# Dacoco Security SDLC Review

OBJECT COMPUTING

YOUR OUTCOMES ENGINEERED

Prepared by:

Janelle Morris
Brandon Lynch

OCI

Apr 13, 2022

# Revision History

| revision | State | date | name | description |
|---|---|---|---|---|
| 1.0 | Original | April 13, 2022 | Janelle Morris/Brandon Lynch | Initial document creation |
| | | | | |

# Table Of Contents

# Overview

OCI has been asked to provide information on best practices for security during the software development lifecycle (SDLC). A secure SDLC is important because it prioritizes the security of software and helps make sure malicious actors do not target the applications. Security threats are too complex to simply patch issues following release, so it's more efficient and effective to integrate security during the coding process.

The WAX blockchain is the basis for the AlienWorlds game. A smart contract security review is underway in a parallel effort. In addition, there is an ongoing effort to review and determine the long-term architecture strategy for the platform which is also a separate initiative. This document will focus solely on secure software development lifecycle best practices.

# Current State

A limited interview was conducted with Dacoco to gain a general understanding of the current state of the SDLC. This information will inform the best practices recommended within this document. In general, Dacoco is in a state of transition. The application has significant technical debt coupled with the need to deliver business value and new features. The initial developer of the system is no longer engaged in development activities. The current team is working to identify and eradicate the technical debt while implementing new features to the system.

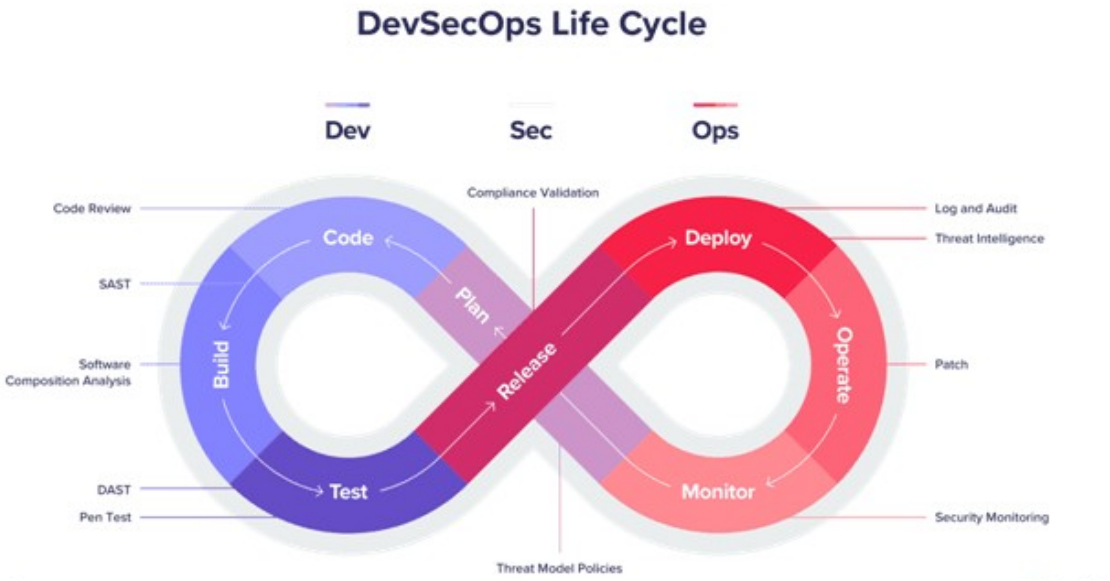
The team is transitioning to agile development from a more waterfall approach. This transition is hampered by the monolithic nature of the current system. In addition, most of the procedures in the SDLC are manual, although the team is working on adding controls. The team is also small in size with single points of failure. For instance, only one system administrator exists to manage the system and the infrastructure.

Security, code reviews, and testing are largely manual. The team is working to automate processes and procedures as time allows. The team is also working on breaking down the monolith into more manageable code modules. Key management currently requires three of the four people who have access to manage the process. However, it is largely based on the trust of one of the three members to be managed effectively.

