Stakeholders' Social Interaction in Requirements Engineering of Open Source Software

Tanmay Bhowmik
Department of Computer Science and Engineering
Mississippi State University
Mississippi State, USA
tb394@msstate.edu

***Abstract*—Requirements engineering (RE) involves human-centric activities that require interaction among different stakeholders. Traditionally, RE has been considered as a centralized, collocated, and phase-specific process. However, in open-source software (OSS) development environment, the core RE activities are iterative and dynamic and follow a rather decentralized software engineering paradigm. This crosscutting characteristic of open-source RE can be conceptualized using the "Twin Peaks" model that weaves RE together with software architecture. Although many weaving mechanisms have been proposed in recent years, lack of theoretical underpinning limits a mechanism's applicability and usefulness in different scenarios. In this research proposal, we hypothesize stakeholders' social interaction as an ecologically valid weaving mechanism of the "Twin Peaks" in open-source RE. We further outline a concrete research plan to examine the generalizability of this weaving mechanism for three activities: requirements identification, requirements implementation, and creativity in RE. Carrying out this research plan will enable us to gain valuable insights to generate guidelines for enhancing software engineering practice in relevant areas.**

***Index Terms*—Requirements engineering; twin peaks; open-source RE; requirements identification; requirements implementation; creativity in RE**

I. Motivation

Requirements engineering (RE) is a set of activities concerned with identifying and communicating the purpose of a software system, and the contexts in which it will be used [1]. The RE process includes the elicitation, modeling, analysis, negotiation, prioritization, and realization of the requirements that the intended software shall meet [1]. Traditionally, RE has been considered as a centralized, collocated, and phase-specific process associated with individual projects or project components. Much of the traditional RE has focused on models and techniques in order to aid identification and documentation of stakeholders and their needs in a form that can be analyzed, communicated, agreed upon, and eventually realized and validated. However, in the open-source software (OSS) development environment, RE activities need to focus on generation, negotiation, adaptation, realization, and maintenance of requirements in a decentralized, iterative, and dynamic software-intensive ecosystem [2]. RE activities in the OSS development paradigm are therefore no longer part of a centralized process specific to a particular phase of software development. Rather these activities are spread, tangled, or otherwise intertwined with the overall software development process in a decentralized manner.

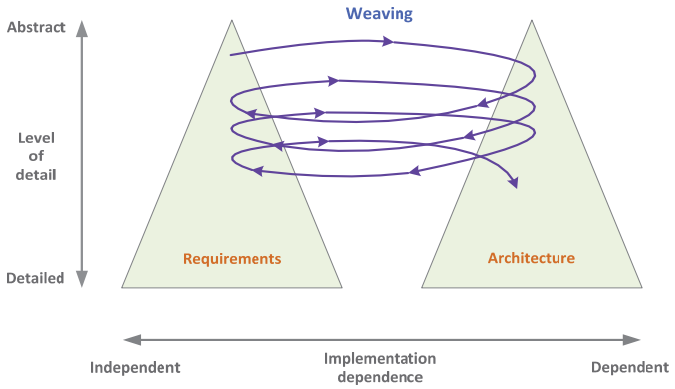

Fig. 1. The "Twin Peaks" of requirements and architecture (adapted from [3]).

Recent research has highlighted the crosscutting characteristics of RE in relation to other software engineering activities. In particular, Nuseibeh [3] identified the highly intertwining nature of RE and software architecture and characterized this relationship in the "Twin Peaks" model. This model, shown in Figure 1, emphasizes on the equal status and an iterative co-development of requirements and architecture. The distinction between requirements and architecture has its root in Jackson's seminal works in RE [4], [5], [6]. Specifically, Jackson distinguished the problem domain and the machine domain. Requirements are located in the problem domain, transformed via the shared phenomena between the two domains, and eventually realized by the source code executing in the machine domain. Despite the distinction, the "Twin Peaks" model emphasizes the weaving mechanism that connects the problem and machine domains. Many approaches have been proposed to instantiate the weaving, including model-based test-driven development [7], architecture and hardware constraints [8], and engineering architecturally significant requirements [9], [10]. However, existing methods mostly follow the traditional, centralized software engineering paradigm. As a result, little is known about how to weave the "Twin Peaks" in the decentralized environment of open-source RE.

