

## DevOps process

door Bart de Best - 06-28-2017

<https://www.itpedia.nl/2017/06/28/devops-process/>

### DevOps process Introduction



When we talk about DevOps, we might think about a group of people who work without frameworks and best practice models like ITIL. After all, each DevOps team is unique and determines themselves how they organize their work. This reasoning is correct, but there are many things debate on this assumption. The work that these people perform is repetitive in nature and is done through a common team goal. So, we can still speak of a process-driven approach within DevOps.

A process consists of people, resources and methods. Regardless of the uniqueness of the approach of each individual DevOps teams, there are solid patterns in this process. In fact, the laws and regulations force controls to be implemented in this process. These laws and regulations are generic and the controls can be generalized as well.

This article describes which aspects of the DevOps process in practice form a certain pattern. It is a umbrella article for a number of follow-up articles. Each article elaborates an aspect in the following format: terms, concepts, patterns, Frequently Asked Questions (FAQ).

### DevOps process

Figure 1 shows an image of a DevOps process. It is not a formal definition of the DevOps process. It is a representation of what is seen by most organizations as a logical set of phases to run iteratively in order to realize a service.

The blue arrows represent the development process and the orange arrows the operating process. These two processes form the methodical core of the DevOps approach. This article does not deal with the resources (application, infrastructure and tools) and the people (functions, roles and culture). These appear in separate follow-up art

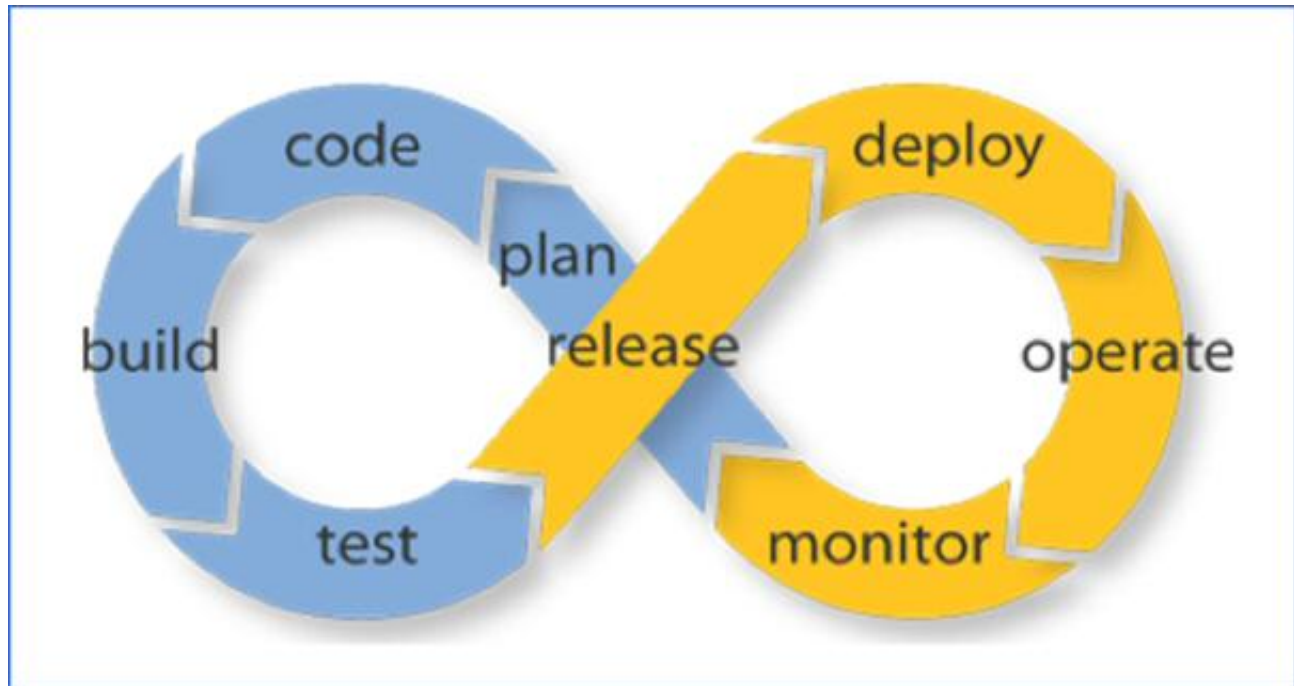


Figure 1. A DevOps process [Exin 2The reason for considering DevOps as a process and writing a set of articles is not to formalize DevOps (which is impossible to accomplish), but rather to classify and exchange stories and best practices in a more structured way. In addition, the construct DevOps becomes more definable and therefore interpretable.

icles.

