



Software Testing

MODULE #10- SOFTWARE REGRESSION TESTING

Topic #1 – Software Regression Testing

**Instructor: Jerry Gao, Ph.D., Professor
San Jose State University**





MODULE #10- SOFTWARE REGRESSION TESTING

What is Software Regression Testing?

Why do we perform Regression Testing ?

Software Regression Strategy

Regression Testing Approaches

Class Firewall Concepts





MODULE #10– SOFTWARE REGRESSION TESTING

What is software regression testing?

Definition: *Regression testing* is the execution of a *set of test cases* on a program in order to ensure that its **revision** does not produce unintended faults. It is a testing process that is done to find the regressions in the system after doing any changes in the product during software maintenance phase.

What are the major testing objective?

- ✓ Retest changed components (or parts)
- ✓ Check the affected parts (or components)





MODULE #10- SOFTWARE REGRESSION TESTING

When do we need to perform Regression Testing ?

- ✓ Change in the requirements and code is modified according to the requirements.
- ✓ New feature is added to the software
- ✓ Defect fixing
- ✓ Performance issue fix

Regression Testing at different levels

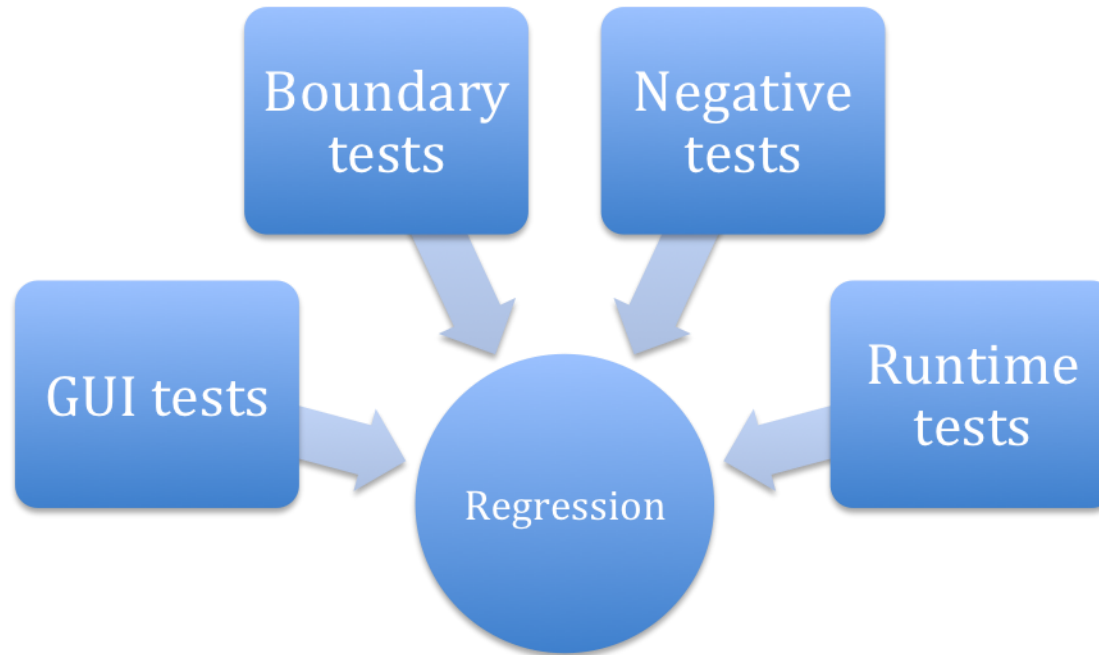
- ✓ Regression testing at the unit level
- ✓ Re-integration
- ✓ Regression testing at the function level
- ✓ Regression testing at the system level