Research on the nature of RE in OSS development has emerged only recently. Alspaugh and Scacchi studied the

978-1-4799-3032-6/14/$31.00 © 2014 IEEE

RE activities of OSS systems and emphasized that many successful OSS projects do not follow the classical, one-time RE process [11]. Dissimilarities between traditional RE and open-source RE was also stressed by Xiao *et al.* [12]. A seminal paper was presented by Ernst and Murphy [13] exploring that in open-source RE stakeholders often capture requirements less formally and elaborate the requirements only after the implementation begins. They introduced the notion of "just-in-time" requirements to characterize the tightly coupled stakeholder relationship in open-source development. Their study shows that a crucial factor in the success of OSS projects is the social interaction among the stakeholders from both problem and machine domains [4], [5]. It is through stakeholders' social interaction that technical aspects are clarified and organizational dependencies are resolved [14]. Inspired by the work of Ernst and Murphy [13], we formulate the central hypothesis of our research.

> **Central hypothesis:** In decentralized OSS development, stakeholders' social interaction serves a unified weaving mechanism of the Twin Peaks, which offers insights into a wide range of RE activities.

This research proposal introduces an ecologically valid [15] weaving mechanism of the "Twin Peaks" that promises a further understanding of RE in a decentralized environment. Successful execution of this research work may also help us generalize the weaving mechanism in the closed-source world where stakeholders' social interaction is often dominant. As several means of stakeholders' social interaction exist in the software world, the biggest challenge in our research is associated with operationalizing social interaction manageable in the scope of this work. Thus, this research abstract describes the state-of-the-art from which we derive the central hypothesis and operationalize stakeholders' social interaction in Section II. We present a concrete plan to test our hypothesis in Section III. Section IV discusses the current progress and contributions, followed by Section V providing the summery and future work of this research.

II. State of the Art

In this section, we describe the background and bedrocks of our research. The goal is to lay the foundation for a novel and improved RE framework for OSS development that will be introduced in Section III. To that end, we first present Jackson's conceptualization and distinction of problem and machine domains [4], [5]. Then we explain the "Twin Peaks" model of requirements and architecture [3], followed by a synopsis on the OSS RE [2] and stakeholders' social interaction in OSS development.

A. The Problem-Machine Environment

Software development is an engineering process with software being the end product. This product shall strive to meet the stakeholder's goals, needs, and desires, which are commonly referred to as requirements in software engineering. In one of the foundational papers, Jackson teased out the meaning of requirements by distinguishing two domains:

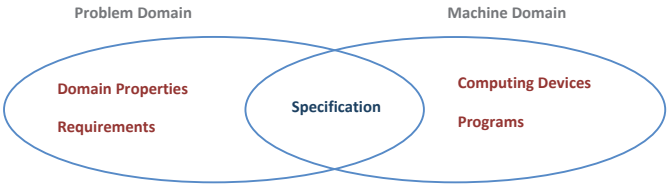

Fig. 2. The problem-machine environment (adapted from [4]).

the problem domain and the machine domain [4]. Figure 2 shows this conceptualization. The problem domain consists of properties and phenomena of the application that give rise to the stakeholder's requirements, whereas the machine domain is private to the intended software and the computing devices in which the software operates.

Requirements are primarily located in the problem domain, as shown in Figure 2. This has several important implications to RE. First, the elicitation and identification of requirements shall focus on the real-world needs of users, customers, and other stakeholders. Second, the transformation of requirements to a software-intensive solution is very open-ended and creative in nature because computing technologies provide many opportunities for possible realization of the requirements. Third, the validation of requirements cannot be performed only in the machine domain (e.g., through verification of specification); rather, the implemented software-intensive system is valid only if it corresponds to the fulfillment of stakeholder's goals. As shown in Figure 2, "specification" serves as the bridge between the two domains. Jackson used designation and assertion to define specification in formal logic (e.g., first-order predicate logic, temporal logic, and deontic logic) [4]. Meanwhile, he emphasized that RE is not a branch of pure mathematics or logic: the meaning of requirements depends crucially on the interpretation by the stakeholders in the two domains.

