DATA DUCTUS

White Paper

The key to a successful automation project

Learnings from 40+ completed network automation projects

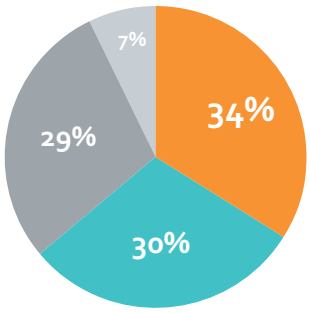
Contents

Introduction

Within software-defined networking (SDN) and Network Function Virtualization (NFV), automation and orchestration are crucial in the cultural shift faced by most communication service providers as they target improved profitability and service agility.

The various work processes of a typical communication service provider are complex and intertwined, with many dependencies, as are the IT systems supporting these processes. This can make it difficult to start an efficient transformation journey. It also explains, in part, why many automation initiatives fail or only deliver a fragment of their promise.

According to a study made by EY in 2016, only 30% of automation efforts result in improved efficiency, while another 29% did not deliver due to parallel manual processes still in use, and a further 34% failed completely.



- Prior investments in automation have not proven to be successful.
- We see an increase in productivity due to more automated processes.
- We are still duplicating automatedand non-automated processes.
- We are not sure.

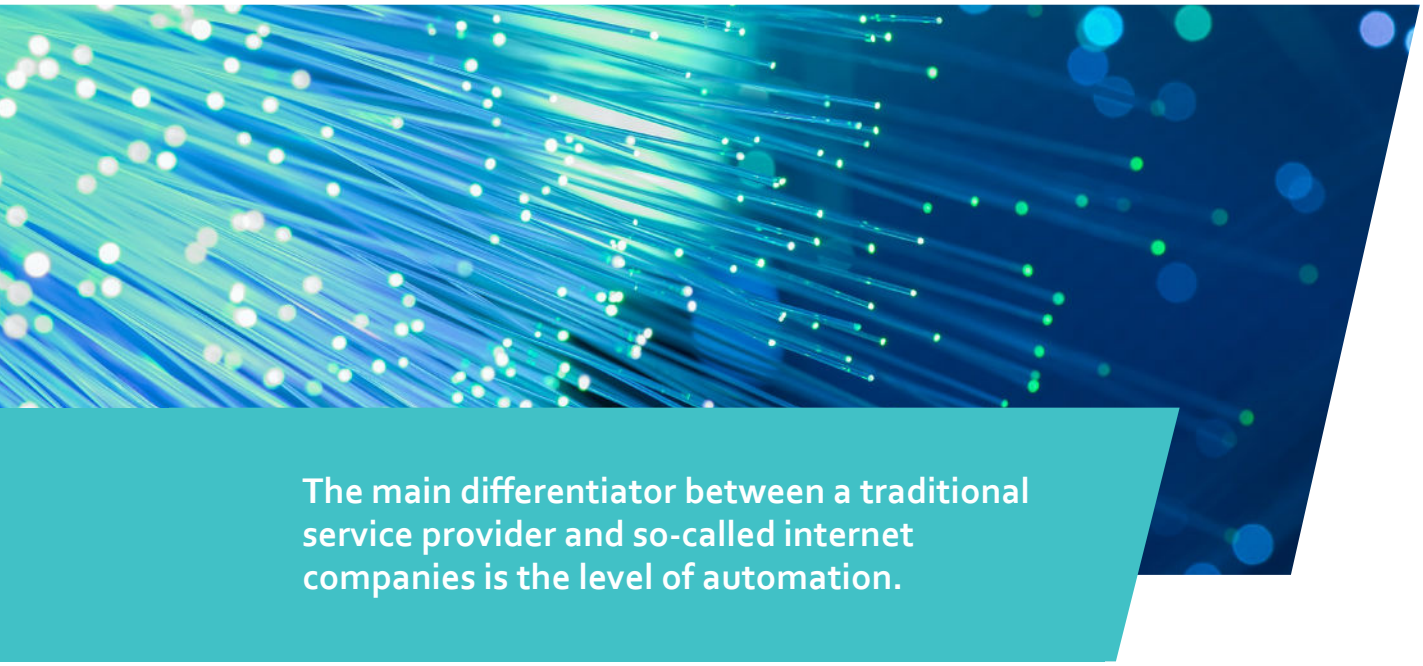
Source: Global Capital Confidence Barometer, 15th edition, EYGM Limited, 2016.