Overall, the technical team is largely aware of the current issues with the system and understands the necessary improvements that need to be made.

# Recommendations

Overall, OCI recommends that Dacoco implement a DevSecOps Lifecycle into its development process. This can be an iterative process that grows in capability over time. The DevSecOps Lifecycle also overlaps with other initiatives currently underway at Dacoco, including the architecture strategy.




*Source: What is DevSecOps?*

DevSecOps stands for development, security, and operations. It's an approach to culture, automation, and platform design that integrates security as a shared responsibility throughout the entire software development lifecycle (SDLC).

DevSecOps means thinking about application and infrastructure security from the start. It also means automating some security gates to keep the DevOps workflow from slowing down. Selecting the right tools to continuously integrate security can help meet these goals. However, effective DevOps security requires more than new tools—it builds on the cultural changes of DevOps to integrate the work of security teams sooner rather than later.

It's important to note that DevSecOps can be implemented in any environment, including Agile.

The following are specific recommendations that can be implemented in order to start implementing a DevSecOps Lifecycle that is more efficient, cost-effective, and secure.

## Documentation

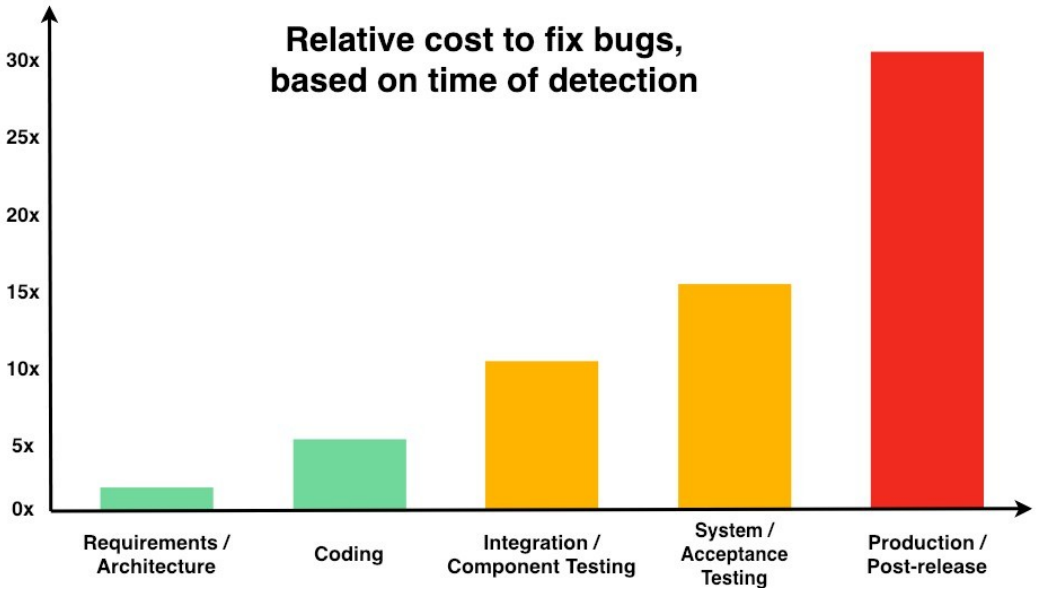
It is extremely important that any procedures that Dacoco wants to implement and standardize be documented. This is a key principle of information security. This documentation should be

comprehensive, organized, and available to anyone who needs it. If these changes are not documented, they cannot be consistently enforced between multiple individuals and projects. In addition, all documentation should be reviewed and validated at a minimum annually.

## Automated Unit Tests

A unit test is a way of testing the smallest piece of code that can be logically isolated in a system. As Dacoco continues refactoring the code base, adding automated unit tests not only improves code quality but helps ensure that security vulnerabilities are not introduced through defects. Unit testing and test automation significantly speed up the feedback loop on defect detection and enable programmers to change code more easily, with more confidence, and with fewer side effects.

NIST (National Institute of Standards and Technology) developed this graphic to illustrate the exponential cost of fixing a bug later in the process. Even with agile development, it is cheaper to fix an issue during development than during integration testing or production. Although there is an initial expense upfront, automated unit tests will save time and money as they identify defects much earlier in the process.




