FOUNDATIONS IN INNOVATION AND TECHNOLOGY CERTIFICATE

Software Tester

FITC 140

Week 1 - Lesson 1

ACKNOWLEDGMENT

We walk gently as a guest working on the unceded homelands of

Nlaka'pamux and Sylix Nation.

We acknowledge the traditional lands of this Nation and it is with gratitude that we now work, learn, and stay here as your guest.

ACKNOWLEDGMENT

We gratefully acknowledge the many Indigenous peoples
that came forward to have their photos taken,
be videotaped, and to share their voices
throughout the Foundations curriculum.

"All My Relations"

Elder Glida is dropping to say, “WELCOME… YOU’RE ONE OF US!”

First Nations
Technology Council

Gathering Time

Time to gather in a circle for some check-in time.

Leave your desks behind and bring your chairs.

Grab a stone so you can pass it around. Whoever talks will hold the stone and everyone else will listen.

Someone from the host Nation can be offered to start the check-in first. If no one is interested then someone else in the group, or the Instructor, will be offered to start it next.

Gathering Time

Pass the sharing stone around, one at a time answer the questions (ask much as you are comfortable):

- Your name, Nation (if you know it), a little bit about yourself, and how you are feeling today
- Why you are involved in the FiiT Program
- Your goal after the course is finished
- Another fun fact about you!

*If you don’t want to share just pass the stone to the next person.

What we'll cover in the course…

- This course introduces students to the foundational concepts and basic skills necessary for pursuing a career in software testing, including methodological approaches, industry-standard tools and best practices related to software testing.Principles of Software Testing

Learning Outcomes

Upon successful completion of this course, students should be able to:

- articulate why software is tested;
- identify the relationship between the software development lifecycle and testing practices;
- describe the role of the software tester and their relation to various stakeholders in a testing environment;
- list major test techniques and their recommended applications;
- identify activities related to test design and how these are carried out and documented;
- execute bothe a scripted and exploratory manual software test;
- report simple bugs and document them in a defect report; and,
- define test automation and identify its advantages in a testing environment.

What have you learned so far in the Fiit program and how does it fit?

Computer Basics & Professional Practice

Communication and Digital Marketing

Network Technician

Web Developer

First Nations
Technology Council

Welcome to

FITC 104

Software Testing

his Photo by Unknown Author is licensed under CC BY-SA-NC

First Nations
Technology Council

Lesson Objectives

- What is Software Testing?
- Software Testing Principles
- Fundamental Test Process
- Psychology of Testing
- Terminology

So, you want to be a Software Tester?

This Photo by Unknown Author is licensed under CC BY-SA-NC

¯_(ツ)_/¯

This Photo by Unknown Author is licensed under CC BY-SA

Let's hear about Maggie's insights into software testing…

Indigenous ways have been around for thousands, and thousands, and thousands of years [time immemorial].

Each Nation has its own teachings.

Do you ever consider whether traditional teachings can be applied to the tech sector?

Maggie Patsey, Indigenous name - Ant'aam Lax Gaa'askw Gaak "Raven looking over the rest of them" is from the Nisga'a Nation's House of Ksim Xsaan. Maggie shares a story about how she imagines software testing being similar to the quality assurance skills they practice.

First Nations
Technology Council

Traditional Teachings and the Tech Sector

"Software Testing is similar to the quality assurance that you apply when you prepare for a dance performance. Dance performance skills require individuals to understand the scope of the messaging, to understand that performers work with the owner of the dance they are performing. The dancers identify roles and plan, monitor and articulate scripts and responsibilities. It also requires that you practice and test to ensure that the dance is seamless. The dance is practiced to identify any potential problems that could be associated with the dance, the story, the surroundings, the prop, or tools. This all requires the dancers to walk through the dance as many times as it takes to prepare for performing. They do this to carry out testing that will be a powerful performance that will impact the audience and tell the story being shared. When the dancers get the kinks out and everything associated with the dance is ironed out in the dance."

What is Software Testing?

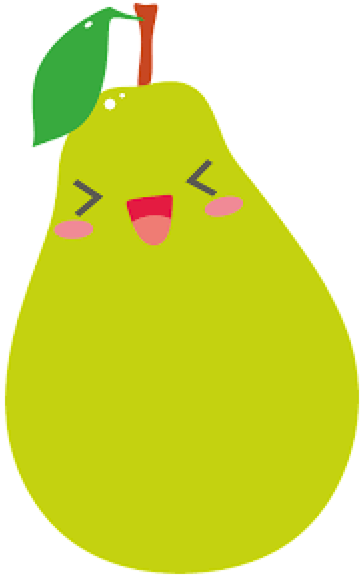
Sarah attended PLATO Testing, the first Aboriginal Software Tester Training in Canada.

She is so excited about her new career as a software tester. The following is what she has learned about why software testing matters.

[What is software testing and why is it important?](#)

First Nations
Technology Council

“Pear” & Share



Have you heard of software testing before?

Can you think of any products you use that may rely on software testing?

Assignment 1

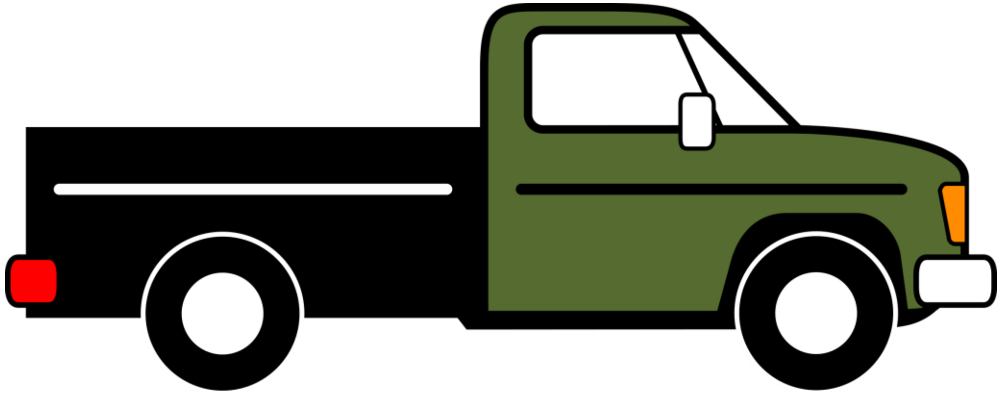
Terminology

First Nations
Technology Council

The following types of businesses, companies, and services all rely on software testing in the research and development of their products:

- Banks
- Airlines
- Healthcare Providers
- Cell Phones, Laptops & Tablets
- Social Media Tools
- Video Games

Software Testing is like....



If you were buying a vehicle, you would expect that it would have four wheels, a steering wheel, an engine, and all the other essential components, and that it should come with appropriate documentation, with all pre-sales checks completed.

The vehicle you see should be the one described in the sales marketing . It should have the correct engine size, the correct colour scheme, and whatever else described.

In short, a level of expectation is set by the marketing, by your experience of sitting in the driving seat, and probably by a test drive. If your expectations are not met you will feel, understandably, misled.

First Nations
Technology Council

"Aboriginal people have a keen eye for differences and that's very key for software testing because you need to find things that the average Joe wouldn't find.“

Shawnee Polchis-Lanteigne (centre),
Software Tester

Shawnee Polchis-Lanteigne

Featured in the Globe and Mail

Software Testing

Why Does Software Fail?

