DevOps & FITNESS Platform

Nicolas TEMPESTINI[(1)], Gilles SALANIE[(1)]

[(1)]*Thales Alenia Space*
26 Avenue Jean François Champollion
31100 Toulouse
Email:nicolas.tempestini@thalesaleniaspace.com,
gilles.salanie@thalesaleniaspace.com

INTRODUCTION

Thales Alenia Space, in the scope of its digital transformation, is actively working at accelerating its “value” time-to-market by promoting a DevOps culture on its products.

WHAT IS “DEVOPS” ? WHAT DOES THALES ALENIA SPACE WANT TO ACHIEVE ?

While DevOps is typically presented as a way to integrate developers and operators teams, the root of DevOps can be found in the Lean engineering approach with a key emphasis on reducing the “delivery” lead time while focusing on “built-in” quality.

Traditionally, the way projects are typically executed includes successive hand-overs between business stakeholders (the ones capturing the “customer” need), engineering teams (writing system specification, coding software and delivering building blocks that are integrated by IVV teams) and Operations teams (the ones that are installing and operating the whole system stack including infrastructures). A typical example is the Galileo project where several parties are delivering into a successive set of environments the working software. In such a complex industrial setup, this flow can take up to several years. However, it is the final responsibility of operators (the ones operating the system) to make sure at any time the system is working properly with the right quality of service. They are the end “victims”. From this example, we can say that the value stream is broken as there is little chance that initial engineers working on the needs know about the real operational impact of what is delivered several years later. As the setup involves different parties, each of them “siloed” in their own activity, the final “feature” quality can be seen as a constant trade-off between cost, schedule and scope of each of them. This is not going towards an efficient way of delivering value to the customer and by derivation, to the end user.

One of the ways to answer to this challenge in the software industry is applying the “Agile” framework. This is also the case in the Space industry where software can be found on-board and in the ground segments. Agility applied at engineering level can be seen as not enough to accelerate the overall value stream to the end user. When carefully implemented, it certainly increases the quality and the regularity of features through a close collaboration of system and software teams. However, it is not solving the final goal of having these features delivered to the end user. Actually, it may increase the fear of business stakeholders who may feel frustrated of not having a clear scope/schedule view. It surely increases the fear of operation teams who still have the final responsibility to make these “more frequent” deliveries to work in production systems. The solution to achieve an overall efficiency would be a way to cope with these fast deliveries up to production while making sure that the business stakeholders can get fast insight on what is being implemented on the system and its real impact on the end users. This is what is called DevOps. DevOps cannot be achieved with some sort of Agility. Many of the DevOps concrete examples embeds Agile principles for instance on the “early and continuous delivery of valuable software” [2].

DevOps is typically represented as a loop of activities that are performed ideally by a group of people cooperating closely with a common objective to deliver a feature fast to production. This engineering practice targets a fully automated pipeline (set of actions) at all stages of the value creation to deliver fast, with the right quality in order to be able to get a fast feedback (and eventually modify what has been delivered previously to improve it). The “blue” parts represents the engineering activities (the DEV part) while the yellow part represents the operators activities (the OPS part).

Fig. 1. Typical DevOps activities

Organization applying DevOps efficiently are organized in a non-traditional way. They usually involve common "teams" doing the Dev and the Ops on a specific part of the system making sure that any modification that is introduced in the system is known and understood by the "OPS" team while it is built.

Task automation enables faster deliveries. The process requires less manual intervention. The highly automated pipeline ensures that each of the stages of the development phase up to the deployment can be assessed factually by a clear status. As no human action is involved, the process is not error prone and can be repeated infinitely. The quality of the delivered modification relies on the quality of this pipeline. The goal of the team is to automate and continuously improve the automated process of delivery, adding any automation step, automated test or automated quality check to make sure the right quality is achieved. Errors becomes opportunities to stop the process, fix it as soon as possible and thus improve it by adding new automated steps and checks that are missing.

From a use case point of view, this approach is valid for commercial satellite projects but also for internal projects where tools, services, test benches are being developed as enabling systems for these projects.

WHY DEVOPS IN THALES ALENIA SPACE ?

Reducing lead time while increasing quality of what is being delivered is the final goal of any industrial company. Taking into account that DevOps can be achieved, it opens new opportunities to deliver faster value in a better adequacy to user needs.

