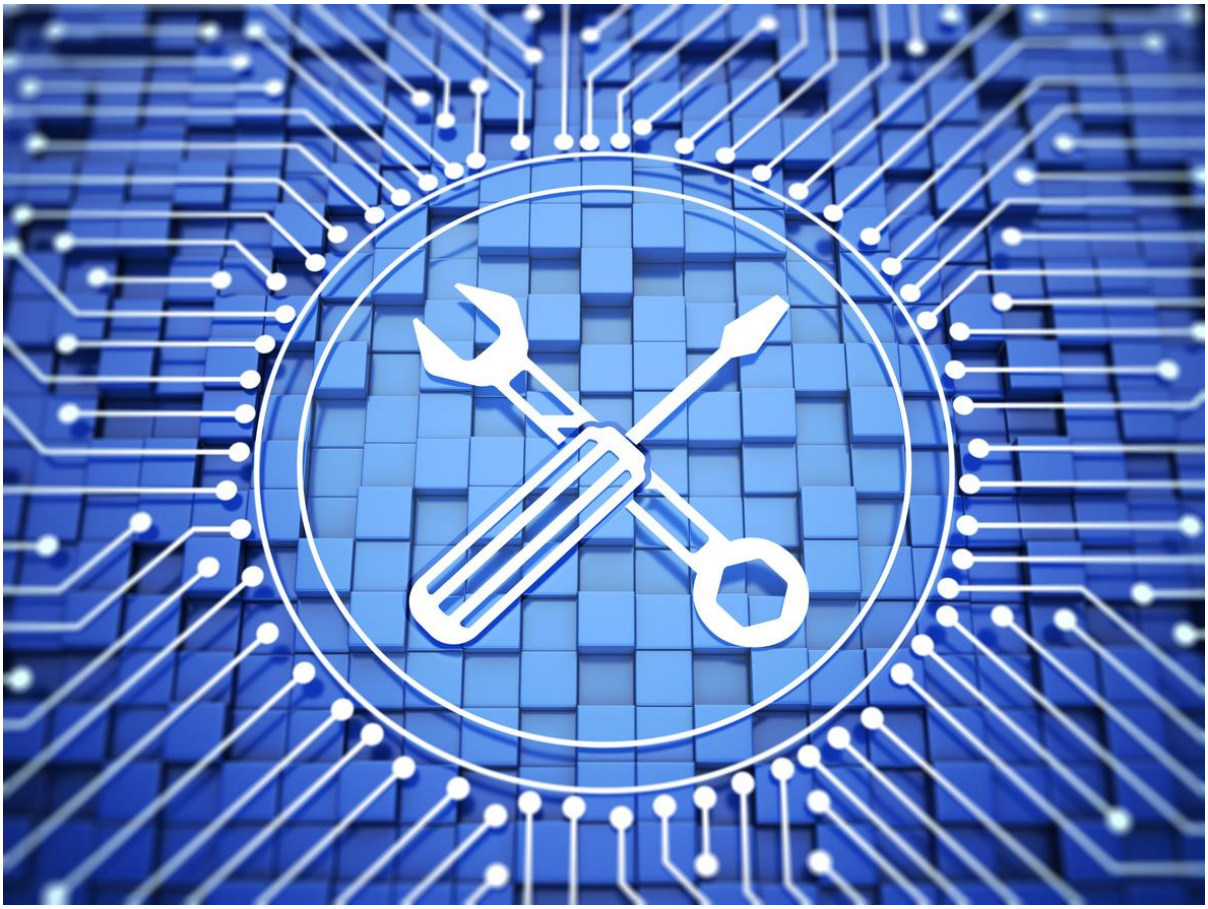
Telco2.0™

Executive Briefing

How to build an open source telco – and why?

How far down the route of using open source software can/should telcos go, and how should they do it?



Roberto Minerva - Associate
roberto.minerva@stlpartners.com

JULY 2017

STL PARTNERS

Executive Summary

In theory, the concept of building a telco based on 'free' open source software is highly attractive to telcos. This is particularly so for operators in low ARPU markets where cost reduction is a critical imperative, and for those wishing to gain more control over their ability to innovate and differentiate.

In practice, the concept comes with three catches:

1. To do it, your organisation needs to possess the skills, organisation and culture to manage the software development and implementation – and most telcos don't have these attributes today.

2. Some other costs, such as those to deploy the open source software, are likely to be higher than in 'traditional' vendor deployed instances, so the cost economics depend on the overall balance achievable.

3. It isn't an 'overnight' solution, so it is likely take years rather than months to get all the necessary pieces in place to successfully migrate to an open source solution.

This report outlines how a telco could go about achieving this transformation in practice, while managing the risks in both in new build ('greenfield') and change-out ('brownfield') contexts. In all cases, it centres on the alignment of the management team on the best approach, and building a team of people with the right skills in the right places.

'Softwarisation' – a trend that can unlock new opportunities

'Softwarisation' - the redefinition of products and services though software - is permeating the telecoms industry, opening opportunities for telcos to create new products and services, and move into new markets, as well as cut costs. Softwarisation is also disrupting long established relationships between vendors and telcos and between telcos and web companies. The traditional approach of relying on the vendor to introduce innovation into the network cannot fully exploit the flexibility that modern software technologies can bring.

Telcos now need to explore new paths that enable them to both reduce their investments in software and increase their flexibility and agility to introduce new functions and services in the network. Both commercial pressures and new technological opportunities make it imperative for telcos to review, and potentially revamp, their software strategy. Deutsche Telekom and NTT, as well as operators in the Americas, have deployed open source solutions with some success. In Latin America, for example, one operator[1] employed a relatively small group of developers to introduce e-commerce and payment-related functionalities into its network using open source packages, after finding vendors' solutions were too cumbersome and expensive.

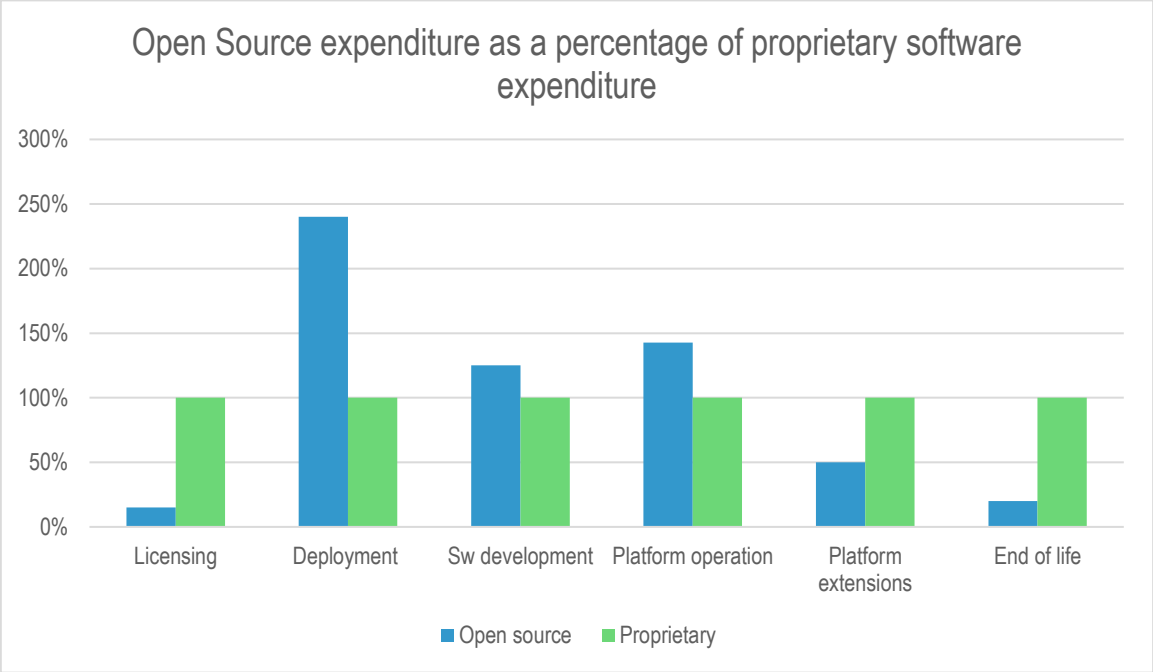
Open Source – a means to achieve softwarisation

In this context, the adoption of open source solutions can be very appealing for many telcos. The most obvious benefit seems to be a drastic reduction in licensing fees, but this positive impact can be offset by additional costs elsewhere in the software development lifecycle (see graph, which estimates the cost differences in percentage terms). In any case, if the telco doesn't master software, any decrease in costs

[1] The operator provided the information on the basis of anonymity

could be short-lived. In fact, an operator that is not able to deal with software will have to pay substantial fees for professional services and support.

Figure 1: Illustrative open source costs versus a proprietary approach



Source: STL Partners estimates (see Weighing the cost of open source, page 23, for details)

Beyond immediate cost considerations, the most compelling reason to adopt open source solutions is to drive a different approach to the evolution of the network infrastructure. A shift to open source could enable a telco to address new market needs and start a digital transformation of the entire company.

