Test Case Generation Technique and Process; Business Driven

Nicha Kosindrdecha and Jirapun Daengdej
Faculty of Science and Technology
Assumption University, Thailand

21 September 2010
Agenda

• Overall
• Research Problem
• Literature Review
• Proposed Method
  – Process
  – Technique
• Evaluation
• Conclusion
Activities:
1. Insert Card
2. Input PIN
3. Select Withdraw
4. Select A/C Type
5. Input Balance
6. Get Money
7. Get Card

Functions:
1. Withdraw
2. Transfer
3. Payment
4. Others

Test Scenario

Test Sequence

Test Data

ATM Machine

Bank

Test Case
### Testing Process

#### Research Problems

1. Large Number of Test Cases.
2. Not Cover Critical Requirements.

#### Methods to Generate Test Cases

1. White-Box Testing
2. Black-Box Testing

#### UML Diagrams

- 2.2.1 UML Use Case Diagram
- 2.2.2 UML Activity Diagram
- 2.2.3 UML State Chart Diagram
- 2.2.4 UML Sequence Diagram

#### Testing Techniques

- 2.1 Derive Tests from Requirements Specification
- 2.2 Derive Tests from Diagrams
1. A Large Number of Test Cases

2. Not Cover Critical Requirements.

Methods to Generate Test Cases

1. White-Box Testing
2. Black-Box Testing

Test Case Generation Techniques

1. Derive Tests from Requirements Specification
2. Derive Tests from Diagrams

Outstanding Problems Motivated This Research

1. Lack of requirement prioritization before test case generation
2. Unable to identify which test cases can be removed during a test case generation process.
3. Large number of test cases due to large number of alternate paths / events.
1. Requirement
2. Design
3. Development
4. Testing
5. Maintenance

1.1 Test Case Generation

1. Requirement
2. Design
3. Development
4. Testing
5. Maintenance

2.1 Test Case Generation

1. Requirement
2. Design
3. Development
4. Testing
5. Maintenance

3.1 Test Case Generation

1. Requirement
2. Design
3. Development
4. Testing
5. Maintenance

3.1 Test Case Generation

1. Requirement
2. Design
3. Development
4. Testing
5. Maintenance

3.1 Test Case Generation

1. Requirement
2. Design
3. Development
4. Testing
5. Maintenance

Problem #1: Lack of Requirement Prioritization Before Test Case Generation Process


(Black-Box)


(White-Box)

(Black-Box)

(Sami et al, 2003; Alberto et al, 1995; C. Ramamoorthy et al, 1976)

*Comparative Test Case Generation Methods
In order to remove a number of above test cases, the following options can be made:

- Option #1 – Remove T2, T3 and T4.
- Option #2 – Remove T2 and T3.
- Option #3 – Remove T2 and T4.
- Option #4 – Remove T3 and T4.
- Option #5 – Remove Only T1.
...
Literature Review

Problem #3: Large number of test cases due to large number of alternate paths / events.

(Peter et al, 2006 (IBM Research); Jim Heumann, 2001; Johannes Ryser et al, 2000; Manish Nilawar et al, 2003)
Overall Proposes

Minimize A Number of Test Cases

Methods to Generate Test Cases (from Use Case Diagram)

A Number of Test Cases

Risk

Quality

S/W Testing

Test Case Complexity

Return on Investment

Requirement Complexity

Customer Satisfaction

Business

Time-to-Market

Risk

Quality

Overall

Problem

Review

Propose

Evaluate

Conclusion

Cover Critical Requirement

Based on Business’s Perspective

10
Overall Proposes

Problem #1: Lack of Requirement Prioritization Before Test Case Generation Process

Propose #1 – Test Case = Use Case + WOW Factors + Marketing Driven Requirement Prioritization

Problem #2: Unable to identify which test cases can be removed during a test case generation process.

Propose #2 – Test Case = Use Case + WOW Factors + Marketing Driven Requirement Prioritization + Alternate Path Points

Problem #3: Large number of test cases due to large number of alternate paths / events.

Propose #3 – Test Case = Use Case + WOW Factors + Marketing Driven Requirement Prioritization + Alternate Path Points + Removable Score

Correlation between ROI and Requirement Complexity

Propose #1 – Test Case = Use Case + WOW Factors + Marketing Driven Requirement Prioritization

Problem #1: Lack of Requirement Prioritization Before Test Case Generation Process

Propose #2 – Test Case = Use Case + WOW Factors + Marketing Driven Requirement Prioritization + Alternate Path Points

Problem #2: Unable to identify which test cases can be removed during a test case generation process.

Propose #3 – Test Case = Use Case + WOW Factors + Marketing Driven Requirement Prioritization + Alternate Path Points + Removable Score

Problem #3: Large number of test cases due to large number of alternate paths / events.
Overall Proposes

Problem #1: Lack of Requirement Prioritization Before Test Case Generation Process

Propose #1 – Test Case = Use Case + WOW Factors + Marketing Driven Requirement Prioritization

Problem #2: Unable to identify which test cases can be removed during a test case generation process.

Propose #2 – Test Case = Use Case + WOW Factors + Marketing Driven Requirement Prioritization + Alternate Path Points

Problem #3: Large number of test cases due to large number of alternate paths / events.

