Extracting new metrics from Version Control System for the comparison of software developers

Marcello Moura$^1$, Hugo Nascimento$^2$ e Thierson Rosa$^2$

Centro de Recursos Computacionais$^1$, Instituto de Informática$^2$
Universidade Federal de Goiás (UFG)
Caixa Postal 131 – 74.001-970 – Goiânia – GO – Brazil
marcello@ufg.br, {hadn,thierson}@inf.ufg.br

Goiânia, 21 de Setembro 2014
Summary I

1. Introduction
2. Extracting fine-grain operations from VCS
3. Metrics for the developers
4. Comparison of the developers
5. The case study
6. Conclusion
Introduction

Extracting fine-grain operations from VCS

Metrics for the developers

Comparison of the developers

The case study

Conclusion
Version Control Systems (VCSs), like Subversion and Git, store revisions of the files of a software development project, registering its historical evolution.
VCSs have been used for:

- **Helping to understand the software development process** – Lopez-Fernandez et al. [2004], Huang and Liu [2005], Girba et al. [2005], Voinea and Telea [2006] and Voinea et al. [2007].

- **Helping to know more about the developers** – Gilbert and Karahalios [2007], Jermakovics et al. [2011], Mockus and Herbsleb [2002], Minto and Murphy [2007], Schuler and Zimmermann [2008], Zhang et al. [2008a,b] and Di Bella et al. [2013].
Our work focuses on understanding the developers by the analysis of their work.

1. We identify and count finer-grain operations at line and file levels that can be extracted from a VCS, like additions, deletions and modifications.
   - This allows to derive a much more detailed and rich information about the work performed by the developers.

2. We calculate a new set of formally defined metrics.

3. Developers are characterized by comparing each one of them against the others.
   - Two comparison approaches for this aim are described.
**Note**: The VCS data cannot be taken as a full and precise description of the software development process.

- It is incomplete and may lead to distinct interpretations. (e.g. Negara et al. [2012])
- Information extracted from a VCS has to be revalidated by the project managers and complemented with their own knowledge.
**Note:** The VCS data cannot be taken as a full and precise description of the software development process.

- It is incomplete and may lead to distinct interpretations. (e.g. Negara et al. [2012])
- Information extracted from a VCS has to be revalidated by the project managers and complemented with their own knowledge.
Summary

1 Introduction

2 Extracting fine-grain operations from VCS

3 Metrics for the developers

4 Comparison of the developers

5 The case study

6 Conclusion
Basic notation:

\( \mathcal{P} \) – a software project in a VCS

\( \mathcal{D} \) – the set of developers that worked on \( \mathcal{P} \).

\( \mathcal{A} \) – the set of all files created during the development of \( \mathcal{P} \).

\( \mathcal{A}' \subseteq \mathcal{A} \) – the set of files that were removed (not reached the final version) of \( \mathcal{P} \).
We mine the VCS for three types of operations: additions, deletions and modifications of files and lines of code.
Extracting fine-grain operations from VCS

Rev(i - 1)
line1
line2
line3

Rev(i)
line0
line1
line2.1

i ← i + 1

@@ -1,3 +1,3 @@
+ line0
| line1
- line2
- line3
+ line2.1

Unified Diff

List of ADD, MOD and DEL Operations of lines in files

H_{a_1}
...
H_{a_2}
h_{i_1}
0_1 \cdots 0_{end}
h_{i_n}
0_1 \cdots 0_{end}
H_{a_m}
...
Summary

1. Introduction
2. Extracting fine-grain operations from VCS
3. Metrics for the developers
4. Comparison of the developers
5. The case study
6. Conclusion
Aspects defined for consideration:

1. **Effort** – represents the total amount of operations of a type performed by a developer.

