FIT5057 Assignment 1

Literature Review Report

Student ID: 33475881
Name: Peichun Shih

Table of Contents

Executive Summary

This report outlines a clear purpose, objectives, and the scope of discussion. The aim is to recommend a solution to develop a secure landing page and to equip project designers and stakeholders with insights to guide decision-making. The definition of key concepts, including Shift Left, DevSecOps, Zero Trust, SWEBoK, and the V testing model, are provided to illustrate the escalating importance of security across every phase of the software development lifecycle.

In addition, the report furnishes a mind map to visualize the interconnection among the key concepts with succinct explanation. Based on these concepts, the suggestion of integrating OAuth 2.0 to APIs is described to enhance the security of the landing page.

The framework formed by the combination of these concepts largely improves cybersecurity and facilitates the construction of more trustworthy APIs. The insights garnered from this report will empower the project designers to develop a more secure landing page with the best practices of the industry.

1) Report Purpose

The report purpose is to address the challenge of developing a secure landing page by adhering to the project governance standards, including Shift Left paradigms, DevSecOps, Zero Trust, SWEBoK, the V testing model, and applying OAuth 2.0 standard.

2) Report Objective and Discussion Scope

The report objective is to define, analyse the key concepts, and synergise the concepts to establish a clear understanding of the interrelationships among them. The understanding will then be leveraged to devise an effective and efficient solution. It discusses how the optimized application of Shift Left methodologies combines Zero Trust within API design to ensure safe access and data protection.

The report provides a structured approach with the profession insights to make well-informed decisions.

3) Concepts Definition

a. Shift Left

In DevOps, Shift Left testing refers to moving the evaluation of testing to the development stage. Accordingly, testing is implemented at the very beginning of the project (Miller, 2015).

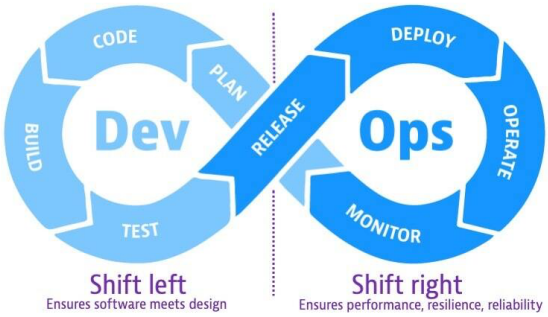


Diagram 1. The illustration of DevOps, Shift Left, and Shift Right (Gunja, 2023)

Shift Left testing makes it possible to detect errors at every stage of the development process. In addition, it is used to ensure that the software fulfills customer requirements, which helps to improve customer experience and software functionality as well as reduce cost (Gunja, 2023). Shift Left further results in

DevSecOps since DevOps teams are devoted to integrating security from the start of SDLC (*What Is DevOps? | IBM*, n.d.).

b. DevSecOps

DevSecOps means DevOps incorporating security within each stage of the agile DevOps lifecycle (*What Is DevOps? | IBM*, n.d.).

DevSecOps strives to solve security issues within earlier phases since it is less complex and less expensive to address these issues. In addition, DevSecOps makes development, security, and operations teams share the responsibility of software security and automates the process of delivering software without postponing the software development cycle (Marsal, 2023).

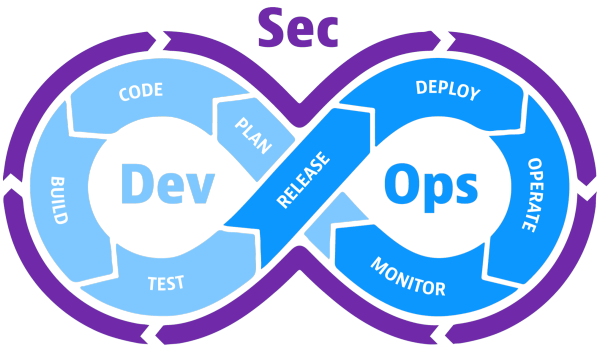


Diagram 3. The illustration of DevSecOps (Marsal, 2023)

c. Zero Trust Architecture

Zero Trust is a critical model for cybersecurity, which offers an identity-centric model for access control (Garbis & Chapman, 2021). It grants trust based on resources (e.g., users, assets, data, applications, servers) instead of the physical network location (Chandramouli & Butcher, 2023). Moreover, Zero Trust implementation guarantees that only if clients are granted access authorisation, they are allowed to connect to the service (Garbis & Chapman, 2021). It also involves strict control on devices for tracking the number of different devices accessing the network and making sure that each device is authorised (GlobalDots, 2022).

