TF

AM Tutorial
4/30/13 8:30AM

Rob Sabourin: On Testing

Presented by:

Rob Sabourin
AmiBug.com

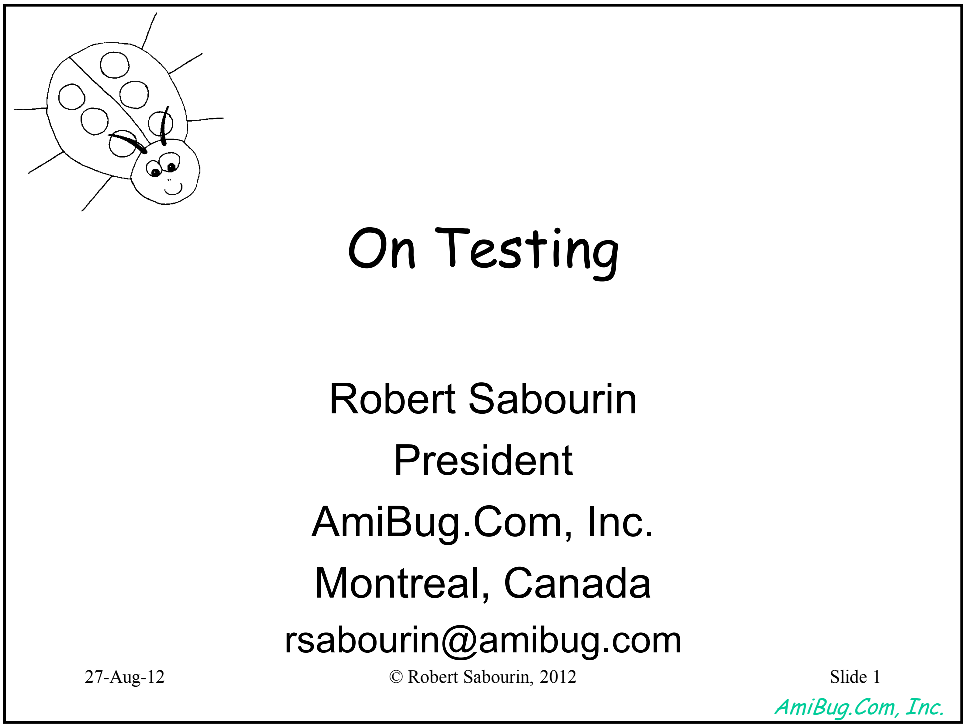
Brought to you by:

SOFTWARE QUALITY ENGINEERING

340 Corporate Way, Suite 300, Orange Park, FL 32073
888-268-8770 · 904-278-0524 · sqeinfo@sqe.com · www.sqe.com

Rob Sabourin

Rob Sabourin, P. Eng., has more than thirty years of management experience leading teams of software development professionals. A well-respected member of the software engineering community, Rob has managed, trained, mentored, and coached hundreds of top professionals in the field. He frequently speaks at conferences and writes on software engineering, SQA, testing, management, and internationalization. Rob wrote I am a Bug!, the popular software testing children's book; works as an adjunct professor of software engineering at McGill University; and serves as the principle consultant (and president/janitor) of AmiBug.Com, Inc. Contact Rob at rsabourin@amibug.com.



On Testing

Robert Sabourin

President

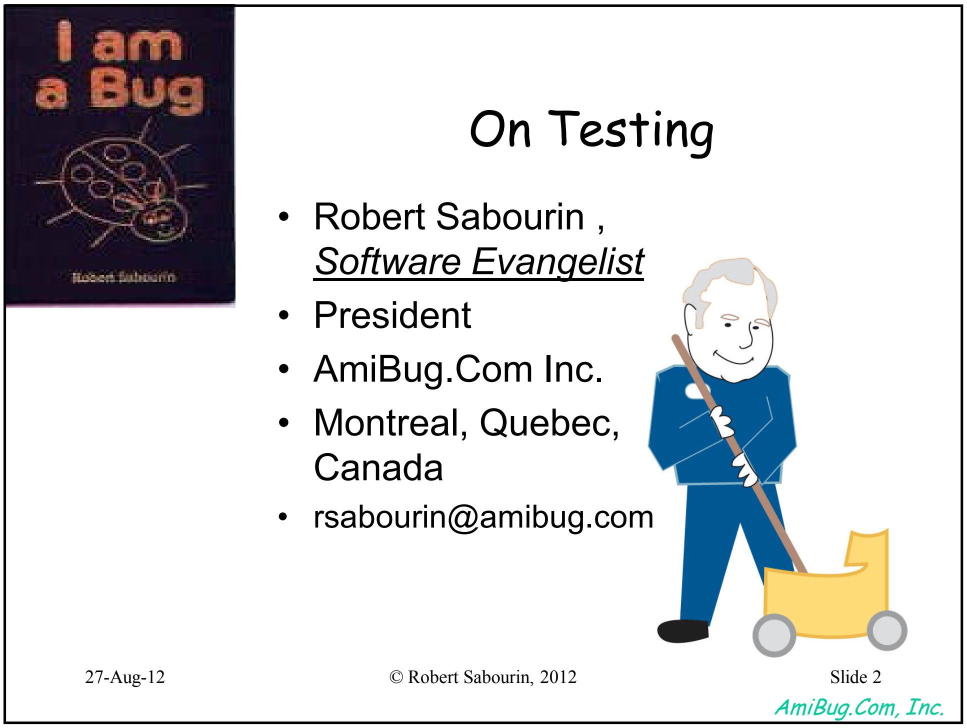
AmiBug.Com, Inc.

Montreal, Canada

rsabourin@amibug.com

27-Aug-12 © Robert Sabourin, 2012 Slide 1

AmiBug.Com, Inc.



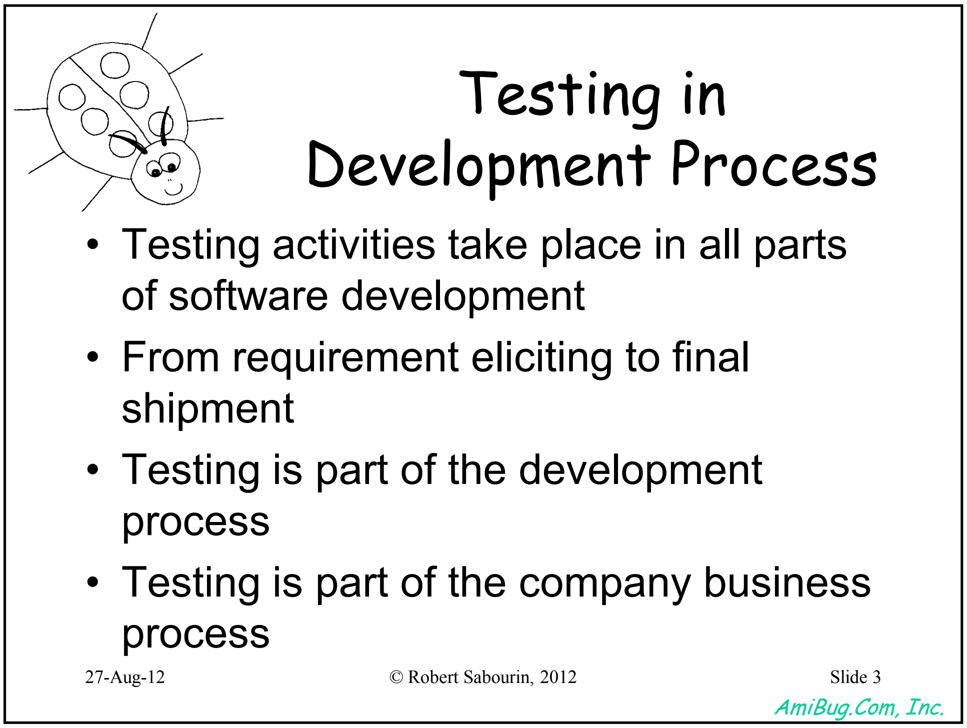
I am
a Bug
Robert Sabourin

On Testing

- Robert Sabourin , *Software Evangelist*
- President
- AmiBug.Com Inc.
- Montreal, Quebec, Canada
- rsabourin@amibug.com

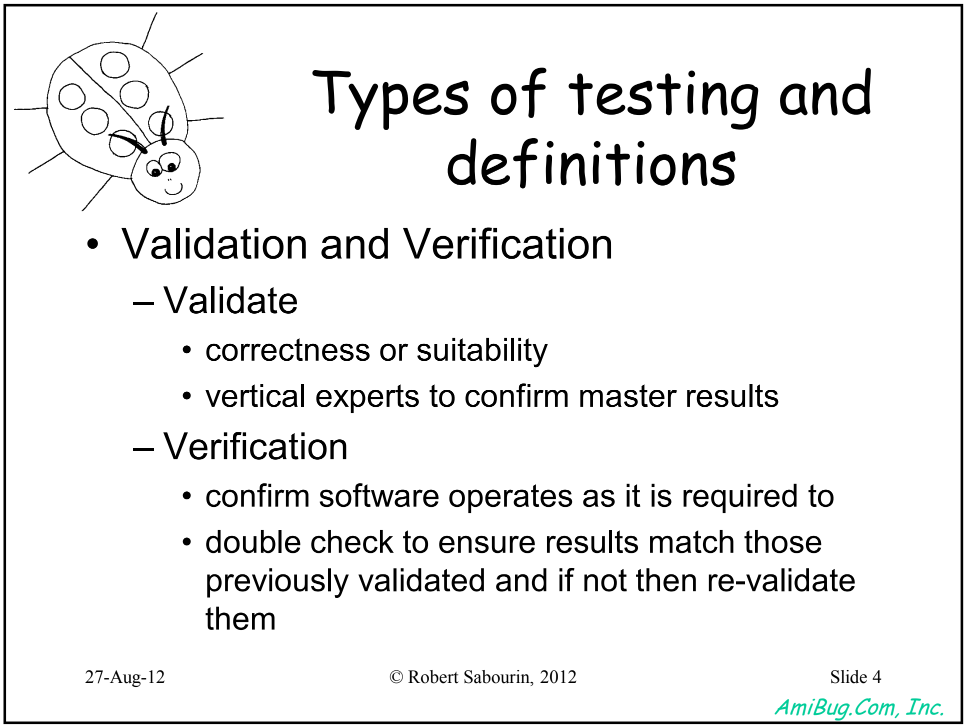
27-Aug-12 © Robert Sabourin, 2012 Slide 2

AmiBug.Com, Inc.



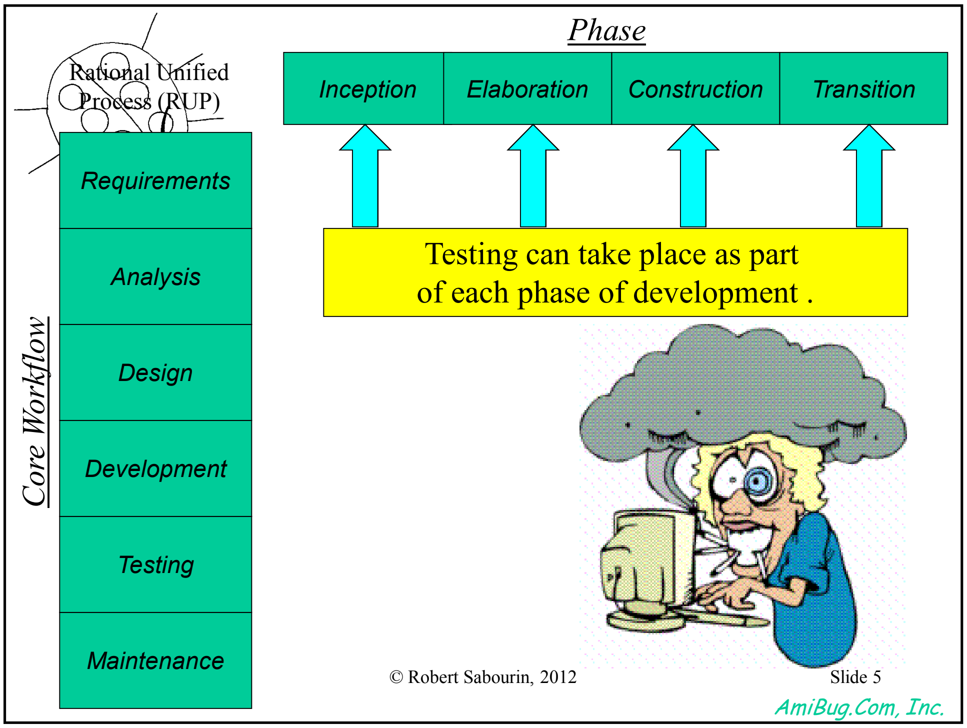
Testing in Development Process

- Testing activities take place in all parts of software development
- From requirement eliciting to final shipment
- Testing is part of the development process
- Testing is part of the company business process



Types of testing and definitions

- Validation and Verification
  - Validate
    - correctness or suitability
    - vertical experts to confirm master results
  - Verification
    - confirm software operates as it is required to
    - double check to ensure results match those previously validated and if not then re-validate them



Phase

Inception | Elaboration | Construction | Transition

Rational Unified Process (RUP)

Core Workflow

Requirements

Analysis

Design

Development

Testing

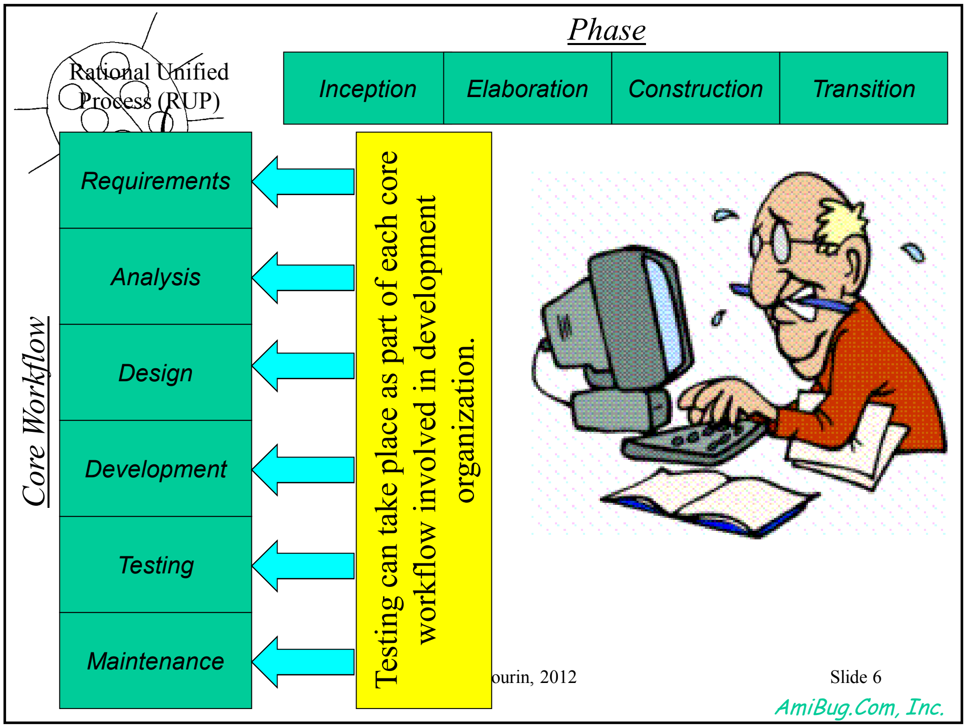
Maintenance

Testing can take place as part of each phase of development .

© Robert Sabourin, 2012

Slide 5

AmiBug.Com, Inc.



Phase

Inception | Elaboration | Construction | Transition

Rational Unified Process (RUP)

Core Workflow

Requirements

Analysis

Design

Development

Testing

Maintenance

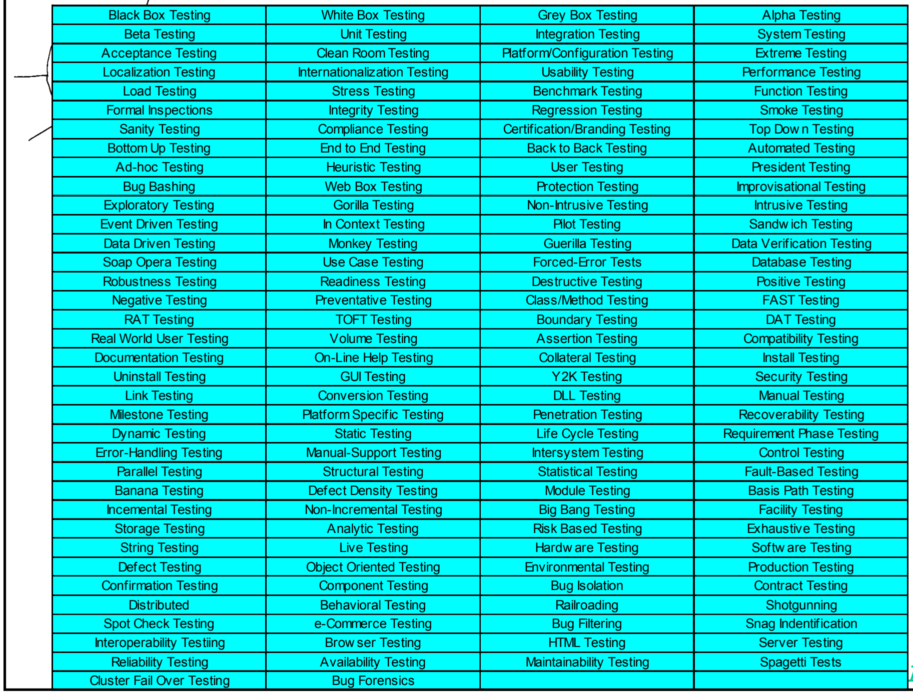
Testing can take place as part of each core workflow involved in development organization.

ourin, 2012

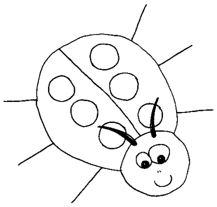
Slide 6

AmiBug.Com, Inc.

Testimonial

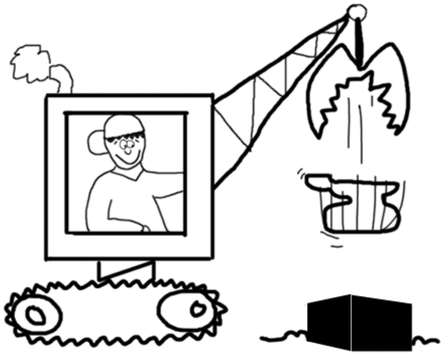


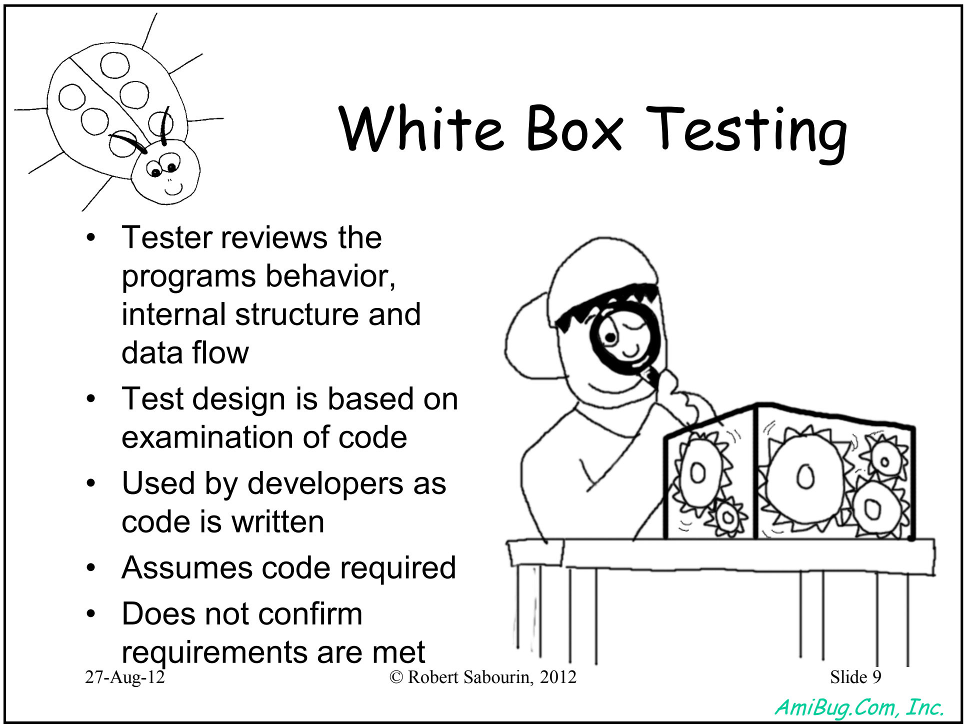
Black Box Testing	White Box Testing	Grey Box Testing	Alpha Testing
Beta Testing	Unit Testing	Integration Testing	System Testing
Acceptance Testing	Clean Room Testing	Platform/Configuration Testing	Extreme Testing
Localization Testing	Internationalization Testing	Usability Testing	Performance Testing
Load Testing	Stress Testing	Benchmark Testing	Function Testing
Formal Inspections	Integrity Testing	Regression Testing	Smoke Testing
Sanity Testing	Compliance Testing	Certification/Branding Testing	Top Dow n Testing
Bottom Up Testing	End to End Testing	Back to Back Testing	Automated Testing
Ad-hoc Testing	Heuristic Testing	User Testing	President Testing
Bug Bashing	Web Box Testing	Protection Testing	Improvisational Testing
Exploratory Testing	Gorilla Testing	Non-Intrusive Testing	Intrusive Testing
Event Driven Testing	In Context Testing	Pilot Testing	Sandw ich Testing
Data Driven Testing	Monkey Testing	Guerilla Testing	Data Verification Testing
Soap Opera Testing	Use Case Testing	Forced-Error Tests	Database Testing
Robustness Testing	Readiness Testing	Destructive Testing	Positive Testing
Negative Testing	Preventative Testing	Class/Method Testing	FAST Testing
RAT Testing	TOFT Testing	Boundary Testing	DAT Testing
Real World User Testing	Volume Testing	Assertion Testing	Compatibility Testing
Documentation Testing	On-Line Help Testing	Collateral Testing	Install Testing
Uninstall Testing	GUI Testing	Y2K Testing	Security Testing
Link Testing	Conversion Testing	DLL Testing	Manual Testing
Milestone Testing	Platform Specific Testing	Penetration Testing	Recoverability Testing
Dynamic Testing	Static Testing	Life Cycle Testing	Requirement Phase Testing
Error-Handling Testing	Manual-Support Testing	Intersystem Testing	Control Testing
Parallel Testing	Structural Testing	Statistical Testing	Fault-Based Testing
Banana Testing	Defect Density Testing	Module Testing	Basis Path Testing
Incemental Testing	Non-Incremental Testing	Big Bang Testing	Facility Testing
Storage Testing	Analytic Testing	Risk Based Testing	Exhaustive Testing
String Testing	Live Testing	Hardw are Testing	Softw are Testing
Defect Testing	Object Oriented Testing	Environmental Testing	Production Testing
Confirmation Testing	Component Testing	Bug Isolation	Contract Testing
Distributed	Behavioral Testing	Railroading	Shotgunning
Spot Check Testing	e-Commerce Testing	Bug Filtering	Snag Indentification
Interoperability Testiing	Brow ser Testing	HTML Testing	Server Testing
Reliability Testing	Availability Testing	Maintainability Testing	Spagetti Tests
Cluster Fail Over Testing	Bug Forensics		



Black Box Testing

- Tester views the program as a black box
- Test is not concerned about the internal behavior and structure of the program
- Test is designed to observe and confirm outcome of program in response to input and system state



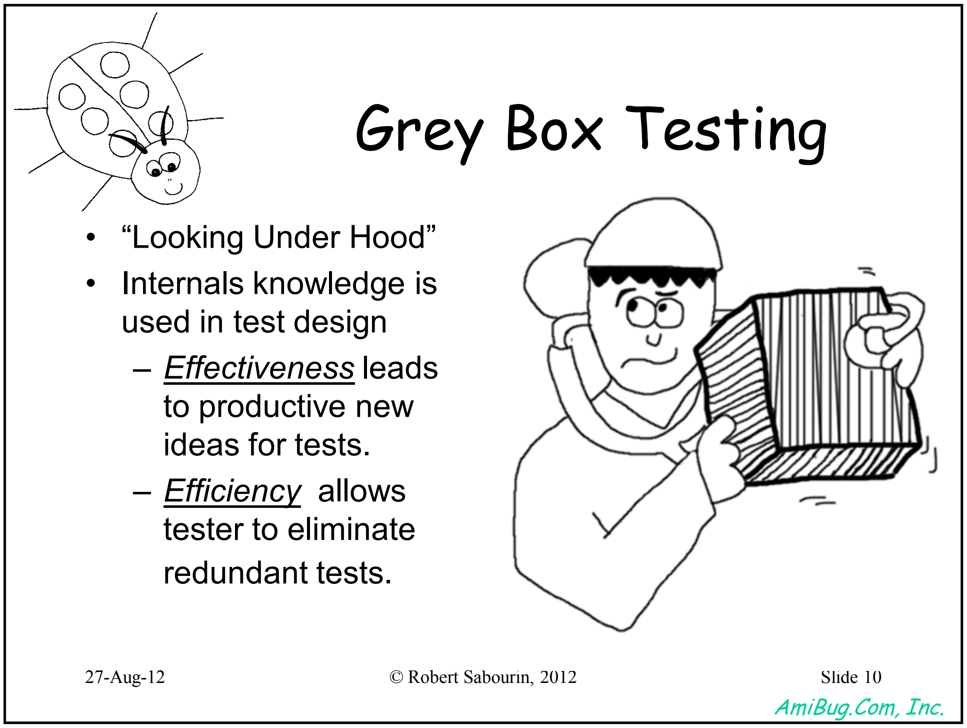


White Box Testing

- Tester reviews the programs behavior, internal structure and data flow
- Test design is based on examination of code
- Used by developers as code is written
- Assumes code required
- Does not confirm requirements are met

27-Aug-12 © Robert Sabourin, 2012 Slide 9

AmiBug.Com, Inc.

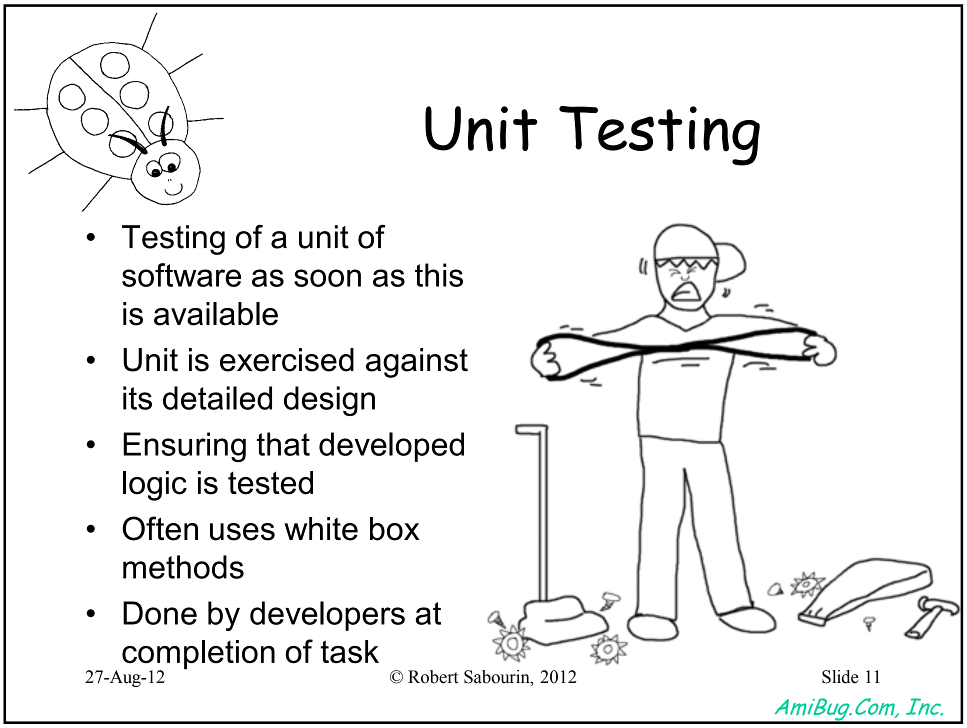


Grey Box Testing

- “Looking Under Hood”
- Internals knowledge is used in test design
  - *Effectiveness* leads to productive new ideas for tests.
  - *Efficiency* allows tester to eliminate redundant tests.

27-Aug-12 © Robert Sabourın, 2012 Slıde 10

AmiBug.Com, Inc.

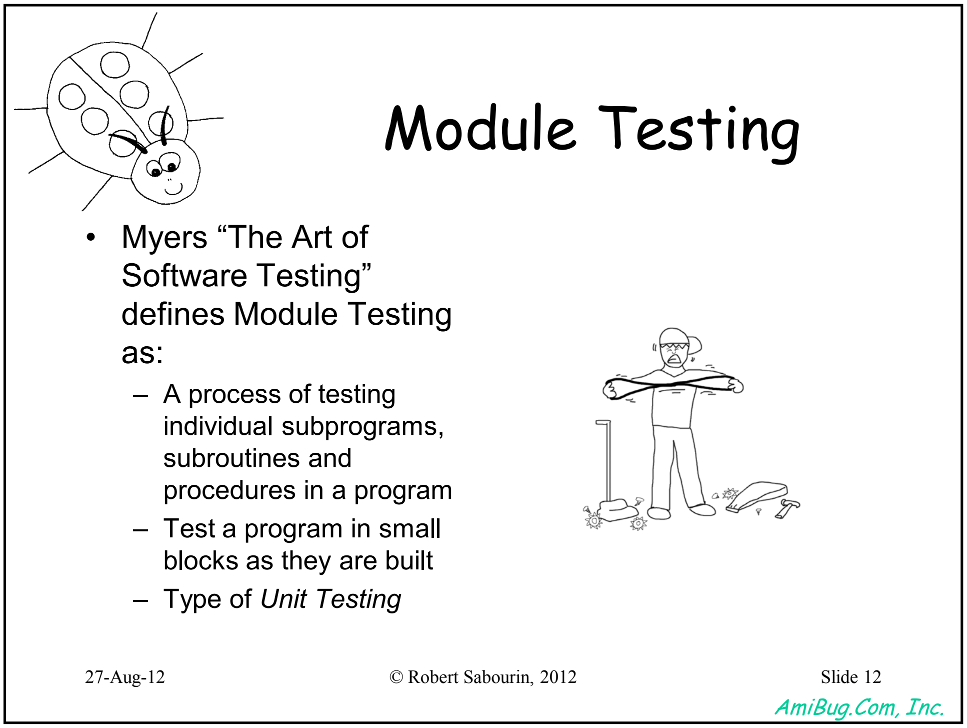


Unit Testing

- Testing of a unit of software as soon as this is available
- Unit is exercised against its detailed design
- Ensuring that developed logic is tested
- Often uses white box methods
- Done by developers at completion of task

27-Aug-12 © Robert Sabourin, 2012 Slide 11

AmiBug.Com, Inc.

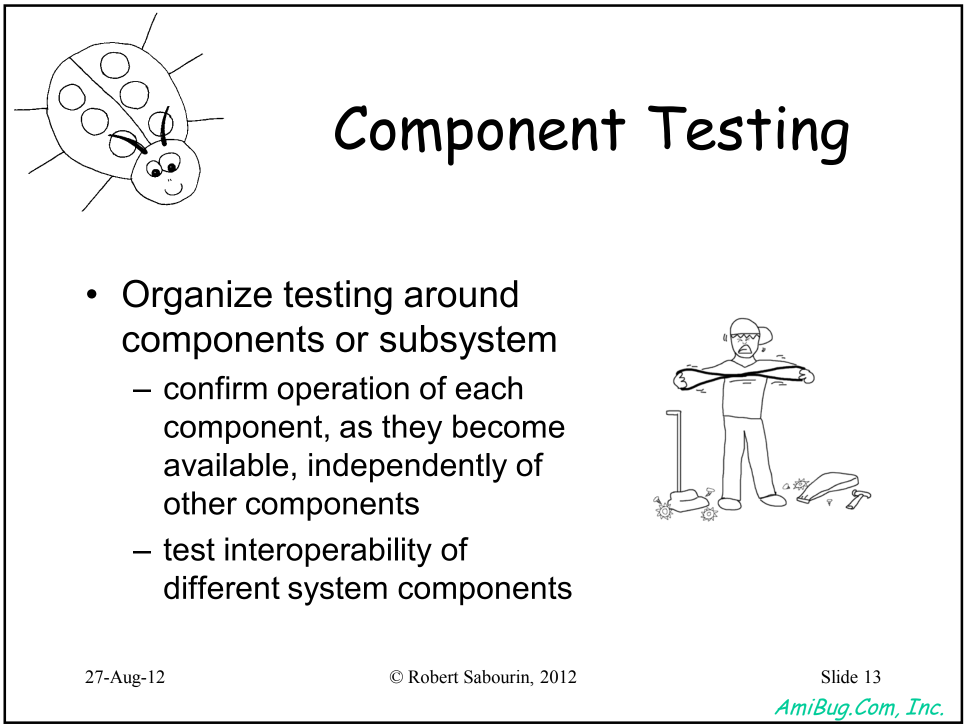


Module Testing

- Myers “The Art of Software Testing” defines Module Testing as:
  – A process of testing individual subprograms, subroutines and procedures in a program
  – Test a program in small blocks as they are built
  – Type of *Unit Testing*

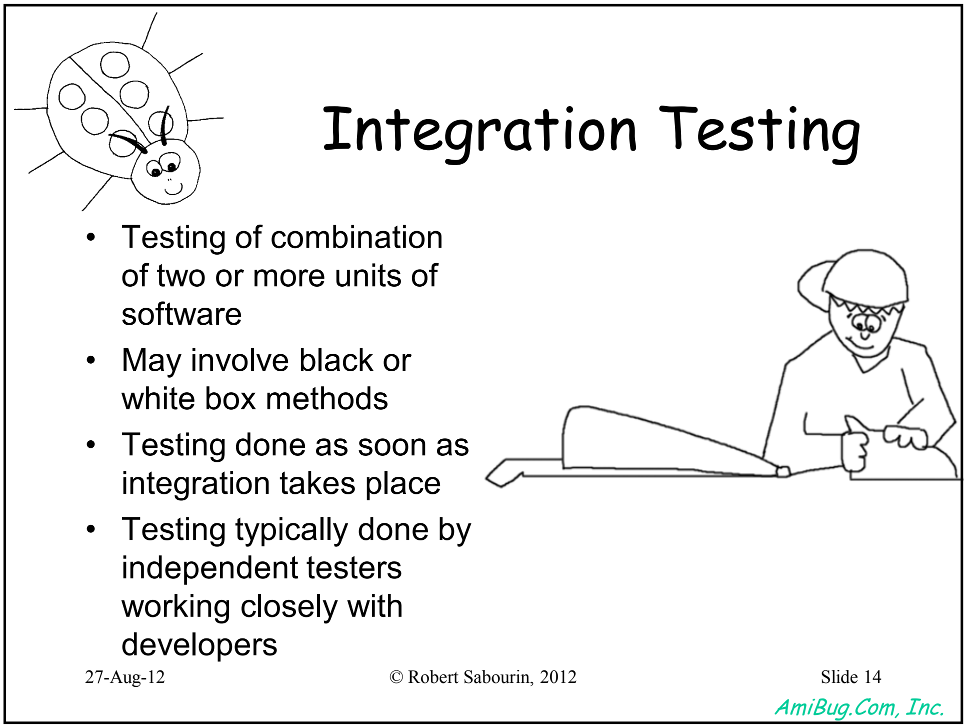
27-Aug-12 © Robert Sabourin, 2012 Slide 12

AmiBug.Com, Inc.