MODULE #10- SOFTWARE REGRESSION TESTING

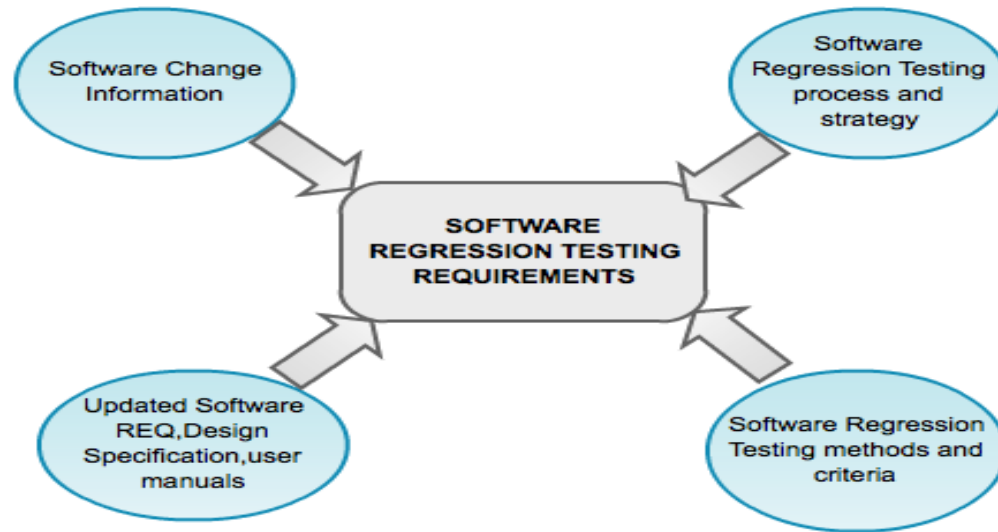
Regression Testing Focuses





MODULE #10- SOFTWARE REGRESSION TESTING

What do we need for Regression Testing



Who performs Regression Testing ?

- ✓ Developers- regression testing at the unit level or integration
- ✓ Test Engineers- regression testing at functional level
- ✓ QA and Test Engineers- regression testing at the system level





MODULE #10– SOFTWARE REGRESSION TESTING

Problems and Challenges in Software Regression Testing

Major problems in software regression testing:

How to identify software changes in a systematic way?

- REQ. specification changes
- Design specification changes
- Implementation (or source code) changes
- User manual changes
- Environment or technology changes
- Test changes

How to identify software change impacts in a systematic way?

- REQ impacts
- Design impacts
- Implementation impacts
- User impacts
- Test impacts





MODULE #10- SOFTWARE REGRESSION TESTING

Problems and Challenges in Software Regression Testing

Major problems in software regression testing:

Major regression testing problems:

- How to use a systematic method or tool to identify changed software parts?
- How to use a systematic method or tool to identify software change impacts?
- How to use a systematic method or tool to identify affected software test cases?
- How to reduce the re-test suites?
- How to select the test cases in a test suite?

Major challenge in software regression testing:

- How to minimize re-testing efforts, and achieve the adequate testing coverage ?





Software Regression Strategy

What is software regression strategy ?

✓ Software test strategy provides the basic strategy and guidelines to test engineers to perform software regression testing activities in a rational way.

Software Regression strategy usually refers to
--> a rational way to define regression testing scope, coverage criteria, re-testing sequence (or order) and re-integration orders.

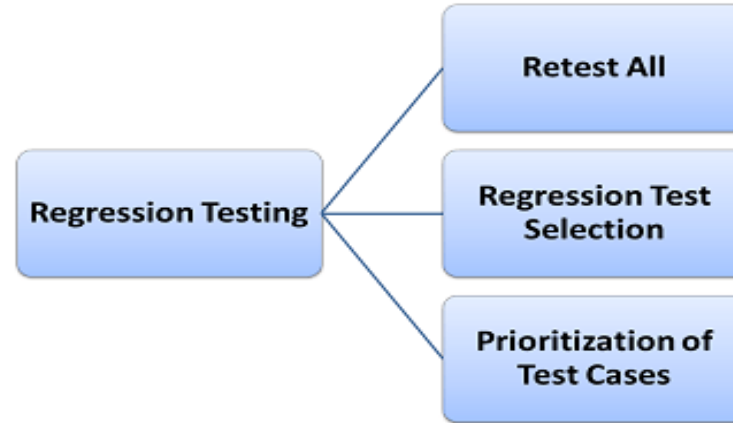
Software regression test models are needed to support the definition of software regression test strategy, test cases, and coverage criteria.





