`



Software Process Decision

Van Lang Admissions

# Revision Table

|  |  |  |  |
| --- | --- | --- | --- |
| Author | Date | Reason for changes | Version |
| Tai Nguyen | 22/9/2016 | Process Development | 1.0 |
| Tai Nguyen | 3/11/2016 | Update Process | 1.1 |

Table of Contents

[Revision Table 1](#_Toc465720024)

[1. INTRODUCTION 3](#_Toc465720025)

[1.1. Purpose 3](#_Toc465720026)

[1.2. Audience 3](#_Toc465720027)

[2. Process 4](#_Toc465720028)

[2.1. Process flow 4](#_Toc465720029)

[2.2. Advantage 4](#_Toc465720030)

[2.3. Disadvantage 5](#_Toc465720031)

[3. Comparison 5](#_Toc465720032)

[3.1. Compare incremental with Waterfall and scrum 5](#_Toc465720033)

[3.2. Process Decision 7](#_Toc465720034)

# INTRODUCTION

## Purpose

Give a Decision about process we will apply to our project

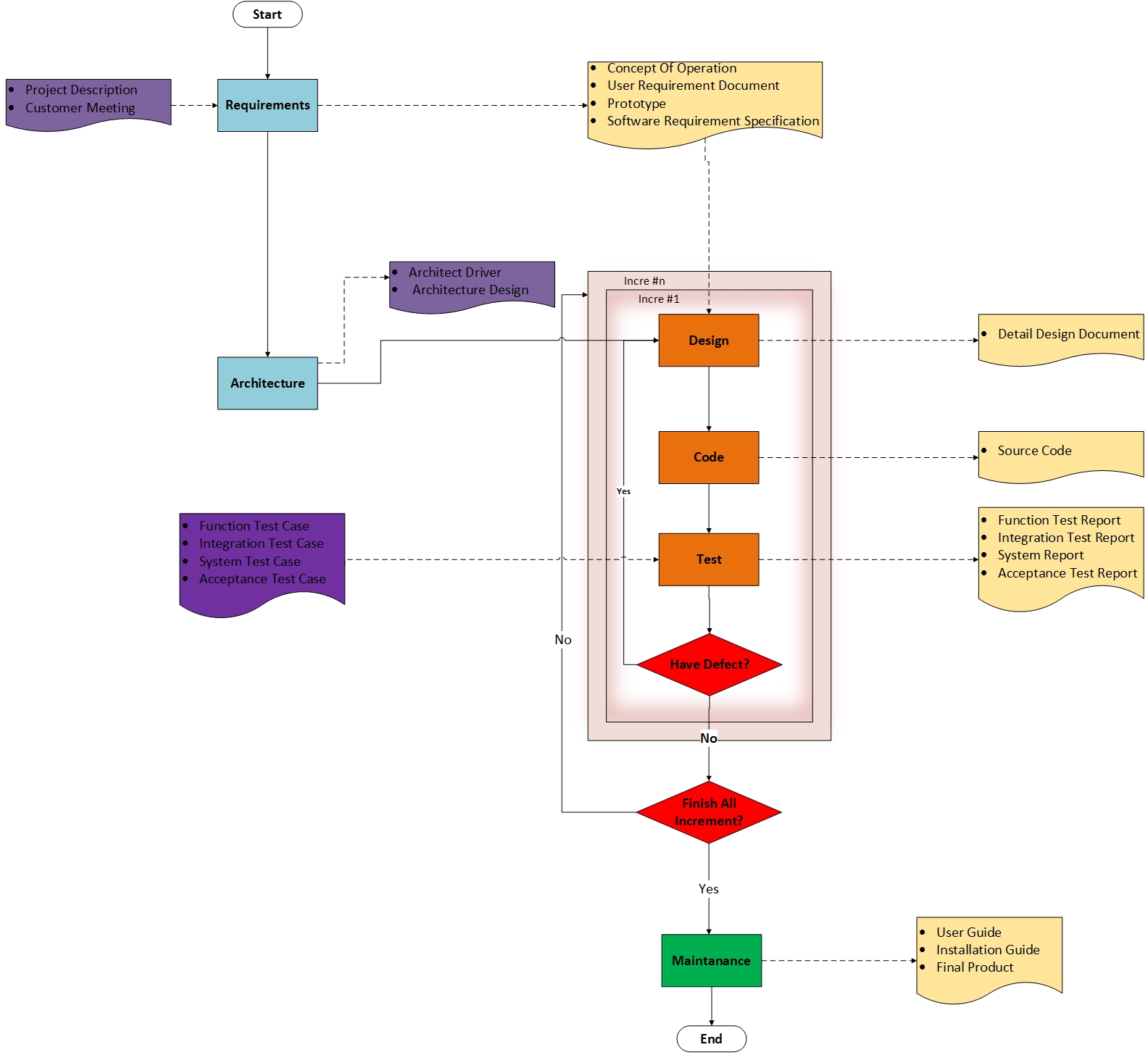
## Audience

The intended audience of the CM Plan is

|  |  |
| --- | --- |
| Intended Audience | Description |
| All team member | All member have a meeting and discuss about process |
| The mentor | People give an advice and support group during implement process. |
| Stakeholder on customer side | People give requirement of project |
|  |  |

# Process

## Process flow



## Advantage

* Used when requirements are well understood
* Iterative in nature; focuses on an operational product with each increment
* Provides a needed set of functionality sooner while delivering optional components later
* Useful also when staffing is too short for a full-scale development
* Generates working software quickly and early during the software life cycle.
* More flexible – less costly to change scope and requirements.
* Easier to test and debug during a smaller iteration.
* Easier to manage risk because risky pieces are identified and handled during its iteration.

## Disadvantage

* Each phase of an iteration is rigid and do not overlap each other.
* Problems may arise pertaining to system architecture because not all requirements are gathered up front for the entire software life cycle.

# Comparison

## Compare incremental with Waterfall and scrum

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | features | | Waterfall Model | Incremental | Scrum |  |
| 1 | Specification of all the Requirements in the beginning | | Yes | Not all and Frequently Changed | Not all and Frequently Changed |
| 2 | Project Cost | | Almost as Estimated | Above Estimated Cost | Almost as Estimated |
| 3 | Guarantee of Success | | Low | High | High |
| 4 | Required Expertise | | Moderate | Moderate | Very High |
| 5 | Overlapping Phases | | No | Yes as Parallel development is there | Yes |
| 6 | Process | | Heavyweight Process | Light weight Process | Light weight Process |
