June 2022

INTERNAL

Secure Development Policy

FUNDS AXIS

Policy title:	Secure Development Policy

Issue	1.0
Approved by:	Trevor Dempster
Approval Date:	June 2022
Next Review Date:	June 2023

Scope:	The policy applies to Funds-Axis Limited and all contractors and other people working on behalf of the company.
Associated documentation:	\ Change Mangement Process \ Secure Development Environment Guidelines \ Information Security Policy for Supplier Relationships \ Supplier Information Security Evaluation Process
Responsibility for Implementation & Training:	Day to day responsibility for implementation: ISO

Distribution methods:	Methods used to communicate this policy: • Training

1. Introduction

The purpose of this document is to set out Funds-Axis's policy in the development of software applications and components in a way which maximises their inherent security.

Secure development contributes to the reliability of the IT environment by ensuring that as many vulnerabilities as possible are designed and tested out of software before it is deployed into the live environment.

Many security breaches around the world occur due to the exploitation of such vulnerabilities in system and application software, including the use of data that was not envisaged when the software was designed and tested.

The growth of cloud applications that are developed using methods such as Agile and DevOps and incorporate very fast development to deployment times means that emerging techniques such as DevSecOps are evolving rapidly to try to keep pace.

This document sets out the precautions that must be taken during the software development Lifecyle to minimise the risk to Funds-Axis whilst ensuring that the benefits set out in the original business case for the software are still realised.

As such, this document will represent an initial design for the enhancement of existing development processes and will be updated on at least an annual basis thereafter as Funds-Axis and its need develop.

This policy should be read in conjunction with the following documents which give more details in specific areas:

- Change Management Process
- Secure Development Environment Guidelines
- Information Security Policy for Supplier Relationships
- Supplier Information Security Evaluation Process

2. Software Development Approaches

The process of software development fits in with the higher-level process of project management of new or enhanced information systems.

This process has the following major stages in a project:

- Proposal
- Planning
- Design and Development
- Transition
- Project Closure

This software development lifecycle sits mainly within the Design and Development stage and consists of the following sub-stages:

- Design and Development
- Business requirements specification
- System design
- Development
- Testing

The way in which the stages of the software development lifecycle are approached will depend upon the development approach used. The two main models of software development used within Funds-Axis are Waterfall and Iterative. The choice of approach will be made on a project by project basis.

2.1 Waterfall Development Approach

The classic Waterfall approach to software development involves the planning and completion of each stage before moving on to the next, in a sequential manner. Functional requirements are defined in detail and signed off before the design stage may begin. In turn, design must be completed before development starts etc. This has the advantage that it is possible to ensure that adequate planning takes place and to include security checkpoints at the end of each stage. These will ensure the inclusion of adequate security criteria at the requirements stage and correct security controls at the design stage.

The common disadvantage with the Waterfall approach is that it is less flexible as circumstances and requirements change. If the project lasts for an extended period of time, there is the danger that what is delivered is no longer what is required.

Similar approaches based on Waterfall include Structured Programming Development, the Spiral Method and Cleanroom.

2.2 Iterative Development Approach

As an alternative to Waterfall, an Iterative approach may be taken. Such approaches include Rapid Application Development (RAD), Prototyping and, more recently, Agile and DevOps.

The Iterative approach typically involves significant stakeholder involvement throughout the development lifecycle and concentrates on producing frequent new versions of the software that may be evaluated and tested before further functionality is added. The process loops round with each of the stages being carried out many times in small iterations (in the Agile method these are called “Sprints”). Approaches such as Ci/CD (Continuous Integration/Continuous Delivery) and Continuous Deployment have evolved from the use of readily available cloud environments, and further reduce the time between code completion and its deployment.

An Iterative approach may be appropriate where exact requirements are less certain and frequent communication between developers and users (and within the development team) is possible.

The inclusion of security requirements and controls within an Iterative development approach needs to be carefully managed to ensure the functionality is not preferred to the exclusion of effective security measures. The speed involved and the potential lack of structured design documentation mean that effective training of developers in security matters and possibly the regular involvement of a security specialist are recommended. The use of roles such as Security Champion or Security Advocate within the development team may be appropriate to achieve the required focus on the security of the code deployed.

The set of techniques used to provide security in a highly iterative development environment is often referred to as “DevSecOps”.

3. Security in the Software Development Lifecyle

This section describes the way in which information security considerations must be incorporated into the various stages within the software development lifecycle.

3.1 Business Requirements Specification

The focus within the business requirements stage is on the functionality of the new system. This will be express in business rather than technical terms and should tie in with the business case that was produced prior to the initiation of the project.

The business is uniquely placed to give clear understanding to the development team of the security requirements of the information that the new system will hold and process. In particular the business requirements must specify:

- The value of the information involved
- The sensitivity of the information – will personal data be held?
- Who the information owner is or will be
- The classification of the information according to the scheme used within Funds-Axis
- The environment in which the information will be accessed or processed – will access be available in public areas?
- The criticality of the new system and the information it holds - what is the business impact if it is not available?
- The legal, regulatory and contractual environment the system must operate within

A risk assessment must be carried out as part of the project to ensure that the implications of the above issues are fully understood by all parties.

3.2 System Design

Based on the risk assessment and the classification of the information that is to be held in and processed by the new system, the design must provide for appropriate security features to be available. These will be largely defined by Funds-Axis's established security architecture as documented in Principles for Engineering Secure Systems.

