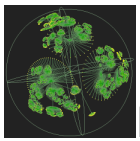
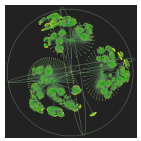
Technical Debt Indexes provided by tools: a preliminary discussion

Francesca Arcelli Fontana Riccardo Roveda **Marco Zanoni**


ESSeRE Lab, DISCo, University of Milano-Bicocca


MTD - 4 Oct 2016, Raleigh

Tecnical Debt Indexes (TDI)

A single number used to control quality and/or debt. *"Any kind of quality index computed by analysis tools"*

A.k.a. Technical Quality Index, Technical Debt/Severity, Deficit Index

- *Q1* How are the quality indexes the tools provide exactly computed?
  - Which features do they take into account?
- *Q2* Which index does take more into account the architectural issues and in which way?
- *Q3* Which are the features not provided or taken into account by the indexes?

Tools

- CAST 7.3.2 (`http://www.castsoftware.com/`)
- inFusion v.1.8.5 (`https://www.intooitus.com/`)[1]
- Sonargraph v.8.8.0 (`https://www.hello2morrow.com/products/sonargraph`)
- SonarQube v.5.2 (`http://www.sonarqube.org`)
- Structure101 v.4.2.10071 (`http://structure101.com/products/`)

[1] its evolution at `http://www.aireviewer.com`

Other tools (examples)

- Massey Architecture Explorer: Antipatterns Score (J. Dietrich et al. 2012), Tangledness metric (S. M. A. Shah, Dietrich, and McCartin 2012);
- Lattix: Stability, Cyclicality, and Coupling metrics;
- STAN: different R. Martin's metrics.

Input information

Information category	CAST	IF	SG	SQ	S101
Architectural Smells, e.g., (Lippert and Rook 2006, Garcia et al. 2009)	yes	yes	yes	no	yes
Code Smells (Fowler 1999, Lanza and Marinescu 2006)	no	yes	no	yes	no
Architecture/Design Metrics, e.g., (Martin 1995)	yes	no	no	no	yes
Code Metrics, e.g., (Chidamber and Kemerer 1994, Lanza and Marinescu 2006)	yes	no	no	yes	yes
Architectural Violations$^{\alpha}$	yes	no	yes	no	no
Coding Rule Violations$^{\beta}$	yes	no	no	yes	no

α: deviations from a reference architecture, i.e., unallowed dependencies

β: detected bad coding practices or excessive values of single metrics (some tools, e.g., SQ, internally refer to the latter as "smells")

Output information

	CAST	inFusion		Sonargraph		SonarQube		Structure101
TDI name →	TDP	QDI	SDI	SDC	TD	TDR	SR	XS
Resolution cost	yes	no	no	yes	yes	yes	yes	no
Keeping cost	no	yes	yes	no	no	no	no	yes
Unity Measure	US$	-	-	US$	Time	-	Rank	LOC

Q1: Used and generated information

As reported in the tables:

- Different TDI output approaches: abstract number, $ cost, time needed, LOC
- Extremely heterogenuous;
- Different terminologies (e.g., smell, metric violation);

but,

- Similar aggregation/drill-down strategy: single indicators are composed in a linear combination, with different weighting schemes;
- Association of costs or weigths to issues/indicators is arbitrary.

Q2: Architectural features

Observations:

- SonarQube ignores architectural information;
- SonarGraph, Structure 101, CAST are more oriented to architecture analysis and use it to integrate their TDIs.

Q3: Missing features

1. Keeping **and** Resolution costs (Principle and Interest);
2. Information unexploited: **no single piece of information** or analysis is supported by **all** tools;
3. Other existing information sources are not present, e.g., historical information
   - History is used only to track values over time

Thank you!

marco.zanoni@disco.unimib.it

References

Chidamber, Shyam R., and Chris F. Kemerer. 1994. "A Metrics Suite for Object Oriented Design." *IEEE Trans. Software Eng.* 20 (6): 476–93. doi:10.1109/32.295895.

Dietrich, Jens, Catherine McCartin, Ewan Tempero, and Syed M. Ali Shah. 2012. "On the Existence of High-Impact Refactoring Opportunities in Programs." In *Proc. 35th Australasian Comp. Sci. Conf. (Acsc'12)*. Australia: ACS.

Fowler, Martin. 1999. *Refactoring: Improving the Design of Existing Code*. Addison-Wesley.

Garcia, Joshua, Daniel Popescu, George Edwards, and Nenad Medvidovic. 2009. "Identifying Architectural Bad Smells." In *Proc. of the 13th Eur. Conf. on Software Maintenance and Reengineering (Csmr 2009)*, 255–58. Germany: IEEE. doi:10.1109/CSMR.2009.59.

Lanza, Michele, and Radu Marinescu. 2005. *Object-Oriented Metrics in Practice*. Springer.

Lippert, Martin, and Stephen Roock. 2006. *Refactoring in Large Software Projects: Performing Complex Restructurings Successfully*. Wiley.

Martin, Robert C. 1995. "Object Oriented Design Quality Metrics: An Analysis of Dependencies." *ROAD* 2 (3).

Shah, S. M. A., J. Dietrich, and C. McCartin. 2012. "Making Smart Moves to Untangle Programs." In *Proc. Csmr 2012*. Hungary: IEEE.