Propose #3 – Test Case = Use Case + WOW Factors + Marketing Driven Requirement Prioritization + Alternate Path Points + Removable Score

*Generate Minimum Number of Test Cases and Cover Critical Requirements*
A TEST CASE GENERATION PROCESS

The following slides present a test case generation process along with marketing-driven requirement prioritization activities.
Propose #1

Propose #1 – Test Case = Use Case + WOW Factors + Marketing Driven Requirement Prioritization

Marketing-Driven Requirement Prioritization Supported by WOW Factors

1. Requirement
2. Design
3. Development
4. Testing
5. Maintenance

Existing Test Case Generation Process

Proposed Test Case Generation Process
Propose #1

1. Requirement
   1.1 Requirement Prioritization
      1.1.1 Classify Requirements
      1.1.2 Prioritize Requirements
   2.1 Test Case Generation
      2.1.1 Extract Use Case Diagram
      2.1.2 Extract Test Scenario
      2.1.3 Generate Test Data and Expected Result
      2.1.4 Minimize Test Cases

Marketing-Driven Requirement Prioritization Supported by WOW Factors

Proposed Test Case Generation Process
Propose #1

A Survey Result of Customer Satisfaction

- Business leaders strongly believe that customer satisfaction is key to success
- Business leaders focus on price, product and service

*Source: Millard, 2006

Marketing-Driven Requirements

(Kano et al, 1984; Harkiranpal S., 2006; Herzberg et al, 1959; Cadotte et al, 1988; Tukman-Mert et al, 2007; Scott McCartney, 2007; James Clausen, 2009)

WOW Factors

Benefit & Cost Model to Prioritize Requirement

(Andrea Herrmann, 2008)

Requirement Complex

(Holly-Parsons-Hann, 2005)
Marketing-Driven Requirement

Customer Satisfaction

Customer's Need

Performance

Delight
(Nice to Have)

Attractive
(Surprise)

Basic
(Must be)

Indifferent
(Don’t Care)

Normal

Very satisfied

Excitement

Dissatisfied

Well Fulfilled

Not Fulfilled

*Kano Analysis Model from Marketing / Sales perspective (Kano et al, 1984; Harkiranpal S., 2006; Herzberg et al, 1959; Cadotte et al, 1988; Brandt et al, 1988)
WOW Factors

WOW Factors Supported Marketing’s View

Implementation Cost

Large
Medium
Small

(Tukman Mert et al, 2007; Scott Mccartney, 2007; James Clausen, 2009)
Example of Requirements

Implementation Cost based on Cost Value Approach

(Joachim Karlsson and Kevin Ryan, 1997)

WOW Factors Supported Marketing's View

Assumption: With the cost-value approach, we assume the above implementation cost for each requirement.
## Requirement Prioritization

### Step 1: Compute Total Estimated Cost

<table>
<thead>
<tr>
<th>Req. Id</th>
<th>Req. Description</th>
<th>Classification</th>
<th>Estimated Efforts for Coding (Man-Hours)</th>
<th>Estimated Efforts for Testing (Man-Hours)</th>
<th>Total Efforts (Man-Hours)</th>
<th>Cost-Value Assignment</th>
<th>Employee Cost per Hour</th>
<th>Total Estimated Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Interbank Fee among Other Banks</td>
<td>Basic</td>
<td>24</td>
<td>24</td>
<td><strong>48</strong></td>
<td>1</td>
<td>$65</td>
<td><strong>$3,120</strong></td>
</tr>
<tr>
<td>2</td>
<td>New Enhancement BASE24 EPS ATM</td>
<td>Basic</td>
<td>40</td>
<td>80</td>
<td><strong>120</strong></td>
<td>1.2</td>
<td>$78</td>
<td><strong>$9,360</strong></td>
</tr>
<tr>
<td>3</td>
<td>Interface with other banks</td>
<td>Basic</td>
<td>40</td>
<td>80</td>
<td><strong>120</strong></td>
<td>1.2</td>
<td>$78</td>
<td><strong>$9,360</strong></td>
</tr>
<tr>
<td>4</td>
<td>Performance Retrieval Transaction for ATM System</td>
<td>Surprise</td>
<td>48</td>
<td>16</td>
<td><strong>64</strong></td>
<td>3</td>
<td>$195</td>
<td><strong>$12,480</strong></td>
</tr>
<tr>
<td>5</td>
<td>Transaction Update with BOT</td>
<td>Surprise</td>
<td>48</td>
<td>40</td>
<td><strong>88</strong></td>
<td>3.1</td>
<td>$202</td>
<td><strong>$17,732</strong></td>
</tr>
<tr>
<td>6</td>
<td>Transaction Cross Check with G-ATM Pool</td>
<td>Surprise</td>
<td>40</td>
<td>42</td>
<td><strong>82</strong></td>
<td>3.2</td>
<td>$208</td>
<td><strong>$17,056</strong></td>
</tr>
<tr>
<td>7</td>
<td>Security Policy at Front End</td>
<td>WOW</td>
<td>32</td>
<td>24</td>
<td><strong>56</strong></td>
<td>5</td>
<td>$325</td>
<td><strong>$18,200</strong></td>
</tr>
<tr>
<td>8</td>
<td>Support VISA, Master Card and JCB</td>
<td>WOW</td>
<td>80</td>
<td>72</td>
<td><strong>152</strong></td>
<td>5.1</td>
<td>$332</td>
<td><strong>$50,388</strong></td>
</tr>
<tr>
<td>9</td>
<td>Operability Monitoring Transaction in ATM Systems</td>
<td>Surprise</td>
<td>40</td>
<td>40</td>
<td><strong>80</strong></td>
<td>3.3</td>
<td>$215</td>
<td><strong>$17,160</strong></td>
</tr>
<tr>
<td>10</td>
<td>Support Mutual Fund Investment</td>
<td>Basic</td>
<td>40</td>
<td>40</td>
<td><strong>80</strong></td>
<td>1.3</td>
<td>$85</td>
<td><strong>$6,760</strong></td>
</tr>
</tbody>
</table>