This whitepaper presents the key factors for a successful automation project. It's based on learnings from over 40 automation projects conducted with Tier-1 and 2 communication service providers and large enterprises around the world.

What is network automation?

When virtualization, fuelled by the cloud trend and emerging SDN and NFV technologies, began to be talked about in the telecom industry, hopes and expectations for what could be achieved in the future were high. The key enabler was that these technologies seemed to provide a programmatic approach through an API that in turn enabled automation.

The main differentiator between a traditional service provider and so-called internet companies is the level of automation. One should also note that SDN and NFV are by no means required for network automation. On the contrary, most of the automation developed today is done without NFV or SDN.



Service providers face challenges in sustaining or growing their revenues, therefore they are forced to identify costs savings. Simply put, this consists of reducing Capex and Opex. NFV and SDN have potential merits for both, and automation can multiply any benefits.

Calculating the actual Opex is often complicated; there are multiple bundled costs, ranging from power consumption to personnel. Contributions to Opex from automation aim at:

1. Reducing the need for network management personnel, thereby combating management costs and the need to manage an increasingly complicated network

2. Improving network utilization / performance by adapting the network to perform accurately and optimally
3. Reducing the time by which new services / customers can be served by the network
4. Reducing costs related to low quality, including the assurance of service KPIs

The above imply that the scope of network automation spans the entire life-cycle of a network: from its design and planning, through installation and commissioning, to the management needed to onboard new customers / services, as well as continuous quality assurance of networks and services including consequent optimization.

Challenges of network automation

Within a network there are multiple elements that are geographically distributed and interconnected. Such complexity – where one network element may support multiple (potentially all) services, and one service operates across elements of the network – cannot be underestimated.

Accordingly, automation (of processes related to network management needs) has to be able to deal with the implications of the services currently operated by the network as well as those that are in a catalog ready to be provisioned, should an instance of the service be ordered. Although this mapping is challenging, it is now increasingly possible to automate it through emerging orchestrator technologies that provide a means to increase the level of abstraction so that it enables focus to be put on the intended services.

Furthermore, the nature of network automation is inherently a programming challenge. The reason being that a "do what I mean" approach is not feasible. The industry has to provide solutions whereby service providers and enterprises can express what their intentions are both at network and service level.

At Data Ductus we refer to the development of automation as a programming exercise irrespective of whether it is a declarative or imperative approach, no matter what information the service provider needs to define or whichever languages are used. The key point is that any network automation solution in run-time needs input that specifies the intended behavior. It also implies that a design-time and run-time view of network automation emerges. These two views both need to be addressed when considering network automation.

Automating service deployment can cause a dilemma if not done properly. Deploying services manually has the benefit of unlimited flexibility. Smart engineers can principally configure services to meet any customer requirement. However, that way of working is not feasible. Service deployment projects take too long and introduce too many errors. Furthermore, the operational cost is too high since you need more staff to cope with the demand.

Accordingly, many service providers and enterprises automate service delivery. Ironically, this type of automation is often counterproductive. Service providers often claim it leads to a culture of "it works so don't touch it". We refer to this as the Catch 22 of Service Orchestration.

A typical example would be when a part of the catalog is automated with a "hard-coded" solution that requires a long and costly software project. Once complete the solution fulfills the goals of fast and error-free configuration, but it does not offer the flexibility to adapt the portfolio to meet new customer and market needs. Another long and costly software project is therefore required to resolve this. Rather than going ahead with the second project there is a risk that the organization in questions falls back to manual configuration.