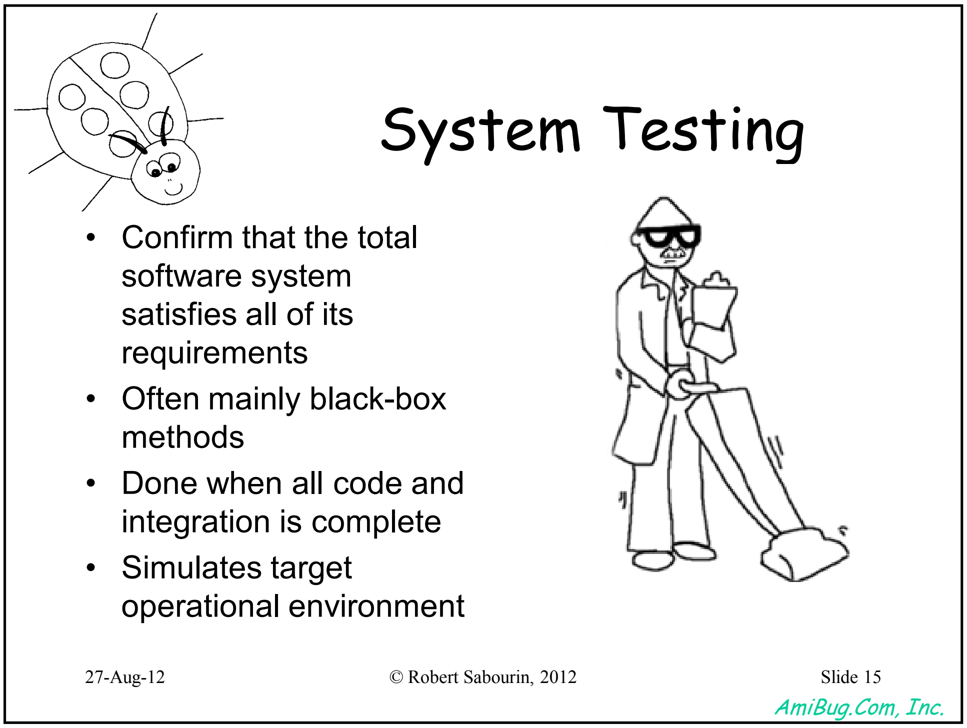
Component Testing

- Organize testing around components or subsystem
  - confirm operation of each component, as they become available, independently of other components
  - test interoperability of different system components



Integration Testing

- Testing of combination of two or more units of software
- May involve black or white box methods
- Testing done as soon as integration takes place
- Testing typically done by independent testers working closely with developers

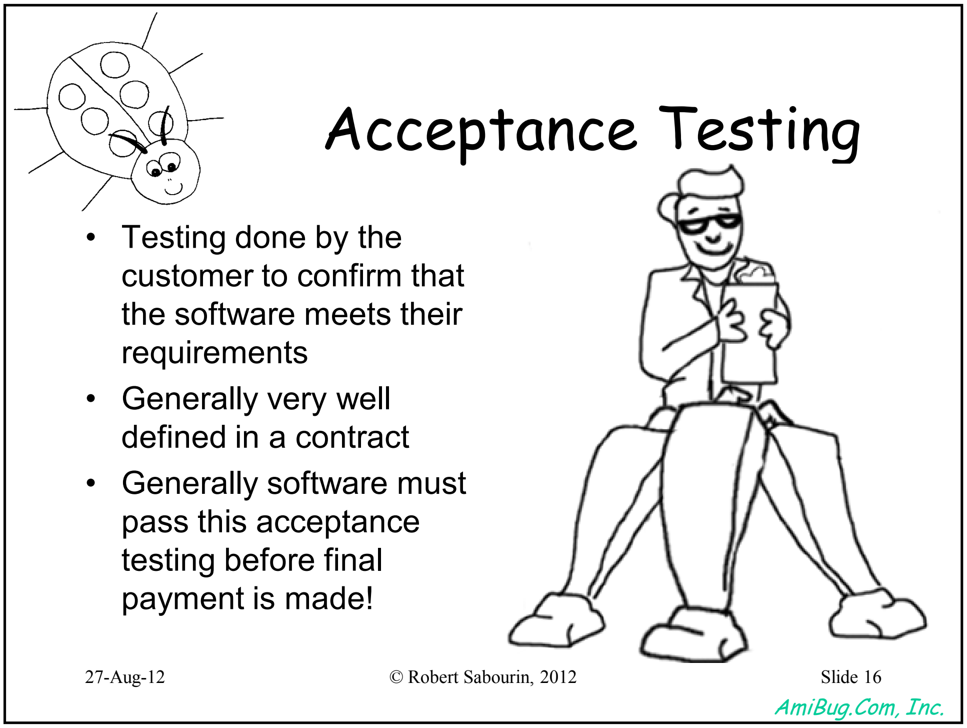


System Testing

- Confirm that the total software system satisfies all of its requirements
- Often mainly black-box methods
- Done when all code and integration is complete
- Simulates target operational environment

27-Aug-12 © Robert Sabourin, 2012 Slide 15

AmiBug.Com, Inc.

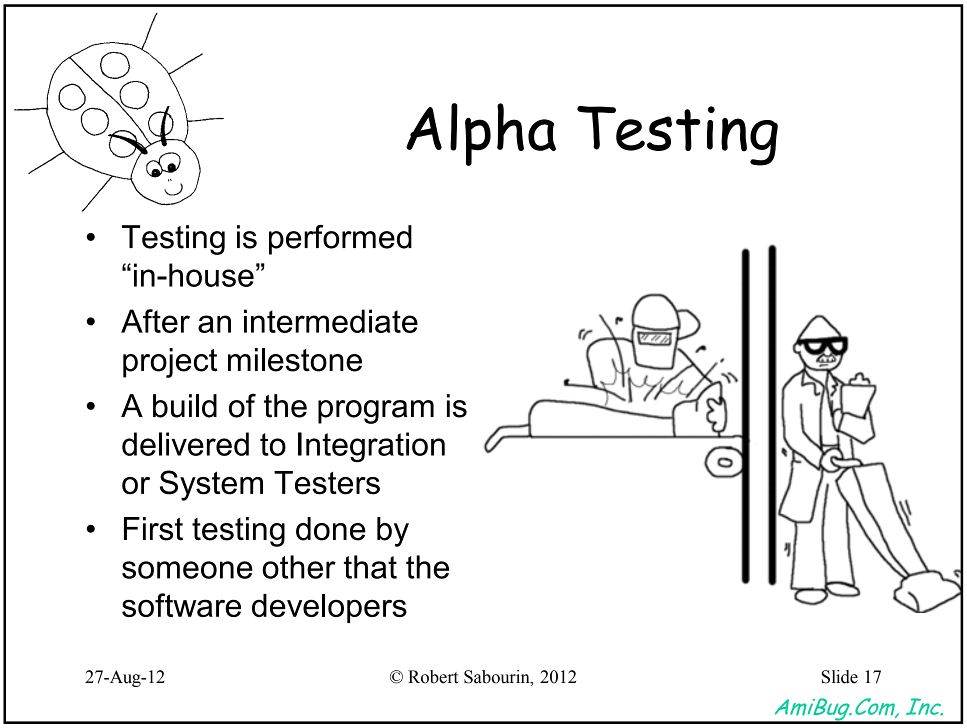


Acceptance Testing

- Testing done by the customer to confirm that the software meets their requirements
- Generally very well defined in a contract
- Generally software must pass this acceptance testing before final payment is made!

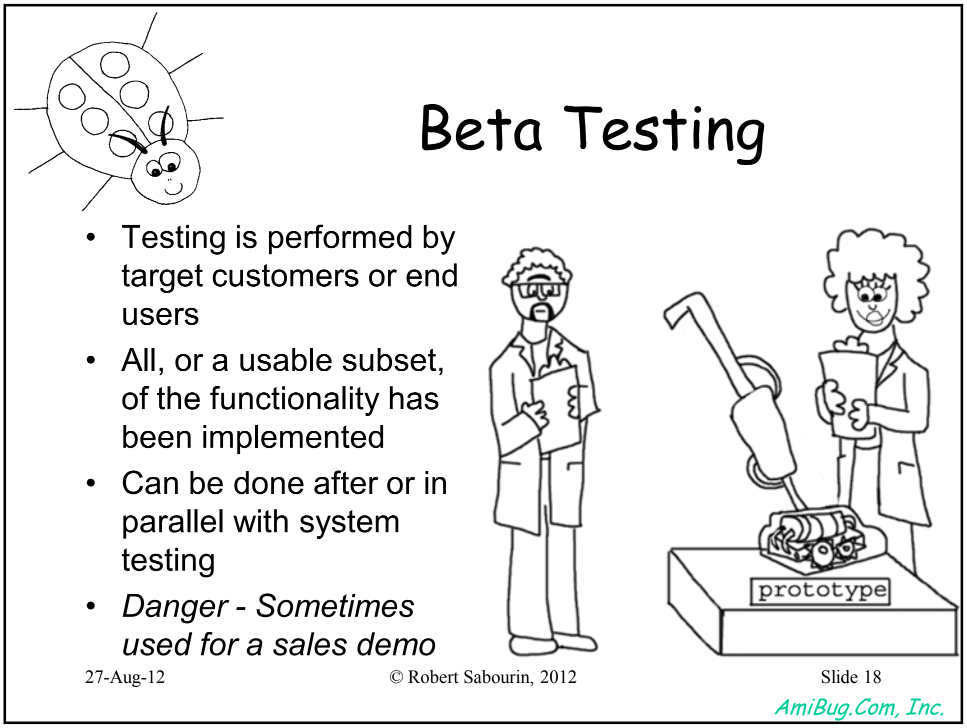
27-Aug-12 © Robert Sabourin, 2012 Slide 16

AmiBug.Com, Inc.



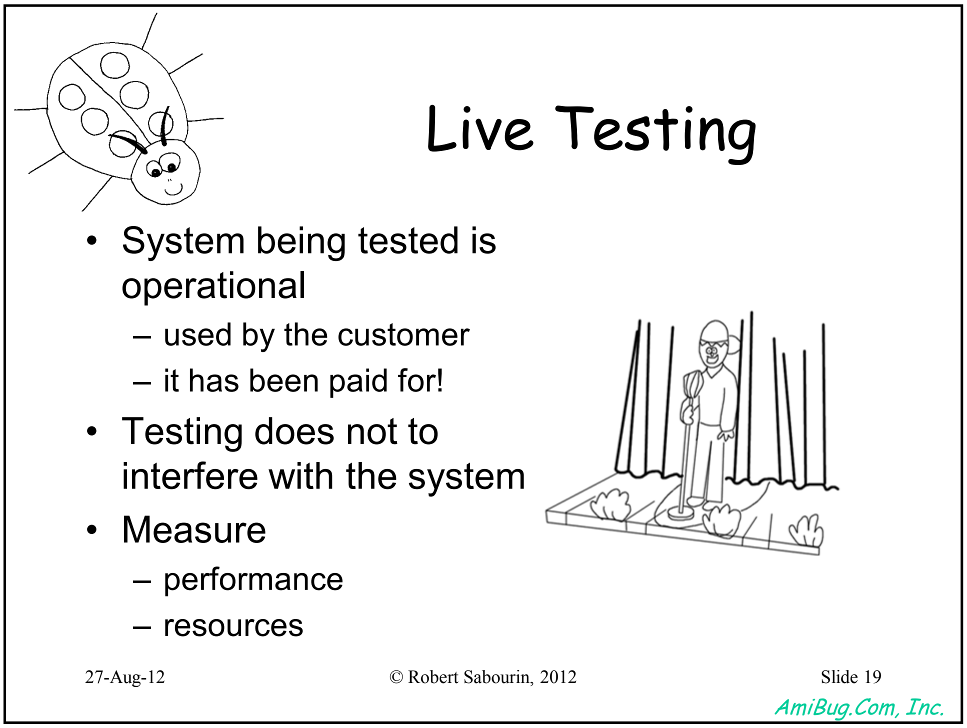
Alpha Testing

- Testing is performed “in-house”
- After an intermediate project milestone
- A build of the program is delivered to Integration or System Testers
- First testing done by someone other that the software developers



Beta Testing

- Testing is performed by target customers or end users
- All, or a usable subset, of the functionality has been implemented
- Can be done after or in parallel with system testing
- *Danger - Sometimes used for a sales demo*

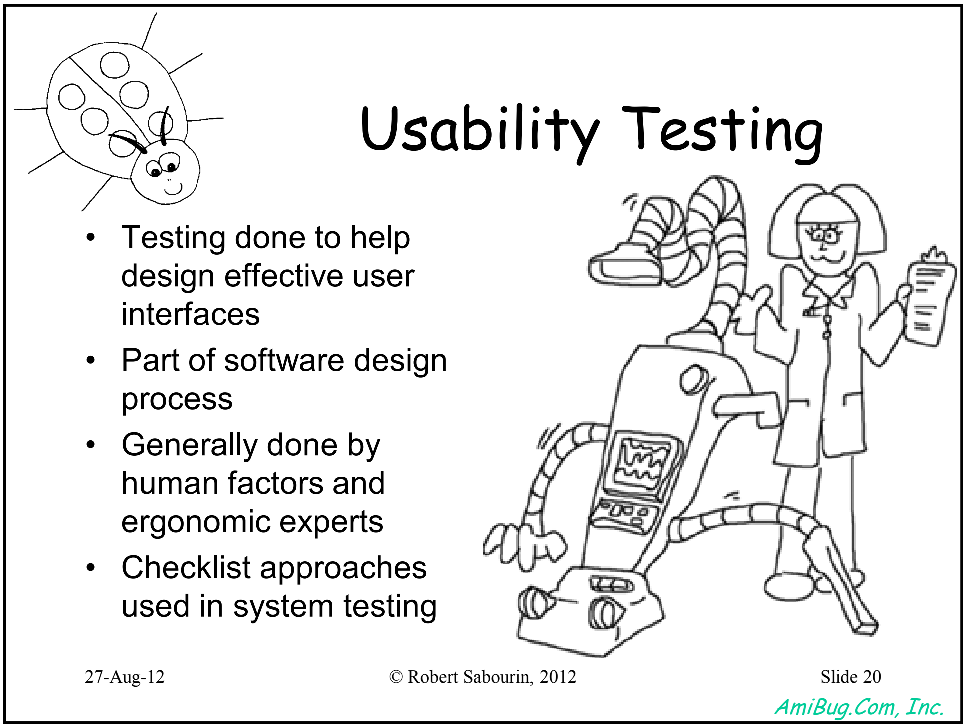


Live Testing

- System being tested is operational
  - used by the customer
  - it has been paid for!
- Testing does not to interfere with the system
- Measure
  - performance
  - resources

27-Aug-12 © Robert Sabourin, 2012 Slide 19

AmiBug.Com, Inc.



Usability Testing

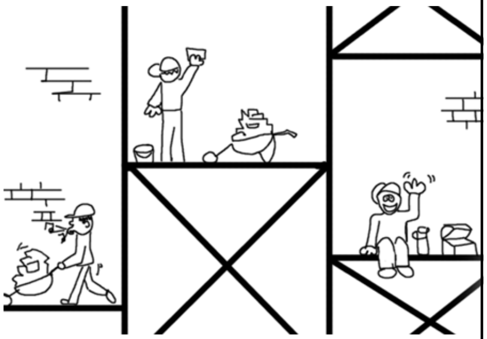
- Testing done to help design effective user interfaces
- Part of software design process
- Generally done by human factors and ergonomic experts
- Checklist approaches used in system testing

27-Aug-12 © Robert Sabourin, 2012 Slide 20

AmiBug.Com, Inc.

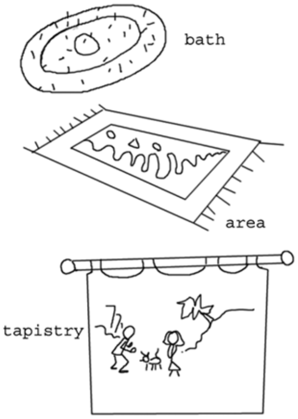
Platform / Configuration Testing

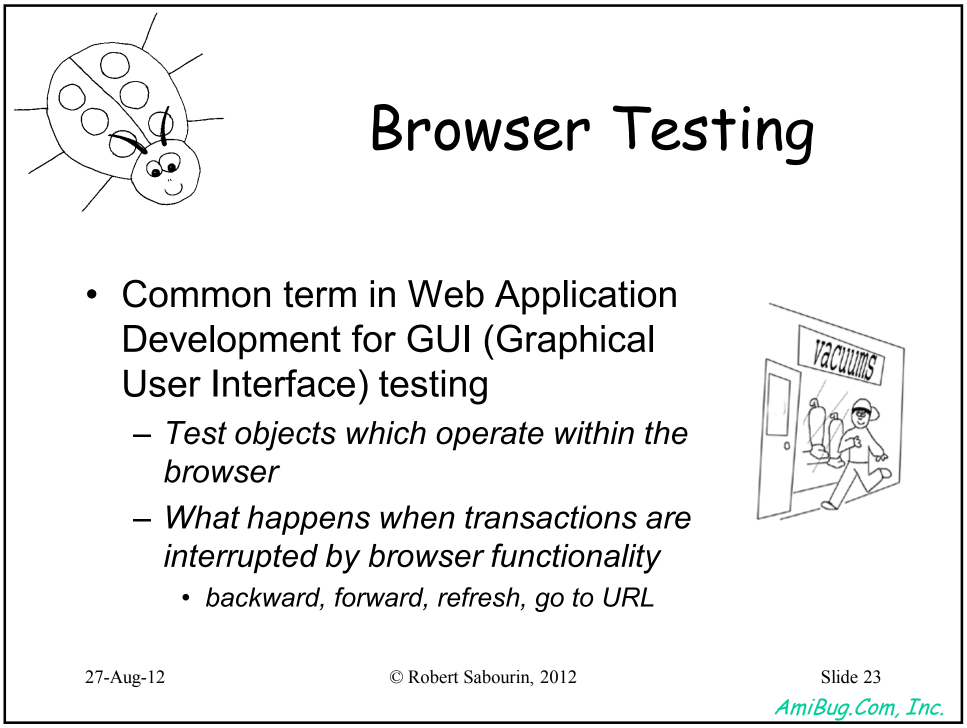
- Ensure functionality operates as required on different hardware and software configurations
- Different versions of operating systems
- Different locales
- Different versions of www browsers, plug-ins
- Different versions of any co-dependent software



Platform Specific Testing

- Testing included in the test plan related to the specific application targe platform!



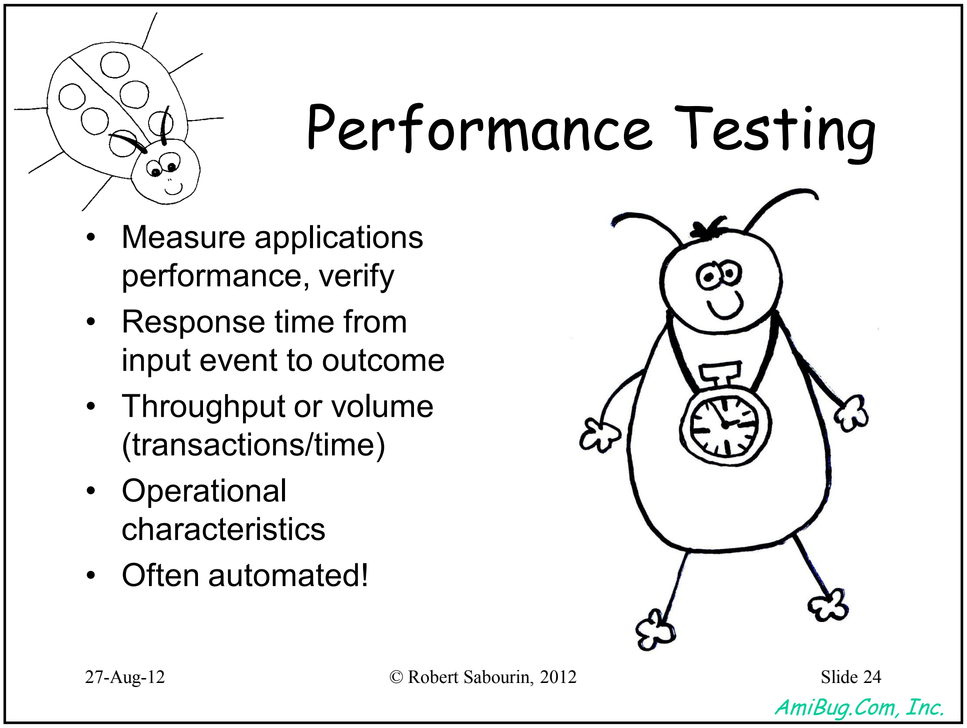


Browser Testing

- Common term in Web Application Development for GUI (Graphical User Interface) testing
  - *Test objects which operate within the browser*
  - *What happens when transactions are interrupted by browser functionality*
    - *backward, forward, refresh, go to URL*

27-Aug-12 © Robert Sabourin, 2012 Slide 23

AmiBug.Com, Inc.

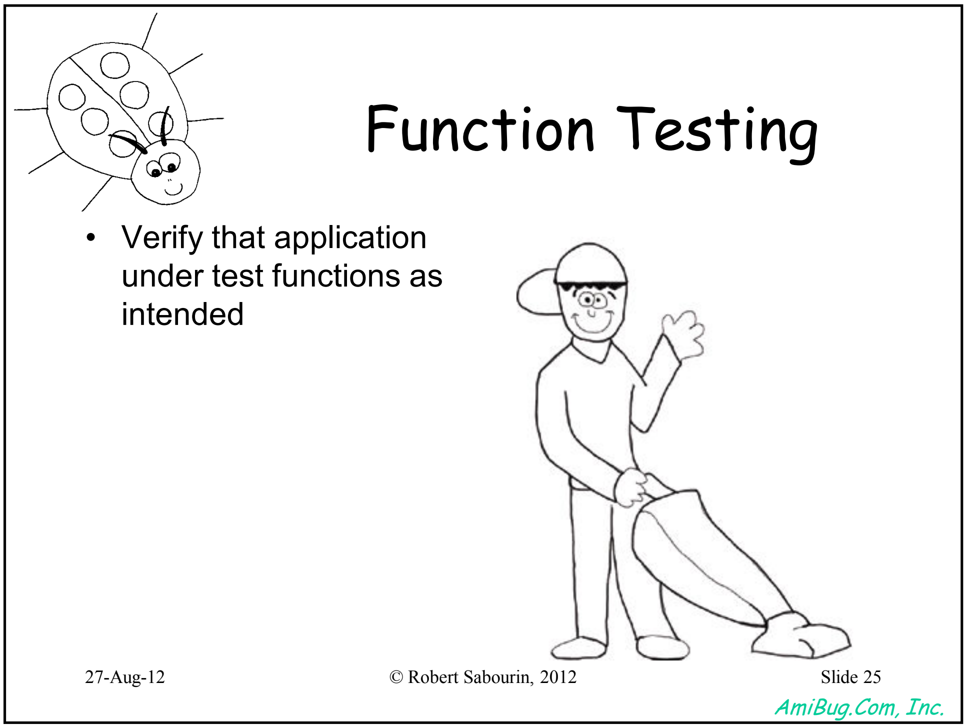


Performance Testing

- Measure applications performance, verify
- Response time from input event to outcome
- Throughput or volume (transactions/time)
- Operational characteristics
- Often automated!

27-Aug-12 © Robert Sabourin, 2012 Slide 24

AmiBug.Com, Inc.

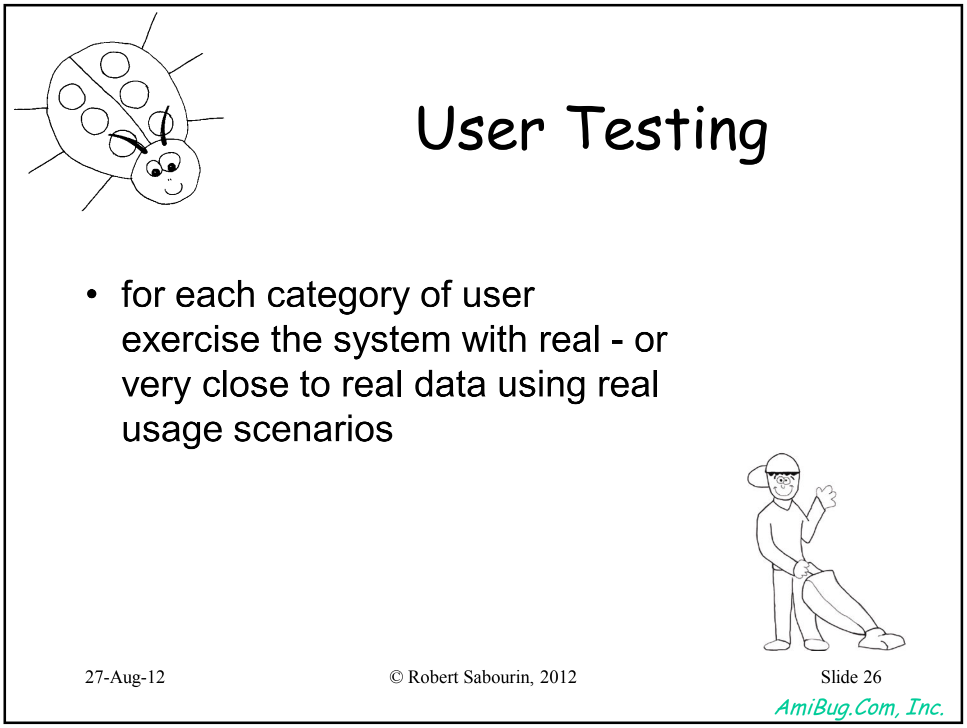


Function Testing

- Verify that application under test functions as intended

27-Aug-12 © Robert Sabourin, 2012 Slide 25

AmiBug.Com, Inc.

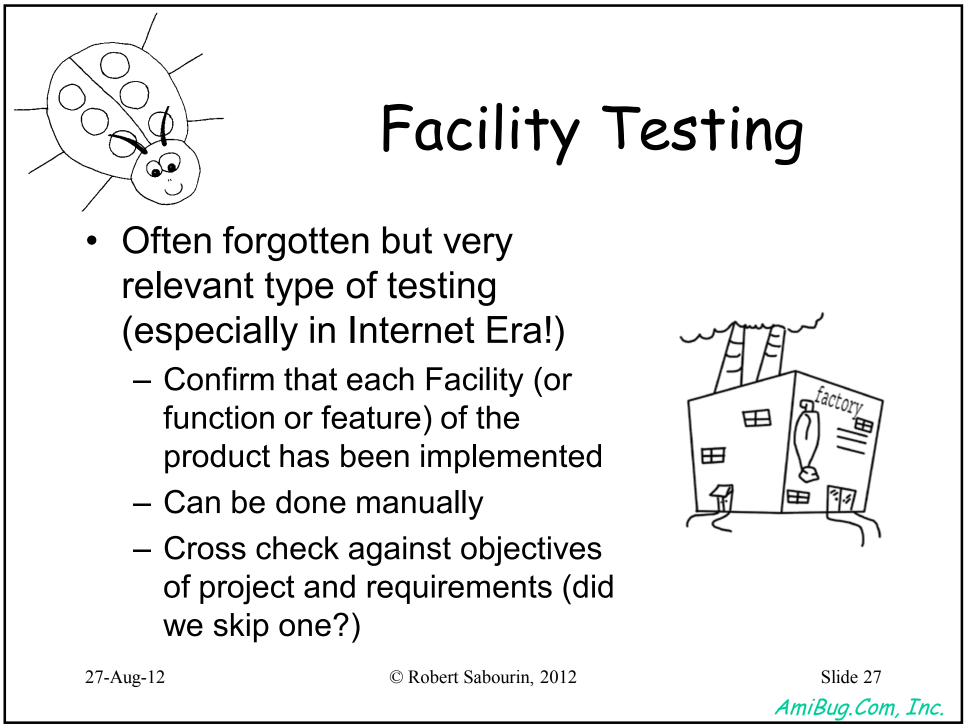


User Testing

- for each category of user exercise the system with real - or very close to real data using real usage scenarios

27-Aug-12 © Robert Sabourin, 2012 Slide 26

AmiBug.Com, Inc.

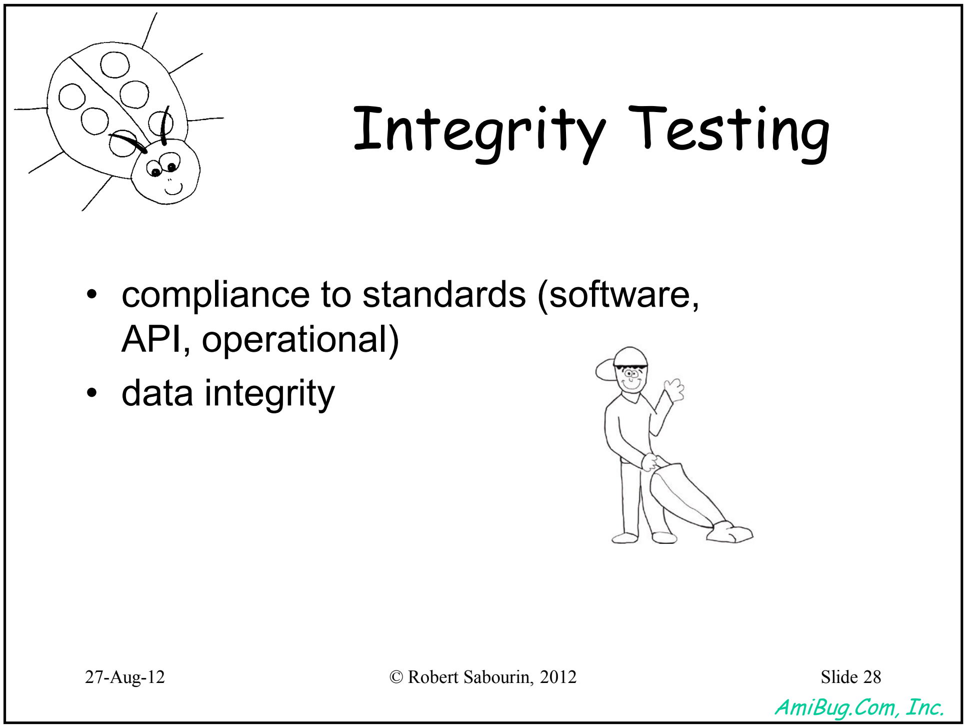


Facility Testing

- Often forgotten but very relevant type of testing (especially in Internet Era!)
  - – Confirm that each Facility (or function or feature) of the product has been implemented
  - – Can be done manually
  - – Cross check against objectives of project and requirements (did we skip one?)

27-Aug-12 © Robert Sabourin, 2012 Slide 27

AmiBug.Com, Inc.

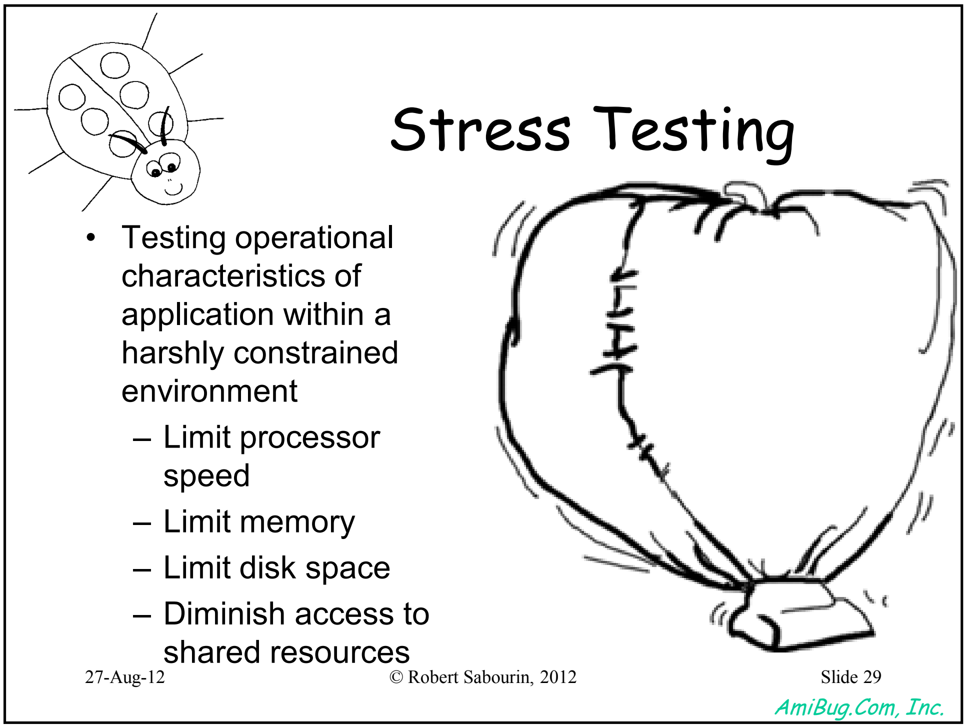


Integrity Testing

- compliance to standards (software, API, operational)
- data integrity

27-Aug-12 © Robert Sabourin, 2012 Slide 28

AmiBug.Com, Inc.

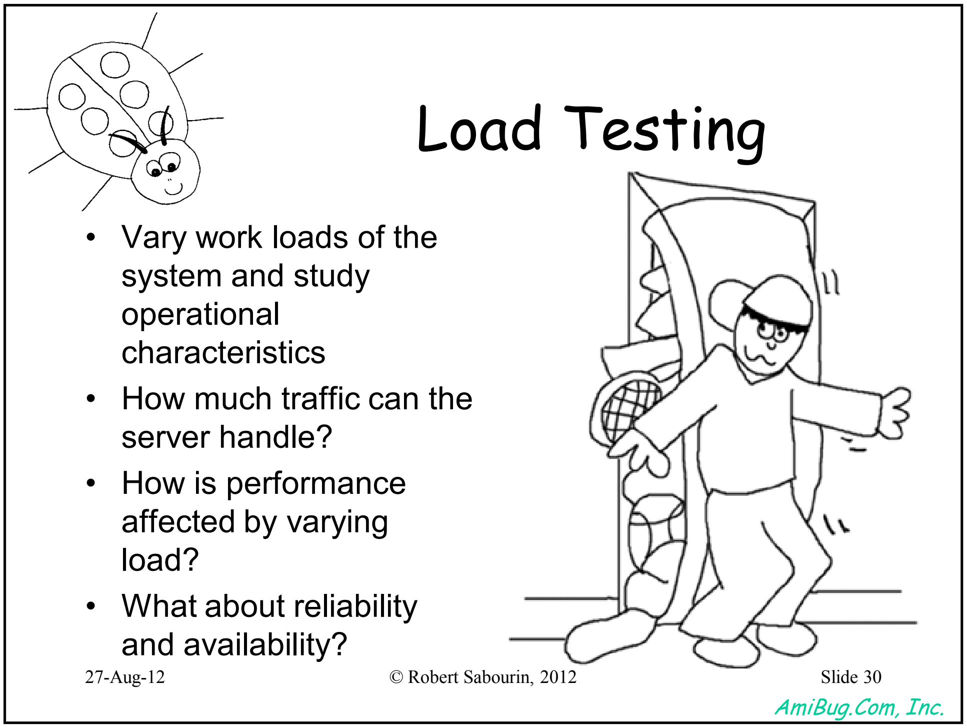


Stress Testing

- Testing operational characteristics of application within a harshly constrained environment
  - Limit processor speed
  - Limit memory
  - Limit disk space
  - Diminish access to shared resources

27-Aug-12 © Robert Sabourin, 2012 Slide 29

AmiBug.Com, Inc.



Load Testing

- Vary work loads of the system and study operational characteristics
- How much traffic can the server handle?
- How is performance affected by varying load?
- What about reliability and availability?

27-Aug-12 © Robert Sabourin, 2012 Slide 30

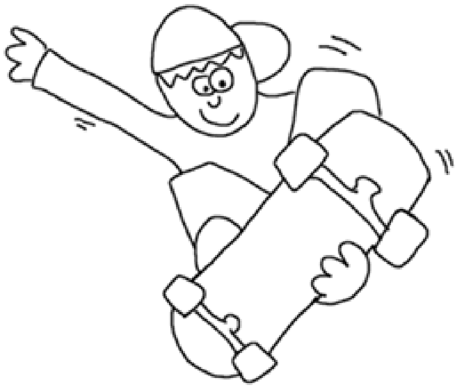
AmiBug.Com, Inc.

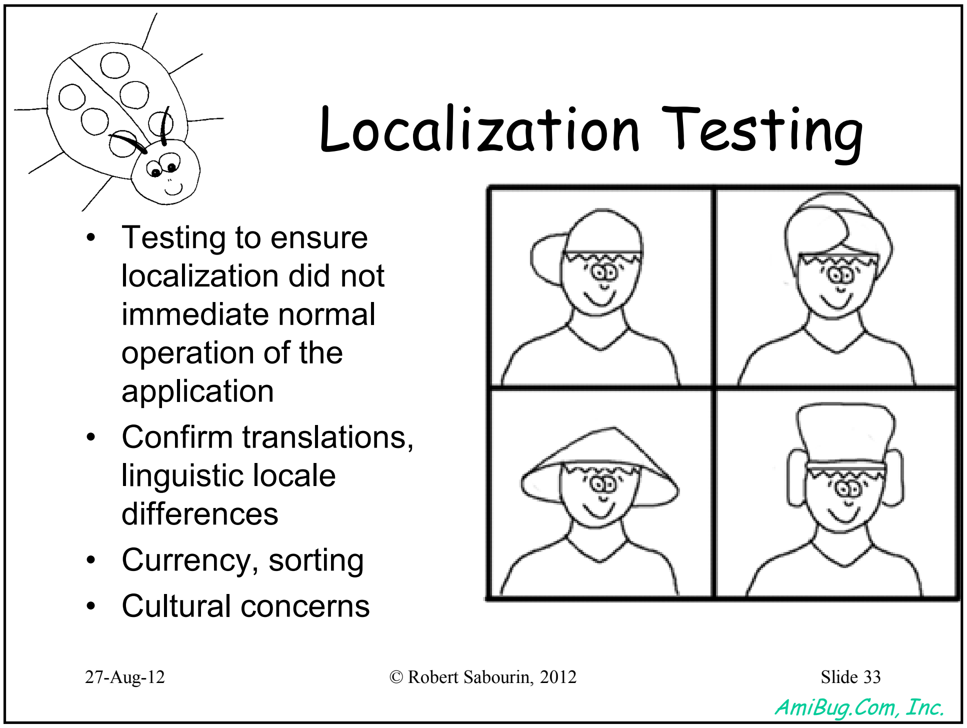
Volume Testing

- Type of load testing in which large amounts of data are processed through the system
- Study behavior of system under test when experiencing extreme processing demands
- Generally automated!

