Clover Demo: Test Coverage in Action

Sara Sprenkle

CISC879

March 15, 2007

Review: When Have You Tested Enough?

- Time? It’s been a couple hours/days/…
- Number of test cases executed? A lot!
- I asked my brother and he’s really smart and he says that it’s enough

Review: When Have You Tested Enough?

- Time? It’s been a couple hours/days/…
- Number of test cases executed? A lot!
- I asked my brother and he’s really smart and he says that it’s enough

➢ Need something more *systematic*: **Coverage Criteria**

Review: Uses of Coverage Criteria

Review: Uses of Coverage Criteria

- “Stopping” rule → sufficient testing
  - Avoid unnecessary, redundant tests

Review: Uses of Coverage Criteria

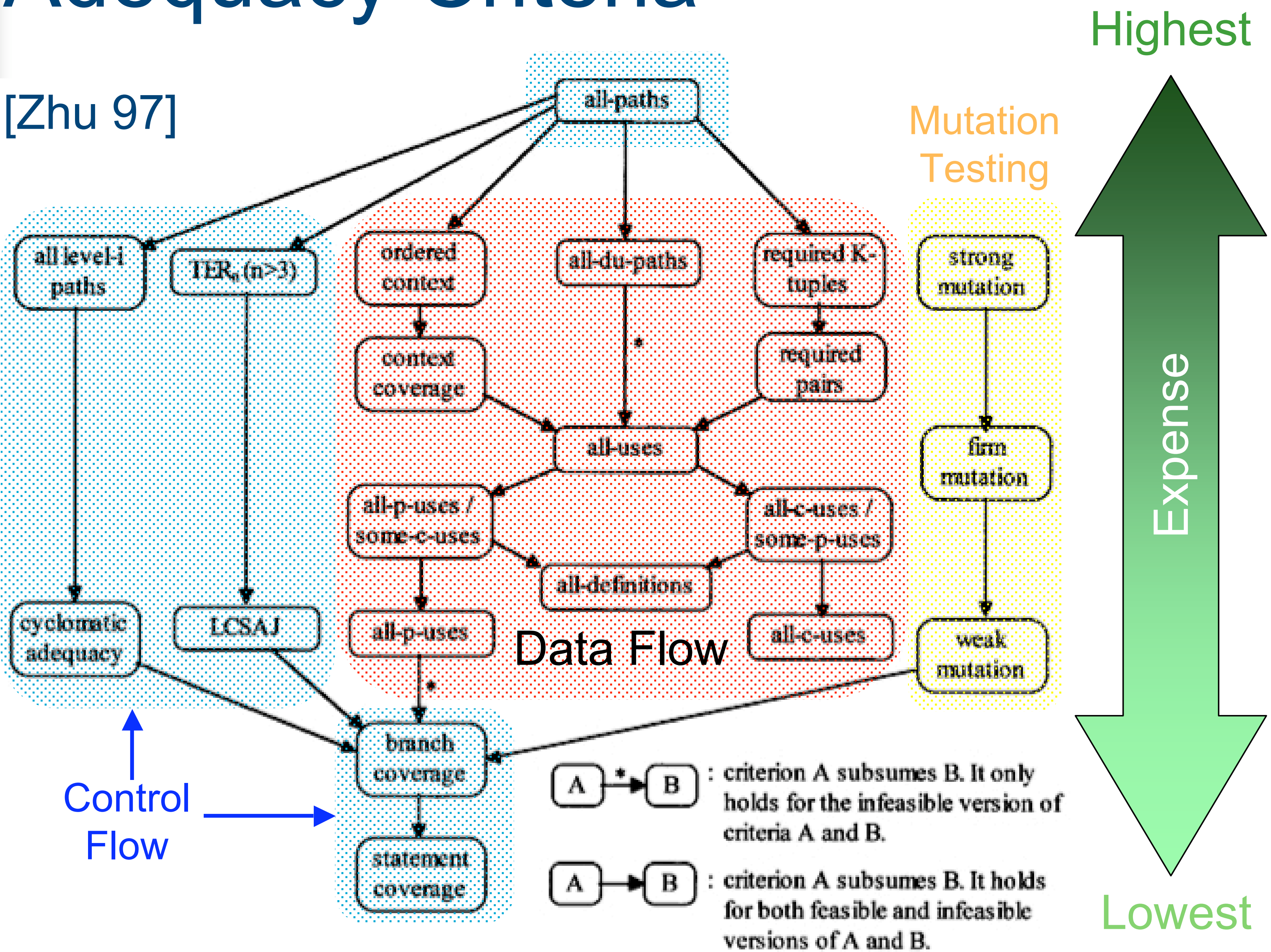
- "Stopping" rule → sufficient testing
  - Avoid unnecessary, redundant tests
- Measure test quality
  - Dependability estimate
  - Confidence in estimate

Review: Uses of Coverage Criteria

- "Stopping" rule → sufficient testing
  - Avoid unnecessary, redundant tests
- Measure test quality
  - Dependability estimate
  - Confidence in estimate
- Specify test cases
  - Describe additional test cases needed

Adequacy Criteria

[Zhu 97]

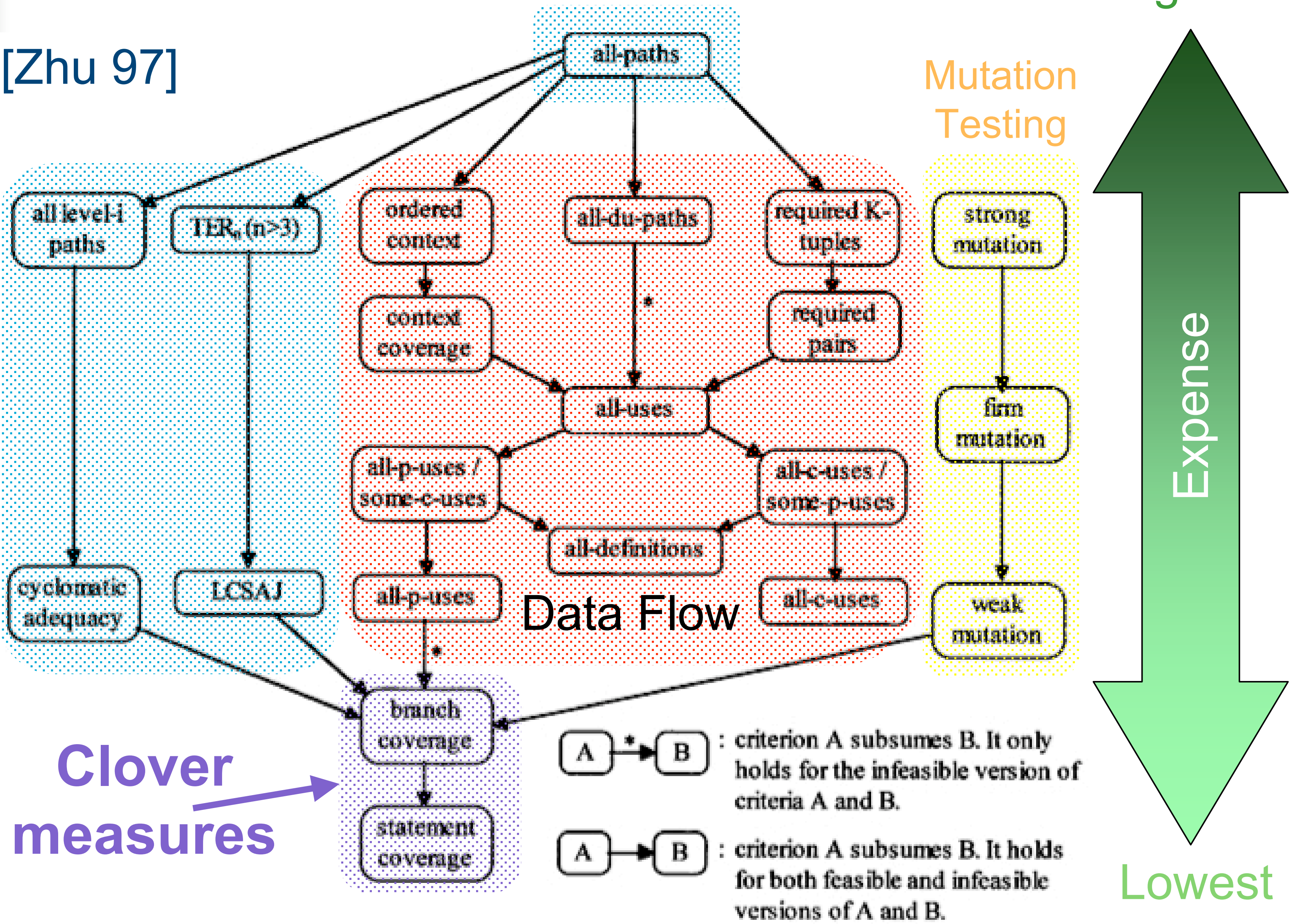


A —*→ B : criterion A subsumes B. It only holds for the infeasible version of criteria A and B.

A → B : criterion A subsumes B. It holds for both feasible and infeasible versions of A and B.

Adequacy Criteria

[Zhu 97]



: criterion A subsumes B. It only holds for the infeasible version of criteria A and B.

: criterion A subsumes B. It holds for both feasible and infeasible versions of A and B.

Statement Coverage

- Cover all **statements** in the program

```
public String exampleMethod(int num) {
    String string = null;
    if (num < 10) {
        string = "" + condition;
    }
    return string.trim();
}
```

Statement Coverage

- Cover all **statements** in the program

Test Suite:

    num=5

```
public String exampleMethod(int num) {
    String string = null;
    if (num < 10) {
        string = "" + condition;
    }
    return string.trim();
}
```

Statement Coverage

- Cover all **statements** in the program

Test Suite:

num=5

```
    public String exampleMethod(int num) {
✓       String string = null;
        if (num < 10) {
            string = "" + condition;
        }
        return string.trim();
    }
```

Statement Coverage

- Cover all **statements** in the program

Test Suite:

num=5

```
    public String exampleMethod(int num) {
✓       String string = null;
✓       if (num < 10) {
            string = "" + condition;
        }
        return string.trim();
    }
```

Statement Coverage

- Cover all **statements** in the program

Test Suite:

num=5

```
   public String exampleMethod(int num) {
✓      String string = null;
✓      if (num < 10) {
✓          string = "" + condition;
       }
       return string.trim();
   }
```

Statement Coverage

- Cover all **statements** in the program

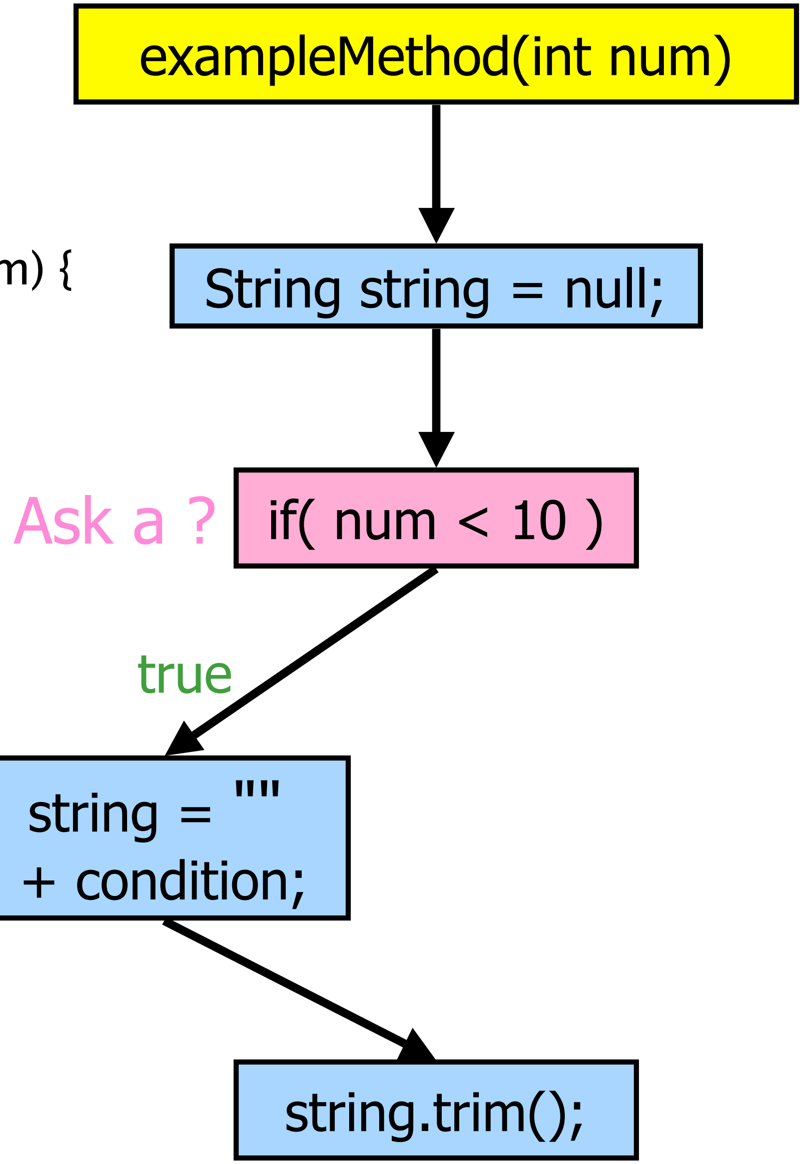
Test Suite:

num=5

```
   public String exampleMethod(int num) {
✓      String string = null;
✓      if (num < 10) {
✓          string = "" + condition;
       }
✓      return string.trim();
   }
```

Program Flow

```
public String exampleMethod(int num) {
    String string = null;
    if (num < 10) {
        string = "" + condition;
    }
    return string.trim();
}
```



exampleMethod(int num)

String string = null;

Ask a ?

if(num < 10)

