

SWE 261P Software Testing and Analysis - Part 3 Report

PDFsam Basic: White Box Testing and Coverage

 JAVA 21

5 JUNIT 5

 MAVEN

JACOCO 0.8.12

Course SWE 261P

Quarter Winter 2026

Team Kingson & Zian & Zhenyu

Repo Github Link:

<https://github.com/eric-song-dev/pdfsam>

Team Members:

- Kingson Zhang: kxzhang@uci.edu
- Zian Xu: zianx11@uci.edu
- Zhenyu Song: zhenyus4@uci.edu

This report documents the **structural (white-box) testing** process of **PDFsam Basic**, covering JaCoCo code coverage configuration, baseline measurement, and targeted test improvements across three non-GUI modules.

Quick Navigation

▼ SWE 261P Software Testing and Analysis - Part 3 Report

- PDFsam Basic: White Box Testing and Coverage

-  Quick Navigation

▼ 1. Structural Testing: Definition and Importance

- 1.1 What Is Structural Testing?
- 1.2 Why Structural Testing Matters
- 1.3 Coverage Metrics
- 1.4 Tool: JaCoCo

▼ 2. Existing Test Suite Coverage — Baseline

- 2.1 Running Baseline Coverage
- 2.2 Modules Under Test

▼ 2.3 Baseline Coverage Data

- pdfsam-model
- pdfsam-core
- pdfsam-persistence

▼ 3. Zhenyu's White Box Testing: pdfsam-model

- 3.1 Target Classes
- ▼ 3.2 Test Implementation Examples

- PdfDocumentDescriptor Tests
 - ToolDescriptor Tests
 - Comboltem Tests
 - Record & Event Validation Tests
- 3.3 Coverage Improvement

▼ 4. Kingson's White Box Testing: pdfsam-core

- 4.1 Target Classes
 - ▼ 4.2 Test Implementation
- ConversionUtils Edge Cases
 - ApplicationRuntimeState Tests
- 4.3 Coverage Improvement

▼ 5. Zian's White Box Testing: pdfsam-persistence

- 5.1 Target Classes
 - ▼ 5.2 Test Implementation
- PersistenceException Tests
 - DefaultEntityRepository Primitive Accessors
 - Repository Tests
 - PreferencesRepository Tests
- 5.3 Coverage Improvement

▼ 6. Test Implementation Summary

- 6.1 New Test Files
- 6.2 Running the White Box Tests

- 6.3 Test Results
- ▼  7. Conclusion
- Key Takeaways

1. Structural Testing: Definition and Importance

1.1 What Is Structural Testing?

Structural testing (white-box testing) is a software testing technique that uses knowledge of a program's internal structure—its source code, control flow, and data flow—to design test cases. Unlike black-box testing, which treats the system as an opaque entity, structural testing examines the *implementation* to systematically exercise different code paths, branches, and conditions.

1.2 Why Structural Testing Matters

Benefit	Description
Reveals Hidden Defects	Tests derived from implementation details uncover bugs in error-handling paths, boundary conditions, and rarely-executed branches
Coverage Measurement	Tools like JaCoCo quantify how much code is exercised (line, branch, method coverage)
Complementary	Fills gaps left by specification-based (black-box) tests
Regression Safety	High structural coverage provides confidence that future changes don't introduce regressions

1.3 Coverage Metrics

Common structural coverage criteria include:

- **Line Coverage:** Percentage of executable statements executed
- **Branch Coverage:** Percentage of decision outcomes (true/false) evaluated
- **Method Coverage:** Percentage of methods invoked
- **Instruction Coverage:** Percentage of bytecode instructions executed

1.4 Tool: JaCoCo

In order to see the test coverage of the current tests in the application, we used a tool called Java Code Coverage (JaCoCo). This plugin can be installed by including it in the [pom.xml file](#).

```
<plugin>
  <groupId>org.jacoco</groupId>
  <artifactId>jacoco-maven-plugin</artifactId>
  <version>0.8.12</version>
  <executions>
    <execution>
      <id>prepare-agent</id>
      <goals><goal>prepare-agent</goal></goals>
    </execution>
    <execution>
      <id>report</id>
      <phase>test</phase>
      <goals><goal>report</goal></goals>
      <configuration>
        <formats>
          <format>CSV</format>
          <format>XML</format>
        </formats>
      </configuration>
    </execution>
  </executions>
</plugin>
```



2. Existing Test Suite Coverage — Baseline

We ran `mvn clean test` with JaCoCo on three non-GUI modules **before** adding any new tests to establish a baseline.

