Test Planning

Testing is an opportunity to measure the quality of a product.

It's also the way we find and remove defects!

Next, we will concentrate on understanding:

- **The strategies and context of a testing effort**
- **The steps to developing a test plan**

Why Do We Test?

"The goal of testing is to uncover and fix as many errors as possible, within the bounds of diminishing returns"

- Futrell, Shafer & Shafer

"Testing cannot show the absence of defects, it can only show that software defects are present"

- Robert Pressman

1. Testing is a process of executing a program with the intent of finding an error
2. A good test case is one that has a high probability of finding an as yet undiscovered error
3. A successful test is one that uncovers an as yet undiscovered error

- Glenford J. Myers (IBM, 1979)

Why Do We Test?

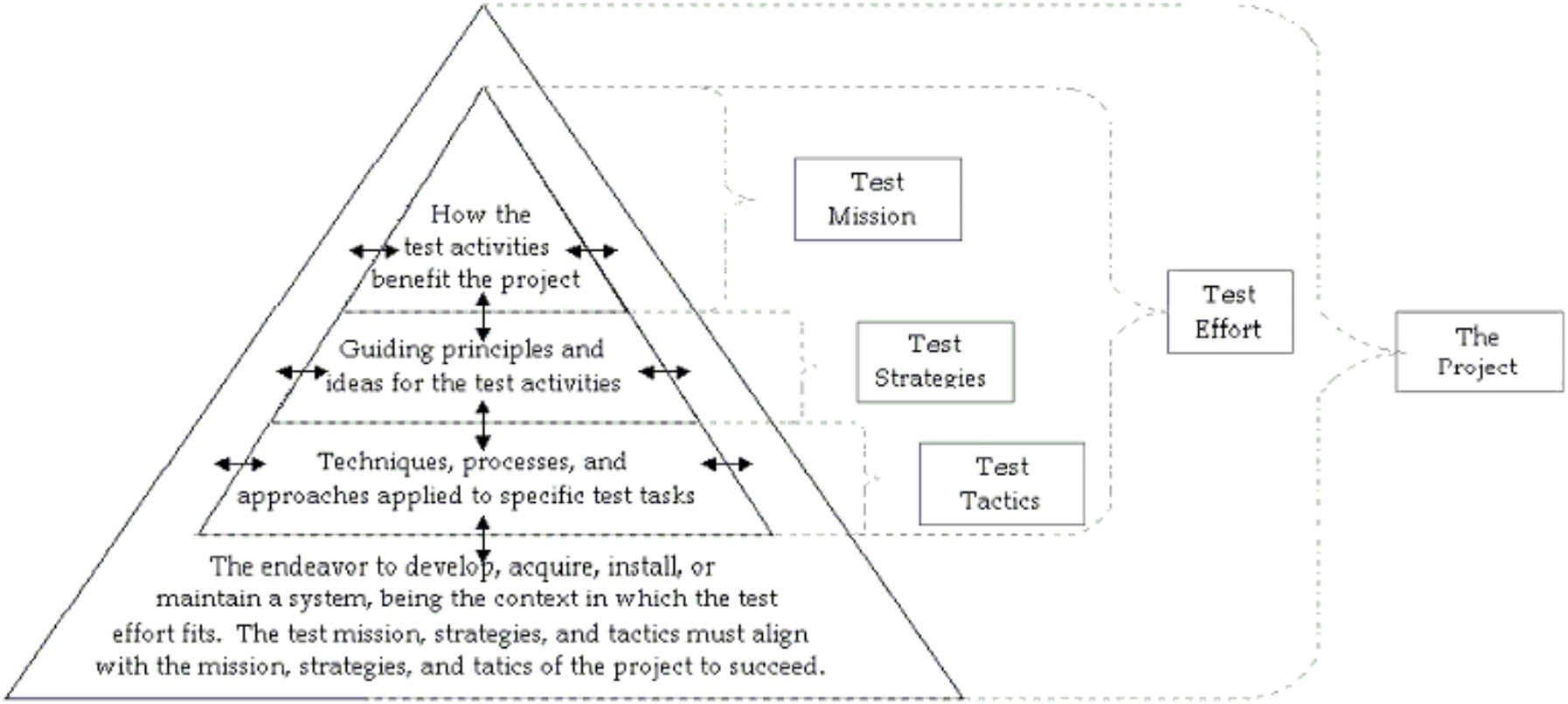
- Improve / maintain quality
- Reduce maintenance costs after implementation
- Smoother, more predictable development time
- Increase customer experience and satisfaction
- Reduce risk of unexpected losses
- Reduce the likelihood of lawsuits

