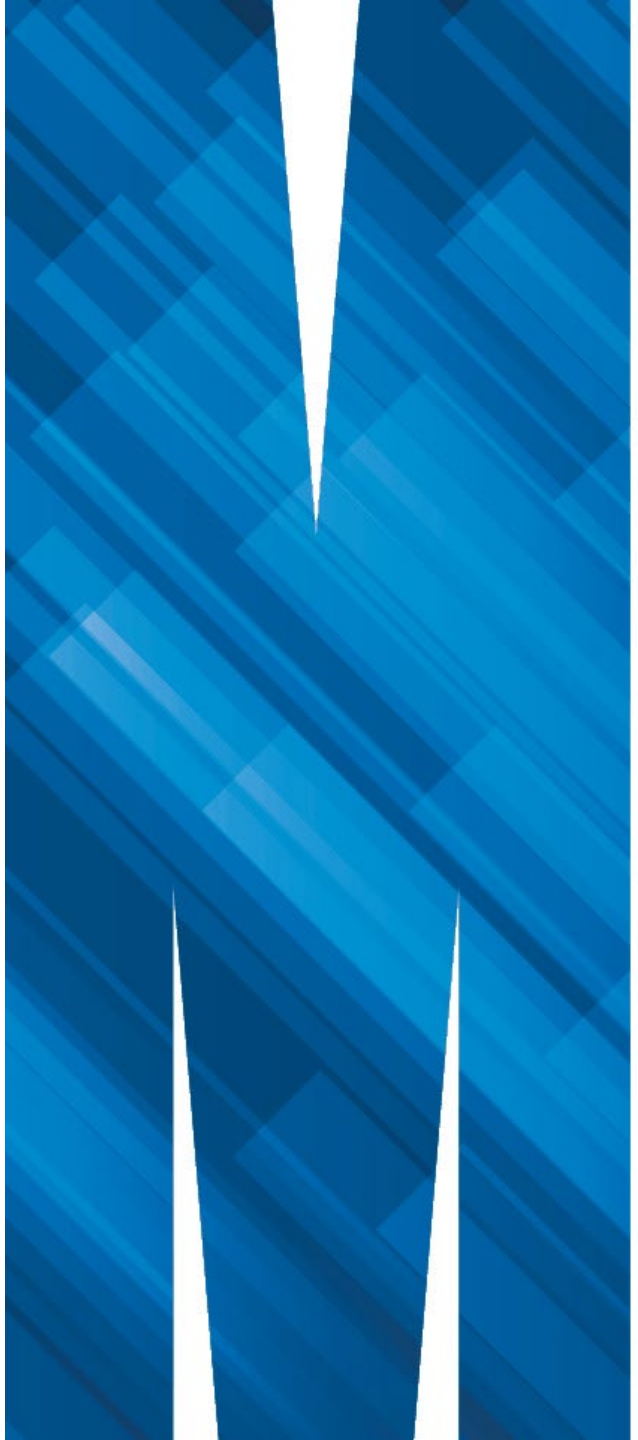
MONASH University

FIT2107
Software Quality and Testing

Lecture 1–Introduction to Software Quality Assurance

- Introduction to the Unit
- Why Software Quality is needed?
- What is Software Quality?
- What are Software Quality Attributes?
- What is Software Quality Assurance?
- Limitations of SQA.

Dr Najam Nazar (Clayton)
Dr Lillian Wang (Malaysia)