2.1 Running Baseline Coverage

```
# Run tests with JaCoCo
mvn clean test jacoco:report -pl pdfsam-model,pdfsam-core,pdfsam-persistence -am '-Dtest=!ZhenyuWhiteBoxTest,'

# Create backup directory
mkdir -p saved-reports/baseline/model saved-reports/baseline/core saved-reports/baseline/persistence

# Copy the report to the backup directory
cp -r pdfsam-model/target/site/jacoco saved-reports/baseline/model
cp -r pdfsam-core/target/site/jacoco saved-reports/baseline/core
cp -r pdfsam-persistence/target/site/jacoco saved-reports/baseline/persistence

# View CSV reports
cat saved-reports/baseline/model/jacoco/jacoco.csv
cat saved-reports/baseline/core/jacoco/jacoco.csv
cat saved-reports/baseline/persistence/jacoco/jacoco.csv

# View HTML reports
open saved-reports/baseline/model/jacoco/index.html
open saved-reports/baseline/core/jacoco/index.html
open saved-reports/baseline/persistence/jacoco/index.html
```

2.2 Modules Under Test

Module	Description	Existing Tests
pdfsam-model	Domain model classes	<input checked="" type="checkbox"/>
pdfsam-core	Core utilities	<input checked="" type="checkbox"/>
pdfsam-persistence	Data persistence layer	<input checked="" type="checkbox"/>

2.3 Baseline Coverage Data

pdfsam-model

Class	Lines Missed	Lines Covered	Line Coverage	Branches Missed	Branches Covered	Branch Coverage
PdfDocumentDescriptor	13	36	73%	3	3	50%

Class	Lines Missed	Lines Covered	Line Coverage	Branches Missed	Branches Covered	Branch Coverage
PdfDescriptorLoadingStatus	0	33	100%	0	2	100%
ToolDescriptorBuilder	4	15	79%	0	0	—
ToolDescriptor	1	14	93%	2	0	0%
PdfRotationInput	8	13	62%	4	2	33%
BulkRotateParameters	6	13	68%	0	4	100%
Workspace	1	12	92%	0	8	100%
ToolCategory	2	10	83%	0	0	—
ToolPriority	0	8	100%	0	0	—
ObservableAtomicReference	0	8	100%	0	0	—
BaseToolBound	1	4	80%	0	0	—
TitleRequest	3	0	0%	0	0	—
NonExistingOutputDirectoryEvent	3	0	0%	0	0	—
NewsData	3	0	0%	0	0	—
ComboItem	15	0	0%	4	0	0%
FileType	14	0	0%	0	0	—
LatestNewsResponse	2	1	33%	0	0	—

To extract line coverage from the CSV report:

```
$ awk -F, 'NR>1 {printf "%-45s line_miss=%s line_cov=%s br_miss=%s br_cov=%s\n", $3, $8, $9, $6, $7}' saved-report.csv | sort
```

	line_miss	line_cov	br_miss	br_cov
ComboItem	15	0	4	0
FileType	14	0	0	0
PdfDocumentDescriptor	13	36	3	3
PdfRotationInput	8	13	4	2
BulkRotateParameters	6	13	0	4
DefaultPdfVersionComboItem	5	0	2	0
ToolDescriptorBuilder	4	15	0	0
ToolIdNamePair	3	0	0	0
StageMode	3	3	2	0
TitleRequest	3	0	0	0
RequiredPdfData	3	0	0	0
OpenType	3	0	0	0
NonExistingOutputDirectoryEvent	3	0	0	0
NewsData	3	0	0	0
FilesDroppedEvent	3	0	0	0
ClearToolRequest	3	0	0	0
WorkspaceLoadedEvent	2	1	0	0
ToolCategory	2	10	0	0
ShowPdfDescriptorRequest	2	1	0	0
SetLatestStageStatusRequest	2	1	0	0
SetActiveContentItemRequest	2	1	0	0
RemovePdfVersionConstraintEvent	2	1	0	0
PremiumToolsResponse	2	1	0	0
PremiumProduct	2	8	0	0
NewImportantNewsEvent	2	1	0	0
LoadWorkspaceRequest	2	1	0	0
LatestNewsResponse	2	1	0	0
AddPdfVersionConstraintEvent	2	1	0	0
WorkspaceCloseEvent	1	0	0	0
Workspace	1	12	0	8
UpdateCheckRequest	1	0	0	0
ToolDescriptor	1	14	2	0
Tool	1	0	0	0
ToggleNewsPanelRequest	1	0	0	0
StartupEvent	1	0	0	0
ShutdownEvent	1	0	0	0
ShowStageRequest	1	0	0	0
ShowLogMessagesRequest	1	0	0	0
SaveLogRequest	1	0	0	0
NoUpdateAvailable	1	0	0	0
HideStageRequest	1	0	0	0
HideNewsPanelRequest	1	0	0	0
FetchPremiumModulesRequest	1	0	0	0
FetchLatestNewsRequest	1	0	0	0
ContentItem	1	0	0	0
ConfirmSaveWorkspaceRequest	1	0	0	0
ClearLogRequest	1	0	0	0

```

CleanupRequest           line_miss=1 line_cov=0 br_miss=0 br_cov=0
ChangedSelectedPdfVersionEvent line_miss=1 line_cov=0 br_miss=0 br_cov=0
BaseToolBound            line_miss=1 line_cov=4 br_miss=0 br_cov=0
UpdateAvailableEvent     line_miss=0 line_cov=3 br_miss=0 br_cov=0
ToolPriority              line_miss=0 line_cov=8 br_miss=0 br_cov=0
ToolInputOutputType       line_miss=0 line_cov=4 br_miss=0 br_cov=0
TaskExecutionRequest      line_miss=0 line_cov=4 br_miss=0 br_cov=0
StageStatus                line_miss=0 line_cov=6 br_miss=0 br_cov=0
StageMode.new StageMode() {...} line_miss=0 line_cov=3 br_miss=0 br_cov=0
StageMode.new StageMode() {...} line_miss=0 line_cov=2 br_miss=0 br_cov=0
SetDestinationRequest    line_miss=0 line_cov=5 br_miss=0 br_cov=0
SaveWorkspaceRequest      line_miss=0 line_cov=8 br_miss=0 br_cov=0
PremiumTool                line_miss=0 line_cov=6 br_miss=0 br_cov=0
PdfLoadRequest             line_miss=0 line_cov=5 br_miss=0 br_cov=0
PdfFilesListLoadRequest   line_miss=0 line_cov=4 br_miss=0 br_cov=0
PdfDescriptorLoadingStatus line_miss=0 line_cov=33 br_miss=0 br_cov=2
ObservableAtomicReference line_miss=0 line_cov=8 br_miss=0 br_cov=0
NativeOpenUrlRequest      line_miss=0 line_cov=3 br_miss=0 br_cov=0
NativeOpenFileRequest     line_miss=0 line_cov=3 br_miss=0 br_cov=0
LoadWorkspaceResponse      line_miss=0 line_cov=6 br_miss=0 br_cov=0
InputPdfArgumentsLoadRequest line_miss=0 line_cov=2 br_miss=0 br_cov=2