*Source: [The exponential cost of fixing bugs - DeepSource]:*

## Backups

Data backups are a copy of your data to be used in the event that the original copy is made unable, lost, or destroyed. Data backups protect against human errors, hardware failures, cyber-attacks, power failures, and natural disasters. Having reliable backups is one of the most

important controls that a company can use to prevent long-term damage following a data breach. Also, backups are important even if your company is never hacked. Accidents can happen, physical devices may get damaged, stolen, or just reach their end of life and shut down unexpectedly.

The system of record for the Dacoco platform is the blockchain, so in the event of a full system failure, the data can be retrieved. However, important data like system logs and configuration files that are not stored on the Blockchain may be lost, so at a minimum, these should be backed up, tested, and saved on a regular basis. In addition, Dacoco needs a backup and recovery process and system(s) for their other important corporate assets: HR data, asset tracking, partner contracts, ERP data, etc.

## Asset Management

Asset management is the process of ensuring an organization's assets are accounted for, deployed, maintained, upgraded, and disposed of when the time comes. This ensures that the valuable items, tangible and intangible, at Dacoco are tracked and used.

Managing assets like cloud resources, OS images, and application dependencies will make it much easier to track and find vulnerabilities, apply patches, and prevent sprawl.

Asset Management is the foundation of other processes like Patch Management.

## Key Management

It is recommended that the keys be transferred to a security model that is well protected and where tracking and logging of all access occurs.  This could be accomplished by an on premises software key management store or through the cloud provider.  The cloud option does ensure that all the security surrounding the vault is up to compliance.

## Logging and Monitoring

Security Logging and Monitoring Failures are on the OWASP Top 10 because it is not only extremely common to overlook this issue, but it is also very important to implement correctly.

Logging and Monitoring help detect, escalate, and respond to active breaches. Without logging and monitoring, breaches cannot be detected. It can be very impactful for accountability, visibility, incident alerting, and forensics as well.

# Pipeline Automation

## Infrastructure Automation

Infrastructure automation is the use of technology that performs tasks with reduced human assistance in order to control the hardware, software, networking components, operating system (OS), and data storage components used to deliver information technology services and solutions.

Applying automation to common management tasks (like provisioning, configuring, deploying, and decommissioning) simplifies operations at scale, allowing you to regain control over and visibility into your infrastructure. This will result in on-time updates, patching, and resource delivery.

Creating Infrastructure as Code (IaC) allows you to easily view the configuration, make changes, and identify configuration or security flaws. This also enables you to run Static Code Analysis tools against this code in order to find vulnerabilities in your infrastructure that would have previously gone unnoticed (see Static Code Analysis).

# Patch Management

Patch management is the process of distributing and applying updates to software. These patches are often necessary to correct vulnerabilities and bugs in the software.

It is critical to develop an effective patch management procedure. This includes monitoring components used in the application for vulnerabilities, understanding the severity, and developing a plan to update, test, and implement the patch. This will ensure that applications and systems are secure and have high availability.

# Secure Code Checklist

The most important Design Principle of Software Security is to **Implement Security by Design and Default**.

Secure coding is to design and develop software by avoiding the weaknesses that lead to security-related vulnerabilities by adhering to the specified security standards, and industry best practices.

To achieve security, it is essential to have a secure coding standard identified to help the team take care of the secure defaults for the software and to help protect it from attacks.

It is essential to define a set of secure coding rules and recommendations to which the source code can be evaluated for compliance so that the testers can carry out conformance compliance testing for each of these secure coding standards.

Developers need to follow a checklist established for secure code practices to ensure that security vulnerabilities are not overlooked.

## Separation of Duties/Single Points of Failure

Separation of duties is both an IT best practice and an audit and control standard that reduces the risk of a malicious or inadvertent breach of system security, data integrity, or the disruption of normal business processes, by requiring that individuals or workgroups not be in a position to control all parts of a transaction, system or business process. Separation of duties is fundamentally about reducing the risk of loss of confidentiality, integrity, and availability of information.

