



Australian Government

Department of Defence

Defence Materiel Organisation

DEFENCE MATERIEL ORGANISATION

AEROSPACE SYSTEMS DIVISION

Software Testing: How Much Is Enough?

Finding The Balance Between Cost, Quality and Assurance

FLGOFF Ryan Lawler





Overview of the Brief

- Background
 - What is Software Testing
 - Why do we test software
 - Who tests software
- Testing for Quality
 - How do we measure software quality
 - Types of software testing
 - Combinations and Permutations
 - Example: Unit Testing





Overview of the Brief

- The Concept of Software Assurance
 - What is software assurance
 - Software assurance and the aviation industry
 - The TAMM and DO-178B
 - Example: Unit Testing
- The cost of Software Testing
 - How much does it cost
 - Why does it cost so much
 - What methods are out there to reduce costs





Australian Government

Department of Defence

Defence Materiel Organisation

DEFENCE MATERIEL ORGANISATION

AEROSPACE SYSTEMS DIVISION

Overview of the Brief

- The Cost Constrained Environment
 - The elements that make up this environment
 - The unique challenges to be faced
 - The affects on software quality
 - The affects on software assurance
 - Finding an appropriate balance





Australian Government

Department of Defence

Defence Materiel Organisation

DEFENCE MATERIEL ORGANISATION

AEROSPACE SYSTEMS DIVISION

References

- Policy:
 - DI(G) OPS 02-2 ADF Airworthiness Management
 - AAP 7001.053 (AM1) Technical Airworthiness Management Manual
- Guidance:
 - Practical Guide to Software Systems Testing
 - DO-178B





Australian Government

Department of Defence

Defence Materiel Organisation

DEFENCE MATERIEL ORGANISATION

AEROSPACE SYSTEMS DIVISION





What Is Software Testing?

- The process of trying to break a software product:
 - Systematic examination of the product.
 - Assessment of functionality against requirements
 - Detection of defects and non-conformances
 - Identification of unexpected behaviour.
- Testers should be trying to break the product
- A product can conform to all requirement and standards, but still contain visible and hidden defects that affect the performance of the system





Australian Government

Department of Defence

Defence Materiel Organisation

DEFENCE MATERIEL ORGANISATION

AEROSPACE SYSTEMS DIVISION

Why Do We Test Software





Australian Government

Department of Defence

Defence Materiel Organisation

DEFENCE MATERIEL ORGANISATION

AEROSPACE SYSTEMS DIVISION

Why Do We Test Software

- To establish how the product will function under specific conditions
- To observe, measure and document the performance of a system.
- To provide assurance that the product meets customer requirements.





Who Tests Software

- Development Engineers
 - Unit Testing
- Definition & Test Engineers
 - System and Performance Testing
- Customer
 - Operational Test & Evaluation
 - Acceptance Testing





Australian Government

Department of Defence

Defence Materiel Organisation

DEFENCE MATERIEL ORGANISATION

AEROSPACE SYSTEMS DIVISION

Software Quality





Australian Government

Department of Defence

Defence Materiel Organisation

DEFENCE MATERIEL ORGANISATION

AEROSPACE SYSTEMS DIVISION

How Do We Measure Software Quality

- Defects discovered per line of code
- Functionality vs. Requirements
- Performance parameters and efficiency
 - CPU Usage, Memory Usage
- Compiler Integrity
 - Consistent memory allocation





Combinations and Permutations

- Testing every possible value for each variable.
- Evaluating condition and decision boxes
 - ie. If $(A > 5 \text{ and } B < 4)$ or $A = 10$
- Coverage of every path
- Integration with the rest of the system and every other system





Australian Government

Department of Defence

Defence Materiel Organisation

DEFENCE MATERIEL ORGANISATION

AEROSPACE SYSTEMS DIVISION

Types of Software Testing

- Unit Testing
- Integration Testing
- System Testing
- System Performance, Integrity and Stability Testing





