**Software Requirement Specification for HeartMate Touch Mobile Application (IOS) Build and Deployment using Azure DevOps**

Table of Content

1. Introduction

          1.1 Purpose of this document

          1.2 Scope of this Document

1.3 Glossary

          1.4 Intended Audience and Reading Suggestions

1.5 References

 2. Overall description

2.1 Product Perspective

          2.2 Dependencies

2.3 Assumptions

3. System Features and Requirements

3.1 Functional Requirements

3.2 Non-Functional Requirements

**1. Introduction**

**1.1 Purpose of this document**

The purpose of this document for the implementation of Build and Release pipeline on Azure DevOps for HeartMate Touch Mobile Application (IOS).

**1.2 Scope of the Document**

This document describes specification and dependencies for the implementation of Build and Release pipeline for the HeartMate Touch Mobile Application on Azure DevOps.

**1.3 Glossary**

|  |  |
| --- | --- |
| **Term** | **Definition** |
| DevOps | DevOps is the combination of cultural philosophies, practices, and tools that increases an organization’s ability to deliver applications and services at high velocity: evolving and improving products at a faster pace than organizations using traditional software development and infrastructure management processes. This speed enables organizations to better serve their customers and compete more effectively in the market. |
| Software Requirements Specification | A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. |
| Self-Hosted Agent | An agent that you set up and manage on your own to run jobs is a self-hosted agent. You can use self-hosted agents in Azure DevOps Server. Self-hosted agents give you more control to install dependent software needed for your builds and deployments |

**1.4 Intended Audience and Reading Suggestions**

The document is intended for HeartMate Touch Mobile Application Dev/release and deployment teams.

**1.5 References for Azure Pipeline**

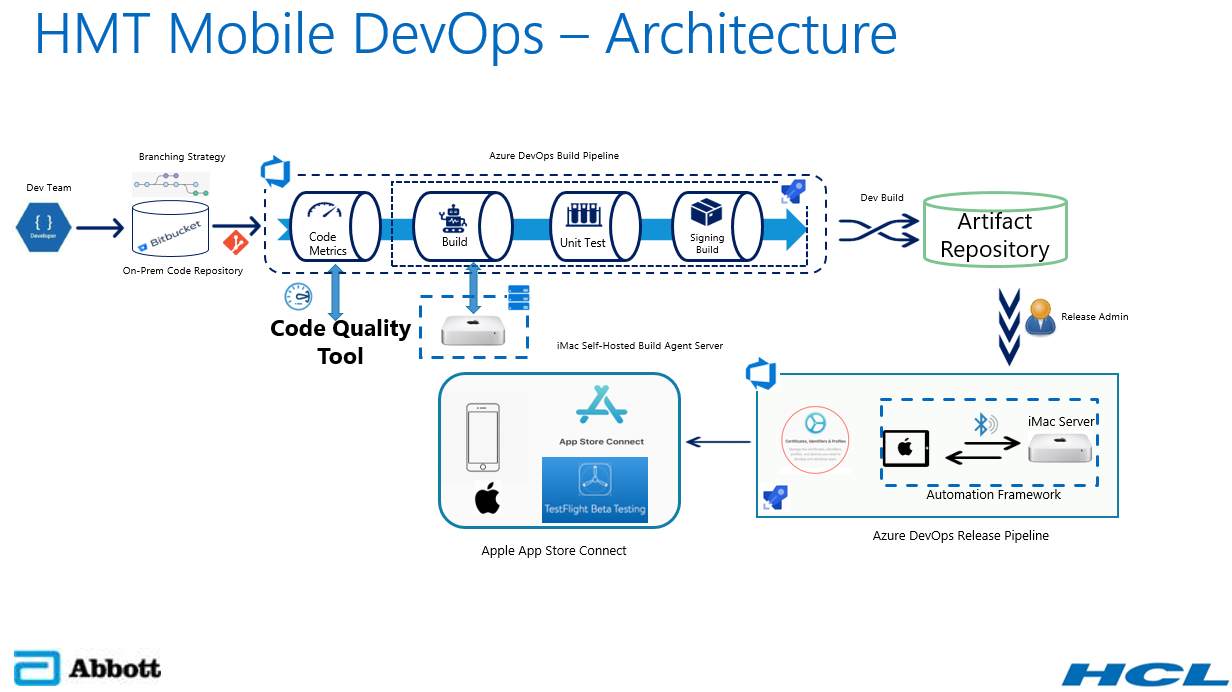
* <https://docs.microsoft.com/en-us/azure/devops/pipelines/ecosystems/xcode?view=azure-devops>
* <https://docs.microsoft.com/en-us/azure/devops/pipelines/get-started/what-is-azure-pipelines?view=azure-devops>

**2. Overall description**

**2.1 Product Perspective**

Need to add About HMT Mobile Apps Details

**Azure DevOps:** Azure DevOps Services provides a platform for implementing the DevOps process across multiple IT segments. This tool supports the various practices under DevOps such as continuous planning, continuous development, continuous integration, continuous testing, continuous deployment/delivery, and continuous monitoring/feedback.