MODULE #10- SOFTWARE REGRESSION TESTING

Regression Testing Strategy



©guru99.com

1) **Reset all**

- ✓ This is one of the methods for regression testing in which all the tests in the existing test bucket or suite should be re-executed.
- ✓ This is very expensive as it requires huge time and resources.



MODULE #10- SOFTWARE REGRESSION TESTING

Regression Testing Strategy

2) Regression Test Selection

- ✓ Instead of re-executing the entire test suite, it is better to select part of test suite to be run.
- ✓ Test cases selected can be categorized as 1) Reusable Test Cases 2) Obsolete Test Cases.
- ✓ Re-usable Test cases can be used in succeeding regression cycles.
- ✓ Obsolete Test Cases can't be used in succeeding cycles

3) Prioritization of Test Cases

Prioritize the test cases depending on business impact, critical & frequently used functionalities .

Selection of test cases based on priority will greatly reduce the regression test suite.



MODULE #10- SOFTWARE REGRESSION TESTING

Regression Test Models

- ✓ control flow graph, state-based behavior diagram
- ✓ object-oriented class diagram
- ✓ scenario-based model
- ✓ component-based Regression model





MODULE #10– SOFTWARE REGRESSION TESTING

Areas to focus on regression testing

- ✓ Test cases which have frequent defects.
- ✓ Functionalities which are more visible to the users.
- ✓ Test cases which verify core features of the product.
- ✓ Test cases of Functionalities which has undergone more and recent changes.
- ✓ All Integration Test Cases.
- ✓ All Complex Test Cases.
- ✓ Boundary value test cases.
- ✓ Sample of Successful test cases.
- ✓ Sample of Failure test cases



Software Regression Process

- Understand and analyze various software changes.

- Understand and analyze software change impacts

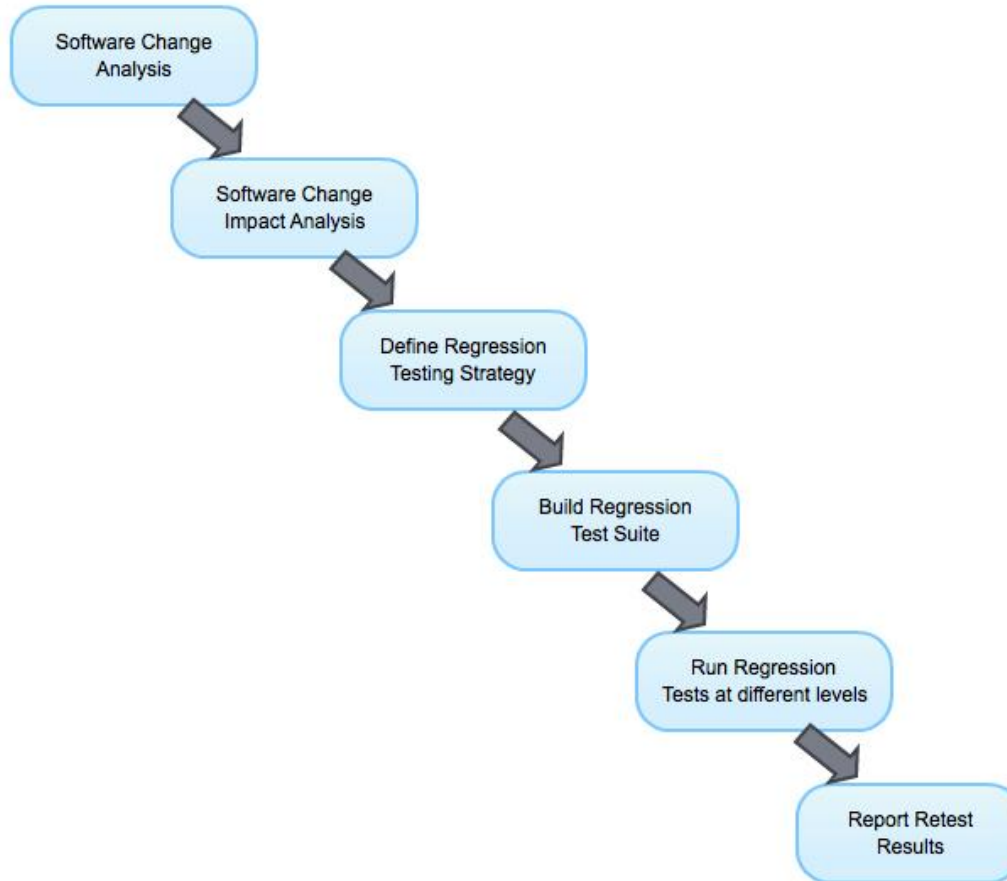
Step #4: Define, select, and reuse test cases to form a regression test suite