- People make mistakes – we're only human!

- Deadlines, complex systems, and changing technology all increase the likelihood of errors in codes

- Environmental conditions can also cause system failures - radiation, magnetism, electronic fields or pollution

Incorrect software can harm:

People

The Environment

Companies

Incorrect software can lead to:

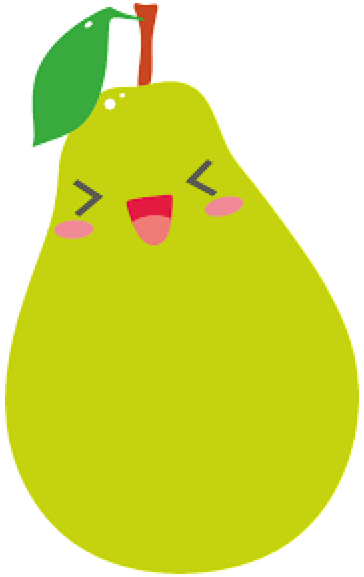
- Loss of money & time
- Frustration
- Loss of business & reputation
- Injury
- sometimes even death

So How Does Software Testing Actually Work?

Software Testing

Netscape Mozilla

“Pear” & Share



What did you think of the Netscape Mozilla documentary?

Does the documentary inspire you to learn more about software testing?

Software Testing

- Also known as Quality Assurance (QA)
- Can be done manually by an individual, or using automated tools
- Requires designing & carrying out specific activities to check whether a software's actual results match expected results
- Helps to identify gaps, errors or missing requirements*

First Nations
Technology Council

Testing

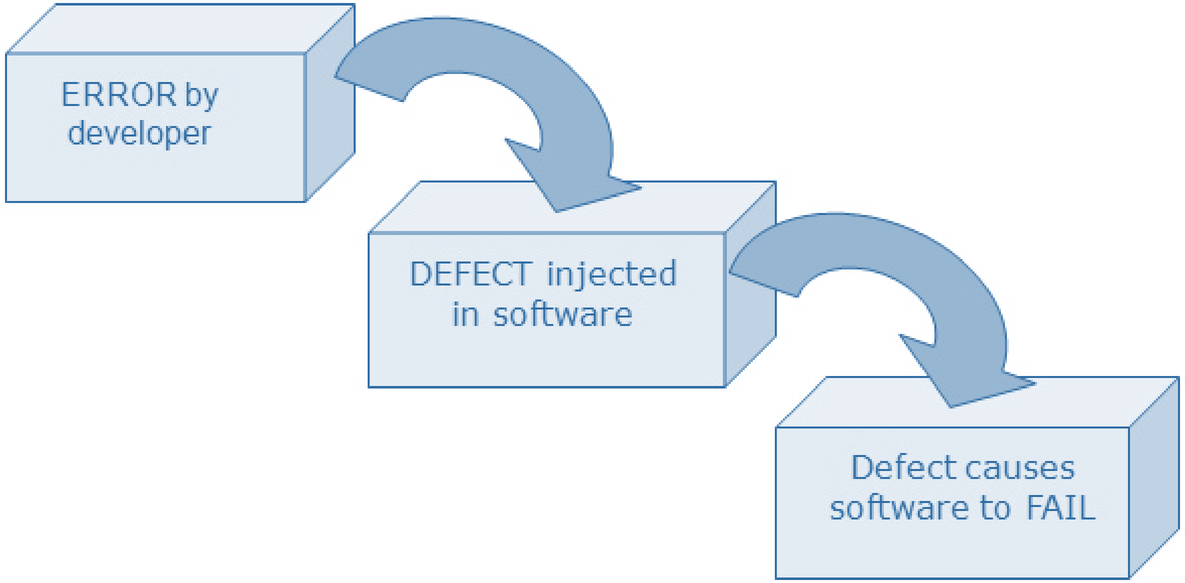
The process of finding and reporting errors or defects in a software's code

Guarantees that any changes or corrections are checked for their effect on other parts of the system

Debugging

The tracking down & fixing of the source of a software failure

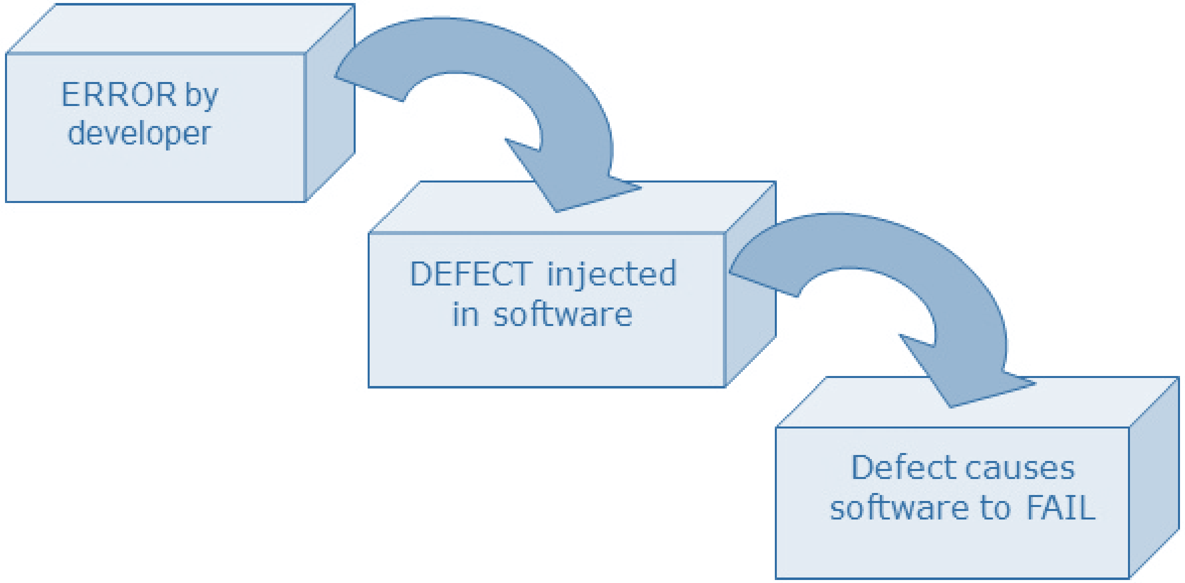
Testers flag bugs to be fixed & work with developers to fix the problem



If a document with an **error** (or mistake) in it is used to build a component, the component will be faulty and will probably display **incorrect behaviour**.

If this faulty component is built into a system, the system is then **defective**.

While failure is not always guaranteed, it is likely that **errors in software will lead to defective components**, and defective components will cause system failure.