```

pdfsam-core

Class	Lines Missed	Lines Covered	Line Coverage	Branches Missed	Branches Covered	Branch Coverage
ApplicationRuntimeState	17	27	61%	2	4	67%
ConversionUtils	8	29	78%	3	13	81%

To extract line coverage from the CSV report:

```
$ awk -F, 'NR>1 {printf "%-45s line_miss=%s line_cov=%s br_miss=%s br_cov=%s\n", $3, $8, $9, $6, $7}' saved-report.csv | sort
```

FileChooserWithWorkingDirectory		line_miss=35	line_cov=0	br_miss=10	br_cov=0
BrandableProperty		line_miss=29	line_cov=0	br_miss=0	br_cov=0
ApplicationContext		line_miss=29	line_cov=28	br_miss=5	br_cov=1
ApplicationRuntimeState		line_miss=17	line_cov=27	br_miss=2	br_cov=4
DirectoryChooserWithWorkingDirectory		line_miss=15	line_cov=0	br_miss=4	br_cov=0
MultiplePdfSourceMultipleOutputParametersBuilder	line_miss=12	line_cov=0	br_miss=2	br_cov=0	
AbstractPdfOutputParametersBuilder		line_miss=11	line_cov=0	br_miss=0	br_cov=0
SinglePdfSourceMultipleOutputParametersBuilder	line_miss=10	line_cov=0	br_miss=0	br_cov=0	
ConversionUtils		line_miss=8	line_cov=29	br_miss=3	br_cov=13
SplitParametersBuilder		line_miss=7	line_cov=0	br_miss=2	br_cov=0
Choosers		line_miss=6	line_cov=0	br_miss=0	br_cov=0
AbstractParametersBuilder		line_miss=5	line_cov=0	br_miss=0	br_cov=0
ObjectCollectionWriter		line_miss=4	line_cov=21	br_miss=0	br_cov=4
EncryptionUtils		line_miss=4	line_cov=15	br_miss=0	br_cov=4
ApplicationPersistentSettings		line_miss=2	line_cov=71	br_miss=3	br_cov=7
Choosers.FileChooserHolder		line_miss=1	line_cov=0	br_miss=0	br_cov=0
Choosers.DirectoryChooserHolder		line_miss=1	line_cov=0	br_miss=0	br_cov=0
XmlUtils		line_miss=0	line_cov=5	br_miss=0	br_cov=4
Validators		line_miss=0	line_cov=16	br_miss=0	br_cov=8
StringPersistentProperty		line_miss=0	line_cov=20	br_miss=0	br_cov=0
RegexValidator		line_miss=0	line_cov=5	br_miss=0	br_cov=4
PositiveIntRangeStringValidator		line_miss=0	line_cov=11	br_miss=2	br_cov=8
PositiveIntegerValidator		line_miss=0	line_cov=2	br_miss=0	br_cov=4
PositiveIntegerStringValidator		line_miss=0	line_cov=4	br_miss=0	br_cov=2
PersistentPropertyChange		line_miss=0	line_cov=1	br_miss=0	br_cov=0
IntegerPersistentProperty		line_miss=0	line_cov=7	br_miss=0	br_cov=0
FileValidator		line_miss=0	line_cov=2	br_miss=0	br_cov=4
FileTypeValidator		line_miss=0	line_cov=7	br_miss=1	br_cov=5
ContainedIntegerValidator		line_miss=0	line_cov=7	br_miss=0	br_cov=0
BooleanPersistentProperty		line_miss=0	line_cov=22	br_miss=0	br_cov=0

pdfsam-persistence

Class	Lines Missed	Lines Covered	Line Coverage	Branches Missed	Branches Covered	Branch Coverage
PreferencesRepository	16	49	75%	0	2	100%
DefaultEntityRepository	19	23	55%	0	4	100%
PersistenceException	4	2	33%	0	0	—
Repository	2	2	50%	0	0	—