This extends not only to the creation and maintenance of user accounts and permissions but also the following areas:

- Data input validation controls
- Data flow
- Data output
- Interfaces with other systems
- Reporting
- Restart and recovery
- Time stamps
- Logging (e.g. of transactions and access)
- Journaling of before and after images
- Batch and transaction counters
- Monitoring facilities
- How non-repudiation requirements will be met
- Ongoing patching arrangements
- Use of cryptography
- Need for digital certificates and signatures

For systems designed as part of a Waterfall approach these aspects will be included as part of the design documentation. If an Iterative approach is used, the development team will need to ensure that these areas are considered during every iteration and that changes do not invalidate controls implemented during an earlier iteration.

3.3 Development

Before starting to write code, a secure development environment must be established for the project. More detail regarding the creation and management of a secure development environment may be found in the document Secure Development Environment Guidelines.

Depending on the coding environment, languages, databases, tools and other components selected, the appropriate guidelines for secure coding and configuration must be adopted. These must be evaluated to ensure they will provide adequate protection from the various types of potential attack identified in the risk assessment, such as:

- Buffer overflow
- Time of Check/Time of Use
- Memory Reuse
- Malformed input

- SQL injection

For a lengthy project it will be necessary to obtain regular updates regarding newly identified vulnerabilities and exploits associated with the technology components in use.

3.4 Testing

During the lifecycle of a software application, many different forms of testing will be carried out, including unit, system, integration, performance, user acceptance and operational acceptance testing. Security controls will to some extent be tested as part of these exercise. However, it is recommended that a separate exercise of security testing be carried out against the security requirements that were established during the business requirements and design stages.

Initial security testing must be carried out within the development project with the same degree of rigour and formality as other forms with a suitable range of test inputs being specified.

Once this has been completed to the development team's satisfaction a further phase of security testing must be carried out by an independent part separate to the development team to verify correct operation of controls.

Adequate controls must be put in place to protect test data. Where appropriate (and with prior approval on each occasion), a live to test copy may be made in order to provide representative test data. However, if this contains sensitive information such as personally identifiable data this must be removed or obscured before use.

In a CI/CD (Continuous Integration/ Continuous Delivery) or Continuous Deployment scenario, in which the testing and deployment of code to the cloud may be automated, testing frameworks such as the OWASP (Open Web Application Security Project) Application Security Verification Standard (ASVS) must be used as a basis for testing application technical security controls.

4. Security in Outsourced Development

Where software development is wholly or partially outsourced to a third party, due care must be taken to ensure that the policies of Funds-Axis are still followed where possible.

Funds-Axis will remain legally responsible for the use of the software created and the information contained within it even though it didn't write the software. Therefore, the same level of rigour must be applied to outsourced software development as that created in-house.

4.1 Selection of Outsourced Developer

Standard procurement procedures must be used in the selection and engagement of an appropriate outsourced developer. Use of these procedures will ensure the developer:

- Is capable of delivering the software to the required standard
- Can meet the delivery timescales required
- Represents best value for Funds-Axis
- Can meet the security requirements specified

Use of sub-contractors by the outsourced developer for any aspects of the development must be understood and an assessment of these sub-contractors included.

Please refer to Supplier Information Security Evaluation Process for further detail on the areas that should be covered.

4.2 Communication of Requirements

The contract with the outsourced developer must require compliance with this policy and include a clear statement of the requirements for secure design, coding and testing of the software. The developer must also be required to establish a secure development in accordance with Funds-Axis standards. These are documented in Secure Development Environment Guidelines.

Requirements definition must be carried out by Funds-Axis so that a clear definition of the software to be created (including its security features) is agreed with the business and used as a contractual starting point for development. Whilst the outsourced developer may in some circumstances assist in the definition of requirements, the exercise should be led, managed and ideally carried out by internal resources so that there is a clear separation between requirements and design/development.

A comprehensive picture of the anticipated threat model faced by the software should be provided to the outsourced developer so that a clear understanding is gained of the types of vulnerabilities that must be avoided if the software is to be secure.

4.3 Supervision and Monitoring

Measures must be put in place to ensure adequate supervision of the activities of the outsourced developer and regular monitoring of progress.

For a large project with significant time gaps between deliverables, an agreed method of verifying interim progress must be in place so that early warning is given of delays.

4.4 Reviews and Acceptance

Review points must be established as part of the project planning process to verify progress and give formal acceptance of the software deliverables created. These will involve appropriate testing activities and code reviews.

The outsourced software developer must be required to provide evidence of the security testing activities carried out during the development, including tests for concealed malware, backdoors and known vulnerabilities.

Where appropriate a security review of developed code may be engaged with a suitable third party with the relevant security expertise.

4.5 Audit of Development Methods

Funds-Axis must have the contractual right to undertake a second party audit of the outsourced development provider. This may be to review whether the development methods used comply to our polices and that all information provided to the supplier is protected by appropriate security controls.

For larger projects it is recommended that an audit be carried out prior to the placing of the order for software development to ensure that assurances given during the scaling process are valid.

4.6 Intellectual Property

Unless the software is license under a formal agreement, contractual arrangements with an outsourced software developer must state that the ownership of the code produced on our behalf rest with Funds-Axis.

It is important that any software that is developed under an outsourcing contract is understood to be our intellectual property. Appropriate legal advice must be taken particularly if the outsourcer is based outside of our home country.

4.7 Escrow

Arrangements must be made for Funds-Axis to be able to legally access the source code of any developments undertaken, in the event that the outsourcer ceases trading for any reason. This must be the case during the development and if appropriate after the code has been delivered.