**Case study DevSecOps & Software Development Lifecycle**

Author: Leon Schrijvers

Date: December 3, 2019

A software development agency is working for large corporations in The Netherlands, e.g. financial institutions, airliners and large e-commerce platforms. The projects that are implemented for these clients, tend to be pretty complex from a technical point of view. Furthermore, for clients, these large-scale projects can be seen as a long-term investment for the client and therefore, the project is required to be actively developed and maintained for a large period of time (e.g. 5 to 10 years).

In these years the teams will likely change as developers move to other projects and companies. To be able to do ensure long support, the development team is following the Agile methodology in combination with the following software development lifecycle (SDLC) procedures:

* There are two teams involved in the project: a development team (responsible for development of the project) and an operations team (responsible for running the project in production);
* The development team uses a *development environment* which is used to develop and maintain the application and a *production environment* where the latest version of the application is deployed to;
* The development team uses a simple branching strategy: active development is done on the *development* branch; this branch is merged to the *master* branch for new deployments;
* The development team has setup a CI/CD pipeline, which contains an automated test set consisting over 1.000 unit tests and integration tests, and can deploy the application to the production environment;
* To streamline the process of receiving and handling change requests (reported issues, new requirements, etc.), all clients are required to report these requests to a central service desk. This service desk prioritizes all input and dispatches it to either the development team or the operations team.

Originally, these procedures were introduced so that developers could easily maintain the application and add new features without worrying too much about breaking the project. However, the development team faces several challenges:

* 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 treats (e.g. 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; 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 quality software and they are starting to cut corners.

As a consultant, you are hired by the company to help them to improve their SDLC approach. To prepare yourself for this task, you start to do some research so you can answer the following 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?

Your task is to come up with a plan with actionable items that the company can use to improve the added value of their SDLC approach. Your plan will contain recommendations in various topics:

* Operational and project managing changes;
* Security development;
* Use tools/techniques differently;
* Expectation management and communication with the client.

On a more technical side provide:

* A schematic overview of you DevSecOps setup including for instance but not limited to deployment diagrams, release management schematics, infrastructure, et cetera.
* A list of tools and techniques that you think will be important for the success of this project
* A working prototype which demonstrates how the chosen technology contributes to handling your chosen automation challenge