Duties should be separated between infrastructure, system administration, and software development. Developers should not have access to production systems, this should be managed by support personnel.

In addition, there are areas with only a single person with the knowledge to maintain the functions. In the event of a security incident or malfunction, if that individual is unavailable, the problem could take longer to resolve.

## Refactoring of Codebase

Refactoring is a change made to the internal structure of software to make it easier to understand and cheaper to modify without changing its observable behavior. Refactoring can not only help improve the overall code quality but it will also improve velocity over time when code is broken down into smaller more manageable pieces. Finally, it can also help isolate vulnerabilities with a smaller footprint. The key to code refactoring to carve out time as a part of the normal release process that is dedicated to this effort. It is recommended to set aside an agreed upon percentage to remediate technical debt.

## Risk Assessment

A risk assessment governs the confidentiality, integrity, and availability of all company assets by identifying, assessing, and rating risks. A Company manages risks appropriately through risk treatment plans. Risks are treated in accordance with the company's risk acceptance statement. Risk management of the company's assets must ensure appropriate controls are used to manage risks. Risks should be assessed in relation to the following factors: business,

systems, processes, and personnel. By conducting a risk assessment, Dacoco will have a better understanding of where they are vulnerable and can prioritize work based on the most critical items.

## Smart Contract Security Audits

Even if a smart contract is bug-free and securely developed, hackers are continually looking to exploit systems in new ways that were never thought of before. This is known as a zero day vulnerability. Conducting an audit after a major change can greatly reduce the risk of introducing a vulnerability into the ecosystem. If no major changes are made, it is an industry best practice to complete an audit annually to assess if any previously unknown vulnerabilities have surfaced and need to be remediated.

## Static Code Analysis

Static code analysis can be performed as part of a Code Review. Static code analysis refers to the running of static code analysis tools that attempt to highlight possible vulnerabilities within source code. Such tools automatically find security flaws with a high degree of confidence that what is found is indeed a flaw.

Static code analyzers can find both code quality and code security issues without much effort for the developer or code reviewer.

### Code Quality

Code quality is important, as it impacts the overall software quality. And quality impacts how safe, secure, reliable, and maintainable your codebase is.

High quality is critical for the team at Dacoco. And it's especially important when developing safety-critical systems.

### Code Security

It was previously mentioned how important code security is. Using a Static Application Security Testing (SAST) tool will help automate the process of creating safe, secure code.

# Appendix A: Resources

This section lists some potential tools and resources that may be used to increase Dacoco's security posture. There are many other tools and resources with the same functionality and this section is not meant to imply that they are the best for Dacoco's individual needs. It is included to show the types of information, features, and functions available in the marketplace. Where possible we have included an open-source option or tools that have a free tier that will serve the current needs.

## Asset Management

- Use cloud-native tools for managing cloud assets
- Snyk (source code dependency management)

## Logging and Monitoring

- OWASP C9: Implement Security Logging and Monitoring
- OWASP Logging Cheat Sheet
- OWASP Error Handling Cheat Sheet

## Patch Management

- Guide to Enterprise Patch Management Planning: Preventive Maintenance for Technology
- 6 Steps for Effective Patch Management
- 10 Step Patch Management Process Template
- Patch Management Best Practices: Process & Flow
- OpenCVE - an open-source vulnerability tracker for tools and software

## Pipeline Automation

- A common way to automate the deployment of infrastructure is with Terraform.

## Risk Assessment

- Cybersecurity Framework
- Risk Management Framework

## Secure Code Checklist

- OWASP Secure Coding Practices Quick Reference Guide
- OWASP Application Security Verification Standard

## Smart Contract Analysis

These automated tools were used during the security reviews of the smart contracts in addition to the manual reviews. These tools are helpful, but should not be relied upon solely to validate the security of a contract.

- Lamington - currently used by Dacoco. It should be noted that this is not actively supported
- Slither -- used for the Solidity contract analysis
- EOSIO contracts -- a few tools were reviewed but were not significantly helpful

Lamington

## Static Code Analysis

- SonarQube (code quality)
- Snyk (code security)