| 7 | Framework type | | Linear | Combination of Linear and Iterative | iterative and incremental |
| 8 | Rework cost | | High | Almost High | High |
| 9 | Testing | | After coding phase Completed | After Every Iteration | After Coding Phase |
| 10 | Customer Involvement | | Low | High, After Each Iteration | High |
| 11 | Basic business Knowledge Required | | Not much | Moderate | Very Much |
| 12 | Suitable Project Size | | Small Scale | Low to Medium Scale | Large Scale |
| 13 | Cost Control | | Yes | No | No |
| 14 | Simplicity | | Simple | Moderate | Simple |
| 15 | Risk Involvement | | High | Low | Not High |
| 16 | Flexibility | | Rigid | Much Flexible | Flexible |
| 17 | Maintenance | | Least Maintainable | Maintainable | Maintainable |
| 18 | Changes Incorporated | | Difficult | Easily | Easily |
| 19 | Reusability | | Least Possible | To some Extent | Yes |
| 20 | Documentation and Training | | Necessary | Yes But Not Much | Limited |
| 21 | Time Frame | | Very Long | Long | Moderate |
| 22 | Availability of Working Software | | At the End of the Life Cycle | At the End of Every Iteration | At the End of Every Iteration A |
| 23 | Customized product | | Least Possible | Much Possible | Possible |
| 24 | Customer Control over Administrator | | Very Low | Yes | Yes |
| 25 | Required Team Creativity | | No | Yes | Yes |
| 26 | Knowledge Transfer | | No | Yes | Yes |
| 27 | Team Size | | Large Team | Not Large Team | Large Team |
| 28 | Primary Objective | | High Assurance | Rapid Development | Rapid Development |
| 29 | Implementation | | Easy | Easy | Moderate |
| 30 | Release Cycle | | Big band(All Functionality at Once) | In Phases | In Phases |
|  | |
|

## Process Decision

* It can be divided into sections to make, such as one parallel web team 1 team app, the part can be divided equally or simultaneously carried out after the completion of 1 new function to function other.
* We will select the Incremental because the project is divided into smaller modules, easy to manage. Each module goes through the requirements phase, design, implementation and testing. A version of the software is launched for the first time, so you have the software earlier in the software lifecycle. Each subsequent versions will add more functionality to the previous version. This process continues until the complete system is achieved. On the other hand, we can test for customers every week or month. We can manage the risks in our projects. We will save our time if we fail a certain period. We just need to redo that period.
* The main requirement is determined. However, no details.
* Require applying new technologies.
* Human resources group on the skills of the new technology is not ready.
* There are some features and target high-risk.