To fully exploit the benefits of open source, it is important to have a digital transformation strategy that can solve some cogent problems of the actual network infrastructure. In cases where the telco needs to technologically renew the infrastructure or to replace old products, open source solutions can help make this transformation more effective. The telco has to decide where to start from and the size of risks that are acceptable.

Deciding how: three implementation options

There are several steps telcos need to become an open source telco (please see Choosing the right path to open source, page 18). The first major step is to choose one of three different options telcos can employ to harness open source solutions:

1. **A tactical open source** option - the telco opportunistically and progressively substitutes obsolete components with new ones that are open source.

2. **A full open source** option - this is the path followed by a telco that has strategically decided to transform its infrastructure by introducing new technologies (e.g., SDN, NFV supported by new virtualised cloud and edge infrastructure) mainly based on open source solutions.

3. **A migrating open source** option; the path followed by a telco that is putting in place an open source based strategy and related infrastructure and is migrating existing functionalities to a new virtualised cloud and edge platform. Not all the functions and components are open source yet, but there is a long-term strategy to substitute existing components and functionalities with new open source components. This operator could introduce new technologies (e.g. 5G-related solutions) based on the open source approach and integrate them into the existing infrastructure.

A second major step is to decide if the telco wants to have full control of the software development or to also use external competences and skills. To be successful, a high degree of control over the software development has to remain in the hands of the operator otherwise they will continue to be locked into vendor solutions. Outsourcing is still possible and in many cases, practical, but only if the telco maintains in-depth knowledge of the software infrastructure within the company.

To make it happen: you need people with the right skills

To become an open source telco, operators need to begin by acquiring considerable expertise and skills in software development; then practise open source development habits, adopt best practices and tools; and finally mix and match the components for building a viable solution. Once a group of professional developers has been created, they should be progressively exposed to larger experimentation and real-world deployments (see How can you build an open source telco?, page 28).

Another issue to be tackled is the company's software culture. In many telco organisations, software is just considered a means to have the job done and not as an enabler for providing new solutions and a way to gain a competitive advantage. Developers are usually neglected and are not considered a valuable and strategic part of the company (in stark contrast to the attitude of the big web companies). Telcos can be process-orientated, elevating the importance of project managers and coordinators. Although these teams ensure that innovation stemming from large external vendors are properly integrated in the network infrastructure, this approach is not going to work with open source. Internal software developers need to be empowered and able to deploy their solutions within the network: the strategy should be to prioritise internal product development over solutions bought from external vendors.

Managing the risks

As the extensive use of open source software could disrupt the way telcos work, moving too fast or relentlessly may be risky. For this reason, it is important to identify well-defined objectives for the transformation and the context in which it will operate. As a rule of thumb, employing open source solutions in the current infrastructure (a brownfield approach) should be confined to well identified products/functionalities and objectives to minimize the risk. Large scale deployments of open source solutions are more likely to be successful in a greenfield context - a situation in which a technological transformation needed to happen anyway (e.g., a new subsidiary, and obsolescent network to be entirely changed, and the like).

Contents

Table of Exhibits

Introduction: why open source?

Commercial pressures and technological opportunities

For telcos in many markets, declining revenues is a harsh reality. Price competition is placing telcos under pressure to reduce capital spending and operating costs.

At the same time, from a technological point of view, the rise of cloud-based solutions has raised the possibility of re-engineering telco operations to be run with virtualised and open sourced software on low cost, general purpose hardware.

Indeed, rather than pursuing the traditional technological model, i.e. licensing proprietary solutions from the mainstream telecoms vendors (e.g. Ericsson, Huawei, Amdocs, etc.), telcos can increasingly:

1. Progressively outsource the entire technological infrastructure to a vendor;
2. Acquire software with programmability and openness features: application programming interfaces (APIs) can make it easier to program telecommunications infrastructure.

The second option promises to enable telcos to achieve their long-standing goals of decreasing the time-to-market of new solutions, while further reducing their dependence on vendors.

Greater adoption of general IT-based tools and solutions also:

- Allows flexibility in using the existing infrastructure
- Optimises and reuses the existing resources
- Enables integration between operations and the network
- And offers the possibility to make greater use of the data that telcos have traditionally collected for the purpose of providing communications services.

In an increasingly squeezed commercial context, the licensing fees applied by traditional vendors for telecommunication solutions start to seem unrealistic, and the lack of flexibility poses serious issues for operators looking to push towards a more modern infrastructure. Moreover, the potential availability of competitive open source solutions provides an alternative that challenges the traditional model of making large investments in proprietary software, and dependence on a small number of vendors.

Established telecommunications vendors and/or new aggressive ones may also propose new business models (e.g., share of investments, partnership and the like), which could be attractive for some telcos.

In any case, operators should explore and evaluate the possibility of moving forward with a new approach based on the extensive usage of open source software.

This report builds on STL Partners' 2015 report, The Open Source Telco: Taking Control of Destiny which looked at how widespread use of open source software is an important enabler of agility and innovation in many of the world's leading internet and IT players. Yet while many telcos then said they crave agility, only a minority use open source to best effect.

In that 2015 report, we examined the barriers and drivers, and outlined six steps for telcos to safely embrace this key enabler of transformation and innovation:

1. **Increase usage of open source software:** Overall, operators should look to increase their usage of open source software across their entire organisation due to its numerous strengths. It must, therefore, be consistently and fairly evaluated alongside proprietary alternatives. However, open source software also has disadvantages, dependencies, and hidden costs (such as internally-resourced maintenance and support), so it should not be considered an end in itself.

2. **Increase contributions to open source initiatives:** Operators should also look to increase their level of contribution to open source initiatives so that they can both push key industry initiatives forward (e.g. OPNFV and NFV) and have more influence over the direction these take.

3. **Associate open source with wider transformation efforts:** Successful open source adoption is both an enabler and symptom of operators' broader transformation efforts, and should be recognised as such. It is more than simply a 'technical fix'.

4. **Bring in new skills:** To make effective use of open source software, operators need to acquire new software development skills and resources – likely from outside the telecoms industry.

5. **… but bring the whole organisation along too:** Employees across numerous functional areas (not just IT) need to have experience with, or an understanding of, open source software – as well as senior management. This should ideally be managed by a dedicated team.

6. **New organisational processes:** Specific changes also need to be made in certain functional areas, such as procurement, legal, marketing, compliance and risk management, so that their processes can effectively support increased open source software adoption.

This report goes beyond those recommendations to explore the changing models of IT delivery open to telcos and how they could go about adopting open source solutions. In particular, it outlines the different implementation phases required to build an open source telco, before considering two scenarios - the greenfield model and the brownfield model. The final section of the report draws conclusions and makes recommendations.

Open Source: Why Now?

Since STL Partners published its first report on open source software in telecoms in 2015, the case for embracing open source software has strengthened further. There are three broad trends that are creating a favourable market context for open source software.

- **Digitisation** – the transition to providing products and services via digital channels and media. This may sometimes involve the delivery of the product, such as music, movies and books, in a digital form, rather than a physical form.
- **Virtualisation** – executing software on virtualised platforms running on general-purpose hardware located in the cloud, rather than purpose-built hardware on premises. Virtualisation allows a better reuse of large servers by decoupling the relationship of one service to one server. Moreover, cloudification of these services means they can be made available to any connected device on a full-time basis.
- **Softwarisation** – the redefinition of products and services though software. This is an extension of digitisation, i.e., the digitisation of music has allowed the creation of new services and propositions (e.g. Spotify). The same goes for the movie industry (e.g. Netflix) or the transformation of the book industry (e.g. ebooks) and newspapers. This paradigm is based on:
  - The ability to digitise the information (transformation of the analogue into a digital signal).
  - Availability of large software platforms offering relevant processing, storage and communications capabilities.
  - The definition of open and reusable application programming interfaces (APIs) which allow processes formerly 'trapped' within proprietary systems to be managed or enhanced with other information and by other systems.

These three features have started a revolution that is transforming other industries, e.g. travel agencies (e.g. Booking.com), large hotel chains (e.g. Airbnb), and taxis (e.g. Uber). Softwarisation is also now impacting other traditional industries, such as manufacturing (e.g., Industry 4.0) and, for sure, telecommunications.

Softwarisation in telecommunications amounts to the use of virtualisation, cloud computing, open APIs and programmable communication resources to transform the current network architecture. Software is playing a key role in enabling new services and functions, better customer experience, leaner and faster processes, faster introduction of innovation, and usually lower costs and prices. The softwarisation trend is very apparent in the widespread interest in two emerging technologies: network function virtualization (NFV) and software defined networking (SDN).

The likely impact of this technological transformation is huge: flexibility in service delivery, cost reduction, quicker time to market, higher personalisation of services and solutions, differentiation from competition and more. We have outlined some key telco NFV/SDN strategies in the report Telco NFV & SDN Deployment Strategies: Six Emerging Segments.

What is open source software?

A generally accepted open source definition is difficult to achieve because of different perspectives and some philosophical differences within the open source community.

One of the most high-profile definitions is that of the Open Source Initiative, which states the need to have access to the source code, the possibility to modify and redistribute it, and non-discriminatory clauses against persons, groups or 'fields of endeavour' (for instance, usage for commercial versus academic purposes) and others.