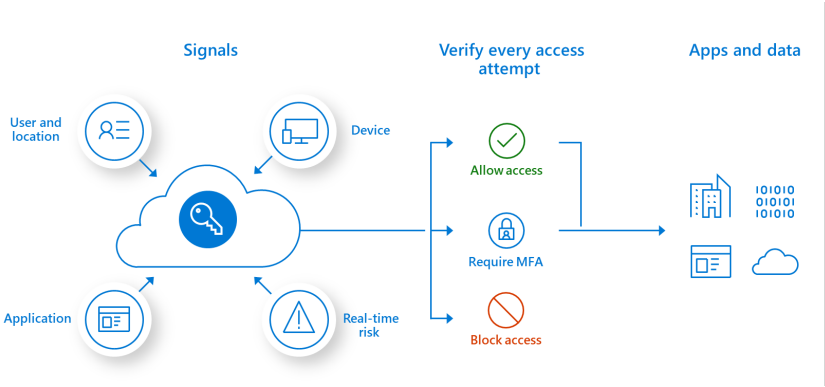


Diagram 4. The illustration of Zero Trust Mechanism (GlobalDots, 2022)

d. SWEBok Software Testing Activities

SWEBoK is short for *Software Engineering Body of Knowledge*, which describes organized knowledge on software engineering process. It manages to clarify and specify the scope as well as the content of software engineering with the aim to establish a globally consistent view toward software engineering.

Software testing in SWEBoK Guide comprises software testing fundamentals, test levels, test techniques, test-related measures, test process, and software testing tools to demonstrate various types of testing and testing techniques or tools generally adopted in the testing process (Bourque & Fairley, 2014).

e. The V Testing Model

The V testing model is established on waterfall development cycle by linking each stage of activities to the corresponding testing usually implemented in the later stage. According to diagram 5, it shows the V shape with the development activities on the left and the corresponding validation on the right (Firesmith, 2013).

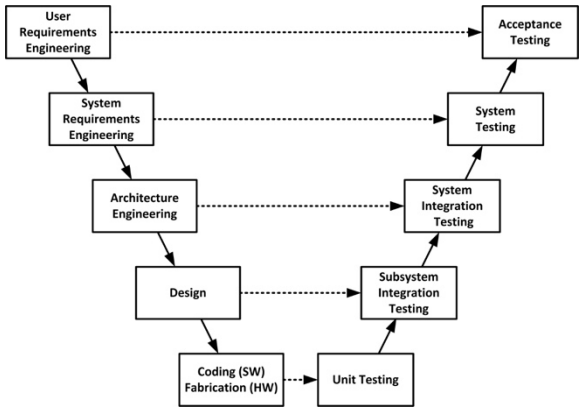


Diagram 5. Traditional Single V Testing Model of System Engineering Activities (Firesmith, 2013)

The V model indicates that testing should start as early as possible in SDLC. However, the simple V model comprises sequential phases, which makes it hard to fit in the agile environment that contains incremental and iterative process. Consequently, this gives rise to two other variants —the double V model and the triple V model to adapt to the agile environment. The double V model also tests every executable work product, and the triple V model further verifies the testing work products (Firesmith, 2013).