The key factors for a successful automation project

#1: Abstraction is essential for success

Although this may seem obvious, it is still critical. If done effectively it can dramatically reduces costs for providers, speed up customer services, and enable labor resources to be focused elsewhere in the business.

When an automation solution (software) operates on network primitives, such primitives need to be well defined. The issue is that most network elements (devices) have very large and complex management data models. Furthermore, data models and APIs differ between equipment vendors.

A key mistake made in projects is to make abstraction equally vague. Automation is about enabling program control, the abstractions need to be defined, precise and very concrete.

So the subset of information from devices that are of interest for automation depends on:

1. The services that are going to be provided on the network (if the scope of automation is service provisioning).

2. The network topology or design (which may or may not utilize certain information).
3. The processes that are going to be automated (and their relation to processes that remain manual).

Points 1, 2 and 3 will give different conclusions depending on your time horizon. Enterprises or service providers trying to look ten years ahead will yield somewhat vague conclusions. It is more feasible to restrict the time horizon and automate processes and activities that are often used and thereby represent considerable improvement potential.

We would advocate an iterative approach, in which abstractions (at all layers) are defined for the challenges at hand, and thereby gain speed in rapid improvement steps.

Customers always want their abstraction models to be reasonably future-proof, so they can be extendable. This applies both to any network models as well as how programming on top of the network is carried out.

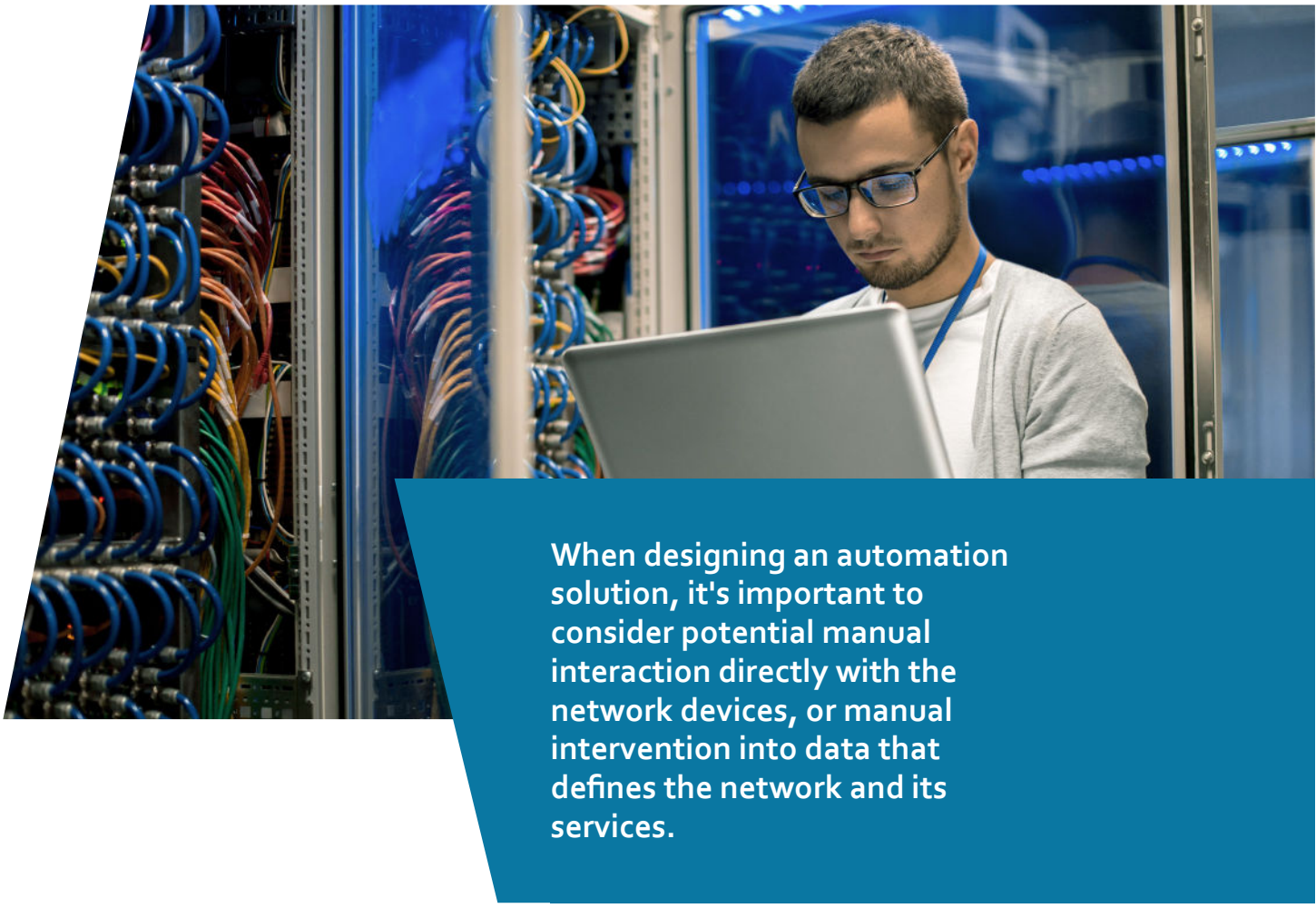
The key to successful abstraction is experience. Having been involved in many such projects, we see patterns and know how to avoid common pitfalls. More importantly, we can identify scenarios and requirements that are likely to occur and create solutions based on specific domain needs.