The value of testing software

- ***Testing must provide business value within the context of the product's development***
  - ***Is the software built right?***
- ***Testing must also provide business value within the context of its intended use***
  - ***Is it the right software?***

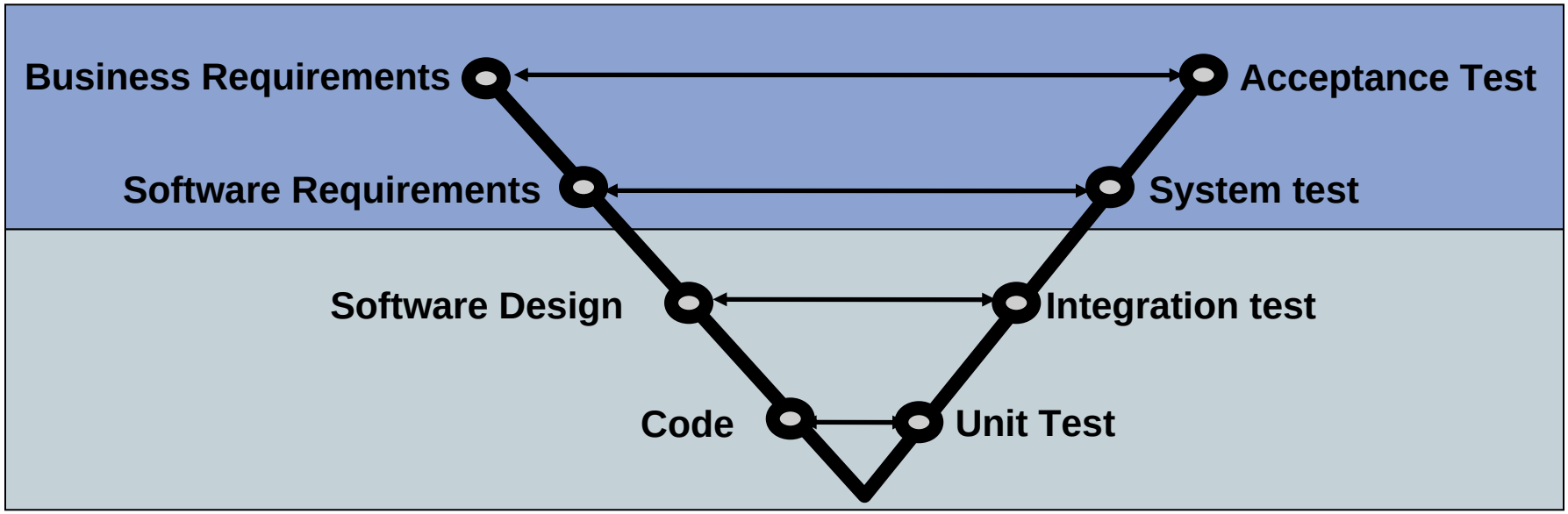
- **Using testing strategies ensures the context is appropriate and the test plan is realistic.**

Testing is Aligned With Project Context



Basic Test Strategy: Testing Levels

- To fit into the project context, it is useful to break testing down into levels aligned with the different phases of the project life cycle.



