S.T.A.R. @
Faculty of Informatics
University of Lugano
Switzerland

Supporting

Test Suite Evolution

through Test Case Adaptation

Mehdi Mirzaaghaei

Fabrizio Pastore

Mauro Pezzè

Test Evolution

```
public final class DateTimeField{
..
```

```
public final class CopticChronology{
..

}
```

```
public class Report{
..
  addRule(int line, String file)
}
```

```
ReportTest{
 test(){
   Report r = new Report();
   r.addRule(5, file );
   assertFalse(r.isEmpty());

}
```

Test Evolution

```
public final class DateTimeField{
..
```

```
public final class CopticChronology{
..

}
```

```
public class Report{
..
  addRule(int line, Context ctx)
}
```

```
ReportTest{
 test(){
   Report r = new Report();
   r.addRule(5, file );
   assertFalse(r.isEmpty());

}
```

Test Evolution

```
public final class DateTimeField{
..
```

```
public final class CopticChronology{
..

}
```

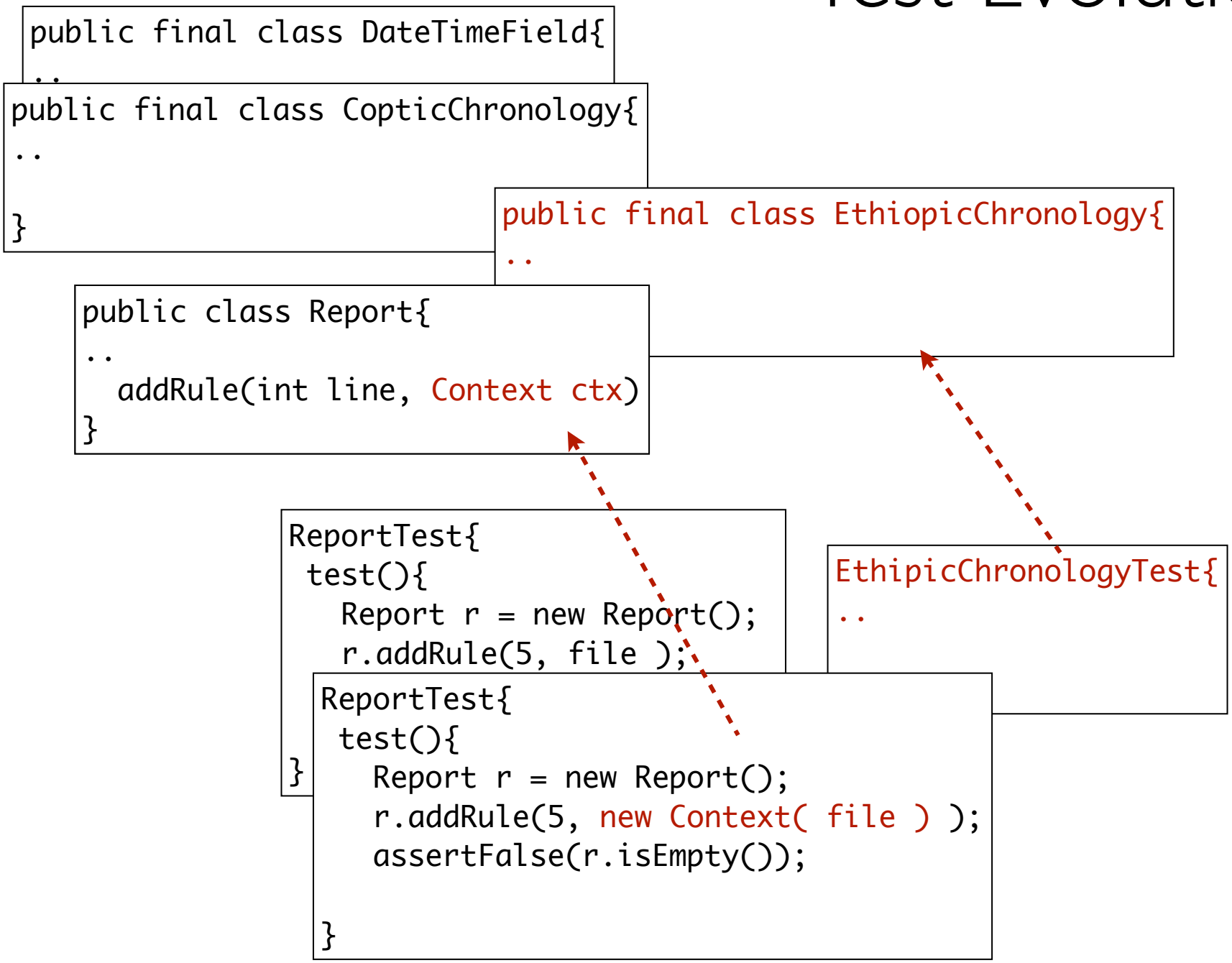
```
public class Report{
..
  addRule(int line, Context ctx)
}
```

```
ReportTest{
 test(){
   Report r = new Report();
   r.addRule(5, file );
}
```

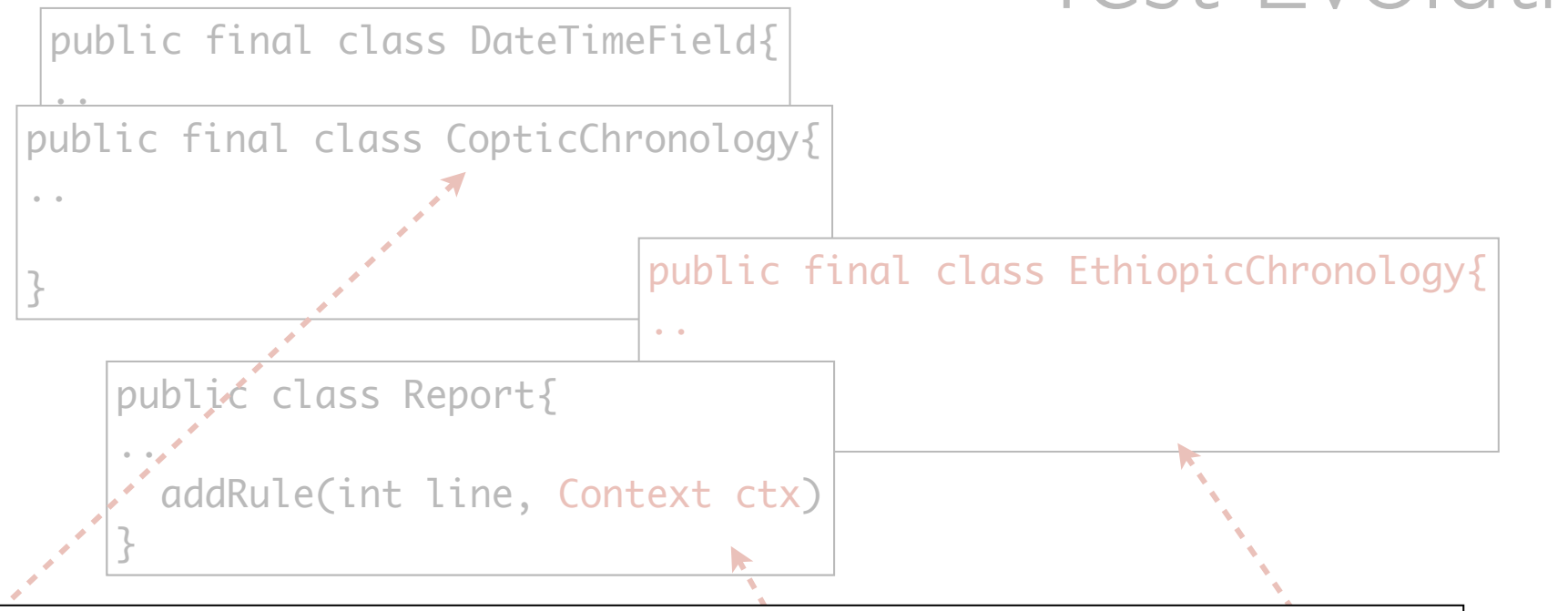
```
ReportTest{
 test(){
   Report r = new Report();
   r.addRule(5, new Context( file ) );
   assertFalse(r.isEmpty());

}
```

Test Evolution



Test Evolution

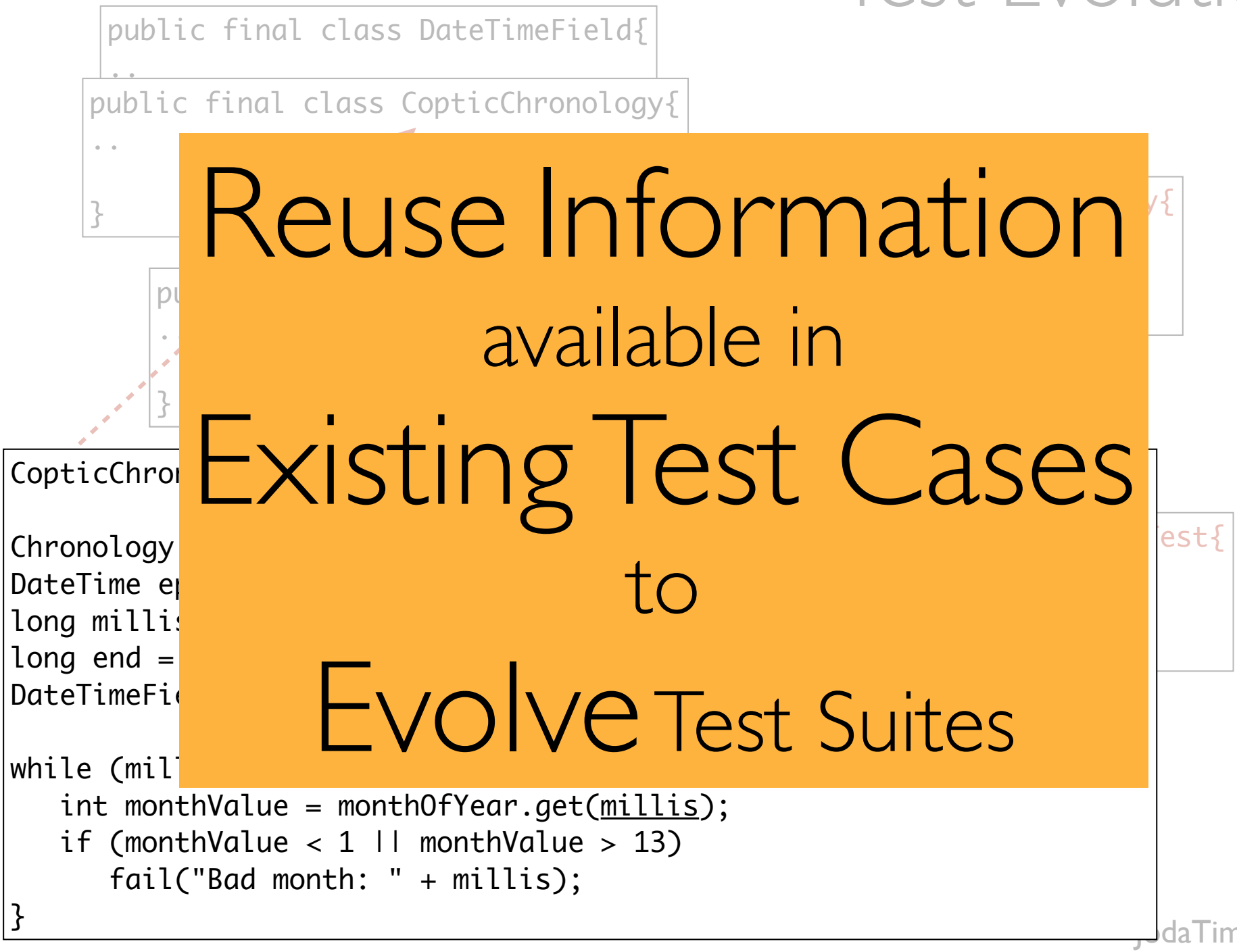



```
CopticChronologyTest{

Chronology COPTIC_UTC = CopticChronology.getInstanceUTC();
DateTime epoch = new DateTime(1, 1, 1, 0, 0, 0, 0, COPTIC_UTC);
long millis = epoch.getMillis();
long end = new DateTime(3000, 1, 1, 0, 0, 0, 0, ISO_UTC).getMillis();
DateTimeField monthOfYear = COPTIC_UTC.monthOfYear();

while (millis < end) {
   int monthValue = monthOfYear.get(millis);
   if (monthValue < 1 || monthValue > 13)
      fail("Bad month: " + millis);
}
```

Test Evolution



```
EthiopicChronologyTest{

Chronology ETHIOPIC_UTC = EthiopicChronology.getInstanceUTC();
DateTime epoch = new DateTime(1, 1, 1, 0, 0, 0, 0, ETHIOPIC_UTC);
long millis = epoch.getMillis();
long end = new DateTime(3000, 1, 1, 0, 0, 0, 0, ISO_UTC).getMillis();
DateTimeField monthOfYear = ETHIOPIC_UTC.monthOfYear();

while (millis < end) {
   int monthValue = monthOfYear.get( new Date(millis) );
   if (monthValue < 1 || monthValue > 13)
      fail("Bad month: " + millis);
} }
```

```
CopticChronologyTest{

Chronology COPTIC_UTC = CopticChronology.getInstanceUTC();
DateTime epoch = new DateTime(1, 1, 1, 0, 0, 0, 0, COPTIC_UTC);
long millis = epoch.getMillis();
long end = new DateTime(3000, 1, 1, 0, 0, 0, 0, ISO_UTC).getMillis();
DateTimeField monthOfYear = COPTIC_UTC.monthOfYear();

while (millis < end) {
   int monthValue = monthOfYear.get(millis);
   if (monthValue < 1 || monthValue > 13)
      fail("Bad month: " + millis);
}
```

Test Care Assistant (TCA)

Add Test Cases for new Modules

- new Class in Hierarchy
- new Interface Implementation
- New Overloaded Method
- New Overridden Method

Repair Compilation Errors due to Signature Changes

- Parameter Addition
- Parameter Type Change
- Parameter Removal
- Return Type Change

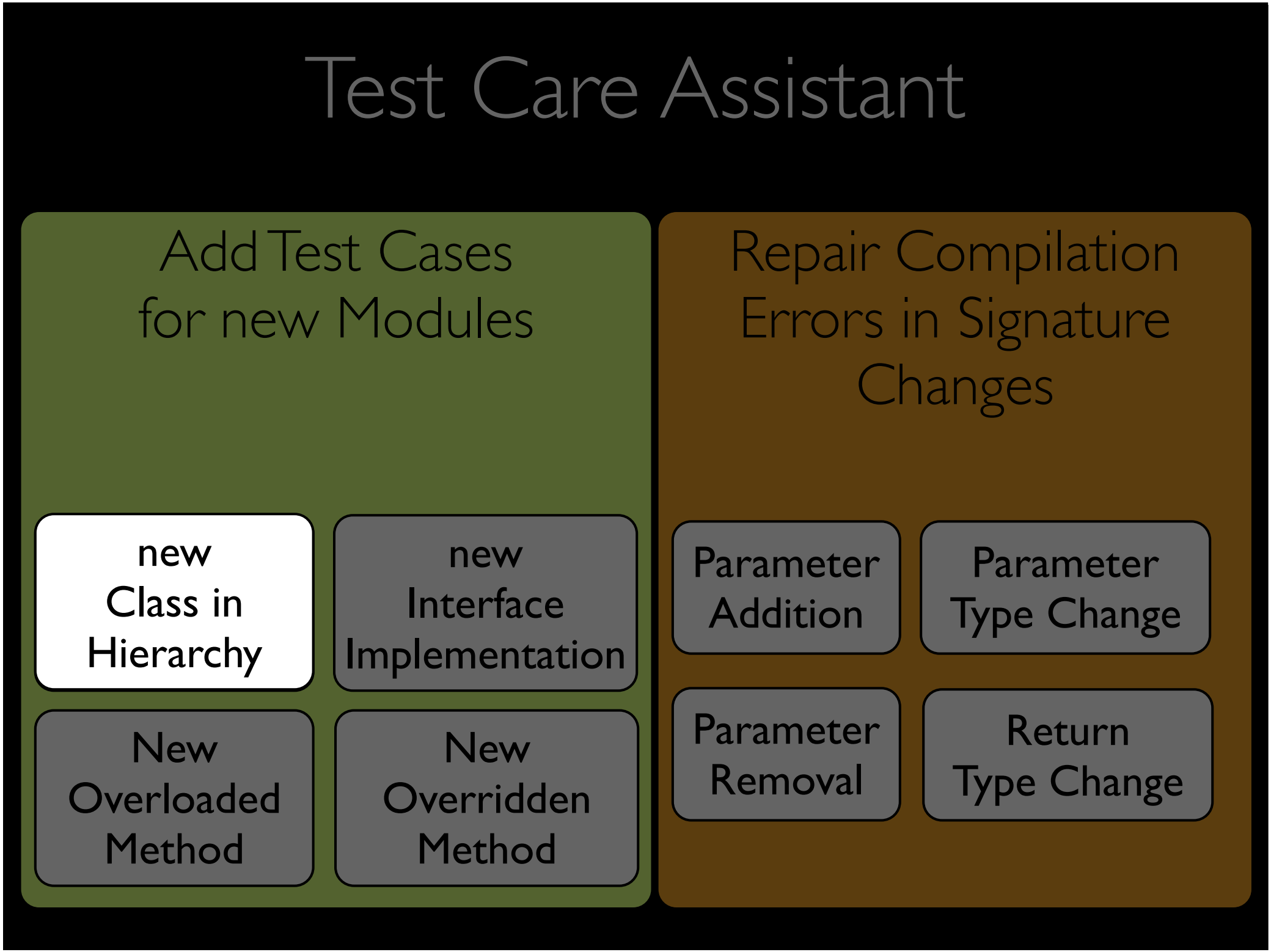
Test Care Assistant (TCA)

