital transformations they are neither given the assistance, nor the "continuous learning" time (a crucial Agile principle that often falls by the wayside due to delivery pressure), to adapt.

In order to integrate in a continuous fashion, there are several practices a development team must master:

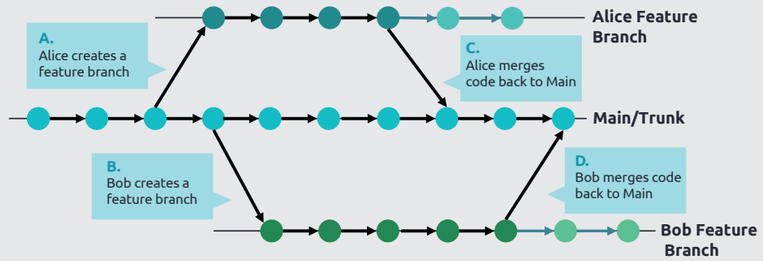
1. Small, frequent changes (which requires support from leadership and product owners).
2. Rapid review feedback (usually through pair/ensemble programming).
3. Good, automated testing (not just coverage, but functional/regression testing.
4. A focus on “constantly shippable” code.
5. Decoupling deployments from releases (through, for instance, feature flags).

Without the first 3 on the list, it becomes impossible to avoid merge conflicts and regressions in the code, which will lead to a cycle of your teams constantly firefighting and struggling with their code every time they want to produce something that can be deployed, ultimately preventing code from flowing through the pipeline rapidly.

Similarly, without numbers 3-5 on the list above, it becomes impossible to control the quality of what is being deployed and what is and is not visible to your end users. As the old computer science adage goes: “Garbage In, Garbage Out”.

To give a concrete example, let us take "Branching Strategy" for version control to illustrate how the choice of approach and practice can impact your CI/CD throughput.

The "Git Flow" model of branching is quite popular in development teams but can lead to situations like this:




The circles are "commits": moments when the code is saved to the version control system. The "main/trunk" line indicates the code that is deployed to higher environments and eventually production and is shared by all the developers in the team/organization.

In this example, Alice "branches" the code at A to create a new "feature branch" so she can work on her new feature in isolation. From this point, her code diverges from on-going development that continues on trunk and she misses any new developments other developers merge to trunk in the meantime.

While Alice is busy, Bob, at B, also decides to branch code to his own feature branch to work on a different feature. Now we have 3 versions of the code that are, temporarily, no longer in sync. Alice is missing any changes from trunk and Bob’s branch while Bob is missing updates from trunk and Alice’s branch.

Eventually, Alice merges her code back with trunk at C. Depending on the volume of changes being produced in the organization, she could be running behind trunk by several tens to several hundreds of changes. If she has changed anything other developers were working on, she will have merge conflicts and she will have to resolve them together with the other developers.

This process can take anywhere from hours to weeks.

Similarly, Bob merges his code back to trunk at D, after Alice, and now has to deal with all the merge conflicts between his code, as well as those introduced by Alice and the other developers.

If you wish to have continuously shippable code, you can't allow either Alice or Bob to simply merge their code back into trunk blindly. They will have to check for merge conflicts before merging their code and spend all that time resolving conflicts (and holding up the release of their features) until everything is working again.

If they do accidentally merge that code into trunk, they might break the entire codebase and make it impossible to deploy to production until these issues are solved.

Lastly, it is typical for feature branches to live on for longer, so this whole process repeats itself as parallel development continues on the feature branches with all the merge/integration issues that comes with it.

This is without complicating things further by adding the possible delays a code review process might add to this situation.

Now imagine the above scenario playing out with 5, 10, or a 100 developers. This is precisely why the practice of Continuous Integration came about – as it is the only way to counteract these issues.

Take a look at the following strategy:

In this approach, often called Trunk-Based Development, the team(s) strive to keep Main/Trunk in an always shippable state. If they branch out code to work on, they keep these branches short-lived, typically there is a span of a day, maximum, or a few hours, between when they branch and when they merge back in.

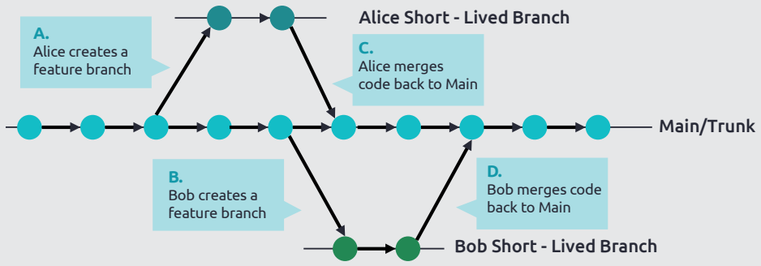
**The advantages are clear:**

1. Branches don't live on, cluttering up the repository
2. It's always clear where to deploy code from when it needs deploying
3. Small merges mean that it's relatively trivial to resolve conflicts or, if you need to roll-back, you don't lose a lot of work.
4. If you're using a review process, small code changes make reviewing them relatively fast, reducing lead time.

Many Agile teams also use pair or ensemble/mob programming to work on stories together. This builds the "review" work into the actual development process as well as improving everyone's knowledge of the code base. In such teams, they can directly commit their code into the main branch as they work, leading to an even faster flow of code changes.

This is where the "Continuous" in "Continuous Integration" comes from and it also demonstrates the power of fast feedback loops.

It requires training and constant practice for teams to develop these skills, but it's an investment that's even more important than investing in the modernization of tooling. In fact, as teams hone their practices, they drive the demand for newer tools (and thus help shape the automation strategy).




Sadly, many organizations end up declaring their digital transformation a failure and return to the "old way" of doing things when the outcomes are not as promised. Further complicating this space is that there is always a newer solution on the horizon. If your DevOps transformation didn't quite deliver the results you were hoping for, perhaps a no/low code platform will solve the problem, or introducing AI Ops or GitOps will produce the desired outcomes?

The temptation to throw yet another methodology or tool stack at the problem is great but only serves to confuse and hamper teams even more as they fall ever further behind on the technology curve.

## Laying the groundwork

Going back to the beginning, you'll remember that I said Continuous Integration is but one of several practices that a team needs to become familiar with in order to practice effective Agile development while making the most of the automation tooling available to them.

While the above examples show concretely how just one practice can interfere with getting the full benefit of a DevOps pipeline, there are a few general principles leaders at an organization can use as a guideline to help them support teams on their transformation journey:

1. **Build slack into a team's day/week/sprint.** Besides having the added benefits of reducing burnout and providing a more sustainable pace, it allows teams to reflect on their work and take the time to learn new skills.
2. **Encourage continuous learning by providing opportunities.** A "self-organizing" team will determine what skills they require to do their jobs and it is crucial they can be honest and transparent about where they might be lacking. Help them close this gap by providing training/coaching opportunities.
3. **Recognize that tools are constantly evolving. What is cutting edge today will be legacy tomorrow.** A team that has a mindset of learning and experimentation will do a much better job of staying up-to-date and being able to capitalize on new developments. They will come to their leaders with ideas for improvements instead of the other way around!
4. **Focus on business outcomes, not productivity.** Productivity is a very difficult metric to quantify and trying to measure it often leads to undesirable results. Focus instead on business outcomes such as whether your delivery to customers has actually become faster and of higher quality. Has your **lead time** improved?

The key takeaway from all of this is to make sure you're investing in the right things, and your people are a great place to start. It's crucial that organizations practice the skill of running, instead of hoping that a new pair of sneakers will solve their performance woes.

Saket Kulkarni
DevOps Coach
LinkedIn

### About the author

Saket Kulkarni is an Agile/DevOps coach and expert in the planning and execution of Agile/DevOps transformations – from introducing best practices to planning the road ahead for future transformation initiatives.

saket.kulkarni@capgemini.com

### Footnotes

[1] Accelerate: the Science behind DevOps: Building and Scaling High Performing Technology Organizations, by Nicole Forsgren et al., IT Revolution, 2018, pp. 9–10

[2] https://speakerdeck.com/charity/cd?slide=17


# SECURITY AND AGILE DEVELOPMENT: STRIKING THE RIGHT BALANCE

To better balance between Security and Agile development, organizations need to build security into their culture.

**IN A NUTSHELL**

- Agility, enablement, automation, shift-left approach, and use of good security tools, are important for a successful Agile security program.
- Agile and DevOps teams move fast and are continuously learning and improving. Security needs to enable them instead of blocking them from moving forward.
- Security must be Agile to keep up with Agile teams. Security must think and act quickly and iteratively, respond quickly, and keep learning and improving along with developers.
- In Agile and DevOps environments, the attack surface is continuously changing. Threat modelling has to be done in a lightweight, incremental, and iterative way.
- Take your security culture seriously and place people in the center

The increasing velocity of change in IT poses a serious challenge for security and compliance, where the focus is on doing things right and minimizing risk, rather than being responsive or providing fast time to market. Few organizations truly understand how security can work in harmony within Agile and DevOps. The future of IT will be faster, more collaborative, and more automated, which creates an entirely new threat landscape. The question is: how do we develop secure software in an Agile environment?

