Blurring Boundaries:

Towards the Collective Team Grokking of Software Product Requirements

by

Robert C. Fuller

B.Sc. (honours), The University of British Columbia, 1974

MBA, Syracuse University, 1984

M.Sc., University of Alberta, 2003

A DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

(Electrical and Computer Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

December 2022

The following individuals certify that they have read, and recommend to the Faculty of Graduate and Postdoctoral Studies for acceptance, the dissertation entitled:

Blurring Boundaries: Towards the Collective Team Grokking of Software Product Requirements

submitted by Robert C. Fuller in partial fulfilment of the requirements for

the degree of Doctor of Philosophy

in Electrical and Computer Engineering

Examining Committee:

Philippe Kruchten, Professor, Electrical and Computer Engineering, UBC
Supervisor

Julia Rubin, Assistant Professor, Electrical and Computer Engineering, UBC
Co-supervisor

Ali Mesbah, Professor, Electrical and Computer Engineering, UBC
Supervisory Committee Member

S. Sidney Fels, Professor, Electrical and Computer Engineering, UBC
University Examiner

Carson Woo, Associate Professor, Accounting and Information Systems, UBC
University Examiner

Abstract

As market expectations of software products continues to become more sophisticated and the competitive landscape for software products grows in intensity, the difference between successful and unsuccessful software products is rapidly becoming less a function of software development methods and more one of how deeply and collectively cross-functional software product development teams achieve a tacit understanding of the product domain, thus creating a context for the team to understand the product requirements and a valid context for the implicit and explicit decision-making that occurs throughout the product development life cycle. Although deep domain understanding cannot be obtained through prevailing requirements engineering methods the way they are usually practiced, many software product development teams do manage to achieve varying degrees of collective grokking of the product domain. However, little is known about what factors support or impede these teams in collectively achieving this deep understanding.

Looking to identify factors that would explain why some teams collectively grok the product domain more deeply than others, I used the Constructivist Grounded Theory research method over a period of three years to study individuals and teams across seven software companies that create products for a diverse range of markets. I found that certain factors of the corporate organisational structure and the product planning process play a significant role in product development teams' potential to collectively develop deep domain understanding. These factors also impact individual and development team dynamics. I identify two essential metaphorical dynamics of broadening the lens and blurring boundaries that successful cross-functional product

teams employ to fully embrace product ownership, visioning, and planning toward achieving this rich context for understanding product requirements.

This study concludes also that the highly specialised nature of many organisational models and development processes is contraindicated for cross-functional product development teams in achieving this deep collective understanding and calls for a revisiting of the mechanistic organisational and product planning practices for software product development. Further, it calls for a shift of emphasis in requirements engineering towards a greater focus on the human factors in requirements engineering, specifically the collective and tacit understanding of requirements and their context.

Lay Summary

Technology products are becoming increasingly complex and often developed by cross-functional teams. An important success factor of these products is how well these development teams obtain a sophisticated empathic and collective understanding of the context of the products. Much of this contextual understanding is tacit knowledge that cannot be easily expressed and, therefore, which the team must often develop for itself. This study examines the characteristics of technology product development teams that do and do not demonstrate aspects of this deep collective understanding as well as characteristics of organisational structures and development process models that support or inhibit teams in achieving this. It finds that high degrees of specialisation which identify clear distinctions of responsibility by individuals and/or teams inhibit the cross-functional teams' ability to develop a collective sense of ownership of the product and a deep collective understanding of the tacit aspects of the context for the product.

Preface

The identification and design of the research program, the execution of all aspects of the research, and the analysis of the research data were all performed solely by the author.

Parts of this dissertation have appeared in previously published work:

- Parts of Chapters 1-3 were previously published by Fuller, R. (2018). Cross-functional teams that grok it: The collective empathic understanding of product requirements. In *Proceedings - 2018 IEEE 26th International Requirements Engineering Conference, RE 2018*, 466–471. Banff, Canada: Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/RE.2018.00063
- Parts of Chapters 1-6 were previously published by Fuller, R. C. (2019a). What T-shirt Are You Wearing? Towards the Collective Team Grokking of Product Requirements. In *The Fifth International Conference on Advances and Trends in Software Engineering, SOFTENG 2019*,,37–40. Valencia, Spain: IARIA. https://www.thinkmind.org/index.php?view=article&articleid=softeng_2019_3_10_90034
- Parts of Chapters 1-6 were previously published by Fuller, R. C. (2019b). Functional organization of software groups considered harmful. In *Proceedings - 2019 IEEE/ACM International Conference on Software and System Processes, ICSSP 2019*,.120–124. Montreal: Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/ICSSP.2019.00024

- Parts of Chapters 1-6 were previously published by Fuller, R. C. (2019c). Blurring Boundaries: Towards the Collective Team Grokking of Product Requirements. In C. Cappiello & M. Ruiz (Eds.), *Information Systems Engineering in Responsible Information Systems, CAiSE Forum 2019,* 87–98. Rome: Springer International Publishing. https://doi.org/10.1007/978-3-030-21297-1_8
- Parts of Chapters 1-6 were previously published by Fuller, R. C. (2020). Broadening the Lens: Towards the Collective Empathic Understanding of Product Requirements. In B. Stepien, University of Ottawa, Canada (Ed.), *The Sixth International Conference on Advances and Trends in Software Engineering, SOFTENG 2020,* 16-21. Lisbon, Portugal: IARIA. http://www.thinkmind.org/index.php?view=article&articleid=softeng_2020_2_10_90031
- Parts of Chapters 1-6 were presented by Fuller, Robert C. (2020). Blurred Boundaries, Towards the Collective Team Grokking of Software Product Requirements at *The 15th Annual Conference of the Interdisciplinary Network for Group Research (INGRoup 2020).*
- A summary of this dissertation, Chapters 1-6, was previously published by Fuller, Robert C. & Kruchten, Philippe (2021). Blurring Boundaries: Toward the Collective Empathic Understanding of Product Requirements. In *Information and Software Technology, 140 (2021),* 106670. https://doi.org/10.1016/j.infsof.2021.106670

This dissertation makes use of the following copyrighted material for which permission has been obtained:

- Figure 3-1 Zig-zag Approach to Data Collection and Analysis (J. W. Creswell, 2012, p. 433) is adapted and republished with the permission of the copyright holder Pearson Education Inc.
- Figure 5-5 People in a box is republished with the permission of the copyright holder zetwe – Can Stock Photo Inc.
- Figure 5-6 People outside the box is republished with the permission of the copyright holder Publitek, Inc.
- Figure 5-9 Job Characteristics Model (Hackman & Oldman, 1980, p. 90) is adapted and republished with the permission of the copyright holder Addison-Wesley
- Figure 5-10 Telephone Game is republished with the permission of the copyright holder Marketoon Studios

This research was approved by the University of British Columbia Behavioural Research Ethics Board (BREB), H16-03352.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

CFPT	Cross-functional Product Team
DWTTY	Do What They Tell You
GUI	Graphical User Interface
HCI	Human-Computer Interaction
IKIWISI	I'll Know It When I See It
MVP	Minimal Viable Product
NaPiRE	Naming the Pain in Requirements Engineering
QDAS	Qualitative Data Analysis Software
RE	Requirements Engineering
SDLC	Software Development Lifecycle
UX	User Experience
XP	Extreme Programming

Acknowledgements

I offer my gratitude to Prof. Philippe Kruchten for his guidance, for the long leash offered me, and for his uncanny ability to always being able to, within arm's reach, put his hand on material relevant to our discussion.

I offer gratitude also to all the participants in this study. Your willingness to participate and especially your enthusiasm for the topic helped validate that this effort was worth it.

Special thanks will never be enough to my wife, Liz, for tolerating me through this journey.

.

Dedication

In memory of my late parents Joyce and Stan

My eternal gratitude for your steadfast support throughout my life. While never understanding this particular intellectual pursuit, your trust in me and in my intellectual restlessness never wavered.

"It is not unreasonable that we grapple with problems. ... Our responsibility is to do what we can, learn what we can, improve the solutions, and pass them on." -- Richard P. Feynman

"I have no special talents. I am only passionately curious" -- Albert Einstein

Chapter 1: Introduction – "I Am All That I Grok"

Of the software product development teams that you are familiar with, ask yourself how many of them actually *grok* the world for which their products are intended? In Robert A. Heinlein's famous 1961 science-fiction novel, *Stranger in a Strange Land* (Heinlein, 1961), Mahmoud explained that to *"'grok' means to understand so thoroughly that the observer becomes a part of the process being observed"* and he later declared, *"I am all that I grok"*. Not to ignore a pop culture word that was not going away, the Oxford English Dictionary later defined the verb *grok* as to 'understand (something) intuitively or by empathy' (Oxford English Dictionary, 1989).

Successful software products are almost always the result of creativity, innovation, and vision. And this success comes from the creativity, innovation, and vision of the entire team, not from any one individual. These elements occur throughout the entire life of the product, not as a single act of creation nor confined to a finite project or phase. In other words, a successful software product is created from software product development teams that collectively *grok.* This research is about grokking, it is about collective grokking, and it is about collective grokking by software product development teams. It is about the product development team's collective grokking of the product domain for the purpose of gaining a deeper understanding of the product requirements and, even more importantly, a deeper understanding of the *context* of those requirements. It is within this context that development teams better understand the product requirements, both expressed and unexpressed, and develop the software product with that deeper understanding. If Mahmoud were part of a software product development team, he might declare, "*we* are all that *we* grok".

Why is this even a topic of interest? Requirements Engineering (RE) is a troubled discipline, as I describe in more detail in this chapter and the next. As noted by Wagner et al. (2019), RE lacks theory. And the methodological practices that do exist in RE not only fail to make project success a likelihood, they barely help avoid project failure. This is evidenced by *incomplete/missing requirements* being cited (Fernandez et al., 2017) as by far the most common reason for project failure, even amongst those organisations using established RE practices. Is it possible that Requirements Engineering research has been aimed at the wrong elements of the problem all along? Is it possible that the label of Requirements Engineering, with all its implications, is the elephant in the room that no one wants to talk about, namely that, for software products, it is not about *requirements* as much as it is about *opportunity* and, further, that perhaps requirements management is not an engineering discipline at all? As I present my research on software product development teams and what supports or inhibits their ability to collectively grok the product domain, I shed further light on these, and other, important questions.

In the remainder of this chapter, Section 1.1 explains the motivation for this study, including my own personal motivation. Section 1.2 offers a background to the requirements understanding problem in terms of the history and evolution of the software development discipline, a description of the requirements understanding problem, and a framing of the research focus. Section 1.3 explores the apparent gap between much of the literature on requirements engineering and the issues faced in industry practice. Finally, Section 1.4 frames the research

objective, offers a high-level rationale for my choice of research method, and outlines the structure of this dissertation.

1.1 Motivation for this Study

Highsmith (2000) noted that exploratory problems, such as those often found in software product development, are defined by uncertainty, risk, change, and speed. He goes on to note that "solving these problems requires … teams that are innovative and creative" (p.99). Hoegl and Gemuenden (2001) showed that teamwork quality and the success of innovative projects is a measure of cohesion, coordination, mutual support, and coordination. Thus, given the complexity of software products, the rate of change of technology and requirements, and the essential dependency on innovation, successful software products are more likely to be created by teams exhibiting these characteristics of cohesion, coordination, mutual support, and coordination than they are by workgroups, simple assemblies of talented individuals. During my long industrial experience, I continually observed that strong software product development teams vary considerably in their ability to collectively grok their product's domain. The reasons for this, however, were not clear to me as a practitioner nor were they to my professional peers. I knew that if the reasons for this variability were better understood, software product development leaders could use that understanding to proactively influence the factors at play - factors that impact the success of their software product teams, the success of the software products, and the success of their companies. Creating abstractions that are actionable by industry practitioners in addressing this issue was the essential motivation for this research. I also identify several related areas of potential inquiry that I feel are of great academic and

industrial importance and so an additional motivation for this study is to provide contribution towards those future studies.

1.2 The Problem

Despite the discipline of software development always having been in the eye of the storm with respect to constant, and often dramatic, change, there remain areas where the discipline clings to the past, seemingly in the hope that the changes are changes of form rather than in kind or at least can be addressed with abstractions of existing methods. When the software is being developed for a market, with no identifiable customer paying for an agreed-upon deliverable with broad and nebulous market dynamics to consider, structure and methods that assume a customer-supplier relationship break down by not providing cross-functional product development teams sufficient domain understanding and context for the software product requirements and for the product as a whole.

1.2.1 Background – How Did We Get Here?

As the 20th century drew to a close, three forces had taken hold which dramatically changed the nature and challenge of software product development. Bill Gates has been quoted as saying that Microsoft was founded with a vision of *"a computer on every desk, and in every home"*. The first of these three forces that I refer to is Microsoft's vision becoming a reality with the computer industry breaking through a price/performance threshold for personal computers which quickly brought computing capability to almost every desk in the workplace and to almost every

home. Thus, was created a significantly larger and more diverse demand for software applications.

The second force was the widespread introduction of graphical user interfaces (GUIs), primarily fuelled by the Macintosh and Windows operating systems. This dramatically enriched the possibilities and complexities for human-computer interaction (HCI) and opened up new dimensions for computer use in both the workplace and in the home, resulting in a much more diverse population of people interacting with software. This richer interaction brought with it an almost exponential growth in the complexity and importance of a critical consideration for software development, namely *usability*, which rapidly grew into software developers having to understand *how* the software would be used and the context of that use, something that heretofore had been quite obvious, and therefore, much less critically important.

Finally, the third force was the introduction and growth of the Internet which fuelled the emergence of totally new business models by introducing entirely new possibilities for accessing data and for using information technology as it was known then (Wasserman, 2011).

In addition to these three forces increasing the art of the possible in software (as well as significantly increasing the complexity of software design, development, and testing), these forces resulted also in more software being developed as *products* for a market (Wasserman, 2011) with *potential* customers, shifting away from the prevailing model of software being developed as bespoke solutions for *known* customers (including internal application development), customers that were paying for the development of the software rather than

simply purchasing a license to use. This bespoke software development model had been the mainstream model for software development before these three forces took hold. The shift towards more software development as a *product* was significant because it alone introduced another significant (and, as will be shown, inadequately addressed) factor of risk and uncertainty into the entire software development process.

These conditions had all the signs of a crisis in the software industry and, in response to this, a Kuhnian *model revolution* (Kuhn, 2012) emerged during this crisis period. This model revolution took a new view on change, risk, and uncertainty within software development in general and this new view was particularly critically important for software *produc*t development. This new *agile* paradigm (Agile Alliance, 2001) accepted that requirements or the understanding of market needs could (and probably, would) change throughout the development lifecycle which was now a *product* life rather than just a *project* life. It asserted that better and deeper understanding of the requirements would emerge throughout the development effort in contrast to contemporary, more structured, Software Development Life Cycles (SDLCs) that strived to lock down requirements in the specification and planning stages in order to minimise the uncertainty of development timeframes, costs, and deliverables. By taking forms of iterative and incremental approaches to solution development and using cross-functional teams to attempt to *discover* the needs throughout the development effort, these new paradigmatic approaches accepted and viewed emergence as a fact of life rather than a failure of the requirements elicitation and analysis activities. "Embrace change" wrote Beck (2004) in his seminal book on eXtreme Programing (XP).

This acceptance of uncertainty along with an increased emphasis on learning throughout the development process placed a new and greater focus on the software development team, recognizing that prescriptive processes were insufficient to ensure project and product success in these complex and emergent conditions. This new focus considered the dynamics of the development team and the team's understanding of the problem domain as critical success factors in delivering software. Thus began a shift towards more cross-functional development teams with the potential to become more empowered and truly own the software product or identifiable portions of it.

The statement from the Agile Manifesto, "Individuals and interactions over processes and tools" (Agile Alliance, 2001) rang loud and clear for those adopting this new paradigm for software development. The *how* to do this would turn out to be a difficult, an oft-misunderstood, and an on-going, problem.

Out of this paradigm shift emerged another entirely new challenge for software product development leadership, one that many were ill-prepared for. Gone were the days of clear objectives, frozen specifications, and simple technologies. Leaders now needed to learn how to build a blended culture, use generative rather than inclusive rules, and foster emergent rather than cause-and-effect practices (Highsmith, 2000). They now needed to lead teams and teams-of-teams successfully under these conditions of rapid change and uncertainty while no longer having a prescriptive development methodology to guide them.

1.2.2 Is It Requirements Engineering or Requirements Understanding?

While the agile approaches that emerged improved many of the issues that had been breaking down during the crisis period, they fell silent on the issue of the development team's understanding of the problem domain. These approaches generally closed their eyes to the issue, choosing to still refer figuratively to a *customer*, whether one actually existed or not. Even today, the product development process in many software product companies still operates as if they are developing for a single customer or stakeholder. Yet, most often, they are not and so they will then often figuratively anoint an internal surrogate (the so-called *"customer on-site"* of XP), hopefully as an authoritative voice with which the development team can iteratively interact to clarify requirements and validate results. These internal roles may carry various titles such as *Product Owner*, *Product Manager*, *Market Analyst*, *Customer Researcher,* etc.

However, as software solutions address more varied and subtle needs, as software technology continues to become increasingly complex, and as software development becomes often more *product* development creating solutions for a whole market as opposed to bespoke development intended for a single customer, a critical challenge emerges, namely how software product development teams gain a deep understanding of the world for which their product is intended, an understanding that cannot be passed on to the team by *any* single voice, much less an internal one that is not part of that product domain world. Certainly, there are analysis techniques, e.g. from marketing or design thinking disciplines, that attempt to *hear* the market and learn about it. Although these are helpful, market inhabitants have tacit knowledge which is rarely, if ever, captured by such methods. As Polyani (2009) states *"people know more than they can tell"*, and

they know more than can easily be observed. A different, yet related, form of this problem commonly occurs with the popular *user story* technique of communicating end-user requirements (Cohn, 2004) when it's later discovered that the story doesn't reflect an actual need but rather simply an articulation of what someone wants, resulting in, *"I know that's what I said I wanted but that doesn't seem to be what I need."* … they know more than they can tell. Sometimes, they cannot even express what they want at all, resulting in IKIWISI (I'll Know It When I See It). Boehm (2000) described an additional form of IKIWISI where the customer initially *thinks* they know what their needs are but their understanding of those needs change over time with continued use.

In earlier times, requirements were less complex. In the past, technology limitations significantly constrained what was possible therefore software needs could be expressed more precisely, unambiguously, and completely. Quite often the requirements came from an identifiable customer with whom the team could validate requirements understanding and agree upon acceptance criteria. In these circumstances, techniques such as having at least one domain expert on, or available to, the team were often sufficient. Today, however, with much more technical and problem complexity, heterogenous customer targets, opaque and rapidly changing market conditions, competitive factors, etc., it is insufficient to simply have one person representing the market domain even if they do have deep, comprehensive understanding because it is only one person's perspective. And, it is even less sufficient to have this deep understanding residing outside the development team. Yet, many software product development organisations continue to operate this way, often resulting in *requirements fixation* (Mohanani et al., 2014) and achieving disappointing results.

Instead, it is important that *everyone* on the development team have a deep domain understanding and it is critical that everyone understands it in a compatible and consistent way. The reason this is so important is that team members (individually, in sub-teams, and across all functional roles on the team) make decisions continually throughout product design and development, indeed the entire product lifecycle, based on their individual understanding of the context of the requirements, and much of that context understanding is tacit. This challenge is well expressed by Berry (1995) when discussing assumptions in requirements engineering amongst team experts:

> It seems that among experts, a common disease is the presence of unstated assumptions. Because they are unstated, no one seems to notice them. Worse than that, it seems that no two people have the same set of assumptions, often differing by subtle nuances that are even more tacit than the tacit assumptions. It is these assumptions that confound attempts to arrive at consensus, particularly because none of the players is even consciously aware of his or her own assumptions and certainly not of the differences between the players' assumptions (p.180)

Thus, it behooves product development teams to strive for a deep collective understanding of the context of their product, a shared mental model of all the elements of the domain, since many decisions made throughout the development lifecycle are made unconsciously and those unconscious decisions are made within the team's understanding of the domain context.

In other words, it behooves them to *grok* that world to the best of their ability. When the term *grok* is used in this dissertation, it refers to cognitive empathy, coupled with skilled perspective-taking. Specifically, I use the definition of cognitive 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" (Krznaric, 2014). Increasingly, the success of software product development teams depends on the degree to which the team *collectively* groks not only the product requirements themselves but also, and very importantly, the *context* for those product requirements and then is able to use that collective understanding of the requirements and context to guide their development decisions, explicit and tacit.

Requirements Engineering (RE), as it is commonly referred, is a set of activities concerned with identifying and communicating the purpose of a software system. The *engineering* term implies that systematic and repeatable techniques are used on requirements to ensure consistency, validity, coverage, etc. While these are important, gaining a full understanding of that other *real* world, is not something that can be accomplished by systematic methods and techniques. For this reason, I argue that Requirements Engineering, as commonly defined, does not address the critical need for Requirements *Understanding*.

To illustrate, I offer a small example from my professional experience of how existential requirements understanding can be for software firms. Two software entrepreneurs, after discussing an operational need with someone in their professional network, decided to start a software company to build a product to address that need. Their network contact worked for a

very well-known and prestigious firm and offered reasonable assurance that the company would license the product once it was built. With the enthusiasm of having product validation and market currency from this well-known company as a launch customer, the start-up was quickly able to secure financing to fund the product development and other start-up costs. The first version of that product was built and licensed to that initial customer whose needs were reasonably well-met by the product, much as everyone involved expected. However, the software start-up was never able to secure a second customer. Other stories can be told where requirements elicitation was limited to customers and potential customers, overlooking the often existentially critical need to identify and respond to the competitive landscape.

Markets are heterogeneous and one company does not reflect the whole of a market. Without deep domain knowledge, this start-up had no context in which to place the requirements they were hearing from this first customer. This lack of deep context limited their ability to validate the completeness of their understanding and it prevented them from being able to assess the relative importance of the requirements they were given against those requirements they were not being given. And finally, without that deep domain understanding, they were unable to determine which needs were unique to this customer versus which might reflect broader market needs.

Not seeing continued traction in the market, the investors were unwilling to fund the company further. It rapidly ran out of funds and closed its doors.

1.2.3 Research Focus and Question

As the saying goes, "a fish doesn't know it's in water", and so the intended users of software products often cannot envisage an ideal or even a conceptually different solution to their needs. Being trapped in that context, they are often unable to even clearly communicate the context in which they operate. So, for software product development teams to understand and define that which they cannot easily see, to understand the *why* more than the *what*, to understand the functionality needs, the supra-functionality needs (attributes that satisfy needs beyond the utilitarian functional needs, including the emotional and cultural relationship between the products and the user), and the context of all those needs, it is necessary somehow for the team to figuratively become one of the those targeted to use the software solution, and to truly learn from that immersion. In other words, it is necessary for them to grok, to understand so thoroughly that they figuratively become a part of the process being observed ("I am all that I grok").

This is difficult. It involves somehow blurring the perceived boundaries between the team's world and the target domain (e.g., a development team in a small, early-stage software product company and a Fortune 100 marketing department). It is difficult to step back from one's own opinions, to learn, observe, and even participate without judgement and it is particularly difficult to be an outsider and obtain an insider's perspective and knowledge. It is also logistically messy. Considering that software solutions are a result of a collaborative cross-functional team effort, this difficulty and messiness is even more acute.

The focus of this research began then on practicing software product teams in action, intending to explore the question of what the internal team factors are that explain differences in team success in collectively grokking the product domain. Specifically, my primary interest is with teams empowered to own their product or some well-defined subset of their product. In this study, I also included teams that were, intentionally and unintentionally, not so empowered. For further contrast, the study included software firms that develop bespoke solutions. The study considered empirical adaptations that these teams make to established software product development practices and methods in order to further their grokking of the context in which their users operate, what I refer to as the *supra-domain* — the business needs, technological, cultural, and political context of the product domain. The research also examined how software product development individuals and teams, who are trained and encouraged to apply their best judgement, suspend those judgements and opinions at critical times in order to connect with and exercise empathy for the domain for which their solution is intended.

Early in this study, however, it became evident that there were differences across participant organisations that were even more pronounced than the differences across teams within any given participant company, suggesting that there might be broader contextual factors at play. In my industrial experience, I had been observing teams in a single environment at any given time so had not witnessed such phenomena across organisations nor had this factor surfaced in peer networking discussions. Specifically, I observed (Fuller, 2019a) that the organisational model (the composition of specific functional groups and the manner in which teams are formed) surrounding cross-functional product teams (CFPTs) has a significant impact on the team itself – on the intra-team communication dynamics, on both the individual and collective sense of

ownership of, and commitment to, the product vision, goals, and plans, as well as on the collective capability of the team to grok, to deeply understand the product domain. This early finding caused me to modify the scope of the study to examine factors which were both internal and external to the teams and to shift the primary focus from factors within individual teams to factors within individual software organisations. I discuss this early finding and my response to it in more detail in Section 4.2. From this, the following specific research question (RQ) emerged and remained throughout the remainder of the study:

> RQ: "What factors support or impede cross-functional software product teams in collectively achieving a deep understanding of the environment for which their product is intended?"

This dissertation focuses on this specific question, aiming to offer actionable insights into factors that support or impede CFPTs in their efforts to collectively grok and thus achieve a deep understanding of the context of their products and, thus, the requirements of those products.

1.3 Gap between Requirements Engineering Research and Industry Practice

This research is positioned in more detail in the next chapter however, here I refer to controversy in the Requirements Engineering (RE) field and the inherent difficulty in investigating the industry experience with RE which in 2012 led to the formation of the **Na**ming the **P**ain **i**n **R**equirements **E**ngineering (NaPiRE) initiative (NaPiRE, 2020), a large-scale community endeavour run by researchers world-wide which periodically surveys current practices and problems of RE within industry. After several open global surveys, NaPiRE cites the most frequently stated reason for project failure being *incomplete/hidden requirements* along with

underspecified requirements (Fernandez et al., 2017). It is important to note that many respondents to the NaPiRE surveys that cite these reasons as the main causes of failure claim also to employ systematic RE process models and/or artefact templates.

I believe this to be a reflection of the point raised in Section 1.2.2 that the *engineering* practices of RE are insufficient to ensure requirements *understanding.*

NaPiRE goes on to note that "there is still a lack of theories in requirements engineering" (Wagner et al., 2019, p. 5). Thus, it appears that both industry and research are looking to each other to further the thinking in RE and, until now, collectively failing to make the progress needed.

Neither industry nor research has expressly embraced a critical aspect of requirements understanding, namely, that product development teams are often creating products intended for a world that they are not part of, a world with tacit elements that cannot be fully articulated. This is a critical consideration and yet one that better engineering methods alone cannot solve. Nor are there existing theories sufficient for guidance. Therefore, there is benefit to be gained from an inquiry into this topic that generates interpretive insights from field data and analysis.

1.4 Research Objective and Guide to this Dissertation

At the beginning of this chapter, the question was asked, "Of the software product development teams that you know, how many of them actually grok the world for which their products are intended?", a question aimed primarily at software product development leaders. A more

actionable question might be, "Why do they not grok more than they do*?"* and, more specifically, "Why do they not grok *collectively* more than they do?".

During my extensive leadership experience in the software industry, I observed that cross-functional product teams (CFPTs) achieve significantly different degrees of success in collectively grokking the product domain, even when they are sharing the same organisational, process, and leadership environments. Software product development leaders have no theories that help explain this. Without explanatory models, industry leaders are unable to proactively nurture conditions and relevant factors to support teams to be as successful in this regard as they can be.

My research objective, therefore, was to gain insights into factors that affect this, insights in the form of abstractions that are both actionable by industry practitioners as well as the basis for further research into important related topics.

The interpretive theory that emerged from this research is aimed at helping industry practitioners explain why certain prevailing techniques and empirical approaches for understanding market domains are often inadequate and why some succeed while others do not. It is intended also to offer interpretive insights into factors affecting how creativity and innovation is permitted, or is constrained, within software product organisations and to offer guidance for more effective approaches.

More specifically, this research examined the organisational and product planning process models that the software product development teams operate within and the results provide insights into what organisations do, often unintentionally, to enable or inhibit their software product development teams' ability to achieve a deep, collective understanding of the context of product requirements.

This research concluded that both these models have a strong influence and this dissertation discusses the impact those models have on the dynamics and success of these teams and their software products. It also highlights behaviours and dynamics that tend to occur within CFPTs in response to the impacts of these organisational and product planning process models.

In addition to assisting practitioners in industry, this interpretive theory illuminates many interesting and important areas of further research. These are discussed in Section 6.4 - Future Work.

Constructivist Grounded Theory (Charmaz, 2014) was used as the research method. This method was selected because, as an interpretive qualitative method, it allows for the development of a theoretical account while grounding that account in empirical data. This flavour of Grounded Theory is suited for exploratory research, for *what is happening here?* type of epistemological research questions that the research question is, as opposed to trying to test or validate existing theory. There are particular benefits to this approach in the field of software product development. For example, the method's inherent capacity to interpret complex and multi-faceted phenomena (Charmaz, 2014) which is a particular characteristic of the software

product development activity. Important also is the method's particular accommodation of social issues and its appropriateness for socially constructed experiences which this research is. Finally, it has an imperative for emergence which fits well with the nature of this research question.

There were two other important reasons for the choice of the Constructivist version of Grounded Theory (GT), particularly over other GT methods. One, this version is well-suited for developing a more nuanced and deeper understanding of a phenomenon, consistent with the motivation for this research. Second, it recognises that the researcher plays an important role and works with the participants to collect and interpret data from participant experiences. The method does not place constraints on the researcher having a priori knowledge which was particularly important considering my extensive industry background. This research method, and my use of it, is described in more detail in Chapter 3.

The interpretive practice of engaging with the data collected from the participants' experiences allowed me to construct an abstract understanding of important phenomena in software product development. I use these abstractions as informative guides to industry practitioners aimed at guiding their decisions and expectations regarding software product development organisational models, development methodologies, and team management.

This dissertation is presented in 6 chapters: the first is this chapter, which introduces the research, provides background to the research problem, and explains the motivation for conducting this research. The second chapter, Initial Literature Review – Situating the Study, is

the preliminary literature review that frames and establishes the general context for this study. Chapter 3 explains the research methodology, the rationale for the choice, and the use, of Constructivist Grounded Theory, and details on how the study was conducted and techniques for data acquisition and analysis. Chapter 4 – Findings presents the results and analysis, presenting the building blocks for the theory of Blurring Boundaries in the context of software product development. Chapter 5 – Discussion discusses the theory, relating Blurring Boundaries to extant literature to position the emergent theory. Also discussed in Chapter 5 are implications for industry practice, offering perspectives and policy recommendations to industry practitioners. Finally, Chapter 6 – Conclusion draws conclusions to the study, highlighting the research contribution, and identifying areas for future research.