Here are two examples :

A first concrete application is around Cyber Security threats. The challenge of managing and solving vulnerabilities in a complex space system is real. Even if a majority of systems are said to be "disconnected", there are likely to be vulnerable if not upgraded regularly. The increasing usage of open source software enforces this statement as vulnerabilities are public with those components. Banking institutions already ensure a very low lead time with respect to vulnerability patching. It is usually performed in less than a few hours. Reaching this performance in a space system is possible by rigorously applying DevOps techniques. This is particularly true for Ground Segments. For sure, upgrading regularly shall be compatible with the level of service required by the application. It means that the software architecture shall be "ready for" DevOps. (See architecture chapter).

Even if the final grail is the one-click seamless deployment to production, DevOps may not necessarily implies automated steps up to production. A project can still require final manual approvals or compliance checks even before being deployed in front of end users. The final production environment may also be a disconnected environment requiring a "manual" transfer at first, before being deployed automatically also. In such a case, the purpose will be to push deliveries automatically as far as possible. DevOps may be enforced up to final IVV Phases on a "pre-prod" environments. In such a case, the "OPS" part is kind of "managed" by the same or a close local team (can be IVV) ensuring that the system is up and running, monitored and tested with the latest version available, refreshed as often as possible. Opening such an environment to the end users, as early continuous deliveries, is the best way to make sure that what is being developed is actually in line with the user requirements. It is also an opportunity for the end user to confirm or adapt new requirements. This is typically closing the agility loop where the users and developers can learn from what is deployed and adapt or modify the product. This loop is a way to achieve the objective of built-in quality.

WHAT ARE THE NECESSARY MEANS ?

The transformation includes 4 working axis:

- **Culture**: The purpose is to challenge how product management and development is performed today. It includes providing more insights on business and users priorities to the product team. It goes through more user interaction, and more agility. Products are developed iteratively. The target is to make sure that the value of what is delivered permanently matches end-user needs. Organizations are reviewed to make sure teams are working closely on the same objective. People are up-skilled through specific trainings and coaching. The end-user is at the heart of product. Business decisions and priorities are taken based on fact from end-users feedback and monitoring. Spreading a DevOps culture is actually melted with spreading an Agile culture.
At Thales Alenia space, we progressively deploy DevOps favorable organizations on new projects. We established a skill matrix that is DevOps-oriented (including all the necessary skills from Dev to Ops) and are reviewing the whole engineering population. The purpose is to be able to identify key experts who can train other trainees and also key gaps leading to specific external support or recruitments.

- **Practices & Tools**: The purpose is to refocus each step of the value creation process on quality and automation. Standard build and run phases (aka. Dev and Ops phases) are enriched with processes and techniques allowing such improvements. This goes through heavy implementation of continuous integration and continuous delivery techniques. Automation brings the capacity to reproduce a complete process without errors as often as necessary. Test automation brings the capacity to assess the quality of the development process. Modifications are made small so that, when an error is detected, the corrective action becomes simple. Deployment phases are automated up to the infrastructure to make sure a change is tested in a relevant environment every time.
A large majority of engineering practices are described in a Thales Alenia Space referential called e-Tooled-UP practices where, for each practice, an engineer can find a detailed description with examples, key enablers and common pitfalls.

In Thales Alenia Space, we use the following tool set to build our pipelines :