## Security isn't time consuming, your process is

A lot has been written about the benefits of Agile development, including its ability to increase agility and productivity, and why organizations must focus on delivering secured applications. But organizations often find it both challenging and complex to implement this way of working. In the end, implementing user requirements always takes precedence over optimizing the IT delivery and ensuring that the application is secure by design. Instead, security is often treated as an afterthought, only looked at towards the end of the process, if there is still some time left to do so at all. Then, the issues that inevitably come to light at that point cause delays to the release, which in turn gives security a bad reputation amongst the developer community.

To counteract this thankless cycle, organizations are constantly looking for ways to integrate security in their software development lifecycle (SDLC), but often come up short. It's not uncommon for businesses to, even in today's digital world, see security as time consuming – mostly because they are waiting until the last minute to address it. Agility, enablement, automation and use of good security tools all play an important role in rectifying this issue – but it's up to you to make sure you have them in place.

In order to address the negative perception of security in Agile, there are four key challenges that need to be solved:

## Security being seen as a "non-functional requirement" (NFR)

This is a requirement that is related to the state of the system, rather than, as the name suggests, functional areas of the system. Security is not immediately visible. The added value is often not immediately seen by all stakeholders. What we often see in organizations is that functional aspects take precedence in a sprint and planning, because direct results can be seen here. Security is done when there is "still time". However, security is an essential component for the functioning of an application. And therefore needs to hold equal weighting when prioritizing tasks and must be taken into account from the beginning of the process, same as functional requirements.

## Security not being a priority for the product owner (PO)

If security is seen by the team as a "non-functional requirement", as stated above, we will often see that a Product Owner prioritizes functional requirements or use cases before the non-functionals which, in this case, would wrongfully include security. PO's will usually do this because it helps them show the direct added value of the team during the last sprint. It's much harder to demonstrate the value of security, and if we are really honest with ourselves, this is one of the main reasons for security coming last in the list of priorities. The direct added value is not seen (or felt) directly, and therefore it's deemed unimportant.

## The structure of user stories

Another thing we commonly see is that security is not seen as a standard part of a user story in most organizations. Predominantly, functionals are described while non-functionals are, consciously or not, not declared. This means that security isn't integrated in user stories and therefore isn't discussed structurally during refinement sessions.

## Lack of Agile-ready security practices and tools

Organizations often struggle with integrating security in the Agile/DevOps environment, because the Agile way of working is mostly aimed at functionality. Besides that, the perception that security disturbs the agility of the process when findings occur creates a "head in the sand" mindset – if we don't see the issue, it doesn't exist. Tools that are used to scan the application are another "extra step" that can make your life easier. While this doesn't fit in with the overall concept of Agile, security can actually be an easy functionality to integrate (without adding extra time or steps), as long as you have the right tools, next to manual checks.

### How can I integrate security while observing Agile best practices?

To strike a better balance between security and Agile development, security must be seen to add value. Currently, it's often seen as a roadblock, but in fact we need to think of it as having the ability to increase the quality of the code and the development processes. Security has to be built into existing user stories, and Agile teams should feel in charge and responsible for carrying out secure development.

To achieve this, a security-first mindset has to be adopted. To ensure that even the most Agile development processes are done securely, automation of security needs to be integrated, and compliance and (code) quality control built into your pipeline.

### Creating a secure development lifecycle in an Agile organization

## Build security in through user stories

A way to include security in the SDLC is through 'evil stories' or misuse cases. Your team should be thinking about what the user wants to do and can do (the happy path), but they should also be thinking about what the users are not allowed to do (the evil path). In evil stories, the Agile team thinks about how an attacker could misuse a function. This forces them to think about what specific actions are needed to defend against threats.

## Threat modelling & security requirements

In the same way that evil stories ask the team to re-examine requirements from an attacker's point of view, threat modelling changes the team's perspective to think about what can go wrong and what can be done about its design phase. This can be done by drawing a rough context diagram of the system's main parts and brainstorming potential threats and attacks from multidisciplinary views. Teams can use existing threat info and attack patterns already observed in production as a basis. Understanding the attack surface and the impact of any changes can help to identify risks and potential weaknesses that need to be addressed. Because the attack surface is continuously changing, threat modelling should be done continuously.

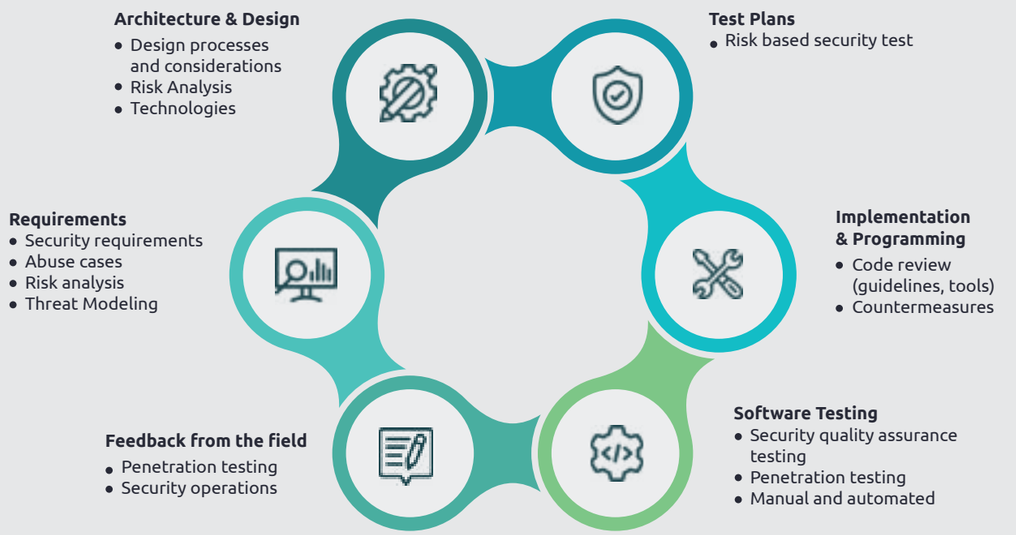
## Put the Agile team in charge of secure development

It is essential for the entire team (e.g. PO, developers, architects, functional and technical application engineers, testers) to know the value of adding security to the process so that they can feel responsible for secure development. Therefore, it is critical to empower your team with knowledge and time. Employees who can think like a hacker will be star players in this process and are the key to ensuring secure and quality software applications.

## Incorporate continuous integration security practices in the SDLC

Security has to be part of the way of working from the very beginning (SHIFT-LEFT). Often, security is only addressed at the end of the process through a pen test or scanning tool. If you think about security from the beginning, issues are discovered earlier and are easier to fix. This saves a lot of time and effort from the team, not to mention the cost of development. Image 1 and 2 show the secure software development touchpoints in a typical Agile/DevOps lifecycle.

**Figure 1:** Secure Software Development Touchpoints (DEVops).






### How to build applications that are secure by design

## Secure coding

Providing safeguards against security vulnerabilities during Implementation & Programming (see image 1) are not yet a standard part of the curriculum in many courses. Knowledge is therefore not always present in teams. However, secure coding starts with changing behavior. The mindset of a developer should, as said before, not only focus on what the application needs to be able to do (functionality), but also what it should NOT be able to do (security). In addition, it is important to offer training on a continuous basis, and to empower your developers with enough time, resources and tools. The moment there's an issue, your people need to have support in solving it. This will help diminish some of the frustration that often comes into play when security issues arise.

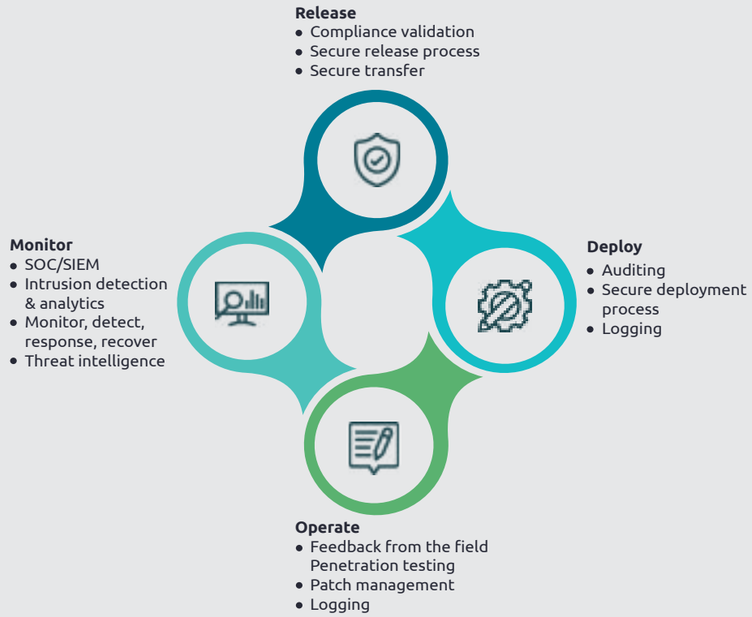
## Code review/SAST/DAST

As you can see in image 1, code review and security testing are essential parts of the development lifecycle. While code review is done in most teams, it can also be helpful to introduce secure coding guidelines, based on known best practices like OWASP top 10, to integrate security in this process. The right tools can help to speed up the security testing process. If you incorporate these in your regression tests and link reporting to this, it becomes a standard part of the test activities you do during the sprint (continuous security testing).