Pre Class Activity

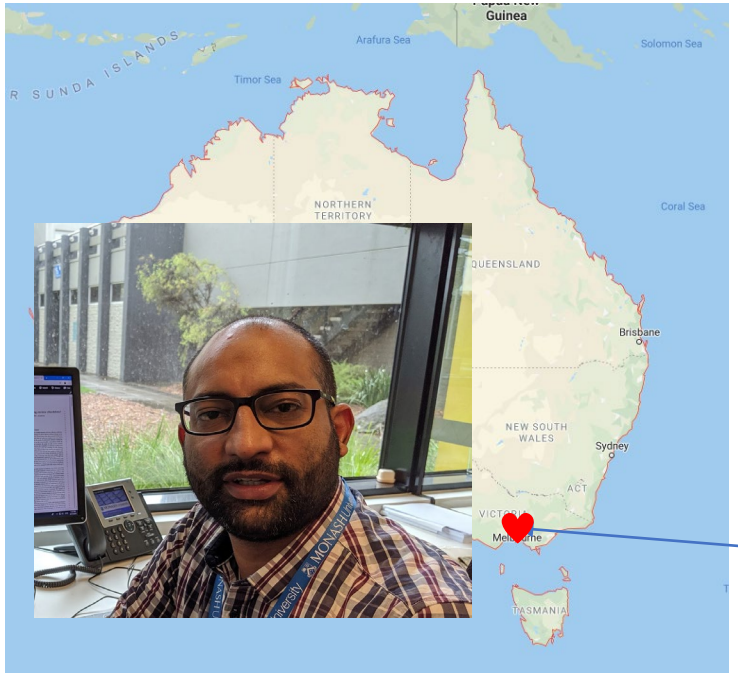
- Fill the following link before attending the class
  - https://forms.gle/qirn6YFokai64GpM9

