**Contents**

[1. Introduction 3](#_Toc102468662)

[2. Purpose 3](#_Toc102468663)

[3. Software development lifecycle 3](#_Toc102468664)

[4. Contribution to Service Value Chain 4](#_Toc102468665)

[5. Software development tools 5](#_Toc102468666)

[6. Responsibilities 5](#_Toc102468667)

[7. Requirements 6](#_Toc102468668)

[7.1 Hardware resources 6](#_Toc102468669)

[7.2 Software resources 6](#_Toc102468670)

[7.3 Packaging and Installation 6](#_Toc102468671)

[7.4 Regulatory requirements 7](#_Toc102468672)

[8. Software test tools 7](#_Toc102468673)

[8.1 Personnel 7](#_Toc102468674)

[8.2 Hardware test platform 8](#_Toc102468675)

[9. System architecture 8](#_Toc102468676)

[9.1 Network architecture capabilities 9](#_Toc102468677)

[10. Risk Traceability Matrix 9](#_Toc102468678)

**DOCUMENT REVISION CONTROL**

REVIEWERS

|  |  |  |
| --- | --- | --- |
| **Name** | **Position** | **Date** |
|  |  |  |
|  |  |  |
|  |  |  |

DISTRIBUTION

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Distributed to** | **Version** | **Distribution format** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

APPROVALS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Version** | **Approved by** | **Designation** | **Approval remarks** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# 1. Introduction

The goal of software development and management is to make sure that the software applications are fit for purpose and easy to use so that customers can get the most out of them. This encompasses the complete lifecycle, from conception to retirement.

# 2. Purpose

The purpose software development and management are to make sure that applications satisfy the needs of both internal and external stakeholders. Functionality, dependability, maintainability, compliance, and auditability are all factors to consider.

# 3. Software development lifecycle

**Initiation**

**Plan**

**Maintenance**

**Develop**

**Deploy**

The software development lifecycle (SDLC) is a process that defines how software is created, from inception to deployment. The SDLC is typically composed of several phases, which can be summarized as follows:

1. Initiation: In this phase, the business objectives for the project are identified and the feasibility of the project is assessed.
2. Planning: This phase involves designing the system, determining the requirements, and creating a plan for testing and deploying the software. This is usually done by the system architect who has a great understanding of the entire IT architecture.
3. Development: The development stage is when programmers write code and build the application based on the design and requirement documents and specifications created by Business analyst and system architect.
4. Testing- QA will go over the software, identifying any flaws or defects that need to be recorded, corrected, and retested. It's critical that the program meets the quality requirements outlined in the requirements document.
5. Deploying and Maintenance- The software is deployed once it has been certified and no bugs or mistakes have been found. The software may then be distributed as is or with suggested enhancements in the object part, depending on the results of the assessment.

After the software has been deployed, it must be maintained. When the client begins to use the designed systems, genuine challenges and requirements arise that must be addressed on a regular basis.

# 4. Contribution to Service Value Chain

Chart, diagram, funnel chart

Description automatically generated

* Plan focuses on providing information on software tools that are essential for strategic and tactical planning.
* Enhance focuses on the information that is needed to improve through the usage of software tools.
* Design & Transition is concerned with the information offered by products and services, as well as the prospects for holistic application design.
* Obtain/build is concerned with acquiring and developing the software resources needed for service and service management.
* Deliver & support focuses on the software application's monitoring and continuous maintenance for value co-creation.
* Value- A software that is fit for use and suits its purpose is deployed.

# 5. Software development tools

|  |  |  |  |
| --- | --- | --- | --- |
| **Hardware** | **Software design** | **Programming and testing** | **Requirements tools** |
| Describe the hardware used for development | Tools used for design | Tools and languages used for development and QA testing | Tools used by Business analyst to document requirements |
|  |  |  |  |
|  |  |  |  |

# 6. Responsibilities

|  |  |  |  |
| --- | --- | --- | --- |
| **Roles** | **Responsibilities** | **Department** | **Reports to** |
| Business analyst | Document business requirements |  | Project Manager |
| Configuration Manager | Setting up tolls for development | IT | Project Manager |
| System Architect | Designs the IT infrastructure | IT | Project Manager |
| Developer | Works on implementation of requirements documented by Business analyst | IT | Project Manager |
| QA | Test the application to ensure it works as per the requirements | IT | Project Manager |