## Open-source security

The deliberate or innocent use of open-source components can expose organizations to different risks, including licensing compliance, security, quality, operational risks, and intellectual property rights infringements (for violating third party copyrights, patents or both). As no one provides indemnities and/or warranties against such potential infringements of open-source components, organizations will have to bear the burden of risk when using them. This high risk could expose businesses to

**Figure 2:** Secure Software Development Touchpoints (DevOPS).




injunctions and damages, and even criminal sanctions.

That is why you must ensure that you know what open-source components are included in your applications, make sure that trusted versions were downloaded from trusted sources, and that these components are kept up to date. SCA tools can do this automatically and can be integrated into your build pipelines to automatically inventory open-source dependencies, identify out-of-date libraries and libraries with known security vulnerabilities, and fail the build automatically if serious problems are found.

## Build compliance and quality control into your pipeline

Implementing controls within the CI/CD pipeline is absolutely necessary. Control objectives do not change, but the way the controls are implemented and how they need to be tested is different.

Standardization of the IT estate results in the consistent and reliable automation of controls. One way of reaching this is by defining and managing system configuration by using code that can be versioned and tested in advance (Infrastructure as Code). This gives more precise insight into the security baselines used to deploy configuration items.

Another advantage is that an auditor can quickly verify whether required infrastructure security configuration settings have been applied, whereas this can be difficult to assess when configurations are managed without automated configuration management.

Also, using a source code repository guarantees complete insight into all changes to the application source code and infrastructure components by recording what was changed, when and by whom.

The truth is, building a thoughtful security culture requires buy in from everyone, not just the security team. As we look ahead to an increasingly digital, remote and online world, there just isn't room for treating security as an afterthought. Besides actually creating more work for yourself in the long run, sub-standard security could spell disaster for your applications and your business. Although it's not shiny or glamorous, with few discerning benefits on the surface, security is the intersection of people, process and technology. And that means it is absolutely imperative for the overall success of not just your security program, but your organization as a whole.


Matthijs Greep

Management Consultant - Security, Privacy & Compliance


### About the author

Matthijs Greep is a Managing Consultant in Information Security and Chapter Lead Security, with a focus on Privacy and Compliance in Application Managed Services.

matthijs.greep@capgemini.com


Sanne Kuijpers

Lead DevOps & Security Center of Excellence

LinkedIn


### About the author

Sanne Kuijpers is a Cybersecurity Consultant with specific expertise in Secure Software Development (SecDevOps).

sanne.kuijpers@capgemini.com





04

# CONVERGENCE OF APPLICATIONS WITH THE PHYSICAL WORLD

1. Software runs the physical world, but the physical world is tough on software
2. What is your sensor telling you?

# SOFTWARE RUNS THE PHYSICAL WORLD, BUT THE PHYSICAL WORLD IS TOUGH ON SOFTWARE

## A new view on IoT systems makes seamless IT/OT convergence a reality.

**IN A NUTSHELL**

- As we strive for a more efficient, innovative business processes, we need to think outside the box while further aligning our IT and OT (Operational Technology) as a united force.
- To do this, we need to harness data from IIoT to act as the lifeforce of both IT and OT, aligning both aspects of your business with a single source of truth.
- It's this high-quality data that will provide advanced insights into your entire organization as a whole, enabling data-driven decision making.
- This kind of transformation can only happen when we shift our mindset to one of continuous integration and continuous deployment.

Today software runs the physical world. But what happens to the stream of data flowing in the opposite direction? The smart machines and robotized sensors at the heart of the Industrial Internet of Things (IIoT) platforms that monitor many modern production and distribution systems provide the raw material for a potential transformation of the IT/OT relationship that has far-reaching consequences for business across sectors.

Naturally, in the new business environment industrialization has to be done in a smart way, with real-world business challenges reflected in the way we develop software. This is why many of the cutting-edge ideas related to modern application development covered by the previous articles converge here. As we try to realize greater rewards from innovations in technology and process efficiency IT and OT should be allies not enemies, as has sometimes been the case in the past. The revolution in IT is also coming to OT – but to realize the full benefits we have to learn to think and operate differently.

### Making smart systems smarter

In this modern industrial environment, control over the physical world is increasingly detailed and sophisticated. Everything is managed by specific software – and it's getting smarter every day. Data from IIoT, the Industrial Internet of Things, can provide the pulse that keeps industrial systems running on time, like a pacemaker, and is already a business driver for shorter time to market, lower costs, predictable quality in production and improved sustainability.

Considered as whole, changes in the way IT/OT are aligned offer a fresh view on business development. It's an outlook that should be of particular interest to clients who are considering innovations such as scaled-down micro factories, which are closer to customers and can scale to meet changing production needs, or more flexible production models, which can respond to shifts in demand and offer shorter product development cycles.

It's about taking advantage of a more general move to data-driven operations. All software business solutions thrive on data, whether it's channeled from the enterprise, the shopfloor or within the value chain. Previously many businesses struggled with a limited view on the real world, but rapid developments in technology and the deployment of smart monitoring systems are creating new distributed environments in many sectors – well beyond early use cases such as power distribution networks or manufacturing production lines.

For example, a French jam manufacturer had invested in new high tech, stainless steel equipment but were still having occasional quality issues with certain production batches. Seasonal fluctuations in the fruit harvest were one factor, but with gelation being such a sensitive process they needed a better overview of production as a whole. By developing a system to gather data from across supply chain and manufacturing processes, including machine data, order data and quality data, and then apply data science to the results, they were able to realize a capability and depth of insight they didn't even know they had.

### Data is the key to convergence

Applying OT and IoT enables you to draw out higher quality data to develop deeper insights into your operations across more locations. If we use healthcare as a metaphor, a medical practitioner might take a patient's temperature or blood pressure to gain a high-level view of that person's overall health, but it's not enough to diagnose specific issues. However, if they analyze blood samples, or carry out an X-ray or ultrasound, the doctor receives higher quality data relating to the patient, enabling them to develop a more detailed picture of the person's physical wellbeing, while having the ability to target pain points. Much like our patient, your organization is full of untapped data that can be leveraged with the right tools and technologies.

When we talk about applying smart software in critical business processes in a controlled manner, the testing and verification requires proper simulation and test environments so that failures are found as early as possible, way before they are released in the real world. Another example is the use of digital twins to create virtual representations of industrial processes for research and testing purposes, using data gathered from sensors and smart machines that's processed through AI and machine learning[1]. For example, a manufacturer of e-cigarettes was experiencing production line quality issues, and we were able to create a simulation of a part of their manufacturing process to find the root cause. By viewing how the virtual environment operated together with the client's subject matter experts,

we were able to pinpoint the most likely cause that resulted in the irregular quality issue, which might otherwise have been impossible to detect.

Data-centric operations such as these clearly have a direct link to IIoT application development. But be careful, it works the other way around, too. Errors and failures in software can have a potentially damaging impact on physical machinery, so it needs to be circumvented to prevent it from becoming the number one outage problem. However, with all the right components in place and accurate monitoring it's possible to imagine a scenario where, in manufacturing terms, "the store" is always open and mission critical software is able to deliver 24/7 production and near-100% uptime.

Ensuring such a model functions smoothly is certain to be challenging and may require specialized oversight, so it's especially important to prioritize collaborative engagements where possible. SMEs and other thought leaders also have a significant role to play, since they help identify solutions in a relationship built around guidance and trust, particularly when companies find it difficult to explore solutions outside of their existing areas of expertise.

In any case, for teams to work successfully on these kinds of solutions over the long term, they need access to state of the art capabilities at both ends of the IT/OT process. Not only will this fuel the ambitions of the best available talent, but it clears the way for OT and IT departments to start talking to each other in ways that are mutually empowering, to develop joint roadmaps together, to understand jointly how to realize the potential benefits of parallel software and hardware development. In our report on Secure Operational Technology[2], we highlight the importance of timely and continuous investment in securing your OT environment, and discuss how this impacts IT/OT projects.