To extract line coverage from the CSV report:

```
$ awk -F, 'NR>1 {printf "%-45s line_miss=%s line_cov=%s br_miss=%s br_cov=%s\n", $3, $8, $9, $6, $7}' saved-report.csv
```

DefaultEntityRepository	line_miss=19	line_cov=23	br_miss=0	br_cov=4
PreferencesRepository	line_miss=16	line_cov=49	br_miss=0	br_cov=2
PersistenceException	line_miss=4	line_cov=2	br_miss=0	br_cov=0
Repository	line_miss=2	line_cov=2	br_miss=0	br_cov=0

⭐ 3. Zhenyu's White Box Testing: pdfsam-model

Test File: [pdfsam-model/src/test/java/org/pdfsam/model/ZhenyuWhiteBoxTest.java](#)

3.1 Target Classes

Class	Package
PdfDocumentDescriptor	org.pdfsam.model.pdf
ToolDescriptorBuilder	org.pdfsam.model.tool
ToolDescriptor	org.pdfsam.model.tool
ToolCategory	org.pdfsam.model.tool
ObservableAtomicReference	org.pdfsam.model
BaseToolBound	org.pdfsam.model.tool
TitleRequest	org.pdfsam.model.ui
NonExistingOutputDirectoryEvent	org.pdfsam.model.ui
NewsData	org.pdfsam.model.news
LatestNewsResponse	org.pdfsam.model.news
ComboItem	org.pdfsam.model.ui
ClearToolRequest	org.pdfsam.model.tool
FilesDroppedEvent	org.pdfsam.model.ui.dnd
OpenType	org.pdfsam.model.io
RequiredPdfData	org.pdfsam.model.tool

3.2 Test Implementation Examples

44 test methods targeting 20+ classes with previously uncovered code paths.

PdfDocumentDescriptor Tests

```
@Test
@DisplayName("hasPassword returns true when password set")
void descriptorHasPassword() {
    var desc = PdfDocumentDescriptor.newDescriptor(mockFile, "secret");
    assertTrue(desc.hasPassword());
}

@Test
@DisplayName("setVersion and getVersion")
void descriptorVersion() {
    var desc = PdfDocumentDescriptor.newDescriptorNoPassword(mockFile);
    desc.setVersion(PdfVersion.VERSION_1_7);
    assertEquals(PdfVersion.VERSION_1_7, desc.getVersion());
}
```

ToolDescriptor Tests

```
@Test
@DisplayName("hasInputType returns false for missing type")
void toolDescriptorHasInputTypeFalse() {
    ToolDescriptor td = ToolDescriptorBuilder.builder()
        .name("test").description("desc")
        .category(ToolCategory.SPLIT)
        .inputTypes(ToolInputOutputType.SINGLE_PDF)
        .build();
    assertFalse(td.hasInputType(ToolInputOutputType.MULTIPLE_PDF));
}

@Test
@DisplayName("supportURL")
void toolDescriptorSupportURL() {
    ToolDescriptor td = ToolDescriptorBuilder.builder()
        .name("test").description("desc")
        .category(ToolCategory.MERGE)
        .supportURL("https://example.com")
        .build();
    assertEquals("https://example.com", td.supportUrl());
}
```

ComboItem Tests

```
@Test
@DisplayName("equals with same reference")
void comboItemEqualsSameRef() {
    ComboItem<String> item = new ComboItem<>("key1", "desc");
    assertEquals(item, item);
}

@Test
@DisplayName("equals with different type")
void comboItemEqualsDiffType() {
    ComboItem<String> item = new ComboItem<>("key1", "desc");
    assertNotEquals(item, "not a ComboItem");
}

@Test
@DisplayName("equals with same key")
void comboItemEqualsSameKey() {
    ComboItem<String> a = new ComboItem<>("key1", "desc1");
    ComboItem<String> b = new ComboItem<>("key1", "desc2");
    assertEquals(a, b);
}
```

Record & Event Validation Tests

```
@Test
@DisplayName("valid construction")
void filesDroppedEventValid() {
    File f = new File("test.pdf");
    FilesDroppedEvent evt = new FilesDroppedEvent("toolId", true, List.of(f));
    assertEquals("toolId", evt.toolBinding());
    assertTrue(evt.acceptMultipleFiles());
    assertEquals(1, evt.files().size());
}
```

3.3 Coverage Improvement

Class	Before Lines	After Lines	Δ Lines	Before Branches	After Branches	Δ Branches
PdfDocumentDescriptor	36	44	+8	3	3	—
ToolDescriptorBuilder	15	19	+4	0	0	—
ToolDescriptor	14	15	+1	0	1	+1
ToolCategory	10	12	+2	0	0	—