### Assumption

The employee cost per hour is $65 (Richard Denny, 2006).

*(Andrea Herrmann, 2008) focus on the requirement prioritization based on benefit and cost prediction / estimation.*
## Requirement Prioritization

### Step 2: Compute Total Charges (for Customer)

<table>
<thead>
<tr>
<th>Req. Id</th>
<th>Req. Description</th>
<th>Classification</th>
<th>Estimated Efforts for Coding (Man-Hours) (a)</th>
<th>Estimated Efforts for Testing (Man-Hours) (b)</th>
<th>Total Efforts (Man-Hours) (c)</th>
<th>Cost-Value Assignment (d)</th>
<th>Employee Cost per Hour (e)</th>
<th>Total Estimated Cost (f)</th>
<th>Charge Customer per Hour (g)</th>
<th>Total Charges (h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Interbank Fee among Other Banks</td>
<td>Basic</td>
<td>24</td>
<td>24</td>
<td>48</td>
<td>1</td>
<td>$65</td>
<td>$3,120</td>
<td>$100</td>
<td>$4,800</td>
</tr>
<tr>
<td>2</td>
<td>New Enhancement BASE24 EPS ATM</td>
<td>Basic</td>
<td>32</td>
<td>16</td>
<td>48</td>
<td>1.1</td>
<td>$72</td>
<td>$3,432</td>
<td>$300</td>
<td>$6,780</td>
</tr>
<tr>
<td>3</td>
<td>Interface with other banks</td>
<td>Basic</td>
<td>40</td>
<td>80</td>
<td>120</td>
<td>1.2</td>
<td>$78</td>
<td>$9,360</td>
<td>$200</td>
<td>$11,600</td>
</tr>
<tr>
<td>4</td>
<td>Performance Retrieval Transaction for ATM System</td>
<td>Surprise</td>
<td>48</td>
<td>16</td>
<td>64</td>
<td>3</td>
<td>$195</td>
<td>$12,480</td>
<td>$300</td>
<td>$15,480</td>
</tr>
<tr>
<td>5</td>
<td>Transaction Update with BOT</td>
<td>Surprise</td>
<td>48</td>
<td>40</td>
<td>88</td>
<td>3.1</td>
<td>$202</td>
<td>$17,732</td>
<td>$310</td>
<td>$20,842</td>
</tr>
<tr>
<td>6</td>
<td>Transaction Cross Check with G-ATM Pool</td>
<td>Surprise</td>
<td>48</td>
<td>40</td>
<td>88</td>
<td>3.1</td>
<td>$202</td>
<td>$17,732</td>
<td>$310</td>
<td>$20,842</td>
</tr>
<tr>
<td>7</td>
<td>Security Policy at Front End</td>
<td>Extra Ordinary</td>
<td>32</td>
<td>24</td>
<td>56</td>
<td>5</td>
<td>$325</td>
<td>$18,200</td>
<td>$500</td>
<td>$23,200</td>
</tr>
<tr>
<td>8</td>
<td>Support VISA, Master Card and JCB</td>
<td>Extra Ordinary</td>
<td>80</td>
<td>72</td>
<td>152</td>
<td>5.1</td>
<td>$332</td>
<td>$50,388</td>
<td>$510</td>
<td>$55,498</td>
</tr>
<tr>
<td>9</td>
<td>Operability Monitoring Transaction in ATM Systems</td>
<td>Surprise</td>
<td>40</td>
<td>40</td>
<td>80</td>
<td>3.3</td>
<td>$215</td>
<td>$17,160</td>
<td>$330</td>
<td>$20,490</td>
</tr>
<tr>
<td>10</td>
<td>Support Mutual Fund Investment</td>
<td>Basic</td>
<td>40</td>
<td>40</td>
<td>80</td>
<td>3.3</td>
<td>$215</td>
<td>$17,160</td>
<td>$330</td>
<td>$20,490</td>
</tr>
</tbody>
</table>

**Assumption** The charge customer per hour is equal to $100 (Richard Denny, 2006).

*(Andrea Herrmann, 2008) focus on the requirement prioritization based on benefit and cost prediction / estimation.*
## Requirement Prioritization