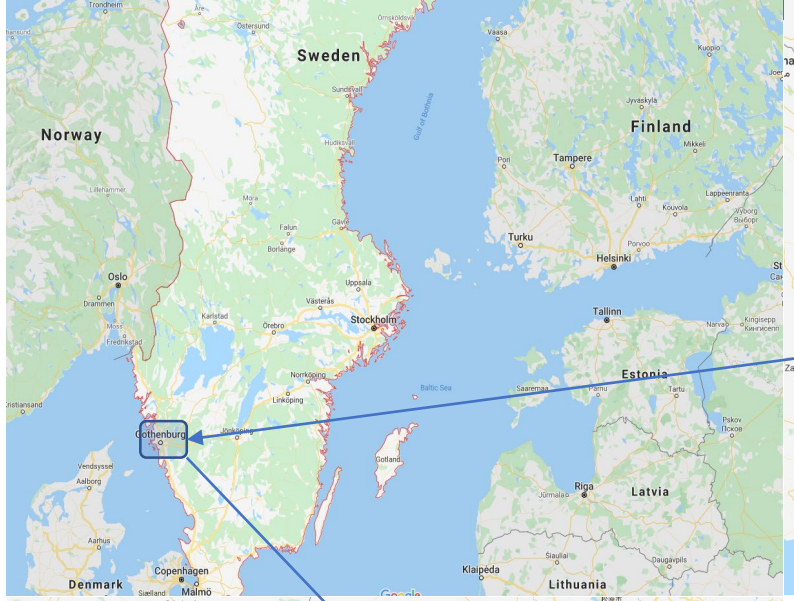
We are here to support you

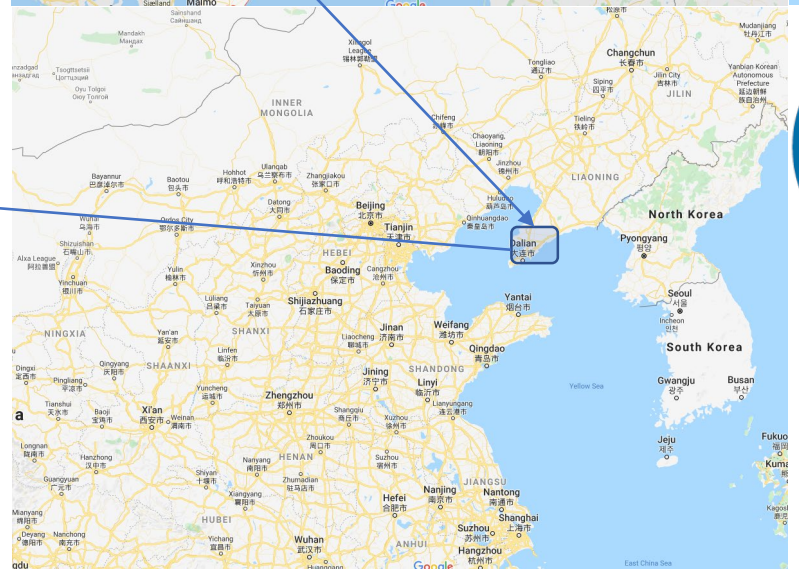
- 2020 is a very challenging year for all of us, and we understand that it is very challenging semester for you
- Even when this situation is a challenge, this challenge is very important for your future career – remote work experience might be a crucial skill for many jobs
- We adjusted all assessment tasks to deal with this situation, we adjusting our teaching to this too, because we would like to help you all to achieve best
- You came form different backgrounds and countries, some of you are now in Melbourne / Australia, some of you are overseas – but we all will work remotely this semester
- I myself wasn't born in here, but I came to call Australia home and joined Monash 3 years ago

MONASH University

Who am I?







Dr Najam Nazar
Lecturer @ Clayton
PhD in Software Engineering

Lecturer and Tutor in Software Engineering units such as FIT3077, FIT2099, FIT5136 etc
Active researcher in Software Engineering in the field of MSR

MONASH University

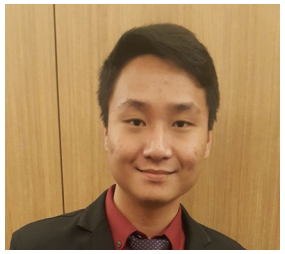
TEACHING STAFF

Dr Lillian Wang
Lecturer @ Malaysia
Professional technologist

Dr Chakkrit Tantithamthavorn
Chief Examiner
ARC DECRA Fellow
Active researcher in Empirical Software Engineering & Mining Software Repository

Ammar Sohail
Tutor

Chamath Divarathne
Tutor



Norman Chen
Tutor

Silpi Dutta
Tutor

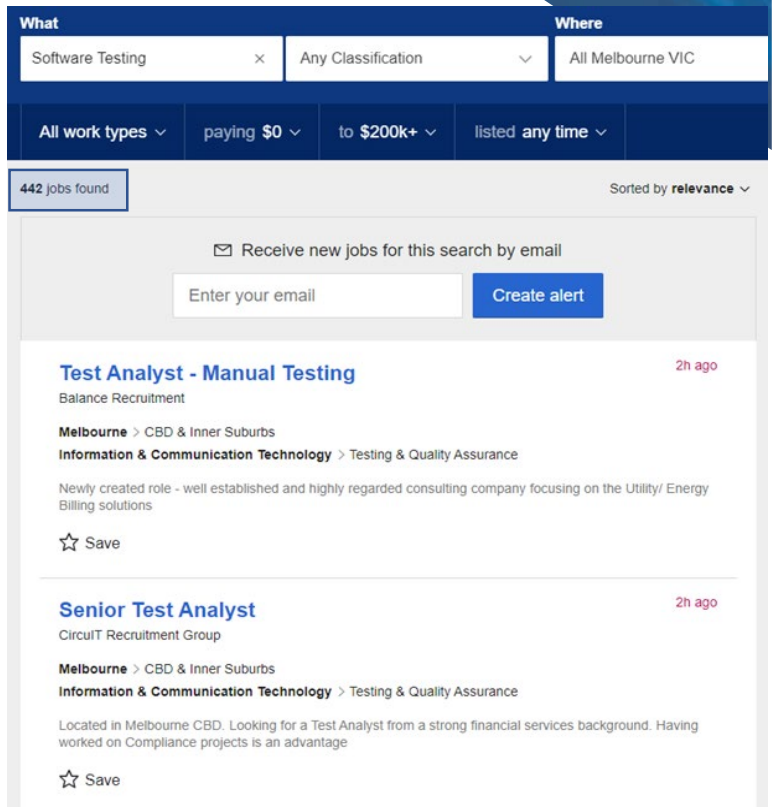
Ting Ting Bi
Tutor

Email: firstname.lastname@monash.edu

MONASH University

Why are we doing this course?