A DATA DUCTUS PROJECT IN ACTION:

One Tier-1 service provider in Asia is running an automation program in which Data Ductus is heavily engaged. The activities are divided into iterations, each of which targets a specific set of services. Each iteration adds new services, which require the underlying network abstraction to be gradually extended. This agile process dramatically shortens time-to-market for new services.

#2: Automation requires robustness

Anybody that has developed telecom software knows that the ideal (error-free) case is rather straightforward, and real efforts related to achieving operational software lie in the amount of scenarios in which all the various errors from non-expected events need to be handled.



When designing an automation solution, it is important to consider potential manual interaction directly with the network devices, or manual intervention into data that defines the network and its services. Typically, the automation solution deals with a subset of network device data. That subset needs to be protected from intervention, and in case we cannot prohibit field personnel from altering data at site, the solution needs to automatically restore the intended data.

The effort needed to develop an automation solution is dependent on how such scenarios can be addressed. Note that we are not arguing that all such scenarios

need to be fully automated. The important lesson is that the scenarios need to be considered, and proper procedures must be defined, e.g., should they be manual or included in the automation solution.

Central in dealing with this is to define where the master data resides (and whether it is controlled by automation).

An automation solution has to deal with the potential scenario of faults in a network device, the repair of them and going live again. A solution might also have to deal with shortages in resources, or intermittent disturbances to network performance due to various issues (including resource competition).