- Different names may be used
- Testing objectives differ at each level
- Users are typically involved in requirements development and corresponding testing

Testing and Project Context

Commercial Off-the-Shelf Software (COTS)

- Considerations when acquiring COTS
  - Will it satisfy the business & technical requirements?
  - What customizations and interfaces will be needed?
  - What is the vendor's past record of test/quality?
- The main goal of a COTS producer is to sell the product to as many buyers as possible.
- The buyers goal is to ensure the product meets the business need and relatively defect free
- Test COTS by installing and integrating the product in the environment where the software will be used
  - Technical support and/or software quality assurance staff test for compatibility and stability
  - ***Users perform acceptance testing in the form of alpha and/or beta tests***

Testing in the Project Context

Custom-Built Software - Developer Testing

- Unit tests each software component / module
- Integration tests the links between components
- System tests end-to-end functionality
- Stress tests peak load conditions
- Volume tests capacity for handling large inputs
- Performance tests response time / other constraints
- Other tests (security, reliability, documentation etc.)
- Considerations during development testing
  - Test environment may not mirror the user environment
  - May use simulated testing data rather than actual data
  - Developers may not interpret results correctly

Testing in the Project Context

Custom-Built Software – User Acceptance

- Determines whether the software meets user requirements
- Can be performed by
  - Representatives from the end-user organization
  - SQA group with end-user representatives
  - Independent group hired by the customer
- Four major concerns
  - correctness
  - robustness
  - performance
  - documentation
- Uses actual data and business process
- Interpreted by business experts
- Uses customer environment
- Any changes after acceptance, are maintenance

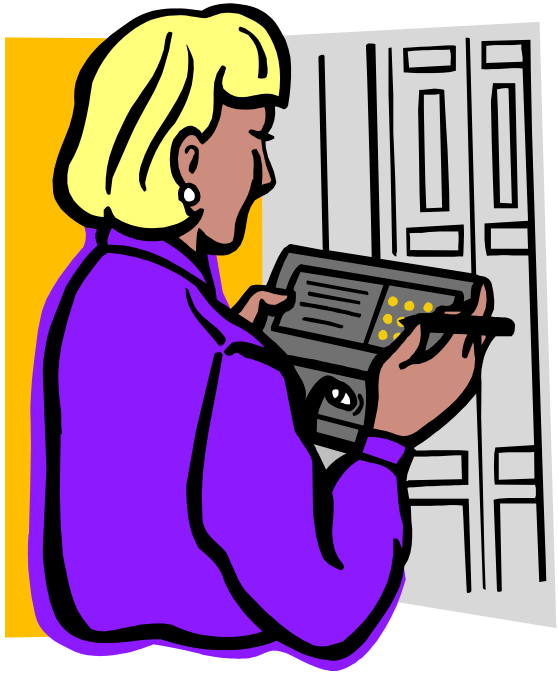
Testing in the Project Context

Regression Testing

- To uncover defects that occur when previously working software no longer works in the same way
- Regression defects are unintended and occur when there are changes to the software
- Includes re-running tests to see if previously fixed defects re-occur or if new defects emerge
- Useful for measuring readiness
- Especially important for maintenance releases that change existing functionality
- Some regression tests may be automated

Testing in the Project Context

- Tools And Techniques
  - Test Processes And Procedures
  - Test Automation Tools
  - Tracking And Reporting Tools
- Training Needs
- Available Resources
  - People
  - Test Environment



System Testing

- A system test is a series of tests designed to fully exercise the system to uncover its limitations and measure its capabilities
  - test the complete system to verify it meets specified requirements
- Common types of system testing:
  - Functional
  - Recovery
  - Security
  - Stress
  - Performance

- In some cases users may participate in system testing

Acceptance Tests

- The purpose of acceptance tests are to confirm that the system is developed in accordance with user requirements and is ready for operational use
- Part of a formal handoff or release process
- Involves end users / customers
- Types:
  - Alpha
  - Beta
  - Benchmark
  - Installation

The Software Quality Institute

Designing a Test Strategy from Goals

- Testing is a way to measure the software's ability to help the business achieve goals
- Acceptance through testing implies there is a threshold of acceptable results
- How will you know when acceptable results have been achieved?
- Goal-Question-Metric (GQM) framework
  - Step 1: Develop goals (for quality)
  - Step 2: List the questions that must be answered to determine if goals are being met
  - Step 3: Specify the metrics, the things to be measured, in order to answer the question