## Changing minds not just machines

Smart hardware is evolving all the time, but really it's software that is making the difference by increasing flexibility and delivering better insights. Ofcourse mapping the physical world onto the digital world, and vice versa, is one of the biggest challenges in application development. Innovations such as Zivid's industrial 3D color cameras for robotic applications give a new view on manufacturing processes[3]. But such tools are only as smart as the software that controls them. In this world of "mechatronics", where different technology domains work closely together during engineering, we need proper intelligent feedback loops that create truly data-driven solutions[4].

At the same time it's clear that understanding and meeting the challenge of what we termed Real Process Management (RPM) requires more than just a technology focused approach. There's a change in mindset needed as well as a change in technology. One of the issues in this domain is finding the right people with the right knowledge and, because the existing technology is often dated or there are legacy systems to work around, companies can find it harder to attract the best young talent in the field.

It's not simply a technological problem. Millennials are used to controlling data-rich systems remotely – they do it all the time on their smartphones. But those kinds of tools are not always available in industrial environments. So to bridge that expectation gap we also need to leverage modern IT developments such as micro-services, edged computing, containerization, version control and CI/CD pipelines to achieve the best application and data rich experience in OT.

## Collaboration, Convergence, Continuity

What needs to be avoided is "poortunity" – situations where an opportunity for improvement has been identified only to be poorly implemented. In overview, it's a three-part development where digital convergence brings about digital collaboration with a goal of digital continuity. A seamless interaction between people, process and technology. We can already see IT and OT moving together – with the potential to boost areas such as time to market and cost of delivery.

To realize the full business benefits, however, it's critical that software is at the foundation of the solution. Lead times in manufacturing are increasingly determined by software capabilities, so the smart factory model needs software designed with that in mind. And while the development process might look like standard IT, it's not. For one thing, deployment will take place across an entirely different environment.

Either way, this isn't about a turf war between competing organizational entities. If we fail to find convergence, then silos develop, and the pace of innovation slows down. Thankfully IT people are known for their flexible approach and ability to work across enterprises supporting different departments, from sales to marketing to operational processes. They have the capability to understand different business domains and can create the conditions for ensuring there's unity between the IT/OT worlds. Similarly, OT people understand interacting with the real physical business world, so by working together they can achieve the needed physical/virtual digital continuity.

Without doubt, exciting times lie ahead for application development in this domain. Not only is it getting smarter every day, there's the potential to connect operational data with real-world business outcomes. Taking advantage of IIoT's complex data flows will require new human skills to evolve alongside the developments in hardware and software. But by starting the conversation on how best to leverage IT expertise to change OT software development, we see the potential to transform the way business is done for the better.


Milé Buurmeijer

Cluster Lead Connected Devices

LinkedIn


### About the author


Milé Buurmeijer is an expert in Digital Roadmaps, Gap Analysis, Innovation Management and IT landscape, with a focus on large enterprises.

mile.buurmeijer@capgemini.com



Arjan de Bruijn

Management Consultant

LinkedIn


### About the author*


Arjan de Bruijn is a Managing Consultant and Technology Lead specializing in IoT and OT.

arjan.de.bruijn@capgemini.com


*This article was written by an external agency through interviews and in close collaboration with the experts.

### Footnotes

[1] IIoT in digital twins: https://capgemini-engineering.com/uk/en/insight/the-benefits-of-building-a-digital-twin-of-your-factory/

[2] Secure Operational Technology: https://www.capgemini.com/nl-nl/bronnen/secure-operational-technology/

[3] Zivid: https://www.zivid.com

[4] Mechatronics: https://capgemini-engineering.com/it/en/tradeshow/international-conference-on-mechatronics-technology-icmt-2019/

# WHAT IS YOUR SENSOR TELLING YOU?

The allocation of reasoning: how (and where) to add the logical reasoning of interpreting the physical measurement of sensors.

**IN A NUTSHELL**

- A sensor is as good as the preservation of the context that comes with it.
- The measured value of a sensor needs history, logic, context and even installation knowledge to be of real value to the software that makes decisions upon it.
- Logic, or "intelligence" can be added to the sensor itself, to the backend where sensor data is gathered but also in between: as a native cloud app.
- A smart sensor compares to a dumb sensor as a 5-day weather forecast compares to a thermometer.
- A sensor measures a physical change. It is the software that turns that change into a value that can be trusted.

As we move to an increasingly digital world, we rely more and more on the input of sensors. This is true not only in a business and industrial environment, but in everyday life too (my fridge tells me the inside temperature of various zones every time I open the door). In this article, I'll discuss how to add logic and reasoning, and perhaps AI, to the measurements of a sensor in order to give us the information we want, rather than the physical measurement it reads. "Is my milk getting off" is a more important question to me than knowing the temperature of the milk's surroundings.

## Allocating reason

A sensor rarely answers our question directly. Unfortunately, I know of no toasters that actually sense the browning of the bread. Instead, we use a timer to prevent burnt toast, based on our own assumption of the average time needed to bake it to our personal preference.

Since a sensor measures a physical quantity, we need to translate that measurement to the context we are interested in. And interpreting that context, the reasoning, is where software comes into play. What would be the best architectural location to implement that reasoning? How do you design the reasoning in a software solution?

When developing IoT solutions, one should be aware of the assumptions one takes when processing sensor data. The reliability of the measurements is an obvious basic requirement. And immediately after that the reasoning starts when the software makes decisions based on the sensor measurements. Those decisions can be taken in the sensor itself, in the middleware handling multiple sensors' inputs, in the backend accumulating the sensor data or in the presentation of that data to you as a system user or to other software systems. So how do we choose the best place for implementation?

As a CIO or Enterprise Architect, when scoping the requirements for a sensor-based solution, don't feel that you need to take it on yourself to decide how and where the sensor data is transformed into human readable presentations. Let your system provider work out the scenarios of implementing the logic and reasoning of the system for you, so you get the results that best fit your requirements.

Capgemini Custom Software Development (CSD) focuses on creating custom software solutions (e.g. custom IoT solutions). As an example of what that means in practical terms, one of our most recent projects came as a result of the Covid-19 pandemic. To enable employees to still utilize office spaces when they needed it, there was a sudden demand for systems that would monitor the number of people at the office and assign available workstations so that everyone could maintain a safe distance during the day. After developing smart sensors that could measure the presence of people (see What makes a sensor device intelligent by Marco van der Pal[1]) we also developed backend logic to interpret the input of those multiple sensors. More recently, we have started the development of virtual sensors that could compile inputs from multiple physical sensors and interpret the readings within its own logic container, so you can gather inputs from a variety of sources in a single sensor.

If we use the example of the SmartOffice solution mentioned above as a guide, given that it leverages a combination of sensors, data gateways and presentation software, I can provide examples of how and where various solutions and applications of logic can add the most value.

## Reasoning sensors (aka smart sensors)

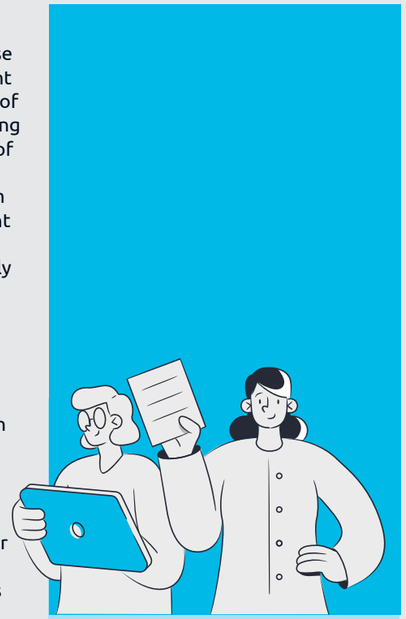
A reasoning sensor has a built-in expectation of the actual real-life context in which it is being used, using filters to isolate a part of the measuring range. For example, a desk sensor is calibrated to detect a human, so it won't react to a mosquito, but it would react to Brutus, the aid-dog of one of your workers, because he's in the same measuring range as a human.

That's fine when you want to know the number of bodies in your office

that produce carbon dioxide, but troublesome if it triggers a software alarm when two “people” are too close to one another for a particular amount of time. This is where the application of reason comes into play. If the reasoning in the sensor is sensitive to behavior of the data pattern, for example sitting at the desk and moving the chair from time to time is identified as a different sensor output than lying under the desk as Brutus does, you’re more likely to get the desired result.

So, the logic container in the sensor may use historic data to recognize patterns and change its state accordingly. AI logic may be used to recognize complex patterns, although this is just a software solution to effectively implement pattern recognition. The power of software running the reasoning in the sensor itself is the fact that it can be tuned or adapted after the sensor is deployed. So for the Solution Architect it means that a single sensor type may be purchased, and the sensors are tuned, software-wise, to accommodate the environment of each and every sensor. Making the sensors context aware also means that the middleware (data collection) or backend (data storage) of the solution doesn’t have to do the reasoning as to the occupation status of the desk. However, if the sensor says that the desk is occupied then the backend still has to know how many desks are in a room to say something about the room occupation. That logic is simpler, but still needs to be managed.

