

## DevOps Plan - The deliverables

door Bart de Best - 07-08-2017

<https://www.itpedia.nl/2017/07/08/devops-plan-the-deliverables/>

### Planning Deliverables, Introduction

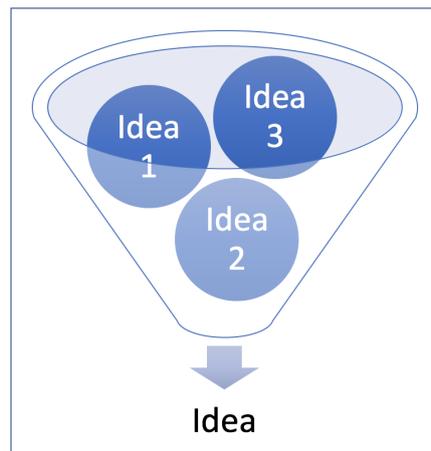


This article describes some of the most important deliverables of a DevOps planning.

### Planning Deliverables, Definitions

#### Funnel

A funnel is a metaphor used to indicate a selection pathway. The graphical representation is a funnel. There are more ideas than can be realized. As an idea gets more shape and the business case becomes clearer, the idea drops into the funnel and finally gets picked up to be planned in the roadmap. The mission, vision, business goal and strategy to realize these are the main driving factors of the funnel. But also aspects that the DevOps organization needs to realize faster time-to-market or higher SLA norms are driving factors.



#### Roadmap

A roadmap is an approach to describe a realization path of a product or service. A product may include an information, application or infrastructure object. A product can be used to offer a service to a customer. The roadmap shows a timeline along which the most important functionalities have to be realized are plotted. The entire portfolio of products and services to prevent bottlenecks has to be taken into account.



### Release planning

Release planning is the basis for the product backlog. The release plan contains the epics that have to be realized. The epics represent the items on the product portfolio. The next release is described in more detail by detailing Epics into features.

### Flow

Within DevOps, often the Lean principles are used. One of the most appealing is flow. A DevOps team should deliver products by working in a particular cadence with no waiting times and no repairs. Therefore, a maximum time to market is realized with the least cost and the desired quality.

### Push and pull

Within DevOps, the pull mechanism is employed that states that the flow is controlled by demand and not by supply. Thus, within supply no stockage is available.

### Technical debt

These terms indicate that choices have been made during the realization that do not deserve the beauty award. The product and / or service is released but require a further redesign to make it future proof. In the worst case, it is a band-aid solution.

## Planning Deliverables, Concepts

### Functional decomposition

This concept describes the idea that the realization of a product or service can take place from coarse to fine. In this way, the realization of ideas in the funnel, roadmap planning, release planning and sprint planning is completed.

## Planning Deliverables, Best practices

More and more organization use Agile principles to create or change the service offering to the business. In many organizations, the product backlog is used as a starting point to collect the features and write them in a user story format. DevOps teams can then realize these features and deploy the created and changed objects in to production. This may be valid for small changes like content management changes. However, for complete new services this approach does not work well in practice. There is a need to have a functional decomposition based on an architecture approach. Often this approach is not appreciated and the slogan “up front definition is not done” is used often. This article explains the benefits of the functional decomposition based on funnel management, a roadmap and a release planning.

### Up front definition is done

Many organizations expect from a DevOps team that the planning only consists of completing the product backlog by the product owner. The top of the product backlog is then refined and contains the highest priority. More downwards, the pieces are bigger and have less priority. In that case, the slogan is often used “up front definition is not done”.

In these organizations, you also see that architecture is only considered per sprint and is formulated incrementally and iteratively. There remains only a kind of solution architect that is directing the DevOps team for the changes that are at hand in the sprint. And often there is not even a role for architecture.

For many simple products and services that need to be realized, this may be sufficient. For complex products and services involving more DevOps teams, or even a complete program has been organized, this is no longer a realistic approach. There must be a functional decomposition.