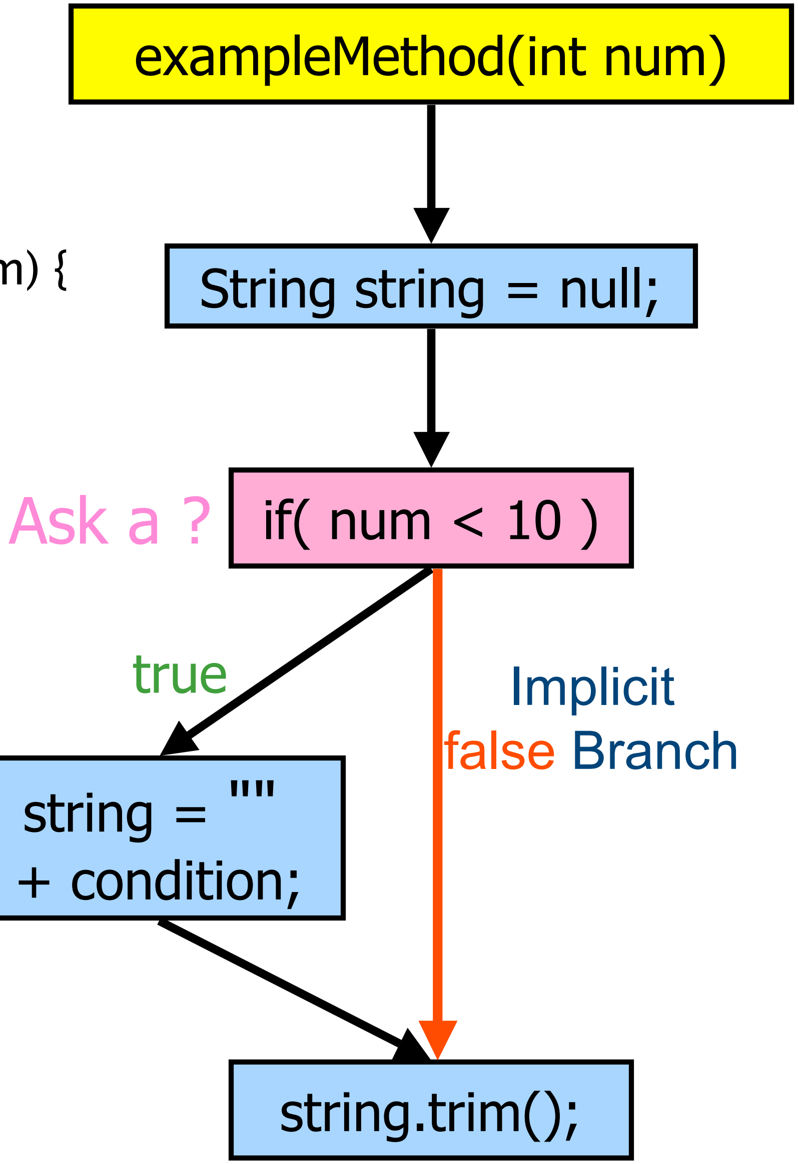
true

string = ""
+ condition;

string.trim();

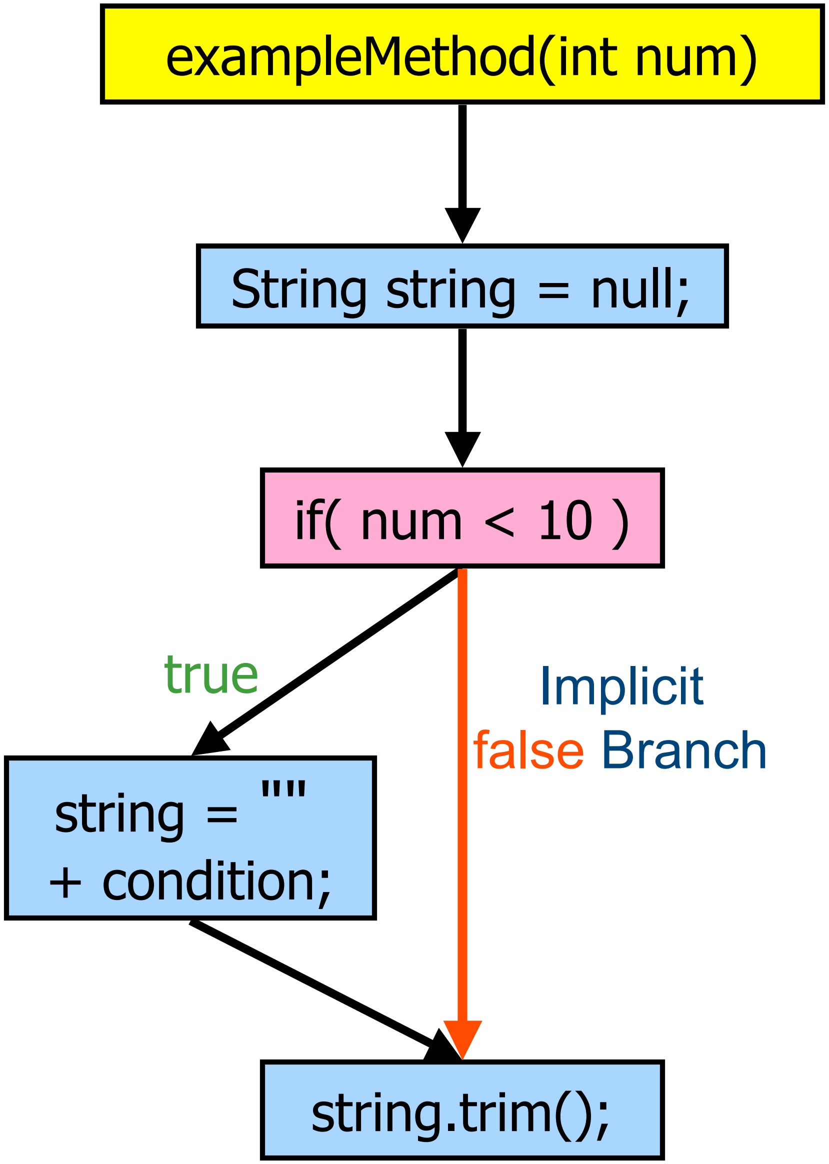
Program Flow

```
public String exampleMethod(int num) {
    String string = null;
    if (num < 10) {
        string = "" + condition;
    }
    return string.trim();
}
```



What Went Wrong?

- Test suite had 100% statement coverage but missed a **branch/edge**
- Try covering all **edges** in program’s flow
  - Also covers all **nodes**
  - Called **Branch Coverage**



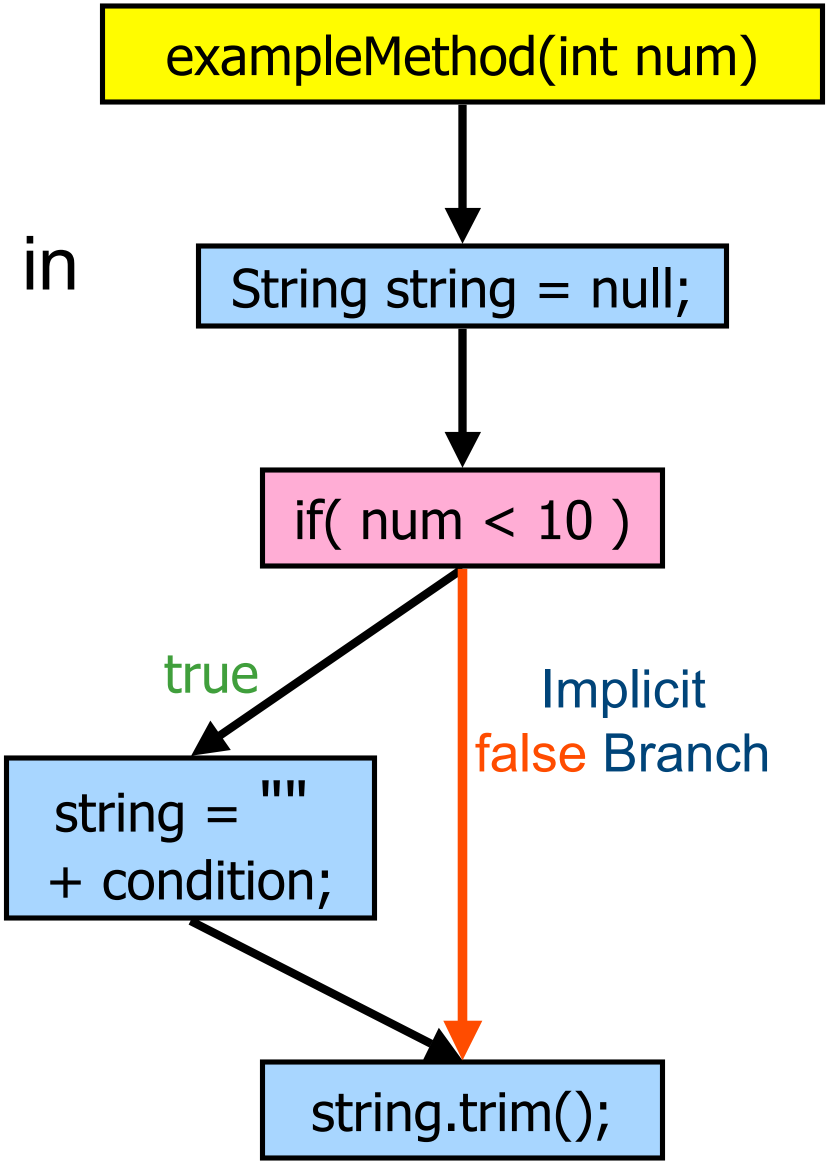
Branch Coverage

- Cover all **branches** in the program

Test Suite:

num=5,

num=10



exampleMethod(int num)

String string = null;

if(num < 10)

true

Implicit false Branch

string = "" + condition;

string.trim();

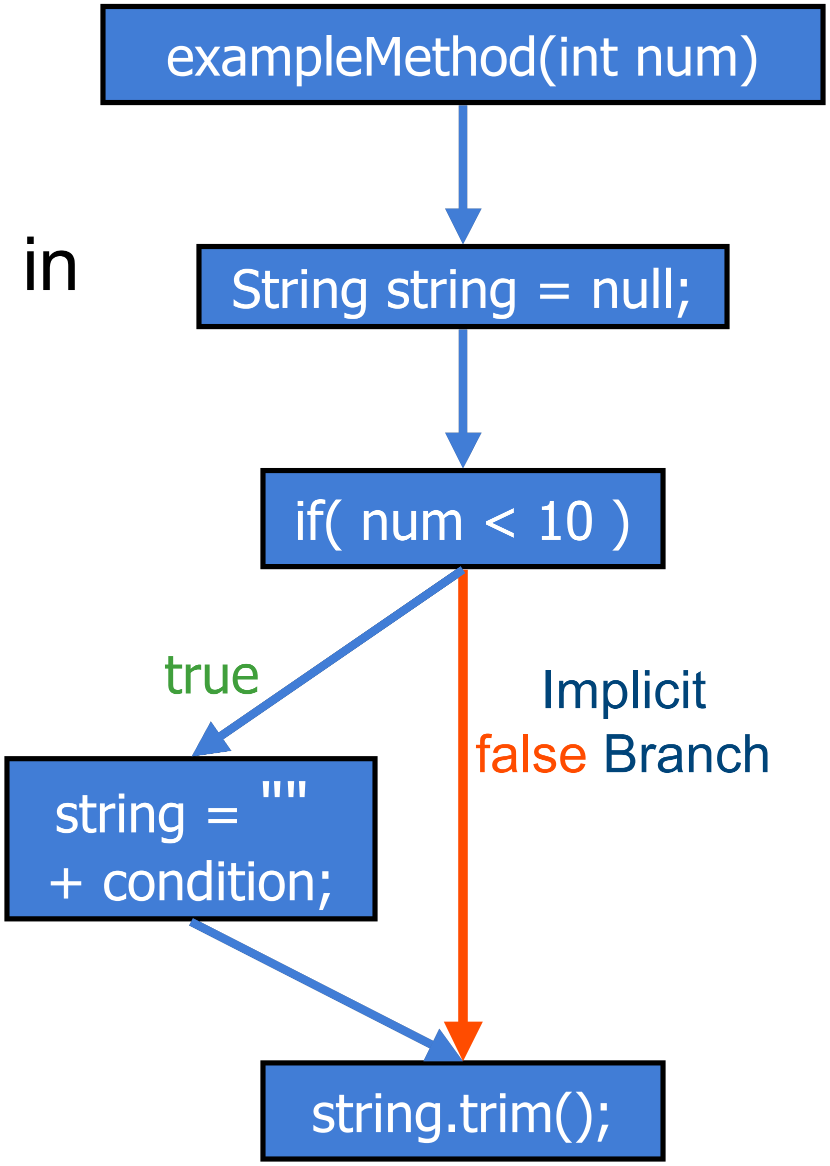
Branch Coverage

- Cover all **branches** in the program

Test Suite:

num=5,

num=10



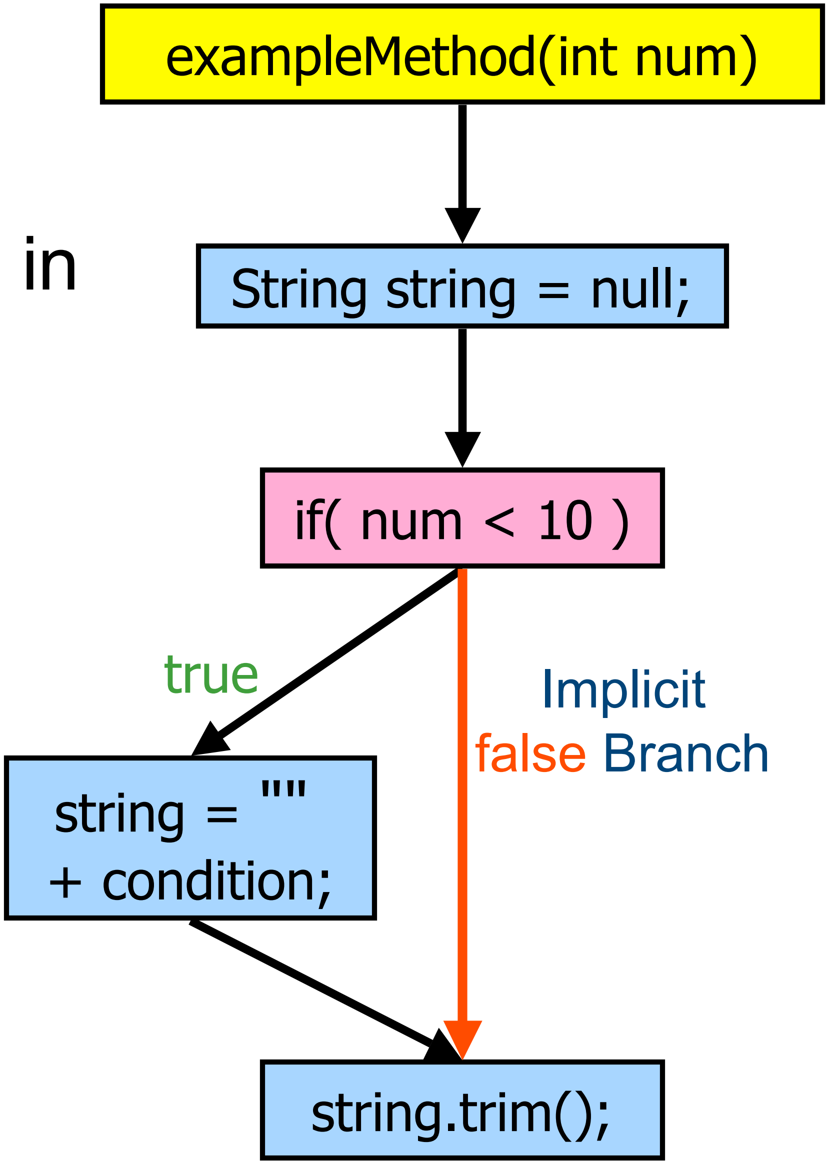
Branch Coverage

- Cover all **branches** in the program

Test Suite:

num=5,

num=10



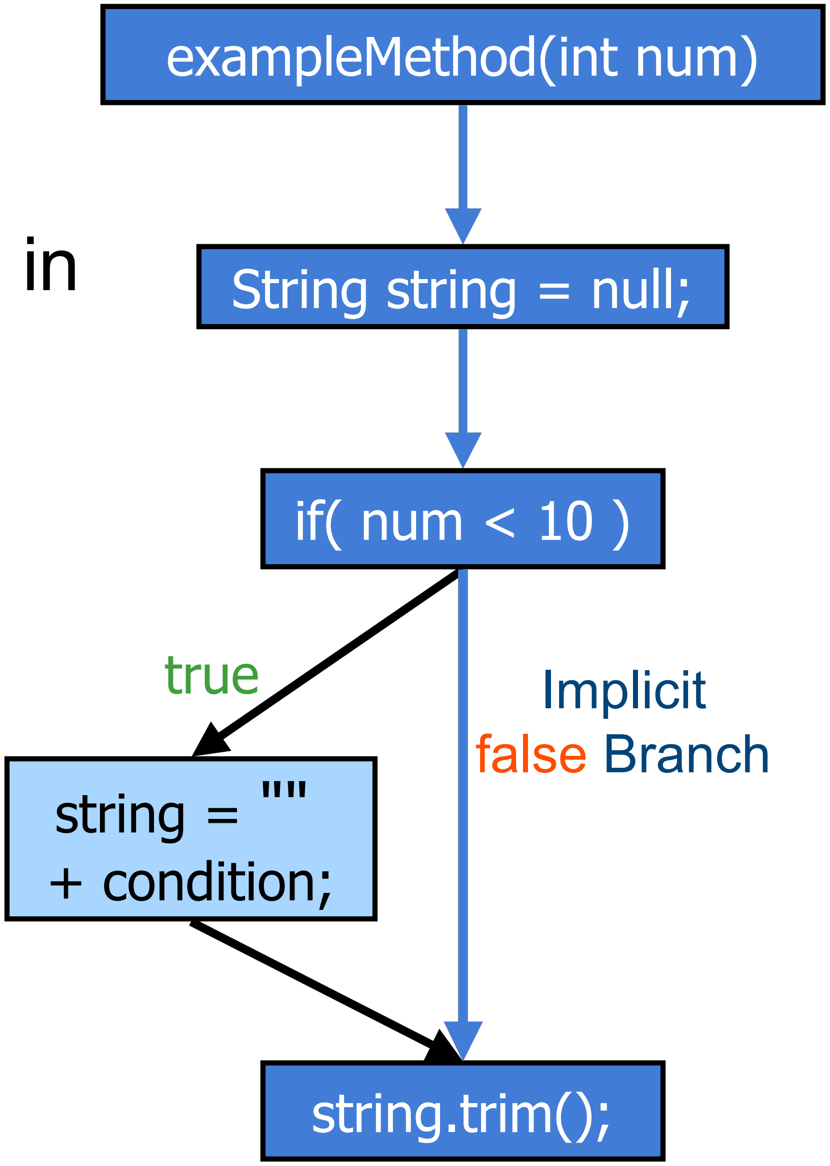
Branch Coverage

- Cover all **branches** in the program

Test Suite:

num=5,

num=10



exampleMethod(int num)

String string = null;

if(num < 10)

true

Implicit false Branch

string = "" + condition;

string.trim();

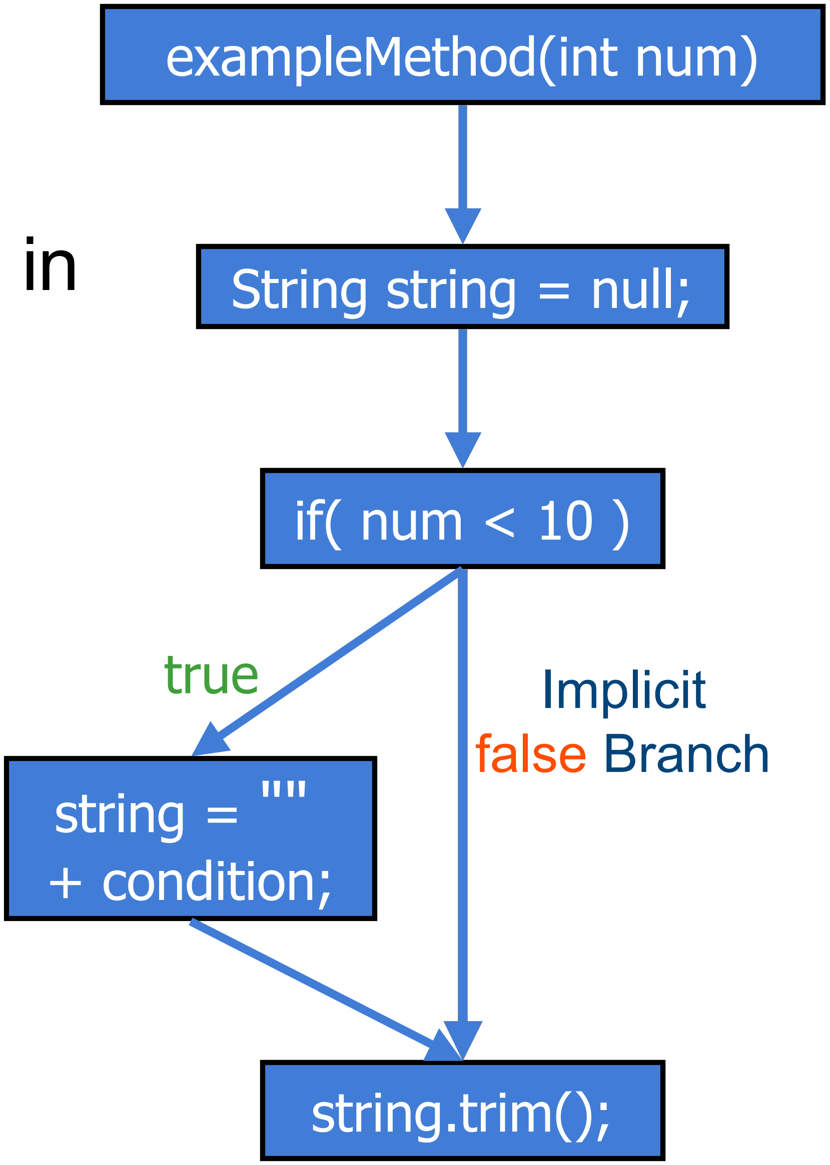
Branch Coverage

- Cover all **branches** in the program

Test Suite:

num=5,

num=10



Example 2

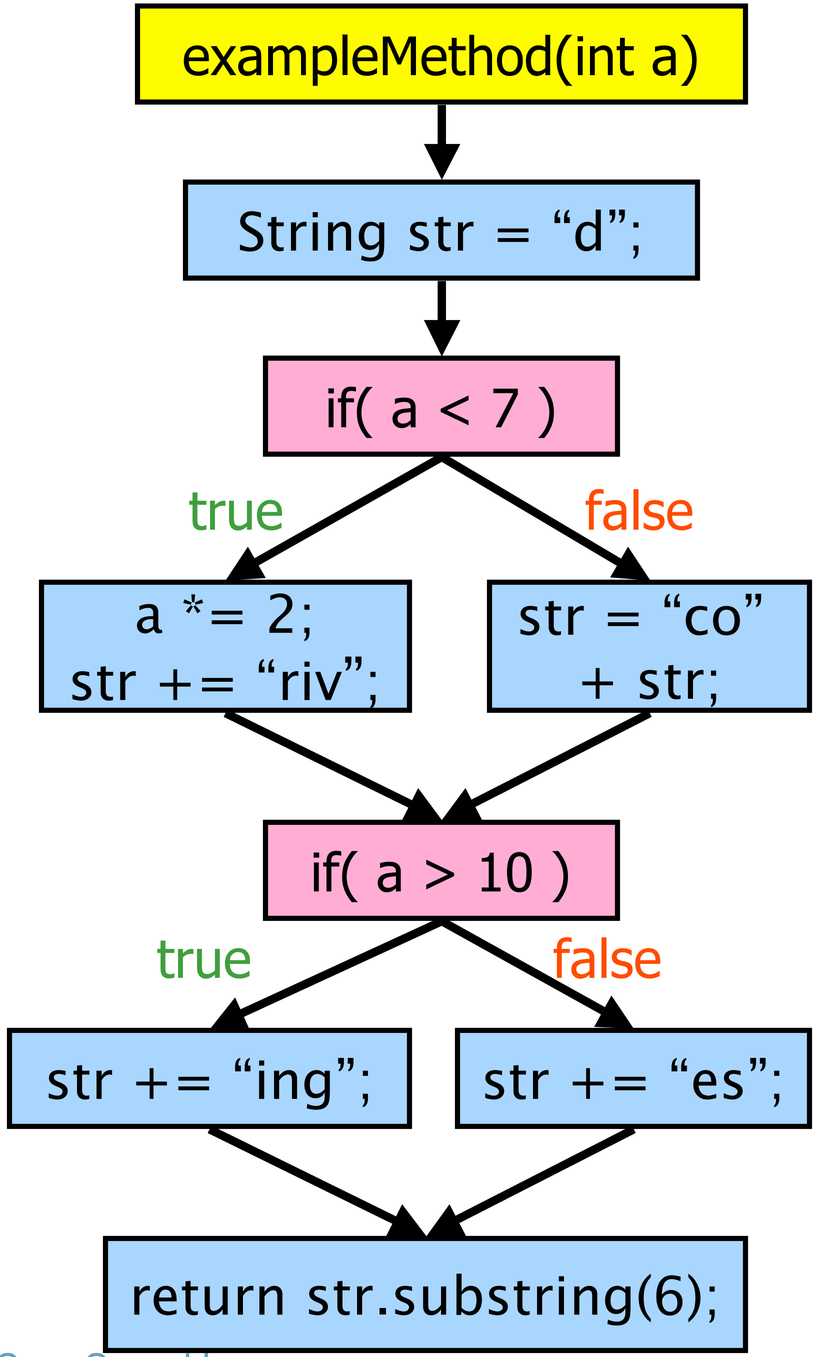
```
public int exampleMethod(int a) {
    String str = “d”;
    if ( a < 7 ) {
        a *= 2;
        str += “riv”;
    } else {
        str = “co” + str;
    }

    if( a > 10 ) {
        str += “ing”;
    } else {
        str += “es”;
    }
    return str.substring(6);
}
```

Example 2

```
public int exampleMethod(int a) {
    String str = “d”;
    if ( a < 7 ) {
        a *= 2;
        str += “riv”;
    } else {
        str = “co” + str;
    }

    if( a > 10 ) {
        str += “ing”;
    } else {
        str += “es”;
    }
    return str.substring(6);
}
```

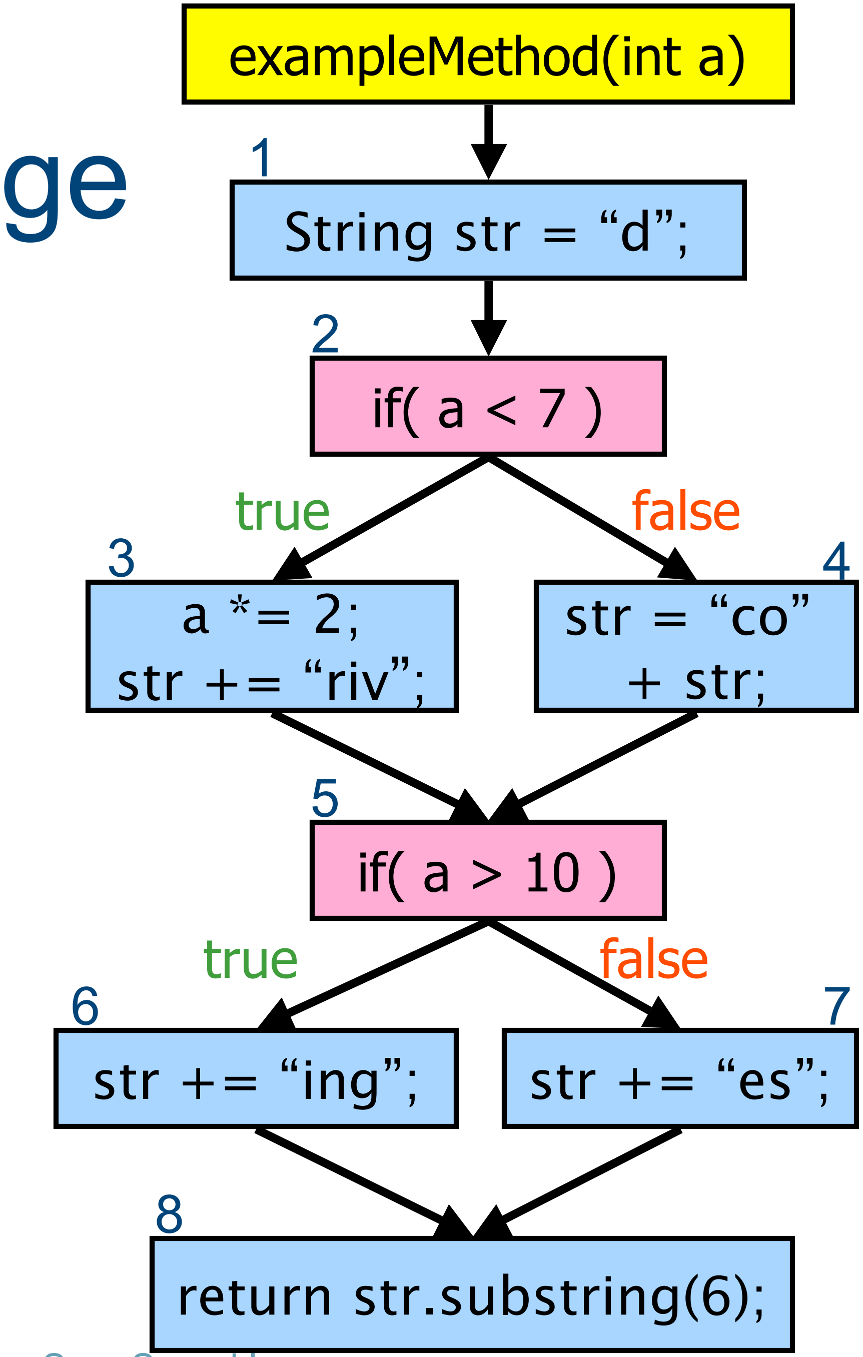


Branch Coverage

Test Suite:

a=3,

a=30



Branch Coverage

Test Suite:

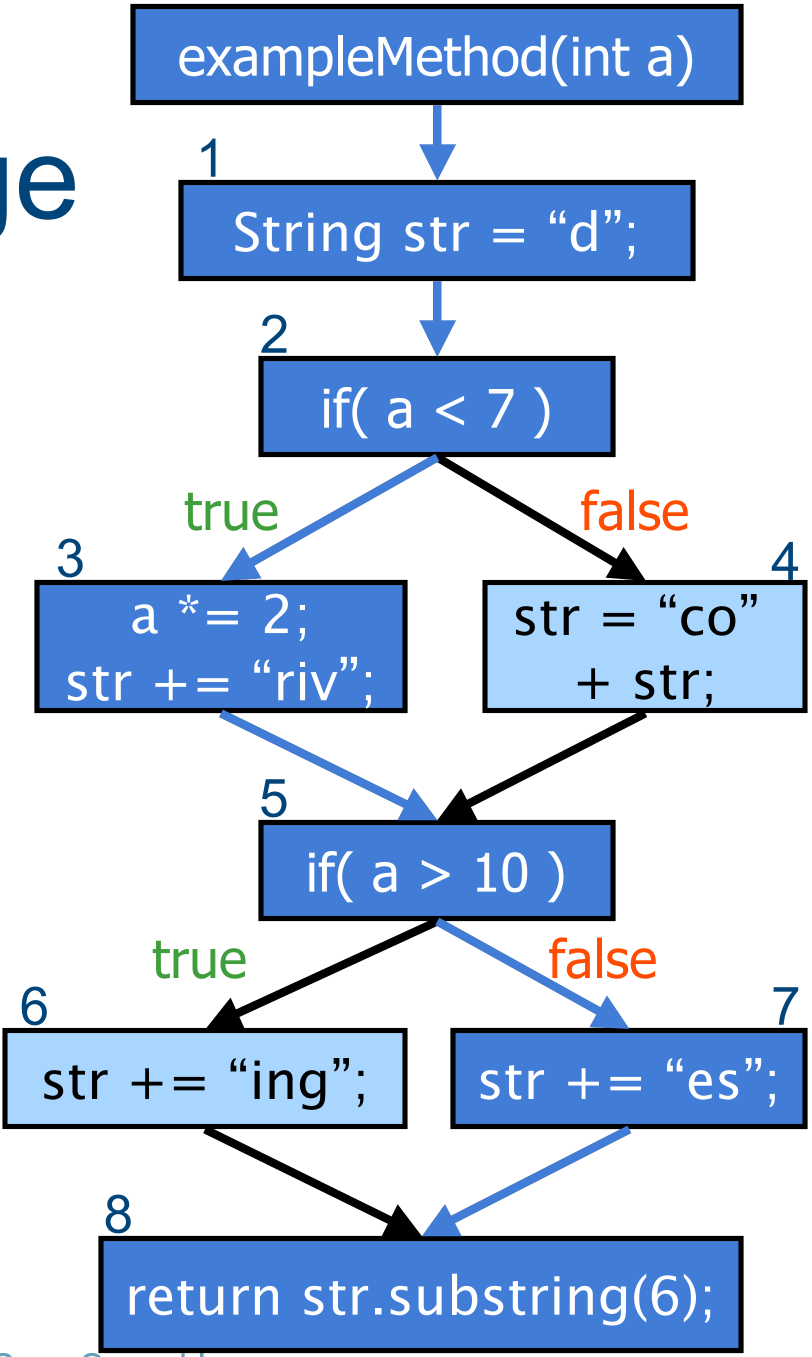
a=3,

a=30

str=“driv”
a=6

str=“drives”

“”



exampleMethod(int a)

1 String str = “d”;

2 if(a < 7)

true

false

3 a *= 2;
str += “riv”;

4 str = “co”
+ str;

5 if(a > 10)

true

false

6 str += “ing”;

7 str += “es”;

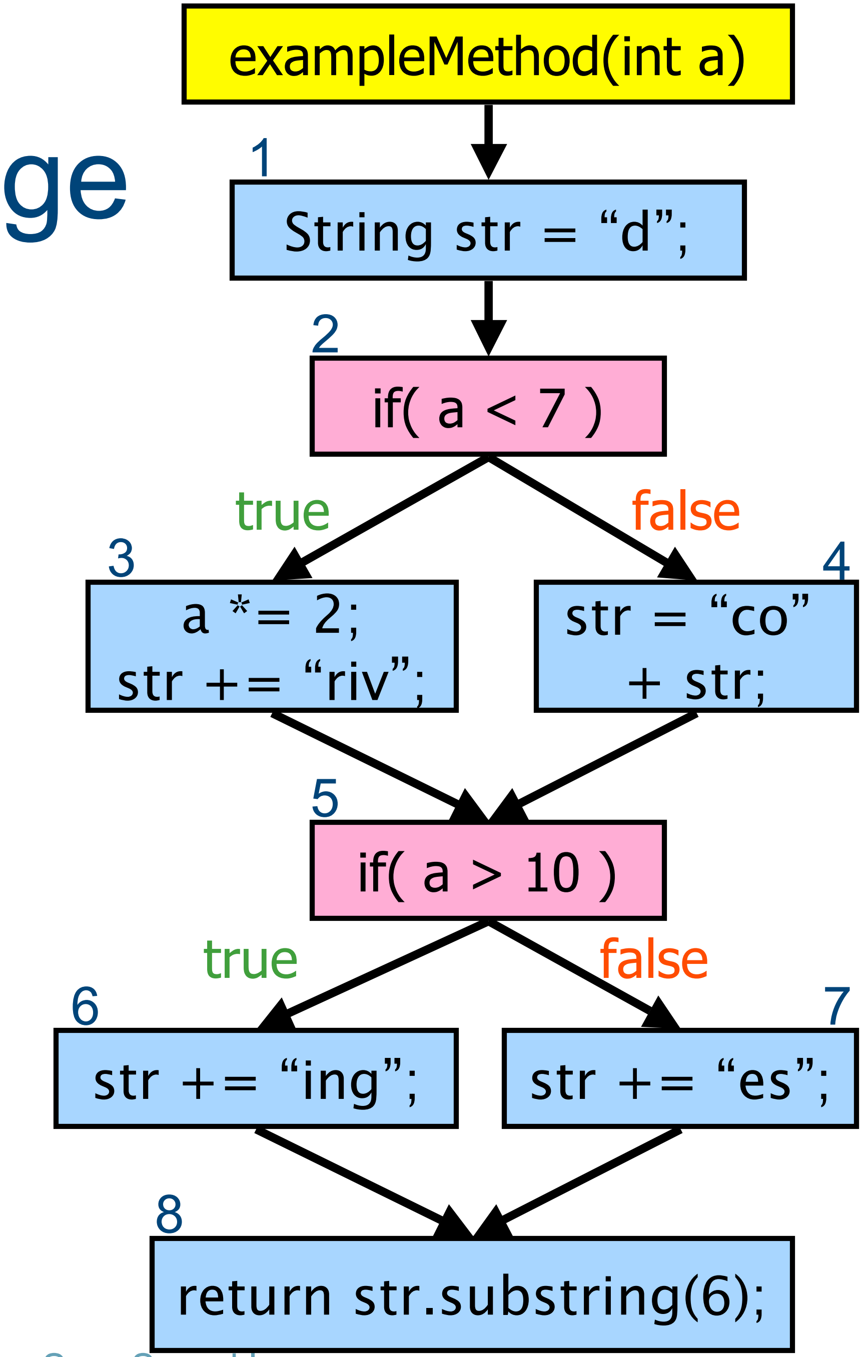
8 return str.substring(6);

Branch Coverage

Test Suite:

a=3,

a=30

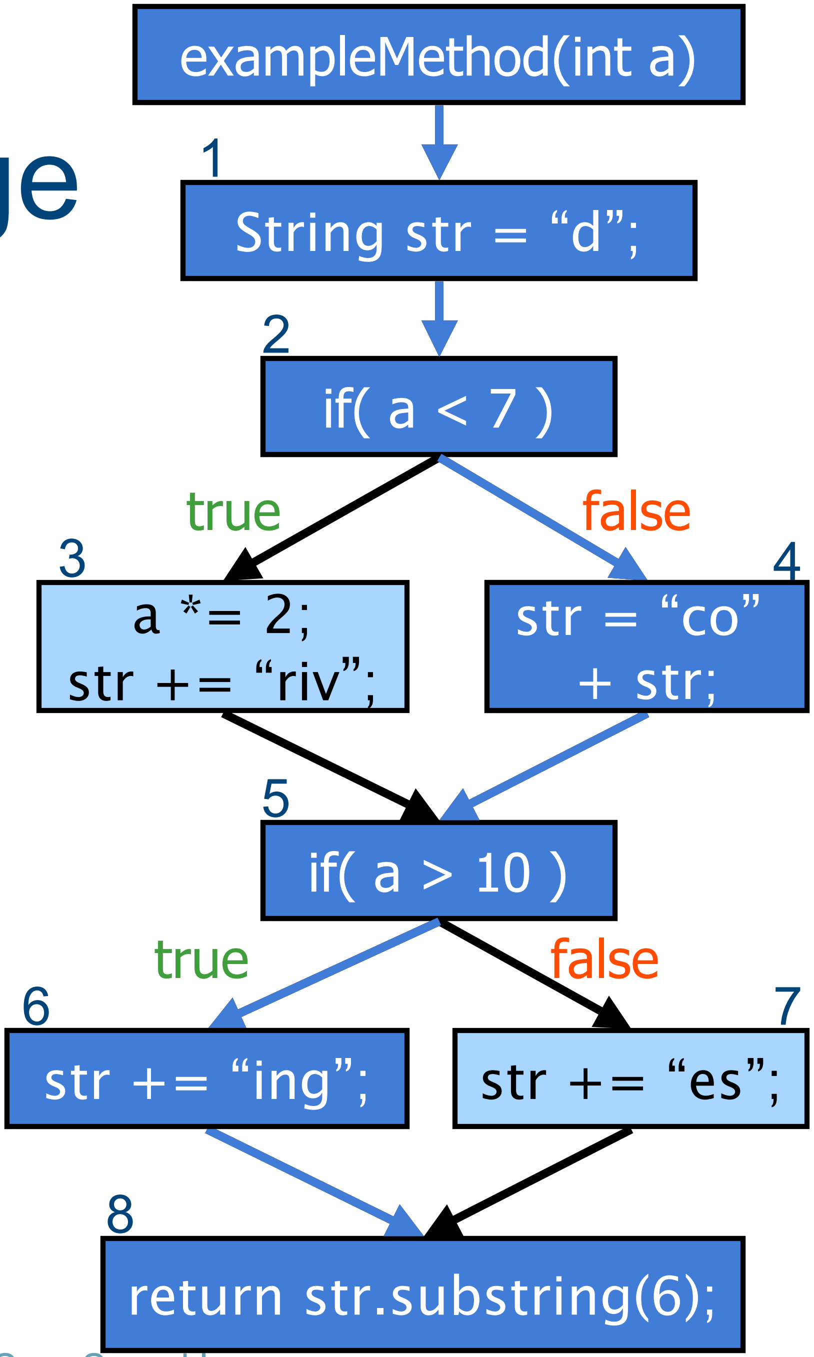


Branch Coverage

Test Suite:

a=3,

a=30



exampleMethod(int a)

1 String str = “d”;

2 if(a < 7)

true

false

3 a *= 2;
str += “riv”;

4 str = “co” + str;

5 if(a > 10)

true

false

6 str += “ing”;

7 str += “es”;

8 return str.substring(6);

str=“cod”
a=30

str=“coding”

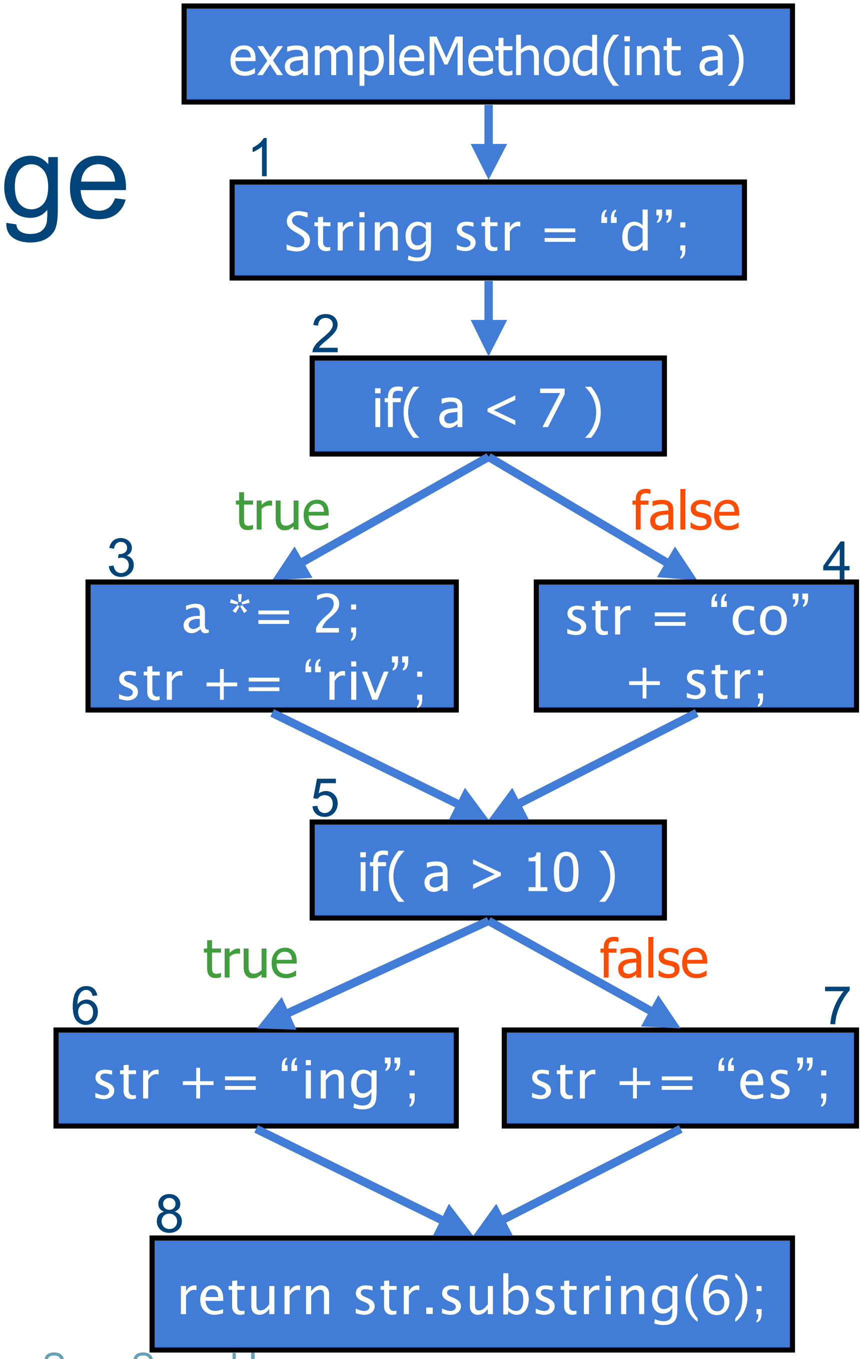
“”

