WHO GUARDS THE GUARDS?
ON THE VALIDATION OF TEST CASE MIGRATION

Presented by Ivan Jovanovikj
Reuse of Test Cases in Migration Projects

Model-Driven Software Migration OMG ADM

User Conference on Advanced Automated Testing
Reuse of Test Cases in Migration Projects

In this presentation: How to validate the migrated test cases in migration scenario?
Validation of Test Case Migration

How to validate a test case migration?

What is a valid test case migration?

Test case migration is a process of transferring test cases into new environments without changing their functionality, i.e., without changing what they test.

Behavioral Equivalence

How to ensure behavioral equivalence in test case migration?
Validation of Test Case Migration

Test Cases → Migration → Migrated Test Cases → Execution → Test Report

- Passed Test Cases (negatives)
  - True negatives
  - False negatives
- Failed Test Cases (positives)
  - True positives
  - False positives

**Problem:** Detects non-existing errors  
**Consequence:** Waste of time by looking for non-existing error

**Problem:** Hides potential errors  
**Consequence:** System malfunctioning/crashes
Validation of Test Case Migration

How to ensure behavioral equivalence in test case migration?

How to avoid/detect false positives and false negatives?

Constructive approaches

Analytical Approaches

Mutation Testing

Cost of Test Case Migration + Cost of Validation of Test Case Migration <= Cost of Developing New Test Cases

Bottom Line
Validation of Test Case Migration

Code Refactoring
- Test Cases
  - Refactoring
  - System
    - System
      - Mutant 1
        - mutant killed
      - Mutant 2
        - mutant survived

Test Case Refactoring
- Refactoring
- Program
  - [Mutation Coverage] = [Mutation Coverage']
- Test Cases
  - Refactored Test Cases

System and Test Case Migration
- Migration
- System
  - Migrated System
- Test Cases
  - Migrated Test Cases

All Test Cases pre- and post-program refactoring should pass

Mutation Coverage = Killed Mutants
All Mutants - Equivalent Mutants
Mutation Analysis Scenarios

- **1.** Mutation of Old System
- **2.** Mutation of New System
- **3.** Mutation of System Migration
- **4.** Mutation of New Test Cases
- **5.** Mutation of Old Test Cases
- **6.** Mutation of Test Case Migration

**System and Test Case Migration**

- Test Cases
  -Mutation
  → Migrated Test Cases
- System
  -Migration
  → Migrated System

**Assumptions**

**Indications**
**Mutation Analysis Scenarios**

### System Mutation

1. **Mutation of Old System**
   - **Old Test Cases** → **New Test Cases**
     - **Old System** → **New System**
   - (1.a) **Old System Mutant** → **New System Mutant**
     - (1.b) **Old System Mutant** → **New System Mutant**

2. **Mutation of New System**
   - **Old Test Cases** → **New Test Cases**
     - **Old System** → **New System**
   - **Old System Mutant** → **New System Mutant**

3. **Mutation of System Migration**
   - **Old Test Cases** → **New Test Cases**
     - **Old System** → **New System**
   - **Old System Mutant** → **New System Mutant**

### Test Case Mutation

4. **Mutation of Old Test Cases**
   - **Old Test Case Mutant** → **New Test Case Mutant**
     - **Old Test Cases** → **New Test Cases**
     - **Old System** → **New System**

5. **Mutation of New Test Cases**
   - **Old Test Case Mutant** → **New Test Case Mutant**
     - **Old Test Cases** → **New Test Cases**
     - **Old System** → **New System**

6. **Mutation of Test Case Migration**
   - **Old Test Case Mutant** → **New Test Case Mutant**
     - **Old Test Cases** → **New Test Cases**
     - **Old System** → **New System**

---

**User Conference on Advanced Automated Testing**
1. **Mutation of Old System**

   - **Old Test Cases** → **New Test Cases**
   - **Old System** → **New System**
   - **Old System Mutant** → **New System Mutant**