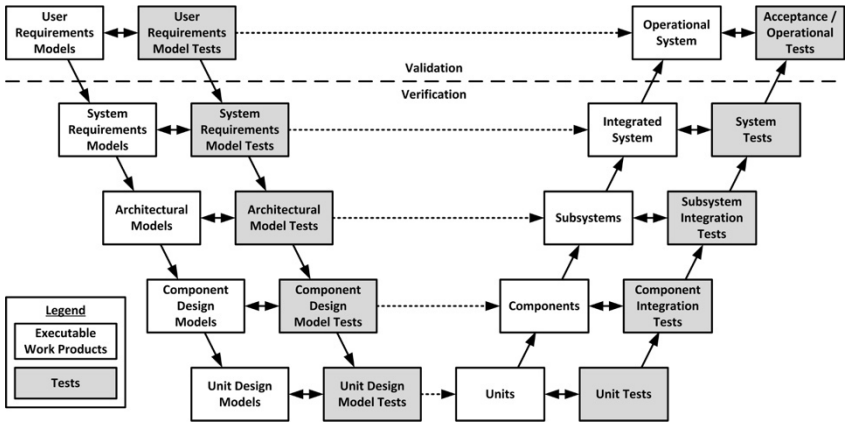


Diagram 6. The Double V Model of Testable Work Products and Corresponding Tests (Firesmith, 2013)

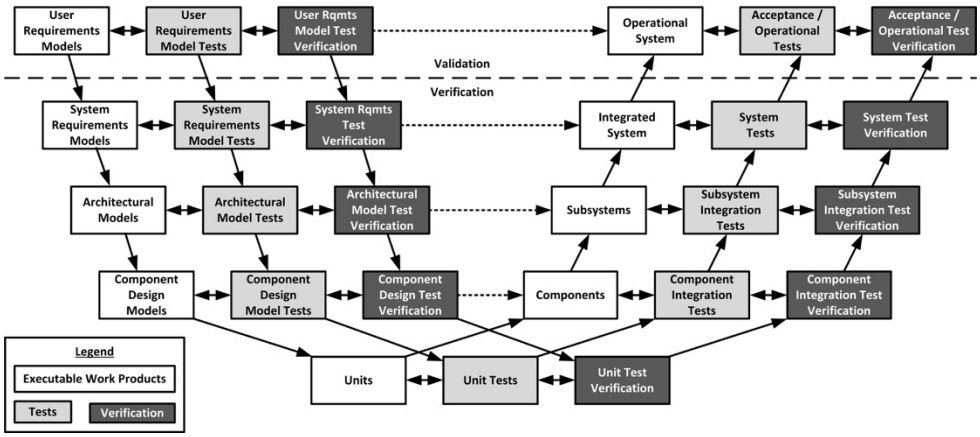
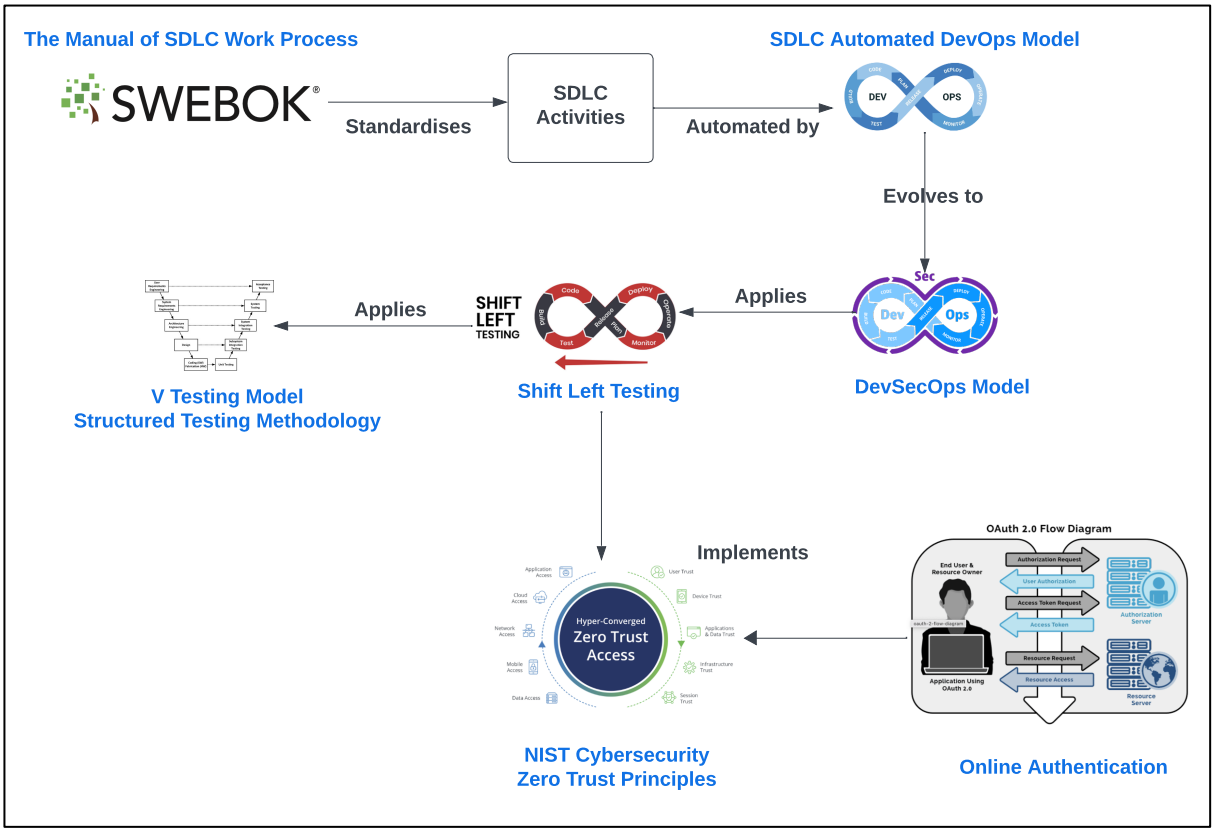