For the purpose of this report, STL defines open source software as follows:

- Open source software is a specific type of software for which the original source code is made freely available and may be redistributed and modified. This software is usually made available and maintained by specialised communities of developers that support new versions and ensure some form of backward compatibility.

Open source can help to enable softwarisation. As an example, it has greatly helped in moving from proprietary solutions in the web server sector to a common software platform (named LAMP) based on the **L**inux operating system, the **A**pache Http server, **M**ysql server, **P**hP programming language. All these components are made available as open source. This essentially means that people can freely acquire the source code, modify it and use it. Modifications and improvements are to be returned to the development community.

One of the earliest and most high profile examples of open source software was the Linux operating system, a Unix-like operating system developed under the model of free and open source software development and distribution.

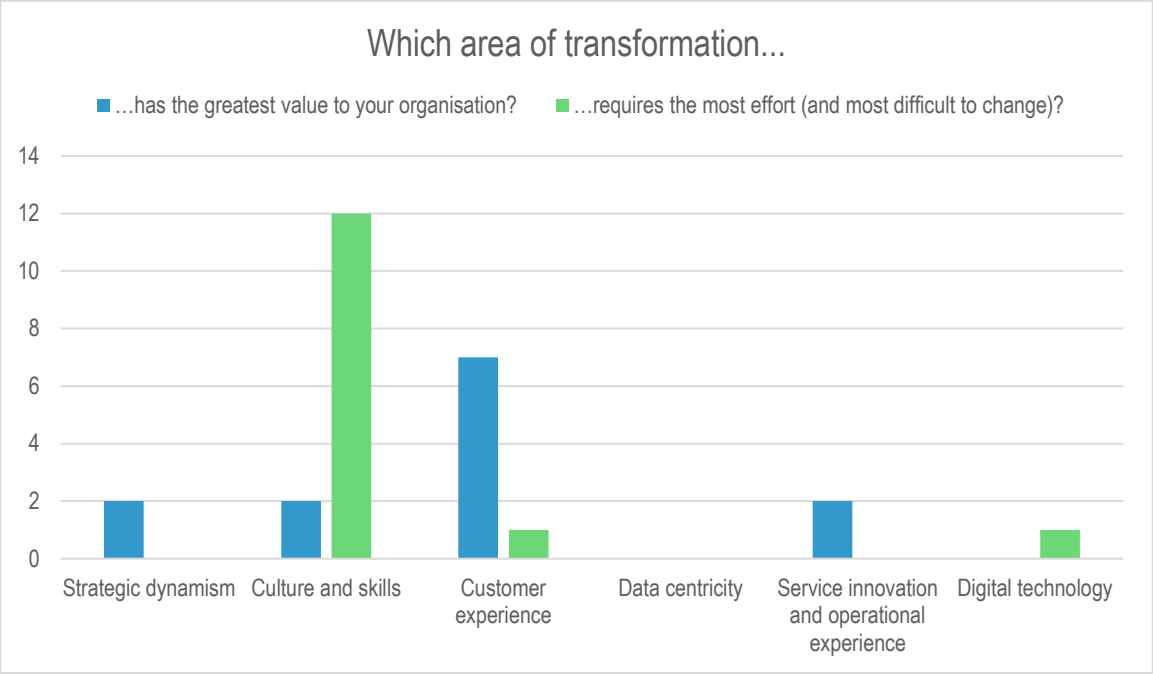
Open source: benefits and barriers

The benefits of using open source

As discussed in our earlier report, The Open Source Telco: Taking Control of Destiny, the adoption and usage of open source solutions are being driven by business and technological needs. Ideally, the adoption and exploitation of open source will be part of a broader transformation programme designed to deliver the specific operator's strategic goals. Operators implementing open source solutions today tend to do so in conjunction with the deployment of network function virtualization (NFV) and software defined networking (SDN), which will play an important role for the definition and consolidation of the future 5G architectures.

However, as Figure 1 shows, transformation programmes can face formidable obstacles, particularly where a cultural change and new skills are required.

Figure 1: Benefits of transformation and the related obstacles



Source: STL Partners[2] - from a survey of telco execs at our Asia Senior Summit workshop in Nov 2016 on 'How to transform'

The following strategic forces are driving interest in open source approaches among telecoms operators:

- **Reduce infrastructure costs**. Telcos naturally want to minimise investment in new technologies and reduce infrastructure maintenance costs. Open source solutions seem to provide a way to do this by reducing license fees paid to solution vendors under the traditional software procurement model. As open source software usually runs on general-purpose hardware, it could also cut the capital and maintenance costs of the telco's computing infrastructure. In addition, the current trend towards virtualisation and SDN should enable a shift to more programmable and flexible communications

[2] STL Partners, Changing Culture: The Great Barrier, p. 23

platforms. Today, open source solutions are primarily addressing the core network (e.g., virtualisation of evolved packet core), which accounts for a fraction of the investment made in the access infrastructure (fibre deployment, antenna installation, and so forth). However, in time open source solutions could also play a major role in the access network (e.g., open base stations and others): an agile and well-formed software architecture should make it possible to progressively introduce new software-based solutions into access infrastructure.

Mitigate vendor lock-in. Major vendors have been the traditional enablers of new services and new network deployments. Moreover, to minimise risks, telco managers tend to prefer to adopt consolidated solutions from a single vendor. This approach has several consequences:

- Telcos don't tend to introduce innovative new solutions developed in-house.
- As a result, the network is not fully leveraged as a differentiator, and can become the full care and responsibility of a vendor.
- The internal innovation capabilities of a telco have effectively been displaced in favour of those of the vendor.

This has led to the "ossification" of much telecoms infrastructure and the inability to deliver differentiated offerings that can't easily be replicated by competitors. Introducing open source solutions could be a means to lessen telcos' dependence on specific vendors and increase internal innovation capabilities.

- **Enabling new services**. The new services telcos introduce in their networks are essentially the same across many operators because the developers of these new services and features are a small set of consolidated vendors that offer the same portfolio to all the industry. However, a programmable platform could enable a telco to govern and orchestrate their network resources and become the "master of the service", i.e., the operator could quickly create, customise and personalise new functions and services in an independent way and offer them to their customers. This capability could help telcos enter adjacent markets, such as entertainment and financial services, as well as defend their core communications and connectivity markets. In essence, employing an open source platform could give a telco a competitive advantage.

- **Faster innovation cycles**. Depending on a vendor makes the telco dependent on its roadmap and schedule, and on the obsolescence and substitution of existing technologies. The use of out-dated technologies has a huge impact on a telco's ability to offer new solutions in a timely fashion. An open source approach offers the possibility to upgrade and improve the existing platform (or to move to totally new technologies) without too many constraints posed by the "reference vendor". This ability could be essential to acquiring and maintaining a technological advantage over competitors. Telcos need to clearly identify the benefits of this change, which represent the reasons, the "why", for the softwarisation.

Overcoming the barriers to using open source

As in all important transformations, the move towards open source and, generally speaking, a software-oriented telco, is an endeavour that requires planning and a clear strategy. Although software has been a strategic asset for operators for a long time, their approach is typically the one of the "intelligent buyer", rather than in-house development. Consequently, many IT departments of major operators are organised for large software project management: intensive software development is essentially outsourced to third parties, while in-house software development is rarely considered strategic. Figure 3 summarises STL Partners' view of the main requirements/barriers facing a telco looking to embrace open source solutions.

Figure 2: The key barriers in the path of a shift to open source

	Barriers	Potential solutions
	Resistance to the need for change	The rationale for open source should be explained across the company
	Failure to properly value programmers	Give software developers recognition and enable them to develop their own skills and abilities within a telco.
	A lack of programming skills	Make mastery of software the mantra of the new telco
	Comfort with vendor lock-in	Encourage risk taking (especially at the very beginning of the softwarisation process)
	Uncertainty about where to start in-house development	Consider piggybacking on the shift to 5G
	Failure to allow for trial and error	Build enough time into development projects for learning and experimentation
	Software solutions can quickly become outmoded.	Develop and improve each component independently from the others, while still ensuring these components can be integrated
	Risk that new protocols and processes are ignored.	Mandate the use of APIs as appropriate
	Need for new methodologies	Study and test agile software development approaches
	Telcos typically lack a developer community.	Hire developers who already have a strong external network.

Source: STL Partners

Steps to overcome the barriers

Accepting the "Why" and overcoming resistance to the need for change. The need for a transformation from a traditional telco to essentially a software telco has to be clearly understood at all levels of the

organisation. The need for change should be shared and a new awareness about the importance of the software has to permeate the company. In addition, a clear strategy needs to be planned and shared. STL Partners previously outlined how to address such issues in the report Five telcos changing culture: Lessons from neuroscience.

Overhauling the culture and addressing the failure to properly value programmers. To be successful in network softwarisation, telcos need to shed the traditional mindset that values “engineering and management of projects” over software development. They also need to eradicate an often deeply held conviction that software-related activities are not of value for a telco. Often the skills and the value of developers have been neglected in favour of project managing the network design people and their competences. Now, telcos need to recognise that software is central to the business. This is an important move from a project management culture to an agile development culture. And it has huge consequences for the entire company organisation.

