Sustainable software development framework

Fredrik Bengtsson

Uppsala University, Department of Informatics and Media
fredrik.bengtsson@im.uu.se

Abstract. The aim of this paper is to present a framework to assert how a software development process can be evaluated to reach sustainability improvements during this process. The role of information systems in sustainability initiative has gotten a lot of attention, both as a culprit by massive energy use, and as an enabler to reach a more sustainable future. Using a stakeholder perspective in regard to sustainability issues during software development activities, possible sustainability actions are proposed; corresponding to the intersection of the three sustainability areas of, *economical*, *social* and *environmental* goals, and the proposed four general software development activities, *planning*, *construction*, *testing*, and *implementation*.

Key words: Software development, sustainability, sustainable software development framework, sustainable IS

1 Introduction

Information systems (IS) are believed to be crucial in the effort to reach a more sustainable world (Elliot 2007; Melville 2010). Software development is one of the basic activities in the creation of IS. Both Elliot and Melville have proposed research questions, in regard to the role of IS, to reach a more sustainable future. The aim of this paper is to present a framework to assert how a software development process can be evaluated to reach sustainability improvements during this process. This would strengthen the claim that IS can be sustainable by widening the scope of potential sustainability actions derived from the use of IS, as well as showing that IS is more than an enabler of sustainability initiatives. Dependant on how this challenge is viewed it can be categorized as belonging to several overarching research questions; these will be further elaborated upon in the section on sustainability.

The aim of this paper falls into several of the research categories proposed by Elliot and Melville (ibid.), the most important reason for this is that software development is such a basic activity in regard to IS that it has an impact directly or indirectly on all undertakings that involves IS. This has two important implications: First, all activities that relies on an IS would benefit from a more sustainable development process from a life cycle perspective. Second, all the proposed research areas in sustainable information systems are still as valid and as important since they are on a more general level in regards to the development of IS.

A conceptual-analytical approach is used to develop the sustainable software development framework. That means that the resulting framework is based on previous empirical studies, theories, models and frameworks (Järvinen 2000).

This paper is structured as follows. The role of sustainability in regards to software development in an organizational setting is discussed, including the concept of sustainability. Next, the shareholder perspective and how it relates to sustainable system development and the software development activities is discussed. Different activities within software development are defined. Finally, the proposed framework for sustainable software development is presented, followed by conclusion and discussion.

2 Sustainability

Sustainability refers to the efforts of minimizing negative economical, environmental and social impacts of an activity (United Nations 2002).

Economical impact is related to direct economic values generated, risks and opportunities for activities caused by climate change, market presence and indirect economic impact.

Environmental impact is typically ecological aspects such as energy consumption, impacts of products and services, biodiversity and emissions.

Social impact covers interaction between the organization and its social context including community interaction, corruption issues and public policies.

As an environmental concern the growing amount of "e-waste" has been in the public eye for some time (Eliot 2007; Ogunseitan et al 2009). During the last few years, however, the focus in relation to information systems has been the energy use, and consequently, the carbon footprint of technology use (ibid.). This could be regarded as a quick fix with an easy target, namely data centers (Kurp 2008; Liu et al. 2009; Torres et al 2008; Kandlur and Keller 2009), and "greener" technology aimed at the consumer market. However, a deeper look at software development in order to facilitate a more sustainable development process would not only make better use of "greener" data centers, it would also facilitate "greener" or more sustainable software.

Sustainability reporting is the benchmark that stakeholders primarily use as the guideline to follow, since sustainability reporting is how an organisation measure and communicate their sustainability improvement. It is not feasible for an organisation, or any other entity, to adhere to all conceivable sustainability indicators. In order to frame the sustainability problem in a manageable way, key indicators are identified by which change in sustainability performance is measured. Although an organisation could make an effort to identify key indicators by their own, the norm is to follow sustainability reporting initiatives such as Global Reporting Initiative (GRI) (Global Reporting Initiative 2010), ISO 14000, AA 1000 from AccountAbility (AccountAbility 2010), etc, or less voluntary adherence to regulatory initiatives.

Sustainability linked to IS research is an emerging area spurred by concerns about climate change and an increased environmental awareness. Important steps to better understand and advance research in this area have been taken by Elliot (2007) and Melville (2010) by proposing relevant research questions in regards to the role of IS in relation to sustainability.