- re-testing at the unit level
- re-testing at integration level
- re-testing at the function level
- re-testing at the system level



MODULE #10- SOFTWARE REGRESSION TESTING

Software Regression Process





MODULE #10- SOFTWARE REGRESSION TESTING

Different Types of Software Changes

Requirements analysis	Requirements Spec. Changes <ul style="list-style-type: none">-> add new functional features-> change current function features-> delete existing function features
System Design	System architecture changes <ul style="list-style-type: none">-> change component interactions-> add new components/subsystems-> update existing components-> delete existing components
	High-level design doc. Changes <ul style="list-style-type: none">-> change state-based behaviors-> change component interfaces-> change database design-> change GUI design-> change function design



MODULE #10- SOFTWARE REGRESSION TESTING

Different Types of Software Changes

System Design Changes	<ul style="list-style-type: none">- Low-level design doc. Changes<ul style="list-style-type: none">-> change algorithm logic-> change component structure
System implementation	<ul style="list-style-type: none">- Component changes<ul style="list-style-type: none">- internal data types and names- internal structures, such as<ul style="list-style-type: none">--> class relationships--> control flow and data flow- internal functions
	<ul style="list-style-type: none">- Component interface changes<ul style="list-style-type: none">- call signatures- message interactions- protocol messages and formats
	<ul style="list-style-type: none">- Technology and/or language changes



MODULE #10– SOFTWARE REGRESSION TESTING

Software Changes Impact

Types of system changes	Types of product impacts
Requirements changes	Affect design, coding, and testing Document update
Design changes	Affect coding and tests Affect associated components Affect system architecture Affect related component interactions
Implementation changes	Affect test cases, test data, test scripts Affect test specification. Code change impact s
Test changes	Affect other tests. Affect test documentation
Document changes	Affect other documents.





Software Testing

MODULE #10- SOFTWARE REGRESSION TESTING

Topic #2 – Software Regression Testing Strategies

**Instructor: Jerry Gao, Ph.D., Professor
San Jose State University**





MODULE #10- SOFTWARE REGRESSION TESTING

Traditional Software Regression Strategy Based on The Firewall Concept

A Module-Based Firewall Concept for Software Regression Testing:

A module firewall in a program refers to a changed software module and a closure of all possible affected modules and related integration links in a program based on a control-flow graph.

With this firewall concept, we can reduce the software regression testing to a smaller scope -->

All modules and related integration links inside the firewall.

This implies that:

- re-test of the changed module and its affected modules
- re-integration for all of related integration links

Similarly, we can come out different kinds of firewalls based on various test models.

