A Beginner's Guide to Automation for FTTH Networks

A technical paper prepared for presentation at SCTE TechExpo24

Marco Gagliostro
Manager, PON Technology & Enterprise Network Operations
Rogers Communications Inc.
marco.gagliostro@rci.rogers.com

Table of Contents

List of Figures

1. Abstract

As the Chinese philosopher Lao Tzu accurately said, “the journey of a thousand miles begins with a single step”, and so does the path to implementing automation for your Fiber to the Home (FTTH) Networks. This endeavor may seem daunting, requiring team cultural shifts, architectural choices, the discovery of suitable tools, and the development of new skill sets. As you embark on this journey, you'll soon realize that the benefits outweigh the challenges. Even small steps towards automating your deployments and network operations can lead to significant improvements in efficiency, error reduction, and overall productivity for your company.

2. Introduction

The convergence of Internet Protocol (IP) subscriber management and access networks in our FTTH deployments creates new network management challenges for Operators. As Cable Operators build their Fiber Networks into Greenfield and Service Expansion areas, they must adapt to architectural and design differences, as well as unique operational encounters in their deployment. Broadband FTTH Networks that are built upon Broadband Network Gateway’s (BNG), Policy Charging and Rating Function (PCRF), Distributed Access Architecture (DAA) nodes and Optical Line terminal’s (OLT) serve a purpose-built function that require the proper orchestration and automation to ease some of these challenges. The predecessors of 10 Gigabit Symmetrical Passive Optical Network (XGS-PON) technology and traditional Element Management Systems’ (EMS) continue to be grandfathered from older technologies (Gigabit Passive Optical Network (GPON) / Digital Subscriber Line (DSL)), which shows their age and lack of flexibility with current automation advances. The future is moving to Software Defined Access Networks (SDAN), to modernize the platform and develop the benefits of Software Defined Network’s (SDN) for Broadband Access Networks.

The race for higher bandwidth is still present, but more than that customers are looking for reliable, highly available, easy to use, and lower costs for their connectivity experience. Your organization’s quest and evolution of your Network Automation Program will translate to better network management, agility, cost, reduced employee friction, and an overall better experience for your customers. Adopting a culture of Development Ops (DevOps) for Network Operations, known as DevNetOps, can accelerate your automation journey. by employing technology and using elements of processes such as Agile, LEAN Six Sigma and Software Development Life Cycle (SDLC) to help you shape and modernize your Network Operations.

This paper is broken up into 3 parts: The first part will share drivers of automation, the cultural considerations and need for clear goals and objectives. The second part will highlight key processes that help to supplement your automation framework. The third part will discuss design considerations and FTTH Automation use-cases. This guide serves as a beginner’s starting point for managing your FTTH networks.

3. Background

According to Precedence Research, the global network automation market is projected to experience substantial expansion, reaching a value of $28.63 billion by 2032 (Figure 1). This growth is driven by the increasing demand for efficient and scalable network management solutions. While still in the earlier stages of the lifecycle it means there is substantial innovation and exploration that will occur.

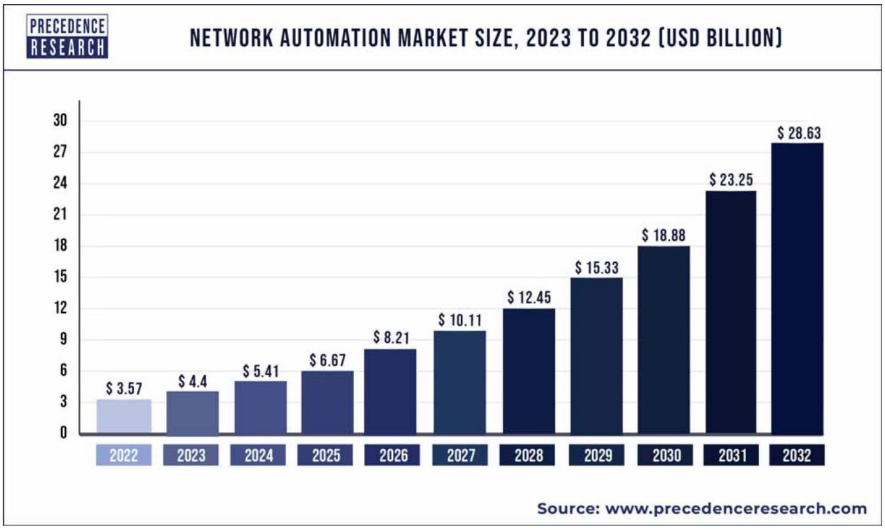


Figure 1: Growth of the Network Automation Market

4. Architectural Reference:

Throughout this document, we will primarily reference the architecture noted in Figure 2 unless otherwise specified. This architecture serves as a foundation for the examples. This reference architecture is characterized by either directly connected OLTs (via directly connected fiber) or using a DAA network for OLTs to connect to the BNG Router, as well as a Traditional EMS/OSS (Operational Support System) backend and PCRF application for handling subscriber data.

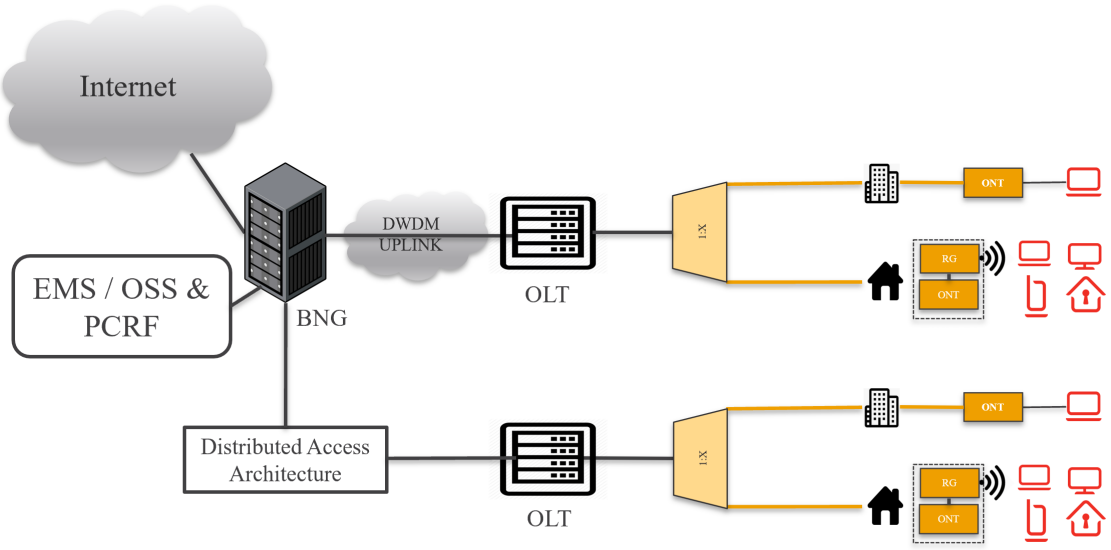


Figure 2: Block diagram of network architecture

5. Legend:

Residential Gateway (RG): Residential Gateway is the Customer Premise Equipment that connects to the WAN and acts as an Internet Gateway for Residential Connectivity Services.

Optical Network Terminal (ONT): The ONT is found at the Customer Premise and provides an interface to the PON Network and to the Residential Gateway. This could also be installed in the RG via a Small Form-Factor Pluggable (SFP) based ONT.

Optical Line Terminal (OLT): The OLT is the central access node where your ONT's connect to a physical PON-based connection. Splitters are connected to your OLT PON Ports to deliver up to 128 customers on a single PON port. This access node comes in several form-factors, including a full-fledged modular or fixed OLT chassis, Clamshell-based OLT, or Pluggable SFP-based OLT that utilizes the switching/routing fabric of a host Network Device.

Distributed Access Architecture (DAA): Provides an IP/Ethernet distribution of the access layer, moving it closer to customers to improve fiber utilization, lower costs, and enhance resiliency. DAA consolidates technologies like DOCSIS® and PON, typically using a spine-leaf architecture. It connects to BNGs for OLT connectivity and can also provide connectivity which enables BNG redundancy to offer better resiliency and availability on your BNG layer.

Broadband Network Gateway (BNG): Provides Internet Routing, Subscriber Authorization and Authentication, IP Assignment, Quality of Service (QoS) and overall subscriber management services for your single, dual, and triple-play services across your network.

Policy Charging and Rating Function (PCRF): This is the application that uses Diameter protocols for authentication, authorization, and accounting, and maintains several key functions such as: enforcing quality of service (QoS) rules, managing data usage policies, and handling real-time charging decisions.

Preface:

This guide assumes that you have business alignment in your pursuit for automation.