Branch Coverage

Test Suite:

a=3,

a=30



exampleMethod(int a)

1 String str = “d”;

2 if(a < 7)

true

false

3 a *= 2;
str += “riv”;

4 str = “co”
+ str;

5 if(a > 10)

true

false

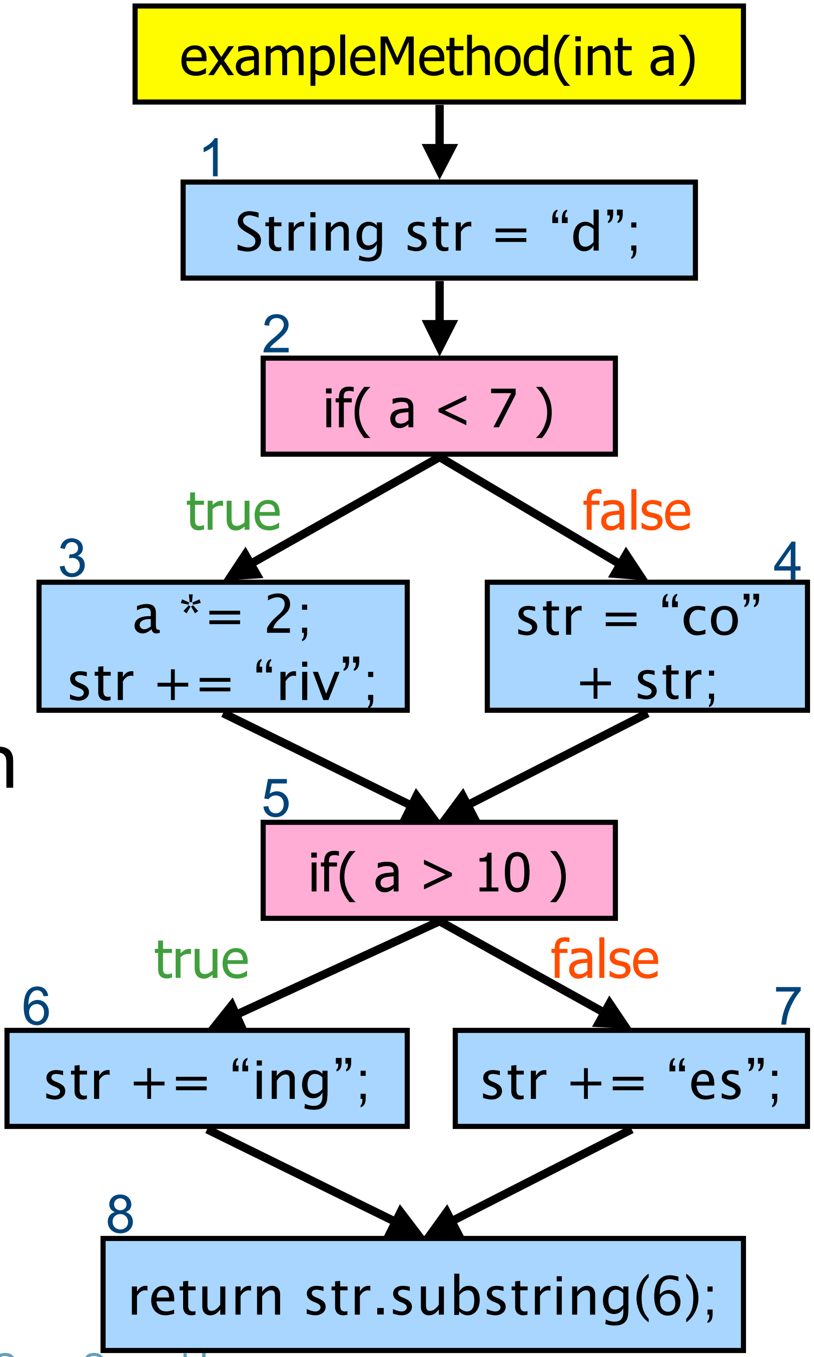
6 str += “ing”;

7 str += “es”;

8 return str.substring(6);

What Went Wrong?

- Test suite had 100% branch (and statement) coverage but missed a **path**
- Try to cover all **paths** in program's flow
  - Also gets all **branches**, **nodes**
  - Called **Path Coverage**
  - **Not** generally **practical**



Code Coverage Tools

- Coverage is used in practice
- You don’t need to figure out coverage manually
- Some available tools to calculate coverage
  - For Java: Clover, JCoverage, Emma, …
  - For C/C++: BullseyeCoverage, CoverageMeter, …
  - For C#: NClover, …
- And many more …
  - Web Resource: http://www.testingfaqs.org/t-eval.html

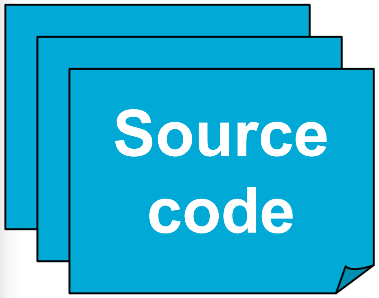
Cenqua's Clover

- Tool to measure code coverage
- Web site: http://www.cenqua.com/clover/
- Code coverage used to
  - Measure **quality** of test suite
  - Improve test suite
  - Determine when to stop testing

A Little History of Clover

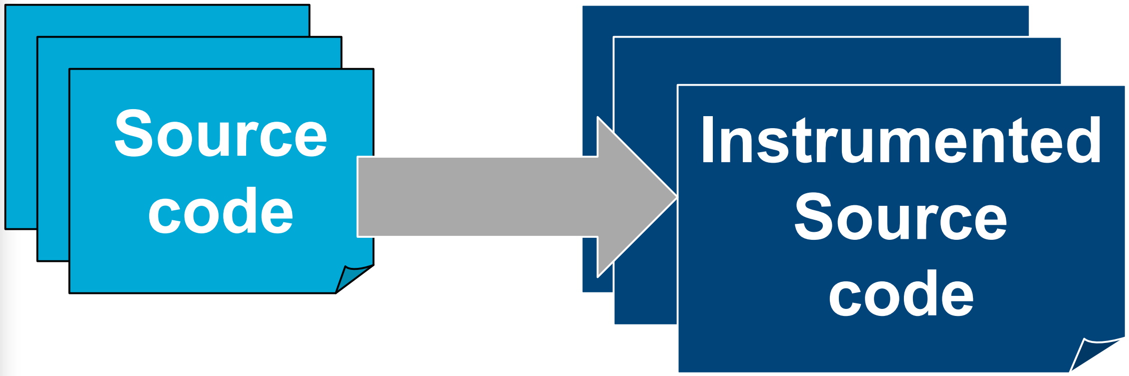
- Initially developed as an internal tool
  - Couldn't find a reasonably priced coverage tool that also
    - Performed well in a continuous integration environment
    - Performed well with large J2EE applications
- Released as a side project in May 2002
- Within 12 months had eclipsed Cenqua's traditional services business
- Name comes from shortened version of "Cover Lover"

How Clover Works



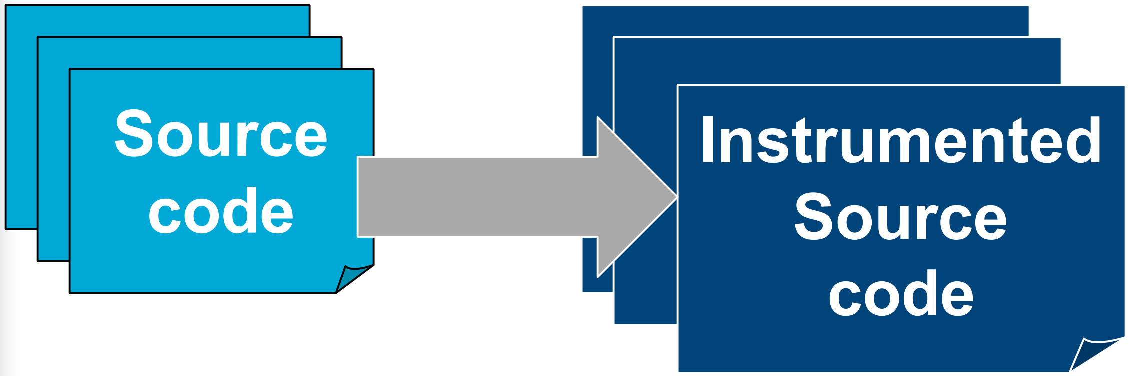
- Instruments source code w/ code to report coverage

How Clover Works



- Instruments source code w/ code to report coverage

How Clover Works

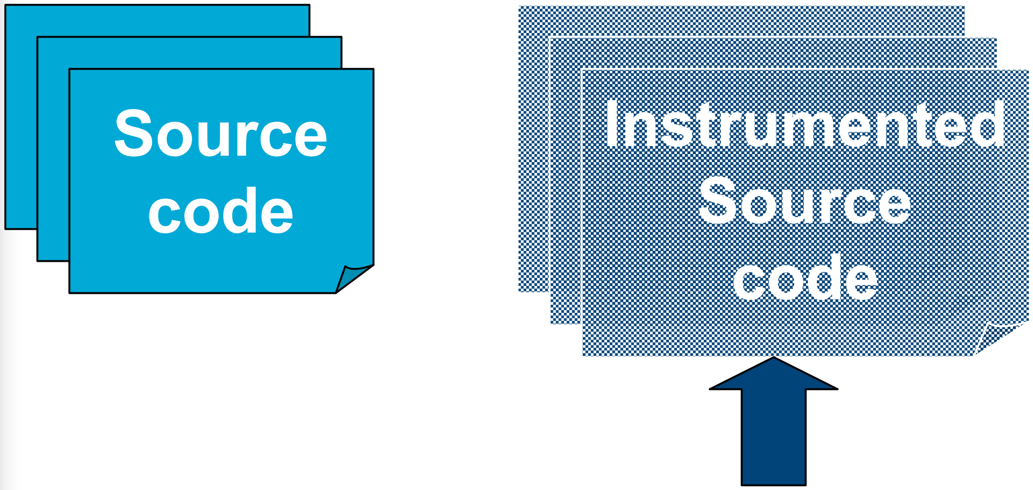


Approximation of Instrumentation

```
…
public void method() {
            _cl_method1_cnt++;
            x=y+z;
            _cl_stmt1_cnt++;
            …
}
…
```