Australian Government

Department of Defence

Defence Materiel Organisation

DEFENCE MATERIEL ORGANISATION

AEROSPACE SYSTEMS DIVISION

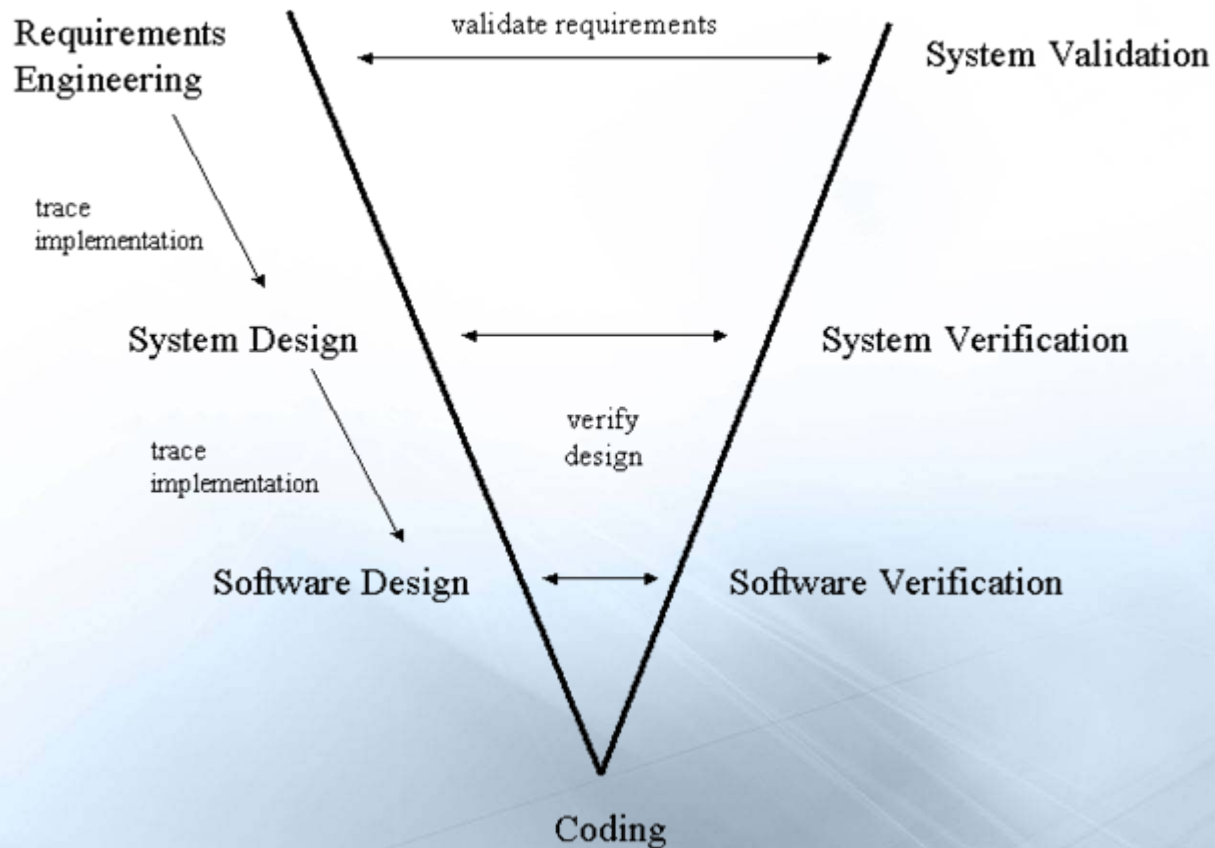
Types of Software Testing

- Regression Testing
- Operational Test & Evaluation
- Acceptance Testing





Software Testing Methodology





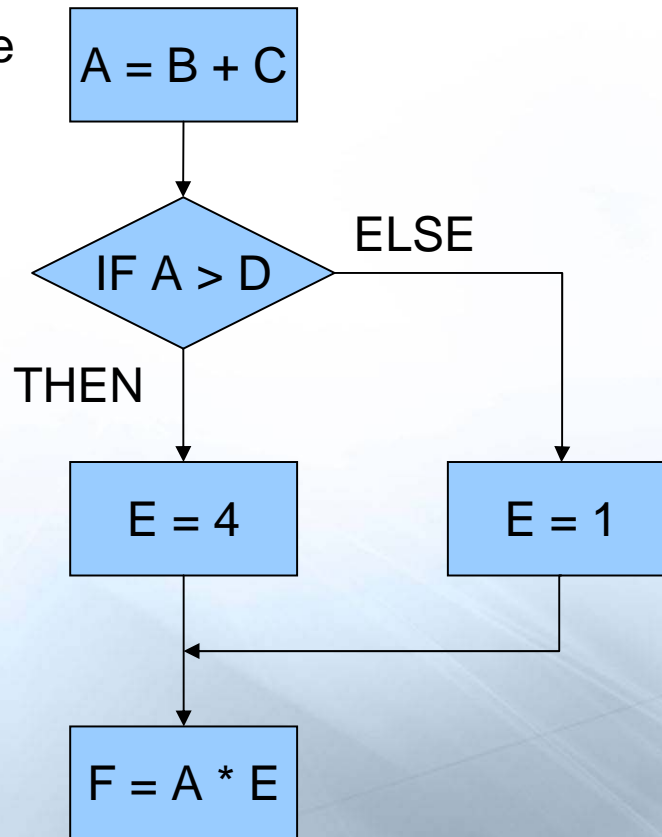