6. Culture

Culture is an important consideration for your Automation journey. While culture touches all aspects of your business, beginning your Network Automation journey will force you to go beyond "*This is the way we always do things*" and into areas of uncertainty, exploration, and discovery. The way to organizational/team growth is not to stay comfortable, but to adapt to new and improved ways of working and problem-solving issues that will have the biggest benefit on your bottom-line.

Webster's dictionary defines culture as: *"The set of shared attitudes, values, goals, and practices that characterizes an institution or organization."* These are enforced by a set of behaviours that your company and team demonstrate on a consistent basis. These behaviours do not occur in grandiose fashion, but in small increments, which compound and develop into something more substantial over time. Building culture is much more than just talking about it. It's defining what your team believes in and how you behave. It's a competitive advantage over your competitors who are struggling in cultures of constant blame and no accountability for outcomes.

6.1. Automation Objectives and Vision

It's vital to ensure your automation efforts are aligned with your company's strategic direction. Automation is the mechanism that you will use to achieve business and operational excellence for your Broadband FTTH network. As a participant and/or leader of your newly started automation journey, it's important that the team understands the objectives and relevant alignment towards the goals of the company. The SMART method (Specific, Measurable, Achievable, Relevant and Time-Bound) is the recommended method to use for objective setting for your automation endeavors. The strategic alignment of your goals to the overall company goals needs to be broken down so the team understands what's tangibly required on their level to achieve organizational goals. It also provides a roadmap for the team to work in unison while reducing tension due to competing goals across the organization. Once you've formulated your goals, share them with relevant parties. Collaboration flourishes when teams share a common vision and invest in each other's success.

Guiding principles or team charters can help to build a common vision and break down responsibilities and behaviours (The **What** (Goal) and the **How** (Success-Measure and Behaviour)), are effective ways to develop this for your team. It's imperative that all team members are part of creating these – written in a simple and concise language, and further continue maintaining these on a regular basis as the team re-focuses and priorities change. These are the core behaviours that you will build as norms within your team. It'll be imperative that all team members take part in keeping everyone on the team accountable for those guiding principles and responsibilities. Recognition and acknowledgement for those that are exhibiting these qualities should be shared on a regular basis to enforce these norms and emphasize their importance within your team.

6.2. Drivers of Automation

When people think of automation, they immediately think of the cost savings that are achieved; however, the benefits extend far beyond financial compensation as they have ripple effects across your organization in terms of productivity, customer satisfaction, and network health.

We'll examine the key factors that propel our automation efforts and the positive results they yield through applicable examples:

- **Support business needs and agility:**
  - *Your Product team wants to launch a new 8 Gigabit Per Second (Gbps) Speed Tier for XGS-PON. Automation tasks may include the new configuration deployment for your OLT, as well testing to ensure your configuration is functional and compliant to your design.*

- **Deploy new infrastructure more effectively (Provisioning):**
  - *Your organization wants to reduce the error rate and complexity when installing new OLT's in the field. Zero Touch Provisioning (ZTP) with cross-domain orchestration allows you to provision all elements of the OLT and associated devices consistently and accurately the first-time reducing re-work.*

    (NB: Zero Touch Provisioning is like Plug and Play provisioning, but for your OLTs. It simplifies the setup process by automatically configuring devices with minimal manual intervention)

- **Increase Quality, Consistency and Security:**

  - *On your BNG routers you maintain configuration for Control-Plane Protection and Security Access-Control-Lists (ACLs) that ensure that your routers are stable and secure. Automation can safeguard that you have configuration compliance reports that tell you if anything has changed without your awareness (i.e. through an incident ticket or unauthorized change), or if there was a mistake in the ACL entry that could expose a security vulnerability on your router.*

- **Improve Data Collection and Processing:**
  - *You have telemetry data from your RG, ONT, OLT, DAA nodes, BNG, DHCP and other platforms. You have too much data to process to make effective decisions during an incident or from post-change activities to assess an activity's success or failure. Automation and reporting capabilities will help you consolidate and analyze large amounts of information more effectively.*

- **Reduce repetitive work:**
  - *As part of your preventative maintenance checks, you manually log in to nodes to run specific data collections. Using automation, you can turn your preventative maintenance work to threshold triggered and only spend time to look at things that matter.*

- **Re-focus skillsets on complex work:**
  - *In the age of knowledge-based workers, they can focus on items that cannot be done (easily) by automation. They can focus on addressing technical debt and recommending or creating new automation of routine tasks.*

Addressing automation around these key drivers will increase your competitive advantage, reduce your network risks, and provide a better experience for your customers who will obtain the benefits of success implementation of your automation initiatives around these areas.

7. Process Improvements

7.1. Agile

A group of interested parties came together in 2001 to develop Agile Values. These were tailored to the software industry, but soon spread to other businesses based on their practicality and ease of understanding. Some of the Agile principles are quite useful to understand for your Automation journey. The Agile manifesto provides an introduction into some of the values that are most important within Agile.

The Agile Manifesto:

We are uncovering better ways of developing software by doing it and helping others do it.
Through this work we have come to value:

- *Individuals and interactions over processes and tools*
- *Working software over comprehensive documentation*
- *Customer collaboration over contract negotiation*
- *Responding to change over following a plan*

Agile project management practices emerged as a response to the shift of industrial-based work into knowledge-based work. Waterfall methodologies are well-suited for projects with clearly defined requirements and a linear workflow. Agile is generally used on projects with greater uncertainty.

Although in many organizations both approaches are used and understanding both are important. Agile's iterative and adaptive nature makes it work well around Automation, which will sometimes require discovery into network integration and programmability solutions to solve complex problems.

7.2. Key Agile Principles in Action

As Operators look to deploy FTTH Networks to complement their DOCSIS ® Networks, Agile practices lend themselves to DevNetOps and Automation initiatives quite well. While all 12 Agile Principles have value there are some key principles to highlight that can help you:

Welcoming Change.

In welcoming change, Agile looks to provide the maximum value to its end customers. Agile recognizes that change is inevitable, issues arise, and your teams need to be flexible to maneuver these in an effective and efficient fashion. If your organization is new to FTTH, there will be things you get right and wrong and your ability to change throughout will be paramount to your success.

Feedback Cycles.

Consistent feedback loops allow end users to provide feedback in collaboration of your design cycles so that the programmers can achieve value-driven delivery. As automation development occurs, assumptions and interpretations are made that can lead to something being built that doesn't meet the intended purpose. Imagine the lost productivity working on something for two months with no feedback only to find out that it was done all wrong? Avoid working in isolation and resisting feedback. Collaboration, transparency, and open and honest feedback cycles make your product better. Demonstrate your progress and share with stakeholders often to ensure that it's meeting their requirements.

Iterative and Continuous Delivery of valuable software.

Agile focuses on time-boxed value phases (iterations) that provide the business with a tangible benefit and used while allowing the team to maneuver through the full complexities of a project. During each phase, planning occurs, and each iteration looks at providing value through defining the requirement (goal) of that iteration, building required artifact/tool, validating, and releasing something that provides a value to the overall project goal. Launching technologies such as FTTH (XGS-PON and beyond) can introduce areas of uncertainty and thus an iterative approach can help the project team adjust to changes while still deploying items in a consistent fashion. Agile uses frequent planning cycles despite that myth that Agile means less structure or planning (then Waterfall) is untrue. Frequent planning helps in the pursuit of addressing issues early and taking necessary steps to avoid roadblocks that prevent completion of your project.

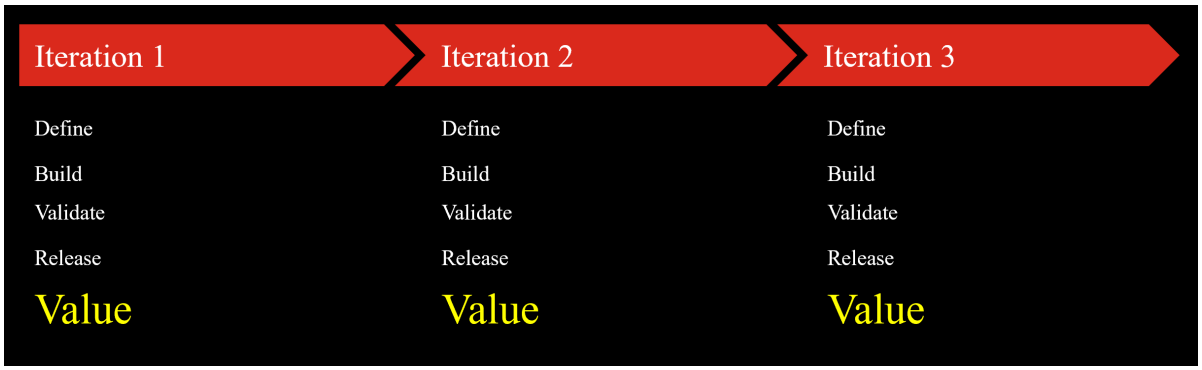


Figure 3: Time-boxed Value Phases (Iterations)

One of the most critical aspects of project managing complex projects is the ability to detect and resolve issues earlier in the project lifecycle. The iterative approach to planning and development helps you recognize, manage, and mitigate risks quicker to ensure they don't end up causing your project to fail. Consider the figure below (Figure 4), the cost increases and your opportunity to adjust decreases as you get closer to launch/operational readiness. The amount of re-work needed at later stages demoralizes the project team and affects the ability to get value out of the project. The last thing that we want to happen is for the project team to become complacent with the quality and focusing on just making it work to complete the project, which ultimately impacts Agile's customer-centric approach to welcoming feedback and building in value iterations.

