Hristina Koleva

New bulgarian university

F66436

PERFORMANCE ANALYSIS REPORT

MITM231 PERSONAL SOFTWARE PROCESS - ADVANCED

Table of Contents

[Planning Performance 2](#_Toc380492721)

[Size Estimating Accuracy 2](#_Toc380492722)

[Size Estimating Error 3](#_Toc380492723)

[Time Estimating Accuracy 4](#_Toc380492724)

[Time Estimation Error 4](#_Toc380492725)

[Productivity 5](#_Toc380492726)

[Quality Performance 6](#_Toc380492727)

[Quality Analisys 6](#_Toc380492728)

[Defect Density and Predictability 6](#_Toc380492729)

[Defect Analysis 7](#_Toc380492730)

[Defect Types – Design 7](#_Toc380492731)

[Defect Types – Code 7](#_Toc380492732)

[Defects/Size – Reviews (Removed) 8](#_Toc380492733)

[Process Performance 9](#_Toc380492734)

[Review Rates 9](#_Toc380492735)

[Defect Removal Rates 9](#_Toc380492736)

[Leverage vs Unit Tests 10](#_Toc380492737)

[Yield vs Code Review Rate 10](#_Toc380492738)

[Yield vs Design Review Rate 11](#_Toc380492739)

[Yield vs A/FR 11](#_Toc380492740)

[Process Improvement 12](#_Toc380492741)

[Process Improvement Areas 12](#_Toc380492742)

[Process Improvement Proposals 12](#_Toc380492743)

[Expected Performance Improvements 12](#_Toc380492744)

[Figure 1 Size Estimation Accuracy 3](#_Toc380492762)

[Figure 2 Size Estimation error 3](#_Toc380492763)

[Figure 3 Time Estimation Accuracy 4](#_Toc380492764)

[Figure 4 Time Estimation Error 4](#_Toc380492765)

[Figure 5 Productivity 5](#_Toc380492766)

[Figure 6 Total Defects Density 6](#_Toc380492767)

[Figure 7 Quality Prediction Accuracy 6](#_Toc380492768)

[Figure 8 Defect Types injected in Design 7](#_Toc380492769)

[Figure 9 Defect Types injected in Code 7](#_Toc380492770)

[Figure 10 Defect Trends in Reviews 8](#_Toc380492771)

[Figure 11 Defect Removal Rates 9](#_Toc380492772)

[Figure 12 Defect Removal Leverage 10](#_Toc380492773)

[Figure 13 CR Rate and Yield 10](#_Toc380492774)

[Figure 14 Code Review Rate vs Yield 10](#_Toc380492775)

[Figure 15 DLDR Rate and Yield 11](#_Toc380492776)

[Figure 16 Design Review Rate vs Yield 11](#_Toc380492777)

[Figure 17 YIELD VS A/FR 11](#_Toc380492778)

# Planning Performance

Planning performance is a set of metrics over the accuracy demonstrated through Planning phase in Size and Time estimations.

## Size Estimating Accuracy

The overall trends are following very similar profile for all of the assignments. What is very obvious from the chart and data below is that I have overestimated the size of the programs from Assignment 2 to 4 and started underestimating it for the programs from Assignment 5 to 7.

The smallest differences between Planned and Actual are for the programs from Assignment 3 and Assignment 6 and the reason for this is the greater percent of Reused code, respectfully from Assignment 2 and Assignment 5.

The reason for overestimation in the first programs is my expectation that more new code (new parts added) would have been necessary, while it was so only for the programs from Assignment 2, since the size of the programs from Assignment 3 and 4 is half of Reused code.

Underestimation of the size of the programs from Assignments 5 to 7 is due to underestimation of the complexity of the whole assignment and expecting the opposite, as for Assignment 2 – 4, the more code would be reused. And it is so but I haven’t taken into account the code modifications due to Design and Code Reviews which also added more lines of code.

Lowe Prediction Interval – LPI (70%) for programs from Assignments 2 to 7 is 45 LOC, Upper Prediction Interval – UPI is 159. 5 out of 6 programs are within the UPI although none falls within the LPI.

4 out of 6 Relative Size estimates are overestimated by the data in the tool, but my conclusion is that this is due to optimizations and seeks for readability and maintainability which in some of the cases has made me modify the code at the Code Review phase and sometimes that has decreased the size.

Another factor is that PROBE method C was used for the programs from Assignment 3 to 5 , Assignment 6 is estimated using PROBE B and only Assignment 7 was estimated with PROBE A.

Figure 1 Size Estimation Accuracy

## Size Estimating Error

Figure 2 Size Estimation error

## Time Estimating Accuracy

Time Estimation error is relatively small for 4 of the programs and is smaller than the Size Estimation error. The programs from Assignment 3 to 6 are estimated using PROBE C and only the program from Assignment 7 is estimated with PROBE B. The confidence in the estimations has not been very high. Reasons are mainly due to lack of experience in estimating coding tasks. The huge error of 80% for Assignment 4 is due to the fact that the whole Program 3 was reused. All of the estimations fall within the statistical predication interval but due to the high variance the intervals are very far from each other, so this cannot be used for valid inferences.

There is obvious dependency between the Time and Size estimation accuracies. Again only Assignment 4 has different profile and can be somehow treated as an outlier.