Add Test Cases for new Modules

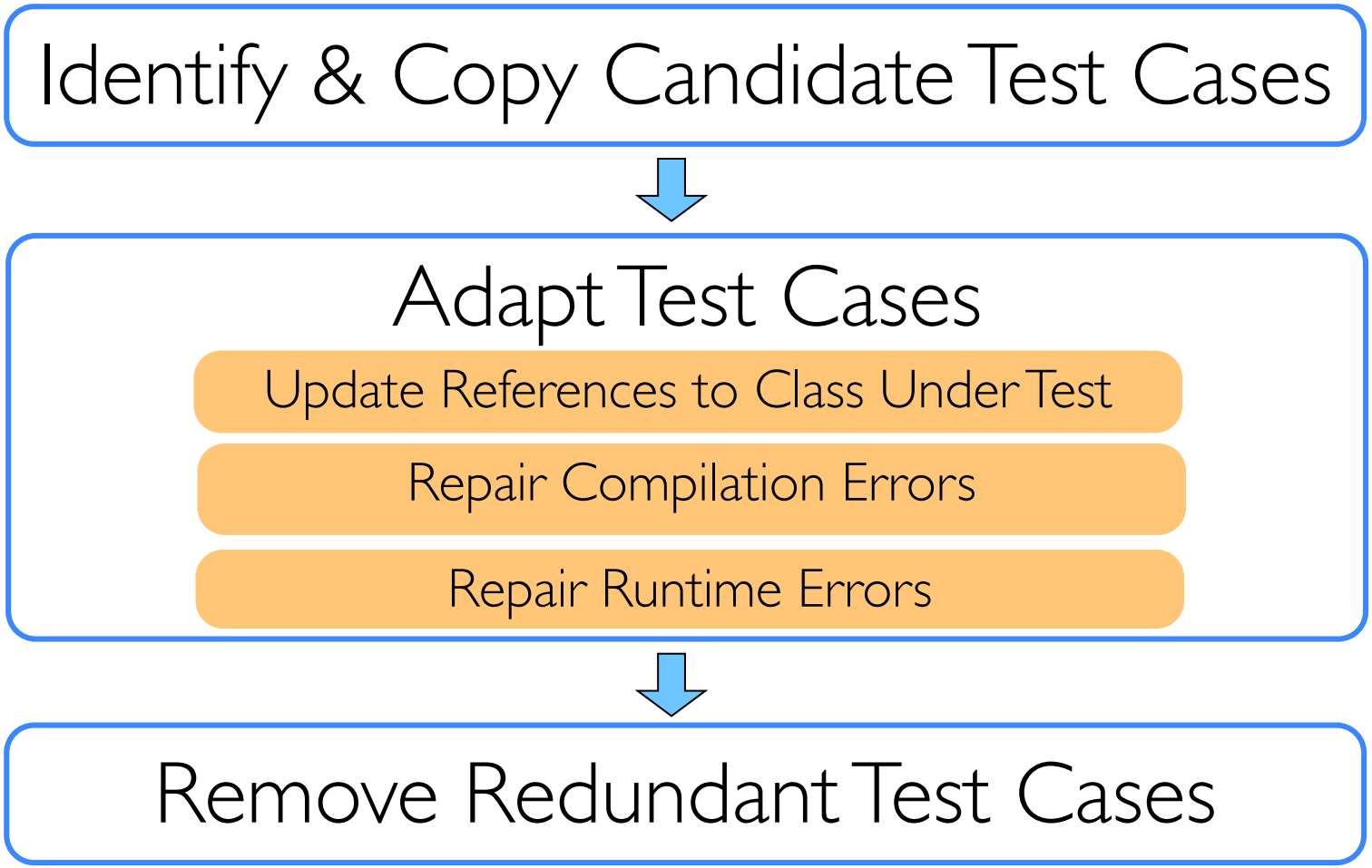
- new Class in Hierarchy
- new Interface Implementation
- New Overloaded Method
- New Overridden Method

Repair Compilation Errors due to Signature Changes

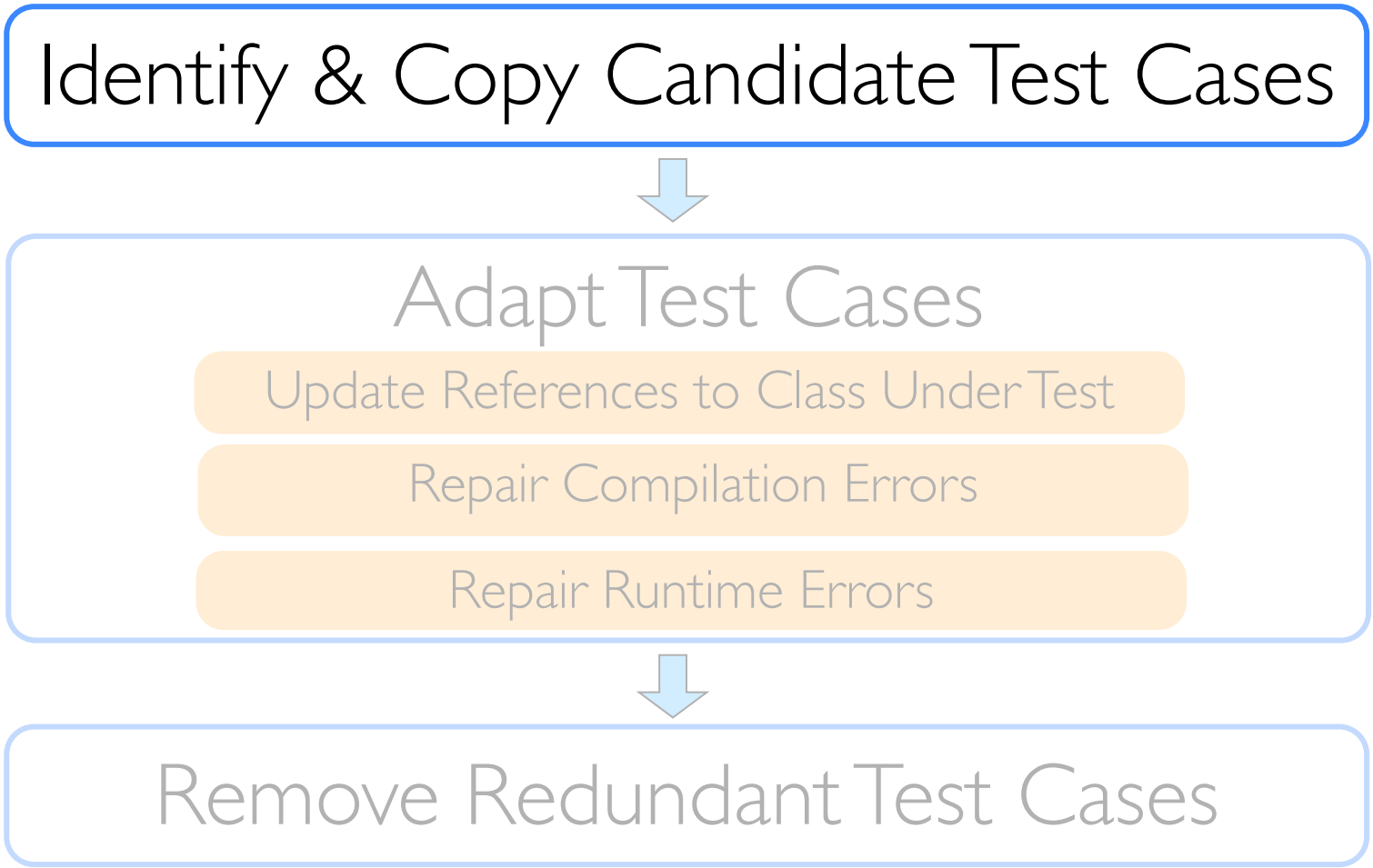
- Parameter Addition
- Parameter Type Change
- Parameter Removal
- Return Type Change


Test Care Assistant
Add Test Cases for new Modules
new Class in Hierarchy
new Interface Implementation
New Overloaded Method
New Overridden Method
Repair Compilation Errors in Signature Changes
Parameter Addition
Parameter Type Change
Parameter Removal
Return Type Change

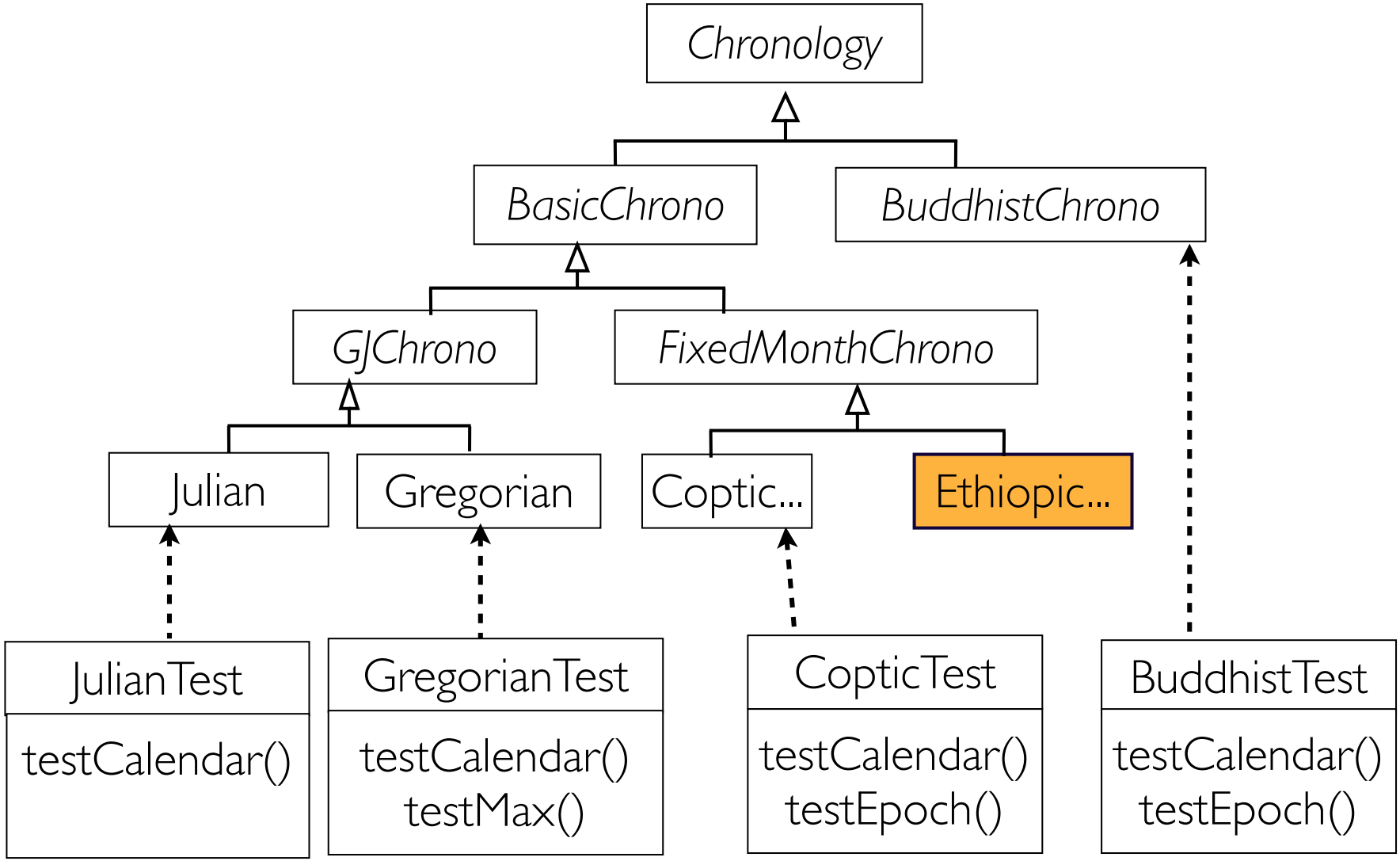
Generate Test Cases for Classes added to a Hierarchy



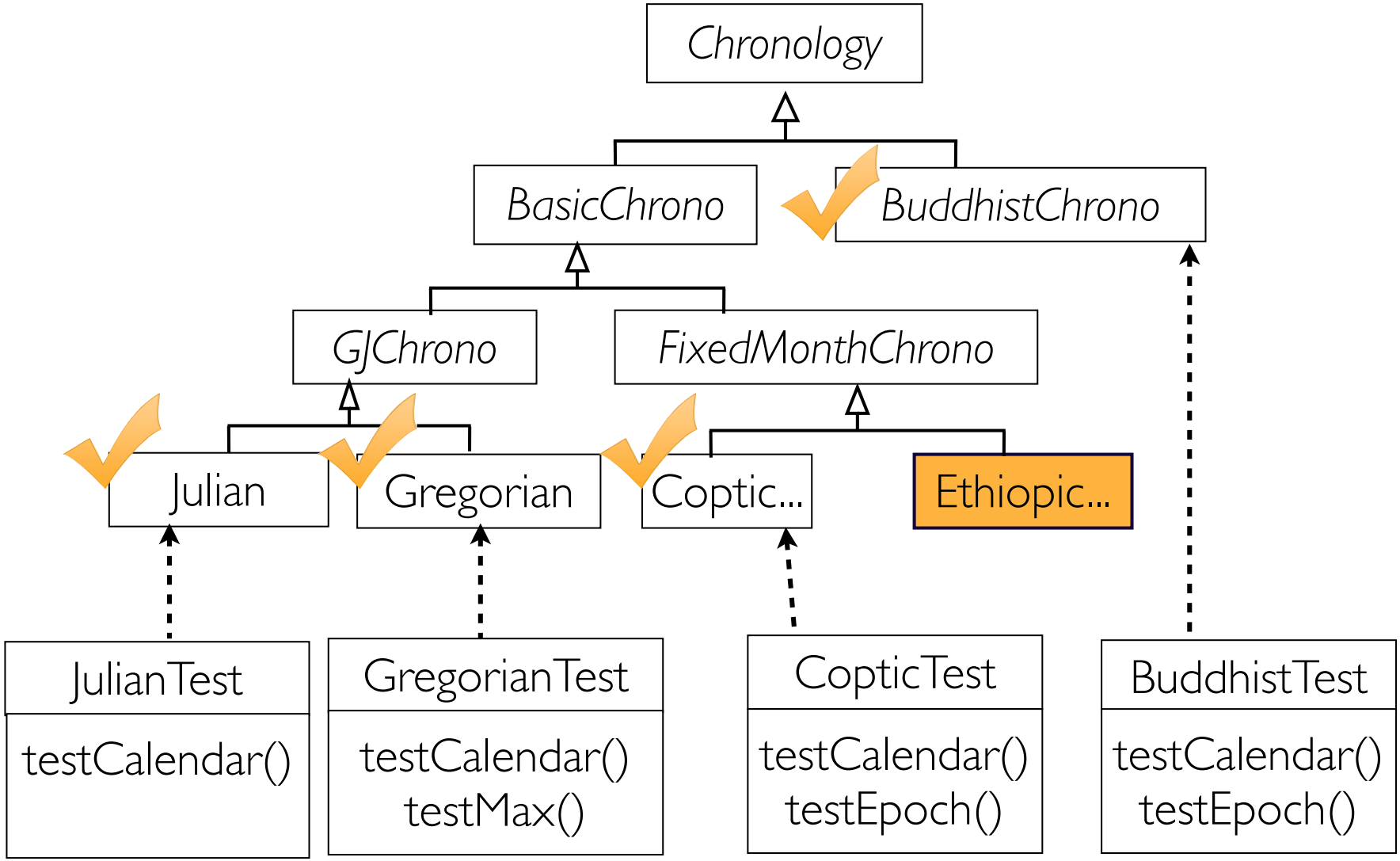
Generate Test Cases for Classes added to a Hierarchy