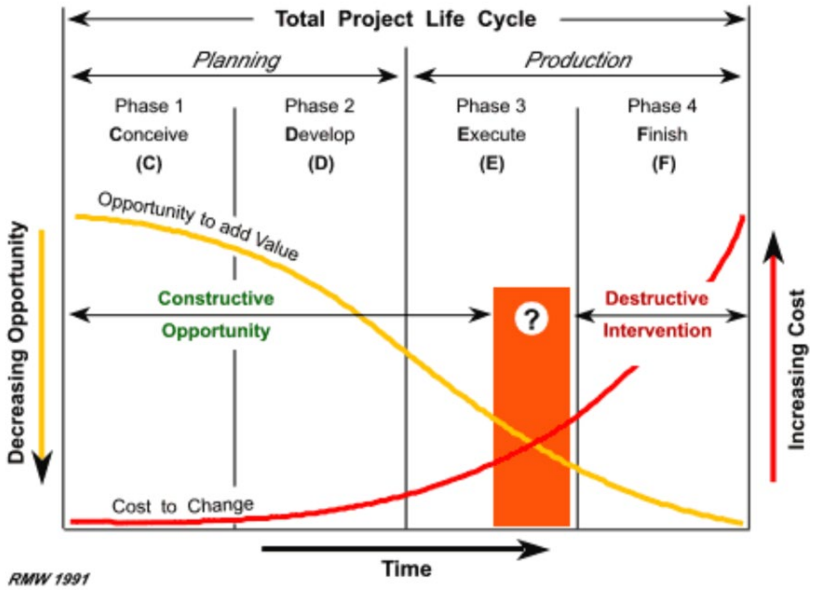


Figure 4: Total Project Life Cycle Cost and Opportunity

Continuous attention to technical excellence and good design enhances agility.

Throughout your journey you want to ensure that you learn to build specifications (guidelines) for code design, code reviews, testing and deployment. Some examples of areas that you will need to investigate include:

- Consider Version Control System (VCS) to manage your code. No different than configuration management tools for your OLT, DAA and BNG node configurations. This allows for your code to be versioned – changes tracked, and development visible to team members. If there are network configuration changes needed, the Automation Specialist can 'branch' off the current configuration and start development on new features. Later those features can be peer reviewed and merged into the main configuration as part of your release and deployment processes.
- Fix Software Bugs, ensure that you have a bug repository to track bugs. If your code doesn't get fixed, it will be discarded as a valid tool and improvements lost. Stay active on fixes and features.
- Acknowledge and rectify previous development shortcuts to prevent technical debt from escalating. Sometimes we don't optimize our code for the sake of getting immediate value. Later we find that refactoring (optimizing) the code will make significant performance benefits.

Manage your technical debt in focused iterations or it will accumulate into a difficult to manage distraction, leading to potential performance or functional issues with your code over time.

- Develop automated unit-test (tests for a certain function area of your code) as well as end to end testing capabilities using automation systems. Ensure your success criteria are fully satisfied before putting it into a production environment.
- Modularity of code is important for Network Engineers. Building libraries and functions with specific purposes that can be re-used in other automations in your company can decrease development time and standardize the procedures involved.

By embracing some of the Agile principles in your DevNetOps practices, you will help to provide a sound methodology for delivering automation programs that enhance customer satisfaction and drives significant value for your organization.

7.3. LEAN Six Sigma

LEAN refers to a set of practices aimed at eliminating waste and maximizing efficiency. It focuses on delivering the most value to customers with fewer resources. LEAN thinking emphasizes continuous improvement and eliminates anything that doesn't add value to the final product or service. This is a natural fit for DevNetOps and Automation.

8 wastes:

- **Defects**: Products or services that don't meet the quality standards for your company.
- **Overproduction**: Producing more than is needed.
- **Waiting**: Idle time due to delays.
- **Non-utilized Talent**: Underusing employee skills.
- **Transportation**: Unnecessary moving of materials or product.
- **Inventory**: Having excess stock.
- **Motion:** Unnecessary movement by people or equipment.
- **Extra Processing:** Unnecessary steps with no-value-add.

Likewise, Six Sigma looks at process improvements to eliminate defects and errors in processes. Together they provide a well-rounded approach to looking at the underlying process, procedures to improve the output and performance of the company.

Six Sigma uses a data-driven approach with a defined structure (DMAIC) for improvement projects:

DMAIC is an acronym for the following:

- **Define:** Identify the problem or opportunity.
- **Measure:** Collect and analyze data to understand the current state.
- **Analyze:** Identify root causes of defects or variations
- **Improve:** Implement solutions to address the root causes.
- **Control:** Monitor and sustain the improvements.

Six Sigma looks to reduce variation and improve quality of your outputs through exploration of your processes and is a key component of your automation program for areas that have high defects which can be solved through automation.

7.4. Applications of LEAN Six Sigma

7.4.1. Process Mapping

Process mapping provides a visual representation of a process, allowing stakeholders to understand its contents, mapping out steps, decision points and handoffs involved. The visual clarity promotes collaboration for team to design more effective processes that reduce friction and increase output.

For FTTH Networks, it's necessary that you have clear processes and workflows on several of the following:

- End to End OLT Turn-Up Process, including new PON Segment expansion.
- New Speed Tier Creation. (Fulfillment)
- New Subscriber IP Block additions based on DHCP usage and trend reporting.
- Performance Management (Impairments and Repair)
- Certifying new OLT or ONT software release.
- Capacity Management (PON Node Capacity segmentation and OLT / PON migrations).
- Addressing last-mile record discrepancies to your fiber plant and physical labelling.
- Managing configuration changes for your BNG, DAA, and OLTs. Transitioning from design, lab testing, to implementation, including how your automation/orchestration tools will be updated as part of this new configuration change.

These are in addition to your companies Incident, Problem, Change and Preventive Maintenance processes.

Understanding and mapping your processes (or what they appear to be) will help provide a starting point to any automation project. What issues are you trying to solve? Are you solving the issue that will provide the most overall benefit to your process?

7.4.2. Value-Stream Mapping

Once you've visualized the process, Value-Stream Mapping documents the value-add vs. non-value add steps in your process.

- Value-Add: Time that directly contributes to the completion of the task.
- Non-Value-Add: Time spent on activities that do not add value to the process.

This shows the balance between tasks and the value that each instills in providing the desired output. Value-stream is useful for identifying areas of automation that could help you deliver more effectively.

Consider the following simplified example (Figure 5) of staging a new Remote-OLT (R-OLT) and completing the configuration required to make it operational. While the process cycle efficiency (valued added time / total time) shows room for improvement, it now forces us to explore and find potential defects/waste. A high process cycle efficiency indicates the process is efficient, with less non-value-added activities. A lower process cycle efficiency suggest that are improvements that can be made to reduce waste and increase efficiency. Now we start to question: why is there 40 minutes between the last two steps? This may illuminate a defect/waste that you want review to improve the productivity of your flow. Finally, we ask ourselves, can automation solve this problem and if so, how. This is a worthwhile venture for all key processes for FTTH.

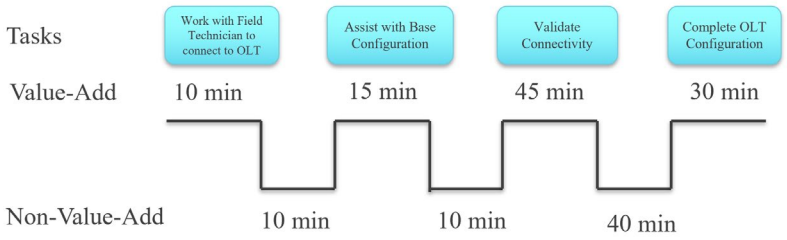


Figure 5: Value-Stream Mapping (R-OLT Turn-Up)

Calculating Process Cycle Efficiency:

How much time provided value add: 100 min.
Non-Value Add: 60 min.
Total Cycle Time (Time from Start to End of a process): 160 min.
Process Cycle Efficiency: 100 min/160 min = 62.5%

7.5. Breakdown of the 8 Wastes

We will discuss some of the 8 wastes in relation to some of the things you may see within your FTTH Network. We examine some examples and potential improvement areas through automation or process improvements.

