# COMP311 Assignment 1

## Questions based on Chapter 1 of Software Testing by Ron Patton

### Instructions

Work in teams of maximum two students. Read the sample chapter handed out. Then discuss and answer the following questions together.

### Marks

This assignment is marked out of 10 and count toward about 2% of your final mark.

### Team

Print the names of both team members below.

Garnett Grant 301188923

1 Garnett Grant Student number 301188923

2 KD Aklilu Student number 301220223

### Questions

1. Which phase in a software development process is cause of main cause for most bugs?

   (1)

   Specification

2. The sample chapter gives five rules for determining what is considered to be a bug in a software product. Which of the following is not one of those five? (1)
   Hint: the wording is changed to make you think.
   1. The product does not perform some functionality required by the specification.
   2. The product offers functionality beyond what is required by the specification.
   3. The product functions in a way that is different from what is described in the specification.
   4. The product specification does not reflect the end-users needs or does not solve the problem that motivated the development of the product.
   5. The product fails to meet non-functional expectations such as performance, learnability or usability.

4, The product specification does not reflect the end-users needs or does not solve the problem that motivated the development of the product.

# COMP311 Assignment 1

3. Which one of the five rules is left out of the list of options for the previous question? (1)
   The software doesn't do something that the product specification doesn't mention but should.

4. Philip Crosby, one of the pioneers of quality assurance coined the phrase Quality is free. Clearly applying QA is not literally "free" because the company must pay salaries of QA professionals and testers and cover the costs of following QA and test processes.
   What did Crosby mean by claiming that quality is free? (2)
   I believe Crosby is claiming quality is free because of the corresponding cumulative cost of "poor quality".
   He believes quality done right the first time is more beneficial
   When contrasting the amount of money, effort, and time that gets put into poor quality activities (retesting, troubleshooting, etc.,) quality ends up in it's own cost classification, i.e., free.

5. In the example of the Y2K bug, do you think that the hypothetical programmer Dave took the wrong approach in storing years as only two-digits. If your group has divided opinions, say so. (2)
   Hint: Give a brief justification that shows deeper analysis than:
   "Dave was wrong because two-digit years caused the Y2K bug."
   I believe Dave was wrong in his approach to storing years as only two-digits because when developing software, scalability should always be in mind.
   "How can I make this easier for the next person".
   Rigorously looking at scenarios and not discarding concerns as irrelevant is important.
   Whether an issue found affects someone 10 years away or tomorrow,
   it should be addressed with some sort of plan to avoid surprises in the long run.

6. Can you give other example of famous product recalls or notorious software bugs that you learned about from the news or experienced personally? DO NOT RECYCLE examples from the chapter! Present in class.
   Hint: You may search the Web if you give the URL of your main reference below.

Briefly describe the problem and the impact it had on the company that produced the product and on users. If possible, apply the wisdom of hindsight to suggest how company could have prevented or reduced the impact of the problem? (3)

The Ariane 5 rocket explosion in 1996 serves as a significant example of the impact of software bugs. Here's a brief overview based on Ron Patton's insights from his book "Software Testing":

**The Problem:**

Nature of the Bug: The explosion was due to a software error involving an integer overflow. This occurred when a 64-bit floating-point number was incorrectly converted into a 16-bit signed integer.

Contextual Oversight: The software, originally used in the Ariane 4 rocket, was not adequately tested or adapted for Ariane 5's different flight conditions.

**Impact on the Company and Users:**

Financial Loss: The failure led to a loss of about $370 million, considering the cost of the rocket and the onboard scientific payloads.

Reputation Damage: For the European Space Agency and the companies involved, the incident was a significant setback, tarnishing their reputation for reliability in the space industry.

**Prevention and Mitigation:**

Rigorous Testing and Adaptation: The incident underscores the importance of thorough testing, particularly when reusing software in different contexts. Adequate testing under Ariane 5's operating conditions might have revealed the issue.

Robust Specification and Design Review: Ensuring that specifications and designs are well-communicated and adapted for the specific context of a project is crucial. In this case, a review of how the software would behave under the new flight conditions of Ariane 5 was necessary.

Proactive Risk Management: Identifying potential risks, especially in critical systems, and managing them effectively could prevent similar incidents.

In summary, the Ariane 5 explosion highlights the critical need for thorough specification, careful adaptation, and rigorous testing in software development, especially for high-stakes projects like space exploration.

URL:_ https://www.esa.int/Newsroom/Press_Releases/Ariane_501_-_Presentation_of_Inquiry_Board_report#:~:text=A%20detailed%20account%20is%20given,seconds%20after%20lift%2Doff).

https://en.wikipedia.org/wiki/Ariane_flight_V88