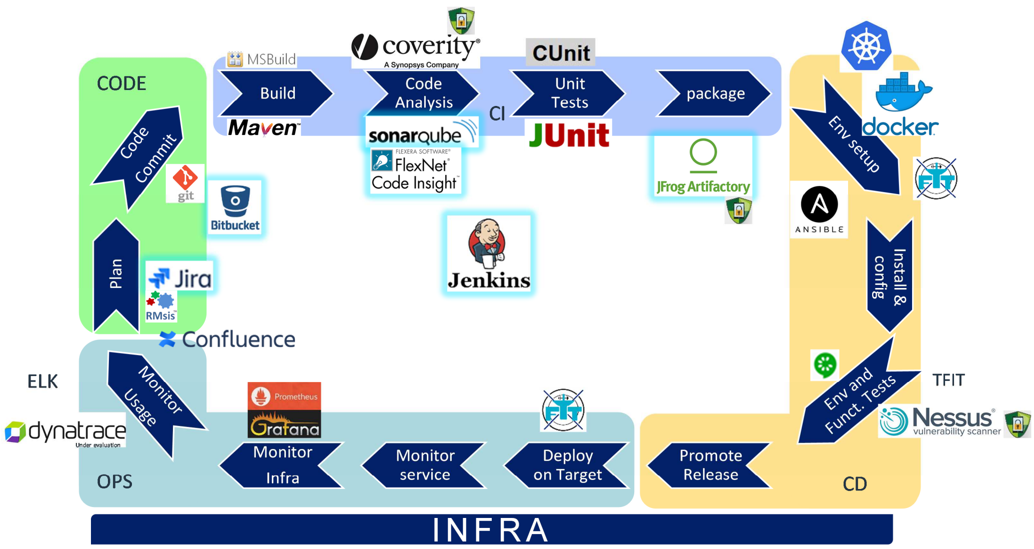

Fig. 2. Toolset in Thales Alenia Space DevOps pipeline

Code is managed through front-ends (Eclipse, Visual Studio Code) connected to GIT repositories. The build is orchestrated with Jenkins. Each developer includes a file called “Jenkinsfile” which aims to describe the continuous integration pipeline to be used for the project. This file is parsed into Git repositories and, upon code modification, the pipeline execution is triggered. To build the software, ephemeral build machines are instantiated on the infrastructure (see FITNESS description below). The ephemeral build environment is running on a Kubernetes cluster running on the platform. As soon as the software is build, code analysis can be performed automatically covering both static and cyber security rules. Then, automated tests are executed at code level (unit test). Once tests are completed, binaries are committed to the binaries repository running in an Artifactory. For each of the continuous integration steps, developers can describe their stopping / failure criteria. Documents may also be generated at this stage (change notice, tests results, interface description, documentation). Once continuous integration is completed, continuous deployment starts. The target integration environment can be built on demand to the infrastructure. On the infrastructure created, the system is deployed. In case the upgrade path shall be tested, the previous environment is kept. During these tests, a set of upper level automated tests are executed. This may include vulnerability tests using Nessus. Another set of documents may be published at this time. Once deployment is completed, and depending on the environment, specific manual tests (as less as possible) may be executed to complete non automated tests. Monitoring is either part of the system that is deployed, either based on the service that the platform offers.

- **Software architecture**: The purpose is to improve software architecture to ensure that when a change is pushed to "production", the system is resilient to this change. Requirements are taken into account in the design: no outage during upgrade, logging capability, close monitoring, etc. They are enforced and verified at each continuous deployment phase during development phase. Operation teams are thus more confident on accepting fast and recurrent changes to the production system. This axis is particularly valid for ground segments software where Cloud Native application is a reality.

From a ground software point of view, this goes through cloud native application principles. Software architectures shall be "disaster aware", aware that the application may be asked to stop at any time (for upgrade purpose for instance). Typical example is the blue-green deployment, where in case of upgrade, a new instance of the application is deployed in parallel of the old one. The application traffic is redirected to the new instance and the old instance is automatically destroyed when the traffic does not go through the old one anymore and that the monitoring of the new instance shows no negative impact. This requires stateless applications. Web technologies are particularly efficient for this. Applications shall be resilient and care about Security (making sure that the design is secure (authentication for instance), Scalability (how to increase the processing power of the application by adding new instance of the application), Deployability (application which can be deployed, configured and upgraded automatically without impacting the on-going service) and Testability (application which can be tested with automated tests). It usually relies on an "API first" approach. In the software industry, common criteria to assess that an application is "cloud native" are the 12 factors [1].

- **Infrastructure / Platform**: A cloud platform is necessary to sustain the collaboration through automated processes for faster value delivery. This platform, which is based on a software defined datacenter, offers the right flexibility, core services and APIs to transform any infrastructure operation into code. A project team is empowered to create ("code") necessary infrastructure resources through a self-service portal and/or automation scripts. This project team can, for instance, recreate its entire testing system (infrastructure, OS, dedicated software, configuration) from scratch at every software modification to make sure the change will be deployable and workable in the end-user environment (production like). In Thales Alenia Space, the "DevOps platform" is shortly called FITNESS and is managed by a DevOps team.

The FITNESS team uses DevOps techniques to deliver new services to the platform users.

THE FITNESS DEVOPS PLATFORM

To support our internal digital transformation, we had to rethink the way we consume the Information Technologies (IT) resources for our daily software development/integration and maintenance activities. First of all, we couldn't afford anymore to purchase dedicated hardware (server, workstation, network, storage…) when starting any development or any prototyping activity in the frame of a new project. We need also to be able to work on a mutualized facility between different teams within the company and external partners. We then naturally had to migrate to a new common platform targeting the following requirements :

- delivering IT resources simply by filling an electronic form
- scalable and maintainable IT infrastructure
- fully integrated with the existing Thales Alenia Space Information System (IS)
- providing native access to the various internal engineering and development environments.