2. **Code-survival** – indicates the amount of operations of a type performed by a developer and not changed later by anyone.
Metrics for the developers

A. Metrics for evaluating developers individually

\[ Effo_{Add}(d) = \sum_{a \in A} \sum_{i=1}^{H_a} \left\{ \begin{array}{ll} 1 & \text{if } o_{1,i}^{a}.\text{devel} = d \\ 0 & \text{otherwise.} \end{array} \right. \]

\[ Effo_{Mod}(d) = \sum_{a \in A} \sum_{i=1}^{H_a} \sum_{j=1}^{h^a_{i,j}} \left\{ \begin{array}{ll} 1 & \text{if } o_{j,i}^{a}.\text{devel} = d \\ \text{and } o_{j,i}^{a}.\text{type} = \text{MOD}; & \text{otherwise.} \end{array} \right. \]
Metrics for the developers

A. Metrics for evaluating developers individually

\[
\text{Surv}_{-}\text{Add}(d) = \sum_{a \in (A - Ar)} \left\{ \sum_{i=1}^{|H_a|} \begin{cases} 
1 & \text{if } o_1^{a,i}.\text{devel} = d \\
& \text{and } \forall o_s^{a,i} \text{ with } s > 1, \\
& (o_s^{a,i}.\text{type} = \text{MOD} \\
& \text{and } o_s^{a,i}.\text{devel} = d); \\
0 & \text{otherwise.}
\end{cases} \right. 
\]

\[
\text{Surv}_{-}\text{Mod}(d) = \sum_{a \in (A - Ar)} \left\{ \sum_{i=1}^{|H_a|} \begin{cases} 
1 & \text{if } o_{\text{end}}^{a,i}.\text{type} = \text{MOD} \\
& \text{and } o_{\text{end}}^{a,i}.\text{devel} = d \\
& \exists w, 1 \leq w < |h^a_i|, \\
& \text{such that } o_w^{a,i}.\text{devel} \neq d; \\
0 & \text{otherwise.}
\end{cases} \right. 
\]
A. Metrics for evaluating developers individually

\[ \text{Surv}\_\text{Add}\_\text{Div}\_\text{Effo}\_\text{Add}(d) = \frac{\text{Surv}\_\text{Add}(d)}{\text{Effo}\_\text{Add}(d)} \]
### Metrics for the developers

#### B. Uncovering and measuring relationships between developers

Also, ADD, DEL, MOD, MOD, MOD, DEL.

<table>
<thead>
<tr>
<th></th>
<th>d1</th>
<th>d2</th>
<th>dn</th>
</tr>
</thead>
<tbody>
<tr>
<td>d1</td>
<td>43</td>
<td>56</td>
<td>102</td>
</tr>
<tr>
<td>d2</td>
<td>22</td>
<td>127</td>
<td>69</td>
</tr>
<tr>
<td>dn</td>
<td>88</td>
<td>241</td>
<td>...</td>
</tr>
</tbody>
</table>

Also, ADD_DEL, MOD_MOD, MOD_DEL.
Line_Add_Mod(x, y) = \sum_{a \in A} \sum_{i=1}^{H_a} \begin{cases} 
1 & \text{if } |h_{i}| > 1 \\
\text{and } o_{1}^{a,i}.devel = x \\
\text{and } o_{1}^{a,i}.type = ADD \\
\text{and } o_{2}^{a,i}.devel = y \\
\text{and } o_{2}^{a,i}.type = MOD; \\
0 & \text{otherwise.} \end{cases}
Metrics for the developers

B. Uncovering and measuring relationships between developers

\[ \text{Line}_{\text{Add} \sum \text{Mod}}(d) = \sum_{y \in D - \{d\}} \text{Line}_{\text{Add} \text{Mod}}(d, y) \]

\[ \text{Line}_{\sum \text{Add} \text{Mod}}(d) = \sum_{x \in D - \{d\}} \text{Line}_{\text{Add} \text{Mod}}(x, d) \]
A project revision is a triple \((r, d, L)\), where:

- \(r\) is the label of the revision,
- \(d\) is a identifier of the developer who made the revision, with \(d \in \mathcal{D}\), and
- \(L\) is a list of pairs \((a, t)\) where \(a\) is a file and \(t \in \{A, M, D\}\) describes the operation.