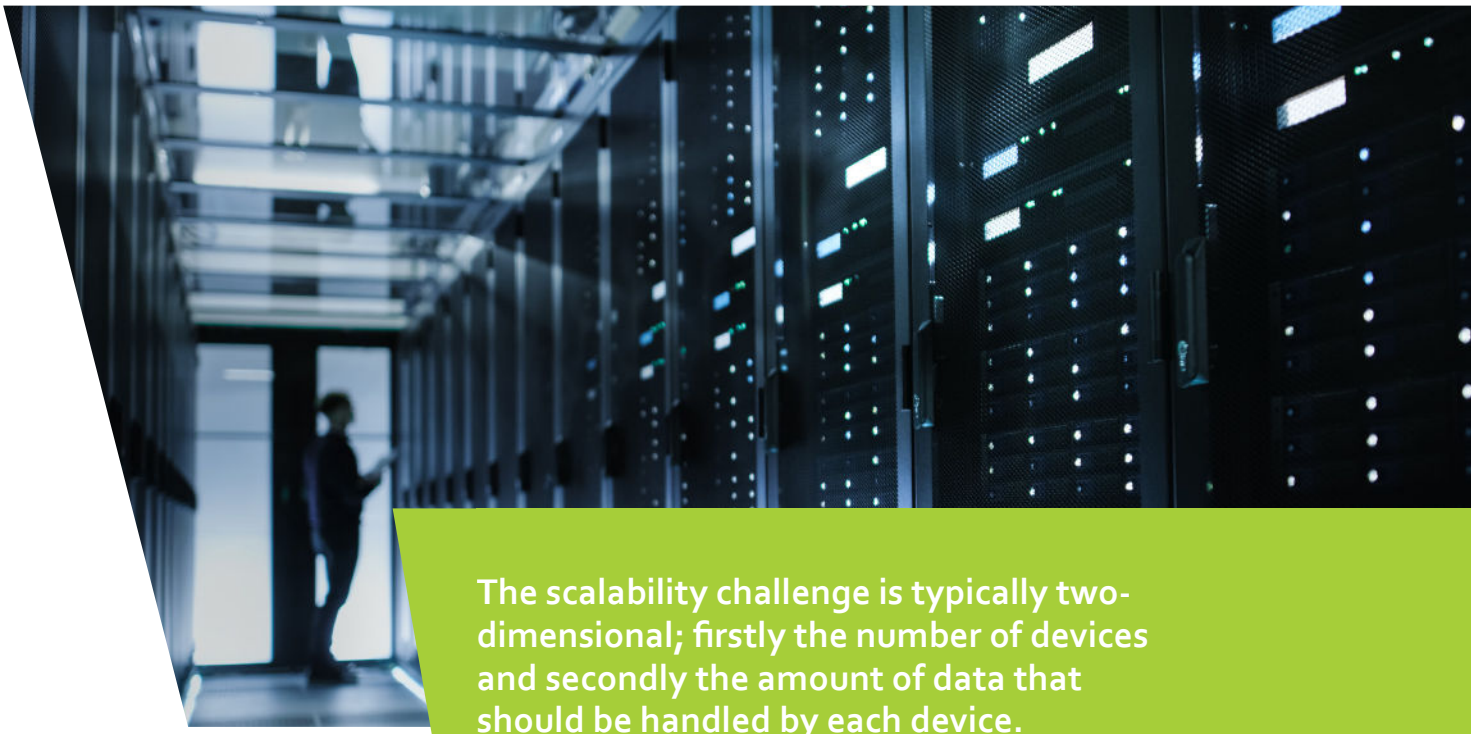
Automation solutions should further support rollback operations. This is reasonably straightforward for network devices that support transactions based on NetConf, or similar. But for devices with traditional CLI interfaces this may quickly become problematic.

A DATA DUCTUS PROJECT IN ACTION:

A national telecom provider uses the orchestrator to periodically synchronize the actual configuration data of their services with the expected configuration data to find out if any out-of-band configuration has been done that affects their services and if so re-deploy the affected service. This has significantly decreased service downtime for the operator.

#3: Scalability requires attention up front

Scalability in this context captures the solution's ability to scale up to a very large number of devices (i.e. to automate the management of a large network). Many service providers' networks are global, and thus include a huge number of network elements (devices). Networks may also be gradually densified, leading to a growth in the number of devices. Furthermore, as networks are modernized, the scope for the automation solution typically grows.



The scalability challenge is typically two-dimensional; firstly the number of devices and secondly the amount of data that should be handled by each device.

Although it may be a good starting point to think about a smaller fraction of your network to get started, this may lead to a solution that does not scale properly. Automation solutions are basically software, and as with all software solutions, their architecture needs to be properly considered up front.

The scalability challenge is typically two-dimensional; firstly, the number of devices and secondly the amount of data (config, performance, etc.) that should

be handled by each device. Therefore, a solution may be perfectly fine for a large network and service, but it may fail when another service type is added.

One approach that Data Ductus has applied for high demanding automation solutions is to cluster a number of orchestrators together that share a common distributed data set. Such a solution brings new challenges related to data distribution, locking of data handling, transaction mechanisms, etc. These solutions work fine, but they represent a different level of investment that should be carefully analyzed and justified before going ahead with such a project.

