Measuring Sustainability Impact of Software

Mfon-Abasi Raphael Idio

Post Graduate Student (M.Sc. Information Technology), Srm University, Chennai.

Abstract

The quality of software architectures determines sustainability of the software to a large extent. Measuring sustainability of a software architecture both during early designs using scenarios and during evolution using scenarios and metrics, which is highly relevant in practice, have great impacts on software sustainability. Sustainability indicators and composite index are also used as powerful tools for policy making and public communication that enhanced software sustainability. Several initiatives exist on measuring sustainability of softwares in diverse fields of study. This article provides a general overview of measuring the impacts of software sustainability.

Keywords

Evolution, Indicators, Metrics, Scenarios, Software Architecture, Sustainability.

Introduction

Software systems with life span of more than 15 years must be carefully designed and implemented, taking notice of maintenance and evolution. Software architectures are major drivers for the sustainability (i.e., cost- efficient longevity) and evolvability of softwares because they influence how quickly and correctly a developer is able to understand, analyse, extend, test, and maintain a software system. According to ISO/IEC 42010 – 2007 definition, “Architecture is the fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution.” Evaluation methods [6, 1, 9] focus mainly on scenario-based methods to evaluate early software architecture designs and do not analyse their suitability for sustainability evaluation. Other surveys [2, 13, 4] provide more breath but do not include architecture-level metrics. An integration of scenario-based and metrics-based methods provide a continuous, pro-active approach towards evolution problem throughout the entire system life-cycle.

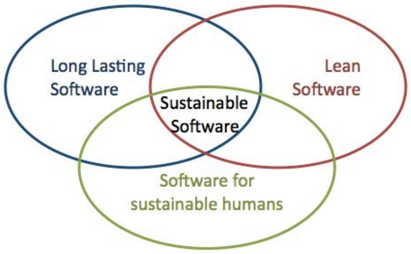
According to Ness *et al.*(2007) “ The purpose of sustainability assessment is to provide decision-makers with an evaluation of global to local integrated nature-society systems in short and long term perspectives in order to assist them to determine which actions should or should not be taken in an attempt to make society sustainable”

In more general terms, sustainability is the endurance of system and processes. The organizing principle for sustainability is suitable development, which includes the four interconnected domains: ecology, economics, politics, and culture. Various concepts are competing for the “sustainable software” name. One is about how well a piece of software will be able to cope with changes. In other words, the goal is to build “long lasting” software. This relates to qualities such as reliability, (self-) adaptability, maintainability or context-awareness of software,as well as development paradigms such as Agile.

The second is about direct environmental impacts of software, such as energy consumption and e-waste from computers made obsolete due to software upgrades. We could call it “Lean software”.

The third one is about indirect effects of software on the environment. In this sense a sustainable software is one that induces sustainable human behaviours. There are infinity of examples, as software has invaded all our activities, in daily life or business, and most have an impact on the way we behave. Consequently any piece of software should be designed in the awareness of its impact on human sustainability. For example, google maps ability to show public transportation routes towards your destination, recreational centers around your locality and lots more.

The diagram below illustrates the famous model for sustainability taking notice of the three concepts mentioned above.



Impacts sustainability's system is user-friendly and affordable system, allowing you to focus on the task at hand. i.e., creating a sustainable business.

A scenario is a brief description of a single interaction of a stakeholder with a system [5]. Scenario-based methods provide techniques for eliciting, documenting, and evaluating software architecture related scenarios against the requirements. Scenarios assessing the sustainability of an architecture are often called change, evolution or exploratory scenarios [3]. A change scenario may impact multiple components. Undesired and costly ripple effects [14] can occur if the change to a component causes changes in dependent components. Thus, loose coupling between system components is a desirable property of a sustainable architecture to avoid such effects.

A review of indicator sets and the categories of indicators are used by specific companies or countries to report sustainability performance. The lack of an open, neutral, inclusive and harmonized set of indicators and indices for sustainable manufacturing. For example, A sustainability measurement infrastructure is being developed by NIST [1].

Sustainability Impacts of Software

Sustainability of software has great impacts on our environment and behavior. A few impacts are enumerated below:

1. Sustainability development trend offers new business opportunity. For example, ICT industry can create new businesses by their environmental impact.

2. Sustainability enables ICT industry to find ways to solve environmental problems in other industries. This means seeking new solutions to reduce the environmental impact of software assets, reducing the impact of hardware through software throughout the full production life cycle.