Class	Before Lines	After Lines	Δ Lines	Before Branches	After Branches	Δ Branches
BaseToolBound	4	5	+1	0	0	—
TitleRequest	0	3	+3	0	0	—
NonExistingOutputDirectoryEvent	0	3	+3	0	0	—
NewsData	0	3	+3	0	0	—
ComboItem	0	15	+15	0	4	+4
LatestNewsResponse	1	3	+2	0	0	—
ClearToolRequest	0	3	+3	0	0	—
FilesDroppedEvent	0	3	+3	0	0	—
OpenType	0	3	+3	0	0	—
RequiredPdfData	0	3	+3	0	0	—
ChangedSelectedPdfVersionEvent	0	1	+1	0	0	—
UpdateCheckRequest	0	1	+1	0	0	—
Total			+56			+5

```
$ awk -F, 'NR>1 {printf "%-45s line_miss=%s line_cov=%s br_miss=%s br_cov=%s\n", $3, $8, $9, $6, $7}' pdfsam-1

FileType line_miss=14 line_cov=0 br_miss=0 br_cov=0
PdfRotationInput line_miss=8 line_cov=13 br_miss=4 br_cov=2
BulkRotateParameters line_miss=6 line_cov=13 br_miss=0 br_cov=4
PdfDocumentDescriptor line_miss=5 line_cov=44 br_miss=3 br_cov=3
DefaultPdfVersionComboItem line_miss=5 line_cov=0 br_miss=2 br_cov=0
ToolIdNamePair line_miss=3 line_cov=0 br_miss=0 br_cov=0
StageMode line_miss=3 line_cov=3 br_miss=2 br_cov=0
WorkspaceLoadedEvent line_miss=2 line_cov=1 br_miss=0 br_cov=0
ShowPdfDescriptorRequest line_miss=2 line_cov=1 br_miss=0 br_cov=0
SetLatestStageStatusRequest line_miss=2 line_cov=1 br_miss=0 br_cov=0
SetActiveContentItemRequest line_miss=2 line_cov=1 br_miss=0 br_cov=0
RemovePdfVersionConstraintEvent line_miss=2 line_cov=1 br_miss=0 br_cov=0
PremiumToolsResponse line_miss=2 line_cov=1 br_miss=0 br_cov=0
PremiumProduct line_miss=2 line_cov=8 br_miss=0 br_cov=0
NewImportantNewsEvent line_miss=2 line_cov=1 br_miss=0 br_cov=0
LoadWorkspaceRequest line_miss=2 line_cov=1 br_miss=0 br_cov=0
AddPdfVersionConstraintEvent line_miss=2 line_cov=1 br_miss=0 br_cov=0
Workspace line_miss=1 line_cov=12 br_miss=0 br_cov=8
Tool line_miss=1 line_cov=0 br_miss=0 br_cov=0
FetchPremiumModulesRequest line_miss=1 line_cov=0 br_miss=0 br_cov=0
ContentItem line_miss=1 line_cov=0 br_miss=0 br_cov=0
WorkspaceCloseEvent line_miss=0 line_cov=1 br_miss=0 br_cov=0
UpdateCheckRequest line_miss=0 line_cov=1 br_miss=0 br_cov=0
UpdateAvailableEvent line_miss=0 line_cov=3 br_miss=0 br_cov=0
ToolPriority line_miss=0 line_cov=8 br_miss=0 br_cov=0
ToolInputOutputType line_miss=0 line_cov=4 br_miss=0 br_cov=0
ToolDescriptorBuilder line_miss=0 line_cov=19 br_miss=0 br_cov=0
ToolDescriptor line_miss=0 line_cov=15 br_miss=1 br_cov=1
ToolCategory line_miss=0 line_cov=12 br_miss=0 br_cov=0
ToggleNewsPanelRequest line_miss=0 line_cov=1 br_miss=0 br_cov=0
TaskExecutionRequest line_miss=0 line_cov=4 br_miss=0 br_cov=0
StartupEvent line_miss=0 line_cov=1 br_miss=0 br_cov=0
StageStatus line_miss=0 line_cov=6 br_miss=0 br_cov=0
StageMode.new StageMode() {...}
StageMode.new StageMode() {...}
ShutdownEvent line_miss=0 line_cov=1 br_miss=0 br_cov=0
ShowStageRequest line_miss=0 line_cov=1 br_miss=0 br_cov=0
ShowLogMessagesRequest line_miss=0 line_cov=1 br_miss=0 br_cov=0
SetTitleRequest line_miss=0 line_cov=3 br_miss=0 br_cov=0
SetDestinationRequest line_miss=0 line_cov=5 br_miss=0 br_cov=0
SaveWorkspaceRequest line_miss=0 line_cov=8 br_miss=0 br_cov=0
SaveLogRequest line_miss=0 line_cov=1 br_miss=0 br_cov=0
RequiredPdfData line_miss=0 line_cov=3 br_miss=0 br_cov=0
PremiumTool line_miss=0 line_cov=6 br_miss=0 br_cov=0
PdfLoadRequest line_miss=0 line_cov=5 br_miss=0 br_cov=0
PdfFilesListLoadRequest line_miss=0 line_cov=4 br_miss=0 br_cov=0
PdfDescriptorLoadingStatus line_miss=0 line_cov=33 br_miss=0 br_cov=2
```

