DRAFT - ICT Project Guidance

Code Management

Version:

0.1

Author:

Sky Sigal, Solution Architect

## Description

<TODO>

## Synopsis

<TODO>

## Contents

[Description 1](#_Toc145232973)

[Synopsis 1](#_Toc145232974)

[Contents 2](#_Toc145232975)

[Synopsis 3](#_Toc145232976)

[Introduction 3](#_Toc145232977)

[Heading Level 3 3](#_Toc145232978)

[Heading Level 4 3](#_Toc145232979)

[Appendices 4](#_Toc145232980)

[Appendix A - Document Information 4](#_Toc145232981)

[Images 4](#_Toc145232982)

[Tables 4](#_Toc145232983)

[References 4](#_Toc145232984)

[Review Distribution 4](#_Toc145232985)

[Audience 4](#_Toc145232986)

[Structure 4](#_Toc145232987)

[Diagrams 4](#_Toc145232988)

[Terms 5](#_Toc145232989)

## Introduction

BOSSCARD/ RAID: Background [], Objective, Options, Scope[In/Out], Stakeholders [Users], Constraints, Assumptions, Risks, Dependencies, Decisions, Deliverables.

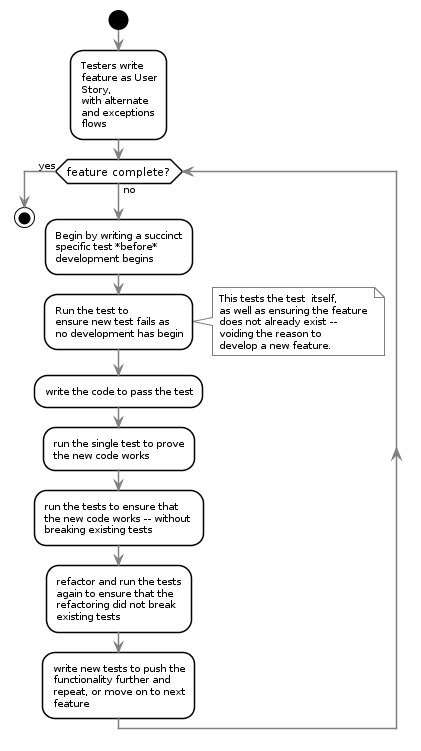
### Confidential Information

Configuration credentials and other confidential information, irrespective of the environment must not be checked in.

### Test Driven

Developers MUST follow an automated Test-Driven Development process.

The following is an example workflow:



### Code Documentation

Code must be documented to facilitate reviews as well as decrease the risk of the market being unable to deliver capable maintenance specialists to take over and continue the work.

**Note**  
A long-running debate has existed in the development world, implying that code comments within code are not required if the code is clean and self-describing.

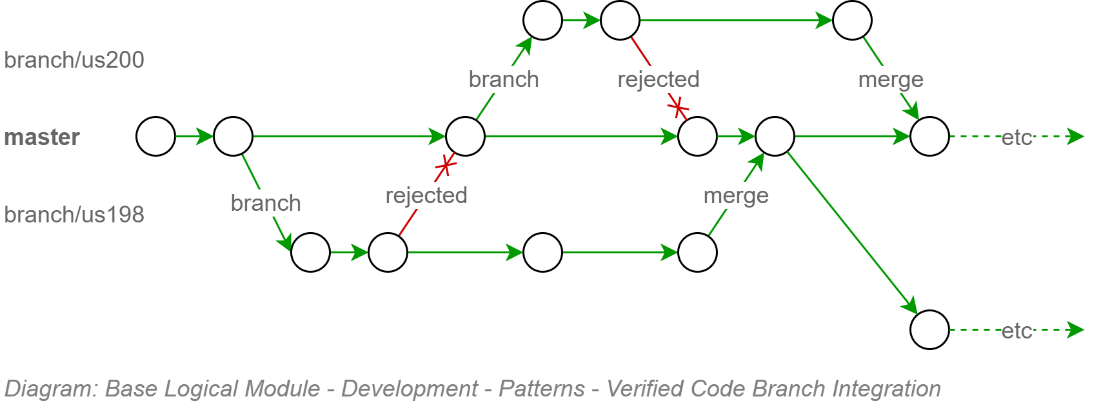
The argument is idealistic and simply wrong.

Code is written to communicate with computers. Comments are written to communicate with humans. You would not say that a book written in French, if written well enough, should not require assistance to allow an English person who is not 100% fluent to read it). Secondly, there are many types and levels of stakeholders in a project, not all access the code. Produce documentation of what is going on, elsewhere.

