Blinded by Simplicity: Locating the Social Dimension in Software Development Process Literature

Blinded by Simplicity*

Locating the Social Dimension in Software Development Process Literature

Johanna Liz Gustavsson†
Computer and Systems Science
Stockholm University
Sweden
Johanna.liz@live.se

Birgit Penzenstadler
Computer Science and Engineering
Chalmers | Gothenburg University, Sweden
Lappenranta Univ. of Techn., Finland
birgitp@chalmers.se

ABSTRACT

The software development process is a complex human, intellectual and labor-intensive activity and human related factors have shown to be the most significant contributors to software system failures. Lacking the ability to identify or quantify these factors, software practitioners will not learn from the failures caused by them. Although, social factors give rise to high failure rates in software development projects they tend to be ignored. Business continues as usual. The inability for software engineers to attain a holistic and inclusive approach will leave the social dimension out and undermine the realization of a fully sustainable software development process.

This paper builds on the master's thesis with the same title completed in December 2019 at Stockholm University. The thesis demonstrates how research literature on software development processes addresses (or not) the social dimension of sustainability from a holistic point of view. The results indicate that the practice of dealing holistically with complexity including the social dimension is still underdeveloped. Further research is suggested regarding the development of adequate supporting tools, social skills, and managerial attitudes and behaviors.

CCS CONCEPTS

• Software and its engineering ~ Software organization and properties • Social and professional topics

*Article Title Footnote needs to be captured as Title Note
†Author Footnote to be captured as Author Note

ICT4S, June,2020, Bristol, UK

https://doi.org/10.1145/1234567890

KEYWORDS

Systematic literature review, Software development process, sustainable computing, computational thinking, social sustainability, holistic approach, human, social and intangible factors, paradigm shift, transformation, failures, complexity, behavior and attitude change, ethics, biases.

ACM Reference format:

Johanna Liz Gustavsson and Birgit Penzenstadler. 2020. Blinded by Simplicity: Locating the Social Dimension in Software Development Process Literature. In *Proceedings of the 7th ACM ICT4Sustainability conference (ICT4S2020). ACM, Bristol, UK, 2 pages.* https://doi.org/10.1145/1234567890

1 Introduction

"We take the view that all design has an impact on sustainability and all software has an impact on the world. Therefore, it is the responsibility of those who are involved in the creation of software to consider this impact carefully" [1]

Sustainable computing is an emerging research field in computer and systems science (CSS). It draws attention to the wider impact software systems have both within and beyond the software community [1] [2]. Here, sustainable computing denotes the act of responsibility when designing algorithms and developing systems within the software development process (SDP), i.e., how to produce software systems that are predictable over time and guided by what is morally right [3]. This should not be confused with sustainable Human-Computer-Interaction (HCI) research, where HCI is a large interdisciplinary research field focusing on the end users' computing satisfaction seen through the interaction, or dialogue, with a computer [4]. Being responsible refers to the degree software engineers adhere to their code of ethics, facilitated by both ACM [40] and IEEE [41], and to protect the public.

CSS carries the ability to provide us with technical solutions that can enable and facilitate us in our quest for a sustainable future. Still, researchers point at our limited knowledge about the social and ethical consequences these abilities hold [1]. The software development process (SDP), which is the focus of this paper, is traditionally perceived as a pure engineering research belonging to the 'technical' school where social and ethical considerations have been neglected. As will be argued, this view is highly unfortunate and needs to be challenged.

When elaborating on sustainability, the most frequently used definition is the one presented in the Brundtland Report [5] saying that the needs of the present should be met without compromising the ability of future generations to meet their own needs. The focus of [5] is sustainable development, i.e. what we need to do in order to attain the state of sustainability. The concept sustainability is not seldom used in short for sustainable development. In CSS the concept sustainability is often used to express purely technical aspects, such as durability, maintenance or serviceability. These aspects are traditionally not dealt with holistically, including social and ecological perspectives. In this paper sustainability refers to the holistic and inclusive process of balancing the three dimensions of ecology, social aspects and economics in order to attain a sustainable state. Balancing the complex interrelatedness between these three dimensions requires a systemic and holistic approach. The aim of this study is to investigate how the social dimension is addressed and holistically interrelated in SDP literature.

Many researchers have found it difficult to find definitive formulations for the concept of sustainability. A suggested reason for this is the wicked, or fluid, nature of this concept [6]. It is persisting and subjected to constant redefinition and resolution in different ways over time [7][8]. Due to the paradoxes, changing requirement and complex interdependencies that wicked problems hold, they cannot easily be modeled by traditional scientific approaches [9]. The difficulty of dealing with complexity within the computational field can explain the inability to agree on a cohesive and explicit definition of sustainability [1][2][8]. According to [8], this inability causes fragmentation of the understanding of the concept and is thereby becoming a source of uncertainty itself,

The inability to define and conceptualize wicked and complex problems, like sustainability, tend to divide CSS into two worldviews:

- *Computational thinking* refers to the conceptual toolkit used by computer professionals for problem-solving. They look for algorithmic solutions to problems, in terms of data manipulation and process control [7]. The concept will be elaborated on further in the next section.
- *Holistic Thinking*, in opposition to computational thinking, acknowledges the complexity that computer systems are embedded in. When applying a holistic approach, it becomes important to understand how social contexts, involving normative and ethical values, produce biases that influence software computing [1][9].

As will be discussed in the next section, the complex SDP is a mixture of technology, people, and management systems, still, CSS research on sustainability is not yet able to attain a holistic and inclusive approach [1][7][8][9]. The tendency is to concentrate mainly on the ecological and economic dimension of sustainability, leaving out the social aspects (also observed by [2]). According to [10], the social dimension of sustainability is devalued in the SDP. He stresses the fact that for the last 40 years' research on "software psychology" has had a minimal impact on software engineer professionals. Further, he argues that when examining the major software system failures, the larger part of the reasons eventually narrows down to human factors.

This paper builds on the master thesis [39] where the research problem raised is twofold: social factors that give rise to high failure rates of software development projects tend to be ignored, due to the challenges involved; and the inability to attain a holistic and inclusive approach will leave the social dimension out and undermine the realization of a sustainable software development process.

The research question addressing these problems is the following: *How does software development process literature address the social dimension of sustainability?*

The next section provides background information. Section III covers research strategy and outcomes; section IV presents a summary of the results together with key findings. Section V holds discussion and limitations, and lastly, section VI offer concluding reflections.

2 Background

2.1 The Complex Software Development Process

Software systems and products are not only developed to assist us in making decisions in a complex environment, but they are also complex enterprises in themselves. Many things can go wrong. The social dimension of sustainability constitutes an intangible space that influences software engineering and gives rise to software failures[1]. Each year enormous amounts of money are wasted on failed software projects with failure rates up to 85 % [12]. The research of Cerpa and Verner [12] shows that 5-15 % of the projects are abandoned before or shortly after delivery, proving inadequate. Despite extensive literature and research for the last 40 years' software development projects still fail for the same reasons, e.g. inadequate requirements and staff, de-motivating management, and adding staff late in the process. They are all examples of human factors relating to the social dimension. According to [12] there is

[1] In this paper failure is referring to failing to deliver stable software products that correspond to end users' needs and expectations, due to social factors. The level of success is correspondingly understood to refer to the ability to reduce failure rates (For further reading on SDP failures [11][12][13][14][45] are suggested).

a perception that software quality is not improving but getting worse.