A project revision sequence is a sequence \(S = \langle (r_1, d_1, L_1), (r_2, d_2, L_2), \ldots, (r_m, d_m, L_m) \rangle\) of project revisions that represent the history of changes made on the files of \(\mathcal{P}\) without going into detail about the changes made on their individual lines.
C. Extending the metrics for the file level

\[ File_{\text{Add}}_{\text{Mod}}(x, y) = \sum_{a \in A} \begin{cases} 1 & \text{if there are triples } (r_i, d_i, L_i) \\
\text{and } (r_j, d_j, L_j) \text{ in } S, \text{ with } i < j, \\
\text{such that } d_i = x, d_j = y, \\
(a, A) \in L_i \text{ and } (a, M) \in L_j, \\
\text{and for which there is no triple } \\
(r_k, d_k, L_k) \text{ with } i < k < j \\
\text{such that } (a, t) \in L_k \\
\text{for any operation of type } t; \\
0 & \text{otherwise.} \end{cases} \]
C. Extending the metrics for the file level

\[
\text{File}_{-}\text{Add}_{-}\sum_{\text{Mod}}(d) = \sum_{y \in \mathcal{D} \setminus \{d\}} \text{File}_{-}\text{Add}_{-}\text{Mod}(d, y)
\]

\[
\text{File}_{-}\sum_{\text{Add}_{-}\text{Mod}}(d) = \sum_{x \in \mathcal{D} \setminus \{d\}} \text{File}_{-}\text{Add}_{-}\text{Mod}(x, d)
\]
Metrics for the developers
D. Metrics regarding commits

\[ \text{Commits}(x, y) = \sum_{i=1}^{\left| S \right|-1} \begin{cases} 
1 & \text{if triples } (r_i, d_i, L_i) \text{ and } (r_{i+1}, d_{i+1}, L_{i+1}) \text{ are such that } d_i = x \text{ and } d_{i+1} = y; \\
0 & \text{otherwise.}
\end{cases} \]

\[ \sum \text{Commits}(d) = \sum_{i=1}^{\left| S \right|} \begin{cases} 
1 & \text{if triple } (r_i, d_i, L_i) \text{ is such that } d_i = d; \\
0 & \text{otherwise.}
\end{cases} \]
Metrics for the developers

\[ \text{Metric}_\text{Rel}(d) = \frac{\text{Metric}(d)}{\sum_{x \in D} \text{Metric}(x)} \]
Summary

1. Introduction
2. Extracting fine-grain operations from VCS
3. Metrics for the developers
4. Comparison of the developers
5. The case study
6. Conclusion
Comparison of the developers
A. Performance-based hierarchy

All metrics should have the same orientation
Comparison of the developers

B. Similarity Comparison

Metrics for all developers

Correlation matrix

List of low correlated matrix

MDS

Some metrics for all developers
1. Introduction
2. Extracting fine-grain operations from VCS
3. Metrics for the developers
4. Comparison of the developers
5. The case study
6. Conclusion
The case study

Evaluating the metrics and the comparison approaches with qualitative assessment on a real software-development project.

The software Weby

- A content management system built by UFG.
- Hosting more than 400 internal web sites\(^1\).
- Considered time (1 year and 7 months).
- Eleven (11) developers contributed to the evolution of the source code.
  - One developer was also the project manager.
- 1,294 code revisions into VCS (Subversion) of UFG.

\(^{1}\)The available at https://github.com/cercomp/weby.
## The case study

<table>
<thead>
<tr>
<th>D.</th>
<th>Commits</th>
<th>Files</th>
<th>Lines</th>
</tr>
</thead>
<tbody>
<tr>
<td>d1</td>
<td>474</td>
<td>482</td>
<td>1,807</td>
</tr>
<tr>
<td>d2</td>
<td>159</td>
<td>47</td>
<td>453</td>
</tr>
<tr>
<td>d3</td>
<td>2</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>d4</td>
<td>170</td>
<td>314</td>
<td>585</td>
</tr>
<tr>
<td>d5</td>
<td>30</td>
<td>43</td>
<td>78</td>
</tr>
<tr>
<td>d6</td>
<td>99</td>
<td>333</td>
<td>367</td>
</tr>
<tr>
<td>d7</td>
<td>61</td>
<td>12</td>
<td>379</td>
</tr>
<tr>
<td>d8</td>
<td>183</td>
<td>848</td>
<td>783</td>
</tr>
<tr>
<td>d9</td>
<td>20</td>
<td>1</td>
<td>34</td>
</tr>
<tr>
<td>d10</td>
<td>24</td>
<td>8</td>
<td>74</td>
</tr>
<tr>
<td>d11</td>
<td>72</td>
<td>7</td>
<td>199</td>
</tr>
<tr>
<td>Total</td>
<td>1,294</td>
<td>2,095</td>
<td>4,765</td>
</tr>
</tbody>
</table>