The next chapter positions the research study in the context of the extant literature from three disciplines.

Chapter 2: Initial Literature Review – Situating the Study

The purpose of this chapter is to situate the research question within the context of three disciplines relevant to this study: requirements engineering (RE), organisational sensemaking, and design thinking/science. As I did not test a theory in this study, the chapter intends to demonstrate that the important themes of the research topic are not currently covered by extant models in any of these disciplines and, therefore, an opportunity exists for my emergent interpretive theory to add to and stimulate the knowledge in these disciplines.

This review takes the form of a critical synthesis of empirical literature in these three disciplines as it relates to important themes in this study. My objective is to show that, in addition to this research topic addressing problems observed in industry, these disciplines alone are inadequate for addressing this topic, possibly pointing to a reality-research gap. The observations that emerged from this study and the analysis that ensued are further related to extant literature in Chapter 5 – Discussion.

The remainder of this chapter is as follows: Section 2.1 discusses the role of the literature review in Grounded Theory methodology and, specifically, how it is viewed in the Constructivist Grounded Theory method that I followed in this study. Sections 2.2 - 2.4 discuss relevant aspects of the research question as they relate to the literature in each of the three select disciplines considered relevant to the research question. The chapter ends with Section 2.5 which offers a summary of the chapter.

2.1 Introduction – Literature review in Constructivist Grounded Theory

Unlike many research methods where existing literature is used as a theoretical background and, therefore, a comprehensive literature review precedes data collection and analysis in order to assist the researcher in conceptualising the research within extant knowledge, Grounded Theory uses existing literature as data to be applied in the analytic activities of the research. In contrast to Constructivist Grounded Theory, the *classical* (Glaser, 1967) and *evolved* (Strauss & Corbin, 2008) methods of Grounded Theory share a core notion that it is important to avoid researchers' *contamination* of the research with a priori knowledge. Classical grounded theory advises against doing a literature review altogether on the substantive area under study prior to the research and then reviewed only in the later stages of the study (Glaser, 1992, p. 31) to prevent the researcher from developing preconceived ideas. Thus, the classical form of grounded theory views a literature review prior to analysis as a constraining influence rather than a guiding one. The *evolved* method as described by Strauss and Corbin shares the same caution but encourages the researchers to use the literature data in all phases of the research provided they maintain an attitude of scepticism throughout (Strauss & Corbin, 2008).

Constructivist Grounded Theory, on the other hand, the method used in this study, takes a different stance on this matter. While none of the Grounded Theory methods would permit researchers' *contamination* of the research product, Constructivist Grounded Theory adds that avoiding the researcher's influential role in the research process is an unattainable task. It also views it as undesirable. Taking a constructivist epistemological stance, this method views that the researcher contributes to, and cannot be purged from, the data collection and analysis as both

are "created from shared experiences and relationships with participants and other sources of data" (Charmaz, 2014, p. 239). In Constructivist Grounded Theory, the resulting theory "*depends* on the researcher's view; it does not and cannot stand outside of it" (Charmaz, 2014, p. 239, emphasis in original). Charmaz maintains that the method's groundedness "results from these researchers' commitment to analyze what they actually observe in the field or in their data" (Charmaz, 1990, p. 1162). Thus, Constructivist Grounded Theory prescribes no location for the literature review, leaving it up to the decision-making process of the researcher. If done early, the method says that it should be revisited to critique and confirm alignment with the research findings.

This is the approach I took. I conducted an initial literature review at the beginning of this study to broadly situate this work with respect to the body of extant literature of disciplines that I felt had some relevance to the research questions. I also used this review to situate myself in relation to the current discourse beyond that which my then-current professional and academic experience had exposed me. Doing this sensitised me to certain concepts and enhanced my "ways of seeing, organizing, and understanding experience that are embedded in our disciplinary lenses" (Charmaz, 2000, p. 515), providing me with a starting point to incorporate certain concepts into the research. This initial literature review was conducted also for a very pragmatic reason, namely to establish sufficient comfort that there had been no prior treatment of this research topic that had heretofore remained obscure. It is this initial literature review that is discussed in the remainder of this chapter.

2.2 Isn't this simply Requirements Engineering?

At the outset, it was natural to imagine this research being firmly situated within the Requirements Engineering (RE) subdiscipline. In other words, the research question seems at the heart of getting the requirements right. It was, and is, reasonable to further relate this study to the specific requirements engineering methods of requirements elicitation and requirements validation. However, as I shall show, this study was more about the general topic of requirements *understanding* than it is about any specific subset of requirements engineering activities, at least as far as requirements engineering is commonly practiced today. Reviewing most of the papers at the IEEE International Requirements Engineering Conference over the past decade, as well as other published papers relating to requirements engineering in many other sources, I found growing sentiments expressed about the challenges and shortcomings of prevailing approaches to requirements engineering (e.g., Schon et al., 2017). As they have tended to historically, predominant approaches focus more on techniques and methods intended to specify and validate detailed requirements than they do on deepening practitioner understanding (individually or collectively) of the requirements and, more importantly, the context of those requirements. Adding to this challenge is the need to understand *market* requirements. The emphasis on techniques and methods is unsurprising – after all, it is called requirements *engineering*.

With these shortcomings and challenges continuing to loom large, there is now strong controversy in the requirements engineering discipline. Some agile development thought-leaders such as Cohn are blunt about it: "The idea of eliciting and capturing requirements is wrong."

(2004). While many researchers hold to prevailing views, believing that we just need better techniques to improve effectiveness, others are echoing the theme of Cohn's critique. E.g., research by Mohanani et al. concluded: "simply using the terms requirements and shall can deleteriously affect designers' creativity" (2014, emphasis in original), introducing the concepts of *requirements fixation*, a disproportionate focus on expressed needs or desires that are labelled as *requirements* with this fixation interfering with the ability to recognise, let alone understand, and much less, even imagine, unexpressed requirements. Viewing certain core requirements engineering activities as being problematic, Ralph and Mohanani noted that: "the specifying, structuring aspects of RE are counterproductive" (2015). And, taking Cohn's position further, Ralph (2013) notes that it is not uncommon for statements presented as requirements to be instead goals, desires, etc. which lack the certainty and unambiguousness that a *requirement* connotes. Citing both ontological and epistemological difficulties with this, Ralph raises questions about how to proceed when there are *"goals, features, conjectures and design decisions mislabeled as requirements"* (p. 4). Given the compulsory denotation of what requirements mean and that requirements engineering methods assume, such *requirements* then are, as Ralph (2013) asserts, *illusory.*

Although the concern is growing, it is not new. While getting the requirements right may be the single most important and difficult part of software product development, more than 20 years ago, Guinan et al. studied team performance during requirements definition and concluded that:

> *team skill, managerial involvement, and little variance in team experience enable more effective team processes than do software development tools and methods.* (1998),

foreshadowing the *"Individuals and interactions over processes and tools"* value that would appear in the Agile Manifesto (2001) three years hence.

This controversy is evidence that some software product development practices still operate in a linear, prescriptive process paradigm (e.g. Waterfall) and are experiencing what Kuhn (2012) described as the incommensurability across paradigms – that is, some methods from a process-driven paradigm are not necessarily appropriate outside of that paradigm due to differences in conceptual frameworks. And while there are certain domains where the more prescriptive *techniques and methods* approach is entirely adequate and appropriate, my focus with this research was on product and problem domains that do not lend themselves well to complete, unambiguous, and perfectly understandable specifications, where the true needs are often unexpressed or even inexpressible and, therefore, where it is necessary for cross-functional product teams (CFPTs) to have their own deep understanding of the product domain beyond that which is (and can be) articulated in any form of a requirements specifications. As Kent Beck (2007) stated in his critique of DWTTY (Do What They Tell You), "there's more to delivering good software than doing what the customer tell you".

As introduced in Chapter 1, this crisis in the requirements engineering (RE) field and the inherent difficulty in investigating the requirements engineering space led to the formation in 2012 of the **Na**ming the **P**ain **i**n **R**equirements **E**ngineering (NaPiRE) initiative (NaPiRE, 2020). This initiative is partly motivated by the view that requirements engineering research is not sufficiently driven by problems emerging from industry nor is it even sufficiently informed about the state of industrial practice in requirements engineering. After several open global surveys,

NaPiRE cites the most frequently stated reason for project failure being *incomplete/hidden requirements* (Fernandez et al., 2017). This is despite those same survey respondents declaring that they were using clear requirements engineering process models and artefact templates. NaPiRE goes on to note that

> *"since requirements engineering … is highly human-based, we face the challenge to create a solid empirical basis that allows for generalisations taking into account the human factors that influence the … discipline"* (Wagner et al., 2019, p. 3),

leading to the observation that *"there is still a lack of theories in requirements engineering"* (Wagner et al., 2019, p. 5).

My research is situated alongside this condition of a) a lack of requirements engineering theories; b) incomplete/hidden requirements being the main cause of project failure; and c) requirements engineering being highly driven by human factors and with the requirements engineering discipline having yet to create generalizations that take these human factors into account. The latter two conditions are specifically and directly related to the research question since there is an increasing dependence on the *development team's* broad and deep understanding of requirements as well as the context of those requirements instead of relying on a separate individual or another team to have that understanding and somehow communicate that understanding effectively to the development team. There is also an increasing dependence on the degree of *collective* understanding of the team due to the growing complexity of the software product being built. With its focus on techniques and methods, prevailing requirements engineering practices fail to address these factors, factors that are increasingly identified as critical for software product

development success. The results of this research could help expand the scope of Requirements Engineering.

2.3 Is this Collective Sensemaking in Organisations?

As this research relates to teams coming to collectively understanding something, it is logical to ask whether this research is about collective sensemaking in organisational settings.

Reviewing literature in the organisational sensemaking field, I saw the focus of many researchers being mainly on the social process of individual identity in successive spheres of membership through interactions with others. There is an extended research area of *collective* (or team) sensemaking (the process by which people give meaning to their *collective* experiences) which takes on both internal and external perspectives (Jones, 2015). The internal perspective is focussed on the representation of collective meaning (Weick et al., 2005). However, a sense of meaning does not necessarily embrace the deep *understanding* aspect of an external domain held by the team as a collective that I investigated. Regarding the external perspective, some researchers, notably Daniel Russell (2009) while looking at the Human-Computer Interaction (HCI) realm, considers collective sensemaking with an external perspective for a broader purpose - to collect and organise information in order to gain insight, to analyse, to transfer. However, although his view establishes sensemaking in a collective location (an information world), he describes a style of engagement of sensemaking that is essentially personal, rather than the collective team-based framing that this research focussed on. My research findings may

contribute to sensemaking in the aspect of external collective sensemaking which appears to be generally underexplored.

Overarching both the above-noted perspectives is the declaration by Weick et al., "sensemaking is retrospective" (2005, p. 411). This strong tone of collective sensemaking being retrospective in nature, making sense of what has occurred in the past, is quite different than the core element of this research which is how to deeply understand the *present*, and trajectory, of a realm that one is not part of, and for that deep understanding of the present providing a basis from which to speculate on the future of that realm. Further research building upon my findings may contribute to collective sensemaking in organisations by expanding from the retrospective view to include on-going activities, current conditions, and future trajectories.

I took note also of the Cynefin framework (Kurtz and Snowden, 2003) which is a sensemaking framework designed to allow for shared understandings to emerge. At first, this framework appeared to offer valuable perspective into how teams ingest, socialise, and collectively store insights. However, as with other collective sensemaking models, it has resonance in early problem-solving stages and for formal and finite periods of time whereas my interest with this research was on the full product lifecycle.

Other researchers (e.g. Klein et al., 2006; Naumer et al., 2008; Kolko, 2010) elaborate further by introducing data-framing into the picture and defining design synthesis as a process of sensemaking, 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 future research building upon this study.

2.4 What about Design Thinking / Design Science?

The design thinking space, being fundamentally about the iterative processes of human-centred idea generation, has a large amount of recent literature regarding empathy-driven design (translating human needs to good experiences) with an emphasis on understanding, e.g. (Koppen & Meinel, 2012; van Rijn et al., 2011; Postma et al., 2012; Woodcock et al., 2018; Dong et al., 2018; Kourprie and Visser, 2009; Kolko, 2014). Woolery (2011) notes that at the core of design thinking is a focus on empathy, or "using a beginner's mindset and immersing yourself in the user's experience to uncover deep needs and insights" (p. 6). I also note the work of David Holston (2011), who places considerable emphasis on designers designing in *context.* Achieving a context through empathic efforts (*grokking*) is a topic that echoes throughout this research.

However, despite recognising the importance of empathy and context, I found that design thinking works fall short of addressing this research question in several critical respects: 1) design thinking tends to position the design activity as an early, albeit iterative, part of an essentially sequential product development process rather than part of an on-going continuous product development effort; 2) it focuses more on the design individual or the design sub-team rather than the entire product development team; 3) when the design sub-team is considered, there is little to no discussion of the team as a *unit* developing a *collective* understanding when referring to empathic capability; and 4) most importantly, design thinking places a heavy

emphasis on the *user* experience (UX) whereas the grokking of the product domain focussed on in this research is a much broader notion of empathic understanding which encompasses the entire context of the product's market and including all stakeholders, competitors, etc., not limited to the users and the user experience (UX). In fact, in many cases, there may not be a user involved with the product at all, yet there is still a complex context for the product. One form of the common *incomplete/hidden requirements* cause of project/product failure referred to in Section 2.2 is overlooking or incorrectly identifying stakeholders. Design thinking focusses on one of, potentially, many types of stakeholders.

In short, the notion of a deep empathic understanding held by a team as a collective and the need for that deep understanding to be that of the broader product and market domain, rather than just an individual product user, appears to be absent in current design thinking. The findings of this research could contribute to the evolution of design thinking.

I reviewed design science literature and found resonance in the design science models described by Wieringa (2014) which acknowledge the challenge that empathy-driven requirements understanding attempts to address. He notes, "stakeholders rarely if ever are able to specify requirements" *and* "requirements are the results of design choices we make, jointly with, or on behalf of, the stakeholders" (p. 52). Wieringa defines a *social context,* as the context for stakeholders, stakeholders being those who affect, or are affected by, a design project. Note that his use of the term *stakeholders* is broader than the design thinking view of *user* and would appear to be more closely aligned with the scope of this research which aims to identify, and fully understand, *all* stakeholders. Further, the *social context* that Wieringa describes is very close to what I refer to as the domain context. Wieringa goes on to promote the use of a

contribution argument (p. 52), the purpose of which is to justify the choices made in creating the artifacts. He notes that the *contribution argument* is a prediction and that it is fallible (p. 53). The reason it is a fallible prediction is that the design researcher uses a *context assumption* when expressing the requirements. This *context assumption* is a guess on the part of the researcher of the *social context.* However, I found no treatment in the design science literature that referred to achieving a more comprehensive, deeper, and on-going understanding of Wieringa's *social context* nor to the necessity of this understanding to be held collectively by the entire development and implementation teams.

As with design thinking, design science has a distinct project orientation, focusing on artifacts to satisfy need. Using methods to justify decisions made and with limited focus on truly understanding, the *social context* suffers from a similar limitation as it relates to this research focus as does mainstream requirements engineering's focus on techniques and methods and, therefore, was an unsuitable conceptual model upon which to base this study.

2.5 Summary

While I was informed by my initial review of current literature in the areas of Requirements Engineering, Collective Sensemaking, and Design Thinking / Design Science, it was clear that this research does not fit entirely within any of these disciplines, as they are currently defined. This research most naturally contributes to Requirements Engineering (RE) by adding to the considerations of human factors in RE methods. It may expand also the view of Collective Sensemaking to include the notion of an on-going understanding of a realm external to the

collective. It may contribute to Design Thinking by introducing the notion of a broader focus than just the user and UX and by embracing the notion that the deep understanding can, and should, be held by the collective development team, not just the design individual. Finally, this research may contribute to design science by offering insights into how to achieve a broader, deeper, and ongoing understanding of Wieringa's *social context*.

As my theory emerged, I was then able to refer back to this material as a guide to more precisely compare the emerging theory with extant literature to refine alignment (and non-alignment) and to situate it within the known theoretical landscape.

Chapter 3: Methodology

The study sought to gain insights into the following research question:

> *What factors support or impede cross-functional software product teams in collectively achieving a deep understanding of the environment for which their product is intended?*

This research question refers to factors affecting a collective grokking of the context of the software product requirements.

The purpose of this chapter is to describe the methodology used in this research. The choice of research methodology was based on the nature of the research question as well as the theoretical perspectives that I hold and that I believe are relevant to this inquiry. These are described in detail in Section 3.2.

This chapter is divided into four sections. In Section 3.1, my philosophical stance and the research design is presented. Section 3.2 describes the research method and includes sub-sections for research context, researcher positionality, participants (including ethical considerations), data collection, and analysis techniques. In Section 3.3, I discuss the topics of research quality and study limitations. Section 3.4 outlines the tools used in the study while Section 3.5 provides a summary of the chapter.

3.1 Philosophical Stance and Selection of Method

This research question is a *what* question, a search for a better understanding of something that is. It is not attempting to prove or disprove a hypothesis nor is it trying to optimise anything. Rather, this is about gaining insights into people and processes, specifically software product development teams, and gaining insights into how they come to collectively understand (or not) the product domain. More specifically, it is about what factors can interfere with or support their ability to do so.

I chose Grounded Theory as the general qualitative research method. Grounded Theory is particularly appropriate in a research situation such as this for several reasons:

- where the theory sought intends to move beyond a superficial description towards an understanding that is contextually situated, that would be useful to practitioners, and could become a framework for further research.
- it is a useful method when studying relatively new areas or when trying to gain a fresh perspective on a well-known area where there has been limited research.
- it allows for the study of social interactions and behaviour. This is particularly important for this research question because, as noted in the previous chapter, there has been limited progress made in understanding the human and social aspects of Requirements Engineering.
- this study is an emergent inquiry, aimed at developing theory upward from the field data (*grounded*), keeping the theory and data close.

This choice of a qualitative research method in an engineering discipline is not as unusual as it once might have been. There has been a steadily increasing recognition of the value and the use of the grounded theory methodology in software engineering research in recent years (e.g. Adolph et al, 2011; Sedano et al, 2019; Hoda and Noble, 2017; Stray et al, 2016; Stol et al, 2016).

My philosophical stance for this research falls into the Interpretive/Constructivist paradigm. This is relevant because the research question aims at better understanding (as opposed to explaining) what is going on with respect to observable phenomena regarding human factors within and around software product development teams. A core belief of the Interpretive perspective is that reality is complex, ever-changing, and socially constructed through interactions. Rather than believing that knowledge can be objectively determined, Interpretivism emphasises that one obtains knowledge through experience and interpretation and that this knowledge obtained is closely tied to the context from which it emerged. My research sought to conceptualise field phenomena in a specifically defined context through engagement with the participants and then to build theory based on the interpretation of their shared experiences that would be useful in their contexts and that could then also be used to inform other contexts.

From my more than 40 years of experience in the software development industry and in the absence of any general theory for software product development, I believe that social construction is an accurate portrayal of how reality is determined and how knowledge is obtained by software development teams, especially those developing products. If I ever had any doubts about this, the fieldwork of this inquiry solidified that belief.

Recognising that there is no one software development process model, no one software product development organisational model, and no seemingly unambiguous set of experiences that software product teams possess, taking an Interpretivist stance was helpful because it embraces the notion of a complex reality and thus views the world through many sets of individual eyes, specifically the eyes of the participants who each have their own worldviews. Interpretivism's inclusiveness accepts multiple viewpoints from different individuals and from different groups to create a richer interpretation of reality. It is this richer, more comprehensive, interpretation that is valuable for the researcher in creating a better and deeper understanding.

Taking my philosophical Interpretivist paradigm stance one step further, I accept the Constructivist epistemological view that individuals not only have their own set of experiences that contribute to reality but that that reality itself is inevitably of our individual construction (Charmaz, 2014). This epistemological stance leads to the belief that by interacting with the participants, the researcher plays a part in the knowledge construction.

This led me to choose, of the established Grounded Theory methods, the Constructivist version of Grounded Theory, as defined by Kathy Charmaz (2014), as my specific research method. This method explicitly aligns with my constructivist epistemological stance. It embraces the belief that knowledge is based upon one's perspective and, therefore, rather than the research product reflecting the reality of the participants' reality, it is influenced by the world view of the researchers and the situations that the researchers rely upon. The method recognises that the researcher has disciplinary and theoretical proclivities, creates relationships with the participants,

and interactionally constructs and renders the data. In other words, the research will always be a construction built from the dialectic interactions between the participants and the researchers. Theory arising from this method, therefore, *depends* on the researchers' view. It is impossible for the research to stand outside it. The theory is constructed *by* the researchers as *a* reality, not *the* reality, using the researchers' background to enrich the co-creation of knowledge and understanding with the participants.

In addition to my philosophical alignment with Constructivist Grounded Theory, the method's view of the researchers' role was a particularly important consideration in this choice because it embraced the extensive subject-matter expertise and industrial experience that I brought into my role as the researcher rather than viewing it as a potential liability to be somehow carefully contained.

3.2 Research Method

Constructing Grounded Theory by Kathy Charmaz (2014) formed the general guidance for my research method, outlining the tenets and techniques I employed. An important characteristic of the Constructivist Grounded Theory method is that, in contrast to a complete data collection phase followed by an analysis phase, an iterative data collection–analysis–theoretical sampling cycle (Charmaz, 2014) is employed which continues until the research reaches a point of *theoretical saturation* (Charmaz, 2014), a point at which new data collected fails to add additional insights to the existing analysis. Whilst the data collected is guided by the general research question or phenomenon of interest, the sampling and data collection throughout is informed by the analysis, analysis that is immediately and continually being performed on the

total data collected at that moment in time. The overall movement of this Grounded Theory analysis approach may be imagined as a *zig-zag* (see Figure 3-1 below), collect data – refine categories – repeat until no new insights into the categories are emerging from the analysis.

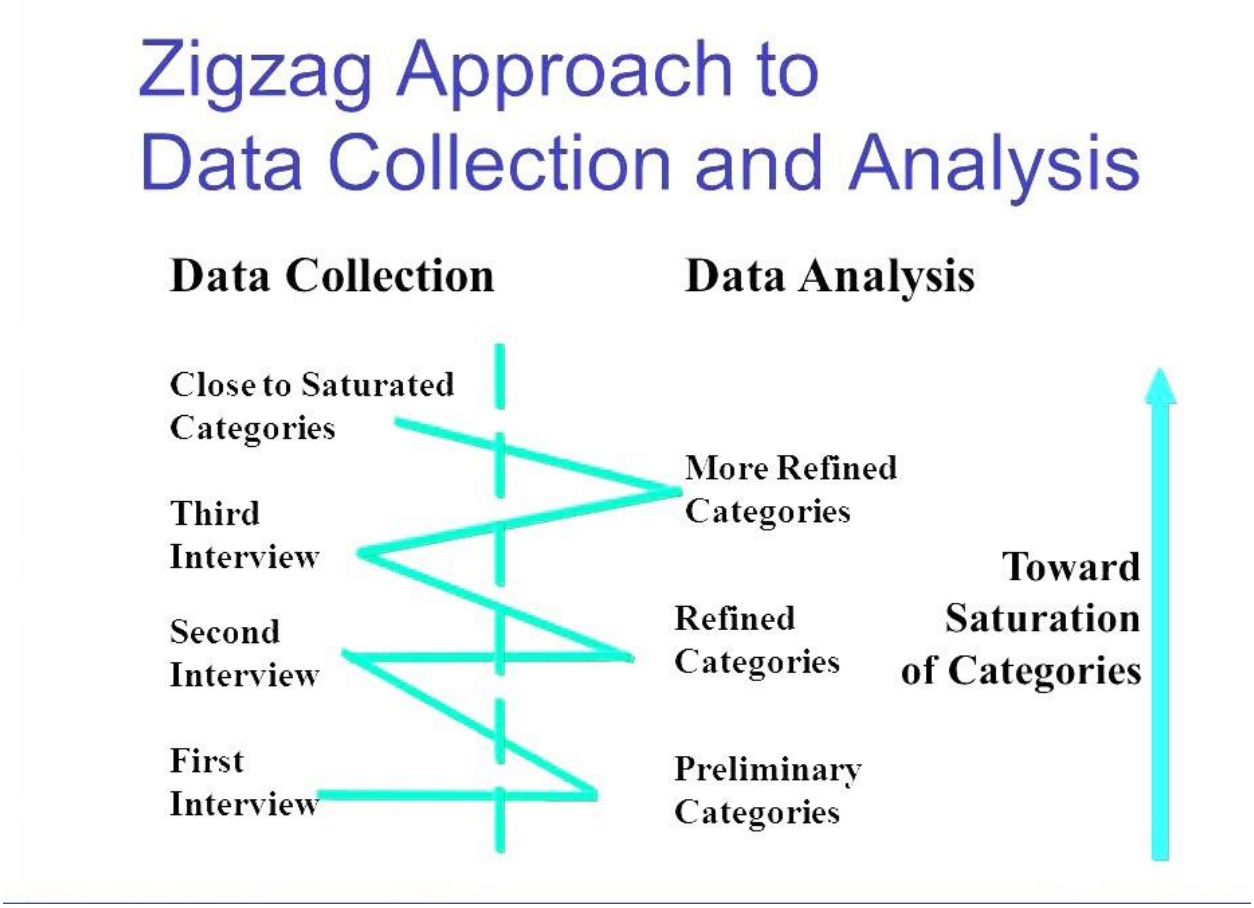


Figure 3-1. Zig-zag Approach to Data Collection and Analysis

Chapter 2 described an initial literature review that was performed prior to commencing the fieldwork to establish sufficient comfort that the topic had not been previously addressed and to shed further light on both controversial and unaddressed topics in the extant literature. It served also to position the inquiry sufficiently for my PhD Comprehensive Examination. In discussing the theoretical concepts that emerged from this study, I engaged further with the literature in Chapter 5 - Discussion.

The remainder of this Section 3.2 - Research Method is as follows: Section 3.2.1 describes the context of the research. Section 3.2.2 discusses the researchers' positionality and reflexivity in this study. Section 3.2.3 describes the participants in the study followed by Section 3.2.4 which discusses how I collected data from those participants. Finally, Section 3.2.5 describes the analysis techniques used and discusses research quality considerations.

3.2.1 Research Context

The research context for this study was companies that create software *products* (as opposed to bespoke software solutions) and, of those, companies that use *some* form of cross-functional teams to develop those products. The primary reason for this context was that this was the industrial context in which I most frequently observed the phenomenon researched while I was in my professional work-life. From personal experience, I believe that many software product companies would find significant value from the research insights (see Section 1.1 – Motivation). The potential for transferability of the research insights outside of this described context is a distinct exercise post-results and is discussed further in Section 6.4 - Future Work.

The companies that participated in this research create products for a broad range of market domains – point-of-sale solutions for restaurants as well as specialised vertical markets, social media marketing, stem cell therapy, enterprise skills management, enterprise payments solutions, workstation and server virtualisation, and advanced aerospace.

Although the research question did not seek cultural nuances to the phenomena studied, it may be important to point out that this was not a monocultural inquiry, reflective of the general character of the software product industry today. The participant companies had software development operations across Canada and the US as well as in Italy, India, and the Philippines. To add to this multicultural landscape, my own personal industrial experience has included software development operations in China, Japan, UK, New Zealand, Australia, Czech Republic, and the Netherlands.

As is the case with many qualitative research methods, I carried out the data collection in a naturalistic setting. Where possible, I conducted individual interviews in participant companies' meeting rooms or in open, semi-quiet common areas. In rare circumstances, interviews were held off-site due to a lack of a suitable meeting location on-site at the scheduled interview time. Group observation sessions were always in the participant companies' group meeting locations.

3.2.2 Researcher's Positionality and Reflexivity

SAGE defines *positionality* as

> *the stance or positioning of the researcher in relation to the … context of the study—the community, the organization or the participant group. The position adopted by a researcher affects every phase of the research process, from the way the question or problem is initially constructed, designed and conducted to how others are invited to participate, the ways in which knowledge is constructed and acted on and, finally, the ways in which outcomes are disseminated and published.*
>
> *(Coghlan & Brydon-Miller, 2014)*

In short, *positionality* defines where it is that the researcher stands in relation to the research participants and their environment. I use the term *reflexivity* to refer to the practice of being mindful of that positionality and thinking about what one has done while one is still engaged in the activity. In other words, being very deliberate and self-aware throughout. This self-awareness is important because the Constructivist Grounded Theory method acknowledges that the researchers' positionality shapes the research process. Specifically, the researchers' positionality shapes what is viewed as important data to collect, how participants respond to the researcher, and how researchers interpret the data collected.

I have worked in the software industry for 42 years, much of that in senior corporate leadership and product development roles, in companies that created products for a wide range of markets. I hold a B.Sc. degree in Honours Computer Science, an M.Sc. degree in Software Technology

and, an MBA in Leadership and Organisation. No participants had any form of relationship with myself that might have imparted influence on the research.

As noted in Section 1.1 – Motivation, this research question arises out of my own deep background in industry where I personally experienced a lack of insights into this topic and strongly felt that had we had deeper understanding on this topic, our effectiveness as senior software product development leaders would be enhanced. Thus, my professional background is the primary factor in the framing and conduct of this research.

Also, as mentioned when describing my Philosophical Stance and Selection of Method in Section 3.1, it is impossible to distance myself from the participants, the research analysis, and the insights that result. Thus, I carried my experience, and the biases inherent, into my interactions with the participants and into my analysis of the data collected.

I expected that my age and deep professional background might have a slight potential to inhibit some participants from engaging with me in frank and open-ended discussions about their experiences and thoughts. However, I found no occasion where participants questioned my positionality nor seemed inhibited by it in any way. What occurred was quite the opposite. My positionality was either not thought about at all by the participants or, in some cases, clearly appeared to be seen as comfort or an opportunity. For the latter, it seemed that I was viewed as someone with enough expertise to *get it*, sufficient senior leadership experience to be able to react constructively during the discussion from that perspective, and independent enough for them to freely express themselves without fear of consequence. There were cases where it

appeared to be a liberating experience – "finally someone I can safely say this to and who will understand!".