The complexity of the software development processes (SDP) depends on many structural and managerial factors. From a managerial perspective the overall development process can be divided into five central components that consume the project: time, resources, money, scope, and quality [15], see figure 1.

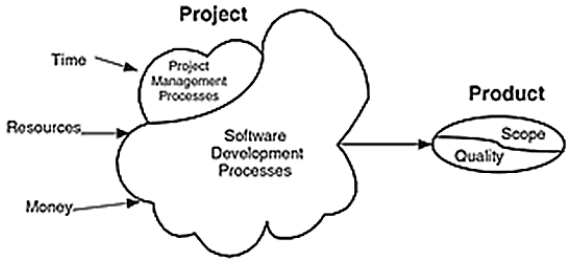

Figure 1. A holistic interpretation [15]

Figure 1. also illustrates the two main processes needed to produce a software product, project management (10-20 % of the whole project, defining, planning and controlling) and the development process (the actual work being done).

Software development is all about the process. It is the fundament that all software projects are based on. The first software process model known, the stage-wise approach (later called the Waterfall model), was defined in 1956 by Herbet D Bennington. Since the 50's, process models have evolved to handle the growing complexity of projects dealing with more dynamic problem domains [15]. [15] identify four main categories of process models that capture the quintessence of all existing models. In table 1. they are organized in a matrix divided into high and low uncertainty of requirement respective complexity of system: incremental models are suited for processes with low uncertainty and high complexity (e.g. the Cleanroom approach); the conventional models for processes with both low uncertainty and low complexity (e.g. the Waterfall approach); the "throw-away prototype" models for processes with high uncertainty and high complexity (also called close-ended prototyping); and fourthly the evolutionary process models suited for high uncertainty and low complexity (e.g. Rapid Application Development and Agile approaches).

		Uncertainty of requirements	
		Low	High
Complexity of system	High	Incremental	Throw-away prototype
	Low	Conventional	Evolutionary

Table 1. Selecting appropriate process models [15]

A contextual approach to software development further discloses the complexity involved in the process. Although the literature on the complexity of software development processes is vast, less attention has been given the situational contexts in which the processes are embedded [9]. The continuously changing situations in the different stages of the development process make the attempt to harmonize a process with its context discomforting since it is beyond our ability to fully control [16]. Before deciding on the most appropriate process model to adopt to a software project, developers need to take a wide range of contextual factors into account. To aid software developers in keeping track of all possible contextual factors, [17] have taken the initiative to the Situational Factors Reference Framework (figure 2.). The framework arranges 44 contextual factors, which are linked to 157 sub-factors, factors known to affect the software development process.

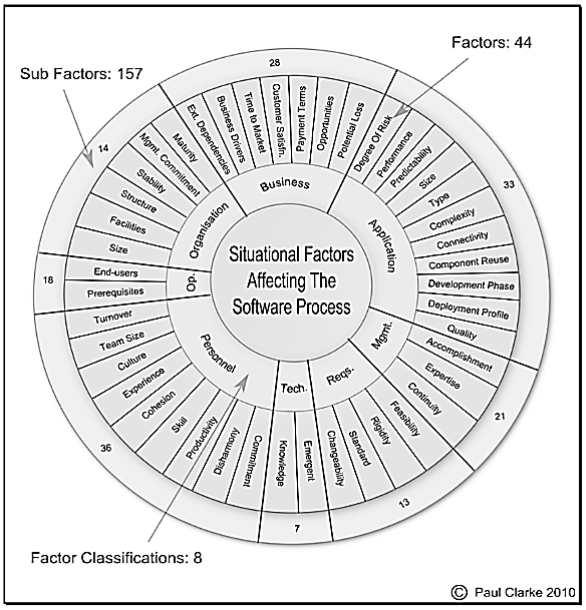

Figure 2. Situational factors affecting the SDP [17]

Given the fundamentals and the prerequisites needed for a successful SDP the social dimension is still not considered. Yet "a successful software project is the result of a complex process involving, above all, people" [18]. The complex process of contemporary software development involves many stakeholders, e.g. product owners, project managers, quality assurance teams, and software developers. [10] emphases the important role human-related factors play in software development, such as selection of the right people for a team. Emotions, moods, and feelings are other factors that have shown correlating with individual task progression according to him.

Social factors are closely linked to ethical and normative considerations. Decisions made by individuals on one level will affect people on other levels or induce a situational change in different layers. To take responsibility for the consequences one's actions afflict on other stakeholders in the development process is

as an important perspective on complexity as the previously mentioned ones. [3] stresses the need for elaborated and more specified ethical codes of conduct to aid each software engineer to become aware of the ethical impact of their decisions through the whole SDP. Figure 3. illustrates the interdependency between different social layers affecting, and likewise being affected by the SDP. [3] states that software engineering has been ascribed bad reputation regarding the avoidance of taking responsibility for failed software systems in the past, playing "the Blame Game". He stresses the "importance that software engineers behave in a way that upholds their profession. They have to be professionally responsible, adhere to their code of ethics and protect the public" [3].

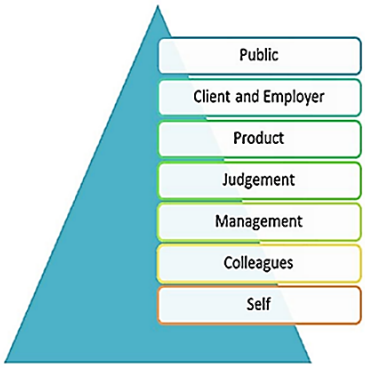

Figure 3. The software engineering code of ethics pyramid [3]

2.2 Computational Thinking - the Root Problem

Computational Thinking (CT) is a fundamentally reductionistic approach. From this perspective technology is perceived as a neutral phenomenon, derived from inputs and outputs in the computer systems. Computational problems are tackled by reducing them to a set of discrete variables that can be mapped onto abstract data types, and to a set of algorithmic steps for manipulating these types of data. In the process, multiple perspectives on the nature of the problem are lost. CT ignores the fact that any expressions of "the problem to be solved" is the result of an ongoing negotiation between the competing needs of a variety of stakeholders. Translating problems into components that can be solved with computers, establish a selective lens through which the world is viewed. From this follows that problems, not easily decomposable (e.g. ethical dilemmas, value judgements, societal change, etc.) will be neglected. This neglect will lead to practices that undermine sustainability [7].

Easterbrook [7] continues saying that while attempting to solve all problems through algorithmic means computer professionals may fail to perceive those problems that cannot be expressed using abstractions of CT. Oversimplifying complex social problems computer professionals nurture an impoverished approach when dealing with such complex matters as sustainability. Building for sustainable computing, or sustainable software use in general, both [7] and [9] stress the need to deal more critically with the reductionism cultivated in the computational field.

