

# PGT Exam Performance Feedback

## 2018/2019 Semester 1

---

COMP61511 Software Engineering Concepts in Practice

Bijan Parsia  
Christos Kotselidis

Comments Please see the attached report.

---

# COMP61511 Exam Feedback

The exam consisted of 31 MCQs (1 mark each) and 2 essays (5 marks each) for a total of 41 marks.

The average time to take the exam was 1 hour 46 min, which is shorter than in previous years which probably reflects the fact that we moved to 2 essays from 3.

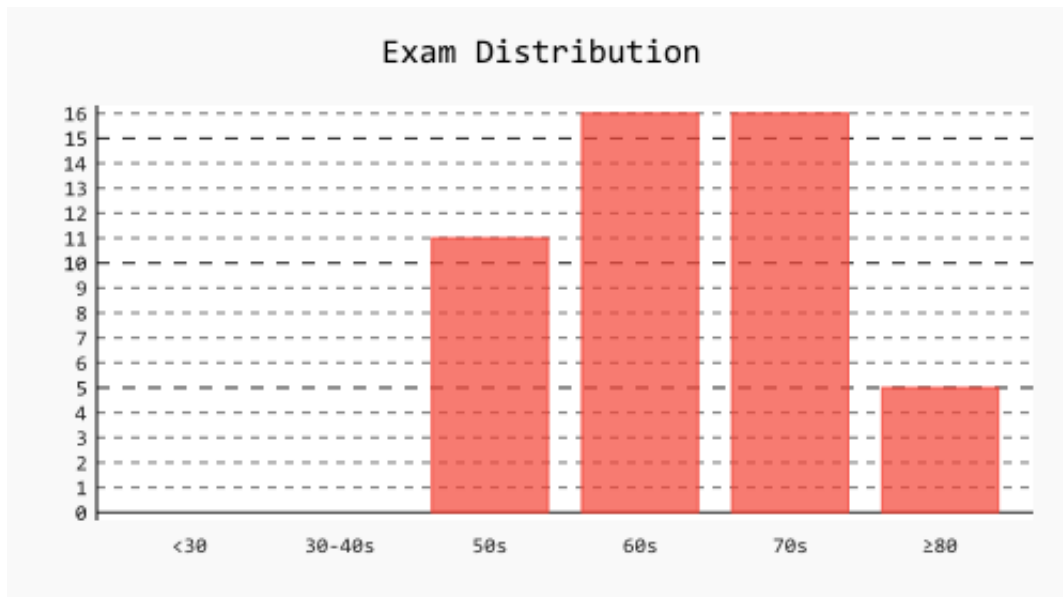
## Overall Exam Performance

---

Here are the basic stats for the exam:

	All	MCQs	Essays
Mean	67.4	67.27	67.81
Median	67.07	67.74	65.0
Stdev	8.96	10.31	10.96
Min	50.0	45.16	45.0
Max	85.37	87.1	100.0

This graph gives shows the distribution of marks in a more granular form.



