

Presentation (Technical Paper)
Ottawa Test Engineering Workshop
Ottawa, Canada
Oct. 21 – 23, 2002

Software Testing in Today's High-tech Industry

Alok Saxena

Cistel Technology Inc.

asaxena@cistel.com 613-226-8344

210 Colonnade Road

Ottawa, K2E 7L5

www.cistel.com





Software Testing in Today's High-tech Industry

Alok Saxena
Cistel Technology Inc.

asaxena@cistel.com 613-226-8344

210 Colonnade Road

Ottawa, K2E 7L5

www.cistel.com





Software Testing

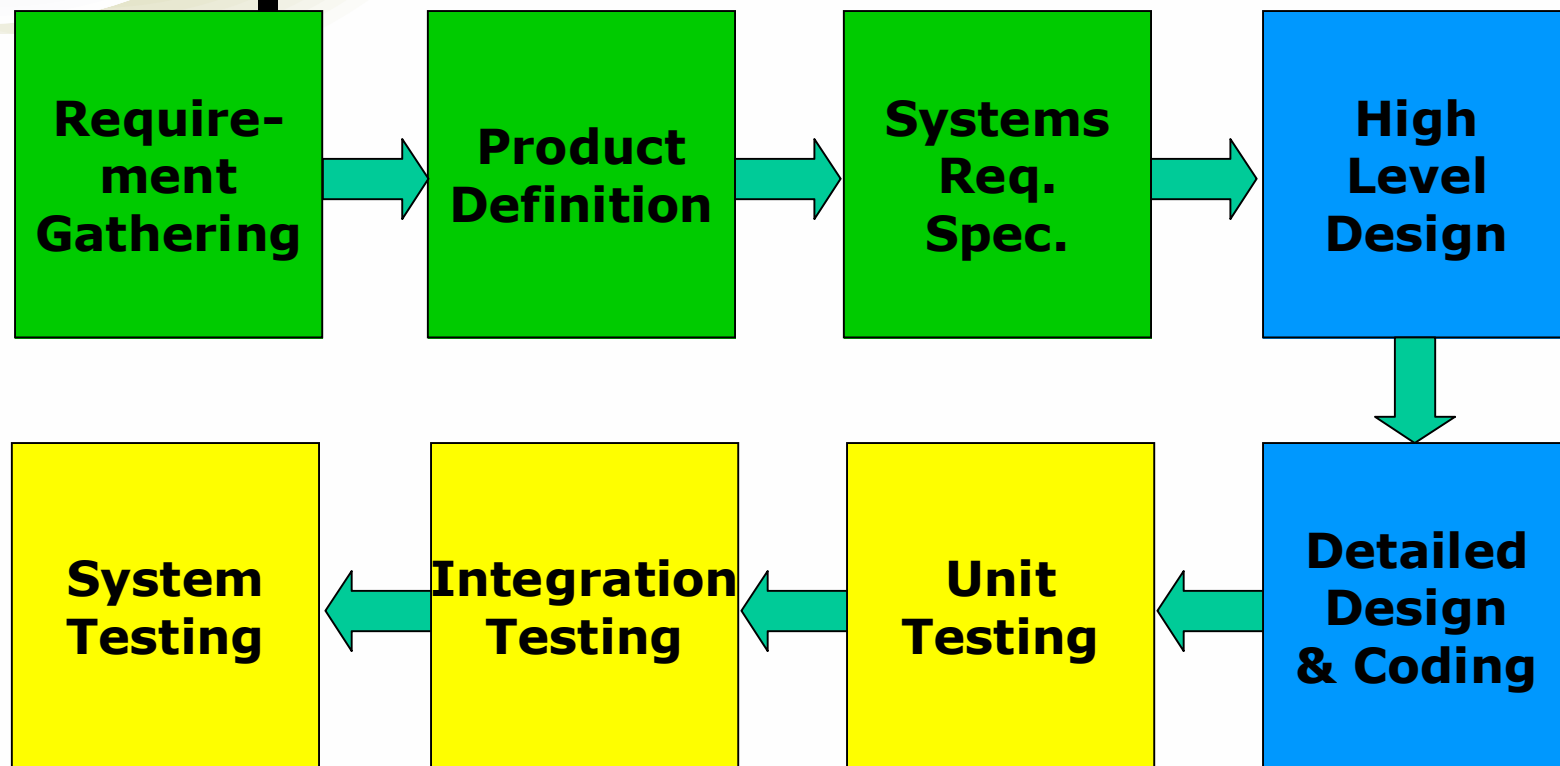
- Suffering in today's environment
 - Lack of resources
 - Cost-cutting
 - Lack of importance placed on testing
 - Urgency to deliver product
 - Quality suffers
 - Lack of understanding
 - Optimum level



Why is Testing so Important

- Quality vs. Quick to market
- Customers demanding quality
- Competition
 - Fiercer now than ever before

Software Dev. Cycle





Unit Testing

- White-box testing
 - Short cycle-time for bug fixes
- One (small group) designer
- Ensures code integrity
- Does not verify interactions among subsystems