- Instruments source code w/ code to report coverage

How Clover Works

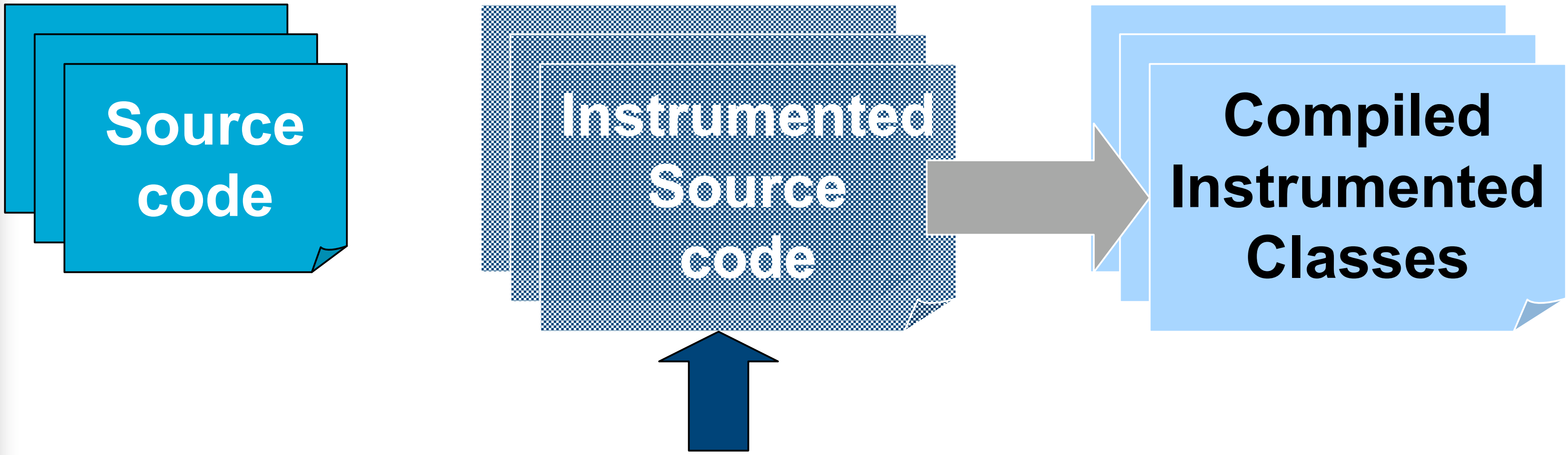

Source code

Instrumented Source code

Doesn’t actually output these

- Compiles instrumented code

How Clover Works



Doesn't actually output these

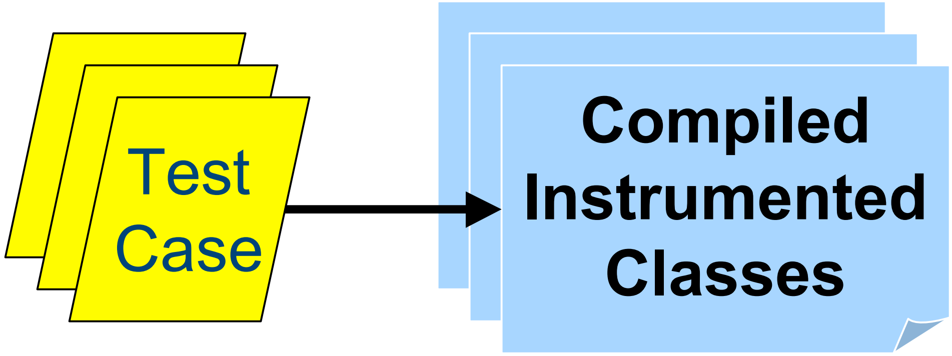
- Compiles instrumented code

How Clover Works

Compiled Instrumented Classes

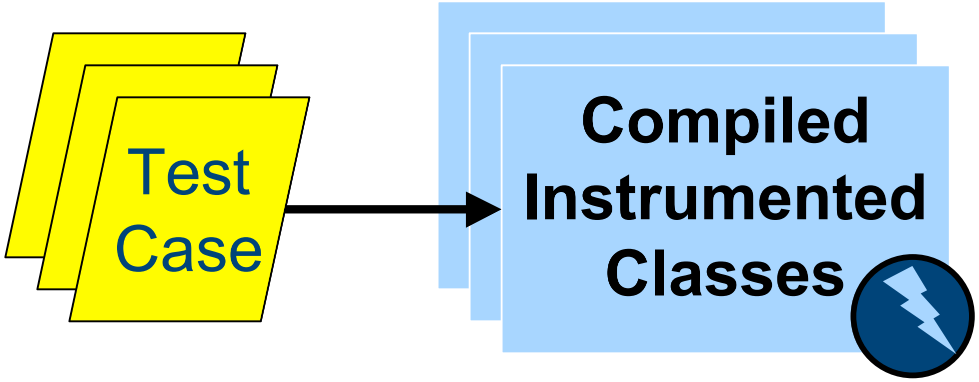
- Records coverage as tests are executed

How Clover Works



- Records coverage as tests are executed

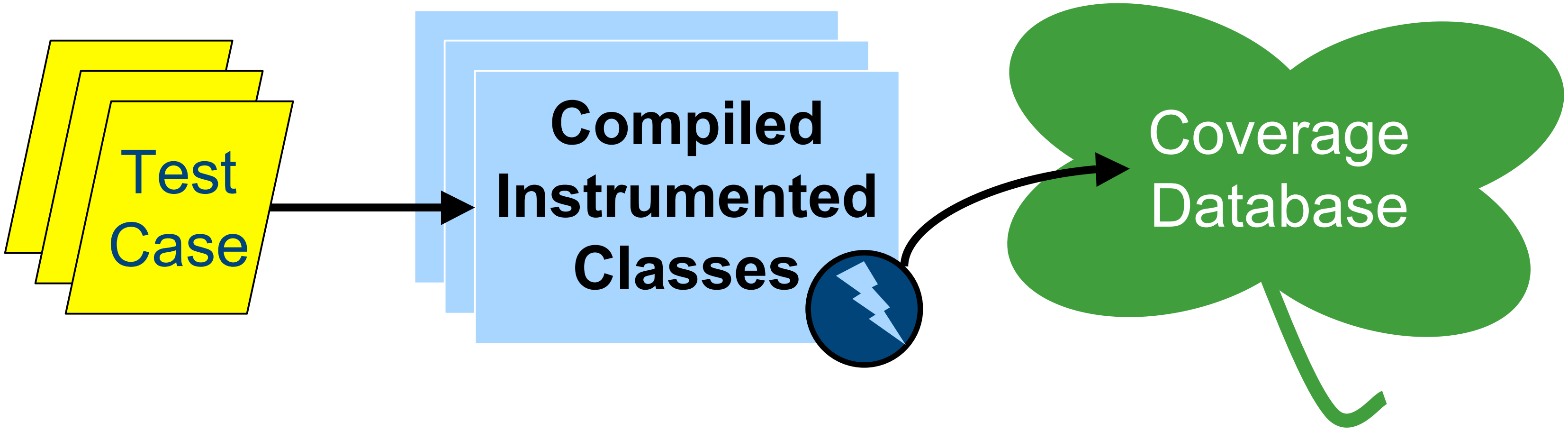
How Clover Works


Test Case

Compiled Instrumented Classes

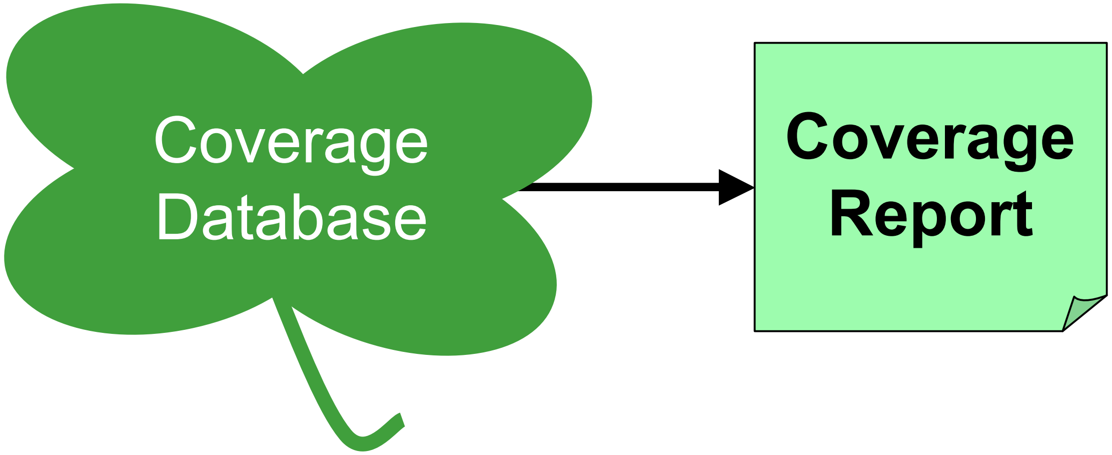
- Records coverage as tests are executed

How Clover Works



- Records coverage as tests are executed

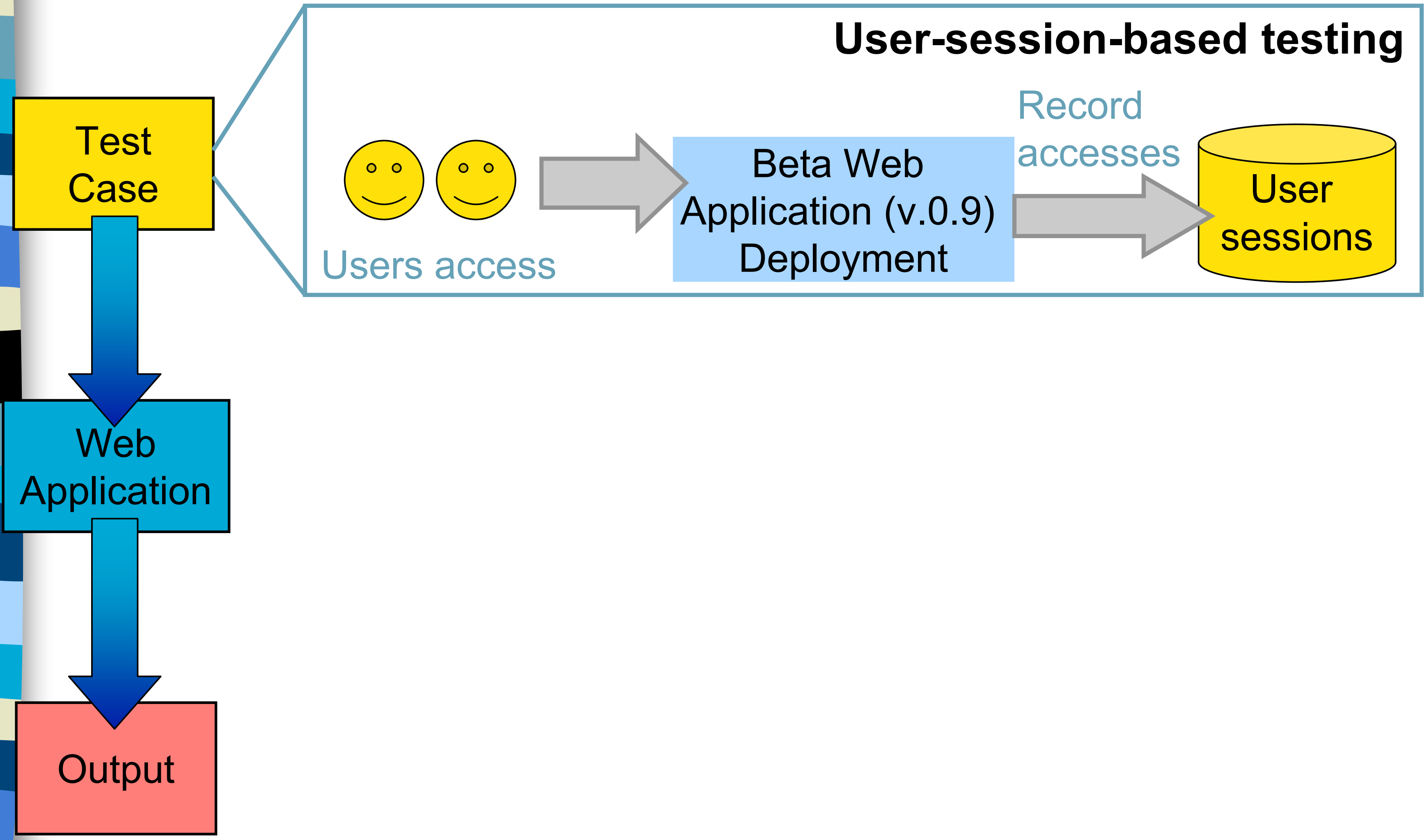
How Clover Works



- Generate coverage report
  - Various formats: XML, HTML, PDF, Plain Text, Swing