This is not to suggest that I ever 'dismissed' my positionality or took it for granted. Quite the contrary. In addition to continually being sensitive to my positionality in all my interactions with the participants, I continually reflected on whether I was seeing what I wanted and/or expected to see or, conversely, whether I might be taking too little account of what looked familiar. While the former (seeing what I wanted to see) is an oft-posed challenge by those unfamiliar with Constructivist Grounded Theory, it was not a major integrity risk in this research study since I was not looking for something specific and, therefore, I didn't know what I wanted to see. Rather, I entered the study without any hypothesis, nor even assumptions since, in dialogues with my professional colleagues on this topic, no speculations surfaced as to possible factors. Thus, I was searching for anything that might offer understanding and be useful to the industry practitioner. The other reflective risk possibility, underplaying what looked familiar, was the subject of continual and ongoing reflexivity on my part. Without this continual reflexivity, I might well have fallen prey to having a lack of interest in what appeared very familiar. Hanging on to a sense of curiosity was the key to avoiding this risk. I eventually came to be comfortable with, and at times excited by, the possibility of understanding something that had been heretofore *hidden in plain sight* or that had earlier been deemed unimportant and, therefore, gone unexamined in the heat of my professional work life. Now, as a researcher rather than a practitioner, I was in a position where I could observe but not be able, nor expected, to *fix* anything. For the practitioner, it is often difficult to be curious when emotionally engaged in an issue or feeling time-pressured to move forward. Having had the experience of being in that

position for so many years and now being close to the issues without having a personal connection to them, I came to better appreciate how prevalent this is in industry and therefore why having curiosity while being able to act upon it is a rare luxury. Admittedly though, in the beginning, it was challenging to maintain a check on so many years of conditioning as a *problem-solver*. Fortunately, reflexivity continually helped temper those instincts throughout the study.

According to Kathy Charmaz, the Constructivist Grounded Theory method rests on the belief that "Knowledge rests on social constructions … influenced by the researcher's perspectives" (Morse, 2009, p. 130), viewing the very notion of a neutral observer as inherently invalid. Rather than a liability to suppress, the method recognises that positionality such as mine can be an asset, while fully recognising the implications. I conducted interviews with questions of my own creation, yet the sessions were semi-structured, allowing the participants to contribute more freely, with me gently guiding the conversations as necessary toward the subject matter at hand. The analyses conducted throughout the study and the ensuing findings are purely my subjective interpretations, they could not have been otherwise. It is this deep industrial experience that enhanced my *theoretical sensitivity*, that is, my ability to generate concepts from data and relate those concepts during data collection and analysis. Theoretical sensitivity is at the heart of constructing grounded theory with creativity playing an important role (Straus & Corbin, 2008; Glaser, 1992). This theoretical sensitivity avoided the temptation to prematurely revert to the literature to look for answers. It was, as Eisner suggests, the expert ability to "see what counts" (1998, p.34) -- the sensitivity to tacit elements of the data, meanings, and connotations that

guided this research, supported fully by the data collected, towards questions that I knew, from experience, mattered.

3.2.3 Participants

The research protocol used received approval from the UBC BREB (Behavioural Research Ethics Board) throughout the study (see Appendix A). It was only after receiving this approval that I began recruiting company and individual team member participants.

3.2.3.1 Company Sampling

The sampling criteria selected participant companies firmly within the Research Context as described in Section 3.2.1. The two main criteria of the research context were that the company developed software as products (as opposed to bespoke solutions only) and, second, that it claimed to have at least some degree of cross-functionality in the teams developing the software products. If the companies didn't have some degree of cross-functionality within their teams, their teams would not be good candidates to observe the phenomena that motivated this research, hence the explicit reference to cross-functional teams in the research question. Provided there was some degree of cross-functionality in the development teams, regardless of the reasons for them being so and regardless of how diverse the functional mix was on the teams, there was at least the potential for teams to exhibit some degree of the collective grokking ability that I wished to study. Within this sampling criteria, companies of various sizes, ages, and market

domains were included to allow for any of these attributes to present themselves as factors addressing the research question.

I purposefully selected one bespoke software development company that did not meet the sampling criteria for being a software *product* company. This was done so for there to be some data to use when examining any assumptions that might be made regarding whether emerging insights were applicable to complex software development environments in general or specific only to software *product* development. This also assisted in a preliminary assessment regarding the transferability of findings beyond the research context.

The second specific sampling criteria for companies was with respect to the domains targeted by their software products. To keep the research more focused, I explicitly sought to examine teams developing products for domains which development team members were unlikely to have had a deep understanding for prior to joining the team. Therefore, excluded from the sampling were certain categories of product domains for which it was reasonable to expect that individuals on a development team might be already familiar with. So, for example, excluded were software technology products such as dev-ops software or software development tools where it was reasonable to expect some development team members would have direct knowledge of, or experience with, similar products and would also likely be the targeted consumers of such products. Other broad product categories excluded were mass-market consumer products, e.g. games (software product development team members could well be current or former consumers of gaming products) or smartphone-based personal apps intended for a broad market. In other words, the company sampling was limited to companies that targeted domains sufficiently

distant from what product development team members might deeply understand, whether from their professional or personal lives.

I approached select senior leaders (CEOs, CTOs, CPOs, etc.) in the targeted companies directly through my professional network, inviting their companies to participate. The reception was very positive and supportive, indicating resonance with the topic and a recognition of its importance and relevance. The credibility brought to the study by my deep industrial experience also contributed to their comfort in agreeing to participate. It was common for the participation 'deal' to agree to thoroughly review the findings and other insights with them once the study was complete, while ensuring, of course, that the anonymity promise made to the individual participants was honoured. While some of these senior leaders participated in the study in the role of individual participant and/or peer validation, they all remained keenly interested in seeing, not only the final results but also insights as they emerged.

Due to the dynamic nature of the software industry, there were unfortunately a few cases where the logistics or the timing was such that corporate participation was not practical (e.g. a company being in the throes of a restructuring or a market pivot) so, while the interest was almost universal, not every company I approached felt they were able to participate. When a company did agree to participate, a senior representative signed a research ethics board Organisation Consent Form (see Appendix B) that provided a high-level description of the research and offered assurances from the researchers and the university to the participant company regarding the responsible use and custodianship of all data collected while engaged with their company. This agreement also afforded me a general nod to approach individuals in the company to

participate as well as general permission to be on the company premises throughout the study. Two companies required me to sign non-disclosure agreements in addition to the Organisation Consent Form. None of the conditions of those agreements introduced any constraints that compromised my normal research activities.

A total of seven companies participated in this study. One company was headquartered in Seattle, USA, and the others in Vancouver, Canada. Several had satellite operations in other countries. The table below illustrates the broad spectrum of market domains targeted as well as a broad range of the number of employees in the participant firms. The nature of the development process is shown in this table only as a note since the essence of the research question transcended development process methods.

	markets served	# employees	age of firm (years)	dev. process model
1	social media marketing	2,000	12	agile as a method
2	enterprise payment solutions	100	25	agile as a method
3	vertical market virtualisation solutions	200	17	agile as a method
4	cellular therapy lab & clinic management	25	22	agile as a paradigm
5	retail	500	21	agile as a paradigm
6	enterprise skills management	13	8	agile as a paradigm
7	satellite & ground imagery (bespoke)	3,000	50	prescriptive method

***Table 3-1*. Participating companies**

3.2.3.2 Individual Team Member Participant Sampling

Once the Organisation Consent Form was signed and a senior representative of the company made a general introduction of myself and this research to the organisation, I proceeded to approach individual team members directly asking if they were willing to participate in one or more semi-structured interviews. No incentives were offered nor provided, and their participation was entirely voluntary. In other words, a company having signed an Organisation Consent Form did not obligate any individual employee to participate. If an individual did agree to an interview, they signed a research ethics board Individual Consent Form (see Appendix C) that described the study in high-level terms and clarified my commitment to them regarding anonymity and responsible custodianship of the data collected. Most individuals that were

approached immediately agreed to participate and to quickly scheduled an interview. Some participants asked to slightly defer scheduling an interview due to leave plans, product release deadlines, etc. All that did agree to participate did so with a degree of apparent enthusiasm while taking personal expressiveness into account. I sensed no hesitation from any individual that agreed to participate.

I found a high degree of consistency in team sizing within the participant organisations and, therefore, I did not include team size in the sampling criteria. Early Agile guidance regarding optimal team size (Schwaber, 2004, p. 118) was typically 7 +/- 2, with a preference towards smaller rather than larger in that range. Except for the one bespoke software company in the company sample, all other companies had adopted this spirit of team size, with an average across these companies being between 5 and 6.

These six companies also took a general approach of defining their development teams as being *complete*, meaning the teams were generally self-contained and autonomous, having sufficient collective skills to solve the complete development problems at hand. The specific functional capabilities represented on the team was determined by the nature of the product. For example, not all products in this study had a need for UX design, security, etc. However, the general sufficiency in team capability, along with some consistency in team sizing, led me to choose not to perform *theoretical sampling* of teams.

The initial sampling strategy for individual team members was simply to talk to team members in various roles and on a variety of teams to gain knowledge and insights into the teams'

operations and dynamics as well as the teams' contextual setting. This research method's continually emerging results of the on-going iterative analysis informs the data gathering for the next iteration. This informed iterative steering of data gathering is the essence of *theoretical sampling* as defined by Charmaz (2014). In other words, what data to collect next and from where. Thus, the sampling strategy evolves as insights emerged rather than being determined a priori. One example of this *theoretical sampling* at work was that at the outset of this study, the focus was primarily, *"what's going on in this team?"*. However, based on emerging results, the focus soon shifted to looking at what was going *around* the team within the company instead of doing deep dives on a team-by-team basis. This was the reason for a shift from interviewing most members on a team to studying more teams in a company. Another form of this sampling strategy evolution at play occurred in the selection of participants. There were times where adjustments were made to select more participants in a particular functional role when the analysis indicated that I had a limited perspective from that function. An example of this occurred when the product management and product design functions were located in a different part of the company and I found myself at one point having closer relationships with the software engineers, thus causing me to put more explicit effort into spending time with the product management participants until I felt I had a broader perspective.

Theoretical sampling helped paint a richer and more complete picture of what was going on by: a) selecting participants in particular roles to provide a perspective on topics emerging from previous interviews or observation sessions, b) adjusting the line of inquiry to focus on emerging questions, and c) selecting specific individuals to explore topics and statements made in an earlier group observation session.

The demographic profile of the individual team member participants was unsurprising, given the character of today's software industry. Most participants' ages were in their 20s or 30s and they typically held university degrees in a relevant discipline (engineering, computer science, marketing, visual design, etc.). They often had prior work experience in other software companies and often in other product domains. Their previous employment often included experience working in other parts of the country or elsewhere in the world and, consistent with the current character of the software industry, the participants were racially and culturally varied. The individual participants' understanding of technical and process concepts was mostly excellent and their ability to communicate on these topics was never a challenge during the interviews. Many did have English as a second language which, at times, required some grammatical adjustment or clarification on my part to ensure that my questions and their responses were understood as intended. At times, I would put additional effort to ensure that I understood local vocabulary correctly, i.e. terms in common use within that specific company that may be unique to that company or that may be a term in common use in industry yet having a company-specific definition or nuance.

I formally interviewed a total of 27 individual team members. In addition to these 27 interviews, many informal conversations were held, some on an on-going basis, with them and with senior executives of several of the participant companies. Additionally, there were individuals that acted as *navigational guides* or hosts and provided support for me as I conducted my activities within the firm. I often used these individuals for validation checks, etc. While I did not formally interview all of them, their input often stimulated insights that influenced the analysis.

I formally attended 20 team and team-of-teams meetings as a silent observer. Table 2 below provides a summary of the participation.

# companies	7
# teams	18
# formal observation sessions	20
# individual team member interviews	27
senior managers	2
senior engineers / team leads	8
intermediate s/w engineers	11
quality assurance specialist	1
product managers	5

***Table 3-2*. Participation summary**

The software industry today is very fluid. Individuals with software development skillsets are able to be very mobile due to their having talents that are in high demand. This environment creates career opportunities for many individuals where they can be quite selective about where they work, what products they work on, what markets the company's products serves, etc. Consequently, it was not uncommon for me to attempt to seek out an individual to invite to participate or to have a follow-up conversation, only to discover that that person was no longer working with the company. In addition, firms are often in a state of change (merging, acquiring, pivoting, downsizing, changing focus, etc.) and I encountered cases where a participant company suddenly experienced an *earthquake* of sorts that resulted in many unexpected voluntary or involuntary departures, relocation within the company, or simply a sudden intense focus. These latter events tended to cause a certain *hunkering down* for a period of time and individuals would

be somewhat less inclined to allocate time for me. The impact of all of this on the research was not significant, but it did create some logistical challenges at times and caused some minor interference with relationship continuity.

3.2.4 Data Collection

The core data collection techniques used in the study were scheduled individual semi-structured interviews and group observation sessions. Being on-site, at times for extended periods, I was also able to have had frequent and unplanned informal discussions as well as observation events that provided further rich material for memos or directly for analysis. No secondary data sources were used.

The iterative data collection and analysis was conducted from January 2017 through March 2020. During this period, approximately 500 hours were spent at the participant companies' premises, including formal and informal contact with individual and team participants.

All individual interviews were loosely guided by the semi-structured interview guide (see Appendix D) which evolved throughout the study as various topics came into and out of focus, an example of *theoretical sampling*. I tried not to use the guide except as a starting point and/or a realignment tool if and when needed. I attempted to create a casual atmosphere and tone to the interviews where the participants felt a freedom to talk about their experiences in a manner that was natural for them personally and to allow for this, I tried to minimise any apparent rigid structure to the interviews. At all times, the actual interview questions were informed by the

emerging analysis and my intuition in the moment. I also used my industrial experience and emerging thoughts to stimulate the conversation, to look for agreement or disagreement, or nudge the conversation towards a general topic I wished to explore. Another technique used, in select circumstances, was to allow a participant to talk at length about something that really mattered to them – their background, point of view on a controversial topic, etc. At times this consumed a considerable amount of time without directly producing useful data however, these types of techniques often produced value later. Sometimes even in the same interview, a more relaxed interaction with the participant seemed to help them feel that it was a casual conversation with a safe party rather than an *interview* or a forced discussion on a topic where they might feel that they had to choose their words more carefully. Many participants, however, were passionate about the research topic and this made the work of conducting the interviews quite easy. I encountered a few cases where an individual became aware through a colleague that I was conducting this study and was so passionate about the topic that *they approached me* asking to participate. In one case, the individual was employed by a company that I had not yet invited to participate. I accepted the individual's offer on the basis that, if they cared about the research topic *that much*, I would accept their offer without hesitation.

All interviews were conducted in-person which afforded me the value of non-verbal communication in addition to the verbal transcript. Interviews were recorded (by permission) on mobile devices and transcribed in their entirety as soon as possible after the interview. I personally transcribed all recorded interviews in Microsoft Word. Although this was very time-consuming, it offered me the value of *re-living* the interaction by hearing the voices again and re-experiencing the conversation more slowly. This very often resulted in insights not fully

appreciated during the interview due to the current of the discussion. See Appendix E for an example of an interview extract with my initial codes and comments made during transcription. Interviews were typically limited to approximately one hour in duration out of respect for the time pressures that many of the participants face in their daily work lives. This also helped to keep the transcription burden to a tolerable level.

Group observation sessions consisted of myself, as unobtrusively as possible, acting as a *fly on the wall* during interactive group meetings. These meetings were most often some form of team or, inter-team, planning sessions. The level of detail in the meetings varied depending on the process model being used within the company and where in the release cycle the meeting was being held. The meetings varied in length from 30 minutes to several hours. These observation sessions were described in rich detail as soon as possible after the session. Reflexive notes were sometimes included when the meeting was being described and/or in subsequent coding activities (see Section 3.2.5.1 Coding). I avoided or kept to a bare minimum, any notetaking during the observation sessions so as not to distract the team in action and to ensure that I was as attentive as possible to the nuances of the meeting. Permission to be on the company premises allowed me to also observe small ad hoc meetings which I did not process as *formal* observation sessions yet were nonetheless useful in understanding some of the communications that occurred outside of the defined rhythms. It also afforded me the opportunity to participate in ad hoc 'watercooler' conversations with individuals that helped not only gain a general pulse of what was going on but also an opportunity to follow-up on specific points that may have surfaced after an observation session or interview.

At the beginning of this research, I expected that the individual interviews would be the primary source of useful data with the group observation sessions placing insights from the interviews into a richer context. As time progressed, however, and the focus shifted towards what was going on *around* the teams within the company more than what was going on *within* the teams, I began to gather more subtle, and often more important, data from the observation sessions. I then found the observation data as being the richer source of useful data with the interview data supplementing and enriching that observation data by adding individual perspectives.

Interpretive qualitative research methods require a certain *entering into* the participants' world. A necessary condition to enter their world as much as I was able was to show respect for the research participants. My primary technique to demonstrate that respect was by establishing rapport with them. This was critical to data collection because there is a real sense that they were *giving* me the data more than I was *collecting* it. My extensive industrial experience gave me a foundation of credibility upon which I was able to establish that healthy rapport. Establishing rapport was important for another reason as well, namely that I would always obtain agreement or invitation from someone on a team (typically a team lead) to attend a group session as an observer. It was necessary to establish sufficient rapport with that individual not simply to receive the invitation to attend but because, in a fairly real sense, they felt they were permitting my attendance on behalf of all the other attendees. In a few cases, I established rapport so well and with enough team members that they viewed me as an honorary member and would sometimes ask me to opine on certain topics that arose in meetings.

The iterative nature of the research method requires that all data be constantly analysed in light of new data collected. The Constructivist Grounded Theory method labels this the *Constant Comparative Method* (Charmaz, 2014), which involves making comparisons in the data during every stage of the analysis throughout the study. In light of the emergent nature of this study and my deep professional experience in the area, the *constant comparative method* occurred almost unconsciously. The specific techniques that were applied during this *constant comparative method* are described in Section 3.2.5 next. My use of the *constant comparative method* was key also to determining when *theoretical saturation* (Glaser & Strauss, 1967, p. 61) was reached. Theoretical saturation is the point at which new data contributes little or nothing to the existing descriptive categories. At this point, the researcher considers the categories *saturated* and deems there to be no point in continuing data collection. The research then moves on to conceptualise higher-level generalisations grounded in and subsuming the initial set of categories.

While the data used in the analysis was limited to the data collected from the participants, the analysis was also shaped by insights gleaned from literature as I assessed the theory as it was emerging in the context of extant scholarship. Therefore, the *constant comparative method* ended only when the study ended.

3.2.5 Analysis Techniques

Data do not provide a window on reality. Rather, the 'discovered' reality arises from the interactive process and its temporal, cultural, and structural contexts (Charmaz, 2000, p. 524).

In the next two sections, an overview is provided on the use of the two core Constructivist Grounded Theory method analytic techniques (Charmaz, 2014) that I used – coding (initial and theoretical) and memoing, described in Sections 3.2.5.1 and 3.2.5.2 respectively. I provide select examples to further illustrate and identify the primary validation actions taken. In Section 3.2.5.3, the technique of clustering is briefly discussed, followed by a discussion on the research quality in Section 3.2.5.4.

3.2.5.1 Coding

---

coding is just one way of analyzing qualitative data, not the way. There are times when coding the data is absolutely necessary, and times when it is most inappropriate

(Saldaña, 2013, p. 40, emphasis in original).

The Coding Manual for Qualitative Researchers by Johnny Saldaña (2013) was an invaluable resource for my coding efforts. Despite having written a very comprehensive book on qualitative research coding, Saldaña's quote above demonstrates his very objective view of this qualitative research analysis technique. For this research, I found coding appropriate, even necessary. And, agreeing with Saldaña, I also found it as just *one* way to analyse the data.

Each instance of the data sources described above was analysed - each interview transcript along with the descriptions and reflections from each observation session. The analysis also included

reflective notes arising from a wide range of stimuli, (see Section 3.2.5.2 next on *Memoing*). Guided by *theoretical sampling* described earlier, the themes that emerged were grounded in the data and reflected in the participants' voices.

The coding took several forms, all being defined Grounded Theory method coding analysis techniques.

First cycle coding is a subjective phase of creating codes from the collected data, summarising what seemed to be valuable in the coding moment. I examined (not necessarily coded) the data line-by-line and word-by-word (a very laborious and time-consuming process). However most codes came about in second cycle coding and memoing.

Throughout this chapter and the next, direct quotations are selected to assist in illustrating key concepts and the emergent theory. Quotes are from transcripts of individual interviews, notes taken from team observation sessions, or researcher memos. These quotes are presented **"bold and in quotations"** while codes/categories/themes from the analysis are shown as ***bold and italicised***. Terms used for codes, categories, and themes are generally my own labels. The reason for this was that while the participants' proficiency in English was generally quite good, they represented a variety of cultural and linguistic backgrounds, thus the grammar and vocabulary was somewhat variable. I found occasions where the same concept would be described by different participants using different terminology and I would sometimes have to clarify the precise meaning of terms and phrases used. This was the case also for company-specific idioms and definitions of common terms. As a result, the use of in vivo terms and

phrases was minimised in the analysis. When in-vivo quotations were used, they were to add richness to the description, not to label categories or concepts. These in-vivo quotations are presented in quotations, without bolding, followed by the source reference.

The first cycle coding techniques used were primarily *descriptive* coding (noting the topic, key point, or concept) with some *initial* (open-ended starting point) (e.g. ***need to communicate***), *process* (noting the activity) (e.g. ***backlog grooming***), and *versus* (X vs/or Y) (e.g. ***less deference***) coding.

Two other first cycle techniques were tried and later discarded. The first was *attribute coding* (experience, role, gender, culture, size, methodology, etc.) which, initially, I thought could be useful but I soon concluded that the emerging themes remained collective and were seemingly unaffected by any of the attributes that had been captured on individuals and companies. As the themes began to emerge along company lines rather than teams, the team attributes became unimportant as well. The other code technique that was discarded was *magnitude coding* which was conducted for a short time (using Agile T-shirt sizing scales) until I concluded that this data was not contributing to the thematic direction of the analysis.

Second cycle techniques used were *theoretical* and *focussed* coding (Charmaz, 2014), which are techniques primarily used to refine the concepts and create categories. I tried *axial* coding, but found it too structured for this study. Most second cycle codes came about through theoretical coding using the *Key Point* approach (Allan, 2003), i.e. identifying key points rather than

individual words to better allow concepts to emerge. As with first cycle codes, second cycle codes and categories could survive, change, or be discarded at any time.

I illustrate a brief example of these analytic movements after discussing the technique of Memoing in the next section.

No *a priori* codes, categories, concepts, or themes were applied to the collected data. This avoided any risk of fitting data to pre-existing concepts. On rare occasions, *in-vivo* coding was used, i.e. using the actual terminology of the participants. There were two primary reasons for limiting the use of *in-vivo* coding. The first was that many participants had English as their second language (ESL) and, representing a wide range of cultural backgrounds, their choice of specific words could vary while referring to the same concept. The second reason was that it was sometimes observed that vocabulary in common use within one company that would not necessarily have the same meaning in another organisation or in the industry at large. In-vivo coding would have applied inappropriate importance to the specific word choices, likely overlooking the emergence of some categories, whereas what was important was what the in-vivo words and phrases were *referring to*, the specific words used were rarely significant. My understanding of the domain and experience in working with highly mixed cultural groups allowed me to interpret the specific choice of words *in context* and create initial code labels to the language (using key point coding) thus minimising the impact of English as a second language, corporate-specific, or non-industry standard vocabulary. When I did use *in-vivo* codes, it was mainly to capture the colour of the language used by the participants, while other terms could have captured the meaning equally well.

As it is inherently with the Grounded Theory method, coding is an emergent, cyclical process. Static codes by themselves do not mean much. Codes simply reflect what is seen at a particular point in time, and that may change. In action, however, they are not simply labels, they are the result of heuristic techniques that link data to ideas. I often renamed, subsumed, or even discarded codes during subsequent cycles of analysis. Being grounded in the data, my view was that if it turned out that a discarded code was useful after all, it would emerge again from the continual searching through the data roughly, an activity colourfully labelled "pawing" by Ryan & Bernard (2003), handling the data multiple times … living with the data. It was through this living with and pawing through the data that fresh relationships, patterns, and concepts emerged. In my experience, the iterative and heuristic nature of the entire coding process cannot be overemphasised. Simultaneous coding (Miles & Huberman, 1994) occurred often throughout where the data supporting a code could, in whole or in part, be grounding for other code(s).

The *constant comparative method* meant that as additional data was collected or fresh thoughts emerged or when different questions were asked of the data or when new lens' on the existing data emerged, opportunities arose for creating more codes and/or refining (even discarding) existing ones. At times, this led into unforeseen areas.

Eventually, when enough category results had emerged, theoretical coding produced core categories, around which as many other categories as possible were grouped. It was through this theoretical coding process that the theme of *Broadening the Lens* became the central storyline. Figure 3-2 below illustrates a summarised example of the structuring from samples from the

study of different types of theoretical & key point codes, categories, themes, and finally to the theory of ***Blurring Boundaries***.

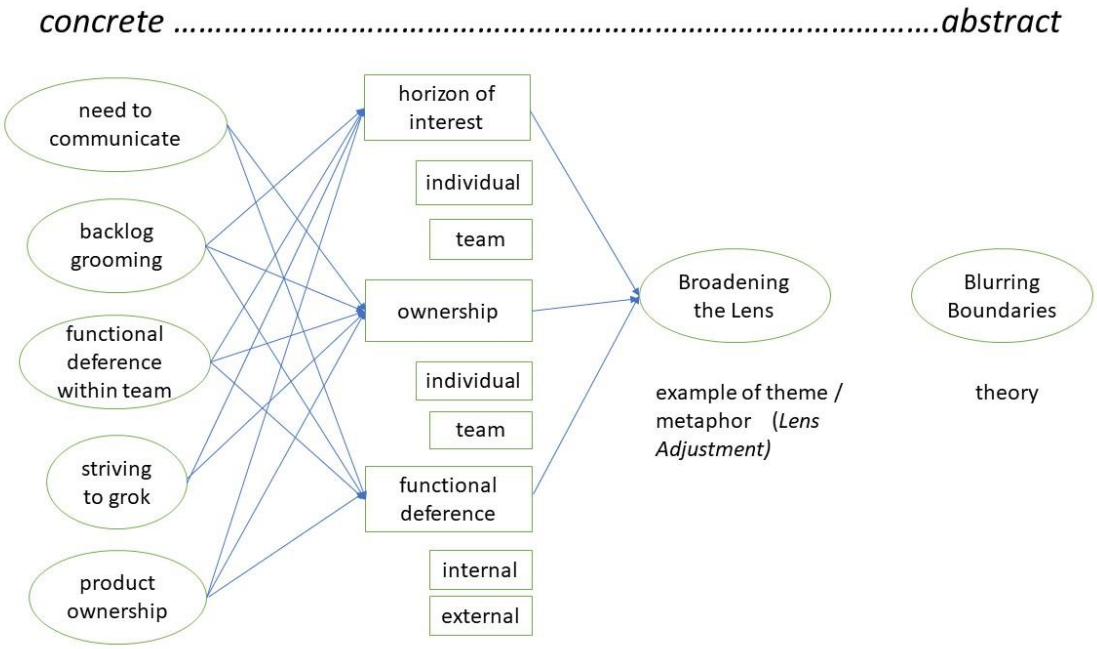


Figure 3-2. Summarised Example of Concept Structuring

In general, I found coding useful, although useful not so much in its own right as much as useful in the sense that Gordon-Finlayson describes, "coding is simply a structure on which reflection (via memo-writing) happens. It is memo-writing that is the engine of grounded theory, not coding" (2010, p. 164, emphasis in original), appearing to be aligned with the views of Saldaña quoted at the beginning of this section. The works of Gordon-Finlayson and Saldaña kept me mindful to ensure that coding did not become reductionist.

3.2.5.2 Memoing

Scrutiny of the notes offers both empirical certainty and intuitive reminders. Insights emerge also from the subconscious … never penned on paper. There are serendipitous connections to be made, if the writer is open to them. Writing and analysis comprise a movement between the tangible and intangible, between the cerebral and sensual, between the visible and invisible. Interpretation moves from evidence to ideas and theory, then back again. There can be no set formulae, only broad guidelines, sensitive to specific cases.

(Judith Okely, 2010, p. 32)

As a core technique in the Grounded Theory method, *memoing* is about capturing the reflections of the researchers about the coding, how the process of the study is unfolding, emergent properties in the data, … – all the analysis possibly leading toward a grounded theory. They are notes made by the researcher, to him/herself, throughout the study to capture and elaborate on codes, reflections, ideas, hunches, etc. relating to the data and categories. This is where data and ideas soak together, and new insights emerge. This is also the primary place where researchers contribute experience, voice, and positionality along with deep reflexivity to intermingle and intertwine into the research findings. Thus, I agree with Gordon-Finlayson (2010) that coding is a feed into reflexive memos and that it is these reflexive memos that are at the heart of the analysis.

I wrote memos whenever an idea occurred that felt like it ought to be captured. It was sometimes during the coding process as thoughts and questions arose, resulting in some codes becoming *mini-memos*. It was not uncommon for a thought to occur while on the way to a meeting or even on a Saturday night over a glass of wine and thus I came to sacrifice memo 'management' in favour of 'creation convenience'. This resulted in having memos on laptops, PCs, various notepads, sticky notes, voice memos, etc. and in various forms, on their own, embedded in interview transcripts/coding, or contained within reflections on observation sessions. Arguably, it was not the best content management strategy however the risk of losing a useful insight or question by not capturing it when it emerged seemed to carry with it a larger cost. Sometimes, I tagged a memo as being superseded by a subsequent memo, at other times tagged or discarded a memo as no longer reflecting my point of view, while allowing for the possibility that I might adopt that viewpoint again later. As I discovered, given the source and nature of what memos intend to capture, it was important that their creation and management have as few rules and constraints as possible. I was always prepared to create categories for the memos if the need became strong enough. As it turned out, I was able to proceed without having to do so. Memos were ultimately infinitely revisable, sometimes embarrassingly informal, and subject to *evolutionary laws*.

By way of example, the following is an excerpt from a multi-page memo in an *Emergent Patterns* category, the second memo on the *Effect of Organisational Model*. This memo became the seed for a subsequent conference paper (Fuller, 2019a).

.... The striking difference in the teams between these two companies is with regard to how the individual product team members see their membership in the product team and how that affects their participation on their product team.

Product team members in Company A participate in team meetings very fully, seemingly with little reservation regarding topic. It's not uncommon for a person to express a point of view quite strongly, sometimes with a hint of deference at the end by saying, "but I'm not a (role)". I've had a couple of team members in this company express some frustrations along the lines of "my opinions are not always valued", meaning they're not always supported, however the important observation is that they still feel free and are willing to express their views in all directions on all topics.

Product team members in Company B are more selective during team meetings in how they participate. On topics relating to their functional role, they will express, debate, etc. very fully. However, on topics relating to a different functional role, they appear to limit their discussion to seeking clarification and understanding and refrain from critique or submitting alternative viewpoints. The oft and cautiously expressed explanation is "I trust them". I've also heard "I'm too busy to get involved in their topics", or "if I get involved, it will look like I don't have enough work to do". It's quite clear that, within those product teams, not only is there a clear segregation of work

performed (typical) but also clear lines of demarcation regarding responsibility/ownership within the team.

The image that arises for me is that the members of the product teams in Company A are all wearing their specific "product" T-shirt, feel they're full members of that product team and feel full ownership of the product, regardless of what functional expertise they're bringing to the team. A team sports analogy may be applicable here.