One of the main challenges in establishing a sustainable approach within computational sciences is the grounded positivistic traditions of CT [7]. The tendency to divide the world into problem types and solution types stands in sharp contrast to the need for embracing complexity and an interdisciplinary, or holistic, mindset [9][19]. Sustainability demands a more systemic approach to the tight interrelations between technology, human behavior, and environmental impacts. It risks being undermined when practices are based on problems reduced to their computational components. Still, as was mentioned earlier, trying to define sustainability may result in unsustainable outcomes, caused by the locking-in effects of definition, and thereby failing to address the complexity

2.3 Sustainable Computing and the Holistic Lens

Sustainability research is lacking, and even absent in several domains of CSS [7][9][14]. The research on sustainability that does exist tends to focus solely on ecology and economy, leaving the third pillar of sustainability out, the social dimension. With its many dimension's sustainability transcends multiple disciplines and require cross-disciplinary expertise [1]. A precondition for acting sustainable is the ability to put on a "holistic lens", allowing reflection and the inclusion of different perspectives. CSS may be a multidisciplinary field but acting sustainable and applying an interdisciplinary mindset is something completely different. [19] argues that to be able to base proposals on approaches and dialogue between different disciplines, you need to involve complex thinking, which enables the inclusion of different perspectives, connecting and discussing them. What characterizes this kind of thinking is the inclusion of different variables, instead of reducing phenomena to what is known and previously defined. It offers more than a limited understanding of the current phenomena of production, it enables people "to envision a future not limited to dual relationships" [19].

Based on human culture and society, sustainability is a normative concept, including aspects as justice, equality, and responsibility. [9] argues that Information and Communication Technology (ICT) is part of the cultural context and thereby also embeds normative values. The main problem with most of the research on sustainable computing is the neglect of these values [20]. The literature lifts several obstacles to a fully inclusive and holistic approach within the CSS. One of the central obstacles, which both [9] and [3] mention, is that focus most of the time is directed to the calculated impact of technology, or on ICT as a tool solving practical problems. ICT is thus treated as a neutral system to be used or studied. Yet, [9] states, if computers are seen as mere tools without accounting for the biases embedded in the cultural context and history from where computing departs, problems and blind spots will appear, e.g. the Heider-Simmel Illusion [48] illustrates how simple shapes are moving in a supposedly random manner, when

in fact they are neither value neutral nor moving randomly[2]. The misconception regarding computing as a strict problem-solving activity, free from ethical deliberations, reinforces the perception that computing is an ethically neutral practice, according to [3]. Software engineers are taught that logic equals problem-solving, not the calculating of the impact probable consequences of software design may cause. This attitude hinders computer professionals' ability to contemplate on possible solutions beyond what is specified by the system.

According to [1], sustainability problems are usually dilemmas to be addressed, not problems to be solved. They remind us that "success may be a moving target", and stress the need for conceptual models, techniques, and tools that support us in communicating, representing, and visualizing relationships between software, systems, and particular aspects of sustainability in their social, economic, and natural environment. In an attempt to address the inability software engineering show in dealing with complexity, they introduce a cross-disciplinary initiative called the Karlskrona Manifesto for Sustainable Design (and Software).

2.4 The Social Dimension of Sustainability

Sustainability is usually thought of as composed of three overlapping, mutually dependent goals: a) to live in a way that is environmentally sustainable, or viable over the very long-term, b) to live in a way that is economically sustainable, maintaining living standards over the long-term, and c) to live in a way that is socially sustainable, now and in the future [21]. In order to safeguard natural systems, viable social systems are essential. However, social sustainability is regarded as the least developed dimension of sustainability [22]. Reviews on general social sustainable literature show that the concept has either been under-theorized or oversimplified. Authors and policymakers tend to derive definitions of social sustainability from their specific professional context, making it a challenge to attain a general definition [22].

When it comes to the influence social mechanisms, such as human factors, have on the SDP, studies to date have only scratched the surface, according to [10]. He argues that one reason for this lack of attention is that programming has not been studied as an individual cognitive activity. Researchers have not looked at personality traits. Regarding the complexity in high-level software development, it involves a range of people and activities. Ignoring human aspects, or not managing them appropriately, can potentially have a huge impact on the SDP, and team effectiveness [18]. [1] suggest a sub-category to the social dimensions, the individual dimension. It refers to the well-being of humans as individuals and should be seen as an integrated part of the social dimension. According to them, software sustainability can only be achieved when social, financial, and technical sustainability aspects of an organization are interdependently dealt with.

When presenting their strategic approach to social sustainability, [23] depart from the Framework for Strategic Sustainable Development (FSSD) developed by [24]. The framework builds on eight sustainable principles, or mechanisms, acting boundary "guardians". Five of these principles address the social dimension, claiming that, in a socially sustainable society, people are not subject to structural obstacles to: health, influence, competence, impartiality, and meaning making [24]. FSSD is a model initially developed to support organizations in their work with sustainable development on a strategic level.

According to [10] "[it] is impossible to exclude the human factors from software engineering expertise during software development because software is developed by people, for people". He continues saying that human errors are much more difficult for developers to deal with and are therefore overlooked in the SDP. Software developers rather move on to problems they are better acquainted with, i.e., technical problems [10]. The continuation of "dependence on abstract mathematical models to capture idealized behavior rather than what happens in the world" is more likely to be encouraged [7].

3 Research Strategy and Outcomes

3.1 Strategy

A systematic literature review (SLR) was conducted following the guidelines suggested by [25]. The choice to target literature oriented to the SDP specifically is motivated by the high failure rates detected among software development projects and the complexity involved in the process. Social factors have shown to be the predominant cause to these failures, which urges a closer look at how the social dimension of the SDP is addressed and whether a holistic approach is applied when doing so. The search has been narrowed to digital libraries belonging to the CSS filed. ACM and IEEE have been selected due to their scope and prominence. To identify relevant literature relating to the research question "software development process" were selected as the main search criteria and sustain* and holistic* were chosen as sub search criteria's, targeting titles and abstracts.

- *"Software Development Process" (SDP)* - SDP was chosen as the main criteria because it embraces the whole process, from contracting, coding to the complete software systems. The SDP constitutes a complex whole where all the sustainable aspects come together: ecologic, social and economic. Therefore, SDP offer a suitable frame for this study.
- *Sustain** - Word combinations of "sustain-" are expected to catch what sustainability implies in relation to the SDP, and it may also include obstacles that hinder SDP to become sustainable.
- *Holistic** - When elaborating on sustainability from the Brundtland report [5] perspective, holistic is the word

[2] See the animation here: https://www.youtube.com/watch?v=VTNmLt7QX8E

normally used to conceptualize the integrating act of balancing the three dimensions, ecology, social and economic. It could be argued that holistic is not a term that would normally be used by the CSS community. That makes sense if sustainability is used as a technological term for maintenance or durability. Research show that a holistic approach is key to sustainable computing, and since sustainability is used and understood as a comprehensive term in this study, choosing the corresponding search-word holistic becomes unavoidable. Previous research is urging for interdisciplinary and holistic approaches within CSS to handle the complexity and anomalies caused by CT. Matches to the search word "holistic" are expected to inform us if a holistic approach is applied in practice, and if then how.

One could argue that 'social' should have been added in the search string. The point here is to see how social aspects are incorporated in a holistic and sustainable perspective, i.e. including all three dimensions, ecology, social and economic. Using the wide concept 'social' as a search criterion would lead to much broader scope of SDP going beyond sustainability and a holistic perspective, which also has been shown in the results. Therefore, searching for social aspects is part of a second coding strategy.

The review is concept centric, meaning that the concepts derived from the search will determine the organizing framework of the review. The articles that use the concepts sustain* and holistic* two times or less has been excluded. The remaining articles have been synthesized guided by approaches related to the three sub-questions. The review has been performed in an explorative and iterative mode, adding layers of perspectives along the way.