Error

A mistake in the software’s code

Defect

The difference between the expected and the actual result – the thing that isn’t working the way it should

Failure

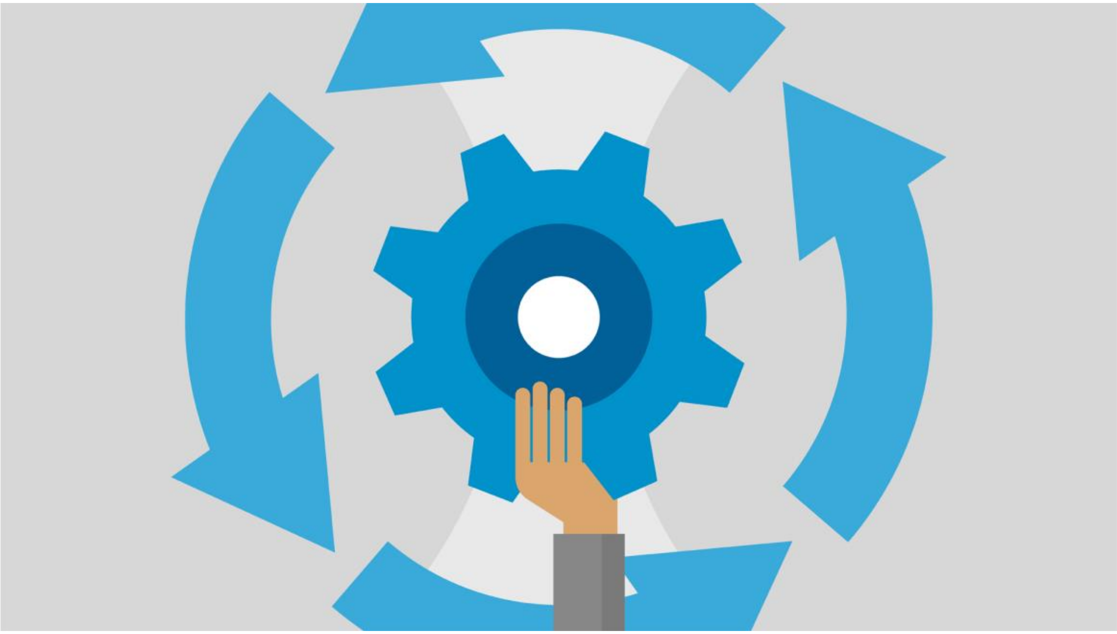
When the defect reaches the end-user and doesn’t work as it is expected to

Testing as a Process

- It’s important to decide what you’re trying to achieve with the testing & set clear goals for each test

- Before testing: design the tests and set them up

- After testing: record the results and check whether the tests are complete



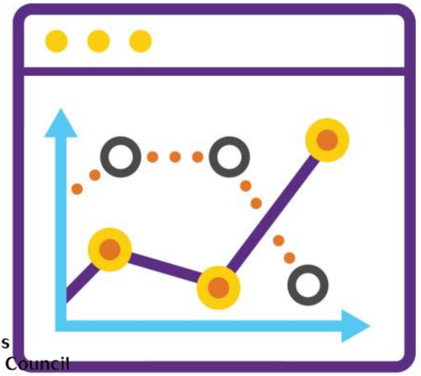
Testing as a Set of Techniques

- Testing costs time and money, so it’s important to be as efficient and effective as possible
- Important to stick to well-proven test design techniques
- Always follow the key principles of software testing to guarantee the best results

Static Testing

Testing which does not require a run-through of the software = the program’s code is not executed

Means that the code, and any related documents, are checked for errors



Dynamic Testing

Testing in which the code is run-through & the software is executed

Checks for functionality & overall performance of the software, ensures expected outcomes meet the actual ones

Activity 1

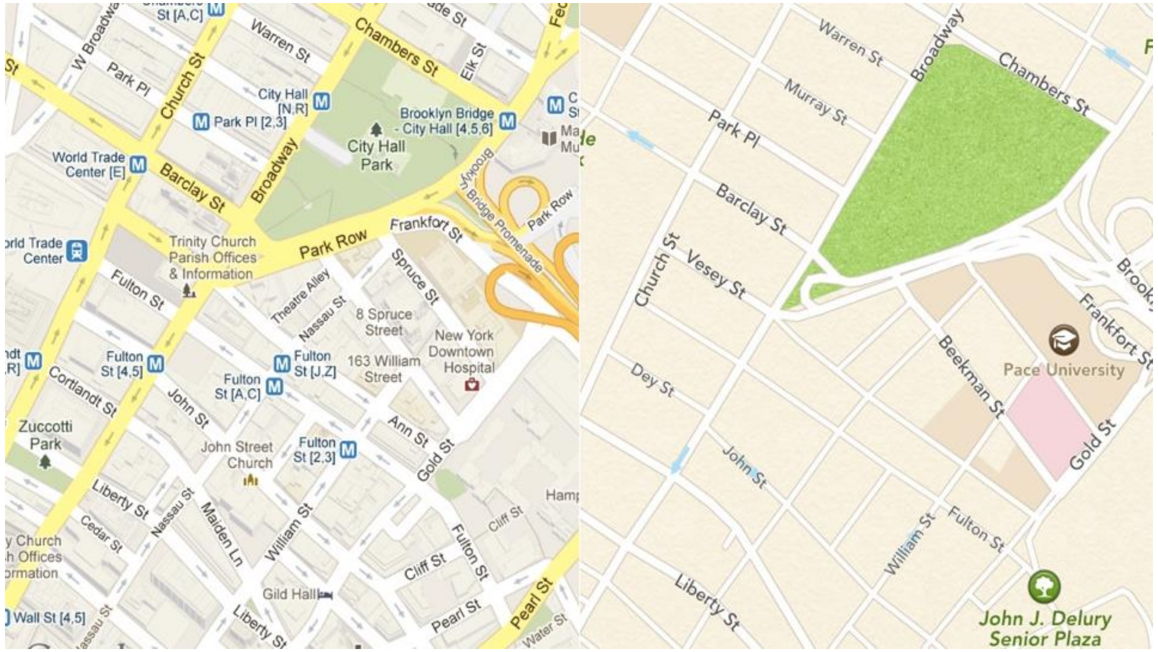
Case Studies

Uncollected Taxes

The Internal Revenue Service has never been an organization to turn down money, except in one now-famous instance in 2006 where it trusted a computer program to call out potential fraud cases in returns claiming refunds. The tax collection agency wasn't aware their program was inoperative until it was too late, costing what the Associated Press via the Houston Chronicle estimates was between $200 million and $300 million in revenue.