B. The Twin Peaks Model

Building upon Jackson's conceptualization, Nuseibeh proposed the "Twin Peaks" model as a way of showing the intertwining relationship between RE and software architecture [3]. Figure 1 illustrates this model. The key observation here is that requirements can influence the architecture that designers select or develop, whereas candidate architectures can restrict designers from meeting particular requirements [3]. The model emphasizes the equal status of requirements and architecture and places each in a peak. The vertical axis indicates the level of details from abstract to detailed, while the horizontal axis presents implementation dependence. The model concurrently starts the requirements and architectural specification at an abstract level considering requirements specification as the problem structure and architectural specification as the solution structure. According to the model, the abstract requirements and architecture go through a simultaneous iterative process that progressively produces separate requirements and architectural specifications at a more detailed level.

It is interesting to note that the "Twin Peaks" model can be viewed as a generalization of Jackson's conceptualization in that "weaving" is used as a general mechanism to link together the problem and machine domains, and "specification" is an instantiation of weaving in Figure 2. In fact, several methods and techniques have been proposed in recent years to implement "weaving", or in other words, to traverse the "Twin Peaks" [16]. Mou and Ratiu [7] introduced model-based test-driven development as a mechanism for linking formalized requirements and architectures in embedded software development. Loft *et al.* suggested that hardware platform plays a crucial part in the requirements and architectural development of software systems with hardware constrains. Engineering architecturally significant requirements [9], [10] could be another means of traversing the "Twin Peaks" as they oftentimes influence a software system's design decisions [17]. However, these existing solutions are rather isolated and mostly built upon traditional centralized software engineering paradigm. As a result, the success in one situation may not be applicable to other scenarios or activities. In this research proposal, we aim to create a theoretically sound weaving mechanism that can be applied to a wide range of "Twin Peaks" activities, especially in a modern open-source RE context.

C. OSS RE and Stakeholders' Social Interaction

Only recently researchers have begun to understand RE in OSS development. In his seminal work, Scacchi [18] identified a set of twenty-odd different types of what he called 'software informalisms' in use across a wide variety of OSS projects. According to Scacchi [18], informalisms like instant messaging and internet relay chat (IRC) provide socially lightweight mechanisms for communicating and coordinating project knowledge. In a further study, Scacchi et al. [19] examined OSS systems from five different application domains and found that very often a new requirement in an OSS system is informally captured through a story telling or a user experience at the initial stage. Ernst and Murphy [13] advanced our understanding about OSS RE by studying the just-in-time requirements management where requirements identification and realization are tightly coupled. In just-in-time RE certain requirements are informally captured and later clarified through stakeholders' social interaction during implementation.

In order to ensure sustainability in the modern competitive market, OSS systems are also subject to deliver new and improved features to the users realizing creative new requirements in short release cycles. Social interaction has been found to be influential on productivity and new idea generation. For example, Pirolli [15], [20] suggested that the way stakeholders interact with each other could influence the novelty of their ideas. Several means of such interactions among stakeholders have been identified in the literature. For example, comments or activities in issue tracking systems, email, and IRC are widely considered as means of social interaction among different stakeholders [21], [22], [23]. Research has also found a dominant use of issue tracking systems by the developers and other stakeholders to interact among themselves and to keep the requirements on track [13].

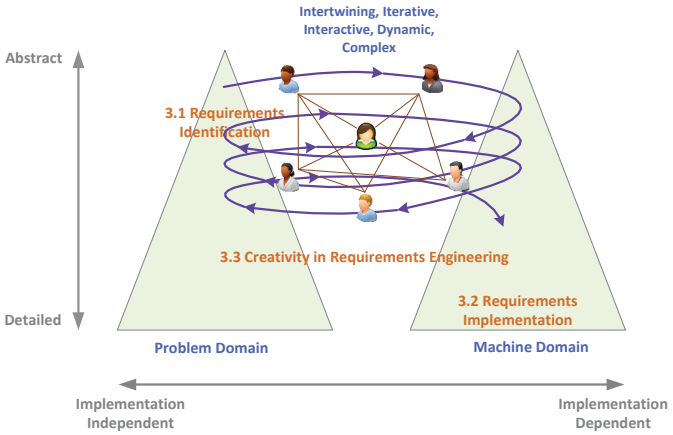

Fig. 3. Using stakeholders' social interaction to unify the "Twin Peaks" in OSS development.

