

# SE 3501

# SOFTWARE VALIDATION AND VERIFICATION

M. Mangong Clement

# SE 3501

# SOFTWARE

# VALIDATION AND VERIFICATION

**Welcome!**

This course is design to provide a comprehensive check of a software system against its specification and to ensure you understand the process in verifying and validating a software produced.

**M. Mangong Clement**

# I. Basics introduction to Software V&V

## ❖ Class Objectives

- ✓ To introduce software verification and validation and to discuss the distinction between them.
- ✓ To discuss the program inspection process and its role in V & V.

## I. Basics introduction to Software V&V

### ❖ Verification Vs Validation

- **Verification**

*“Are we building the product right”*

- ✓ The software should conform to its specification.

- **Validation**

*“Are we building the right product”*

- ✓ The software should do what the user really requires.

## I. Basics introduction to Software V&V

### ❖ Software Verification and Validation

- These are activities that checks the software against its specification.
- Every project must verify and validate the software it produces.

## I. Basics introduction to Software V&V

### ❖ Software Verification and Validation

#### ➤ *Who is responsible to organize it?*

- The project management is responsible for organizing software verification and validation activities, the definition of software verification and validation roles, and the allocation of staff to those roles.

## I. Basics introduction to Software V&V

### ❖ Software Verification and Validation

#### ➤ *How is V & V done (1)*

- By checking that each software item meets specified requirements;
- By checking that each software item before it is used as an input to another activity;

## I. Basics introduction to Software V&V

### ❖ Software Verification and Validation

#### ➤ *How is V & V done (1)*

- By ensuring that checks on each software items are done, as for as possible , by someone other than the author;
- By ensuring that the amount of verification and validation effort is adequate to shoe each software item is suitable for operational use.



# I. Basics introduction to Software V&V

## ❖ Verification Vs Validation

### ■ **Verification**

- ✓ Should check the program meets its specification as written in the requirements document.

Example:

- This may involve checking that it meets its functional and non-functional requirements.

# I. Basics introduction to Software V&V

## ❖ Verification Vs Validation

- **Verification(1)** – it means the

- ✓ Act of reviewing, inspecting , testing, checking, auditing, or otherwise establishing and documenting whether items, processes , services or documents conform to specified requirements.

# I. Basics introduction to Software V&V

## ❖ Verification Vs Validation

- **Verification(2)** – it means the
  - ✓ Process of evaluating a system or component to determine whether the products of a given development phase satisfy the conditions imposed at the start of the phase.
  - ✓ Formal proof of program correctness

# I. Basics introduction to Software V&V

## ❖ Verification Vs Validation

- **Verification** activities - include
  - ✓ Technical reviews, walkthroughs and software inspections
  - ✓ Checking that software requirements are traceable to user requirements;
  - ✓ Checking that design components are traceable to software requirements;
  - ✓ Unit, integration, system and acceptance testing
  - ✓ Audit.

# I. Basics introduction to Software V&V

## ❖ Verification Vs Validation

### ■ **Validation**

- ✓ Should ensure that the product meets the customers expectations.

#### *Example:*

- This goes beyond checking its meets its specification; as system specifications may not always accurately reflect the real needs of users.

# I. Basics introduction to Software V&V

## ❖ Verification Vs Validation

### ▪ **Validation (according to ANSI/IEEE)**

- ✓ It's the process of evaluating a system or component during or at the end of the development process to determine whether it satisfies specific requirements.
- Validation is , therefore , 'end-to-end' verification.

# I. Basics introduction to Software V&V

## ❖ Verification and Validation Process

- **A whole life-cycle process**

- ✓ Must be applied at each stage in the software process.

- **Has two principal objectives**

- The discovery of defects in a system
- The assessment of whether or not the system is usable in an operational situation.

