Context

- **Software Maintenance/Bug Detection concepts:**
  - Bugs per KLOC (Fumio Akiyama, 1971)
  - Bugs per Cyclomatic Complexity (Thomas McCabe, 1976)
  - Bugs impact software maintenance efforts

- **Software development studies often use data insights from:**
  - Software configuration management repositories (SCM)
  - Source code systems (SC)
  - Bug tracking (BT) tools

- **They usually do not include:**
  - Knowledge about the adopted software process followed
  - Metadata collected from the development environments (IDEs) about user activities
The Problem

Represent a “Perfect-World”
Lack of development process dimension
Not real-time
Research Questions

▪ **RQ 1:** To what extent can process mining discover accurate models representing developers’ behaviour?
  ▪ Operationalize data collection and analysis

▪ **RQ 2:** Can we assess the efficiency of software development teams by using process mining techniques?

▪ **RQ 3:** The assessment of team’s proficiency, performed by a process expert, is reflected in the quality of the produced models?
The Solution

Represent a “Perfect-World”
Lack of development process dimension
Not real-time

Understand Developers behaviour
Assess development process quality
Predict / Suggest developer actions
The Experiment (SPOTS Project)

- **Task**
  - Detect Code Smells using JDeodorant
    - Long Method, God Class, Feature Envy, Type Checking

- **Real-Time Collection**
  - Analytics Plug-in
    - Eclipse development events
    - Multiple geographies
    - Multiple teams
    - Real-time

- **Archiving**
  - Local CSV File
  - Microsoft Azure Event Hub
    - Using Java API
    - Stored in JSON Blob files

- **Loading**
  - CSV and Azure data loaded into MySQL
  - Exported to CSV and loaded into PROM 6.8
  - Converted to IEEE XES format
Data Structure

- **Collection**
  - Eclipse Plugin

- **Used Properties**
  - group
  - timestamp_begin
  - timestamp_end
  - categoryName
  - commandName
  - username, fullname

```json
{
    "source": "Azure",
    "group": "T-101",
    "Schedule": "Day",
    "session": "Project",
    "timestamp_begin": "2018-05-07 16:53:52.144",
    "timestamp_end": "2018-05-07 16:54:04.468",
    "gender": "Female",
    "fullname": "Ana Sample",
    "username": "ana",
    "workspacename": "Workspace1",
    "projectname": "/g2o/another",
    "filename": "/g2o/another/AncestorTest.java",
    "extension": "java",
    "categoryName": "Eclipse Editor",
    "commandName": "File Editing",
    "categoryID": "org.eclipse.ui.internal.EditorReference",
    "commandID": "iscte.plugin.eclipse.commands.file.edit",
    "categoryDescription": "All events in the Eclipse Editor",
    "commandDescription": "File being edited in workbench",
    "platform": "Eclipse",
    "platform_branch": "Eclipse Oxygen",
    "platform_version": "4.7.3.M20180330-0640",
    "java": "1.8.0_171-b11",
    "timezone": "Europe/London",
    "locale": "en",
    "os": "Windows 10 win32",
    "os_arch": "x86_64",
    "plugin": "iscte.analytics.plugin.eclipse",
    "plugin_version": "1.0.0.201804144",
    "plugin_jar": "iscte-analytics-plugin-eclipse-1.0.0.201804144.jar",
    "topic": "event"
}
```