Building upon the research on OSS RE, we try to leverage stakeholders' social interaction to tackle the weaving of "Twin Peaks". This weaving mechanism indeed incorporates the real world scenarios, thereby can address a wide range of evolving and practical "Twin Peaks" activities in a decentralized software development paradigm. Following the literature, we also plan to consider stakeholders' activities over the issue tracking systems in order to operationalize their social interaction. Next, we tease out our research plan to test stakeholders' social interaction as a weaving mechanism of the "Twin Peaks".

III. Research Plan

The central hypothesis of our work is that: "stakeholders' social interaction serves a unified weaving mechanism to link together the Twin Peaks, which offers insights into a wide range of open-source RE activities". Figure 3 conceptualizes our proposed research. We hypothesize that such a mechanism has two main advantages: (1) it is general to account for a wide spectrum of RE activities; and (2) it is ecologically valid [15] so that the research findings can provide potentially transformative benefits to software practitioners.

To test the central hypothesis, we choose to study three activities, namely requirements identification, requirements implementation, and creativity in RE as shown in Figure 3. Horizontally, these activities span the problem and machine domains and their intersection. Vertically, they also cover different levels of details in the engineering process. Our choice follows purposive sampling in that the findings from these representative cases can be used for theoretical generalization [24]. Although, the chosen activities are apparently discrete, we plan to study them from a common perspective, i.e., stakeholders' social interaction. Furthermore, in our study design, these three activities not only strike a proper balance over the "Twin Peaks", but also represent typical and critical cases so that the lessons learned can be informative to a wide variety of important RE tasks. Next we describe in more detail

about each activity and outline our research plan along these dimensions.

A. Requirements Identification

Requirements identification, a human-centric activity involving multiple stakeholders, includes determining the organizational or customer needs that must be addressed by the delivered artifacts [25]. It has been long accepted that stakeholders' social interaction is indispensible in successful identification and negotiation of requirements [26], [27], [19]. In fact, tools and techniques, such as IBM Jazz, focus groups, and repertory grids, have been developed to support social interaction in such RE activities [26], [28], [29], [30]. Despite the development of such technological supports, much less is known about how stakeholders' social interaction and their organizational arrangements influence requirements identification.

Study Plan: Following the conceptual framework of our research in Figure 3, we propose to investigate *how stakeholders' social interaction contributes to new requirements identification in a decentralized OSS development environment*. In particular, we plan to conduct this investigation in light of the theory of new idea generation widely used in sociology and anthropology [15], [31], [32]. According to such theory, diverse social interaction can help a stakeholder contribute new ideas. Therefore, we intend to consider the stakeholders and their social interaction as independent variable of this study, whereas the dependent variable is the number of new requirements proposed by the stakeholders [33]. We operationalize 'new requirements' as the features that are freshly appeared (proposed) and promptly realized (closed). These requirements are interesting because they are both new, practical, and adaptive to the task constraints [34], [35]. We start examining Firefox, Mylyn [36], and CONNECT Gateway [13], three large scale OSS systems in our study. In order to support our operationalization of new requirements, we intend to investigate requirements that have been proposed and realized between releases.

Expected Outcomes: The outcomes of this study are expected to provide insights on stakeholders' social interaction and new idea generation. The preliminary analysis gives us promising results. For example, we notice that stakeholders interacting with a diverse group of people generally contribute a higher number of new requirements. In particular, 37.23%, 30.60%, and 44.83% of the new requirements for Firefox[1], Mylyn [36], and CONNECT Gateway[2] have been proposed by stakeholders with diverse social interactions. The detailed findings of this study are reported in [37].