Once orchestrators are clustered, they can also provide a higher availability and resiliency for single orchestrator faults.

A DATA DUCTUS PROJECT IN ACTION:

One EU headquartered provider had a global network footprint that could not be supported by a single orchestrator solution. We developed a clustered solution with shared data that is capable of configuring a network, creating required service levels and monitoring that the services provided meet agreed SLAs. In-built scalability enables the system to grow with the provider's offering (handling 100,000's of end points) and individual client's needs.

#4: Get your software culture right

Automation is software. Automation is related to your business. Your business is supposed to be agile (at least it should be soon). This implies that the teams responsible for your automation need to perform best-in-class.

Software teams should be an integral part of an enterprise or service provider as they are business critical and essential to a successful DevOps approach.

Not only do the teams need to understand basic software and its development, they also need to understand high-availability and high performance (telco grade) software that needs to be developed according to the highest quality standards. Management has to have clear lines of responsibility, be devoted to targets and have extensive domain knowledge on the topics to be automated

It's no secret that there are order-of-magnitude differences among developers, so, basing resource decisions on employees' hourly rates is bad practice. We believe that software teams should be an integral part of an enterprise or service

provider as they are business critical and are essential to a successful DevOps approach. Such a setup, of course, includes setting up the appropriate DevOps tool-chains.

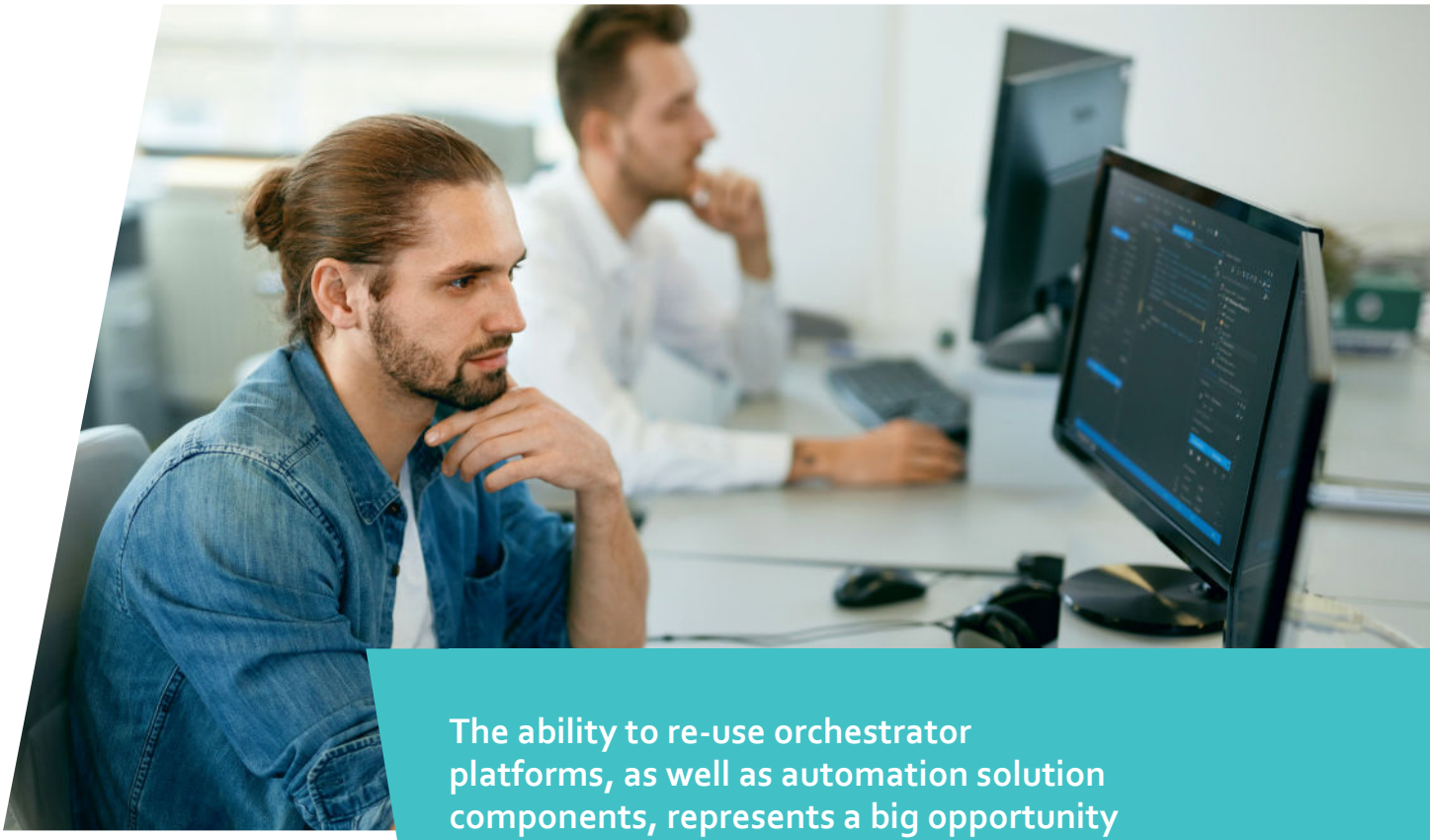
When building up such teams it is a good idea to learn from best practices. At Data Ductus, we have extensive experience from helping organizations transform their ways of working into a DevOps operation. By beginning small, identifying the ideal automation projects and working iteratively, results-based change can be implemented, even in risk-averse organizations.

