

WHITE PAPER

What Happened to Digital Transformation in the Telecommunications Industry?

Jag Siva



waylay

TABLE OF CONTENTS

Introduction	2
Findings	2
Managing Scope	3
Managing Change	3
Leverage Your Legacy	4
Keep Your Experts & Customers Close	4
Build for Purpose	5
Embrace Failure	5
Waylay & Transformation	6
Summary	7

INTRODUCTION

What happened to digital transformation in the telecommunications industry? By their own account, this has been a strategic imperative for the past 30 years. But what has changed? Long hold times with call centers, extended outages, inflexible networks and poor customer experience are common complaints across companies, regardless of geography. Internet giants, data center players and SD WAN providers are all encroaching on both the traditional as well as the OTT areas that seemed so lucrative for carriers just a few years ago.

FINDINGS

Through the decades, we have seen the urgency and need for transformation. Transformation to migrate legacy data and systems. Transformation to support new technologies. Transformation to reduce costs and improve operations. Transformation to introduce operational agility, better customer experience and so on. But what has happened to all these initiatives? Why did they not materialize as intended or became a never-ending journey? It certainly is not from the lack of effort or desire.

In recent years, we are again seeing increased focus and attention around digital transformation and automation. Again, while truly successful examples have been implemented, they are rare. While these types of initiatives appear to be in the DNA of the internet giants, carriers' ambitions are hindered by their own massive inertia from within and from outside their organizations. The reality is telecom services providers have been extremely successful companies for many years. As such, it is very hard to look outside for a different operating and technology innovation model.

My experience in the industry over the past 25+ years points to an approach that deviates from the current thinking that is no longer serving the industry well. Everybody agrees on the needed outcomes, but are we aligned on the approach?

We all have our complaints with the inflexibility and response times of our local carriers. Historically, this was the price we paid for a robust, universally accessible service that was always delivered at a consistent and predictable level. The limited number of services carriers provided, while not offering the choices of bells and whistles we see now, were mostly always available and usually worked as we expected. This was no accident. It was the result of an engineering led approach to solving problems. The customer's concerns and values were considered but exacting performance was prized over everything else.

This, however, is no longer suitable for the challenges faced by our carriers. We now interact with our service providers for a myriad of services, some critical, others not. We are also offered options by players in adjacent spaces—faster, cheaper and more flexible. In some ways, the comparison is unfair. Would I expect the same quality of service availability from one of the web giants that I would expect from my incumbent carrier? No. If a dongle hanging off of my TV streaming my favorite show was to fail, I wouldn't even know what number to call. At best, I would check my Wi-Fi connection and reboot the device.

MANAGING SCOPE

Traditional telco systems, such as inventory, billing, order management and fulfillment, are massive in scope and complexity. Replacing these systems present insurmountable cost and business continuity risks. The multitude of so-called modernization projects undertaken in the 2000's and 2010's, largely led by systems integrators, have created a new era of legacy, and mostly left the limitations of the original systems in the hands of the carriers to deal with. But this wasn't all. These initiatives also introduced another layer of complexity with enterprise product catalogs and with bespoke orchestration and workflow systems that were bolted on to adequately interwork the numerous silos that were present in the OSS/BSS infrastructure.

With this tenuous past, it does not take a giant leap to understand the current situation. Where do you start? The destination is clear, but the path to this future is far more challenging to achieve.

Scope creep is our enemy. The temptation is always to modernize from the ground (current systems) up. But does this deliver real business value and/or improved customer experience? Migrating data and capability from dozens of systems that are often decades old is a complex and expensive undertaking. It is difficult to build a business case to support this, let alone assume the business continuity risk that it inherently brings.

“
Traditional telco systems are massive in scope and complexity
and replacing them present insurmountable cost and business continuity risks.
”

One approach is to modernize from the top (customer) down, not from the bottom (network) up. Prioritize capabilities that touch customers and employees. If data resides in a legacy system and is critical to the process or function being modernized/automated, consider encapsulating and federating that system for your purpose. The concepts of encapsulation are simple enough; create a wrapper around the legacy that exposes its useful data and processes while leaving the system largely intact.