### Step 3: Compute ROI

<table>
<thead>
<tr>
<th>Req. Id</th>
<th>Req. Description</th>
<th>Classification</th>
<th>Estimate Efforts for Coding (Man-Hours) (a)</th>
<th>Estimate Efforts for Testing (Man-Hours) (b)</th>
<th>Total Efforts (Man-Hours) (c)</th>
<th>Cost-Value Assignment (d)</th>
<th>Employee Cost per Hour (e)</th>
<th>Total Estimated Cost (f)</th>
<th>Charge Customer per Hour (g)</th>
<th>Total Charges (h)</th>
<th>Estimated Benefit (i)</th>
<th>ROI (%) (j) = ((l/f) * 100)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Interbank Fee among Other Banks</td>
<td>Basic</td>
<td>24</td>
<td>24</td>
<td>48</td>
<td>1</td>
<td>$65</td>
<td>$3,120</td>
<td>$100</td>
<td>$4,800</td>
<td>$1,680</td>
<td>54%</td>
</tr>
<tr>
<td>2</td>
<td>New Enhancement BASE24 EPS ATM</td>
<td>Basic</td>
<td>32</td>
<td>16</td>
<td>48</td>
<td>1.1</td>
<td>$72</td>
<td>$3,433</td>
<td>$110</td>
<td>$5,280</td>
<td>$1,837</td>
<td>52%</td>
</tr>
<tr>
<td>3</td>
<td>Interface with other banks</td>
<td>Basic</td>
<td>40</td>
<td>80</td>
<td>120</td>
<td>1.2</td>
<td>$78</td>
<td>$5,60</td>
<td>$120</td>
<td>$14,400</td>
<td>$8,800</td>
<td>60%</td>
</tr>
<tr>
<td>4</td>
<td>Performance Retrieval Transaction for ATM System</td>
<td>Surprise</td>
<td>48</td>
<td>16</td>
<td>64</td>
<td>3</td>
<td>$195</td>
<td><strong>$12,480</strong></td>
<td>$300</td>
<td><strong>$19,200</strong></td>
<td><strong>$6,720</strong></td>
<td>54%</td>
</tr>
<tr>
<td>5</td>
<td>Transaction Update with BOT</td>
<td>Surprise</td>
<td>48</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$27,280</td>
<td>$9,548</td>
<td><strong>54%</strong></td>
<td><strong>54%</strong></td>
</tr>
<tr>
<td>6</td>
<td>Transaction Cross Check with G-ATM Pool</td>
<td>Surprise</td>
<td>40</td>
<td>42</td>
<td>82</td>
<td>3.2</td>
<td>$208</td>
<td><strong>$17,056</strong></td>
<td>$320</td>
<td><strong>$26,240</strong></td>
<td><strong>$9,184</strong></td>
<td>54%</td>
</tr>
<tr>
<td>7</td>
<td>Security Policy at Front End</td>
<td>Extra Ordinary</td>
<td>32</td>
<td>24</td>
<td>56</td>
<td>5</td>
<td>$325</td>
<td><strong>$18,200</strong></td>
<td>$500</td>
<td><strong>$28,000</strong></td>
<td><strong>$9,800</strong></td>
<td>54%</td>
</tr>
<tr>
<td>8</td>
<td>Support VISA, Master Card and JCB</td>
<td>Extra Ordinary</td>
<td>80</td>
<td>72</td>
<td>152</td>
<td>5.1</td>
<td>$332</td>
<td><strong>$50,388</strong></td>
<td>$510</td>
<td><strong>$77,520</strong></td>
<td><strong>$27,132</strong></td>
<td>54%</td>
</tr>
<tr>
<td>9</td>
<td>Operability Monitoring Transaction in ATM Systems</td>
<td>Surprise</td>
<td>40</td>
<td>40</td>
<td>80</td>
<td>3.3</td>
<td>$215</td>
<td><strong>$17,160</strong></td>
<td>$330</td>
<td><strong>$26,400</strong></td>
<td><strong>$9,240</strong></td>
<td>54%</td>
</tr>
<tr>
<td>10</td>
<td>Support Mutual Fund Investment</td>
<td>Basic</td>
<td>40</td>
<td>40</td>
<td>80</td>
<td>1.3</td>
<td>$85</td>
<td><strong>$6,760</strong></td>
<td>$130</td>
<td><strong>$10,400</strong></td>
<td><strong>$3,640</strong></td>
<td>54%</td>
</tr>
</tbody>
</table>

\[
\text{Estimated Benefit} = \text{Total Charges} - \text{Total Estimated Cost}
\]

\[
\text{ROI} = \left( \frac{\text{Estimated Benefit}}{\text{Total Est. Cost}} \right) \times 100
\]
## Requirement Prioritization

