What T-shirt Are You Wearing?

Towards the Collective Team Grokking of Product Requirements

Rob Fuller
Electrical and Computer Engineering
The University of British Columbia
Vancouver, Canada
email: rfuller@ece.ubc.ca

***Abstract*—This paper describes requirements engineering research that examines how cross-functional product teams, as a collective, create and nurture a shared mental model that accurately represents the external product domain and its realities and provides the context for understanding the requirements. In the conduct of this study, organisational factors have been identified that support or inhibit teams from achieving this deep collective understanding.**

Keywords—empathy-driven development; collective sensemaking; design science; requirements validation; product team organisation.

I. Introduction

In the late 1990s, a “model revolution” [1] began to emerge that took a new view on change, risk, and uncertainty. These ‘agile’ models typically embraced the possibility that requirements could change throughout the development effort in contrast to many earlier Software Development LifeCycles (SDLCs) that strived to lock down requirements in the specification and planning stages.

These new models had a greater focus on the software development team, usually cross-functional, as a critical success factor in delivering software. These teams often have the necessary collection of functional expertise and capacity in each functional area to be essentially self-sufficient. Many software development companies have gone even further, empowering their cross-functional teams to truly own the product. This approach is now quite common, no longer adopted only by industry thought-leaders. It is these organisations and teams that are the main focus of this research.

The rest of this paper is structured as follows. In Section II, we outline the general problem and in Section III, the general research question and importance. Section IV identifies the focus of the research, who is being sampled in the study. Section V is a summary of the literature review followed by Section VI, where we discuss the research methodology. Section VII briefly describes the status of the research to-date and in Section VIII, emerging observations, which is the main topic of this paper. Finally, Section IX offers thoughts on future direction.

II. The Problem

While agile models improve many of the issues that were breaking down during the crisis period, they still generally cling to the notion that there is a customer, an authoritative voice that the development team can interact with iteratively to clarify requirements and validate results. However, as software development became more product development intended for a market, a new and critical challenge emerged for software teams and that is how to gain a deep understanding of the world for which their product is intended. Certainly, techniques to ‘hear’ from the market are helpful but, as Polyani [2] noted, market participants have tacit knowledge -- people can know more than they can tell and they know more than can be easily observed. This is evidenced by an all-too-common experience when it is discovered that the so-called requirement does not reflect an actual need but rather simply an articulation of what someone wants -- “I know that is what I said I wanted but that does not seem to be what I need.” -- they know more than they can tell.

It is important that the entire team gains this deep understanding because team members (individually, in sub-teams, and in all functional roles) make decisions almost continually based on their understanding of the *context* of the requirements, and much of that context understanding is also tacit.

Thus, it behooves product development teams to strive for a deep, collective understanding of the context of their product, that other world for which their product is intended, a shared mental model of the supra-domain, since many design and implementation decisions are unconsciously made within the team’s understanding of the domain context. The success of the team, of their product, and often of the software company itself rests upon how well they do this. Teams do this with varying degrees of success. Some achieve reasonable success seemingly instinctively, while many struggle ineffectively. Software development leaders are often able to observe this phenomenon but have no theories that help explain why.

While labels are being used to describe what some they think this deep understanding (grokking) of that external world is (e.g., “empathy-driven development”), there does not appear to be any clear definition of what it is, but rather simply labels of what some think may be happening.

III. Research Question and Importance

The purpose of this qualitative study is to develop a substantive theory that answers the following research question:

“what are the factors that support or impede multi-disciplinary software product teams, empowered to own their product, to collectively achieve a deep understanding of the environment for which their product is intended?”

This theory will help industry practitioners explain why certain prevailing techniques and empirical approaches for understanding software solution needs are often inadequate, why some succeed while others do not. These insights will offer guidance for more effective software development approaches.