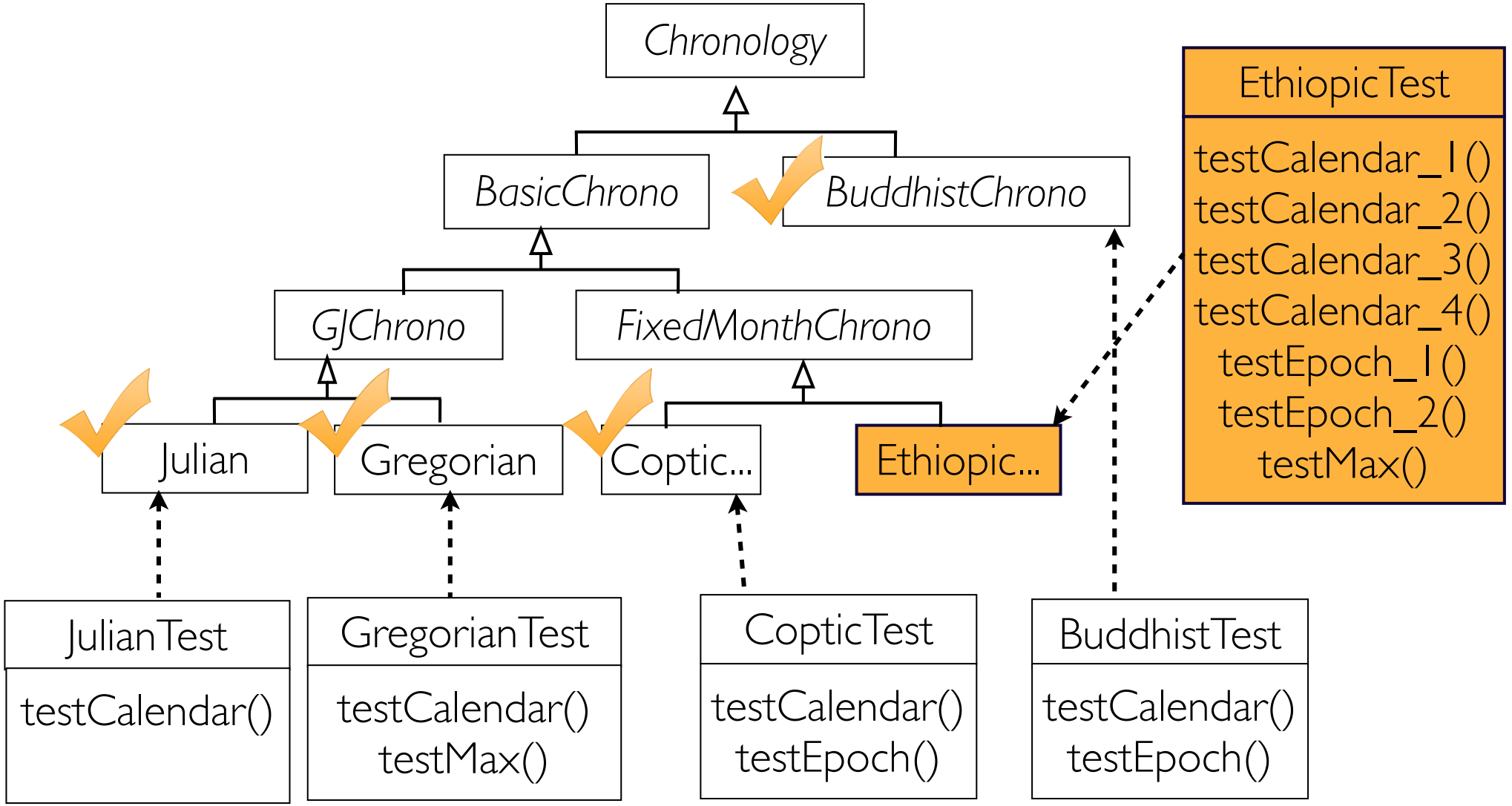
Identify & Copy Candidate Test Cases



Identify & Copy Candidate Test Cases

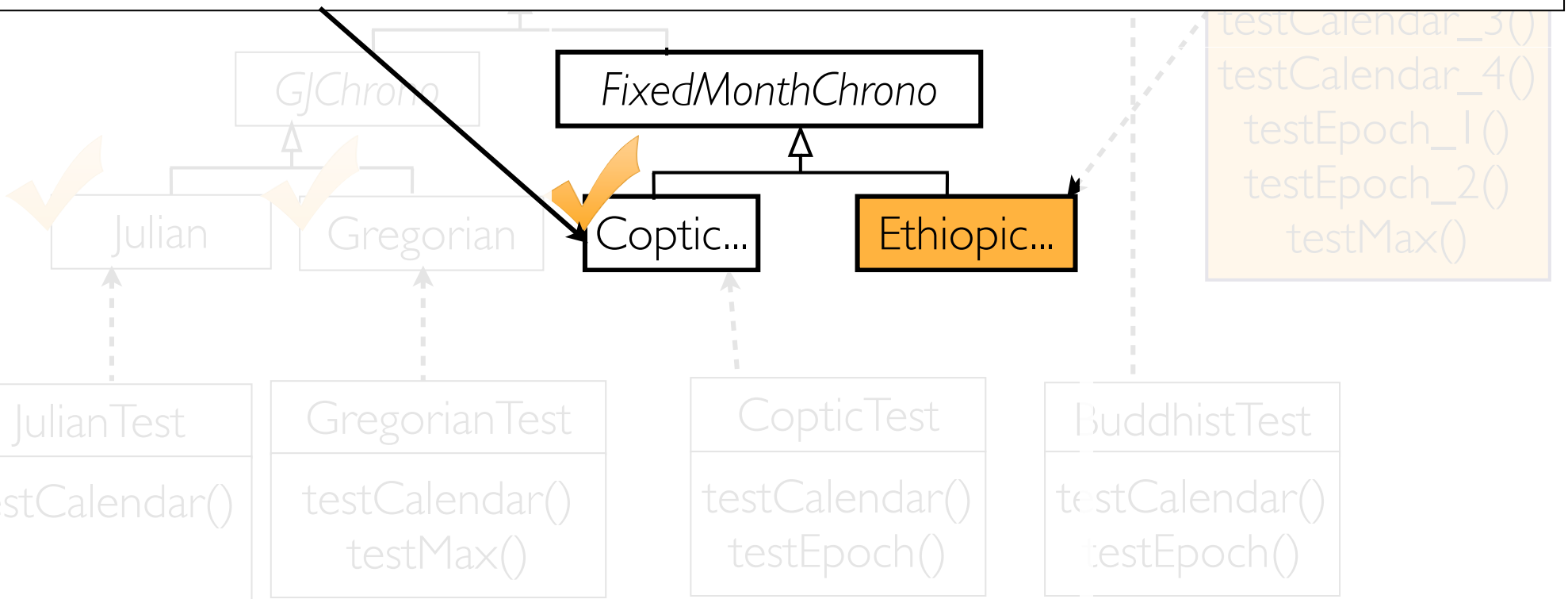


Identify & Copy Candidate Test Cases



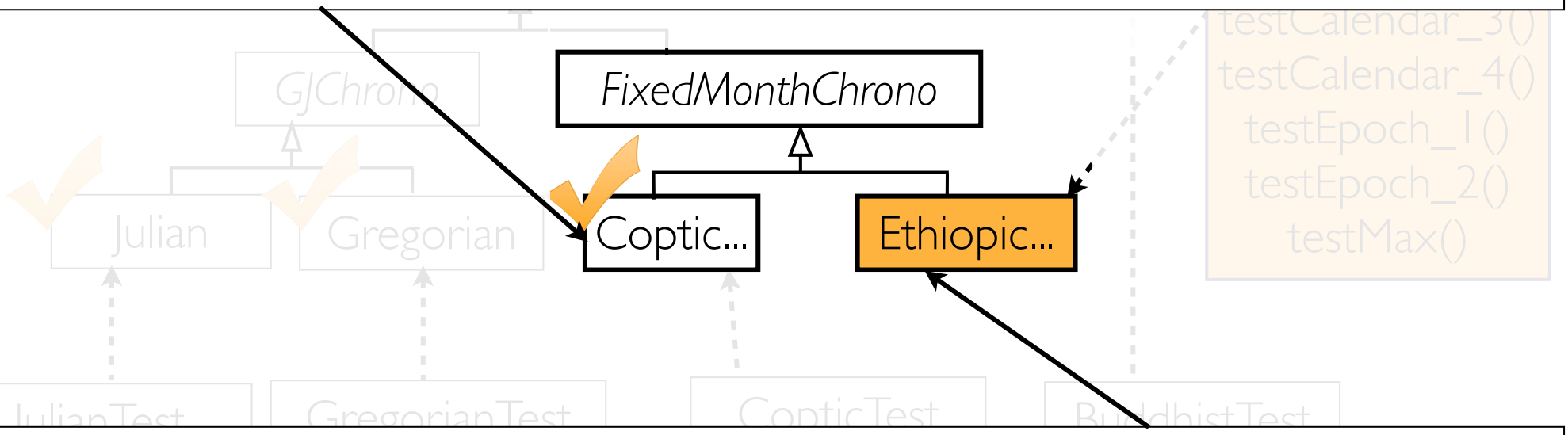
Identify & Copy Candidate Test Cases

```
Chronology COPTIC_UTC = CopticChronology.getInstanceUTC();
DateTime epoch = new DateTime(1, 1, 1, 0, 0, 0, 0, COPTIC_UTC);
assertEquals( CopticChronology.AM, epoch.getEra() )
...
```



Update References to Class Under Test

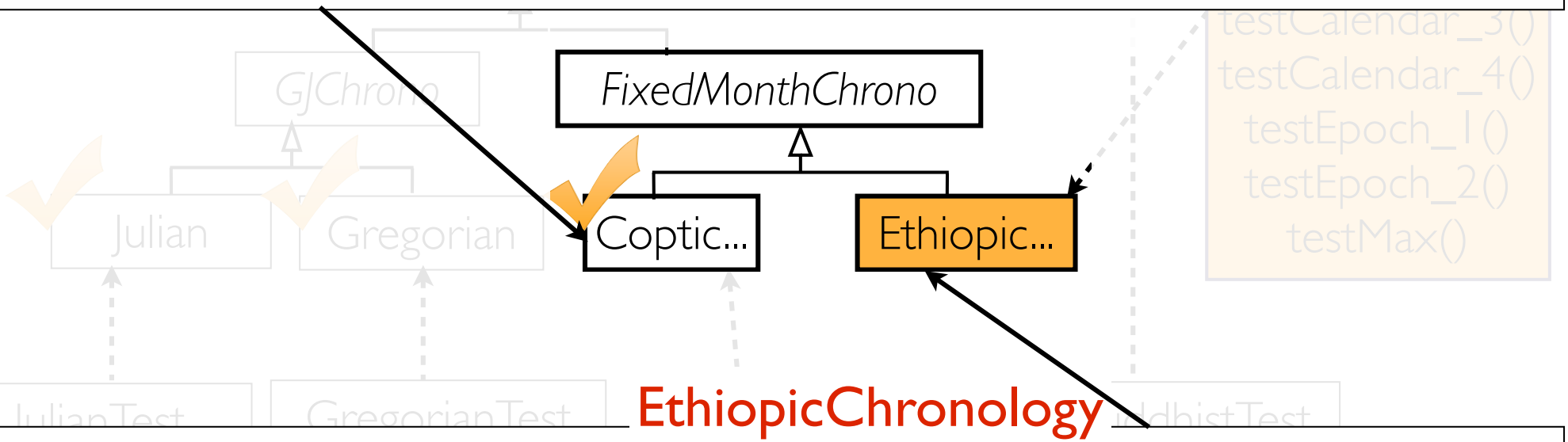
```
Chronology COPTIC_UTC = CopticChronology.getInstanceUTC();
DateTime epoch = new DateTime(1, 1, 1, 0, 0, 0, 0, COPTIC_UTC);
assertEquals( CopticChronology.AM, epoch.getEra() )
...
```




```
Chronology COPTIC_UTC = CopticChronology.getInstanceUTC();
DateTime epoch = new DateTime(1, 1, 1, 0, 0, 0, 0, COPTIC_UTC);
assertEquals( CopticChronology.AM, epoch.getEra() )
...
```

Update References to Class Under Test

```
Chronology COPTIC_UTC = CopticChronology.getInstanceUTC();
DateTime epoch = new DateTime(1, 1, 1, 0, 0, 0, 0, COPTIC_UTC);
assertEquals( CopticChronology.AM, epoch.getEra() )
...
```

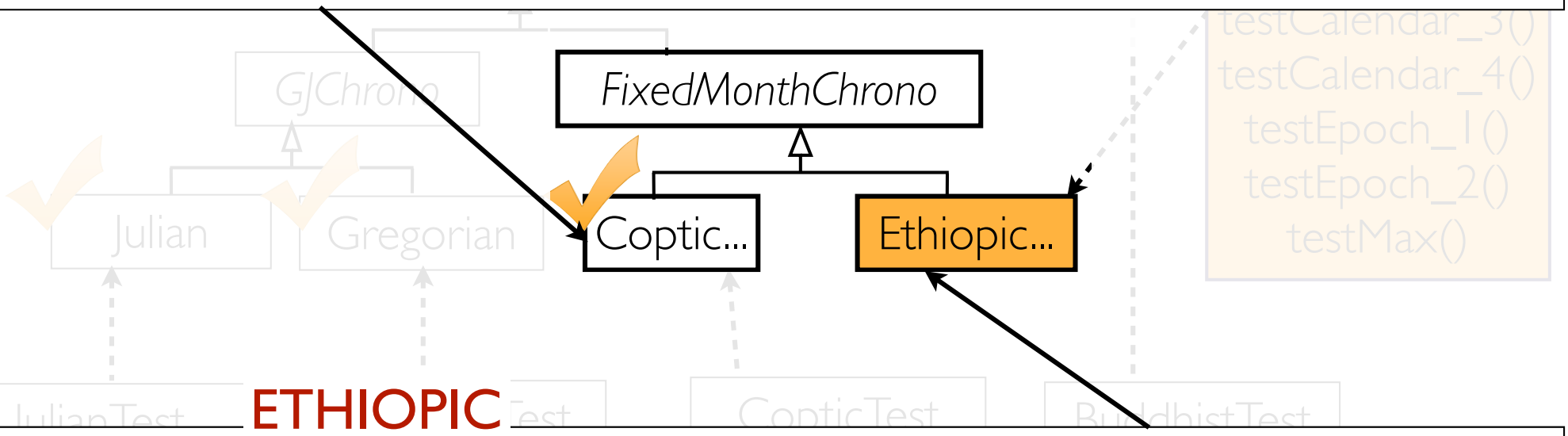



```
Chronology COPTIC_UTC = CopticChronology.getInstanceUTC();
DateTime epoch = new DateTime(1, 1, 1, 0, 0, 0, 0, COPTIC_UTC);
assertEquals( CopticChronology.AM, epoch.getEra() )
...
```

EthiopicChronology

Update References to Class Under Test

```
Chronology COPTIC_UTC = CopticChronology.getInstanceUTC();
DateTime epoch = new DateTime(1, 1, 1, 0, 0, 0, 0, COPTIC_UTC);
assertEquals( CopticChronology.AM, epoch.getEra() )
...
```



ETHIOPIC

```
Chronology ~~COPTIC~~_UTC = EthiopicChronology.ETHIOPIC_UTC();
DateTime epoch = new DateTime(1, 1, 1, 0, 0, 0, 0, ~~COPTIC~~_UTC);
assertEquals( EthiopicChronology.AM, epoch.getEra() )
...
```