### Step 4: Compute Correlation Ratio

<table>
<thead>
<tr>
<th>Req. Id</th>
<th>Req. Description</th>
<th>Classification</th>
<th>Total Efforts (Man-Hours)</th>
<th>ROI (%)</th>
<th>Requirement Complexity Factor</th>
<th>Correlation Ratio (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Interbank Fee among Other Banks</td>
<td>Basic</td>
<td>48</td>
<td>54%</td>
<td>3</td>
<td>18%</td>
</tr>
<tr>
<td>2</td>
<td>New Enhancement BASE24 EPS ATM</td>
<td>Basic</td>
<td>48</td>
<td>54%</td>
<td>3</td>
<td>18%</td>
</tr>
<tr>
<td>3</td>
<td>Interface with other banks</td>
<td>Basic</td>
<td>120</td>
<td>54%</td>
<td>3</td>
<td>18%</td>
</tr>
<tr>
<td>4</td>
<td>Performance Retrieval Transaction for ATM System</td>
<td>Surprise</td>
<td>64</td>
<td>54%</td>
<td>3</td>
<td>18%</td>
</tr>
<tr>
<td>5</td>
<td>Transaction Update with BOT</td>
<td>Surprise</td>
<td>88</td>
<td>54%</td>
<td>3</td>
<td>18%</td>
</tr>
<tr>
<td>6</td>
<td>Transaction Cross Check with G-ATM Pool</td>
<td>Surprise</td>
<td>82</td>
<td>54%</td>
<td>3</td>
<td>18%</td>
</tr>
<tr>
<td>7</td>
<td>Security Policy at Front End</td>
<td>WOW</td>
<td>56</td>
<td>54%</td>
<td>3</td>
<td>18%</td>
</tr>
<tr>
<td>8</td>
<td>Support VISA, Master Card and JCB</td>
<td>WOW</td>
<td>152</td>
<td>54%</td>
<td>4</td>
<td>13%</td>
</tr>
<tr>
<td>9</td>
<td>Operability Monitoring Transaction in ATM Systems</td>
<td>Surprise</td>
<td>80</td>
<td>54%</td>
<td>3</td>
<td>18%</td>
</tr>
<tr>
<td>10</td>
<td>Support Mutual Fund Investment</td>
<td>Basic</td>
<td>80</td>
<td>54%</td>
<td>3</td>
<td>18%</td>
</tr>
</tbody>
</table>

(Holly-Parsons-Hann, 2005) “Measuring Requirement Complexity”

<table>
<thead>
<tr>
<th>Number of Hours</th>
<th>0-8</th>
<th>9-40</th>
<th>41-160</th>
<th>161-320</th>
<th>321-480</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complexity</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

*Note: 1 is very low, 2 is low, 3 is medium, 4 is high and 5 is very high.*
**Requirement Prioritization**

**Step 5: Prioritize Requirements**

<table>
<thead>
<tr>
<th>Req. Id</th>
<th>Req. Description</th>
<th>Classification</th>
<th>Total Estimated Cost</th>
<th>Correlation Ratio (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Interbank Fee among Other Banks</td>
<td>Basic</td>
<td>$3,120</td>
<td>18%</td>
</tr>
<tr>
<td>2</td>
<td>New Enhancement BASE24 EPS ATM</td>
<td>Basic</td>
<td>$3,432</td>
<td>18%</td>
</tr>
<tr>
<td>3</td>
<td>Interface with other banks</td>
<td>Basic</td>
<td>$9,360</td>
<td>13%</td>
</tr>
<tr>
<td>4</td>
<td>Performance Retrieval Transaction for ATM System</td>
<td>Surprise</td>
<td>$12,480</td>
<td>18%</td>
</tr>
<tr>
<td>5</td>
<td>Transaction Update with BOT</td>
<td>Surprise</td>
<td>$17,732</td>
<td>18%</td>
</tr>
<tr>
<td>6</td>
<td>Transaction Cross Check with G-ATM Pool</td>
<td>Surprise</td>
<td>$17,056</td>
<td>18%</td>
</tr>
<tr>
<td>7</td>
<td>Security Policy at Front End</td>
<td>Extra Ordinary</td>
<td>$18,200</td>
<td>18%</td>
</tr>
<tr>
<td>8</td>
<td>Support VISA, Master Card and JCB</td>
<td>Extra Ordinary</td>
<td>$50,388</td>
<td>13%</td>
</tr>
<tr>
<td>9</td>
<td>Operability Monitoring Transaction in ATM Systems</td>
<td>Surprise</td>
<td>$17,160</td>
<td>18%</td>
</tr>
<tr>
<td>10</td>
<td>Support Mutual Fund Investment</td>
<td>Basic</td>
<td>$6,760</td>
<td>18%</td>
</tr>
</tbody>
</table>

In case that Correlation Ratio is the same, consider the low total estimated cost.

*(Andrea, 2008)*

Prioritized Req. = \{R10, R2, R3, R1, R4, R6, R9, R5, R7, R8\}

**WOW Factors Supported Marketing’s View**
A TEST CASE GENERATION TECHNIQUE DERIVED FROM USE CASE DIAGRAM

The following slides present a method to generate test cases derived from UML Use Case diagram.
Alistair Cockburn classified use cases into the below three categories.