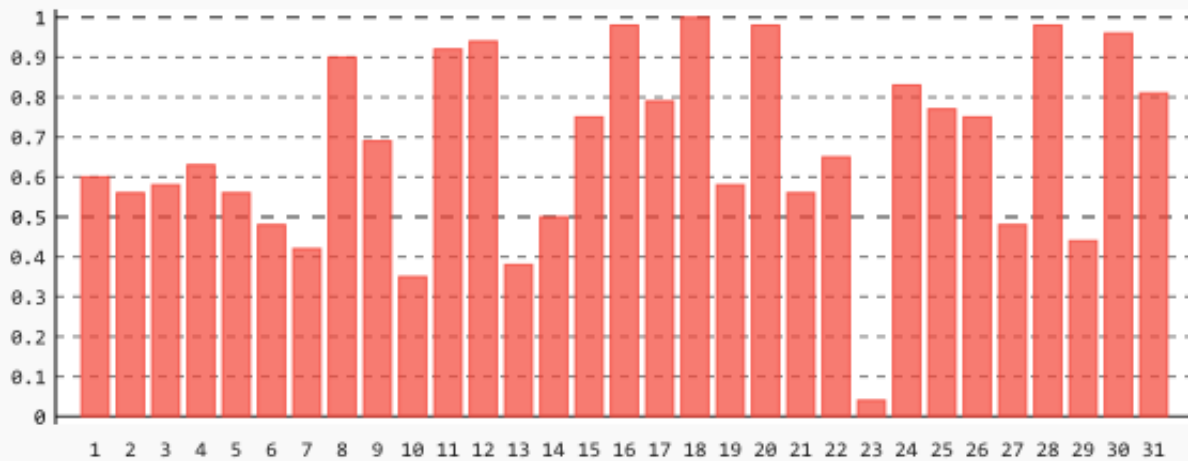
## Question Breakdown

---

There are two sorts of question in this exam: automarked questions (e.g., multiple choice questions (MCQs), true/false, multiple response, etc.) and manually marked questions (e.g., essay questions).

## Automarked Questions

### Automarked Question Average Scores



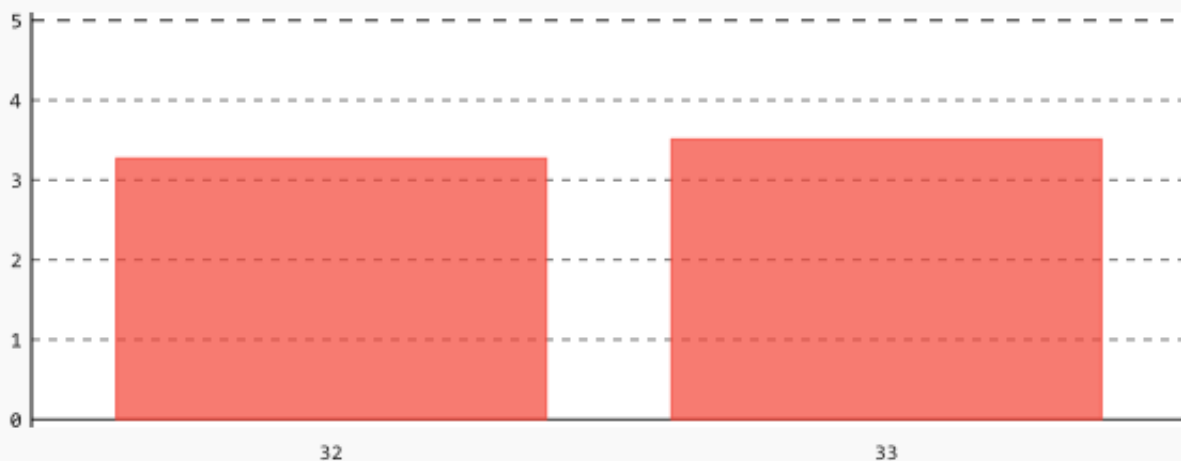
Most questions fell into the “medium” difficulty range with perhaps an unusually high level of “easy” questions (i.e., >90% passing). This itself isn’t necessarily an issue but may reflect better preparation.

Q23 was very hard for this cohort and a bit tricky. The definition of refactoring is producing a *sufficiently functional* version with an improved internal quality. This doesn’t mean that all functionality is untouched! “Irrelevant” behavior might be touched and even somewhat relevant functionality might be. We saw this with the argparse refactoring.

It’s not enough to sort of memorise a definition. This cropped up again in the essays

## Manually Marked Questions

### Manually Marked Question Average Scores



## General Question Feedback

The students performed well in both essay questions and exhibited a high level of understanding with respect to refactoring and McConnell’s paradox.

- **Q32** mean = 65% (3.27 out of 5; min = 2.0; max=5.0):

In the majority of the cases students failed to expand on mitigation strategies for technical debt or lower code quality

- **Q33** mean = 70% (3.51 out of 5; min = 2.5; max=5.0):

A significant amount of students defined refactoring as the transformation of a program to improve its internal qualities while being functionally equivalent.

The correct answer is the transition to a *sufficiently functionally equivalent* program.