OpenType	line_miss=0 line_cov=3 br_miss=0 br_cov=0
ObservableAtomicReference	line_miss=0 line_cov=8 br_miss=0 br_cov=0
NoUpdateAvailable	line_miss=0 line_cov=1 br_miss=0 br_cov=0
NonExistingOutputDirectoryEvent	line_miss=0 line_cov=3 br_miss=0 br_cov=0
NewsData	line_miss=0 line_cov=3 br_miss=0 br_cov=0
NativeOpenUrlRequest	line_miss=0 line_cov=3 br_miss=0 br_cov=0
NativeOpenFileRequest	line_miss=0 line_cov=3 br_miss=0 br_cov=0
LoadWorkspaceResponse	line_miss=0 line_cov=6 br_miss=0 br_cov=0
LatestNewsResponse	line_miss=0 line_cov=3 br_miss=0 br_cov=0
InputPdfArgumentsLoadRequest	line_miss=0 line_cov=2 br_miss=0 br_cov=2
HideStageRequest	line_miss=0 line_cov=1 br_miss=0 br_cov=0
HideNewsPanelRequest	line_miss=0 line_cov=1 br_miss=0 br_cov=0
FilesDroppedEvent	line_miss=0 line_cov=3 br_miss=0 br_cov=0
FetchLatestNewsRequest	line_miss=0 line_cov=1 br_miss=0 br_cov=0
ConfirmSaveWorkspaceRequest	line_miss=0 line_cov=1 br_miss=0 br_cov=0
ComboItem	line_miss=0 line_cov=15 br_miss=0 br_cov=4
ClearToolRequest	line_miss=0 line_cov=3 br_miss=0 br_cov=0
ClearLogRequest	line_miss=0 line_cov=1 br_miss=0 br_cov=0
CleanupRequest	line_miss=0 line_cov=1 br_miss=0 br_cov=0
ChangedSelectedPdfVersionEvent	line_miss=0 line_cov=1 br_miss=0 br_cov=0
BaseToolBound	line_miss=0 line_cov=5 br_miss=0 br_cov=0

✨ 4. Kingson's White Box Testing: pdfsam-core

Test File: [pdfsam-core/src/test/java/org/pdfsam/core/context/KingsonWhiteBoxTest.java](#)

4.1 Target Classes

Class	Package	Key Uncovered Code
ConversionUtils	org.pdfsam.core.support.params	Blank/null input, "last" keyword, open-ended ranges, prefixed dash
ApplicationRuntimeState	org.pdfsam.core.context	maybeWorkingPath null/blank/file/dir/string, activeToolValue , workspace , theme
ObjectCollectionWriter	org.pdfsam.core.support.io	writeContent().to(Path) with content and empty list

4.2 Test Implementation

22 test methods targeting boundary cases and uncovered branches.

ConversionUtils Edge Cases

```
@Test
void toPageRangeSetBlankReturnsEmpty() {
    var result = ConversionUtils.toPageRangeSet("");
    assertTrue(result.isEmpty());
}

@Test
void toPagesSelectionSetLastPage() {
    Set<PagesSelection> result = ConversionUtils.toPagesSelectionSet("last");
    assertEquals(1, result.size());
    assertTrue(result.contains(PagesSelection.LAST_PAGE));
}

@Test
void toPageRangeSetOpenEndedRange() {
    Set<PageRange> result = ConversionUtils.toPageRangeSet("5-");
    assertEquals(1, result.size());
    PageRange range = result.iterator().next();
    assertEquals(5, range.getStart());
    assertTrue(range.isUnbounded());
}
```

ApplicationRuntimeState Tests

```
@Test
void workingPathInitiallyEmpty() {
    ApplicationRuntimeState state = new ApplicationRuntimeState();
    assertEquals(Optional.empty(), state.workingPathValue());
}

@Test
void maybeWorkingPathWithValidDirectory(@TempDir Path tempDir) {
    ApplicationRuntimeState state = new ApplicationRuntimeState();
    state.maybeWorkingPath(tempDir);
    assertEquals(Optional.of(tempDir), state.workingPathValue());
}

@Test
void maybeWorkingPathWithRegularFile(@TempDir Path tempDir) throws IOException {
    Path file = Files.createTempFile(tempDir, "test", ".txt");
    ApplicationRuntimeState state = new ApplicationRuntimeState();
    state.maybeWorkingPath(file);
    // Should resolve to parent directory
    assertEquals(Optional.of(tempDir), state.workingPathValue());
}
```

