

CS-577b Quiz2

QM3 2/26/01

1. Which of the following is **not** a recognized software testing technique?

a) Data-flow testing	b) Domain testing	c) Destructive testing
d) Logic-based testing	e) Path testing	f) Syntax testing
g) Transaction-flow testing	h) Transition testing	i) None of the above

2. In MBASE, what determines the kinds and amounts of testing to be performed:

3. Circle all of the following that are practiced in CS577b:

a) **Automated** Analysis

Simple compiler syntax checking.
Basic compiler or additional tools capabilities for static module-level code analysis, and syntax- and type-checking.
Some compiler extensions for static module and inter-module level code analysis, and syntax- and type-checking.
Basic requirements and design consistency; and traceability checking.
Intermediate-level module and inter-module code syntax and semantic analysis. Simple requirements/design consistency checking across views.
More elaborate requirements/design view consistency checking. Basic distributed-processing and temporal analysis, model checking, symbolic execution.
Formalized [Consistency-checkable pre-conditions and post-conditions, but not mathematical theorems] specification and verification.
Advanced distributed-processing and temporal analysis, model checking, symbolic execution.

b) Execution Testing and Tools

Ad-hoc testing and debugging. Basic text-based debugger
Basic unit test, integration test, system test process. Basic test data management, problem tracking support. Test criteria based on checklists.
Well-defined test sequence tailored to organization (acceptance, alpha, beta, flight, etc. test). Basic test coverage tools, test support system. Basic test process management.
More advanced test tools, test data preparation, basic test oracle support, distributed monitoring and analysis, active assertion checking. Metrics-based test process management.
Highly advanced tools for test oracles, distributed monitoring and analysis, assertion checking. Integration of automated analysis and test tools. Model-based test process management.