In most telcos, a software developer doesn’t rank highly in the corporate hierarchy. Programme managers or coordinators are generally deemed more valuable. Telcos need to recognise, appreciate and reward these very important professional figures. The leading Internet companies have done this since their inceptions and have prospered as a result. These companies are now technology leaders, while many telcos could hardly be recognised as technology companies. As well as being given recognition, software developers should be able to develop their own skills and abilities within a telco: relevant equipment and tools should be available, while new ways of work organisation could be tested.

Mastering software and acquiring appropriate programming skills. Deep insight on IT and web technologies is needed, as is practical know-how about software technologies, such as:

- New software methodologies and programming techniques;
- Practical experience about large software project management (the definition, development, deployment and maintenance of large software platforms);
- Software platforms, i.e., the way to incrementally create large software platforms able to leverage the company’s assets;
- The usage, value and need for open application programming interfaces (APIs) for taking advantage of the software flexibility.

Mastery of software should be the mantra of the new telco and this approach should be embraced and supported by the top management and be practised by all the departments of the company. The company’s efforts should be directed to development, rather than buying products and external integration. A large portion of the company should be proficient in developing new software solutions, and in using, modifying and improving all the existing software products. Although external development can still be valuable, knowledge of how the software is designed and developed must be held internally.

Fostering vendor independence and encouraging risk taking. The telco needs to dismantle the cultural mindset of vendor lock-in. Many senior and middle managers in telcos are comfortable with dependence on a major vendor, as this is regarded as a low-risk approach. It is very easy to demand the development and deployment of consolidated solutions vendors. The “nobody ever got fired for choosing IBM" syndrome is alive and well in many telcos, impairing entrepreneurship on the part of the management. Instead, a willingness to take risks (especially at the very beginning of the softwarisation process) should be supported across the organisation.

Identifying what to change first and overcoming inertia: What to build first is an essential decision. This is a real challenge: What platform? For what purpose? Here, clear business perspective plans are essential. For instance, the 5G time horizon offers an opportunity to think and develop an open source solution for the future now. As the 5G network will be substantially different from the current one, this could be a safe place to start because the entire mobile industry is currently keen to experiment and develop new solutions for this new architecture. As established vendors will develop products and solutions to support the deployment of 5G, telcos will have a back-up option in the case of failure of internal projects. Moreover, telcos will be better placed to judge and evaluate vendors' 5G solutions, as they will have more practical and effectual knowledge of the issues and intricacies following their own internal experimentation and software development. Another approach is to start from burning issues within the company and to try to make a difference by empowering internal development. Different issues are relevant to different operators: a better BSS/OSS platform[3]; the migration of obsolete solutions to the cloud; a better and more modern customer care approach; and many more. These problems represent opportunities to move towards open source and create a new platform. In any case, a clear statement and vision about "what" should be defined.

Creating time to experiment and allowing for trial and error. Building a software platform takes a long time and involves many risks. Large software projects with long development periods frequently fail, especially in organisations that do not have or have not acquired the relevant know-how. The project timeline for the creation of a new platform should allow enough time for development groups to grow and to learn. The team should be able to implement a set of small-scale experiments with a trial and error approach so that the different development teams can learn from their mistakes. The timeline and scope for experimentation should be clearly defined up front.

Avoiding obsolescence and ensuring software solutions evolve. Together with long development times comes the quick technological obsolescence of the platform, i.e., new and better technologies are made available. The web industry has approached the problem by developing a set of basic and independent solutions (for instance, the aforementioned LAMP bundle). In this case, each component is developed and improved independently from the others, while still ensuring these components can be integrated. The involvement of large open source communities can help to speed up the development of open source solutions, while enabling experimentation with different technological paths. They can develop and improve different components (independently from each other and sometimes in competition with other technologies or groups) keeping up the pace of technological evolution. Users of these solutions can select and integrate the components and exploit the evolution of the components' capabilities. These independent groups often pursue different technological paths that would not be possible to follow in a monolithic approach (e.g., within one single company).

Promoting the use of APIs so that new protocols and processes aren't ignored. Creating a platform is just the first enablement step. It is then important to use it internally and to create functions, services and applications on top of it. In particular, the extensive use of the platform's APIs is crucial. Without this step, the platform won't be used and will be a futile effort. The telco should also seriously consider opening the interfaces externally to the company. This means tackling the relevant security issues and being able to support and nurture a community of developers.

Senior managers should impose the use of APIs, as Jeff Bezos did with his famous mandate in the early 2000s about the internal usage of APIs. According to a blog post by a former Amazon staffer, that mandate went something along these lines:

[3] We have previously discussed virtualization of OSS in the report NFV and OSS: Virtualization meets reality

"All teams will henceforth expose their data and functionality through service interfaces.

Teams must communicate with each other through these interfaces.

There will be no other form of interprocess communication allowed: no direct linking, no direct reads of another team's data store, no shared-memory model, no back-doors whatsoever. The only communication allowed is via service interface calls over the network.

It doesn't matter what technology they use. HTTP, Corba, Pubsub, custom protocols -- doesn't matter. Bezos doesn't care.

All service interfaces, without exception, must be designed from the ground so that they can be used externally. That is to say, the team must plan and design to be able to expose the interface to developers in the outside world. No exceptions.

Anyone who doesn't do this will be fired".

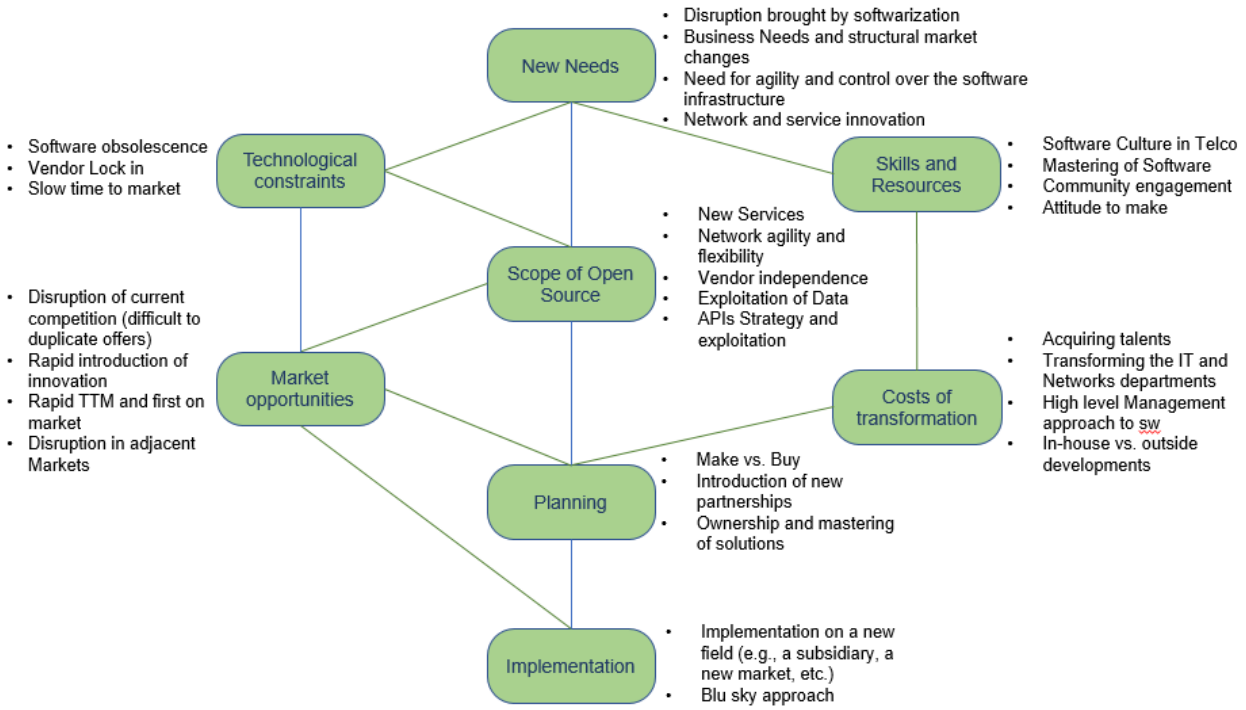
Although Bezos' mandate, as described by this former staffer who now works for Google, seems excessively rigid and dogmatic, it highlights how Amazon (and other leading Internet companies) regard APIs as absolutely fundamental to what they do.

Defining the "How" by introducing new methodologies: New agile software development approaches and methodologies should be studied and tested and, if they prove valid, they should be accepted and used. Note, these methodologies refer to the "How" and are by far less important than the "Why" and the "What". Indeed, telcos should not deem agility as the panacea for their software issues: some transformation projects fail because they focused on "how" to develop things and not on "why" and "what" to develop. Agility and new methodologies are important, but they must be treated for what they are: a means to reach the important results (the "why and what").

Winning third party support and building a developer community. Open source platforms need the support of a community that can help with the development of components of the platform (as in the LAMP example) and development on top of the platform. Aware that there are different types of developers, telcos will need to decide how to attract, reward and grow an appropriate community to support their open source initiatives. One way to kick-start this process is to hire developers who already have a strong external network.