Adapt Candidate Test Cases

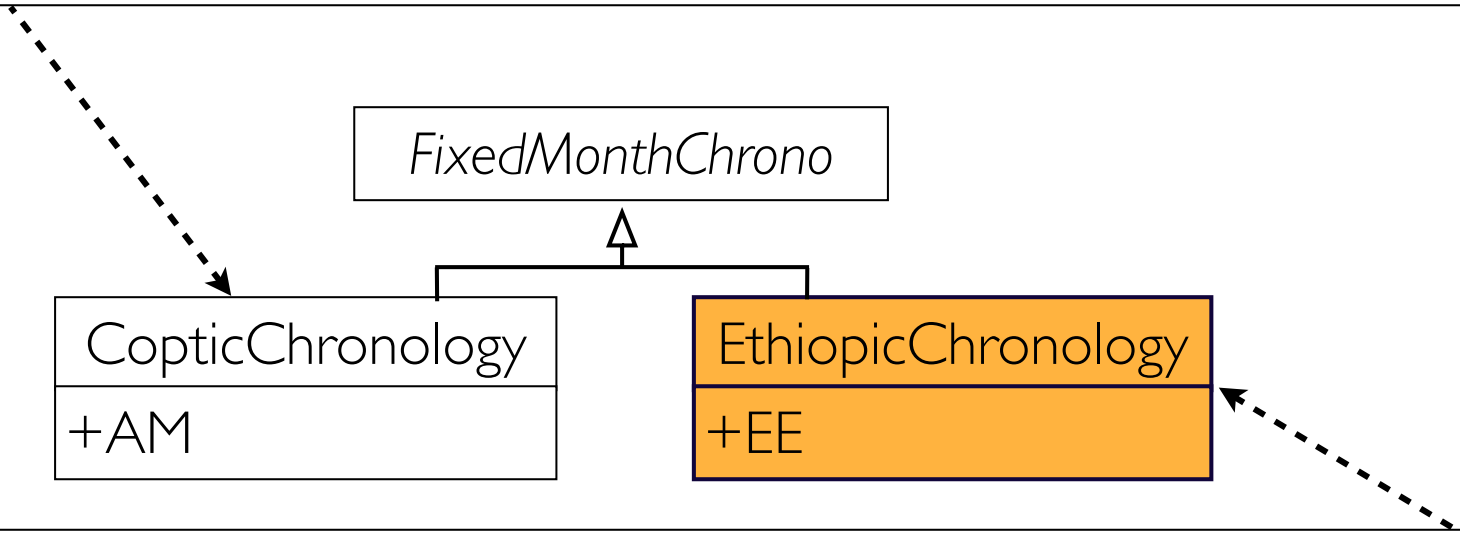
Undefined Fields

Missing Constructors

Missing Methods

Adapting Undefined Fields

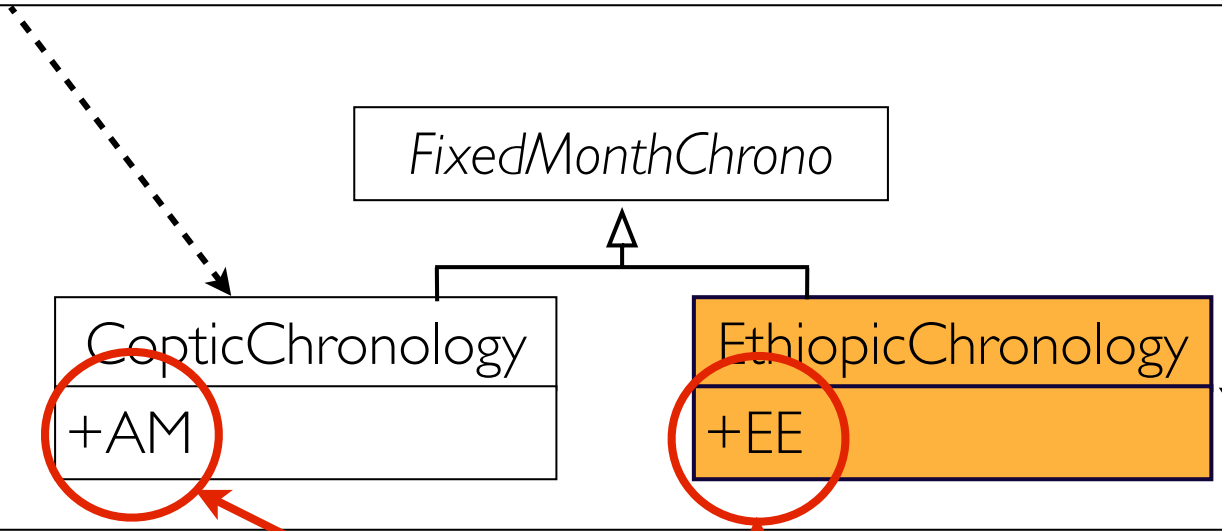
```
Chronology COPTIC_UTC = CopticChronology.getInstanceUTC();
DateTime epoch = new DateTime(1, 1, 1, 0, 0, 0, 0, COPTIC_UTC);
assertEquals( CopticChronology.AM, epoch.getEra() )
...
```




```
Chronology ETHIOPIC_UTC = EthiopicChronology.getInstanceUTC();
DateTime epoch = new DateTime(1, 1, 1, 0, 0, 0, 0, ETHIOPIC_UTC);
assertEquals( EthiopicChronology.AM, epoch.getEra() )
```

Adapting Undefined Fields

```
Chronology COPTIC_UTC = CopticChronology.getInstanceUTC();
DateTime epoch = new DateTime(1, 1, 1, 0, 0, 0, 0, COPTIC_UTC);
assertEquals( CopticChronology.AM, epoch.getEra() )
...
```




```
Chronology ETHIOPIC_UTC = EthiopicChronology.getInstanceUTC();
DateTime epoch = new DateTime(1, 1, 1, 0, 0, 0, 0, ETHIOPIC_UTC);
assertEquals( EthiopicChronology.AM, epoch.getEra() )
```

```
class CopticChronology {

    /** Serialization lock */
 long serialVersionUID = -59L;

 /**
 * Constant value for ...
 */
 public static final int AM =
  DateTimeConstants.CE;

 /** A singleton era field. */
 private DateTimeField
  ERA_FIELD = new
  SingleEraDateTimeField("AM");
```

```
class EthiopicChronology {

 /** Serialization lock */
 long serialVersionUID = -59L;

 /**
 * Constant value for ...
 */
 public static final int EE =
  DateTimeConstants.CE;

 /** A singleton era field. */
 private ateTimeField
  ERA_FIELD = new
  SingleEraDateTimeField("EE");
```

+AM

+EE

```
Chronology ETHIOPIC_UTC = EthiopicChronology.getInstanceUTC();
DateTime epoch = new DateTime(1, 1, 1, 0, 0, 0, 0, ETHIOPIC_UTC);
assertEquals( EthiopicChronology.AM, epoch.getEra() )
```

```
class CopticChronology {

    /** Serialization lock */
 long serialVersionUID = -59L;

 /**
 * Constant value for ...
 */
 public static final int AM =
  DateTimeConstants.CE;

 /** A singleton era field. */
 private DateTimeField
  ERA_FIELD = new
  SingleEraDateTimeField("AM");
```

```
class EthiopicChronology {

 /** Serialization lock */
 long serialVersionUID = -59L;

 /**
 * Constant value for ...
 */
 public static final int EE =
  DateTimeConstants.CE;

 /** A singleton era field. */
 private ateTimeField
  ERA_FIELD = new
  SingleEraDateTimeField("EE");
```

+AM

+EE

```
Chronology ETHIOPIC_UTC = EthiopicChronology.getInstanceUTC();
DateTime epoch = new DateTime(1, 1, 1, 0, 0, 0, 0, ETHIOPIC_UTC);
assertEquals( EthiopicChronology.AM, epoch.getEra() )
```

```
class CopticChronology {

    /** Serialization lock */
 long serialVersionUID = -59L;

 /**
 * Constant value for ...
 */
 public static final int AM =
  DateTimeConstants.CE;

 /** A singleton era field. */
 private DateTimeField
  ERA_FIELD = new
  SingleEraDateTimeField("AM");
```

```
class EthiopicChronology {

 /** Serialization lock */
 long serialVersionUID = -59L;

 /**
 * Constant value for ...
 */
 public static final int EE =
  DateTimeConstants.CE;

 /** A singleton era field. */
 private ateTimeField
  ERA_FIELD = new
  SingleEraDateTimeField("EE");
```

+AM

+EE

```
Chronology ETHIOPIC_UTC = EthiopicChronology.getInstanceUTC();
DateTime epoch = new DateTime(1, 1, 1, 0, 0, 0, 0, ETHIOPIC_UTC);
assertEquals( EthiopicChronology.AM, epoch.getEra() )
```

```
class CopticChronology {

    /** Serialization lock */
 long serialVersionUID = -59L;

 /**
 * Constant value for ...
 */
 public static final int AM =
  DateTimeConstants.CE;

 /** A singleton era field. */
 private DateTimeField
  ERA_FIELD = new
  SingleEraDateTimeField("AM");
```

```
class EthiopicChronology {

 /** Serialization lock */
 long serialVersionUID = -59L;

 /**
 * Constant value for ...
 */
 public static final int EE =
  DateTimeConstants.CE;

 /** A singleton era field. */
 private ateTimeField
  ERA_FIELD = new
  SingleEraDateTimeField("EE");
```

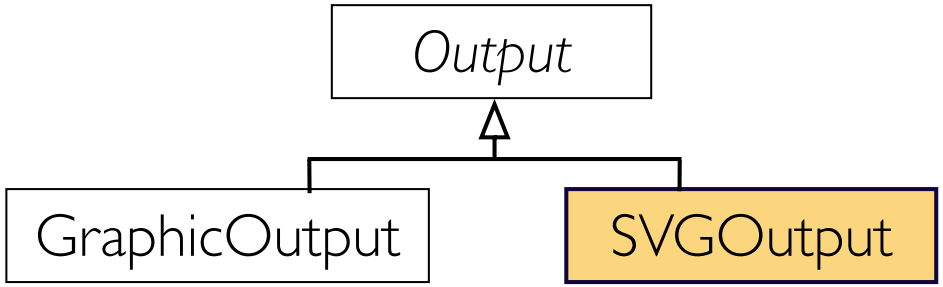
+AM

+EE

```
Chronology ETHIOPIC_UTC = EthiopicChronology.getInstanceUTC();
DateTime epoch = new DateTime(1, 1, 1, 0, 0, 0, 0, ETHIOPIC_UTC);
assertEquals( EthiopicChronology.AM, epoch.getEra() )
```

EE

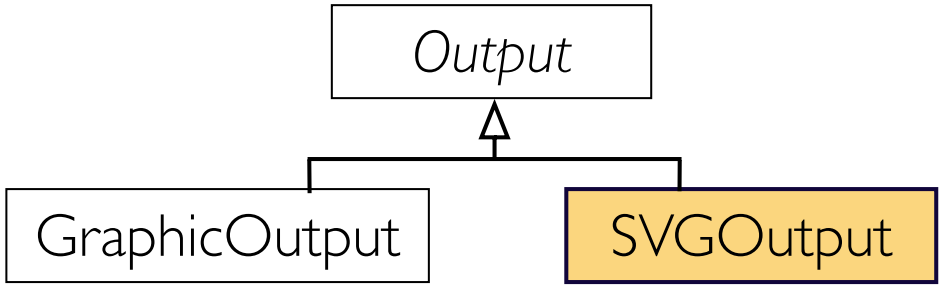
Adapting Undefined Constructors




```
public void testDefaultFont() {
 g = new GraphicsMock();
 output = new GraphicsOutput(
      g,
      DefaultEnv.DEFAULT_FONT,
      fgColour,
      bgColour);
 [..]
 assertEquals("graphicsOutput",
output.toString());
}
```

```
public void testDefaultFont() {
 g = new GraphicsMock();
 output = new SVGOutput(
      g,
      DefaultEnv.DEFAULT_FONT,
      fgColour,
      bgColour);
 [..]
 assertEquals("graphicsOutput",
output.toString());
}
```

Adapting Undefined Constructors



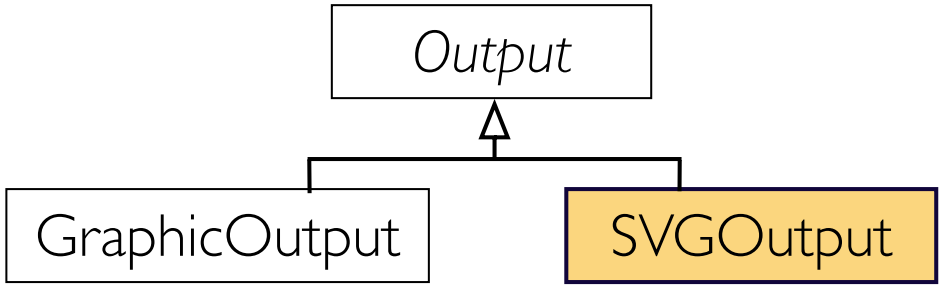

```
public void testDefaultFont() {
 g = new GraphicsMock();
 output = new GraphicsOutput(
      g,
      DefaultEnv.DEFAULT_FONT,
      fgColour,
      bgColour);
 [..]
 assertE
output.t
}
```

```
public void testDefaultFont() {
 g = new GraphicsMock();
 output = new SVGOutput(
      g,
      DefaultEnv.DEFAULT_FONT,
      fgColour,
      bgColour);

 ertEquals("graphicsOutput",
 ut.toString());
```

Compilation error:
The constructor SVGOutput(Graphics2D, Font, Color, Color) is undefined

Adapting Undefined Constructors




```
public void testDefaultFont() {
 g = new GraphicsMock();
 ou                    GraphicsOutput(

                                    DEFAULT_FONT,

 [
 g                                   LA",
ou                                  ));
}
```

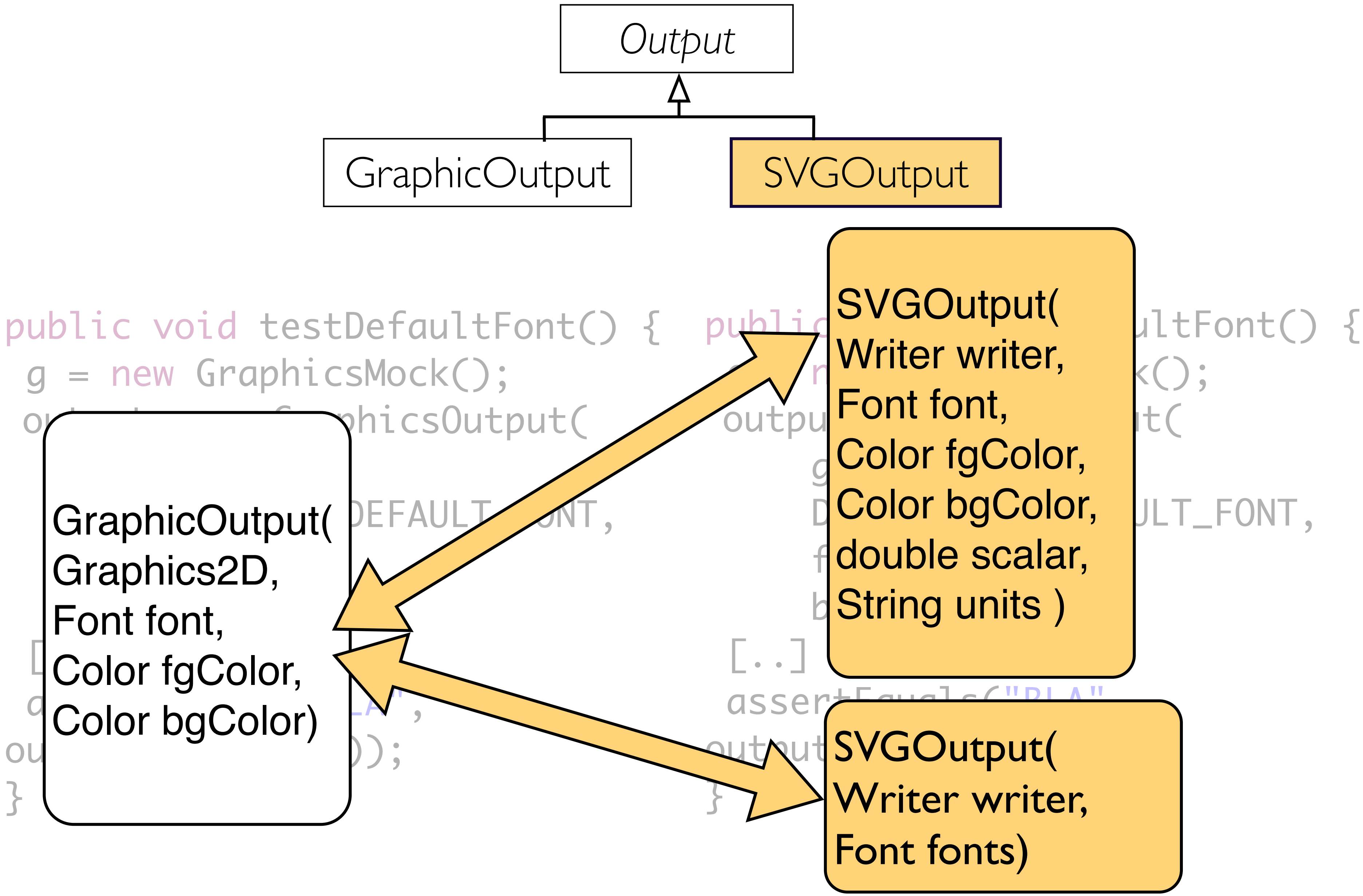
GraphicOutput(
Graphics2D,
Font font,
Color fgColor,
Color bgColor)

```
public                              ultFont() {
 g = n                              k();
 outpu                              t(
      g
      D                             ULT_FONT,
      f
      b
 [..]
 assert
output
}
```