Product teams in Company B appear to be more assemblies of individuals bringing specific functional expertise with a looser sense of product ownership. It's as if each person is wearing a T-shirt representing their functional department and including a (possibly temporary) badge on it that identifies what product team the person belongs to at the moment. The primary affiliation seems to be with the functional department and the product is more the current assignment. From the outside, it may look like a product team, but it feels more like a collection of consultants or hired guns. ….

Memo journal entry, 2018.June.06.

Following is an example taken from my analysis notes of a thread of analysis illustrating various techniques discussed above – *initial coding*, *constant comparative method*, *focussed coding*, *key point coding*, reflexive *memoing*, and *theoretical coding* – and how they are able to co-create and foster model emergence.

Interview data (with key point coding): (in response to the interviewer's probing for insights into cross-functional tension) *"PM would say this is what I want and dev would say but that's not what you can get. Neither side understood why." - Participant 07, company W.*

Key Point: ***strong functional distinctions***

Interview data (with initial coding): (in response to the interviewer's question to a developer about his interactions with UX designers on the team who reported into a different department than did this developer) *"…really only to ask detail questions that might come up when I'm writing the code. But I'm careful not to criticise their designs.* ***They're the experts, I trust them.****". – Participant 20, company V.* (initial code in bold)

Constant Comparative Method (with focussed coding): (seeing echoes of the ***strong functional distinctions*** and this "***they're the experts, I trust them***" sentiment in other interviews, I created a *concept* of ***showing functional deference within the team***. Later, as indications of a distinct *lack* of functional deference in certain other teams emerged, I renamed the concept ***functional deference within the team*** with an intent to examine why it existed strongly in some teams and was notably absent in others).

Observations session data (with reflexive memoing): (observing development teams interacting with dissimilar functions outside of their team, I began to see functional deference also appearing in a different dimension). From the researcher's

observation note from a product road-map presentation by product management to the development team, *"… a rip-roaring debate between the dev team and the PMs about the vision of the product and how it might evolve. Stark contrast with the passive, near silent, reception by the dev team at W in a meeting with a very similar purpose. This is functional deference (or not, in this case) but between teams of different roles rather than between roles within a dev team." - Memo journal entry, 2018.Aug.15.*

Constant Comparative Method (with theoretical coding): (comparing this memo noting inter-team deference with the data and codes I had relating to functional deference *intra*-team, I then created a *category* of ***functional deference*** which had both intra-team and inter-team sub-categories).

Comparable analytic journeys occurred for other categories which were related, grouped, and ultimately led to the theory of Blurring Boundaries.

Figure 3-3 below shows a high-level view of the Grounded Theory analysis model, illustrating the relationships between data collection, coding, and memoing that iteratively nurture the emergence of categories, themes, and theory.

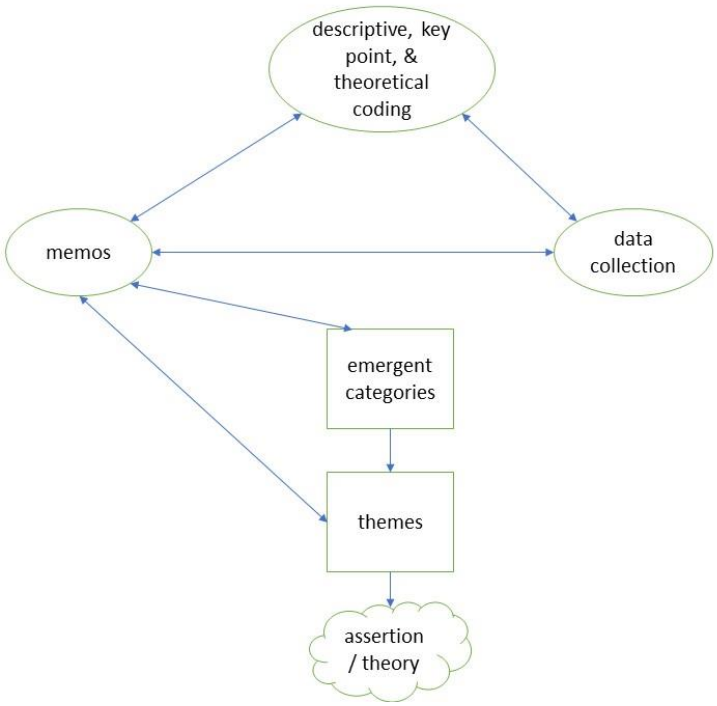


Figure 3-3. Grounded Theory analysis model

3.2.5.3 Clustering

Clustering is a diagrammatic technique to explore ideas, categories, processes, etc. and their relationships. As with memoing, clustering does not create lasting artefacts, rather they are aids in conceptualising, and as such, there are no rules, simply some guidelines to help a researcher get started.

I tended to use clustering as a non-textual technique to play with ideas whenever I felt the relationships amongst categories or concepts were seemingly opaque and needed some visualisation assistance. At other times, it was a tool to combat writing blocks. Sometimes it

was a useful technique to share thoughts during validation conversation. Overall, however, it was not a key tool in the study.

3.3 Research Quality and Limitations

3.3.1 Research Quality

Charmaz (2014) notes that expectations for quality in the grounded theory research method depend on who forms the expectations and for what purposes. She offers the following four evaluation criteria that account for the empirical study and development of the theory: *credibility*, *originality*, *resonance*, and *usefulness*. I will discuss each of these criteria briefly and what I did in the conduct of this research to address them.

Credibility refers to the degree of intimate familiarity achieved in the setting and topic, whether there is sufficient data and range of observations, and the soundness of the analysis performed.

I had intensive, on-going involvement with participants in the form of extended participation and the ability to *live* in the participants' workplace. With this close involvement, the data collected was grounded in the experiences of a large number and a wide range of participants and settings. This provided richer data that was more direct (i.e., less dependent on inference) and permitted repeated observations and interviews, all of which helped rule out spurious associations, systematic biases, and premature theories. I collected and used rich data (transcribed interviews, thick descriptive notetaking of observations) that helped provide a fuller and more revealing picture of what was going on. Data collection only stopped once I felt that *theoretical saturation*

had been reached, when new data failed to add significant additional insights, avoiding premature claims. The reflexivity inherent in the core memoing and constant comparative method techniques that I used helped minimise bias. Interview transcripts, observation session notes, and memos were reviewed soon after they were created to capture thoughts while they were new. They were also reviewed regularly during the constant comparative method allowing for possible enrichment of the analysis with the passage of time and with new data. As part of the standard closing of an interview, all interviewees were offered an opportunity to review the interview transcript once it was complete. Not surprisingly, not one interviewee expressed an interest in doing so. Not only were they very busy but it is also likely that they trusted that the transcription of the recording would be accurate. I also conducted participant (and non-participant) checks to obtain reactions on the emerging analysis and conclusions being drawn at the moment which provided early and multiple perspectives on the analysis and helped rule out possibilities of misinterpretation.

Originality asks whether the research created fresh ideas and insights, if it introduces new concepts to consider, and whether it challenges, extends, or refines current ideas or practices.

This aspect of the research quality was a continual challenge throughout much of this study since, from my extensive industry background, everything I observed had a ring of familiarity to it. Often the inner voice would say, "There's nothing new in all this, I've seen it, or variations of it, many times over the years.". Which oft-times led to the question, "So then why is it still a problem?". Then I remembered that my time in industry was characterised by the pressures of deadlines and other urgencies, thus did not afford me, nor my professional peers, the opportunity

to step back, take the time to reflect, analyse, compare, etc. Hence, what was appearing obvious was so only because of my positionality. I submitted several conference papers on elements of this research as it unfolded throughout the study and enjoyed a very high acceptance rate which I believe was due in large part to the originality of the topic itself and also to the findings. Reviewer comments often noted that the topic was an important one and conference organisers said they felt the topic would stimulate lively conversation. This was echoed throughout the study during the peer validation discussions that I held with participants as well as with peers in my professional network. Finally, in challenging certain common industrial practices, I am revealing some relatively unexamined assumptions behind some of those practices, providing an additional lens on originality.

Resonance, as it relates to research quality, is about whether the results make sense to those living the experience, whether they care and whether it helps them.

Based on my experiences in industry, I constantly directed the study towards topics that would be relevant in practice. Throughout the study, I was often approached proactively by company executives and development practitioners expressing an interest in seeing the final results when they were complete. This speaks to some degree of resonance and reflects also the point made above that leaders in the technology space are too busy to do the research and analysis that this study has done. When it was appropriate for me to discuss emerging insights with participants or question them further to learn from their perspectives or to confirm my own, there was always enthusiastic willingness to engage, despite demanding work pressures. As noted earlier, there were also cases where individuals who were not participants, upon hearing of the study via their

professional network, proactively approached me asking if they could participate. The topic resonates and is one that many care about, in a variety of roles from software engineers and Scrum masters to CEOs.

Usefulness in research quality is viewed in terms of whether the insights are actionable and if the applicability is broad enough to be interesting.

Applicability of the findings is confined to *the context of software product development* (the research context) and was assessed primarily via on-going peer reviews of the emerging insights and resulting theory with software product development leaders in the study, in my professional network, and at technical conferences. This typically took the form of networking and *shop talk* with colleagues. Not only were their reactions useful, but simply verbalising my findings and thoughts often generated new valuable insights and ideas. I received no feedback regarding applicability that was contrary to what the results indicated and, as with senior executives in the participant companies, several expressed strong interest in receiving the published findings. Nothing in my analysis nor the peer reviews challenged my claim of sufficient applicability of the findings in the general context of software product development. Future assessment regarding the transferability to other complex product development areas remains open.

3.3.2 Limitations

Throughout the research, I employed various strategies (Maxwell, 2012) to mitigate threats to validity, as well as taking actions to allow an assessment of the degree of transferability of the

findings. Qualitative research studies of this type are subject to several limitations and this was no exception. I review quality limitations here by discussing three topics: *comprehensiveness of the data*, *appropriateness of the interpretation*, and *applicability of the claims*.

3.3.2.1 Comprehensiveness of the Data

I maintained intensive, ongoing involvement with the individual and organisational participants. This extended participation also included the capability to 'live' in the participants' workplace, observing and interacting, formally and informally.

It was through this intensive involvement that all data was collected, and it was upon that data that reflections were made. This ensured that the data was grounded in the experiences of a large number of participants and in a range of settings, providing richer data, data that was more direct, i.e. less dependent on inference. This on-going involvement allowed also for repeated observations of teams and individual interviews, both formal and informal. It also offered me a convenient opportunity to re-examine and review observations and analysis with participants, helping to rule out spurious associations, systematic biases, and premature theories. This provided intimate familiarity with the topic and settings and sufficient data to merit my conclusions (Charmaz, 2014).

I did not do content analysis. That is, no development process artefacts were included in the analysed data. This was a conscious decision as I did not feel that insights into the research question were likely to be found in those materials. However, there was one topic that arose that

was outside the scope of the study that might be explored using content analysis, namely, the role of domain modelling in the team's attempt to capture insights and create some form of collective memory of that learning experience. This could be a useful topic for future research. In my sampling, however, there was no formal domain modelling performed by any of the participant teams.

Similarly, I did not treat development methodology as an attribute in the analysis. The participant organisations all had different variations on methodologies commonly used in industry and, while I have opinions that the methods witnessed were not all equally effective, I believe that the factors relating to the research questions were not methodological in nature. Future research could confirm or disprove this.

3.3.2.2 Appropriateness of the Interpretation

I collected and used rich data (transcribed interviews and thick, descriptive notetaking of observations) that provided a fuller and more revealing picture of what is going on. All individual participants were offered, as part of the standard closing of the interview, an opportunity to review the transcription once it was complete. None have chosen to, which I interpret as trust in the interviewer and the process. However, many expressed a strong interest in seeing the final research findings when they are made available, demonstrating a keen interest in the topic and the potential usefulness of the findings - early indication of resonance and support for usefulness, according to the Charmaz (2014) definition of research quality.

Triangulation, data collected from a range of participants and settings, further reduced the risk of chance associations and systematic biases. The diversity in participants included a range of corporate types (size, age, markets), teams, and types of individuals (experience levels and functional roles).

Participant (and non-participant) checks were conducted periodically throughout the study to obtain reactions to both the emerging analysis and conclusions drawn. This offered multiple perspectives on the analysis and helped to rule out possibilities of misinterpretation. No feedback has yet been received that is contrary to what the findings have indicated.

Due to the interpretive nature of the Constructivist Grounded Theory method and the role the researcher perspective plays, replication of results is inherently difficult. However, I am ensuring auditability by maintaining detailed records of data collected from which one can see the general stability to the moment I declared *theoretical saturation* had been reached.

The measures taken to assess the appropriateness of the interpretation do not alter the fact that the interpretation of the data is mine and mine alone. It reflects what I saw and felt was relevant to the research question, filtered through my own professional experience. I performed several validation techniques to achieve sufficient comfort that the interpretation and resulting analysis resonated with participants and industry peers and, on that basis, I stand behind the analysis. However, different eyes notice different things.

3.3.2.3 Applicability of the claims

This research result is substantive theory applied to software product development in restricted domains. In the theoretical sampling as described in Chapter 3, I explicitly excluded product domains likely to be familiar to the development teams and make no claims regarding transferability to such domains. Nor do I claim transferability to bespoke software development, software development which has a much more *transactional* nature with less reliance on the development team understanding the more tacit aspects of the context of the product requirements. My belief is that the theory may not transfer to those non-product types of software development however, this is a question for future research to confirm. As the theoretical sampling included only software product development, I make no assertions regarding transferability to complex technology product development spaces other than software. My belief is that this theory is quite likely to apply to high-tech product development generally where there is a critical dependency on innovation and creativity applied to complex problem domains. However, that transferability is also an area for future research.

The transferability claims that are made were checked primarily via reviews of the resulting theory with industry peers (software product development leaders).

While participants were selected from companies with a broad cross-section of attributes, there remains the possibility that another sample set could produce materially different data. If this were found to be the case, I believe it would not invalidate the findings from this data set, but rather create a new perspective on the transferability of the findings.

I have very long and deep professional experience in the software product industry and I recognised at the outset of this research that this strong positionality (my position in relation to the study) shapes my objectivity and subjectivity in many aspects of perspective in this study. While acknowledging the challenges this presents, I consider this experience, and the bias it creates, to be an asset to this research. As pointed out in Chapter 3, the Constructivist Grounded Theory method embraces researcher experience as a manageable asset to the research. Eisner (1998) referred to the expert ability to "see what counts" – the sensitivity to tacit elements of the data, meanings, and connotations. This guided my research, supported fully by the collected data, towards questions and insights that I believe matter. Simply put, the researcher *groks* the world of the participants. Also, my many years of experience working alongside the same types of people as the participants afforded me considerable comfort, understanding, and ability to gain rapid rapport with the participants which resulted in dialogue that would have been very difficult to achieve by researchers lacking this level of industrial experience.

This credibility along with the resonance received from the participant checks and peer reviews support a high degree of usefulness of the findings. As a result of this analysis, discussions with industry peers and participants, and my own professional experience, I offer recommendations for industrial practice in Chapter 5 - Discussion.

3.4 Tools

Various tools were employed during this study. Early in the study, NVivo (2018) was used, a qualitative data analysis software (QDAS) product, to provide a repository for the study's data

and analysis results and tools for performing certain analytic activities. However, I found that I was spending too much time working with the tool itself than working with the data. As the datasets were not technically complex, the tool served mostly as a data repository and, for that purpose, it was too excessive for my needs and carried too much overhead. Other tools used included mobile devices to record interviews which were later transcribed using word processing software. Initial coding most often occurred directly in those transcribed documents. Subsequent coding analysis was performed with word processing and spreadsheet software, sticky notes, and simple diagramming tools. Memos were captured using various tools - word processing software, voice recorders, pen and paper, and mobile note apps (primarily Evernote (2020)). Mendeley (2018), a software product for managing research material, was used as a reference management tool.

3.5 Summary

The goal of this chapter was to outline the research method used to address the research question. I discussed the research context, researcher positionality and reflexivity, participants, data collection, analytic techniques, and research quality.

Although the rationale for selecting Grounded Theory for this research and, specifically, selecting the Constructivist flavour of the Grounded Theory method seemed sound at the outset, I am even more convinced of its appropriateness now that the study is complete. The iterative nature of the method allowed the research to be steered based on the findings and conditions rather than struggle when something was not unfolding as might have been expected. The

approach of “let the data speak for itself” provided the confidence that what emerged was real and valid rather than finessed to fit pre-existing beliefs. The entire method resonated very strongly with my own principle-based Agile mindset.

Reflecting on this through an even more personal lens, this study was a first-hand experience of the difference between “knowledge” and “understanding”, a distinction that became core to the findings. Knowledge is something one can gain through learning. Understanding is something gained through lived experience. It occurs to me in hindsight that the Grounded Theory method is “requirements engineering” applied to the problem of understanding. This research approach allowed me to “live” the experience, to some degree, with the participants. I was “stepping into their world”, as well as I could, to create a context within which I could better understand the details of what I observed. This distinction between knowledge and understanding is particularly relevant as it points to a deeper essence of the research question itself – how do software teams collectively move from *knowledge* of the product domain to *understanding* of the product domain? As I discuss in the following chapters and concluded, a key element in teams moving from knowledge to understanding is collective empathy.

In the next chapter, I present the important findings of the study.

Chapter 4: Findings

4.1 Introduction

This chapter presents the key insights from the analysis of the data collected in this study; insights that then led to the substantive theory that emerged.

The purpose of this study was to examine what factors influence (support or impede) cross-functional software product teams (CFPTs) in collectively achieving a deep and shared understanding of the product domain. Using the Constructivist Grounded Theory method as described in detail in Chapter 3, I iteratively analysed the data collected on-site from several software product companies. From that analysis emerged the select grounded categories and themes presented in this chapter.

The data was collected from observation sessions of teams in action and from individual interviews. These teams and individuals were sampled from software product companies that developed products for markets, markets not likely to be intimately understood by the typical member of a CFPT. The results emerged iteratively as additional data fuelled deeper interpretation. Insights were grounded in the coding and reflexivity that nurtured the evolution of codes to categories to more theoretically oriented themes.

At times the results and observations are discussed here using specific terms to provide an indication of the frequency of applicability. These frequency references refer to participant teams as well as participant companies. The phrases “strongly” and “most” are used to discuss

concepts presented by approximately 80% of the participants or more. The term "few" was used to indicate concepts presented by approximately 25% or fewer of the participants.

Toward answering the research question, many categories were identified that contributed to the analysis. Of these, four major categories factored more prominently than the others and which, upon further analysis, led to the identification of two factors that directly answer the research question. These major categories were - ***functional deference***, ***primary affiliation***, ***horizon of interest***, and ***alignment with expectations***. These are defined in detail in this chapter.

Both of the influencing factors identified from the major categories are settings surrounding CFPTs within companies, namely:

1) The organisational structure surrounding the team.
2) The product planning process and the team's role in that process.

The findings illustrate how these two factors, separately and jointly, can inhibit a product development team's ability to grok the domain of their product or, conversely, how they can allow the freedom for a team to do so to the extent of its ability. These results were grouped around the thematic metaphors of ***Distinct Boundaries*** and ***Broadening the Lens*** to form the foundation for the emergent theory of ***Blurring Boundaries*** which is presented and discussed in Chapter 5.

In the rest of this chapter, Section 4.2 next describes a description of the evolution of the research question during the study. Section 4.3 provides a description of the major categories

that emerged, while Section 4.4 explains how each of the two factors answering the research question emerged from examination of the major categories. Sections 4.5 and 4.6, respectively, describe these two influencing factors in detail. Finally, Section 4.7 summarises the chapter, describes the themes that emerged from the analysis, and prepares for the discussion in Chapter 5 which positions and interprets the emergent theory and discusses implications for practice.

4.2 Allowing the Research Question to Evolve

At its inception, this study was motivated around an initial research question which asked what CFPTs did differently from one another that would explain why some teams collectively grokked the product domain better than others. I believed that this question could be answered by sampling a sufficient number of teams in a variety of software product companies, examining the internal conditions and dynamics of those teams, and analysing the differences between them that would explain the phenomenon observed in my industrial experience. However, it became apparent early in the study that any differences in collective domain understanding between teams in any one company were subtle compared to the striking differences observed in behaviours between the *collection* of teams in different companies. This suggested that there could be strong environmental factors at play, i.e. factors relating to the differences between the companies more than between teams within any one company. I was then faced with a choice. One option was to stay with the original research question. This would require a reset of the study with a different theoretical sampling strategy that would control for those environmental factors by sampling companies that were each large enough to offer a sufficient number of

participating teams to permit the examination of differences between teams within reasonably consistent contexts.

The other option was to broaden the lens of the study to examine the environmental differences that impacted the teams' ability to understand the product domains so dramatically. As the environmental impact on the teams was so striking, this option was chosen, with the belief that the findings were more likely to have quickly actionable benefit to a greater number of industry practitioners. It was also reasoned that the results from this revised research question would likely be valuable consideration for researchers or practitioners in attempting to answer the initial research question in the future.

In summary, allowing the research question to evolve led to it shifting from what individual teams did differently to explain their different levels of grokking the product domain to what external factors impacted a team's ability to progress towards grokking their product domain as stated in Chapter 1. This was a shift from examining *how* teams collectively grokked to examining what factors impacted their *potential* to collectively grok.

4.3 Influencing Categories

From the data collection and use of the analytical techniques described in the previous chapter, many categories surfaced and many of those are used throughout this chapter to tell the story that emerged. Four major categories became prominent in the analysis. These major categories were used as lenses into the organisational structure and product planning process factors noted in the

introduction to this chapter. As much of the discussion in this chapter revolves around these four major categories, they are described in more detail in the next four sections, offering rich examples of each as they are used to describe how they contributed to the two factors of the research question that emerged. The major influencing categories are:

1. ***functional deference***
   - between individuals within teams,
   - between teams and other teams.
2. ***primary affiliation***
   - for the individual team member within the overall organisational structure,
   - for the team to the product and to the product plans.
3. ***horizon of interest***
   - for the individual team member within the team,
   - for the team with respect to product planning.
4. ***alignment with expectations***
   - between senior leadership and the team.

Each of these is described in detail in the following sections and they are examined further in Sections 4.5 and 4.6 in the discussion of how they appear in the context of each of the two environmental factors that emerged that influence the potential for CFPTs to deeply and collectively understand the product domain.

Note that these categories are neither binary nor are they mutually independent. Although the research question and path of inquiry did not call for measurement, underlying the categories in both contexts in which they occur is a spectrum along which the individual and/or CFPTs may be positioned. Generally, I observed that an individual or team would be positioned nearer to one end of the spectrum or the other and, for the purposes here, it was sufficient to simply note which end they were nearest and whether there appeared to be effort expended by the individual or team to move along the spectrum or not. This is discussed further throughout the remainder of this chapter.

From the interpretation of the data, I identified two themes that were woven through these four categories, namely the themes of ***Distinct Boundaries*** and ***Broadening the Lens***, the latter being a unifying metaphorical mechanism which is useful to describe movement along the spectra within established boundaries. I describe these themes in Section 4.7, which provides the entry to the description and discussion of the theory of ***Blurring Boundaries*** in Chapter 5.

4.3.1 Category 1 – *Functional Deference*

Do members of a team contribute as if they are fully committed to the team and striving for the success of the team or do they contribute as if they are 'an expert in attendance' to provide specialist expertise in a well-defined role?

I am using the term *deference* to refer to the behaviour of a team or an individual when one yields (displays submission and respect) to others solely due to the others' specific functional expertise or role being different from one's own. I observed functional deference at the

individual level (between team members having differing functional expertise or roles) and at the team level (between teams with differing mandates within the product planning process).

At the individual level, many examples were observed of team members limiting their contributions on the team to topics directly relating to their particular competencies and they tended to show marked deference to other team members with different functional competencies (internal ***functional deference*****)** regarding topics outside their area of primary functional expertise. For example, a security specialist not expressing an opinion on a usability matter. Conversely, I observed many examples of other individuals showing little to no inhibition due to differences in functional expertise when they were conversing with others on the team or explicitly expressing their points of view. Note that showing little deference was neither a sign of disrespect nor an indication that the functional expertise differences were not recognised. Rather it was a reflection that they were placing the interest of the product first and foremost (***product ownership***) and were contributing as much as they could offer to the team and to the product's success by not allowing relative competency differences to interfere with their contribution or their level of participation.

There is a version of this section's introductory question that would refer to the team as a whole instead of individual team members. To that question, I also observed an *external* form of ***functional deference*** that could exist between teams having different functional mandates relating to the product (e.g. Engineering and Product Management), whether in the same or different departments.

4.3.2 Category 2 – *Primary Affiliation*

What T-shirt are team members wearing?

I observed that individuals displaying strong internal ***functional deference*** on teams tended to behave as if their *home* was a group outside of the team (often a defined functional group) and that their membership on the team was essentially viewed by them as an "assignment", even if it were for an unspecified period of time. As these participating teams were cross-functional, the members on the team could come from a variety of *homes*. During this research, when conducting mutual introductions with a new participant, members on teams showing strong internal ***functional deference*** almost invariably expressed their identity ***(primary affiliation)*** as their competency and/or home group or department first and then, and only sometimes, would identify what team they were currently a member of, even in cases where they had been on that team for several years.

In contrast, individuals on teams with less internal ***functional deference*** (which tended to result in greater ***team cohesion***) identified and positioned themselves, their ***primary affiliation***, as being on the specific team. Period. There was no other *home* for them. When one individual on one of these teams was asked what would happen if they wished to no longer be on this particular team, the person replied, "I'd be job hunting" (Participant 04). One of the participant companies has a well-known corporate profile with multiple products in a large and long-established product domain yet it was common for the product team members to have LinkedIn profiles showing the product name as the company they work for, displaying little (and usually, no) reference to the company itself, a strong display of ***product ownership*** and ***primary***

affiliation with the team. Although these team members were very clear what their individual mix of competencies were, they seemed to view those competencies more as how they contribute the most to the team as opposed to being the defining reason why they are on the team at all. These individuals were very clear that they were part of the product *team* and committed to the success of the *product*. They also smiled more.

As a collective, it was observed that CFPTs with strong ***team cohesion*** (gained from having low levels of internal ***functional deference*** and team members with stronger ***primary affiliation*** towards the team) strove towards more ***product ownership*** and, therefore, collectively cared more about the *content* of the product plan. Teams on the other end of the spectrum with weaker ***team cohesion*** showed to have a lower sense of collective ***product ownership*** and tended to focus much more solely on the *execution* of a narrower interpretation of the near-term plan, showing more ***functional deference*** to other functional teams regarding the plan content.

4.3.3 Category 3 – *Horizon of Interest*

How far into the future does the team as well as individual team members concern themselves?

Although potentially caused by a variety of reasons, which is discussed further in Section 4.4, some team members have a degree of uncertainty, and therefore tentativeness, about their team membership into the future. Comments heard from team members in this state were typified by one direct quote from a participant, "I'm on this team… for now" (Participant 20), who quickly added that they were fully engaged with the team *at the moment* but that there was a reasonable

likelihood that they might find themselves assigned to a different team in the foreseeable future (hence the "for now" phrase in the quotation), which shortened the participant's ***horizon of interest***. In contrast, individuals that saw their ***primary affiliation*** being fully with the team, having no competing organisational context for them to position their future, did not generally display discernible tentativeness to the team nor to their interest in the future of the product, hence a longer ***horizon of interest***.

Looking at the team as a collective unit, it was noted that teams with strong internal ***functional deference*** and lower ***team cohesion*** had a shorter ***horizon of interest***, focussing more narrowly on the plan in front of them (the iteration plan, release cycle, major release, etc.) and paying notably less attention to its context, the longer-term product roadmap. Whereas teams with less internal ***functional deference*** and higher ***team cohesion*** had a longer ***horizon of*** interest – with a broader lens, asking more and broader questions, and putting more effort into looking at the longer-term product roadmap in order to better understand and contextualise the current and near-term workplan (***product ownership***). The factors influencing how far into the future they might concern themselves are discussed in Section 4.6.

4.3.4 Category 4 - *Alignment with Expectations*

How aligned is senior leadership's expectations of the team's mandate and empowerment with that which the team holds and understands?

I observed that teams with low ***team cohesion*** and high internal ***functional deference*** were often less concerned about whether they were as self-sufficient, capable, and/or productive as they

might be, viewing it as having been "management's decision" (Participant 14) to define the structure and process models as they currently were. In other words, they held a view that someone or some group had made conscious choices for these matters to be as they were and that the team was not empowered to change those decisions.

In contrast, I found that by placing the interests of their product foremost (strong ***product ownership***), teams with low internal ***functional deference*** and stronger ***team cohesion*** were more inclined to advocate for what they felt was in the best interest of the product (***striving to grok***), attaching the definition of team success with the success of the product. "**It's our product**", as Participant 03 declared. In these cases, with respect to the degree of product ownership held by the teams, there was strong alignment shown between what senior leadership *expected* and the teams' behaviour.

4.4 A Tale of Two Factors

So what is going on here?

The first insight that occurred when examining these major categories was that the topic of the degree of ***product ownership*** (individual and collective) threaded through them all. The second observation was that there were two opposing characterisations in common for all the four major categories described above and I discovered that each participant team fell strongly toward one of the polarised ends of the spectrum (although not at the extreme end) for each of the categories described. In other words, no team fell clearly in the middle of the spectrum for any category.

Even more significant was the observation that no participant company had teams on both ends of any of the category spectrums. This was strong indication that the companies themselves had clustered into one of two types when viewed through the lens of these major categories. What was different between these two groups of companies? The answer to that question would be a partial answer to the research question.

The first difference between these two groups of companies became quickly evident, the proverbial *elephant in the room*, namely that there was a fundamental difference in the organisational structure between the groups of companies, or more specifically, the understood expectations of the organisational structure. Evidence in the data appeared very early, showing that the organisational structure impacted the character and behaviour of cross-functional product teams (CFPTs) in certain important respects. The impact was so strong that, for a period of time, it appeared as if it might be the *only* major environmental factor. Earlier findings of this organisational structure impact in two conference papers (Fuller, 2019a, 2019b). Those findings have subsequently been expanded and matured and are presented in the remainder of this chapter.

As the data collection and analysis progressed, several major categories having nuances began to emerge that were not completely explained by the influence of the organisational structure context. As clarity of these nuances came into focus, it was seen that, in addition to the organisational structure, the product planning process was another, albeit related, influencing factor on CFPTs and their potential, and to some degree their motivation, to grok the product domain. When referring to the product planning process, I am not referring to any specific

methodology but rather to the nature and degree of the CFPT participation in the planning process, defined or assumed. Specifically, how (or whether) CFPTs participate in (and the degree to which they may feel responsible for) the core elements of product visioning, strategic planning, and road mapping.