Transportation: Unnecessary moving of materials or product.

During the turn-up of a new OLTs, your Field Technician is e-mailing other staff member to obtain required turn-up data related to staging this OLT.

Potential Automation Use-Cases to Address:

- Engage in Zero Touch Provisioning to reduce human interaction while and simplify the overall process.
- Centralized data-repositories/asset-management tools that provide the necessary visibility to data required. Enable self-serving capabilities through a Chatbot or Wiki Page to assist with this.

Inventory: Having excess stock.

Using a software tunable Dense Wavelength Division Multiplexing (DWDM) SFP transceiver for your OLT uplink vs. a fixed channel SFP, will allow you to reduce the number of inventories you hold for your deployments and sparing depots. It will also simplify your automation to maintain a consistent approach and not have to build additional decision points, constraints, or configuration changes to account for minor differences.

Potential Automation Use-Cases to Address:

While the output of this use case is not automation per-se, it will have an impact on your automation initiatives. Simplifying your deployments in terms of the number of disparate part numbers (or revisions) will lessen the amount of time needed for regression testing and addressing configuration differences in your automation. Where possible simplify your product choices to avoid having to account for differences in the configurations and holding excess inventory.

Motion: Unnecessary movement by people or equipment.

Sending a technician to a customer site to validate Internet speeds to ensure they're attaining their subscribed speeds.

Potential Automation Use-Cases to Address:

- Embedding a Gateway Speed Test application on your CPE (RG or ONT), which is ingested into a common reporting/alarming dashboard.
- Build out several dashboards for your customers service experience. Speed issues may be resultant of other impairments (Signal degradation, Optical Power Level, Capacity, Upstream Issue, etc.) that need to be exposed and visible. Data collection and appropriate flags can help identify issues and steer the technician to the right place.

Waiting: Idle time spent waiting for resources.

If you're using manual provisioning workflows for turning up your OLT, you will have dependencies on those others configuring your DAA or BNG for connectivity.

Potential Automation Use-Cases to Address:

- Zero Touch Provisioning will reduce the number of people required to do the work, increase success-rates, and avoid waiting/idle time caused by manual processes and procedures.
- To provision your DAA network and/or BNG for that OLT, as well as associated inventory records, you'll need to have a cross-domain orchestrator to provisioning this flow.

Overproduction: Producing more than is needed.

Your team oversees the logical configuration on the BNG or DAA network for each OLT. Your team is resource constrained and your configuration is not automated in any fashion. Your team is being asked to pre-configure 50 OLTs on these devices with varying required in-service dates ranging from 5 days to 3 months for OLT turnover.

Potential Automation Use-Cases to Address:

- Overproduction can be the result of not following Agile principles. In the case where you have limited staff, they need to be focused on extracting immediate value and not putting effort into pre-provisioning devices that are 3 months from completion. This approach, which is like Just-in-Time Manufacturing (JIT), can be used to alleviate over-producing.

Extra-Processing: Unnecessary steps with no-value-add.

Your team is copying values from one spreadsheet to another to capture the data for another team.

Potential Automation Use-Cases to Address:

- Use an intake method where the data is entered once, validated by a system, and then goes to your provisioning team for implementation.
- Utilize structured data methods like JSON (JavaScript Object Notation) and YAML (YAML ain't Mark-up Language) to help to build inputs into your automation, which can also be used as a trigger for other workflows and avoids moving data around unnecessarily.

Defects: Products or services that don't meet the quality standards for your company.

Your team misconfigures a new IP Subscriber block, which now means you're handing out invalid IPs to customers that are unrouteable. Your call center is receiving calls and sending truck-rolls to customer sites that cannot be resolved by your Service Technicians.

Potential Automation Use-Cases to Address:

- If this is a result of human error, then you can use automation and programming constraints to ensure that the input configuration meets the criteria that you provide so that you can catch some of the common mistakes E.g., the input must be within the following IP Block (x.x.x.x/netmask to y.y.y.y/netmask) only. Further you can conduct automated steps to validate other logic/syntax faults on your IP Management platform and BNG routers.
- Build automated test plans (post-checks) to confirm that these IP addresses are routable after your change activity.

Non-Used Talent: Underusing employee skills.

Your team has resource bottlenecks when conducting important ONT upgrades in the network. The complexity of ONT upgrade execution is low compared to other tasks that are done daily.

Potential Automation Use-Cases to Address:

- While not automation per se, Agile teaches us that specialization in a certain area can lead to bottlenecks due to waiting for resources to become available. For tasks like ONT upgrades, employing generalists can help create a larger pool of resources and avoid these constraints.
- Related to Automation, by having automation the routine and less risky work, your staff are freed to do things that bring more value to the team, like optimizing your network design, reviewing Problem Management/Chronic issue review, or fine-tuning your event management practices.

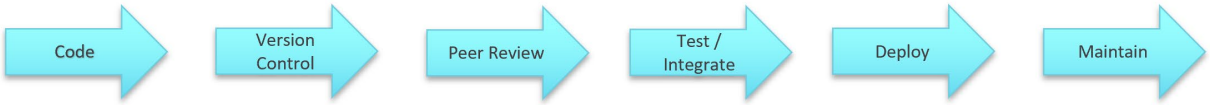
7.6. Introducing DevNetOps

DevNetOps has its roots in DevOps, which is used for software engineering, and applies those methodologies and practices into Network Operations. It's a movement that's not just about technical excellence, but about breaking down silos, increasing collaboration, and optimizing delivery of network platforms and services. DevNetOps is built using concepts from LEAN, Agile and SDLC (Software Development Lifecycle) and introduces the concepts of Network as Code (NaC) and NRE (Network Reliability Engineer) to our technical vernacular.

7.6.1. What is Network as Code

Network as code is the process of codifying your network devices and configuration and enveloping them in the same rigor and processes that application developers would go through. Taking the best processes from software developers, we can leverage this in a Network Operations setting.

The following illustrates a high-level flow of DevNetOps Network as Code principles. Code (Network Configuration) gets placed into version control, where changes to that configuration can be introduced, changes are peer reviewed, tested and integrated into the lab, deployed to production and finally into the operational lifecycle (bug fixes, maintenance releases.)



7.6.1.1. Integration Considerations for Network Operations

Using Network as Code as a concept, what should we consider when utilizing these network programmability and software/coding best practices in the realm of Network Operations.

- Implement error-detection and handling using a layered approach. Create the necessary guardrails for safe network execution.
- Use secure methods for storing your code and passwords. Follow your organizational practices for data security and privacy. You want to ensure your automation hasn't resulted in new security vulnerabilities and need to test for such.
- Review release notes and alerts from the vendor to ensure there are no current issues related to the automation and executions that you're attempting to do. Newer automation features are less mature than standard (classic) Command Line Interface (CLI) and may have software bugs that risk the network's stability.
- Test your code in lab, deploy in pre-production environment (no live customers) and re-test during your first deployment with real customers, before completing changes en mass. This stepped approach will provide you real data that can help you advance on your deployment with confidence. During lab testing you also need to test your rollback capabilities to ensure that if a problem arises, you're able to restore the network quickly with minimal impact to your customers/network.
- Establish your criteria for success is clearly defined. If you have this documented and well-understood, you would be able to implement auto-rollback capabilities in your code in the future.
- Develop using libraries and modules for common tasks to avoid re-creating code and developing new configurations for something that doesn't warrant it. E.g. Establishing an SSH (Secure Shell) login and passing CLI credentials to the node.
- Utilize the networking developer community to shorten development cycles, increase understanding and limitation. Many vendors such as Cisco, Nokia, Arista, Ciena, Juniper have vast communities/forums dedicated to automation on their platforms.
- Develop team (or company) standards for automation. For example, where are network configurations stored, how repositories are updated/maintained, where's documentation stored? What tools can be used? What scripting/programming languages are supported? How are bug fixes tracked? Etc.
- Make sure that documentation is available for understanding and using automation to its fullest capability while completing the right level of documentation without 'over-producing'.
- Spend ample time to maintain bug fixes, new features, regular maintenance, and code optimization. Establish them as regular business-as-usual processes.
- Incorporate an automation first mentality from the on-set of any new Technology/Design/Feature or change in your network.