Despite the fact that Elliot and Melville use different methodologies to identify their proposed research questions, the questions are very similar on a conceptual level. Elliot presents a taxonomy

based on Barki et al.'s (1993) IS research categories, extended to incorporate ICT sustainability. Melville has based his research questions by correlating IS research questions in a sustainable setting too the Belief-Action-Outcome (BOA) framework for research on sustainability. The framework depicts the links between believes in an organizational or societal structure on a macro level, via individuals on a micro level or directly to behaviour on macro level in social systems or organizations. This behaviour leads to actions and subsequent outcomes.

Examples from Elliot (2007), named *Taxonomy for Environmental Sustainability of ICT* :

- External environment: economic, legal, political and social.
- IT hardware and software.
- Organizational environment, including characteristics, functions and tasks.
- IS management, including hardware, software, personnel, projects, planning, evaluation and other management issues.
- IS development and operations, life cycles, IS development, implementation and operations.

In total Elliot's taxonomy consists of eight topics of sustainability ICT research. Sustainable software development as a research question does fall into all of these at some level. The five examples listed above could all be applicable since they all touch on questions relevant during the software development life cycle. Those topics that are omitted from the list are relevant but to somewhat lesser degree.

Examples from Melville (2010), based on the *Belief-Action-Outcome (BOA) Framework for IS Research on Sustainability*:

- How can different research methodologies, such as life cycle analysis and integrated assessment, be applied to examine complex problems involving information systems, organizations, and the environment?
- What is the impact of information systems on beliefs about the natural environment and environmental sustainability?
- What design approaches are effective for developing information systems that influence human actions about the natural environment?
- What is the association between information systems and organizational and sustainability performance?

In contrast to Elliot, Melville has chosen to present his proposed research topics as direct questions for researchers to explore. In total, Melville put forward ten research questions where two are divided into a) and b) giving the total number of twelve. These are in turn divided amongst five different domains; philosophical perspective and theory, research methodology and data sources, belief, action, and finally outcome. Except for the first domain the others are represented with one example from each domain in order, and are selected based on relevance to questions regarding the process of software development. As with the examples from Elliot's taxonomy these examples are put forward to show the wide span of research questions that could be investigated based on sustainable software development.

In relation to the proposed research questions, software development touches on several of these, but does not, due to its fundamental nature within IS answers up to any specific subset of research questions. Furthermore, the software development life cycle is fundamental to IS since it reflects the process by which the software is created and as a consequence the physical infrastructure to support it.

3 Stakeholder

According to Fassin (2008); "A *stakeholder* refers to any individual or group that maintains a *stake* in an organization in the way that a shareholder possesses a share". The notion of a wide range of potential parties that are affected by organizations doings are shared by the concept of triple bottom line accounting. Traditionally the success of a corporation is measured by the financial bottom line or earnings of the corporation. In triple bottom line accounting the idea is that a successful or healthy corporation should also take social and environmental performance into account (Norman & MacDonald 2004; Gray 2006) based on John Elkington (1997; 2001), thus, shifting from a *shareholder* view to a *stakeholder* view. And it is the idea of a triple bottom line and a multitude of stakeholders that is the foundation for sustainability reporting, and in practice this idea has had and still has an impact amongst companies as well as legislators (Norman and MacDonald 2004; Gray 2006). However, both Gray, and Norman and MacDonald are critical, not to the potential good that can result from a strive to a more sustainable future, but to the usability of sustainability reporting in Gray's case and triple bottom line in the case of Norman and MacDonald.

Mitchell et al. (1997) presented a typology that identifies eight different types of stakeholders, based on three attributes; power, legitimacy and urgency. This model was later confirmed by Agel et al. (1999). Power is basically to what degree a party in a relationship can impose its will, regardless of means used to achieve it. Legitimacy represents what is conceived as desirable social good. Finally, urgency is the degree to which a stakeholder demand for immediate attention based on for example time constraints, perceived importance, etc. A stakeholder may possess one, two or three of this attributes. The number of attributes present in a stakeholder determines the stakeholder's salience, with more attributes present in a stakeholder the more salient it is. Furthermore, a stakeholder can lose or gain attributes and the extent to which a stakeholder can act or influence based on this attributes can change (ibid.). The eight different types of stakeholders are categorized as depicted in figure 1 based on the classes of attributes it possesses:

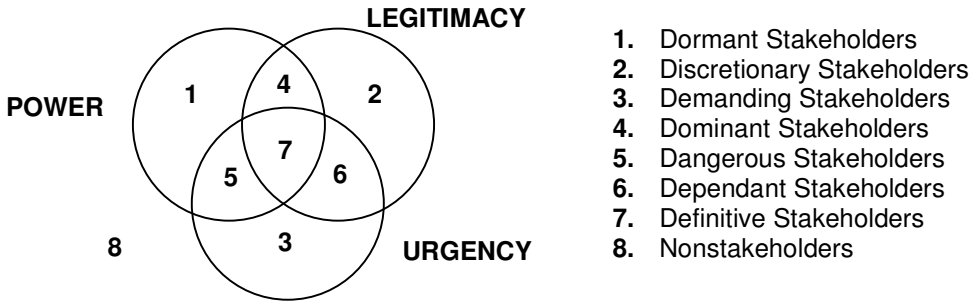

Figure 1: Stakeholder typology (Mitchell et al. 1997)

In order from least salient to most salient stakeholder:

Nonstakeholders are stakeholders that do not have any power over the activities, any legitimate concern, or express any urgency concerning the activity.
Dormant, **Discretionary**, and **Demanding** stakeholders have one of the three attributes thus salience will be low. *Dormant* stakeholders have power they could assert on the activity but little reason to do so. *Discretionary* stakeholders are affected by the activity but do not have any power

over it, and no urgent claims. *Demanding* stakeholders have no power or legitimacy in regards to the activity, but still feel a need to express their view on the activity.
Dominant, **Dangerous**, **Dependant** stakeholders have two of the three attributes. *Dominant* stakeholders are an important group with both power and legitimacy and could thereby have a great impact on the activity. *Dangerous* stakeholders lack legitimacy but have power and urgency, representatives of this group are usually not under the influence of those with legitimate claims to the activity and there by a potential risk to the activity. *Dependant* stakeholders have both legitimate and urgent claims but lack the power to influence the activity directly.
Definitive stakeholders have all the attributes, power, legitimacy and urgency. These stakeholders are often represented by a dominant stakeholder that expresses a need for urgency, thereby stepping forward as a definitive stakeholder and as result being the most prioritized stakeholder in regards to the activity. Dependant and potentially dangerous stakeholder could also become a definitive stakeholder by accruing the missing attribute, power in the first case and legitimacy in the second case.

To know how to be more sustainable we have to know how an activity affects its surroundings, to know this we have to identify the stakeholders involved in order to identify how they can change and/or influence an activity in order to be more sustainable.

The purpose of sustainability reporting is to show that there is a progress towards a more sustainable state. From a triple bottom line and stakeholder perspective this is to track sustainability changes by indicators that affect stakeholders that do not have the means to influence the activity themselves, e.g. dependant and demanding stakeholders, but also to be vigilant for nonstakeholders that potentially could acquire stakeholder attributes.

According to this stakeholder view combined with a sustainability outset, to meet the needs of affected stakeholders, dominant and definitive stakeholders are those that easiest can affect the software development life cycle. These stakeholders assert power and can thereby influence how things are done. Enforcing policies, deciding which key sustainability indicators that should be prioritised and which indicators that are less feasible or should be ignored all together. Who these stakeholders are will change during the development process, and the same stakeholder can gain or lose attributes during different activities. Furthermore, power, urgency, and legitimacy are all variable; a person can still be dominant, but someone else can step up and be even more so.

4 Software development activities

There are a multitude of software development methodologies. All of these methodologies are perceived as enabling a path to follow in order to produce the desired software. In retrospect, they all have strong points as well as weaknesses, thus the invention of new methods to counter flaws in previous methods or to meet new demands. Nevertheless, they share the same purpose, and as a consequence can be divided into a few generic activities.

In order to identify the generic activities during the software development life cycle a literature review was done (cf. e.g. Galin 2004; Sommerville 2004; Budgen 2003; Pfleeger and Atlee 2006; Maciaszek 2007; CTG.MFA-003 1998; Abrahamsson et al., 2002). Each model was then analyzed to identify the constituent development phases. Based on these phases, four activities could be identified, categorized as follows; planning, construction, testing, and implementation.