- IT graduates think testing doesn't require any programming skills
  - What if I don't like writing documents? This course sounds tedious and …%^&&%$@….
- I am excellent in programming
  - Hmm….I don't want to be tester.
- IT industry is very dynamic and rapidly changing.
  - Software Quality is critical.
  - Software testing is essential.
- Being software engineer, knowing testing is a critical skill
- …and coding is mandatory for this unit
  - Python
- Many Industrial Jobs in testing



MONASH University

Think about your future career

- Take care about yourself and each other ☺
- Even when we do not have the course on Campus or online, use the lecture, tutorial and practical sessions for networking.
- You are in charge of your own work, including deadlines and academic integrity.
- Be resilient, don’t give up and if you have an issue try to think not only about how to solve it but also how to use this as an opportunity to learn
- Your tutor is a first line of contact and if required will involve Head Tutor or Lecturer.

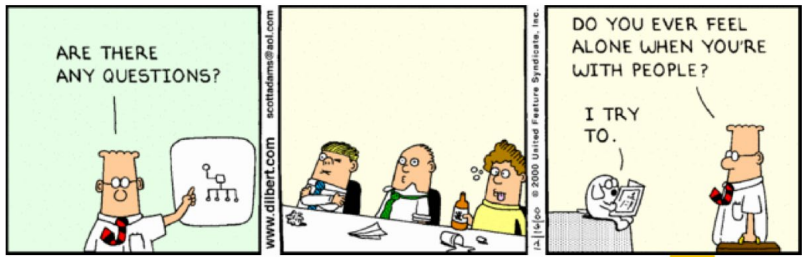
MONASH University

Where & When & How?

- Lecture: general ideas + “big picture”+ theory + examples + inspiration + pre-recorded + online.
- Workshops: more examples + more practice + past exams + online.
- Assignments: real problems, assessed, deadline.
- Quizzes: true/false, mcq, short answers -> revision.
- 1hr lecture + 2 hr workshop.
  - May not enough
  - Require extra effort
- Give your tutor or me your feedback early.
- Written Course Notes (Mandatory)
  - Reading Material

Questions + Emails

- Ask
- Ask Early
- Ask preferably during lectures, tutorials and consultations.
- Use Moodle discussion forums -> always encouraged.
- Emails
  - Always put FIT2107-S2-2020 at the beginning of the subject
  - No subject emails will be ignored….
  - Use your official Monash email to contact
  - Use your full name in the salutation area (at the end of the text)
- I'll regularly make announcements on forum
  - Check regularly.
- I'll regularly send emails
  - I didn't check my email
  - Oh! it went into spam/junk…
  - 😟 Not acceptable



MONASH University

Course Changes 2020

- Based on the Industry trend as well as on feedback from students this course is revised each year
- Key Changes
  - More focus on Software Testing
  - More programming
  - Code Reviews are introduced
  - Quizzes are updated
- Lecture slides + reading material + workshops released weekly on Moodle.
- Assignments will be released two weeks prior to the start date.

Unit Preview + Schedule

Watch my videos in week 0 on Moodle about the unit preview and schedule.

Week	Topic	Workshop	Assessment
1	Software Quality - Introduction	Quality Attributes	
2	Introduction to Software Testing	Software Testing	
3	Black-Box Testing - I	Black-Box Testing - I	
4	Black-Box Testing - II	Black-Box Testing - I	
5	White-Box Testing - I		Assignment 1
6	White-Box Testing - II	White-Box testing - I	
7	Unit Testing	White-Box Testing - II	
8	Gitlab + Continuous Integration	Unit Testing + Mocking	
9	Mocking		Assignment 2
10	Inspections, Walkthrough, Code Reviews	Code review	
11	QA Metrics	Python metrics	
12	Review		Assignment 3

Unit Assessments

Watch my videos in week 0 on Moodle about the unit preview and schedule.