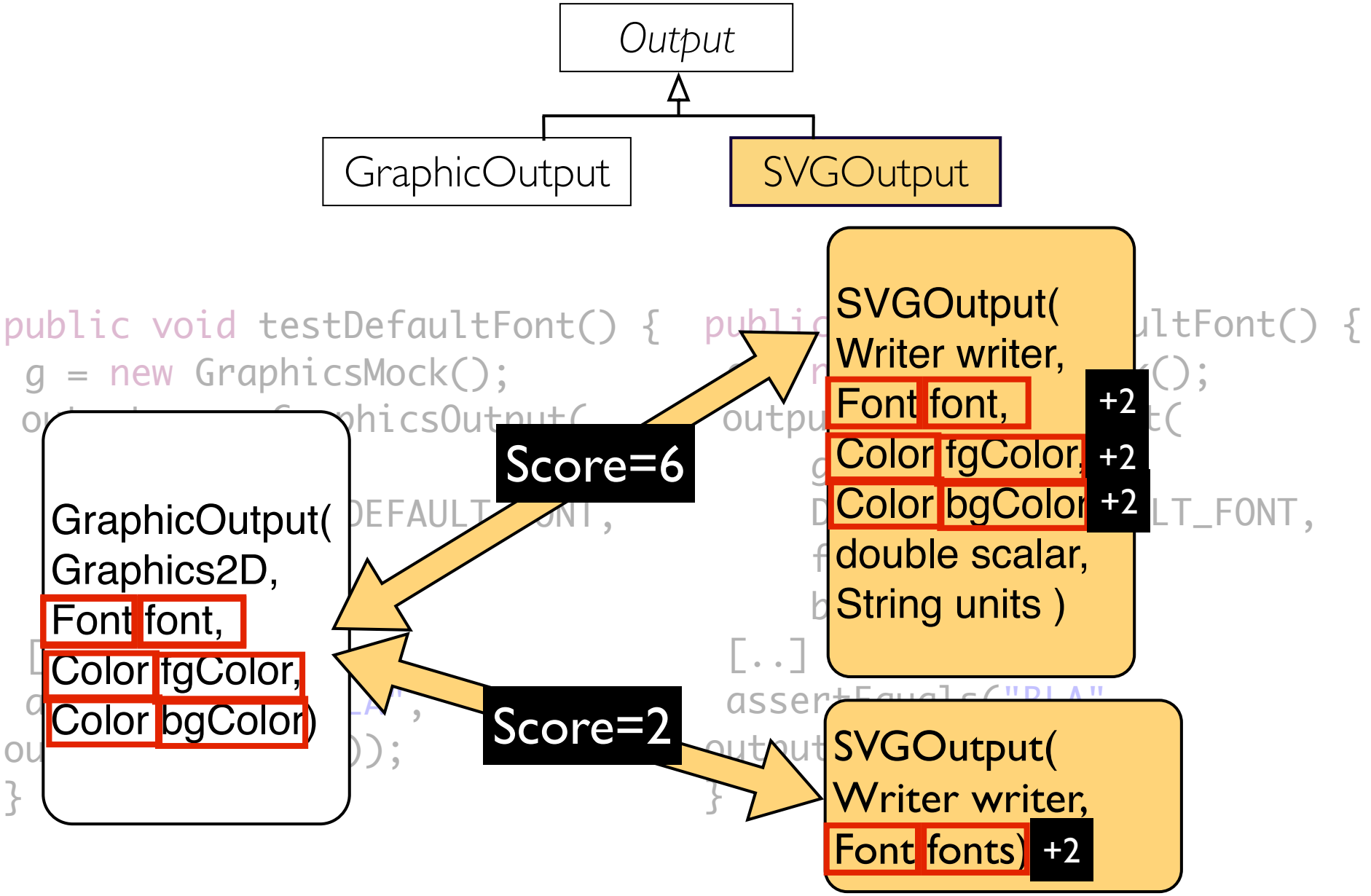
SVGOutput(
Writer writer,
Font font,
Color fgColor,
Color bgColor,
double scalar,
String units)

SVGOutput(
Writer writer,
Font fonts)

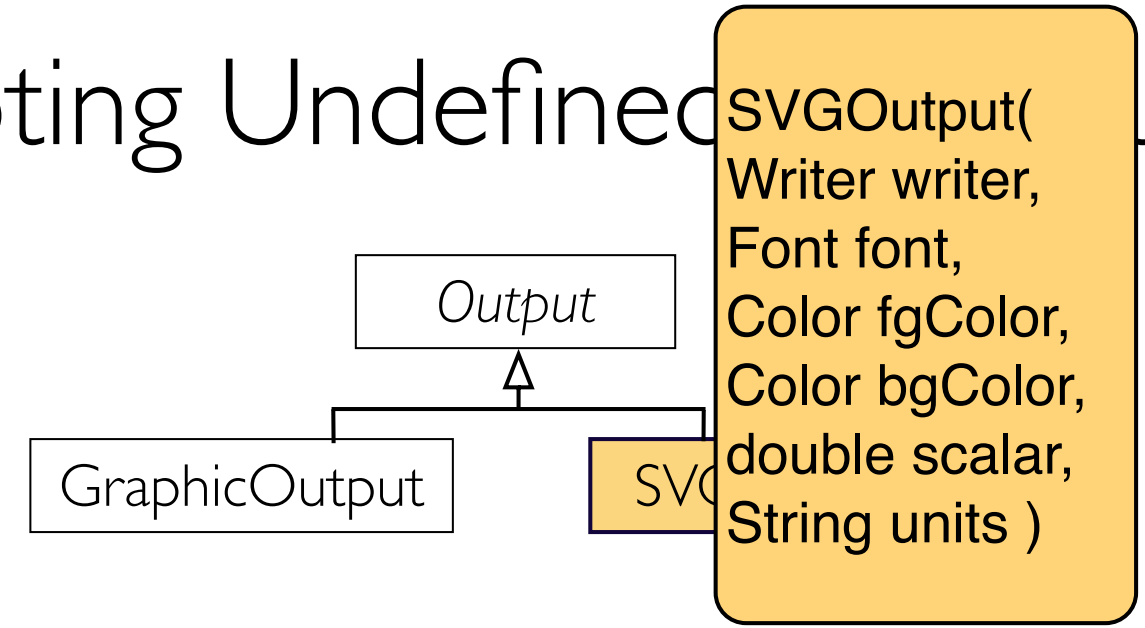
Adapting Undefined Constructors



Adapting Undefined Constructors



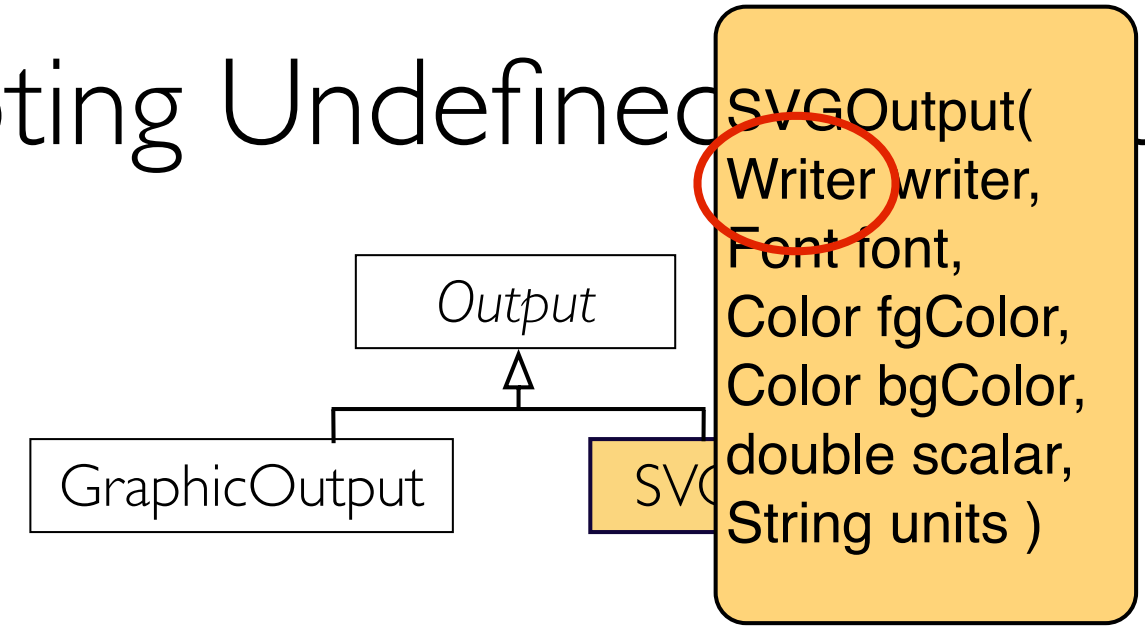
Adapting Undefined uctors




```
public void testDefaultFont() {
 g = new GraphicsMock();
 output = new GraphicsOutput(
      g,
      DefaultEnv.DEFAULT_FONT,
      fgColour,
      bgColour);
  [..]
 assertEquals("graphicsOutput",
output.toString());
}
```

```
public void testDefaultFont() {
 g = new GraphicsMock();
 output = new SVGOutput(
      ....,
      DefaultEnv.DEFAULT_FONT,
      fgColour,
      bgColour,
       ...,
        ...;
 [..]
 assertEquals("graphicsOutput",
output.toString());
}
```

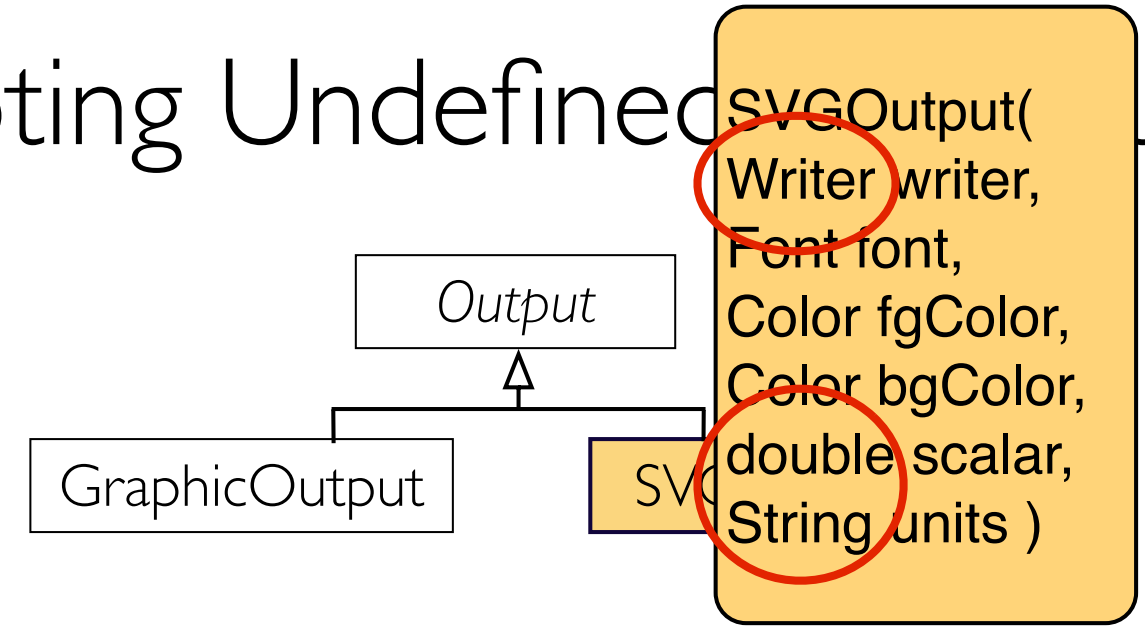
Adapting Undefined uctors




```
public void testDefaultFont() {
 g = new GraphicsMock();
 output = new GraphicsOutput(
      g,
      DefaultEnv.DEFAULT_FONT,
      fgColour,
      bgColour);
 [..]
 assertEquals("graphicsOutput",
output.toString());
}
```

```
public void testDefaultFont() {
 g = new GraphicsMock();
 output = new SVGOutput(
      new StringWriter( EasyMock(Writer)),
      DefaultEnv.DEFAULT_FONT,
      fgColour,
      bgColour,
        ...,
         ...;
 [..]
 assertEquals("graphicsOutput",
output.toString());
}
```

Adapting Undefined uctors




```
public void testDefaultFont() {
 g = new GraphicsMock();
 output = new GraphicsOutput(
      g,
      DefaultEnv.DEFAULT_FONT,
      fgColour,
      bgColour);
 [..]
 assertEquals("graphicsOutput",
output.toString());
}
```

```
public void testDefaultFont() {
 g = new GraphicsMock();
 output = new SVGOutput(
      new StringWriter( EasyMock(Writer)),
      DefaultEnv.DEFAULT_FONT,
      fgColour,
      bgColour,
      0,
      "");
 [..]
 assertEquals("graphicsOutput",
output.toString());
}
```

Barbecue 1.5b1

Adapting Runtime Failures

Test Failures: apply ReAssert

Daniel et. al.[ASE09,ISSTA10]

```
public void testDefaultFont() {
 g = new GraphicsMock();
 output = new SVGOutput(
      new StringWriter(EasyMock(Writer)),
      DefaultEnv.DEFAULT_FONT,
      fgColour,
      bgColour,
      0,
      “”;
 [..]
 assertEquals("graphicsOutput", output.toString());
}
```

Adapting Runtime Failures

Test Failures: apply ReAssert

Daniel et. al.[ASE09,ISSTA10]

```
public void testDefaultFont() {
 g = new Graphi
 output = new SV
      new String
      DefaultEnv
      fgColour,
      bgColour,
      0,
      “”;
 [..]
 assertEquals("graphicsOutput", output.toString());
}
```

junit.framework.AssertionFailedError:
expected:<SVGOutput> but
was:<graphicsOutput>

Adapting Runtime Failures

Test Failures: apply ReAssert

Daniel et. al.[ASE09,ISSTA10]

```
public void testDefaultFont() {
 g = new GraphicsMock();
 output = new SVGOutput(
      new StringWriter(EasyMock(Writer)),
      DefaultEnv.DEFAULT_FONT,
      fgColour,
      bgColour,
      0,
      “”;
 [..]
 assertEquals("graphicsOutput", output.toString());
}
```

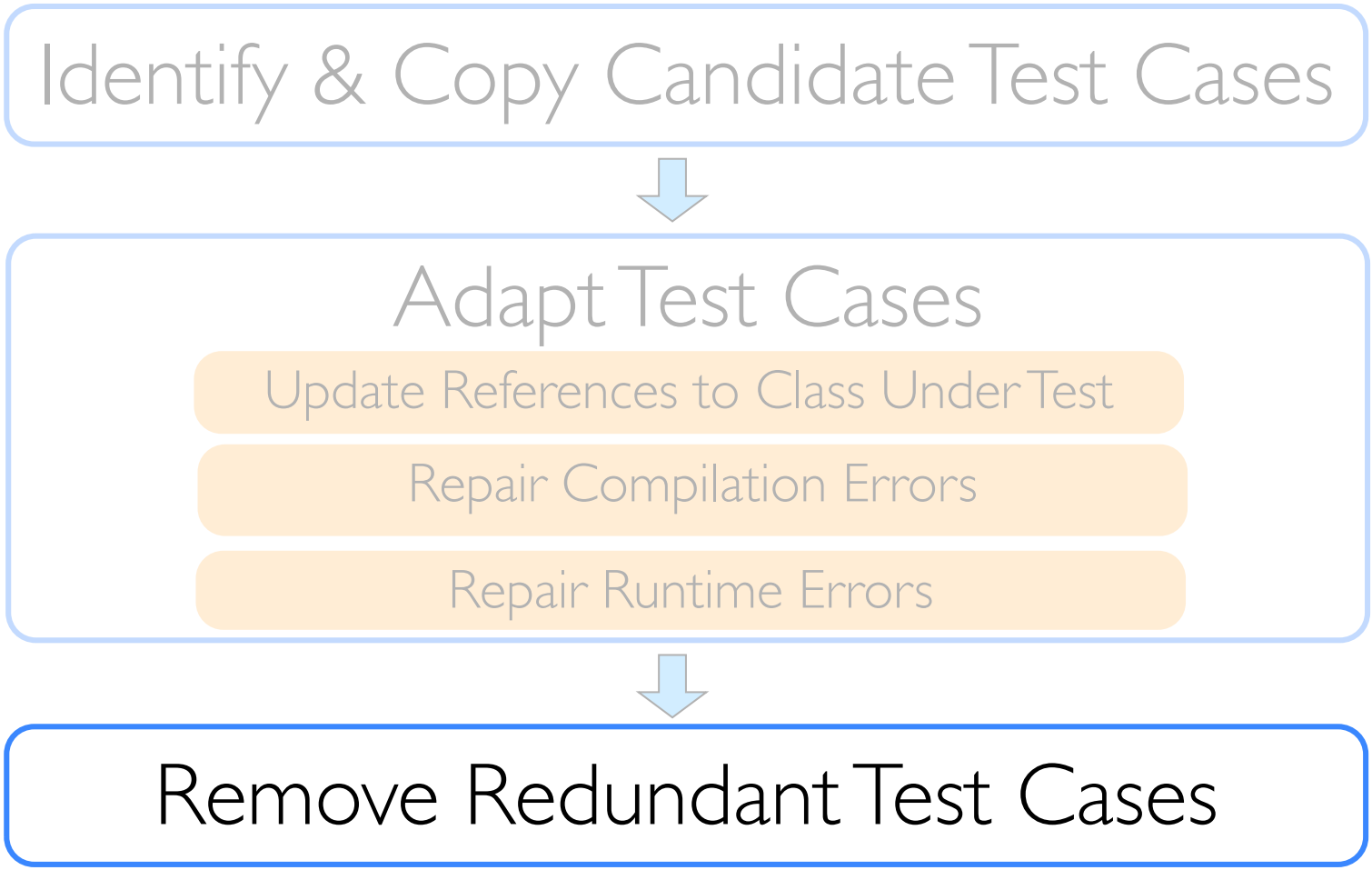
"SVGOutput"