How Do We Test – Unit Testing

Proposed design change

Could be implemented using approx 8 lines of code

Static testing:

Inspection, review of source code against design



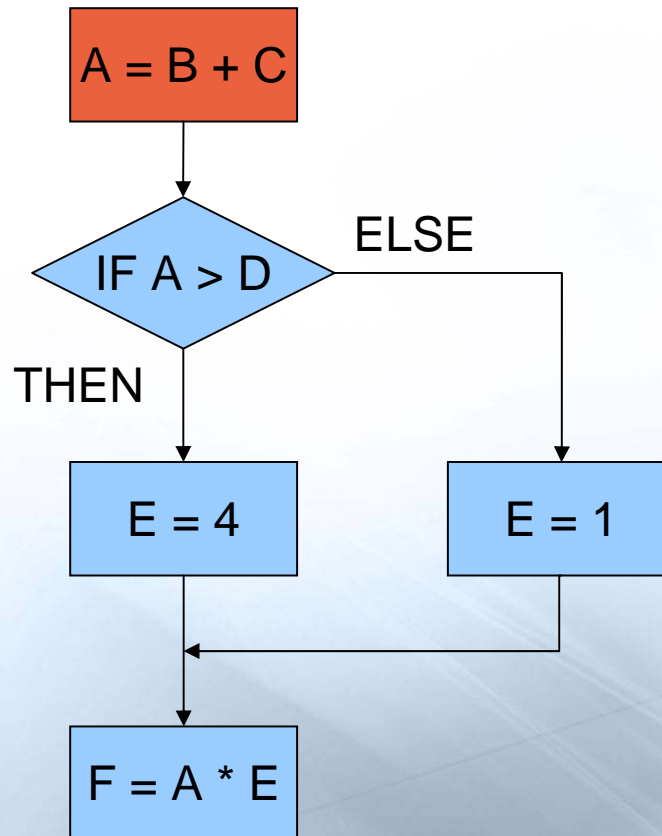


How Do We Test – Unit Testing

TC1: Verify A given values for B and C

Need to ensure you verify an addition has taken place.