In summary, the first challenge is to clearly define the scope of the move towards an (open source) platform: which parts of the network infrastructure will be challenged first by in-house development. Then the "How" should be determined in terms of methodologies, tools and software components. Figure 3 summarises the decisional steps required to evaluate a move to an open source strategy.

Figure 3: Shaping an initial strategy for the adoption of open source solutions



Source: STL Partners

Choosing the right path to open source

As discussed in Figure 3, a full shift to open source may require a large organisational transformation. For some telcos, this transformation may be too difficult, expensive or just risky. In other cases, there is not an urgent need to adopt this transformation, because neither the market nor the current approach requires change yet.

Each telco has to decide if and how to make the transition to open source. This possible transition can be applied in full or in specific fields. For instance, a progressive transformation could be implemented in the operational part of the telco software architecture. For example, management products could be progressively substituted by similar or newer open source counterparts, without disrupting the entire software infrastructure. Some telcos need to migrate obsolete solutions (e.g., based on out-phased hardware and software) to new IT platforms, opening opportunities to test virtualisation and open source alternatives. In this case, a relatively small transformation project could achieve this migration and, at the same time, increase the skills and know-how on open source. Another viable option is to adopt open source in the new network components. In this case, the implementation of 5G networks could enable a transition from experimentation to deployment.

There are essentially three **different options** telcos can take to harness the potential of open source solutions:

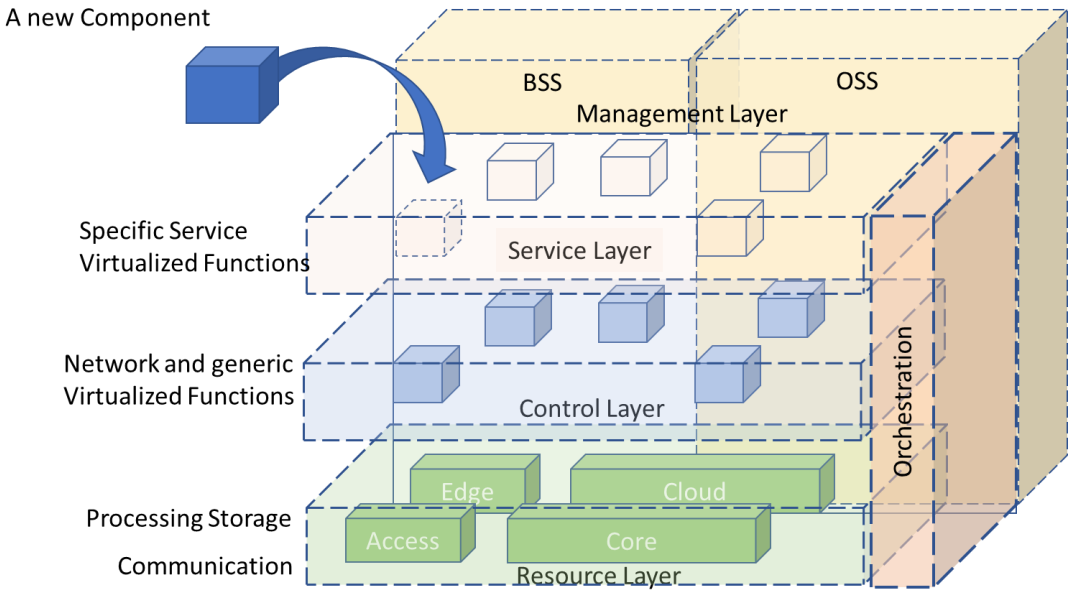
1. **A tactical open source** option: a telco opportunistically and progressively substitutes obsolete components with new ones that are open source. Figure 5 depicts how an open source component could be slotted into a telco's existing software architecture.

2. **A full open source** option: this is the path followed by a telco that has strategically decided to transform its infrastructure by introducing new technologies (e.g., SDN, NFV supported by new virtualised cloud and edge infrastructure) mainly based on open source solutions.

3. **A migrating open source** option: the path followed by a telco that is putting in place an open source based strategy and related infrastructure and is migrating existing functionalities to a new virtualised cloud and edge platform. Not all the functions and components are open source yet, but there is a long-term strategy to substitute existing components and functionalities with new open source components. This operator could introduce new technologies (e.g. 5G-related solutions) based on the open source approach and integrate them into the existing infrastructure.

It will take a long time for an established telco to become a pure open source telco because of the existing legacy of software and systems. However, greenfield operators, including established telcos entering new markets, could try this approach. As they need to introduce a totally new infrastructure, it would be an ideal opportunity to adopt a state-of-the-art software network architecture (see the penultimate chapter for more discussion of this opportunity).

Figure 4: A new open source component in an existing infrastructure



Source: STL Partners

Selecting the right IT delivery model

As well as selecting which open source option they wish to pursue, the telco needs to consider which IT delivery model they want to adopt. Again, there are essentially three models:

1. **Traditional model** (vendor dependency): Telco buys blackbox, proprietary software, vendor sorts it all out
2. **Hybrid software model** (intelligent buyer): Telcos buys hardware with some APIs, buys software from a variety of vendors with some APIs
3. **Full software model**: Buy commercial off-the-shelf hardware, buy or build software, fully customisable

1. Traditional model (vendor dependency)

In this model, the telco specifies its high-level architecture and then works with vendors to determine which components and products fit its blueprint. The telco pays for each customisation of the solution by the vendor. Typically, the telco buys hardware and proprietary software (even if many of the components may be based on open source). The development, integration and testing phases are carried out by the vendors. Once the solution is up and running, the telco will operate it, but any new function, change or addition is negotiated with the relevant vendor in accordance with an agreed release plan. In case of malfunctioning or underperforming issues, the vendor is responsible for the problem resolution. The telco doesn't need to do software development, and if new software is developed in the platform, the telco may be considered responsible for issues and problems affecting the entire infrastructure. With reference to Figure 4, the telco see the entire architecture as a set of blackboxes that may be accessible via APIs.

2. Hybrid software model (intelligent buyer)

The telco considers which open source components can by deployed for specific services or functionalities and be integrated with the legacy infrastructure. Evolution to open source in this case take places by substituting old software packages with new open source solutions that do not require modifications or changes to the basic platform of the telco. The introduction of new solutions is limited to the phase out of old packages that are obsolete or the introduction of new functionalities on top of the existing software infrastructure. Some vendors can support this approach through the supply of components or specialised platforms that offer and support APIs. In this case, the border between the competences and the responsibility of the vendor and those of the telco are made clear by the segmenting of the functionalities. With reference to Figure 5, the telco sees the architecture as a combination of black boxes and programmable and modifiable pieces of software.

3. Full software model

In this model, the telco uses commercial off-the-shelf (COTS) hardware to run programmable components and packages accessible via APIs. The large majority of these components and packages will be modifiable and open (essentially open source). In principle, there are no functions or features that cannot be customised or modified directly by the operator. The software architecture is fully modular and the telco has the ability to change and update any single feature of the platform. With reference to Figure 5, the telco sees the platform as transparent and all the updates and evolution work are its own responsibility.

Choosing the right model for the right scenario

To determine the best approach to becoming an open source telco, a clear business proposition is needed. This section considers three overarching business strategies that will influence the telco's approach.

- **Connectivity provider**: The telco focuses on creating a lean and robust infrastructure that supports the connectivity needs of its customers. Basic communication services will be offered and the entire network architecture is organised to be as effective and reliable as possible. The goal is to deliver robust, high-speed connectivity.
- **Platform provider** (an enabler): The telco focuses on creating basic network functions and exposing them by means of APIs. The goal is to create a feature-rich and programmable platform for customers and third parties. The telco will need flexibility in the allocation and usage of connectivity resources, as well as a flexible and programmable management system that can account and charge for usage resources. The goal here is to maximise the usage of the platform resources.
- **Service provider**: the telco provides full services and charges accordingly. The interface with the customer is at the level of services and applications. The goal is to maximise service revenue from customers.

Figure 5: Different kinds of telcos need to select different delivery models

	Traditional model (vendor dependency)	Hybrid software model (intelligent buyer)	Full software model
Connectivity provider	🟢	🟡	🔴
Platform provider	🔴	🟡	🟢
Service provider	🔴	🟡	🟢

Source: STL Partners

Connectivity provider case

To execute this strategy, the major goal of the infrastructure is to deliver fast and cheap connectivity. The telco is not interested in fancy or feature-rich services, but it will want to take advantage of an optimised infrastructure that requires minimal maintenance. The traditional IT delivery model is most likely to prove effective and economical. In fact, the telco could seek to fully outsource the network infrastructure to a reference vendor and focus on marketing and customer care.

However, the hybrid IT delivery model could be useful if major customers require a specific tailoring of functionalities to fulfil special requirements. In this case, the telco may consider using specialised open source software, if the required modifications would be too expensive to source from a vendor.

The full software IT delivery model would only be applied in a very few and extreme cases: a new experimental network deployed in a specific situation by the telco looking for full control or a situation in which new technologies are available and their deployment could disrupt other operators' business.