The need for Network Engineers to incorporate elements of software development are part of the new landscape and next generation of Network Engineers. This doesn't forego the need for strong networking capabilities, but recognition of where the industry is and where it will continue to develop.

7.6.2. Perspectives in Automation

Depending on the perspective you have within your automation journey, you will inevitably have different considerations that you should think about.

As a leader:

- Not surprisingly some team members will be threatened by automation. It's change and ongoing risk to current comfort levels. In fact, people have found their own manual shortcuts for routine, mundane tasks, and believe it's the best way. You must be persistent in your approach to show the benefits and value of automation and what is brings towards their day-to-day work. For those that resist automation, keep them updated and ask for their feedback along the way. Even better than this, as you develop a culture of automaton, your team will incorporate this as part of their norms, self-manage these ideals across the entire team, and look at ways to continually improve everything they do.
- You must be able to show your own vulnerabilities and express a keen sense of learning new things. It will be infectious to your team. You may not have done programming or worked in a DevNetOps environment, so share that and be focused on learning, absorbing, and enabling your team to grow. While your role authority will always be present, try to break down the walls and show them that it's a level playing field in this new pursuit– everyone's input matters equally.
- Allow for experimentation in a sandbox. With any new exploration there will be stumbles along the way. Learning and sharing are key in this. Have your team share what they have done, what they succeeded or failed at, and what they learned for next time. Build this dialog through lightly structured automation sessions where team members can demonstrate live-coding and allow for questions throughout.
- Build an exceptional continuous improvement process where blame-less root-cause analysis (RCA) can occur and lessons are shared, documented, and used for future improvements. Google Site Reliability Engineering is an excellent source of information as to how to structure your Post-Mortem discussions (Google - Site Reliability Engineering (sre.google)) Of course, a lot of this is rooted in having a strong company and team culture that is built upon a foundation of trust and accountability.
- Find ways to measure your team's success with automation. How much time was avoided to work on other important items? How many more OLTs were turned up? How many incidents were diagnosed using automation? How many were resolved? Etc.
- Work with your team to re-design your Architectural/Engineering practices to take an automation conscious approach. Evaluate new platforms with an eye for programmability, inter-operability, operating-cost savings, and features that easily integrate into your new DevNetOps tools and processes, amongst the pure technical specifications.
- Work with your company to strengthen your LEAN Six Sigma, Agile, and DevNetOps practices. It will not be a perfect fit everywhere but take those principles, understand them, implement what makes sense, discard what doesn't, and adjust as required. Finally, become an advocate for these, and help to promote positive changes within your organization.
- Learn to develop new talent. It is rare to find strong network engineers that also have rich experience in network automation. Luckily this is becoming less of an issue in recent years but build a strong onboarding program and regular development check-ins to ensure that key skills are progressing and being utilized.
- Understand that automation is a slow evolution – it's not a short-term strategy, but an integral part of business going forward.

- Work with your team to create and prioritize a list of automation opportunities with the expected cost avoidance and time-savings that will be achieved. Highlight those opportunities that solve your teams' biggest hurdles and timewasters.
- Establish the right processes to protect your network (and staff). This includes process and guidelines for transitioning your changes through the DevNetOps tools and methods.

Leaders may not be the experts in automation but must learn to establish the right environment, urgency, focus, and staff development on automation while maintaining the necessary boundaries of safety and stability over your network. They must also ensure that automation initiatives align with the business objectives of the organization.

For those that are Network Engineers and want to contribute to the automation programs what items should you consider?

As eager network engineers:

- Read and participate in automation forums and communities – asking a lot of questions and answering for new engineers. Understand that the team is made of a series of puzzle pieces, and everyone has a different piece that contributes to the sum of the whole. Some will be good at scripting or building pseudo-code, others will have ability in process development and enhancements. Recognize (and appreciate) the value in others' role towards your automation initiatives.
- Document your network service flows and performance KPIs. You must establish a deep understanding of your network (protocols, flows, functions). Likewise, you need to understand the telemetry and key measurements and capabilities from each platform. Figure 6 and Figure 7 show the use of documenting these to better understanding how to extract the right datasets from your equipment to guide and make data-driven decisions. In equal light it's fundamental to understand the correlations and relationship between your data so that you can extract valuable information.

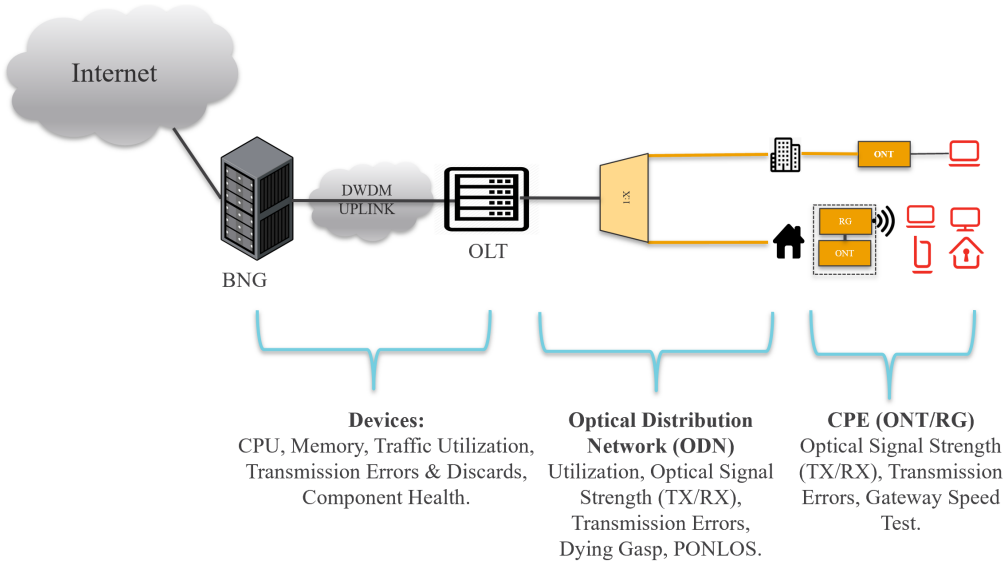


Figure 6: Network Platform KPI (High-Level)