Dr. Victor Basili authored the Goal, Question, Metric (GQM) paradigm

Step 1: Develop Goals

- Analyze 'what' - the object under measurement can be a process, model, product, metric, etc.
- For the purposes of (why) – for example: evaluating, predicting, assessing, improving, controlling, etc.
- In terms of – The quality focus of the object such cost, effectiveness, correctness, friendliness, etc.
- From the viewpoint of (who's perspective - the people that measure the object)
- In the context of (characteristics of the environment where or when the measurement takes place)

Goals

- **Examples of Goals for Acceptance Testing?**

Step 2: Questions

- **Questions should be aimed at the operational level**
- **Answering the questions will determine whether the goal has been reached**
- **Examples of questions for Acceptance Testing Goals?**

Step 3: Metrics

- **Specify the metrics or measures needed to answer the questions**
- **May be derived from key words in the questions**
- **Examples of metrics needed to answer the Acceptance Testing questions?**

Other Test Strategies

- Analytical
  - Risk-based
  - Object-guided
  - Fully-informed
- Model-based
  - Scenario-based
  - Use-case-based
  - Domain-based
- Methodical
  - Learning-based
  - Function-based
  - State-based
  - Quality-based

Gathering and Reporting on Goals

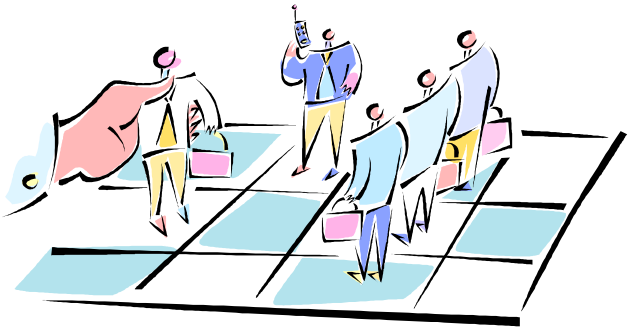
- Make sure mechanisms are in place to gather metrics
- Collect, validate and analyze the data in time to make corrective actions if necessary
- Provide constructive feedback
- Keep it simple at first and make improvements over time

Risk-Based Testing Strategies

- Identify the specific risks to system quality that need to be tested
- All key stakeholders are involved in the risk analysis
- Consider different types of risk: technical, functional, business.
- Prioritize those risks
- Determine the appropriate way and amount of testing for those risks

More Test Strategies

- Standards or Best Practice
  - Formal Standard (IEEE 829)
  - Agile
  - Automated Random
  - Automated Functional
- Dynamic Test Strategies
  - Bug hunting
  - Intuitive
  - Exploratory
- Philosophical
  - Exhaustive
  - Shotgun
  - Externally-guided



Characteristics of Testing Strategies

- Perform reviews in order to eliminate errors early in and throughout the life cycle and thus "build-in" quality
- Employ different testing techniques different times and on different parts of the system
- Early testing is always conducted by the developer. Larger systems require independent test groups
- Debugging is not testing, but it is included in the test strategy
- Plan for re-testing
- Testing will not uncover all errors – target critical, high-risk areas

Success Factors

- Develop effective communications and relationships between developers and users
- Specify requirements in a way that can be verified
- Fully develop testing goals that are directly linked to acceptance criteria
- Participate in formals reviews of the software requirements to remove defects prior to development
- Conduct a formal reviews on the test strategy, plans and cases
- Learn from problems and mistakes and continuously improve the testing process

