My Flings with Data Analysis

Venkatesh-Prasad Ranganath
Kansas State University

Dagstuhl Software Development Analytics Seminar
June 25, 2014

The described efforts were carried out at Microsoft Research and Microsoft.
Observations

1. Involve the users early
2. Don’t fear the unconventional
3. Embrace dynamic artifacts
   - Logs, core dumps, usage profiles, etc.
4. Have a feedback loop involving the users
5. Domain knowledge is a power pellet
6. Questions and answers first, methods later
7. Relevant features and simple methods work really well
8. Quick trumps slow and utility trumps quick
9. Automate (whenever possible)
10. Presentation matters
11. Models are good, simple explanations are better
Is USB 3.0/Win8 ≈ USB 2.0/Win7?

When a USB 2.0 device is plugged into a USB 3.0 port on Win8, the USB 3.0 stack in Win8 should behave as the USB 2.0 stack in Win7 (along both software and hardware interfaces).

Collaborators: Randy Aull, Pankaj Gupta, Jane Lawrence, Pradip Vallathol, & Eliyas Yakub
One could ....
Patterns-based Compatibility Testing

DispatchIrп *forward alternates with* IrпCompletion &amp; PreIпCompleteRequest when
IOCTLType=IRP_MJ_PNP(0x1B),IRP_MN_START_DEVICE(0x00), irпID=SAME, and
IrпSubmitDetails.irп.ioStackLocation.control=SAME
<table>
<thead>
<tr>
<th>Device Id</th>
<th>Detected</th>
<th>Reported</th>
<th>False +ve</th>
<th>Unique</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9844</td>
<td>478</td>
<td>465</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>2545</td>
<td>15</td>
<td>11</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>743</td>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1372</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>26118</td>
<td>55</td>
<td>55</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>26126</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>2320</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>27804</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>34985</td>
<td>115</td>
<td>98</td>
<td>4</td>
</tr>
<tr>
<td>10</td>
<td>51556</td>
<td>59</td>
<td>56</td>
<td>3</td>
</tr>
<tr>
<td>11</td>
<td>695</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>1372</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>3315</td>
<td>24</td>
<td>23</td>
<td>1</td>
</tr>
<tr>
<td>14</td>
<td>9299</td>
<td>3</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>
Test Suite Reduction

Collaborators: Naren Datha, Robbie Harris, Aravind Namasivayam, & Pradip Vallathol
Patterns-based Test Suite Reduction

Cluster-n-Select
Results

- 50% reduction in test suite size
- 75-80% bugs uncovered (v/s human-based baseline)
- Fully automated (except for threshold setting)
- Flying without safety net
Structural & Temporal Patterns (w/ Data flow)

21 = fopen("passwd.txt", "r") .... fclose(21)
Observations

1. Involve the users early
2. Don’t fear the unconventional
3. Embrace dynamic artifacts
   • Logs, core dumps, usage profiles, etc.
4. Have a feedback loop involving the users
5. Domain knowledge is a power pellet
6. Questions and answers first, methods later
7. Relevant features and simple methods work really well
8. Quick trumps slow and utility trumps quick
9. Automate (whenever possible)
10. Presentation matters
11. Models are good, simple explanations are better
Opportunities

• Combining of dynamic and static artifacts/techniques
  • Time to move beyond repositories
• Using data analysis to improve techniques
• Exploring/designing language for
  • Query
  • Visualization
• Scaling
  • Distributed Computing
  • GPUs