Integration Testing

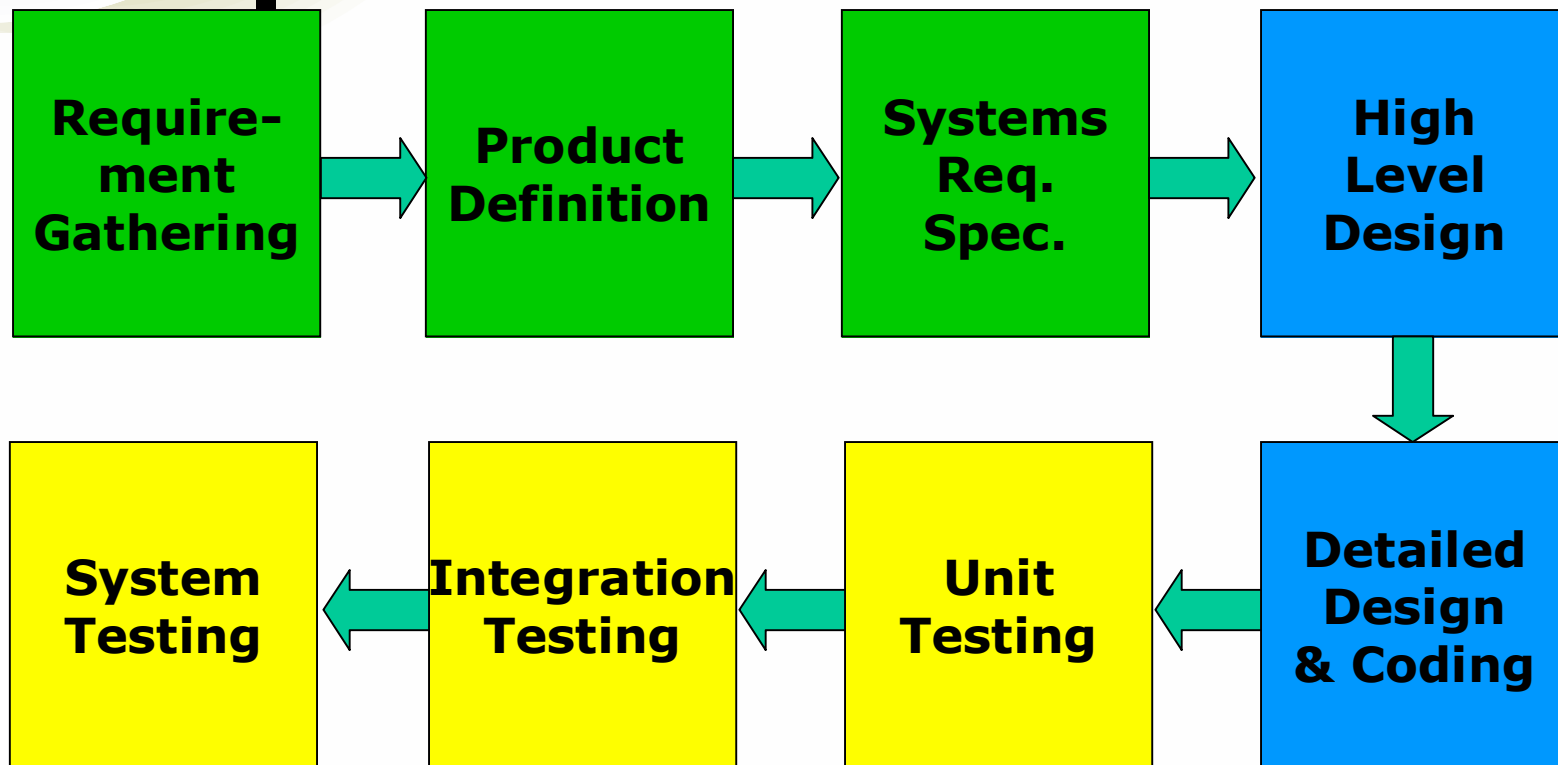
- White-box and Black-box testing
 - It should be more black than white
 - Longer cycle time for bug fixes
 - Expensive if IT uncovers UT bugs
- Ensures the integrity of subsystem interfaces
- More than one phases possible
- Responsibility of test organization



System Testing

- Black-box testing
 - Much longer cycle time for bug fixes
 - Much more expensive to fix bugs
- Verifies the overall system against the requirements
- More than one phase possible
- Responsibility of Product Validation or Customer Acceptance groups

Software Dev. Cycle





Typical Problems

- Lack of emphasis on Unit Testing
 - Urgency to finish the coding
 - Designers are overloaded
- Consequences
 - Integration testing group gets bogged down into white box testing
 - Too-much software churn
 - Expensive



Typical Problems

- Lack of emphasis on Integration and System Testing
 - Lack of resources within testing group
 - Pressures to meet deadlines
 - Belief that UT has covered most testing
- Consequences
 - Subsystem interactions not tested thoroughly
 - End-to-end system behavior
 - Poor software quality



Solutions

- Unit Testing should provide complete code coverage
- Integration Testing should focus only on subsystem interactions
- Functional Testing should verify the overall product behavior



Allocation of Testing Time

- Unit Testing time
 - schedule it along with coding time
- Integration Testing
 - Two cycles needed to ensure quality software
- System Testing
 - Relatively short cycle, if UT and IT have been done properly
 - Cost of fixing a bug rises exponentially



Recommendations

- Integration and System Testing groups should be at arms length from Design groups
- System Testing group should be combined with the Systems Engineering group (Sys Req.)



Recommendations

- Outsource Integration and System Testing
- Benefits of Outsourcing:
 - Tester at arms length
 - Avoid the expenses of large testing group
 - Have resources on a as-needed basis
 - Improved product quality
 - Reduced Cost