3.2 Outcomes

The literature search resulted in 47 articles in total (see table 2.). Except for one article that appeared both in IEEE and ACM, the data are derived mainly from IEEE.

In a first coding process a word count was conducted to get an overview of the distribution of the two sub-search criteria. The appearance of the concepts was divided into the categories: title, abstract, and text. The findings offered an indication of what articles were relevant for the study (see appendix 1.). After having scanned the articles and investigating how the concepts were attributed, articles mentioning the concepts two times or less was regarded as irrelevant, 25 articles remained for scrutiny.

	RQ: *How does software development process literature address the social dimension of sustainability?*	
Main Search Criteria	**"Software development process"**	
Sub Search Criterias	**Sustain***	**Holistic***
ACM (Title/Abstract) 5 April 2018 and 30 March 2020	0/3	0/3
IEEE (Title/Abstract) 6 April 2018 and 30 March 2020	0/25	0/20
RESULTS	47 articles remained after filtering out duplicates	

Table 2. Articles Collected

To facilitate the answering of the research question, three sub-questions were used: *SQ1 - How is the concept sustainability addressed?* – Several studies have already been made reviewing how sustainability is defined in computer and systems science in general. The main purpose of this question is to give perspective on how the concept is used and explained within a SDP context. *SQ2 - Is a holistic approach applied when relating to sustainability?* – The definition of sustainability used here implies a holistic approach. Holistic refers to the balancing of the three dimensions of sustainability: ecological, social, and economic. This question intends to explore how the literature relate to the concept holistic. It is also expected to give an indication of how the three dimensions of sustainability is dealt with in relation to each other. *SQ3 - What social aspects are highlighted and of influence on the SDP?* – The reason to identify social aspects is not due to lack of knowledge about their existence, rather it is how they are approached if addressed at all. Is there any attempt to integrate the social dimension of sustainability into the SDP, if so, then how?

In order to derive answers from the collected material, elaborative questions were formulated relating to the two first sub-questions. Meanwhile, performing this data search social aspects relating to the SDP emerged that was equally searched for in all articles in order to get a more comprehensive picture of how, if, and why social aspects could be of importance to the SDP. Those aspects that appeared to have a direct influence on the SDP were filtered out, processed and summarized into three categories: challenges identified; human sources of risk and their influence on the SDP; and solutions offered. The guiding questions and social aspect elaborated on are shown in table 3. The conclusions of the outcomes will be discussed in the next section.

SUB-QUESTIONS	SUPPORTING QUESTIONS	
SC1 - How is the concept sustainability addressed?	Is it relating to the Brundtland definition?	Other aspects of concern?
	Is sustainabiliy equated to sustainance/maintainance?	
	Does a definition or conceptual discussion of the concept exist?	
	Contextual notes	
SQ2 - Is a holistic approach applied when relating to sustainability?	Does 'Holistic' refer to the three dimensions of sustainabiliy?	
	What dimension(s) of a holisitc approach is addressed?	
	Is 'Holistic' referring to managing compelxity in general (e.g. systemic, non-linear, contextual, inclusive, interconnectedness, multidimensional)?	
	Does a definition or conceptual discussion of the concept exist?	
	Contextual notes	
SQ3 - What social aspects are highlighted and of influence to the SDP?	Social, group, team, people, human and individual	
	Human well-being and needs	
	Behaviour, sentiment and emotion	
	Ethics, values, culture, norm	
	Trust	
	Collaboration	
	End-user, customer, client	
	Human infrastructure, intellectual and intagible capital (IC)	
	Contextual notes	

Table 3. Assigning data to sub-research categories

4 Results and Analysis

The literature search in ACM and IEEE resulted in 25 articles relevant for review. Fifteen articles were method oriented, six articles had an explorative approach, and further four articles presented frameworks of how to manage complexity. Regardless of the overall themes and purposes of the selected articles, fourteen of these articles addressed social aspects influencing the SDP. The one and only article elaborating on both the concepts sustainability and holistic [26] have been cited fifteen times since 2014, and the article cited most times (eighty-five times) was published in 1993 [11]. Seven of the eighteen articles matching sustain* offer, more or less developed, a definition or conceptual discussions related to the sub-search criteria. Appendix 2. presents an overview of these findings.

4.1 Summary of Results

A synthesis of the results from the sub-research questions reveals the following:

1. Except for one article [26], the concept sustainability is not used nor presented in a systemic way, acknowledging the complex interplay between social, ecological and economic factors influencing the SDP. In most of the articles the concept of sustainability is neither defined nor conceptually discussed, although used. Several articles [6][11][31][29][43] discuss the challenges of implementing a sustainable approach within the software community. They stress the need for raising sustainability awareness and for the facilitation of communicating sustainability holistically, including the social dimension. Most articles use the concept sustainability either in a vague sense or it is used to express product maintenance and life-length.
2. Except for one article [26], none of the authors offer a holistic approach when presenting their research, even if sustainability is framed by the definition of the Brundtland report [5]. Beyond these articles only two articles [47][49], out of which mention the concept holistic, defined and conceptually discussed the concept.
3. Several authors deem the social dimension to be of grand importance and highly influential to the SDP. Except for one article [26], these social factors are not systemically presented, nor part of a holistic approach. Rather, the social factors are related to complexity in general, and demonstrated to induce even more complexity to the SDP, causing confusion and higher failure rates.

Although the influence of social aspects on the SDP are well documented, this study indicates that the practice of dealing with the social dimension in a holistic manner is less of a priority in SDP management and implementation. None of the articles addressing influential social aspects in the SDP are referring to them as essential components of a sustainable system. A concluding reflection is that when used in SDP literature, sustainability and a holistic approach, whether defined through the definition of [5] or not, are concepts describing complexity and interrelations in general.

4.1 Key Findings

- The low sample can be indicating a too narrow search scope or limited search string. It may also act as a confirmation of previous statements on lack of interest for sustainable computing, or the inability to incorporate holistic approaches into the SDP, and to solve the challenges involved neglecting the social dimension. Two thirds of the collected articles are ten years old, and half of the articles are five years old or less.
- Most of the articles use both concept's 'sustainability' and' holistic' without any working definition nor conceptual discussion. Using key-concepts with floating definitions raises the need for contextualization when dealt with, especially when it comes to systemic problems, such as wicked issues.
- The results show no lack of awareness regarding the influence social factors have on the SDP. Although, if not listing a range of factors they focus on one isolated factor at the time, still failing to take a holistic grip of all the factors that affect the SDP simultaneously. Factors tending to be of equal importance.
- Suggested steps towards the realization of a paradigm shift:
  - Universally accepted and adopted procedures, and a total risk management ethics [11]. By conceptualizing and integrating sustainable aspects into the SDP [26] software practitioners are provided the support that allows them to view and define sustainable actions [27].
  - A systematic and structured process with a common ground for communications is suggested, which creates a mutual understanding of the whole system [11][28][42]. Also, standardized collaboration approaches are advised [29].
  - Decision support in weighting tangible and intangible values to one another [26][30][31].
  - The invitation of non-experts to participate in the SDP, allowing the problem space to be jointly explored cross-disciplinarily [6][29][32]. Also, when employing a user-centric approach contextual awareness is crucial, including culture, gender, class and accessibility aspects [43][44]
  - The enforcement of these changes should also involve attitude and behavior changes towards intangible social factors influencing the SDP [13][45][46] and the inclusion of social, behavioral and management sciences in the software engineering curriculum [11].
  - Above all, the ability for introspection and awareness of the part developers themselves play in aggravating flawed situations are paramount, e.g. caused by biases, value-systems and reductionistic methods [43]. Reflection is encouraged regarding how to transform computational thinking and value-neutral perspectives on the SDP to a more holistic value-based approach [47], also including the social dimension.
- Among the articles that do discuss a holistic approach to sustainability, an unexpected tendency was noticed. The social dimension was related to environmental and biological aspects, rather than social, cultural or interpersonal.