How I Use Clover

My Approach to Web Application Testing



User-session-based testing

Users access

Beta Web Application (v.0.9) Deployment

Record accesses

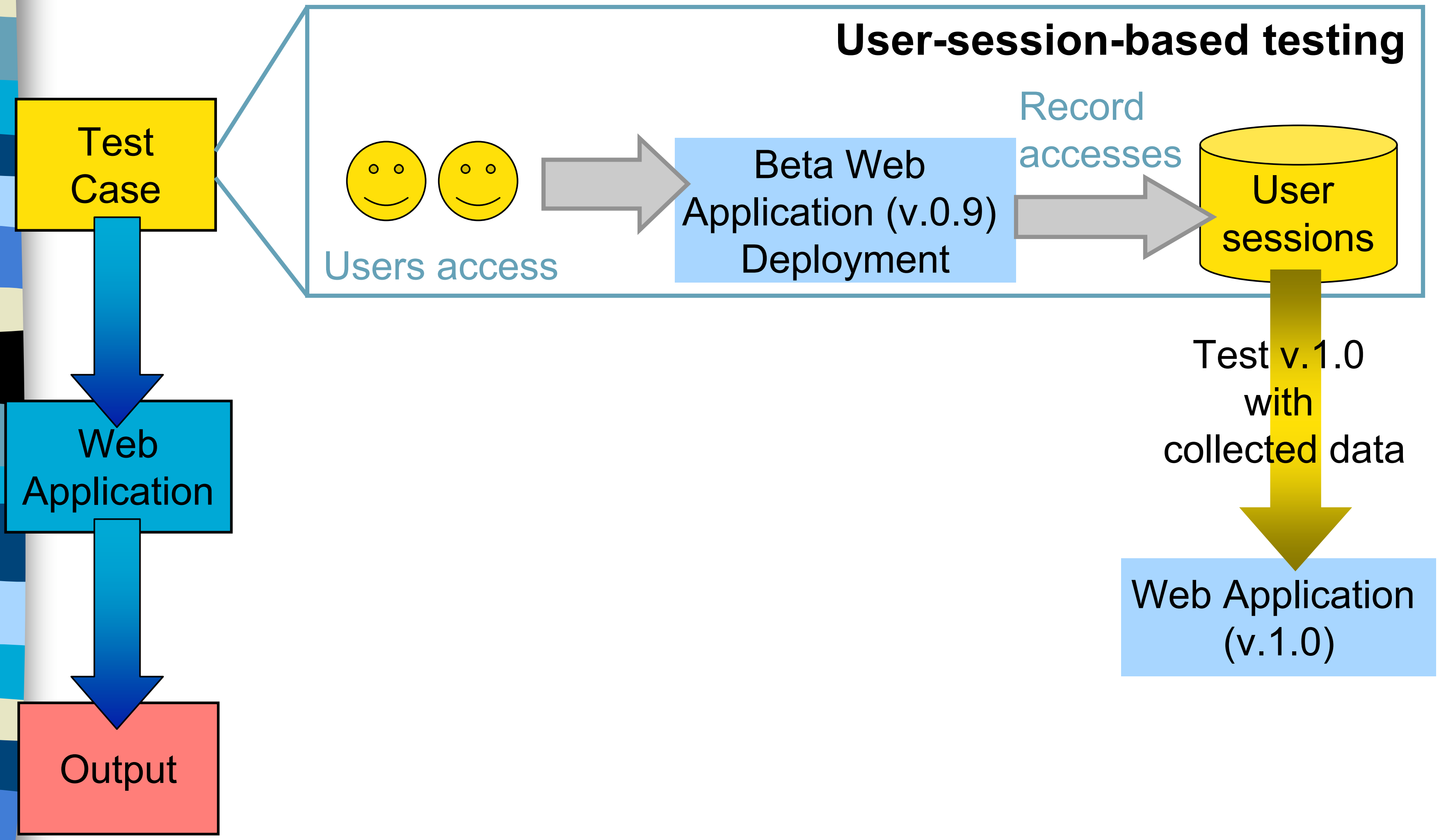
User sessions

Test Case

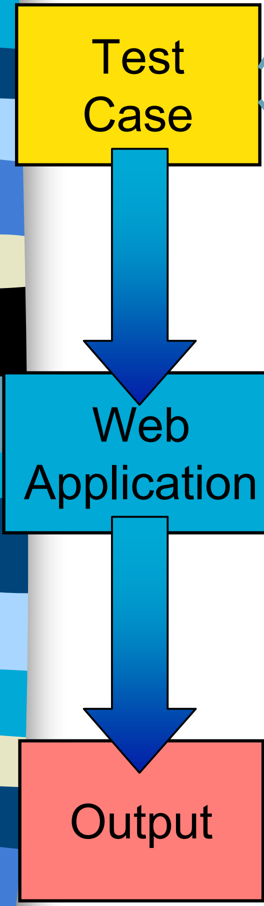
Web Application

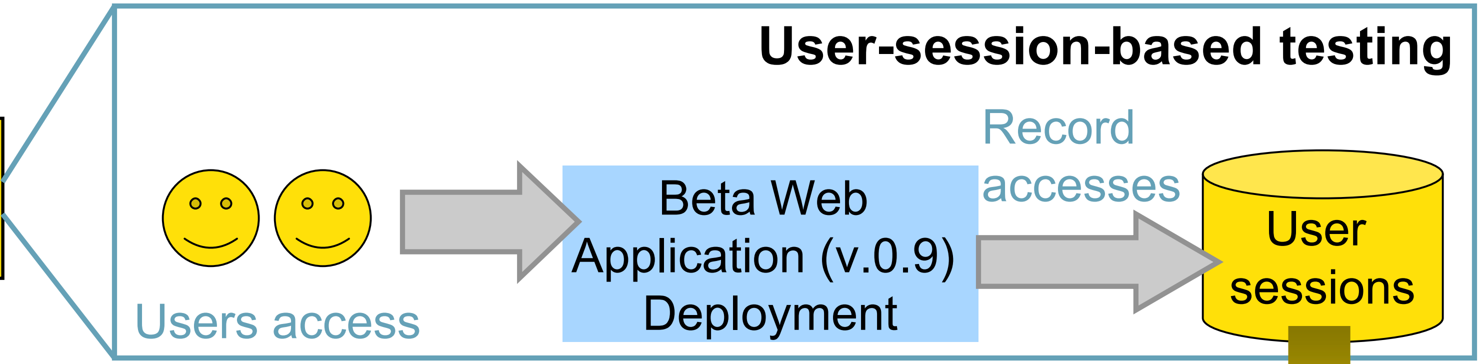
Output

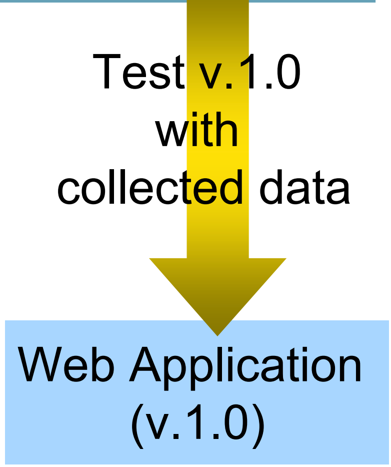
My Approach to Web Application Testing


Test Case
Web Application
Output
User-session-based testing
Users access
Beta Web Application (v.0.9) Deployment
Record accesses
User sessions
Test v.1.0 with collected data
Web Application (v.1.0)

My Approach to Web Application Testing

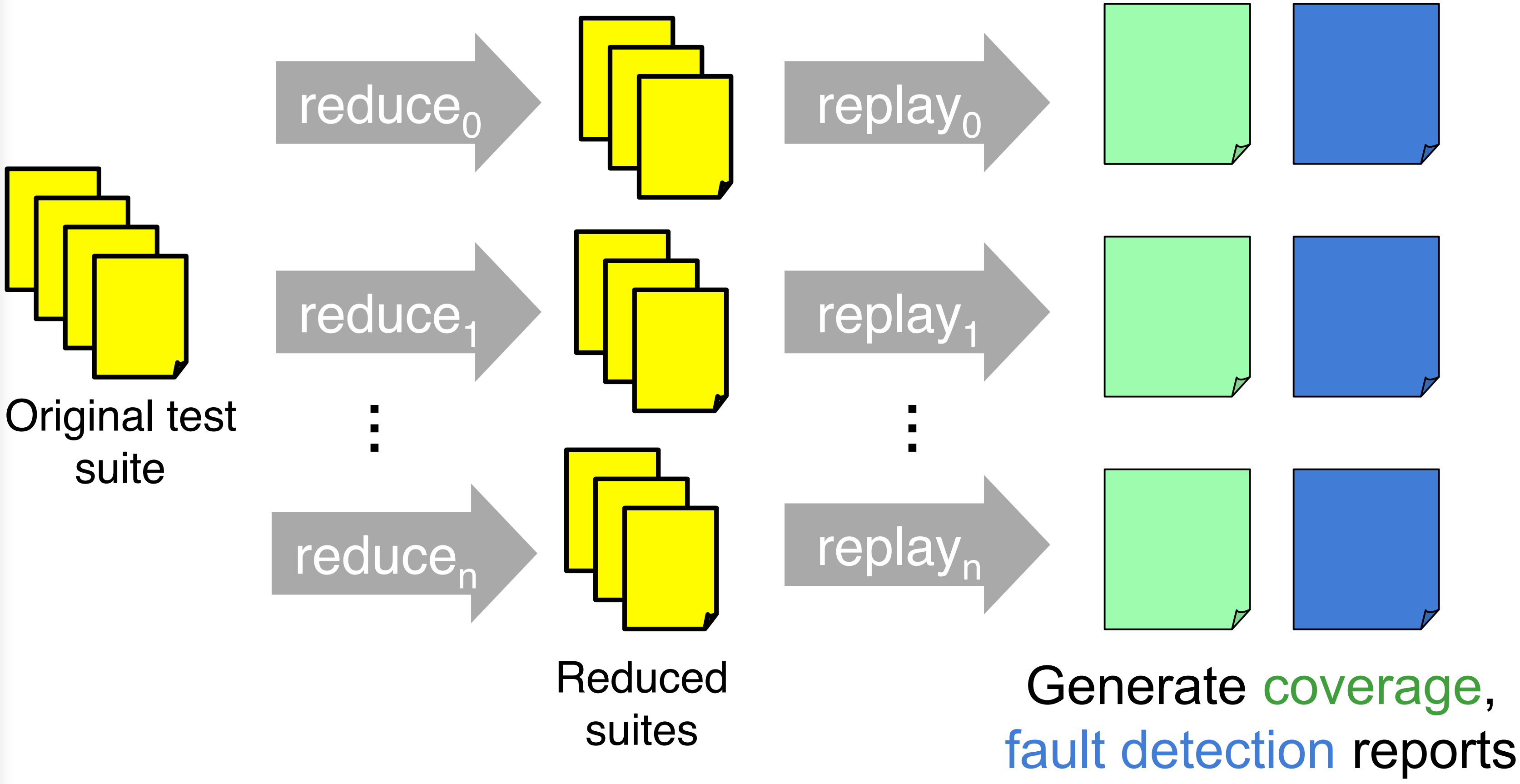



User-session-based testing
Users access
Beta Web Application (v.0.9) Deployment
Record accesses
User sessions



- Test cases: real user accesses
  - Test code users frequently access
  - Likely redundant

Evaluating Reduced Test Suites



Original test suite

reduce$_0$

reduce$_1$

⋮

reduce$_n$

Reduced suites

replay$_0$

replay$_1$

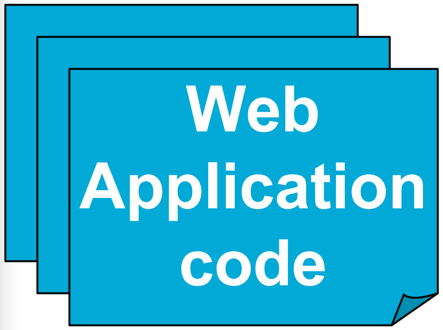
⋮

replay$_n$

Generate coverage, fault detection reports

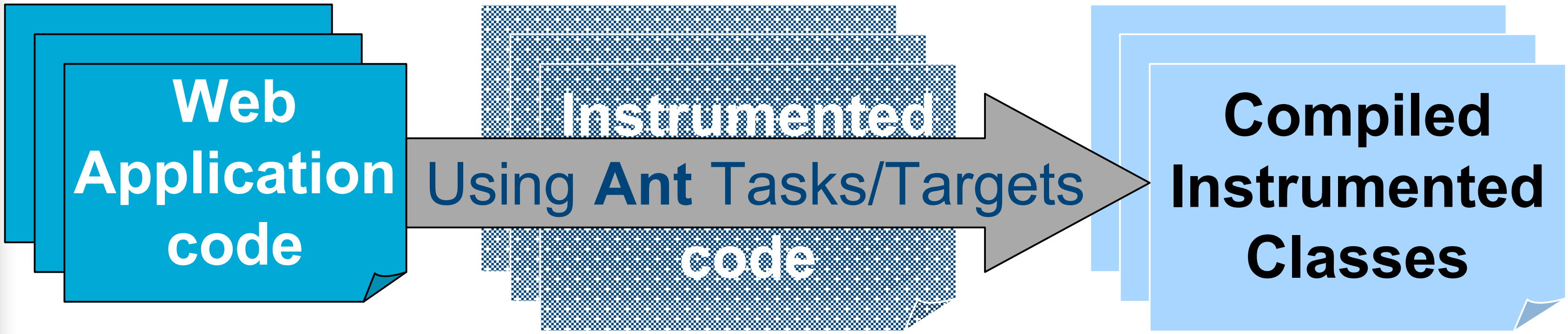
- Reduce original test suite
- Compare coverage of reduced test suites