These two influencing factors are not mutually independent. The context of the organisational structure sets the tone for the structure and methodology of the product planning process. However, the product planning process is not *determined* completely by the organisational structure and may have some additional and independent dynamics.

Separately and collectively, these two contexts within a software product company have a significant influencing factor on a team's collective motivation and potential to grok its product's domain. It is the grokking of the product domain that is the foundation for the team's ability to more deeply understand the product requirements and place those requirements into a richer context for execution than could otherwise be the case. The following figure illustrates the spectrum relationship between these two influencing factors, a diagram built upon as this chapter unfolds.

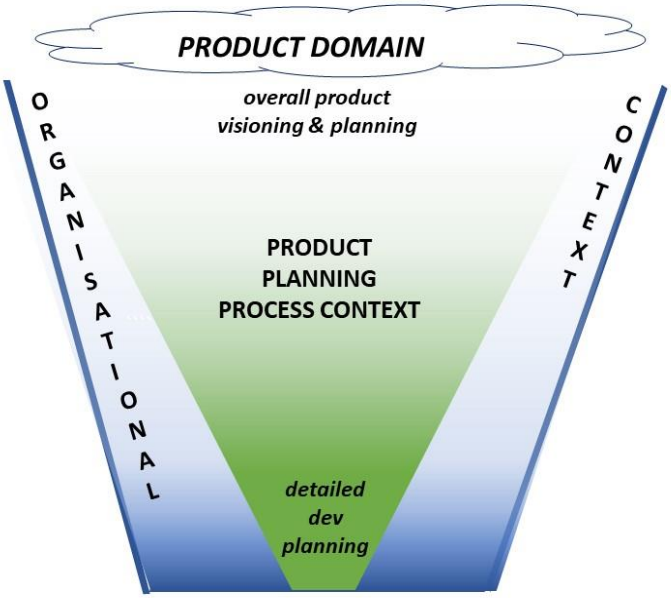


Figure 4-1. Influencing Factors

The next section of this chapter provides an in-depth examination of the first influencing factor, the Organisational Model. This is followed by Section 4.6, offering a similar in-depth examination of the second influencing factor, the Product Planning Process. The four major categories, along with several non-major categories, serve as valuable lenses through which to examine and describe each of these two factors; as a result, these categories are referred to frequently throughout the remainder of this chapter.

4.5 Factor 1 - Organisational Model Impact

To facilitate the discussion of the Organisation Model impact, two figurative and extreme team models were created to illustrate the influences of this model. Note that these are extreme

characterisations and few participant teams were so defined. However, all teams sampled had characteristics that put them closer to one of these extremes than the other.

Figure 4-2 illustrates one end of this team model spectrum where there are distinct functional groupings in the organisation (e.g., server-side engineering, UX design, security, mobile development, product management, testing, architecture, etc.) and where those groups manage the functional resources and contribute functional specialists to a CFPT. Within the teams, there was a strong recognition of, respect for, and sense of belonging to, the functional groups outside the team, i.e., the team member's *home* group. The *home* groups are illustrated in the figure by the very distinct colour, representing functional, boundaries within the team. There is an ambiguous boundary surrounding the team, composed of the aggregation of the different mandates of the participating functional groups. This extreme form of a team model operates as more of a work group. For the purposes of this discussion, this is labelled as an *Assembly of Experts* team (Figure 4-2).

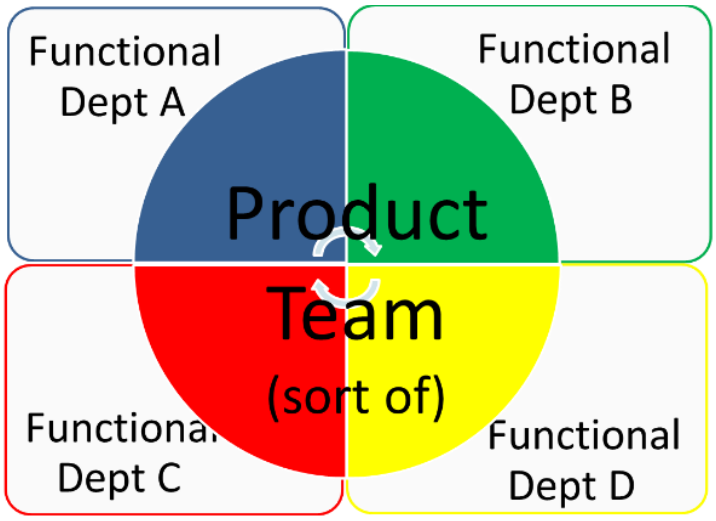


Figure 4-5. Assembly of Experts Team

Figure 4-3 below illustrates a contrasting model at the opposite end of this figurative team model spectrum. In this illustration, there are either no functional home groups for team members or else the functional groups exist but have softer, more flexible, mandates. Whatever functional distinctions may exist within the team are soft and often fade even further towards the centre as the team works together, committed to the same outcomes. The work individuals perform on the team is influenced by, but is not limited by, their particular functional competencies. There is an unambiguous team mandate, illustrated in this figure by the very clear distinct boundary surrounding the team. For discussion purposes here, this is labelled as a *True* Team model (Figure 4-3).

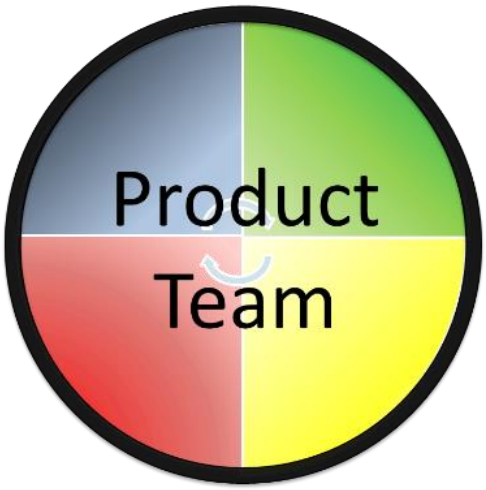


Figure 4-6. True Team

Of the six participant companies in the study that developed software products for a market, three had clearly defined and rigid functional organisational structures while the other three did not. As noted above in Section 4.4, all participant teams presented themselves with respect to the major categories in a manner consistent with the organisational model surrounding the team.

Examining teams that participated in this study against these illustrative team models, several categories from the data were useful to contrast these two models. The first four are forms of the major categories introduced above and as they appear in the context of the Organisational Model Impact.

1. ***functional deference***. In the context of the Organisational Model Impact, this is deference displayed between differing functions within the team (*internal* ***functional deference***).

2. ***primary affiliation***. In the context of the Organisational Model Impact, this refers to the location within the organisational structure where the individual team member feels his/her primary affiliation.
3. ***horizon of interest***. In the context of the Organisational Model Impact, this refers to how far into the future an individual on the team concerns him/herself as well as how far into the future does the team as a whole concern itself.
4. ***alignment with expectations***. This refers to how aligned are senior leadership expectations regarding the team's degree of product ownership with that actually felt and held by the team.
5. ***team cohesion***. The degree of unity of purpose within the team. This is heavily determined by the degree of internal ***functional deference*** that exists within the team combined with the individuals' sense of ***primary affiliation***.
6. ***product ownership***. The degree of ownership the team feels towards the development plans and the product future. This is the result of ***team cohesion*** and ***horizon of interest***.

I now discuss each of the four major categories in the context of this Organisational Model Impact on CFPTs' ability to ultimately grok the product domain.

4.5.1 *Functional Deference* (Internal)

Do members of a team contribute as if they are fully committed to the team and striving for the success of the team or do they contribute as if they are 'an expert in attendance' to provide specialist expertise in a well-defined role?

4.5.1.1 *Functional Deference* (Internal) - Assembly of Experts

When a functional organisational structure exists in the software product enterprise (e.g., separate groups and sometimes even sub-groups) for specific functions (e.g., server-side engineering, UX design, security, mobile development, product management, testing, architecture, etc.) with each group assigning individuals as required to form cross-functional product teams (see Figure 4-2), it was found that team members generally limited their team contributions to topics directly relating to his/her particular competencies. In doing so, they displayed marked deference to team members with functional competencies different from his/her own on topics outside their area of primary functional expertise (internal ***functional deference***).

Evidence of strong internal ***functional deference*** (and low ***team cohesion***) was often freely expressed verbally. As one participant software engineer put it in an interview when referring to the UX design team members:

> *"***I don't criticise their designs.** ***They're the experts, I trust them****".*
>
> *(Participant 15)*

Notes from observations of *Assembly of Experts* teams in action along with other quotes from interviews with team members on these teams were very consistent in this explicit display of internal ***functional deference***. This behaviour was shown very strongly and consistently throughout the study. Of the 14 participant teams that were in the three companies with a

distinct functional organisation, all 14 displayed strong ***functional deference*** within the teams and exhibited overall behaviour as described above as *Assembly of Experts* teams.

Although ***team cohesion*** was influenced by other factors as well (e.g. ***primary affiliation***), it ws noted that high levels of internal ***functional deference*** weakened the sense of cohesion in the team. In other words, it was more difficult for an *Assembly of Experts* team to feel and act truly united towards a common purpose.

Over time, the following coded quotes emerged from analysis of the interviews and observation sessions and contributed to the eventual creation of the theme of ***distinct boundaries*:**

"**I just do my job and they do theirs**"

"**They're the experts, I trust them**"

"**I think someone else is looking after that**"

"**I just do what I'm asked to do**"

There is very strong *us and them* language being used here by members on these types of teams, illustrating a strong distinction and separation between the functions (***distinct boundaries***) within the team, inhibiting a single sense of *team* from emerging. A single sense of team is a pre-requisite condition for a CFPT to develop a collective understanding of the product requirements and their context. Without it, there is no *collective* in which to position product requirements and their context nor is there a basis to reconcile assumptions. Reflecting the divisions represented by the functional organisational model outside the team, these members were conducting themselves as someone *playing a specific role* on the team in stark contrast to *fully being* on the

team. On these teams with strong internal ***functional deference***, when the term "us" was used, it was most often referring to team members with the same or similar functional role.

These ***distinct boundaries*** were also reflected in the body language within teams where it was observed that interactions amongst team members having the same, or similar, functional roles were often notably more open, relaxed, and exploratory while there were more formal or structured interactions between team members in differing functional roles.

4.5.1.2 *Functional Deference* (Internal) - True Teams

In contrast, within *True* teams, the low level, and often the absence, of internal ***functional deference*** presented itself strongly. Of the three companies without a strong functional organisational model, the teams in two of them displayed almost no evidence of ***distinct boundaries*** (little to no internal ***functional deference***) while the one team in the third company displayed so only very mildly, reflective of the very polite and passive corporate culture of that participant company. However, it was so mild in this case that it did not appear to erode ***team cohesion*** to any discernible degree.

When I did observe what might be taken as internal ***functional deference*** being displayed between functions in these teams, it turned out not to be ***functional deference*** at all, but rather very strong ***product ownership***. For example, these events might occur after a team member had fully and passionately expressed his/her opinion, followed by a brief qualifying statement, acknowledging expertise differences and showing respect for fellow team members yet not

letting those functional role differences prevent the individual from fully expressing what he/she felt was best for the product (***product ownership***). To illustrate, the following is an excerpt from researcher notes taken from an observation session of a team planning and design meeting:

> *"a developer gave a fairly long explanation of how she thought some product capability ought to be in terms of functionality, usability, and adaptability for future needs. Not only a long explanation, but a very impassioned one as well. She really cared. At the end of her 'speech', she declared, '**however, I'm not a designer'**". (team C, session 1)*

The participant in this quotation was showing respect for her fellow team members that were in a different functional role than her (in this case, UX design). Yet, caring more about the product than concern for any functional sensitivities that might (or in this case, did not) exist on the team, she ensured that her point of view did not go unexpressed.

Collaborative behaviours were observably higher in these *True* teams that demonstrated lower levels of internal ***functional deference***. Examples of collaborative behaviours were observed by the amount of assistance offered and provided between differing functions in the team as well as evident in the language used in team interactions. Higher collaborative behaviours were reflected in the language used by team members which included much more collective references, being more '*we*' oriented (referring to the team as a unit), and less likely to include connotations of '*them*' when referring to differing intra-team functions.

More collaborative behaviour also contributed to more balanced workloads. In daily stand-ups, for example, it was not uncommon to hear phases along the lines of *what can I do?, how can I help?,* or *perhaps we can work together on that?* from a team member that was idle or felt she/he had a lighter workload than others or simply felt they could make a positive contribution to a work item. In contrast, team members on *Assembly of Experts* teams with strong internal ***functional deference***, having finished his/her tasks and confirmed that no one was requesting anything specific from them, might well be playing table tennis while his/her fellow team members were finishing his/her tasks.

These *True* teams also had much less interaction with, rarely even making reference to, other organisational units in the company, illustrating a stronger sense of self-sufficiency and ***clarity of ownership***. This was a result of markedly less ***functional deference*** shown within the team and the ***team cohesion*** being notably higher, with individual team members placing the interests of the product and deep understanding of the requirements foremost and above any functional differences or tensions within the team. Often when those interactions outside of the team did occur, they happened as a matter of course and rarely became topics for team discussion.

Individuals on these true teams were ***broadening the lens*** to see a broader picture and expand his/her contextual understanding. In doing so, they softened, even eliminated, ***distinct boundaries*** between functions on the team. With weaker ***distinct boundaries*** in the team, they achieved stronger ***team cohesion***, greater ***product ownership***, and a sense that they were all part of the same team, sharing the same purpose, which allowed them to approach the requirements (near- and long-term) with the collective knowledge and entire expertise held by the team.

This internal ***functional deference*** is distinct from, yet influences, the other major categories that I will now discuss in the context of this organisational model factor - ***primary affiliation, horizon of interest,*** and ***alignment with expectations.***

The beginning of Section 4.5.1 asked whether members on a team contribute as being 'on the team' or whether they contribute as if they are 'an expert in attendance'? The answer is strongly influenced by the organisational model surrounding the team.

4.5.2 Primary Affiliation

What T-shirt are team members wearing?

4.5.2.1 *Primary Affiliation* - Assembly of Experts

On *Assembly of Experts* teams, the team member's sense of ***primary affiliation*** was stronger towards that individual's functional group than it was toward the product, the software product team, and the team mission. Simply put, an individual in this organisational model locates themself more by his/her functional competency and the organisational structure (i.e. his/her functional group) than they do by what team they are a member of. As a typical example, one participant in this organisational model introduced himself this way – "I'm a member of the security group in the engineering department, currently working with the Product S team" (security specialist, team S, company Z).

This is unsurprising since these cross-functional teams do not situate completely within a single functional grouping thus, where there is a strong functional organisational structure, there is a competing organisational identity for those team members, i.e. the function group vs the product team. I observed that, in circumstances where there could be conflict between the two, e.g. if the preferred approach to addressing a product need would be contrary to the standard practice of a functional group, certain individuals would most often feel safer favouring his/her obligation to his/her home functional group. To do otherwise would risk creating inter-departmental tension outside of the team and my observations indicated that functional connections were often stronger than team connections under these conditions. This ***psychological safety*** factor and its role in innovation and a team's sense of common purpose is discussed further in Chapter 5 – Discussion. A clear instance of this was observed in a product planning meeting where a debate occurred between a developer responsible for architecture and much of the rest of the team. The developer in question felt obligated to honour an architectural pattern favoured by the engineering department while other members on the team felt the pattern would not support the fullness of certain product requirements and likely make future product evolution efforts more difficult. While appreciating the arguments posed by others on the team, the developer/architect stood firm, reflecting a stronger ***primary affiliation*** to his functional group than to the product.

In this *Assembly of Experts* model, the enterprise has typically not defined these CFPTs in a manner where an individual can resolve the broader external ***distinct boundaries*** between different product functions represented on the team. In conversations I held had with executives in the participant companies, some indicated that it had not occurred to them that affiliation conflict might occur, expressing platitudinal statements along the lines of, "we're one big team

here". Others either felt or assumed that the functional grouping distinctions had been purposely established in the distant past for specific reasons and that primary affiliation in favour of the functional group was how it ought to be. The organisational metaphor and the implications for practice are discussed further in Chapter 5 - Discussion.

When product development work was underway, team members with a strong ***primary affiliation*** to his/her functional group tended to take a narrower, more specialist, viewpoint on product issues (***distinct boundaries***), reflecting the functional focus of his/her home group. This presented itself as individual team members, depending on his/her functional expertise, being much more concerned with *how*, *what*, or *why* a product was to be built (depending on his/her role) but rarely all three, showing a specialist framing of the work, e.g. "**I just do what I'm asked to do**" or "**I do my job and they do theirs**".

While the deference and affiliation dynamics in this *Assembly of Experts* model might be viewed as a lack of concern for the team's work product, no evidence for this emerged from the data. That is, there was no indication that a high degree of internal ***functional deference*** and an individual's ***primary affiliation*** favouring the functional group over the team and its product reflected any diminished concern for the quality of the work performed. On this type of team, the concern for technical quality, however defined, took a degree of precedence over a broader concern for product quality. Whereas, on teams closer to the *True Team* model, *product* quality concerns tended to be given somewhat more importance than did that for technical quality, which would often be viewed as a matter that could be addressed at a later time. It's important to

note, however, that no team was observed to be on either extreme of this team model spectrum, thus all teams held some concern for *both* product and technical quality.

In other words, what might appear as a lack of concern for the team's work product by individuals on *Assembly of Experts* teams was more an acute awareness of his/her ***primary affiliation*** combined with a shorter ***horizon of interest***. ***Horizon of interest*** is discussed in the Section 4.5.3.

4.5.2.2 *Primary Affiliation* - True Teams

In contrast, the primary affiliation of members on *True* teams was clear. It was the team and its mission. ***Broadening the lens***, individuals looked beyond just 'what they were asked to do' and connected more to the team as a whole and, therefore, to the product and the future success of the product (***product ownership***). When these team members and these types of teams were being interviewed in this study, they would very often lead the introduction of themselves as being on his/her particular team (or working on a particular product) and secondarily (and not even always) identified what the particular competencies were that they brought to the team. As noted earlier, this contrasted with *Assembly of Experts* team members that would sometimes not even mention what team they were on.

So, what T-shirt are team members wearing? Figuratively speaking, *Assembly of Experts* team members wear functional T-shirts, perhaps with some indication of what team they are on at the moment. This would be akin to an imaginary scenario of the goaltender for a hockey team wearing a jersey from the *ABC Goaltender Supply Company* with a patch saying *today playing*

for …. In contrast, on a *True* hockey team, all players, regardless of position, are wearing the *team* jersey with forwards sometimes defending and defensemen sometimes scoring goals.

4.5.3 Horizon of Interest

How far into the future does the team as well as the individual team members concern themselves?

4.5.3.1 *Horizon of Interest* - Assembly of Experts

For *Assembly of Experts* teams, there was a more limited ***horizon of interest*** held by individual team members. With a team member's sense of ***primary affiliation*** being stronger towards his/her functional group than towards the team, he/she had a sense of having a *home* to go back to and so his/her participation on a team tended to be viewed by them and the rest of the team as being one of an *assignment* which brought with it a diminished feeling of being an integral part of the team. Therefore, they tended to have an expectation of team reassignment at some point in the foreseeable future. This increased expectation of mobility reduced an individual's intrinsic connection to the product team (lowering ***team cohesion***) and, therefore, reduced that individual's interest in the full product itself and the longer-term product roadmap. When a team member expected to be reassigned, even if it were not expected soon, he/she was likely to hold a certain tentativeness to his/her commitment to both the product and the product team and therefore less likely to behave as if they are fully committed. As quoted earlier, *"*I'm on this team… *for now"* (Participant 20).

Even detailed development decisions were affected. It was observed during development discussions and decision-making, less consideration was given to elements further into the future on the product roadmap (shorter individual ***horizon of interest***) due to the *specialist* framing these team members have on the work in front of them.

As it was with one of the participant companies, this condition of tentativeness can be created or exacerbated by Human Resources practices that encourage the systematic and regular internal movement of employees from one job or team to another within the company, based on the belief that it supports a shortened ramp-up for new employees, fosters alignment, strengthens engagement, and enhances career development. Whether these policies actually achieve those objectives is unclear but, in the case observed, it was clear that the practice came at a cost of shortening the team members' ***horizon of interest*** due to those individuals having more uncertainty regarding how long they might be on the team. As one engineering manager expressed while lamenting the consequences of this situation: "I don't know how a true *team* can emerge in this condition" (Participant 10), referring to the shorter ***horizon of interest*** that individual team members showed under those conditions as well as to the increased on-boarding effort required by the entire team as a result of greater churn in team membership.

Thus, these *Assembly of Experts* teams were generally much less inclined to play the long game (less ***product ownership***). Instead, the focus was aimed more narrowly at the current and near-term plans, "**we just do what the plan** (or **Product Management**) **says**". With greater internal ***functional deference***, and ***primary affiliation*** favouring the functional group over the team, and

with a tentative ***horizon of interest***, a complete commitment to the full product and the product's long-term development roadmap was inhibited.

4.5.3.2 *Horizon of Interest* - True Teams

In contrast, *True* teams exhibited a more complete sense of collective ***product ownership*** for their product and a commitment to the product's longer-term success (longer *team* ***horizon of interest***), engaging in more comprehensive discussions regarding the product requirements (***striving to grok)***. The motivation of ***striving to grok*** resulted in the team ***broadening the lens*** in an attempt to achieve a deeper understanding of the details (the *what*) of the requirements specifications and to achieve also some understanding of the context (the *why*) of the requirements to the best of its ability.

Regardless of functional competency and role, members of teams in the *True* team model tended to develop a more holistic and collective perspective of what they were doing (***striving to grok***) and what the long-term plan was. All team members tended to care about what, why, and how of the product, reflecting a stronger sense of (collective) ***product ownership***, ownership of the product beyond simply his/her individual contribution to it. As Participant 03 stated in a very matter-of-fact tone during an interview, "**it's <u>our</u> product**".

So, how far into the future do individual team members and the team itself concern themselves? *Assembly of Experts* teams and team members looked only as far as necessary to deal with the work facing them at the moment (iteration plan, release cycle, critical initiative, etc.) while *True*

teams and members on those teams look as far ahead *as they were able* in order to provide as rich a context as possible for the execution of their current plan and formulation of future plans.

4.5.4 Alignment with Expectations

How aligned is the senior leadership's expectations of the team's mandate and empowerment with that which the team holds and understands?

4.5.4.1 *Alignment with Expectations* - Assembly of Experts

As noted above, individuals on *Assembly of Experts* teams tended to have a lower sense of total ***product ownership.*** That is, they generally showed less investment in the overall success of the product, **"I just do what I'm asked to do***"* and, therefore, appeared less invested in both the team itself and in the long-term plan for the product. All 14 teams in companies with a distinct functional organisational model surrounding the CFPT acted in ways that demonstrated a strong focus on the *near-term* roadmap, but with very little attention given to the *long-term* plans and vision for the product.

With lower ***team cohesion***, shorter ***horizon of interest***, and less ***product ownership***, *Assembly of Experts* teams were much less likely to take collective responsibility for the product's success or failure. With a near-term *(narrow lens)* on the product plan, they would take ownership for the work they performed, but not necessarily for the overall product result. With less collective ***product ownership***, individuals were more likely to passively criticise rather than attempt to

understand and possibly influence the vision and broader plan. For example, the following quotes are taken from observation notes of a planning session of team M:

> ***"It isn't the strategy I would have put together, but it's not my call"***
>
> ***"I don't know if they understand the market very well"***

They had strongly held views that indicated that they cared somewhat about the product but the distinct *I and they* language demonstrated a limit to that caring by recognising ***distinct boundaries*** and ***functional deference*** resulting in a sense of having little control or influence.

This passivity-like behaviour appeared also in the form of the team being less likely to advocate for what they feel might be in the best interest of the team and product. These teams tended to be somewhat less concerned about having limited scope due to organisational or expertise limitations and, therefore, were less likely to advocate for the optimum level of team resources to meet the roadmap expectations. Nor were they likely to even try to obtain specific functional competencies in order to assume a broader scope or plan.

This is an example of the aforementioned point about teams in this organisational model that have less ***team cohesion*** and greater ***functional deference*** being less likely to have collective ownership for the definition of the work output (lower ***product ownership***). In other words, they are more willing to defer to the decisioning of his/her respective functional home group regarding resourcing, plans, process, etc. than they are to strongly and collectively advocate for what is in the best overall interest of the product.

There were differences in the awareness and understanding of this phenomenon held by senior leadership in these participant companies. For 8 of the 14 teams observed in this category, senior management admitted to being generally aware of this (low) level of collective ***product ownership*** and desired to change it but admitted to being unclear as to its cause and, therefore, what corrective steps to take. For the remaining 6 teams, there was a significant disconnect between the low sense of collective ***product ownership*** and what senior leadership thought was actually the case. In other words, these leaders held a belief that the teams were much more cohesive and more fully and collectively committed to the product and its long-term roadmap than the teams actually were.

For all 14 teams, senior leadership generally felt that the CFPTs had been assembled with representatives from various functional groups as an expedient way to integrate expertise and to operate across the functions in the organisation to address the growing complexity and the need for innovation in a highly competitive product world. Factors relating to creativity and innovation are discussed in Chapter 5 – Discussion. However, contrary to senior leadership intention and expectations, these teams behaved as work groups with a short **team *horizon of interest***, focussing almost exclusively on the near-term work in front of them ("**I just do what I'm asked to do**"). In further discussions with these senior leaders about these observations, they sometimes reacted with surprise, stating that they were unaware that these organisational choices would likely have the impact on individuals and teams as they were having. In one case, and based on my professional experience this is possibly more common than this research might suggest, the response was a knowing nod, with the leader adding that the design of the

organisational structure and solution to its problems were beyond that leader's control. This point is discussed further in the Implications for Practice section of Chapter 5.

4.5.4.2 *Alignment with Expectations* - True Teams

In contrast, all 4 teams *without* a strong functional organisation surrounding them (Figure 4-3) regularly positioned their actions in the context of the longer-term plan and vision of the product (***broadening the lens***). The data showed no evidence of misalignment between the teams' actions and understanding of their mandate with that of the expectations and understanding held by the senior leadership.

In summary, I found that the more a team can adjust its lens, the more it is able to align its definition of success with what the company expects of the team (***alignment with expectations***).

4.5.5 Summary

Collectively, these four major categories (***functional deference***, ***primary affiliation***, ***horizon of interest***, and ***alignment of expectations***) illustrate that a CFPT's position and potential progression along the continuum with an *Assembly of Experts* group on one end of the spectrum, focussing mostly just on the work in front of them, and a *True* empowered, cohesive team on the other end of the spectrum, committed to the long-term success of the product, is heavily influenced by the broader functional organisational structure surrounding the team. It is influenced also by the nature of the organisational structure, not simply the existence of it. I.e. if a broader functional structure does exist, how strong those functional group distinctions are will strongly influence its effect. Figure 4-4 below graphically illustrates the limiting pressure

(downward pointing arrows) on each of the major categories experienced by teams with a strong functional organisational model (*Assembly of Experts*).

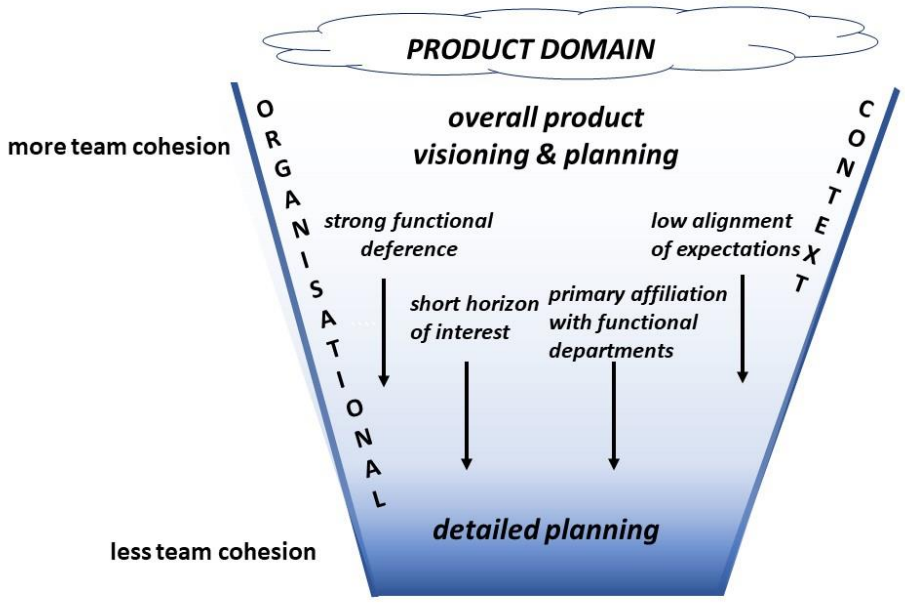


Figure 4-7. Effect of Major Categories - Strong Functional Organisational Model

Figure 4-5 below illustrates the enablement (upward pointing arrows) in each of the categories afforded to teams surrounded by a weak or non-existent functional organisational model (*True* Teams). This potential is exercised by the teams ***broadening the lens*** as much as they are able as they strive to move higher in the column and achieve a broader context.

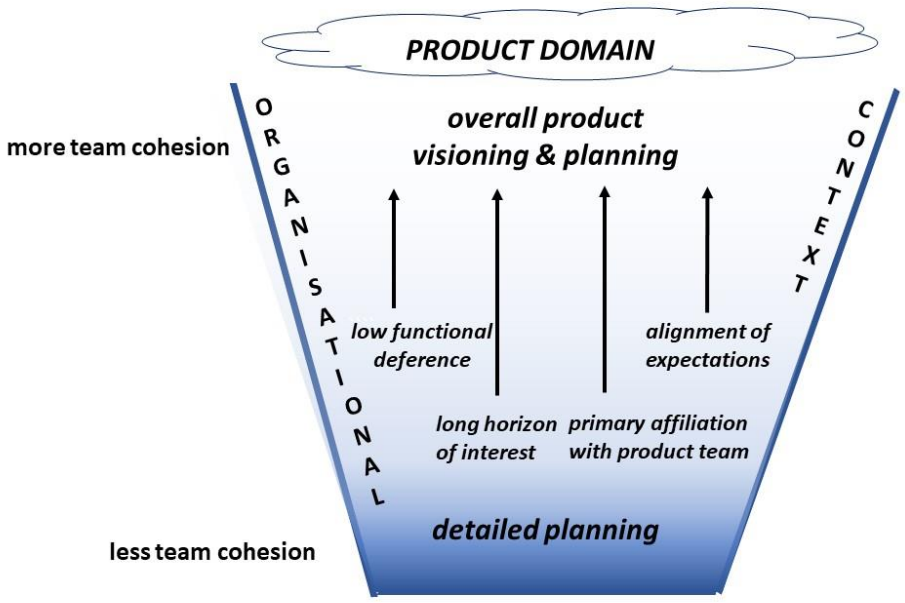


Figure 4-8. Effect of Major Categories - Weak or Non-existent Functional Organisational Model

4.6 Factor 2 - Product Planning Process Impact

The process for product planning also presented strongly as having a significant influence on cross-functional product teams (CFPTs), specifically on their motivation and potential to exercise their ability to collectively grok the product domain. As illustrated in Section 4.5 regarding the Organisational Model Impact, the effects on these teams can take many forms. My primary interest in the Product Planning Process factor was with how (if at all) CFPTs participated in the core elements of product visioning, strategic planning, and road mapping and also with how much the teams appeared to own any of the process elements that they did participate in.