- Although, several articles stir to ethical reflection, only one article [45] expressively talks about the need for ethical considerations related to the SDP. The authors stress the importance of socio-technical and ethical issues. This also involve the usage of symbols and pictures, as well as the need for understanding aesthetic values. No other article discusses ethics or responsibility issues in software engineering on a wider societal scale.

5 Discussion and Limitations

5.1 Discussion

How does software development process literature address the social dimension of sustainability? As is concluded in the previous section the SDP literature does not address the social dimension from a holistic approach, and if addressed at all social factors are not referred to as an essential component of a sustainable system. Given that social factors are the predominant cause to software development project failures, and that these failure rates do not seem to decrease over time, it is noteworthy that the field of computer and systems sciences do not have more to offer regarding radical and liable approaches to solve this problem. The business as usual approach might not be acceptable for much longer.

The results presented in this study appear to be aligned with, in contradiction to, and to be lacking crucial perspectives outlined in previous research. There seem to be no doubt that the social dimension is of crucial importance to SDP success. Knowledge do exist regarding the influence of social factors, but they are not addressed in a holistic way. Further research that focuses on how to bridge the two conflicting worldviews represented in CSS, i.e., computational and holistic thinking, is encouraged. The former has shown inadequate to deal with complexity and wicked problems such as sustainability issues and the social dimensions of the SDP. In order to overcome the obstacles preventing us to incorporate a more holistic mindset in the SDP and in CSS as a whole, this study suggests four prime directions for further research:

1. Adequate tools and approaches need to be developed that support the integration of the social dimensions when managing and accomplishing a sustainable SDP.
2. The absence of ethical considerations and discussions about the impact each single developers' biases, sentiments, values or worldviews have on the SDP does not ease a transformative process, rather it encourages an ongoing disciplinary introspection where these aspects are seriously reflected upon and support is built into the process to keep awareness present at all times.
3. Also, an attitude and behavioral change is needed in the managerial layers in relation to social factors that has shown to be crucial for SDP success.
4. Realizing a fully systemic and holistic approach to the SDP implies a paradigm shift within CSS, providing social skills to deal with an increasingly complex emerging future.

5.2 Limitations

This study is framed by previous research on sustainable computing and has been delimited to the software development process with its two constituents, software development and process management.

The low sample of 47 articles originating mainly from one of the used databases is noteworthy. It is difficult to know whether the results would have given different answers if the search would have been extended to other digital libraries[3] or to non-academic papers. Also, additional search criteria could perhaps have given a bigger sample size. At times sustainability was used with reference to the word maintainability, or durability. The question is what contribution the use of 'maintenance', as a search criterion, would offer in relation to an inclusive holistic approach to sustainability, balancing the three dimensions, ecology, social, and economic. Most of the collected articles lacked conceptual definitions of the search-words, although different alternative implicit understandings can be derived. These alternative understandings could maybe serve as additional search criteria's in an extended study in the future.

Delimiting this study to the CSS domain solely is a limitation. Some branches of sustainable Human-Computer-Interaction research align with the field of software design although the gap between them is significant [33]. How these two fields overlap regarding the social dimensions of sustainability could have been elaborated on more but given the limited time frame of the study this relationship need to be explored elsewhere. Given the interdisciplinary nature of sustainability and the broad usage of computational technology, a search in other domains, such as social, political, psychological, human sciences etc., could maybe offer a bigger sample.

The two frameworks Framework for Strategical Sustainable Development [24] and the Karlskrona Manifesto for Sustainable Design (and Software) [1] have been suggested as potential supportive structures that could be applied in further research. Whether other supportive frameworks or models could serve as better alternatives is an inquiry that goes beyond the scope of this study. Maybe other approaches taken from sustainable development research may also prove appropriate, such as complex adaptive systems [34], the doughnut economics perspective [35] or Bendell's Deep Adaptation Agenda [36]. Also, Theory U, founded by MIT's Otto Scharmer [37], may serve as a supporting framework for creating social sustainable computation.

[3] The limited search result encouraged an extended search using the same search-words in a few other digital libraries providing CSS literature together with other disciplines, e.g. Journal of Statistical Software (JSS), SpringerLink, and ScienceDirect. This search was made November 11th 2019. No matches were found.

6 CONCLUDING REFLECTIONS

Software development projects continue to demonstrate high failure rates. Addressing this problem is important due to the significance it has for the wider impact software has within and beyond computational science. As previous research has pointed out the predominant causes behind these failures are social factors. From a sustainable perspective software success is dependent on how well the three dimensions: ecology, social and economy, are balanced and integrated with each other. What both previous research indicates and the results of this study verify, is that although there are many attempts to address sustainability aspects of the software development process (SDP) they fail to incorporate the social dimensions in the equation. On the other hand, the results also show that when elaborating on social factors of influence researchers tend to isolate them from the bigger picture failing to handle them as essential components of a sustainable whole. If the social dimension constitutes a predominant part of the reasons behind high failure rates of software development projects, it is noteworthy that we have not seen more effort and resources invested in finding new and better approaches within the SDP field.

Utterly, this study elaborates on the challenges involved in bridging the two worldviews of computer and systems science. These challenges are not limited to software engineering, rather it is the concern of all societal sectors bounded by stiff managerial systems, and to those affected by them, e.g. if we could come to terms with social factors like translation problems between clients and developers' worldviews, lack of contracting skills and transparency of the process, we could save immense amounts of tax-money used to maintain flawed IT systems. Money that could be allocated for better purposes.

Encouraging interdisciplinary cross-sectorial collaborations could provide fruitful soil for innovative perspectives on future solutions on how to integrate complex social factors. Measurement or computing based on predefined static representations of a past reality, disconnected from context or the world as it is, may blind us from what is right in front of us and from what lies ahead. When supportive systems and structures are based on simplified assumptions about the world, what solutions can we then expect them to offer us when dealing with complex societal problems? Problems usually constitutes parts of their own solution and if complexity is one of these parts systemic solutions should be strived for. This entails holistic and interdisciplinary approaches, which include the social dimension as well.

The aim of this study is not to provide an answer to how this paradigm shift can or should be realized. Instead it offers a contribution to a bigger discussion that is of concern to all of us – What should our mutual future look like? Becoming aware of our blind spots may enable us to embrace an emerging future with open eyes.

"Systems developed and implemented without a sound set of principles are doomed to unpredictability, with the consequence that success can neither be sustained nor failure avoided." [38]

ACKNOWLEDGMENTS

The authors would like to thank David Sundgren, June Verner, Binnur Balkan, Emir Konuk, Cecilie Helmer, Emilia, Ola Gustafsson, Letica Duboc, Luiz Fernando Capretz, Fabio Queda Bueno da Silva, Patricia Lago and Jorge Zapico.