Moura, Nascimento e Rosa
The evaluation was conducted through two assessments involving four steps each:

1. Calculation of the values of a set of metrics for all developers.
2. Computation of the hierarchy of classes and the MDS visualization.
3. Interview with the project manager, aiming to verify if the classes and the visualization produced by the comparison approaches match his/her perception about the developers.
4. Analysis and interpretation of the results obtained from the interview.
## Formulário de Entrevista

<table>
<thead>
<tr>
<th>Nome do Entrevistado:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Nome do Projeto:</td>
<td></td>
</tr>
<tr>
<td>Cargo:</td>
<td></td>
</tr>
<tr>
<td>Formação:</td>
<td></td>
</tr>
<tr>
<td><strong>Local e Data:</strong></td>
<td></td>
</tr>
<tr>
<td>1  Explicar os dados existentes e as métricas. (Explicar o que o sistema desenvolvido faz)</td>
<td></td>
</tr>
<tr>
<td>2  Apresentar a classificação por classe de dominância. (Explicar o significado de cada classe)</td>
<td></td>
</tr>
<tr>
<td>3  Perguntas sobre a classe de dominância.</td>
<td></td>
</tr>
<tr>
<td>a) “Essa separação faz sentido para você?”</td>
<td></td>
</tr>
<tr>
<td>b) “Se você fosse escolher um ou mais desenvolvedores para um projeto futuro, esta classificação ajudaria? Por quê? Quais os desenvolvedores você escolheria?”</td>
<td></td>
</tr>
<tr>
<td>c) “Você classificaria os desenvolvedores dessa mesma forma? Por quê? Se não, como seria sua classificação?”</td>
<td></td>
</tr>
<tr>
<td>d) “Tem algum desenvolvedor que você acha que foi classificado equivocadamente?”</td>
<td></td>
</tr>
<tr>
<td>4  Apresentar a visualização em MDS. (Explicar o que significa a distância entre dois desenvolvedores)</td>
<td></td>
</tr>
<tr>
<td>5  Perguntas sobre a visualização em MDS.</td>
<td></td>
</tr>
<tr>
<td>e) “Os desenvolvedores que estão próximos são, de fato, parecidos na sua produção técnica? Eles produzem resultados semelhantes?”</td>
<td></td>
</tr>
<tr>
<td>f) “Como você rotularia (daríamos nomes com base em alguma característica de similaridade) os “grupos” de pessoas visualmente próximas?”</td>
<td></td>
</tr>
<tr>
<td>g) “Há alguma discrepância ou semelhança entre os resultados das classes de dominância, apresentadas anteriormente, e a visualização MDS atual?”</td>
<td></td>
</tr>
<tr>
<td>6  Perguntas sobre o conjunto total de métricas.</td>
<td></td>
</tr>
<tr>
<td>b) “Você concorda que quanto maior for o valor obtido em cada uma dessas 4 métricas melhor foi o desempenho do desenvolvedor? Por quê?”</td>
<td></td>
</tr>
<tr>
<td>i) “Quais outras métricas (da planilha completa) você acha interessante/util para uma avaliação dos desenvolvedores? Por quê?”</td>
<td></td>
</tr>
</tbody>
</table>

---

Moura, Nascimento e Rosa

Extracting new metrics from VCS ... 30 / 48
The case study
A. Metrics and comparisons computed in the first assessment