B. Requirements Implementation

Requirements implementation represents the core phase in realizing requirements in a software system. It is during the implementation phase when the developers write source code that transforms the conceptual abstractions captured by requirements into deliverable features in software [35]. Requirements realization is not an isolated activity performed by sole developers as a developer typically interacts with other stakeholders in order to clarify the technical and organizational constraints [14]. Furthermore, not all newly proposed requirements are immediately implemented by the developers. The requirements are prioritized during implementation and some complicated ones with high coding demand may be postponed for later versions [13]. As requirements implementation in a timely manner is very important in the modern deadline driven software development, developer productivity is crucial in a software system's sustainability. Research has shown that a person's interaction with others can impact her problem solving abilities [15], [38]. Despite the importance of social interaction during requirements implementation [13], [14], little is known about how social interaction among developers and other stakeholders affects developer productivity in terms of implementing the requirements.

Study Plan: Following the proposed research framework (cf. Figure 3), we plan to investigate *how stakeholders' social interaction influences developer productivity in terms of implementing requirements in the OSS development paradigm.* To that end, we plan to consider developers and their social interaction with other stakeholders as independent variable, and developer productivity as dependent variable [33]. Following existing research [39], we operationalize developer productivity as average delta per month, i.e., average number of LOC added and/or deleted by a developer per month. Our intention is to study two large and successful OSS projects, namely Firefox[1] and Mylyn [36]. Existing research suggests that social interaction can lead to improved use of available information, potentially providing a positive effect on a person's productivity [15], [38]. Thus, we anticipate this study will provide interesting findings that can help software practitioners better manage available resources to address time critical requirements.

Expected Outcomes: The outcomes are expected to shed some light on the impact of stakeholders' social interaction on their productivity. The study is currently in progress and we plan to report the results in details in one of our upcoming publications.

C. Creativity in RE

Creativity, defined by Sternberg, can be considered as "the ability to produce work that is both novel (i.e. original and unexpected) and appropriate (i.e. useful, adaptive to task constraints)" [34]. According to Maiden *et al.* [35], creativity in RE is the capture of requirements that are new to the project stakeholders but may not be historically new to humankind. Creative requirements could be obtained by exploring, combining, and transforming existing ideas in the conceptual space [40] that includes problem and machine domains in RE [4], [5]. In this way, creativity in RE can be considered as a product of the intertwined nature of the "Twin Peaks" that allows us to conceptualize swinging and sharing ideas and experiences between the domains (cf. Figure 3). Thus, creativity is placed in between the "Twin Peaks" signifying the importance of both application and implementation domains.

Creativity in RE has been identified to be a collaboratively creative process where collaboration enables exposure to diverse ideas [41]. In an RE environment, therefore, stakeholders' social interaction works as a means to interchange knowledge and concepts between the "Twin Peaks" that can be explored, combined, or transformed to obtain creativity requirements. Techniques, such as generating requirements with scenario, have been proposed to support creativity while exploring information analogical to the current context [42], [43]. To further advance the knowledge of creativity in RE, we plan to investigate stakeholders' social interaction from a combinational creativity perspective, i.e., making unfamiliar connections between familiar possibilities of requirements [44].

Study Plan: We propose to examine *how stakeholders' social interaction can be utilized to achieve combinational creativity in RE*. To that end, we plan to mine existing requirements and associated stakeholder discussions and obtain familiar ideas in terms of dominant topics [45]. In order to find unfamiliar connections between these familiar ideas to generate new requirements, we plan to exploit linguistic techniques, such as part-of-speech (POS) tagging [46]. For example, we can use POS tagging to identify the most common objects and the most common verbs from the topics commonly discussed by stakeholder groups. Flipping the POSs can provide us unfamiliar verb-object combinations that can then be elaborated to obtain new requirements. To further assess the creativity merit of these requirements, we propose to conduct a human subject study considering the requirements as independent variable and the creativity ratings provided by the humans as dependent variable. We consider Firefox[1] and Mylyn [36] as the subject systems and mine existing requirements and stakeholders' comments in order to create new requirements.