**Brief Use Case**
- Use Case Name
- Use Case Number
- Goal / Purpose

**Casual Use Case**
- Use Case Name
- Use Case Number
- Goal / Purpose
- Summary

**Fully Dressed Use case**
- Use Case Name
- Use Case Number
- Goal / Purpose
- Summary
- Actors
- Stakeholders
- Pre-Conditions
- Post-Conditions
- Basic Event
- Alternative Events
- Business Rules
- Notes
- Version
- Author
- Date

**Use Case Diagram**

**Withdraw**

**Transfer**

**Customer**

Limitation:
(a) The information for each use case must be fully completed.
(b) Alternative paths for each use case must be completed. In practical, we assume that test engineers may need to complete the information in format of steps / paths.
## Fully Dressed Use Case

<table>
<thead>
<tr>
<th>Use Case Id</th>
<th>Use Case Name</th>
<th>Purpose</th>
<th>Summary</th>
<th>Actor(s)</th>
<th>Stakeholders</th>
<th>Pre-Conditions</th>
<th>Post-Conditions</th>
<th>Basic Event</th>
<th>Alternative Events</th>
<th>Business Rules</th>
<th>Notes</th>
<th>Version</th>
<th>Author</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>UC-001</td>
<td>Withdraw</td>
<td>To withdraw money from ATM machines</td>
<td>To allow bank's customers to withdraw money from ATM machines anywhere in Thailand.</td>
<td>Customers</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>1. Insert Card</td>
<td>1. Select Inquiry</td>
<td>(a) Input amount &lt;= Outstanding Balance</td>
<td>No</td>
<td>v1.0</td>
<td>Test</td>
<td>27-Jul-10</td>
</tr>
</tbody>
</table>
### Use Case Scenario

**Step 1: Extract Use Case Diagram**

<table>
<thead>
<tr>
<th>Case Id</th>
<th>Case Name</th>
<th>Pre-</th>
<th>Stakeholder</th>
<th>Pre-Cond.</th>
<th>Post-Cond.</th>
<th>Basic Event</th>
<th>Alternative Events</th>
<th>Business Rules</th>
<th>Notes</th>
<th>Version</th>
<th>Author</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>UC-001</td>
<td>Withdraw</td>
<td>No</td>
<td>Customers</td>
<td>No</td>
<td>No</td>
<td>1. Insert Card</td>
<td>1. Select balance</td>
<td>a) Input amount</td>
<td>No</td>
<td>v1.0</td>
<td>Test</td>
<td>27-Jul-10</td>
</tr>
<tr>
<td></td>
<td>To withdraw money from ATM machines</td>
<td></td>
<td>To allow bank's customers to withdraw money from ATM machines anywhere in Thailand.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Scenario 001

<table>
<thead>
<tr>
<th>Id</th>
<th>Purpose</th>
<th>Summary</th>
<th>Pre-Cond.</th>
<th>Post-Cond.</th>
<th>Basic Scenario</th>
</tr>
</thead>
</table>

### Scenario 002

<table>
<thead>
<tr>
<th>Id</th>
<th>Purpose</th>
<th>Summary</th>
<th>Pre-Cond.</th>
<th>Post-Cond.</th>
<th>Basic Scenario</th>
</tr>
</thead>
</table>

---

**Use-Case Cross Reference**

- Scenario-001
- Scenario-002

- Use-Case Reference: UC-001
# Test Scenario Generation

## Step 2: Extract Test Scenario

<table>
<thead>
<tr>
<th>Id</th>
<th>Summary</th>
<th>Pre-Condition</th>
<th>Post-Condition</th>
<th>Basic Scenario</th>
<th>Use-Case Cross Reference</th>
</tr>
</thead>
</table>
| Scenario-001 | To withdraw money from ATM machines anywhere in Thailand. | No            | No             | 1. Insert Card  
  2. Input PIN  
  3. Select Withdraw  
  4. Select A/C Type  
  5. Input Balance  
  6. Get Money  
  7. Get Card | UC-001 |
| Scenario-002 | To withdraw money from ATM machines anywhere in Thailand. | No            | No             | 1. Insert Card  
  2. Input PIN  
  3. Select Inquiry  
  4. Select A/C Type  
  5. Check Balance  
  6. Select Withdraw  
  7. Select A/C Type  
  8. Input Balance  
  9. Get Money  
  10. Get Card | UC-001 |

## Test Scenario Id

<table>
<thead>
<tr>
<th>Id</th>
<th>Purpose</th>
<th>Summary</th>
<th>Pre-Condition</th>
<th>Post-Condition</th>
<th>Basic Scenario</th>
<th>Use Case Scenario</th>
</tr>
</thead>
</table>
| TS-001 | To withdraw money from ATM machines anywhere in Thailand. | To allow bank's customers to withdraw money from ATM machines anywhere in Thailand. | No            | No             | 1. Insert Card  
  2. Input PIN  
  3. Select Withdraw  
  4. Select A/C Type  
  5. Input Balance  
  6. Get Money  
  7. Get Card | Scenario-001 |
| TS-002 | To withdraw money from ATM machines anywhere in Thailand. | To allow bank's customers to withdraw money from ATM machines anywhere in Thailand. | No            | No             | 1. Insert Card  
  2. Input PIN  
  3. Select Inquiry  
  4. Select A/C Type  
  5. Check Balance  
  6. Select Withdraw  
  7. Select A/C Type  
  8. Input Balance  
  9. Get Money  
  10. Get Card | Scenario-002 |
# Test Case Generation