REFERENCES

[1]. Christoph Becker, Ruzanna Chitchyan, Leticia Duboc, Steve Easterbrook, Birgit Penzenstadler, Norbert Seyff, **and** Colin C. Venters. 2015. Sustainability Design and Software: The Karlskrona Manifesto. In *IEEE/ACM 37th IEEE International Conference on Software Engineering*. Florence (May 2015) 467-476. https://doi.org/10.1109/ICSE.2015.179

[2] Nina Wolfram, Patricia Lago, **and** Francesco Osborne. 2017. Sustainability in software engineering. In *Sustainable Internet and ICT for Sustainability (SustainIT)*, Funchal (December 2017), 1-7. https://doi.org/10.23919/SustainIT.2017.8379798

[3] Peter Oriogun, Olatunji Akinbule, Chinwe Ibecheozor, **and** Zayyad Nyako. 2012. Software engineering ethical decision making and professional responsibility. In *African Conference on Software Engineering and Applied Computing*. Gaborone (September 2012), 7–14. https://doi.org/10.1109/ACSEAC.2012.9

[4] Thomas T. Hewett, Ronald M Baecker, Stuart K. Card, Tom Carey, Jean B Gasen, Marilyn Mantei Tremaine, Gary Perlman, Gary W Strong, **and** William Verplank. 1992. *ACM SIGCHI Curricula for Human-Computer Interaction*. Technical Report. ACM, New York. https://doi.org/10.1145/2594128

[5] Gro H. Brundtland. 1987. *Report of the World Commision on Environement and Development: Our Common Future. United Nations*. https://doi.org/10.1007/978-94-007-0753-5_441

[6] Peter Newman, Maria Angela Ferrario, Will Simm, Stephen Forshaw, Adrian Friday, **and** Jon Whittle. 2015. The Role of Design Thinking and Physical Prototyping in Social Software Engineering. In *IEEE/ACM 37th IEEE International Conference on Software Engineering*. Florence (May 2015), 901-909. https://doi.org/10.1109/ICSE.2015.181

[7] Steve Easterbrook. 2014. From Computational Thinking to Systems Thinking: A conceptual toolkit for sustainability computing. In *Proceedings of the 2014 conference ICT for Sustainability*. Stockholm (August 2014), 235–244. https://doi.org/10.2991/ict4s-14.2014.28

[8] Friedrich Chasin. 2014. Sustainability: Are We All Talking About the Same Thing? State-of-the-Art and Proposals for an Integrative Definition of Sustainability in Information Systems. In *Proceedings of the 2014 conference ICT for Sustainability*. Stockholm (August 2014), 342–351. https://doi.org/10.2991/ict4s-14.2014.42

[9] Jorge. L. Zapico. 2014. Blinded by data: The risks of the implicit focus on data in ICT for Sustainability. In *2nd International Conference on ICT for Sustainability*. Stockholm (August 2014),148–154. https://doi.org/10.1109/ICSE.2015.181

[10] Luiz Fernando Capretz, Fahem Ahmed, **and** Fabio Queda Bueno da Silva. 2017. Soft Sides of Software. In *Information and Software Technology*. Vol. 92, 92-94. Elsevier https://doi.org/10.1016/j.infsof.2017.07.011

[11] C. G. Chittister, and Yacov Y. Haimes. 1993. Risk Associated with Software development: a Holisitc framework for assessment and management. In IEEE Transactions on Systems, Man, and Cybernetics (May/June 1993). Vol. 23(3), 710-723. https://doi.org/10.1109/21.256544

[12] Narciso Cerpa and June. M. Verner. 2009. Why did your project fail? In *Communications of the ACM* (December 2009). Vol. 52(12), 130–134. https://doi.org/10.1145/1610252.1610286

[13] Guillermo Villasana and Rodolfo Castello. 2014. An Agile Software Quality Framework Lacking. In *World Congress on Computer Applications and Information*

Systems. Hammamet (January 2014), 1-4. https://doi.org/10.1109/WCCAIS.2014.6916549

[14] Haslinda Sutan Ahmad Nawi, Othman Ibrahim, Nur Syufiza Ahmad Shukor, Siti Fatimah Omar, Irny Suzila Ishak, and Azizah Abdul Rahman. 2015. Understanding sustainability: An exploration of the is literature. In *Journal of Theoretical and Applied Information Technology*. Vol. 73(3), 447–455. Jatit. ISSN: 1992-8645

[15] Christian Dawson, and Ray Dawson**.** 2013. Software Development Process Models: A Technique for Evaluation and Decision-Making. In *Knowledge and Process Management* (November 2013). Vol. 21(1), 42–53. https://doi.org/10.1002/kpm.1419

[16] Paul Clarke, Rory V. O'Connor, **and** Brian Leavy**.** 2016. A complexity theory viewpoint on the software development process and situational context. In *Proceedings of the International Conference on Software and Systems Process*, ACM, New York (May 2016), 86-90. https://doi.org/10.1145/2904354.2904369

[17] Paul Clarke and Rory V. O'Connor**.** 2015. Changing Situational Contexts Present a Constant Challenge to Software Developers. In: Rory O'Connor et al. (eds) Systems, Software and Services Process Improvement. In *Proceedings of 22nd European Conference* EuroSPI 2015. Ankara (September/October 2015), 100–111. https://doi.org/10.1007/978-3-319-24647-5_9

[18] Giuseppe Destefanis, Marco Ortu, Steve Counsell, Stephen Swift, Michele Marchesi, **and** Roberto Tonelli. 2016. Software development: do good manners matter? In *PeerJ Computer Science* (July 2016). Vol.2, e73. https://doi.org/10.7717/peerj-cs.73

[19] Ivan Bolis, Sandra N. Morioka, and Laerte I. Sznelwar. 2017. Are we making decisions in a sustainable way? A comprehensive literature review about rationalities for sustainable development. In *Journal of Cleaner Production* (March 2017). Vol. 145, 310–322. Elsevier Ltd. https://doi.org/10.1016/j.jclepro.2017.01.025

[20] Christian Sandvig, Kevin Hamilton, Karrie Karahalios, and Cedric Langbort. 2016. Automation, algorithms, and politics| When the Algorithm Itself is a Racist: Diagnosing Ethical Harm in the Basic Components of Software. In *International Journal of Communication* (August 2016). Vol. 10, 4972–4990. ISSN: 1932–8036

[21] Jesse Dillard, Veronica Dujon, and Mary C. King (eds). 2008. *Understanding the Social Dimension of Sustainability*. Taylor & Francis. ISBN:10: 0-415-96465-2

[22] Merlina Missimer, Karl-Henrik Robèrt, and GöranBroman. 2017. A strategic approach to social sustainability – Part 1: exploring the social system. In *Journal of Cleaner Production* (January 2017). Vol. 140(Part 1), 32-41. Elsevier https://doi.org/10.1016/j.jclepro.2016.03.170

[23] Merlina Missimer, Karl-Henrik Robèrt, and GöranBroman. 2017. A strategic approach to social sustainability – Part 2: a principle-based definition. In *Journal of Cleaner Production* (January 2017). Vol. 140(Part 1), 42-52. Elsevier https://doi.org/10.1016/j.jclepro.2016.04.059

[24] Göran Broman and Karl_Henrik Robèrt. 2017. A framework for strategic sustainable development. in *Journal of Cleaner Production* January 2017. Vol. 140(Part 1), 17-31. Elsevier. https://doi.org/10.1016/j.jclepro.2015.10.121