Expected Outcomes: Among the expected outcomes of this study are the new requirements for subject systems with promising average creativity ratings provided by human analysts. For instance, the words 'text' and 'number' are found to be a common object and a common verb respectively and flipping the POSs gives us an unfamiliar verb-object combination 'text-number' in the context of Firefox[1]. Elaborating this combination, we obtain a Firefox requirement in our study "Firefox user can text phone number from the web page" with an average creativity rating of 3.5 on a 5-point Likert scale [47]. The findings of this study are detailed in [48].

IV. Progress

So far, some progresses have been made following the research framework proposed in this paper. In particular, we have conducted our planned studies on requirements identification and creativity in RE. The detailed findings from these studies are reported in papers [37] and [48], submitted to a journal and the RE 2014 conference respectively, that are currently under review.

Our study investigating the role of stakeholders' social interaction in identifying new requirements has revealed that generally social interaction positively impact new requirements identification [37]. Further interesting finding from this study, however, is that stakeholder's role together with social interaction plays an important part in this context [37]. These findings attest stakeholders' social interaction as a weaving mechanism of the "Twin Peaks" in open-source RE. Furthermore, this work advances the fundamental understanding about the interplay between stakeholders' organization in their social network and the attributes of the software artifacts contributed by them.

Our study on exploiting stakeholders' social interaction to facilitate combinational creativity in RE [44] has also shown promising results [48]. Following the research plan outlined in Section III-C and detailed in [48], we were able to obtain several new requirements for both Firefox[1] and Mylyn [36] where the human subject study revealed encouraging results about the novelty of these requirements. The outcomes of this study not only stress stakeholders' social interaction as a weaving mechanism but also advance the current solutions that facilitate creativity practice in the RE process of OSS systems.

The proposed study on stakeholders' social interaction and productivity is currently in the data collection and analysis phase. Based on the initial trial analysis conducted on the collected data, we are optimistic about obtaining valuable insights about both the "Twin Peaks" model and the open-source RE process from this study as well. We plan to wrap up the current phase of the study in a couple of months and intend to report our detailed findings in a journal article.

V. Summary and Future Work

In this research proposal, we have formulated a central hypothesis to guide our overall research. The contributions of our work lie in the novelty of the hypothesis that exploits the interaction among the stakeholders as an underlying mechanism to weave a variety of human-centered activities in the "Twin Peaks" of modern open-source RE. Guided by the unified conceptualization, we have developed concrete research plans to test the central hypothesis via three activities: requirements identification, requirements implementation , and creativity in RE. Carrying out the research plans will enable us to gain valuable insights, which we will use to update the assumptions if necessary and to generate guidelines for enhancing software engineering practice in relevant areas.

Our future work includes further examining stakeholders' social interaction from other RE activities, such as requirements prioritization and negotiation. We also plan to conduct further studies to test our hypothesis in commercial software development environment thereby pushing the knowledge from our overall research towards the software development paradigm in general.

Acknowledgment

I express my sincere and deepest gratitude to my advisor, Dr. Nan Niu for his kind support and guidance throughout this research. The research is partially supported by the U.S. NSF (National Science Foundation) Grant CCF-1238336.

REFERENCES

[1] B. Nuseibeh and S. Easterbrook, "Requirements engineering: a roadmap," in *Proceedings of the Conference on The Future of Software Engineering*, ser. ICSE '00, 2000, pp. 35–46.

[2] M. Jarke, P. Loucopoulos, K. Lyytinen, J. Mylopoulos, and W. Robinson, "The brave new world of design requirements," *Information Systems*, vol. 36, no. 7, pp. 992–1008, 2011.

[3] B. Nuseibeh, "Weaving together requirements and architectures," *IEEE Computer*, vol. 34, no. 3, pp. 115–119, 2001.

[4] M. Jackson, "The meaning of requirements," *Annals of Software Engineering*, vol. 3, no. 1, pp. 5–21, 1997.

[5] ——, "Problems and requirements," in *Proceedings of the 2nd IEEE International Symposium on Requirements Engineering (RE)*, 1995, pp. 2–8.