The software of the sensor is sometimes referred to as firmware, and that’s because it sits firmly in its own hardware environment. This type of software needs hardware (processor or controller, memory, I/O ports) to run, which makes the entire sensor more costly. The software itself usually is quite hardware dependent and therefore very technical. And because this type of software is not easily updated when in operation (updating means more hardware components to establish communication) the software may not be flexible to changing environments. So smart sensors are great, but they aren’t the perfect fit for every situation.



This kind of architecture makes the configuration and maintenance of the back office environment a big task. For example, any time there is a change in the physical environment, like moving a desk, it must be taken into account in the back office solution as well.

## Back office logic

A very typical and widely used implementation scheme of logic is to do the data interpretation and calculation at the system back office. In this scenario, the sensors themselves may be simple and cheap, which is fine, because all they’re doing is sending their physical measurement to the back office, which is where the real work happens. In our SmartOffice example, temperature sensors send the temperature, while PIR sensors send an exceeded threshold in heat radiation. The back office adds context to the data (the floor characteristics, the data behavior patterns when desks are being used) and calculates the likelihood of whether or not a person is sitting at a desk.

This kind of architecture makes the configuration and maintenance of the back office environment a big task. For example, any time there is a change in the physical environment, like moving a desk, it must be taken into account in the back office solution as well. The back office solution must be a perfect digital twin of reality in order to maintain the accuracy of its reasoning. And that digital copy is hardly ever a true twin.

One of the hardest challenges in IoT is to be prepared for all the events and changes that physical installations face in real life. So as a software designer, you have to focus on the most probable conditions and make sure the rare conditions or situations do not cause reasoning problems.

That said, a major advantage of back office reasoning is the ability to notice patterns or situations that individual sensors cannot. If a chair is broken and nobody chooses the desk with the broken chair, the desk sensor only reports “no occupation”. The back office logic however sees that the occupation of that desk has a different using pattern than the other desks in that room. In the office this is probably not a typical use case as workers will solve that situation themselves but for a train section this may be important information. Perhaps the seat is too dirty to sit on? In theory, the logic could be capable of spotting a piece of gum on chair 3A.

From a software development viewpoint, the solutions tend to be quite demanding on the back office environment, because it’s a central unit where all data comes in and where all logic is processing that data. Not only that, but the software needs to be flexible enough to serve small installations as well as multi-building environments. Different modules need to be developed to offer different functionality, for example knowing the availability of a meeting room, in addition to the workstations in the communal spaces. If all sensor processing logic is done in the back office environment it means the logic becomes a serious and complexly integrated part of the whole solution.

## Logical/virtual sensors

Centralizing and decentralizing computer power has been a constantly evolving wave in IT throughout its existence, driven primarily by the cost of hardware performance, by the performance needs of software and the technical difficulties of communicating quickly and efficiently.

The reasoning in an implementation architecture is the heart of the software logic and may take place at different locations, both centralized and decentralized. In the age of cloud solutions, anything can be virtualized. It’s not uncommon to have virtual PCs, virtual systems, or virtual users, so of course it stands to reason that the same can be true for virtual sensors, which we touched on briefly in the beginning of this article.

For the Capgemini SmartOffice solution, we are now in the process of finalizing the implementation of virtual sensors that combine the power of smart sensors with the context awareness of back office systems. The result is a very flexible software component that resides in a totally virtual world. By taking input from multiple physical sensors, it can add logic, reasoning and context awareness in a flexible and maintainable software component architecture. On the outside, you see a single sensoring device, but what it’s actually doing is sending the sensor updates as MQTT messages (Message Queuing Telemetry Transport – a lightweight, publish-subscribe network protocol that transports messages between devices) to its subscribers. It also acts as a data push component and sends updates via WebSocket API (SignalR), and has a REST API interface (a request-response interface commonly used in Web applications).

Because this type of sensor architecture is virtual and pure software, the flexibility and scalability are endless. As the desk sensors send the individual desk occupation to the backend, the virtual “room sensor” uses the same sensor inputs to guard a maximum of 15 occupied desks in that room and the virtual “floor sensor” does the same for the entire building floor. The data from the floor is sent to another COTS application used by the facility department to manage heating and air quality that is being managed at floor level. With virtual sensors it becomes possible to build, combine and deploy new sensors in an IoT solution without even connecting a single new piece of sensor hardware.

Apart from offering new information derived from bundling multiple sensor inputs, the virtual sensor is also applicable as a watchdog for your physical sensors and their operating parameters. Things like data analysis, error handling and also managing redundancy amongst sensors can all be done by virtual sensors. Using this approach, the physical sensors and back office applications become loosely coupled rather than hard wired, providing new functional possibilities and flexibility when it comes to maintenance and support during operation.

## Designing logic means designing trust

What this all really comes down to is that when designing and building an IoT solution, it's important to realize that sensors simply communicate the physical value they sensed. There is no such thing as a people counter that can be quickly screwed to a door to help you manage capacity. At its very core, the sensor only measures differences in infrared light – which is not enough to do much of anything without logic and reasoning.

The software logic and reasoning of that measured data is what turns the sensor into a people counter. If you want to trust the outcome of the sensor measurements (and obviously you do) one could say that designing and implementing the logic and reasoning in an IoT solution is actually designing the trust of the data.

What I hope has become clear over the course of this article is that there are different architectural locations where the reasoning software can be implemented, and realistically there is no good or bad location per se. In my opinion, designing the logic is an integral part of designing the overall IoT solution – from sensor level through to presentation layer. In some situations, sensors can become smart by adding context awareness. In others, the reasoning is done purely in the backend and simple sensors are fine. And in some situations, the flexibility and scalability of virtual sensors is perfect for your specific IoT challenge.

The moral of the story is, don't blindly trust your sensors to do what you need them to do; design the trust by designing the reasoning.

Wybe Horsman

Manager Capgemini High Tech Center

LinkedIn

### About the author

Wybe Horsman is the Manager at the High Tech Center at Capgemini NL, a project center for custom IoT (or sensor-based) software projects and mobile solutions.

wybe.horsman@capgemini.com

### Footnotes

[1] https://www.capgemini.com/2020/09/what-makes-a-sensor-device-intelligent/





05

# EVERYTHING IN MODERATION

## Making the case for good-enough Quality Assurance

**IN A NUTSHELL**

- Organizations are balancing out their QA efforts enough.
- Leaders in Agile realise that moderating quality principles, deliver better value in the long run.
- The key is to find the right balance between quality and agility.

The future of Quality Assurance in an Agile organization should be focused on “good-enough”. Quality is a trade-off between risk, cost and value. That won’t change. What will change is what qualifies as good-enough testing, which will differ in every case and evolve over time.

There is zero doubt that the adoption and consumption of software is accelerating. As consumers’ wants and needs change, organizations and their software must learn how to adapt. Combine cultural shifts, like the rapid transition to working from home, the high pace of new emergent technologies and the urgency to find a way to be quick to market, and the acceleration makes sense. Yet, the increased speed of development and a myopic focus on only some quality principles at the expense of others is setting organizations back. The solution is simple: moderate quality assurance to fit the organization’s needs while still allowing for future change. This can be done by pushing for quality while also defining the necessary requirements for good-enough quality assurance. Only by doing both can we find a balance that allows us to move forward confidently into the future.

We often see our customers undertaking different quality activities, when in reality they are overdoing it. To tackle this issue, I’d like to talk about some examples of quality principles, and how moderating them can bring value to your business.

## Products should be built fast, so they can fail fast

Generally, the theory is that product development needs to move quickly to the minimal viable product (MVP) stage. So, typically, you should build your product as soon as you can to get that MVP out quickly, to then gather feedback on that product early and see where it fails to meet expectations. Without stating the obvious, getting this feedback in a timely fashion will help to save time on rework.

When we observe leaders in Agile, we see that, commonly, they opt for adding some moderation in speed. By slowing down the development they enable themselves more time to spend on the early conversations, the discussions about the requirements of the product, and adding space for thinking. This is essential, as people are not always good at combining the various ways of thinking associated with building, the creative process, and analysis all at once.

Build fast, yes, but when we talk about moderation, I believe there’s a pace that is just fast enough to deploy quickly, while still allowing time for the right amount of reflection, with plenty of room for careful thought as well as building the MVP.

## The customer is always right

Product quality lies in the eye of the beholder. The user-based definition of quality has saturated the QA way of working. Because, well, your customers must be the only ones capable of defining quality given they are the only ones who truly know what they need. Right?

And yet, we often find that in actuality customers rarely know exactly what they need, nor do they have the ability to articulate those needs in a coherent, comprehensive way so that we may act upon them.