Planning: Activities that precedes construction, this does not including the customers (internal or external) activities before they become a part of the development process. However, customers are often involved in the planning activities during the development.
Construction: Typical activities are design, coding and other activities tied to the actual production of the software, based on the deliverables, or equivalent, from the planning activities.
Testing: Activities that involves testing, evaluation and refinement of artifacts created during construction activities. Testing and construction activities are often closely linked in an iterative manner.
Implementation: The final activity when the developed software is installed or by other means are handed over to be used by the customer. In some cases implementation activities are linked to testing and construction.

Activity / **Model**	*Planning*	*Construction*	*Testing*	*Implementation*
Waterfall Model	• Requirement definition • Analysis	• Design • Coding	• Testing	• Installation and conversion • Regular operation and maintenance
Prototyping	• Requirement definition	• Design • Prototyping • Creation Modification	• Assessment • Refinement	• System implementation • Maintenance
Exploratory Model	• Initial specification development	• System construction/ modification	• System test	• Implementation
Spiral Model	• Planning • Risk assessment/ analysis	• Design • Construction	• Customer evaluation	• Implementation
Reuse Model	• Requirements • Definition of objects	• Collection of objects • Coding new objects • Assembly	• Evaluation	• Delivery
Object-Oriented Model	• Requirement definition • Object-oriented analysis	• Object-oriented design • Reusability survey of components library • System construction	• System tests • Customer's evaluation	• Installation and conversion • Operation and maintenance
Agile Model	• Requirement specification • Design	• Code	• Unit test • Integration test • System test	• Maintenance • Implementation

Table 1: Activities in software development life cycle models.

It should be noted that the exact naming of the phases might differ within each model dependant on the explicit method used; the naming in *table 1* should be viewed as an abstraction to encompass the different wordings used by the methods belonging to a specific model. Furthermore, some of the models include partly or wholly iterative methods; this iterative nature is not depicted in the table.

During the software development lifecycle many different stakeholders are represented. For each activity, several stakeholders can be identified, who or what they represent depend on the actual development case, methods applied, parties involved, domestic and international legislations applicable, the state of sustainability adherence, etc. In some way or another they have an interest in the outcome or success of the undertaking.

By identifying stakeholder that *can* influence sustainability outcomes, a focal point for efforts that can influence activities that has sustainability impact is also identidied. Furthermore, by identifying these stakeholders, it will be possible to adapt if circumstances change that calls for new strategies to ensure that the sustainability goals will be met, even if the distribution of sustainability impact from sub-activities are different.

5 Sustainable software development

The challenge when trying to identify factors and activities that can contribute to a more sustainable software development is that the development process is built up by many different activities, and these activities are often carried out or influenced by many different stakeholders. From a managerial point of view, it is the claims of these stakeholders that the managers have to evaluate against each other as well as how their claims affect both the development process and sustainability change. The purpose of the proposed framework is to aid in the identification of sustainability actions based on stakeholder activities in relation to sustainability goals *figure 2*. By identifying stakeholders that are either in a position to impose change in a more sustainable direction or stakeholders whose influence over the development enforces a more sustainable attitude, sustainability actions can be identified and implemented.

		Software development activities			
		Planning	*Construction*	*Testing*	*Implementation*
Sustainability areas	Economical				
	Environmental				
	Social				

Figure 2: Sustainability action framework.

Within each of the four software development activities, potential sustainability actions can be identified, correlated to each of the three sustainability areas. These actions can broadly be defined as: (1) Binding actions governed by legislation, these should ideally already be taken into

consideration, but can potentially still be improved. (2) Strategic actions, such as environmental branding and sustainability reporting, these are self governed actions. (3) Practical actions, these are actions taken to satisfy claims from primarily dependant and demanding stakeholders to forestall future claims backed by more influential stakeholder. Actions taken to appease discretionary and even dangerous stakeholders would also fall into this category.