[6] M. Jackson and P. Zave, "Deriving specifications from requirements: an example," in *Proceedings of the International Conference on Software Engineering (ICSE)*, 1995, pp. 15–24.

[7] D. Mou and D. Ratiu, "Binding requirements and component architecture by using model-based test-driven development," in *Proceedings of the 1st International Workshop on the Twin Peaks of Requirements and Architecture (Twin Peaks)*, 2012, pp. 27–30.

[8] M. S. Loft, S. S. Nielsen, K. Norskov, and J. Jorgensen, "Interplay between requirements, software architecture, and hardware constraints in the development of a home control user interface," in *Proceedings of the 1st International Workshop on the Twin Peaks of Requirements and Architecture (Twin Peaks)*, 2012, pp. 1–6.

[9] J. Cleland-Huang, "Meet elaine: A persona-driven approach to exploring architecturally significant requirements," *IEEE Software*, vol. 30, no. 4, pp. 18–21, 2013.

[10] N. Niu, L. Xu, J.-R. Cheng, and Z. Niu, "Analysis of architecturally significant requirements for enterprise systems," *IEEE Systems Journal (Accepted)*, 2013.

[11] T. A. Alspaugh and W. Scacchi, "Ongoing software development without classical requirements," in *Proceedings of the International Conference on Requirements Engineering (RE)*, 2013, pp. 165–174.

[12] X. Xiao, A. Lindberg, S. Hansen, K. Lyytinen *et al.*, "computingrequirements in open source software projects," in *Proceedings of the International Conference on Information Systems (ICIS)*, 2013.

[13] N. A. Ernst and G. Murphy, "Case studies in just-in-time requirements analysis," in *Proceedings of the International Workshop on Empirical Requirements Engineering at RE*, 2012, pp. 25–32.

[14] L. Panjer, D. Damian, and M. Storey, "Cooperation and coordination concerns in a distributed software development project," in *Proceedings of the International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)*. ACM, 2008, pp. 77–80.

[15] P. Pirolli, "An elementary social information foraging model," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, 2009, pp. 605–614.

[16] M. Mirakhorli and J. Cleland-Huang, "Traversing the twin peaks," *IEEE Software*, vol. 30, no. 2, pp. 30–36, 2013.

[17] I. Ozkaya, L. Bass, R. L. Nord, and R. S. Sangwan, "Making practical use of quality attribute information," *IEEE Software*, vol. 25, pp. 25–33, 2008.

[18] W. Scacchi, "Understanding the requirements for developing open source software systems," in *IEE Software*, vol. 149, no. 1, 2002, pp. 24–39.

[19] W. Scacchi, C. Jensen, J. Noll, and M. Elliott, "Multi-modal modeling of open source software requirements processes," in *Proceedings of the International Conference on Open Source Software*, 2005, pp. 1–8.

[20] P. Pirolli, *Information foraging theory: Adaptive interaction with information*. Oxford University Press, 2007.

[21] C. Bird, D. Pattison, R. D'Souza, V. Filkov, and P. Devanbu, "Latent social structure in open source projects," in *Proceedings of the ACM SIGSOFT International Symposium on Foundations of Software Engineering (SIGSOFT/FSE)*, 2008, pp. 24–35.

[22] M. Cataldo and J. Herbsleb, "Communication networks in geographically distributed software development," in *Proceedings of the ACM Conference on Computer Supported Cooperative Work*. ACM, 2008, pp. 579–588.

[23] T. Wolf, A. Schroter, D. Damian, and T. Nguyen, "Predicting build failures using social network analysis on developer communication," in *Proceedings of the International Conference on Software Engineering (ICSE)*, 2009, pp. 1–11.

[24] R. K. Yin, *Case study research: Design and methods*. SAGE Publications, 2008, vol. 5.

[25] I. Sommerville and G. Kotonya, *Requirements engineering: processes and techniques*. John Wiley & Sons, Inc., 1998.

[26] D. Zowghi and C. Coulin, "Requirements elicitation: A survey of techniques, approaches, and tools," *Engineering and Managing Software Requirements*, pp. 19–46, 2005.