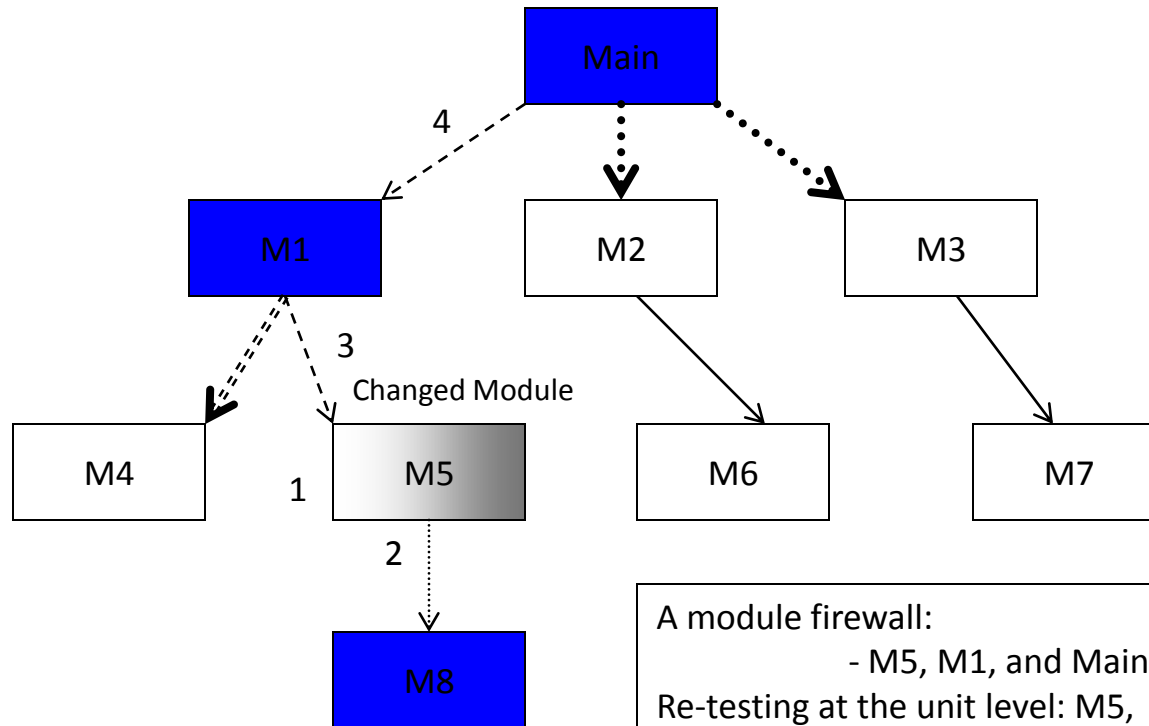


- data firewall, function firewall
- class firewall, state/transaction firewall



MODULE #10- SOFTWARE REGRESSION TESTING

A Changed Module Firewall for Regression Testing



A module firewall:

- M5, M1, and Main

Re-testing at the unit level: M5,

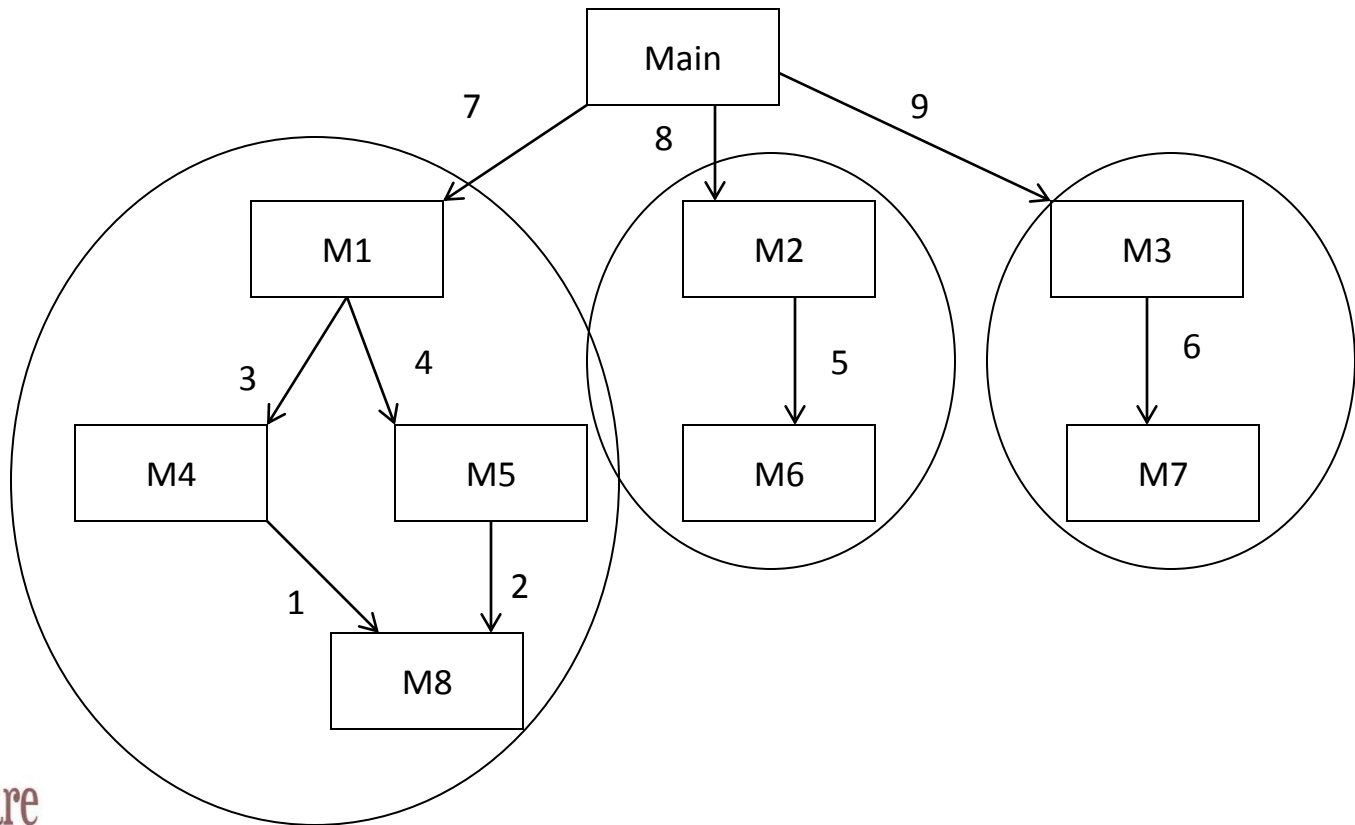
Re-integration:

2, 3, 4



MODULE #10- SOFTWARE REGRESSION TESTING

Regression Testing for State-Based Behavior Changes





Object-Oriented Software Regression Strategy

An OO Software Regression Testing Strategy Based on the Class Firewall Concept:

- Identify changed classes
- Identify affected classes using the concept of Class Firewall
- Apply the Class Test Order strategy to classes in a class firewall to perform class re-testing at the unit level
- Use the Class Test Order to re-integrate classes together.
- Select, reuse, and define test cases based on the class firewall and change information.





MODULE #10– SOFTWARE REGRESSION TESTING

The Class Firewall Concept

A class firewall concept in OO Software is very useful for OO regression testing.

What is a class firewall?

- A class firewall is a closure set of all classes that are directly or indirectly dependent on the changed class in an OO program.
- The class firewall provides the safe scope of regression testing for an OO software after changing a class.
- Similarly, we can apply to many changed classes.