4.3 Coverage Improvement

Class	Before Lines	After Lines	Δ Lines	Before Branches	After Branches	Δ Branches
ApplicationRuntimeState	27	33	+6	4	4	—
ConversionUtils	29	32	+3	13	16	+3
Total			+9			+3

```
$ awk -F, 'NR>1 {printf "%-45s line_miss=%s line_cov=%s br_miss=%s br_cov=%s\n", $3, $8, $9, $6, $7}' pdfsam-1

FileChooserWithWorkingDirectory           line_miss=35 line_cov=0 br_miss=10 br_cov=0
BrandableProperty                         line_miss=29 line_cov=0 br_miss=0 br_cov=0
ApplicationContext                        line_miss=29 line_cov=28 br_miss=5 br_cov=1
DirectoryChooserWithWorkingDirectory      line_miss=15 line_cov=0 br_miss=4 br_cov=0
MultiplePdfSourceMultipleOutputParametersBuilder line_miss=12 line_cov=0 br_miss=2 br_cov=0
ApplicationRuntimeState                   line_miss=11 line_cov=33 br_miss=2 br_cov=4
AbstractPdfOutputParametersBuilder       line_miss=11 line_cov=0 br_miss=0 br_cov=0
SinglePdfSourceMultipleOutputParametersBuilder line_miss=10 line_cov=0 br_miss=0 br_cov=0
SplitParametersBuilder                   line_miss=7 line_cov=0 br_miss=2 br_cov=0
Choosers                                line_miss=6 line_cov=0 br_miss=0 br_cov=0
ConversionUtils                          line_miss=5 line_cov=32 br_miss=0 br_cov=16
AbstractParametersBuilder                line_miss=5 line_cov=0 br_miss=0 br_cov=0
ObjectCollectionWriter                   line_miss=4 line_cov=21 br_miss=0 br_cov=4
EncryptionUtils                          line_miss=4 line_cov=15 br_miss=0 br_cov=4
ApplicationPersistentSettings           line_miss=2 line_cov=71 br_miss=3 br_cov=7
Choosers.FileChooserHolder              line_miss=1 line_cov=0 br_miss=0 br_cov=0
Choosers.DirectoryChooserHolder         line_miss=1 line_cov=0 br_miss=0 br_cov=0
XmlUtils                                line_miss=0 line_cov=5 br_miss=0 br_cov=4
Validators                              line_miss=0 line_cov=16 br_miss=0 br_cov=8
StringPersistentProperty                line_miss=0 line_cov=20 br_miss=0 br_cov=0
RegexValidator                          line_miss=0 line_cov=5 br_miss=0 br_cov=4
PositiveIntRangeStringValidator        line_miss=0 line_cov=11 br_miss=2 br_cov=8
PositiveIntegerValidator                line_miss=0 line_cov=2 br_miss=0 br_cov=4
PositiveIntegerStringValidator          line_miss=0 line_cov=4 br_miss=0 br_cov=2
PersistentPropertyChange               line_miss=0 line_cov=1 br_miss=0 br_cov=0
IntegerPersistentProperty              line_miss=0 line_cov=7 br_miss=0 br_cov=0
FileValidator                           line_miss=0 line_cov=2 br_miss=0 br_cov=4
FileTypeValidator                      line_miss=0 line_cov=7 br_miss=1 br_cov=5
ContainedIntegerValidator              line_miss=0 line_cov=7 br_miss=0 br_cov=0
BooleanPersistentProperty              line_miss=0 line_cov=22 br_miss=0 br_cov=0
```

⭐ 5. Zian's White Box Testing: pdfsam-persistence

Test File: [pdfsam-persistence/src/test/java/org/pdfsam/persistence/ZianWhiteBoxTest.java](#)

5.1 Target Classes

Class	Package	Key Uncovered Code
PersistenceException	org.pdfsam.persistence	Two constructors: message-only, cause-only
DefaultEntityRepository	org.pdfsam.persistence	Primitive type accessors: saveInt / getInt , saveLong / getLong , saveString / getString , saveBoolean / getBoolean , keys()
PreferencesRepository	org.pdfsam.persistence	saveInt / getInt , saveLong / getLong , saveBoolean / getBoolean , keys() , clean
Repository	org.pdfsam.persistence	getInt , getLong

5.2 Test Implementation

27 test methods were implemented to achieve comprehensive coverage, specifically targeting primitive types, interface default methods, and hard-to-reach exception handling blocks.

PersistenceException Tests

```
@Test
@DisplayName("PersistenceException: message-only constructor")
void persistenceExceptionMessageOnly() {
    PersistenceException ex = new PersistenceException("test error");
    assertEquals("test error", ex.getMessage());
    assertNull(ex.getCause());
}
```

```
@Test
@DisplayName("PersistenceException: cause-only constructor")
void persistenceExceptionCauseOnly() {
    Throwable cause = new RuntimeException("root cause");
    PersistenceException ex = new PersistenceException(cause);
    assertEquals(cause, ex.getCause());
}
```

DefaultEntityRepository Primitive Accessors

```
@Test
@DisplayName("DefaultEntityRepository: saveInt and getInt")
void entityRepoSaveGetInt() throws PersistenceException {
    entityRepo.saveInt("intKey", 42);
    assertEquals(42, entityRepo.getInt("intKey", 0));
}

@Test
@DisplayName("DefaultEntityRepository: getInt default value")
void entityRepoGetIntDefault() throws PersistenceException {
    entityRepo.delete("intMissing");
    assertEquals(-1, entityRepo.getInt("intMissing", -1));
}

@Test
@DisplayName("DefaultEntityRepository: saveLong and getLong")
void entityRepoSaveGetLong() throws PersistenceException {
    entityRepo.saveLong("longKey", 123456789L);
    assertEquals(123456789L, entityRepo.getLong("longKey", 0L));
}
```