Extreme Testing

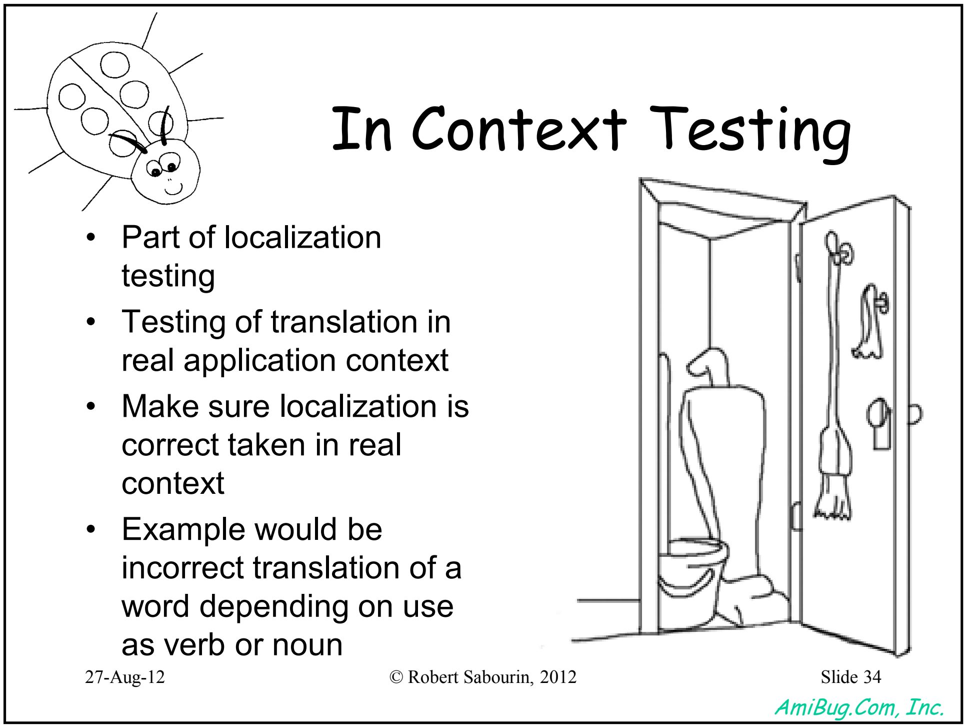
- Testing done as part of an extreme software development process
- Testing is against story board scenarios
- Tightly coupled with development (one iteration at a time)
- Interact with customer of project for acceptance testing





Localization Testing

- Testing to ensure localization did not immediate normal operation of the application
- Confirm translations, linguistic locale differences
- Currency, sorting
- Cultural concerns

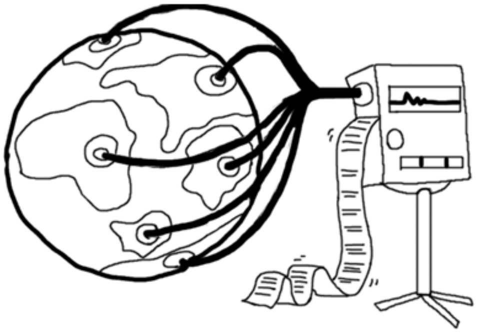


In Context Testing

- Part of localization testing
- Testing of translation in real application context
- Make sure localization is correct taken in real context
- Example would be incorrect translation of a word depending on use as verb or noun

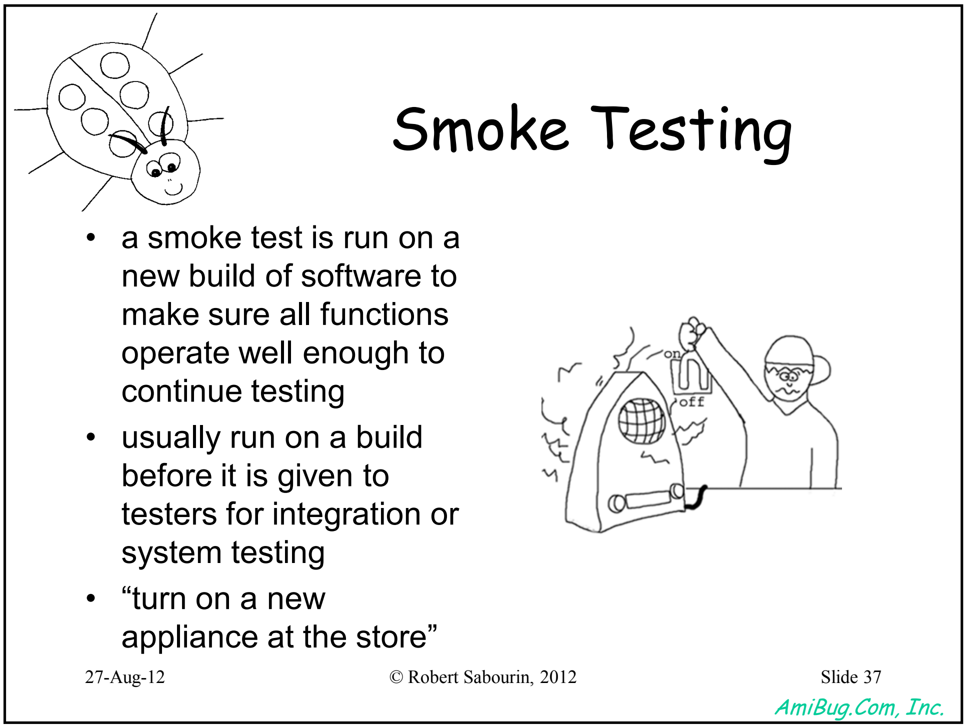
Internationalization Testing

- An internationalized program is able to be localized to operate with many different languages and data representations including those using multi-byte character sets
- Does internationalized software still behave as it did originally?



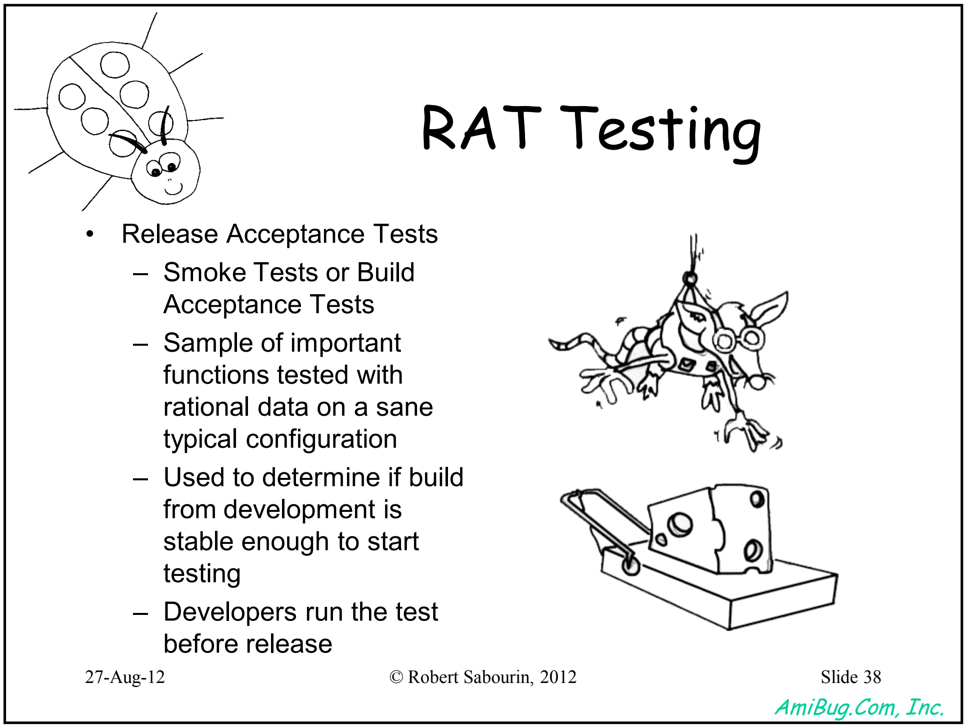
Regression Testing

- previously executed tests are re-executed against a new version of the application
  - have code changes broken something tha used to work
  - have we introduced new defects
  - typically first part of a testing iteration
  - often automated



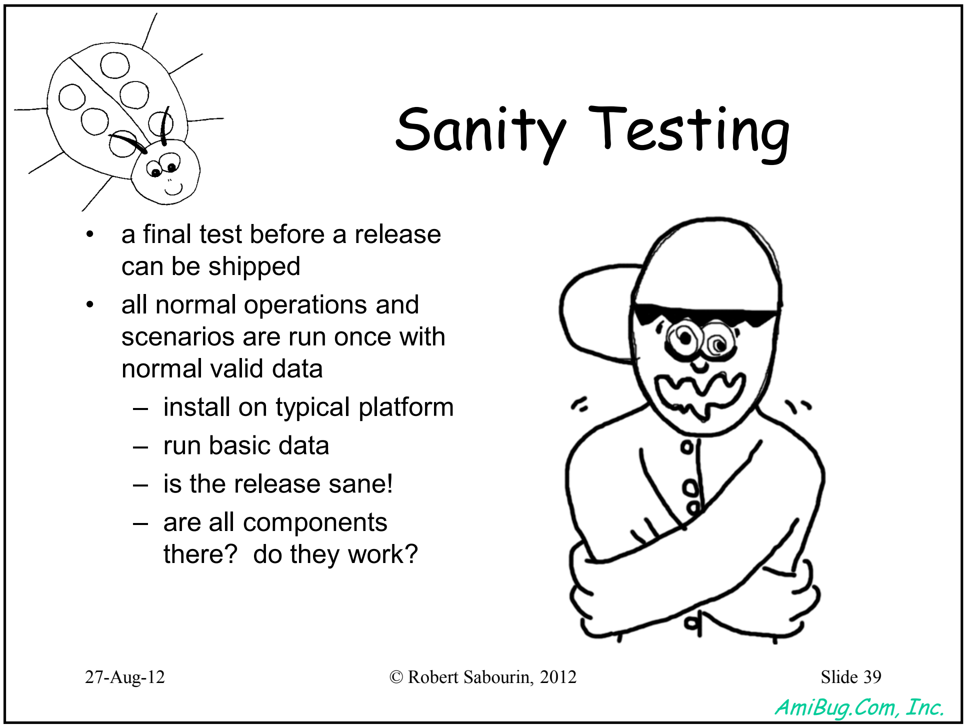
Smoke Testing

- a smoke test is run on a new build of software to make sure all functions operate well enough to continue testing
- usually run on a build before it is given to testers for integration or system testing
- “turn on a new appliance at the store”



RAT Testing

- Release Acceptance Tests
  - Smoke Tests or Build Acceptance Tests
  - Sample of important functions tested with rational data on a sane typical configuration
  - Used to determine if build from development is stable enough to start testing
  - Developers run the test before release

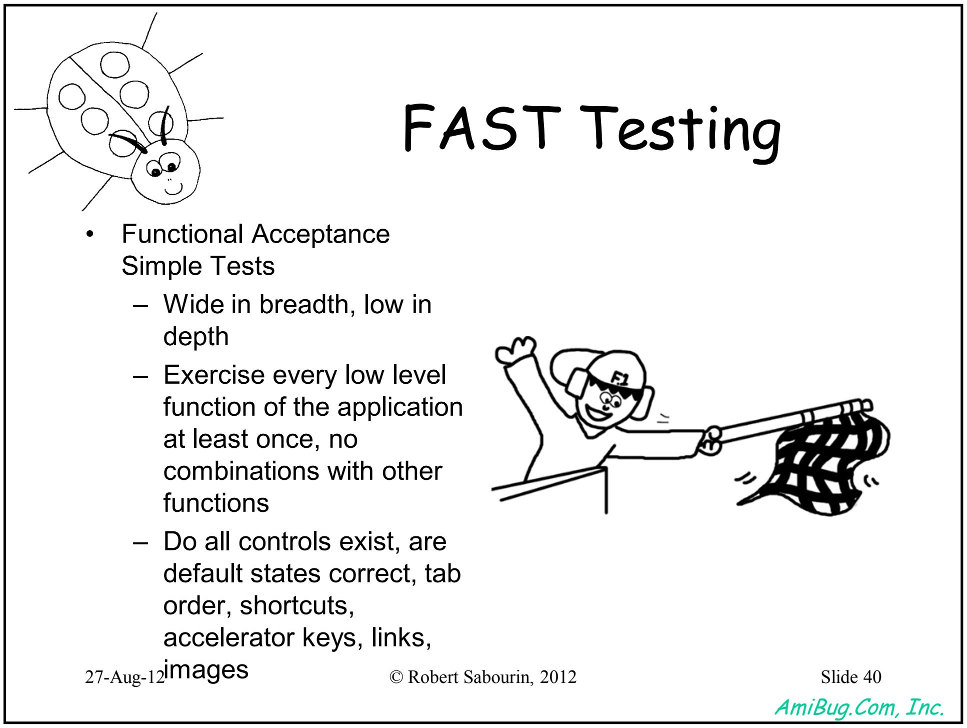


Sanity Testing

- a final test before a release can be shipped
- all normal operations and scenarios are run once with normal valid data
  - install on typical platform
  - run basic data
  - is the release sane!
  - are all components there? do they work?

27-Aug-12 © Robert Sabourin, 2012 Slide 39

AmiBug.Com, Inc.

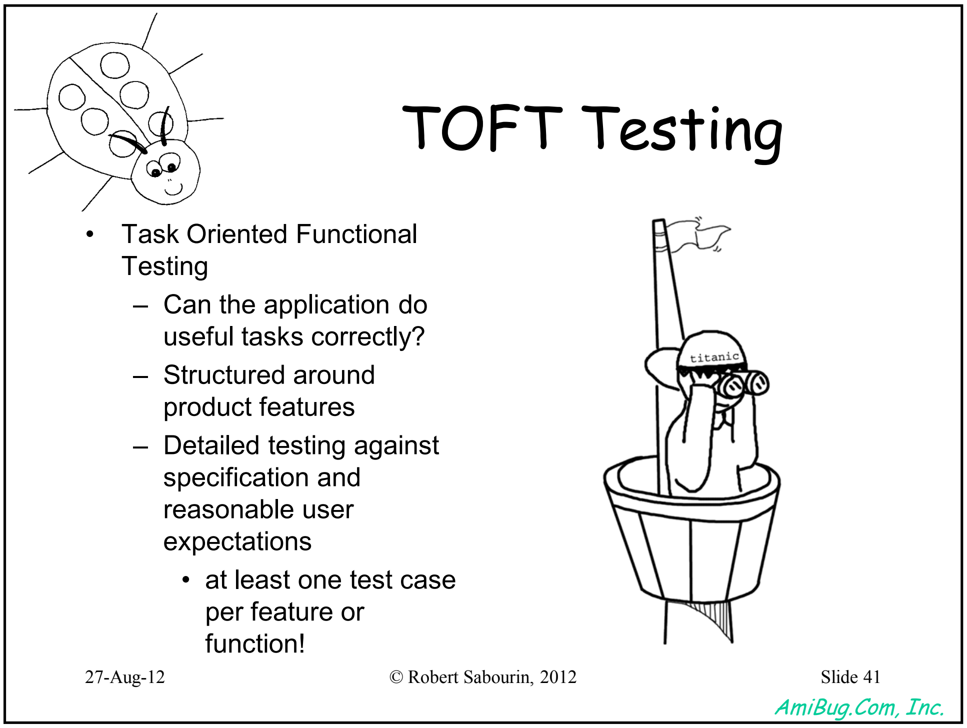


FAST Testing

- Functional Acceptance Simple Tests
  - Wide in breadth, low in depth
  - Exercise every low level function of the application at least once, no combinations with other functions
  - Do all controls exist, are default states correct, tab order, shortcuts, accelerator keys, links, images

27-Aug-12 © Robert Sabourin, 2012 Slide 40

AmiBug.Com, Inc.

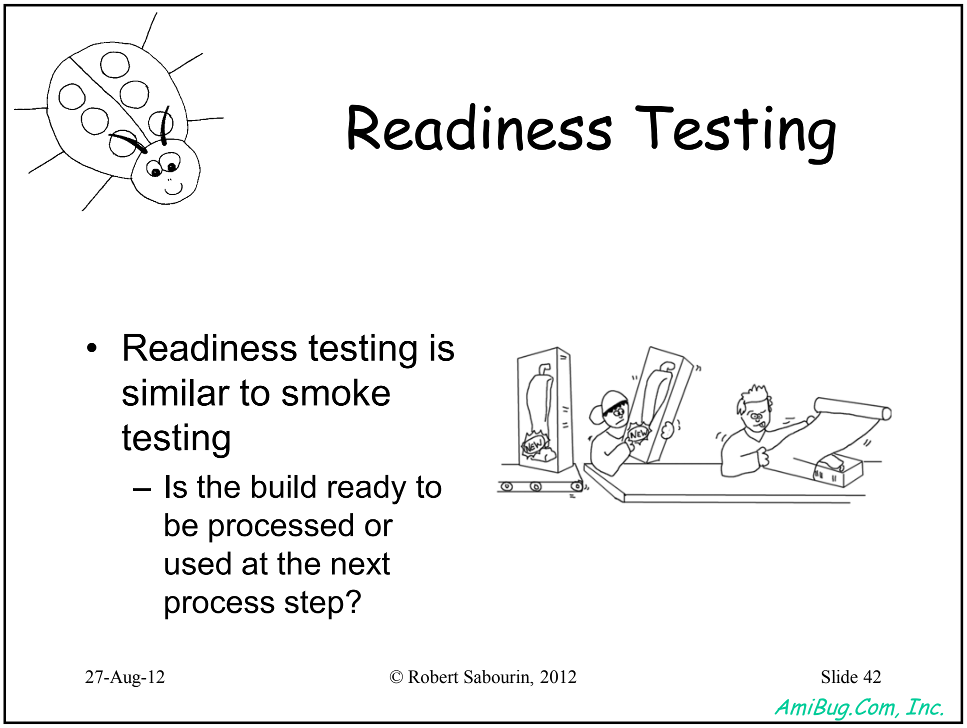


TOFT Testing

- Task Oriented Functional Testing
  - Can the application do useful tasks correctly?
  - Structured around product features
  - Detailed testing against specification and reasonable user expectations
    - at least one test case per feature or function!

27-Aug-12 © Robert Sabourin, 2012 Slide 41

AmiBug.Com, Inc.

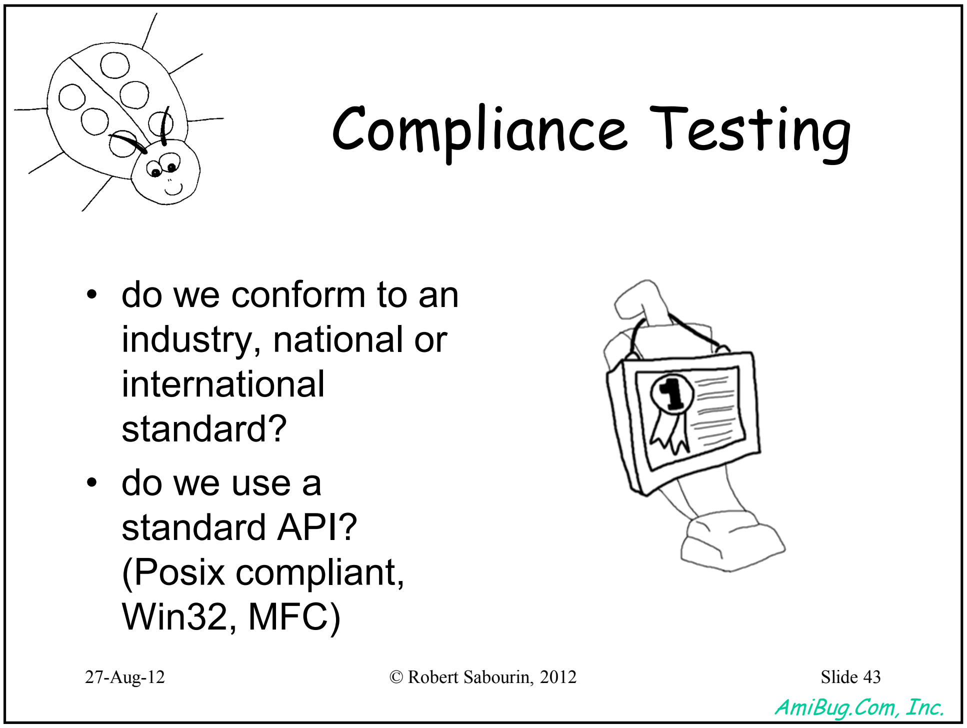


Readiness Testing

- Readiness testing is similar to smoke testing
  - Is the build ready to be processed or used at the next process step?

27-Aug-12 © Robert Sabourin, 2012 Slide 42

AmiBug.Com, Inc.

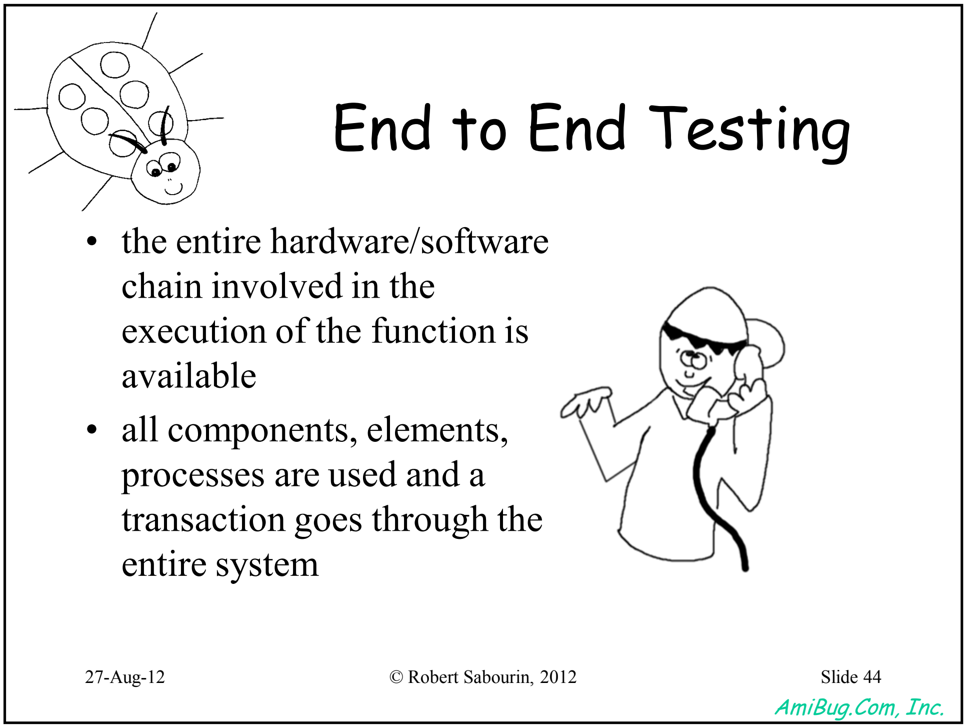


Compliance Testing

- do we conform to an industry, national or international standard?
- do we use a standard API? (Posix compliant, Win32, MFC)

27-Aug-12 © Robert Sabourin, 2012 Slide 43

AmiBug.Com, Inc.

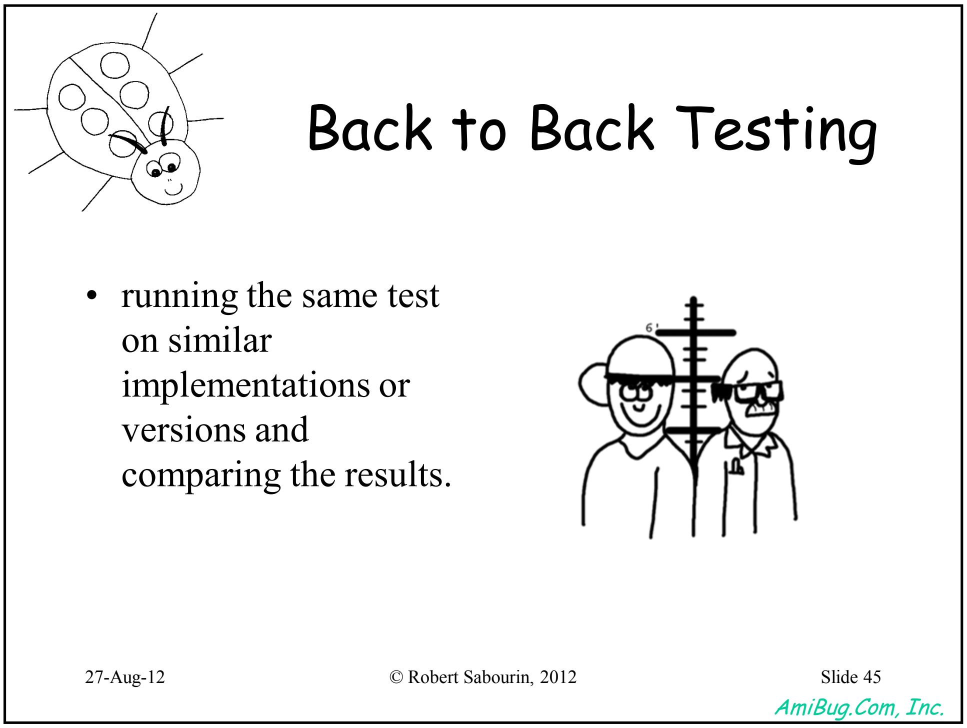


End to End Testing

- the entire hardware/software chain involved in the execution of the function is available
- all components, elements, processes are used and a transaction goes through the entire system

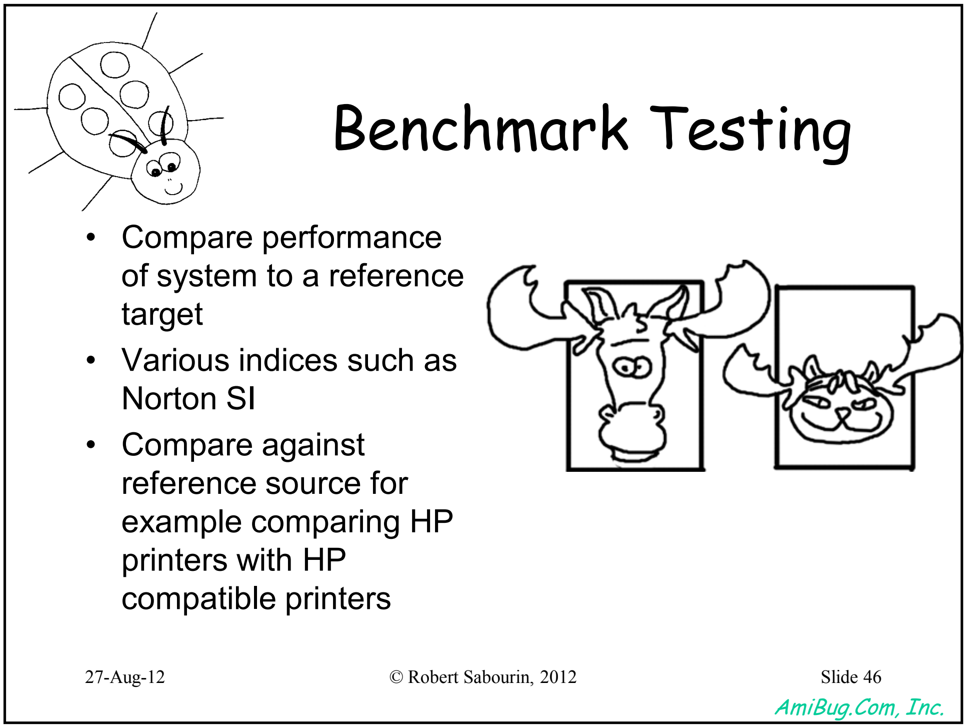
27-Aug-12 © Robert Sabourin, 2012 Slide 44

AmiBug.Com, Inc.



Back to Back Testing

- running the same test on similar implementations or versions and comparing the results.

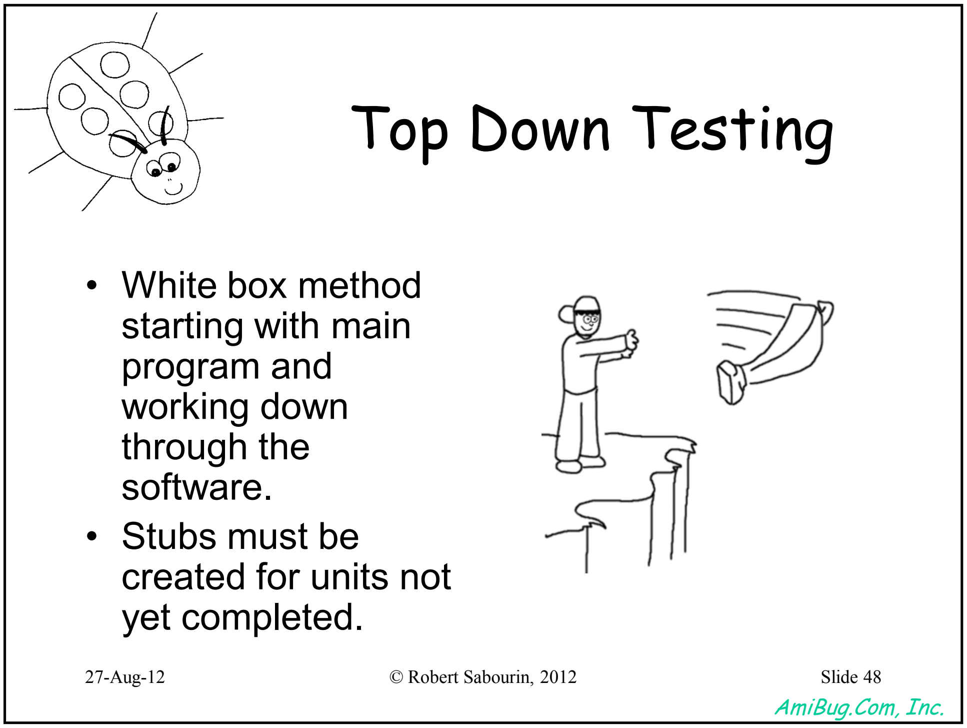


Benchmark Testing

- Compare performance of system to a reference target
- Various indices such as Norton SI
- Compare against reference source for example comparing HP printers with HP compatible printers

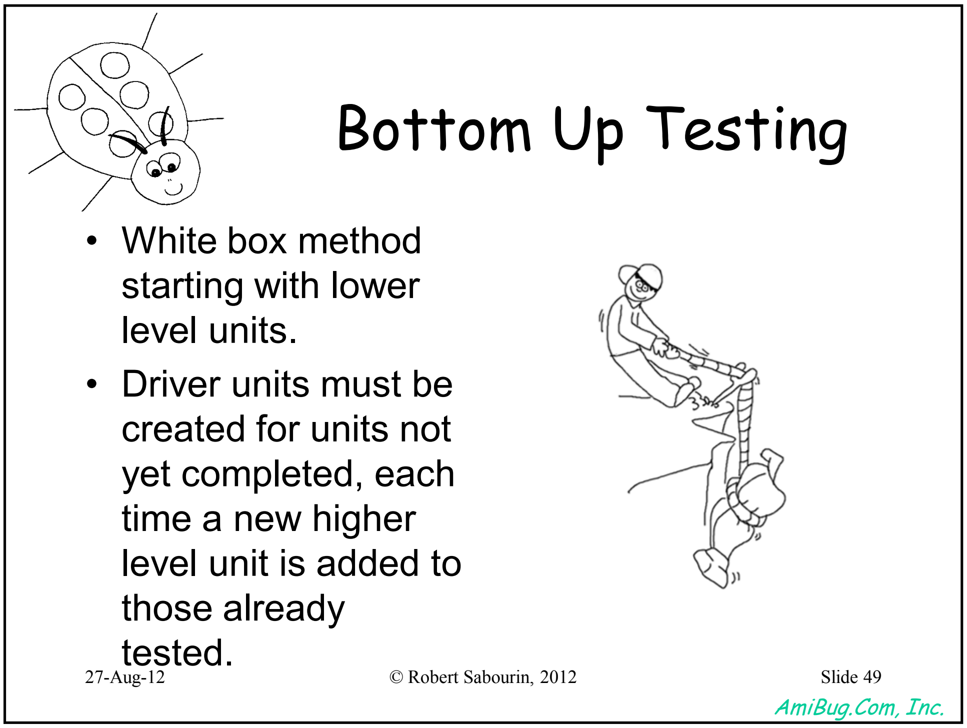
Certification/Branding Testing

- third party testing done to confirm that the application conforms to criteria for certification or branding program
  - Microsoft Windows certification



Top Down Testing

- White box method starting with main program and working down through the software.
- Stubs must be created for units not yet completed.

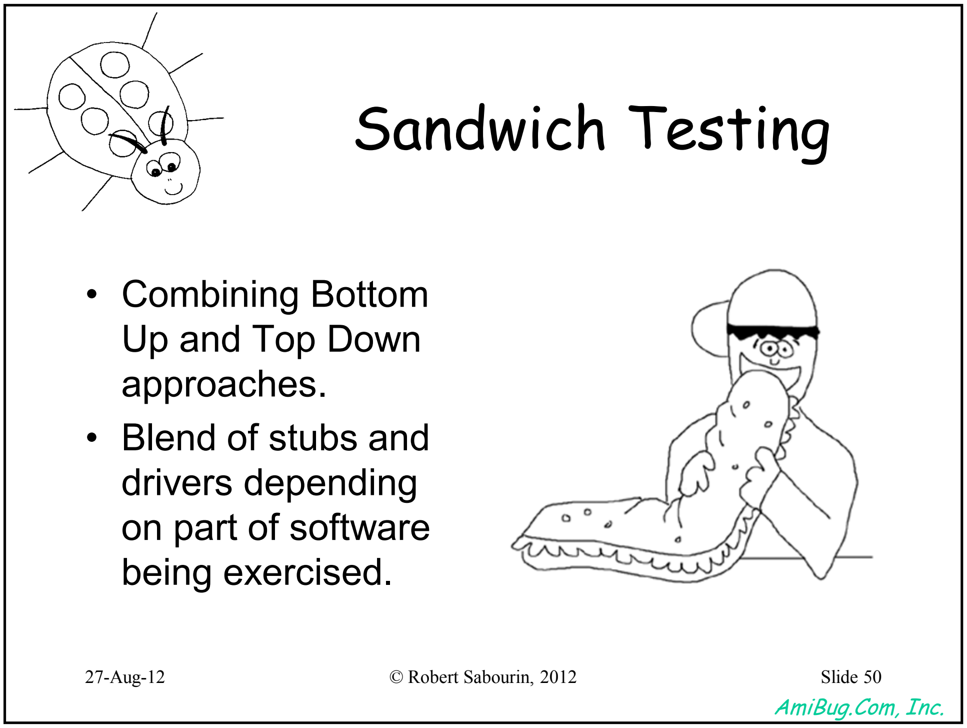


Bottom Up Testing

- White box method starting with lower level units.
- Driver units must be created for units not yet completed, each time a new higher level unit is added to those already tested.

27-Aug-12 © Robert Sabourin, 2012 Slide 49

AmiBug.Com, Inc.

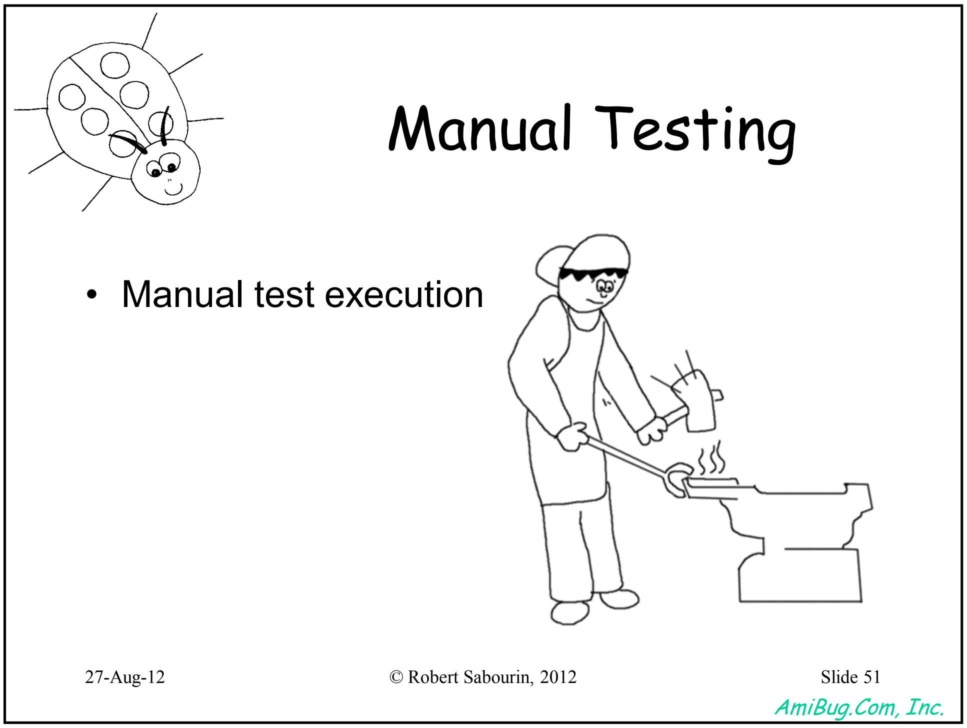


Sandwich Testing

- Combining Bottom Up and Top Down approaches.
- Blend of stubs and drivers depending on part of software being exercised.

27-Aug-12 © Robert Sabourin, 2012 Slide 50

AmiBug.Com, Inc.

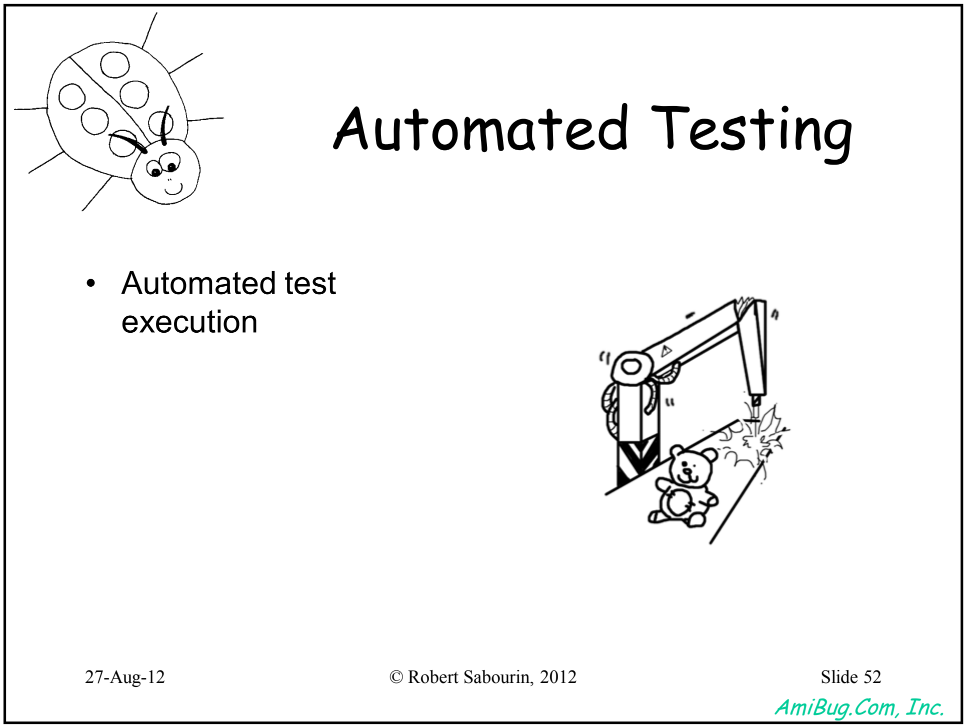


Manual Testing

- Manual test execution

27-Aug-12 © Robert Sabourin, 2012 Slide 51

AmiBug.Com, Inc.