Platform provider case

In this scenario, the focus of the telco is on the network platform. As such, it needs a high level of programmability. Implementation of open source solutions, especially those supported by a large developer community, through the full software IT delivery model should provide the necessary flexibility and evolve the platform components at an appropriate pace. Although this platform could reside on COTs hardware, some adaptation towards proprietary interfaces may be required to integrate valuable functions into the platform. With interfaces toward the service layer (in which the telco itself isn't competing), developers and third parties can use open source or proprietary approaches for developing applications on top of the platform. In fact, the telco should consider nurturing a large community to populate the platform with useful applications. At least initially, an open source approach may fit better with this goal.

The hybrid IT delivery model is also applicable here. Even proprietary software components that behave as black boxes, can provide reliable and well-defined functions that contribute to the richness and effectiveness of the platform. A specific case could refer to the usage of a basic set of functions (a skeleton of a platform) that can be extended to create new middleware on top of it. However, there may be the risk of vendor lock-in.

The traditional IT delivery model is not well suited to the execution of this business strategy. There is a danger that the telco will simply end up reselling a fully-fledged platform and professional services supplied

by a vendor to its customers. As the vendor should be able to promptly solve issues and problems, it is questionable whether the telco could add value by acting as a mediator and providing an additional level of support. If there is not enough value added by the telco, customers may find it convenient to migrate to the platform offered directly by the vendor or to move to another platform that offers better professional services.

Service provider case

As this type of telco is focusing on the provision of services, it needs an IT platform that requires minimal maintenance and upgrading. This type of telco is more interested in the usage of the platform than in the development of it.

The full software IT delivery model can be applied across the service layer and to the development environment so that the telco has full control and responsibility of the services offered and delivered to the customer. The underlying platform must support open and well-defined APIs, but it doesn't need to be open source. A proprietary platform guarantees prompt and risk-free support, but is less performant if extensions and fast evolution is required. In that case, an open source platform offers the possibility to directly develop extensions and tailored solutions for specific valuable customers. In this case, telcos may find new kinds of vendors have convincing propositions. A vendor with an open source platform could guarantee a level of support similar to the traditional model, while also allowing the telco to access the intricacies of the platform when and if needed. If the telco decides to adopt the full software IT delivery model across the board, it would be wise to create competences and development groups within the telco and within the developers' community that can modify, change and improve the platform, as necessary.

The hybrid IT delivery model is a viable option, in this case, both at the service and the platform layers. The telco could mix and match open source and proprietary components at both levels. It may be preferable to have some homogeneity at the platform level to avoid the risk of spending a lot of time in finding malfunctions in different types of components that could be out of the control of the telco. In the service layer, the use of open source software would enable the telco to be responsive to the needs of customers.

Although some proprietary packages can add value and are necessary for extending the service offering, experience suggests this approach needs to be applied carefully. For example, the use of proprietary software has limited the impact of the Intelligent Network standard architecture. As the Intelligent Network service delivery platforms (and even more the Intelligent Network nodes) are proprietary, they are offering the same functionalities to each service provider, making it difficult for the innovative service players to distinguish themselves from more traditional ones. As a result, a new service is soon replicated by competitors and the platform providers are soon able to provide the necessary platform changes to all their clients. While at the technical level, Intelligent Network platforms are working, at a business level they have not yet delivered up to their promises.

Finally, the traditional IT delivery model isn't an appropriate way to execute this business strategy. Although it would provide a strong and stable platform, this delivery model would limit the telco's ability to differentiate its service offering from that of its competitors.

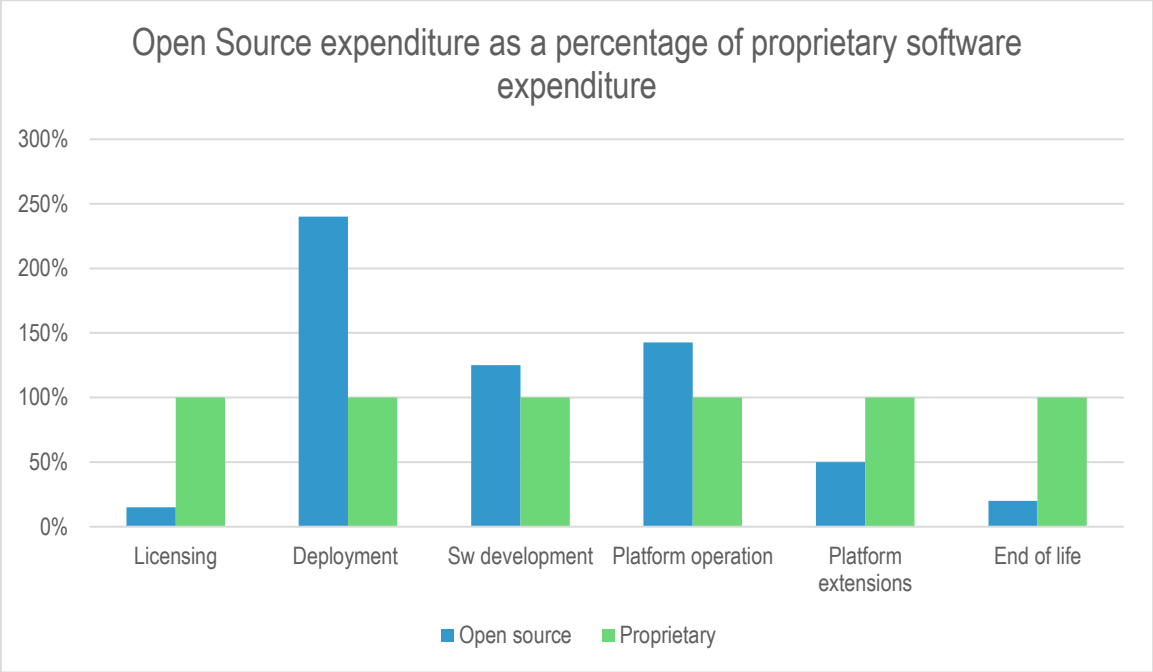
Weighing the cost of open source

Although open source solutions don't attract licensing costs (an evident advantage of open source), there are other cost factors a telco needs to consider. This section sets out the basic considerations for a telco looking to compare the cost of open source and proprietary software. To better understand the possible issues, the full software life cycle needs to be considered and any hidden costs identified.

Figure 6 gives an illustrative estimate of how the various costs compare across open source and proprietary (in terms of percentages). In the chart, we estimate each stage of the open source software lifecycle costs as a proportion of the cost of proprietary vendor software solutions. So, as illustrated below, we estimate that licensing costs for open source software are around one seventh of proprietary software, but deployment costs are more than twice as high[4].

As a rule of thumb, telcos should pay close attention to the licensing and the deployment support costs because they are two major features that may be discriminant in terms of adoption of open source solutions. In addition, if the solution needs to be customised extensively, then open source is a viable and convenient approach. Obviously, the creation of skills and attitude to software development is a prerequisite of the usage of open source solutions, unless the user pays for professional services.

Figure 6: Illustrative estimate of Open Source costs versus a proprietary approach



Source: STL Partners estimates

Below, we explain the differences between each stage of software deployment for open source versus proprietary software that underpin our estimates of the cost structure.

[4] Our illustrative estimates are based on the experience of the author, and external discussions with other practitioners. Clearly the actual savings will vary from project to project, both in terms of the relative savings and the absolute costs in each stage.

At first glance, open source solutions have a clear advantage over proprietary ones. However, even if the licensing fee is zero or very low there are costs related to installation and deployment, customisation and tuning up of the platform. In the telecoms sector, proprietary software usually comes with some support for instantiation, deployment and maintenance. Typically, an open source solution has to be compiled, installed and deployed by the company itself.

Moreover, proprietary software could also be installed and provided under precise service level agreements (SLAs), with related costs, while in a typical open source implementation, no SLA can be requested.

Cost of licensing of the software: Open source solutions are available with minimal or no licensing fees. This is a big difference with respect to proprietary software: licensing is the way vendors can exploit their software, but to make this approach valuable to customers, licensing also covers updates and some forms of support to the customers that open source solutions do not usually provide.

The full installation and deployment costs: This relates to the time and resources required to set up the solutions. With proprietary software, deployment is guaranteed by the vendor, making it hassle free to the customer. The costs of deployment are generally hidden in licensing or covered in a specific contract. It is rare for customers to have to deploy proprietary software without support.

In an open source environment, the telco will have to pay these costs, but this allows it to customise the installation and the deployment to meet their needs, and this means more agility and responsiveness. Paying for the installation or deployment could be in the form of professional services, or through an in-house development team. Thus, a lot of the savings on the licensing costs must be put into the deployment stage.

A specific case within the deployment stage is software for supporting cloud computing and virtualization, OpenStack, which is a major open source product used in many major deployments. In this case, the installation is performed directly by the company that will finally operate the platform. The configuration and tune up of the platform is a very long procedure, particularly if there are no professional services (or community services) that can be used to cut down the installation and deployment time. Some aspects of the documentation could also be missing. Mastery of OpenStack solution is needed to customise the solution to the needs of the company using it. Many operators know how cumbersome the installation and the deployment of OpenStack can be, and, in some cases the deployment of a new version of the platform is delayed because of the reluctance of the internal development team.