How I Use Clover

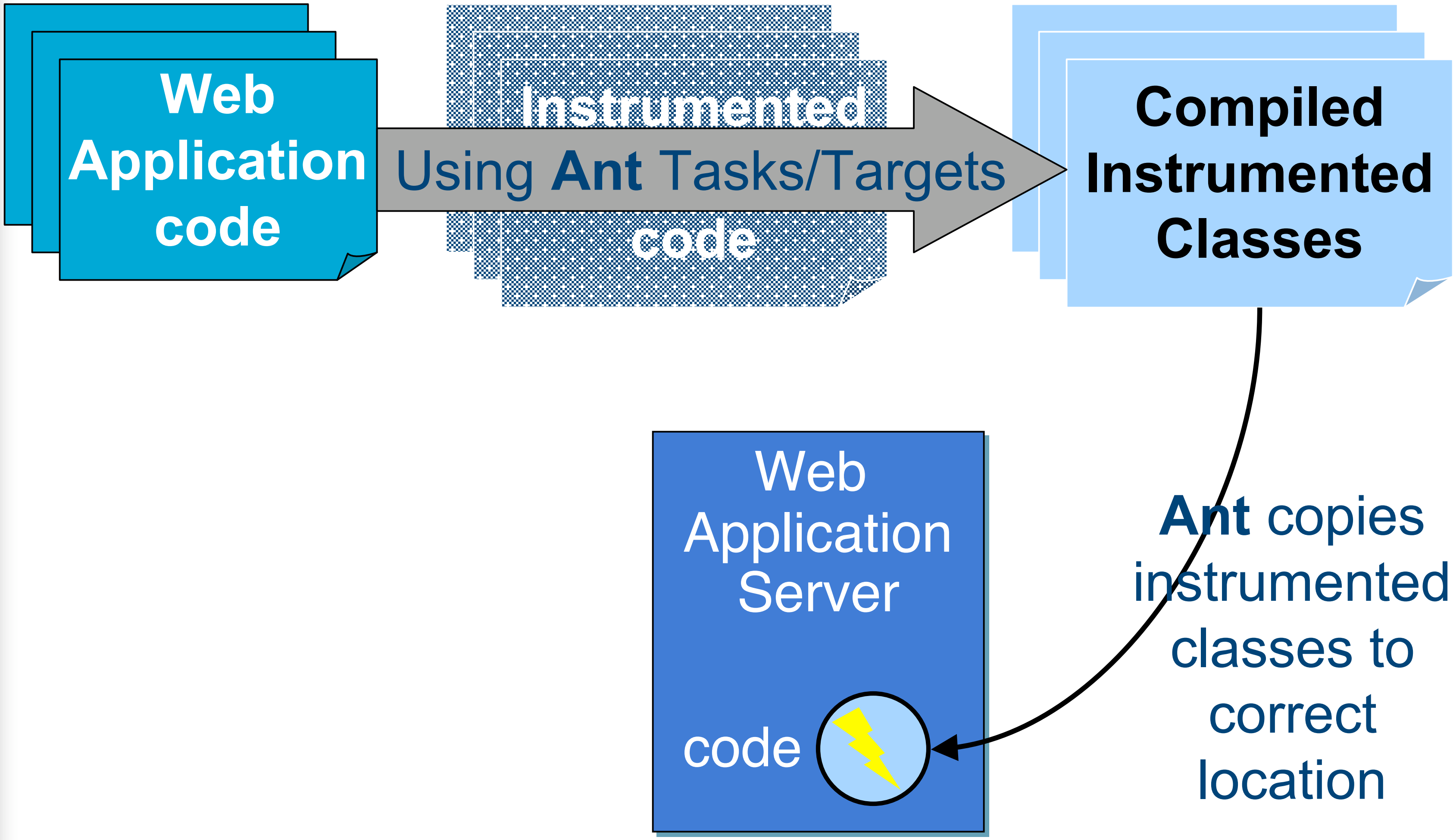


Web Application code

How I Use Clover



How I Use Clover



Web Application code

Instrumented code

Using **Ant** Tasks/Targets

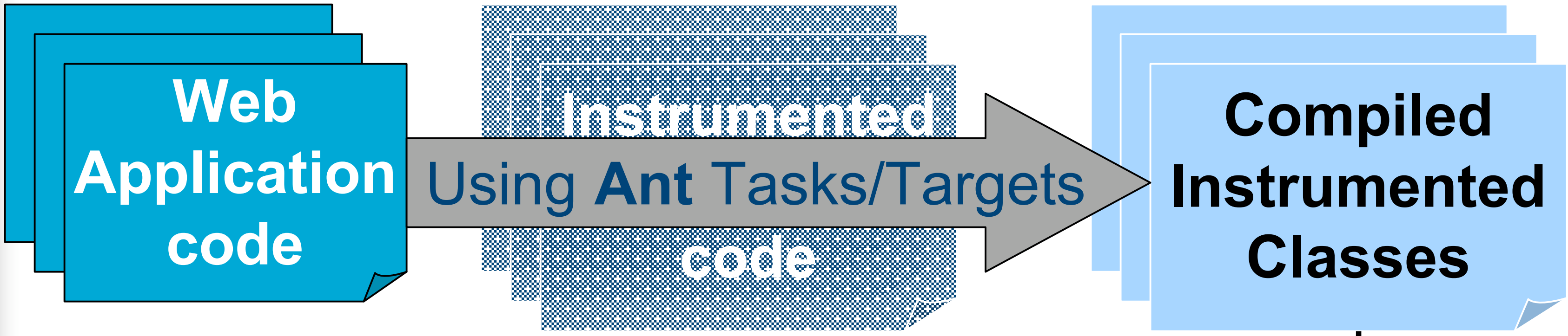
Compiled Instrumented Classes

Web Application Server

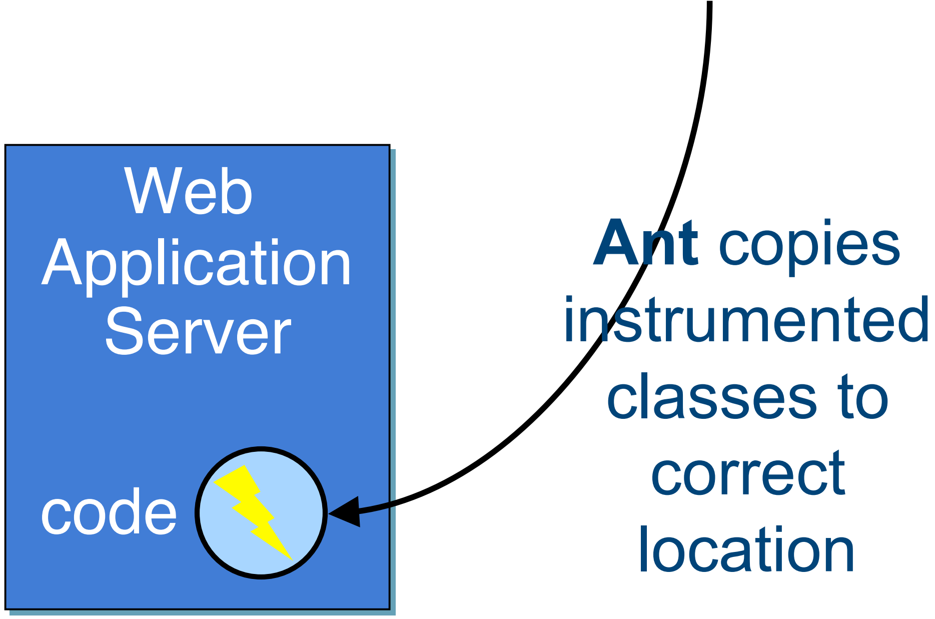
code

Ant copies instrumented classes to correct location

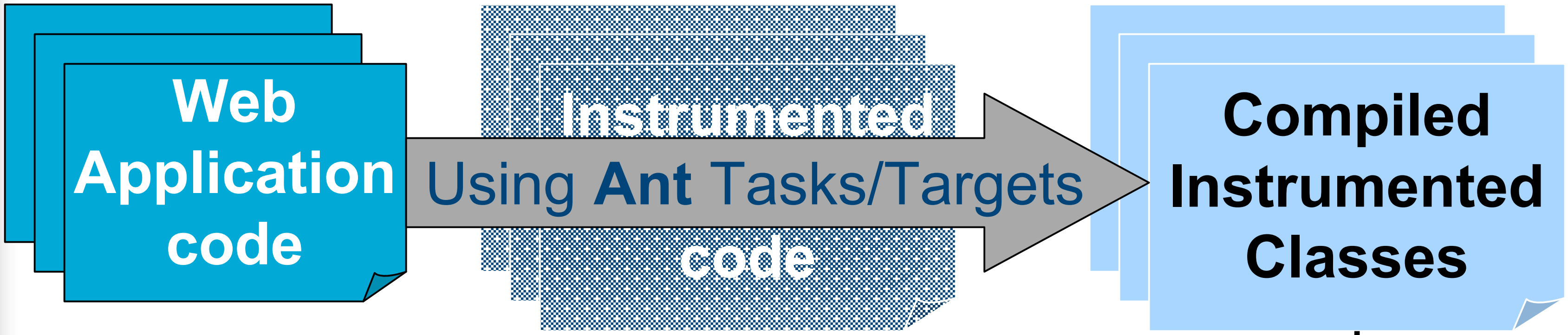
How I Use Clover



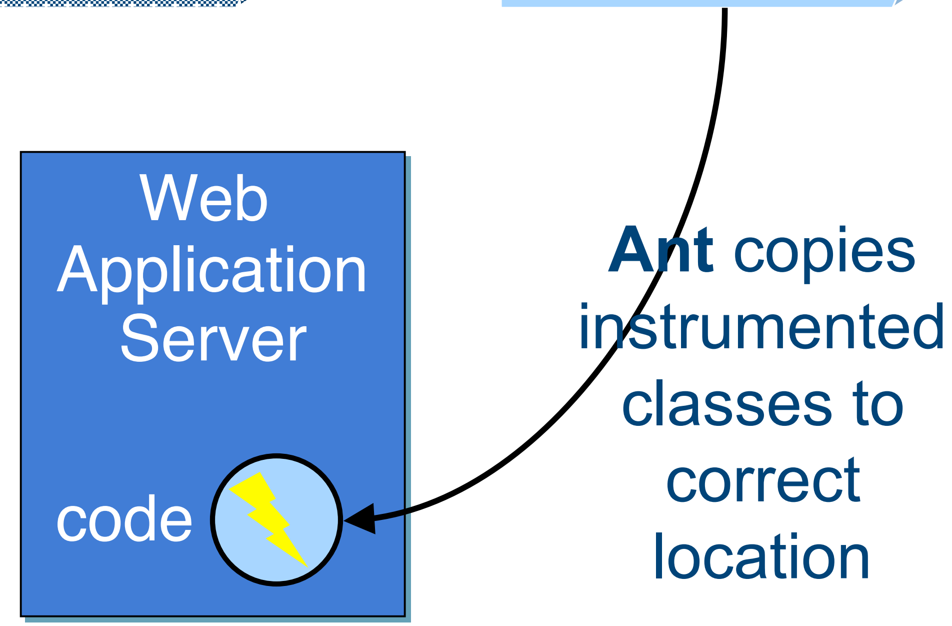
- Execute test suites



How I Use Clover



- Execute test suites
- Generate coverage reports

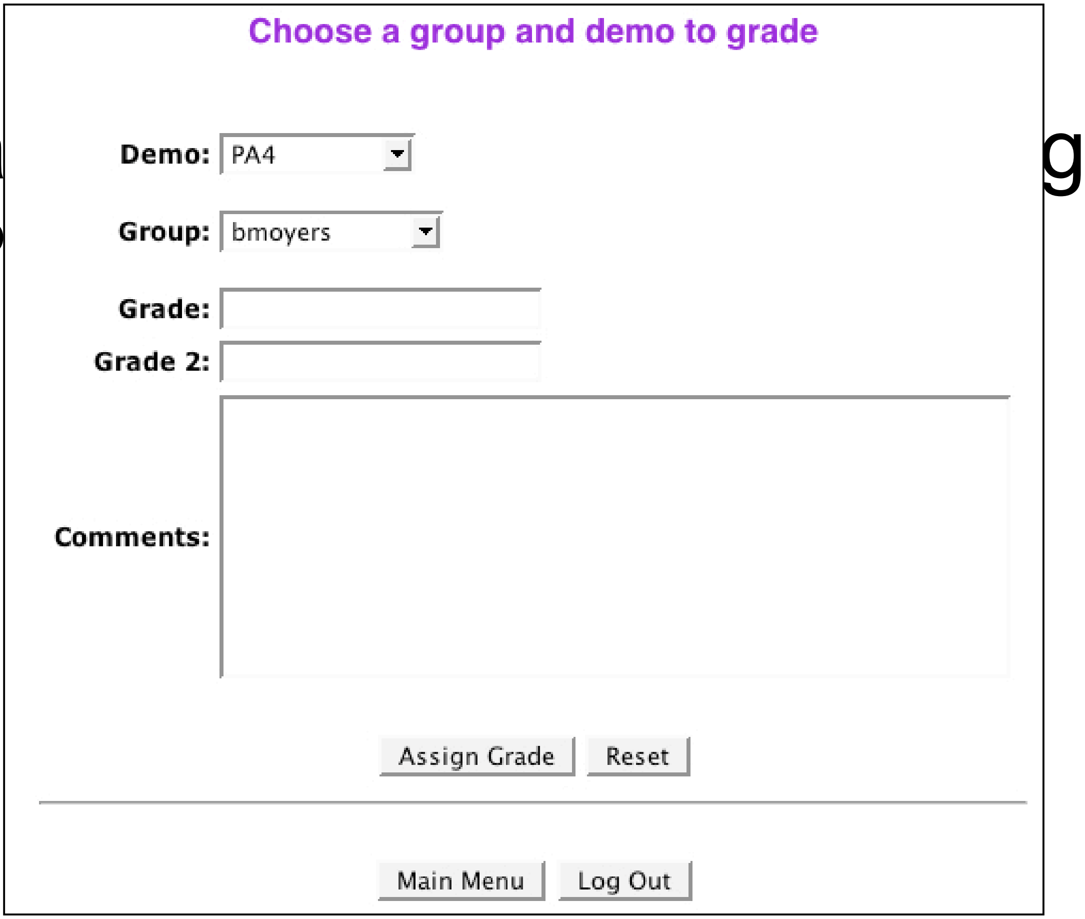


How I Use Clover

- Example: Course Project Manager (CPM)

How I Use Clover

- Exa ger
(CP



Choose a group and demo to grade

Demo: PA4

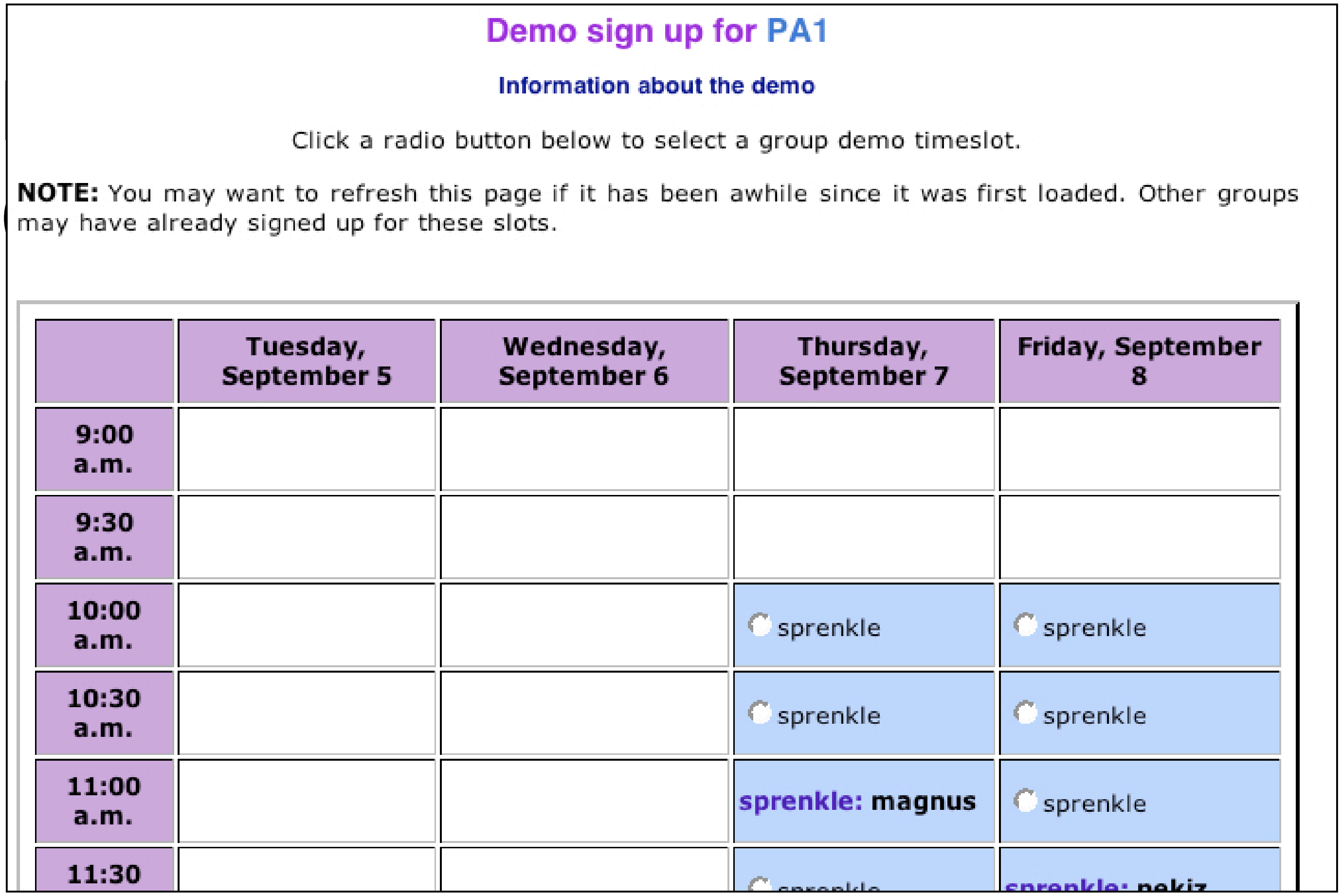
Group: bmoyers

Grade:

Grade 2:

Comments:

How I Use Clover



Demo sign up for PA1

Information about the demo

Click a radio button below to select a group demo timeslot.

NOTE: You may want to refresh this page if it has been awhile since it was first loaded. Other groups may have already signed up for these slots.

	Tuesday, September 5	Wednesday, September 6	Thursday, September 7	Friday, September 8
9:00 a.m.				
9:30 a.m.				
10:00 a.m.			○ sprenkle	○ sprenkle
10:30 a.m.			○ sprenkle	○ sprenkle
11:00 a.m.			sprenkle: magnus	○ sprenkle
11:30			○ sprenkle	sprenkle: nekiz

Clover Coverage Reports

- HTML report
- XML report

Clover Features

- Fast, accurate coverage measurement
- Directives to choose which code to instrument
  - E.g., can exclude certain methods
- Integrated with Apache Ant, Maven
- Multiple report formats
  - Historical reporting too

More Clover Features

- Plugins for IDEs
  - Integrate testing/coverage into development
  - Eclipse, IntelliJ IDEA, Eclipse, NetBeans, JBuilder
- Can measure coverage for **distributed** applications
- Can use interactively
  - Look at coverage during testing process

Future of Clover: Clover 2.0

- Includes more than just coverage reports
- Results of test cases (pass/fail)
- Which tests hit which code
- Analyzes coverage results
  - Where to focus testing