First Nations
Technology Council

Mapping Glitches

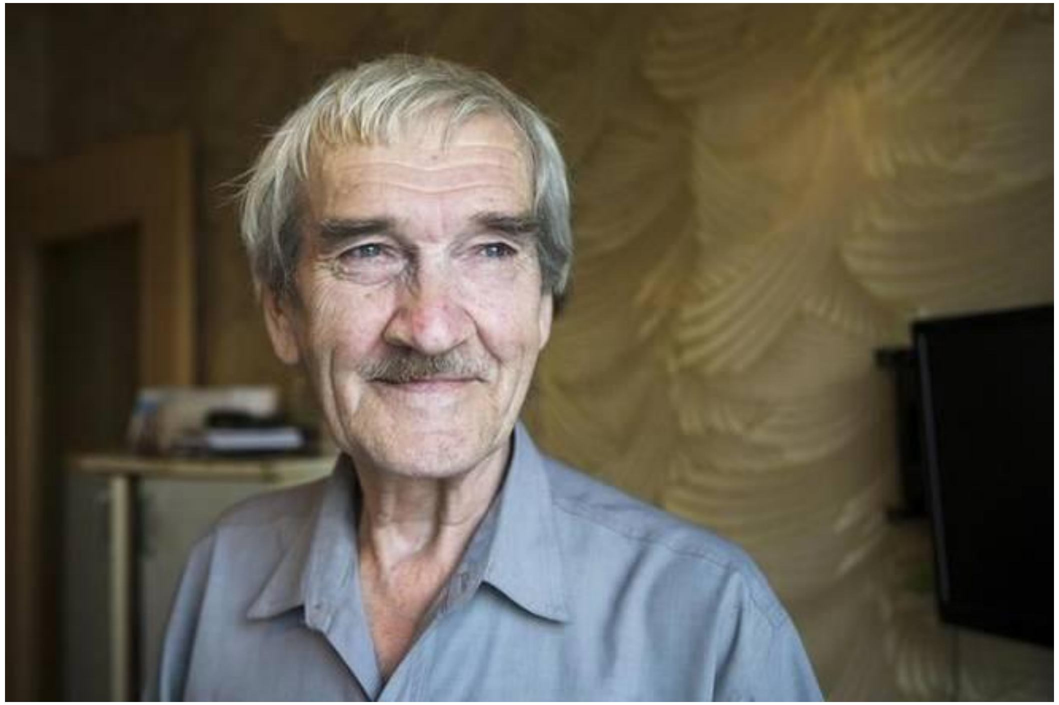


With the 2012 Apple iOS 6 update, the company decided to kick the superior Google Maps platform to the curb in favor of its own system. Unfortunately, it did a poor job of mapping out locations resulting in one of the most epic fails of the mobile computing movement. The software missed entries for entire towns, incorrectly placed locations, gave incorrect locations for simple queries, confused satellite imagery obscured by clouds, and more.

Millions in the Dark

On August 14, 2003, a blackout across eight US states and Canada affected 50 million people. PC Authority described the cause, a race condition bug, as something that occurs when "two separate threads of a single operation use the same element of code." Without proper synchronization, the threads tangle and crash a system. That's what happened here with the result 256 power plants offline. The major disruptions manifested themselves in the form of cellular communication with the best form of communication during the outage said to be a laptop using a dial-up modem.

Crisis Avoided



Russia — or Soviet Air Defence officer Stanislav Petrov, to be exact — may have saved the world on the night of September 26, 1983, when the Soviet nuclear early warning system malfunctioned and erroneously reported that the US had launched an attack on his country. Petrov later told the Washington Post that he "had a funny feeling in my gut" that the alarm was false, which was confirmed through further investigation.

Washington Post

Software Testing World Cup Finals 2016

There's a World Cup?

World Cup

Software Testing Principles

Testing can be a **complex** activity.

Because of this, some general testing **principles** have been developed over time that **help to guide testers**, and prevent the types of problems described previously.

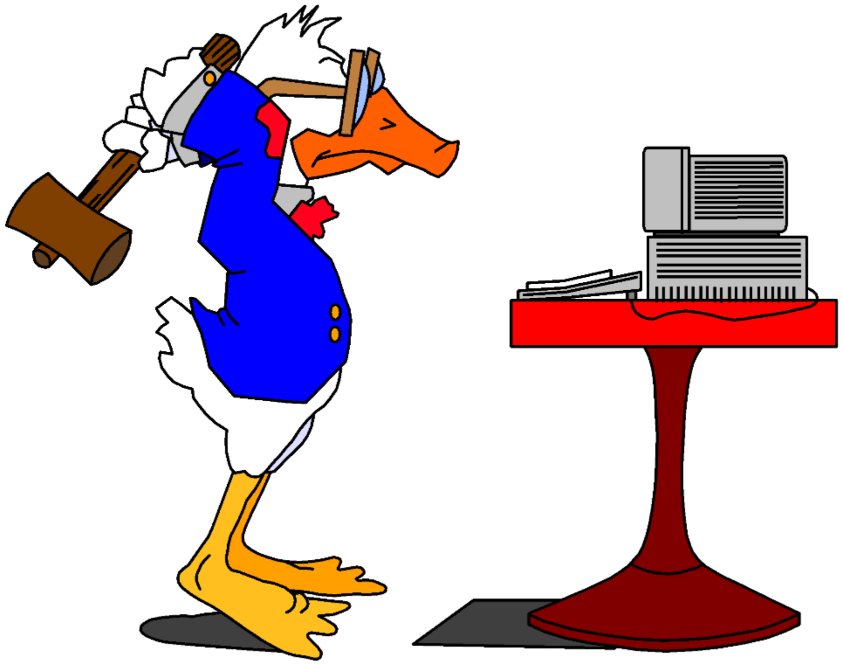
Testers use the following principles, along with test techniques, to create the **best possible conditions** for finding errors and making sure software does **what it's supposed to do**.