Repository Tests

```
@Test
@DisplayName("Repository: getInt uses Supplier for default value")
void repositoryGetIntWithSupplier() throws PersistenceException {
    entityRepo.delete("intSuppMissing");

    assertEquals(100, entityRepo.getInt("intSuppMissing", () -> 100));
}

@Test
@DisplayName("Repository: getLong uses Supplier for default value")
void repositoryGetLongWithSupplier() throws PersistenceException {
    entityRepo.delete("longSuppMissing");

    assertEquals(999L, entityRepo.getLong("longSuppMissing", () -> 999L));
}
```

PreferencesRepository Tests

```
@Nested
class ZianExceptionCoverageTest {
    private final PreferencesRepository brokenRepo = new PreferencesRepository("/invalid//double/slash/path");

    @Test
    @DisplayName("getInt triggers PersistenceException on invalid path")
    void getIntException() {
        assertThrows(PersistenceException.class, () -> brokenRepo.getInt("anyKey", 0));
    }

    @Test
    @DisplayName("getLong triggers PersistenceException on invalid path")
    void getLongException() {
        assertThrows(PersistenceException.class, () -> brokenRepo.getLong("anyKey", 0L));
    }
}
```

5.3 Coverage Improvement

Class	Before Lines	After Lines	Δ Lines	Before Branches	After Branches	Δ Branches
DefaultEntityRepository	23	42	+19	4	4	—
PreferencesRepository	49	61	+12	2	2	—
PersistenceException	2	6	+4	0	0	—
Repository	2	4	+2	0	0	—
Total			+37			0

```
$ awk -F, 'NR>1 {printf "%-45s line_miss=%s line_cov=%s br_miss=%s br_cov=%s\n", $3, $8, $9, $6, $7}' pdfsam-1

PreferencesRepository                                line_miss=4 line_cov=61 br_miss=0 br_cov=2
Repository                                         line_miss=0 line_cov=4 br_miss=0 br_cov=0
PersistenceException                               line_miss=0 line_cov=6 br_miss=0 br_cov=0
DefaultEntityRepository                           line_miss=0 line_cov=42 br_miss=0 br_cov=4
```

6. Test Implementation Summary

6.1 New Test Files

File	Location	Author
ZhenyuWhiteBoxTest.java	pdfsam-model/src/test/java/org/pdfsam/model/	Zhenyu Song
KingsonWhiteBoxTest.java	pdfsam-core/src/test/java/org/pdfsam/core/context	Kingson Zhang
ZianWhiteBoxTest.java	pdfsam-persistence/src/test/java/org/pdfsam/persistence/	Zian Xu

6.2 Running the White Box Tests

```
# Run all white box tests together
mvn clean test jacoco:report -pl pdfsam-model,pdfsam-core,pdfsam-persistence -am

# Run individual test files
mvn test jacoco:report -pl pdfsam-model -Dtest=ZhenyuWhiteBoxTest
mvn test jacoco:report -pl pdfsam-core -Dtest=KingsonWhiteBoxTest
mvn test jacoco:report -pl pdfsam-persistence -Dtest=ZianWhiteBoxTest

# View CSV reports
cat pdfsam-model/target/site/jacoco/jacoco.csv
cat pdfsam-core/target/site/jacoco/jacoco.csv
cat pdfsam-persistence/target/site/jacoco/jacoco.csv

# View HTML reports
open pdfsam-model/target/site/jacoco/index.html
open pdfsam-core/target/site/jacoco/index.html
open pdfsam-persistence/target/site/jacoco/index.html
```

6.3 Test Results

```
$ mvn test jacoco:report -pl pdfsam-model -Dtest=ZhenyuWhiteBoxTest
Tests run: 44, Failures: 0, Errors: 0, Skipped: 0
BUILD SUCCESS
```

```
$ mvn test jacoco:report -pl pdfsam-core -Dtest=KingsonWhiteBoxTest
Tests run: 28, Failures: 0, Errors: 0, Skipped: 0
BUILD SUCCESS
```

```
$ mvn test jacoco:report -pl pdfsam-persistence -Dtest=ZianWhiteBoxTest
Tests run: 27, Failures: 0, Errors: 0, Skipped: 0
BUILD SUCCESS
```

7. Conclusion

This report documents our application of **structural (white-box) testing** to PDFsam Basic through three team members testing three non-GUI modules:

Team Member	Module	Δ New Tests	Δ Lines	Δ Branches
Zhenyu Song	pdflsaml-model	+44	+56	+5
Kingson Zhang	pdflsaml-core	+28	+9	+3
Zian Xu	pdflsaml-persistence	+27	+37	0
Total		+99	+102	+8

Key Takeaways

- **JaCoCo** provides systematic, automated coverage measurement in CSV/XML format
- **Structural testing** complements the partition testing (Part 1) and FSM testing (Part 2) by revealing untested implementation paths
- **Coverage-guided test design** helps prioritize which classes and branches need additional tests

The white-box tests complement the **partition testing** from Part 1 and **FSM testing** from Part 2, providing a third perspective on the same codebase and moving toward comprehensive test coverage.