Each of these two processes can again be considered as a set of phases, procedures or better yet another set of processes. They form always recurring steps to achieve the same results with a defined purpose.

Furthermore, it is important to recognize that these processes are sequentially drawn, but of course also have feedback loops. For example, during the test process, the code process may be triggered to resolve defects, which is a waste in the DevOps process that should be prevented as much as possible. Furthermore, processes are often wider than the indicated scope. Thus, the monitor process also includes monitoring the entire DevOps process.

Finally, the named deliverables per process are not limitative summaries but only a summary of the most important results. The following articles discuss the DevOps best practices. These articles are not intended to define DevOps but rather a set of stories how these processes can be implemented.

## Planning process

### *Purpose:*

The purpose of this process is to determine with the stakeholders what the business case is of the new service offered to the customer, what the roadmap is of delivery and what is the best approach is for the realization.

### *Deliverables:*

1. Stakeholders are appointed.
2. Business case is determined.
3. Roadmap is established.
4. Approach was chosen.

### *Explanation:*

The stakeholders are the parties who deliver (non) functional requirements and pay the solution. The business case is about the added value for the customer and the estimate in time and money to realize this includes the risks to be controlled. The roadmap is a timeline of quarters that indicate the release of building blocks. Each building block is defined in on the level of epics. The approach is the manner in which the development process takes place. This can be Agile or Waterfall. In practice, this is for most organizations that embrace DevOps chose in 80% of the business cases for an Agile approach. At the end of this process, the product backlog was filled and a team has been set up for which a budget has been set.

Following this initial plan, there is either a Waterfall Project or a DevOps approach. In the DevOps approach the service will be created in an incrementally in an iterative manner. For each increment, a detailed schedule is drawn up, epics are translated into features and features in stories. This is often used for Agile Scrum development and for Kanban operations. But all possible combinations are used in practice.

## Code process

### *Purpose:*

The purpose of this process is to generate source files in order to generate application and infrastructure components in the build process.

### *Deliverables:*

1. Features are translated to stories.
2. (non) Functional requirements with measuring rules are determined.
3. Acceptance Test Courses are defined.
4. Technical requirements are determined.
5. Test cases are defined.
6. Source files are created saved and a version management system

### *Explanation:*

The source files consists files consists of the source code that developers create in order to generate application components. However source files also consists of scripts that the operations people create in order to configure the infrastructure components. To make this possible, use is made of the selected features. A feature is a (non) functional description of what needs to be delivered in business terms. A story is the technical translation into what needs to be done in terms of source code (application) and scripting (infrastructure).

Because more developers are often working on the source code and it also changes over time, it is important to provide all source code files and scripts with a unique version.

## Build process

### *Purpose:*

The purpose of the build process is to convert source code into object code (compile) if applicable and also to perform a number of unit tests. The ultimate goal is to qualify within 5 minutes for the source code to qualify for check-in in the baseline.

### *Deliverables:*

1. Object code has been created and the errorlog contains no errors.
2. Object code is stored in an artefact repository.
3. Quality of the source code has been checked.
4. Unit tests are performed and the test results do not contain any errors.
5. Source code is checked in the baseline.

### *Explanation:*

Source code is written in a programming language that is appropriate for the problem area. An example is PHP for a website and Javacode for a business function. This source code can sometimes be used directly in production when it comes to a

language that can be interpreted as PHP, Basic and SQL. However, it is often necessary to convert the source code into an operating system readable format (object code). The result of this process are objects that are also called artefacts. If the source code scan and the unit test cases are successful, the source code is checked into the baseline.