1. 100% Testing is Not Possible

- Testing absolutely everything is not possible

- Therefore, set priorities & determine a suitable level of risk

- The most important parts of a software should be identified & tested

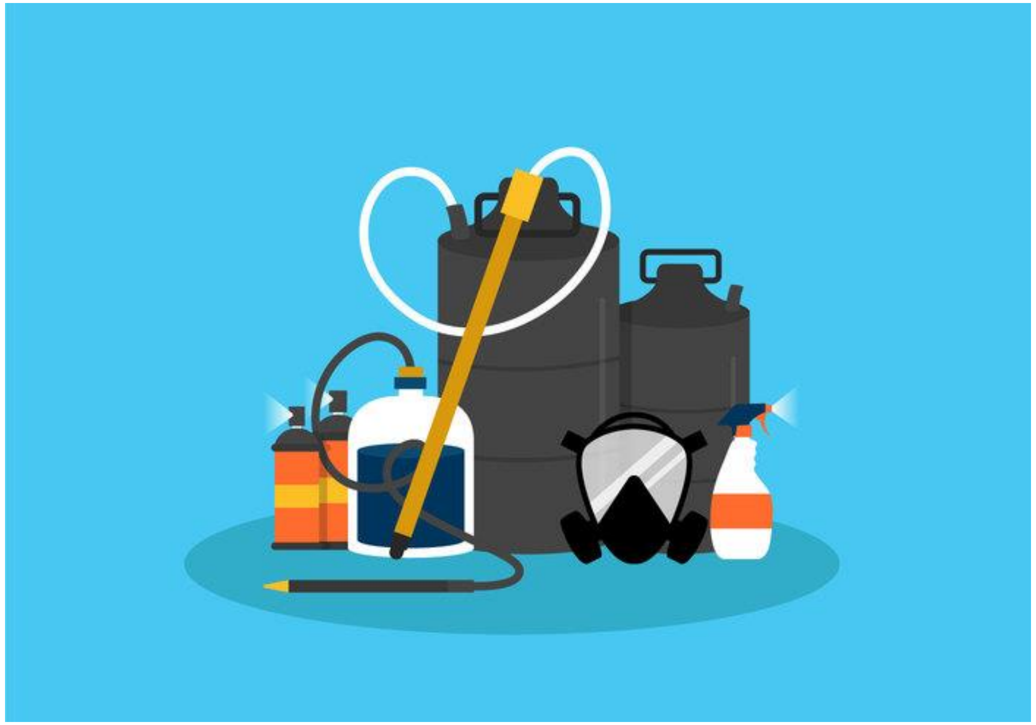


2. Defect Clustering

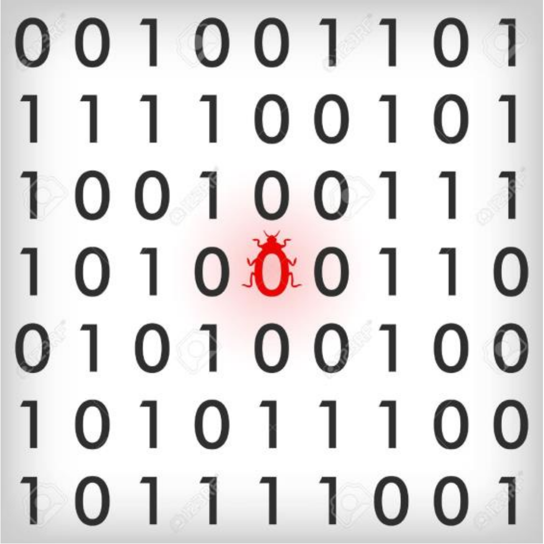
- Approx. 80% of problems are found in 20% of a software's modules
- The higher the number of defects, the more need for targeted testing in that area
- Defect clusters should not be the only places where a software is tested

3. Pesticide Paradox

- Repeating tests over and over again won't find any new defects
- It's important to use different techniques to find different types of defects
- Testing approaches should be regularly reviewed & revised



4. Testing Shows Presence of Defects



- The main purpose of testing is to find defects, not to show that a software is error-free

- Software testing reduces the chances of undiscovered defects

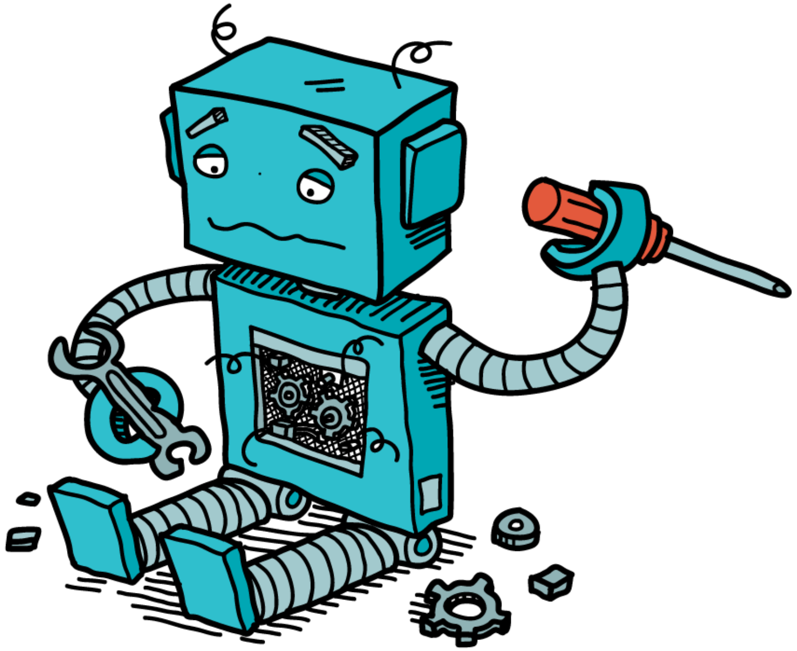
- A software that clears testing may still fail due to other issues

5. Absence of Error

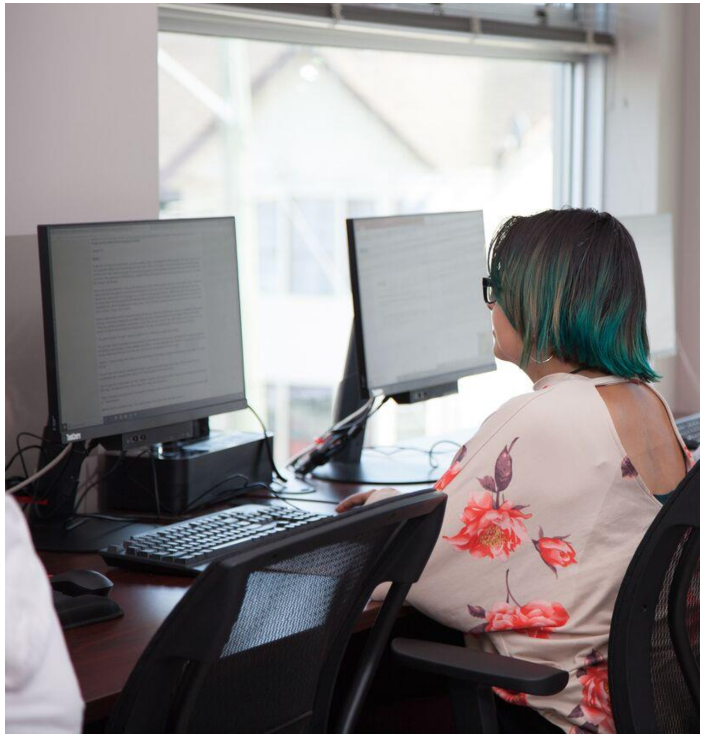
- It's possible that software which is 99% bug free is still unusable

- Finding & fixing defects doesn't help if the system itself doesn't meet the user's needs

- Software testing is also about making sure that software does what it's supposed to do



6. Early Testing



- Testing should begin as early as possible in a software's development
- The earlier a defect is found, the cheaper and easier it is to fix
- As soon as the requirements of a software module are outlined, it is possible to start testing

7. Testing is Context Dependent

- The way you test will depend on what you're testing = all software is not the same

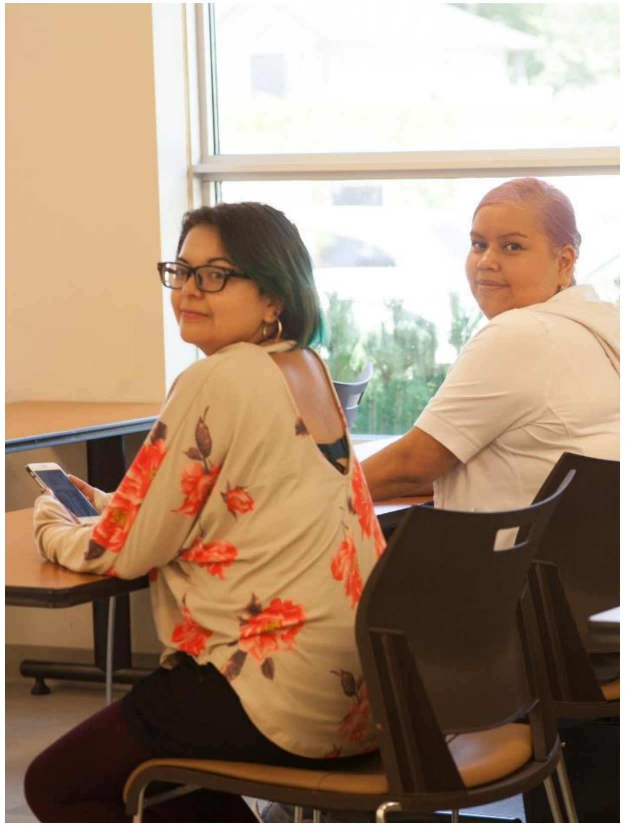
- Risk can be a major factor in deciding the type of testing that is necessary

- The higher the possibility of losses, the more investment into testing that is likely

Fundamental Test Process

The fundamental test process is composed of **five steps** that make up all aspects of testing – from the early planning phases to the final stages of reporting.