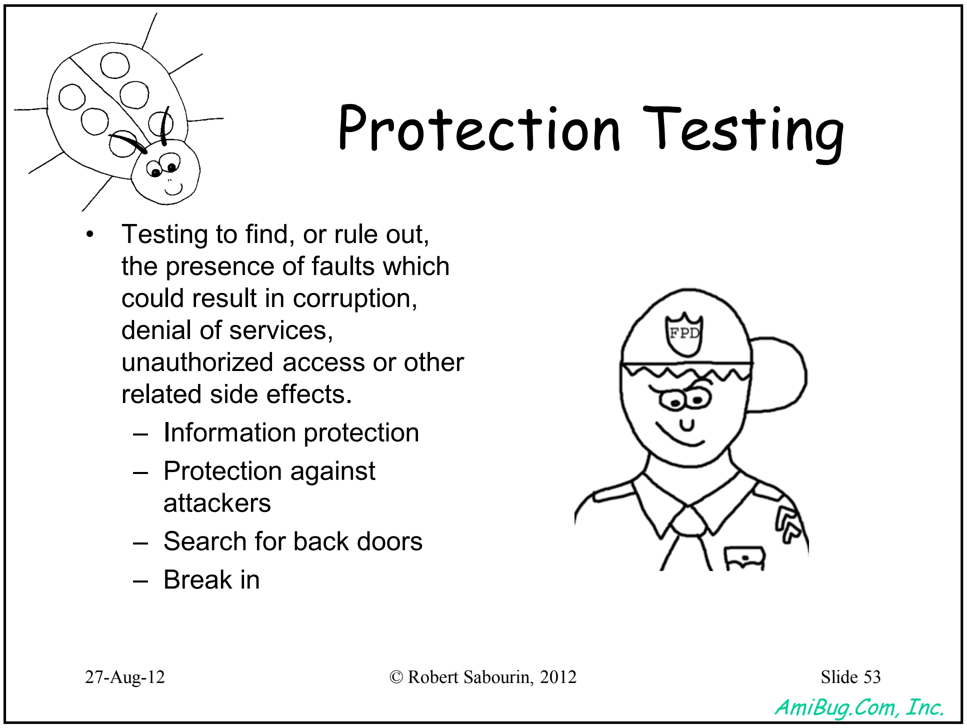


Automated Testing

- Automated test execution

27-Aug-12 © Robert Sabourin, 2012 Slide 52

AmiBug.Com, Inc.

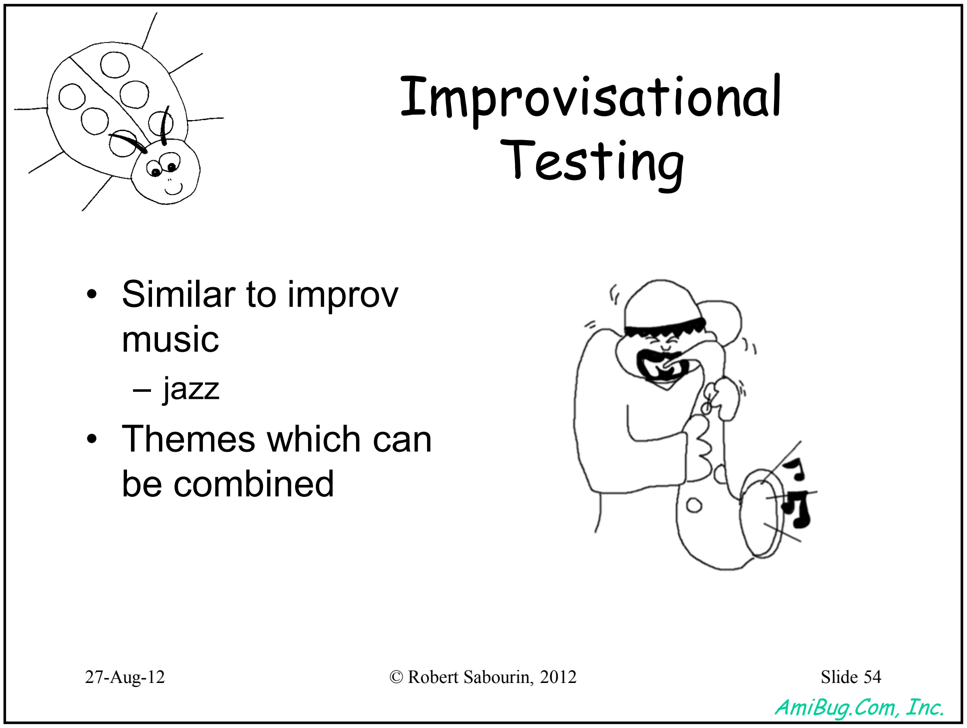


Protection Testing

- Testing to find, or rule out, the presence of faults which could result in corruption, denial of services, unauthorized access or other related side effects.
  - Information protection
  - Protection against attackers
  - Search for back doors
  - Break in

27-Aug-12 © Robert Sabourin, 2012 Slide 53

AmiBug.Com, Inc.



Improvisational Testing

- Similar to improv music
  - jazz
- Themes which can be combined

27-Aug-12 © Robert Sabourin, 2012 Slide 54

AmiBug.Com, Inc.

Clean Room Testing

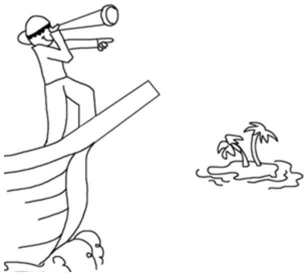
- Testing done as part of a clean room software engineering process
- High reliability
- Statistical analysis to determine test cases by sampling set of possible input data and conditions
- Analytic technique

Analytic Testing

- Test planning based on a detailed analysis of all application functions and operations
- Test cases derived from a detailed analysis of the technical specification

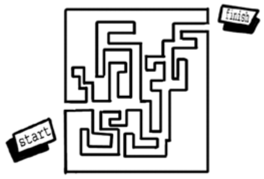
Exploratory Testing

- Concurrent
  - testing
  - test design
  - test planning



Ad-hoc Testing

- Random
- Not systematic

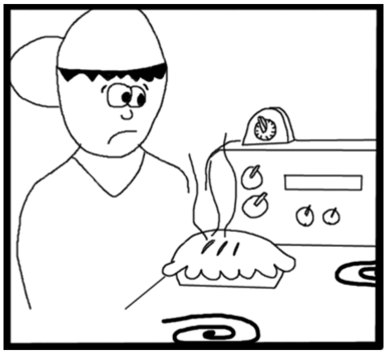


Heuristic Testing

- Testing using heuristic reasoning and taking advantage of the technical insight, critical thinking and experience of the tester
- Heuristic is defined as:
  - of or relating to exploratory problem solving techniques that use self-education

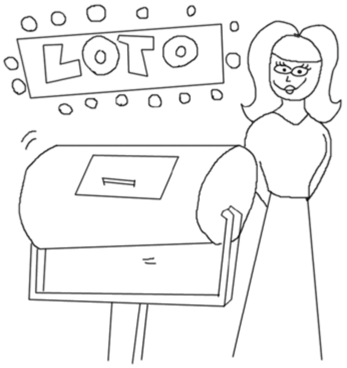
Event Driven Testing

- A type of automated testing
- Test script execution is triggered by an external event or interrupt
- Run test A when event B occurs
  - *Web Site Monitors*
  - *Run consistency check if DBMS is 80% full*



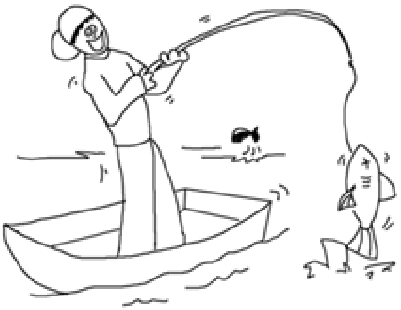
Data Driven Testing

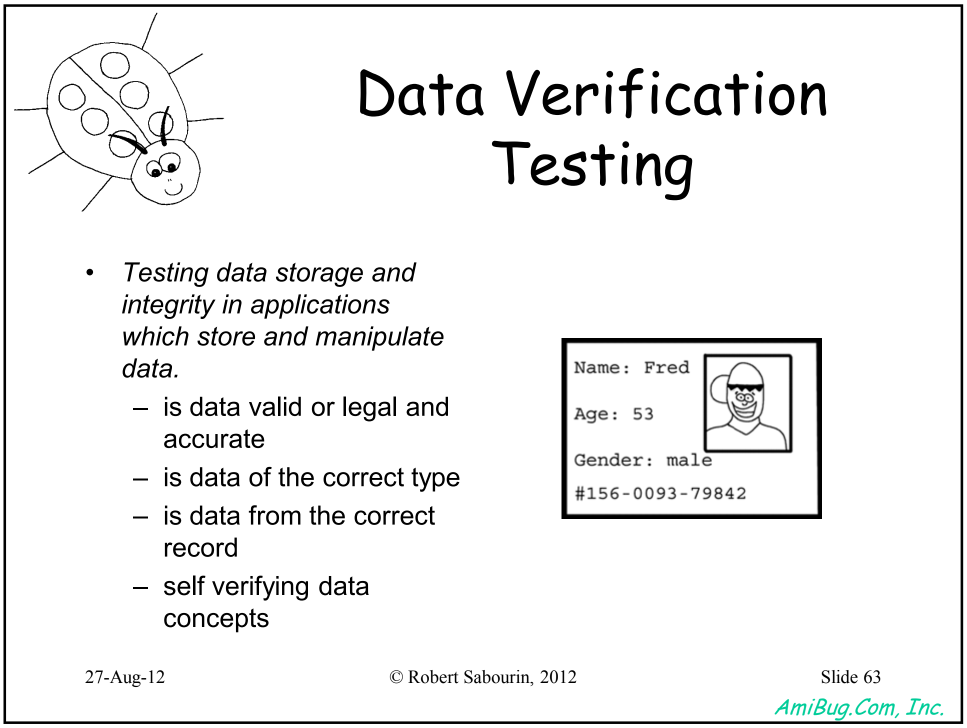
- Test script or procedure in which *test data* is separated from *test operations or actions*
  - parameterized data
  - one set of data per test case
  - run the same test script with different data!
  - data is deliberate, not arbitrary
  - testing is repeatable
  - test can be automated



Capture Playback Testing

- *Tool* is used to record all input events of a simulated user
- Test case is automated by replaying all input events
- Difficult to maintain!
- Can be used as a first step in designing a data driven test automation script!





Data Verification Testing

- *Testing data storage and integrity in applications which store and manipulate data.*
  - is data valid or legal and accurate
  - is data of the correct type
  - is data from the correct record
  - self verifying data concepts

Name: Fred

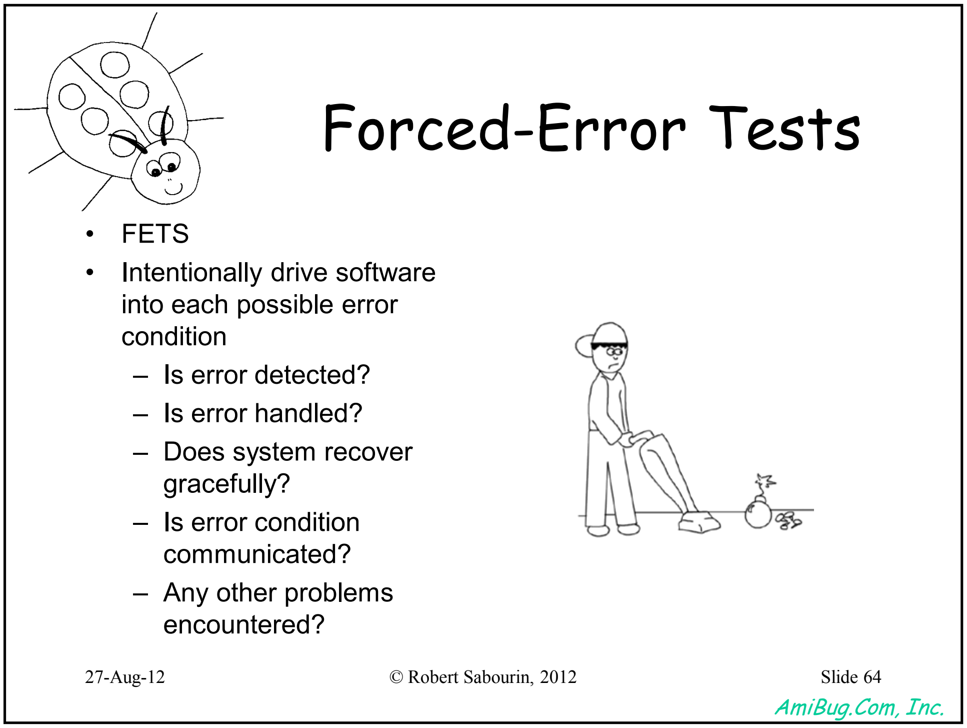
Age: 53

Gender: male

#156-0093-79842

27-Aug-12 © Robert Sabourin, 2012 Slide 63

AmiBug.Com, Inc.

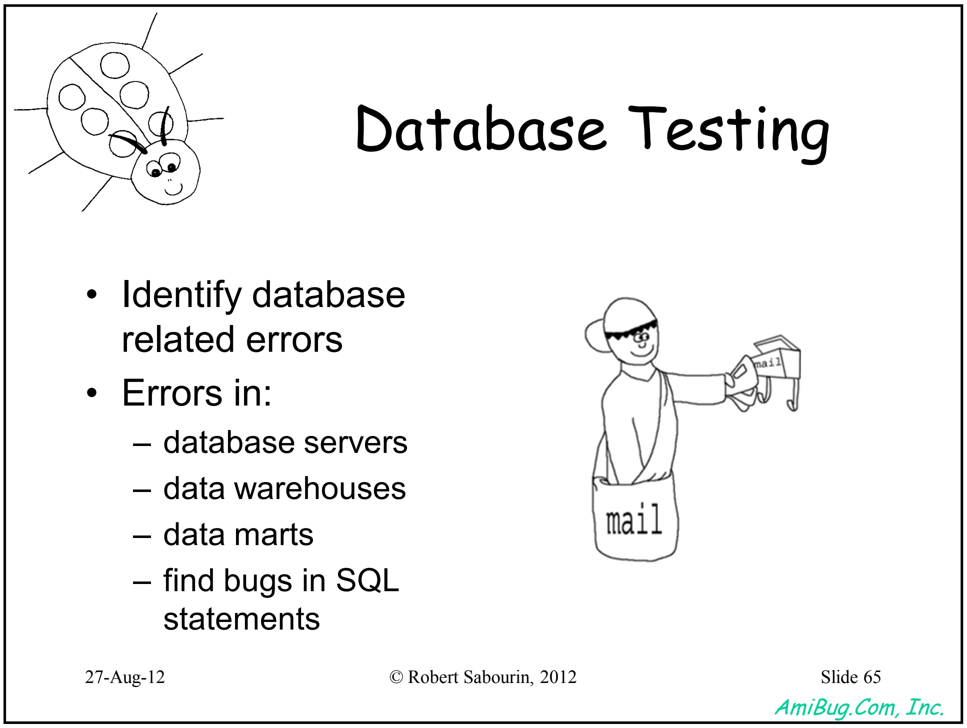


Forced-Error Tests

- FETS
- Intentionally drive software into each possible error condition
  - Is error detected?
  - Is error handled?
  - Does system recover gracefully?
  - Is error condition communicated?
  - Any other problems encountered?

27-Aug-12 © Robert Sabourin, 2012 Slide 64

AmiBug.Com, Inc.

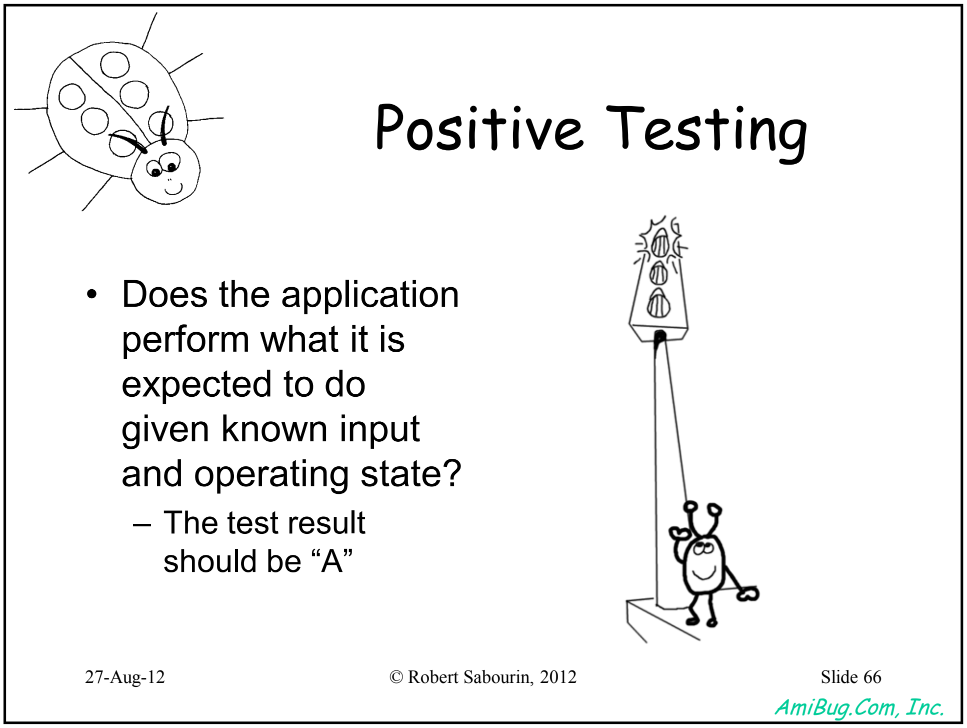


Database Testing

- Identify database related errors
- Errors in:
  - database servers
  - data warehouses
  - data marts
  - find bugs in SQL statements

27-Aug-12 © Robert Sabourin, 2012 Slide 65

AmiBug.Com, Inc.



Positive Testing

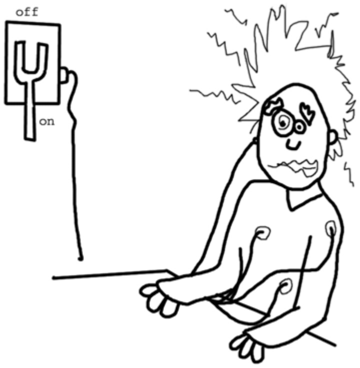
- Does the application perform what it is expected to do given known input and operating state?
  - The test result should be “A”

27-Aug-12 © Robert Sabourin, 2012 Slide 66

AmiBug.Com, Inc.

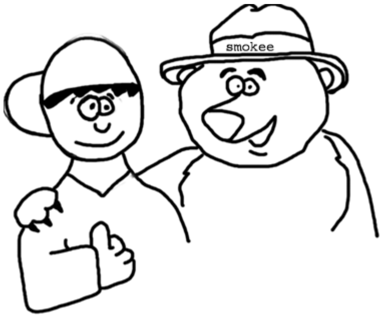
Negative Testing

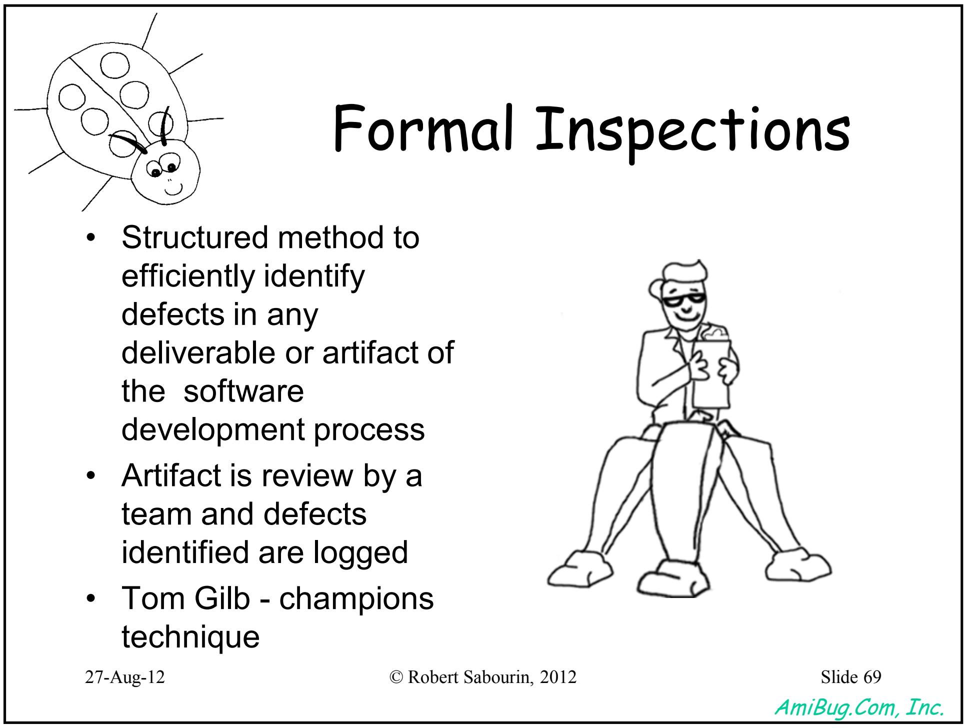
- Does the application not perform what it is not expected to do given known input and operating state?
  - The test result should not be “A”
  - we are testing to ensure it is not “A” we do not care if it is “B” or “C”!



Preventative Testing

- Testing used to avoid introducing defects in software
- We use preventative testing before we code!
  - Formal Inspections
  - Design Reviews
  - Walkthroughs
  - Peer Code Reviews



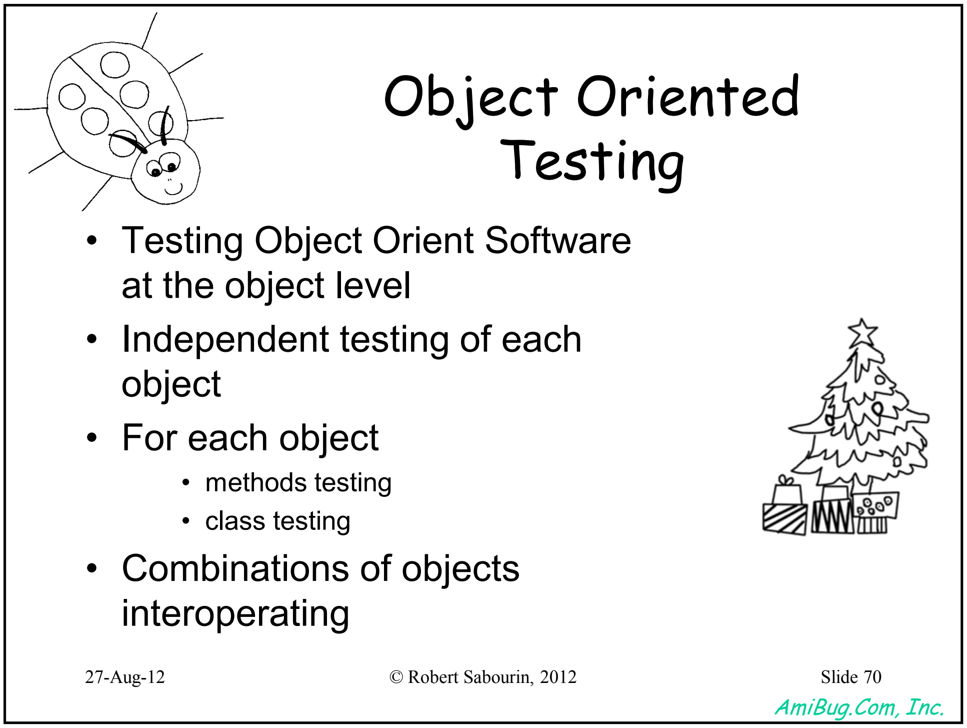


Formal Inspections

- Structured method to efficiently identify defects in any deliverable or artifact of the software development process
- Artifact is review by a team and defects identified are logged
- Tom Gilb - champions technique

27-Aug-12 © Robert Sabourin, 2012 Slide 69

AmiBug.Com, Inc.



Object Oriented Testing

- Testing Object Orient Software at the object level
- Independent testing of each object
- For each object
  - methods testing
  - class testing
- Combinations of objects interoperating

27-Aug-12 © Robert Sabourin, 2012 Slide 70

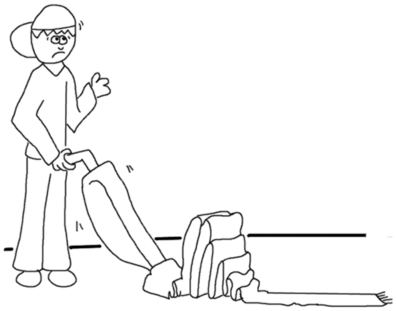
AmiBug.Com, Inc.

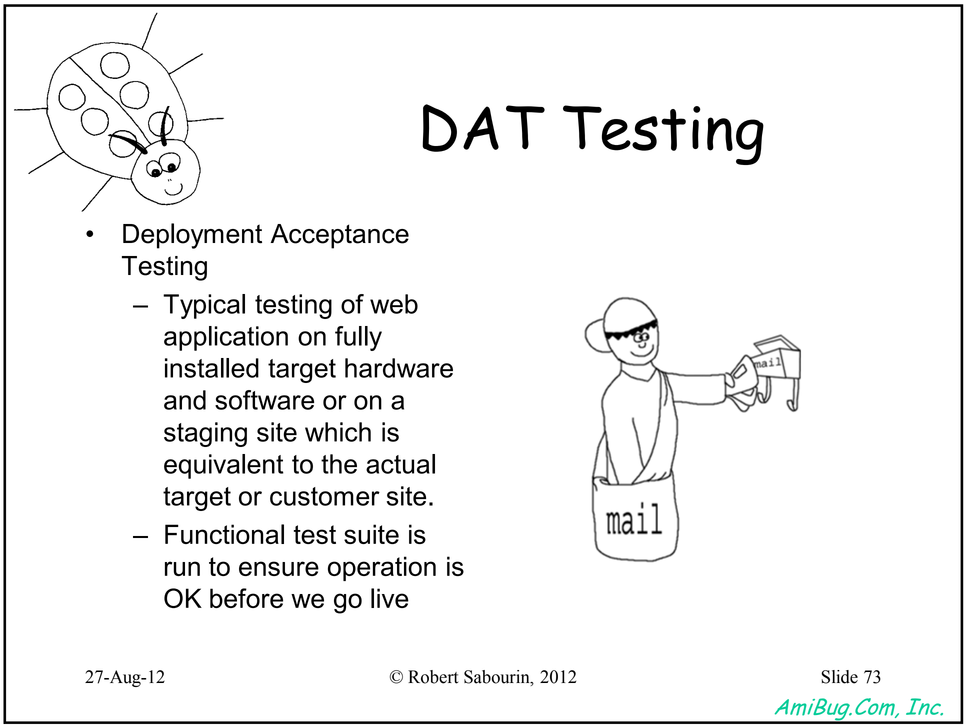
Class/Method Testing

- OOP component testing
  - Testing classes of an object
  - Testing methods of a class
  - Independent of other objects in the system
  - White box method

Boundary Testing

- Test with extreme input values
  - Lower and upper boundaries
  - Any edge conditions
  - Above and below extreme values
- Test to generate extreme output values
  - May or may not require extreme input values
  - Zero divide, overflow
- Record Sizes



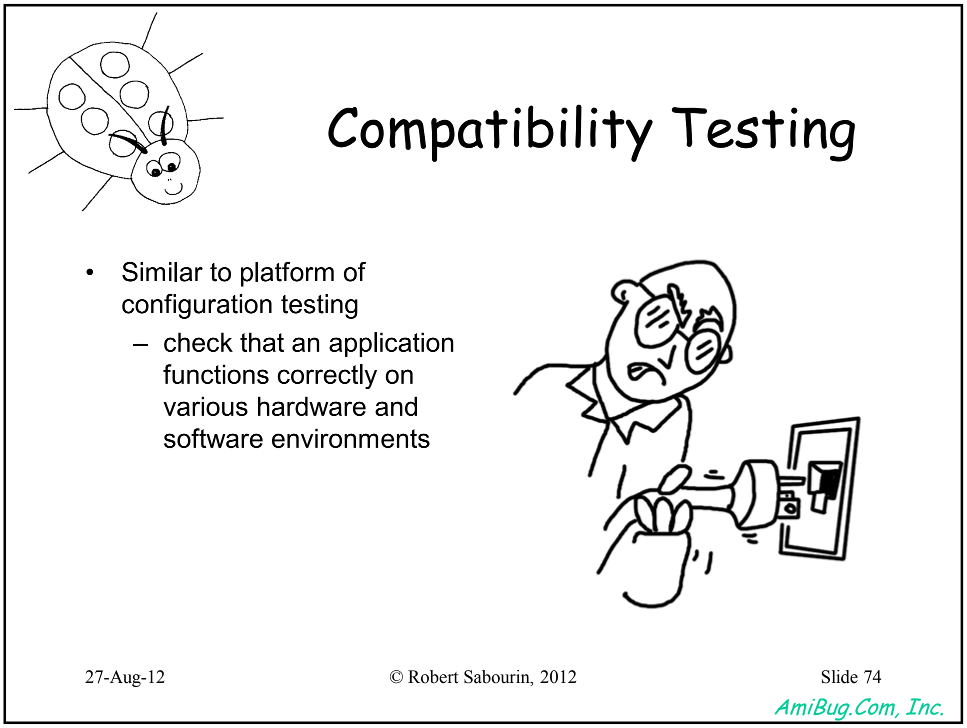


DAT Testing

- Deployment Acceptance Testing
  - Typical testing of web application on fully installed target hardware and software or on a staging site which is equivalent to the actual target or customer site.
  - Functional test suite is run to ensure operation is OK before we go live

27-Aug-12 © Robert Sabourin, 2012 Slide 73

AmiBug.Com, Inc.

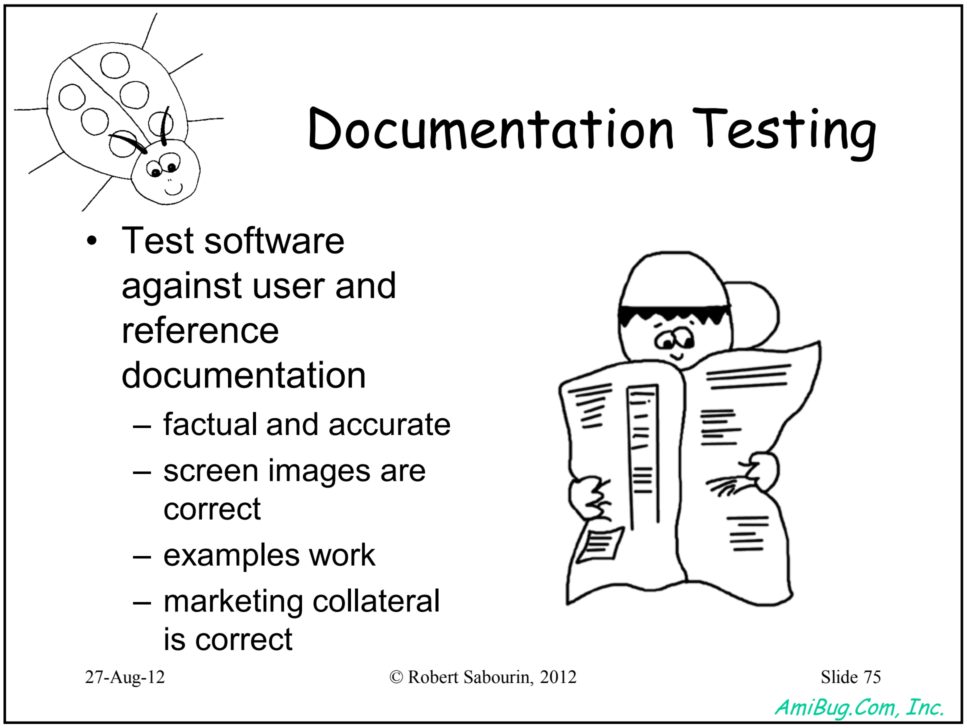


Compatibility Testing

- Similar to platform of configuration testing
  - check that an application functions correctly on various hardware and software environments

27-Aug-12 © Robert Sabourin, 2012 Slide 74

AmiBug.Com, Inc.

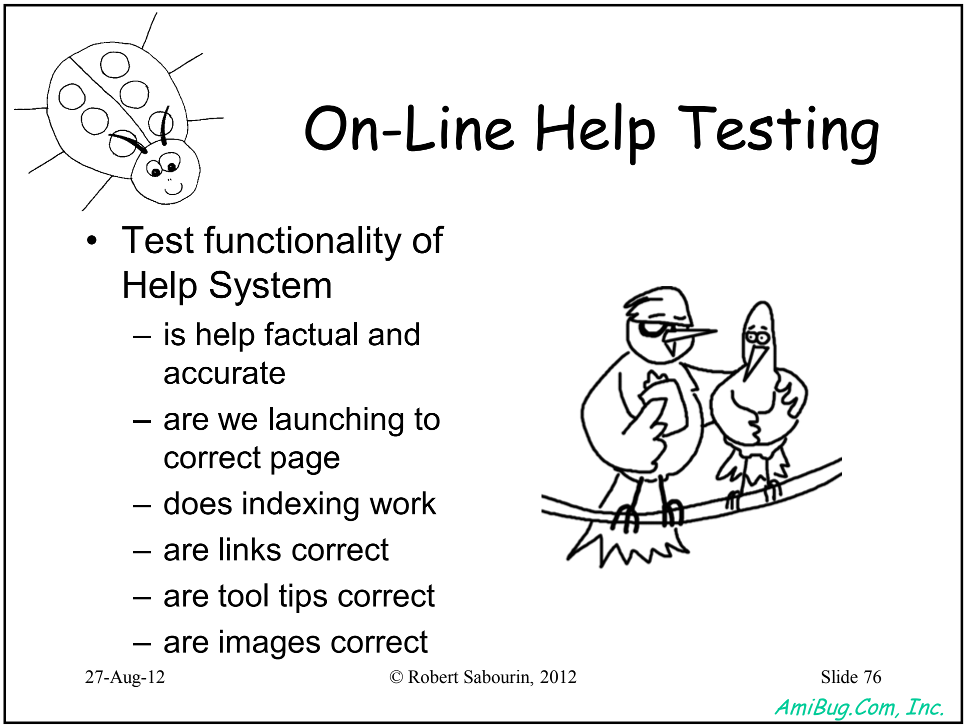


Documentation Testing

- Test software against user and reference documentation
  - factual and accurate
  - screen images are correct
  - examples work
  - marketing collateral is correct

27-Aug-12 © Robert Sabourin, 2012 Slide 75

AmiBug.Com, Inc.

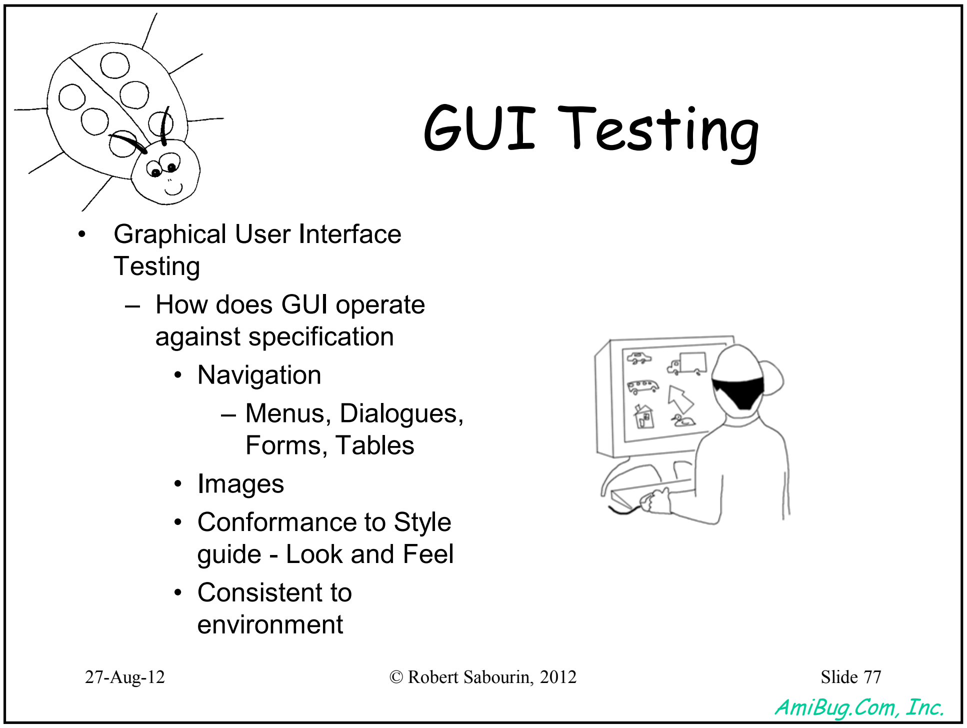


On-Line Help Testing

- Test functionality of Help System
  - is help factual and accurate
  - are we launching to correct page
  - does indexing work
  - are links correct
  - are tool tips correct
  - are images correct

27-Aug-12 © Robert Sabourin, 2012 Slide 76

AmiBug.Com, Inc.