A more moderate approach is to include multiple definitions for quality. For example, a product-based approach, where quality is based on quantifiable attributes of the product. Or the manufacturing-based approach, where the aim is the quality improvement that leads to cost reduction. There are a few more; the point is, no single approach is better than another. The goal is moderation in each, finding a combination of cost and value measurements that go beyond purely the customer viewpoint.

## Automate all the tests!

It is inevitable to include test automation for app development. In fact, if we’re honest, organizations are going to need a large amount of test automation in place. It’s a best practice for a reason. The trouble lies in keeping the test automation fit-for-purpose. Namely, tests that reduce risk and are in support of the development and maintenance of the product.

In the last Capgemini World Quality Report[1], we can see across our clients that the level of automation is shrinking compared to previous years. This is because organizations are starting to understand that the cost of automation efforts can actually outweigh the benefits.

Moderation in test automation requires pruning almost as frequently as producing new test cases. How many automated test cases have your teams got rid of lately? Often as a product progresses, some checks and tests become obsolete. Honestly, some are never any good to begin with. I would suggest reassessing the granularity, modularity and reusability of your test cases.

## A bug is cheaper to fix if found early

Initiatives to shift software testing efforts "left" to the early design, specification, and code testing phases should pay off handsomely in reduced defects. However, this does not cover the whole picture. Risks of failure should be considered, and the ability to fix an error in production quickly is crucial. A comparison of the cost of further time spent “shifting left”versus the value of those further improvements should be compared to other options in the software development lifecycle.

Especially when we consider options on the “right” such as beta releases, feature toggles and A/B testing, which could be used to accommodate controlled and fast experimentation in production.

“At every point in the life cycle, we must compare the present quality of the product against the cost and value of further improvement.” - James Bach (Good Enough Testing)[2].

That cost and value shouldn’t only be determined in terms of whether the improvements are on the “left” or “right” of the software development cycle.

Initiatives to shift software testing efforts "left" to the early design, specification, and code testing phases should pay off handsomely in reduced defects. However, this does not cover the whole picture.

## Continuous Delivery to production

Improving the frequency at which changes can make it into production is essential. What can happen, though, is that in focusing on optimizing the time-to-market, quality measures, such as testing, start to be perceived as a hindrance. In the Capgemini Quality for DevOps Teams report[3], we give practical examples and valuable knowledge on how organizations can implement quality as part of their DevOps culture.

For example, it is recommended to examine the metrics used to measure delivery success closely. Delivering in one second versus one hour is impressive only when it also delivers the right value and shows the impact on the code. There should always be a focus on more than just speed and frequency, otherwise it's only about shortening the time to market at the expense of other important quality attributes.

The other blind spot here is that the CI/CD pipelines need management like any other software project. This is often either completely forgotten or centralized beyond usefulness. The key here again is a balance. Management of those pipelines while also preventing that pipeline solutioning is promoted to a goal in and of itself, hogging too many valuable resources that would be better spent elsewhere.

## We should be as Agile as we can be

Yes, we need our teams to be Agile, particularly in the pursuit of finding a way of working that is adaptable and allows for autonomy in making improvements. However, there is a risk that with, for example, twenty teams, they will each develop twenty different ways of working. A situation that is not ideal for measurement across teams, nor for managing cross-team improvement. Boundaries counter-intuitively allow agility to form, mostly by reacting to those limits.

For example, there are clear guidelines to the architecture that may be used, and deviation from this causes maintenance issues. Yet, for one feature, there is a clear case to deviate for the sake of the business value the change can give. The collision between the two allows navigation and rebalancing of both limits, the essential conversations that need to occur between expanding the architecture standards and what constitutes enough business value to outweigh the increased maintenance. Autonomy without limits isn't ideal, agility within boundaries is a way to keep resetting and staying adaptable. We touch on this concept in more depth in our article, "Is Agile Dead?".

## Adding moderation

The risk of over commitment cannot be stressed enough. It is common to become too heavily invested in continuing to the finish line to the point that we become blind to the art of moderation.

Making the hard choices to delete test cases, building features, and even cancelling entire projects feels like failure. However, these actions are necessary for maintaining a balance where quality is good-enough to provide value to your business without overburdening your teams.

Finding the right balance point is a lot like standing on one leg. Everything must move around to stabilize the body until an equilibrium is found. While balancing on that one leg is a precarious position, it is incredible for building the right muscles to stay standing. The key to this balance is fundamentally in the principle of moderation, not to overexert one way or the other to prevent yourself from falling.

Quality assurance is shifting to maintain that balance with the organizations. The test organization of the future is organized according to the good-enough definitions of quality for that specific scenario. Lately, an increasing number of quality advocates, quality coaches, and even quality catalyst roles have been emerging.

This trend signifies that it is fundamentally becoming a role that facilitates the idea of finding that sweet spot, the Goldilocks of QA that's "just right".


Sanne Visser

Managing Consultant Quality Engineering

LinkedIn


### About the author


Sanne Visser is a Managing Consultant within the Quality Engineering Department at Capgemini in the Netherlands, with a technical focus on E2E testing in complex software systems.

sanne.visser@capgemini.com


### Footnotes

1] https://www.capgemini.com/research/world-quality-report-wqr-20-21/

[2] James Bach "Good Enough Testing" https://www.satisfice.com/download/a-framework-for-good-enough-testing

[3] https://www.sogeti.com/explore/reports/enterprise-devops-report-2020-21/

### Further Reading

https://hbr.org/2021/04/research-how-to-get-better-at-killing-bad-projects

https://seths.blog/2019/03/move-fast-and-break-things-isnt-a-worthy-slogan/

https://www.infoq.com/articles/agile-stoicism-testing/

https://hbr.org/2018/06/in-praise-of-extreme-moderation

https://sloanreview.mit.edu/article/what-does-product-quality-really-mean/

06

# INTELLIGENCE VERSUS PRIVACY

AI is becoming an increasingly core component in application development, but it's also data hungry. To protect users, we need to infuse our intelligent apps with more privacy.



**IN A NUTSHELL**

- Although it requires a specific approach, it is possible for AI and privacy to coexist.
- Differential privacy is the key to intelligent applications – already in use by tech giants like Apple.
- Synthetic data is the way forward for all future-facing Application Development.

Applications are becoming increasingly intelligent and sophisticated, incorporating advanced AI and Machine Learning to meet the needs and expectations of customers and users. So it's no big surprise that the development of applications now requires more data than ever before. But this shift in app development demands more interactions between production data and AI during the development phases.

From an end user perspective, building AI technologies into your applications will prove beneficial in the overall usability of the final product, but using data and AI in app development also raises some additional concerns around privacy. Because when it comes to intelligent assistants, smart home appliances, your phone, watch or other wearables, the information that all of these devices and apps gather is becoming increasingly private, intimate and, essentially, off limits. In order to make use of this rich production data for application development in a way that complies with data protection guidelines, we need to understand how we can assure that we make the most of this abundant information while still adhering to privacy guidelines.

## The intelligence versus privacy paradox

In the past, the majority of the data we gathered came predominantly from online activity. Now, we have the ability to tap into a lot more personal "offline" sources, like home environments or even data and information concerning peoples' medical health. Having access to and using this data comes with a great deal of responsibility. How do you make sure that by using it you're not overly impacting your customer's privacy? Yet there is a paradox at play here, because it's that precise offline data that will help you to build more intelligent apps – the kind that your customer is demanding. A dilemma.

This paradox between privacy and intelligence is where a lot of businesses struggle. According to GDPR, data is any information relating to a natural person or "data subject". This includes everything from names, addresses, and emails to credit card details, social media posts and other similar information. Regulations like GDPR help to govern and hold companies and individuals accountable for the way they use and gather the data, and it is absolutely crucial that these rules are properly observed.

This means that while AI gets trained on and learns from data, it's your responsibility to ensure that any of the data used in this process isn't sensitive, doesn't violate the privacy of others and is not being used to break any laws. However, this is not to say that you can't collect information. In fact, data is becoming increasingly valuable to businesses. So, it doesn't take a genius to work out why companies are creating and curating more of it than ever. However, as it starts to pile up, it's important to understand how to manage it.

So how can we deal with the intelligence vs privacy paradox? Well, there are numerous ways of using data in a more private manner, the most promising being Differential Privacy and Synthetic Data.

## Differential privacy

Differential privacy simultaneously enables data scientists to extract useful insights from datasets containing personal information, while also offering stronger privacy protections. Using a statistical method for protecting individual privacy, it allows you to perform a query on a dataset of sensitive information so you can learn something useful from it. The clever part is that the results produced are nearly indistinguishable from uniformly random data – meaning it's almost impossible for the individual's sensitive data to be exposed.

