



KUOK (SINGAPORE) LIMITED

Secure Development Policy

Classification	PROTECTED
Document Reference	ISMS20009
Document Version	2.1
Document Status	Final

Revision History

Version	Date	Change Number	Summary of Changes
1.0	28 Dec 2018	NA	First copy
2.0	04 Mar 2020	NA	Change of Approver
2.1	19 Nov 2020	NA	Change of Approver


Document Review

Date of Next Scheduled Review
1 st Dec 2021

Authors

Name	Title
Tan Chin Tang	Senior Manager, IT Security

Approval

Name	Position	Signature	Date
Thai Kum Foon	Acting CIO, Group IT		17/2/2021

Contents

1	INTRODUCTION	1
2	SOFTWARE DEVELOPMENT APPROACHES	2
2.1	AGILE DEVELOPMENT APPROACH.....	2
2.2	ITERATIVE DEVELOPMENT APPROACH.....	3
3	SECURITY IN THE SOFTWARE DEVELOPMENT LIFECYCLE.....	4
3.1	BUSINESS REQUIREMENTS SPECIFICATION	4
3.2	SYSTEM DESIGN.....	4
3.3	DEVELOPMENT	5
3.4	TESTING	6
4	SECURITY IN OUTSOURCED DEVELOPMENT.....	7
4.1	SELECTION OF OUTSOURCED DEVELOPER	7
4.2	COMMUNICATION OF REQUIREMENTS.....	7
4.3	SUPERVISION AND MONITORING.....	8
4.4	REVIEWS AND ACCEPTANCE	8
4.5	AUDIT OF DEVELOPMENT METHODS	8
4.6	INTELLECTUAL PROPERTY	8
4.7	ESCROW	8
	THIS PAGE IS INTENTIONALLY LEFT BLANK	9

1 Introduction

The purpose of this document is to set out Kuok Singapore Limited'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 transitioned 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.

This document sets out the precautions that must be taken during the software development lifecycle to minimise the risk to the organisation. While 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 Kuok Singapore Limited and its needs develop.

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

- ISMS18004 Change Management Policy
- ISMS18011 Release and Deployment Management Policy

2 Software Development Approaches

The process of software development fits in with the higher level process of project management of new (or significantly changed) IT services. This is described in the document ISMS20001 Design and Transition of New or Changed Services.

This process sets out the following major stages in a project:

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

The 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 Kuok Singapore Limited are Agile and Iterative. The choice of approach will be made on a project by project basis.

2.1 Agile Development Approach

Agile process models in software development break away from the classic Waterfall Model and its variations. Instead of the fixed sequence of “specification, design and implementation”, the project is carried out in very close and direct collaboration with the customer. The specification gradually occurs during implementation.

Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product. Agile Methods break the product into small incremental builds. These builds are provided in iterations. Each iteration typically lasts from about one to three weeks. Every iteration involves cross functional teams working simultaneously on various areas like –

- Planning
- Requirements Analysis
- Design
- Coding
- Unit Testing and
- Acceptance Testing.

At the end of the iteration, a working product is displayed to the customer and important stakeholders.

One important advantage of agile process models is the collaboration with the customer. The customer receives what he needs and not what he has specified. This is an important benefit for projects whose requirements are still unclear at the beginning or are subject to major changes caused by external influences.

Another advantage is the breakdown of bureaucratic structures, e.g. fewer documents need to be created and maintained. A typical criticism of classic process models is the distance from the client or user. While the specification still occurs in close cooperation, the client often has virtually no influence on the definition of the product during implementation. Agile software development on the other hand generally attempts to manage with less bureaucracy and fewer explicit rules.

2.2 Iterative Development Approach

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

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").

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 that 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 is recommended.

3 Security in the Software Development Lifecycle

This section describes the way in which information security considerations should 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 expressed 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 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 data be held?
- Who the information owner is or will be
- The classification of the information according to the scheme used within the organisation
- 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 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.

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 should 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 should be established for the project. Depending on the coding environment, languages, databases, tools and other components selected, the appropriate guidelines for secure coding and configuration should be adopted. These should 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 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 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 should be carried out by an independent party separate to the development team to verify 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 in order to provide representative test data. However, if this contains sensitive information such as personally identifiable data this should be removed or obscured before use.

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 Kuok Singapore Limited are still followed where possible.

Kuok Singapore Limited 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 should be used in the selection and engagement of an appropriate outsourced developer. 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 best value for the organisation
- Can meet the security requirements specified

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

Please refer to ISMS08001 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 should require compliance with this policy and include a clear statement of the requirements for secure design, coding and testing of the software. The developer should also be required to establish a secure development environment in accordance with Kuok Singapore Limited standards.

Requirements definition should be carried out by Kuok Singapore Limited 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. While 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 should 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 should be in place so that early warning is given of delays.

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

4.5 Audit of Development Methods

Kuok Singapore Limited 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 used comply to our policies and that all information provided to the vendor 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 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 the ownership of the code produced on our behalf rests with Kuok Singapore Limited.

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

4.7 Escrow

Arrangements should be made for Kuok Singapore Limited 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 should be the case during development and if appropriate after the code has been delivered.

This page is intentionally left blank