Listing 1: Sample Eclipse Event Instance
Data demographics

- **Resources Data**
  - 180 students
  - 52 students valid data

- **Teams Selected**
  - Project /jasml-0.10

- **Project Events vs. Generic Events**

<table>
<thead>
<tr>
<th>Resource</th>
<th># Distinct</th>
</tr>
</thead>
<tbody>
<tr>
<td>Students</td>
<td>52</td>
</tr>
<tr>
<td>Workspaces</td>
<td>43</td>
</tr>
<tr>
<td>Projects</td>
<td>14</td>
</tr>
<tr>
<td>Files</td>
<td>573</td>
</tr>
<tr>
<td>Command Categories</td>
<td>22</td>
</tr>
<tr>
<td>Commands</td>
<td>240</td>
</tr>
<tr>
<td>Operating Systems</td>
<td>5</td>
</tr>
<tr>
<td>Operating System Architectures</td>
<td>2</td>
</tr>
<tr>
<td>Eclipse Bundles</td>
<td>2</td>
</tr>
<tr>
<td>Eclipse Bundle Versions</td>
<td>10</td>
</tr>
<tr>
<td>Java Versions</td>
<td>18</td>
</tr>
<tr>
<td>Timezones</td>
<td>3</td>
</tr>
<tr>
<td>Locales</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Team</th>
<th>TM</th>
<th>UCC</th>
<th>UCA</th>
<th>UEA</th>
<th>PE (#/%)</th>
<th>GE (#/%)</th>
<th>TE (#)</th>
</tr>
</thead>
<tbody>
<tr>
<td>T-43</td>
<td>4</td>
<td>10</td>
<td>38</td>
<td>39</td>
<td>790/85.13%</td>
<td>138/14.87%</td>
<td>928</td>
</tr>
<tr>
<td>T-41</td>
<td>2</td>
<td>10</td>
<td>37</td>
<td>40</td>
<td>615/77.75%</td>
<td>176/22.25%</td>
<td>791</td>
</tr>
<tr>
<td>T-02</td>
<td>3</td>
<td>12</td>
<td>41</td>
<td>24</td>
<td>552/74.80%</td>
<td>186/25.20%</td>
<td>738</td>
</tr>
<tr>
<td>T-26</td>
<td>2</td>
<td>8</td>
<td>28</td>
<td>22</td>
<td>360/77.25%</td>
<td>106/22.75%</td>
<td>466</td>
</tr>
<tr>
<td>T-23</td>
<td>1</td>
<td>9</td>
<td>23</td>
<td>22</td>
<td>276/93.24%</td>
<td>20/6.76%</td>
<td>296</td>
</tr>
<tr>
<td>T-21</td>
<td>1</td>
<td>9</td>
<td>27</td>
<td>23</td>
<td>272/77.71%</td>
<td>78/22.29%</td>
<td>350</td>
</tr>
<tr>
<td>T-24</td>
<td>1</td>
<td>8</td>
<td>26</td>
<td>13</td>
<td>181/89.60%</td>
<td>21/10.40%</td>
<td>202</td>
</tr>
<tr>
<td>T-01</td>
<td>4</td>
<td>4</td>
<td>12</td>
<td>20</td>
<td>134/97.10%</td>
<td>4/2.90%</td>
<td>138</td>
</tr>
</tbody>
</table>

| REF  | 1  | 4   | 12  | 20  | 134/97.10%| 4/2.90%   | 138    |

**Legend:**
- TM - Team members
- UCC - Unique Command Categories
- UCA - Unique Command Actions
- UEA - Unique Edited Artifacts
- PE - Project related events
- GE - Generic Eclipse events
- TE - Total events
The Analysis

- **Statechart Plugin (ProM 6.8)**
  - Hierarchical Discovery
  - Structured Names Heuristic ( | )
The Analysis

- **Statechart Plugin (ProM 6.8)**
  - Hierarchical Discovery
  - Structured Names Heuristic (|)
  - Using Alignments

![Diagram of Statechart Plugin and its applications](image)
The Analysis

- **Process Comparator (ProM 6.8)**
  - Compares Behaviour
  - Differences between teams

The following list ranks pairs of event logs based on their percentage of control-flow differences. We recommend that you select the highest ranked pair as a starting point:

- (88.51%) PROFESSOR vs EIC1-43
- (88.50%) PROFESSOR vs EIC1-01
- (87.60%) PROFESSOR vs EIC1-41
- (86.84%) PROFESSOR vs EIC2-02
- (86.49%) ICI-24 vs EIC2-01
- (85.11%) PROFESSOR vs ICI-26
- (85.04%) PROFESSOR vs ICI-24
- (83.33%) PROFESSOR vs ICI-21
The Results

- **Simplicity/Complexity**
  - Teams T-26, T-24, T-41 produced models less complex
  - T-01 and T-02 generated models with higher complexity