Assessment task	Value	Due date
Blackbox Testing	15%	Submission in week 5
White Box & Unit Testing	20%	Submission in week 9
Code Review and Critique	15%	Submission in week 12
Quizzes	10%	Fortnightly
Exam (2 hours)	40%	To be advised

Assignments teams

- Working in teams
  - Pairs in most cases
  - Or odd numbers discuss with tutor
- Registration
  - On Moodle
  - No later than week 2
- Team relocation requests may be considered?
  - Depends on the situation.
- Conflicts & Issues
    - Notify your tutor asap.

MONASH University

Assignments Marks

- Each student will a receive final mark based on a final mark for assignment and their contribution.
  - Bonus and penalties may apply.
- WBA or contribution is a hurdle for passing the assignments.
- Marking will be done on an external tool
  - MADAM
  - Also move to Moodle for record keeping.
- Marking Feedback
  - Will be provided in a two weeks time
  - Delays may happen
- Issues
  - Discuss with tutor

PART - 2

SOFTWARE QUALITY

Your PC ran into a problem and needs to restart. We're just collecting some error info, and then we'll restart for you. (0% complete)

If you'd like to know more, you can search online later for this error: HAL_INITIALIZATION_FAILED

Queensland Health Payroll

- In 2009 Queensland government attempted to replace to the pyroll system
- The process didn't go well.
- Queensland Health's 85,000 workers went without pay, or were overpaid
- The project of $384 million cost over $1.25 billion
- An audit found that the project's difficulties were caused by
  - woeful project scope definition
  - poor governance
- The inquiry commission by Supreme court further revealed
  - unwarranted urgency
  - lack of diligence.
- Results?
  - The incumbent government lost the election due to this scandal.
  - IBM had to pay millions in legal battle.

https://delimiter.com.au/2013/08/07/anatomy-of-qld-health-it-disaster%E2%80%A8-ibm-should-never-have-been-appointed/

Malaysian banks

- On 8th April 2014, Microsoft announced that they will stop supporting Windows XP
- 90% of Malaysia's ATM were running on Windows XP.
- Although being notified a potential exploit on XP, 3 local banks choose not to do anything about it because the old ATM software cannot be run on Windows 7/8.
- They will need to spend extra money on new hardware and software
- The software developers didn't bother to test the system
- Results?
    - A group of hackers got away with RM3million (~1 million AUD)

http://www.themalaymailonline.com/malaysia/article/banks-knew-atms-were-open-to-attacks-source-claims
http://www.financetwitter.com/2014/09/here-is-how-malaysian-atms-were-hacked-of-rm3-million-by-latin-americans.html

MADAM Marking Tool

- Prototype marking report tool
- Used in many SE units
- Built in a terrible hurry.
- Had a “bug” – took 30 seconds to send reports, was causing timeouts
- Code was functionally correct
- Algorithm - search entire directory path to send each report.
- Tested? Well as made in a hurry, no adequate testing done.
- Result?
    - Delayed marking feedback.

What is Software Quality?

- Quality means the degree to which a product or a process meets requirements (functional quality).
- Supports the delivery of functionality requirements such as maintenance, robustness etc. (non-functional quality).
- What makes software quality so hard?
  - Software systems are growing in **complexity**: Modern systems are composed of millions of lines of code!
  - Some **quality requirements** are difficult to specify in an unambiguous way
  - Software systems are **intangible**: Our senses cannot help us understand them
  - Software systems are **malleable**: Small changes can have huge repercussions
  - **Limited human resources** for finding defects: Window for finding/fixing defects is small
- The focus may be ‘fitness for purpose’ rather than specification conformance.