In addition to assisting practitioners in industry, this interpretive theory aspires to illuminate areas of potential further research. For example, how are technically-oriented people (primarily millennials) working in teams (typically cross-functional) and following a rational process to create software solutions able to develop, nurture, and incorporate 'squishier' skills into a process that strives to be as rational and deterministic as possible? Or, how does that which cannot be easily observed nor expressed be equally understood and preserved within a team? Or, how does empathic appreciation of the context of a software solution translate across individuals, organisations, business domains, cultures?

IV. Focus of the Research

As the saying goes, “a fish does not know it is in water”, thus the intended users often cannot clearly communicate the context in which they operate because they are trapped in that context. Thus, it is necessary for the team to somehow become one of the people targetted to use a software solution, and to truly learn from that immersion. It is very difficult to be an outsider and obtain an insider’s perspective and knowledge. It is especially difficult for a team to do this collectively. And it does not easily fit into established software engineering practices nor is it well-supported by software engineers’ training.

Thus, the focus of this research is practicing software product teams in action, specifically teams empowered to own their product. It also examines the empirical adaptations these teams make to established software engineering and design practices that represent an empathic-based development approach towards evolving an increasingly accurate understanding of the context in which their users operate, the supra-domain - the business needs, technology, culture, and politics. It also examines important organisational factors that either allow or inhibit a team’s ability to collectively grok the domain.

V. Literature Review

Literature was reviewed in 3 areas - requirements engineering (specifically elicitation), design science, and collective sensemaking.

This inquiry may be seen as primarily placed within software requirements engineering, specifically requirements elicitation (attempting to obtain and understand the true needs). We looked at all the accepted papers for the IEEE International Requirements Engineering Conference over the past 10 years, plus many related papers published in other publications. There are increasingly more views being expressed about the shortcomings of prevailing approaches to requirements elicitation which tend to focus on techniques and methods rather than deepening practitioner and team understanding. We believe this is evidence that some software product development efforts still operate in the ‘process-driven’ paradigm and are experiencing what Kuhn [1] described as the incommensurability across paradigms. While acknowledging that the ‘techniques and methods’ approach is entirely appropriate in certain domains, our focus is on problem domains that don’t lend themselves well to clear specifications. Controversy aside, the intent of requirements elicitation is to understand the true software needs and, therefore, this inquiry will contribute to the requirements engineering discipline.

In the design science space, we found a considerable scholarship regarding empathy-driven design, e.g., (Koppen and Meinel [3], van Rijn et al. [4], Postma et al. [5], Woodcock et al. [6], Dong et al. [7], Kourprie and Visser [8], Kolko [9]). However, this research falls short of addressing our inquiry question in three critical respects: 1) the focus is on the design activity as part of an essentially sequential product development process rather than design as part of an on-going continuous product development effort, 2) rather than considering the whole development team, they tend to focus on the design individual or just the design team, and, 3) when it does consider the design team, it tends not to be viewed as a unit to consider regarding its empathic ability. There are design science models described by Wieringa [10] that acknowledge the challenge that empathy-driven requirements understanding attempts to address (using very different vocabulary) but stops short of suggesting how those challenges are, or could be, addressed. We believe this inquiry could enrich those models and generally contribute to the design science field.

In the collective sensemaking field, the collective (team) is usually considered only insofar as its relationship to the organisation, not to its understanding of a specific domain outside of the organisation. Some researchers, notably Daniel Russell [11] look at sensemaking from Human-Computer Interaction (HCI) perspective and, although his view positions sensemaking in a collective context (the information world), he describes a style of engagement of sensemaking that is essentially personal, not collective. The Cynefin framework (Kurtz and Snowden [12]) is a sensemaking framework that is particularly useful for collective sensemaking in that it is designed to allow shared understandings to emerge which could be insightful with respect to how teams ingest, socialise, and collectively store insights. As with other collective sensemaking models, it has resonance in early problem-solving stages and for formal and finite periods of time. Other researchers (Klein et al. [13], Naumer et al. [14], Kolko [15]) elaborate further by bringing data-framing into the picture and defining design synthesis as a process of sense-making, trying to make sense of chaos. The data-framing activity of sensemaking lends itself to being part of a long-term collective effort to understand and therefore may have some relevance to this inquiry.