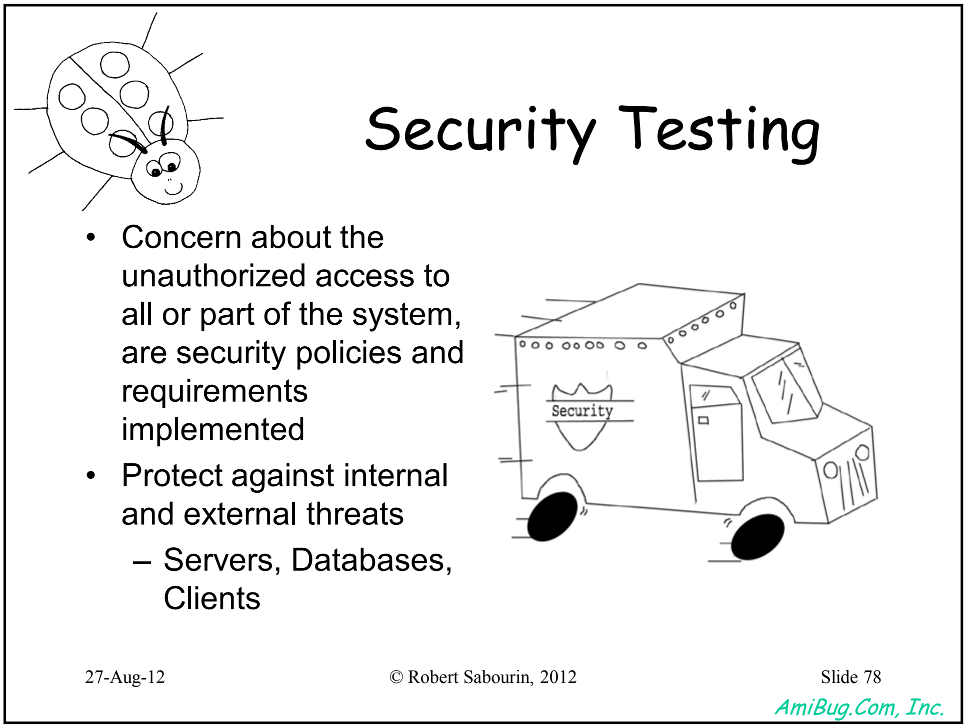


GUI Testing

- Graphical User Interface Testing
  - How does GUI operate against specification
    - Navigation
      - Menus, Dialogues, Forms, Tables
    - Images
    - Conformance to Style guide - Look and Feel
    - Consistent to environment

27-Aug-12 © Robert Sabourin, 2012 Slide 77

AmiBug.Com, Inc.



Security Testing

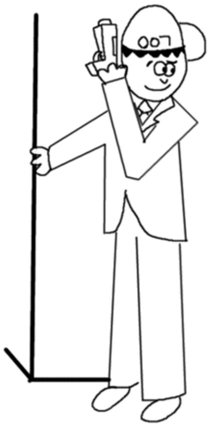
- Concern about the unauthorized access to all or part of the system, are security policies and requirements implemented
- Protect against internal and external threats
  - Servers, Databases, Clients

27-Aug-12 © Robert Sabourin, 2012 Slide 78

AmiBug.Com, Inc.

Penetration Testing

- Evaluate effectiveness of network defenses
  - External expertise
  - Conducted before system is live
  - Continue on live system
  - Work from outside perimeter
  - Simulate work of hackers

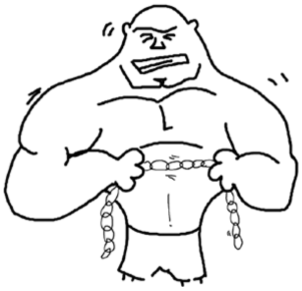


27-Aug-12 Slide 79

AmiBug.Com, Inc.

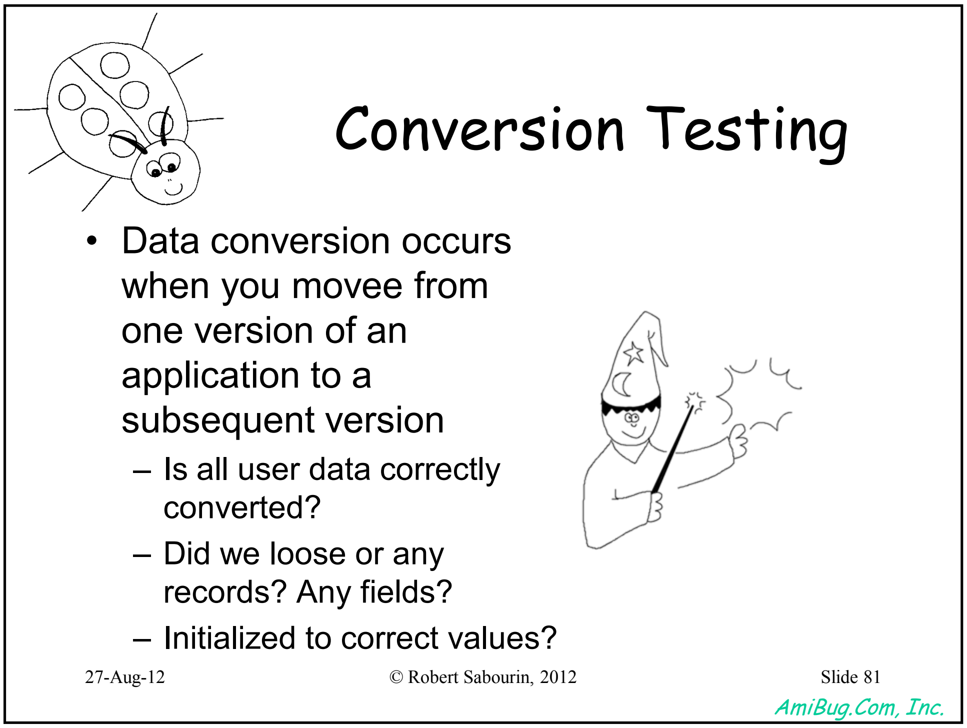
Link Testing

- Test all links between all pages of a web site
  - Point to the correct page?
  - Is page accessible?
  - Are references relative or absolute?
- Several automated link testing tools



27-Aug-12 Slide 80

AmiBug.Com, Inc.

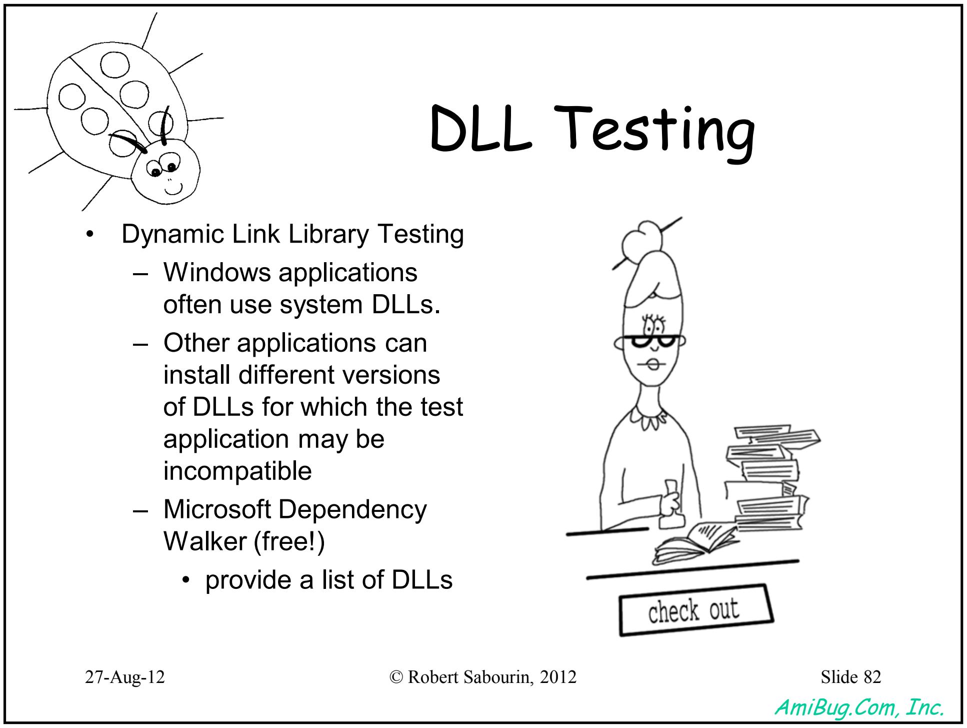


Conversion Testing

- Data conversion occurs when you movee from one version of an application to a subsequent version
  - Is all user data correctly converted?
  - Did we loose or any records? Any fields?
  - Initialized to correct values?

27-Aug-12 © Robert Sabourin, 2012 Slide 81

AmiBug.Com, Inc.

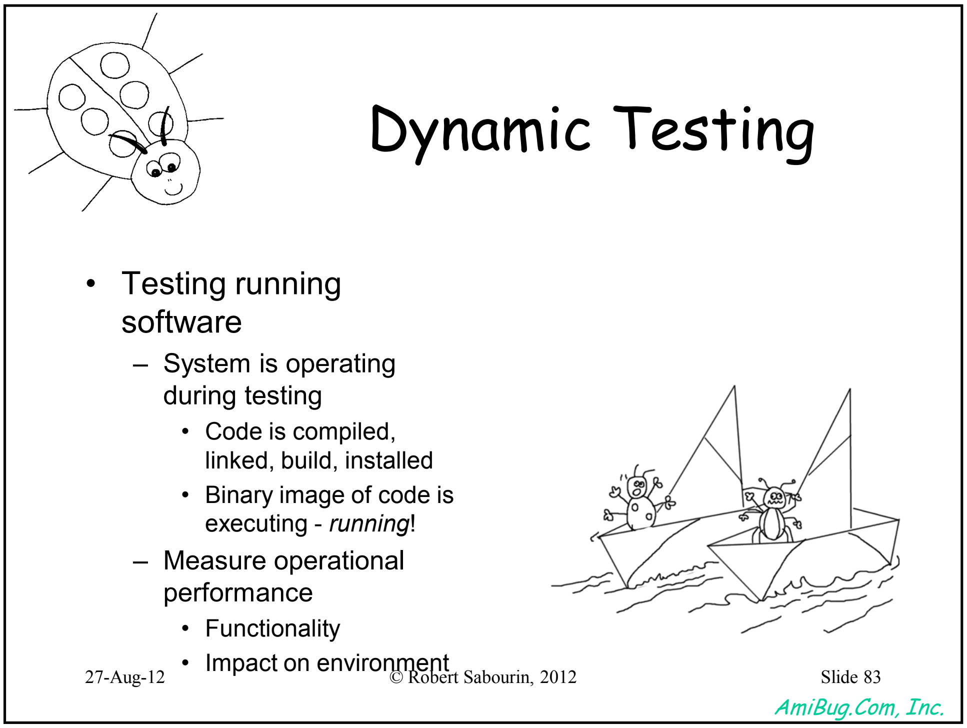


DLL Testing

- Dynamic Link Library Testing
  - Windows applications often use system DLLs.
  - Other applications can install different versions of DLLs for which the test application may be incompatible
  - Microsoft Dependency Walker (free!)
    - provide a list of DLLs

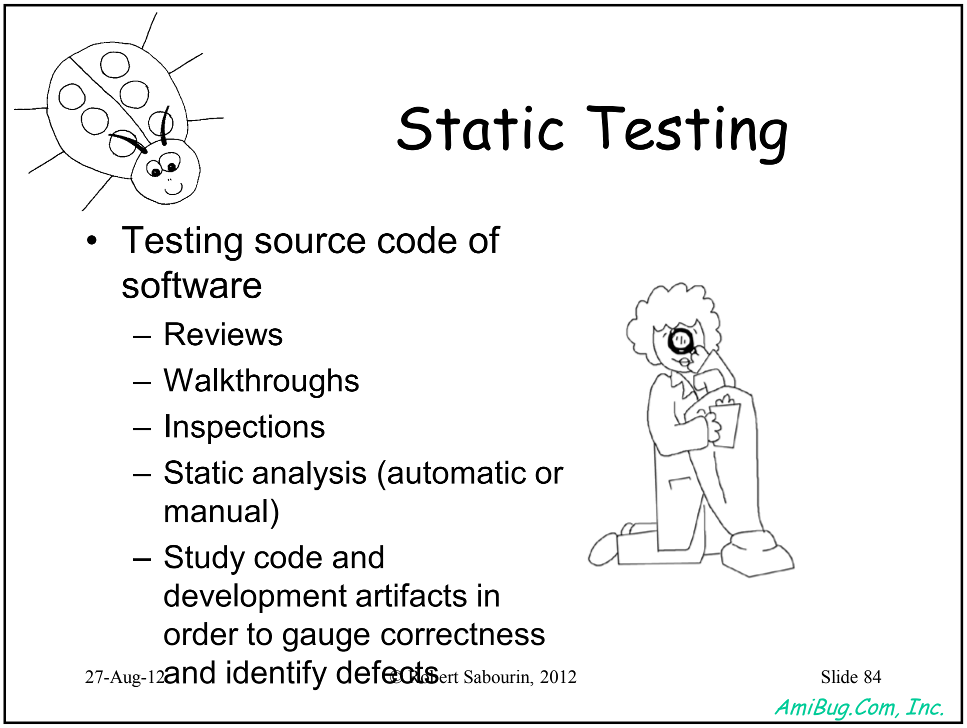
27-Aug-12 © Robert Sabourin, 2012 Slide 82

AmiBug.Com, Inc.



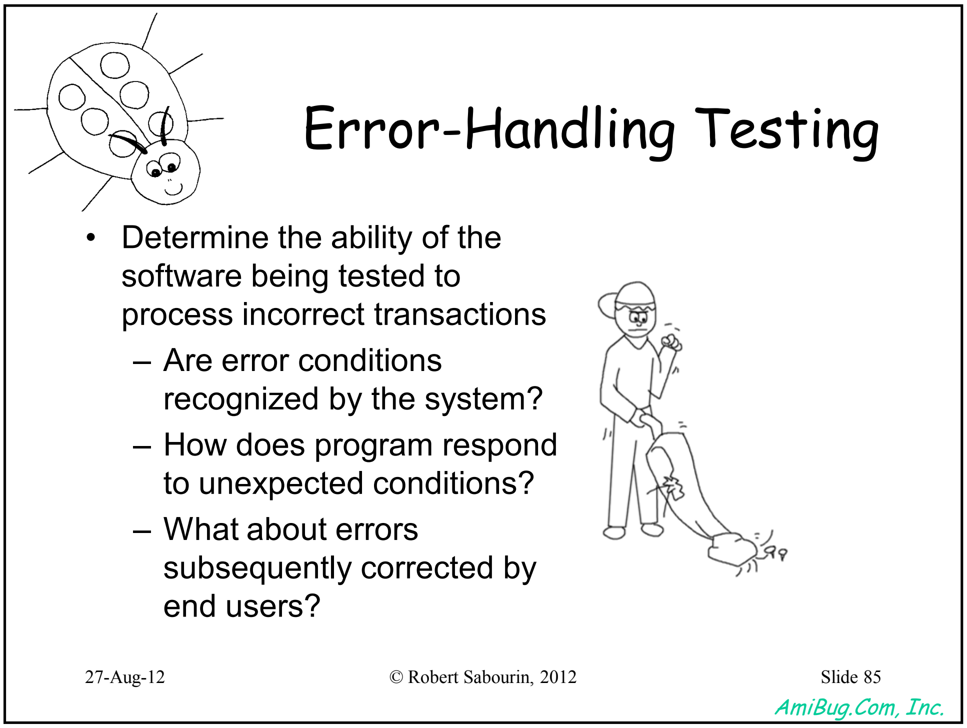
Dynamic Testing

- Testing running software
  - System is operating during testing
    - Code is compiled, linked, build, installed
    - Binary image of code is executing - *running*!
  - Measure operational performance
    - Functionality
    - Impact on environment



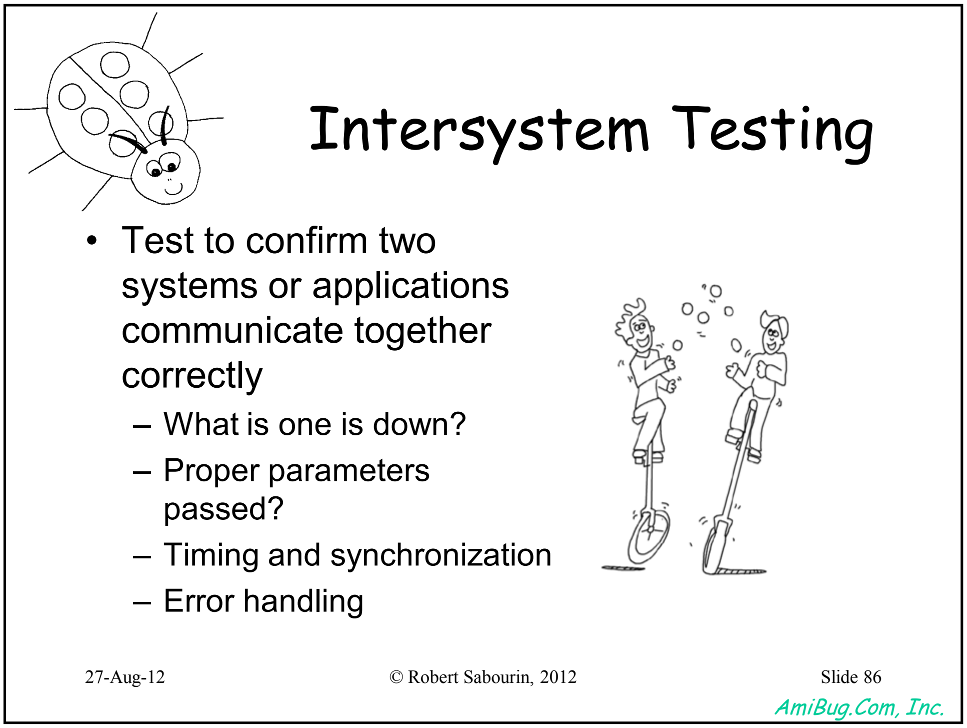
Static Testing

- Testing source code of software
  - Reviews
  - Walkthroughs
  - Inspections
  - Static analysis (automatic or manual)
  - Study code and development artifacts in order to gauge correctness and identify defects



Error-Handling Testing

- Determine the ability of the software being tested to process incorrect transactions
  - Are error conditions recognized by the system?
  - How does program respond to unexpected conditions?
  - What about errors subsequently corrected by end users?



Intersystem Testing

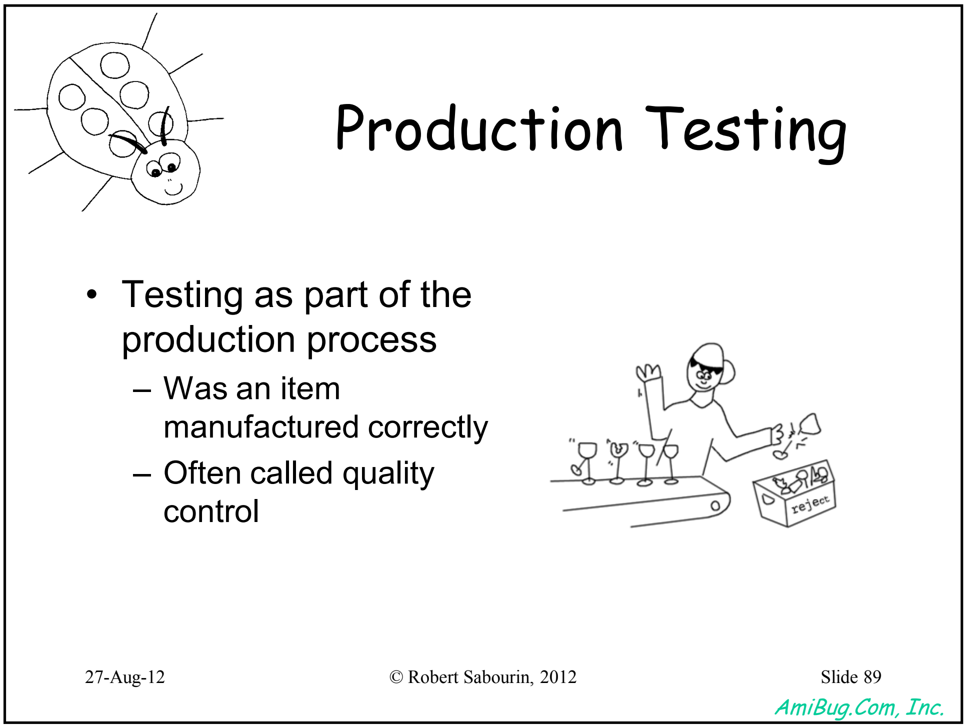
- Test to confirm two systems or applications communicate together correctly
  - What is one is down?
  - Proper parameters passed?
  - Timing and synchronization
  - Error handling

Control Testing

- Part of system testing to ensure that controls on application are correct:
  - accurate and complete data
  - authorized transactions
  - audit trail
  - integrity of processing
- Accounting systems as an example!
  - authentication, electronic authorization

Big Bang Testing

- Synonymous to *Non-incremental Testing*
  - Do not do Integration Testing until the entire system has been build
  - Attack all at once
    - Risky
    - No leverage!

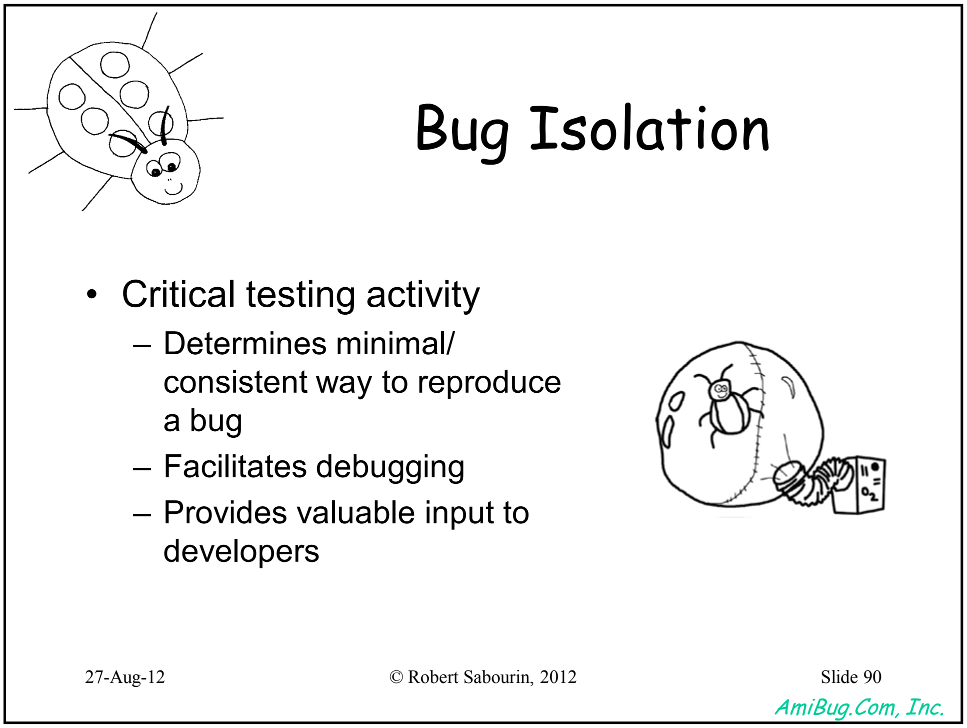


Production Testing

- Testing as part of the production process
  - Was an item manufactured correctly
  - Often called quality control

27-Aug-12 © Robert Sabourin, 2012 Slide 89

AmiBug.Com, Inc.



Bug Isolation

- Critical testing activity
  - Determines minimal/ consistent way to reproduce a bug
  - Facilitates debugging
  - Provides valuable input to developers

27-Aug-12 © Robert Sabourin, 2012 Slide 90

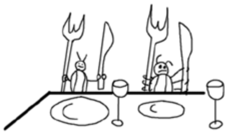
AmiBug.Com, Inc.

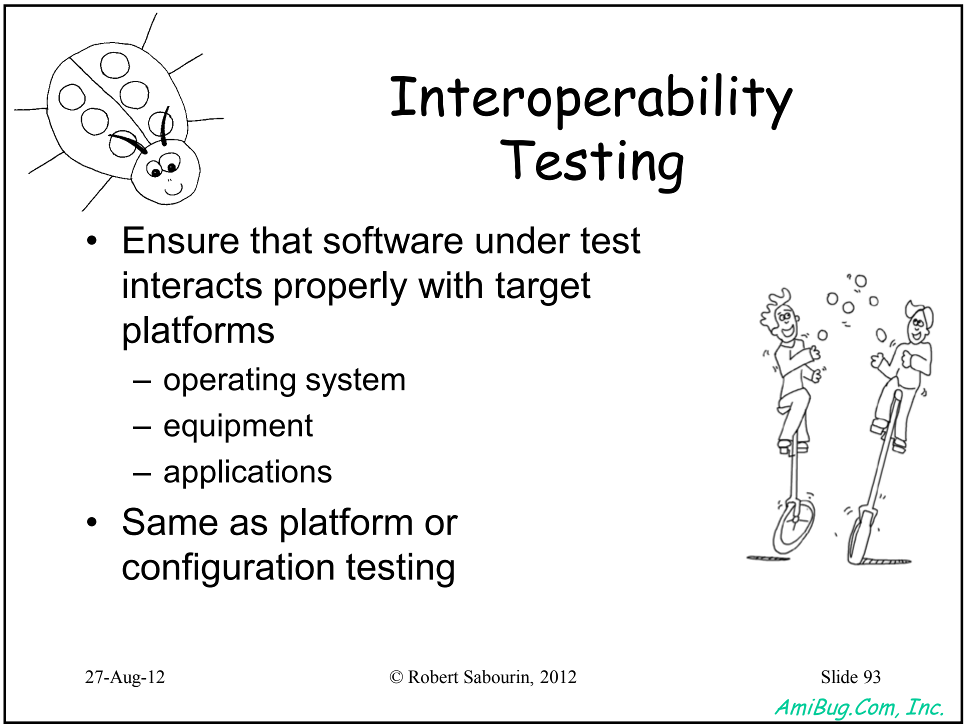
Confirmation Testing

- Typically:
  - Tester finds the bug
  - Product/Development leads prioritize the bug
  - Developer fixes the bug
  - Tester confirms that the fixed bug is really fixed in the appropriate software build

Behavioral Testing

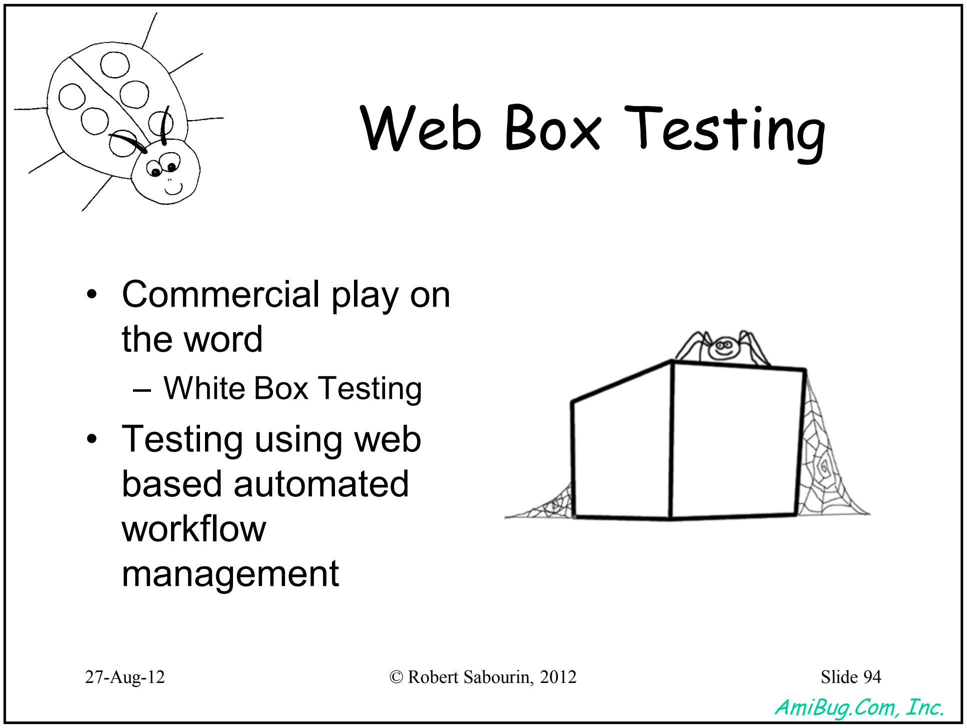
- Testing based on what a system is supposed to do
- Tests based on how a system functions
- Tests based on how a system is supposed to be used
- Black box testing





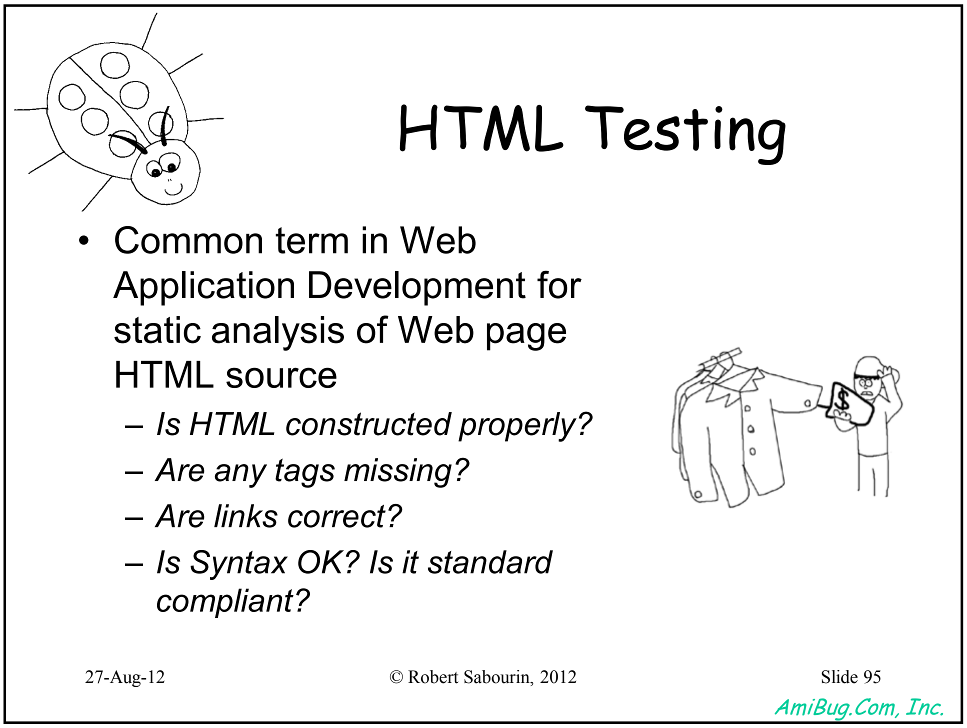
Interoperability Testing

- Ensure that software under test interacts properly with target platforms
  - operating system
  - equipment
  - applications
- Same as platform or configuration testing



Web Box Testing

- Commercial play on the word
  - White Box Testing
- Testing using web based automated workflow management

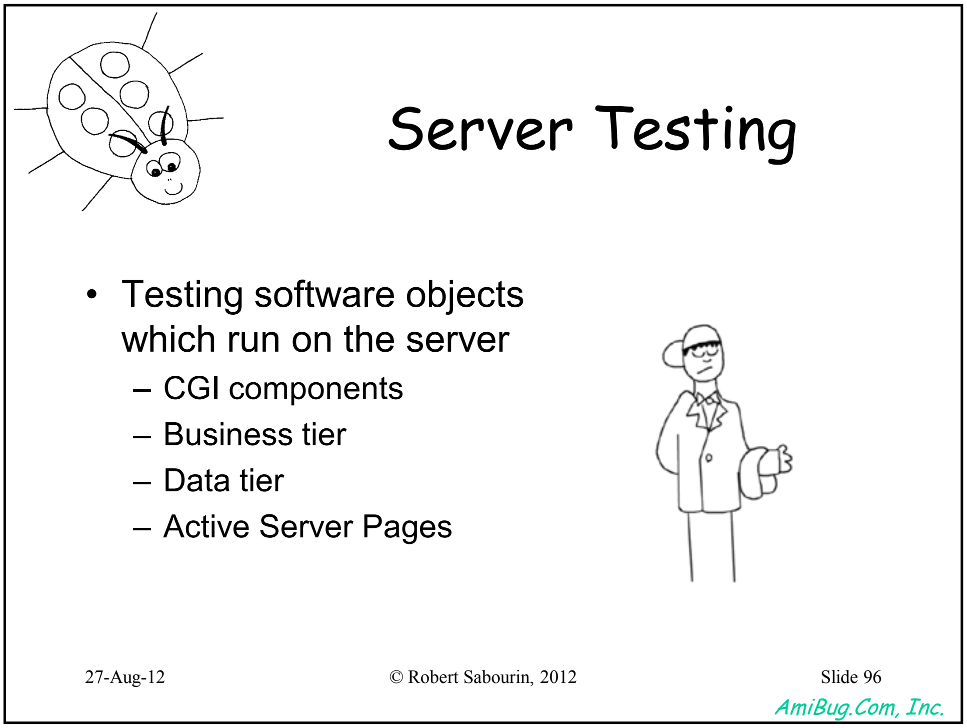


HTML Testing

- Common term in Web Application Development for static analysis of Web page HTML source
  - *Is HTML constructed properly?*
  - *Are any tags missing?*
  - *Are links correct?*
  - *Is Syntax OK? Is it standard compliant?*

27-Aug-12 © Robert Sabourin, 2012 Slide 95

AmiBug.Com, Inc.

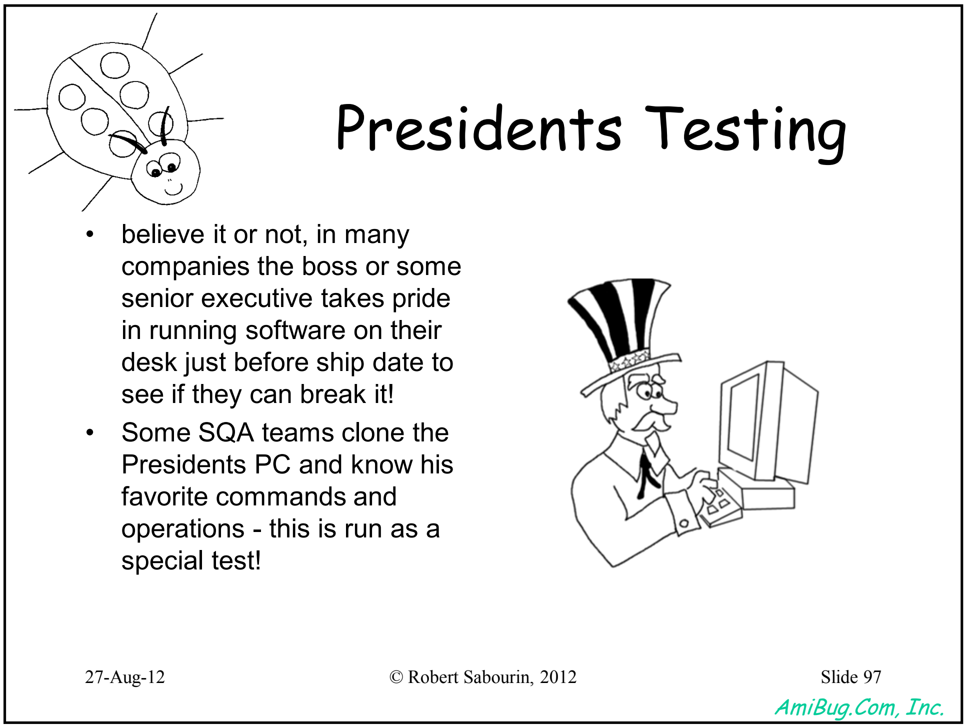


Server Testing

- Testing software objects which run on the server
  - CGI components
  - Business tier
  - Data tier
  - Active Server Pages

27-Aug-12 © Robert Sabourin, 2012 Slide 96

AmiBug.Com, Inc.