Software Fitness for Purpose

- Has the software been properly tested?
- Is the software sufficiently dependable to be put into use?
- Is the performance of the software acceptable for normal use?
- Is the software usable?
- Is the software well-structured and understandable?
- Have programming and documentation standards been followed in the development process?
- Good quality software must answer **YES** to the above questions!!!

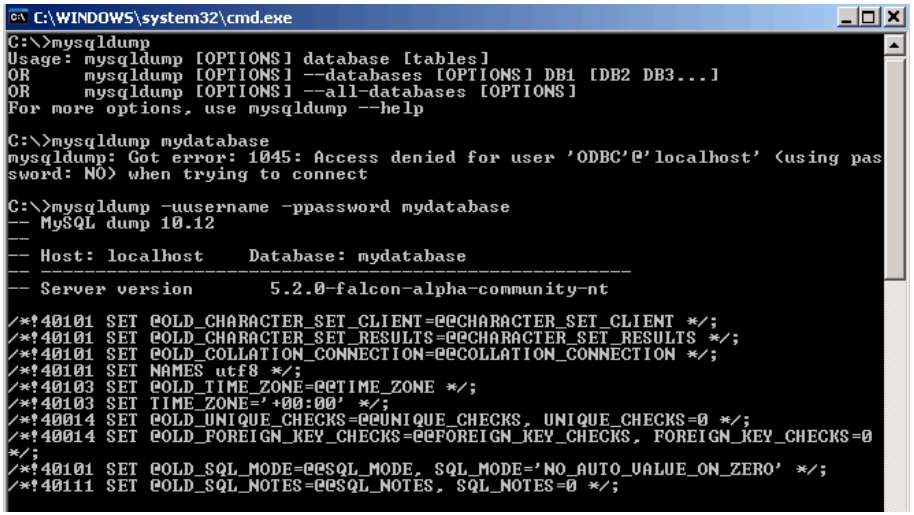
Software Quality Attributes (ISO 25010)

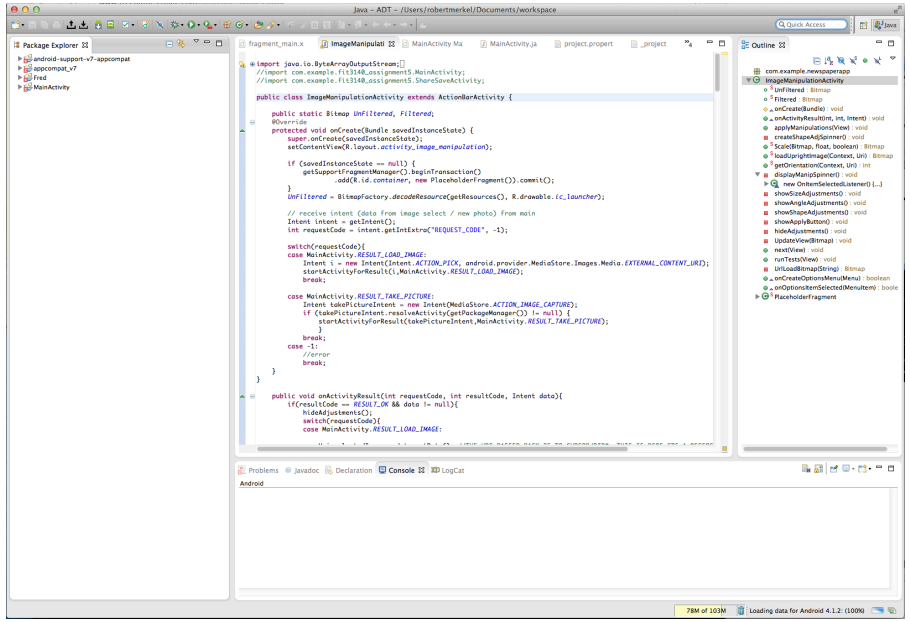
Product Quality							
Functional Suitability	**Reliability**	**Performance Efficiency**	**Usability**	**Maintainability**	**Security**	**Compatibility**	**Portability**
Functional completeness	Maturity	Time behaviour	Appropriateness recognisability	Modularity	Confidentiality	Co-existence	Adaptability
Functional correctness	Availability	Resource utilization	Learnability	Reusability	Integrity	Interoperability	Installability
Functional appropriateness	Fault tolerance	Capacity	Operability	Analysability	Non-repudiation		Replaceability
	Recoverability		User error protection	Modifiability	Accountability		
			User interface aesthetics	Testability	Authenticity		
			Accessibility				