Always measure scope inclusion against business value and customer experience. Most importantly, define scope in manageable programs of work—chunks that have enough to demonstrate business value, but are still reasonable to operationalize in months, not years

MANAGING CHANGE

As they say, change is the one thing that is certain. Even more certain than change is that the rate of change will only increase over time. If one is able to contain scope to manageable chunks, the duration of the project then faces less change requests or scope changes. By delivering capability in smaller chunks, you can ensure that your system will have a better chance of not being obsolete before it's even deployed.

Once your system is in production, there will still be more changes, that is simply the nature of business operations. The best approach for this is a platform and architecture, along with internal processes that

allow and embrace change. Change, whether fixes for new functionality, should be the new business as usual. Change should not be an exception, but rather a way of life.

Is your transformation approach and underlying technology able to support this? A system that is capable of embracing a high change velocity model is imperative to the transformation program being relevant on day one and into the future.

LEVERAGE YOUR LEGACY

Far too often, we see carriers attempting to modernize legacy platforms. It is an expensive and near impossible proposition to migrate the millions, and in some cases 100 of millions of records that are currently held in these systems. These systems will go away, but that day is not today. They service legacy networks that are still critical to today's infrastructure. Eventually, that infrastructure will be replaced, and these systems will die on the vine. In the meantime, why not encapsulate and federate this data and make it accessible to your modernized operational and business systems? The data needed from these systems will be still accessible and your modernized and automated process will have full use of them without disrupting the legacy services.

KEEP YOUR EXPERTS & CUSTOMERS CLOSE

Large transformation programs are prone to "translation errors." Months are spent gathering requirements followed by a lot of heavy lifting in development and testing. Typical projects are measured in years, not months. If a production system sees the light of day, a few years later, requirements will have changed, the underlying networking technology, at a minimum, will have been extended, and the system will likely be less valuable than planned at the outset.

“**Large transformation programs are prone to errors.** Months are spent gathering requirements. Years later, they will have changed.”

Let us remember that most of these large-scale programs are led by the systems integrators (SI). While the SI brings strong competencies in the IT and programmatic aspects of the projects and the carriers' needs, operational requirements are still derived from the people doing those jobs day-to-day. Requirements are gathered from hundreds of interviews, documents, and observations of the various activities. These are then compiled into large requirements documents, followed by several design phases. This is the epitome of waterfall, big bang, methodology.

Finally, the development starts followed by multiple phases of infrastructure builds and testing. During this arduous process, often involving several people from multiple organizations, both details and business objectives fall through the cracks, requirements get missed or misunderstood, and when the time finally comes for a UAT, the system is not quite what the

customer's domain expert outlined. Often, the original domain expert or user that shaped the requirements may have moved on to a new role or even no longer be with the company.

How do we avoid this? When we view this in conjunction with the rate of change, there's one approach that rises to the top. We need to push capability toward our users and customers. This means we thin out the bespoke development required and offer our domain experts the ability to implement business rules and feature changes in flight, right at their fingertips. Similarly, provide customers with a limited capability to change features and rules that are relevant to them on their own.

While this may not work for all scenarios, it can certainly reduce the number and frequency of change requests to a production system. It will also help avoid the "everything and the kitchen sink" approach to scoping, because there is a much simpler and a more effective way of introducing change into a production system.

BUILD FOR PURPOSE

With today's modern application platforms, we have the luxury of leveraging all kinds of horizontal, meaning multi-capable and highly configurable, platforms to provide building blocks across organizations and systems. Gone are the days when we built siloed monoliths. Plan, design and architect for flexibility. It is far better to build applications that are function specific from the perspective of the customer, not bottom up from the perspective of the network.



It is far better to build applications that are function specific from the perspective of the customer, not bottom up from the perspective of the network.