   (1.a) Test Case Migration
   (1.b) System Migration

2. **Mutation of New System**

   - **Old Test Cases** → **New Test Cases**
   - **Old System** → **New System**
   - **Old System Mutant** → **New System Mutant**

   ![Diagram](image)

**Artefacts**
- Mutant (System or Test Case)
- Reverse engineered Mutant
- Migrated Mutant

**Activities**
- Mutant generation
- Mutant migration
- Reverse engineering of Mutant
- Test execution against system mutant
- Test execution against system
**Mutation Analysis Scenarios**

### Assumptions
- Suitable mutation framework exists
- Mutant reverse engineering is possible

### Indications

```plaintext
if migrated system mutant is killed then
  if old system mutant is killed then
    Expected case
  else
    if old system mutant is equivalent then
      No indication
    else
      Scenario 1a should be revisited
else //migrated system mutant NOT killed
  if old system mutant is killed then
    At least one migrated test case is a false negative
  else
    if old system mutant is equivalent then
      No indication
    else
      Scenario 1a should be revisited
```

### Artefacts
- Mutant (System or Test Case)
- Reverse engineered Mutant
- Migrated Mutant

### Activities
- Mutant generation
- Mutant migration
- Reverse engineering of Mutant
- Test execution against system mutant
- Test execution against system

---

**User Conference on Advanced Automated Testing**

© All rights reserved
Mutation Analysis Scenarios

### Artefacts
- Mutant (System or Test Case)
- Reverse engineered Mutant
- Migrated Mutant

### Activities
- Mutant generation
- Mutant migration
- Reverse engineering of Mutant
- Test execution against system mutant
- Test execution against system
Mutation Analysis Scenarios

**4 Mutation of Old Test Cases**

- **Old Test Case Mutant**
- **Old Test Cases** \(\rightarrow\) **New Test Case Mutant**
- **Old System** \(\downarrow\) **Old Test Cases** \(\rightarrow\) **New Test Cases** \(\downarrow\) **Old System** \(\rightarrow\) **New System**

**5 Mutation of New Test Cases**

- **Old Test Case Mutant**
- **Old Test Cases** \(\rightarrow\) **New Test Case Mutant**
- **Old System** \(\downarrow\) **Old Test Cases** \(\rightarrow\) **New Test Cases** \(\downarrow\) **Old System** \(\rightarrow\) **New System**

<table>
<thead>
<tr>
<th>Artefacts</th>
<th>Activities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mutant (System or Test Case)</td>
<td>Mutant generation</td>
</tr>
<tr>
<td>Reverse engineered Mutant</td>
<td>Mutant migration</td>
</tr>
<tr>
<td>Migrated Mutant</td>
<td>Reverse engineering of Mutant</td>
</tr>
</tbody>
</table>

New Test Cases and Old Test Cases.

Test execution against system mutant.

Test execution against system.
Mutation Analysis Scenarios

Assumptions
- Suitable mutation framework exists
- Mutant reverse engineering is possible

Indications

\[
\text{if } \text{old test case mutant fails} \quad \text{then} \\
\text{Expected case}
\]

\[
\text{else} \\
\begin{align*}
\text{if } \text{old test case mutant is equivalent} & \quad \text{then} \\
\text{No indication} \\
\text{else} & \\
\text{Bad smell for test case migration}
\end{align*}
\]
Mutation Analysis Scenarios

### Artefacts
- Mutant (System or Test Case)
- Reverse engineered Mutant
- Migrated Mutant

### Activities
- Mutant generation
- Mutant migration
- Reverse engineering of Mutant
- Test execution against system mutant
- Test execution against system

**Diagram:**
- Old Test Case Mutant
- New Test Case Mutant
- Old Test Cases → New Test Cases
- Old System → New System
- Test Case Migration
- System Migration
- 6 Mutation of Test Case Migration
Comprehensive discussion on the application
Thank you for your attention

Software Innovation Campus
Paderborn University
Fürstenalle 11
33102 Paderborn

Ivan Jovanovikj
Tel.: (05251) 60-6841
ivan.jovanovikj@sicp.uni-paderborn.de
https://www.sicp.de/