Similarly, by examining each paring of development activity and sustainability area, relevant stakeholders can be identified. In order to facilitate sustainability actions relevant stakeholders has to advocate change. In accordance with Mitchell et al (1997), dominant and definitive stakeholders are those that are in a strong position to implement change. Dormant stakeholders have power and could potentially be influenced to support change. Other groups of stakeholders are less likely to be able to directly influence and support sustainability actions, although they may be significant for other reasons as discussed above. An important factor to relate to is the willingness of a stakeholder to advocate change, and in the extreme case stakeholders that counteracts sustainability actions.

Important stakeholders to a software development project depend on a variety of factors. The development method used can influence both the number of stakeholders and the importance of stakeholders, e.g. users (Kujala 2003), the organizational structure and size of the development firm as well as the customer influence relevant stakeholders, where the involved parties are located in the world and compared to each other are relevant to identify important stakeholders, etc.

The purpose of the framework is to identify activities that can be made more sustainable from a sustainability onset. In a specific case this requires that; (1) the sustainability indicators in the chosen sustainability reporting model is known and understood by the managers, (2) relevant activities in regards to the sustainability indicators can be identified within the chosen development method, and (3) the managers can identify relevant stakeholders in regards to the activities that can advocate the change needed.

In *figure 3* some possible actions with positive sustainability change are shown. These changes are based on two conditions: First, a set of possible sustainability actions are proposed e.g. based on a sustainability reporting scheme. Second, the identified stakeholders that have claims on the project and results of the project, e.g. development managers, programmers, the customer, and end users are willing to align their claims with sustainability actions. From this a subset of sustainability actions are agreed upon.

		Software development activities			
		Planning	*Construction*	*Testing*	*Implementation*
Sustainability areas	Economical	E-meetings to avoid unnecessary cost due to travelling. **1**	**2**	**3**	Install on virtual servers to minimize hardware and energy cost. **4**
	Environmental	E-meetings to avoid unnecessary emissions due to travelling. **5**	Virtualizing the development environment to minimize CO_2 emissions during the construction. **6**	Set up remote testing environments to minimize travel. **7**	**8**
	Social	Involve end users in planning activities. **9**	**10**	Involve end users in testing. **11**	Organize training in the use of the new software. **12**

Figure 3: Sustainability actions during software development.

Lets consider an example, result shown in *figure 3*, not all of the stakeholders involved in the project where located in close proximity to each other, thus e-meetings was proposed as an alternative to physical meetings during some planning activities, this had the benefit of contributing to both economical and environmental goals (*cell 1 and 5 in figure 3*). End user involvement could contribute to social goals during planning, testing, and implementation activities by emphasizing accepts important to them and thereby, hopefully, the adoption and acceptance of the software *(cell 9, 11 and 12)*. Since the idea of managing some of the planning activities remotely it was also deemed feasible to set up a remote testing environment thus adding an environmental goal to the testing activities *(cell 7)*. Finally, the customer proposed to install the software in a virtual server environment to save cost on new hardware *(cell 4)*. The empty cells represent that no feasible sustainability action where identified. For instance, the virtualizations proposed (*cell 6 and 4*) could influence economical and environmental goals respectively. But due to the existing IT-infrastructure no such goals were deemed feasible.

The presented example is deliberately somewhat simplistic but it shows that it is possible to ascribe sustainability actions and subsequent change to the software development process. In a real world example there would most likely exist several sustainability actions corresponding to the different cells. This example also lacks the added complexity of weighing positive sustainability actions versus negative sustainability actions.

6 Conclusion

To be more sustainable is to promote and enforce change that has an effect compared to previous comparable activities. The proposed framework supports the identification of such actions. With the conclusions drawn in this paper the next step for further research would be to apply the framework in a real life case to test its applicability in a practical setting. Furthermore, this would aid in

identifying weaknesses in the framework that need more rigorous grounding in both theory and practice.

The aim of this paper was to present a framework to assert how a software development process can be evaluated to reach sustainability improvements during this process. From this onset, this paper argues that the software development life cycle can be analyzed to identify actions to make this process more sustainable. Furthermore, this paper argues that, stakeholders that can advocate and champion sustainability change are vital to these efforts. Hitherto, it has been IS in use as an enabler for sustainability actions, and IS use of hardware and energy that has caught the attention of scholars and the public. By considering the sustainability impact of software development, this stage in an IS lifecycle can be incorporated into a sustainability life cycle analysis of IS.