Important to remember is that this is **not always a one-way process**. Some steps may need to be **repeated**, as defects are detected, test approaches are modified, and exit criteria are determined.



1. Planning & Control
2. Analysis & Design
3. Implementation & Execution
4. Evaluating Exit Criteria & Reporting
5. Test Closure Activities

Planning & Control

Test planning and control include the following tasks:

- Creating a document that describes the overall approach to testing & its goals
- Comparing the progress of testing against the test plan
- Making adjustments to the plan to meet end goals

Analysis & Design

Testing analysis and design include the following tasks:

- Identifying test conditions
- Designing the tests
- Detailing set-up of the test environment & necessary tools

Implementation & Execution

Testing implementation and execution include the following tasks:

- Prioritizing and running tests
- Collecting data on test results & comparing actual results with expected results
- Re-executing tests that previously failed, to confirm fixes

Evaluating Exit Criteria & Reporting

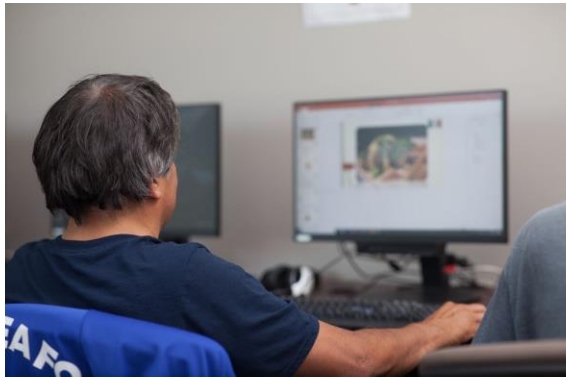
Exit criteria and reporting include the following tasks:

- Determining when to stop testing
- Assessing if more tests are needed, or if the exit criteria should be changed
- Writing a test summary report

Test Closure Activities

Test closure activities include the following tasks:

- Making sure that the planned deliverables are actually delivered
- Finalizing & archiving testware for later use
- Evaluating how the testing went & lessons learned for future projects



Psychology of Testing

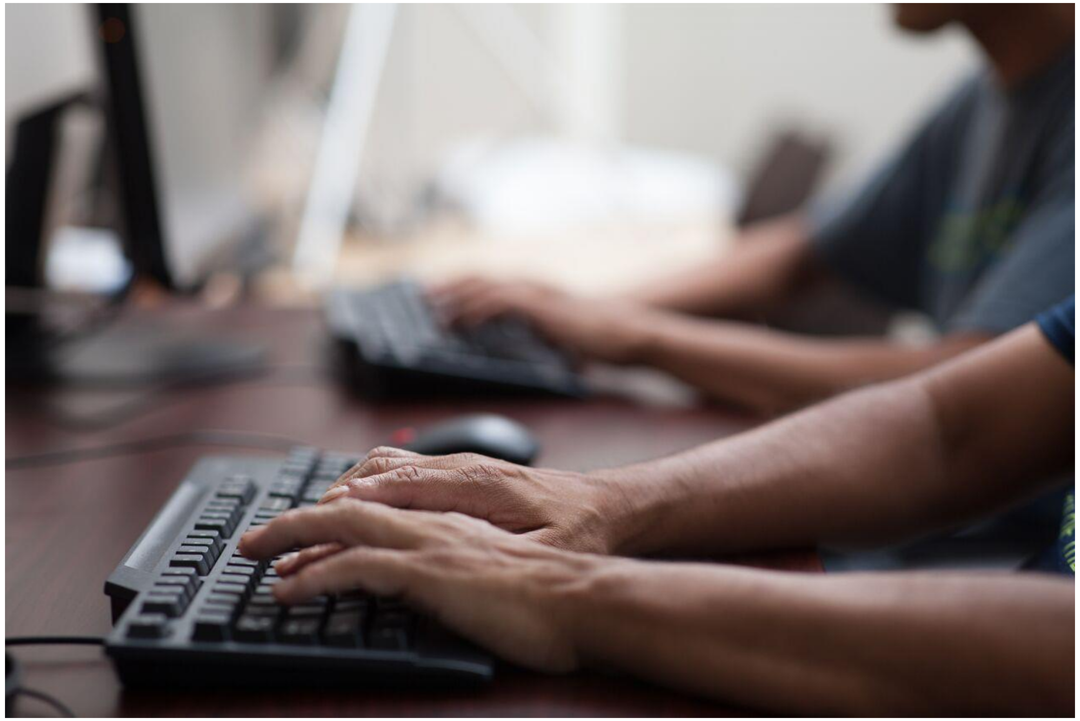
A variety of different people may be involved in the total testing effort, and they may be drawn from a broad set of backgrounds.

Some will be developers, some professional testers, while others may be users hired to assist with acceptance testing.

Who ever is involved in the testing needs at least some understanding of the skills and techniques of testing to make an effective contribution to the overall testing effort.

Who’s testing?

- Software Developers
- Software Testers
- Project Managers
- End Users
  the people who will actually be using the software

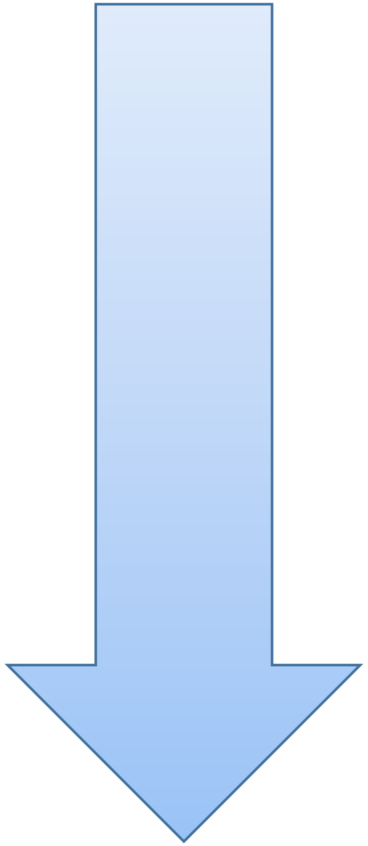


Test Independence Scale

Testing can be more effective if it is not done by the individual(s) who wrote the code, for the simple reason that the creator of anything (whether it is software or a work of art) has a special relationship with the created object. The nature of that relationship can mean that flaws in the created object may be invisible to the creator.