This method also makes it possible to answer sensitive questions about private information without leaking data that relates to other people. But there is a tradeoff when it comes to accuracy. It's also worth keeping in mind that the more differentially private your data becomes, the more you need to keep an eye on bias in your models because amplification can have disparate effects on underrepresented groups in your data.

From protecting medical records to enabling the collection of useful data about individuals under regulatory constraints, differential privacy can be applied to the analysis of all types of sensitive information about individuals. This can include medical histories, financial records, website clickstreams and even election voting records. It even prevents an entity with access to the raw data from linking together different instances of data concerning a single individual. So it's no wonder this method has been gaining traction amongst data scientists for the last 10 years.

Apple has been one of the biggest adopters of differential privacy techniques, one being an algorithm called Local Differential Privacy within the Safari browser, which helps to guarantee users' privacy by removing identifiers from the data and transmitting it via an encrypted channel to be stored for no more than three months.

Because of its versatility, many of the big tech companies have started incorporating differential privacy heavily within their organizational structure. The method gives them the ability to preserve privacy while still harnessing the data required by hungry AI to improve their services. However, while it has gained the seal of approval from some of the largest names in the business, differential privacy is not the only way to create a more privacy-oriented approach to AI and intelligent apps.

## Synthetic data

Synthetic data looks and feels just like the real data, with all the characteristics and relationships present, but without the physical human attached to it. At the moment, current practice for most companies is to simply anonymize data, which isn't regulated by GDPR but still comes with risks. Anonymized data can't be traced back to a certain individual, but recent studies have shown that it could still be traced back to the underlying individuals, which doesn't rule out the possibility of adversarial attacks (Nature Communications, 2019)[1].

The advantages of using synthetic over real data are two-fold. First, you can create an entire dataset that looks and feels like your real data but without the security risk of any data breach – which is ideal for companies that operate in highly regulated industries. Second, this solution is completely scalable, meaning we can create endless amounts of data using only a very small sample of real data. The only down-side is that you do still need to create it from a representative sample of the actual data.

Companies can use synthetic data for software testing where using production data has GDPR challenges. Often, the access to large quantities of good test data is challenging. In this instance, synthetic data provides a great solution as it removes the GDPR challenges and can provide your test team with the levels of high-quality test data that they require. The other advantage of using synthetic data in a testing scenario is that it often has more variety in the dataset than the actual production data itself, so testers can test with more unique cases.

## The future of data and AI

From an ethical standpoint, it's good that we are struggling with the moral dilemma of data vs privacy, because it shows that we are taking the privacy of our customers seriously, instead of prioritizing the release of new products and services. Although using private data would indeed enhance the customer experience, the risks are too high.

That's why Application Development teams are constantly striving for new developments, new techniques and new approaches to overcome the issues of privacy, building a world in which we can respect our consumers while also meeting their needs. The way we use data and AI is changing, and so the adoption of new techniques, like the ones outlined in this article, are just one of many next steps we need to take in the future of data and application development.

Mark Oost

Global CTO in Analytics and AI Services, Sogeti

LinkedIn

### About the author

Mark Oost is Global CTO in Analytics and AI Services at Sogeti, using his expertise to lead the strategic development of Sogeti's Global Artificial Intelligence portfolio.

mark.oost@sogeti.com

**Footnote:** [1] https://www.capgemini.com/2020/09/what-makes-a-sensor-device-intelligent/





07

# THE APPLICATION OF ETHICS

Consumers and users of software applications expect those same applications to behave ethically – even being capable of fostering ethical behavior in companies and society.

**IN A NUTSHELL**

- Consumers expect applications to be ethical.
- Ethics can supply significant drivers of positive change.
- Value sensitive design offers a method to design ethical applications.
- Purpose-led organizations are better suited to create ethical applications.

A significant number of people expect organizations, like businesses and government bodies, to use algorithms that are ethical, meaning that they are transparent, fair and non-discriminatory. So how can we embed the ethical discussions around applications into our organizations?

## The Importance of Ethical Applications in Society

Coded Bias[1], currently shown on Netflix, is a must-see film. The documentary shows how software applications, based on Artificial Intelligent (AI) algorithms, perpetuate inequality and discrimination. Facial recognition applications are failing to recognize women of color[2]. Police surveillance systems flag innocent bystanders as wanted criminals[3]. These AI-applications have caused a lot of uproar in the AI-community, but also in general society. Do we really want computer systems that make decisions on our behalf without knowing if those decisions are fair?

> **"We are all grappling every day with algorithmic determinism. We are all being scored"[4]. (Amy Webb in Coded Bias)**

There are many applications that influence the lives of both customers and citizens. So much so, that software applications decide whether you are eligible for loans, parole, or social benefits. Because these decisions can have severe personal repercussions, evaluating the ethics of application development becomes crucial. As these applications become more prominent, we need to establish some form of control. How can we assess that these decisions are fair and balanced? Well, the assessment starts with the design of the application itself.

It is the purpose and use of the software that causes the ethical problems. The questions we need to ask ourselves are: is the purpose of the application ethically sound? Are the consequences of the application acceptable for individuals, minorities, the poor, and society? Are our fundamental human rights, like freedom and justice, ensured when using these applications?

## Ethics of software applications

Software applications are human products, as are any of the other artefacts we've built into modern society. The difference being that software products are far more complex than simple household utensils. The more intrusive and impactful a product is, the more intense the discussions need to be. This can make the ethical discussion complicated and volatile.

> **"Customers expect transparent and fair [algorithmic] interactions, and with clear accountability"[5]. (Capgemini)**

We see this in the ongoing discussions about the negative impact of social media. To begin with, social platforms were praised for their ability to bring people together, but now the discussion has shifted focus to the power and influence of Big Tech on society and its values.

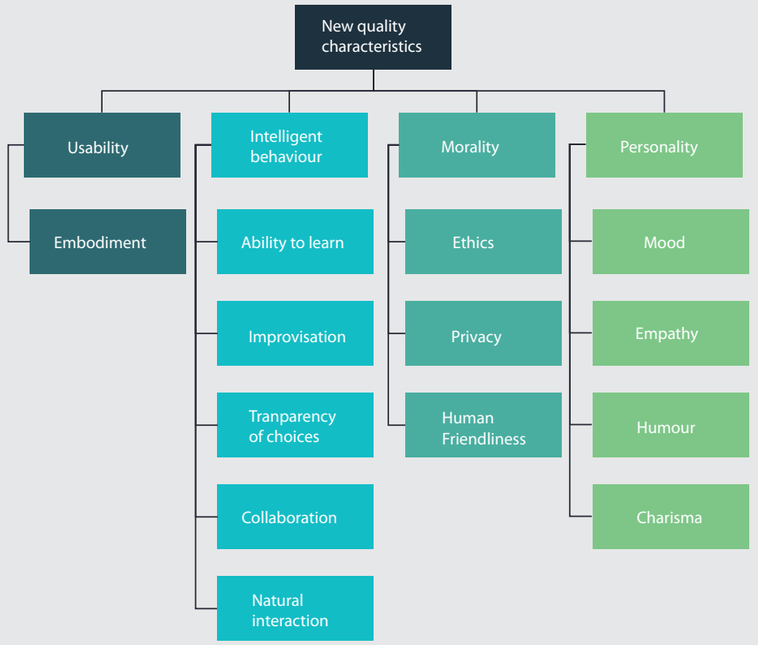
"While consumers may not be changing behavior en masse, they're starting to recognize the scope of the challenge for themselves and for society", says Eric Turkington from Fahrenheit212[6]. Turkington argues that "responsible use" is something that tech companies should actively be discussing, and even advocating.

## Technology is not value neutral

There is a consensus among philosophers that technology isn't value neutral. Until fifty years ago we believed that the user of a product embodied values, not the product itself. Like that old phrase, "guns don't kill people, people do." This is now being regarded as outdated. Products embody values because the use of a product makes it possible to realize values. But a product may also have negatives which violate those morals or ethics.

For example: a car may offer freedom to its driver, but it exhausts harmful gases into the environment. To some extent car drivers are responsible for the use of their cars and the emissions this use causes. But it is not the user,

**Figure 1: The new quality characteristics of intelligent machines (Source: Sogeti).**




i.e., the driver or owner, that has decided the car would have those consequences. He might be aware of the consequences and stop using the car. But it is the designer of the vehicle that has accepted the positives and negatives as morally acceptable in the first place.

> **"Developers must think about to whom their product or service is relevant, how the rights of these people are protected, and what consequences they may face"[7]. (Anna-Mari Rusanen, University of Helsinki)**

To put it bluntly: it is the designer that created the artefact in the first place. And he or she, as a moral agent, should have made a conscious decision whether normal use of his design would be morally acceptable. Moreover, the designer should take all imaginable consequences into account, including the possible misuse of the product or software application, for that matter. "In fact, design is an inherently ethical activity," writes Jet Gispen, designer at A/BZ[8].