Class Firewall Application:

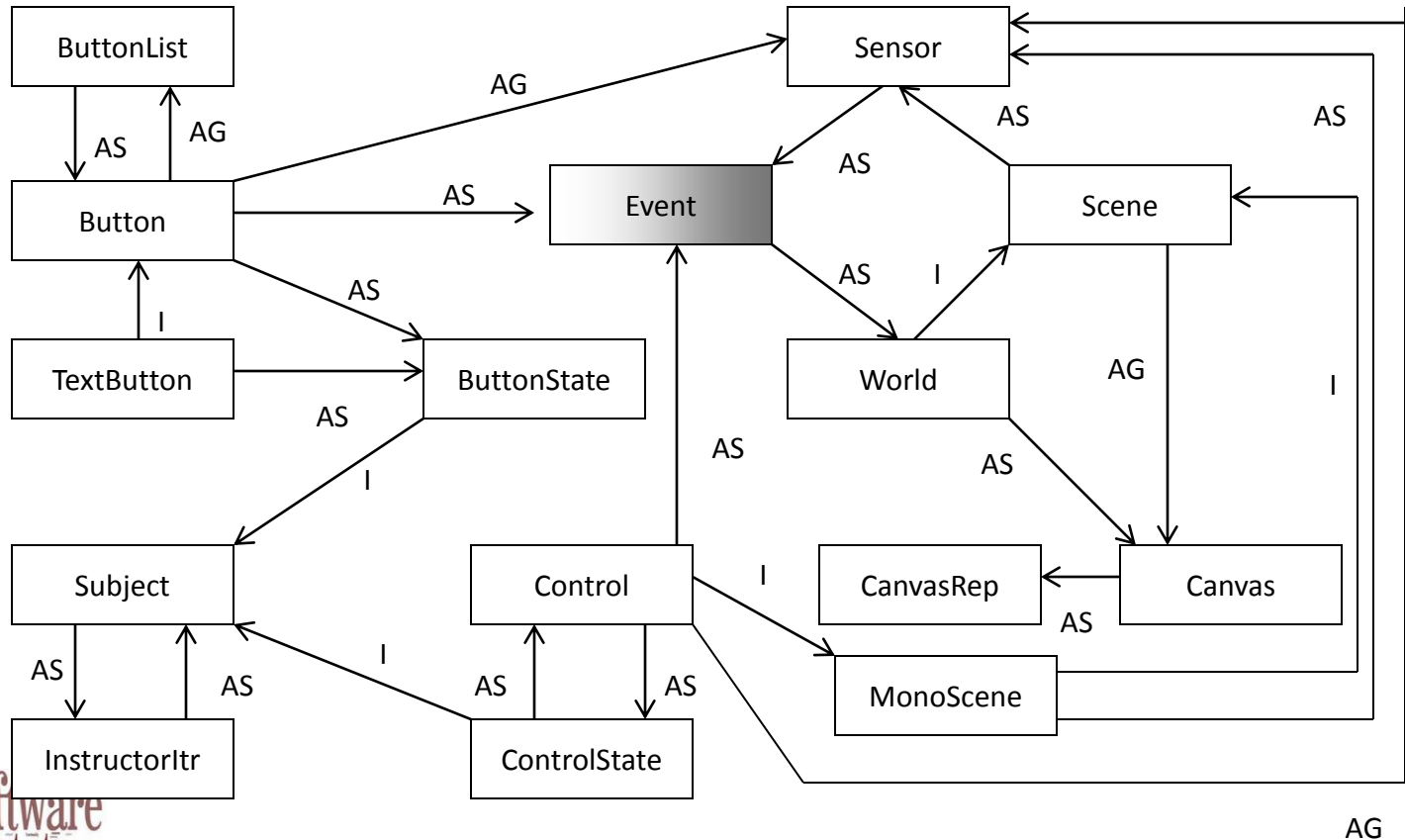
- With this class firewall concept, we can narrow down the class regression testing scope, including unit re-testing, and re-integration.
- Based on the class firewall and changed information, we can select, define, and reuse class test cases for regression testing.