Adapting Runtime Failures

Uncaught Exceptions:

involve users

Add Test Cases



Test Care Assistant

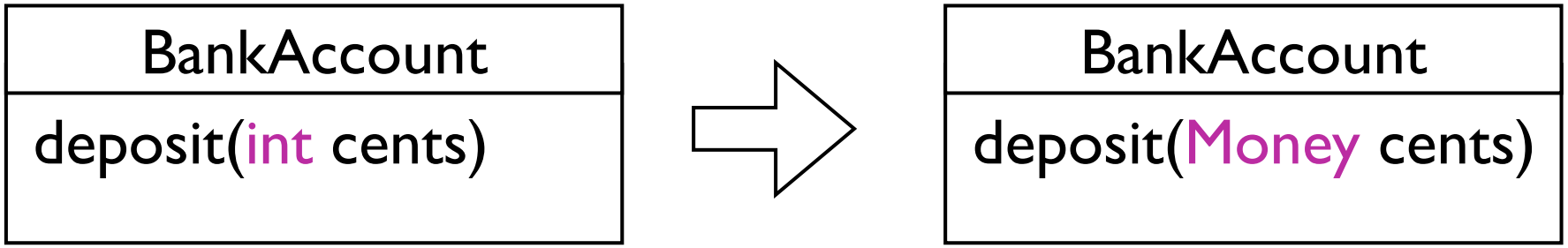
Add Test Cases for new Modules

- new Class in Hierarchy
- new Interface Implementation
- New Overloaded Method
- New Overridden Method

Repair Compilation Errors in Signature Changes

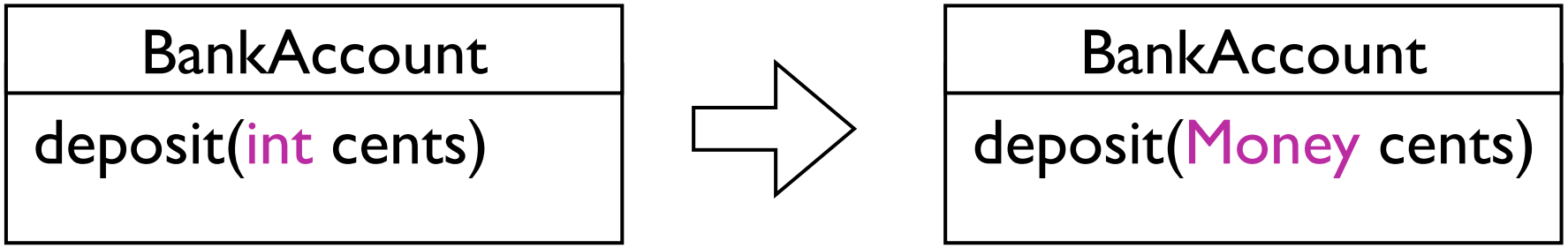
- Parameter Addition
- Parameter Type Change
- Parameter Removal
- Return Type Change

Repair Signature Changes




```
BankAccount account = new BankAccount();
int amount = 500;
account.deposit(amount);
assertEquals( 500, account.getBalance());
```

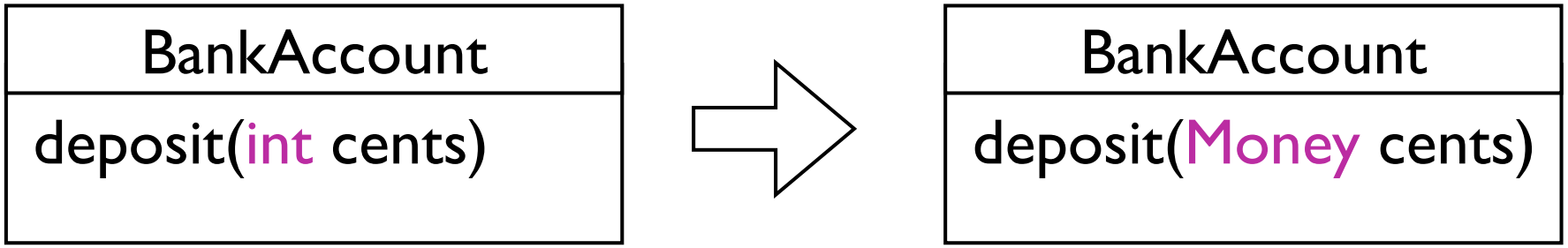
Repair Signature Changes



Compilation error:
Method *deposit(Money)* in the type BankAccount is not applicable for the arguments (int)

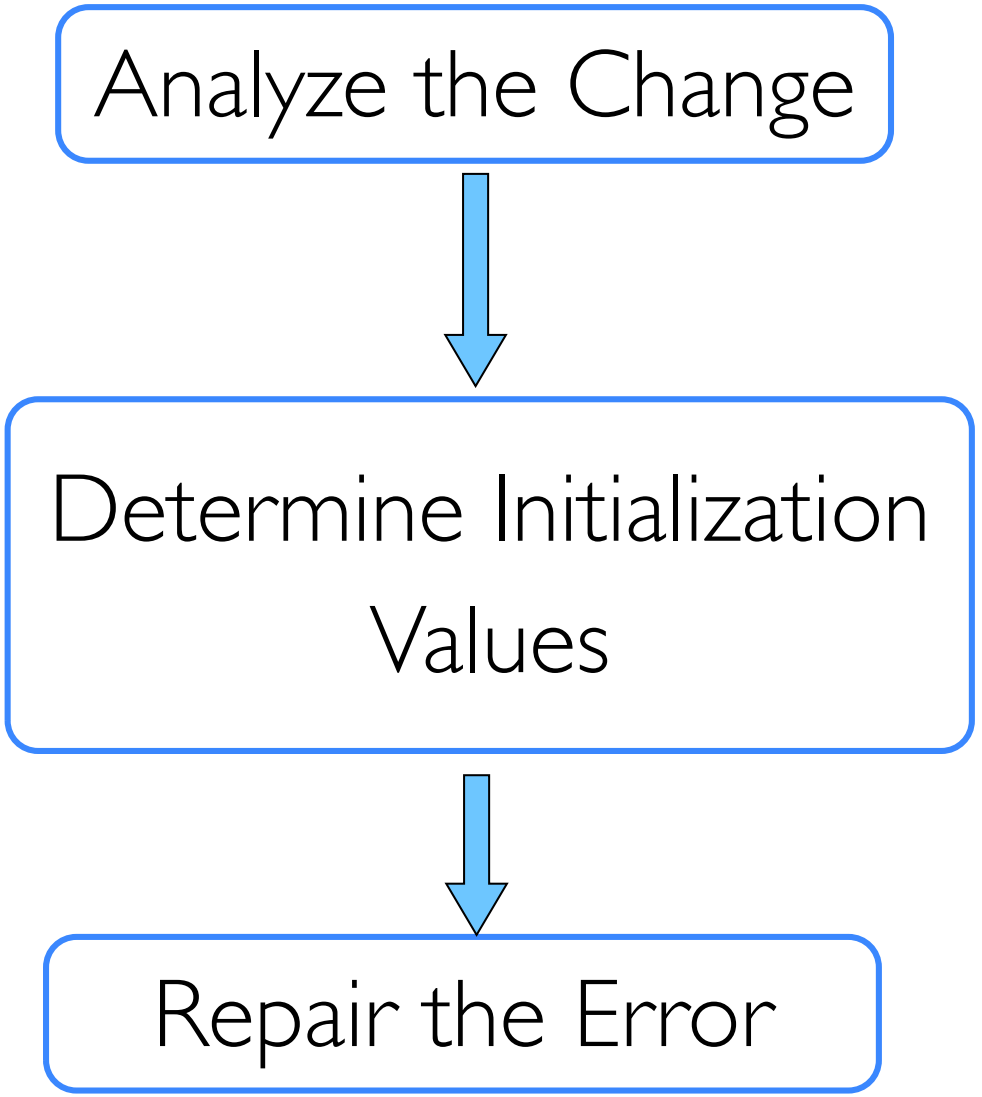
```
BankAccount account = new BankAccount();
int amount = 500;
account.deposit(amount);
assertEquals( 500, account.getBalance());
```

Repair Signature Changes

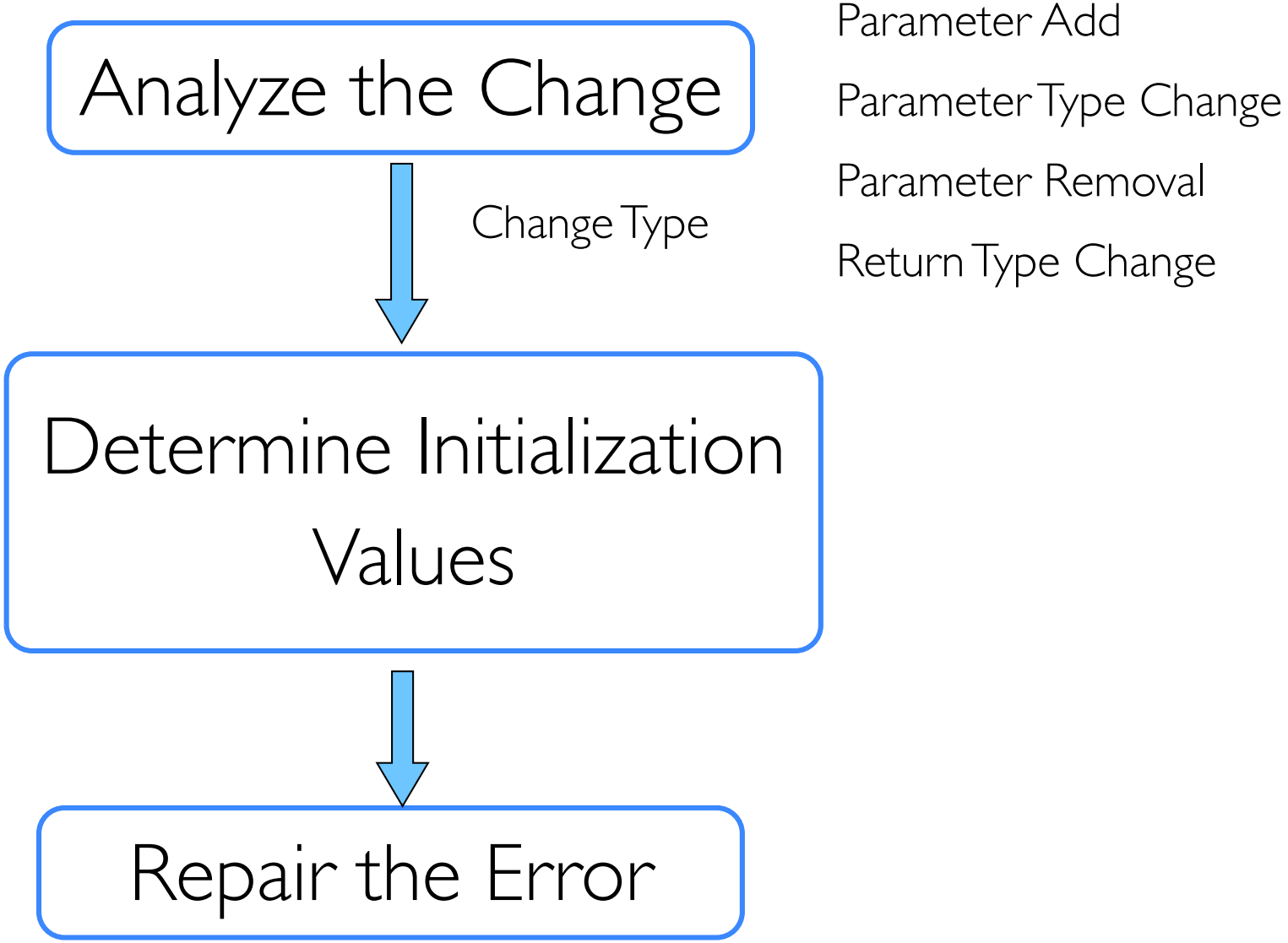



```
BankAccount account = new BankAccount();
Money money = new Money(500);
account.deposit(money);
assertEquals(500, account.getBalance());
```