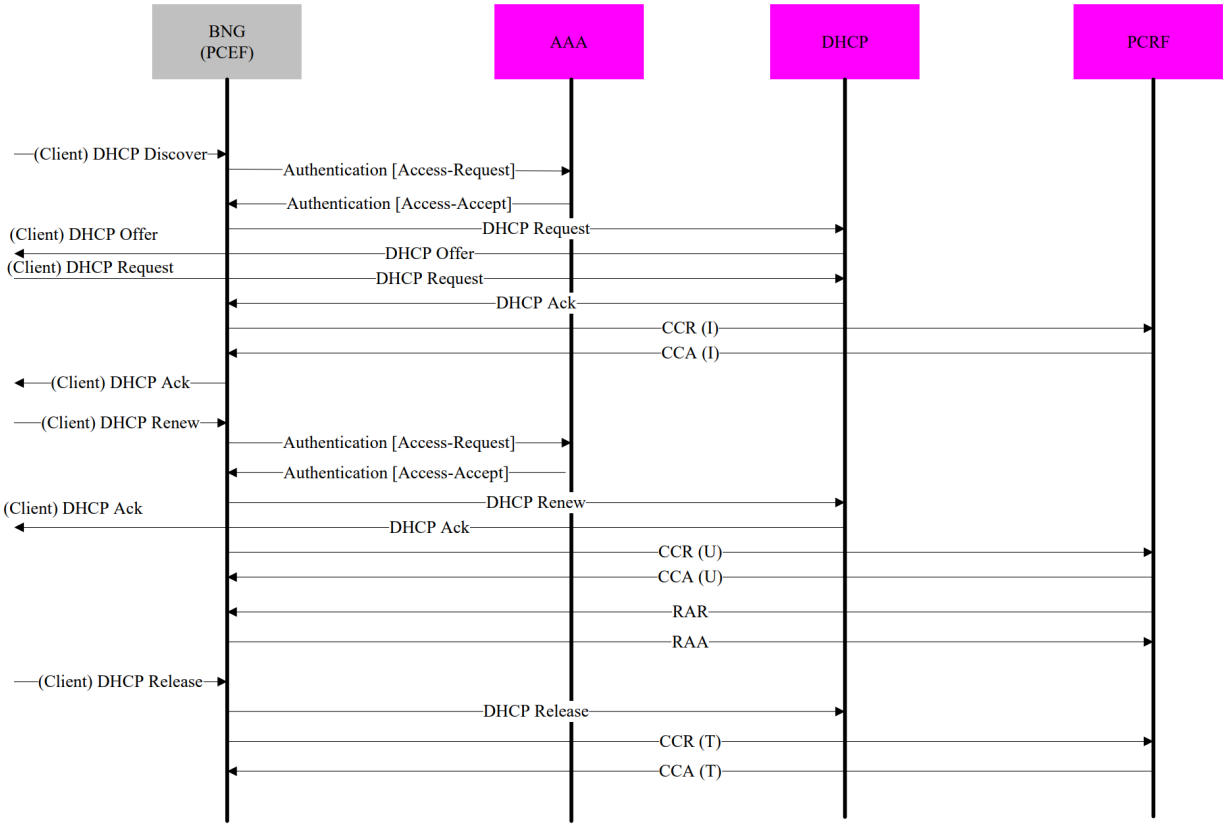


Figure 7: Service Flow Diagram: RG Obtains IP from DHCP Server

- Continuously develop key new skill sets, whether it be on the process track or the programming/automation track. Find your niche area and exploit it. Recognize that you will need to continue to balance skill sets between strong networking and process and/or scripting/automation capabilities.
- Find small tasks and look for creative ways to automate and make it part of your day-to-day work. Share successes and failures with your team.
- Become highly involved in creating the culture around you. Hold your teammates to a high expectation based on agreements that you made through the setting of your objectives, guiding principles, or team charters.
- IMPORTANT: Ensure that you're adhering to your company's security practices and privacy policies. Practice safety first mentality when working with automation. It can accomplish remarkable things but can also do a lot of harm if you do not fully understand the expectations of running it.

8. Automation Framework

8.1. Laying the Ground Work

You've looked at your culture and established clear norms and values that you wish to promote. You've implemented a LEAN Six Sigma program to uncover opportunities, drive continuous improvement and reduce friction in your processes and workflows. You've received help from understanding Agile

methodologies to work in iterative fashion that is customer-centric and works on bringing immediate value to your business. You've looked at how DevNetOps can help you structure your automation. These automation building blocks serve as the starting point for your automation journey.

NB: I will refer to "pipelines" throughout the following section. Pipelines are a crucial aspect of DevNetOps, ensuring that the automated processes for building, testing, and deploying network infrastructure are efficient and reliable.

8.1.1. Automation Building Blocks

Lab Environment

Building a scalable and easy to use lab environment where we can test our automation prior to launching into production environment is a necessary part of your automation journey.

Key considerations –

- Review the use of virtual simulator(s) that can be used to confirm the execution of scripts/automations, and new features. This doesn't exclude the necessity of a physical lab for testing hardware capabilities. In most cases it will be a mix of the two with different purposes and use-cases.
- Your lab environment, while not production, should still use diligence and production-like practices, such as version control, pipeline management and change management.
- Document the environment and set up practices to support and maintain the nodes.
- Review tools such as GNS3, EVE-NG, and Containerlab (to help implement virtualized lab topologies). Using containerized versions (Container Network Function or CNF) may allow you to build, tear-down lab environments as needed for further flexibility.

Configuration Management Framework

This is the building block that manages your devices and enables automation to your network elements in a simple, straightforward, and mediated fashion. It should be built on safety/stability first, but easily scalable and flexible to meet your ever-changing requirements. More than likely your tool-belt will have several tools to deal with different situations and platforms that you encounter.

Key considerations –

- Review tools like Ansible/Salt/Chef and Puppet to decide what tools to use to configure your various network elements that can provides flexibility and interoperability between legacy and newer platforms depending on their connectivity options.
- Several of the large vendors openly share their automation interfaces, whether it uses Application Programming Interface (API) methods or uses Remote Procedure Call (RPC) for connectivity and eXtensible Markup Language (XML) to issue commands or gather data. In addition, many vendors offer Python scripts (on-box automation), within the node to perform specialized tasks, or off-box automation to connect to your configuration management systems.

Network-As-Code (NaC)

This is using Software Development principles into the context of Network Operations and specifically Network Configurations. This means your network configurations are put into a version control system (VCS) and allows you manage your configuration (or versions of) through a separate system. As an

example, you can split your current configuration into a new development workspace, so you can work on it without interruption of your current "golden" configuration (or standard baseline configuration). Once it's taken through your automation processes (approvals, testing/validation, and implementation), it can be integrated into your nodal configuration template as your new 'golden' configuration. VCS will assist in ensuring the integrity of the configuration and templates are maintained, organized, and changes transparent to all authorized users.

Key considerations –

- Develop coding standards, specifications, and procedures on how to utilize your version control system for updating modifying or deleting network configurations.
- Create the procedure of placing your configurations and templates into repositories. Use your automation configuration management platform to execute scheduled network configuration collections.

Pipeline Management

Pipeline management is the process of scheduling, automating, and managing your automation. This provides ways to confirm the status, provide real-time input (like passwords or attributes) and report on the success or failure of your automation. The two most popular are Jenkins and Gitlab.

Network Provisioning Pipeline:

Network provisioning is the function of configuring net new nodes (OLT, BNG, DAA) with your standard configuration template and integrating them into your production environment. Managing your Golden Template requires discipline to ensure your deployments are accurate and through your standard template are pushing consistent configurations that are less error-prone and maintain solid standards for security and configuration.

Key Considerations –

- If you're using CLI-based automation, considering creating your Golden Template in a templating language that can be called upon by your automation.
- Network Provisioning Pipeline should be in-tune with your network asset-management systems and work-flow systems to ensure that key provisioning data is passed to the automation.
- Consider Zero-Touch Provisioning to eliminate a lot of the manual/repetitive work required for turning up new nodes (where possible).

Network Configuration Pipeline:

Create pipelines that can be used for deploying configuring changes uniformly across your network– safely and effectively. For your FTTH network, you will have several requirements to upgrade OLT software, update global configurations on your BNG, and make consistent adjustments across a potentially large access network. These cannot be done manually, so the Network Configuration Pipeline deals with the production network and modifying, adding, removing configurations.

Key Considerations –

- Put safeguards in your code so that it doesn't cause impact or use automation frameworks or vendor tools that have inherent safeguards built into them. This pipeline needs to work within the configuration that is already there, whereas your provisioning pipeline deals with all net-new configurations.
- Ensure that you have quick, straightforward, and tested rollback in case of unplanned impact.
- Once you're ready for production nodes, always conduct the first change of a mass-deployment as a manual change to a node to ensure production success prior to mass-deployment. Depending on the risk profile, slowly scale your network changes to larger increments and frequently review KPI and service performance indicators as measures towards the success of your change.
- Investigate tools that abstract the complexity of automation and use a low code method. Low code is a tool that minimizes the need for coding and abstracts the complexity of the automation into a friendly GUI, which allows Network Engineers to focus more on the networking aspects then the programming aspects.

Troubleshooting Pipeline

Create a pipeline for collecting troubleshooting data related to a service and/or platforms involved in an incident. This should allow for the collection and analysis of the current state to decide how to proceed with resolving the incident. This may include SYSLOG, Event/Fault Management (SNMP Trap), and other indicators (such as counters) to effectively analyze a situation on a macro level.

Key Considerations –

- Identify and consolidate the necessary data from various sources (SNMP Trap Data, SYSLOG, Counters and Dashboards). This will help with the ability to properly characterize an issue to improve the quality of the diagnosis and resolution.
- Keep in mind performance constraints to avoid self-imposing issues. For example, ONTs have limited bandwidth to collect all object IDs.
- Review indicators/object collections that illustrate the same thing and minimize duplication, where possible.
- Understand service flows and network platform characteristics. What parameters to collect and how the protocols and service-flows interact with each other.
- Parse the key information that's important towards resolution.
    - Caveat: On older CLI (command line) based platforms, parsing, and structuring output data can be a challenge. There are tools that can help, which use Regular Expressions (RegEx) to conduct pattern matching, and then can help structure the data so that it can be used purposeful throughout your automation.
- Your Troubleshooting pipeline will lead to auto-diagnostic capabilities for analyzing faults, and eventually auto-restoral capabilities for resolving issues in the network. For a beginner, it's recommended that you will start with auto-diagnostic capabilities to collect data quickly, eventually leading to targeted auto-restoral once a clear fingerprint of an issue is established.

Compliance Pipelines:

Configuration Compliance is especially important for automation. You need to understand what configuration is compliant to your approved and standard configurations, as well as easily deciding non-standard configurations in the network and be able to report on both. There are various tools to provide this functionality, either open-source, or proprietary.