VI. Method of the Research

As the primary interest is on substantive theory generation, rather than extending or verifying existing theories, we are taking an interpretive epistemological stance, employing a Grounded Theory approach, as developed by Glaser and Strauss [16], and using the Constructivist Grounded Theory methodology described by Charmaz [17]. Grounded Theory is highly applicable in research such as this because it is explicitly emergent and can generate theory relating to a specific research situation. This is an area that is a relatively new, where there has been limited research, and where field data will come from observations and interviews, conditions for which Grounded Theory is particularly well suited.

Grounded Theory has been used successfully as a research method to study Agile software development teams: Adolph et al. [18], Dagenais et al. [19], Coleman and O'Connor [20], Martin [21], Hoda [22].

The research uses theoretical sampling (Charmaz [17]) where the analysis of the data collected prior informs the selection of and inquiry with the next participants. Individual participants and corporate sites selected are ones involved with software product development (teams developing software for market) and that claim to have cross-functional product development teams. The primary data collection methods are observations of team meetings and team interactions, enriched by semi-structured interviews with open-ended questions that can allow real issues to emerge.

Participants are carefully recruited through our professional network and via direct outreach to select software product organisations. Where permitted, we hold interviews in the participant's workplace to allow for environmental context to enrich the interview data. Also, where we have approval from the organisations involved, we locate ourselves as unobtrusively as possible in the workplace to allow for more casual direct observation as an additional data source which may direct further data collection and analysis. The individual interviews conducted are recorded whenever permitted. Our many years of leadership with the types of people that are participants affords us considerable comfort, understanding, and rapid rapport.

Iterative data collection and analysis (formulation, testing, and redevelopment of propositions) allows participant sampling and questions to purposefully evolve as patterns emerge in the data until we reach a theory. We use the NVivo software tool to analyse the unstructured qualitative data collected. Data collection will stop once the analysis indicates the achievement of theoretical saturation, the point at which gathering more data reveals no new properties nor yields any further theoretical insights about the emerging grounded theory (Charmaz [17]). This ensures a degree of consistency in the analysis.

Our professional experience allows for a certain considered positionality and we recognise that this shapes our objectivity and subjectivity of many aspects of perspective in this study. While acknowledging the challenges, we consider this experience, and the bias it creates, to be an asset to this research. As Eisner [23] suggests, the expert ability to "see what counts" -- the sensitivity to tacit elements of the data, meanings and connotations -- will guide the research, supported fully by the collected data, towards questions that matter.

Quality in research of this nature is generally assessed in terms of validity and generalizability, which, together, determine some measure of usefulness. During the research, we employ various strategies (Maxwell [24]) to mitigate threats to validity (credibility, dependability, reliability). Intensive, on-going involvement, e.g., extended participation and the ability to live in the participants' workplace, provides richer data and data types, less dependence on inference, and opportunity for repeated observations and interviews, all which will help rule out spurious associations and premature theories. The collection and use of rich data (transcribed interviews, thick descriptive note-taking of observations) help provide a more complete and revealing picture of what is going on. Participant checks (obtaining participant and peer feedback on the data collected and conclusions drawn) help rule out possibilities of misinterpretation. Triangulation (collection from a range of participants and settings) reduces the risk of chance associations and systematic biases. Finally, we will be transparent with any discrepant evidence or negative cases. We intend to assess transferability of final results within the context of software product development primarily via reviews of the resulting theory with software product development leaders and, further, to draw comparisons with non-product software development teams to further refine the specificity of transferability claims.