To implement this "IT as a service" approach, we've designed this IT infrastructure on the basis of the Software Defined Data Center (SDDC) concept, introduced by VMware a few years ago. In this kind of IT infrastructure every resource such as CPU, storage and network is virtualized and can be defined/configured through a piece of code. From the end-user perspective, we talk about Infrastructure As a Code (IAC) which is basically the textual description of the target virtualized IT infrastructure in which takes place the development/integration and maintenance activities. The various technologies have been selected with the following rationales :

- VMware vSphere hypervisor CPU virtualization has been selected as a market-leading product, because it is largely used in our ground segment products and infrastructure, and heavily used in the Thales Alenia Space information system.
- VMware vSAN Software Defined Storage (SDS) solution has been selected because it is based on standard x86 server H/W and because it is linearly scalable (throughputs and I/Os) by adding new servers to the platform.
- VMware NSX-V Software Defined Network (SDN) solution has been selected to provide on demand network capabilities (Routing, VPN, firewalling, load balancing…) as part of the virtualized infrastructure. The use of NSX-V improves the security by segregating the network traffic and connectivity between the various projects hosted on the DevOps Platform. The integration capabilities with third-party security solutions (e.g. agent less anti malware, IPS/IDS) makes NSX-V even more attractive.
- VMware vRealize Suite has been finally selected as the overall monitoring and automation solution for this on-premises virtualization platform. This VMware suite provides a very efficient supervision and capacity planning through VROps Manager. VRealize Automation provides the user portal through which the users deploy their working environment of the DevOps Platform.

Since the commissioning of this DevOps Platform in Q2 2018, the main usage has been some software development and early system integration activities. Indeed, usually, the system integration activities started only when the target customer hardware was available and finally, very late in the project's life cycle. As far as the testing approach allows it, the early use of this platform allows to start system integration as soon as the first piece of code is available. Once the target H/W infrastructure is available for testing, the system level integration can focus on performance, while in parallel the new software release integration can take place on the DevOps Platform. At this stage, software artifacts can be deployed and tested indifferently on the target H/W infrastructure or on the virtualized environment.

The figure 3 shows an example of some ongoing projects where software development and system integration activities take place whether on the Devops Platform or on the Target customer platform (still on Thales Alenia Space premises).

In this example we are integrating many different technologies on the DevOps Platform with a focus on the interfaces. For instance this project uses a block-oriented distributed file system but also a framework and image processing algorithms. The typical use of the DevOps platform is to work first on the deployment of the various components and integrate them as close as possible to the target environment. In the frame of this project, the major part of the distributed file system has been prepared and tested on the DevOps Platform, while the fine performance tuning is taking place on the final target platform. Regarding the image processing algorithms of this project, we limited the test of these components to standalone tests, where the distributed and real-time processing is to be tested on the target platform to assess the exact performance on the final hardware.