As teams were examined with respect to the impact the Product Planning Process factor had, four major categories appeared most strongly – external ***functional deference***, ***primary affiliation***, ***horizon of interest***, and ***context of team definition of success.***

1. ***functional deference***. In the context of the Product Planning Process Impact, this is deference displayed between teams of differing functions in the product planning process (*external* ***functional deference***).
2. ***primary affiliation***. In the context of the Product Planning Process Impact, ***primary affiliation*** refers to where it is that the team places its primary focus when executing on the product plans – from the near-term plan alone on to the entire product roadmap.
3. ***horizon of interest***. Tightly tied to ***primary affiliation***, in the context of the Product Planning Process Impact, ***horizon of interest*** refers to how far into the future is the concern held by the team with respect to the product plans.
4. ***definition of success.*** As a flavour of ***alignment with expectations*** in the Organisational Model context, ***definition of success*** refers to the success criteria the team uses to determine its own success. Specifically, the analysis examined the context that the team has available to it, if any, to decide what criteria it will use and how it assessed its own performance against those criteria.

In the next three sections, these categories are discussed in the context of this Product Planning Process Impact and its effect on CFPTs' ability to ultimately grok the product domain. The next section discusses the external ***functional deference*** category, Section 4.6.2 discusses team ***primary affiliation*** and team ***horizon of interest*** together as they are closely aligned, and finally Section 4.6.2 discusses the team's ***definition of success*** and the context for it.

4.6.1 *Functional Deference* (External)

Do cross-functional product teams contribute to the product visioning and planning with a sense of complete or partial ownership of the product plan and the planning process or do they show deference to other functional groups and contribute in a manner as if they are 'a collection of experts on call'?

I found that *Assembly of Experts* teams (Figure 4-2) showed more ***functional deference*** to the other functional groups (i.e. respecting ***distinct boundaries*** in the planning process model). Consequently, they tended to focus strongly on the work in front of them but were less likely to act as a unit for any product cause beyond their work, respecting accountabilities of other involved groups. This collective character of the team presents itself in a similar manner as does the individual *specialist* on an *Assembly of Experts* team.

Conversely, the closer a team was to the *True* team model (Figure 4-3), the more it held the interests of the product foremost and advocated for the product interest to teams with other functional responsibilities, e.g. service delivery, marketing, product management, etc.). The more strongly they did this, the less *external* ***functional deference*** they were displaying. In other words, the more they were softening the ***distinct boundaries*** between functional groups involved in product planning and development.

4.6.2 Primary Affiliation & Horizon of Interest – Product or Plan?

Are the cross-functional product teams focussed entirely on the near-term plan or do they attempt to put the near-term plan into the context of the longer-term plan and vision for the product?

Although there are important nuanced differences between the major categories of the ***primary affiliation*** of the team and its ***horizon of interest*** in this context of the Product Planning Process, there are also strong interplays between them, so these are discussed together in this section.

Primary affiliation in this setting is primarily about the disposition of the team towards being more solely focussed on the near-term development plan versus being *also* very concerned with the product roadmap more broadly. Team ***horizon of interest*** in the product planning process setting is about how far into the future on the product roadmap a team looks for the purpose of creating context when making development decisions in the near term. The former is an indication of ownership ***(product ownership)***, the latter one of creating context ***(striving to grok)***.

Teams that were 'handed a plan', with little or no involvement in its creation, tended to focus more on the *execution* of the plan, reading that plan more literally and putting less effort into deeply understanding the *content* of the plan (**"we just do what the plan says"**). With little context of the plan handed to them in relation to the longer-term product roadmap, they had a narrower lens on the plan details and simply executed the plan handed to them as best they could. Success criteria for these teams tended to be related to the successful execution of near-term

plans i.e., iteration success, meeting a release deadline, solving a critical defect, etc. In other words, these teams' ***primary affiliation*** in the planning context was more towards the near-term plan (however and why it was that that plan may have been created) than it was towards the product overall and the long-term product roadmap. The following is an extract from the researcher reflective notes taken after a team observation session which expresses this view by contrasting the low degree of socialised product vision within the team against another team previously observed:

> *"… and to repeat one of my reflections from the C team observation session, 'The vision develops, evolves, and socialises in the team through constant discussion with others within and outside the team, NOT via a form of specification to define, ground, and document. This constant discussion includes continually (or easily) revisiting ground previously covered (pawing).'"*
>
> *- (notes from Team X observation session)*

The team being observed here had just been handed a plan and, although there were resources available to the whole team and to individual team members to answer questions, etc., they were still left with a low level of affiliation to the product and its overall plan and were very focussed on executing just what the current plan specified, making almost no attempt to create greater context for that plan.

Conversely, teams that were more involved in the plan creation were further along on the spectrum towards the team having a ***primary affiliation*** with the product ("**it's our product**").

By ***broadening the lens*** and ***striving to grok***, the team used their sense of ***product ownership*** to cultivate a deeper, collective, contextual understanding of the product plan and product domain. With this deeper understanding of the product plan and domain, for any requirements that *did* get simply handed to the team, the team had a basis upon which to ask deeper and broader questions in their effort to create understanding and context.

Where a team is situated along the spectrum of ***primary affiliation*** in the Product Planning Process setting heavily influences the team's default ***horizon of interest***. A ***primary affiliation*** with the product and long-term plan tends to result in a longer ***horizon of interest***. The more a team is "spoon-fed" development plans without meaningful involvement in the planning process, the narrower their lens is, i.e. the more limited their default context is for meeting the requirements. In other words, a team with a ***primary affiliation*** with the near-term plan will, by default, have a shorter ***horizon of interest***. However, even with a ***primary affiliation*** to the short-term plan, a team may take steps to create as long a ***horizon of interest*** as they are able for the purpose of creating a richer context within which to make near-term development decisions. Since this longer ***horizon of interest*** doesn't occur naturally with a near-term ***primary affiliation***, it takes intention and concerted effort on the part of the team within the constraints created by the Product Planning Process.

Teams having some sense of ownership for (or, at least, interest in) the plan *content* and the product overall will ***broaden the lens*** to provide context for current *and future* requirements (***striving to grok***) before ***narrowing the lens*** again to make detailed development decisions.

4.6.3 Context of Teams' *Definition of Success*

Why do teams broaden the lens and strive to grok at all?

The data showed that the Product Planning Process Factor constrains or permits teams to ***broaden the lens*** in their efforts in ***striving to grok*** as much as they are capable. The question I then asked myself was, "why do some teams strive to own as much as they are capable of rather than simply being content with owning less, if that's all the company expects them to?"

As it was with the Organisational Model Factor, I also observed that in the Product Planning Process realm the degree to which a team might be ***broadening the lens*** to understand a bigger picture (and then narrow it again to focus on the work to be done) was reflected in the verbal language used by the teams. The broader the team's planning scope was, the more I observed conversations indicating a deep understanding of, or at least references to, product needs. These indicated that the team was taking a product domain perspective and were also considering product/market opportunities, competitive factors, etc. (***broadening the lens***). Then, with that broader perspective as context, the team discussed more granular detail in the requirements as necessary (***narrowing the lens***) for development. The more a team was able to ***broaden the lens***, the more they were able to be curious and open-minded, attempting to explore more (***striving to grok***) with the view "**it's our product**". The broader the perspective held by the team, the more the *detailed* planning discussions tended to occur in stride rather than being the main focus of discussion. To illustrate, an excerpt from researcher notes of a planning and design session:

"striking contrast in overall discussion content between this planning session and the one I attended recently with Team Q. While this meeting 'did' result in an agreed-upon iteration plan with sized stories, etc. those details emerged almost as a natural by-product of the team's discussion on broader product and market matters. The iteration plan details weren't ignored, but there was very little 'focus' on them. Team Q's meeting (which was very common and typical of 'by the book' agile teams) consisted almost solely of discussion of stories, sizing, etc."

- (team R, session 2)

As noted in this memo excerpt, discussions held by teams with a limited ability to ***broaden the lens*** tended to reference domain considerations much less. This is because they have less domain knowledge to consider. The conversations in these situations were almost entirely about internal entities and artefacts such as stories, requirement specifications, acceptance criteria, other functions/teams, processes, etc., and very little, if any, about the product and product domain.

What a team sensed as its permitted boundaries was often reflected in what completed work the development team took pride in and celebrated, e.g. a successful iteration, meeting a release deadline, getting an important feature on the release train, or the pride of being a team that created a product that is successful in the market. Simply put, these were the boundaries set by the team for the space within which it would define its own success. Normally, there was a base-level context for the definition of team success as defined (explicitly or implicitly) by the

organisation. Some teams observed were using those externally-set measures of success as the starting point. The effort the team expended to expand team success measures beyond that starting point (to the extent permitted by the influencing factors) appeared to be efforts toward creating identity and greater meaning via collective sensemaking. This point is discussed further when the metaphorical adjustment mechanism, ***broadening the lens***, is examined in detail in Chapter 5 - Discussion.

I observed that *all* teams had a natural propensity to want to own *something*. ***Striving to grok*** is evidence of a team ***wanting to own***. This appeared to be true whether the ***primary affiliation*** of a team in the product planning context was towards the near-term plan and the execution of that plan or more towards the plan *context* and product overall. It appeared also to hold whether the team had a short or a long ***horizon of interest***. Thus, even narrowly focussing on the current iteration's tasks and taking pride in a successful iteration is still an example of ***wanting to own***. In progressing team ownership more broadly towards the team's potential, a team is ***broadening the lens*** enough, but only enough, on the product development plan to match what they are able or allowed to see. This affords them the richest context of the requirements allowable, given the team's ability and the constraints of the organisational structure and product planning process settings.

4.6.4 Summary

In summary, the Product Planning Process is a significant factor in determining the potential for cross-functional product teams to grok the product domain as much as they are capable of in

order to create a context for a richer understanding of the requirements. This setting can offer a team free reign, it can limit it mildly with a high ceiling, or it can suffocate it with a low ceiling. Figure 4-6 below graphically summarises the directional pressure the major categories have on the teams in an environment where they have little to no meaningful participation in the product planning process.

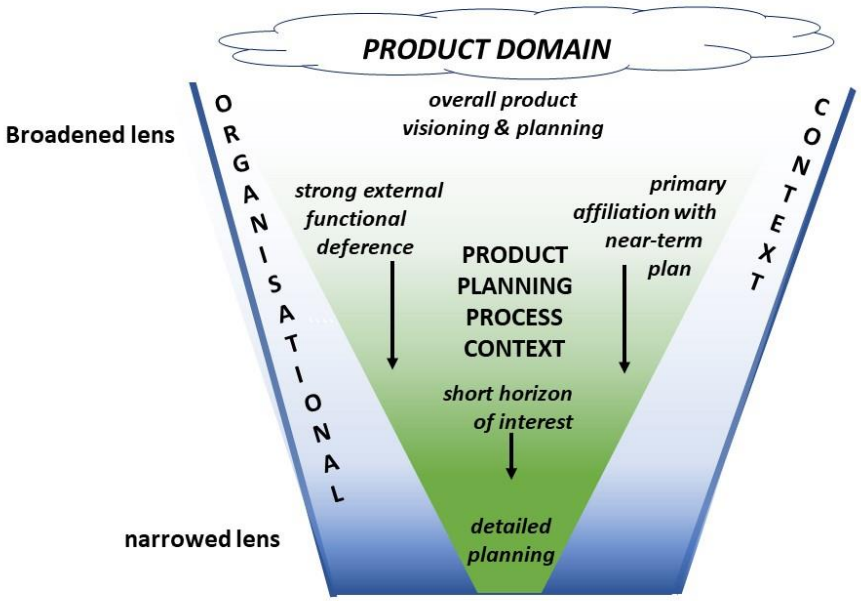


Figure 4-9. Effect of Major Categories – Minimal Product Planning participation

Figure 4-7 below graphically depicts the directional pressure of those same categories for teams that have meaningful involvement in the product planning process and are ***broadening the lens*** as they push upward in the continuum in each of the categories.

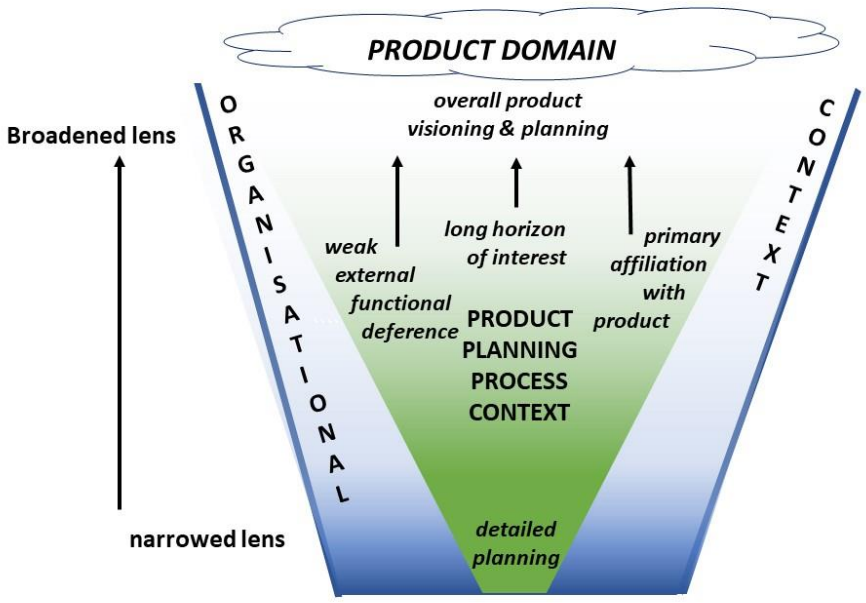


Figure 4-10. Effect of Major Categories – Meaningful Product Planning participation

4.7 Conclusion

These results offer two answers to the question of what factors support or impeded cross-function software product teams (CFPTs) in collectively achieving a deep understanding of the product domain for the purposes of achieving a deeper and collective understanding of the context of the product requirements. The two influencing factors identified are the Organisational Model and the Product Planning Process.

Presenting the results in this chapter, two important themes are referenced that emerged from the interpretation of the data and that persist throughout the analysis. These are the condition of ***distinct boundaries*** (referring to a spectrum of the hardness of boundaries in both of the

influencing factors) and the action of ***broadening the lens*** (with the complementary action of ***narrowing the lens***) to adjust what can be seen and how clearly. These themes provide the foundation for the emergent theory of ***Blurring Boundaries*** which is introduced and positioned in Chapter 5 - Discussion. I describe these two themes and the emergent theory next.

4.7.1 Distinct Boundaries

The spectrums upon which an individual or team finds themselves and, to some degree, can navigate along with all the major categories – ***functional deference***, ***primary affiliation***, ***horizon of interest***, and ***alignment with expectations*** – all have embedded within them a notion of *boundaries.* Whether those boundaries exist and, if they do, their clarity and hardness. I observed that some organisations purposefully had no boundaries in one or both of the influencing factors. When boundaries did exist, some were observed that were very hard and distinct whereas others had boundaries that were flexible, sometimes even somewhat ambiguous. Whether the boundaries were distinct or ambiguous, organisations were seen defining boundaries as being very rigid or as having a certain degree of softness and adjustability, depending on the circumstances and pressures placed upon them.

Boundaries represent an important topic. Boundaries define the limits of how a team identifies itself and how the team defines what it owns and what work output it uses as success criteria. Specifically relating to the research question, the data showed that harder and more distinct boundaries are barriers for the team to collectively grok and therefore are, arguably, contraindicated. This contraindication topic is discussed in detail in the next chapter as the

emergent theory of ***Blurring Boundaries*** is explored alongside prevalent organisational and process metaphors within software product companies today.

4.7.2 Broadening the Lens

Frequent references have been made in this chapter to the figurative action of ***broadening (***and ***narrowing) the lens*** as an explanatory metaphor to describe the mechanism that was observed being used by individuals and CFPTs to navigate along the spectrums described, to adjust the scope and clarity of what they saw and what they could then focus on. The breadth of a lens affects what they saw within the team and between teams (organisational realm), how they saw and understood the plan (product planning realm), and how deeply they understood the product domain.

Broadening the Lens illuminates a broader picture, with less detail, exposing control boundaries, relationships, and patterns that cannot be seen when the lens is more focussed. Seeing a bigger picture provided the opportunity for the team to gain a broader understanding and, with that broader understanding, was then able to more purposefully and knowledgeably re-focus (***narrowing the lens***) to do the work. ***Narrowing the lens*** tightens the scope, reduces the context, and brings detail into focus.

Although ***broadening the lens*** is a figurative action, not all CFPTs could simply broaden its lens as much as they might wish. The data showed that, within whatever hard structural limitations that may exist (organisational or process dictates) plus other softer constraints, a team attempts to define and honour ownable boundaries. In doing so, both individuals and teams would ***colour***

within the lines they are given, allowed to be in, or are able to create (***want to own***). In other words, a team is unlikely to be ***broadening the lens*** beyond its permitted boundaries. To do so would create organisational tension or conflict that the team is unlikely to be able to reconcile. This is discussed further in the next chapter.

To the degree to which a team is able and permitted, ***broadening the lens***, as a metaphorical mechanism, allows teams to become more informed and, therefore, make more knowledgeable decisions. Although, outside the scope of this research, a broader context for the product development held by the team is likely to be a factor in the amount and severity of avoidable errors, accidental complexity, and technical debt. It also provides a basis for team communications and near-term product design decisions to be more closely aligned with long-term plans, product vision, and domain realities.

This mechanism also greatly facilitates the team's collective capability to explore further, to be more creative, and to innovate more, a point discussed further in the next chapter.

4.7.3 Summary

The purpose of this study was to identify and examine factors that impact CFPTs' ability to grok the product domain, thereby creating a deep, collective context within which to understand the product requirements.

The findings presented in this chapter strongly show that both the organisational structure surrounding the CFPTs and the product planning process are factors that can either impede or support CFPTs to become cohesive, to develop a strong sense of ownership of the product and the roadmap, and to collectively strive towards deeply and collectively understanding the product domain. This analysis of the data gathered uses major categories as distinct lens through which to examine these two factors and their impact on CFPTs.

The examination also identified two recurring themes in the data that lead to the emerging theory of ***Blurring Boundaries***. This theory is described, positioned, and discussed further in the next chapter.

Chapter 5: Discussion

5.1 Introduction

The primary objective for this study was to identify at least some of the factors that contributed to the observation that some software product development teams grokked the product domain more deeply than others. Chapter 2 positions the topic of this collective empathic understanding within an overview of the related extant literature in three areas relating to the research question - requirements engineering, collective sensemaking, and design thinking. It demonstrates a gap in the knowledge base and established that, while the research is aimed at the requirements engineering (RE) sub-discipline of software engineering, the research question is not addressed by any of the disciplines surveyed in the initial literature review.

The literature review sets the broad direction for this study and contributes to the theoretical sensitivity that helped refine the research question during the study and acted as a backstop throughout the research. Informed by the literature review, this chapter summarises the findings as they relate to the research question and then proceeds to discuss the significance of the findings in the context of further theoretically sampled extant literature.

The discussion in this chapter synthesises key elements of the findings into a substantive theory that contributes to the explanation of a rephrased research question which is why cross-functional product teams (CFPTs) in some companies generally have greater success at developing a deep, collective understanding of the product domain than do teams in certain other

companies. This discussion of the substantive grounded theory makes liberal references to the findings in Chapter 4 to demonstrate that this theory is grounded in the data and analysis from this study.

The substantive grounded theory of ***Blurring Boundaries*** presents a new understanding of the social and process elements of software product development and, specifically, software product requirements engineering. In addition to describing the theory and the main contribution it makes, support for the theory is provided by illustrating alignment of the findings with 1) the motivation, satisfaction, and effectiveness within software product teams, 2) support for the previously identified importance of human factors in requirements engineering whilst offering increased clarity of certain aspects of those factors, and 3) arguments for the contraindication of the *as-machine* metaphor (Morgan, 2006) commonly found in software product companies in the design of their organisational structures and their product planning processes.

In the rest of this chapter, Section 5.2 summarises the research findings described in Chapter 4, focussing on the research question. Section 5.3 offers a detailed discussion and positioning of those findings, providing the foundation for the theory of ***Blurring Boundaries*** which is presented in Section 5.4. Finally, Section 5.5 discusses the direct implications of this theory for industrial practice that emerges from further framing of the research results.

5.2 Summary of Findings – to the RQ and beyond

The research question (RQ) was "what factors support or impede cross-functional software product teams (CFPTs) in collectively achieving a deep understanding of the environment for which their product is intended?", a question that has its roots in observations made during my extensive industrial experience. Chapter 4 presents the findings relevant to this question that derived from the continual analysis throughout the study performed on the data gathered from observations of CFPTs and individual interviews with team members of those teams that developed software as products. The research findings provide at least a partial answer to the research question, specifically identifying two major factors that influence the behaviour of cross-functional software product development teams, both at the level of a collective unit and as well at the individual level of members on those teams. These two factors are 1) the organisational structure surrounding the CFPTs and, 2) the product planning process. The influences of these two factors impact the potential available to the teams to develop a deeper, collective grokking of the product, the product plans, and the product domain towards achieving a deeper, collective understanding of the context of the product requirements.

The analysis indicates that if a team is cohesive, whatever context it obtains tends to be more *collective* than if the team is not cohesive (Organisational Model factor). There is also a direct relationship between how much of the product visioning & planning the team owns, or is at least heavily involved with, and how *comprehensive* the team's context of the product plan is (Product Planning Process factor).

Together, the *collectiveness* and *comprehensiveness* of the team's contextual understanding of the product plan allows or dampens the team's motivation and effectiveness in its attempts to collectively grok the product domain for the purpose of creating a more accurate and comprehensive understanding of the context of the product requirements. It is to this that this inquiry aimed to offer a deeper understanding.

The following three sections comment further on each of the two factors and then identify some common elements in the research data behind them.

5.2.1 The Organisational Model factor

"Individuals and interactions over processes and tools"

(Agile Alliance, 2001)

The first influencing factor identified in response to the research question was the organisational model within which the cross-functional product teams (CFPTs) are situated. The data showed that if functional distinctions exist in the organisation model outside the team structure, the character of the CFPTs is affected. More specifically, the data showed that the stronger the functional distinctions surrounding the team, the less ***team cohesion*** there tends to be, and the shorter the ***horizon of interest*** is likely to be for both the team as a whole as well as for the individual team members. ***Team cohesion***, in this context, is an inverse function of the degree of affiliation ambiguity an individual team member feels and, therefore, how much tentativeness

that individual has regarding his/her team membership. Distinct from, yet closely related to, ***team cohesion***, ***horizon of interest*** in the Organisational Model context refers to how far into the future the individual team members as well as the team as a collective see themselves being committed to the team and to the product. If there are no functional distinctions surrounding the CFPT, or if those distinctions are soft, the findings indicate there is more team cohesion, i.e. stronger individual team member affiliation both to the team and to the product. There will also tend to be a longer ***horizon of interest*** held by both the individual team member and the team as a collective unit regarding the team's engagement with the product's development.

Figure 5-1 illustrates a team with strong functional organisation surrounding it. This condition creates affiliation conflict for individuals, inhibits team cohesion, and dampens the ***horizon of interest*** for both the team and the team members, causing the team to focus mainly on the details of the near-term development plan and execution.

"there was so much to grok, so little to grok from"

(Heinlein, 1961, p. 19)

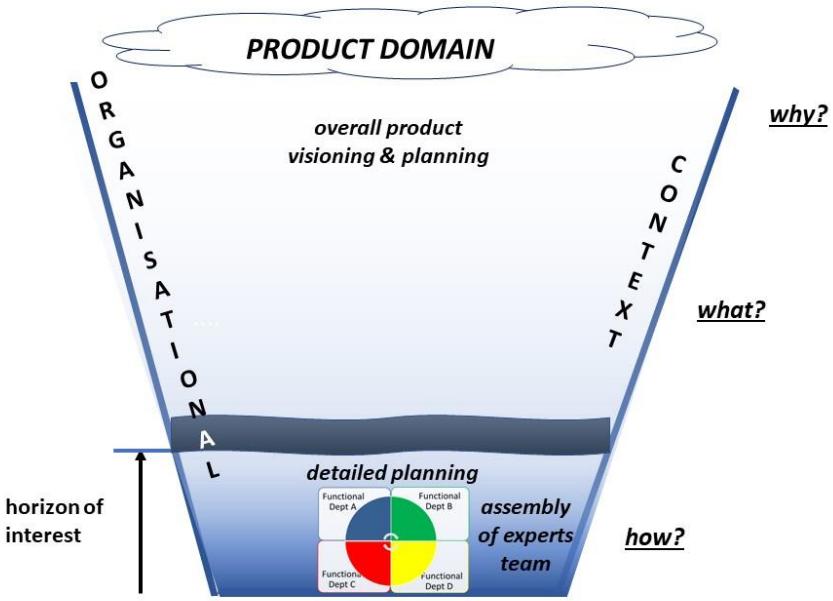


Figure 5-1. Short Horizon of Interest

Figure 5-2 below illustrates a condition where there is no, or at least soft, functional organisation surrounding the team. This allows the team to have greater team cohesion and it elevates, or even removes, the ***horizon of interest*** ceiling imposed by the organisational model. There also remains the impact of the product planning process to consider which is discussed next.

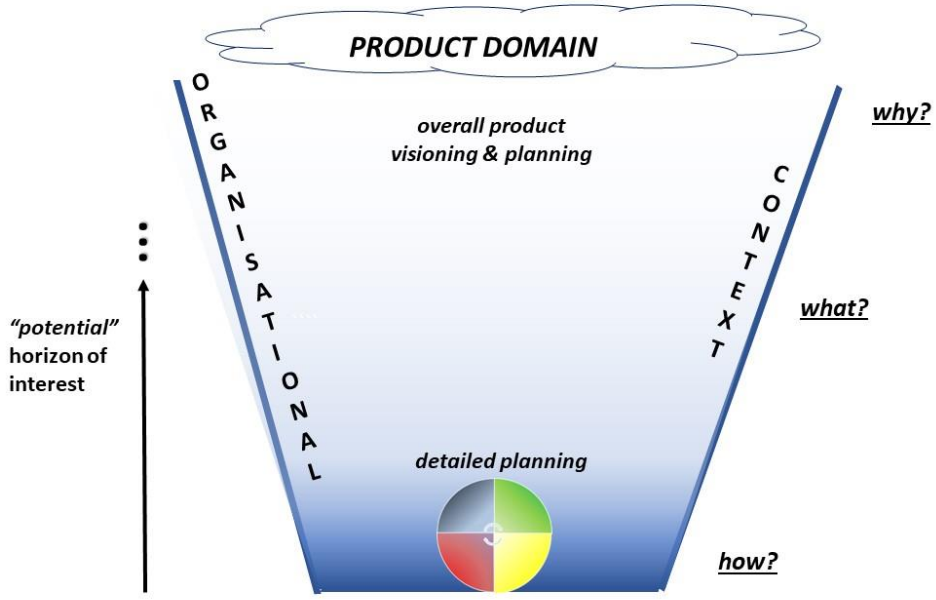


Figure 5-2. Longer Horizon of Interest

5.2.2 The Product Planning Process factor

"since requirements engineering … is highly human-based, we face the challenge to create a solid empirical basis that allows for generalisations taking into account the human factors that influence the … discipline"

(Wagner et al., 2019, p. 3)

The second factor identified in the findings that responded to the research question is the product planning process, i.e. product visioning, strategic planning, requirements engineering, road

mapping, and execution. The findings indicate that when CFPTs are more engaged with, even to the point of owning, these key product planning process elements, they tend to be more committed to the long-term success of the product (***product ownership***). They have a longer ***horizon of interest*** and will, to the extent of their collective ability, achieve a broader and deeper understanding of the product itself and of the context of the product plans (through ***striving to grok*** and ***wanting to own***). By ***horizon of interest*** in this Product Planning Process context, I am referring to how far into the future does the team have a deep interest in the product requirements as well as in the overall product vision and direction.

Teams that are not permitted to own, or at least be deeply involved with, these planning activities are limited in their understanding of the context of the product plan and in the motivation of specific requirements. This can be as extreme as the teams being spoon-fed detailed development tasks ("**I just do what I'm asked to do**"). A team's limited understanding, due to it being less deeply involved with the product planning process activities, leads to the team being less committed to the product overall. These teams remain committed to the technical excellence of their development work product; however, they tend to be more focussed on the near-term development plan, the only part of the planning process in which, by default, they are deeply involved. In some cases, this may also keep the team aligned with expectations held of them by senior leadership.

Figure 5-3 below illustrates this condition described where a CFPT is limited in its ***horizon of interest*** by constraints placed on the team's involvement in the product planning process. This constraint 'blanket' has a strong negative impact on the final outcome because it represents an additional boundary across which plans must be communicated. Plans communicated across that

barrier are not detailed technical specifications, rather they are a higher-level description of requirements and plans, yet the development team on the receiving end of those plans have a limited context within which to understand them ("**we just do what Product Management says**"). Software product development without a rich context of the market needs is one of the contributing factors to the NaPiRE survey results that show *incomplete/hidden requirements* being a major cause of project failure (Wagner et al. 2019). This communication challenge is discussed in more detail in section 5.3.2 - Requirements Engineering is a Team Sport.

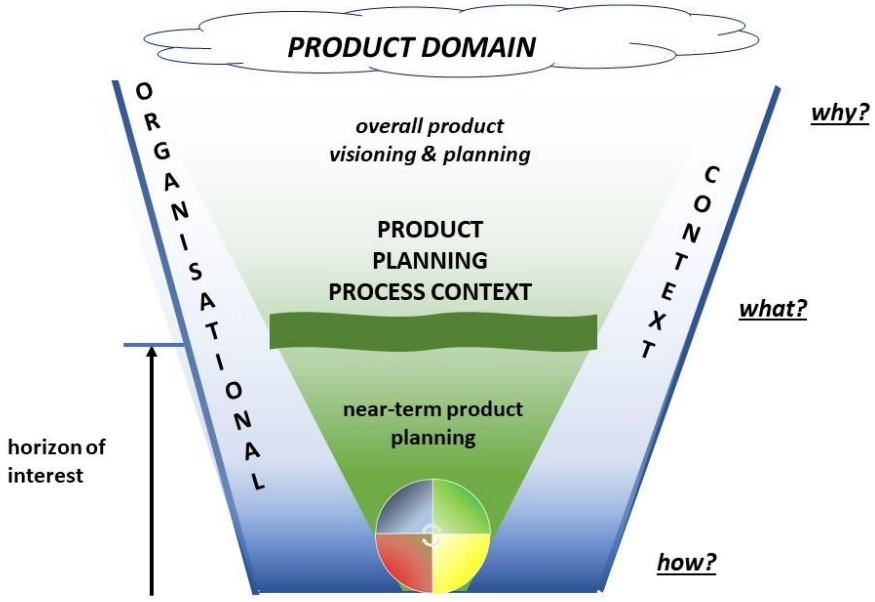


Figure 5-3. Process-constrained Horizon of Interest

Where there are few or no constraints imposed on the team by the product planning process (or at least flexible constraints) and where the CFPT is deeply involved throughout, the team has a greater cohesion around the plan and assumes a full horizon of interest in the product towards the

product domain, from visioning through to development and back again, as the team strives to achieve domain context by grokking the product domain. See Figure 5-4 below.

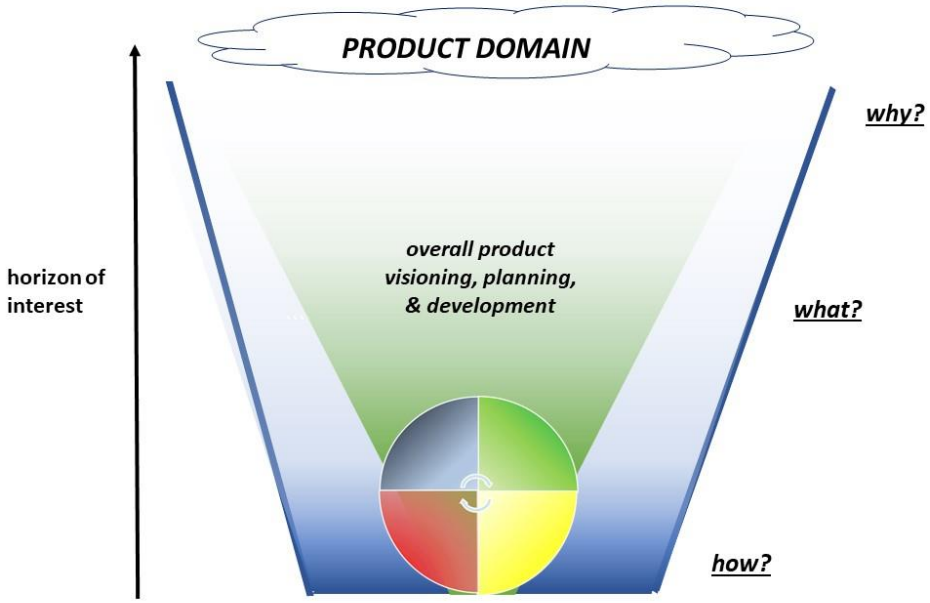


Figure 5-4. Unconstrained Horizon of Interest

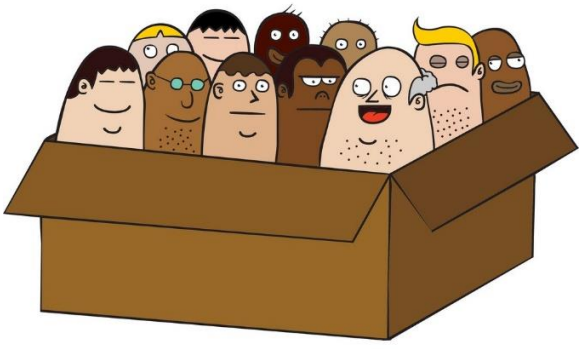
5.2.3 "and beyond"

The research question was stated in a simple '*what*' form and the two factors just described provide an accurate and useful answer to that question. The substantive grounded theory that emerged is perceived as the product of complex interactions and layers of processes emerging from the impact of the factors found in response to the research question. However, throughout the analysis, two recurring themes contributed to a conceptual framing of the research findings in other respects beyond simply answering the research question. These themes are referenced in

Chapter 4 and are discussed further in this section. In Section 5.3, following, I explore them even further in the context of additional concepts that arose.

The first of these framing themes that persisted throughout the analysis of both factors was one labelled ***distinct boundaries***. The term *boundaries* is referring to role definitions, process stages, etc. with the corresponding interfaces that are necessary for communication and hand-off when boundaries exist between collaborating entities.

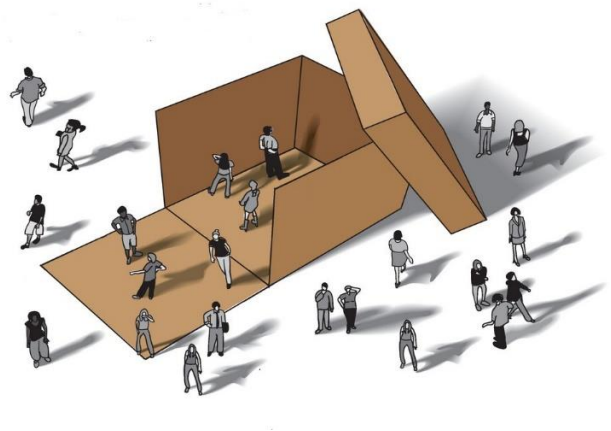
Distinct Boundaries – stronger and more well-defined delineations within and between organisational units and planning process elements. ***Distinct boundaries*** results in less empowered and more constrained CFPTs that were observably less committed to the overall product result. This lower commitment to the overall product result is a consequence of these teams being less able to deeply and collectively understand the product domain and, therefore, less able to have a rich context of the market requirements.



(teams tightly constrained are as if in a box can grok only what they are allowed to see and can do only what they are allowed to do)

Figure 5-5. people in a box

No, Soft, or Flexible Boundaries – weaker or more flexible delineations within and between organisational units and planning process elements. It was observed that these softer constraints tended to result in CFPTs feeling more empowered and committed to the product's long-term future. With teams more able to exercise their potential to understand the product domain deeply and collectively, they achieved richer contexts for the market requirements.



(teams less constrained are more able to grok to their collective potential and more able to do what they feel needs to be done)

Figure 5-6. people outside the box

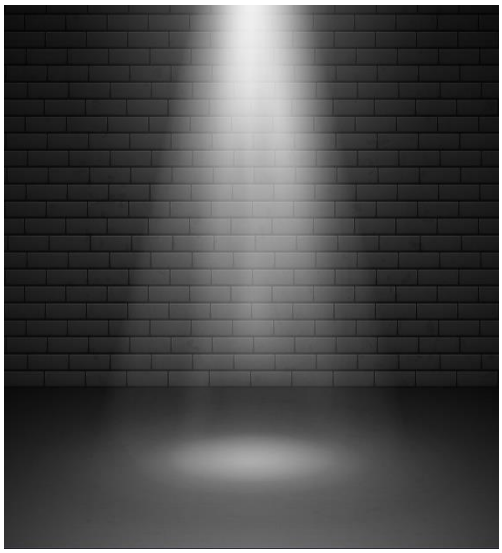
Within the findings presented in Chapter 4, another persistent theme was described, a metaphorical mechanism which I labelled ***Broadening the Lens***. This is a simple, figurative description of what individuals and teams do in order to see a bigger picture, a picture that can offer them insights into relationships, patterns, structure, etc., to create context. The reverse of that metaphorical mechanism (***narrowing the lens***) is what individuals and teams do to focus on detail. To use the "T-shaped" analogy introduced by Johnston (1978), a broader lens allows a

horizontal bar for a team member to expand whereas a narrower lens limits its span and, thus, that team member's contribution to collaboration, innovation, and creativity.

The benefit of this action of broadening and then narrowing the lens is that the detail in focus is now more informed, benefiting from the richer context obtained when the lens was broader. The limits on how far the ***broadening the lens*** action can go is determined by what constraints (***distinct boundaries***) exist, or not, on the individuals and teams from one or both of the organisational context and the product planning process context (see Figures 5-1 & 5-3 respectively). In cases where those limits are strong (shown by the barriers appearing lower in each of the images in those figures), a somewhat narrower lens is all the teams can achieve.

(a narrower lens focusses on an area of interest showing detail but provides limited context of the targetted area)

Figure 5-7. Narrow Lens



(a broader lens shows less detail of the targetted space but illuminates a broader picture, providing more context)

Figure 5-8. Broadened Lens

(imagine, if you will, a team having a requirements specification for the area in focus in Figure 5-7 with no further context provided. If the team was not able to broaden its lens to create further context, it would likely be unaware of the important presence of the brick wall shown in Figure 5-8)

Before moving on to position these findings further, it is important to clarify that the influencing factors and their impact limit CFPTs grokking potential. How well teams *actually* grokked with fewer or no limitations was not within the scope of the RQ. One might imagine a runner having shackles on his/her ankles. The shackles clearly limit the runner's ability to run fast. With the shackles removed, the runner could then demonstrate whether he/she was capable of - a world-class runner, a mediocre runner, or a runner unable to run any faster than when the shackles were on. However, the shackles would have to be removed in order to know.

Both the theme of ***distinct boundaries*** and the metaphorical mechanism of ***broadening the lens*** are recurring concepts that are woven throughout the discussion next in Section 5.3 as the implications of the findings are positioned and discussed.

5.3 Discussion and Positioning of Findings

Section 5.2 provides an overview of this study's answer to the research question. This answer is the basis for several implications for practice, which are described in Section 5.6. Section 5.2.3 highlights two select themes that help frame the research findings as they relate to topics beyond the research question which will now be discussed in more detail.

As the substantive grounded theory of **Blurring Boundaries** was emerging, several topics surfaced in the findings that warrant further discussion since they help in: a) further positioning the grounded theory, b) pointing to possibilities for future research, and c) offering more specific implications and guidance for industrial practice. These topics are: a) why do some teams strive to grok at all and what are some of the factors at play? b) the importance of human factors in requirements engineering, identified in the pre-research literature review and echoed throughout the study, and c) the contraindication of the strength and prevalence of the *as-machine* industrial metaphor in the organisational structure and the product planning process model used today in many software product companies. These topics are discussed in the following three sections.

5.3.1 Teams Strive to Grok

"to design something really well, you have to get it. you have to really grok what it's all about. it takes a passionate commitment to really thoroughly understand something, chew it up, not just quickly swallow it."

– Steve Jobs (as cited in Wolf, 1996)

As Chapter 1 states, this research was about collective grokking. It was about CFPTs, collectively striving to deeply understand the product and the context of the product, with all its potential. And it was about those teams owning the product and its destiny as much as their ability and the organisation allows.

The answer to the research question identified factors that can inhibit how much a team can grok or, conversely, factors that can allow a team to do so as much as it is capable of (see Figures 5-1 & 5-3). Throughout the research, teams were observed that attempted to grok more than they currently were (at times even pushing against the limitations or ***distinct boundaries*** placed upon them), and also observed were teams that did not seem to try to grok any more than they currently did.

Why? Why *do* some teams strive to grok in the first place? And why don't they *all* continually strive to grok even more? Chapter 4 notes that teams appear to have a propensity to *own*

something but that observation alone does not explain *why*. Why does a team have a propensity to own something in the first place? Although developed 40 years ago, Hackman and Oldham's Job Characteristics Model (Hackman & Oldman, 1980) is still highly relevant in today's software product development and offers some insights into this question. Figure 5-9 illustrates a visual summary of the model.

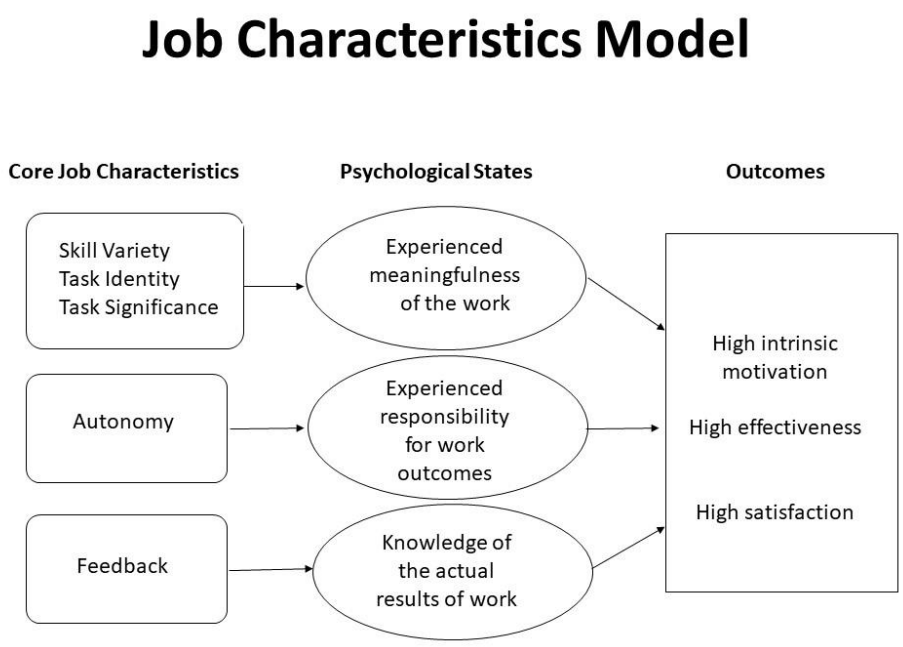


Figure 5-9. Job Characteristics Model

Adapted and reprinted with permission from Work Redesign by Hackman & Oldman, 1980, p. 90. Copyright 1980 by Addison-Wesley.

The Job Characteristics Model is helpful here because it provides some insights into conditions that create outcomes that are relevant to the question about why teams strive to grok. Specifically, the model describes three psychological states that contribute to the work-related outcomes of motivation, satisfaction, and effectiveness.

The first state in the model is *experienced meaningfulness of the work* leading to *high intrinsic motivation.* This refers to the degree to which the work is felt as intrinsically meaningful and that it reflects one's values when presented to other people and/or the external environment.

With software technology today being so ubiquitous in our economic and social fabric along with software development skills being in such high demand, my assertion is that individuals are very much in control of this need to *experience meaningfulness of the work* simply out of the luxury of having rich choices about where they work, whom they work with, what products they work on, etc. Driven by the individual's interests and values, he/she is able to gravitate towards situations and conditions (products, jobs, and companies) that reflect what matters to them and that offer work opportunities with the skill variety, identity, and significance to match. In the conditions of the software industry currently, competent software development professionals rarely need to feel stuck in a meaningless, dead-end job in which they have little choice but to put up with. This opens the doors for individuals to achieve high intrinsic motivation which then is fuel for the effort needed to ***strive to grok***.

The second psychological state in the model is *experienced responsibility for work outcomes* leading to *high effectiveness*. Several categories appeared in the findings that relate to this experience of responsibility for outcomes. The primary one is ***wanting to own***. Both at the individual and team level, ***wanting to own*** is about the desire to be accountable to make decisions and to be responsible for the results of those decisions. The size and complexity of most software products today make the creation of the product and its ongoing development a

team sport and, therefore, I shall describe the categories that relate to this psychological state in the context of the collective team as much as possible rather than only at the individual level.

In the study, I observed that CFPTs having strong ***team cohesion*** will ***want to own*** *something*, leading to an increase in *experienced responsibility for work outcomes*, and thus will ***strive to grok*** (in an effort to increase effectiveness), within the ***distinct boundaries*** they are constrained by. This effort to deeply understand what the team is allowed to own maximises the team's experience of responsibility and moves these teams in a positive direction toward addressing the widespread cause of project failure identified by NaPiRE (Wagner et al., 2019), namely *incomplete/missing requirements*.

Assembly of Experts teams, with lower autonomy (see Figure 5-1), have a much more limited definition of what the *work* is for which they can *experience responsibility for work outcomes*. For these teams, responsibility is often limited to task excellence since what they own (and feel they *could* own) is nearer-term, more narrowly defined, and therefore less collective. These teams will ***strive to grok*** in a more confined context, mainly limited to the detailed work in front of them. This limited examination of the context of the requirements is an example of the framing effect that Mohanani et al. (2014) identified as a cause of a cognitive bias in requirements engineering, namely, *requirements fixation* (p. 896).

The third psychological state in the model is *knowledge of the actual results or outcomes* leading to *high satisfaction.* As the *Ship First, Fix Later* approach becomes increasingly less acceptable

as a business practice in the software industry, it becomes more critical to *understand it first.* As Jim Highsmith (2000) stated, "*The greatest risk we face in software development is that of overestimating our own knowledge.*" (p. 14). Closely related to the propensity of ***wanting to own***, empowered product development teams look for *feedback* as one method to obtain some *knowledge of the actual results* which is motivated by the propensity to want to *understand* (***striving to grok***).

I observed that cohesive teams (those with little to no internal ***functional deference***) and those that were not being spoon-fed their development tasks, were collectively ***striving to grok*** the product domain to their ability, looking for *knowledge of the actual results of work* using as many feedback sources as they had access to and, additionally directly ***broadening the lens*** towards creating a richer collective context with which to more accurately perceive and analyse actual results or outcomes.

Another category in the analysis relates strongly to this third psychological state of the Job Characteristics Model, namely, the ***horizon of interest***. Teams that were empowered and cohesive tended to *hold the vision and deal with what is in front of them.* With a longer ***horizon of interest*** (playing the long game), they exhibited an additional interest in having *knowledge of the actual results of work*, e.g. success of the product in the market. As this feedback on the results of work relates not only to work completed, but it relates to the short- and longer-term product planning as well. Having lower ***functional deference*** and less individual tentativeness with respect to membership on the team, these teams experience (both individually and collectively) more full participation with, and commitment to, the long-term product roadmap

(***product ownership***) and therefore it becomes more important to them to have *knowledge of the actual results of work*.

On the other end of the spectrum, *Assembly of Experts* teams appeared to be less motivated by longer-term matters due to constraints imposed organisationally and/or by the process model. These teams are then low on this psychological state of *knowledge of the actual results of work* due to their shorter ***horizon of interest***. Moreover, with them often being able to focus only on an iteration or the next release plan and using their own fabricated story points as the only metric of success, they have little context with which to understand any *knowledge of the actual results of work*.

In summary, there is a close relationship between the collective job characteristics and psychological states as described by Hackman & Oldman's Job Characteristics Model (1980) and the team models used in this study. One explanatory illustration of this is the observation that *True* teams ***strive to grok*** more than *Assembly of Experts* teams and the suggestion that team *motivation*, *satisfaction*, and *effectiveness* are all increased the closer the team is able to move toward operating as a *True* team. Framed another way, the Job Characteristics Model suggests that the longer the team's ***horizon of interest*** and the more it is able to ***broaden the lens*** toward the product domain, the more motivation, satisfaction, and effectiveness it achieves.

5.3.2 Requirements Engineering is a Team Sport (not a Manufacturing Process)

Chapter 2 references a declaration from NaPiRE that after 40 years of research and practice, the discipline of requirements engineering (RE) still struggles, despite significant contributions made over those years in the form of improved tools, methods, and processes (Wagner et al., 2019). This same NaPiRE report acknowledged that RE is highly driven by human factors, yet there is a lack of generalisations in RE that take these factors into account. The spirit of these assertions echoed throughout the study as I observed software teams in action.

In response to the RE discipline being in a troubled state, Mafra et al. (2016) proposed a guideline of recommendations for development organisations. Of their 22 recommendations, six call for the adoption of, or improved adherence to, established RE methods while 10 recommendations refer to the adoption of existing software development processes and team practices. Five of the recommendations assume the existence of an identifiable 'customer' and only one recommendation refers to acquiring deeper domain knowledge.

I refer to this excellent paper by Mafra et al. not to critique the content but rather as an example to highlight my assertion that human factors are the true crux of the RE crisis. This critical problem in RE will not change by simply applying better discipline in adhering to current best practices. The majority of the recommendations made by Mafra et al. (16 of 22) refer to established practices, yet the NaPiRE surveys indicate that many of the organisations reporting dissatisfaction with RE claimed to be using established RE methods on those failed projects. Simply put, industry experience indicates that the current use of existing RE methods alone is

insufficient to address this crisis. The same argument holds for better adherence to software development process methods alone.

Nor is it likely that this problem will be solved solely by *improving* the methods of requirements elicitation, validation, and specification. As NaPiRE notes, the RE discipline has been developing these methods for 40 years, and yet there remains deep dissatisfaction with RE in industry.

While it might be hard to argue against more discipline or better methods in practice, there are many who assert that these alone are insufficient to address this issue. As referenced earlier, many industry thought leaders have strong opinions on this matter - "Individuals and interactions over processes and tools" (Agile Alliance, 2001), "the specifying, structuring aspects of RE are counterproductive" (Ralph & Mohanani, 2015), and "the idea of eliciting and capturing requirements is wrong." (Cohn, 2004). A constructive path forward is offered here.

I note that five of the 22 recommendations by Mafra et al. (2016) make reference to a 'customer', suggesting that the relationship between developers and the holders of the domain knowledge is still viewed as a significant source of requirements understanding. Add to this that only one of the 22 recommendations referred to increasing the understanding of the business domain and it was qualified with an "if needed". This points to the heart of this crisis, namely, when software is being developed as a product for a marketplace rather than a bespoke software deliverable to an identified customer, where does the deep domain knowledge reside?

The graveyard of software companies is full of firms that did not realise that neither a single customer nor a single prospect is an accurate representation of the market - rather they are simply unique examples. Obtaining the first happy customer is not hard, finding a second happy customer with the same solution that was specifically developed for the first customer requires a development approach that is *market*- rather than *customer*- driven.

However, the RE discipline has not faced this issue directly. Practices of RE elicitation and validation typically make reference to the role of a customer and yet, in the software *product* development context, development teams need to understand the *market* deeply, not simply *customer* needs, and certainly not a *single* customer's needs. Common practices in use in industry often insulate the software product development life cycle (SDLC) from this complication by inserting a 'customer representative' or 'customer surrogate', etc. into the process, essentially saying, "let's just proceed as if we have a customer".

This *customer* approach is simply insufficient for developing software *products* and so it is unsurprising that so many of the recommendations made by Mafra et al. would point to getting better at communicating with the 'customer', consistent with the RE improvement theme for decades being "we just need to get more disciplined at this". Rich and complete understanding of the market needs and requirements, competitive landscape, etc. along with the context in which they reside, cannot be dependably communicated to the product development team in a manner akin to the children's 'telephone game' where information (and understanding) is communicated (and morphed) link-to-link along the complex chain of the SDLC process. See

Figure 5-10 – Telephone Game for a light-hearted, mild, and yet all-to-typical, example that characterises the ‘telephone game’.

Figure 5-10. Telephone Game

Copyright 2008 Marketoon Studios. Reprinted with permission.

Participants in the telephony chain can often be aware of the problem but not always in a position to correct it. A quote from a senior software engineer participant in this study expressed this well:

"I spent a lot of my time trying to understand what the PO (Product Owner) wants and I don't even know if that's necessarily what the market needs" - WA08

It is easy to see that the more links there are in the communication chain, the more both explicit and tacit information loss is likely to occur. Therefore, adding more roles and methods into the RE process is likely to worsen the problem. While this phenomenon is commonly seen in communications between departments (e.g., product management - engineering), it can also be experienced within teams, particularly if the team is not very cohesive with high degrees of ***functional deference***.

It is important to point out that the grokking of the product domain by CFPTs includes deeply understanding the *tacit* knowledge of that external product domain and its market context. More sophisticated analytical techniques and comprehensive templates cannot capture this tacit, yet critical, understanding. Even capturing and transmitting what can be *explicitly* (and, simply) expressed is difficult in this manner as Figure 5-10 above illustrates.

So why has this outdated approach to RE persisted for decades if it is so ineffective? In *The Ambiguities of Experience*, James March offers one possible insight into why the RE discipline is so fixated on stories passed along a chain to communicate market needs and product requirements. He writes,

> *"the argument in favor of samples can be seen as … some kind of intellectual foundation for a deep human predilection to stories. Arguments are mobilized to provide a justification for the … intelligence gained from converting instances of experiences into possible [emphasis added] understandings of the world. Since the ability and eagerness of the human intellect to conjure utilitarian [emphasis added] justifications for intuitive prejudices is one of the more endearing human traits, the efforts are often elegant and even persuasive." (March, 2010, p. 116)*

In other words, Walt Kelly's Pogo was right: "We have met the enemy and he is us" (Kelly, 1972). Next Section 5.3.3 offers another, more foundational and actionable, explanation.

The research findings show that empowered, cohesive teams do ***strive to grok***. If progress is made by teams in this quest, it reduces, towards even eliminating, the 'telephone game' by transmitting as much knowledge and understanding as possible directly to the team and allowing the team to collectively absorb as possible directly, gaining some tacit knowledge, richer domain context, and minimising the shortcomings of multiple communication links. The degree to which the team is able to do this is subject to the constraints of the organisational and product planning process model in place surrounding the team, however, their efforts in ***striving to grok*** demonstrates the team's desire to experience responsibility for work outcomes (Hackman & Oldman, 1980) and grokking with whatever they have to grok with. In cases where teams feel sufficiently safe psychologically (Edmondson and Lei, 2014), this can even involve the team

pushing (blurring) boundaries that were otherwise distinct (challenging the constraints). Section 5.4 discusses this notion of ***blurring boundaries*** further.

As noted, the NaPiRE initiative acknowledges that human factors are a major characteristic of the requirements engineering discipline (Wagner et al., 2019). However, ***distinct boundaries*** both in the organisational model and in the product planning process constrain the human element in the requirements engineering process by deepening this 'telephone game' communication style. Specifically, as it relates to communicating product requirements, product experience in the market, and market conditions, it results in costly information omissions, spillage, and incorrect understandings – lost in translation. Not to mention the important explicit and tacit knowledge that was never captured in the first place.

Not only is the understanding of the context of product requirements impacted by the failure to infuse domain knowledge directly into the product teams, product *design* is also impacted as noted by Ralph and Mohanani (2015), "increased domain knowledge improves design performance" (p. 2).

My findings lead me to strongly agree with NaPiRE that human factors are a major characteristic of the requirements engineering discipline. I will state this even more strongly and argue that, in the current software product development environment, human factors are the most critical factor in requirements engineering. I assert that the discipline will continue to struggle until more attention is given to the fundamental issue of how to have external product domain knowledge deeply understood by the CFPTs. How can we help those teams grok their unfamiliar and

complex product domains? Better tools and methods alone will be insufficient to address this need.

"the perfect ideal involves the developers themselves as customers"

(Coplien & Harrison, 2005, p. 117)

5.3.3 "As-Machine" models are counterproductive

"Optimization stifles emergence, not only because individuals feel restricted but also because optimization reduces the breadth and scope of interconnectedness and relationships." ***(Highsmith, 2000, p. 288)***

Upon reflection, and with my extensive experience in industry as a contextual background, few of the detailed findings toward answering the research question seem particularly surprising. While the analysis provided a deeper understanding of what was going on, on the surface, the situations observed were ones that I had often seen in my professional experience. The obvious question is, why? Why do companies get in the way of their software product teams' desire to understand, to own, to make the product and the company as successful as possible? This is even more puzzling considering that this phenomenon is not particularly new, as NaPiRE attests (Wagner et al., 2019).

The research question was a 'what' question, not a 'why' one. The answer to the research question identified two quite basic constructs, namely the organisational model and the product

planning process. Rephrasing the question above, why do companies, specifically, *software product* companies, adopt the organisational and product planning process models that they do? In this section, I discuss some of the possible reasons for this and the implications of adopting the models so commonly found in industry.

My findings are consistent with team studies by Gladstein (2006) and Ancona & Caldwell (2008) which indicate that contextual factors (e.g., organisational structure, resources available, and functional mix) have a greater influence on team effectiveness than do internal team processes. Specifically, my data shows that the contextual factors of a team have a significant impact on internal team character and behaviour including, but not limited to, processes. This section focuses on four of these impacts on team character. Section 5.3.3.1 discusses the organisation's ability to learn, Section 5.3.3.2 looks at functional diversity in the teams, Section 5.3.3.3 addresses the topic of the stability of team membership, and Section 5.3.3.4 makes some brief comments on the location of functional roles outside of the team. Section 5.3.3.5 concludes with a discussion of the guiding metaphors behind the organisation and process models.

5.3.3.1 The Learning Organisation

"[the] functional organisational structure works well in a stable environment where business strategies are less inclined to need changes or updating"

(Awa, 2016, p. 1)

Note that none of these conditions mentioned in Awa's quote above reflect the nature of the software products industry today which faces strong pressures for continual change - change due to technology, fashion, economics, and social conditions. Moreover, pressures from competitive forces in this industry are much more often based on innovation and clock speed of change than they are based on lower cost (Amjera & Maynard, 2015). Yet, the industry continues to design organisational structures and processes with ***distinct boundaries*** and narrowly defined roles, driven by a strong tone of *optimisation*.

One argument for the focus on efficiency and optimisation often comes out of the mantras such as '*fail fast, fail often*' or '*no product plan survives contact with the market'* or other such approaches that have a similar essence as the Minimal Viable Product (MVP) concept from the Lean Thinking discipline, e.g. Eric Ries (2014). These approaches take a fast, iterative, and experimental approach to development and learning where some proponents (mistakenly) argue that efficient structures and processes are necessary in order to experiment frequently and then to adapt rapidly based on the learning of the experiments. The findings of this study suggest that greater CFTP grokking of the product domain minimises and softens that market contact shock and, most importantly, maximises the learning from the market contact experience.

However, as noted, unless the CFPT is given the freedom to grok the product domain, they will not have the psychological state of *knowledge of the actual results of work* (Hackman & Oldman, 1980) and, therefore, will not quickly nor completely learn from the MVP-like experiments, nor will their learning be as direct and rich. This is because a great deal of the learning from experiments is *soft* knowledge. Communication of all knowledge gained from the market

experience, especially *soft* knowledge, to the CFPT that occurs across ***distinct boundaries*** in the organisation will suffer from the ineffectiveness and information loss at each of the communication links (the 'telephone game').

Emphasis on rapid learning requires a *learning organisation* as described by Senge (2006) which, at its heart, is an organization skilled at creating, acquiring, and transferring knowledge. This means that a learning organisation is also skilled at modifying its behaviour to reflect new knowledge and insights. One of the critical dimensions of team learning as defined by Senge is, "the potential for many minds to be more intelligent than one mind" (p. 236). Senge identifies a second critical dimension of a learning organisation as having "the need for innovative, coordinated action" (p. 236). The first of these, many minds are more intelligent than one mind, is a characteristic of the *True* team model, as Chapter 4 describes. The second critical dimension, the need for innovative, coordinated action, is an imperative for any software product company in a highly competitive market. Neither of these critical dimensions are likely to exist in an organisation that has boxed in its CFPTs with ***distinct boundaries***.

Simply put, software product teams that are constrained when ***striving to grok*** are not part of a *learning organisation*. If they are not able to grok the product domain, they are not able to *learn fast* by quickly and collectively absorbing the market experience of a product release or changing conditions.