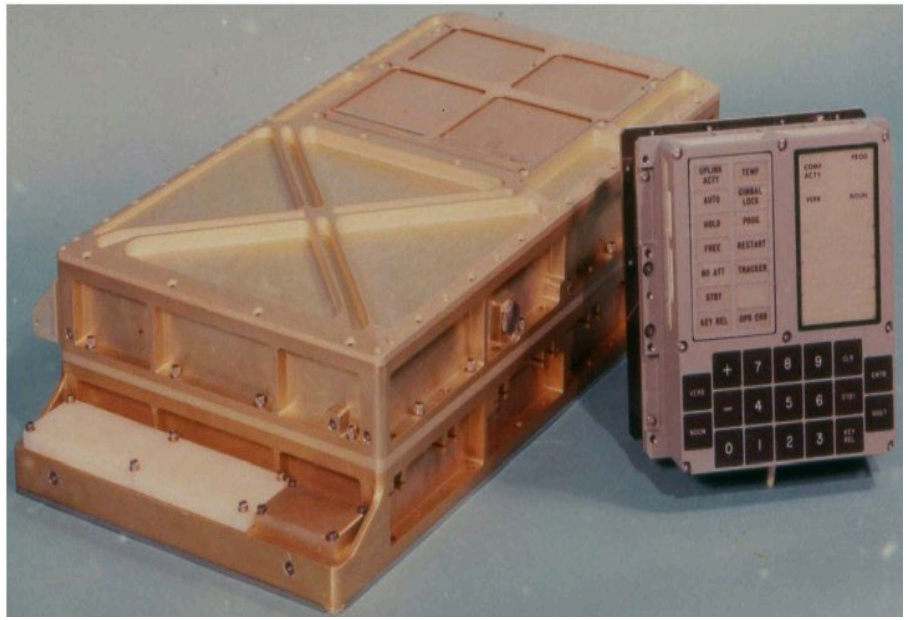
Source: L. Garcés, F. Oquendo and E. Y. Nakagawa, "A Quality Model for AAL Software Systems," 2016 IEEE 29th International Symposium on Computer-Based Medical Systems (CBMS), Dublin, 2016, pp. 175-180, doi: 10.1109/CBMS.2016.46.

MONASH University

Software Quality Attributes - Examples







MONASH University

What is Software Quality Assurance?

