DevSecOps

**PREPARED FOR**

Erik van der Schriek

Niko Kuijpers

Gertjan Schouten

**PREPARED BY**

Maarten Blömer

Nick Krijgsman

Faruk Aydin

Jursley Gonzalez

Vincent Andersen

# Table of Contents

[**Table of Contents**](#_z6lhuq9emniv) **1**

[**Case Study**](#_k6jx82pctwds) **2**

[Case goal:](#_bpxkra1gvikb) 2

[Case problems:](#_tz830mgo6nzk) 2

[Case questions:](#_3vn6lbsuph60) 2

[Formulate learning objectives](#_58tw4qf8lsdh) 2

[Brainstorming:](#_llbf4q7qnu37) 3

[**Independent study**](#_zc6h8lveooxw) **5**

[Subject (Faruk)](#_eaou76j8xww7) 5

[Subject (Maarten)](#_97ho52d59gzk) 5

[Subject (Vincent)](#_v2xj76ghlwj7) 5

[Subject (Nick)](#_qo5qk3cr0w3h) 5

[Subject (Jursley)](#_505gm0evad4t) 5

[**Matrix**](#_cg2taz46zkvz) **6**

[**Sources**](#_qwibiw7iqp2l) **8**

# Case Study

## Case goal:

*Your task is to come up with* ***a plan with actionable items*** *that you can use* ***to improve*** *the added value of* ***the SDLC approach*** *for your group project.*

## Case problems and questions:

### Problems

* The runtime of the test suite is large (e.g. a couple of hours) and sometimes tests fail without any known reason;
* In case of reported issues, both teams have a hard time to figure out which team is responsible for investigating and fixing the issue;
* It is not clear when and what features and bug fixes are released;
* Team is unable handle security threats (e.g. a recent SQL injection attack could have been avoided);
* It takes too long for the company to resolve production issues (MTTR is too high); it is not clear when (and if) a bug was solved and in what conditions it was appearing in the first place; the company already paid fines for not complying to SLA with the clients;
* The client has the feeling that the current change request procedure via the service desk doesn't add any value to the project;
* The development team members feel a lot of pressure, which reduces their focus on delivering software of good quality and they are starting to cut corners.

