Software Engineering for Industry Coursework 2: AcmeTelecom Billing System

Yufei Wang (yw6312, s5)
Paul Gribelyuk (pg1312, a5)
Yawei Li (yl8012, s5)
Jun He (jh1212, s5)
Xiaoxing Yang (xy212, s5)

December 2, 2012

1 Introduction

In this report we relied heavily on Test Driven Development (TDD) to analyze the initial codebase and then build on it to add a business-critical feature in a consistent way. We added Framework for Integrated Testing (FIT) to understand the results produced by the billing system as well as how to proceed in developing software which would pass these tests.

2 Analysis of Original Code

Initial dependency graph showed that the code was coherent and decoupled, although it lacked interfaces to a lot of classes for easy testing and feature-building. `BillingGenerator` acts as an intermediary between `BillingSystem` and `Printer`.

3 Refactoring and Testing

Creating interfaces for testing purposes Creating Mock tests Creating Unit tests for container classes Breaking `HtmlPrinter` singleton interface and changing constructor to take a different `PrintStream` other than `System.out`.

Extracted calculation part of original `BillingSystem` into a new class to allow flexibility in future design.

4 Creating a DSL

5 Implementing the New Billing System

6 Acceptance Tests

- GivenPeakPeriod:
- GivenTheFollowingCustomers:
- GivenTheseCallsAreMade:
- GivenTheSystemIsInitialized:
- TheBillShows:

7 Conclusions and Recommendations

Tools used (Git, Jenkins, IntelliJ Coverage Report, IntelliJ Dependency Matrix). Problems enountered like refactoring for Tiny Types, Mocking with concreted classes rather than interfaces.