Key Considerations -

- Learn a templating language, such as Jinja2 to help you set up your standard configuration templates. These templates can be compared against your live configuration to detect anomalies or defects that will result in service impacts/degradation or security exposures.
- Your configuration templates become the standard for which you configure your network elements on and has been rigorously tested (prior) and is known to be stable. Deviations against your standard template are non-standard. You want your non-standard configuration to be clearly known, as it represents an elevated risk in planning some of your change activities and needs to be well understood.
- Take time to build non-compliance reports and review regularly to ensure items do not go undetected. Missing key configuration can result in issues that cannot be seen immediately until the customer uses a feature, or vulnerability exploited. Our goal is to reduce preventable faults to zero and automation is a key part of achieving this.

Reporting Pipelines:

One of the things that Automation can aid with is collecting copious amounts of data and interpreting and analyzing the data to provide insights that may not easily be seen by manual checks of the data. This can be used for preventative maintenance, capacity, service experience and overall improvements.

Key Considerations -

- As been previously mentioned: understand your workflows, performance KPI's and relationship to other business drivers (such as service truck-roll data and customer experience metrics). Build reports that provide near-real-time and historical views of your service experience.
- Identify and review items that are embers but have not yet turned into fire.
- Famed Business Theorist Peter Drucker once said, "What gets measured, gets managed." Work with your performance analytics teams to build meaningful data that represents the true end to end service experience for your customer.

These serve as key focus areas for your beginning automation journey. Of course, as you progress through beginner to mastery things will become clearer and you'll customize what works well for you and your business.

8.2. Process Measurements

In the past Mean-Time-To-Repair (MTTR) alone played a large part of measuring your Incident Management success-rate, however there are new measurements being looked at which can be used to measure improvements through automation. Due to the increasing complexity of networks and platforms, automation inevitably to be applied to help analyze, collect, and report across several disparate platforms and services. This is the case with FTTH services, which span multiple devices (CPE, Access, Aggregation, BNG, Distribution, Core) and services (Diameter, DHCP, DNS, OSS, etc.).

Some of the process measurements that have become more prominent and can assist in measuring tool effectiveness and automation initiatives:

Mean-Time-To-Detect (MTTD). Simply put, how is long is it taking for you to first detect a performance or network issue? Being able to consolidate many sources of network data (SNMP Traps, SYSLOG, etc.) and build correlations between the data can help pinpoint and detect a problem that may not be easily seen by traditional event messages. In addition, the presence of software/hardware bugs results in anomalies that do not alarm or present usable logs. These 'hidden' issues require comprehensive

collection and analysis of data sources to help detect. Effectively detecting an issue means that you can work towards restoration quickly and decisively.

MTTD = (Total time between failures and detection) / (Total number of failures) (Note: The Time between failures and detection are captured when the initial symptom or incident cause is recognized.)

Mean-Time-To-Understand (MTTU) – Beyond simply detecting issues, understanding the incident will help you in diagnosing the issue quicker. By piecing together relevant activities in a timeline, you can gain a deeper understanding of the issue's origin and triggers. For example, when DHCP assignments fail, there are several potential areas of failure. Is it the DHCP infrastructure itself, the application, the network connectivity, the CPE, or even authentication issues that are preventing from DHCP leases from persisting. From a scoping exercise, where is the issue occurring (region, area) and where isn't the issue happening.

Understanding the issue leads to more effective root-cause identification/diagnosis and eventual restoral. Being successful at this means that data insights and automation will be needed to provide valuable insights that humans alone may have difficulty in gathering efficiently.

MTTU = (Total time to understand) / (Number of network incidents). (Note: The Time to understand is gathered when the team has a grasp of the issue and starts working on restoral activities.)

This diagram (Figure 8) shows the relationship between Mean-Time-Before-Failure (MTBF), Mean-Time-To-Diagnose, Mean-Time-To-Restore (MTTR), and Mean-Time-To-Failure (MTTF)

MTTD (Mean-Time-to-Detect) and MTTU (Mean-Time-To-Understand) are distinct metrics looking at different areas of your incident process but can be complementary towards lowering your MTTD (Mean-Time-To-Diagnose) and obtaining a potential cause of the issue.

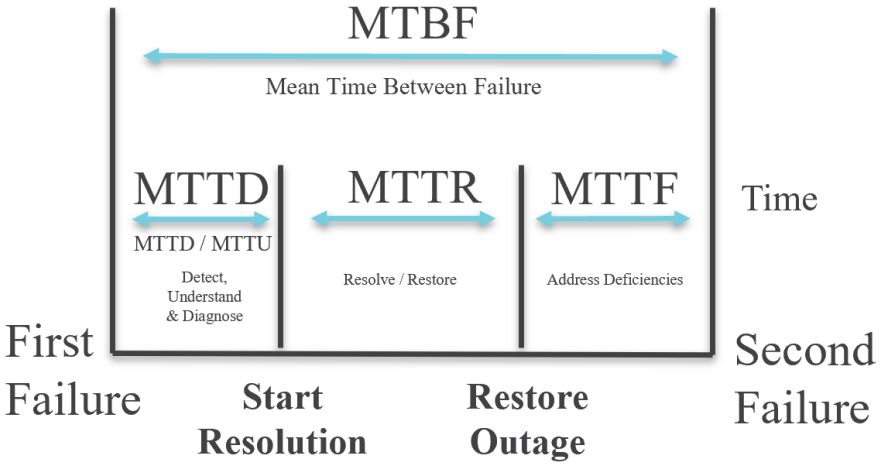


Figure 8: Incident Process Measurements

9. Future Designs/Architectures

There have been paradigm shifts around the networking industry over the last several years. The focus on software defined networking has now transitioned into the Access network space. This standard known as SDAN (Software Defined Access Network) is the term that marries the software-defined networking

space and its network programmability, flexibility, and scalability with use-cases for Broadband Access Networks.

SDAN platforms offer a range of benefits, including scalability, flexibility, and open architecture. This enables Network Operators to easily expand their networks, adapt to changing requirements, and integrate with various tools and technologies. SDAN uses a Controller in a Data Centre or Cloud infrastructure to perform control and management functions for access networks. Some of the compute functions of the NE can be moved into software and the separation of Control-Plane and Data-Plane can exist. By leveraging SDAN, Network Operators can seamlessly implement DevNetOps practices network functions that might be difficult or time-consuming on traditional platforms.

Some of the benefits of SDAN:

- OLT Provisioning becomes easier through templated configuration that is supported and pushed from the Controller.
- Network Operations and intelligence intrinsically becomes part of your environment.
- Vendors are starting to focus their development in these areas to enrich the overall experience.
- Modernize your architecture with Open APIs and flexible programming / connectivity options that can be tailored to you and your environment.
- The Controller is your source of truth (master) for nodal information and configuration and is always in sync with your Network Element (NE). Out-of-Sync changes can be fixed automatically or prompted for correction without the need for extensive compliance validations.
- The Controller's ability to store configuration data and push it to the OLT when it becomes available is a key advantage of SDANs.
- Automation and network configuration activities are inherent on the Controller.
- Provisioning becomes easier. Either through standard templates being ‘pushed-down’ to the NE, or through Zero-Touch Provisioning.

Figure 9 shows the Traditional EMS uses protocols such as Simple Object Access Protocol (SOAP) and Simple Network Management Protocol (SNMP) to manage the platforms, whereas the SDAN based system uses more flexible and programmable protocols such as Open API, REST (RESTful API), YANG modelling language and Network Configuration Protocol (NETCONF) for management and provisioning functions. SDAN and Traditional will inevitably be together and this co-existing is important to explore and understand.

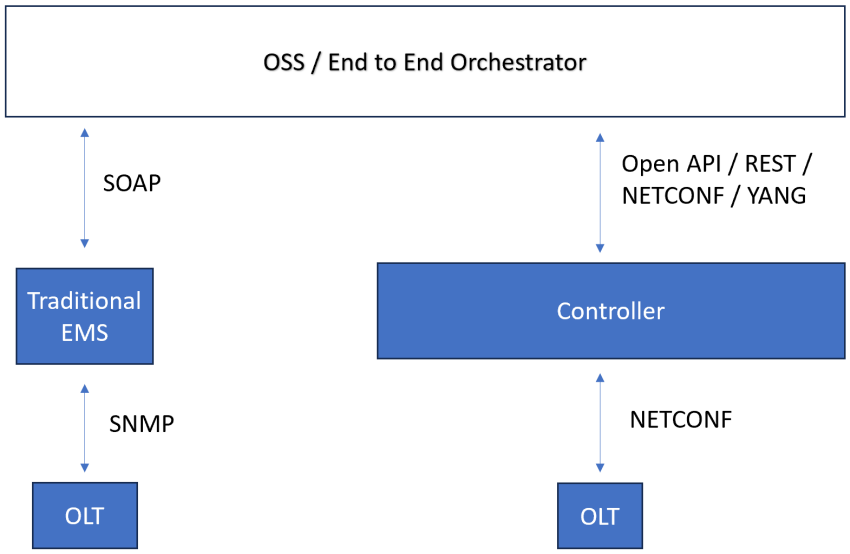


Figure 9: Traditional vs. SDAN based setup.

