

---

# TEAM 2 POLICY

for

CMPT371

Version 1.0 approved

Prepared by Team 2

University of Saskatchewan

November 4, 2019

# Contents

<b>1</b>	<b>Policy</b>	<b>3</b>
1.1	Members . . . . .	3
1.2	Roles and Responsibilities . . . . .	3
1.2.1	Project Manager . . . . .	3
1.2.2	Design Lead . . . . .	3
1.2.3	Test Lead . . . . .	3
1.2.4	Build Master . . . . .	4
1.2.5	Risk Officer . . . . .	4
1.3	Meetings . . . . .	4
1.4	Git . . . . .	4
1.5	Builds . . . . .	5
1.6	Prototypes . . . . .	6
1.7	Tardiness . . . . .	6
1.8	Pull Requests . . . . .	6
1.9	Read and Understood . . . . .	6
1.10	Positive Work Environment . . . . .	6

# 1 Policy

## 1.1 Members

Team Roles	
Member	Role(s)
Evan	Project Manager
Mesa	Development Lead
Eileen	Test Lead
Braunson	Build Master, Development Team
Kevin	Test Team
Clinton	Test Team
Anurag	Development Team
Amanda	Test Team, Risk Officer
Camille	Development Team

## 1.2 Roles and Responsibilities

### 1.2.1 Project Manager

The project manager will be responsible for organizing meetings, meeting with stakeholders, establishing priority of implementations, coordinating between accountable positions, and other administrative work. Generally, the project lead will not directly code. In cases of disagreements the project lead has executive decision making power.

### 1.2.2 Design Lead

The design lead is in charge of development. Their goal is to complete the priorities set out by the stakeholder, organize the development team to ensure cohesion, and selecting technologies for use in the project. They should also ensure code quality and work closely with the test lead and build master.

### 1.2.3 Test Lead

The test lead has to ensure there is sufficient test coverage that adequately tests use cases of the stakeholder. They will have oversight over the whole test team and is in charge of test plans.

### **1.2.4 Build Master**

The build master ensures the build pipeline is maintained with a server to deploy as staging and production environments. The pipeline managed by the build master should include automating tests, notifying group members of build issues, and ensuring quality code is checked into the project.

### **1.2.5 Risk Officer**

The risk officer is a rotating, part-time role which evaluates risk continuously during project development. They are responsible for documenting mitigation strategies and proactively taking steps to reduce risks to the project.

## **1.3 Meetings**

Meetings and class room attendance is considered mandatory. If an individual cannot attend a meeting they are to notify the project lead ahead of time and must read the meeting minutes afterwards/when they are able to/at their earliest opportunity. If more than three meetings are missed in the course of the term that person must talk to the project manager, if any more meetings are missed, the team will talk to Dr. Osgood.

The first ten minutes of class will be used as an informal stand up. If a team 2 member cannot attend they are to post on slack what they have done since last time and what they plan to do for next time.

## **1.4 Git**

The project follows the Git Flow paradigms. The only modification is we do not strictly use release branches, releases are tagged in master when a milestone is reached.

For more information on how Git Flow works, see [here](#).

All features must be reviewed via a pull request by at least two people and the build must pass before it can be merged to the staging environment. Every pull request should be associated with an issue that has already been triaged.