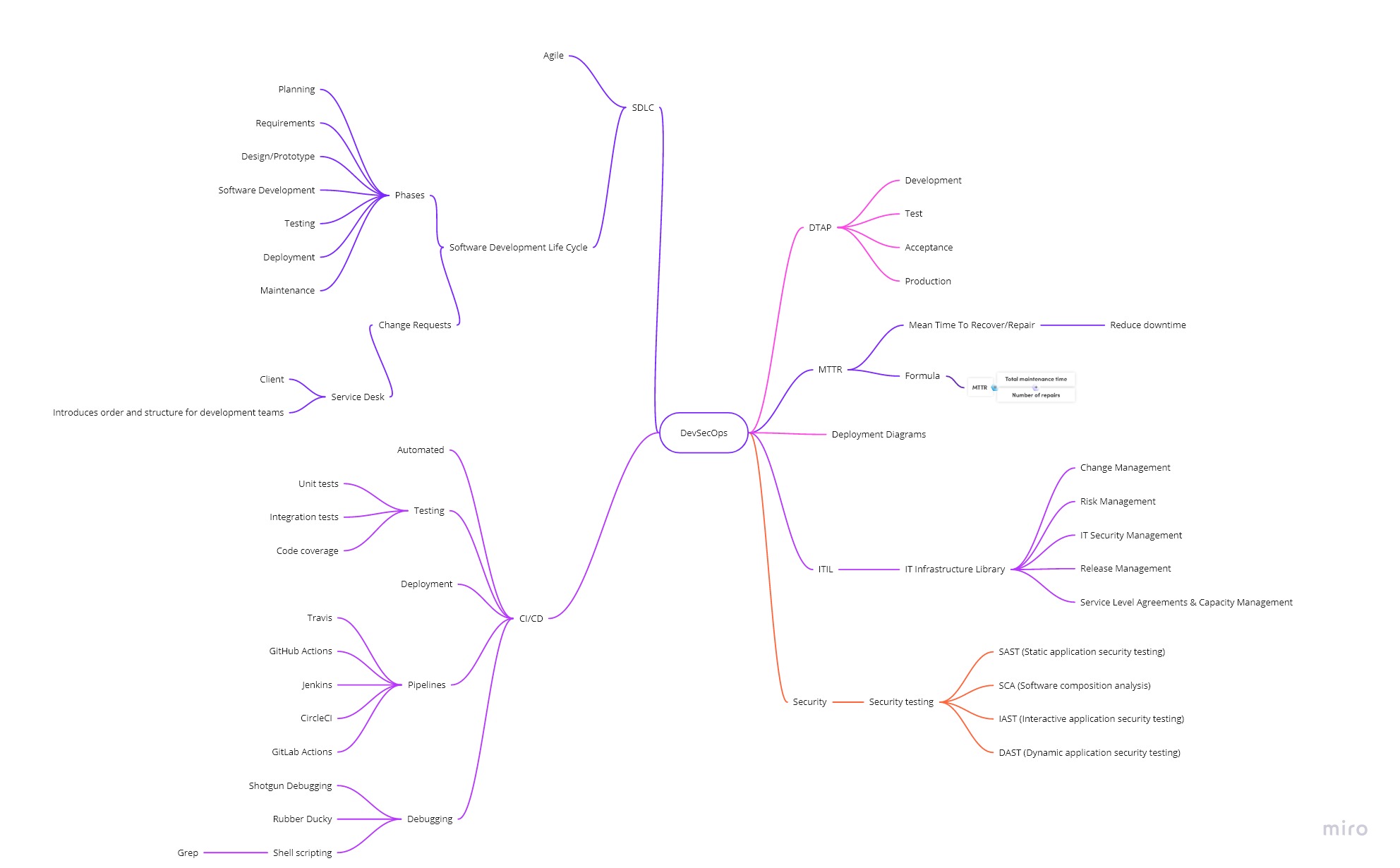
### Questions

* What is SDLC exactly and what are typical challenges in the field of SDLC that organizations encounter in large scale projects?
* What is a change request procedure, and why do the company and client need to follow this procedure?
* What is MTTR and why is this important for the client?
* Are there any existing tools, techniques, patterns or best practices that could be used in this case?

## Case research questions:

1. What is release management and how does it look like in our project? (Maarten)
2. What is change management and how does it look like in our project? (Vincent)
3. What is risk and security management and how does it look like in our project? (Faruk)
4. What is capacity management and how does it look like in our project? (Nick)

## Brainstorming:

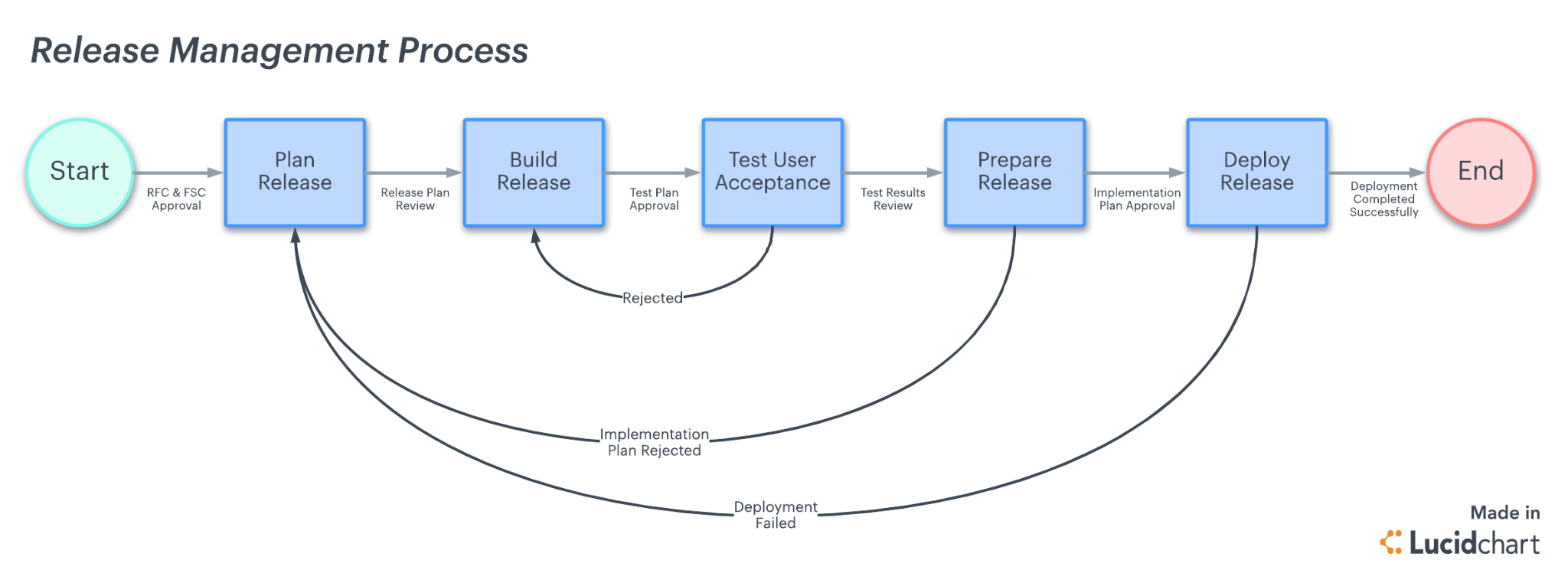


# Independent study

## What is release management and what does it look like? (Maarten)

### Theory

“Release management oversees all the stages involved in a software release from development and testing to deployment. Release management is required anytime a new product or even changes to an existing product are requested.”