## I. Basics introduction to Software V&V

### ❖ Verification and Validation Goals

- V&V should establish confidence that the software is fit for its purpose.
- This does not mean completely free of defects rather, it must be good enough for its intended use and the type of use will determine the degree of confidence that is needed.



## I. Basics introduction to Software V&V

### ❖ Verification and Validation Confidence

- *To what extent should a verification and validation be done?*

*It depends on :*

- ✓ The Software function
- ✓ The User expectation
- ✓ The Marketing environment

## I. Basics introduction to Software V&V

### ❖ Verification and Validation Confidence

- *The external of V & V confidence (1)*

*It depends on :*

- ✓ **The Software function**

- The level of confidence depends on how critical the software is to an organization.

## I. Basics introduction to Software V&V

### ❖ Verification and Validation Confidence

#### ■ *The extern of V & V confidence (2)*

*It depends on :*

#### ✓ **The user Expectations**

- Users may have low expectations of certain kinds of software.

## I. Basics introduction to Software V&V

### ❖ Verification and Validation Confidence

- *The extern of V & V confidence (3)*

*It depends on :*

- ✓ **The Marketing environment**

- Getting a product to market early may be more important than finding defects in the program.

## I. Basics introduction to Software V&V

### ❖ Types of software verification

- *Software inspection ( Static verification)*
- *Software testing ( dynamic verification)*

## I. Basics introduction to Software V&V

### ❖ Types of software verification(1)

- ***Software inspection ( Static verification)***
  - ✓ Its concerns with analysis of the static system representation to discover problems.
- May be supplement by tool-based document and code analysis.

## I. Basics introduction to Software V&V

### ❖ Types of software verification(2)

#### ■ *Software testing ( Dynamic verification)*

- ✓ Its concerns with exercising and observing product behavior.
- The system is executed with test data and its operational behavior is observed.
- It reveals the presence of errors Not their absence.

# I. Basics introduction to Software V&V

## ❖ Lesson Two



# I. Basics introduction to Software V&V

## ❖ Approaches to verification – Comparison(1)

- **Testing:**
  - ✓ **Purpose:** Reveals failures
  - ✓ **Limits:** Small subset of the domain ( $\Rightarrow$  risk of inadequate test set)
- **Static verification:**
  - ✓ **Purpose:** Consider all program behaviors (and more)
  - ✓ **Limits:** *false positives, may not terminate*

## I. Basics introduction to Software V&V

### ❖ Approaches to verification – Comparison(2)

- ***Inspection/review/walkthrough:***
  - ✓ ***Purpose:*** Systematic in detecting defects
  - ✓ ***Limits:*** Informal
- ***Formal proof:***
  - ✓ ***Purpose:*** Prove correctness
  - ✓ ***Limits:*** complexity/cost (requires a formal specification)

## I. Basics introduction to Software V&V

### ❖ Program testing

- *Can reveal the presence of errors Not their absence*
- *The only validation technique for non-functional requirements as the software has to be executed to see how its behaves.*
- *Should be used in conjunction with static verification to provide full V&V coverage.*

# I. Basics introduction to Software V&V

## ❖ Types of testing

### ■ *Testing and debugging(1)*

✓ *What is the difference between these two?*

- *Defect testing and debugging are distinct processes.*
- *Verification and validation is concerned with establishing the existence of defects in a program*
- *Debugging is concerned with locating and repairing these errors.*

# I. Basics introduction to Software V&V

## ❖ Types of testing

- **Testing and debugging(2)**

- ✓ *What is the difference between these two?*

- *Debugging involves formulating a hypothesis about program behavior then testing these hypotheses to find the system error.*

## I. Basics introduction to Software V&V

### ❖ V & V planning

- *Careful planning is required to get the most out of testing and inspection processes.*
- *Planning should start early in the development process.*
- *The plan should identify the balance between static verification and testing.*
- *Testing planning is about defining standards for the testing process rather than describing product tests.*