### Funnel management

An anti-pattern of a funnel approach is an organization in which development teams receive in each sprint new and changed assignments because something has to be realized fast. In this case funnel management does not exist, even no flow exist. In fact, these organizations do not have a DevOps control from a mission, vision, business purpose and strategy. And if there is a direction, then the DevOps teams are disconnected from this direction.

In such a situation, it is important to recognize the benefits of funnel management. The main benefits are located in:

- Funnel management defines the stakeholders that are important in the realization of the business objective. Think of marketing, business managers, architects, HRM managers, Product owners, security officers and serving.
- Funnel management requires a recurring prioritization at strategic level and ensures that shifts in interest per stakeholder are timely signaled.
- The stakeholder cooperation ensures that the involvement and commitment to change remain high.
- The stakeholders can be well guided by the architect to sketch the target architecture that embodies the business objective, strategy and ideas.
- The ideas needed to better shape DevOps can be included in the funnel. For example, the establishment of pipeline deployment in the cloud instead of on-premises (locally on its own server)
- Funnel management offers an opportunity to direct the development of products and services where otherwise development depends on the issues of the day.
- Funnel management offers the ability to define the business case by determining both costs and benefits. The cost is not only the cost of the DevOps teams but also the cost of the countermeasures to mitigate the risks. Risks not only relate to the feasibility and practicality (time and money) but also the realization of SLA norms. By involving the stakeholders up front, it is possible to elaborate or transform delusions into valuable ideas.

Funnel management is for some organizations just a matter of implementing it. Other organizations need a transition strategy. It is important to especially determine why it is necessary to interrupt the sprint and to avoid the flow. A first start may be to distinguish two types of DevOps teams. One team is only driven from the funnel and gets via a roadmap and release planning work while the other team gets only ad hoc business-like incidents, emergency cases, ad hoc developments etcetera. This creates an opportunity for at least one DevOps team to demonstrate the benefits of DevOps.

### Roadmap

A roadmap is drawn up per product or service. The simplest way to compile a roadmap is to draw a diagram with two axes. The vertical axis is the stakeholders and the horizontal axis the quarters in a year. Next, each cell can indicate which functionality for a stakeholder is realized in what quarter.

The classification of functionality and quality in blocks per quarter sounds simple. But that's just a step too fast. From architecture, it is important to check whether the order is logical. There are always certain aspects that need priority because these conditions are preconditioned, such as security in the form of authentication, authorization provisioning, encryption et cetera. However, usually these are standard features that can be used.

Different it is with the interfaces between the other products and services. Each product and service has its own roadmap. The sum of these roadmaps must be overall a harmonious whole. As a product or service occurs in multiple chains, the planning of the roadmap will be more difficult. Therefore, road mapping and funnel management without architecture is impossible.

### Release planning

A release planning is quite easy to realize when the roadmap is already set up. The elements of the future functionality remain vague. The actual benefits are described in the detail before the first release. Although DevOps uses a continuous pull mechanism with a steady flow, a functional decomposition in the form of release planning is very important.

An anti-pattern is an incremental and iteratively generated application whose new features cost more time and quality is deteriorating. This is because the new features do not fit properly with the existing functionality. Often there is no more time for refactoring, where old products and services are re-designed. Release planning can only reserve time to include refactoring of products and services in the planning, as from roadmap planning.

Another anti pattern of release planning is the introduction of technical debt. By including in the funnel and roadmap aspects that are necessary for a good DevOps process and the product / service architecture, technical debt can be avoided. By incorporating architecture through the entire lifecycle from idea to deployment through architecture, countermeasures for technical debt can be taken into account.



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 \[Best 2015\]. 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 Plan - The deliverables

Description

This article describes some of the most important deliverables of a DevOps planning. Release planning is the basis for the product backlog. The release plan contains the epics that have to be realized. The epics represent the items on the product portfolio. The next release is described in more detail by detailing Epics into features.

Author

Bart de Best

Publisher Name

ITpedia

Publisher Logo



ITpedia