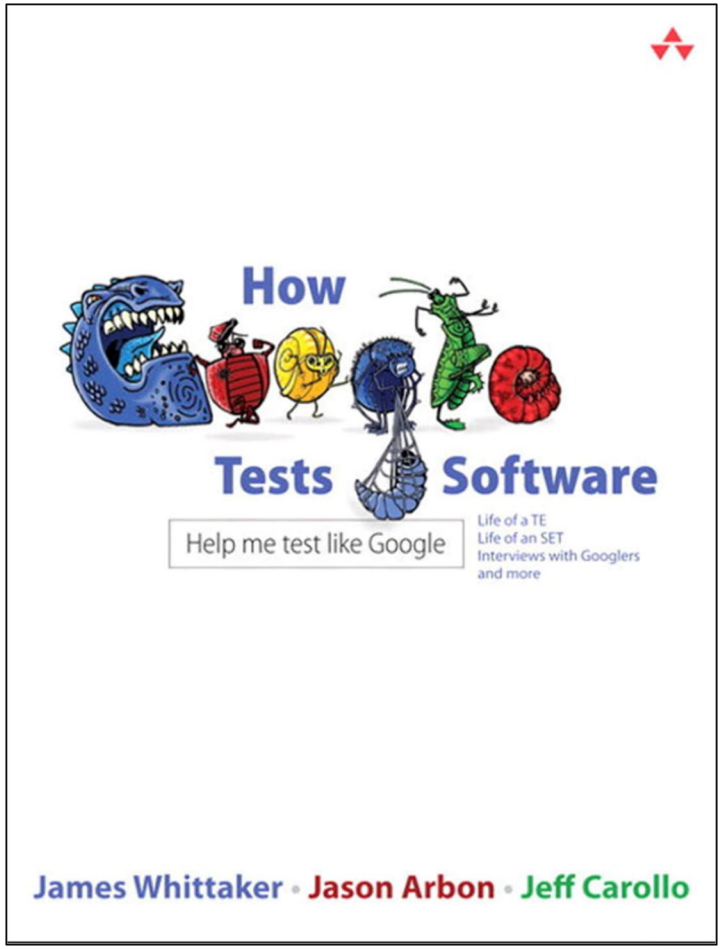
- A set of activities that define and assess the adequacy of software processes to provide evidence that establishes confidence that the software processes are appropriate for and produce software products of **suitable quality** for their intended purposes. (IEEE)

- A planned and systematic pattern of all actions necessary to provide adequate confidence that an item or **product conforms to adequate quality.**

- A set of activities designed to **evaluate the process** by which the products are developed or manufactured.

How Google Tests Software?

"Quality is more an act of prevention than it is detection"



https://www.amazon.com/Google-Tests-Software-James-Whittaker/dp/0321803027

Software Engineer (SWE)

- Write code/test code (including test-driven design)
- Write unit tests
- Engineers own quality for everything they touch

Software Engineer in Test (SET)

- Review designs
- Check code quality and risk
- Refactor
- Partner with SWE aiming to increase quality and test coverage, rather than adding new features

Test Engineer (TE)

- Write test automation scripts
- Interpret test results
- Drive test execution
- Product experts
- Quality advisers
- Risk analyzers

MONASH University

Limits of SQA

- No SQA technique can remove all risk
  - Blame Alan Turing
- In practice, reducing risk to super low levels is very costly
  - NICTA verified microkernel
  - 1 LOC per programmer/month!
- Need to balance costs of SQA with magnitude of risk

Software Verification & Validation

- Validation: Are we building the right product?
- Verification: Are we building the product right?

- What artefacts we QA?
  - Executable artefacts
  - Formal Artefacts
  - Informal Artefacts

Closing Notes!

- Python will be mainly used for FIT2107
  - assignments, workshops, exams, etc.
- **Warning:** This is not a Python Unit but require basic knowledge of Python language.
- If you don’t have much experience in Python
  - It’s better to start learning Python NOW!!!!
- **Expectation:** By the end of Week 1, all students must be able to program and compile Python scripts on an IDE (e.g., PyCharm or VS Code or command line).

Summary

- This unit has three assignments, 5 quizzes, and an exam.
- Python will be mainly used for FIT2107.
- We focus on functional quality and focus on quality attributes.
- Validation: Are we building the right product?
- Verification: Are we building the product right?

Questions??


YOU SCHEDULED THE END OF THE TEST PHASE AFTER THE START OF THE PRODUCTION PHASE.
WE'RE FEELING CONFIDENT.
IT'S TOO BAD THAT BEING SMART DOESN'T COME WITH SOME SORT OF GOOD FEELING LIKE THAT.
Dilbert.com DilbertCartoonist@gmail.com
5-17-11 © 2011 Scott Adams, Inc. Dist. by Universal Uclick