VII. Status of the Research

To-date, we have been working with 5 software companies, 4 of which produce commercial enterprise-class software products (skills management, retail, stem cell therapy, and social media marketing management) and using common Agile approaches. The 5th company develops large-scale aerospace and satellite systems and adopts some Agile philosophies within a large-scale, structured project management methodology. All companies are leaders in their markets. They range in size from 10 to several hundred employees and in maturity from 2 to 50 years old. To-date, 9 product development teams across these companies have participated, resulting in 15 individual semi-structured interviews conducted and 17 team sessions observed. More data gathering is scheduled and more organisations and teams are being recruited.

VIII. Emerging Observations

The first emerging observation is that the organisational model surrounding the cross-functional team significantly impacts the cross-functional team dynamics, individual participation and sense of primary allegiance. Where there is, e.g., a software engineering department, a design department, and a product management department, contributing members to cross-functional product teams, the intra-team dynamics are often strikingly different than when there is no functional organisation surrounding the teams. In the former case, team members are more likely to temper their contributions, identifying more with their functional affiliation than with the product mandate. The analogy we use here is that each are wearing a functional t-shirt (e.g., the software engineering department t-shirt with a small insignia that indicates the person is assigned to a particular product team at the moment). In addition to observing this in team interactions, this also appears in the language, "*I just do my job and they do theirs*", "*I trust them*", "*I think someone else is looking after tha*t", "*I just do what Product Management (or Design) says*", "*I am on this team -- for now*". A software engineer in this environment is much more likely to care about the how and defer to others on the what and why. In contrast, organisations that do not have a functional structure surrounding the cross-functional product teams tend to have teams with a more complete sense of ownership for their product and richer intra-team interactions. The t-shirt analogy here is that they are all wearing the same product t-shirt with perhaps an insignia that identifies their functional competency. On these teams, sense of *team* is much stronger, thus the language does not refer to 'them'. All team members are more likely to care about what, why, and how because there is a stronger sense of collective ownership for the product, not just their particular contribution to it. We plan to probe this phenomenon further and look at definitions of success and how they may be defined similarly or not across these two models.

The second emerging observation relates to expectation of mobility which appears to be inversely related to an individual's intrinsic connection to the product and, therefore, the product team. We have observed two pressures that inhibit an individual's inclination to be all-in. One pressure is where there is a high degree of staff turnover impacts product development team resourcing. After a certain length of time, people in these environments come to expect they will be reassigned soon and thus have a certain tentativeness to their commitment to the product and the product team. The other pressure is similar, however, more intentional, where there is a Human Resource department policy that encourages a high degree of mobility with respect to team assignment, e.g., 20% of technical staff should change teams every year. This seems to be rooted in a belief that mobility is healthy for the individual and/or adds to corporate robustness. A telling quote from an engineering manager in one of these situations, *"I do not know how a true 'team' can emerge this way."*.

IX. Conclusion and Future Work

Product development is a social process; thus, the organisational dimension is the elephant in the room, a critical factor for success or failure of software product teams. The two observable phenomena surfacing strongly in the analysis to-date both fall into a category of what an organisation may do, consciously or otherwise, to support or inhibit, a cross-functional team to be all it can be. There appears to be a certain blurring of functional boundaries necessary for a team to become a true product team rather than a collection of functional experts assembled around a product.

In the context of requirements engineering, we use the definition of empathy to be *the ability to imaginatively step into another domain, understand the perspectives of those in that domain, and use that understanding to guide decisions* [25]. Stepping into that other domain also involves a certain temporary *blurring of the boundaries* in order to truly understand perspectives in that domain. Thus, further inquiry is needed to determine if this blurring of functional boundaries is a necessary condition for the team to collectively grok the target domain.

Acknowledgment

This work is supported in part by the Institute for Computing, Information and Cognitive Systems (ICICS) at UBC.

References

[1] T. S. Kuhn, The Structure of Scientific Revolutions. 4th ed. University of Chicago Press, 2012.

[2] M. Polanyi, "The Tacit Dimension," Knowledge in Organisations, 1997.

[3] E. Koppen and C. Meinel, "Knowing People: The Empathetic Designer," Design Philosophy Papers, vol. 10, no. 1, pp. 35-51, 2012.