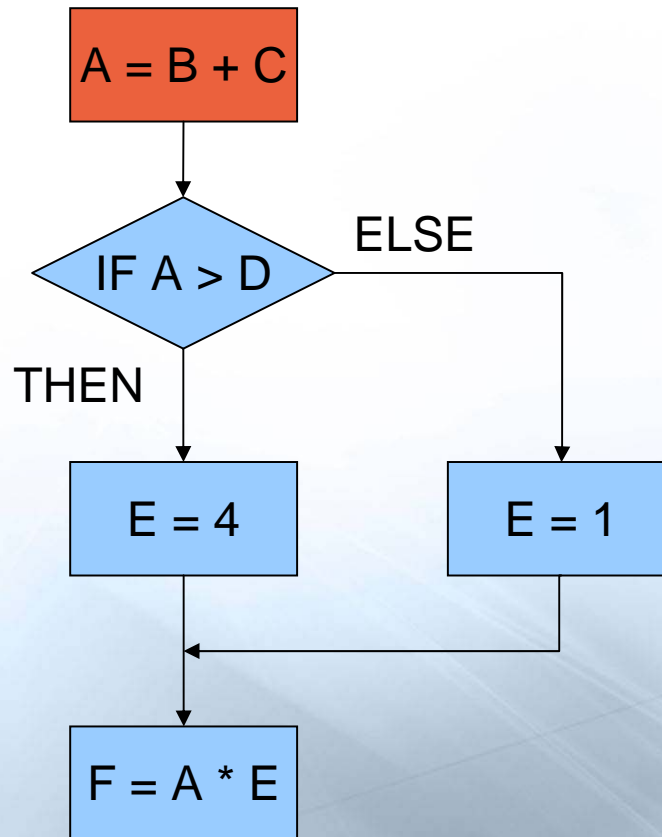
Given B,C = 2
A = B + C, or
A = B * C





How Do We Test – Unit Testing

TC2: Verify A does not exceed its max/min limits given max/min values of B and C



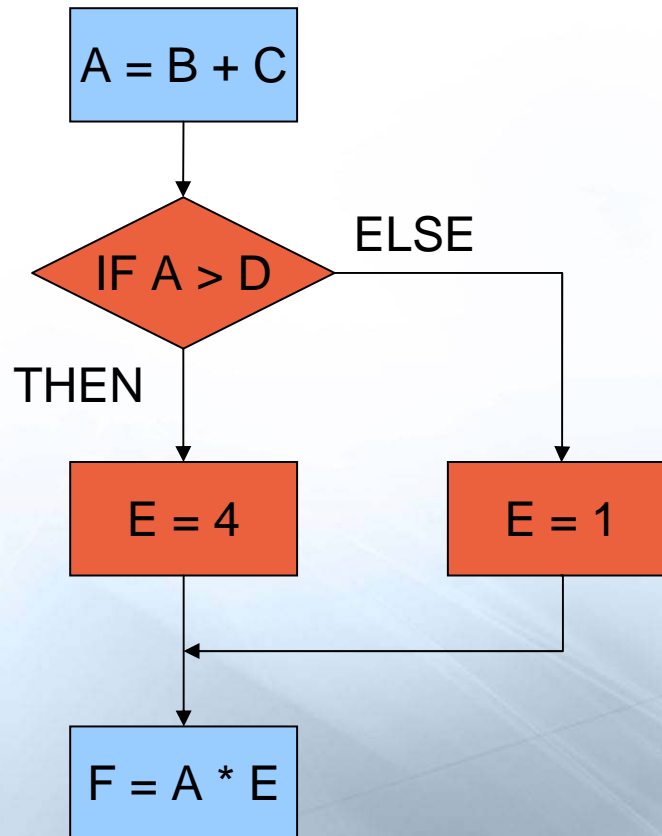


How Do We Test – Unit Testing

TC3: Verify that $E = 4$ if $A > D$

TC4: Verify that $E = 1$ if $A = D$

TC5: Verify that $E = 1$ if $A < D$





Australian Government

Department of Defence

Defence Materiel Organisation

DEFENCE MATERIEL ORGANISATION

AEROSPACE SYSTEMS DIVISION

Software Assurance

Assurance

Don't worry, it'll be fine.
Trust me





What is Software Assurance

- Is a measure of confidence that software should function correctly.
- Is based on the amount of testing, type of testing, and the rigour applied.
- Uses logic to remove duplication of testing





Australian Government

Department of Defence

Defence Materiel Organisation

DEFENCE MATERIEL ORGANISATION

AEROSPACE SYSTEMS DIVISION

What is Software Assurance

- We cannot say for certain that a product functions correctly under all conditions.
- For complex software, you cannot test every single combination and permutation
- Compliance with an assurance level is not a guarantee





Australian Government

Department of Defence

Defence Materiel Organisation

DEFENCE MATERIEL ORGANISATION

AEROSPACE SYSTEMS DIVISION

Software Assurance in Aviation

- Aviation is a regulated industry
- The consequences can be extreme if you get it wrong
- Software must be designed, implemented and tested to meet accepted assurance levels





TAMM – Regulation 3.5.3

- Software Integrity Management System
 - Configuration management of Software CI's
 - Software Tool Identification
 - Development of safety-related software to satisfy TAR recognised software assurance standard / Software Assurance Matrix
 - Software load control
- The focus is on knowing what software we are using, where it is used, and having assurance that an aircraft is safe to fly using that software