Software development costs: Once the platform is up and running, there is the software development phase of applications and services on top of the platform. Essentially, the software development costs of proprietary and open source solutions may be quite similar. However, the support in this phase is very different: for an open source solution, tutorials and development support is left to the community and its related capability, meaning some issues have to be solved by the in-house team without much support. On the proprietary side, the vendor could provide tutorials and professional services help, either as part of the licensing fees or as an additional cost.

Platform operational costs: Generally speaking, there are substantial differences between open source and proprietary software in this phase. Proprietary software usually provides a precise schedule for updates and improvement of subsequent versions of the software. This is not always the case with open source. Furthermore, maintenance and updates in open source sometimes have to cope with new versions or forks of important projects that are not necessarily backward-compatible with the current version of the platform. This could provide new and enhanced functionalities, but could create delay and new costs for a change in release and some rewriting of old software.

Platform extensions: In contrast to platform operations, platform extensions in open source can come for free and may be well timed to address the market needs. For extensions developed in-house, there are clearly costs of employing skilled software developers. However, these can be mitigated by working with the development community – extensions that have general value can be promoted within a user community and some of the costs and time can be shared with other entities.

Proprietary solutions (even if provided with code) are closed, meaning that any extension is developed by the vendor. Customisation can be possible but it may be very expensive and constrained by the development cycles and schedule of the vendor. Customer development is, in principle, possible, but is comes with important costs: the customer can lose warranties and support from the vendor and must cover all development costs.

End of life: Ultimately, the platform will become obsolete, so the costs in this phase consist of closing down a platform, and possibly porting the services onto another platform or solution. Here the vendor lock-in is evident: the manufacturer/vendor of the proprietary solutions will try to delay the migration to other solutions (especially if offered by another provider) and their professional services teams will put in considerable effort to guarantee a smooth migration to a new platform. This is one of the forms of vendor lock in, which can cost a substantial amount of money in terms of loss of market opportunities or delays in entering new markets.

In the open source case, the end of life is generally smoother and not dependent on vendors. Some forms of dependency could come from need to learn how to adapt functions to a new software platform, but this is also the case for proprietary software. In general, the migration from an open source solution to another one is easier because of less vendor resistance.

In summary, if a telco has a good set of skills in software development and management, the open source option could be cheaper and more efficient (in terms of agility and responsiveness) than proprietary software. However, many telcos' IT departments contain more project management skills than real development skills. A shift from management towards building software solutions internally could require a substantial investment in human resources and skills. At an early stage, this could result in an increase in IT costs and in a decrease in efficiency. In the longer run, this strategic choice could result in a substantial capability to operate in a software-based environment. The telco needs to ask how much time and resources could be devoted to a transformation into a software company.

Another related issue is the question of who will drive the software transformation within a telco? The two natural candidates are the CIO and the CTO, both of whom will have to have considerable software development competencies to successfully run their business. Although with the softwarisation trend, the differentiation between the two will soon no longer make sense, in the meantime the telco should consider the following criteria, depending on its strategic choices:

1. If the path to open source adoption starts from the transformation of the operational and management infrastructure (BSS and OSS), then the clear lead is the IT department;

2. If the first step in the transformation path is represented by the adoption of NFV/SDN, then the driving force will be the Network department.

Independently from the driving force, it is clear that the IT and Network relationship needs to be synergistic, partly to consider another essential issue: the exploitation of the telco's data. There is an urgent need to create an interdisciplinary approach between IT and Network based on the newer software technologies. And this osmosis has to happen fast.

Which telcos are using open source today?

Many telcos are already engaged with activities related to open source exploitation. Some are involved in standard organisations, or consortia for promoting the development and adoption of open source solutions in specific applications domains. In addition, some of them are even cooperating with web companies to create a telecom global infrastructure based on open source (the Telecom Infra Project, TIP5, launched by Facebook). Obviously, many related activities are also going on in the NFV/SDN realm (see for instance Open Day Light6 consortium).

A major European operator, Deutsche Telekom, has completed a migration to a full IP network in Croatia introducing advanced NFV/SDN functionalities. The solution employs some open source components, such as OpenStack[7]. This implementation is important because it demonstrated the feasibility of many NFV/SDN concepts, while using open source components. It is important to note DT chose to experiment in a relatively small operating company with a need to refurbish and improve the network infrastructure. This has allowed for experimentation in a relatively small environment, to create competences for the global organisation, and to export them across the international group. In the operating company, the cultural resistance to change is likely to have been low, given it had the opportunity to do something new and relevant for the entire corporation. As the focus was on NFV/SDN innovation, the lead was in the hands of network department.

In another example, smaller in terms of effort and innovation, but extremely important from the IT point of view, a telco[8] in Latin America has used open source solutions to introduce e-commerce and payment-related functionalities in its network. The traditional approach, i.e. ask a large vendor to provide a solution, was unsatisfactory for this company because the vendors were all proposing oversized solutions for supporting e-commerce and payment capabilities. These solutions were aimed at a large general retail organisation that could sell everything from connectivity to fridges. This fully-fledged infrastructure was too expensive and ill-suited to the business goal of this telco. Instead, it decided to use open source packages and to integrate them into its current infrastructure. A relatively small group of developers was able to integrate, customise and fully exploit the solution. The solution is totally under the control of this group and it is now possible to customise, modify, and improve it according to the changing needs of the market and the specific objective of the telco in this specific sector. The development the group is fully motivated to continue pursuing this endeavour.

In the US, many major operators are embracing the open source approach. However, the US market is different to other parts of the world, as it is easier to hire people with relevant software development skills and COTS hardware and cloud computing may be cheaper than elsewhere. One of the bigger US operators (undisclosable) is now trying to address relevant groups within the open source community to drive development towards its needs. One possible strategy for orientating these communities towards the needs of the operator is to hire relevant people with a reputation within these communities and ask them to address the needs of large scale deployment of open source solutions especially in the realm of NFV/SDN.

In East Asia, there is also interest in open source deployment. For instance, NTT has launched an open source project applied to the needs of an Internet Exchange operator aiming at substantially reducing the time for accommodating the requests of clients. The GoBGP project[9], based on an open source Internet

---

[5] Interested readers may find more information at https://telecominfraproject.com/
[6] https://www.opendaylight.org/
[7] Axel Clauberg "Deutsche Telekom TeraStream: A Network Functions Virtualization (NFV) Using OpenStack Case Study" available at https://www.a10networks.com/sites/default/files/case-study-files/A10-CS-80103-EN.pdf
[8] STL Partners isn't at liberty to disclose the identity of the operator
[9] http://www.ntt.co.jp/news2016/1609e/160930b.html

routing control software provided by NTT, is operational and fulfilling the design requirements. It is based on SDN developments and OpenStack usage. By means of this project, NTT has increased its technological expertise in network control software and its integration and improvement of OSS. The solution is also offering relevant APIs for controlling the infrastructure.

How can you build an open source telco?

To become an open source telco, operators need to begin by acquiring considerable expertise and skills in software development; then practise open source development habits, adopt best practices and tools; and finally mix and match the components for building a viable solution. Once a group of professional developers has been created, they should be progressively exposed to larger experimentation and real-world deployments.

In essence STL Partners sees three phases to becoming an open source telco:

1. **The skill acquisition phase** during which selected telco developers are used as mentors for other younger or less expert programmers.

2. **The experimentation phase** in which the development teams practise and refine their skills on solutions that could be potentially disruptive and be deployed.

3. **The deployment phase** in which the new skills and the trained people are engaged in real projects that will lead to a deep transformation.

Note that these phases are not intended to be carried out in a sequential “waterfall” model, they are the steps that a junior programmer or a person to be re-trained should be going through in order to transform the organisation. Software competences should be pervasive in an open source telco, and time is needed to enlarge the development base.

In the medium term, most people in the telco’s network and IT departments should master software. Depending on the size of the company, a great effort must be spent training existing personnel, as well as hiring relevant skills that are missing.

Ideally, the nucleus from which to start is the innovation group either in the network or the IT department. In fact, a blend of the two departments could help to overcome artificial borders between real-time and back office operations. Competent people in these sectors should have relevant skills in software development and in network control and management. As previously discussed, a vision is needed to ignite the right people and skills.

In telcos undertaking innovative initiatives related to the future 5G architecture, these initiatives could be a vehicle through which an operator could develop the internal competences required by an open source telco.

Skill acquisition phase

For the skill acquisition phase, the telco should seek to exploit the existing development competences of “group leaders”. Ideally, each leader should be able to build industrial software and lead small development teams. They may be certified developers. Each development team, which will be made up of junior developers seeking to reach a professional level in software development, would be assigned a particular goal with relevance to the company. For example, a team could be asked to implement an NFV orchestrator or develop global control capabilities for a SDN-based network. Guided by their group leader, the team members would then learn to program with modern techniques and experiment with the introduction of programmability and security within the software infrastructure of an operator. The size of these groups should be small to ensure a good flow of information and effective mentoring. The more experts the telco has, the more groups it can create, enabling it to spread a new approach to implementing software more rapidly within the company. The training period should allow for different iterations. Particularly skilled people

emerging from the first iteration could become group leaders for subsequent iterations. As one of the objectives is to create well-crafted components the telco can use, each group needs enough time to achieve meaningful results in development.