[25] Jane Webster and Richard T. Watson. 2002. Analyzing the Past to Prepare for the Future: Writing a Literature Review. In *MIS Quarterly* (June 2002). Vol. 26(2), xiii–xxiii, ISSN: 02767783

[26] Stefanie Betz and Timm Caporale, 2014. Sustainable Software System Engineering. In *IEEE Fourth International Conference on Big Data and Cloud Computing*. Sydney (December 2014), 612-619, https://doi.org/10.1109/BDCloud.2014.113

[27] Bokolo Anthony, Mazlina Abdul Majid, and Awanis Romli. 2017. A model for adopting sustainable practices in software based organizations. In *8th International Conference on Information Technology*. Amman (May 2017), pp. 26-35, https://doi.org/10.1109/ICITECH.2017.8079911

[28] Sanath S. Shenoy and Raghavendra Eeratta. 2011.Green software development model: An approach towards sustainable software development. In *Annual IEEE India Conference.* Hyderabad (December 2011), 1-6. https://doi.org/10.1109/INDCON.2011.6139638

[29] Stanley Ewenike, Elhadj Benkhelifa, and Claude Chibelushi. 2017. Cloud Based Collaborative Software Development: A Review, Gap Analysis and Future Directions. In *IEEE/ACS 14th International Conference on Computer Systems and Applications*. Hammamet (November 2017), 901-909, https://doi.org/10.1109/AICCSA.2017.220

[30] I. M. Hampton and B. W. T Quinn. 2000. Software project measurement criteria. In *Proceedings First Asia-Pacific Conference on Quality Software*. Hong Kong (October 2000), 258-264, https://doi.org/10.1109/APAQ.2000.883799

[31] Sebastian Barney, Claes Wohlin, and Aybuke Aurum. 2009. Balancing software product investments. In *3rd International Symposium on Empirical Software Engineering and Measurement*. Lake Buena Vista (October 2009), 257-268, https://doi.org/10.1109/ESEM.2009.5316001

[32] Harvey P. Siy, James D. Herbsleb, Audris Mockus, Mayuram Krishnan, and George T. Tucker. 2001. Making the software factory work: lessons from a decade of experience. In *Proceedings Seventh International Software Metrics Symposium*. London (April 2001), 317-326. https://doi.org/10.1109/METRIC.2001.915539

[33] Carl DiSalvo, Phoebe Sengers, and Hrönn Brynjarsdóttir. 2010. Mapping the landscape of sustainable HCI. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Atlanta (April 2010), 1975-1984. ACM https://doi.org/10.1145/1753326.1753625

[34] John H. Miller and Scott E. Page. 2007. *Complex Adaptive Systems: An Introduction to Computational Models of Social Life*. Princeton University Press, New York, Woodstock ISBN-13: 978-0-691-13096-5

[35] Kate Raworth. 2018. *Doughnut Economics: Seven Ways to Think Like a 21st-Century Economist*. Random House Business Books. ISBN: 9781847941398

[36] Jem Bendell. 2018. Deep adaptation: a map for navigating climate tragedy. Institute for Leadership and Sustainability (IFLAS) Occasional Papers vol. 2. University of Cumbria. Ambleside, UK. 2018 (Unpublished) Last assessed at 5/2/2020: http://insight.cumbria.ac.uk/id/eprint/4166/.

[37] C. Otto Scharmer. 2016. *Theory U: Leading from the Future as it Emerges*. (2nd ed.). McGraw-Hill Education, USA. ISBN: 9781626567986

[38] G. Michael McGrath and Lorna Uden. 2000. Modelling 'Softer' Aspects of the Software Development Process: An Activity Theory based Approach. In *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences.* Maui January 2000, 1-9. https://doi.org/10.1109/HICSS.2000.926778

[39] Johanna L. Gustavsson, 2019. Blinded by Simplicity: Locating the Social Dimension in Software Development Process Literature. Master thesis at Stockholm university, December 2019

[40] ACM Council, "ACM Code of Ethics and Professional Conduct," adopted by 6/22/18. Last assessed at 5/2/2020: https://ethics.acm.org/code-of-ethics/

[41] IEEE Board of Directors, "IEEE Code of Conduct," approved by June 2014. Last assessed at 5/2/2020: https://www.ieee.org/content/dam/ieee-org/ieee/web/org/about/ieee_code_of_conduct.pdf

[42] Barbara Scozzi, Kevin Crowston, U. Yeliz Eseryel, and Qing Li. 2008. Shared Mental Models among Open Source Software Developers. In *Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS)*, Waikoloa (January 2008), 306-306. https://doi.org/10.1109/HICSS.2008.391

[43] Heike Winschiers and Barbara Paterson. 2004.Sustainable software development. In Proceedings of the 2004 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries (SAICSIT '04). South African Institute for Computer Scientists and Information Technologists (October 2004), ZAF, 274–278. https://dl.acm.org/doi/abs/10.5555/1035053.1035090

[44] Wiesław Kopeć, Radosław Nielek, and Adam Wierzbicki. 2018. Guidelines towards better participation of older adults in software development processes using a new SPIRAL method and participatory approach. In *Proceedings of the 11th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE '18)*. ACM New York (May 2018), 49–56. https://doi.org/10.1145/3195836.3195840

[45] Dharmender Singh Kushwaha, Ranjit K Singh, and Arun Kumar Misra. 2006. Cognitive web based software development process: towards a more reliable approach. *In SIGSOFT Softw. Eng* (July 2006). Notes 31(4), 1–6. https://doi.org/10.1145/1142958.1142963

[46] Paula Rachow, Sandra Schröder, and Matthias Riebisch. 2018. Missing Clean Code Acceptance and Support in Practice - An Empirical Study. *In 25th Australasian Software Engineering Conference (ASWEC)*, Adelaide (November 2018), 131-140. https://doi.org/10.1109/ASWEC.2018.00026

[47] Atabek Murtazaev, Sungwon Kang, Jongmoon Baik, and Jihyun Lee. 2010. An Approach to Defining a Value-Based Software Development Process. In *IEEE/ACIS 9th International Conference on Computer and Information Science*, Yamagata (August 2010), 690-695. https://doi.org/10.1109/ICIS.2010.79

[48] Fritz Heider and Marianne Simmel. 1944. An Experimental Study of Apparent Behavior. In *The American Journal of Psychology* (April 1944). Vol. 57(2), 243–259. https://doi.org/10.2307/1416950

[49] Luca Ponzanelli.2014. Holistic recommender systems for software engineering. In *Companion Proceedings of the 36th International Conference on Software Engineering (ICSE Companion 2014).* New York (May 2014), 686–689. ACM https://doi.org/10.1145/2591062.2591081

APPENDIX 1.

