

McMASTER UNIVERSITY

SMARTSERVE

SOFTWARE & MECHATRONICS CAPSTONE

Development Process & Implementation

Authors:

Christopher McDonald
Harit Patel
Janak Patel
Jared Rayner
Nisarg Patel
Sam Hamel
Sharon Platkin

Professor:

Dr. Alan Wassyng

Teaching Assistants:

Bennett Mackenzie
Nicholas Annable
Stephen Wynn-Williams
Viktor Smirnov



Last compiled on October 29, 2017

Contents

1	Introduction	2
1.1	Project Overview	2
1.2	Naming Conventions & Terminology	2
2	Team Members	2
2.1	Roles & Responsibilities	3
2.2	Meeting Schedule	3
2.3	Communication Pipelines	4
2.4	Handling Change	4
3	Version Control	4
4	Process Workflow	5
5	Onboarding Process	5

List of Figures

1	Revision History	1
2	Feature-Branch and Dev Git Workflow	5

Date	Revision	Comments	Author(s)
October 28, 2017	1.0	Added structure and content	Christopher McDonald

Figure 1: Revision History

1 Introduction

1.1 Project Overview

SmartServe is an autonomous table tennis training system for a wide range of table tennis players to aid in diagnosing and improving their performance. The system does so by shooting balls toward the player and detecting successful returns. It can then adapt to the player's weaknesses and expose them by playing more to them over time. This alleviates problems of finding and working with a coach or training many players in a short amount of time. The system will be deemed a success if table tennis players enjoy and can see value in using our system in place of or in tandem to a coach.

The development started at the beginning of the Fall 2017 academic turn and will conclude at the end of the Winter 2018 term. The team was assembled in the capstone course for Software and Mechatronics Engineering disciplines.

1.2 Naming Conventions & Terminology

- **Git:** a distributed versioning control system
- **Master Branch:** the main branch of the GitHub repository
- **Pull Request:** a request to add changes from one branch into another done via a merge commit or rebase
- **Slack:** a messaging platform for teams to subscribe and publish to channels or send messages between two or more team members
- **DevOps:** development operations

2 Team Members

The team members are as follows:

- Christopher McDonald
- Harit Patel
- Janak Patel
- Jared Rayner
- Nisarg Patel
- Sam Hamel

- Sharon Platkin

In addition to the list above, Dr. Alan Wassyng will be the Project Advisor. He will be assisted by Bennett Mackenzie, Nicholas Annable, Stephen Wynn-Williams and Viktor Smirnov for marking and providing advice to the team.

2.1 Roles & Responsibilities

See Table 1 for a breakdown of roles & responsibilities.

Table 1: Roles & Responsibilities Breakdown

Role	Member(s)	Responsibility
Scribe	Sharon Platkin	Taking notes for all meetings Posting notes to Slack after meetings
DevOps Developer	Christopher McDonald	Solving issues related to GitHub Configuring CI Tools Merging code changes into the <i>master</i> branch
Team Contact	Christopher McDonald	Handle a majority of communication with Project Advisor and Teaching Assistants
Hardware Team Member	Janak Patel Jared Rayner Nisarg Patel	Handle most hardware focused work tasks
Software Team Member	Christopher McDonald Harit Patel Sam Hamel Sharon Platkin	Handle most software focused work tasks

2.2 Meeting Schedule

The team currently has weekly meetings scheduled on Tuesdays at 16:30 and Fridays at 16:30 for a duration of 50 minutes. Every member of the team is expected to attend. Every other Tuesday meeting must cover reflecting on the previous 2 weeks and planning the following 2 weeks for what work needs to be done in that time. All other meetings will be for progress updates, clarifying any work going forward and removing roadblocks which inhibit optimal performance.

The Hardware and Software team will schedule meetings on an ad-hoc basis, inviting any members which have a stake in the content being discussed. In the event Sharon Platkin is not present and therefore cannot be the Scribe, a member will be elected by and among the attendees.

2.3 Communication Pipelines

All inter-team communication will be done through Slack. This will allow team members to communicate via channels or groups. Examples of the channels currently used are *hardware*, *software*, *documentation*, *meetings*, *deliverables* and *git*. The *git* channel will give updates on open PRs and the *deliverables* channel gives reminders 2 days ahead of all deliverable deadlines. Meeting notes will be posted in the *meetings* channel.

When communicating with the TAs or Project Advisor, email will be used and the Team Contact will relay all relevant information to all other team members.

2.4 Handling Change

In the event any aspect of the project must be changed, all members can be involved in the decision making. In the absence of a unanimous decision, the team must decide if the change is paramount to the outcome of the project. If not, the decision will be left to a vote by all meeting attendees. If so, a teaching assistant will be involved in the decision making to aid in arriving to a unanimous decision. The Project Advisor will also be involved at the discretion of the teaching assistant.

Once the change has been decided, all documentation and code changes will be given the highest priority to reflect the change made. This will ensure no further work will be done using documentation or code that does not reflect the team's ideals.

3 Version Control

The technology used for version control will be *git*. The *git* repository will be hosted by GitHub and can be found [here](#). The workflow we will be using is a feature-branch workflow with the addition of a two-stage release process. For all releases, changes will be made via a Pull Request (PR) into the *master* branch from the *dev* branch. Therefore, any demonstrations and deliverables can be found by checking out the *master* branch. When making any changes, an ad-hoc branch will be made off the *dev* branch. Once completed, a PR can be made from that branch into the *dev* branch. This pertains to documentation and source code. A flowchart for this can be found in Figure 2.

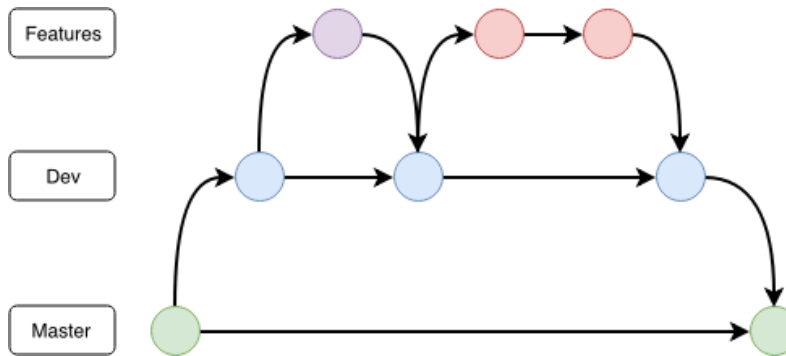


Figure 2: Feature-Branch and Dev Git Workflow

4 Process Workflow

In order to facilitate the development of the system, work will be organized into 2-week long sessions. At the beginning of the week, the team will consider the next 2-3 deliverables and schedule them in accordingly. The team will attempt to finish each deliverable a week or more before the deadline. This allows the team to be ahead of schedule and gives them the ability to respond to other items should they arise. During meetings before the end of each 2-week session, any priorities will be adjusted as needed. When the work session is completed, items will be placed into the next session or placed in the backlog until the next session.

5 Onboarding Process

The following administrative work must be done to onboard another team member:

- Invite them to Slack team
- Share team's Google Calendar with them
- Add them as a collaborator on GitHub
- Add them to the list of authors on the documentation template

The new member will be expected to read all current revisions of documentation made for the project. Any questions which arise can be asked during meetings or on Slack. After a comprehensive understanding of the project, tasks can be assigned to them and any education will be done on an ad-hoc basis.