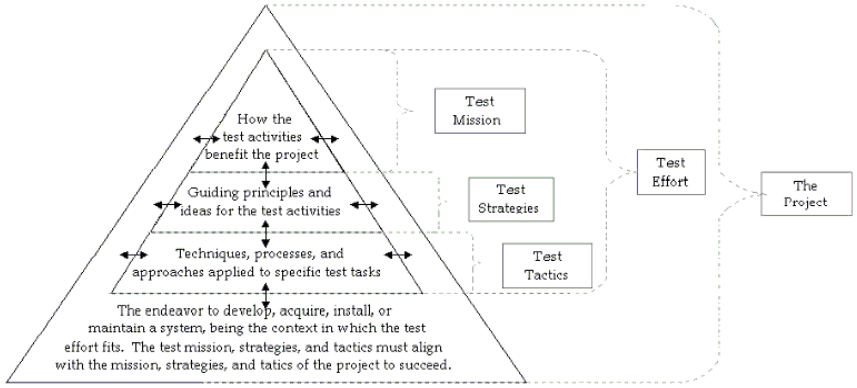
Testing Principles

- Tests should be planned long before testing begins
- All tests should be traceable to requirements
- The Pareto principle applies to software testing
  - 80% of all errors will likely be in 20% of the software
- Testing should begin "in the small" and progress toward testing "in the large"
- Exhaustive testing is not possible
- An independent third party may conduct testing

"The test team should always know how many errors still remain in the code - for when the manager comes by."

Test Planning

- **Managing the testing of a large system is difficult**
  - **applicable strategies identified**
  - **techniques and tools put in place**
  - **numerous test cases**
  - **defect reporting**
  - **defect fixing**
  - **numerous phases**
  - **multiple cycles**



- **Testing is a project within a project, therefore…**
- **Testing must be planned, resources must be allocated and a schedule needs to be set!**

Benefits of a Test Plan

- Establishes a test schedule
- Acts as a service agreement between testers and developers
- Identifies what will be tested and risks involved in testing
- Communicates
  - Methods which will be used to test the product
  - Required resources:
    - hardware
    - software
    - people / skills / knowledge
  - Testing schedule (tests phases and cycles)
  - Process for managing the testing effort

Developing a Test Plan

- Describes the testing activity
  - Determine which tests you need to perform
    - Hint: Use your strategies!
  - Create a test plan using a template
  - Begin planning and developing test cases
  - Review the test plan
  - Incorporate feedback
  - Get approval
  - Follow the plan

Typical Test Plan Contents

- Introduction
- Scope
- Test Plan Strategy
- Test Environment
- Schedule
- Control Procedures
- Control Activities
- Functions to be Tested
- Functions not to be Tested
- Risks and Assumptions
- Deliverables
- Test Tools
- Approvals
- Exit Criteria

Test Plan Sections

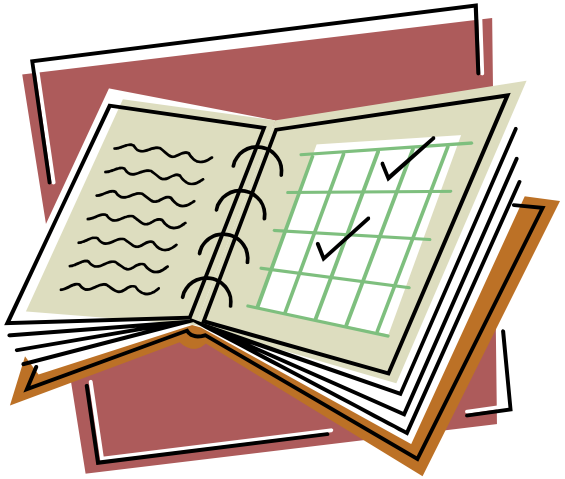
- **Introduction**
  - **brief description of system being tested, providing overview of its features**
- **Scope**
  - **high level description of what will and will not be tested**

Test Plan Sections

- **Test Plan Strategy**
  - **outlines goals of test plan**
  - **establishes procedures for conducting the tests**
  - **establishes priorities for testing**
  - **brief description of each test type**
- **Test Environment**
  - **hardware and software configurations required to conduct testing**

Test Plan Sections

- **Schedule - start and end dates for**
  - **development of the test plan**
  - **designing test cases**
  - **executing test cases**
  - **problem reporting**
  - **developing the test summary report**



Test Plan Sections

- **Control Procedures**
  - **detailed instructions for problem reporting**
  - **indicate**
    - **reporting mechanisms**
    - **how problems should be prioritized**
      - **severity levels**

        1 = missing functions, function not working

        2 = work around exists for non-working function

        3 = cosmetic errors and spelling mistakes
    - **how fixes will be accepted**

Test Plan Sections

- Control Activities
  - When walkthroughs and reviews will be conducted
  - Change control processes
  - Persons responsible
- Functions to be tested
- Functions not to be tested and reasons why not