[27] E. Kavakli, "Goal-oriented requirements engineering: A unifying framework," *Requirements Engineering Journal (REJ)*, vol. 6, no. 4, pp. 237–251, 2002.

[28] J. P. Rodríguez, C. Ebert, and A. Vizcaino, "Technologies and tools for distributed teams," *IEEE Software*, vol. 27, no. 5, pp. 10–14, 2010.

[29] R. Agarwal and M. R. Tanniru, "Knowledge acquisition using structured interviewing: an empirical investigation," *Journal of Management Information Systems*, pp. 123–140, 1990.

[30] N. Niu and S. M. Easterbrook, "So, you think you know others' goals? a repertory grid study," *IEEE Software*, vol. 24, no. 2, pp. 53–61, 2007.

[31] R. S. Burt, *Brokerage and closure: An introduction to social capital*. Oxford University Press, USA, 2005.

[32] J. N. Cummings, "Work groups, structural diversity, and knowledge sharing in a global organization," *Management science*, vol. 50, no. 3, pp. 352–364, 2004.

[33] D. Montgomery and G. Runger, *Applied statistics and probability for engineers*. Wiley, 2010.

[34] R. J. Sternberg, *Handbook of creativity*. Cambridge University Press, 1999.

[35] N. Maiden, S. Jones, K. Karlsen, R. Neill, K. Zachos, and A. Milne, "Requirements engineering as creative problem solving: A research agenda for idea finding," in *Proceedings of the International Requirements Engineering Conference (RE)*, 2010, pp. 57–66.

[36] M. Kersten and G. Murphy, "Using task context to improve programmer productivity," in *Proceedings of the ACM SIGSOFT International Symposium on Foundations of Software Engineering (SIGSOFT/FSE)*, 2006, pp. 1–11.

[37] T. Bhowmik, N. Niu, and P. Singhania, "On the role of structural holes in requirements identification: An exploratory study on open source software development," *ACM Transactions on Management Information Systems (TMIS) Special Issue on Complexity Systems Evolution: Requirements Engineering Perspective*, (Major revision under review).

[38] S. Clearwater, T. Hogg, and B. Huberman, "Cooperative problem solving," *Computation: The Micro and the Macro View*, pp. 33–70, 1992.

[39] A. Mockus and J. Herbsleb, "Expertise browser: a quantitative approach to identifying expertise," in *Proceedings of the International Conference on Software Engineering (ICSE)*, 2002, pp. 503–512.

[40] M. A. Boden, *The creative mind: Myths and mechanisms*. Routledge, 2003.

[41] M. Mahaux, L. Nguyen, O. Gotel, L. Mich, A. Mavin, and K. Schmid, "Collaborative creativity in requirements engineering: Analysis and practical advice," in *Proceedings of the International Conference on Research Challenges in Information Science (RCIS)*, 2013, pp. 1–10.

[42] I. K. Karlsen, N. Maiden, and A. Kerne, "Inventing requirements with creativity support tools," in *Requirements Engineering: Foundation for Software Quality*. Springer, 2009, pp. 162–174.

[43] K. Zachos and N. Maiden, "Inventing requirements from software: An empirical investigation with web services," in *Proceedings of the International Requirements Engineering Conference (RE)*, 2008, pp. 145–154.

[44] N. Maiden, "Requirements engineering as information search and idea discovery (keynote)," in *Proceedings of the International Requirements Engineering Conference (RE)*, 2013, pp. 1–1.

[45] E. Linstead, C. Lopes, and P. Baldi, "An application of latent dirichlet allocation to analyzing software evolution," in *Proceedings of the International Conference on Machine Learning and Applications (ICMLA)*, 2008, pp. 813–818.

[46] E. Brill, "A simple rule-based part of speech tagger," in *Proceedings of the Workshop on Speech and Natural Language*, 1992, pp. 112–116.

[47] R. Likert, "A technique for the measurement of attitudes," *Archives of psychology*, 1932.

[48] T. Bhowmik, N. Niu, A. Mahmoud, and J. Savolainen, "Ongoing software development without classical requirements," in *The International Conference on Requirements Engineering (RE)*, 2014 (Under review).