# 7. Requirements

## 7.1 Hardware resources

|  |  |
| --- | --- |
| **Requirement ID** | SRS-XXX-080 SAMPLE |
| **Category** | Must have |
| **Title** | Hardware configuration |
| **Description** | On a PC with the following basic configuration, XXX should run with the expected response times:  • 2 GB of RAM |
| **Version** |  |

## 7.2 Software resources

|  |  |
| --- | --- |
| **Requirement ID** | SRS-XXX-090 SAMPLE |
| **Title** | Software configuration |
| **Description** | XXX runs in the following software environment:  •(describe OS version), |
| **Version** | V2.0 |

## 7.3 Packaging and Installation

|  |  |
| --- | --- |
| **Requirement ID** | SRS-XXX-090 SAMPLE |
| **Title** | Packaging |
| **Description** | XXX shall be delivered on yyy media |
| **Version** | V2.0 |

|  |  |
| --- | --- |
| **Requirement ID** | SRS-XXX-090 SAMPLE |
| **Title** | Installation |
| **Description** | XXX shall be delivered on yyy media |
| **Version** | V2.0 |

## 7.4 Regulatory requirements

Software design can be influenced by regulations. This is the case, for example, with the FDA's upcoming Unique Device Identification program. An about window is a useful tool for determining the program version and providing a UDI. In Europe the CE mark may be somewhere in the GUI:

|  |  |
| --- | --- |
| **Requirement ID** | SRS-XXX-090 SAMPLE |
| **Title** | CE mark |
| **Description** | XXX shall display the CE Mark in the “About…” window.  The CE Mark is displayed with the 4-digits number of the notified body |
| **Version** | V2.0 |

# 8. Software test tools

Identify the program that was used to conduct the test:

* Operating systems and service packs
* Operating system drivers (if specific for you)
* Tools for backup and recovery
* Web, blogs, content management systems, and database engines; memory, disc use, CPU, and network analyzers; and test coverage or test management tools.
* Any modest (or big) software you built to run the tests
* Simulator, data generator of software or hardware that you don't have

## 8.1 Personnel

|  |  |  |  |
| --- | --- | --- | --- |
| **QA tasks** | **Tools** | **Responsible** | **Reports to** |
| Test the user interface of the application | Selenium | UX designer | QA Manager |
|  |  |  |  |
|  |  |  |  |

## 8.2 Hardware test platform

Describe the location of the test platform and, if applicable, the hours of operation. Describe the hardware used in your offices to test your software. Accurately identify the following hardware items:

* If conventional computers and servers are used
* Processor
* Memory
* Hard Disk
* Hardware configuration

If you employ specialized hardware (hardware simulator for a machine you don't have, hardware lent by a customer or a third party, electronic card, medical device, etc.)

* Their purpose
* Name
* Manufacturer
* Configuration, version
* Firmware version.

# 9. System architecture

Describe the hardware/software architecture's reasoning in terms of capabilities:

* Public appearances (for example response time, user mobility, data storage, or any functional performance which has an impact on architecture)
* Patient/user safety
* Anti-misuse protection
* Repairs and maintenance (cold maintenance or hot maintenance),
* Flexibility and adaptability
* Scalability and accessibility
* Backing up and restoring
* Security of hardware and software: fault tolerance, redundancy, emergency stop, crash recovery.
* Internationalization, administration, monitoring, and audit

## 9.1 Network architecture capabilities

If the device is connected to a network, explain the hardware and network architecture here:

* Bandwidth
* Network faults
* Data loss
* Inconsistent data
* Inconsistent data timing
* Cyber security

# 10. Risk Traceability Matrix

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Risk** | **Software element** | **Software unit** | **Comments** |
| 1 | Data out of range | Data reader module | Reader controller |  |
|  |  |  |  |  |
|  |  |  |  |  |

# 11. Requirement traceability Matrix

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Req ID** | **Requirement Description** | **Source/Requestor** | **Department** | **Business need** | **Test Strategy** | **UAT Responsibility** | **Status** |
| 1 | Change the table component to a graph | Allan | Sales | Better representation of data | Use cases to be developed | Follow the test steps as defined in use case | Done |
| 2 | Create a new category to sort the result set | Tanya Harper | Pricing | Will help the pricing department to automate selection of categorized data | Automated testing needs to be done | Check the categories in base table | On hold |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |