Factors affecting Technical Debt Raw data from a systematic literature map

Villar, Alberto
Matalonga, Santiago

Documento de Investigación No.12
Facultad de Ingeniería
Universidad ORT Uruguay
31 de marzo de 2014
ISSN 1688-6372

Documento de Investigación
Factors affecting Technical Debt Raw data

from a systematic literature map

Alberto Villar (Facultad de Ingeniería, Universidad ORT Uruguay)
Santiago Matalonga (Facultad de Ingeniería, Universidad ORT Uruguay)

Documento de Investigación No.12

Facultad de Ingeniería
Universidad ORT Uruguay

31 de marzo de 2014
Factors affecting Technical Debt
Raw data from a systematic literature map
(Documento de trabajo)

Alberto Villar (avillar@uni.ort.edu.uy)
Santiago Matalonga (smatalonga@uni.ort.edu.uy)

Marzo de 2014
Technical Debt Reading List.
Results from a Systematic mapping study

ABSTRACT

This document presents the complete list of references that have been short listed during the systematic review process carried out during the months of April-September 2012. The objective of the systematic review was to identify current research trends in technical debt and to explore the relationship between technical debt measures and agile software development. This document includes 352 references that are categorized according to their relevance to technical debt research.
This document is intended as a reading list for technical debt research. It details the 352 articles that have been reviewed during the systematic review performed by Alberto Villar from April 2012 until July 2012.

This report is the second intermediate result of an ongoing literature mapping project. After each iteration new knowledge has been developed.

This iteration was commissioned in order to explore the software metrics and factors that could affect a project’s technical debt.

Results of this systematic review have been sent to publication outlets and are being considered.

The references are organized according to a classification criteria driven by their application on the subject of Technical debt.

The following exclusion criteria were defined:

- The topic did not relate to software development (D_NET)
- The article is not accessible with the authors subscription level (NA)
- The article was filtered by authors after reading the abstract (D_PA)
- The article was filtered by author after reading the full paper (D_PL).
- Filtered because the registry did not represent a full paper (D_ITOC).
- Repeated papers were also filtered (REP).

The following table summarizes the filtering process.

### Keywords

Keywords have been enhanced from last iteration to encompass the full software development lifecycle:

- “technical debt”;
- “design debt”;
- “analysis debt”;
- “requirements debt”;
- “testing debt”;
- “configuration management debt”;
- “process debt”;
- “architecture debt”;
- “architectural debt”;
- “people debt”;
- “documentation debt”;
- “code quality debt”.

### Search Engines

The search was carried out during the month of August 2013 in ACM; Springer; IEEE; SciVerse; and Citeseerx.

### Summary of results

<table>
<thead>
<tr>
<th>D_NET</th>
<th>NA</th>
<th>D_PA</th>
<th>D_PL</th>
<th>D_ITOC</th>
<th>REP</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>27</td>
<td>2</td>
<td>0</td>
<td>9</td>
<td>20</td>
<td>58</td>
</tr>
</tbody>
</table>
Technical Debt Reading List.
Results from a Systematic mapping study

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>1</td>
<td>5</td>
<td>12</td>
<td>1</td>
<td>24</td>
</tr>
<tr>
<td>0</td>
<td>7</td>
<td>0</td>
<td>8</td>
<td>1</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>53</td>
</tr>
<tr>
<td>42</td>
<td>48</td>
<td>8</td>
<td>8</td>
<td>31</td>
<td>27</td>
<td>164</td>
</tr>
</tbody>
</table>

Factors affecting technical debt

The following table presents the factors that have been identified in the literature. Factors have been classified according to their relevance to the different stages of a software development lifecycle.

<table>
<thead>
<tr>
<th>Factores</th>
<th>cod</th>
<th>Life cycle</th>
<th>Factores (N_AD - N_TD_A)</th>
<th>N_T</th>
<th>N_T</th>
<th>N_N</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>deterioro de la arquitectura</td>
<td>F_D</td>
<td>AYD</td>
<td></td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>deterioro del código / diseño</td>
<td>F_D</td>
<td>AYD, Impl</td>
<td></td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>Falta de transparencia hacia otros stakeholders</td>
<td>F_F</td>
<td>TO DO</td>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>short-term versus long-term goals (nivel código y diseño)</td>
<td>F_S</td>
<td>AYD, Impl</td>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>short-term versus long-term goals (nivel arquitectura)</td>
<td>F_S</td>
<td>AYD, Impl</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>short-term versus long-term goals (nivel general)</td>
<td>F_S</td>
<td>TODO</td>
<td></td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>cambios rápidos en tecnologías</td>
<td>F_C</td>
<td>TODO</td>
<td></td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>nuevos requerimientos en entornos legados</td>
<td>F_N</td>
<td>Requerimientos</td>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>requerimientos mal definidos y/o cambiantes</td>
<td>F_R</td>
<td>Requerimientos</td>
<td></td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>cambios en el equipo</td>
<td>F_C</td>
<td>TODO</td>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>falta de tiempo / budget</td>
<td>F_F</td>
<td>TODO</td>
<td></td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>falta de fondos para realizar los cambios necesarios</td>
<td>F_F</td>
<td>TODO</td>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>falta de conocimiento/experiencia</td>
<td>F_F</td>
<td>TODO</td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>mal control de versionado</td>
<td>F_S</td>
<td>SCM</td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>poco esfuerzo y poca motivación</td>
<td>F_P</td>
<td>TODO</td>
<td></td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>insuficiente testing</td>
<td>F_IT</td>
<td>TEST</td>
<td></td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>excesivo testing manual</td>
<td>F_T</td>
<td>TEST</td>
<td></td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>
Table 3

<table>
<thead>
<tr>
<th>Technical Debt Reading List.</th>
<th>EM</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>mala integración y manejo de releases</td>
<td>F_REL</td>
<td>SCM</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>compromiso en una dimensión en detrimento de otra</td>
<td>F_DIM</td>
<td>TODO</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Like-to-like migration</td>
<td>F_PLL</td>
<td>TODO</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>bajo refactoring y rediseño</td>
<td>F_BRD</td>
<td>TODO</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>pobre codificación en etapa de desarrollo</td>
<td>F_CPD</td>
<td>TODO</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>gerenciamiento de proyectos caótico</td>
<td>F_GPC</td>
<td>TODO</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>time to market</td>
<td>F_TTM</td>
<td>TODO</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>obsolescencia tecnológica</td>
<td>F_OTN</td>
<td>TODO</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>cambio de estandares de calidad en el tiempo</td>
<td>F_CES</td>
<td>TODO</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>nuevas oportunidades de negocio</td>
<td>F_NOS</td>
<td>TODO</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>costos de implementación / rework</td>
<td>F_CLR</td>
<td>AYD, Impl</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>pobre adherencia a estandares de desarrollo</td>
<td>F_AES</td>
<td>AYD, Impl</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>adquisición de software legado</td>
<td>F_ASL</td>
<td>TODO</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>imprudencia</td>
<td>F_IMP</td>
<td>TODO</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 3

Papers regarding Technical debt research
- **N_Nom**: Are those papers that only mention technical debt.


Technical Debt Reading List.
Results from a Systematic mapping study


- N_TD_C: Are those papers whose scope on the Technical debt subject is limited to code.


Technical Debt Reading List.

Results from a Systematic mapping study


Technical Debt Reading List.
Results from a Systematic mapping study


Technical Debt Reading List.
Results from a Systematic mapping study


Technical Debt Reading List.

Results from a Systematic mapping study


Technical Debt Reading List.
Results from a Systematic mapping study


- N_TD_A: Are those papers with a broader scope on technical debt (encompassing concepts like poor architecture (Hunter and Spann 2008), poor requirements(Ivanović, America, and Snijders 2012), etc).


Technical Debt Reading List.
Results from a Systematic mapping study

- N_TD_I: Are those papers with an integral view on the development process when talking into account technical debt. For instance, (Klinger et al. 2011) take into account project stakeholders when analyzing technical debt decisions.


Technical Debt Reading List.
Results from a Systematic mapping study


Papers regarding agile reflection adoption

- AG_OT: Though agile concepts appeared in the title and abstract, the research was not relevant to the context of reflection.
  


Technical Debt Reading List.
Results from a Systematic mapping study
R. L. Baskerville, L. Mathiassen, J. Pries-Heje, and J. I. DeGross, Eds., Crossing the Chasm in

Methodology for Multi-channel Adaptive Web Information Systems,” World Wide Web, vol. 10, no. 4,

S. Biffl, A. Aurum, B. Boehm, H. Erdogmus, and P. Grünbacher, Eds., Value-Based Software

T. Binder, J. Löwgren, and L. Malmberg, Eds., Margot Brereton Designing from Somewhere – A


2007.


J. Bosch and P. Bosch-Sijtsema, “From integration to composition: On the impact of software
product lines, global development and ecosystems,” Journal of Systems and Software, vol. 83, no. 1,

S. Bressan, J. Küng, and R. Wagner, Eds., From Extreme Programming to Extreme Non-
programming: Is It the Right Time for Model Transformation Technologies?, vol. 4080. Berlin,

semiconductor lasers,” in OFC/IOOC . Technical Digest. Optical Fiber Communication Conference,
1999, and the International Conference on Integrated Optics and Optical Fiber Communication, 1999,
vol. 2, pp. 137–139.

L. M. Camarinha-Matos, H. Afsarmanesh, and M. Ollus, Eds., Advanced Collaborative Business

1A, pp. 512–515.

C. Chrysanthou, J. Boksiner, J. Scott, and T. J. Garner, “Effects of nearby objects on fading and
coupling in Line-Of-Sight environments,” in 2010 - MILCOM 2010 MILITARY

W. W. C. Chung and M. F. S. Chan, Agile Manufacturing: The 21st Century Competitive

A. M. Clarke, A. Borghesani, D. W. Smith, P. Ossieur, P. D. Townsend, R. Jensen, and N.
1–3, 2011.
Technical Debt Reading List.

Results from a Systematic mapping study


Technical Debt Reading List.

Results from a Systematic mapping study


Technical Debt Reading List.
Results from a Systematic mapping study


Technical Debt Reading List.

Results from a Systematic mapping study


Technical Debt Reading List.

Results from a Systematic mapping study


Technical Debt Reading List.
Results from a Systematic mapping study


- AG_NI: The articles mentioned agile reflection but do not dive into specific reflections activities.
Technical Debt Reading List.
Results from a Systematic mapping study


M. Toleman, A. Almeida, F. Darroch, and M. Ally, “Aligning Adoption Theory with Agile System Development Methodologies.”

Technical Debt Reading List.
Results from a Systematic mapping study

- **AG_Ref**: The papers in this category explicitly reference the topic of reflection adoption in agile contexts.

  
  
  
  
  M. Lamoreux, “Improving agile team learning by improving team reflections [agile software development],” in Agile Development Conference (ADC’05), 2005, pp. 139–144.
  
  
  
  

- **AG_CT**: These articles were determined to be marginally relevant, since they mention improvement and learning in an agile context without necessarily referencing reflection activities.

  
  
  
  
  
  
  
Technical Debt Reading List.

Results from a Systematic mapping study


• AG_RNC. This research mentions reflection or retrospective activities but applies them as control mechanisms for the process and not as learning opportunities. In our opinion, misunderstanding the agile values.