

Visual Facts: A Cross-Platform Program Fact Visualization Tool

Nigel Lee

School of Computer Science and Engineering

Nanyang Technological University

July 13, 2021

Submitted in Partial Fulfillment of the Requirements for the Degree of
Engineering Science (Computer Science) of the Nanyang Technological
University

This page is intentionally left blank.

Abstract

Hello

Contents

Abstract	i
1 Introduction	1
2 Preparations	2
2.1 Requirements Gathering	2
2.2 Project Management	2
2.2.1 Agile Principles	3
2.2.2 Scrum Units of Time	3
2.2.3 Scrum Units of Work	3
2.2.4 Alternate Project Management Techniques	4
3 Architecture	5

Chapter 1

Introduction

Chapter 2

Preparations

This chapter details the preparations conducted prior to developing the software project.

2.1 Requirements Gathering

An initial series of requirements gathering was conducted by researching on the topic and discussing with my project supervisor. The initial requirements gathering served as the groundwork to plan stories and epics ¹ for the project. The following requirements were gathered:

- Blah

2.2 Project Management

The project was conducted in the style of Agile Software Development² with Scrum techniques³. The Agile method is based on the Agile Manifesto for Software Development⁴ written in 2001. The manifesto highlights several guiding principles towards light weight software development processes.

While the terms *Agile* and *Scrum* are often use interchangeably, *Agile* refers to the philosophy laid out by the original Manifesto while *Scrum* refers to an actual practice of the Agile philosophy.

¹see 2.2.3

²blah

³blah

⁴blah

2.2.1 Agile Principles

The Agile principles which are most relevant to this project are itemized below:

- Customer satisfaction by early and continuous delivery of valuable software
- Welcome changing requirements, even in later development
- Deliver working software frequently
- Sustainable development, able to maintain a constant pace

For the purpose of this project, *customers* refer to my supervisor.

2.2.2 Scrum Units of Time

Sprints

Sprints are the basic unit of time in the Scrum method. Sprints usually span 2 - 4 weeks and incorporate a specific scope of work to be done. The following events are usually conducted during the sprint:

- **Sprint Planning:** Conducted at the start of the sprint, the scope of work and tasks to be done are determined here.
- **Backlog Refinement:** Usually conducted in the middle of the sprint. Tasks that are on the backlog are prioritized. If the pace of work for the current sprint is ahead of schedule, additional tasks in the backlog can be introduced into the current sprint.
- **Sprint Retrospective:** Conducted at the end of the sprint. An after action review is done of the sprint that just occurred, detailing tasks that went well and possible improvements.

For the purpose of this project, I adopted a 2 week sprint and involved my supervisor in the sprint planning and retrospective.

2.2.3 Scrum Units of Work

Work in the Agile method is can be described with the following terms:

Story

The story is an actionable description of how the user intends to interact with a software system. It is usually written in the following structure:

As a user, I expect the system to X when I do Y

The story is often created during initial requirements gathering and is used to plan epics.

Epic

An epic is a description of the work to be done to provide functionality for a story. Epics are usually large bodies of work that span multiple sprints. Epics can be broken down into individual technical tasks.

Task

A task is the smallest unit of work in the Agile method. Tasks usually have a single scope in a single code repository.

2.2.4 Alternate Project Management Techniques

The main counterpart to the Agile method is the Waterfall method. In contrast to the Agile method, the Waterfall method involves a sequential workflow from analysis and design to development and testing.

While the Waterfall method excels at delivering a product according to the initial spec, it is rigid to changing requirements.

Chapter 3

Architecture