DO-178B – Software Assurance

- Is an internationally accepted guidance produced by RTCA
- It is not a standard, it cannot guarantee quality
- Uses the concept of Software Assurance Levels
- Provides guidance for assessing compliance with each software level
- Is the guidance of choice for Software Assurance in Aviation





Example – Unit Testing

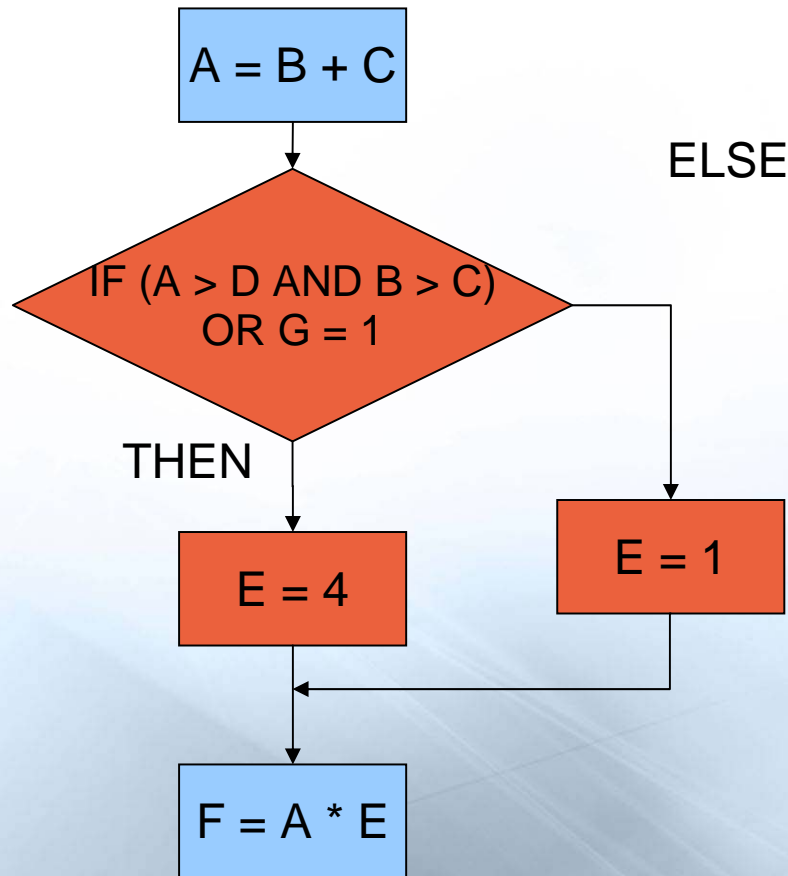
TC: Verify $E = 4$ when
 $A > D, B > C, G = 1$

TC: Verify $E = 4$ when
 $A > D, B < C, G = 1$

TC: Verify $E = 4$ when
 $A < D, B > C, G = 1$

TC: Verify $E = 4$ when
 $A < D, B < C, G = 1$

Etc.





Example – Unit Testing

| | X | Y | Z | Result |
|---|---|---|---|--------|
| 1 | 0 | 0 | 0 | E |
| 2 | 0 | 1 | 0 | E |
| 3 | 0 | 0 | 1 | T |
| 4 | 0 | 1 | 1 | T |
| 5 | 1 | 0 | 0 | E |
| 6 | 1 | 1 | 0 | T |
| 7 | 1 | 0 | 1 | T |
| 8 | 1 | 1 | 1 | T |

$X = 1$ when $A > D$

$Y = 1$ when $B > C$

$Z = 1$ when $G = 1$





Modified Condition/Decision Coverage

| | X | Y | Z | Result | X | Y | Z |
|---|---|---|---|--------|---|---|---|
| 1 | 0 | 0 | 0 | E | | | 3 |
| 2 | 0 | 1 | 0 | E | 6 | | 4 |
| 3 | 0 | 0 | 1 | T | | | 1 |
| 4 | 0 | 1 | 1 | T | | | 2 |
| 5 | 1 | 0 | 0 | E | | 6 | 7 |
| 6 | 1 | 1 | 0 | T | 2 | 5 | |
| 7 | 1 | 0 | 1 | T | | | 5 |
| 8 | 1 | 1 | 1 | T | | | |





