Bare Necessities

Version 2.0

Document Details

Document Path	**http://192.168.1.102:9073/Mindshare/5gpal/policies/OPD_Bare Necessities**
Document Version	2.0
Document Date	17-Jul-2018
Document Status	Final
Circulation Type	Internal
Circulation List	All@5G

Document Revision History

Date	Version	Content Added/Revised	Prepared by/Revised by	Approved By
Apr 2008	1.0	Base Document	Shogan	Ananth RK
Nov 15, 2008	1.1	Adopted new naming conventions, Added sections on engineering documents etc,	5G PG Team	Sumitra Seshan
Dec 12, 2011	1.2	Added general necessities section	5G PG Team	
08 May 2016	1.3	Rephrased the bare necessities section, removed .net specific sections.	Badri	Ananth RK
17 Jul 2018	2.0	Updated the bare necessities as per the current practices	5G PG Team	Sumitra Seshan

TABLE OF CONTENTS

1.0 INTRODUCTION **4**

1.1 PURPOSE 4
1.2 AUDIENCE 4

2.0 BARE NECESSITIES **4**

1.0 Introduction

1.1 Purpose

This document lists the must-follow engineering processes for every project involving code development.

1.2 Audience

All members of 5G engineering group.

2.0 Bare necessities

All development initiatives in 5G must meet the following bare necessities:

1. **Each project must have a standard coding style guideline and must follow it. Projects can also adopt coding style tools like StyleCop/FxCOP, Pep as available for .Net and Python.**

2. **Have automated test cases. This should at least include functional unit tests and integration tests.**

3. **The project's code repository must have separate branches to track on-going development as well as various releases. The release branches must have access control to prevent direct write and must only be modified via Pull Requests that are reviewed and merged by the leads of the project.**

4. **Code review requests to be sent to reviewers against the development branches. Upon closure of the code review issues, the project leader(s) will then merge the changes to the appropriate release branch. This ensures the release branches will not have any code without reviews.**

5. **Have automated script(s) to build and deploy the code from the release branch. During the build, the automated test cases shall be executed and**

only upon all test cases are passed, the deployment of code shall be started. Build failures during this build is to be considered as critical failure of the project. Team can adopt tools like Jenkins for deploying and scheduling these automated script(s) for continuous integration and deployment.

-End of Document-