Release testing => black box … doesn't have knowledge

Use cases => basis **(ATM system)**

Performance testing => emergent properties … part of release

Stress testing => maximum design load … causing defects

Component (unit) testing => individual in isolation

Interface testing => interface errors … invalid assumptions in interfaces

Test case design => the goal is to create a set of tests are effective ….

Requirements based testing => requirement and derive

Partition testing => input data and output results different classes

Structural testing => **white box testing … derivation program structure … objective: exercise all prog statements**

Path testing => white box testing strategy … **objective**: ensure each path is executed at least once

Nodes: program decision / **Arcs:** flow of control

Regression testing => check that changes … broken previously working code

Linguistic metrics => measuring properties of program text without interpreting

Structural metrics => structural relations … control-flow and data-flow … properties of flowgraph

M = L – N + 2P **(L: links, N: nodes, P: disconnected parts)**

Validation: evaluating software ***at the end of software development*** to **ensure compliance with intended usage**

validation: does software system **meets the user's real needs**? **(includes: usability testing, user feedback)**

Verification: determining whether the products of a ***given phase of the software development process*** **fulfill** the **requirements established** in the **previous phase**

verification: does software system **meets the requirements specifications**? **(includes: testing, inspections, static analysis)**

Validation: "Are we building **the right product**"? / **Verification:** "Are we building **the product right**"?

System that is consistent with its specification is **(dependable.)**

Safe: A safe analysis has **no optimistic inaccuracy**

Sound: Soundness is a term to describe **evaluation of formulas.**

Complete: Completeness, **like soundness**, is a term to describe **evaluation of formulas.**

black-box testing => little understanding of software **internals**

white-box testing => full understanding of software **internals**

Error: is a **mistake** made in the **code**; that's why we cannot execute or compile code.

Fault: is a state that causes the **software to fail** to accomplish its **essential function**.

Failure: If the software has **lots of defects**, it leads to **failure** or **causes failure**