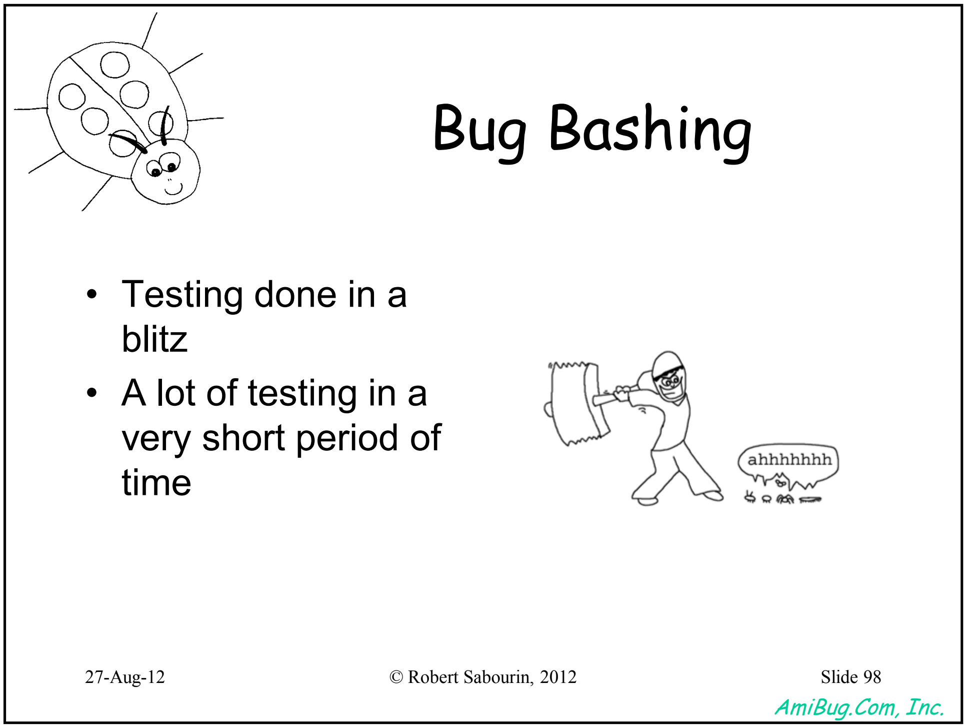


Presidents Testing

- believe it or not, in many companies the boss or some senior executive takes pride in running software on their desk just before ship date to see if they can break it!
- Some SQA teams clone the Presidents PC and know his favorite commands and operations - this is run as a special test!

27-Aug-12 © Robert Sabourin, 2012 Slide 97

AmiBug.Com, Inc.

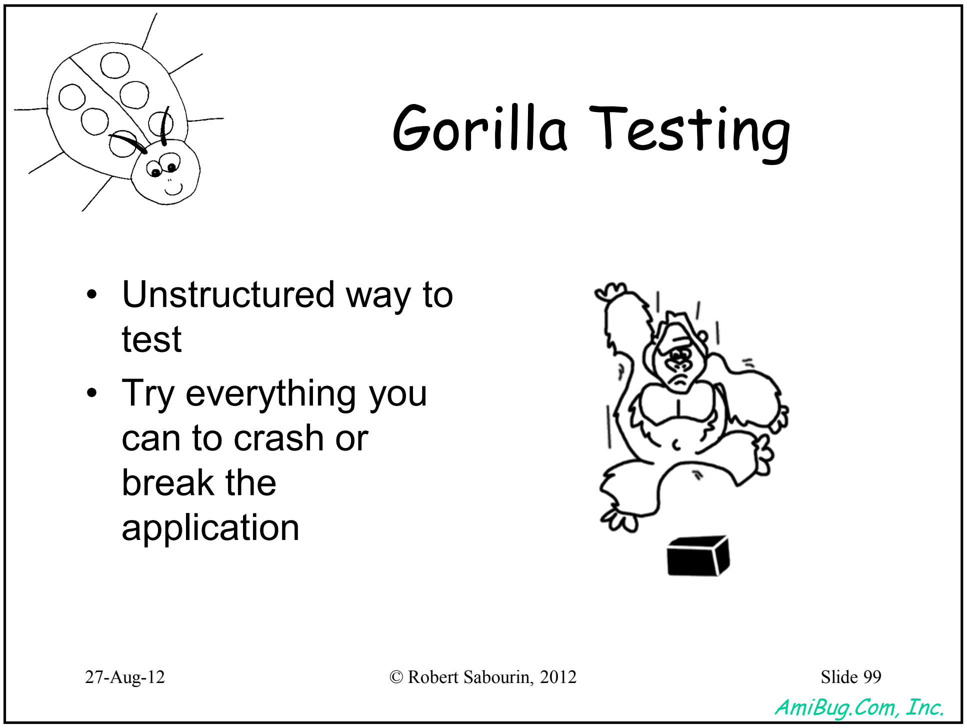


Bug Bashing

- Testing done in a blitz
- A lot of testing in a very short period of time

27-Aug-12 © Robert Sabourin, 2012 Slide 98

AmiBug.Com, Inc.

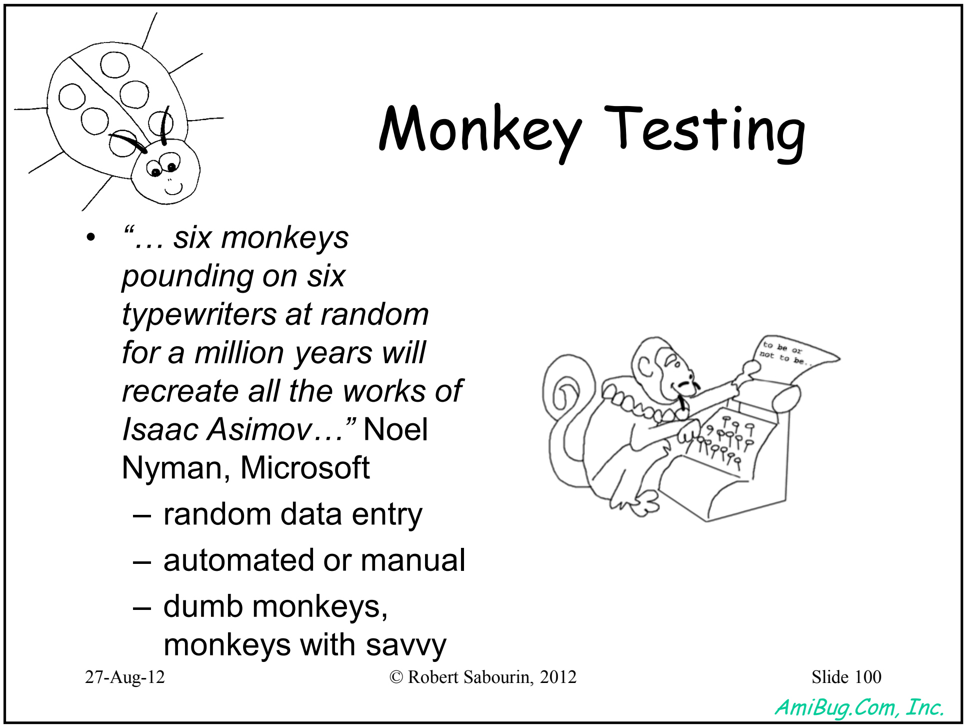


Gorilla Testing

- Unstructured way to test
- Try everything you can to crash or break the application

27-Aug-12 © Robert Sabourin, 2012 Slide 99

AmiBug.Com, Inc.



Monkey Testing

- *"… six monkeys pounding on six typewriters at random for a million years will recreate all the works of Isaac Asimov…"* Noel Nyman, Microsoft
  - random data entry
  - automated or manual
  - dumb monkeys, monkeys with savvy

27-Aug-12 © Robert Sabourin, 2012 Slide 100

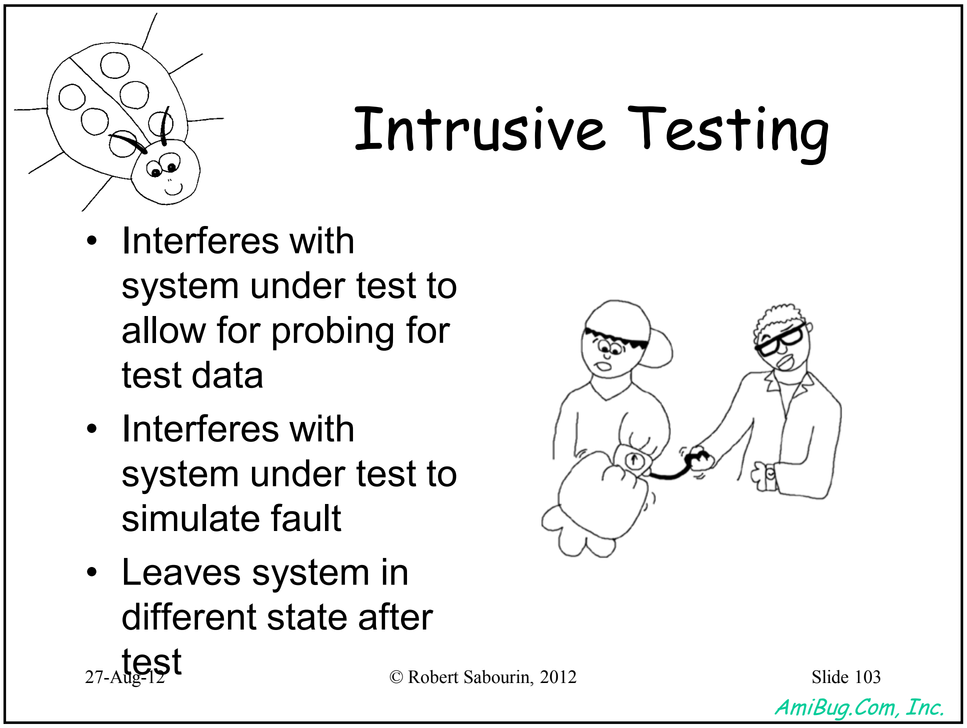
AmiBug.Com, Inc.

Guerrilla Testing

- Wage a Guerrilla attack on the software being tested, observe behavior
  - Do evil things
  - Damage things
  - Intentionally corrupt data

Non-Intrusive Testing

- Black Box
- Does not interfere with system under test
- Leaves system in same state after test

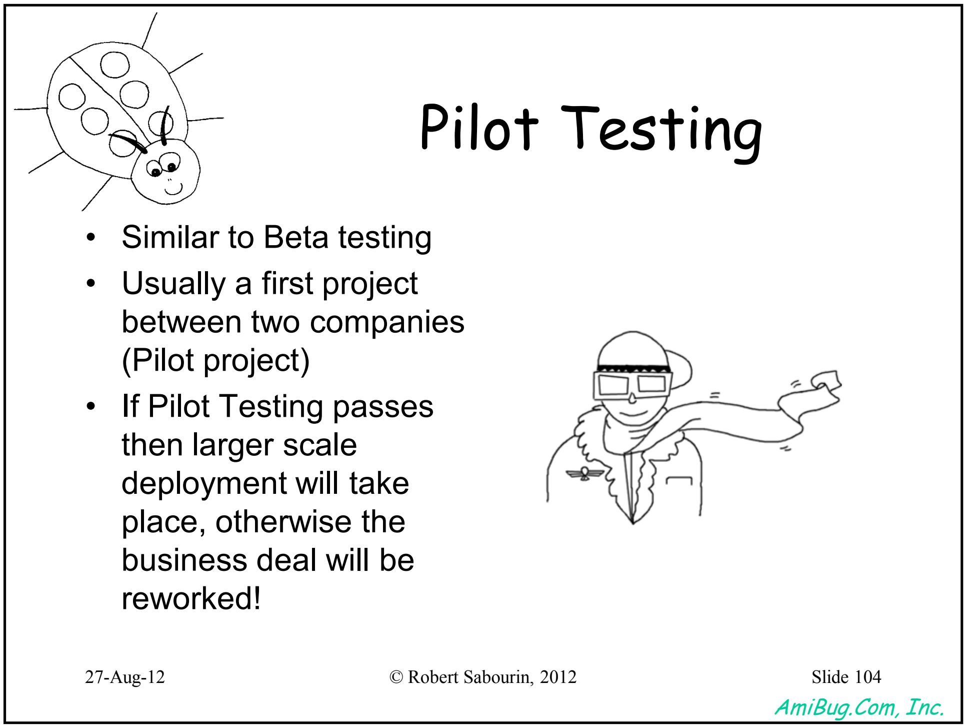


Intrusive Testing

- Interferes with system under test to allow for probing for test data
- Interferes with system under test to simulate fault
- Leaves system in different state after test

27-Aug-12 © Robert Sabourin, 2012 Slide 103

AmiBug.Com, Inc.

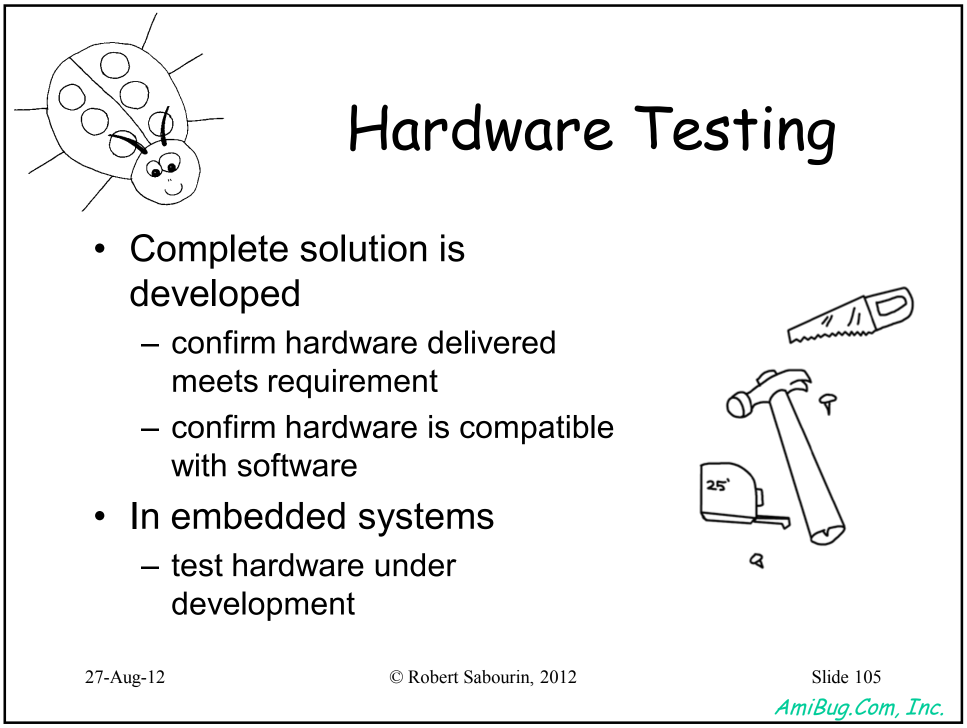


Pilot Testing

- Similar to Beta testing
- Usually a first project between two companies (Pilot project)
- If Pilot Testing passes then larger scale deployment will take place, otherwise the business deal will be reworked!

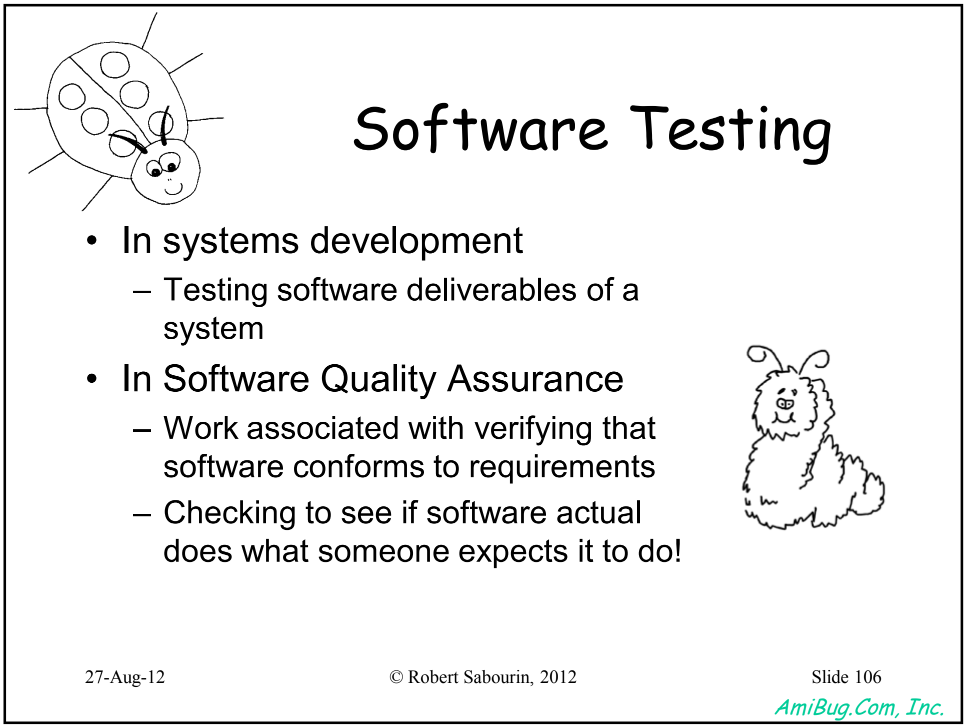
27-Aug-12 © Robert Sabourin, 2012 Slide 104

AmiBug.Com, Inc.



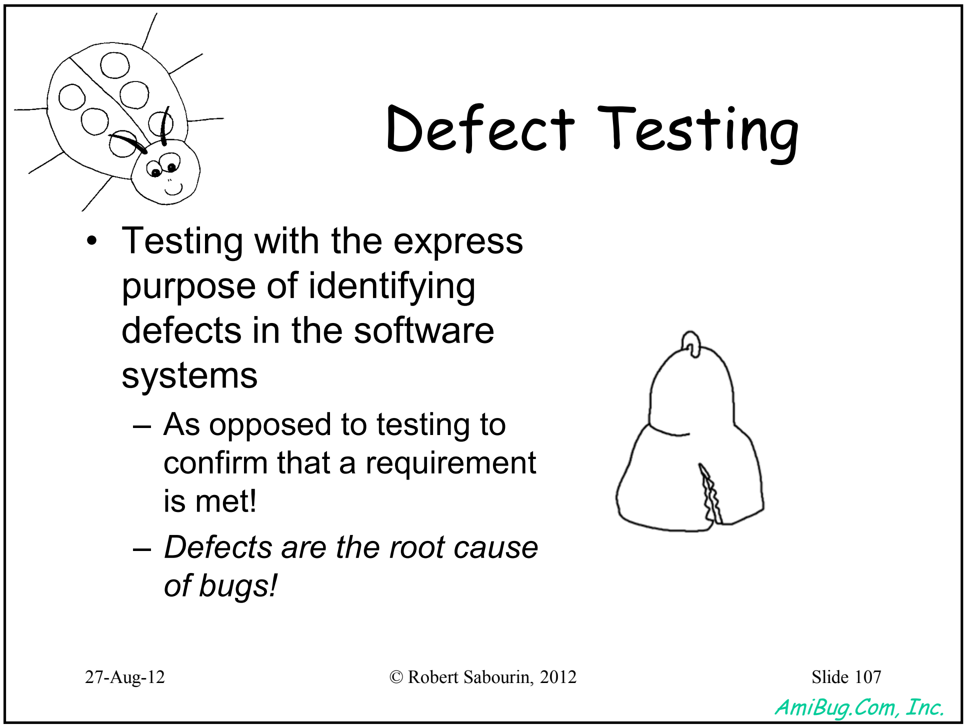
Hardware Testing

- Complete solution is developed
  - confirm hardware delivered meets requirement
  - confirm hardware is compatible with software
- In embedded systems
  - test hardware under development



Software Testing

- In systems development
  - Testing software deliverables of a system
- In Software Quality Assurance
  - Work associated with verifying that software conforms to requirements
  - Checking to see if software actual does what someone expects it to do!

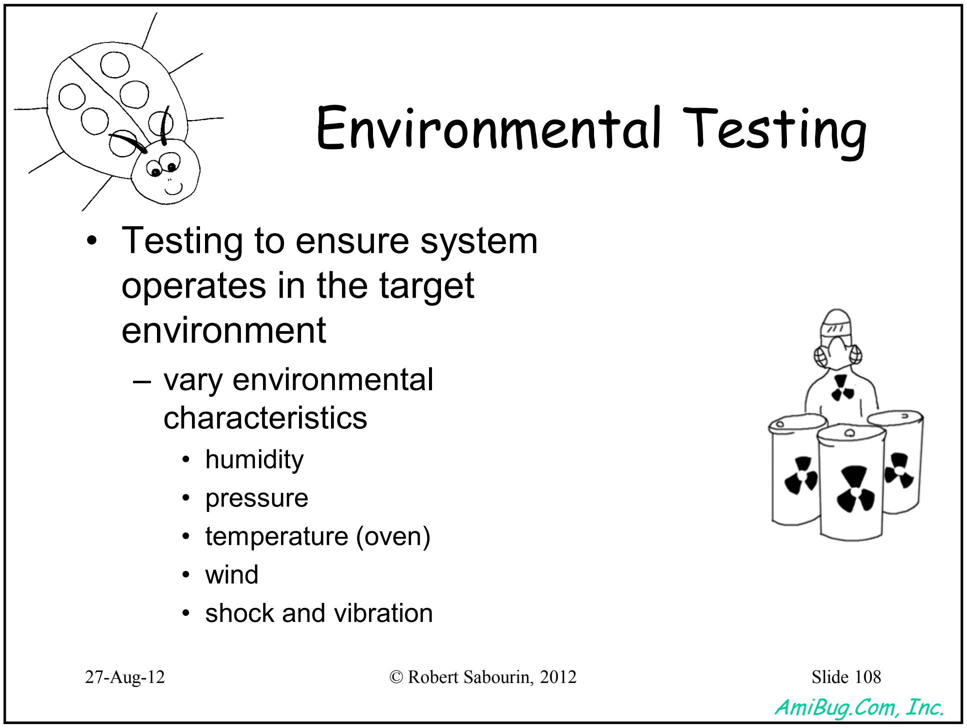


Defect Testing

- Testing with the express purpose of identifying defects in the software systems
  - – As opposed to testing to confirm that a requirement is met!
  - – *Defects are the root cause of bugs!*

27-Aug-12 © Robert Sabourin, 2012 Slide 107

AmiBug.Com, Inc.

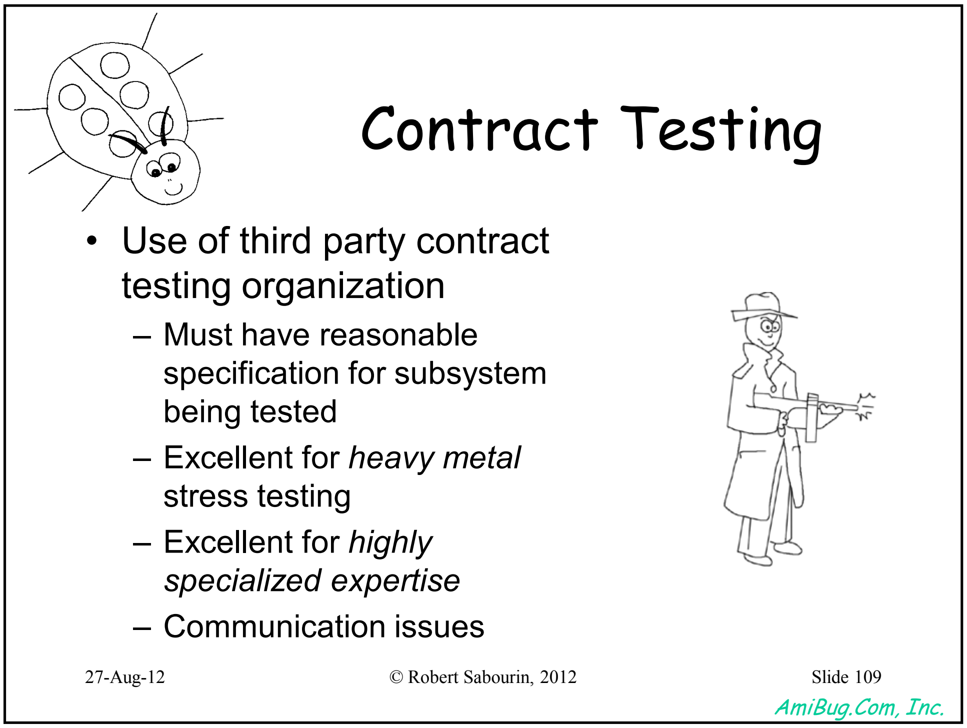


Environmental Testing

- Testing to ensure system operates in the target environment
  - – vary environmental characteristics
    - humidity
    - pressure
    - temperature (oven)
    - wind
    - shock and vibration

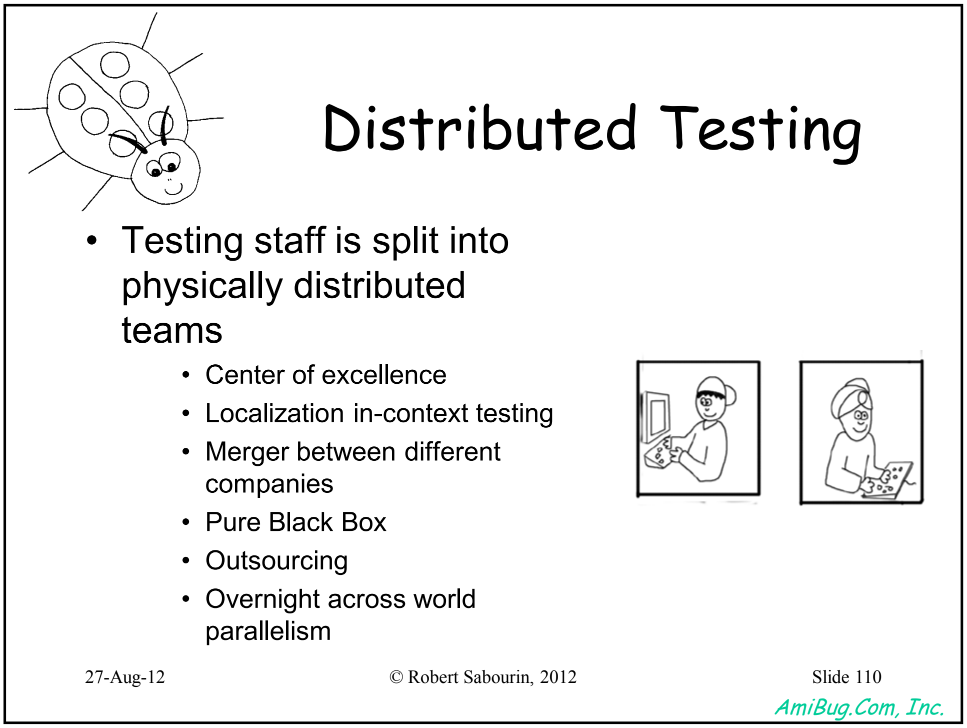
27-Aug-12 © Robert Sabourin, 2012 Slide 108

AmiBug.Com, Inc.



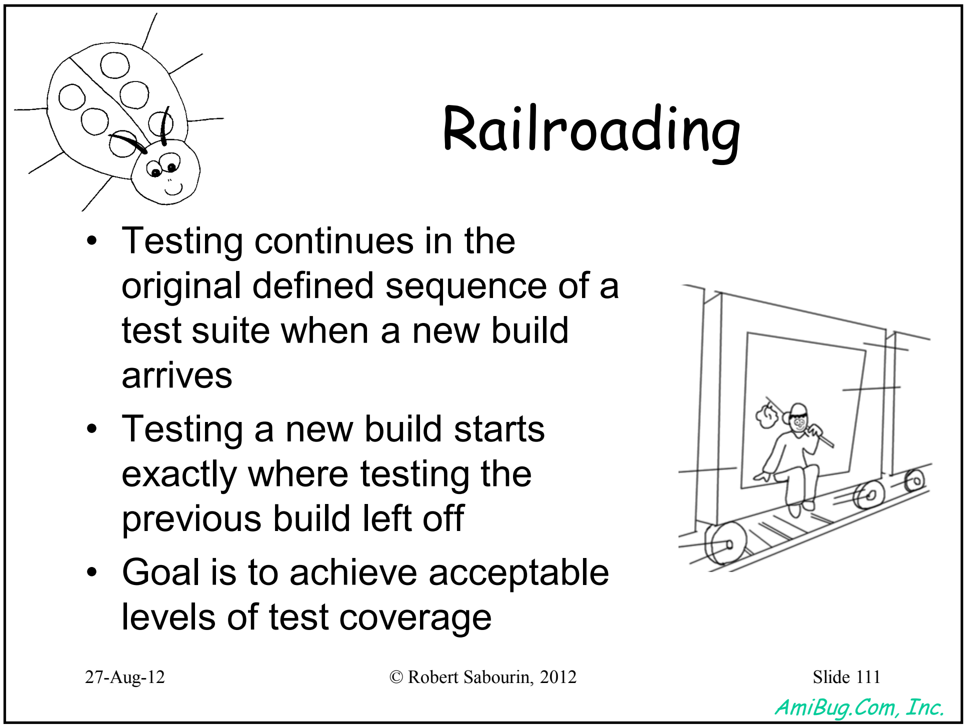
Contract Testing

- Use of third party contract testing organization
  - Must have reasonable specification for subsystem being tested
  - Excellent for *heavy metal* stress testing
  - Excellent for *highly specialized expertise*
  - Communication issues



Distributed Testing

- Testing staff is split into physically distributed teams
  - Center of excellence
  - Localization in-context testing
  - Merger between different companies
  - Pure Black Box
  - Outsourcing
  - Overnight across world parallelism

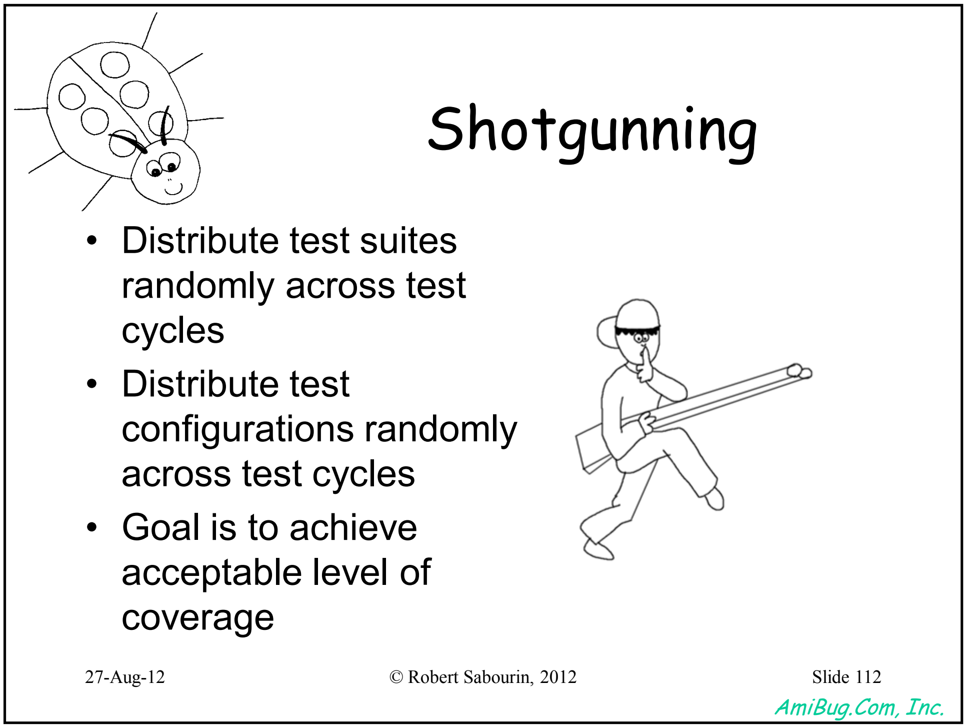


Railroading

- Testing continues in the original defined sequence of a test suite when a new build arrives
- Testing a new build starts exactly where testing the previous build left off
- Goal is to achieve acceptable levels of test coverage

27-Aug-12 © Robert Sabourin, 2012 Slide 111

AmiBug.Com, Inc.

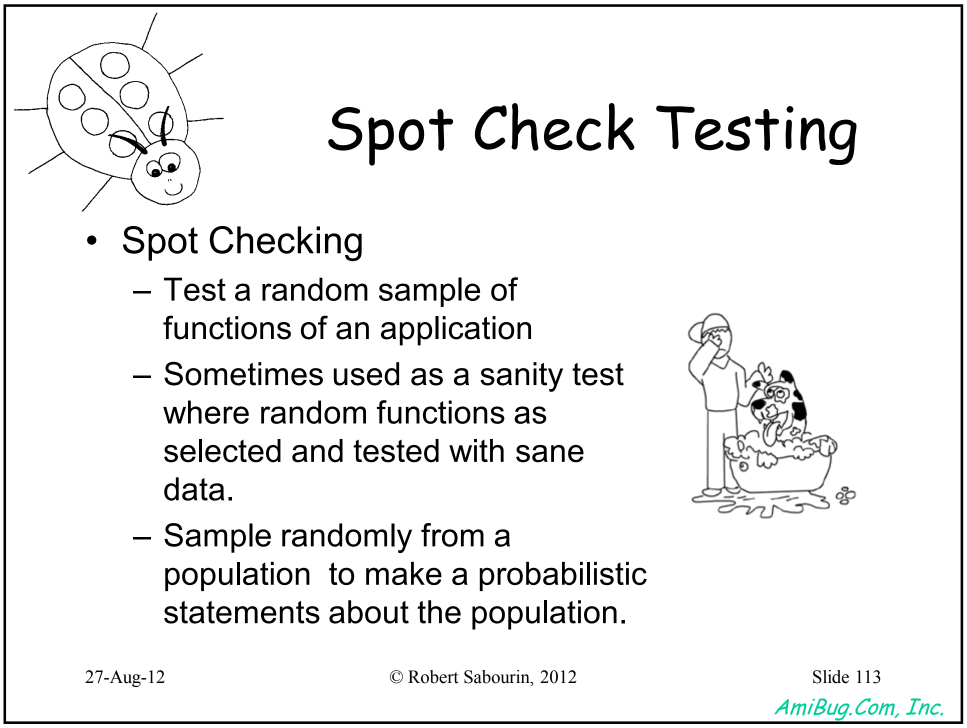


Shotgunning

- Distribute test suites randomly across test cycles
- Distribute test configurations randomly across test cycles
- Goal is to achieve acceptable level of coverage

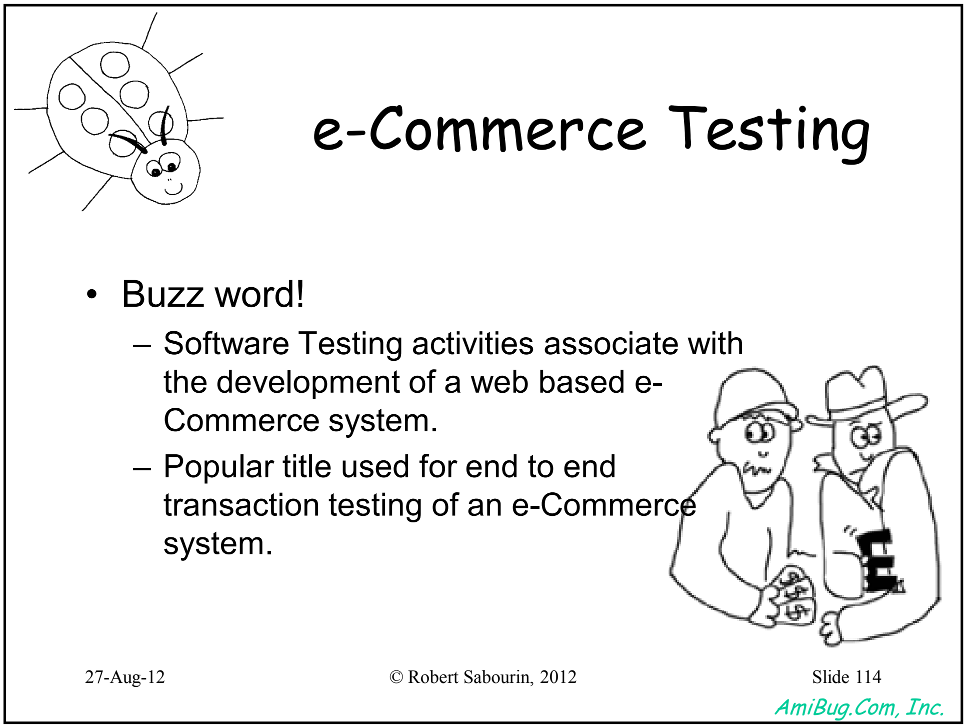
27-Aug-12 © Robert Sabourin, 2012 Slide 112

AmiBug.Com, Inc.



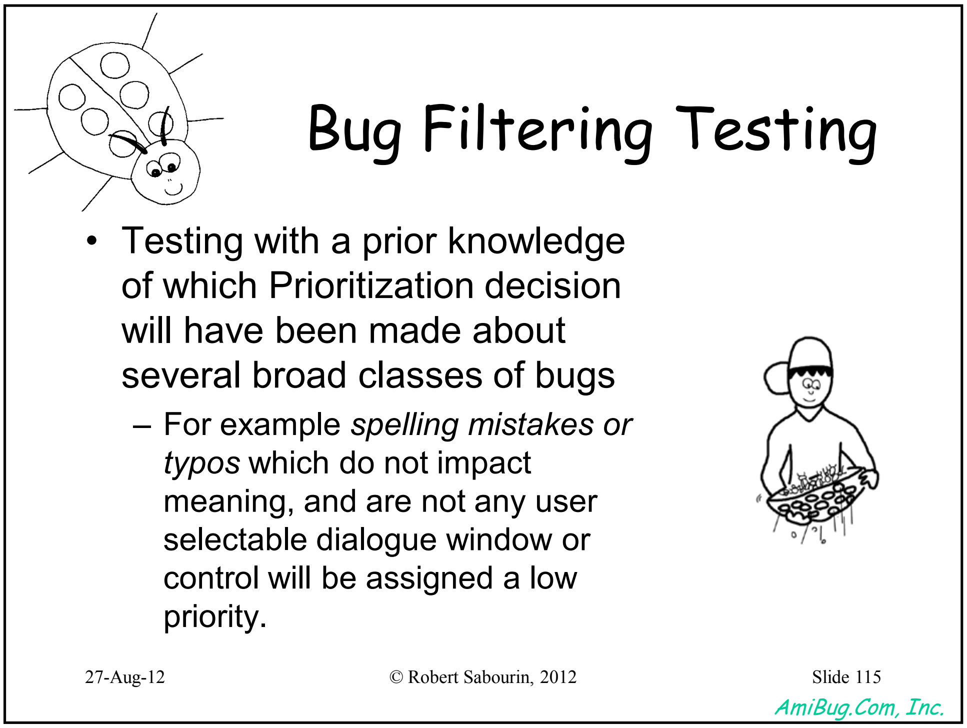
Spot Check Testing

- Spot Checking
  - Test a random sample of functions of an application
  - Sometimes used as a sanity test where random functions as selected and tested with sane data.
  - Sample randomly from a population to make a probabilistic statements about the population.