<table>
<thead>
<tr>
<th>Test Scenario Id</th>
<th>Purpose</th>
<th>Summary</th>
<th>Pre-Condition</th>
<th>Post-Condition</th>
<th>Basic Scenario</th>
<th>Use Case Scenario</th>
</tr>
</thead>
</table>
| TS-001           | To withdraw money from ATM machines | To allow bank's customers to withdraw money from ATM machines anywhere in Thailand. | No            | No            | 1. Insert Card  
2. Input PIN  
3. Select Withdraw  
4. Select A/C Type  
5. Input Balance  
6. Get Money  
7. Get Card | Scenario-001 |
| TS-002           | To withdraw money from ATM machines | To allow bank's customers to withdraw money from ATM machines anywhere in Thailand. | No            | No            | 1. Insert Card  
2. Input PIN  
3. Select Inquiry  
4. Select A/C Type  
5. Check Balance  
6. Select Withdraw  
7. Select A/C Type  
8. Input Balance  
9. Get Money  
10. Get Card | Scenario-002 |

## Step 3: Generate Test Data and Expected Result

<table>
<thead>
<tr>
<th>Test Case Id</th>
<th>Purpose</th>
<th>Summary</th>
<th>Pre-Conditions</th>
<th>Post-Conditions</th>
<th>Basic Scenario</th>
<th>Test Scenario</th>
<th>Input Data</th>
<th>Expected Result</th>
<th>Actual Result</th>
<th>Status Pass / Fail</th>
</tr>
</thead>
</table>
| TC-001       | To withdraw money from ATM machines | To allow bank's customers to withdraw money from ATM machines anywhere in Thailand. | No             | No             | 1. Insert Card  
2. Input PIN  
3. Select Withdraw  
4. Select A/C Type  
5. Input Balance  
6. Get Money  
7. Get Card | TS-001 | PIN, Amount Balance | Customers get money and the balance is calculated successfully. | ... | ... |
| TC-002       | To withdraw money from ATM machines | To allow bank's customers to withdraw money from ATM machines anywhere in Thailand. | No             | No             | 1. Insert Card  
2. Input PIN  
3. Select Inquiry  
4. Select A/C Type  
5. Check Balance  
6. Select Withdraw  
7. Select A/C Type  
8. Input Balance  
9. Get Money  
10. Get Card | TS-002 | PIN, Amount Balance | The outstanding balance is displayed. Customers get money and the balance is calculated successfully. | ... | ... |
## Test Case Generation

### Testing Matrix Table

<table>
<thead>
<tr>
<th></th>
<th>Use Case 1</th>
<th>Use Case 2</th>
<th>Use Case 3</th>
<th>Use Case 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>T2</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>T3</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>T4</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

**Step 4: Minimize Test Cases**

In order to remove a number of above test cases, the following options can be made:

- Option #1 – Remove T2, T3 and T4.
- Option #2 – Remove T2 and T3.
- Option #3 – Remove T2 and T4.
- Option #4 – Remove T3 and T4.
- Option #5 – Remove Only T1.
- ... 

Problem #2: Unable to identify which test cases can be removed during a test case generation process.
## Propose #2

### Step 1: Create Testing Matrix between Test Cases and Alternate Paths of Use Cases

An Example of Testing Matrix Table between Test Cases and Paths / Events in Use Case

<table>
<thead>
<tr>
<th>T1</th>
<th>Path #1 of Use Case 1</th>
<th>Path #2 of Use Case 1</th>
<th>Path #1 of Use Case 2</th>
<th>Path #2 of Use Case 2</th>
<th>Path #3 of Use Case 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>T2</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T3</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T4</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>T5</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>T7</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Overall: [Diagram showing flow through Overall, Problem, Review, Propose, Evaluate, and Conclusion stages.]

### Problem

### Review

### Propose

### Evaluate

### Conclusion
Propose #2

<table>
<thead>
<tr>
<th>Path #1 of Use Case 1</th>
<th>Path #2 of Use Case 1</th>
<th>Path #1 of Use Case 2</th>
<th>Path #2 of Use Case 2</th>
<th>Path #3 of Use Case 2</th>
<th>Weight</th>
<th>Value (No of Covered Paths)</th>
<th>Alternate Path Points (Weight * Value)</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>2</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>T2</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>T3</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>T4</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>2</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>T5</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>T6</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>T7</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

From UCP method (Gautam Banerjee, 2001), the weight factors for alternate paths can be found as follows:
- **Weight** = 1 when a number of paths is: <= 3.
- **Weight** = 2 when a number of paths is: 4 <= paths <= 7.

**Problem #3:** Large number of test cases due to large number of alternate paths / events.
Propose #3

**Step 1: Compute Complexity**

**Step 2: Calculate Removable Score**

Removable Score = Alternate Path Points \* Complexity

**Step 3: Remove Test Cases with Minimum Risk Driven Factors**

Let Complexity Factor (Hans Schaefer, 2005) measures a total number of test steps in each test case.