MODULE #10- SOFTWARE REGRESSION TESTING

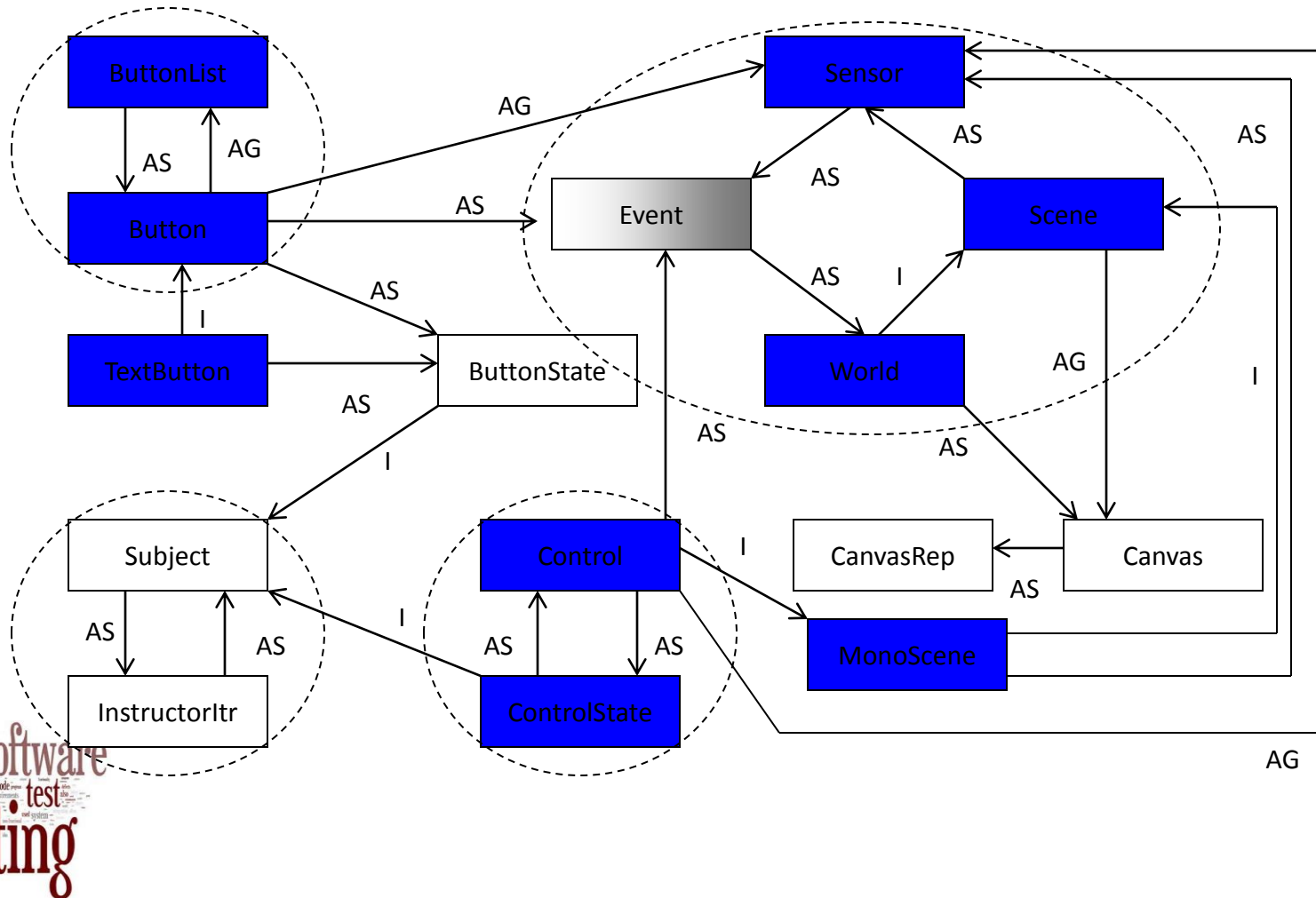
A Changed Class in An Class Relation Diagram





MODULE #10- SOFTWARE REGRESSION TESTING

A Class Firewall for an Object-Oriented Programs





MODULE #10- SOFTWARE REGRESSION TESTING

A Class Test Order for The Class Firewall