Therefore, the ultimate goal should be to have the software tested by someone who was not involved in its creation; this approach is called **test independence**

But independence comes at a price: it is much more expensive to hire an outside testing agency than to test a program internally



Test Independence Scale

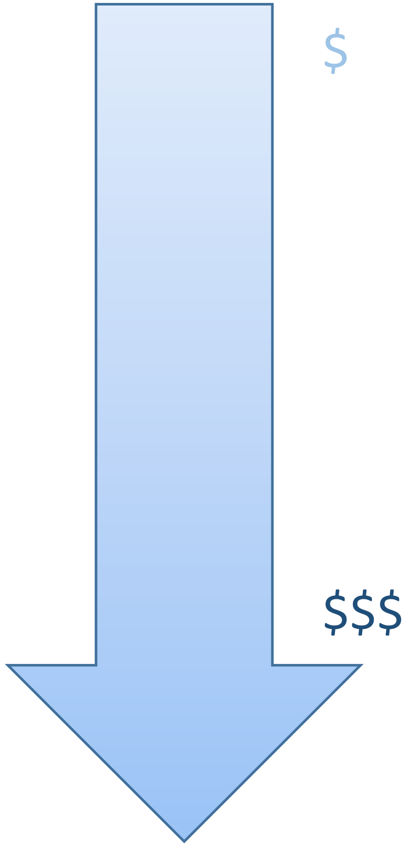
Limited Independence

Those who wrote the code

Members of the same development team

Members of an independent test team outside of the developer group

Members of a different company or testing agency



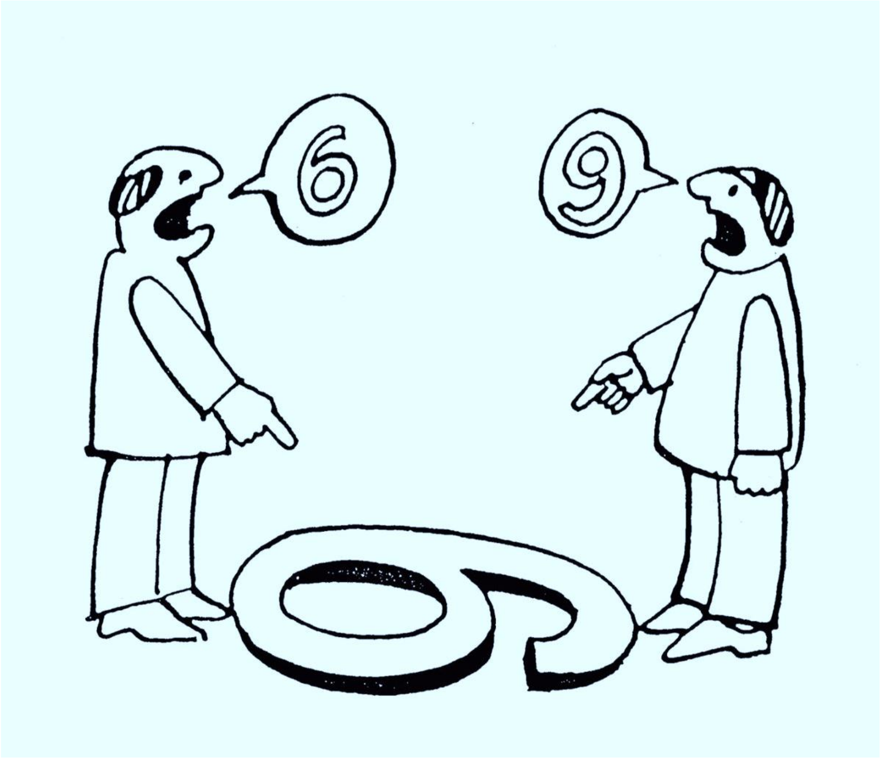
Maximum Independence

Testing and Communication

- Testers (QA) and Developers tend to work together closely on testing teams
- Testers and developers think in different ways
- Getting developers to also test software has advantages, but also difficulties
- If a defect is found in software, the software author may see this as criticism

Testing and Communication

- Working together makes sure the focus is kept on delivering a quality product
- One of the tester's main jobs is to capture issues that can come out of miscommunication
- Defect reports need to be raised against the software, not against the individual
- At the end of discussions, confirm that you have both understood and been understood



Activity 2

Testing and Ethics

First Nations
Technology Council

Testing Trivia Game

1. What is testing?

    a) The automated programs used to detect software failures

    b) The steps required to ensure unchanged areas of a software have not been affected

    [✓ c) The process of finding and reporting errors or defects in a software's code]

2. Debugging is the tracking down & fixing of the source of a software failure

    a) False

    [✓ b) True]

3. Running the same set of tests will not continue to find new defects. Which of the seven testing principles does this illustrate?

    a) Defect clustering

    [✓ b) Pesticide paradox]

    c) Absence of error

4. What is the relationship between errors, defects and failures?

[✓ a) Errors in software will lead to defective components, and defective components will cause system failure]

b) Defects and failures are the same thing, both due to the presence of errors

c) There is no relationship between errors, defects and failures

5. Static testing is….

a) Measuring the static electricity present on any piece of computer hardware

[✓ b) Testing which does not require a run-through of the software, and is instead checked manually]

c) When the expected outcomes of a test do not match the actual outcomes

6. Dynamic testing is…

- [✓ a) Testing in which the code is run-through & the software is executed to measure actual outcomes against expected ones]
- b) Measuring how quickly a software responds to different types of testing
- c) Testing which requires the tester to try the software on different types of hardware

7. Which, in general, is the least required skill of a good tester?

- [✓ a) Able to write software]
- b) Having good attention to detail
- c) Able to be relied on

8. Approximately 80% of problems are found in 20% of a software’s modules

- a) False
- [✓ b) True]

9. How much testing is enough?

a) This question is impossible to answer

[✓ b) The answer depends on potential risk, expectations, time available and funds]

c) The answer depends on the maturity of your developers

d) The answer should be standardized for the software development industry

10. The main objectives of testing include…

a) Detecting early defects

b) Gaining confidence in a software system

c) Preventing defects

[✓ d) All of the above]

11. Software testing is the most effective when it is started…

[✓ a) At the beginning of a software’s development]

b) After both internal and external design are complete

c) At the end of a software’s development

12. Tests are prioritized so that we…

a) Find more faults

[✓ b) Do the best testing in the time available]

c) Shorten the time required for testing

13. A reliable software application will be the one which is unlikely to cause a failure

a) False

[✓ b) True]