For example, T1 is used to test path #1 of use case 1, path #2 of use case 1, path #1 of use case 2, and path #2 of use case 2. Each path requires a number of test steps, represented in the left-hand side table. Thus, the total number of test steps is 3+3+4+2, which is equal to 12.
EVALUATION

The following slides present an evaluation method, results and discussion.
Evaluation Factors:

1. **A Number of Test Cases**: This is the total number of generated test cases, expressed as a percentage, as follows:
   \[
   \% \text{ Size} = \left( \frac{\# \text{ Size}}{\# \text{ of Total Size}} \right) \times 100
   \]

2. **Percentage of Critical Requirement Coverage**: This is an indicator to identify a number of requirements covered in the system, particularly critical requirements [10]. Due to the fact that one of software testing goals is to verify and validate requirements covered against the system, therefore, this factor should be considered.
   \[
   \% \text{ CRC} = \left( \frac{\# \text{ Critical}}{\# \text{ of Total}} \right) \times 100
   \]

3. **Total Time**: This is a total number of times running the generation methods in the experiment. This metric is related to time used during testing development phase (e.g. design test scenario and produce test case). Therefore, less time is desirable.
   \[
   \text{Total Time} = \text{Preparation Time} + \text{Compile Time} + \text{Running Time}
   \]
<table>
<thead>
<tr>
<th>Attribute</th>
<th>Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirement ID</td>
<td>Randomly generated from the following combination: Req + Sequence Number.</td>
</tr>
<tr>
<td></td>
<td>For example, Req1, Req2, Req3, …, ReqN.</td>
</tr>
<tr>
<td>Description</td>
<td>Randomly generated from the following combination: Des + Sequence Number same as Requirement ID.</td>
</tr>
<tr>
<td></td>
<td>For example, Des1, Des2, Des3, …, DesN.</td>
</tr>
<tr>
<td>Type of Requirement</td>
<td>Randomly selected from the following values: Functional and Non-Functional.</td>
</tr>
<tr>
<td>Classification</td>
<td>Randomly generated from: Basic, Surprise and WOW.</td>
</tr>
<tr>
<td>Estimated Efforts for Coding</td>
<td>Randomly generated from 1 to 480 hours.</td>
</tr>
<tr>
<td>Estimated Efforts for Testing</td>
<td>Randomly generated from 1 to 480 hours.</td>
</tr>
<tr>
<td>Cost-Value Assignment</td>
<td>Randomly generated from 1 to 5 such as 1, 1.1, 1.2, 1.3, 3, 3.1, 3.2, 3.3, 5, 5.1, 5.2 and 5.3.</td>
</tr>
</tbody>
</table>
## Approach to Generate Random Requirements

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use case ID</td>
<td>Randomly generated from the following combination: ( uCase + \text{Sequence Number} ). For example, ( uCase_1, uCase_2, \ldots, uCase_n ).</td>
</tr>
<tr>
<td>Purpose</td>
<td>Randomly generated from the following combination: ( \text{Pur} + \text{Sequence Number same as Use case ID} ). For example, ( \text{Pur}_1, \text{Pur}_2, \ldots, \text{Pur}_n ).</td>
</tr>
<tr>
<td>Pre-condition</td>
<td>Randomly generated from the following combination: ( pCon + \text{Sequence Number same as Use case ID} ). For example, ( pCon_1, pCon_2, \ldots, pCon_n ).</td>
</tr>
<tr>
<td>Basic Path</td>
<td>Randomly generated from the following combination: ( uCase + \text{Sequence Number} ). For example, ( basic_1, basic_2, \ldots, basic_n ).</td>
</tr>
</tbody>
</table>
| Alternate Paths | (a) Random a number of paths from 1 to 10  
(b) For each path, randomly generated from the following combination: \( altPath + \text{Sequence Number} \). For example, \( alt1, alt2, \ldots, alt3 \). |
Compare Percentage of A Number of Test Cases, Critical Req. Coverage and Total Time

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Excellent</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Poor</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Percentage</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A Number of Test Cases</td>
<td>97.80% 98.00% 97.00%</td>
<td>80.80%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Critical Requirement Coverage</td>
<td>20.20% 19.80% 19.00%</td>
<td></td>
<td>53.20%</td>
<td></td>
</tr>
<tr>
<td>Total Time</td>
<td>20.48% 31.82% 31.61% 30.20%</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Result

Heumann's Method
Ryser's Method
Nilawar's Method
Our Proposed Method
### Discussion

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>A Number of Test Cases</th>
<th>Critical Requirement Coverage</th>
<th>Total Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heumann’s Method</td>
<td>1</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>Ryser’s Method</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Nilawar’s Method</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Proposed Method</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

5-Excellent, 4-Very good, 3-Good, 2-Normal and 1-Poor.
A Number of Test Cases

Existing Test Case Generation Methods

Proposed Test Case Generation Method

Proposed Method

Heumann’s Method

Ryser’s Method

Nilawar’s Method
Critical Requirement Coverage

A Number of Test Cases

Critical Requirement Coverage

Proposed Method

Existing Test Case Generation Methods

Proposed Test Case Generation Method

Heumann’s Method

Ryser’s Method

Nilawar’s Method

Overall

Problem

Review

Propose

Evaluate

Conclusion

High

Low

Smaller

Larger
Future Works:

- Enhance the ability to weight and prioritize both of requirements and use cases’ alternate paths.
- Improve the ability to compute a level of damage in the risk value.
- Include an estimated cost and test execution as weight prioritization factors.
- Incorporate other UML diagrams to automatically generate test cases.
Thank You