Diagram 7. The Triple V Model of Work Products, Tests, and Test Verification (Firesmith, 2013)

4) Concepts Interrelationship Summary (mind-map)

Regarding the definition demonstrated, the mind map of the concepts is illustrated as follows.



The interrelated key concepts create a holistic technique to manage software development and security. It starts from SWEBoK to standardise the activities. The development and operation teams then collaborate to work through the entire process. In DevOps, testing and security become the focus, leading to DevSecOps and Shift Left testing. Shift Left testing is applied to waterfall methodology, giving rise to the V testing model. Due to the adoption of security-prioritised mechanism, Shift Left testing and online authorisation within APIs utilise Zero Trust architecture to manage risks and security from planning to deployment. On the whole, more secure software products will be created by incorporating these concepts.

5) Recommended Writing Sections and Brief Description

To secure the landing page, applying OAuth 2.0 is recommended.

I. What is OAuth 2.0 Standard?

OAuth 2.0, short for “Open Authorisation”, is designed to secure API authentication and authorisation. It allows a third-party website or application to

represent a user to access resources, which are owned by other web apps without sharing users' credentials (*What Is OAuth 2.0? - Auth0*, n.d.).

OAuth 2.0 is implemented based on Zero Trust architecture. An access token is used as the authorisation to access resources on behalf of the user. Adopting OAuth 2.0 to develop the landing page guarantees more reliable API.

II. How is OAuth 2.0 Applied in the Weverse Login and Platform Service Access?

According to the flow of OAuth 2.0 authorisation (*What Is OAuth 2.0? - Auth0*, n.d.), the overview of applying OAuth 2.0 in Weverse landing is as follows:

- As a client application, Weverse is registered to the authorisation server (e.g., Google, Apple, or Kakao), and it will receive a client identifier and client secret as the trust between Weverse and the authorisation server.
- For example, if a user wants to login to Weverse platform with the Google account. Weverse will redirect the user to Google's login page, and the user will provide the credentials. After Google authenticates, the scopes that Weverse request will be presented to the user.
- If the user chooses to grant access, Google will generate an access token (or an authorisation code) for Weverse. Weverse exchanges the access token to Google, requesting to access the user's resources, such as user data, on Google platform.
- Since an access token has a lifespan, Weverse can request a new access token without the user's reauthentication. The always-refreshing token ensures ongoing and secure access.