<table>
<thead>
<tr>
<th>D.</th>
<th>Surv_Add</th>
<th>Surv_Mod</th>
<th>Surv_Add_Div_Effo_Add</th>
<th>Surv_Mod_Div_Effo_Dist_Mod</th>
</tr>
</thead>
<tbody>
<tr>
<td>d1</td>
<td>102,817</td>
<td>539</td>
<td>0.932</td>
<td>0.253</td>
</tr>
<tr>
<td>d2</td>
<td>3,188</td>
<td>294</td>
<td>*0.734</td>
<td>*0.609</td>
</tr>
<tr>
<td>d3</td>
<td>0</td>
<td>0</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>d4</td>
<td>41,929</td>
<td>410</td>
<td>0.952</td>
<td>0.455</td>
</tr>
<tr>
<td>d5</td>
<td>1,185</td>
<td>21</td>
<td>*0.682</td>
<td>*0.437</td>
</tr>
<tr>
<td>d6</td>
<td>50,630</td>
<td>479</td>
<td>0.979</td>
<td>*0.807</td>
</tr>
<tr>
<td>d7</td>
<td>483</td>
<td>163</td>
<td>*0.432</td>
<td>*0.612</td>
</tr>
<tr>
<td>d8</td>
<td>83,409</td>
<td>1,302</td>
<td>0.973</td>
<td>0.632</td>
</tr>
<tr>
<td>d9</td>
<td>55</td>
<td>211</td>
<td>*0.539</td>
<td>*0.875</td>
</tr>
<tr>
<td>d10</td>
<td>225</td>
<td>43</td>
<td>*0.415</td>
<td>*0.605</td>
</tr>
<tr>
<td>d11</td>
<td>1,053</td>
<td>315</td>
<td>*0.884</td>
<td>*0.734</td>
</tr>
</tbody>
</table>
### Equivalence Classes

<table>
<thead>
<tr>
<th>Equivalence Classes</th>
<th>Developers</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>d1, d6, d8</td>
</tr>
<tr>
<td>2</td>
<td>d4</td>
</tr>
<tr>
<td>3</td>
<td>d2, d11</td>
</tr>
<tr>
<td>4</td>
<td>d5, d7, d9</td>
</tr>
<tr>
<td>5</td>
<td>d10</td>
</tr>
<tr>
<td>6</td>
<td>d3</td>
</tr>
</tbody>
</table>
The case study

### Equivalence Classes

<table>
<thead>
<tr>
<th>Equivalence Classes</th>
<th>Developers [first]</th>
<th>Developers [second]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>d1, d6, d8</td>
<td>d1, d6, d4, d8</td>
</tr>
<tr>
<td>2</td>
<td>d4</td>
<td>d2, d11</td>
</tr>
<tr>
<td>3</td>
<td>d2, d11</td>
<td>d5, d7, d9</td>
</tr>
<tr>
<td>4</td>
<td>d5, d7, d9</td>
<td>d10</td>
</tr>
<tr>
<td>5</td>
<td>d10</td>
<td>d3</td>
</tr>
<tr>
<td>6</td>
<td>d3</td>
<td></td>
</tr>
</tbody>
</table>
The case study
Introduction

Extracting fine-grain operations from VCS

Metrics for the developers

Comparison of the developers

The case study

Conclusion
Conclusion I

- We presented new formal definitions and metrics that allow the extraction of basic but important information from projects hosted in VCSs.
- We considered measures of efforts and code-survival.
- Two approaches were suggested for comparing the developers.
- A case study with a real software project was carried out. The results showed the usefulness of the metrics and of the comparison approaches.
- The new metrics may help to unveil interesting facts.
- But there are limitations in the use of VCS data. The logs are in general incomplete and can lead to ambiguous interpretation.
Conclusion II

- We tried to compensate this weakness by involving the project manager.
Future Work

Future investigations include:

- formulating new metrics;
- using other techniques to compare the developers;
- improving the diff analysis for detecting other types of operation;
- exploring more sources of data.
Questions?
Extracting new metrics from Version Control System for the comparison of software developers

Marcello Moura¹, Hugo Nascimento² e Thierson Rosa²

Centro de Recursos Computacionais¹, Instituto de Informática²
Universidade Federal de Goiás (UFG)
Caixa Postal 131 – 74.001-970 – Goiânia – GO – Brazil
marcello@ufg.br, {hadn,thierson}@inf.ufg.br

Goiânia, 21 de Setembro 2014