A unit test differs from other test types because these test cases remain within the context code of the object code. Thus, no checks are made on the object code collaboration with other components of the application or infrastructure. This because of the 5 minute requirement of the build process. This is the source code that is shared in the DevOps team to form a new service together.

### Test process

#### *Purpose:*

The purpose of the test process is to perform the integration tests and system tests using all source code and infrastructure scripting created in the sprint so far.

#### *Deliverables:*

1. The system and integration tests have been successfully executed
2. The service complies with the technical requirements and measurement requirements.
3. The technical acceptance tests have been performed and the test results do not contain any errors

#### *Explanation:*

The integration test is defined as testing the proper collaboration of the application components. The system test is defined as the end-to-end test of the service to be realized, including the infrastructure test cases.

## Release process

#### *Purpose:*

The purpose of the release process is to determine that the new version of the service meets the (non) functional requirements.

#### *Deliverables:*

1. The service complies with the stated (non) functional requirements and measurement requirements.
2. The acceptance tests have been performed and the test results do not contain any errors.

#### *Explanation:*

The integration test is defined as testing the proper collaboration of the application components. The system test is defined as the end-to-end test of the service to be realized, including the infrastructure test cases.

## Deployment process

#### *Purpose:*

The purpose of the deployment process is to roll out the new service into the product environment in as much automated manner as possible through deployment pipeline.

#### *Deliverables:*

1. The production environment has been adapted through the infrastructure scripts.

#### *Explanation:*

The automated creation of new versions of services is an important aspect of DevOps. The mechanism used for this is the deployment pipeline. In fact, this pipeline is not limited to this deployment process. The pipeline is already starting to set requirements with measurement in a format that can be automated (code process) and ends with automated rolling out of new services and rolling out patches (operate process) to keep that service permanently in compliance with the requirements. Also

the reset of an older version of a service is also part of the deployment pipeline's functionality.

## **Operate process**

*Purpose:*

The purpose of the operating process is to keep the service up and running conform the agreed functionality and quality of the service in order to ensure business continuity.

*Deliverables:*

1. Secure environments and data
2. Executed management tasks
3. Prevent incidents
4. Observed and persistent incidents

*Explanation:*

Keeping the service running properly is actually a multiple process. It concerns both the underlying infrastructure (infrastructure management), the applications (application management) and the data (information management). In order for a service to work properly, many measures need to be taken in the code process. Please note that the resolution of application bugs is not part of this process but of the code process.

## **Monitor process**

*Purpose:*

The purpose of the monitoring process is to monitor the agreed service standards. The scope is not just the production environment but the entire DevOps process.

*Deliverables:*

1. Established monitoring instructions
2. A defined event catalog
3. An equipped monitor facility
4. Determined (potential) deviations in the production standards
5. Registered events
6. Service reports

*Explanation:*

Monitoring the proper operation of the service is actually a multiple process. It concerns both the underlying infrastructure, the applications and the data. In order for a service to work properly, many measures need to be taken in the code process. Please note that the inclusion of incidents is not part of this process but of the code process.

## **Relationship to the DevOps framework**

Figure 2 gives an overview of the relationship between the DevOps process and the DevOps framework as described in the DevOps article.