Figure 3 Time Estimation Accuracy

## Time Estimation Error

Figure Time Estimation Error

## Productivity

The trend of the productivity is very much explainable by the essence of the Assignments. From A2 to A3 there is a slight decrease because of the introduction of the Design and Code Review checklist, although it is very small since half of the code was reused. The significant increase between A3 and A4 is because of the gained awareness of the model and tasks practices so far and again because of significant percent of reusable (tested) code. The proportional decrease from A4 to A5 is due to the fact that A5 is from the Advanced level, and thorough Design Specification and Verification phases. The programming task itself was also of higher level than the previous. Then, with A6 and A7 a stable level of productivity is reached.

Figure Productivity

# Quality Performance

## Quality Analisys

### Defect Density and Predictability

The profile of total count of defects in the programs from Assignment 2 to 4 and then from 5 to 7 is similar because and correspondingly decreasing. This is due to the reused code in A 3 and 4 and A 6 and 7. The slight increase between A 4 an A 5 is due to the difference in the tasks and increased level of complexity of the requirements for the two assignments. Obviously due to the Design and Coding standards, reviews and verifications the total number of defects has decreased more than 3 times if we also count A 1 (where the number of defects – not shown in the chart – was 9 with Density 40).

Figure Total Defects Density

Defects number has been relatively predictable throughout all of the assignments. The only significant differences are for Assignments 3 and 7 due to the reused code and overestimating of the programs complexity. Anyhow, the planned number of defects has been always higher than the actual which is the better case, than underestimating the potential risks.

Figure Quality Prediction Accuracy

## Defect Analysis

### Defect Types – Design

Figure Defect Types injected in Design

### Defect Types – Code

Figure Defect Types injected in Code

Since the Function defects have been most common, they have also been the most expensive (in time) to fix. More than 3 times than Interface and Data defects and more than 10 times in comparison to the forth most common defect type, which has been the Assignment type.

### Defects/Size – Reviews (Removed)

Ever since the coding standards and reviews were introduced in the process, I have continuously been increasing the defects found by reviews in comparison with the defect found by test, until no bugs have entered Unit Test phase in the programs from Assignment 6 and 7. Handling of the defects and early detection is of the strong areas where improvement can be treated as a secondary goal, after estimation processes are improved.

Figure Defect Trends in Reviews

# Process Performance

## Review Rates

Average review rates are not extremely high due to me being new to the introduced technics so it has taken me more time to review. Overall impression is that the numbers are still reasonable and justifiable. Average Design Review rate is approximately 80 LOC per Hour. Average Core Review rate is around 100 LOC per Hour.

## Defect Removal Rates

Defect removals rates decrease due to the overall descend of number of defects.

Figure Defect Removal Rates

## Leverage vs Unit Tests

From the chart below, we can easily conclude that Design and Code Review phases have been more effective than the Unit Test phase, which proves the improvement in early and cost-efficient defect detection.

Figure Defect Removal Leverage

## Yield vs Code Review Rate

It will not be very reliable and valid inference if we try to find a relationship between Code Review rate and Yield, which could be probably explained with lack of significant experience in these types of tasks.

|  |  |
| --- | --- |
| **CR Rate** | **Yield** |
| 43.76 | 75 |
| 112.99 | 50 |
| 32.57 | 100 |
| 168.81 | 100 |
| 136.75 | 100 |

Figure 13 CR Rate and Yield

Figure Code Review Rate vs Yield

## Yield vs Design Review Rate

We cannot make a valid inference about a relationship between Design Review rate and Yield, which could be probably explained with lack of significant experience in these types of tasks.

|  |  |
| --- | --- |
| **DLDR Rate** | **Yield** |
| 93 | 66.67 |
| 144.38 | 100 |
| 64.39 | 100 |
| 34.87 | 100 |
| 61.49 | 0 |

Figure 15 DLDR Rate and Yield

Figure Design Review Rate vs Yield

## Yield vs A/FR

Appraisal to failure ratio shows the relation between time spend in design review and code review against the time spend in test (compile time is 0). It can be seen that A/FR from the programs from Assignments 5 to 7 is up to 2 times higher than A/FR for the programs in Assignment 3 and 4 which increased the Yield to 100% or with more around 20% average.

|  |  |  |
| --- | --- | --- |
| **Assignment** | **A/FR** | **Yield** |
| **A3** | **3.57** | **83** |
| **A4** | **1.41** | **67** |
| **A5** | **4.36** | **100** |
| **A6** | **5.62** | **100** |
| **A7** | **7.95** | **100** |

Figure YIELD VS A/FR

# Process Improvement

## Process Improvement Areas

Finally, we can conclude that the area that needs serious improvements is the Planning phase; respectfully greatest errors have been made during Size and Time estimations.

Review (and Test) phases have been the most predictable and this is explainable by my experience as a Quality Assurance professional rather than software developer.

## Process Improvement Proposals

Size Estimation Proposal

* Increase focus on the Planning phase.
* Analyze the requirements more thoroughly
* Identify unclear requirements and use more efforts on them
* Identify and research for ready-made solutions for unknown technologies or problems, prototypes and walkthrough tutorials

Time Estimation Proposal

* More practice in Coding tasks
* Careful analysis at the design phase, spend more time there

## Expected Performance Improvements

Size Estimation Improvement Goals

* Fall within both statistical prediction intervals by decreasing the Size Estimation Error with 20%
* Decrease Time Estimation Error with 10%