Process Extraction from Development Artifacts

Abram Hindle, Michael W. Godfrey, Richard C. Holt

Software Architecture Group
David R. Cheriton School of Computer Science
University of Waterloo
Canada
http://swag.uwaterloo.ca/
{ahindle,migod,holt}@cs.uwaterloo.ca
Process Extraction

• Extract information about:
  – behavior
  – topics / focus
  – requirements
  – Software Development Life Cycle
  – repeating behaviors

• from...
Development Artifacts

- Wand of dispel demo demons
- Blessed wand of developer motivation (4:3)
- Rusty cursed plate mail named “contract deadline”
- Blessed dagger named “debug”
- Potion of Bad Smell Detection
Artifacts

- Mailing lists
- Bug tracker events
- Source Control Repositories
- Source Code
- Documentation
- Test
- Build system
Figure 1: Remember the Rational Unified Process?
What is that diagram

- We have work-flows or areas of emphasis
- Plotted across time
  - across phases
- We could provide this to stake-holders
Self Reflection

- Not in OO terms, analysis of one’s self
- Analysis
- Can we reflect on a project’s processes, focuses, behaviors?
  - Can we do this automatically?
Dashboard versus Time-line

- Dashboard: An indicator of state
- Time-line: An indicator of past state correlated with time
- Speedometer versus Odometer
Who needs a Time-line

- Stake-holders not intimately familiar with development
  - Managers
  - New Developers
  - Developers after 2 weeks vacation
  - Acquisitions
Proposal

- Automatically generate something like the RUP lumpy diagram
- Provide overviews in a time-line form but allow investigation
  - Allow zooming
  - Allow slicing
How does our previous work relate?

- Time-lines
- Metrics
- Evolution
YARN on a Timeline
Release Patterns

Revisions per Time Unit summed per time unit before and after a release

Source Code Revisions per Time Unit (day) Smoothed Summed Near Release

Test Revisions per Time Unit (day) Smoothed Summed Near Release

Build Revisions per Time Unit (day) Smoothed Summed Near Release

Documentation Revisions per Time Unit (day) Smoothed Summed Near Release

Linear Regression
Indentation Metrics

Get the Diff

```c
void square( int * arr, int n ) {
    int i = 0;
    for ( i = 0 ; i < n ; i++ ) {
        arr[ i ] *= arr[ i ];
    }
}
```

Measure the Indentation

<table>
<thead>
<tr>
<th>Raw Indentation</th>
<th>Logical Indentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>044840</td>
<td>011210</td>
</tr>
</tbody>
</table>

Produce Summary Statistics

<table>
<thead>
<tr>
<th>Metric</th>
<th>Raw</th>
<th>Logical</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOC</td>
<td>6.000</td>
<td>6.000</td>
</tr>
<tr>
<td>AVG</td>
<td>3.330</td>
<td>0.833</td>
</tr>
<tr>
<td>MED</td>
<td>4.000</td>
<td>1.000</td>
</tr>
<tr>
<td>STD</td>
<td>2.750</td>
<td>0.687</td>
</tr>
<tr>
<td>VAR</td>
<td>9.070</td>
<td>0.567</td>
</tr>
<tr>
<td>SUM</td>
<td>20.000</td>
<td>5.000</td>
</tr>
<tr>
<td>MCC</td>
<td>2.000</td>
<td>2.000</td>
</tr>
<tr>
<td>HVOL</td>
<td>152.000</td>
<td>152.000</td>
</tr>
<tr>
<td>HDIFF</td>
<td>15.000</td>
<td>15.000</td>
</tr>
<tr>
<td>HEFFORT</td>
<td>2127.000</td>
<td>2127.000</td>
</tr>
</tbody>
</table>
Large Changes

Proportional Distribution of Extended Swanson Maintenance Classes

Extended Swanson Categories
- Boost
- EGroupware
- Enlightenment
- Evolution
- Firebird
- MySQL 5.0
- PostgreSQL
- Samba
- Spring Framework

Proportion of Commits
Topic Trends
Topic Trends Proposed
Frequency Analysis
How do we combine these?

- Luckily we have local time correlations of events
- We can slice up data sources
Time-line Example
Time-line Topics

Changes Per Day

Development Topic A

Development Topic B

Development Topic C
Time-line Topics Select

Changes Per Day

Development Topic A

Development Topic B

Development Topic C
Time-line Topics Sliced

Changes Per Day

Development Topic A

Development Topic B

Development Topic C
What are work-flows we can look for

- STBD
- Communication measurements
  - Design & Implementation
  - Bugs, fixing, debugging
  - “Nonfunctional”
What to do with the data

• Visualization
  – Dashboard / Time-line

• Analysis
  – Event analysis
  – Windowed Analysis
  – Correlation
  – Tagging
Further Analysis

• Can we spot patterns of phases
  – Identify phases
  – Find repeating sequences of phases?
Conclusions

- Aim to visualize and analyze data about a project with respect to entities and time
- Produce a tool to act as dashboard but also a time-line of the topics and behaviors of a project
- Produce a methodology and tool communicate project activities to stake-holders
- Tool should help automate parts of project retrospectives