Modified Condition/Decision Coverage

| | X | Y | Z | Result | X | Y | Z |
|---|---|---|---|--------|---|---|---|
| 1 | 0 | 0 | 0 | E | | | 3 |
| 2 | 0 | 1 | 0 | E | 6 | | 4 |
| 3 | 0 | 0 | 1 | T | | | 1 |
| 4 | 0 | 1 | 1 | T | | | 2 |
| 5 | 1 | 0 | 0 | E | | 6 | 7 |
| 6 | 1 | 1 | 0 | T | 2 | 5 | |
| 7 | 1 | 0 | 1 | T | | | 5 |
| 8 | 1 | 1 | 1 | T | | | |

Independent X's: 6 and 2

Independent Y's: 6 and 5

Independent Z's: 1 and 3, 2 and 4, 6 and 5

Only 4 Tests Required: 2,4,5,6





Modified Condition/Decision Coverage

| | X | Y | Z | Result | X | Y | Z |
|---|---|---|---|--------|---|---|---|
| 1 | 0 | 0 | 0 | E | | | 3 |
| 2 | 0 | 1 | 0 | E | 6 | | 4 |
| 3 | 0 | 0 | 1 | T | | | 1 |
| 4 | 0 | 1 | 1 | T | | | 2 |
| 5 | 1 | 0 | 0 | E | | 6 | 7 |
| 6 | 1 | 1 | 0 | T | 2 | 5 | |
| 7 | 1 | 0 | 1 | T | | | 5 |
| 8 | 1 | 1 | 1 | T | | | |

Only 4 Tests Required: 2,4,5,6





Australian Government

Department of Defence

Defence Materiel Organisation

DEFENCE MATERIEL ORGANISATION

AEROSPACE SYSTEMS DIVISION

The Cost of Software Testing





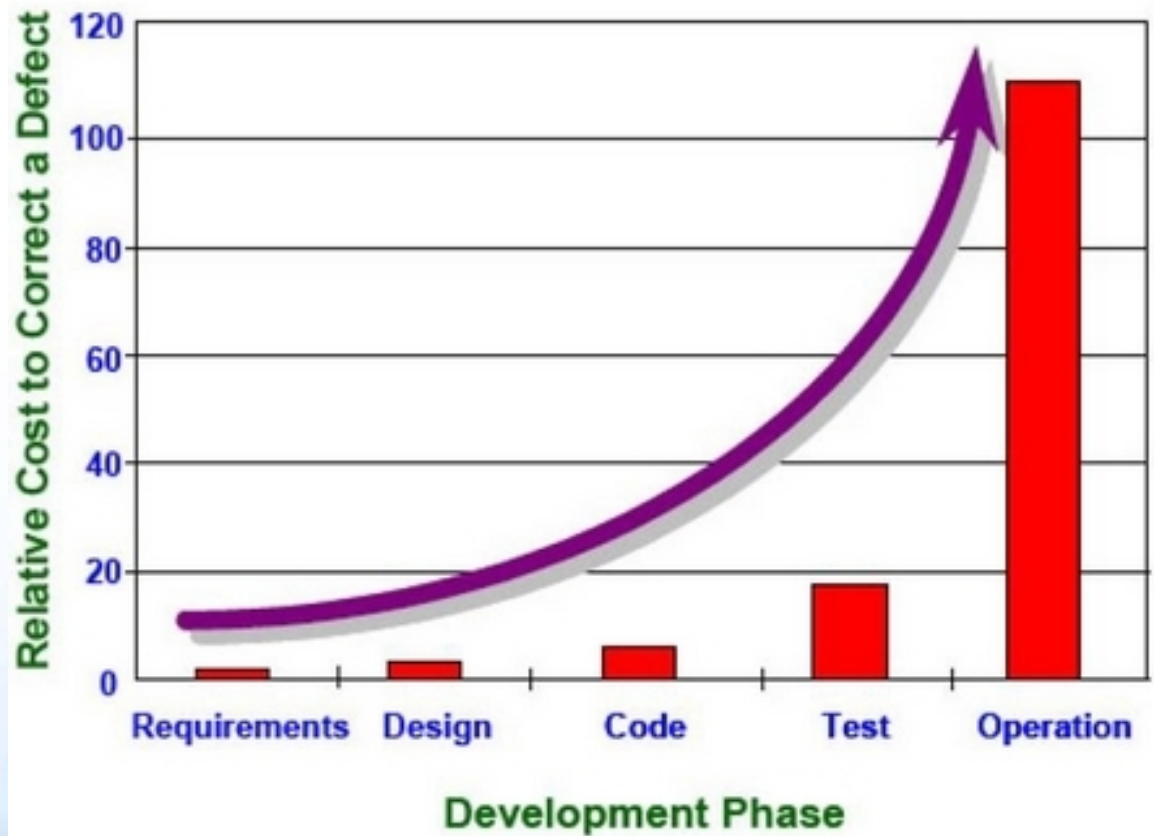
How much does testing cost?

