**Chapter 1- The Case for Software Quality**

That in 1953, Francis Crick and James Watson described the structure of deoxyribonucleic acid (DNA) as a double-helix structure and thus began a scientific pursuit of the Promethean power of genetics. But unraveling the sheer volume and complexity of the genetic information contained in DNA was a computational problem far ahead of its time. It was the magic of software decades later that was the deciding factor in unlocking the genetic code and the full promise of DNA research.

It’s hard to imagine that the instruments necessary to perform such a search would be possible without software, which played a role not only in the design and use of the instruments themselves but also in the analysis of the data they produce. And now, thanks again to software, the universe is as close as your home computer, vastly increasing the number of eyes looking for exoplanets. If an earth like exoplanet is ever found, you may rest assured that the magic of software will be central to its discovery and confirmation

Do we trust software to perform these globally important tasks? Software bugs have been responsible for many disasters, from stranded ships to exploding rockets to the loss of life and fortune. Software is designed by imperfect humans, and our creations and inventions echo this unnerving tendency to fail. Bridges collapse, airplanes crash, cars break down, and nothing we build manages to successfully navigate.

Name some ways in which software has changed the world?

Software is used by businesses to communicate Worldwide. It is used to operate some of the most complex medical machines. It regulates nuclear plants, electrical grids, and other utilities that provide the basics of life.

How technology changing the way children are raised?

Technology use can cause social and behaviour problems in children because it minimizes the amount of time kids spend interacting with others. Make sure to monitor child’s social media use and be aware of the types of websites they’re visiting and the games they’re playing online.

How is technology changing the way teenagers interact with their peers?

The use of digital applications has also dramatically changed the way adolescents relate to their peers, access information, and engage in social relationships, and also had a profound influence on their health, including their well being

How has technology changed business? Government?

Technology have made elements of business such as manufacturing, communication, purchasing, sales and advertising easier and more effective for businesses. Changes in technology have included E-mail enables written messages to be sent instantly to others, and files can be shared as attachments.

Can you name five negatives about technology and software?

1. Depression and other Mental health issues

2. Lack of sleep

3. Obesity

4. Learning Barriers

5. Decreased Communication and Intimacy

6. Cyberbullying

8. Loss of Privacy

A software failure caused loss of human life.

Therac-25 caused several deaths and injuries due to software bugs. The machine was designed to deliver precise doses of radiation to cancer patients, malfunctioned and delivered massive overdoses of radiation to some patients.

**Chapter 4 Exploratory Testing in the Large**

**Exploring Software**

The techniques presented in the preceding chapter help software testers make the numerous small decisions on-the-fly while they are running test cases. Those techniques are good for choosing among atomic inputs and arranging atomic inputs in combination or in sequence.

Testers must make many decisions while they are testing. There are big decisions, like how to obtain realistic data to simulate customer databases. There are small decisions, like choosing what string of characters to enter in a text box. Without the proper mindset and guidance, testers can end up wandering aimlessly around an application’s interface looking for bugs that may or may not actually be there and gaining only poor coverage of the application in the process.

• To gain an understanding of how an application works, what its interface looks like, and what functionality it implements

• To force the software to exhibit its capabilities

• To find bugs

**CHAPTER 5 Hybrid Exploratory Testing Techniques**

**Scenarios and Exploration**

This chapter shows how the exploratory testing mindset can be combined with more traditional scenario-based and scripted testing. This hybrid technique relaxes much of the rigidity ordinarily associated with scripting and makes good use of the exploratory testing guidance presented in the last two chapters. It also allows teams that are heavily vested in existing scripts to add exploratory testing to their arsenal.

Traditional scenario testing is very likely to be a familiar concept for the reader. Many testers write or follow some sort of script or end-to-end scenario when they perform manual testing. Scenario testing is popular because it lends confidence that the product will reliably perform the scenario for actual user.

There is no formal definition of scenarios that I know of which really helps testers. Some scenarios are like maps, providing only general guidance, and others are more like printed driving directions with step-by-step instructions for every turn and intersection. In general, scenarios are written prose that follow no fixed format but describe how the features and functionality of the software under test work to solve user problems.

**Applying Scenario-Based Exploratory Testing**

Testers often use scenarios that describe user intent to test the software. Scenario testing works because it mimics the way a real user would behave, and thus it finds bugs that, if they survived testing, would plague actual users.

**Introducing Variation Through Scenario Operators**

Exploratory testing can be combined with scenario testing to help a tester explore minor and even major variations on a specific scenario. Where a scenario describes specific actions for a tester to take, the techniques described next can be used to permute those actions and create deviations from the scenario that will test different states and code paths. Where a scenario describes general activity, these techniques can be used to select among the possible choices and allow a tester to consider alternate paths in a more methodical manner.