

Expert opinion

7 key success factors for your DevOps procedure

By Maxime Orand, Consultant, Hardis Group

Although on paper the DevOps procedure looks like a winner, we have to accept that sometimes, despite the best will in the world, in reality it is not a compelling solution. This is due to a lack of preparation, and because the steps are not followed correctly. 7 essential points to be familiar with before starting (or to correct a procedure that has got bogged down).

1 – Use modeling

Reduction of cycles alone is not sufficient to completely streamline the processes. On the contrary, multiple deliveries are each a potential source of friction. To reduce this risk, it is therefore recommended to identify the optimization options as early as possible. This is done in particular not only by modeling all the delivery processes, but also by giving precise unambiguous definitions of responsibilities and the deliverables associated with each step in the cycle. In other words, the automation and rapidity of DevOps do not do away with the need for formal delivery and documentation processes, and for introduction of human check points.

This effort to be transparent also represents an opportunity to shed light on certain tasks, so that they are no longer sacrificed to planning requirements. Of course the tasks concerned are operating tasks performed right at the end of the production chain, but in the same way as other processes they determine the quality of the overall service rendered.

2 – Emphasize culture rather than tools

Too often, a change in methods is limited to acquisition of new tools. However not only is a change in culture just as important to ensure success, it is even a prerequisite: tools are simply a means of implementation. Once the modeling mentioned above has been put in place, it's time to look at human resources (existence, training, recruitment, etc.), organization, governance and more generally the information given to teams about the objectives to achieve. Although the steps to support change and reorganization of the teams are time-consuming and complex, they are vital in order to achieve full participation in the approach.

In this respect, the sharing and decision-making ceremonies are a major component of cultural change. These meetings enable the different working groups to interact and communicate effectively, so they can share their issues and align themselves with the challenges of each iteration. The introduction of new tools and automation mechanisms can only fully bear fruit once the cultural changes have taken place.

3 – Design a common baseline

By definition, each team sees each process through its own spectrum, depending on its specialization (development, testing, integration, operations). But to maximize synergy and ensure DevOps succeeds, it is vital for all teams to have the same overall vision: definition of a common language, contextualization of each team's working methods in the entire delivery cycle, etc. This will also lead to the creation of collaborative tools in the field: follow-up of bugs and incidents, delivery schedule, project schedule, service commitments, shared backlogs, etc.

So the work environment is common to all teams, but also information is accessible for everybody, in a system that favors better awareness-raising for each individual about other people's imperatives and constraints. The objective is to create overlapping zones (developers who have access to logs and monitoring, operators who carry out tests themselves, etc.) in order to encourage mutual understanding between teams, while ensuring better accountability of all involved.

4 - Keep to the sequencing of the DevOps approach

To progress from agility to DevOps is no easy matter. Although the objective is to achieve continuous operations, the approach requires us to implement 3 essential steps: continuous integration, continuous delivery and continuous deployment. Each step is dependent on its predecessors. It is only when a high delivery frequency is achieved (as a result of the continuous integration and delivery steps) that there will be a need to automate deployments, and following on from that, to automate operations (disaster recovery plan, monitoring, administration, restoration).

There is no sense in attempting to put everything in place at the same time - in fact that could even be counter-productive. It's better to stabilize each step to make sure the next one will be successful. However there is no need to wait for a step to reach complete maturity before starting the next one. Right from the start the chain should be envisaged in its entirety, to anticipate dependencies between tools, and to make development teams aware of how their code will be used.

5 – Measure and improve performance

Every opportunity must be seized to review the different processes: sprint review, bug review (quality), deployment review, incident review, performance review, etc. These must be systematic reviews where each individual can (and must) contribute to continuous improvement. There is no point in trying to pass the buck. Conclusions must be shared and systematically give rise to actions that are monitored and prioritized at project level.

In this context, it is necessary to set up KPIs to monitor the project over a period of time and create continuous improvement loops: number of deliveries performed compared to those expected, number of bugs detected compared to incidents in production, duration of deployments, use of infrastructure, etc. When a problem is identified, a decision must be taken whether or not to automate its identification, even correction. Investments made to tackle a problem in advance are often more profitable than makeshift solutions at the end of the chain.

6 – Attempt matrix organization and transparency

We have seen that in a DevOps team the transparent definition of each person's responsibilities makes it easier to identify their role. Matrix organization is an extremely effective means of going further and developing the concepts of sharing and communication. A DevOps profile and a tester are associated with each group of developers responsible for a functionality or block of components in the project, and in parallel they both belong to the Quality and Operation groups. This guarantees "embodiment" of the overlaps and overall cohesion.

It also reinforces communication, which is vital for DevOps to succeed, since withholding or concealing information can have a particularly severe impact on this type of organization. Managers must simultaneously impose very frequent communication and behave in a positive manner: singling out a responsibility is pointless. Finding solutions together is much more effective. When it comes to DevOps, transparency is not a recommendation, it's a necessity.

7 – Integrate the challenges of operations and support

Until recently agile methods and DevOps focused mainly on integration and continuous deployment, but they are used increasingly now in production phases. Operators are now active participants in the system. They must clearly state their demands, expectations, and use cases, and make the other teams aware of the issues encountered in production. This must be an integral part of the non-functional requirement collection phases, particularly within the context of operation automation.

Similarly, support must not limit itself to watching incidents occur in succession. Support teams must enrich the use cases tested by the quality team's scripts and suggest new monitoring approaches to developers and operators. In brief, by collecting demands or through continuous improvement, the organization must make it possible to take into account this aspect of the application's life cycle.

About Hardis Group

Hardis Group is a consultancy firm, digital services company and independent software vendor, helping its customers to transform their business model, digital value chain and logistics operations. It develops and integrates business, technological and digital solutions tailored to its customers' needs and challenges, improving their competitiveness and operational performance.

Hardis Group has developed expertise in the areas of insurance, distribution, industry and energy, e-health, and logistics services. Today this expertise enables it to offer comprehensive solutions, using an agile approach incorporating co-construction, innovation and continuous improvement.

Since its creation in 1984, the company has built its growth on a pragmatic approach and values of efficiency and firm commitment, to both its 2,500 customers and its 850 employees. In 2016, Hardis Group achieved sales of €83.3 million. As well as its head office in Grenoble, it has five other offices in Lyon, Paris, Lille, Bordeaux and Nantes.

www.hardis-group.com

Press contacts

Anjuna
Elodie Cassar
elodie.cassar@anjuna.fr
Tel. +33 9 64 15 31 27
GSM: +33 6 80 53 82 94

Hardis Group
Hélène Leclercq
helene.leclercq@hardis.fr
Tel: +33 4 76 70 98 41