**Source Code Management (SCM): Bitbucket (OnPrem)**

Bitbucket is a Source Code Management Tool and self-managed solution that provides source code collaboration for professional teams of any size, across any distance.

On-Premises Bitbucket is the code repository configured within the organization infra network to check-in the code developed by the developers.

**Build Agent: Xcode**

Xcode is an IDE - an integrated development environment – created by Apple for developing software for macOS, iOS, watchOS, and tvOS. (Xcode Version Need to check)

Xcode can be used to build the HeartMate Touch Mobile Application (IOS) and generate release package as an Artifact (IPA).

For more Information, about Xcode -https://developer.apple.com/library/archive/documentation/ToolsLanguages/Conceptual/Xcode\_Overview/BuildingYourApp.html#//apple\_ref/doc/uid/TP40010215-CH53-SW1

**Build Pipeline**

Whenever the developer triggers the DevOps pipeline manually, the Build process will start the build process and publish the release package into Artifact Server.

Firstly, it will clone the code from on premises Bitbucket then it will build the code with Xcode (MacOS Machine Self-Hosted Agent), and publish the artifact into Artifact Repository.

**Release Pipeline**

Once release package published into Artifactory (Jfrog/Nexus) from Build pipeline, release team will trigger Release pipeline to publish the package from artifactory server to target machine (App Store Connect).

**Artifact Repository**

An artifact repository is a software application designed to store these artifacts generated by Continuous Integration Pipeline.

It is storing, versioning and maintaining the Artifacts (IPA) generated by the Build pipeline.

Need Check with Developers(Jfrog/Nexus)

**Code Quality Tool**

Code Quality Assurance tool that collects and analyzes source code, and generate reports for the code quality of the project. It combines static and dynamic analysis tools and enables quality to be measured continually over time.

Code Quality Tool to detect bugs, vulnerabilities, and code smells in the code and it generate the reports.

Need Check with Developers (SonarQube/Coverity

**App Store Connect**

App Store Connect is a suite of web-based tools for managing apps sold on the App Store for iPhone, iPad, Mac, Apple Watch, Apple TV, and iMessage. As a member of the Apple Developer Program, you’ll use App Store Connect to submit and manage apps, invite users to test with TestFlight, add tax and banking information, access sales reports, and more.

App Store Connect can be used to submit the HeartMate Touch Mobile Application into TestFlight for Beta testing.

**TestFlight**

TestFlight makes it easy to invite users to test apps and App Clips and collect valuable feedback before releasing apps on the App Store. You can invite up to 10,000 testers using just their email address or by sharing a public link.

Take advantage of TestFlight by uploading a beta build of your app or App Clip to App Store Connect. Testers will use the TestFlight app to install your app and provide feedback. TestFlight supports apps for iOS, iPadOS, macOS, tvOS, watchOS, and iMessage, as well as automatic updates to ensure that testers always test the latest available build. Up to 100 apps can be tested at a time, internally or externally, and multiple builds can be tested simultaneously. Builds remain active for 90 days after upload.

TestFlight can be used to Deploy HeartMate Touch Mobile Application for Beta testing.

**2.2 Prerequisites**

The following Prerequisites for HeartMate Touch Mobile Application Build and Release implementation.

* Azure DevOps account for Build and Release pipeline.
* Bitbucket (On-Premises) -Source Code Repository.
* MacOS Dedicated Machine as a Self-Hosted Agent.
* Code Quality and Code Coverage stool.
* Artifact Repository.
* Install Xcode on MacOS Machine.
* Access to an Apple Account to get a Provisioning profile and p12 (Apple Certificate) to sign for iOS application.
* App Store Connect Account (Beta).

**2.3 Assumptions**

* Need access for Azure DevOps Account.
* Need access for Bitbucket repository (OnPrem).
* Need Dedicated MacOS Machine with Xcode.
* Need access Code Quality tools.
* Need access for Artifact Repository.
* Need Apple Account to get a Provisioning profile and p12 (Apple Certifacate) to sign iOS application.

**Servers Requirement**

In this Azure Pipeline required MacOS Machine (Self-Hosted Agent) for Build and Release the HeartMate Touch Application (IOS)

MacOS Machine Specification

MacOS Machine Version 10.15: "Catalina"

RAM-

Data Disk-

Need to check with Developer

**3. System Features**

**3.1 Functional Requirements**

**Dev Team**

* Developer should have access to Azure DevOps <https://dev.azure.com/ABT-HF-Software> “HeartMate Touch” Project.
* Developer should able to select HeartMate Touch pipeline” and click “Run Pipeline”
* Developer should have access to view the Artifactory to check the signed build was created after pipeline ran successfully.
* Developer should download the IPA file and install into IOS device to do UI testing.

**Release Team**

* Release team should have access to CD pipeline to run the pipeline and Deploy Artifact (IPA) into App Store Connect for Beta testing on TestFlight.
* The final release package will be uploaded to Windchill.

**3.2 Non-Functional Requirements**

This section is defined non-functional requirements.

Non-functional requirements include

* Pipeline running timeline
* Performance monitoring – Dashboard in Azure DevOps
* Email notification when build gets failed / success