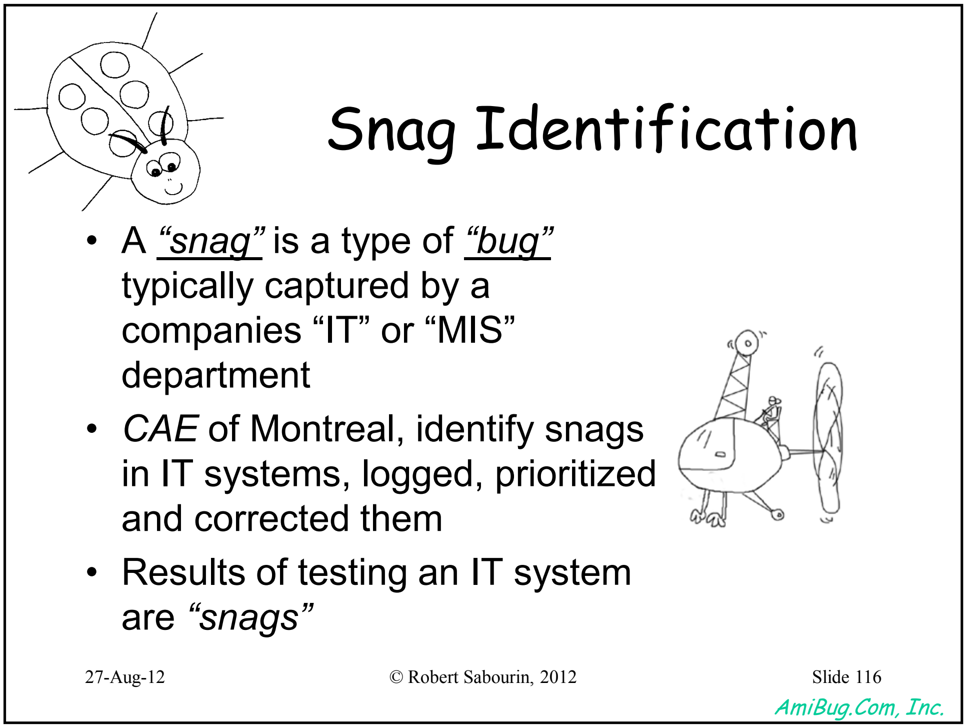
e-Commerce Testing

- Buzz word!
  - Software Testing activities associate with the development of a web based e-Commerce system.
  - Popular title used for end to end transaction testing of an e-Commerce system.



Bug Filtering Testing

- Testing with a prior knowledge of which Prioritization decision will have been made about several broad classes of bugs
  - For example *spelling mistakes or typos* which do not impact meaning, and are not any user selectable dialogue window or control will be assigned a low priority.



Snag Identification

- A *<u>“snag”</u>* is a type of *<u>“bug”</u>* typically captured by a companies “IT” or “MIS” department
- *CAE* of Montreal, identify snags in IT systems, logged, prioritized and corrected them
- Results of testing an IT system are *“snags”*

Reliability Testing

- Testing to determine the reliability of software
  - Statistical samples
  - Accelerated life testing
  - Project MTBF
    - Mean Time Between Failures

Availability Testing

- Testing to determine the availability of software
  - Can users connect?
  - Does application respond to input?
  - How many failed attempts to load a page occur as the system is loaded?
  - Can the system run 24/7?

Maintainability Testing

- Testing to determine whether the software can be maintained after commercial deployment
  - Can field installations be upgraded?
  - What it the associated PITA factor?
  - Can code be modified by developers not familiar with code base?
  - Inspections/Reviews may be needed

Spaghetti Tests

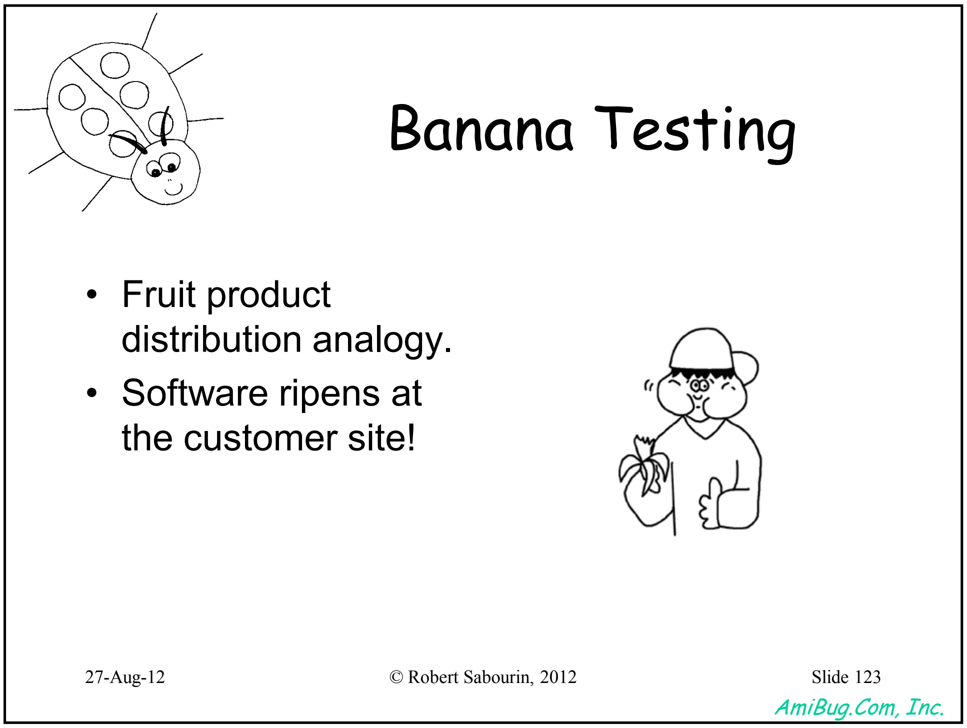
- Term describing a disorganized collection of automated test scripts.
  - Similar to Spaghetti Code
  - Tests are hard to maintain
  - Get more complex when modified
  - Hard to understand
  - Lack design

Cluster Fail Over Testing

- Multiple server systems
  - Ensure that if, for any reason, one node of a system fails that work is distributed to other nodes
  - Common in horizontally scalable Web or e-Commerce applications

Bug Forensics

- When a computer system abnormally stops or unexpectedly fails, Bug Forensics are the actions taken to identify the exact time, place and cause of system death
- *Quincy TV metaphor*

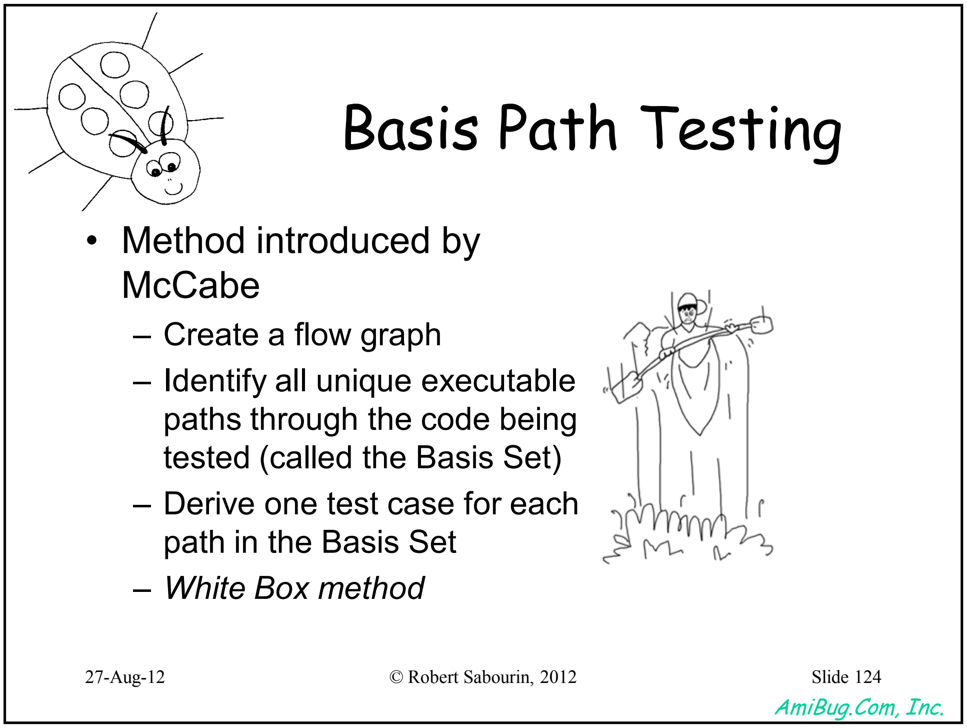


Banana Testing

- Fruit product distribution analogy.
- Software ripens at the customer site!

27-Aug-12 © Robert Sabourin, 2012 Slide 123

AmiBug.Com, Inc.

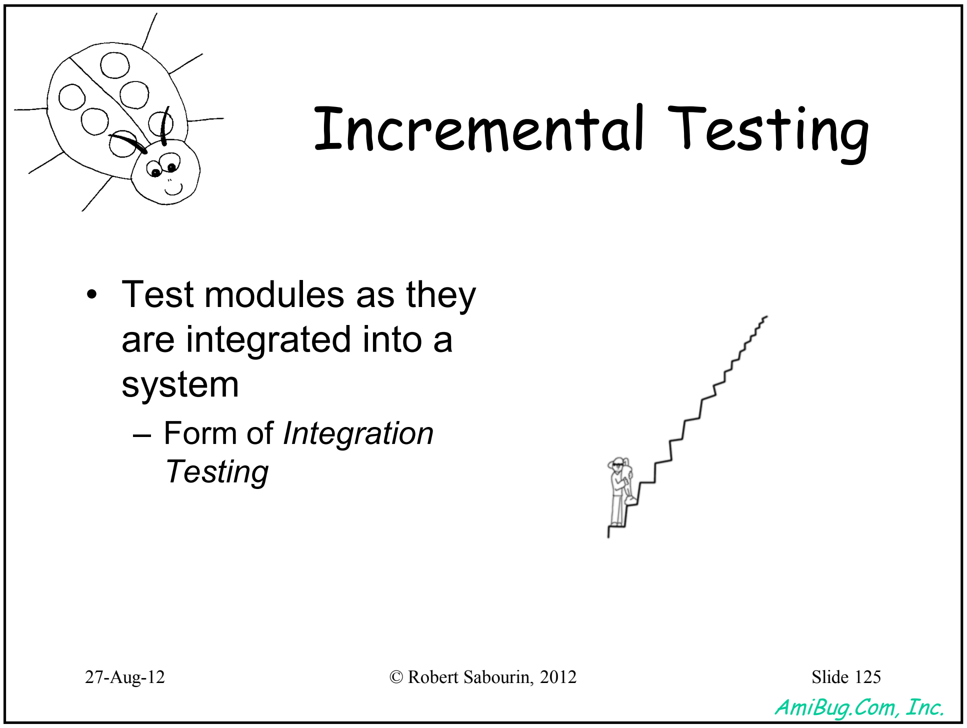


Basis Path Testing

- Method introduced by McCabe
  - Create a flow graph
  - Identify all unique executable paths through the code being tested (called the Basis Set)
  - Derive one test case for each path in the Basis Set
  - *White Box method*

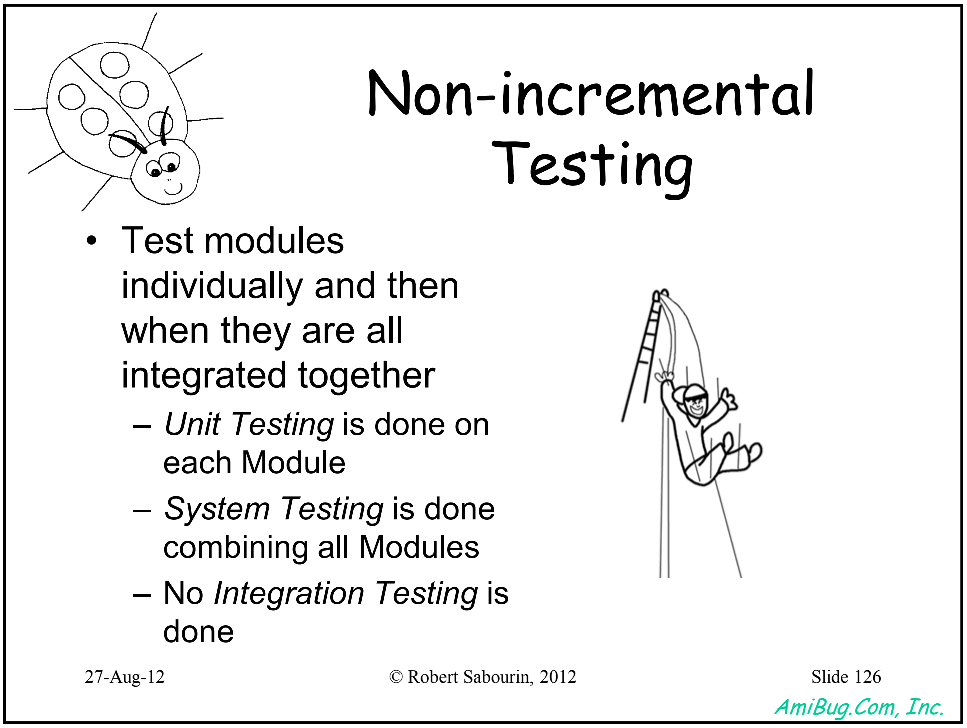
27-Aug-12 © Robert Sabourin, 2012 Slide 124

AmiBug.Com, Inc.



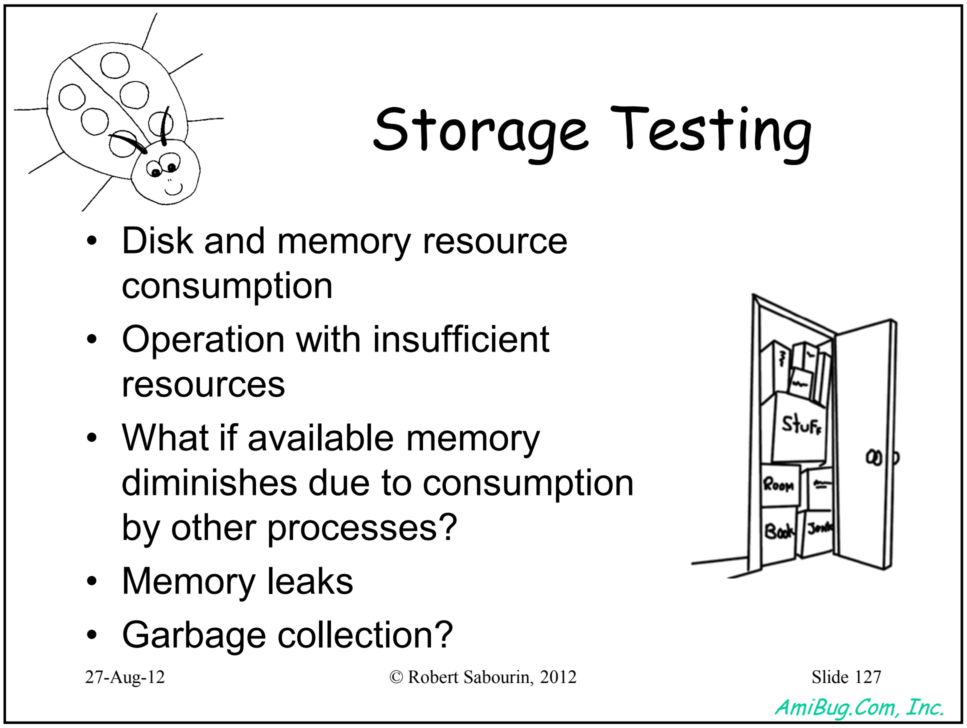
Incremental Testing

- Test modules as they are integrated into a system
  - Form of *Integration Testing*



Non-incremental Testing

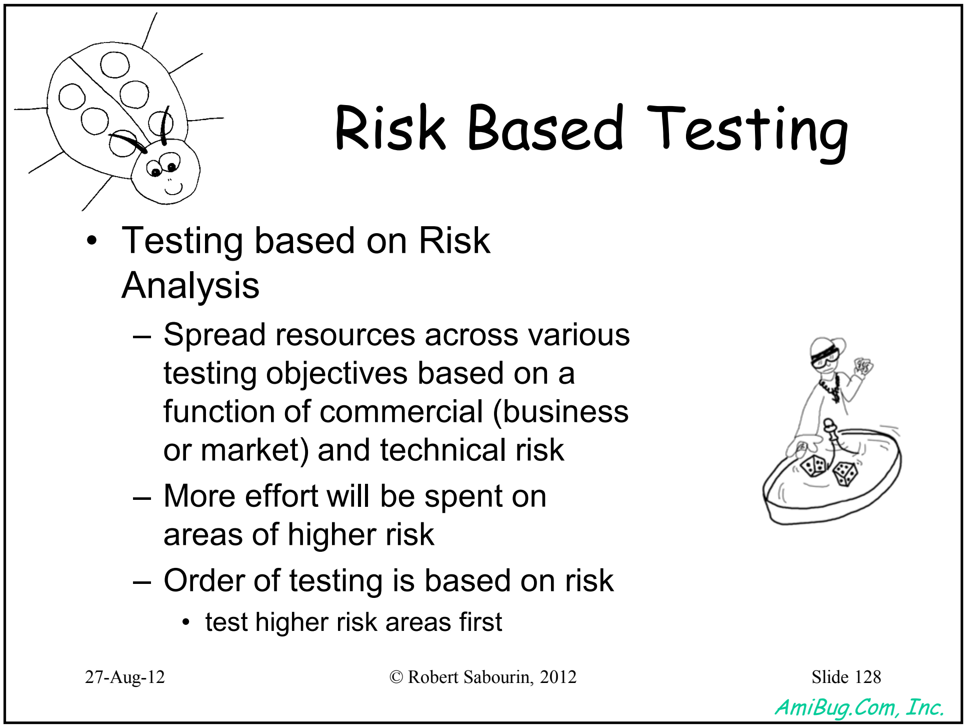
- Test modules individually and then when they are all integrated together
  - *Unit Testing* is done on each Module
  - *System Testing* is done combining all Modules
  - No *Integration Testing* is done



Stuff
Room
Book
Junk

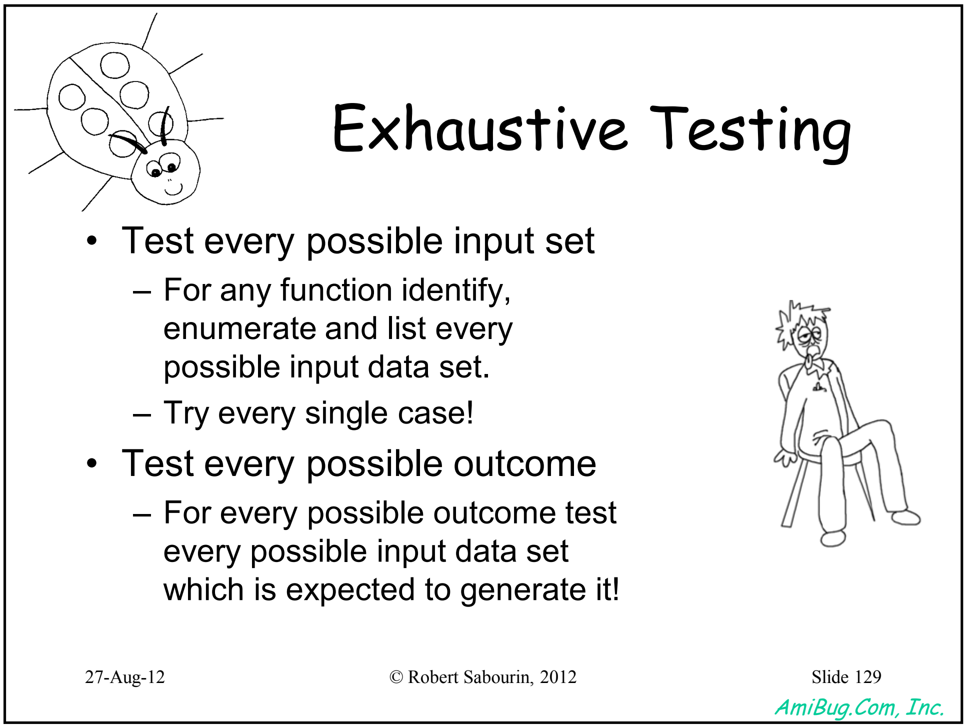
Storage Testing

- Disk and memory resource consumption
- Operation with insufficient resources
- What if available memory diminishes due to consumption by other processes?
- Memory leaks
- Garbage collection?



Risk Based Testing

- Testing based on Risk Analysis
  - Spread resources across various testing objectives based on a function of commercial (business or market) and technical risk
  - More effort will be spent on areas of higher risk
  - Order of testing is based on risk
    - test higher risk areas first

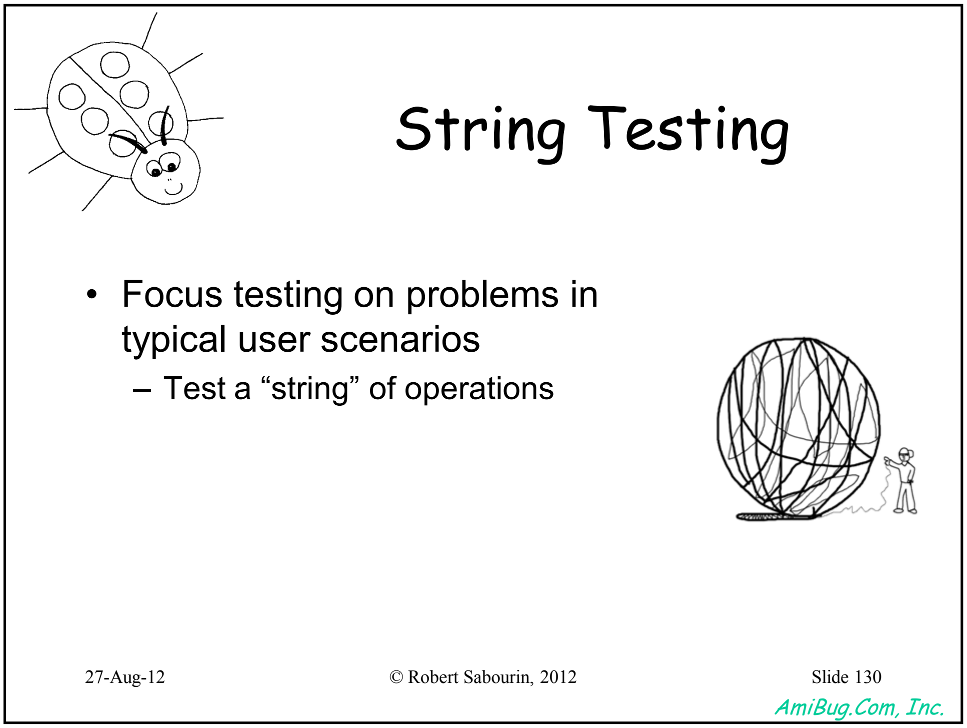


Exhaustive Testing

- Test every possible input set
  - For any function identify, enumerate and list every possible input data set.
  - Try every single case!
- Test every possible outcome
  - For every possible outcome test every possible input data set which is expected to generate it!

27-Aug-12 © Robert Sabourin, 2012 Slide 129

AmiBug.Com, Inc.

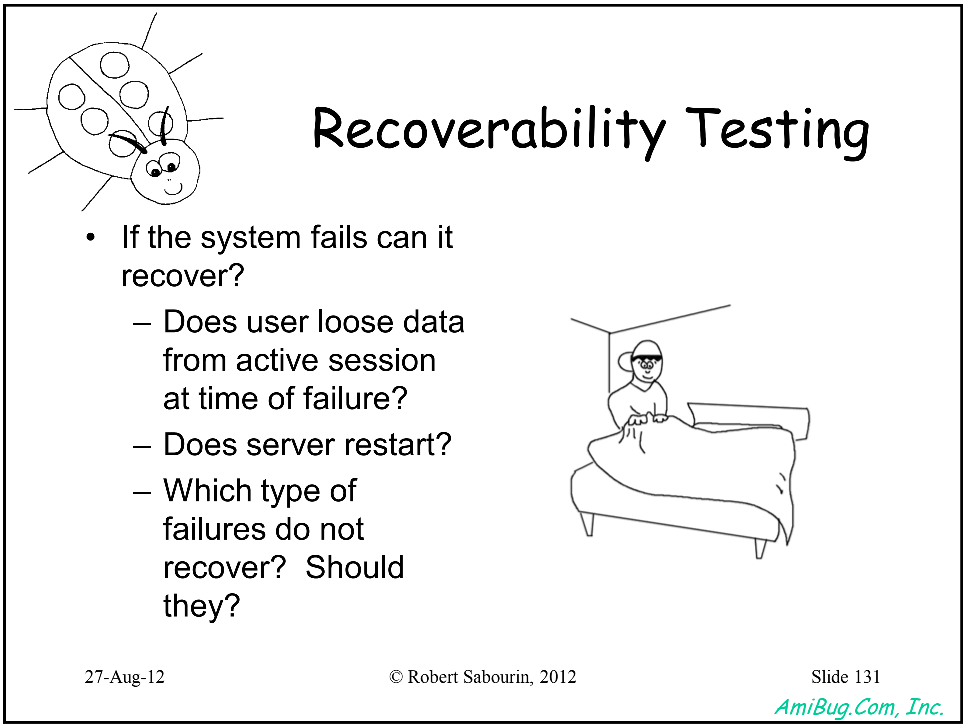


String Testing

- Focus testing on problems in typical user scenarios
  - Test a “string” of operations

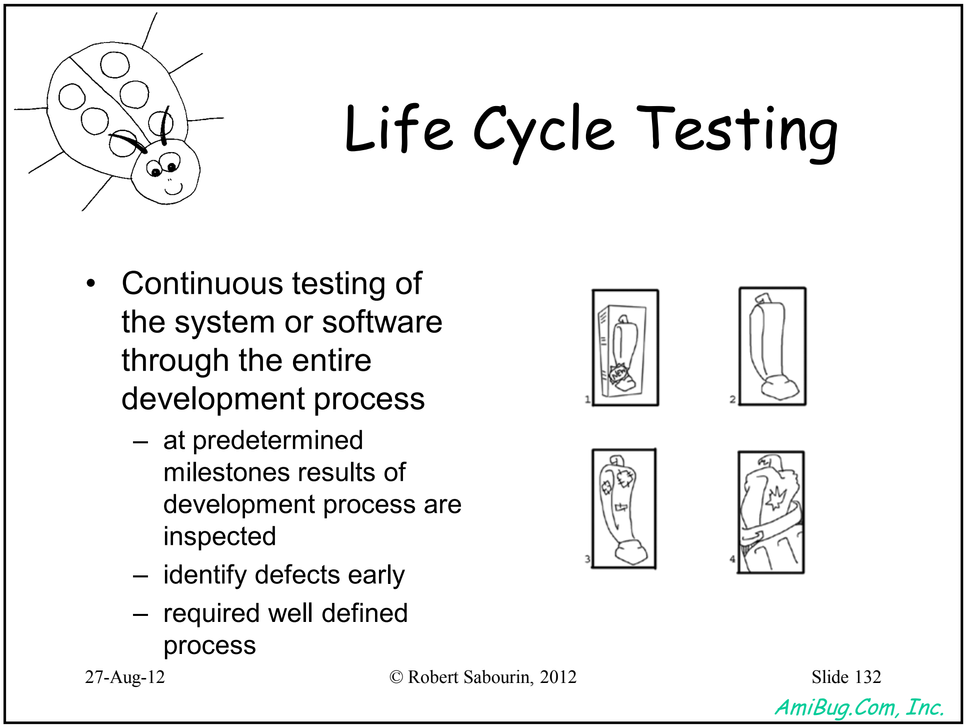
27-Aug-12 © Robert Sabourin, 2012 Slide 130

AmiBug.Com, Inc.



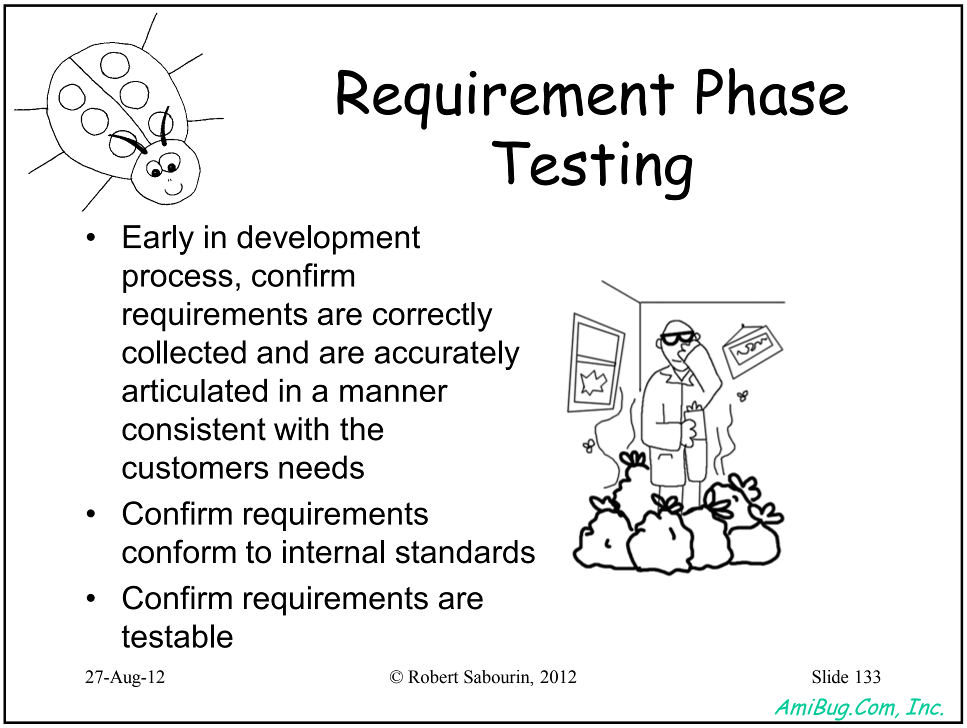
Recoverability Testing

- If the system fails can it recover?
  - Does user loose data from active session at time of failure?
  - Does server restart?
  - Which type of failures do not recover? Should they?



Life Cycle Testing

- Continuous testing of the system or software through the entire development process
  - at predetermined milestones results of development process are inspected
  - identify defects early
  - required well defined process

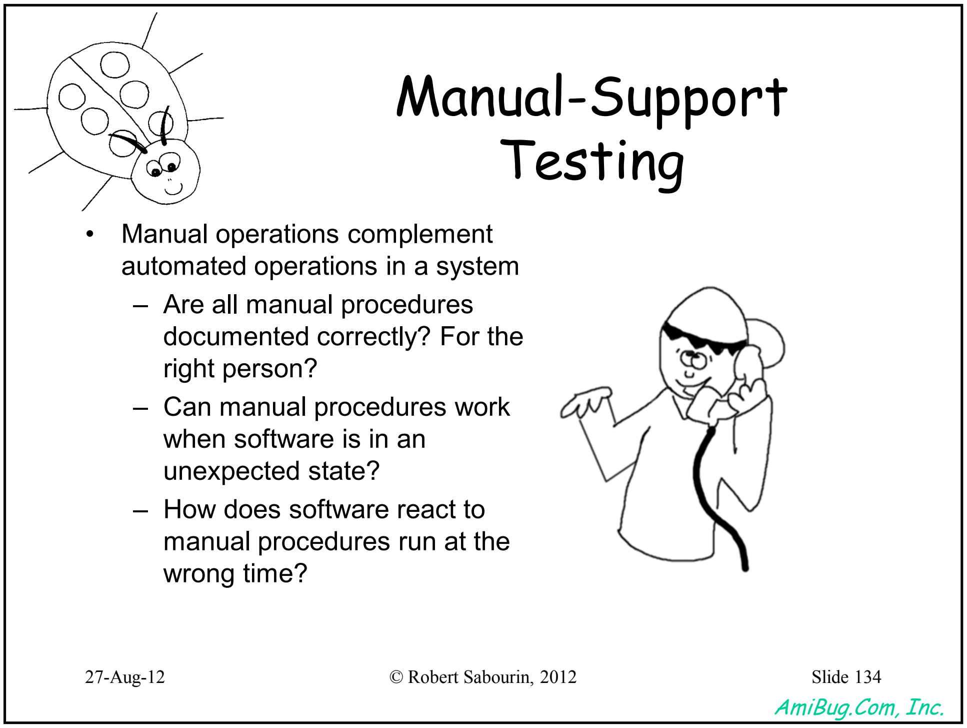


Requirement Phase Testing

- Early in development process, confirm requirements are correctly collected and are accurately articulated in a manner consistent with the customers needs
- Confirm requirements conform to internal standards
- Confirm requirements are testable

27-Aug-12 © Robert Sabourin, 2012 Slide 133

AmiBug.Com, Inc.

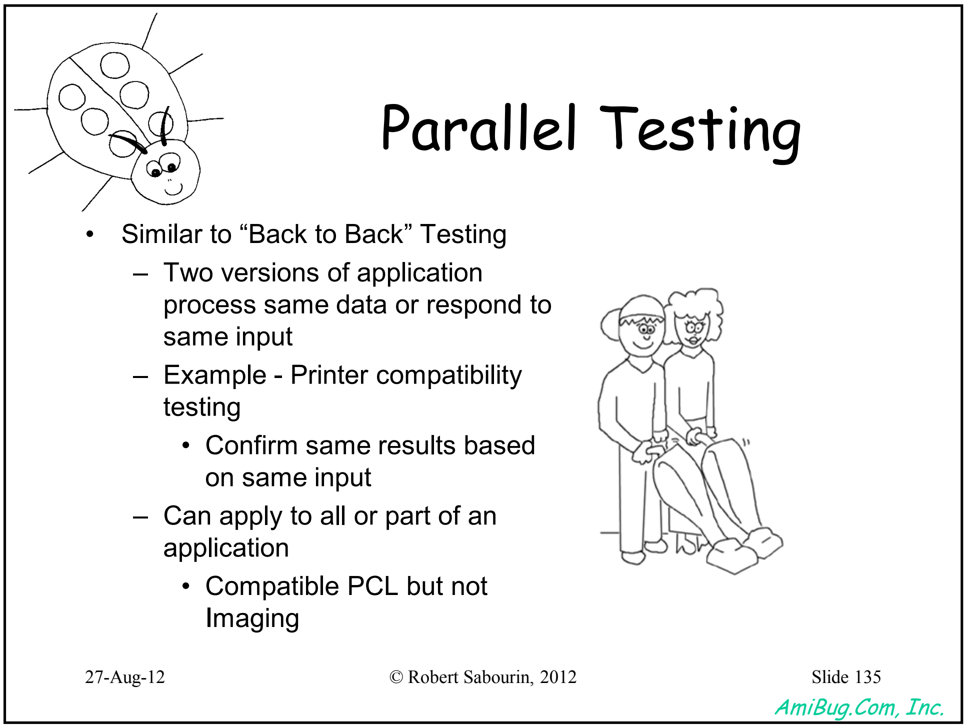


Manual-Support Testing

- Manual operations complement automated operations in a system
  - Are all manual procedures documented correctly? For the right person?
  - Can manual procedures work when software is in an unexpected state?
  - How does software react to manual procedures run at the wrong time?

27-Aug-12 © Robert Sabourin, 2012 Slide 134

AmiBug.Com, Inc.

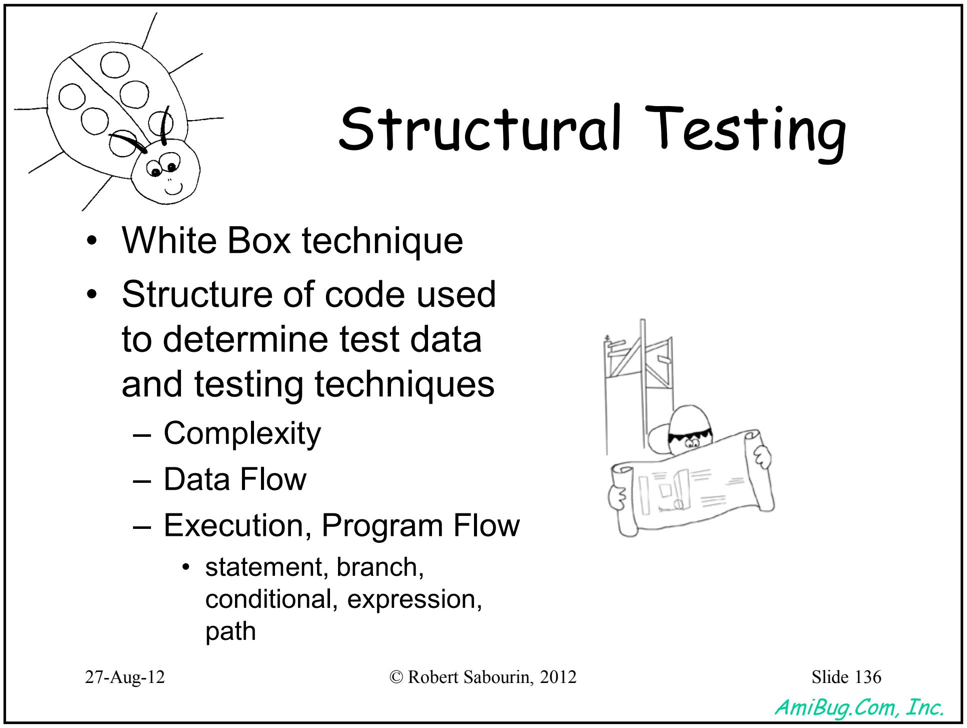


Parallel Testing

- Similar to “Back to Back” Testing
  - Two versions of application process same data or respond to same input
  - Example - Printer compatibility testing
    - Confirm same results based on same input
  - Can apply to all or part of an application
    - Compatible PCL but not Imaging

27-Aug-12 Slide 135

AmiBug.Com, Inc.