# I. Basics introduction to Software V&V

## ❖ The V-model of development

## I. Basics introduction to Software V&V

### ❖ The structure of a software test plan

- *The testing process*
- *Requirement traceability*
- *Tested items*
- *Testing schedule*
- *Test recording procedures*
- *Hardware and software requirements*
- *Constraints*



## I. Basics introduction to Software V&V

### ❖ The structure of a software test plan(1)

- *The testing process*

- ✓ *A description of the major phases of the testing process.*

- *Requirement traceability*

- ✓ *Users are most interested in the system meeting its requirements and testing should be planned so that all requirements are individually tested.*

## I. Basics introduction to Software V&V

### ❖ The structure of a software test plan(2)

- *Tested items*

- ✓ *The products of the software process that are to be tested should be specified.*

- *Testing schedule*

- ✓ *An overall testing schedule and resource allocation for this schedule.*
- ✓ *This is obviously linked to the more general project development schedule.*

## I. Basics introduction to Software V&V

### ❖ The structure of a software test plan(3)

#### ■ *Test recording procedures*

- ✓ Its not enough simply to run tests.
- ✓ The results of the tests must be systematically recorded.
- ✓ It must be possible to audit the testing process to check that it been carried out correctly.

## I. Basics introduction to Software V&V

### ❖ The structure of a software test plan(4)

- *Hardware and software requirements*
  - ✓ *This section should set out software tools required and estimated hardware utilization.*
- *Constraints*
  - ✓ *Constraints affecting the testing process such as staff shortages should be anticipated here.*

## I. Basics introduction to Software V&V

### ❖ Inspections and testing

- ✓ *They are complementary and not opposing verification techniques*
- ✓ *Both should be used during the V & V process*
- ✓ *Inspections can check conformance with a specification but not conformance with the customer's real requirements.*
- ✓ *Inspections cannot check non-functional characteristics such as performance, usability, etc*

## I. Basics introduction to Software V&V

### ❖ Software inspections

- ✓ *These involve people examining the source representation with the aim of discovering anomalies and defects.*
- ✓ *Inspections do not require execution of a system so may be used before implementation*
- ✓ *They may be applied to any representation of the system ( requirements, design, configuration data, test data, etc.)*
- ✓ *An effective technique for discovering errors.*

## I. Basics introduction to Software V&V

### ❖ Program inspections

- ✓ *Formalized approach to document reviews*
- ✓ *Intended explicitly for defect detection ( not correction)*
- ✓ *Defects may be logical errors, anomalies in the code that might indicate an erroneous condition (e.g an uninitialized variable) or non-compliance with standards.*

## I. Basics introduction to Software V&V

### ❖ Inspection pre-conditions

- ✓ *A precise specification must be available*
- ✓ *Team members must be familiar with the organization standards*
- ✓ *Syntactically correct code or other system representations must be available*
- ✓ *An error checklist should be prepared*
- ✓ *Management must accept that inspection will increase costs early in the software process*
- ✓ *Management should not use inspections for staff appraisal –finding out who makes mistakes.*



# I. Basics introduction to Software V&V

## ❖ Inspection Procedure

- ✓ *System overview presented to inspection team.*
- ✓ *Code and associated documents are distributed to inspection team in advance*
- ✓ *Inspection takes place and discovered errors are noted.*
- ✓ *Modifications are made to repair discovered errors*
- ✓ *Re-inspection may or may not be required.*

# I. Basics introduction to Software V&V

## ❖ Inspection Roles(1)

### ■ *Author or owner*

- ✓ *The programmer or designer responsible for producing the program or document.*
- ✓ *Responsible for fixing defects discovered during the inspection process.*

### ■ *Inspector*

- ✓ *Finds errors, omissions and inconsistencies in programs and documents.*
- ✓ *May also identify broader issues that are outside the scope of the inspection team.*