*Appearance of concepts with regards to Sustain** (Search april 2018)

22 Articles	Betz 2014	Anthony 2017	Salam 2016	Sheaoy 2011	Newman 2015	Barney 2009	Ewenike 2017	Terrel 2015	Byer 2009	Chiitister 1993	Hampton 2000	Sly 2001	Majumdar 2015	McGrath 2000	Auer 2003	Adeklie 2010	Barzilay 2009	Hussein 2008	Magies 2016	Prokech 2017	Schirmeier 2015	Villasana 2014
Sustain*	167	155	33	12	9	6	4	4	3	3	3	3	2	2	1	0	0	0	0	0	0	0
Title	*	*	*	*																		
Abstract	*	*	*	*	*	*			*				*	*	*							
Text	*	*	*	*	*	*	*	*	*	*	*	*	*	*								
Holistic*	8	0	0	0	0	0	3	0	0	7	0	0	0	0	0	2	1	2	2	4	1	9
Title										*												
Abstract	*						*			*						*	*	*	*	*	*	*
Text	*						*			*						*		*		*		*

*Appearance of concepts with regards to sustain** (Additional search March 2020)

25 Articles	Winschiers 2004	Kopec 2018	Massimo 2003	Scozzi 2008	Ching 2018	Rachow 2018	Scott 2017	Wang 2018	Kushwaha 2006	Ponzanelli 2014	List 2005	Magies 2016	Schirmeier 2015	Adeklie 2010	Ahmed 2008	Barzilay 2009	Niazi 2003	Murtazaev,2010	Kaur 2015	Das 2014	Kaswan, 2015	Gupta 2015	Bhujang 2017	Nanda 2017	Athirthan 2018	
Sustain*	13	10	4	4	3	3	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Title	*		*																							S
Abstract	*	*				*		*																		D
Text	*	*	*	*	*	*	*	*																		P
Holistic*	0	0	0	0	2	0	3	0	3	10	2	2	1	2	2	1	3	4	0	0	0	0	0	0	0	
Title							*			*	*															
Abstract					*				*	*		*	*	*	*	*	*	*								
Texts					*		*		*	*	*			*	*		*	*								

*Appearance of concepts with regards to Holistic** (Search april 2018)

22 Articles	Villasana 2014	Betz 2014	Chiitister 1993	Prokech 2017	Eweniike 2017	Adeklie 2010	Hussein 2008	Magies 2016	Barzilay 2009	Schirmeier 2015	Anthony 2017	Salam 2016	Sheaoy 2011	Newman 2015	Barney 2009	Terrel 2015	Byer 2009	Hampton 2000	Sly 2001	Majumdar 2015	McGrath 2000	Auer 2003
Holistic*	9	8	7	4	3	2	2	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Title			*																			
Abstract	*	*	*	*	*	*	*	*	*	*												
Text	*	*	*	*	*	*	*															
Sustain*	0	167	3	0	4	0	0	0	0	0	155	33	12	9	6	4	3	3	3	2	2	1
Title		*									*	*	*									
Abstract		*									*	*	*	*	*		*		*	*	*	*
Text		*	*		*						*	*	*	*	*	*	*	*	*	*	*	

*Appearance of concepts with regards to holistic** (Additional search March 2020)

25 Aricles	Ponzanelli. 2014	Murtazaev 2010	Scott 2017	Kushwaha 2006	Niazi 2003	Ching 2018	List 2005	Magies 2016	Adeklie 2010	Ahmed 2008	Schirmeier 2015	Barzilay 2009	Winschiers 2004	Kopec 2018	Massimo 2003	Scozzi 2008	Rachow 2018	Wang, 2018	Kaur 2015	Das 2014	Kaswan 2015	Gupta 2015	Bhujang 2017	Nanda 2017	Athirthan 2018	
Holistic*	10	4	3	3	3	2	2	2	2	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
Title	*		*				*																			S
Abstract	*	*		*	*	*		*	*	*	*	*														D
Texts	*	*	*	*	*	*	*		*	*																P
Sustain*	0	0	2	0	0	3	0	0	0	0	0	0	13	10	4	4	3	2	0	0	0	0	0	0	0	
Title													*		*											
Abstract													*	*			*	*								
Text			*			*							*	*	*	*	*	*								

Sustain appear in name of conference

APPENDIX 2.

	Articles of Relevamce	Theme of Article	Model/Method/ Approach, 15	Explorative Approach, 6	Framework Appraoch, 4	Paper citation (Google Scholar 8th April)	No definition nor conceptual discussion of sustainability
1	**Betz and Caporale, 2014. S***ustainable Software System Engineering (167 S/8H)*	Holistic				15	
2	**Anthony et al., 2017** *A Model for Adopting Sustainable Practices in Software Based Organizations (155 S)*	Env-econ				0	
3	**Salam and Khan, 2016** *Developing Green and Sustainable Software: Success Factors for Vendors (33 S)*	Env-econ				1	
4	**Shenoy and Eeratta, 2011** *Green Software Development Model An Approach towards Sustainable Software development (12 S)*	Env-econ				51	
5	**Newman et al., 2015** *The Role of Design Thinking and Physical Prototyping in Social Software Engineering (9 S)*	Socio-econ				29	
6	**Barney et al., 2009** *Balancing Software Product Investments (6 S)*	Socio-econ				20	
7	**Ewenike et al., 2017***Cloud Based Collaborative Software Development: A Review, Gap Analysis and Future Directions (4 S/3 H)*	Socio-econ				0	
8	**Terrel et al., 2015** *Scientific Software Communities (4 S)*	Economic				1	
9	**Byer and Shahmehri, 2009** *Prioritisation and Selection of Software Security Activities (3 S)*	Economic				9	
10	**Chittister and Haimes, 1993** *Risk Associated with Software Development: A Holistic Framework for Assessment and Management (3 S/7 H)*	Socio-econ				85	
11	**Hampton and Quinn, 2000** *Sofware Project Measurement Criteria (3 S)*	Economic				4	
12	**Siy et al., 2001** *Making the Software Factory Work: Lessons from a Decade of Experience (3 S)*	Socio-econ				20	
13	**Proksch et al., 2017** *Enriching In-IDE Process Information with Fine-Grained Source Code History (4 H)*	Socio-econ				17	
14	**Villasana and Castelló, 2014** *An agile software quality framework lacking (9 H)*	Economic				19	
15	**Winschiers and Paterson. 2004.** *Sustainable software development. (S13/H0)*	Socio-econ				28	
16	**Kopeć et al., 2018.** *Guidelines towards better participation of older adults in software development processes using a new SPIRAL method and participatory approach. (S10/H0)*	Socio-econ				11	
17	**Singh et al., 2006.** *Cognitive web based software development process: towards a more reliable approach. (S0/H3)*	Socio-econ				8	
18	**Scott et al., 2017**. A *Holistic Capstone Experience: Beyond Technical Ability. (S2/H3)*	Socio-econ				1	
19	**Ponzanelli, 2014.** *Holistic recommender systems for software engineering. (S0/H10)*	Economic				7	
20	**Massimo and Alberto, 2003.** *USDP application: cutting out a sustainable process for the public local administration. (S4/H0)*	Economic				0	
21	**Scozzi et al., 2008.** *Shared Mental Models among Open Source Software Developers. (S4/H0)*	Economic				43	
22	**Rachow et al., 2018.** *Missing Clean Code Acceptance and Support in Practice - An Empirical Study. (S3/H0)*	Socio-econ				2	
23	**Ching and Mutuc,2018.** *Evaluating Agile and Lean Software Development Methods from a System Dynamics Perspective. (S3/H2* **)**	Socio-econ				0	
24	**Niazi and Shastry,2003.** *Role of requirements engineering in software development process: an empirical study. (S0/H3)*	Socio-econ				44	
25	**Murtazaev et al., 2010**. *An Approach to Defining a Value-Based Software Development Process. (S0/H4)*	Economic				7	