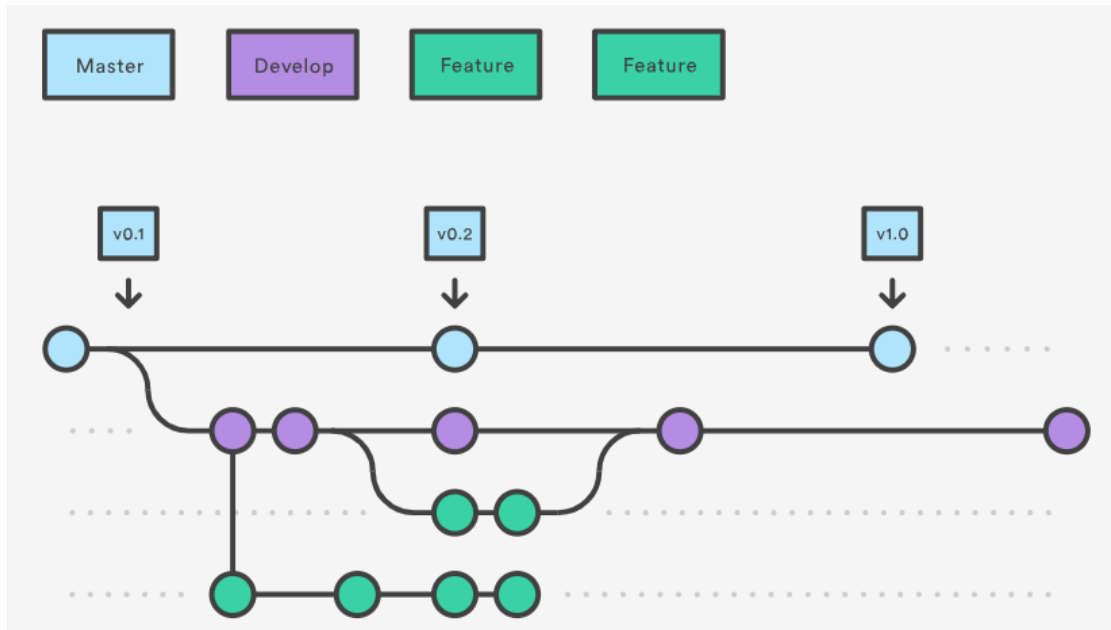


Figure 1.1: A diagram of Git Flow.

## 1.5 Builds

No member should knowingly push broken code to the repository. All members should locally test the quality of their code before pushing to the repository.

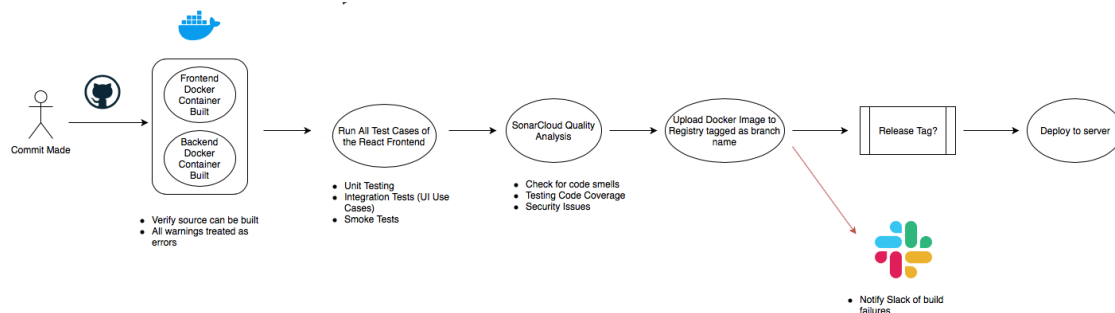


Figure 1.2: A diagram of the build pipeline.

Some of the steps are subject to change as the project evolves. Users who break the build are responsible for resolving the issue before it is merged into the staging environment.

## **1.6 Prototypes**

Prototypes should be placed in the prototypes directory with a README file that details the objective of the prototype and the outcome. The outcome will serve as a "lessons learned" for use in the project.

## **1.7 Tardiness**

Excessive and repeated lateness will be met with the formal issuing of a yellow-card. Being late will be considered being more than 10 minutes late without any sort of valid excuse.

## **1.8 Pull Requests**

It is expected that people check pull requests at least once a day. If an item is not reviewed by a team member within 24 hours, any member can add themselves as a reviewer and 'yoink' the review from you. Although no formal action is taken, it reflects extremely badly on you if you're always letting your reviews get yoinked by more attentive teammates

## **1.9 Read and Understood**

A R&U (read and understood) response is mandatory in slack for messages, if you do not respond in slack with a :ladies\_and\_gentleman:, the PM (project manager) will PM (private message) you to ensure you've read the message and understand the implications

## **1.10 Positive Work Environment**

In order to promote a healthy and safe work environment, all ideas and contributions will be met with open and honest discussion.