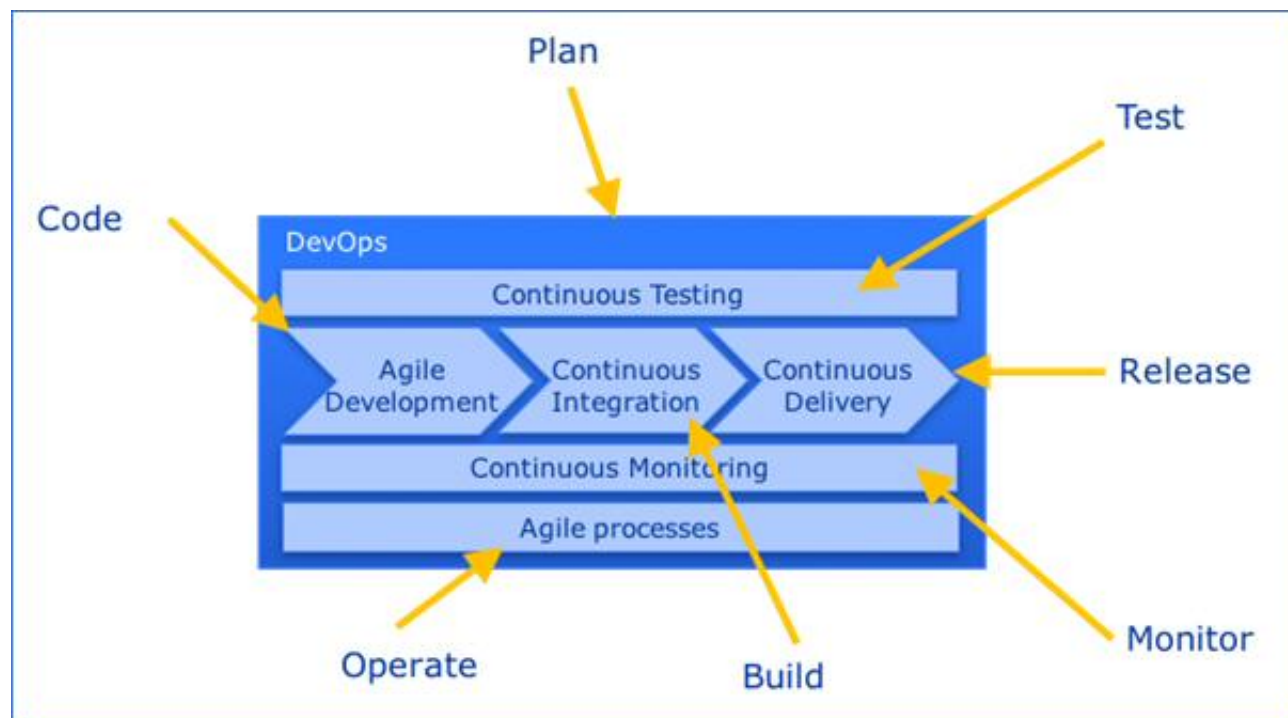


Figure 2. The relationship between the DevOps process and the DevOps framework.

This image is intended to be indicative. There are no sharp lines to draw. But it gives us a reasonable coherence. Here is a brief explanation of Figure 2, indicating in particular that the relationship is not a pure one-to-one relationship.

**Plan:** The plan includes all activities of DevOps, both initially for the entire roadmap and the last incremental delivery of the service.

**Code:** Agile Development mainly covers the aspects that are named in the code process.

**Build:** Continuous Integration mainly includes the build process. Inside the unit tests are also taking place.

**Test:** Continuous Testing is more than the test process as described in this article because it includes the entire lifecycle of all test types such as the unit test cases of the build process.

**Release:** Continuous Delivery is more than just a release rollout. It includes the entire deployment pipeline that is already in Agile Development with the definition of executable test case.

**Operate:** The Agile processes actually include all DevOps processes and not just the operate process. The entire DevOps process is an Agile process.

**Monitor:** Continuous Monitoring includes not only the production phase but the entire DevOps process.

[Exin 2017] Basic training material of DevOps master.



Discuss with us about this article on [LinkedIn](#).

## More information

### Related Books:

[Agile Service Management with Scrum, ISBN: 9789071501807](#)

[Agile Service Management with Scrum in Practice, ISBN:9789071501845](#)

### Related training sessions:

- [DevOps Foundation](#)
- [DevOps in practice](#)
- [DevOps Master](#)
- [Masterclass DevOps Operations](#)
- [Masterclass DevOps Application Development](#)
- [Masterclass DevOps Architecture](#)

**Related Article:**

[Service Management](#)

Summary



Article Name

DevOps process

Description

A process consists of people, resources and methods. Regardless of the uniqueness of the approach of each individual DevOps teams, there are solid patterns in this process. In fact, the laws and regulations force controls to be implemented in this process. These laws and regulations are generic and the controls can be generalized as well.

Author

Bart de Best

Publisher Name

ITpedia