- **Differences**
  - T-24 & T-26 are similar in Control-Flow
  - The ones more similar to the Professor

- **Performance**

<table>
<thead>
<tr>
<th>Team</th>
<th>Proficiency</th>
<th>Process Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>T-43</td>
<td>0</td>
<td>23h:49m</td>
</tr>
<tr>
<td>T-41</td>
<td>0.73</td>
<td>18d:3h</td>
</tr>
<tr>
<td>T-02</td>
<td>0.75</td>
<td>11d:22h</td>
</tr>
<tr>
<td>T-26</td>
<td>0.75</td>
<td>12d:16m</td>
</tr>
<tr>
<td>T-23</td>
<td>0.72</td>
<td>12d:13m</td>
</tr>
<tr>
<td>T-21</td>
<td>0.02</td>
<td>8d:12h</td>
</tr>
<tr>
<td>T-24</td>
<td>0.64</td>
<td>47m:14s</td>
</tr>
<tr>
<td>T-01</td>
<td>0</td>
<td>10d:7h</td>
</tr>
<tr>
<td>REF. Log</td>
<td>–</td>
<td>23m:05</td>
</tr>
</tbody>
</table>

![Graph showing differences in control-flow]
Threats to Validity

- Used a group of undergraduate students
  - They may not have yet the maturity needed for this type of request

- Students may have used, in the same computer, the same user/key pair to activate the collection plugin even if there was multiple people coding

- Some users have stopped the collection mechanism in the Eclipse plugin which makes it impossible to understand what they were doing during that period.

- Few teams have made a pause in the exercise, causing it to express more execution time than what was really needed.
  - The process mined reflects these times, but indeed, that was idle time
Assessing Software Development Teams’ Efficiency using Process Mining

João Caldeira, Aachen, 08/07/2019

Introduction

Research Problems

Solution

Results

Conclusion

Summary

▪ Able to collect and give team context to development data originated from distributed individuals (RQ1)
  ▪ Understand better the magnitude of differences among users environments

▪ Assess the quality of discovered development processes (RQ2)
  ▪ Detect the teams with best performance

▪ Evaluate the developers performance vs. a expert performance (RQ3)
  ▪ Compare teams’ work from a Control-Flow dimension to a reference process model
Next Steps

- **SPOTS Project**
  - Software Process On-the-run Tracking System
  - Prepare Micro-Services Engine
  - Deliver Dashboards Portal

- **Refine the Collection Plugin**

- **Evaluate Code Complexity vs. Process Complexity**

- **More Experiments**
  - Social Perspective
  - Performance
  - Predict on Process Smells

- **Plugin Ecosystem**
  - Explore Insights
  - Understand Diversity
  - Understand Variability

<table>
<thead>
<tr>
<th>Resource</th>
<th># Distinct</th>
</tr>
</thead>
<tbody>
<tr>
<td>Students</td>
<td>52</td>
</tr>
<tr>
<td>Workspaces</td>
<td>43</td>
</tr>
<tr>
<td>Projects</td>
<td>14</td>
</tr>
<tr>
<td>Files</td>
<td>573</td>
</tr>
<tr>
<td>Command Categories</td>
<td>22</td>
</tr>
<tr>
<td>Commands</td>
<td>240</td>
</tr>
<tr>
<td>Operating Systems</td>
<td>5</td>
</tr>
<tr>
<td>Operating System Architectures</td>
<td>2</td>
</tr>
<tr>
<td>Eclipse Bundles</td>
<td>2</td>
</tr>
<tr>
<td>Eclipse Bundle Versions</td>
<td>10</td>
</tr>
<tr>
<td>Java Versions</td>
<td>18</td>
</tr>
<tr>
<td>Timezones</td>
<td>3</td>
</tr>
<tr>
<td>Locales</td>
<td>3</td>
</tr>
</tbody>
</table>
Assessing Software Development Teams’ Efficiency using Process Mining

João Caldeira, ISCTE-IUL
Fernando Brito e Abreu (ISCTE-IUL), José Reis (ISCTE-IUL), Jorge Cardoso, (DIE/UC, Huawei Munich Research Center)

Aachen, June 24th, 2019