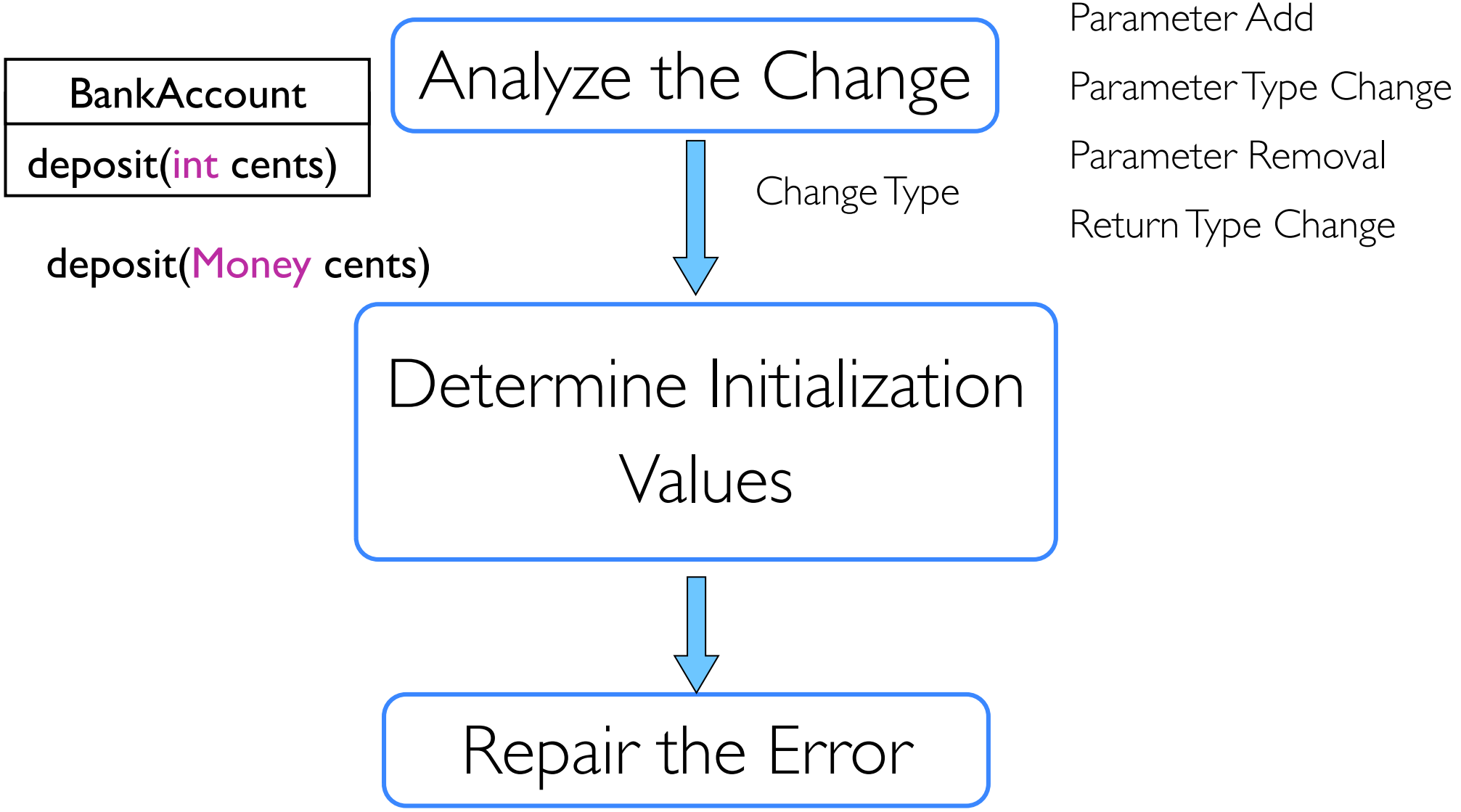
Repair Signature Changes



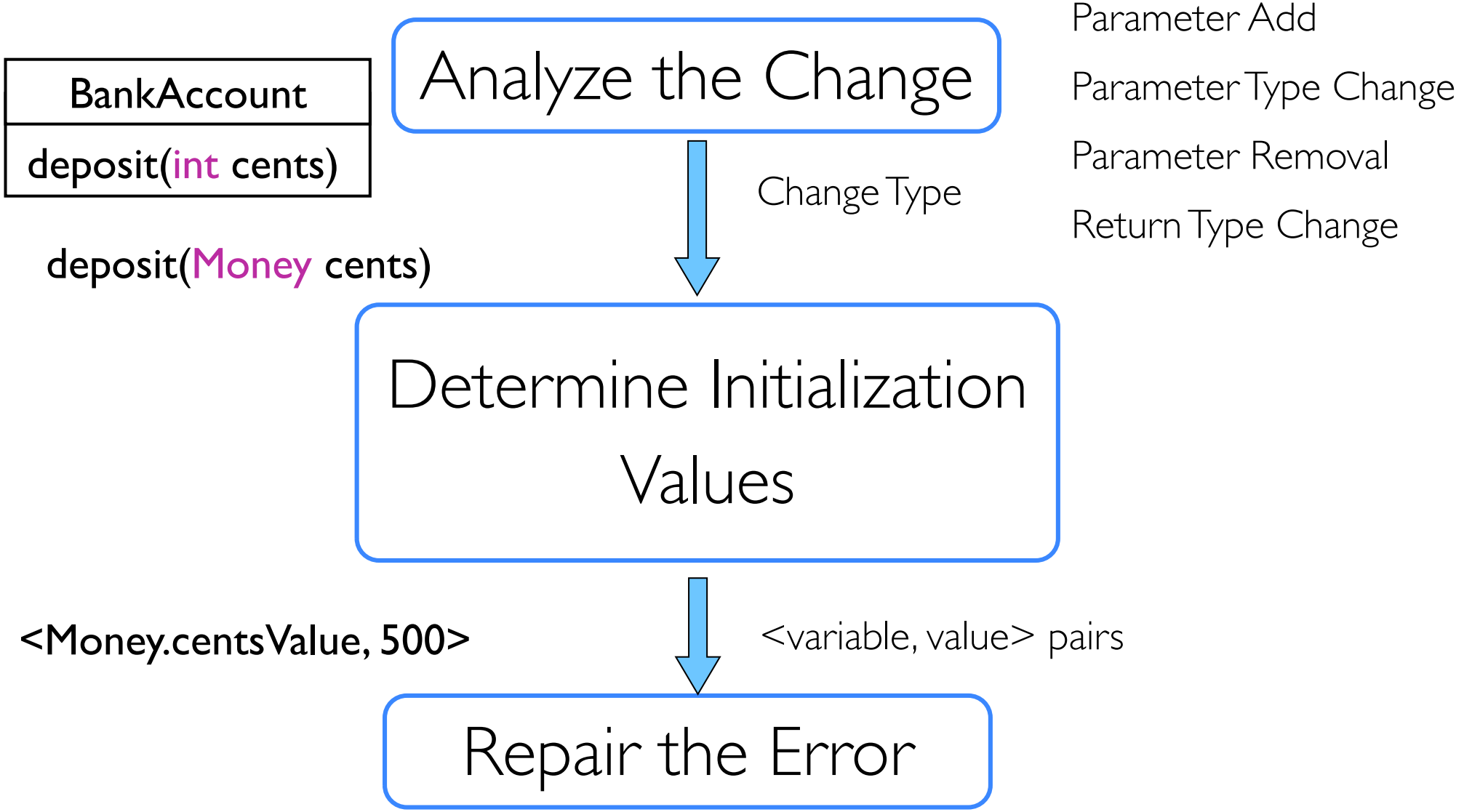
Repair Signature Changes



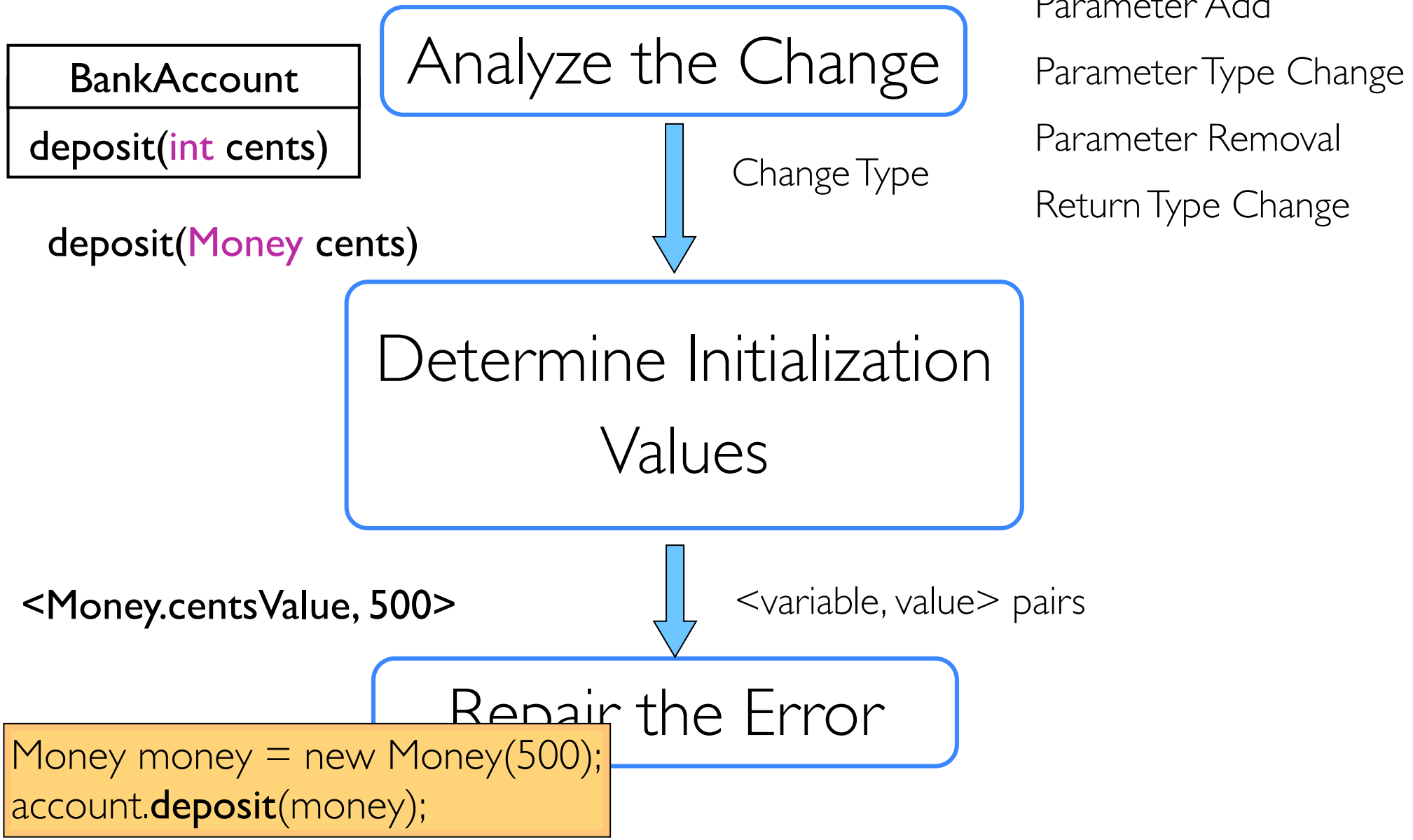
Repair Signature Changes



Repair Signature Changes



Repair Signature Changes



EVALUATION

Q1: APPLICABILITY

Q2: EFFECTIVENESS

Test Care Assistant

Add Test Cases for new Modules

- new Class in Hierarchy
- new Interface Implementation
- New Overloaded Method
- New Overridden Method

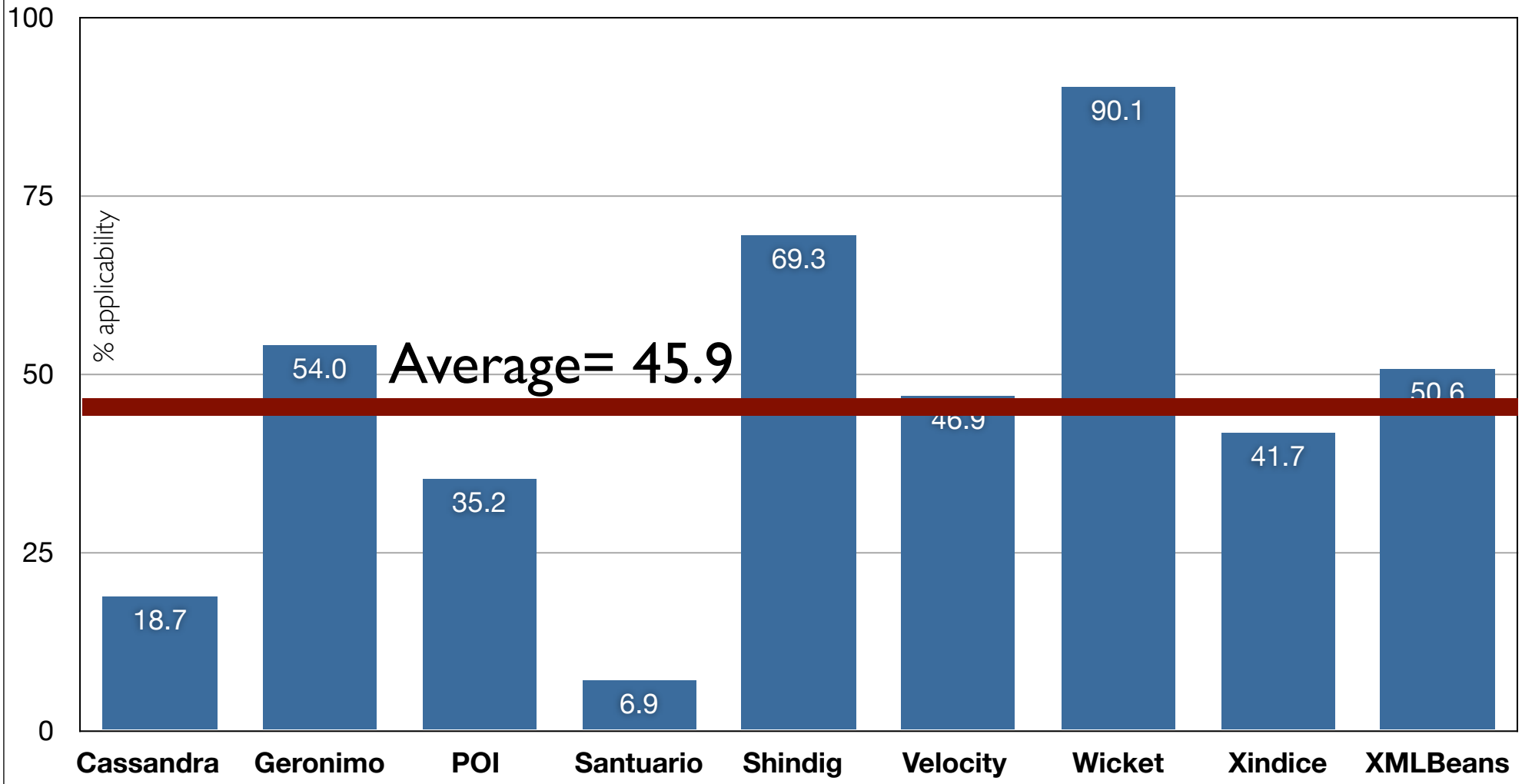
Repair Compilation Errors due to Signature Changes

- Parameter Addition
- Parameter Type Change
- Parameter Removal
- Return Type Change

Repair Signature Changes Applicability

Subject	# versions
Cassandra	19
Geronimo	4
POI	6
Santuario	4
Shindig	6
Velocity	15
Wicket	7
Xindice	9
XMLBeans	7

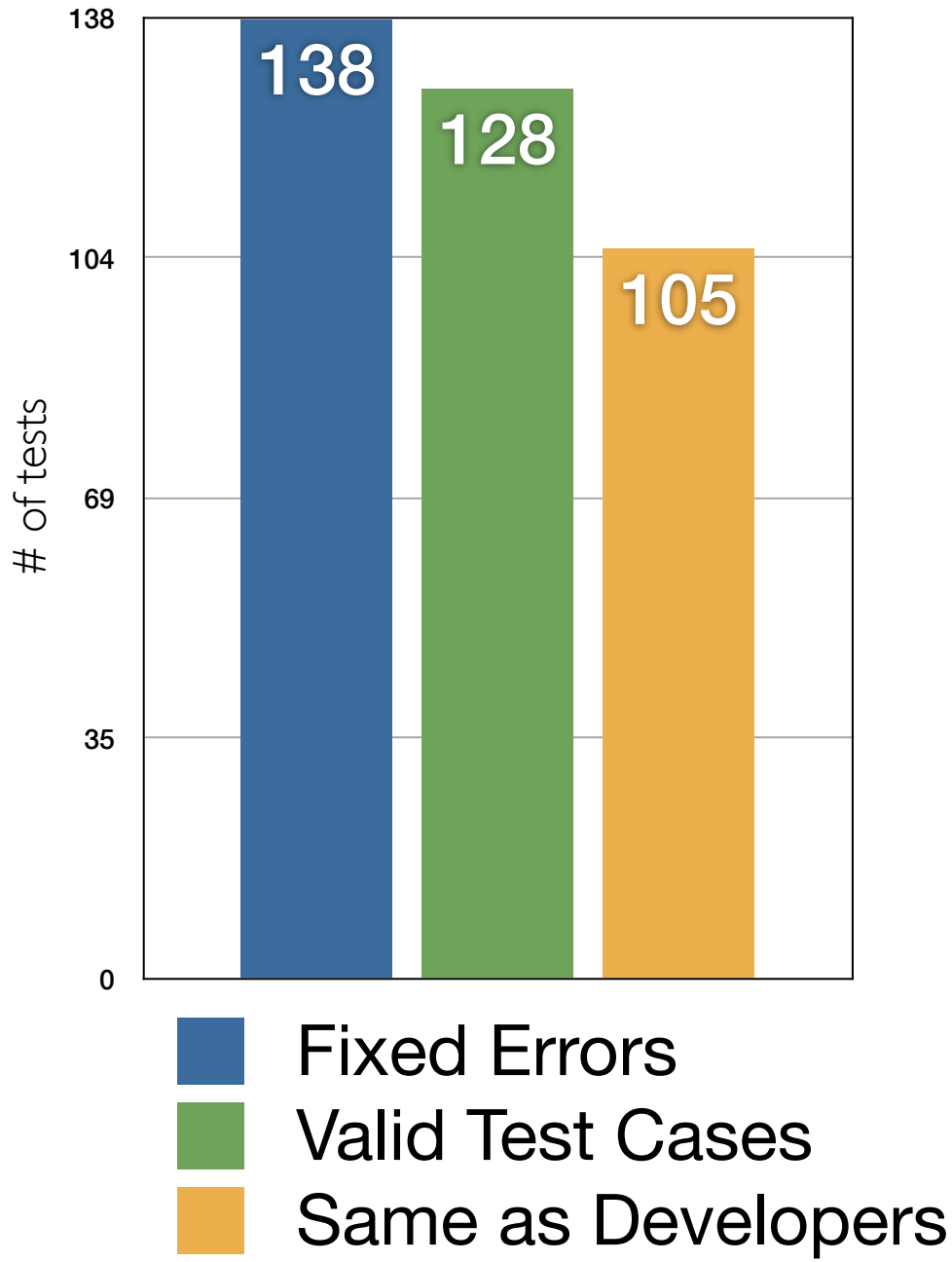
Repair Signature Changes Applicability



Repair Signature Changes Effectiveness

Project	# Versions	# Tests
PMD	13	**99**
JodaTime	2	**19**
JFreeChart	6	**21**
	21	**138**

Repair Signature Changes Effectiveness



Test Care Assistant

Add Test Cases for new Modules

- new Class in Hierarchy
- new Interface Implementation
- New Overloaded Method
- New Overridden Method