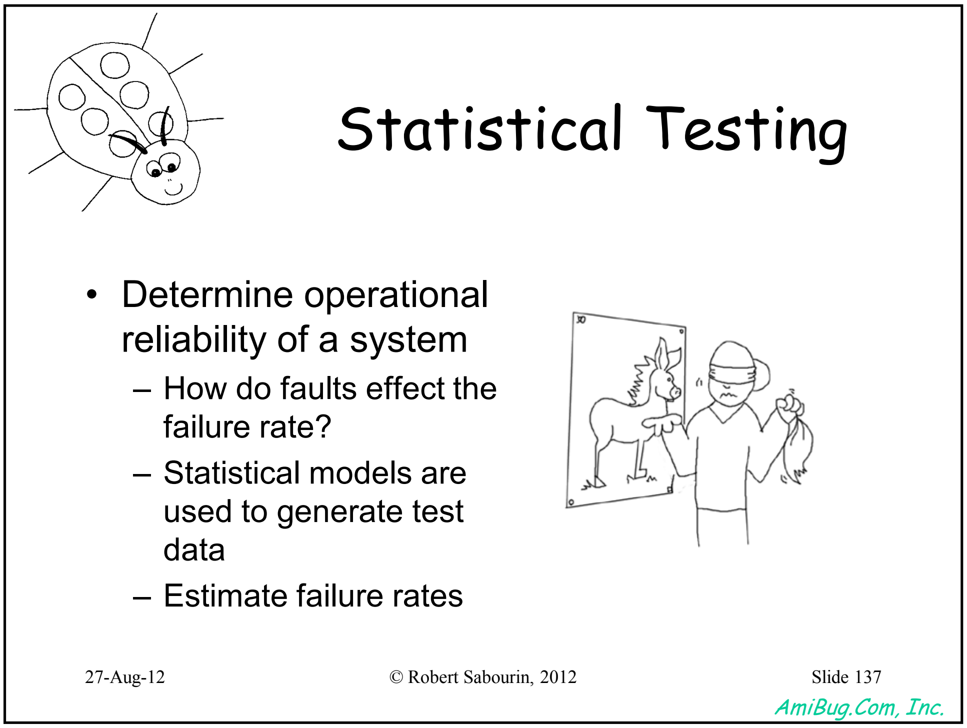


Structural Testing

- White Box technique
- Structure of code used to determine test data and testing techniques
  - Complexity
  - Data Flow
  - Execution, Program Flow
    - statement, branch, conditional, expression, path

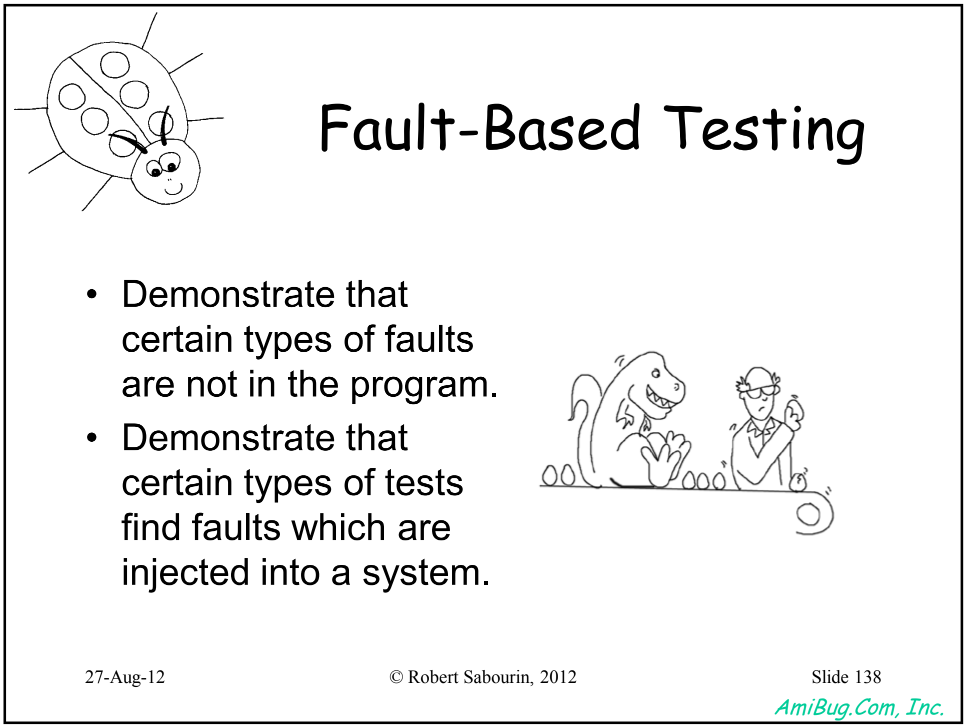
27-Aug-12 Slide 136

AmiBug.Com, Inc.



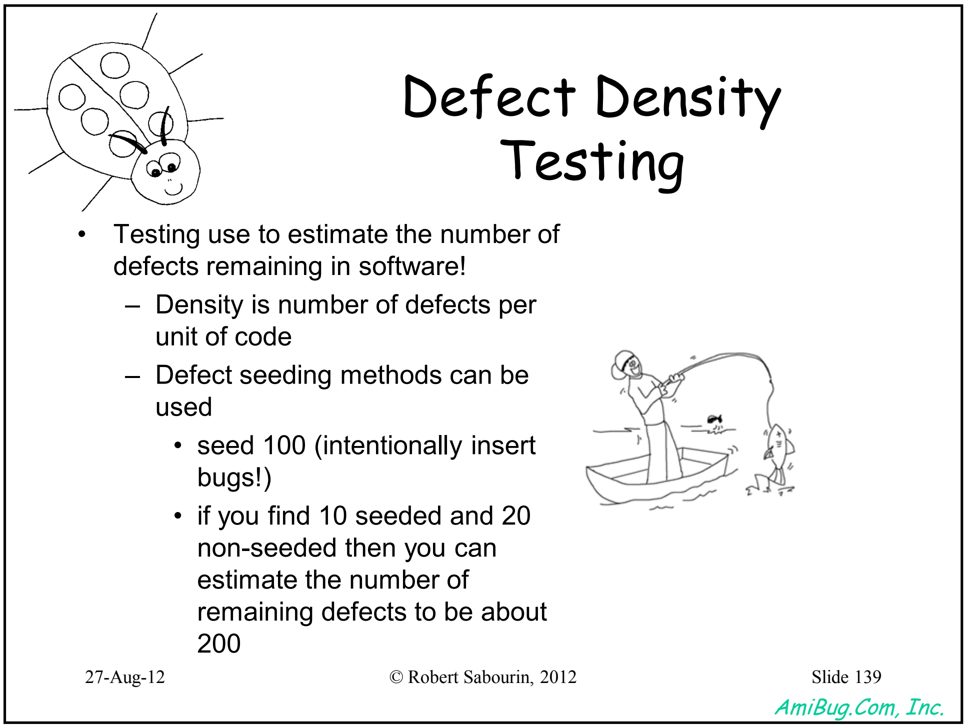
Statistical Testing

- Determine operational reliability of a system
  - How do faults effect the failure rate?
  - Statistical models are used to generate test data
  - Estimate failure rates



Fault-Based Testing

- Demonstrate that certain types of faults are not in the program.
- Demonstrate that certain types of tests find faults which are injected into a system.

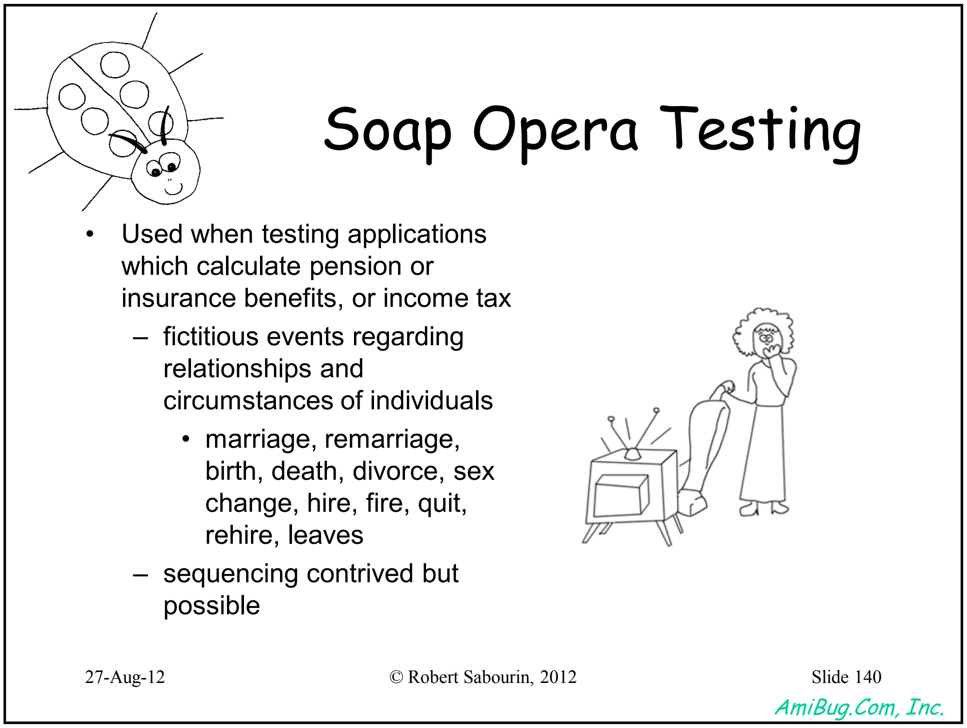


Defect Density Testing

- Testing use to estimate the number of defects remaining in software!
  - Density is number of defects per unit of code
  - Defect seeding methods can be used
    - seed 100 (intentionally insert bugs!)
    - if you find 10 seeded and 20 non-seeded then you can estimate the number of remaining defects to be about 200

27-Aug-12 © Robert Sabourin, 2012 Slide 139

AmiBug.Com, Inc.

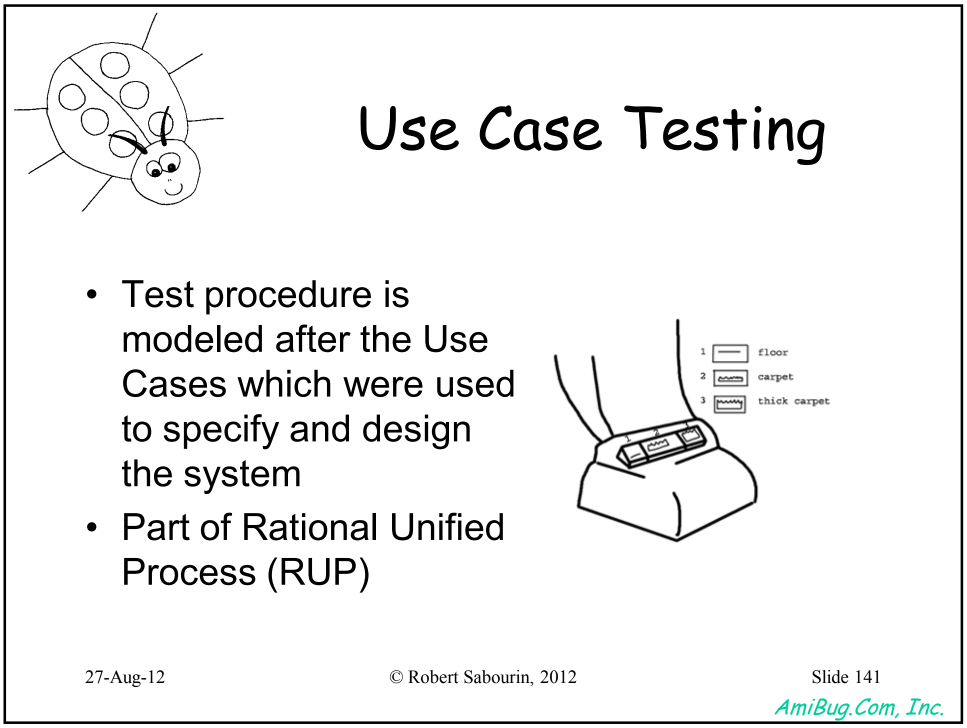


Soap Opera Testing

- Used when testing applications which calculate pension or insurance benefits, or income tax
  - fictitious events regarding relationships and circumstances of individuals
    - marriage, remarriage, birth, death, divorce, sex change, hire, fire, quit, rehire, leaves
  - sequencing contrived but possible

27-Aug-12 © Robert Sabourin, 2012 Slide 140

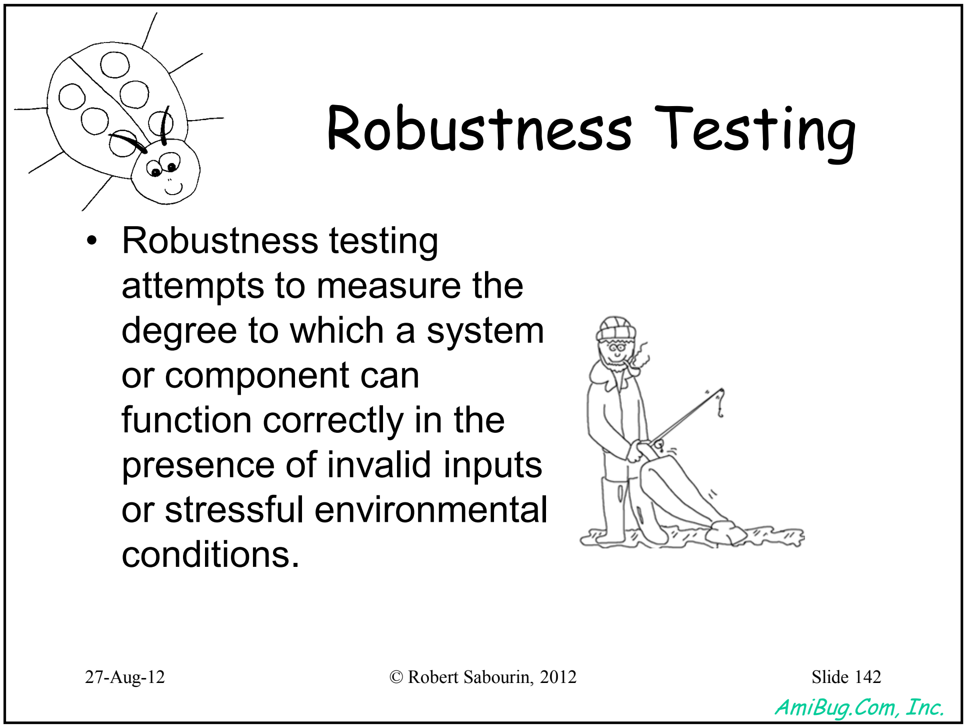
AmiBug.Com, Inc.



Use Case Testing

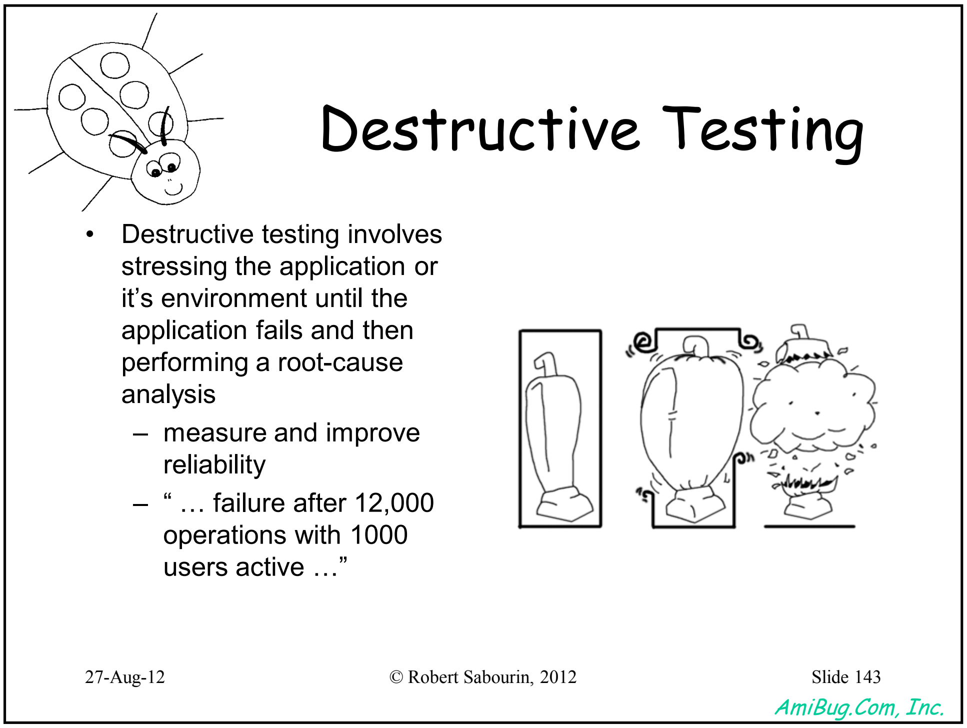
- Test procedure is modeled after the Use Cases which were used to specify and design the system
- Part of Rational Unified Process (RUP)

1 floor
2 carpet
3 thick carpet



Robustness Testing

- Robustness testing attempts to measure the degree to which a system or component can function correctly in the presence of invalid inputs or stressful environmental conditions.

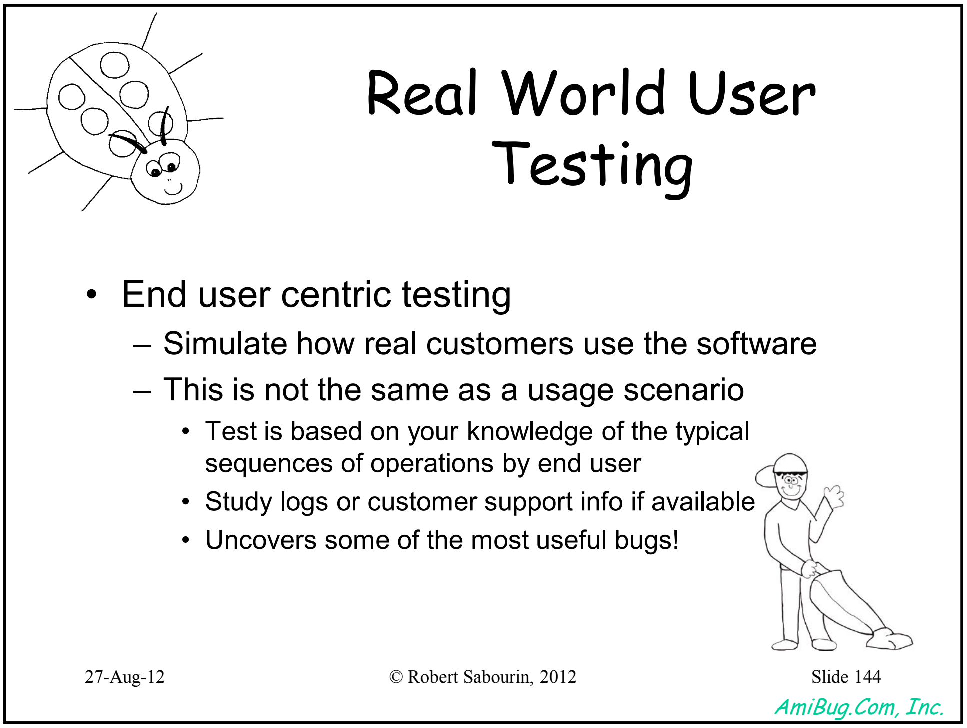


Destructive Testing

- Destructive testing involves stressing the application or it’s environment until the application fails and then performing a root-cause analysis
  - measure and improve reliability
  - “ … failure after 12,000 operations with 1000 users active …”

27-Aug-12 © Robert Sabourin, 2012 Slide 143

AmiBug.Com, Inc.

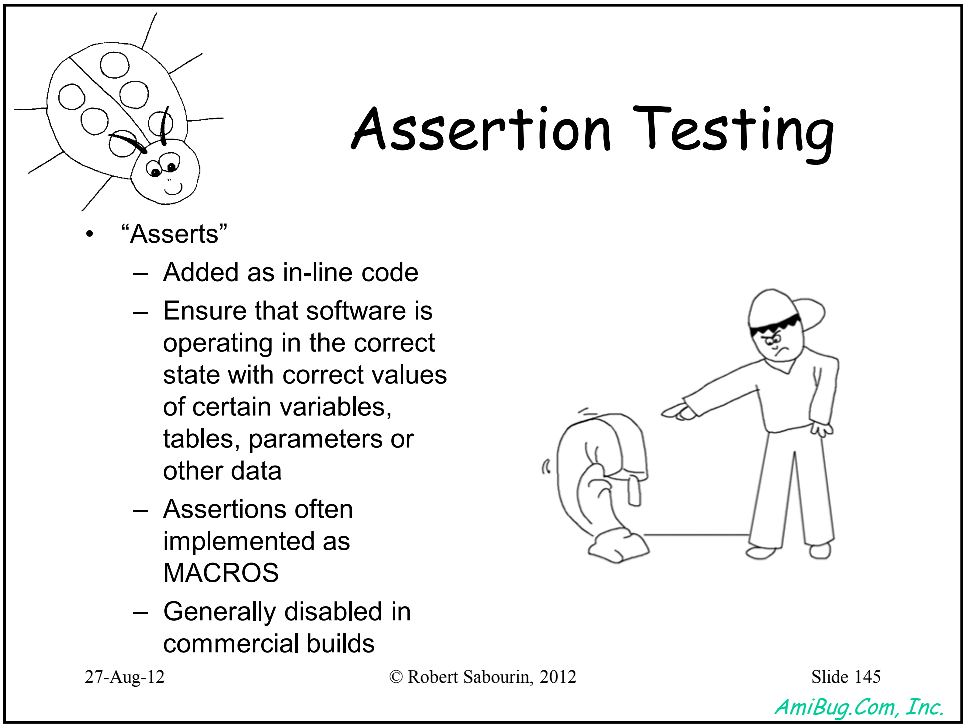


Real World User Testing

- End user centric testing
  - Simulate how real customers use the software
  - This is not the same as a usage scenario
    - Test is based on your knowledge of the typical sequences of operations by end user
    - Study logs or customer support info if available
    - Uncovers some of the most useful bugs!

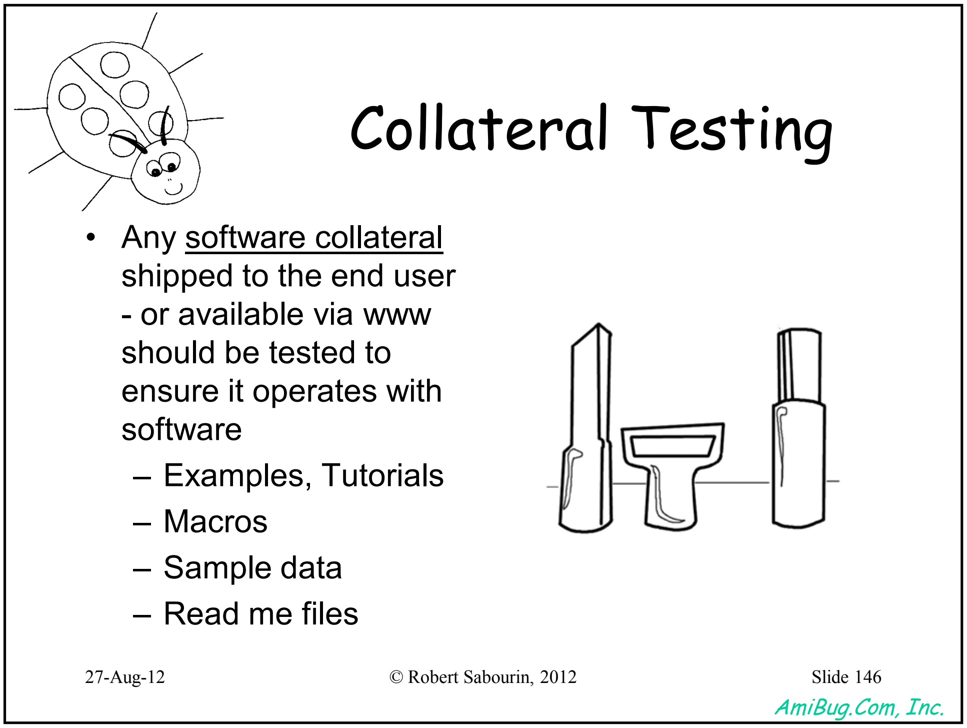
27-Aug-12 © Robert Sabourin, 2012 Slide 144

AmiBug.Com, Inc.



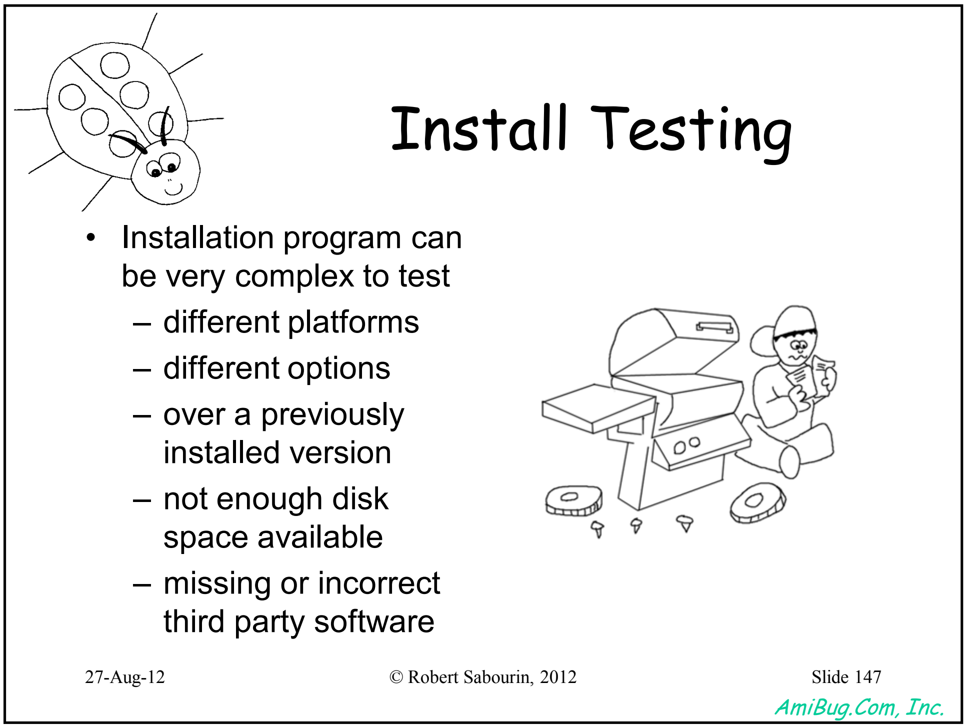
Assertion Testing

- “Asserts”
  - Added as in-line code
  - Ensure that software is operating in the correct state with correct values of certain variables, tables, parameters or other data
  - Assertions often implemented as MACROS
  - Generally disabled in commercial builds



Collateral Testing

- Any <u>software collateral</u> shipped to the end user - or available via www should be tested to ensure it operates with software
  - Examples, Tutorials
  - Macros
  - Sample data
  - Read me files

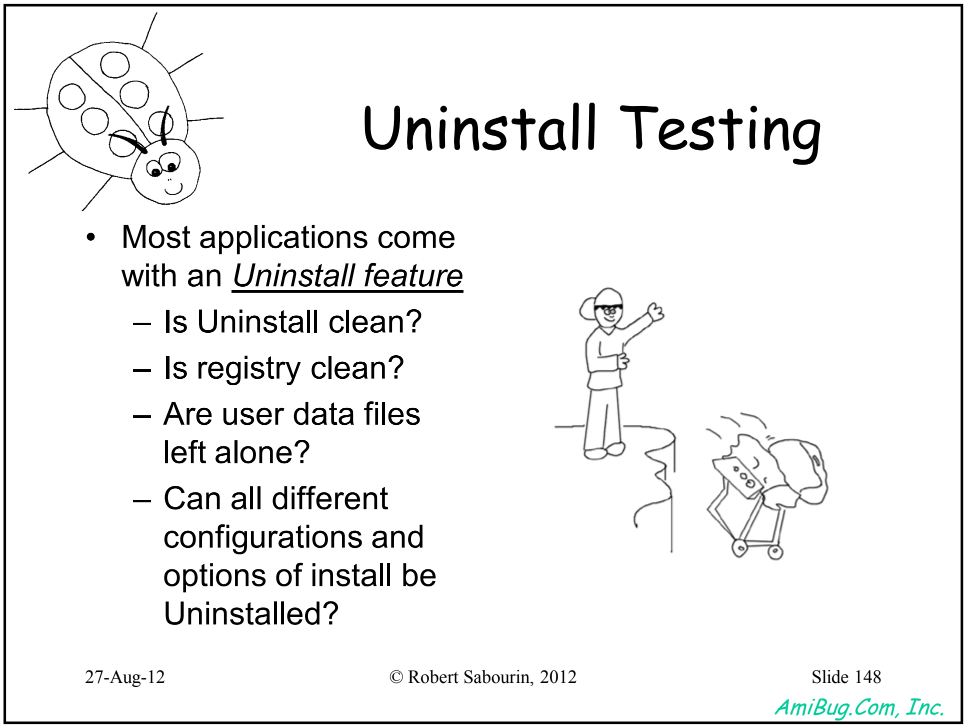


Install Testing

- Installation program can be very complex to test
  - different platforms
  - different options
  - over a previously installed version
  - not enough disk space available
  - missing or incorrect third party software

27-Aug-12 © Robert Sabourin, 2012 Slide 147

AmiBug.Com, Inc.



Uninstall Testing

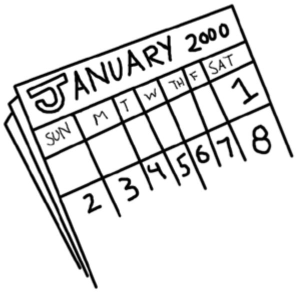
- Most applications come with an *Uninstall feature*
  - Is Uninstall clean?
  - Is registry clean?
  - Are user data files left alone?
  - Can all different configurations and options of install be Uninstalled?

27-Aug-12 © Robert Sabourin, 2012 Slide 148

AmiBug.Com, Inc.

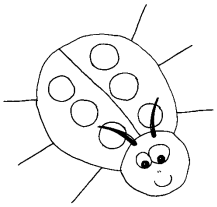
Y2K Testing

- Year 2000 testing is still needed!
  - Dominant software testing issue up to and including Year 2000
  - Testing of any date aware aspect of the application to confirm handling of forward and backward time calculations, dates, leap years
  - Examples: age computation, expiry dates



Milestone Testing

- Testing which must be passed before software is considered to have completed a mileston
  - For all project milestones stakeholders agree on milestone passage criteria and suitable testing which confirms the achievement!
  - Decide this early to avoid conflict and politics!

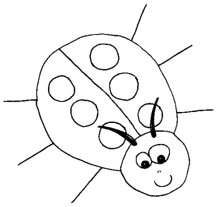


Agile Testing

- Testing tasks which take place as part of an Agile development process

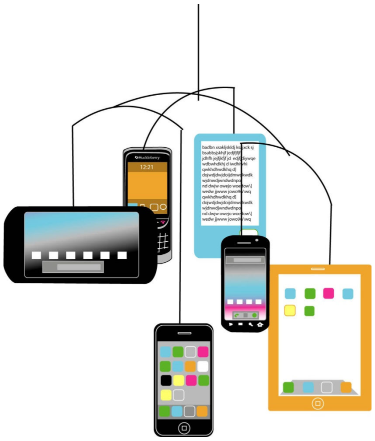
27-Aug-12 © Robert Sabourin, 2012 Slide 151

AmiBug.Com, Inc.



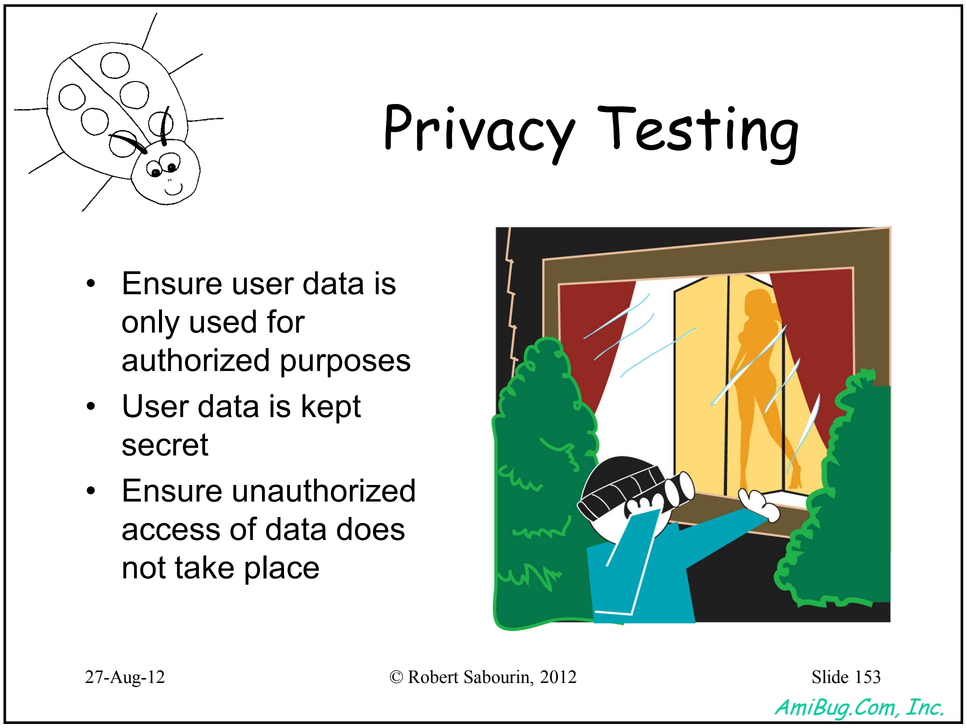
Mobile Testing

- Testing software integrated into handheld mobile computing devices



27-Aug-12 © Robert Sabourin, 2012 Slide 152

AmiBug.Com, Inc.

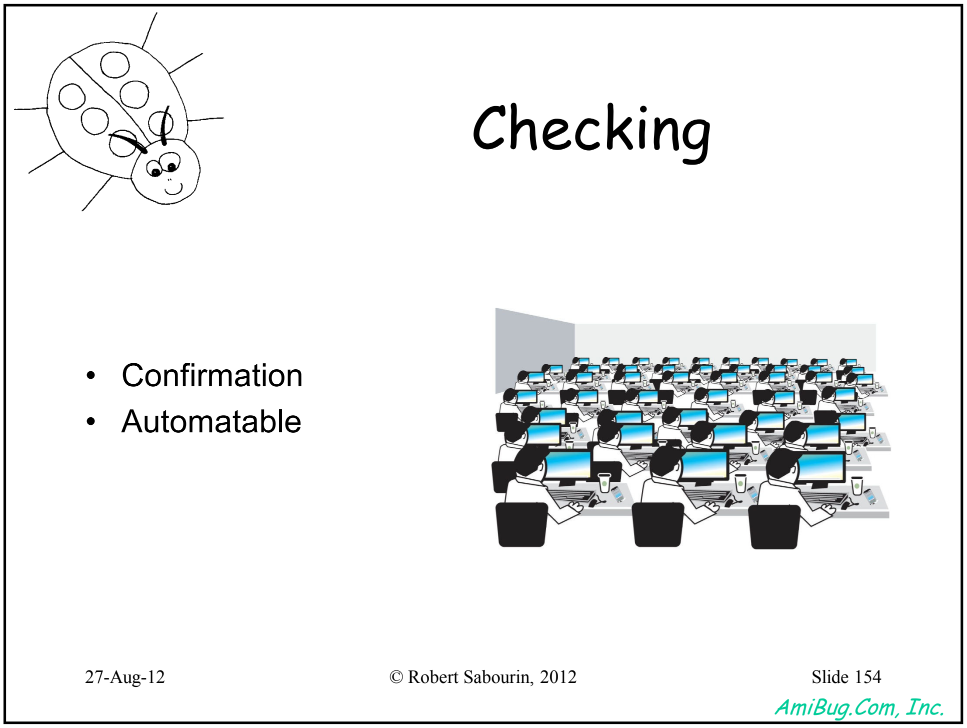


Privacy Testing

- Ensure user data is only used for authorized purposes
- User data is kept secret
- Ensure unauthorized access of data does not take place

27-Aug-12 © Robert Sabourin, 2012 Slide 153

AmiBug.Com, Inc.



Checking

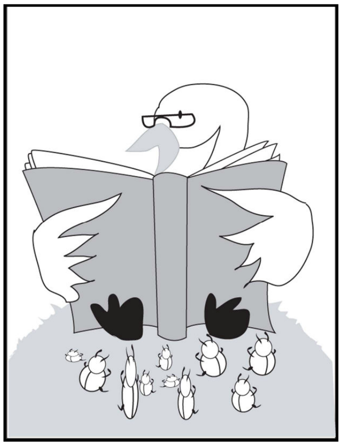
- Confirmation
- Automatable

27-Aug-12 © Robert Sabourin, 2012 Slide 154

AmiBug.Com, Inc.

Story Testing

- Acceptance tests of user stories
- Fit criteria
- Business facing tests



Thank You

- Questions?