References

Abrahamsson, P., Salo, O., Ronkainen, J. and Warsta, J. (2002). *Agile software development meyhods Review and Analysis*, VTT Technical Research Center of Finland, Oulu

AccountAbility (2010) [online], available: http://www. accountability.org [accessed 06 April 2010]

Agel, B. R., Mitchell, R. K., and Sonnenfield, J. A. (1999). *Who matters to CEOs? An investigation into stakeholder attributes and salience, corporate performance and CEO values*, Academy of Management Journal, 42(5): 507-525

Barki, H., Rivard, S. and Talbot, J. (1993). *A keyword classification scheme for IS research literature: An update*, MIS Quarterly, 17(2): 209-226

Budgen, D. (2003). *Software Design*, Second edition, Pearson Education Limitid, Harlow.

CTG.MFA-003 (1998). *A survey of Systems Development Process Models, Center for Technology in Government*, University at Albany/SUNY

Elkington, J. (1997). *Cannibals with Forks: The Triple Bottom Line of 21st Century Business*, Capstone Publishing, Oxford.

Elkington, J. (2001). *The Chrysalis Economy: How Citizen CEOs and Corporations Can Fuse Values and Value Creation*, Capstone Publishing, Oxford.

Elliot, S. (2007). *Environmental Sustainable ICT: A Critical Topic for IS Research?*, in F. Tan and J Thong (eds) Proceedings of PACIS 2007, AIS, 4-7 July, Auckland New Zealand

Galin, D. (2004). *Software Quality Assurance From Theory to implementation*, Pearson Education Limitid, Harlow.

Global Reporting Initiative (2010) [online], available: http://www.globalreporting.org [accessed 06 April 2010]

Gray, R. (2006). *Social, environmental and sustainable reporting and organisational value creation? Whose value? Whose creation?*, Accounting, Auditing & Accountability Journal, 19(6): 793-819

Järvinen, P.H. (2000). *Research questions guiding selection of an appropriate research method.* In Proceedings of the 8th European Conference on Information Systems, Vienna, 124-131

Kujala, S. (2003). *User involvement: a review of the benefits and challenges*, Behaviour & Information Technology, 22(1): 1-16.

Kandlur, D. D. and Keller, T. W. (2009). *Green Data Centers and Hot Chips*, DAC'09, July 26-31, 2009, San Francisco, California, USA

Kurp, P. (2008). *Green Computing*, Communications of the ACM, 51(10): 11-13

Liu, L., Wang, H., Liu, X., Jin, X., He, W., Wang, Q. and Chen, Y. (2009). *GreenCloud: A New Architecture for Green Data Center*, ICAC-INDST'09, June 16, 2009, Barcelona, Spain.

Melville, N. P. (2010). *Information Systems Innovation for Environmental Sustainability*, MIS Quarterly, 34(1): 1-21

Mitchell, R., Agle, B. and Wood, D. (1997). *Towards a theory of stakeholder identification and slience: defining the principle of who and what really counts*, Academy of Management Review, 22(4) pp 853-886

Norman, W. and MacDonald, C. (2004). *Getting to the bottom of "Triple bottom line",* Business Ethics Quarterly, 14(2): 243-262

Maciaszek, L. A. (2007). *Requirements analysis and system design*, Third edition, Pearson Education Limited, Harlow

Ogunseitan, O. A., Schoenung, J. M., Saphores, J-D. M. and Shapiro, A. A. (2009). *The Electronics Revolution: From E-Wonderland to E-Wasteland*, Science, Vol. 326. no. 5953: 670 – 671

Pfleeger, S. L. and Atlee, J. M. (2006). *Software Engineering Theory And Practice*, Third edition, Pearson Prentice Hall, Upper Saddle River.

Sommerville, I. (2004) *Software Engineering*, Seventh edition, Pearson Education Limited, Harlow

Torres, j., Carrera, D., Hogan, K., Gavaldà, R., Vicenç, B. and Poggi, N. (2008). *Reducing wasted resources to help achieve green data centers. 22nd IEEE International Symposium on Parallel and Distributed Processing*, IPDPS 2008, Miami, Florida USA, April 14-18, 2008

United Nations, (2002). *Report of the World Summit on Sustainable Development*, Conference Publication 199/20