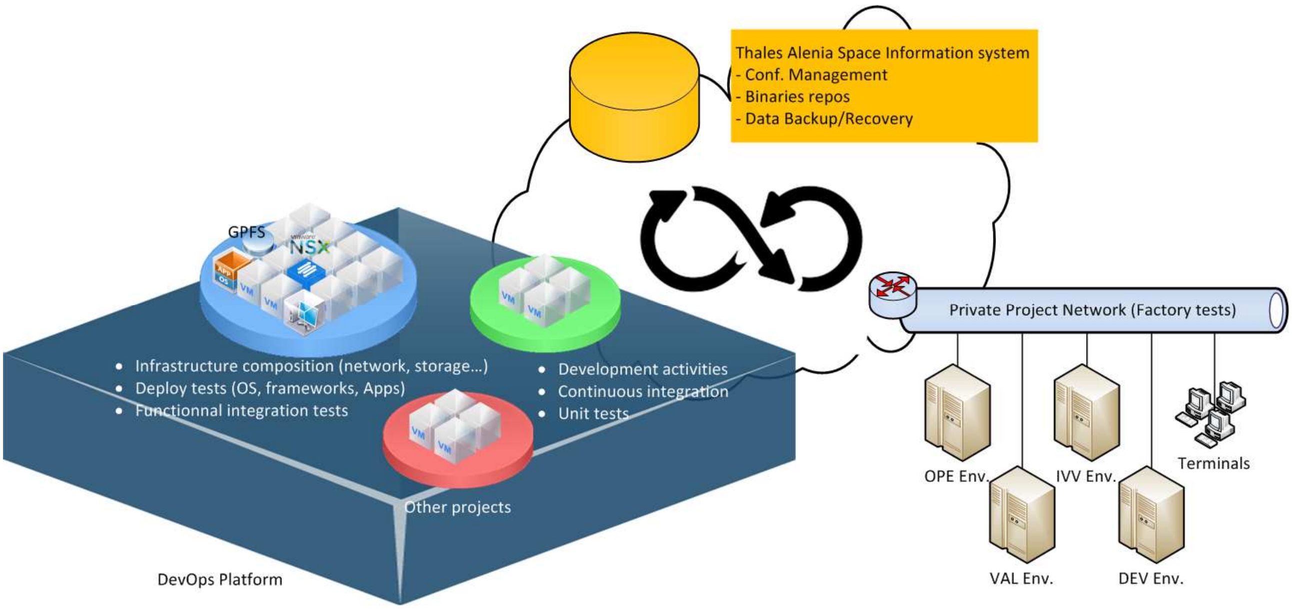

Fig. 3. Typical project setup for development and integration activities

Another important use case is the maintenance of ground systems components. Due to its native capability to instantiate and deploy on demand a complete virtualized ground segment infrastructure, the DevOps platform is definitely the right candidate to host the software maintenance activities. This trend follows Thales Alenia Space's effort in the design phases, to avoid any dependency between software and underlying hardware. The first expected result is the capability to limit, and even more avoid, the need for a "reference" platform. Several on-going projects at bid stage already make the assumption that the warranty/maintenance phase will only take place on the DevOps platform.

Generally, most of the hardware components involved in our ground segment infrastructure designs exists as virtual appliances (virtual storage, network and security appliances…). If not, we still have the capability to interact with actual hardware as far as this hardware provides standard TCP/IP interface or the communication protocol can be carried over IP (RS485, RS232...).

Finally the flexibility of this mutualized platform enables and simplifies the co-working of multi-disciplinary teams.

Beyond the regular development/integration and maintenance activities running on this platform, we made this platform flexible enough to be able to support benchmarks and experiments. The general approach is to add temporarily some hardware resources to the DevOps platform, providing access to a few new technologies. Once the evaluation of this new technology confirms the interest for our Ground segment solutions, the integrated DevOps Team launches the integration of the solution on the DevOps Platform. The new technology is then available "as a service" to every Thales Alenia Space users.

As an example, the use of GPU for massive computing has been evaluated on the DevOps platform with the benchmark of a deep learning algorithm in charge of object detection within Satellite images. The success of the bench confirmed the interest to provide some "GPU as a service" on the DevOps Platform. The integration of additional nodes equipped with PCIe GPU cards is then planned for 2019.

Another example of features to be introduced "as a service" is the emulation of alternative CPU architecture such as SPARC or ARM on top of the native x86 DevOps Platform servers. Indeed, beyond the ground segments software solution, the embedded software represents a large part of our development and integration activities. The need for early integration capabilities is the same because the target H/W availability can be even more critical compared to ground segment hardware. Both of the architectures mentioned here-above correspond respectively to the various rad hard CPU that are used on Thales Alenia Space platform and/or payloads . The "emulation as a service" foreseen here aims at delivering on demand virtual resources capable of running native binary codes for those CPUs. The overhead introduced by on-the-fly instruction transcoding, is largely balanced by the scalability of virtual resources on the

DevOps platform. The main challenge remains here in the simulation of software interfaces through software stubs or H/W virtual appliances.
In the specific case of ARM, a new opportunity for on-demand CPU resources came from VMware, announced by mid-2018 with the native support of ARM Architecture in its vSphere virtualization solution [3]. From a DevOps platform perspective, this service could be suggested as a new cluster based on a farm of ARM System On Chip (SOC).

CONCLUSION

The paper has detailed the way Thales Alenia Space is moving to the DevOps approach in its software development/integration and maintenance activities. This project takes place in a larger company digital transformation.

From a technical point of view, the Thales Alenia Space DevOps Platform contributes actively to innovation by delivering regularly new technologies to the user hand and providing an environment to validate solutions similar to the target infrastructure, which allows to detect issues a lot more in advance and gain time in the value delivery, reducing the time to market of our different products.

From a cultural point of view, this co-working area is definitely a way to break siloes between historically distinct activities.

Reference :

[1] https://en.wikipedia.org/wiki/Twelve-Factor_App_methodology
[2] https://agilemanifesto.org/principles.html
[3] https://www.vmware.com/radius/taking-innovation-to-new-and-unexpected-levels-at-vmworld-2018/