EMBRACE FAILURE

When all we had was PSTN and some basic data networks, it may have been possible to build robust enough networks and systems that "never" failed. We are in quite a different place now. Software, networks, and service complexities are multiple orders of magnitude greater.

What we can be certain of is that failures will occur. Rather than extract that last bit of robustness out of the system or network at the cost of diminishing returns and elaborate complexity, why not embrace the failure—build resilient systems, failover and load-balancing capabilities, and the myriad of technologies that are now available to us?

At some point the measure of "robustness" becomes a philosophical one and loses perspective in the context of business value and customer experience.

WAYLAY & TRANSFORMATION

When I first joined Waylay, I had a lot of time to think about what telecom transformation would mean in the context of Waylay. Waylay is a multi-industry automation platform that services customers in fintech, manufacturing, logistics, smart buildings, oil & gas, and of course, telecommunications.

The first thing that struck me is that automation is a journey, one of building capability, monitoring and measuring how that capability is performing and, eventually, taking the training wheels off and allowing it to become autonomous. You then may extend and iterate as you introduce new processes and capabilities onto the platform. This is what I call biting off small chunks of manageable scope.

Data comes from a multitude of sources, some modern, some legacy, some with proper APIs and others without. Ingested data can come from modern APIs or from encapsulated or federated systems—no need to modernize the legacy infrastructure. Data may be used transactionally to support the automation process but does not need to be persisted and duplicated within the automation platform, because the automation platform can talk to the source of truth without the need to become one. Transactional and time-series data can be persisted if desired depending on the nature of the automation and service itself. Push data, pull data or streaming data, whatever the data need is, there's a flexible and efficient way to access and use it.

Scope can be defined as entire systems or individual processes being automated. Building rules and relationships happen within a low-code, citizen developer context. Although this may appear to be chaotic at a first glance, best practice is to partition the user communities and personas of these systems and empower them at the right subject matter expert level to do what they need to do.

“
Automation is a journey, one of building capability, monitoring and measuring how that capability is performing and allowing it to become autonomous.
”

Create multiple, fit for purpose applications that use the same data and resource infrastructure. Run it on your own cloud or a third-party cloud and scale up or down as your needs demand.

This type of approach lends itself to looking at automation and transformation as incremental improvements to the working process you perform—not a wholesale replacement of an old system to a new system.

I earlier said that transformation, modernization, automation, and etc., appear to be in the DNA of the internet giants. But we also rarely see them announce a big transformation program. It is happening within these organizations all the time. It is in their DNA. Their approach is not to spend years building a system and then use it as is for another number of years. Theirs is an evolutionary approach that embraces change and builds it into their daily lives and operations. Carriers can do the same thing by changing their perspective and by incorporating software platforms that are built for change.

SUMMARY

Carriers need to rethink their approach to transformation and automation in a number of key areas. First, scope. Don't boil the ocean, bite off manageable chunks of scope that deliver business value in phases. Second, change velocity. Customer needs have changed much faster in 2022 when compared to decades ago. Third, modernizing legacy. This is a massive undertaking. So, use it as is, it will sunset on its own when it is no longer of value. Next, complexity. Move decisions and execution closer to your experts and stop trying to build systems of ONE that support all networks and all customers. We need to rethink this and build systems and networks that are "fit for purpose." Finally, accept that it is impossible to build something that never fails, but rather, accept that it WILL fail and be prepared to deal with such failures.



About the Author

Jag Silva is EVP of Global Sales at Waylay with more than 25 years of global software sales and services experience in a wide range of organizations, from startups to fortune 500s. Jag has an established background in strategy, sales, services, and go-to-market execution and a strong record in building and managing P&Ls of all sizes, from 50M to 1B, with an emphasis on driving high-quality revenue streams and long-term customer success.



Blekersdijk 33a, 9000 Gent, Belgium
+32 9 311 55 66 | www.waylay.io
info@waylay.io



© Copyright 2022 Waylay. All rights reserved.
Waylay is a registered trademark of Waylay NV.