Recently the discussion of Intent-Based-Network (IBN) has become prevalent. Intent-Based-Networking is the evolution of model-driven architectures. Intent-Based-Networking works on a higher-abstracted method understanding what you want to have configured vs. knowing how to provision it. It relies on service models and learning to build your services through automation within the guidelines that you have set for it.

10. Automation Use Cases

Zero touch provisioning

Zero touch provisioning allows Operators to deploy new OLT devices in a quick automated fashion. For large scale operators, this is a key requirement. There are several benefits to this, including less human error, better utilization of resources, and able to scale to meet demands on the business. Of course there is significant preparation to conduct before this. DHCP, TFTP and automated workflows need to be built with your asset management systems to ensure that you can provision dependent network devices (BNG, DAA node.)

Figure 10 shows a high-level Zero Touch Provisioning Operation, which relies heavily on your DHCP architecture.

High-level architectural view

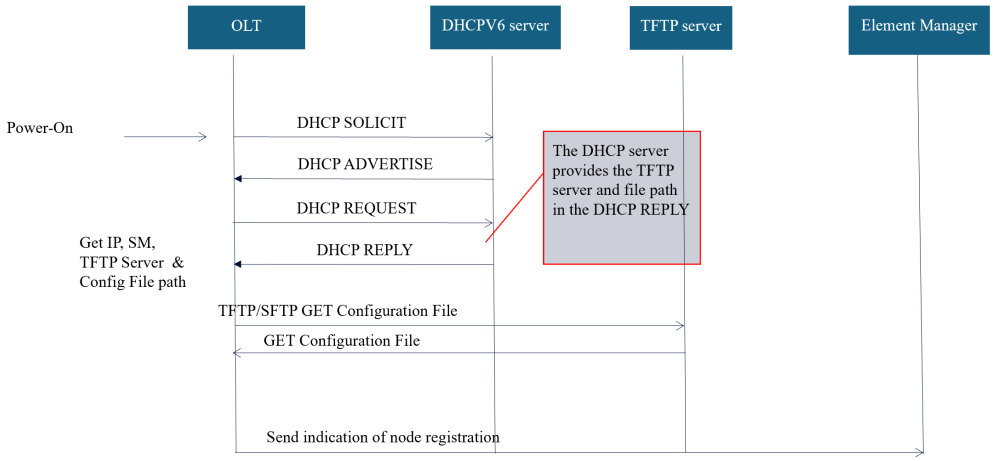


Figure 10: Zero Touch Provisioning Flow

Provisioning to your BNG or DAA node will require your element manager (or SDAN Controller) to connect to your end-to-end orchestration platform to configure the other Platforms. Additionally, integration with asset-management database and work-flow system to ensure the proper end to end provisioning, sequencing and record management across multiple platforms is required.

sequencing and completion of upstream dependencies are completed.

Method of Procedure (MOP) Script Templates.

One of the easiest and best ways to start your automation program is to try something as simple as Templating methods. Where you need several MOPs for executing against several platforms, each with specific differences. Jinja2 (open source) is a great program to start templating your scripts for execution using CSV, YAML, JSON as inputs, and producing procedural documents for repetitive and consistent tasks.

Global Policy Management (BNG).

To keep uniformity over your BNG devices, there are several vendors that use a Global Policy Management capability to propagate configurations of routing policies, ACLs, and other policy-statements. These are automated solutions that are inherent and do not require additional code to utilize them. Identify things that your vendors can do inherently, and you may already pay for, as a first step.

ONT Automated Provisioning.

One key feature that may be included in your FTTH system is the capability for self-installation and auto-provisioning of the ONT. This process involves the ONT and Element Manager/Controllers communicating to transmit the ONT Serial Number and ONT ID to your Northbound provisioning interfaces. This data exchange enables the fulfillment of ONT connectivity and service enablement.

This self-installation process is known as bottom-up provisioning, which is triggered by the ONT discovery message. It works in tandem with top-down provisioning, which is data that originates from your BSS (Business Support Systems) and fulfillment platforms. Is the service authorized, what is the

service tier, etc. Together, these processes ensure that the ONT and subscribers are properly authorized for services. By integrating top-down and bottom-up provisioning, you can automate service provisioning across your network.

With SDAN this comes as part of the feature-set, but less recent OLT systems require customization to do this based on the discovery of new ONTs.

11. Conclusion

This paper discusses the importance of automation for managing Fiber-To-The-Home (FTTH) networks. While implementing automation can seem complex, the benefits outweigh the challenges. Automating deployments and network operations can significantly improve efficiency, reduce errors, and boost overall productivity.

The paper highlights key points:

- A strong organizational foundation is crucial, including a DevNetOps culture and processes like Agile and LEAN Six Sigma.
- Traditional network management systems lack flexibility for automation due to older protocols.
- The future lies in Software Defined Access Networks (SDAN) for easier automation.
- Customer demands include reliability, affordability, and ease of use, all achievable with the assistance of automation.

Abbreviations

AAA	authentication, authorization, and accounting
ACL	access-control list
API	application programming interface
BNG	broadband network gateway
CLI	command line interface
CNF	container network function
DAA	distributed access architecture
DEVOPS	development operations
DEVNETOPS	development network operations
DHCP	dynamic host configuration protocol
DMAIC	define, measure, analyze, improve, control
DOCSIS	data over cable service interface specification
DSL	digital subscriber line
DWDM	dense wavelength division multiplexing
EMS	element management system
FTTH	fiber to the home
GBPS	gigabit per second
GPON	gigabit passive optical network
IEEE	Institute of Electrical and Electronics Engineers
IP	internet protocol
IPoE	internet protocol over ethernet
ITU	International Telecommunication Union
JSON	javascript object notation
KPI	key performance indicator
MSO	multiple system operator
MTTR	mean-time-to-restore (or repair)
MTTD	mean-time-to-diagnose (or detect)
MTBF	mean-time-before-failure
MTTF	mean-time-to-failure
MTTU	mean-time-to-understand
NAC	network-as-code
NE	network element
NETCONF	network configuration
NRE	network reliability engineer
OMCI	ONT management control interface
ODN	optical distribution network
OLT	optical line terminal
ONT	optical network terminal
OSS	Operations Support System
PCRF	policy charging and rules function
PCEF	policy charging and enforcement function
PON	passive optical network
QoS	quality of service
R-OLT	remote optical line terminal
REST	restful (api)
RPC	remote procedure call

SCTE	Society of Cable Telecom Engineers
SDLC	software development life cycle
SFP	small form-factor pluggable
SDN	software defined network
SDAN	software defined access network
SOAP	simple object access protocol
TX/RX	transmitter / receiver
VCS	version control system
XGS-PON	10 Gbps symmetric passive optical network
XML	eXtensible markup language
YAML	YAML Ain't Markup Language
ZTP	zero-touch provisioning

Bibliography & References

Beck, K., et al. (2001) The Agile Manifesto. Agile Alliance. http://agilemanifesto.org/)

Cisco Systems, Inc. (n.d.). Network Automation Trends and Strategy. [Whitepaper]. Retrieved from https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/network-automation-strategy-wp.html

Clemm, A., Ciavaglia, L., Granville, L., and J. Tantsura, "Intent-Based Networking - Concepts and Definitions", RFC 9315, DOI 10.17487/RFC9315, October 2022, https://www.rfc-editor.org/info/rfc9315.

George, M. L., Rowlands, D., & Kastle, B. (2005). *Lean Six Sigma pocket toolbook: A quick reference guide to 100 tools for improving quality and speeding up results*. McGraw-Hill

Griffiths, M. (2021). *PMI-ACP exam prep*. RMC Publications.

Kerpez, K., Cioffi, J., Ginis, G., Goldburg, M., Galli, S., & Silverman, P. (2014). Software-defined access networks (SDAN). *IEEE Communications Magazine, 52*(9), 152-159.

Murphy, N. R., Robbins, J., & Kurn, C. (2016). *Site reliability engineering: How Google runs production systems*. O'Reilly Media.

Nokia. (n.d.). *Triple play service delivery architecture guide*. Retrieved from https://www.nokia.com

Nokia. (n.d.). *Nokia SDAN Use Cases Brochure* [White Paper]. Retrieved from Nokia.

Pinto, I., & Chaudhry, F. (2024). *Automating and orchestrating networks with NetDevOps*. Cisco Press.

Wideman, Max. (2001). Project Management Simply Explained: A Logical Framework to Help Your Understanding.