Test Plan Sections

- **Risks and Assumptions**
  - **"Regression testing will be automated"**
  - **"All testing personnel will have the knowledge of the product and will be trained in testing procedures prior to testing"**
  - **"The computers will be available during all work hours"**

Test Plan Sections

- Describe Testing Deliverables
  - Examples
    - Other test plans
    - Test cases
    - Test incident report
    - Test summary report
- Test Tools
  - in-house or externally acquired

Test Plan Sections

- Approvals
  - Reviewed and approved by individuals affected by or dependent upon the document
- Exit Criteria
  - Identifies when the testing phase is complete
    - No priority-one open problems
    - All functions identified in the requirements document are present and working
    - No more than three priority-two problems open

Estimating the time for testing

- How long will it take to run all tests?
  - Count the tests and categorize them
  - Estimate time each category of test will take
  - Determine how many testers will be involved
  - Determine tester productivity rate
- How many defects will be found and how long will it take to confirm the defect has been resolved?
  - Estimate the number of defects in each category
  - Estimate the time to re-run tests
- How long will regression testing take?
  - Select regression test cases in each category
  - Determine how many regression tests will be added as a result of defect

Other factors in estimating test time

- Consider the type of testing strategy- Scripted vs. exploratory
- Develop a regression testing strategy (repeat tests or just run once?)
- Generate a schedule using test cycles
- Defective deliverables increase testing time – effective reviews will decrease testing time
- A higher rate of defects means more time spent re-testing
- Be realistic about tester productivity – no one is productive 100% of the time!
  - Emails, meetings, vacation, sick time
  - Blocking defects

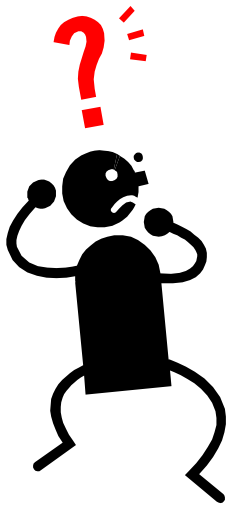
How long to get the defects out?

Developer Activities to resolve or correct a defect:

- **Recreate the problem**
- **Analysis and fix**
- **Retest**

User Activities

- **Regression testing**
- **Retest**
- **Documentation and reporting**



How long will testing take?

	Cycle 1			Cycle 2		
	Complex	Moderate	Simple	Complex	Moderate	Simple
Time to Execute One Test	80	15	5	80	15	5
Time to Document/Report/Resolve	30	15	15	30	15	15
How Many Tests in Cycle	5	16	35	10	8	22
Time to Run Test (row 3 x row 5)	400	240	175	800	120	110
Estimated Defects in Cycle	2	5	8	4	3	3
Defect Fix Time	60	75	120	120	45	45
Retest time	160	75	40	320	45	15
Number of Regression tests	1	4	6	5	11	18
Time to Execute Regression Tests	80	60	30	80	15	5
Total Estimate Cycle Time	700	450	365	800	165	155
Total Hrs of Test for Cycle	25.3			18.7		
Tester Productivity(hrs/day)	6			6		
Man Days of Testing	4.2			3.1		
Number of Testers	2			2		
Days of Testing for Cycle	2.1			1.6		

Test Planning Checklist

- Have major test phases been properly identified and sequenced?
- Has traceability between test criteria & requirements been established?
- Are major functions demonstrated early?
- Is the test plan consistent with the overall project plan?
- Has a test schedule been explicitly defined?
- Are test resources and tools identified and available?
- Has a test record-keeping mechanism been established?
- Has exit criteria been defined?

Summary

- **Use testing strategies to determine what is the right approach to testing for your project.**
- **Testing is a project within a project – so treat it like one…**

> Like death and taxes, testing is both unpleasant and inevitable.
>
> - Ed Yourdon

…develop a test plan! You won't avoid testing, but you'll minimize the unpleasantness!

Next...

- **Test Execution**
  - **Developing Test Cases**
  - **Running Tests**
  - **Evaluating Results**
  - **Reporting Results**