### Verified Code Branch Integration

The Version Control Service employed to deliver the solution performs \*Continuous Integration\* activities and verifies submitted code feature branches before integrating the code with the protected `master` branch.

If the Version Control Service rejects the code due to it failing tests -- or a `Pull Request` reviewer (see elsewhere in this View) has manually rejected the submission -- the developer has to fix it and try again before the Version Control Service will allow the submitted feature branch to be integrated with the protected `master` branch:

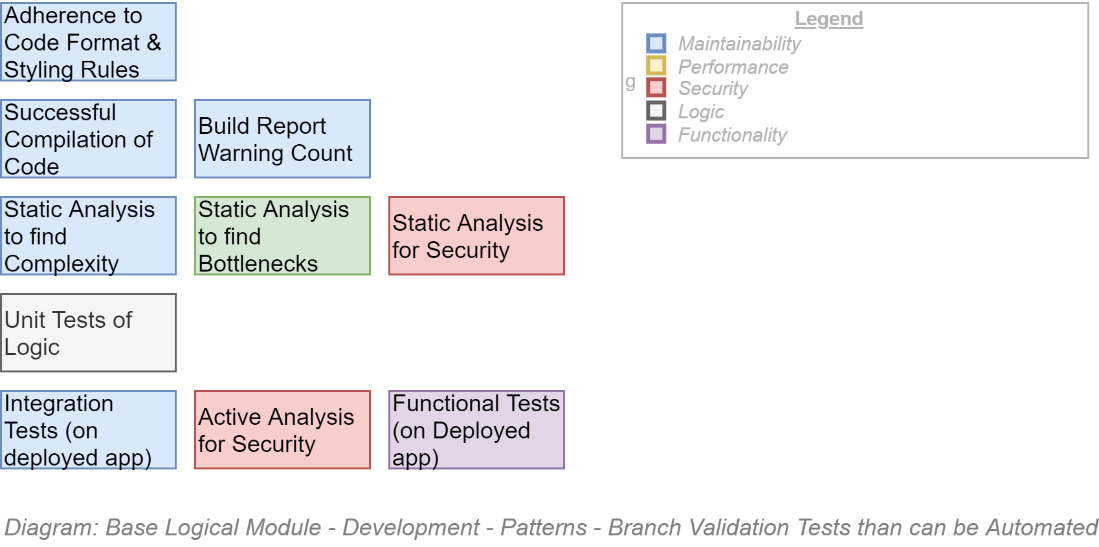


This upfront effort configuring the continuous integration pipeline to protect the code base from getting polluting, protects other developers from downloading poor code and working around it – only to have to remove the work arounds when the original poor code is fixed.

### Automated Tests

Static Tests can be performed on the source code itself, or the components created by compiling the source code. Dynamic tests can only be applied once the components have been deployed to a target environment.

The following categories are recommended for testing by automation.



The project is not expected to automate every test possible before beginning to develop the information system – but the project manager and dev team lead are expected to put aside resource time to implement in an ongoing continuous manner till they are completed.

**Note:**  
The importance of these automated tests are two-fold:   
- ensuring the quality of code during the main upfront development phase,  
- putting in place before handover to different (support) developers the automated safeguards required to ensure the initial quality of the product does not degrade over service’s whole service lifecycle duration.

Appendices

Appendix A - Document Information

### Versions

* 1. Initial Draft

### Images

[Figure 1: TODO Image 2](#_Toc144995112)

### Tables

[Table 1: TODO Table 3](#_Toc145048484)

[Table 2: TODO Table 2 3](#_Toc145048485)

### References

**There are no sources in the current document.**

### Review Distribution

The document was distributed for review as below:

|  |  |
| --- | --- |
| Identity | Notes |
|  |  |
|  |  |
|  |  |

### Audience

The document is technical in nature, but parts are expected to be read and/or validated by a non-technical audience.

### Structure

Where possible, the document structure is guided by either ISO-\* standards or best practice.

### Diagrams

Diagrams are developed for a wide audience. Unless specifically for a technical audience, where the use of industry standard diagram types (ArchiMate, UML, C4), is appropriate, diagrams are developed as simple “box & line” monochrome diagrams.

### Terms

Refer to the project’s Glossary.

##### IT

: acronym for Information, using Technology to automate and facilitate its management.

##### ICT

: acronym for Information & Communication Technology, the domain of defining Information elements and using technology to automate their communication between entities. IT is a subset of ICT.