Similar to how scrum works in sprints, within each sprint there’s a cycle to be followed. The release cycle usually follows the following steps: planning, build, test, prepare, deploy.

It is important to firstly start figuring out what the stakeholders want and need. What things are required for the project such as tools and time constraints. In other words, clarify business needs. This is followed by planning and staging the project into different segments, creating user stories, creating a release risk profile, and assessing how changes will interact with each component in the application. It is important that all parts of the application are tested during development, accessing risks, and dependencies. This should also ensure that you publish a fully tested application. Lastly make sure to plan for the future. When you release a new version you should be preparing to improve and expand on it.

### Our plan

We already implemented most of this by using Jira to plan out our tasks in a backlog to keep track of everything. For each sprint we take the features that need to be developed, build then, test them and prepare them for release.

## What is change management and how does it look like in our project? (Vincent)

### Theory

The ***purpose*** of ITIL Change Management is to control the lifecycle of all changes and help in implementing beneficial changes with minimum disruption to IT services.

ITIL defines a *change* as:

*“the addition, modification of removal of anything that could have an effect on IT services"*

Where they distinguish between 3 different types of changes:

1. Standard Changes: Pre-authorized, low-risk Changes that follow a well-known procedure.
2. Emergency Changes: Changes that must be implemented immediately, for example to resolve a Major Incident.
3. Normal Changes: All other Changes that are not Standard Changes or Emergency Changes.

The ***objective*** of this process is to:

* respond to business and IT requests to ensure alignment of services with business needs.
* ensure that changes are introduced in a controlled manner, optimising business risk.
* ensure timely and successful change implementation to meet business needs.
* Use standard processes and record every change.

The ***scope*** of this process is to handle:

* changes to architecture, tools, metrics, processes and documentation.
* Addition, removal or modification of a service or a CI or associated documentation.
* Changes to any of the five aspects of service design

With this in mind the goal of ITIL change management can be summarised as:  
“Responding to the customer’s changing business and while doing this maximizing value to the customer and reducing incidents, disruption, and re-work.”

### 

### Our plan

We ourselves don’t have an assigned role or team for the change management that handles change requests for us. We ourselves evaluate and add any necessary changes or requests for changes into the backlog of our Jira project.   
In order to improve in our current situation we could be asking ourselves some questions when a change request is made:

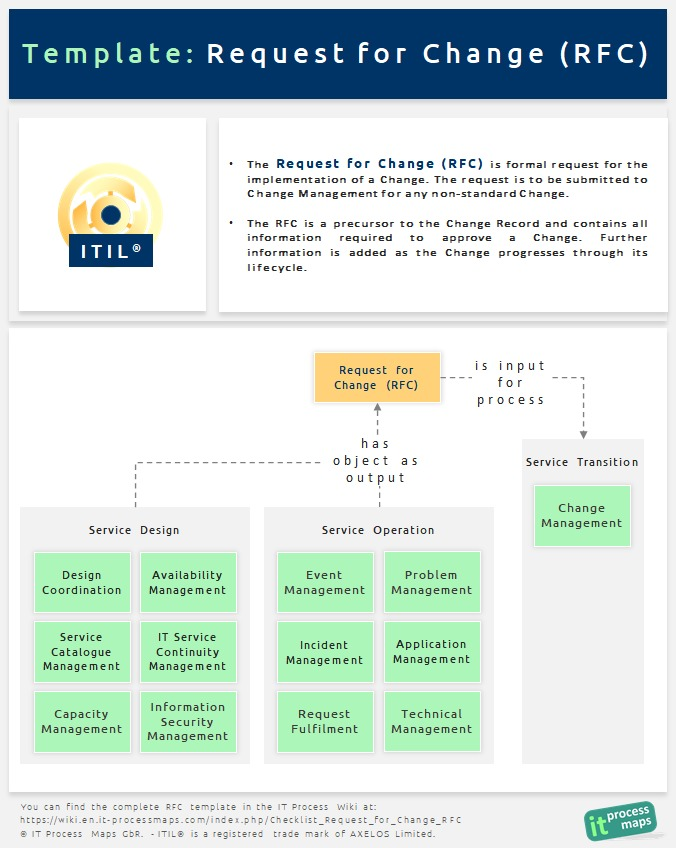
* Who raised the change?
* What is the reason for the change?
* What is the required return?
* What are the risks involved?
* What resources are required to deliver the change?
* Who would be responsible for building, testing and implementing the change?
* What is the relationship between this change and other changes?

