

# **CYBERSECURITY E-DEGREE**

## **DevSecOps Policy Creation**

Provide an existing company which has a strategic goal - i.e., Increase Revenue, Optimize Value to Clients etc.

- List the pain points or the drivers to this requirement

Planning - understand what is the current process and tooling that they have i.e. Planning, ALM, Integrated Development Environment, Deployment Tools, Monitoring Tools

Find the gaps of the Automation Tools - Plan, Build, Test and Release

Expectations

1. is to be able to fill the gaps in tooling to provide a full automation and improve speed, acceleration and release of software

2. how much improvement has been imposed with the new proposed system against the current

**COMPANY NAME: Codeplay Software Limited**

**Location: Scotland, Edinburgh; United Kingdom**

### **Pain Points:**

- 1) **Business Demands (UX/UI):** This business needs great application experience (ease of use), fast response time, efficient cross-collaboration and reduced IT cost.

If the UX/UI (User Experience and User Interface) is poorly planned, then the users may find the app very difficult to use. Application developers and

interface designers should be proactive and be on the lookout right from day one when the app has been released to the masses for problems with the UX/UI and not waiting for customers' complaints.

There is a lack of measuring the components that make up the UX/UI and so often, companies may release the application (Web, Mobile, Desktop) for all to download onto their devices. This would have a paramount effect on the company in terms of fewer sales, bad reviews, angry customers, etc.

- 2) **Software Development Lifecycle Disruptions:** DevOps is a software development phase that bridges the gap between software development and IT operations by improving the relationship between the two. If some reason **the organization still use old software lifecycle** (Waterfall, Iterative) to deliver applications to enterprise clients, there is bound to be a problem. The problems include:

- Poor resource planning
- Limited Documentation
- Fragmented Output
- No Finite End – No clear vision of what the final product would look like
- Difficult to measure the progress of the overall application lifecycle

- 3) **Application Management:** There is a problem with checking whether to use commercial or open-source tools for application development and management. That a particular software is free does not mean that it can provide ease of use for the developers using it and/or help deliver and meet the organization's goals to serve the customers effectively and efficiently. This brings the question of whether the data it generates can help to make better decisions in the organization.

**A lack of test resources** can also be a pain point. Modern software now relies on **codeless testing and bots** for testing purposes. This can have

effects in extending the wait times for software to be developed, tested and released on different devices.

- 4) Security – Lack of Security especially in the back-end servers. This cannot be taken for granted. Databases can be hacked by an adversary via **SQL Injection**.

## **SOLUTIONS**

- 1) Security with modern software solutions should be built into every software development during the planning and development stage rather than when it has been released. The security should be able to maintain the **CIA** (Confidentiality, Integrity and Availability) of customers' data.
- 2) Organizations should always strive to have the right tools and the right people for the job to meet modern software development demands. The use of **Containerization** and **Infrastructure-as-a-Code** is important. This would help in continuous planning, continuous testing, and continuous delivery within the organization.
- 3) Organizations should have autonomous teams where software developers and operations teams should be able to make decisions and apply changes without a convoluted decision-making process.

## **DevSecOps Components – People, Process, and Tools**

- 1) **People:** The people (employees of an organization) will make DevOps successful within the organization. Digital transformation employed in modern organizations delivering goods and services that are agile, available and scalable with continuous integration requires new technologies. But these technologies require people (commonly the IT Personnel) with the right skill set to use these technologies in the organization. Furthermore, the proper IT personnel also need soft skills to deliver.

2) **Process – Tooling:** With the right technology in place, the process of properly delivering the product would be achieved. Organizations should have a customer-centric attitude towards their product. The following is the process involved for the organizations to use the automation tools to deliver:

a. **Planning:** There should be a **Planning Portfolio Management**. This is a planning tool. When an organization has an idea or a goal to improve on their goods and services or to create new software for their customers – the question is, how will they achieve it? They need market research and creating models (probably database models in the case of the database design paradigm) before presenting the idea to the executives for approval for implementation to go ahead. The design process involves when the developers and IT operations team collaborate.

b. **Building and Testing:** The software developers would use a good Integrated Development Environment (IDE) to build the application. This may include **Visual Studio**, **Android Studio** (for android, C++, etc.), **Eclipse**, **IntelliJ IDEA** etc. Such tools should have components for automation to **Git** and **GitHub** for code repositories and to track changes to any set of files that are updated frequently.

**Quality Assurance** is very important during the development stage.

Continuous testing e.g., **Functional testing** would help to know whether the input would deliver the right or expected output.

Automated software testing tools the company can employ for this purpose include **Unified Functional Testing**, **AccelQ**, **AVO Assure**, **Test Complete**, **Test Rigor** etc. The next question is whether the software can scale. **Performance Testing** should be performed to determine the speed and stability of the software program under a heavy workload. Some of the examples of automated performance test tools that the organization can use could be **IBM Rational Performance Tester**, **WebLOAD**, **LoadNinja**, **HeadSpin**, **LoadView** etc.

Finally, there should be **Security testing**. This involves testing the applications (including the network) for any vulnerabilities that might

hinder the success of the application. This is to avoid exposing customers' data. Examples of such tools to be used can be **Wireshark, Burpsuite, Nmap, SQLMap, Zed Attack Proxy (ZAP), Google Nogotofail** etc. This is to ensure that the customer's data are strictly confidential, have integrity and are always available when needed.

- c. **Delivering/Release:** Release is the set of changes that are created to be delivered to the production environment. Release activities refer to things like **Configuration Management, Release and Deployment Management, Designing, Planning, Rollout Planning and testing communication**. Consequently, there is a release process that makes this possible and a release package that contains the code and artefacts needed to make the desired change.

The **release management tools** help with the development, testing, deployment and support of software releases. Delivery of software has changed over the years from floppy disks to CDs, one-off downloads (Waterfall and Agile) and finally cloud technology (SaaS -Agile and DevOps which involves constant responsiveness to customers' feedback and monthly subscription which finance further development).

The **Release Management process flow** for introducing changes into the IT environment includes – approved changes, release planning, release building, acceptance testing, release preparation and Release and Deployment.

Furthermore, the release automation tool to achieve the above includes - **Microsoft Azure DevOps, GitLab, RedHat Ansible Automation Platform, Digital.ai Release, AWS Amplify**, etc. These tools provide the organization with a collaborative culture that brings together developers, Project Managers, Marketers and contributors to develop software. They can automate testing, code changes and releases and pipelines for continuous delivery of the applications.