

# Introduction to Software Dev Notes

Marie Burer

16-1-24

# Contents

<b>1</b>	<b>Overview</b>	<b>3</b>
<b>2</b>	<b>Git and Version Control</b>	<b>4</b>
<b>3</b>	<b>Software Development Process Models</b>	<b>4</b>
<b>4</b>	<b>Code Construction</b>	<b>6</b>

# 1 Overview

Tools:

- Warp
- Eclipse
- Git

What is Warp?

- Existing code base
  - Not huge, but bigger than other class projects
  - Useful because it reflects real world, you won't join a team and write code from scratch
- Written by Professor Goddard for a research project called WARP
  - Initially written in Swift, then rewritten in Java for the course
  - Primary purpose is to develop programs for network communication in wireless sensor networks

Why Warp?

- Get used to reading and modifying others' codes
- Learn to refactor code

- Rich opportunity to add:
  - documentation
  - pre- and post-conditions
  - testing
  - new features
- Learn from Professor Goddard's experiences and mistakes.

## **2 Git and Version Control**

## **3 Software Development Process Models**

Beginner's Model: Code and Fix

- Conceptual Development
- Code and Fix
- Release Product

Software Development Life Cycle

- Conception
- Requirements gathering
- Design

- Coding and debugging
- Testing
- Release
- Maintenance
- Retirement

#### Software Development Model Categories

- Plan driven models:
  - Strict methodology
  - Clearly defined phases
  - Heavy weight
  - Works best for large contracts
- Agile development models:
  - incremental and cyclic
  - small, frequent releases
  - less documentation
  - Works well for start ups

#### Waterfall Process Model

- Conceptual Development

- Requirements Analysis
- Architectural Design
- Detailed Design
- Code and Debug
- System Testing
- Release and Maintenance

## 4 Code Construction

Who is the code for?

- The computer
  - Must fulfill requirements and implement the design
- You and other programmers
  - Must be readable and easy to understand