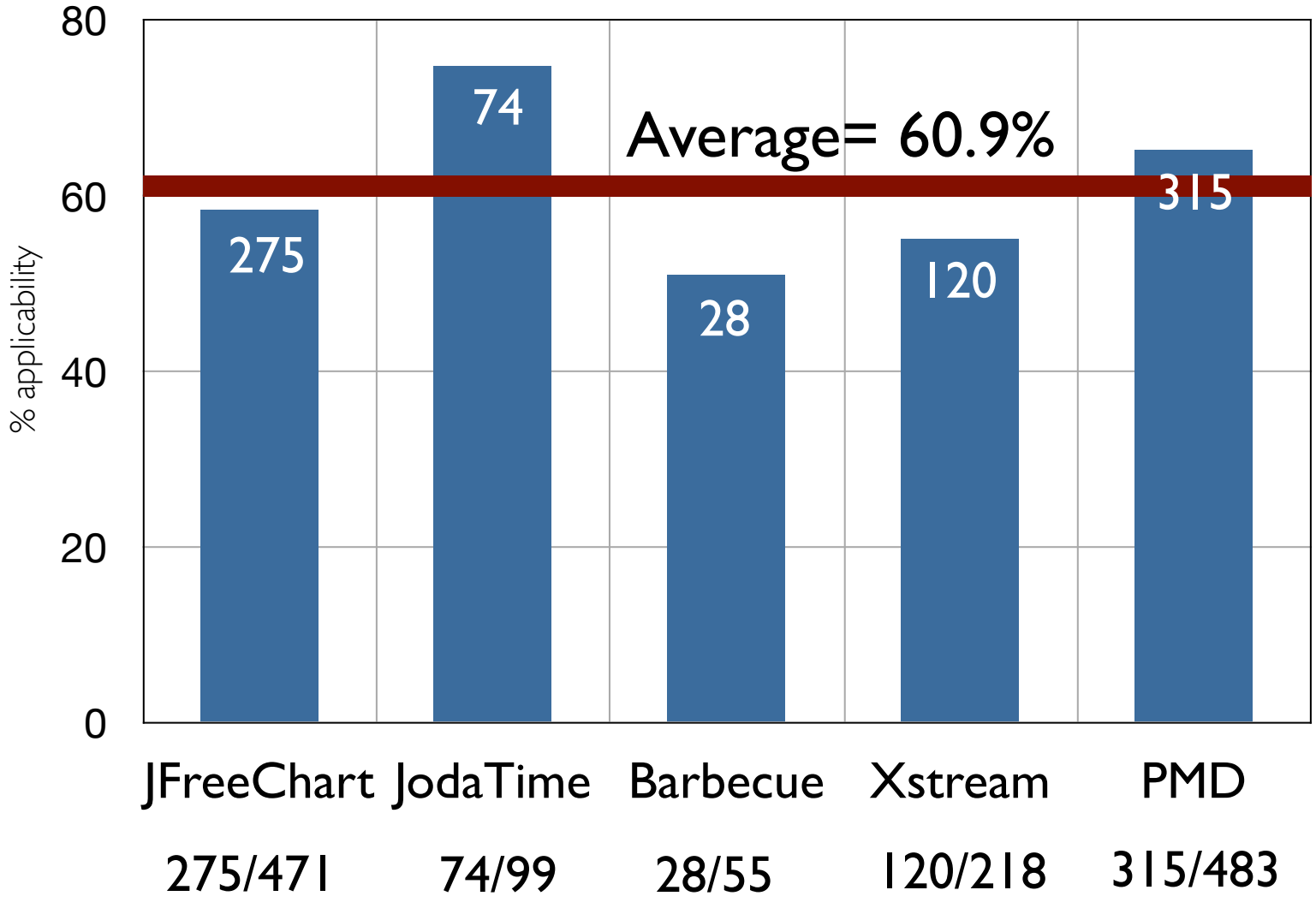
Repair Compilation Errors in Signature Changes

- Parameter Addition
- Parameter Type Change
- Parameter Removal
- Return Type Change

Applicability

Software	version	KLOC	classes
JFreeChart	0.96	217	471
JodaTime	1.62	64	99
Barbecue	1.5	9	55
Xstream	1.31	25	218
PMD	4.2	65	483

Applicability



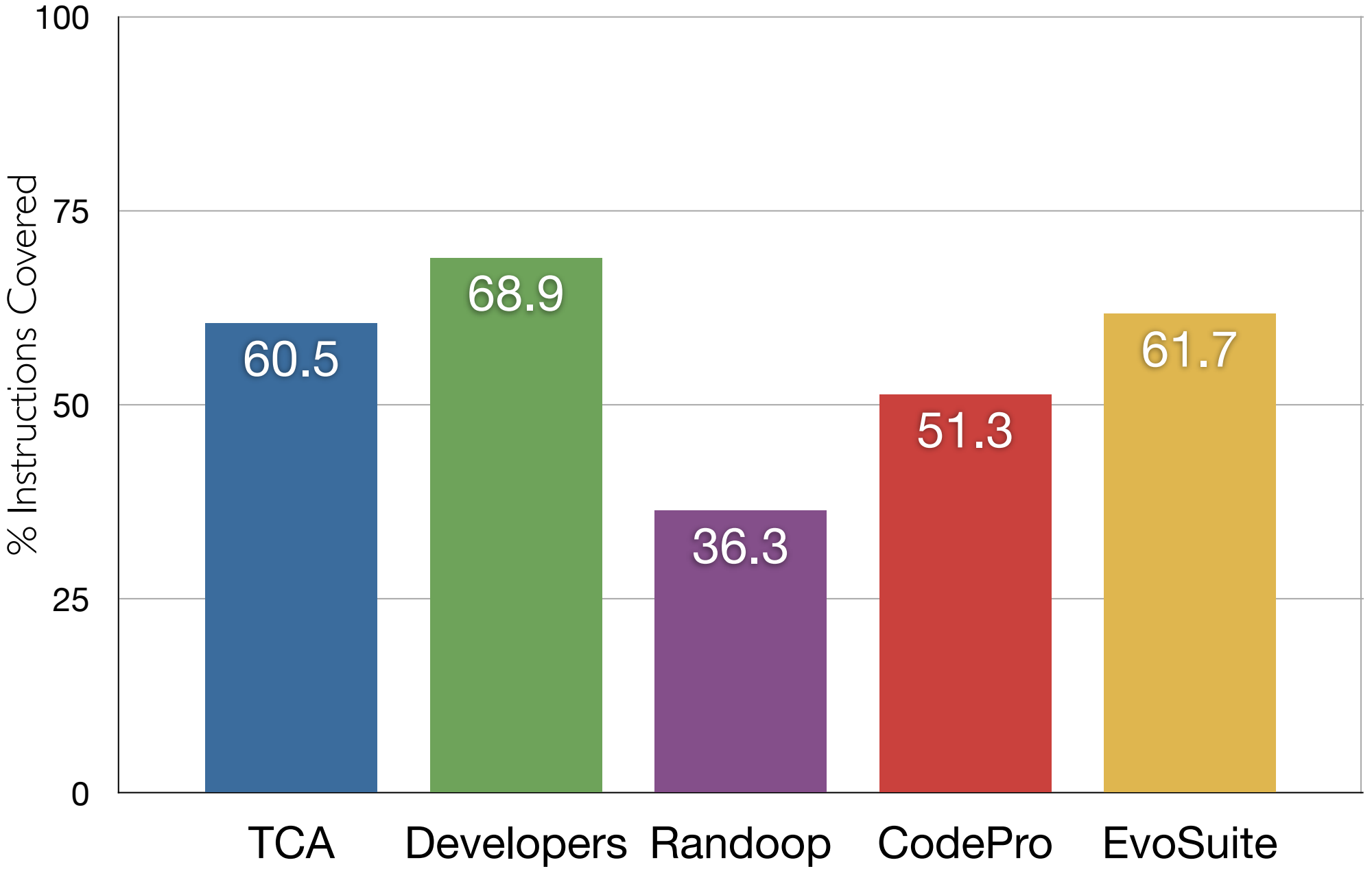
% classes for which TCA can generate tests

Effectiveness

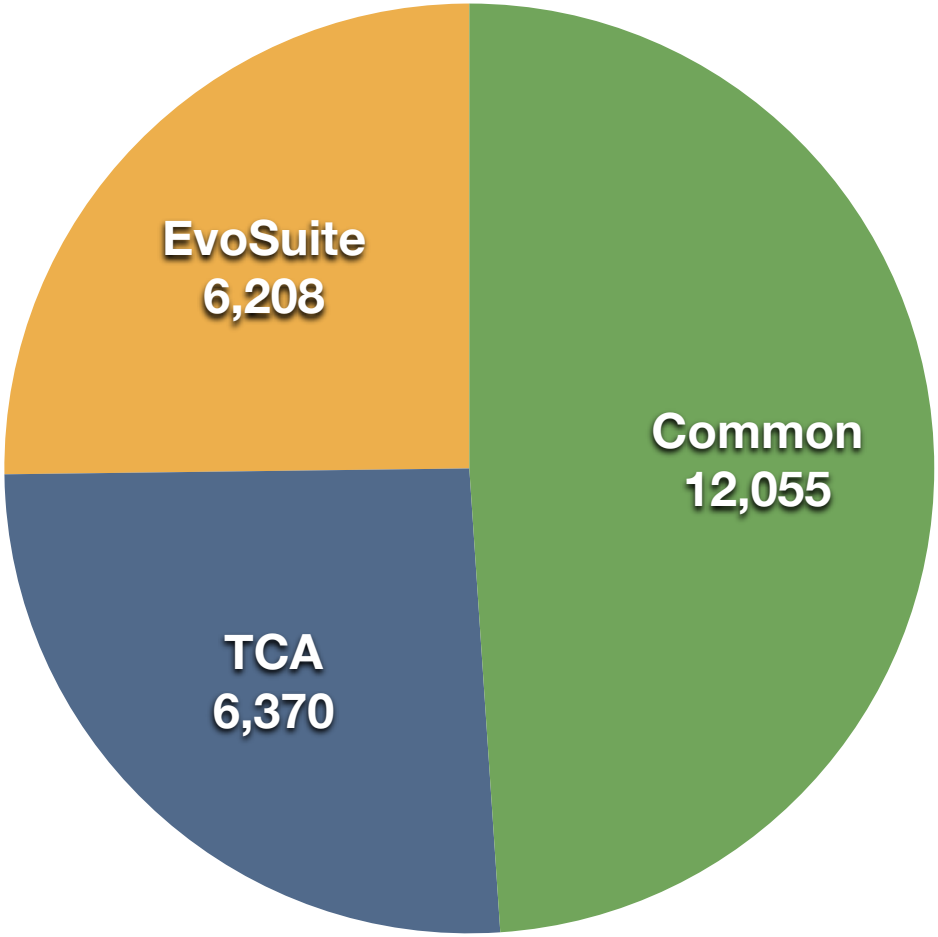
TCA	Developer's	Test Generation Tools
 Randoop	Google CodePro AnalytiX™ 	
• random test generation • run for 50 sec on each class	• industrial, coverage based • run for each classes	• search-based testing • run with assertion generation

Subjects	classes
Xstream 1.31	36
PMD 4.2	56
Barbecue 1.5b1	12
JodaTime 1.62	32
JfreeChart 1.013	204
Total	**340**

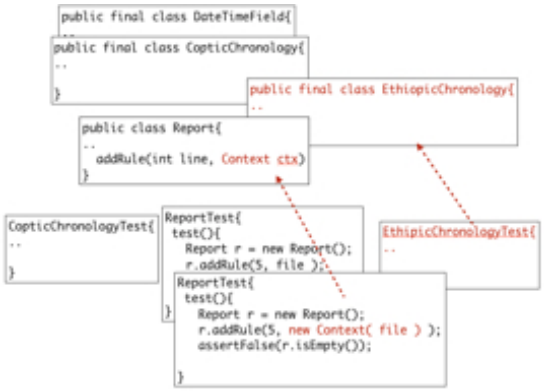
Coverage Results



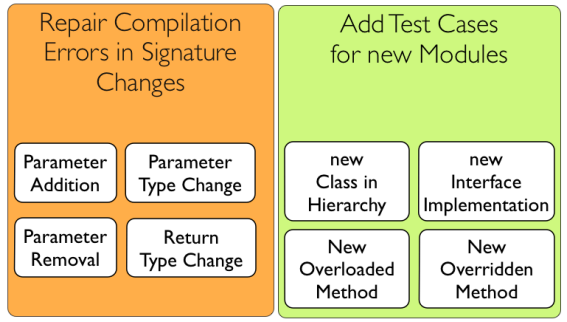
TCA and Evosuite are Complementary



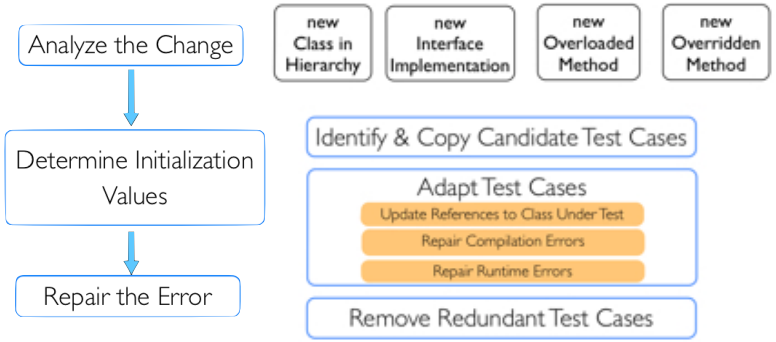
Covered Statements



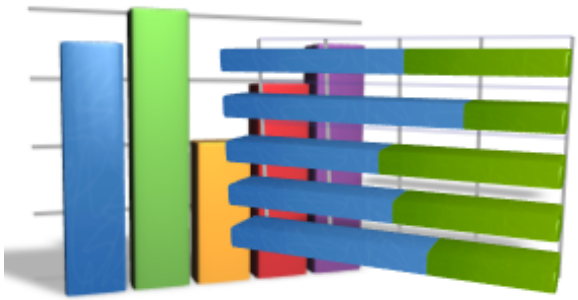
Test Evolution



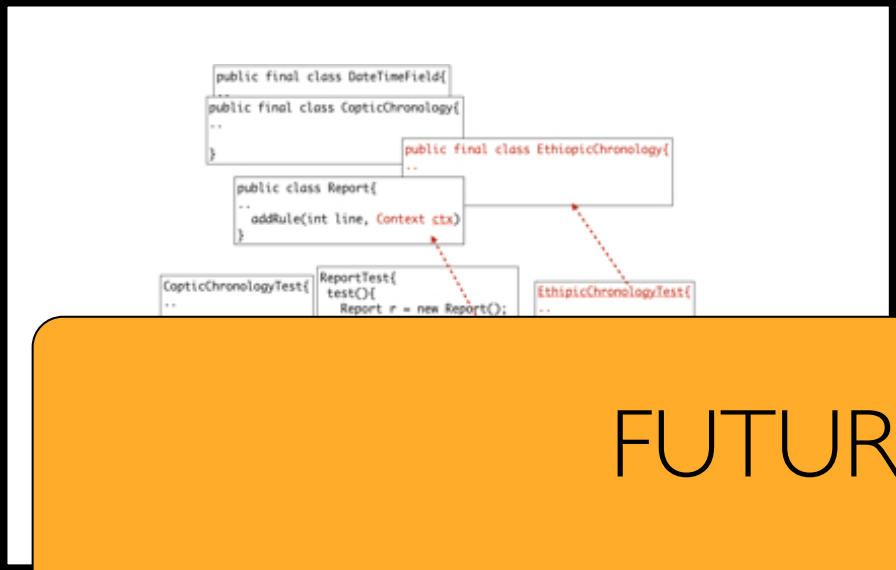
Test Reuse Algorithms

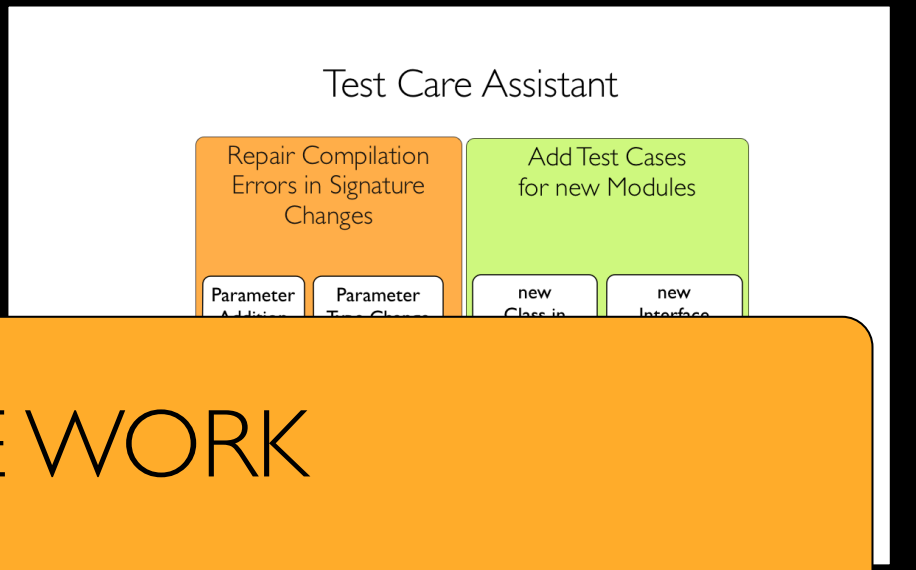


Test Care Assistant



Evaluation





FUTURE WORK

- Evaluate other Test Reuse Algorithms
- User study on readability of generated tests

Test Care Assistant

Evaluation