## Ethics-by-design, privacy-by-design, and security-by-design

Over the last few decades, we have developed methods to incorporate security and privacy requirements into the application development processes. Basically, the starting point is that the project should take those requirements into account from the very beginning. Security and privacy considerations must be part of the vision or conceptualization of the application.

With ethics, the same way-of-working applies. You should incorporate ethics from the start of the design process. Some experts even say that ethics by design needs to encompass security and privacy, because both are ethical values too. Ethical consideration should also be part of the design of the application. You cannot apply ethics as an afterthought unless you're prepared to shred all the work you have done and start all over again. Ethics is an inherent quality of an application, the attributes of which can be tested like any other feature[9].

## Ethical guidelines are not sufficient

But the question remains, where do we get the quality attributes for the ethics of applications? In the area of Artificial Intelligence, new guidelines, rules and regulations are drafted. These documents describe, in global terms, what features an AI-application must have in order to become ethical. But the rules have some limitations:

1. They describe the requirements in broad terms, only the why and the what. How these rules can be translated into measurable requirements is up to the designer and its environment. At this moment, experts are warning that too loose of an interpretation of the rules does not meet the criteria of being truly ethical.
2. The rules only reflect societal norms and values. Your own company ethics are not considered. So, designers have to also research the culture, purpose, and governance of the organization. All to obtain a full set of ethical requirements for application development.

It is the domain of business ethics that tries to create frameworks for ethics within organizations and businesses. And, in my opinion, ethics for applications will only work when these ethics comply with or are embedded into the company ethics.

> **"Beyond the ethical implications for businesses to become more purpose-driven, consumers are also increasingly evaluating companies based on their environmental, social, and corporate governance factors"[10]. (Rhea Cai, Capgemini)**

## Value sensitive design

Besides the required function of a system (the functional requirements), we also need to consider the quality of how well it performs that function (non-functional requirements), such as performance or ease of use. But there is a third type of requirement that is equally important to success: the moral requirements. Moral requirements are about how the design affects individuals and society: the moral impact.

Value sensitive design (VSD) has been developed to cater for all three kinds of requirements, especially the ethical ones. "Value sensitive design (VSD) is a theoretically grounded approach to the design of technology that accounts for human values in a principled and comprehensive manner. VSD originated to address design issues within the fields by emphasizing the ethical values of direct and indirect stakeholders", as stated by Wikipedia[11].

Value Sensitive Design is one of the methods to systematically investigate the values of the stakeholders, both internally and externally to an organization. But how can we create a shared set of values, when every individual may have different value sets? How do we avoid those conflicts? VSD proposes dialogues to find common values and to build on those.

**"Respecting human values in a design is not always straightforward. It may require creative thinking to find solutions that reduce potential negative impacts on stakeholders"[12]. (Marlies van Steenbergen et al., Sogeti)**

For large projects, like building a dam or a nuclear power plant, the size of the projects allows us to elaborate on values extensively. But for application development, most projects are too small to thoroughly evaluate values to be incorporated into the application from scratch every time. In my opinion, you should have a set of values for application development ready at hand.

By using this method, you can re-evaluate these values for each project according to the specific purpose of the application. It also allows the project team to become familiar with these values, and it becomes habitual to incorporate them into the application design.

## Purpose-led-organizations

So, the question to ask yourself is: how do I obtain a set of shared values within my organization? In 2020 Capgemini started a program of making companies aware of their purpose. This leads to a purpose aware culture, where ethics is an intrinsic part. But within purpose-led companies, ethics are not just a compliance issue, limiting the business possibilities. On the contrary, ethics can create new opportunities. "Creating a purpose-driven culture across all levels of an organization begins with recognizing that social impact and profitability are far from mutually exclusive concepts." writes Rhea Cai, consultant at Capgemini[13].

By being aware of the purpose of your organization, it's societal relevance and ethical position, an ethical framework that makes sense can be derived. And of course, societal norms and values are part of this framework because the organization is also part of society. This ethical framework should inform the guidelines and requirements for application development.

**"We follow our Code of Business Ethics no matter where we operate and whatever the economic circumstances"[14]. (Capgemini)**

Building software applications with ethical requirements and guidelines – next to the functional and non-functional requirements – creates better and more relevant applications for consumers, users, and society. And this is not only necessary from a purpose-led perspective. Capgemini is convinced that people, purpose and profit are intertwined in ways that allow for companies to become significant drivers of positive change, while also driving performance and growth.

Consumers are increasingly evaluating companies based on their environmental, social, and corporate governance factors, meaning the time for stricter ethical guidelines in application development is long overdue.

Building software applications with ethical requirements and guidelines – next to the functional and non-functional requirements – creates better and more relevant applications for consumers, users, and society.

Reinoud Kaasschieter

AI and ECM expert

### About the author

Reinoud Kaasschieter is an AI and ECM expert whose core expertise lies in AI ethics and application design.

reinoud.kaasschieter@capgemini.com

### Footnotes

[1] Coded Bias, an American documentary film directed by Shalini Kantayya. https://www.codedbias.com

[2] Antoaneta Roussi: Resisting the rise of facial recognition; Nature; 18-11-2020; https://www.nature.com/articles/d41586-020-03188-2

[3] Fieke Jansen & Sarah Chander: Why EU needs to be wary that AI will increase racial profiling; EU Observer; 19-4-2021; https://euobserver.com/opinion/151556

[4] Coded Bias on Facebook: https://www.facebook.com/codedbiasmovie/posts/we-are-all-grappling-every-day-with-algorithmic-determinism-we-are-all-being-sco/235819614631873/

[5] AI and the Ethical Conundrum; Capgemini; 2020; https://www.capgemini.com/research/ai-and-the-ethical-conundrum/

[6] Eric Turkington: Should My Phone Have a Surgeon General's Warning?; Fahrenheit212; 2021; https://www.fahrenheit-212.com/boiling-point/should-my-phone-have-a-surgeon-generals-warning

[7] Sandra Theres Dony: The Ethics of Artificial Intelligence: Helsinki University; Techstory, 04-04-2021; https://techstory.in/the-ethics-of-artificial-intelligence-helsinki-university/

[8] Jet Gispen: Ethics for designers; 2017; https://www.ethicsfordesigners.com

[9] Saunders, Paul: Testing of artificial intelligence – AI Quality Engineering Skills; Sogeti; 27-02-2020; https://www.sogeti.com/explore/blog/testing-of-artificial-intelligence/

[10] Rhea Cai: Why purpose, profit and people are now inseparable; Capgemini; 18-10-2020; https://www.capgemini.com/2020/10/why-purpose-profit-and-people-are-now-inseparable/

[11] Value sensitive design; Wikipedia; last edited 23-03-2021; https://en.wikipedia.org/wiki/Value_sensitive_design

[12] Steenbergen, Marlies van, et al.: Value sensitive architecture; Sogeti, 02-09-2020; https://labs.sogeti.com/whitepaper-value-sensitive-architecture/

# THE CAPGEMINI APPLICATIONS UNLEASHED THOUGHT LEADERSHIP TEAM 2021

## Foreword
Ronald van den Berg

## Introduction
Ron Tolido

## Editor in Chief
Anya Buchan

## Project management and marketing team
Rik Pennartz, Aditi Basu, Mohammad Salim Zia

## Design & Graphics
Johanna Achterberg, Sujoy Karmakar & Ashim Karmakar

## Thought Leaders
Anand Gupta
Alex Bulat
Arjan de Bruijn
Claire Knoester
Fina Piazza
Jade Eloff
Maarten Stam
Marien Krouwel
Mark Oost
Matthijs Greep
Milé Buurmeijer
Rahul Khandelwal
Reinoud Kaasschieter
Rich Minns
Rik Pennartz
Robert Kooloos
Ruud de Wit
Saket Kulkarni
Sanne Kuijpers
Sanne Visser
Sjoukje Zaal
Stijn Follet
Wybe Horsman

## With special thanks to
Esmee van de Giessen
Rens de Leeuw
Roger Wannee
Sander Zijlstra

# About Capgemini

Capgemini is a global leader in partnering with companies to transform and manage their business by harnessing the power of technology. The Group is guided everyday by its purpose of unleashing human energy through technology for an inclusive and sustainable future. It is a responsible and diverse organization of 290,000 team members in nearly 50 countries. With its strong 50 year heritage and deep industry expertise, Capgemini is trusted by its clients to address the entire breadth of their business needs, from strategy and design to operations, fueled by the fast evolving and innovative world of cloud, data, AI, connectivity, software, digital engineering and platforms. The Group reported in 2020 global revenues of €16 billion.

**GET THE FUTURE YOU WANT | www.capgemini.nl**

## For more details contact:

**Capgemini Nederland B.V.**
P.O. Box 2575, 3500 GN Utrecht
Tel. + 31 30 689 00 00
***www.capgemini.nl***

IN/1B-038.21