- The cost of testing is proportional to the rigour of testing applied.
- The cost of testing is inversely proportional to the defects found.
- If few defects are found during testing, the cost may be more than half your budget
- If many defects are found during testing, the cost of reworking the design will overshadow the cost of testing.
- Defects found during the in-service phase will require the initiation of a new development project in order to be resolved.





The Benefits of Software Testing





Australian Government

Department of Defence

Defence Materiel Organisation

DEFENCE MATERIEL ORGANISATION

AEROSPACE SYSTEMS DIVISION

Why Does it Cost So Much?

- Development of Test Cases
- Performance of Tests
- Documentation of Results
- Resolution of anomalies
- Review of Test Cases, Performance, and Documentation
- Witnessing activities
- Done for all products at all levels





Australian Government

Department of Defence

Defence Materiel Organisation

DEFENCE MATERIEL ORGANISATION

AEROSPACE SYSTEMS DIVISION

How Can we Reduce That Cost?

- Amount of testing
- Type of testing
- Level of rigour
- Customer testing during development
- Acceptance of defects





The Cost Constrained Environment





Australian Government

Department of Defence

Defence Materiel Organisation

DEFENCE MATERIEL ORGANISATION

AEROSPACE SYSTEMS DIVISION

The Elements of a Cost Constrained Environment

- Poor Estimation
- Poor Requirements
- Budget / Schedule Recovery
- Strategic Reform Program





Australian Government

Department of Defence

Defence Materiel Organisation

DEFENCE MATERIEL ORGANISATION

AEROSPACE SYSTEMS DIVISION

The Unique Challenges

- Testing can make up half the project budget
- Being such a high cost makes it a target for cost reduction activities
- The decision to cut costs is forced upon the project
- There is push back because regulations must still be met





The Unique Challenges

- Testing become a highly visible activity
- Increased pressure on testers to get through test activities as quick as possible with as few errors as possible
- Pressure to accept defects increases as development cycle progresses.
- Automated testing reduces cost, but also reduces need for personnel





Australian Government

Department of Defence

Defence Materiel Organisation

DEFENCE MATERIEL ORGANISATION

AEROSPACE SYSTEMS DIVISION

Impact on Quality

- Reduced rigour in testing may lead to defects not being detected
- Defects may be detected but remain untreated due to the cost of fixing them.
- Bugs in the fleet can lead to perception of poor quality software
- Poor quality software changes will result in a reluctance to fund future software changes





Australian Government

Department of Defence

Defence Materiel Organisation

DEFENCE MATERIEL ORGANISATION

AEROSPACE SYSTEMS DIVISION

Impact on Assurance

- Is a paradigm shift in the philosophy of testing.
- Instead of testing to break a product, we are testing to ensure compliance to standards.
- Pressure to reduce level of assurance because assurance does not guarantee quality.





Finding an Appropriate Balance

- If testing is to be minimised where possible, all stakeholders need to accept that there may be undetected defects in the code.
- All stakeholders need to agree on a software assurance guidance and assurance level that is appropriate for the functionality being provided
 - DO-178B Software Level A for Safety Related and Airworthiness functionality
- Developers and Testers need to explore efficiency gains through automation of tools.
- Need to improve requirement determination and estimation techniques





Australian Government

Department of Defence

Defence Materiel Organisation

DEFENCE MATERIEL ORGANISATION

AEROSPACE SYSTEMS DIVISION

Questions??