14. Which of the following is not phase of the Fundamental Test Process?

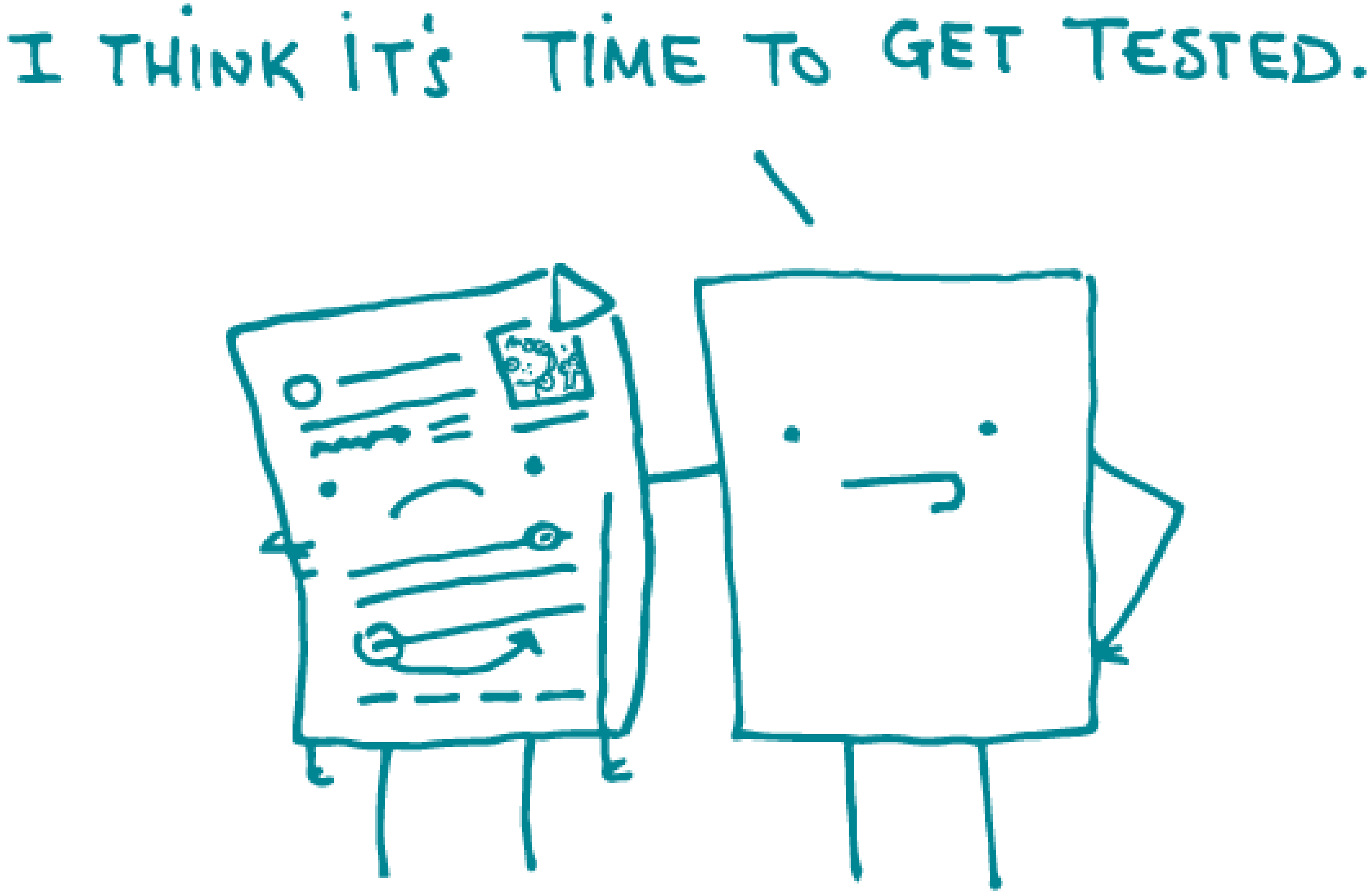
- a) Test Planning & Control
- [✓ b) Requirement Analysis]
- c) Test Implementation & Execution
- d) Evaluating Exit Criteria & Reporting

15. Prioritizing and running tests, collecting data on test results & comparing actual results with expected results are a part of which phase of the Fundamental Test Process?

- a) Test Closure Activities
- b) Test Analysis & Design
- [✓ c) Implementation & Execution]

Class Wrap-Up

First Nations
Technology Council



First Nations
Technology Council

Image Credits

- First bug - https://www.pinterest.ca/pin/571394271450948191/
- What we'll cover - http://www.cybersuccess.biz/course/software-testing-course/
- Have you heard of testing - https://www.invensis.net/blog/it/top-5-essentials-software-testing/
- Software testing is like - https://www.autoraptor.com/millennial-car-buying-habits-dealership-needs-know/
- Aboriginal people in software testing - https://beta.theglobeandmail.com/report-on-business/small-business/startups/indigenous-software-testing-company-pushes-boundaries-in-new-brunswick/article35459512/?ref=http://www.theglobeandmail.com&
- Tax - http://daverocker.org/tax-season-tips-filing-early-can-reduce-stress-maximize-your-return/
- Mapping glitches - https://www.phonearena.com/news/Apples-iOS-6-beta-Maps-app-compared-to-Google-Maps_id31266
- Unexpected consequences - http://www.noshiz.com/index.php?philo_ID=776
- Why does software fail- https://riddlestory.wordpress.com/category/it-quality-management/
- Error defect failure - http://qa-qcarena.blogspot.ca/2013/05/
- People - http://www.freepik.com/free-icon/group-of-people_755293.htm
- Environment – http://www.freepik.com/free-icon/leaf-black-shape-eco-symbol_743986.htm
- Incorrect software - http://www.jfmaccountants.com.au/software-and-technology-solutions.htm
- Head scratch - https://lauriekendrick.wordpress.com/2011/10/19/its-wednesday-kids/head-scratch/
- Software and Quality - http://techbodhi.co.in/courses/job-oriented/software-testing/
- Time cost quality triangle - https://www.linkedin.com/pulse/time-cost-quality-welcome-tradeoff-scott-moffatt
- Making decisions - http://www.savagechickens.com/
- Testing - https://www.acefastservices.co.uk/software-testing/
- Debugging - https://www.iconexperience.com/o_collection/icons/?icon=debug
- Testing as process - https://www.lynda.com/Business-Skills-tutorials/Process-Improvement-Fundamentals/365729-2.html
- Testing Principles- https://www.lynda.com/Windows-tutorials/Configure-software-update-synchronization-Windows-Update/612192/654829-4.html
- Static & Dynamic Testing - https://www.synopsys.com/software-integrity/security-testing.html
- Software Testing Principles- https://www.lynda.com/Project-Management-tutorials/Project-Management-Foundations-Quality/159186-2.html
- Exhaustive testing - http://ousard.blogspot.com/2011/08/searching-searching-man.html
- Defect clustering - https://clipartxtras.com/download/6fb75010620df7a33d8a1bad180827f364ec23b2.html
- Pesticide paradox - https://www.vecteezy.com/vector-art/102558-pest-control-supply-vectors
- Absence of error - https://promoboxx.com/compare/why-dam-systems-dont-work
- Testing context dependent - http://www.subject-7.com/learn-more/
- Early testing - https://www.lynda.com/Business-Software-tutorials/Managing-Do-Lists/365217-2.html
- Presence of defects - https://www.123rf.com/photo_10792282_software-bug-hidden-bug-among-binary-codes.html
- Context dependent - https://www.visma.com/brandbook/visual-identity-imagery-illustrations2/#undefined
- Fundamental Test Process - https://websupportnow.com/product/premium-professional-wordpress-maintenance-plan/
- Planning & control - http://www.iconarchive.com/tag/control-panel
- Analysis & design - https://www.iconfinder.com/icons/379472/graph_magnifier_icon
- Implementation & execution - http://www.p2energysolutions.com/services/implementation
- Evaluating criteria - http://www.iconarchive.com/show/colorful-long-shadow-icons-by-graphicloads/Hand-thumbs-up-like-2-icon.html
- Test closure - https://freeiconshop.com/icon/agreement-icon-flat/
- Psychology of testing - http://www.all-about-psychology.com/psychology-ebook.html
- Who's testing - http://freedesignfile.com/132535-different-people-pattern-seamless-vector-01/
- Developer vs. QA - https://www.utest.com/articles/cat-and-mouse-qa-tester-vs-developer-funny-comments
- Testing and Communication - https://sominotes.com/2017/09/01/ect-300-educational-technology-what-is-the-communication-process-and-learning/
- Questions - http://indsoft.com/comments.html
- Time to get tested - https://communicatehealth.com/2015/09/testing-techniques-part-3-testing-print-materials/

First Nations
Technology Council