Asking ourselves these questions would broaden our view of the impact on our process and our application when we would implement the change.

In order to measure and report performances or a process, IT service or activity we could use a Key Performance Indicator. These can be as well defined as one must have them but for now one of the standard ones to measure the overall performance of an RFC process is:

## 

|  |  |
| --- | --- |
| **Key Performance Indicator (KPI)** | **Definition** |
| **Number of Major Changes** | Number of major changes assessed by the CAB (Change Advisory Board) |
| **Number of CAB Meetings** | Number of CAB (Change Advisory Board) meetings |
| **Time for Change Approval/ Rejection** | Average time from registering an RFC with Change Management until a decision on the RFC is reached (i.e. until it is either approved or rejected) |
| **Change Acceptance Rate** | Number of accepted vs. rejected RFCs |
| **Number of Emergency Changes** | Number of Emergency Changes assessed by the ECAB (Emergency Change Advisory Board) |

Within the context of a bigger company where there would be multiple teams available we simply could use one of the RFC templates such as:  


The steps would then be followed within the diagram where each responsible party will handle the RFC within their dedicated area of expertise which would then produce a result of acceptance or possible rejection of the RFC.

## What is risk and security management and how does it look like in our project? (Faruk)

### Theory

“Security Risk Management is the ongoing process of identifying these security risks and implementing plans to address them. Risk is determined by considering the likelihood that known threats will exploit vulnerabilities and the impact they have on valuable assets.”

5 steps to manage a risk:

1. Risk identification

State the source of the risk.

1. Risk analysis

Now that you know what the risks are, you need to consider the likelihood and impact of these risks.

1. Risk evaluation

Prioritize investments where it is most needed and conduct estimates based on likelihood positioning.

1. Risk treatment

The work you are doing to end a risk completely or manage and tolerate the risk.

Also known as ‘risk response planning’

1. Monitor and review the risk

Describe the processes for monitoring and review. Collect feedback on each risk and evaluate the steps. Get to know what to improve.

### 

### Our plan

We currently do not have a set procedure for this. We mainly use the OWASP risks as risk sources. In order to have more insight into possible risks, I suggest that we brainstorm about possible attack options. This can be done by looking at code, but also by thinking about which structure we maintain during the realization of the project. After we know what the possible risks are, we can continue with step 2 of the 5 steps above.

## What does capacity management in our project look like? (Nick)

### 

### Theory

First things first to know how we apply or going to apply capacity management in our project, we need to know what is capacity management.

*“Capacity management aims to ensure that the capacity of IT services and the IT infrastructure is able to deliver the agreed service level targets in a cost effective and timely manner. The Capacity Management process considers all resources required to deliver the IT service, and plans for short, medium and long term business requirements.“*

Capacity management has four sub-processes which allows for more clear objectives.

* **Business capacity management**; create capacity and performance requirements based on the business needs and plans. This also includes future plans and should ensure that capacity and performance needs can be fulfilled.
* **Service capacity management**; predict, manage and control the performance and capacity of operational services. This requires proactive and reactive measures to ensure the SLA is met.
* **Component capacity management**; predict, manage and control the performance of IT resources and individual components.
* **Capacity management reporting**; provide information about capacity, utilization and performance.

### 

### Our plan

In our current situation we haven’t talked about capacity as much. As we didn’t run any tests on capacity and performance yet. Things that we have implemented are: Prometheus, to check on the load of certain services, and a capacity limit within the pipeline. The capacity limit is there to make sure we stay within our budget.

**Prototype**

## Prototype infrastructure

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## Tool List

|  |  |
| --- | --- |
| **Tool** | **Description** |
| **Kubernetes** |  |
| **Github** |  |
| **Docker** |  |
| **Azure Pipelines** |  |
| **Prometheus** | Monitors load on services |
| **Helm** |  |

# Sources

|  |  |
| --- | --- |
| **Researcher** | **Source** |
| Maarten | https://www.lucidchart.com/blog/release-management-process |
| Nick | https://wiki.en.it-processmaps.com/index.php/Capacity\_Management |
| Nick | https://www.lucidchart.com/blog/itil-capacity-management-service-design |
| Maarten | https://www.plutora.com/software-release-management/what-is-release-management |
| Vincent | https://wiki.en.it-processmaps.com/index.php/Change\_Management |
| Vincent | https://wiki.en.it-processmaps.com/index.php/KPIs\_Change\_Management |
| Vincent | https://wiki.en.it-processmaps.com/index.php/Checklist\_Request\_for\_Change\_RFC |
| Vincent | https://blog.masterofproject.com/itil-change-management-process/ |
| Vincent | https://www.cherwell.com/it-service-management/library/essential-guides/essential-guide-to-itil-change-management/ |
|  |  |