Practical 1 Review

(version 1.6)

Conrad Hughes

University of Edinburgh

Software Testing, 5 March 2010

Outline

1. Marks

2. Improvements

Statistics

- Mean: 71%
- Standard deviation: 18%

Know the question

- Briefly re-state it in your own words so I know you understand it. If you then answer a different question to what I intended, at least I can see that you're following a logical process and try to give marks for this.
- Look up your terms, and define them
  - Very few people defined "relative URL"
  - Three mentioned RFC definitions for URLs or related issues (1738, 1808, 2396, 2732, 3986); somewhat more had some mention of a definition.
    - (1808, 2396 and 2732 are obsolete)
- Task 1.1 (specification) was one of the weakest.

Research your subject

- Sometimes you need to learn more about the area you're working in:
  - Extremely few people noticed that StringTokenizer is a legacy API (not quite deprecated, but close).
- Cite your references:
  - Very few people cited more than one reference other than my slides, and mentioned these citations in their main text.
  - It's not compulsory or expected, but does show that you've done your reading (and reading will usually improve the quality of your understanding and your work).
  - Beware the university regulations on plagiarism (linked from both ST practicals)...

On citation: paraphrase

Ntafos[1] states more extensive testing strategies aren't clearly better than using combinations of weaker strategies.

. . .

[1] Ntafos, *A comparison of some structural testing strategies*, IEEE Transactions on Software Engineering, June 1988.

GOOD

On citation: quote

Ntafos[1] states that “is not at all clear that adopting a more extensive strategy is preferable to using a combination of simpler strategies or extending a simpler strategy.”

. . .

[1] Ntafos, *A comparison of some structural testing strategies*, IEEE Transactions on Software Engineering, June 1988.

FINE

On citation: overquote

Ntafos[1] states:

> *"It is not at all clear that adopting a more extensive strategy is preferable to using a combination of simpler strategies or extending a simpler strategy. For example, consider testing an IF-THEN structure using branch and segment testing. To achieve branch testing we need to use two paths while segment testing can be achieved with just one path. Assuming that test data are selected in a similar fashion from the path domains, we can claim that branch testing will be more effective than segment testing. However, if we use two independent test cases for segment testing, it follows that segment testing will be more effective than branch testing in detecting errors within the body of the THEN branch. This points out a major deficiency that all structural testing strategies share. Many errors along a path can only be detected if the path is executed with values from some subset of its subdomain."*

. . .

[1] Ntafos, *A comparison of some structural testing strategies*, IEEE Transactions on Software Engineering, June 1988.

NOT SO GOOD

On citation: unclear quote

It is not at all clear that adopting a more extensive strategy is preferable to using a combination of simpler strategies or extending a simpler strategy. For example, consider testing an IF-THEN structure using branch and segment testing. To achieve branch testing we need to use two paths while segment testing can be achieved with just one path. Assuming that test data are selected in a similar fashion from the path domains, we can claim that branch testing will be more effective than segment testing. However, if we use two independent test cases for segment testing, it follows that segment testing will be more effective than branch testing in detecting errors within the body of the THEN branch. This points out a major deficiency that all structural testing strategies share. Many errors along a path can only be detected if the path is executed with values from some subset of its subdomain.

. . .

[1] Ntafos, *A comparison of some structural testing strategies*, IEEE Transactions on Software Engineering, June 1988.

BAD

On citation: plagiarism

It is not at all clear that adopting a more extensive strategy is preferable to using a combination of simpler strategies or extending a simpler strategy. For example, consider testing an IF-THEN structure using branch and segment testing. To achieve branch testing we need to use two paths while segment testing can be achieved with just one path. Assuming that test data are selected in a similar fashion from the path domains, we can claim that branch testing will be more effective than segment testing. However, if we use two independent test cases for segment testing, it follows that segment testing will be more effective than branch testing in detecting errors within the body of the THEN branch. This points out a major deficiency that all structural testing strategies share. Many errors along a path can only be detected if the path is executed with values from some subset of its subdomain.

DISCIPLINARY ACTION

Pay attention!

- Many people wrote tests which failed for reasons completely independent of the intended test (e.g. `new URL()` throwing an exception).
- Many people wrote tests which passed for the wrong reasons
  - Eclipse and JUnit make it easy to check this stuff!
- Lots of you tested `toRelativeURL(null, null)` — beware errors masking each other!

Manage your time

- I think a huge number of people ran out of time:
  - T1 average was 77%, T2 70%, T3 69%
  - Even though T3 is probably easiest.
- Pay attention to question value
  - Task 3 was worth 40%, but many only wrote half a page for this. Others wrote ten (that's too much, but there's a middle ground).

(Pretend) I'm an idiot

- Explain everything you do.
  - If you're being clever, explain *very* carefully.
- Show your workings.
- Code is important, but if you make mistakes in your code and haven't told me what you intended (in your report), then it's very difficult for me to mark. Many people wrote nothing about their erroneous variants (I diffed them and read the comments, don't worry).