A DATA DUCTUS PROJECT IN ACTION:

We have helped transition several hardware-focused clients into working in a more software-driven way. This has meant developing methods for high quality agile software projects in hardware environments and introducing automated test routines and iterative development.

#5: Reduce development time with reusable sofware

There is no "silver bullet" when it comes to automation. At the end of the day it is about making or using software. The total amount of software needed is often substantial.



The ability to re-use orchestrator platforms, as well as automation solution components, represents a big opportunity that makes some automation projects feasible.

This implies that there is an inherent software development, maintenance and quality problem present, as for any software. The ability to re-use orchestrator platforms, as well as automation solution components, represents a big opportunity that makes certain automation projects feasible that would otherwise have been too large to undertake.

The emergence of orchestrator platforms has become more and more common in recent years. It represents a huge enabler for automation projects. Re-using software is however not limited to a pure platform approach, it can also be utilized at the solution layer. Making software re-usable normally comes at a

cost. If such software could be re-used at a later stage, the cost is well worth the investment, otherwise not.

Where possible, we utilize pre-packaged software components from our software library. These solutions typically represent various frameworks upon which the customer solution can be tailored. This approach helps us reduce project time and costs while improving the quality of the final solution. In fact, most projects are realized at least 25% quicker using this approach.

A DATA DUCTUS PROJECT IN ACTION:

In three consecutive projects for different clients, we reduced the development costs by 25% by building on our pre-packaged software components as well as increasing the levels of test automation.

About Data Ductus

Data Ductus is an independent IT consultancy company with operations in Europe, US and Asia through its 11 offices. Our business spans telecom automation, IoT as well as IT support and service management.

We provide game-changing solutions to telecom service providers and enterprises, including network and service automation and abstraction that scales. We address the specific needs of verticals, and can provide 24/7 support of our solutions to customers.

These solutions are built on a strong portfolio of 40+ telecom automation projects for tier-1 and tier-2 service providers and large enterprises located across the globe.

As a leading telecom consultant and partner to leading service providers, we can offer consultancy on automation solutions spanning from requirements collection and architecture definition over solution development, to solution delivery and support. When doing this we always use the available experience and software assets that we have gained throughout our projects.

Read more about our network orchestration and automation services
at our website:
https://www.dataductus.com/network-orchestration-and-automation/

or contact us for further discussions

+46 (910) 73 23 00 | info@dataductus.se | www.dataductus.com

DATA DUCTUS