3. The advances in computer hardware over the last few decades have been absolutely staggering and scientists have been aggressive about applying these advances to the process discovery.

4. Cloud computing has made available essentially limitless storage and computing power on demand.

5. The connection between individuals that have been enabled by the internet have accelerated scientific communication, enable new types of discourse, and provided the opportunity to directly transfer data and technology between researchers from diverse fields and backgrounds.

6. Most importantly, for younger researchers and those inexperienced in computation, high level languages and packages have enabled more direct development of tools, reducing the hurdles necessary to bring these developments into bear on a scientific problem.

Conclusion

This paper covers an overview of measuring various impacts of software sustainability. Although there are various international efforts on measuring sustainability, only few of them have an integral approach, taking into account environmental, economic and social aspects. In most cases the focus is on one of the three aspects. Indicators of sustainable development should be selected, revisited and refined based on the appropriate communities of interest. Thus, indices should be constructed within a coherent framework.

Future Work

Practitioners can use this review to tailor their own sustainability evaluation method based on the referenced methods and tools. Researchers can loopholes in the body of work and create systematic sustainability evaluation methods. This review identifies a need for more empirical studies on architecture evaluation.

Acknowledgment

The author would like to thank the Almighty God for making this dream come true and also her tutor: S. Kanchana for the inspiration.

References

[1] Babar, M. A. and Gorton, I. "Comparison of scenario-based software architecture evaluation methods." In *Proc. 11th Asia-Pacific Software Engineering Conf.*, APSEC '04, pages 600–607, Washington, DC, USA, 2004. IEEE Computer Society.

[2] Barcelos, R. and Travassos, G. "Evaluation approaches for software architectural documents: a systematic review". In *Ibero-American Workshop on Requirements Engineering and Software Environments (IDEAS'06)*, 2006.

[3] Bengtsson, P., Lassing, N., Bosch, J., and Vliet, H.V. "Architecture-level modifiability analysis (ALMA)." *Journal of Systems and Software*, 69(1-2):129–147, 2004.

[4] Breivold, H. and Crnkovic, I. "A Systematic Review on Architecting for Software Evolvability." In *21st Australian Software Engineering Conference*, pages 13–22. IEEE, 2010.

[5] Clements, P., Kazman, R. and Klein, M. "Evaluating software architectures: methods and case studies." Addison-Wesley Reading, MA, 2002.

[6] Dobrica, L. and Niemel, E. "A survey on software architecture analysis methods." *IEEE Trans. on Softw. Eng.*, 28(7):638–653, July 2002..

[7] Heiko, K. 2011. “Sustainability Evaluation of software Architectures: A systematic Review” Industrial Software Systems, ABB Corporate Research, Ladenburg, Germany.

[8] http://blogs./se.ac.uk/impactof socialsciences/2014/07/10/working-towards-sustainable-software-for-science/

[9] Kazman, R., Bass, L., Klein, M., Lattanze, T., and Northrop, L. M. “A basis for analyzing software architecture analysis methods.” *Software Quality Journal*, 13(4):329–355, 2005.

[10] Kocak, S. A., Alphtekin G. I., “The impact of Improving Software Functionality on Environmental Sustainability”. 2013. IBM Canada.

[11] Ness, B., Urbel-Piirsalu, E., Anderberg, S., Olsson, L. 2007. “Categorising tools FOS sustainability assessment.” Ecological Economics 60, 498 – 508.

[12] PenZenstadler B. “Towards a Definition of Sustainability in and for Software Engineering.” 2013. ACM New York, NY, USA.

[13] Roy, B. and Graham, T. “Methods for Evaluating Software Architecture: A Survey.” Technical Report 545, Queen’s University at Kingston, Ontario, Canada, Kingston, 2008.

[14] Schneidewind, N. F. “The state of software maintenance.” *IEEE Trans. Softw. Eng.*, 13:303–310, March 1987.

[15] Shaw, C. F., Che-Bung J., and Gang, L. 2010. “Development Overview of Sustainable Manufacturing Metrics” Manufacturing Engineering Laboratory, National Institute of Standards and Technology (NIST).

[16] Singh, R. K., Murty, H. R., Gupta S. K., Dikshit, A. K. 2012. “An Overview of Sustainability assessment Methodologies.” Elsevier ltd. Publishers India. http://www.sciencedirect.com/science/article/pii/s1470160x08000678