1. Introduction

This document aims to set out GBS's policy for developing software applications and components that maximize 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 transitioned into the live environment. Many security breaches worldwide 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.

This document outlines the precautions that must be taken during the software development lifecycle to minimize the organization's risk while ensuring that the benefits set out in the software's original business case are still realized. As such, this document will represent an initial design for enhancing existing development processes and will be updated annually as GBS and its needs develop.

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

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

- Business requirements specification
- System design
- Development
- Testing

The development approach will determine how the stages of the software development lifecycle are approached. The two main models of software development within GBS 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 planning and completing each stage sequentially before moving on to the next stage. 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 of ensuring adequate planning and including security checkpoints at the end of each stage is possible. These will ensure the inclusion of proper security criteria at the requirements stage and correct security controls at the design stage.

The common disadvantage of the Waterfall approach is that it is less flexible as circumstances and requirements change. If the project lasts for an extended period, the danger is 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

An Iterative approach, such as Rapid Application Development (RAD), Prototyping, or Agile, may be taken as an alternative to Waterfall.

The Iterative approach typically involves significant stakeholder involvement throughout the development lifecycle. It concentrates on producing frequent new software versions that may be evaluated and tested before further functionality is added. The process loops around, with each stage being carried out many times in small iterations (in the Agile method, these are called “Sprints”).

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.

Including security requirements and controls within an Iterative development approach must be carefully managed to ensure that functionality is not preferred over effective security measures. The speed involved and the potential lack of structured design documentation mean that practical developer training in security matters and possibly the regular involvement of a security specialist are recommended.

3. Security in the Software Development Lifecycle

This section describes how information security considerations should be incorporated into the various stages of the software development lifecycle.

3.1 Business Requirements Specification

The business requirements stage focuses on the new system’s functionality. This will be expressed in business rather than technical terms and should tie in with the business case produced before the project’s initiation.

The business is uniquely placed to give a 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 should specify:

- The value of the information involved
- The sensitivity of the information – will personal or protected data be held?
- Who the information owner is, or will be
- The classification of the information according to the scheme used within the organization
- 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 should be carried out as part of the project to ensure that all parties fully understand the implications of the above issues.

3.2 System Design

Based on the risk assessment and the classification of the information to be held in and processed by the new system, the design must provide appropriate security features to be available. These will be primarily defined by GBS'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

These aspects should be included in the design documentation for systems using a Waterfall approach. If an Iterative approach is used, the development team must ensure that these areas are considered during every iteration and that changes do not invalidate earlier controls.

3.3 Development

Before writing code, a secure development environment should be established for the project. The Secure Development Environment Guidelines provide more details regarding creating and managing a secure development environment.

The appropriate guidelines for secure coding and configuration should be adopted depending on the coding environment, languages, databases, tools, and other components selected. These should be evaluated to ensure they will provide adequate protection from the various types of potential attacks 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 a software application's lifecycle, many different forms of testing will be carried out, including unit, system, integration, performance, user acceptance, and operational acceptance testing. To some extent, security controls will be tested as part of these exercises. 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 should be conducted within the development project with the same rigor and formality as other forms, with a specified range of test inputs. Once this has been completed to the development team's satisfaction, an independent party separate from the development team should conduct a further security testing phase to verify the correct operation of controls.

Adequate controls should 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 to provide representative test data. However, it should be removed or obscured before use if it contains sensitive information, such as personally identifiable or protected data.

4. Security in Outsourced Development

Where software development is wholly or partially outsourced to a third party, care must be taken to ensure that GBS policies are followed.

GBS will remain legally responsible for using the software created and its information, even though it didn't write the software. Therefore, the same level of rigor must be applied to outsourced software development as that created in-house.

4.1 Selection of Outsourced Developer

Standard procurement procedures should be used to select and engage an appropriate outsourced developer. The use of these procedures should ensure the developer:

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

The outsourced developer's use of subcontractors for any aspects of the development should be understood, and an assessment of these subcontractors should be included. Please refer to the Supplier Information Security Evaluation Process for further details on the areas that should be covered.

4.2 Communication of Requirements

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

GBS should define requirements so that a clear definition of the software to be created (including security features) is agreed upon with the business and used as a contractual starting point for development. While the outsourced developer may assist in defining requirements in some circumstances, 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 of the types of vulnerabilities must be avoided if the software is to be secure.

4.3 Supervision and Monitoring

Measures should be taken to ensure adequate supervision of the outsourced developer's activities and regular progress monitoring.

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

4.4 Review and Acceptance

Review points should 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 should 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 suitable third party with relevant security expertise may review the developed code's security.

4.5 Audit of Development Methods

GBS should have the contractual right to undertake a second-party audit of the outsourced development provider. This may be to review whether the development methods comply with our policies and whether appropriate security controls protect all information provided to the supplier.

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

4.6 Intellectual Property

Unless the software is licensed under a formal agreement, contractual arrangements with an outsourced software developer should state that GBS owns the code produced on our behalf. Any software developed under an outsourcing contract must be understood to be our intellectual property. Appropriate legal advice should be taken, mainly if the outsourcer is based outside our home country.

4.7 Escrow

Arrangements should be made for GBS to legally access the source code of any developments undertaken if the outsourcer ceases trading for any reason. This should be the case during development and, if appropriate, after the code has been delivered.

5. Review and Updates

This policy must be reviewed and updated annually or when significant changes occur to the development environment, security landscape, or organizational structure.

6. Reference

Documents
Change Management Process
Secure Development Environment Guidelines
Supplier Management Policy
Secure Development Environment Guidelines
Secure Development Practices
Secure Implemented Measures