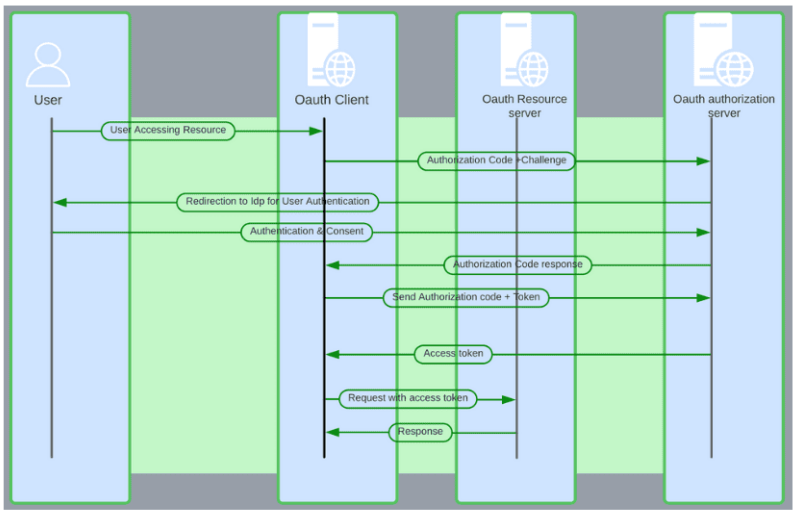


Diagram 7. OAuth authentication and authorisation mechanism (*Unlocking Zero Trust: The Role of OAuth in Security Architecture*, n.d.)

III. How are OAuth 2.0 Capabilities Tested?

Based on the key concepts mentioned, testing is required throughout entire SDLC to ensure OAuth 2.0 components and flows work as intended. For example:

- **Unit testing** should be adopted to test individual components, including testing the authorization server, token issuance, validation, and so on.
- **Integration testing** would be applied to test each function to make sure that each flow works correctly and conforms to the specification, such as the flow of access token operation, user authentication flow, and client credentials flow.
- **System testing** ensures that the entire system can be deployed smoothly. For example, it helps to check the compatibility and consistency of OAuth 2.0 implementation across different client types (e.g., web application or mobile apps), evaluate the performance, verify the error handling mechanism, as well as test the security vulnerabilities.

6) Conclusion

The synergy among these key concepts forms a holistic framework to secure API functions in this project. Comprehensive and early testing contributes to identify and address issues in a controlled environment. It allows the implementation of OAuth 2.0 to be secure and reliable, and further ensures that the APIs are trustworthy as well as aligned with the governance standards of the project.

References

Miller, S. (Host). (2012-present). *SEI Podcast Series* [Audio podcast]. Software Engineering Institute. https://www.sei.cmu.edu/publications/podcasts/index.cfm

Gunja, S. (2023, February 7). Shift left vs shift right: A DevOps mystery solved. *Dynatrace News*. https://www.dynatrace.com/news/blog/what-is-shift-left-and-what-is-shift-right/

Marsal, J. (2023, June 1). What is DevSecOps? And what you need to do it well. *Dynatrace News*. https://www.dynatrace.com/news/blog/what-is-devsecops/

Garbis, J., & Chapman, J. W. (2021). What is zero trust? In *Apress eBooks* (pp. 7–18). https://doi.org/10.1007/978-1-4842-6702-8_2

Chandramouli, R., & Butcher, Z. (2023). A Zero Trust Architecture Model for Access Control in Cloud-Native Applications in Multi-Location Environments. *NIST Special Publication*.

GlobalDots. (2022, January 30). *Zero trust explained*. https://www.globaldots.com/resources/blog/zero-trust-explained/

Bourque, P., & Fairley, R. E. (2014). Guide to the Software Engineering Body of Knowledge (SWEBOK(R)): Version 3.0. In *IEEE Computer Society Press eBooks*. https://dl.acm.org/citation.cfm?id=2616205

Firesmith, D. (2013, November 11). *Using V Models for Testing*. Software Engineering Institute. Retrieved August 24, 2023, from https://insights.sei.cmu.edu/blog/using-v-models-for-testing/

R, B. (2018, November 14). Begin with Shift Left Testing - QA touch. *QA Touch*. https://www.qatouch.com/blog/begin-with-shift-left-testing/

What is OAuth 2.0? - Auth0. (n.d.). Auth0. https://auth0.com/intro-to-iam/what-is-oauth-2

Unlocking Zero Trust: The role of OAuth in Security architecture. (n.d.). WWT. https://www.wwt.com/lab/unlocking-zero-trust-the-role-of-oauth-in-security-architecture