The second step is to consolidate the knowledge. This can be done by defining a set of tools, methodologies and languages designed to support the later development phase. Moreover, experiences and solutions arising from the experimental phase could be shared within the developer community (e.g., the best available NFV orchestrators). This consolidation can result in a fast and challenging learning mechanism, the acquisition of some software components, the creation of a common background and the development of team spirit.

The experimentation phase

After the training phase, the developers should engage in larger projects designed to prove the feasibility of deploying an advanced solution within friendly customers. Still, in this experimentation phase, people should be free to learn and to increase their knowledge by applying it to important use cases.

In this phase, experts hired from external companies could be integrated into the teams or could take on the leadership of particularly important experiments. Demonstrating strong commitment to the development of open source components, these teams should be the driving forces behind the shift to a new approach to software. In this phase, the nurturing of external open source communities is of particular importance. The telco should seek to build a strong relationship with these communities (even by hiring the leaders of particularly important open source components). The communities could be relevant also for running experiments with “friendly customers” that can help resolve issues.

The experimentation phase should also lead to the progressive consolidation and improvement of the supporting open source development infrastructure, together with greater precision in the choice of reference open source components (e.g. major platforms, or environments).

The development phase

This phase involves the actual exploitation of the acquired skills for the purpose of large open source software development. Skilled people with experience of working on open source tools and awareness of the best practices take on the tasks required to transform the business. These professionals are aware of the importance of their role, the importance of their development and have enough knowledge and the right relationships to win the co-operation of the developer community. These developers will be required to promote, within the reference communities, those specific developments needed to complete and improve the open source solutions used by the telco.

This section has depicted a general “spiralling” model for skill sharing and improving, but telcos operating in a greenfield or a brownfield situation may need to take a different approach. These scenarios are discussed in the following two sections.

Greenfield model

In the case of a new company or division, the process of building an open source telco could be simplified depending on the number of people being hired and their skill levels. In a new environment in which people have been selected for their competences in software development, the important thing is to have leaders with relevant experience in carrying out large software projects. A lot of iteration and consolidation could be avoided, while some basic decisions about tools and software development methodologies could be taken by a small group of experts. Moreover, new recruits could be selected on the basis of the technologies and tools they are expert in. It would be important to hire people that have experience of working beyond the

established telecoms vendors. Ideally, an aspiring open source telco would hire expertise from networking companies or from large cloud providers. As these recruits have already been exposed to networking, virtualisation and open source usage, their learning curve could be faster.

In a greenfield organisation, the negative impact of the telco mindset could be minimised from the very beginning if the leaders are free to operate as a software company.

A special greenfield case is the launch of a new company in a new country by an established telco. In this case, there is the opportunity to deploy a modern telecommunications infrastructure, which anticipates technological evolution, from scratch. This case can allow the telco to combine the freedom of a greenfield experience with the possibility to also harness “big company insights” and practical experiences for driving and implementing transformation.

Brownfield model

If an established traditional telco wants to become an open source telco, it will need to address significant issues. One is related to the relevance that the transformation group has within the organisation. Without the continuous support of the top management, the transformation will be hindered by the company culture and it will be difficult to impose.

The group of people to be trained in open source development should be protected from internal politics and given enough time to produce results. In the beginning, the results are unlikely to be very substantial compared with those delivered by the existing (vendor supported) engineering groups already operating in the company. For this reason, such comparisons should be avoided. The transformation to an open source telco requires time, investment and perseverance to be effective.

The transformation needs the clear support of a shared vision within the company about the “Why” and “What” the transformation team should be doing. In this case, the “How” will be much less important. As much as possible, the transformation team should be put in a similar situation to the greenfield scenario. This means that the supporting management has to understand that mistakes and errors will happen and they may impact time and investments. Nevertheless, the top management has to create links with the transformation team and ensure its members have the right levels of commitment and engagement. They represent the possibility for a company to change and move to a better way to do business. The transformation team should feel the importance of its goals, as well as the associated pressure.

The human resources department can play a key role in empowering developers, by favouring a change in mindset towards development capabilities. The HR team need to understand and correctly evaluate the skills of developers and their ability to have an impact on developer communities. To that end, a brownfield company may need to introduce new metrics and new evaluation processes to ease the transformation. We discussed the importance of new metrics to measure digital transformation in our report Telco transformation: The 20 metrics that matter.

The identification of a few easy targets for the transformation team - and their subsequent achievement - should give enthusiasm and prove to those in traditional IT and network teams that the process is working.

Last, but not least, top management may need to be educated to better evaluate the effort and the importance of the software development for the new telco. Ideally, there will be software-related knowledge in the leading team of the brownfield company. Without this awareness, the journey to become an open source telco is very close to mission impossible.

Conclusions and recommendations

Controversial and challenging, yet often compelling

The adoption of open source within established and project-based companies is somewhat controversial. It requires a fundamental change in approach to engineering and to human resources. In recent years, telcos are less and less technological companies and more process-orientated ones. Many IT staff don't have modern software expertise and don't comprehend the importance of software. The approach is usually to rely on an established vendor in order to introduce innovation. Although software is driving innovation, there isn't a push to fully understand and accept this challenge. Moreover, the work of developers can be undervalued compared to that of project managers and controllers.

The traditional separation between network control and network management is still evident in many telcos, meaning the software is not seen as an enabling factor for a business transformation, but it is simply a means to improve a specific sector (the network or the management).

Yet open source solutions can drive softwarisation - the use of virtualisation, cloud computing, open APIs and programmable communication resources to transform the network architecture. The net result could be a better customer experience, leaner and faster processes, rapid innovation, and usually lower costs and prices.

Some operators, such as Deutsche Telekom, have taken initial (and successful) steps towards the softwarisation of the entire infrastructure. This proves that with strong management support for focused projects based on open source components, telcos can achieve relevant transformation objectives.

Still any plan for widespread adoption of open source needs to be carefully evaluated to identify potential hidden costs and the skills that will be needed to complement the telco's existing human resources.

Recommendations for different kinds of telcos

An operator's approach to open source will differ depending on the type of telco. There are also some regional differences that can impact the effective adoption of open source solutions. Generally speaking, small- or medium-sized telcos may be engaged in a hybrid approach, that is to cope with ossification and vendor lock in by introducing fully controllable and programmable solutions based on open source. Relatively small teams of skilled developers can be aggregated to successfully implement effective solutions with a real impact on the market.

Big operators could adopt a larger scale approach, potentially following the playbook adopted by Deutsche Telekom: start in a smaller regional context, operate in essentially a greenfield situation, empower the software developers and allow them enough time to reach objectives, export the skills and experiences to the corporate level. The bravest large operators could decide to address the current needs of the entire network infrastructure and be disruptive. In this way, they could lead innovation, disrupting the old telecoms market.

There are several steps both large and small operators should consider:

- Build a deep understanding of the need to change approach and create a clear (open source) software vision and strategy.

- Instil software skills on the top management and on the basis of the pyramid. Value the developers and provide them the context and the tools to work proficiently. Senior management should back this effort to make it viable.
- Try to progressively increase the number of skilled developers and software experts in order to transform the company.
- Integrate network control and network management into meaningful projects and activities. The difference between the two is blurring and will ultimately be insignificant.
- Have a strategy to decide how to develop software in-house (at least the understanding and the ability to modify it according to new requirements must stay in the company). Outsourcing is still an option, but not for the most important software solutions.
- Engage the developer community, and let it address some cogent needs of the network platform. Telcos need to cultivate people that can create relationships with external developers.

Some of the initial activities and experiences with open source will be difficult and fail, but the skills of employees will increase and in the long run they will make the difference. It is important to regain the ability to develop software within the telco and become less reliant on vendor-driven innovation.

Telcos have to choose whether they want to be innovators (taking the risks of likely failures) or followers and wait for generally available innovation brought by consolidated vendors. When making this decision, the telco should consider what would happen if one competitor is able to fully exploit its development capabilities. Free/Iliad in France has provided a glimpse of how disruptive this can be. The operator's successful Freebox - a single device combining an ADSL modem/router and an IPTV set top box/media player, which was developed in-house - has transformed the market and rival operators have been forced to follow the innovator along this new path.

STL Partners and Telco 2.0: Change the Game

STL Partners delivers original and strategic research, consulting and advisory services for telecoms, technology and media businesses that seek transformational success in the digital economy. It:

- Helps clients create opportunities, make new connections, deal with threats, and drive strategy, plans and effective actions.
- Specialises in changing business models, driving innovation and growth, and is the behind 'Telco 2.0' – the leading visionary benchmark for success in Telecoms.
- Key practice areas include: Transformation; Disruptive Strategies in Communications, Content and Commerce; Cloud and Enterprise ICT; and Future Networks.

To get involved, please call +44 (0) 247 5003 or email contact@stlpartners.com to engage with us through:

- STL Partners Research, which includes the Telco 2.0 Executive Briefing Service, in-depth streams on the key practice areas, and the widely read Telco 2.0 industry blog and newsletter
- Bespoke Consulting and analytical services, typically helping clients evaluate opportunities, develop new propositions and business models, and develop 'go to market' strategies.
- Expert and interactive support for specific engagements with key market contacts and new connections.