# I. Basics introduction to Software V&V

## ❖ Inspection Roles(2)

- *Reader*

- ✓ *Presents the code or document at an inspection meeting*

- *Scribe*

- ✓ *Records the results of the inspection meeting*

- *Chairman or moderator*

- ✓ *Manages the process and facilitates the inspection*
- ✓ *Reports process results to the chief moderator.*

# I. Basics introduction to Software V&V

## ❖ Inspection Roles(3)

- *Chief moderator*

- ✓ *Responsible for inspection process improvements, checklist updating, standards development etc.*

# I. Basics introduction to Software V&V

## ❖ Inspection Checklists

- *Checklist of common errors should be used to drive the inspection*
- *Error checklists are programming language dependent and reflect the characteristics errors that are likely to arise in the language*
- *In general, the 'weaker' the type checking, the larger the checklist*
- *Example : **Initialization, constant naming, loop termination, array bounds , etc.***

# I. Basics introduction to Software V&V

## ❖ Inspection Check 1 (1)

### ■ *Data faults*

- ✓ *Are all program variables initialized before their values are used?*
- ✓ *Have all constant been named?*
- ✓ *Should the upper bound of arrays be equal to the size of the array or size-1?*
- ✓ *If character stings are used , is a delimiter explicitly assigned?*
- ✓ *Is there any possibility of buffer overflow?*

# I. Basics introduction to Software V&V

## ❖ Inspection Check 1 (2)

### ■ *Control faults*

- ✓ *For each conditional statement, is the condition correct?*
- ✓ *Is each loop certain to terminate?*
- ✓ *Are compound statements correctly bracketed?*
- ✓ *In case statements, are all possible cases accounted for?*
- ✓ *If a break is required after each case in case statement, has it been included?*

# I. Basics introduction to Software V&V

## ❖ Inspection Check 1 (3)

### ■ *Input/output faults*

- ✓ *Are all input variable used?*
- ✓ *Are all output variables assigned a value before they are output?*
- ✓ *Can unexpected inputs cause corruption?*



# I. Basics introduction to Software V&V

## ❖ Inspection Check 2 (1)

### ■ *Interface faults*

- ✓ *Do all function and method calls have the current number of parameters?*
- ✓ *Do formal and actual parameter types matches?*
- ✓ *Are the parameters in the right order?*
- ✓ *If components access share memory, do they have the same model of the shared memory structure?*

# I. Basics introduction to Software V&V

## ❖ Inspection Check 2 (2)

### ■ *Storage management faults*

- ✓ *If a linked structure is modified, have all links been correctly reassigned?*
- ✓ *If dynamic storage is used, has space been allocated correctly?*
- ✓ *Is space explicitly de-allocated after it is no longer required?*

# I. Basics introduction to Software V&V

## ❖ Inspection Check 2 (3)

- *Exception management faults*
- ✓ *Have all possible errors conditions been taken into account?*

# I. Basics introduction to Software V&V

## ❖ Inspection Rate

- ✓ *500 statements/ hour during overview*
- ✓ *124 source statement./hour during individual preparation*
- ✓ *90-125 statements/hour can be inspected*
- ✓ *Inspection is therefore an expensive process*
- ✓ *Inspecting 500lines cost about 40man/hour effort – about 2800 Euro at UK rages/*

# I. Basics introduction to Software V&V

## ❖ Automated static analysis

- ✓ *Static analyzers are software tools for source text processing.*
- ✓ *They parse the program text and try to discover potentially erroneous conditions and bring these to the attention of the V & V team*
- ✓ *They are very effective as an aid to inspections- they are a supplement to but not a replacement for inspections.*

# I. Basics introduction to Software V&V

## ❖ Assignment :

- ✓ *The values checks of static analysis*
- ✓ *Stages of static analysis*
- ✓ *The various type of static analysis tools*
- ✓ *How to use the static analysis*

# Welcome!

*This course is design to provide a comprehensive check of a software system against its specification and to ensure you understand the process in verifying and validating a software produced.*



## QUESTIONS