[4] H. Van Rijn, F. S. Visser, P. J. Stappers, and A. D. Özakar, "Achieving empathy with users: the effects of different sources of information," CoDesign, vol. 7, February 2015, pp. 65–77, 2011.

[5] C. Postma, E. Zwartkruis-Pelgrim, E. Daemen, and J. Du, "Challenges of Doing Empathic Design: Experiences from Industry," Int. J. Des. Vol 6, No 1, pp. 59-70, 2012.

[6] A. Woodcock, D. McDonagh, J. Osmond, and W. Scott, "Empathy, Design and Human Factors," Advances in Usability and User Experience, pp. 569-579, 2018.

[7] Y. Dong, H. Dong, and S. Yuan, "Empathy in Design: A Historical and Cross-Disciplinary Perspective," Advances in Neuroergonomics and Cognitive Engineering, pp. 295-304, 2018.

[8] M. Kouprie and F. S. Visser, "A framework for empathy in design: stepping into and out of the user's life," Journal of Engineering Design, vol. 20, no. 5, pp. 437–448, 2009.

[9] J. Kolko, Well-Designed: How to create empathy to create products people love. Harvard Business Review Press, 2014.

[10] R. Wieringa, Design Science Methodology for Information Systems and Software Engineering. Springer, Berlin, 2014.

[11] D. Russell and P. Pirolli, "An Overview of Sensemaking: A View from the Workshop CHI 2009," Sensemaking Work. CHI, pp. 1–2, 2009.

[12] C. F. Kurtz and D. Snowden, "The New Dynamics of Strategy: Sense-making in a Complex-Complicated World," IBM Syst. J., vol. 42, no. 3, pp. 462–483, 2003.

[13] G. Klein, B. Moon, R. Hoffman, and K. Associates, "Making Sense of Sensemaking 2: A Macrocognitive Model," IEEE Intell. Syst., vol. 21, no. 5, pp. 88–92, 2006.

[14] C. Naumer, K. Fisher, and B. Dervin, "Sense-Making: a methodological perspective," CHI2008 Work. SenseMaking Florence, 2008.

[15] J. Kolko, "Sensemaking and Framing: A Theoretical Reflection on Perspective in Design Synthesis," 2010 Des. Res. Soc. Conf., pp. 1–9, 2010.

[16] B. G. Glaser and A. L. Strauss, The discovery of grounded theory: Strategies for qualitative research. Piscataway, NJ: Aldine Transaction, 1967.

[17] K. Charmaz, Constructing grounded theory: a practical guide through qualitative analysis. London: Sage, 2006.

[18] S. Adolph, W. Hall, and P. Kruchten, "Using grounded theory to study the experience of software development," Empirical Software Engineering., vol. 16, no. 4, pp. 487–513, 2011.

[19] B. Dagenais, H. Ossher, R. K. E. Bellamy, M. P. Robillard, and J. P. De Vries, "Moving into a New Software Project Landscape," in ICSE '10 Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering, pp. 275–284, 2010.

[20] G. Coleman and R. O'Connor, "Using grounded theory to understand software process improvement: A study of Irish software product companies," Information Software Technology, vol. 49, no. 6, pp. 654–667, 2007.

[21] A. M. Martin, "The Role of Customers in Extreme Programming Projects," PhD thesis. Victoria University of Wellington, New Zealand, 2009.

[22] R. Hoda, "Self-Organizing Agile Teams : A Grounded Theory," PhD thesis. Victoria University of Wellington, New Zealand, 2011.

[23] E. W. Eisner, The enlightened eye: Qualitative inquiry and the enhancement of educational practice. Upper Saddle River, NJ: Prentice-Hall, 1998.

[24] J. A. Maxwell, Qualitative research design: An interactive approach. Thousand Oaks, Calif.: SAGE Publications, 2012.

[25] R. Krznaric, Empathy: why it matters, and how to get it. New York: Penguin Random House, 2014