In a rebuke against the highly structured and mechanistic approach to organisation and process design for a learning organisation, Hong (1999) declares that "it is only through this dramatic

change of structural paradigm that the ideal notion of the ``learning organization'' as argued by Peter Senge (1990) will be realized" (p. 184). Edmondson and Lei (2014) noted that "Challenging the status quo and offering ideas to improve process can be a vital force in helping organizations learn" (p. 27). Yet, if ***distinct boundaries*** are viewed as 'do not cross' borders, teams will be reluctant to ***broaden the lens*** further than the boundaries indicate, since doing so may not be safe, thus inhibiting the organisation's ability to be a *learning organisation*.

> *"The vision develops, evolves, and socialises in the team through constant discussion, NOT via a form of specification to define, ground, and document. This constant discussion includes continually (or easily) revisiting ground previously covered (pawing)."*
>
> *(Team C, Organ X observation notes)*

5.3.3.2 Functional Diversity

"results indicate that functional diversity is positively related to creativity and innovation" (Reiter-Palmon et al., 2012, p. 298)

With this thought from Reiter-Palmon and wanting to explore Awa's point further about more rigid functional organisational structures being better suited to stable environments that are less likely to require change, I approached senior leaders in companies (both formal participants in the study as well as some outside of the participant group) that had an overall functional structure in place. I asked them about their intentions when they formed their CFPTs, specifically whether

they formed them as special cases or whether creating them indicated that the main product development activities were no longer well-handled through the coordination of larger functional groups. The most common response received was that they had formed CFPTs with an instinctive belief that it would improve both innovation and productivity and many added that they also expected it would simplify project management. This is consistent with the views of Reiter-Palmon et al. (2012) and with the fast experimentation models mentioned in the previous section. Based on these responses, it appeared that forming cross-functional teams was a tactical attempt by companies with functional organisational structures to counteract the negative consequences of those structures.

As I reference above, studies have shown that team behaviours which emphasise collaboration, open discussion, and inclusive team behaviours are critical for team creativity and innovation (e.g., Mitchell et al., 2009). Awa (2016) goes on to point out that, "[the] challenge of the functional structure is the tendency for employees to take a specialist viewpoint" (p. 2). This *specialist viewpoint* can be external ***functional deference*** (between functional groups) or internal ***functional deference***, tending towards *Assembly of Experts* team behaviour. This supports my findings that when a broader functional organisational model exists, the individuals on CFPTs and those CFPTs themselves tend to exhibit that *specialist viewpoint*, presenting behaviours that are contrary to the results that senior leaders expressed to us that they were striving for when they created those teams.

In addition to the *specialist viewpoint* looking at the product plans through a narrower lens while respecting ***distinct boundaries***, the strong ***functional deference*** that the *specialist viewpoint*

creates increases the number of communication paths within the team, inhibiting high-bandwidth communication across functions (a core principle of the Agile paradigm) and segmenting goal setting, thus interfering with team creativity and innovation. This was explained by Goel (1995) when he asserted that significant mental processing happens in a realm of imprecise thought, proto-thoughts of ideas whose boundaries have not yet been demarcated. ***Distinct boundaries*** and strong ***functional deference*** also make the team less collectively intelligent, as explained by Woolley et al. (2010) who found that collective intelligence was positively correlated with the equality in distribution of conversational turn-taking (p. 686).

In summary, my findings support the view that functional diversity has a positive effect on team learning and innovation provided the functions are cohesive (i.e., working together as a *True* Team) as opposed to simply working side-by-side (as in the *Assembly of Experts* style teams). Put another way, functional diversity in action adds to creativity and innovation by ***softening boundaries*** and encouraging high-bandwidth, less mechanical, and more creative communications. Functional diversity in appearance alone does not.

5.3.3.3 Team Membership Stability

what team am I on (today)?

Team cohesion and productivity are both negatively impacted when the team unit itself is not viewed with the same *organic* consideration as are the individuals. There are many team development models that identify membership turnover as an erosion factor for ***team cohesion***. One of the well-known ones referenced frequently in software development circles is from Bruce

Tuckman (1965) who, with particular emphasis on the team as a cohesive unit, described a Forming-Storming-Norming-Performing model of team development evolution. In his model, there is a *norming* stage where ingroup feeling and cohesiveness develop and eventually lead to a *performing* stage where roles become flexible and the team starts to express more creativity, a softening of ***distinct boundaries***. There is an implication of his model that suggests that significant change to team composition while in the norming or performing stages of team development creates backward pressure towards the storming or even the forming stages. Tuckman hints at this by noting, "…expect "newness" of the group to be greeted by orienting behavior and resultant unsureness and insecurity" (Tuckman, 1965, p. 396).

Certainly, there are times when there is a need to change team membership in order to bring in different expertise, perspectives, capacity, etc. and the resultant orientation effort and 'chemistry' adjustment are to be expected. With Tuckman's point in mind, the advice would be to ensure that the benefit is worth the cost. With some teams in this study, several factors frustrated their efforts to become a *performing* team (very frequent team member reassignment, corporate human resources rotation policy across teams, high overall turnover, etc.), resulting in the teams spending an inordinate amount of time in the formative stages (*forming* or *storming* stages in Tuckman's model) compared to time spent in the ideal *performing* stage. Those CFTPs finding themselves perpetually in the *forming* or *storming* states remain closer to an *Assembly of Experts* workgroup, possibly with functional sub-groups inheriting similar traits, but locked into a short ***horizon of interest*** with a narrow lens.

5.3.3.4 Functional Location in the Organization

I observed that in cases where relevant functions in the SDLC were not all represented within the CFPT (e.g. in some cases, product management or UX design), functional deference shown between functional groups that relate to the product's development often shared many of the same characteristics as they appeared internally in Assembly of Experts cross-functional teams. Any strong deference displayed between functional groups (e.g. between a product management group and a user experience design group) relating to a product had a similar effect as with complete CFPTs showing strong internal functional deference - dampening the expression of categories such as, wanting to own, striving to grok, horizon of interest, and broadening the lens by those functional groups, holding what Awa (2016) called "a specialist viewpoint" and constrained to only using a narrow lens. This is most evident as organisations scale and the functional roles and groupings become more entrenched with their own gravity, precise mandate, and distinct boundaries.

5.3.3.5 The Wrong Metaphor?

"It is impossible to develop new styles of organization and management while continuing to think in old ways" (*Morgan, 1997, p. 63*)

There is an imperative for a successful software product company to be a learning organisation, to strive for creativity and innovation through functional diversity, to foster stable and cohesive teams that look into the future, and for that organisation to support functional groups outside the

cross-functional product development teams to achieve similar behaviours. Striving towards this, I repeatedly observed actions such as ***broadening the lens*** and ***striving to grok***, actions that are aimed at supporting those imperative objectives just mentioned and, yet, hindered in doing so because those actions ran opposition to the mechanistic organisation and process models and mindsets in place in those organisations and so commonly seen across the software product industry.

Are we using the wrong metaphor in software product development organisations? In a technology-oriented industry such as software product development, specialisation, standardisation, and measurement in organisational structures and process models is almost intuitively implemented, and very often even intentional. This might be expected since the industry (and the technology space, in general) is dominated, and led, by analytical minds, minds that *build complex things*. The things these analytical minds build are like machines - a series of specialised components with precisely defined purposes, connected in a logical order in order to produce an output. Thus, we often see organisational and process models in software organisations defined using this same *as-machine* industrial metaphor as described by Morgan (2006). Another reason that the adoption of this metaphor is so natural for software companies is that the principles that relate to the management of information, a large component of the roots of the software industry, are deeply rooted in the mechanistic approach, namely, information *systems*.

Typically, this metaphor is adopted instinctively by software organisations, almost without questioning the choice. When there is consciousness about designing the organisation and

process models, the decisions are usually driven by a relentless quest to *optimise*, striving for predictability, to produce more in less time and at less cost. Frequently, there are additional arguments used that refer to the value of focus and clarity, the need to decompose complexity, etc. Yet, as referenced earlier, these are not the main critical success factors that are at play for much of today's software industry - rather the industry depends on creativity and innovation. Yet, this study shows that when the *as-machine* metaphor-like approach is applied to organisational structure and process models, there is an inhibiting effect on both creativity and innovation.

In the quest to *optimise*, the industry puts great effort into measuring whether we are doing the thing right. However, there are often few, if any, useful indicators to identify whether we are doing the right thing. For the most part, whether the right thing is being done becomes a post hoc assessment. For minds embedded in the *as-machine* organisational and management metaphor, this assessment often does not include examining the underlying causes. The problem is not simply that more attention ought to be placed doing the right thing over doing the thing right, but rather to repeat Jim Highsmith (2000), "optimization stifles emergence" (p. 288). Thus, if the software organisation is putting significant effort into optimisation, it is already dampening its own possibilities. Later, Highsmith (2001) states this more broadly:

> *"The frequency of change and the speed of the market have created a raw, hostile environment, one where the basic tents of process improvement, software engineering, and command and control management are insufficient*

for success…. In these environments, adaptation is significantly more important than optimization." (p. 252).

The impact of adopting any metaphor is broad. In Metaphors We Live By (1980), Lakoff & Johnson wrote:

"… we define our reality in terms of metaphor and then proceed to act on the basis of the metaphors. We draw inferences, set goals, make commitments, and execute plans, all on the basis of how we in part structure our experience, consciously and unconsciously, by means of metaphor." (p. 158).

Once established, metaphors become entrenched and it is difficult and painful to change them. Morgan (1997) noted, "It is impossible … while continuing to think in the old ways", thus to change a metaphor is akin to what Kuhn (2012) described as a paradigm shift. Attesting to this difficulty in changing an established metaphor we only have to witness the emergence and evolution of the Agile model of software development that began in the late 1990s. As the heavily prescriptive software development life cycles reflecting industrial manufacturing metaphors eventually came to be ill-suited for the rapidly changing and increasingly complex software product world, a new paradigm emerged, *agile*. Yet, even for an industry conditioned to rapid and continual change, this shift of a life-cycle paradigm is still *conceptually* in-progress in many software development organisations after more than 20 years. For some, there is comfort, even safety, in rigidly-defined structure and those people are resistant to even softening boundaries, much less eliminating them.

I argued (Fuller, 2019b) that this mechanistic metaphor has outlived its appropriateness in software organisations, particularly when the metaphor is applied to the organisational model with its consequential impact on requirements engineering. I now make the same assertion regarding the contraindication of the *as-machine* metaphor when applied to the product planning process. The traditional model of writing software specifications and writing software to those specifications with the requisite iteration in the process to address change and uncertainty is 2-dimensional development. The empathic-based software product development I am referring to, with cross-functional teams grokking the *product* domain, adds depth to the process in the form of tacit and unexpressed explicit knowledge along with deep, as if lived, understanding. This makes product development more like a 3-dimensional activity. If the team groks the domain, they have context for the product needs, and they will *feel* the development.

Thus, the challenges of software product requirements engineering, with its multi-layered tacit dimensions, *other world* target, continual change and ambiguity, and dependence on learning, creativity, innovation, and emergence requires a more responsive and organic approach and cannot be adequately addressed using a mechanistic metaphor.

> *"the old (hierarchical structures and managerial processes) simply can't handle rapid change."* *(Kotter, 2012, p. 5)*

5.4 Blurring Boundaries

the future cannot be reimagined in a box

What is it about the organisational model and/or the product planning process that can have such inhibiting, almost suffocating, effects on product teams? Certainly, when it is applied, the *as-machine* metaphor just discussed can contribute to the stifling of learning, creativity, innovation, and adaptation. However, I felt that a deeper conceptualisation was needed to better understand the fullness of what was going on. An answer was found when I examined the theme of ***distinct boundaries*** and the metaphorical mechanism of ***broadening the lens*** more closely to also include a relationship between them.

The ***distinct boundaries*** referred to are the distinctions between differing roles within a CFPT, the boundaries between functional groups outside of the team, and the (often vast) space between the CFPT and the external product domain. The first two are created by the software product organisation and my findings indicate that the softer those are (even as far as being non-existent), the more it is possible for the boundary between the team and the domain to be softened and transcended.

Distinct boundaries within the organisation clarify and strengthens the organisational awareness of specific 'roles' which, as mentioned earlier, is usually driven by an efficiency or optimisation objective. Recent efforts to define broader functional roles, such as 'full-stack developer', are partially aimed at reducing functional boundaries and the constraining effect they can create, particularly in environments that require a high degree of agility, mostly in the form of greater flexibility in balancing workloads. However, these are superficial adaptations that do not change the underlying metaphor in the organisational model and, therefore, do not completely address the causes of the inhibiting consequences of ***distinct boundaries***.

Why *is* ***distinct boundaries*** an issue for software product development? And specifically, why is it an issue for requirements engineering? I offer three reasons. The first is the issue of the 'telephone game' communication problem mentioned earlier. The more boundaries there are to scale between the originator of the message (or the source of the information) and the receiver, the more information loss of explicit knowledge there will be when communicating across the boundaries. And further, the more distinct the boundaries between the *links in the chain*, the more the information loss.

The second issue with ***distinct boundaries*** arises when the market needs and product requirements are being elicited and analysed by a function outside the CFPT. How the collective CFPT or individual team members obtain the *tacit* knowledge of the product requirements (including the context of the requirements) and also how the team obtains the tacit knowledge of the learning from the experience of the product in the market becomes significantly more difficult. Tacit knowledge can only be obtained by experience, not communicated along a chain nor by someone across a ***distinct boundary***. Ferrari et al. (2016) showed that ambiguity can play an important role in disclosing tacit knowledge in requirements elicitation. However, requirements analysts, operating on the other side of a ***distinct boundary***, in their effort to convey clear and precise requirements, will tend not to include ambiguous considerations in the requirements communicated to the development team. When faced with an ambiguous situation, they will often, consciously or subconsciously, assign their own meaning to what they perceive is ambiguous in an effort to be precise. The RE discipline has long had the battle cry of striving for *clear and concise requirements*. Ambiguity, as insightfully rich as it sometimes may be, does

not have a welcomed place in the *requirements* engineering world. Picking up on the assertion from Ralph (2013) that requirements are illusory, perhaps we might consider substituting the word *requirements* for *opportunity* and then, perhaps, have a more adaptable view on how best to manage them, benefitting from the learning opportunity ambiguity and tacit knowledge has to offer.

The third issue with ***distinct boundaries*** relates to the CFPT grokking the product domain itself for the purpose of obtaining a deep context within which to understand the product requirements. This is important because all software is developed in context. It is that context that frames the individual and collective understanding of the requirements, however it is that those requirements may be obtained. This context also guides product decisions that are made collectively and also by each individual team member, the explicit decisions and the tacit ones, throughout the entire development life cycle. I am reminded here of The Allegory of the Cave in Plato's Republic (Halliwell, 1988) where the prisoners, being confined to the cave, had no choice but to believe that the shadows they saw on the cave wall were how things truly were. They had no other reality. Without the ability to walk out of the cave and experience the world for what it truly was, they had to rely on shadows, representations that came to them, without context, and without their own direct experience.

Many software product development teams are in that allegorical cave. Reality is often selected and interpreted *for* them, reflected to them like the shadows on the cave wall. Shadows of shadows along the "telephone game" chain of communication of market needs and requirements. My findings indicate that ***distinct boundaries*** in the organisational model and/or the product

planning process model inhibit the CFPT's ability to grok the product domain - to turn around, blur the boundaries, walk out of the cave, and directly see the source of the shadows. The world as it is, the domain for which their product is intended.

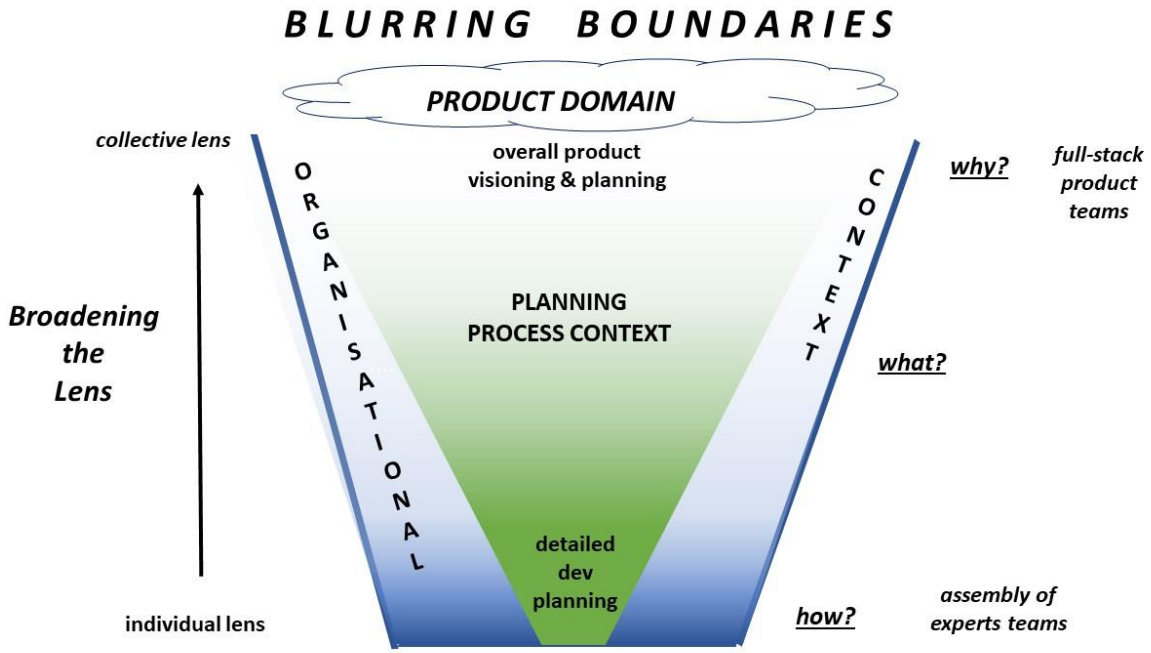


Figure 5-11. Blurring Boundaries

5.4.1 Introducing Blurring Boundaries

The term ***blurring boundaries*** is used many times in previous sections. This section discusses it more thoroughly.

The metaphorical effort of ***broadening the lens*** by product teams to see and understand a broader picture may be applied internally to understand and own the product plan and it may be

applied externally toward understanding the product domain, both of which are applications of the metaphor and are key mechanisms in ***striving to grok***.

The spectrum of collective grokking of the product domain ranges from not asking questions ("**just do what the story says**") through to *intellectual* domain *knowledge* (learned knowledge of vocabulary, workflows, objectives, etc.) and finally on to *true felt (as if lived)* domain *understanding*. The more a team groks, the further along this spectrum it moves (upward in the funnel shown in Figure 5.11). The Oxford English Dictionary defines the verb *grok* as to 'understand (something) intuitively or by empathy' (Oxford English Dictionary, 1989). To understand something by empathy is, to use Krznaric's (2014) definition of cognitive empathy, "*the ability to imaginatively step into another domain, understand the perspectives of those in that domain, and use that understanding to guide decisions"* (p. x). This is perspective taking, perceptual and conceptual perspective taking.

Thus, in the context of requirements engineering, I suggest that empathy, specifically collective cognitive empathy, is a fundamentally important ability for the team to possess in order for it to more deeply understand the external product domain. A team cannot collectively create imaginative proposals for an alternative future of a product domain that the team itself does not deeply understand. Exercising the ability to grok, to cognitively empathise, to figuratively step into that other domain, involves a certain temporary softening of the ***distinct boundaries*** between the team as a collective and the product domain itself. Integrating the theme of ***distinct boundaries*** and the metaphorical mechanism of ***broadening the lens***, I introduce the grounded substantive theory of ***Blurring Boundaries***.

Blurring boundaries is temporarily softening the distinctions that exist towards more deeply understanding perspectives on the other side of the boundary. Applying this to the requirements engineering aspect of software product development and to the team's actions in ***blurring boundaries*** between itself and that product domain so that it may more deeply understand the domain context, represents significant challenge, effort, and ability. ***Broadening the lens*** is a key technique used in addressing that challenge, a necessary technique for the team to be able to see the other domain in its context. ***Blurring boundaries*** clears the way for ***broadening the lens*** to be more effective. It allows the team to get closer to the product domain, with its lens broadened as much as possible, towards more deeply understanding (that is, *to grok*).

5.4.2 Constraints on Blurring Boundaries

As this research notes, teams have a propensity towards striving to grok. However, the team is motivated to strive and can only make progress towards actually grokking the product domain to the extent that the organisation is not constraining the team's ability to do so. When individuals or teams attempt to see a bigger picture by broadening the lens and reach a pre-mature limit of how broad they can adjust their lens, it is because they have encountered a distinct and hardened boundary. I have shown that these boundaries can exist in both the organisational model and in the product planning process. Either of these factors can create and harden boundaries, constraining how far broadening the lens can be taken and thus determining the potential value of the action.

5.4.3 Collective Empathy

Most literature on empathy describes empathy as an ability inherently directed at a different individual/group and that literature tends to discuss it primarily as a *personal* ability. However, the intergroup theory from Smith et al. (2007) suggests that empathy can become collective and that as such it can be a palpable attribute of the group that defines the group as more than just the aggregation of individuals' attributes. It suggests also that empathy strengthens the bonds of group membership (***team cohesion***). They claim that, as a collective ability, empathy carries more powerful action tendencies than do the sum of the individual abilities. My observations support this view. I observed that teams that were ***blurring boundaries*** (by ***striving to grok*** the product domain) were teams that were closer to the *true* team model and, the closer they were to that model (i.e., the more ***team cohesion*** they had along with clear and broad ***product ownership***), the more they seemed to be attempting to blur the boundaries in a *collective* manner.

To summarise, for a CFPT to be able to figuratively step into another domain, an external product domain, and to do so collectively, the team must see itself as a cohesive unit (***team cohesion***). This can be achieved only when there is a high level of transparency across all functions on the team, with little to no ***functional deference*** shown within the team, and a strong sense of ***team ownership*** for the product, felt both within the team as well as outside of the team within the company. In other words, a team with a strong sense of being a collective, with a strong product mandate, and the ability to ***blur boundaries*** towards grokking the external product domain. It requires team members to feel psychologically safe (Edmondson and Lei, 2014; Google re:Work, 2020), have open minds, a high level of curiosity, and a strong common purpose (Mitchell et al., 2009). If any of these are weak or missing, efforts toward market

discovery and solution innovation are inhibited (Reiter-Palmon & Harms, 2017), grokking the product domain will be weak or non-existent, and the results are very likely that the product will not meet the market needs as well as desired.

Boundaries demark the organisation from its environment and define the character of its inner world. I submit that boundaries play a significant role in the success or failure of products for a software product company and, therefore, for the company overall. Traditional models no longer meet the needs of today's software product companies with the requirement for exceptional creativity and innovation, rapid adaptation to market and technological change, and deep market understanding. While there is a lot of experimentation occurring in industry, the organisational and process implications of these pressures are still incompletely explored and understood, even though, as noted by Santos & Eisenhardt (2005), "the study of organizational boundaries is foundational" (p. 505). I hope to contribute insights in this respect.

5.5 Implications for Practice

As Chapter 1 notes, a core motivation for this research was to create insights that would be helpful to industry practitioners managing the entire life cycle of developing and managing software products. In addition to this motivation giving rise to the research question, it also guided my choice of method, sampling of participants, and validation of results. Reflecting upon the findings through the lens of my own professional background, I present the following implications of the findings for industrial practitioners, primarily to leaders in these software product companies that are in a position to effect change. These are presented in the form of

factors to be aware of and recommendations to consider. This is offered in the hope that increasing the awareness of the impacts that certain software product development leadership practices have will help organisations and teams move towards achieving a deeper and more collective understanding of product requirements, resulting in more successful products and software companies. While some of what follows may read as conventional wisdom, I found that conventional wisdom and conventional practice are often disconnected.

Implication 1 – Organisational Design

There is a significant difference between a product team member feeling a primary sense of affiliation to a product, being a full member of the product team, while bringing his/her particular competencies to that team versus an individual feeling primarily affiliated with their functional group or department in the company and being assigned to bring his/her skills to a particular team. For example, what T-shirt would your back-end software engineer wear - a functional group T-shirt with a specific product badge on it, or specific product team T-shirt, possibly with badge indicating his/her primary competency on the team (Fuller, 2019a)? While the words may sound subtle the impact is significant.

Recommendation: eliminate (at least minimise) any ambiguity a team member may have regarding primary affiliation. This is important in order to create cohesive teams that have minimal intra-team deference across functions, higher bandwidth communications, and more unified goal setting, all leading to greater creativity and innovation. There are other ways to address topics such as knowledge management, functional leadership, and career

development without organising in such a way that the individual feels his/her primary affiliation toward a functional grouping rather than toward the product and product team.

Implication 2 – Team Membership

Changes in team membership are sometimes necessary for a variety of reasons (skills adjustment, interpersonal considerations, etc.), however unnecessary team membership changes come with a price. Having individuals and, by extension, the entire team being committed to the long-term vision and plan for the product is in part dependent upon an expectation that each team member is on the team for the long term.

Recommendation: make changes in team composition only when necessary and when the benefit to the team justifies the inevitable price the team will pay for the change.

Implication 3 – Team Mandate

Work groups are content to consume a short-term work plan. Conversely, cohesive and healthy product teams want to own something and be committed to it for the long term. These teams will define ownable boundaries for themselves if those boundaries are not already specified.

Recommendation: provide the product development teams with unambiguous ownership of a product or subset of a product. Define that ownership in terms that are as broad and meaningful as possible, something that the team can feel inspired by and proud to own.

Implication 4 – Collective Grokking of the Product Domain

Cohesive teams also want to deeply *understand* what they own. If they are not being spoon-fed tasks or having the product domain interpreted for them, they will try to collectively grok it themselves.

Recommendation: expect the product development teams to deeply understand the product domain and ensure that the teams are explicitly aware of this expectation. Minimise the amount of domain *interpretation for the team* that occurs elsewhere in the organisation and then communicated to the team. Encourage *direct acquisition of domain knowledge by the product development teams*.

Implication 5 – Team involvement in Product Visioning & Planning

Much is lost when the product development teams do not participate in the higher-level product visioning and planning, determining the *what* and *why* behind the requirements. Teams that are less involved in these activities, make more assumptions (many of them tacit), create more unintended complexity, and take more time. In short, run a greater risk that the work product will fail to meet the potential outcome. Additionally, this limited involvement results in an erosion of deep ownership and two-dimensional software product development. Product development teams that do not own (or do not meaningfully participate) in product visioning and planning are less likely to take collective responsibility for product success or failure.

Recommendation: give the product development teams as much ownership as possible for product visioning and planning. At the very least have them deeply involved in all key product visioning and planning activities.

Chapter 6: Conclusion

This final chapter summarises the conclusions reached in this study – Blurring Boundaries, Towards the Collective Team Grokking of Software Product Requirements. I start with Section 6.1 which summarises the research, then Section 6.2 provides a brief description of the research conclusions. Section 6.3 discusses the research contributions both to industrial practice and to academic research, situated in the context of current gaps in requirements engineering research. Finally, in Section 6.4, I offer some thoughts regarding future research possibilities emerging from this study.

6.1 Research Summary

At the outset, this research aimed to generate substantive theory about "what factors support or impede cross-functional software product teams in collectively achieving a deep and shared understanding of the product domain?". This goal is important because the success of software products, and indeed the success of the software companies themselves, is impacted, often determined, by how well software product development teams deeply understand the context of the product requirements. Software product development practitioners have no theories to guide them in this regard.

The conclusions reached are the result of a Constructivist Grounded Theory method study that involved 18 development teams in 7 software product companies over a period of 3 years. The research findings are based on detailed notes and reflections from team observation sessions,

transcripts of individual interviews, and reflective researcher memos. A detailed discussion in Chapter 5 analysed the theory, which emerged from the data, and positioned it within extant literature, primarily requirements engineering, software product development process, and organisational scholarship. In addition to identifying factors that addressed the research question, the findings also provide insights into the underlying cause of these factors which led to reflections on implications for industrial practice. Several topics of possible future research also emerged from the analysis and reflection.

6.2 Research Conclusions

Addressing the research question, I found factors impacting CFPTs' internal dynamics, ownership of their product, time horizon of interest, and level of collective team grokking achieved by the team of the product domain. Two factors were identified that have significant impact on these product teams, 1) the organisational structure surrounding the team and, 2) the team's role in the product planning process. There is a common thread of *boundaries* between teams/groups/individuals appearing throughout the analysis and ***blurring boundaries*** is presented as a unifying theme that describes what teams do, in various contexts, in their effort to grok, i.e. to deeply understand the context of product requirements by imaginatively stepping into that other domain. Using a metaphorical technique of ***broadening the lens***, this purposeful blurring of boundaries occurs, to varying degrees, within the teams, between the team and the product planning process, and between the team and the product domain. Other relevant and useful conceptual categories appearing frequently throughout the analysis involved ***team cohesion***, ***horizon of interest*** for individuals and teams, propensity of teams to ***want to own***,

functional deference within and between teams, and evidence of ***striving to grok*** the product domain on the part of both individuals and teams.

I found a direct relationship between the strength of the mechanistic paradigm behind the design of these two influencing factors and the strength of the influence the factors had in inhibiting the ability of cross-functional software product teams to grok. As a result, I concluded that the *as-machine* metaphor for organisation and process models in software product organisations is contraindicated due to the negative effect they have on creativity and innovation, elements which are vital for the success of software product companies.

6.3 Research Contributions

This research offers contributions to both requirements engineering researchers as well as to software product industry practitioners.

Requirements engineering research has identified widespread dissatisfaction in industry in the area of RE while noting also that many that express dissatisfaction claim to be following widely accepted requirements engineering practices. Research has also recognised that the discipline of requirements engineering is highly human-based and the literature notes that there is yet no solid empirical basis for generalisations in requirements engineering that take these human factors into account.

My research contributes to both these gaps in requirements engineering research. It offers insights into one of the critical causes of incomplete/missing requirements being the major cause of software project failure, namely that cross-functional software product development teams themselves are critical for project success, not simply the methods followed and, further, that these teams require a deep understanding of the *context* of the requirements. The context for product requirements is the market domain, not simply the customer domain. Extant requirements engineering methods alone are not (and I argue, will not be) sufficient to satisfy this need. Further, there are often organisational and process impediments constraining the teams' potential to obtain context for the requirements. This research specifically identifies two factors that inhibit or enable the teams to grok the product domain.

Contribution is made also to requirements engineering research by providing some generalisations in the discipline that consider the human factors involved, underscoring the emerging awareness of the importance of human factors in requirements engineering. Specifically, I emphasise that the practices of requirements engineering should be augmented to explicitly address the importance of the development team obtaining direct and deep domain knowledge.

I also offer important insights aimed at practitioners. Aspects of these results sit in critique of mechanistic views of organisational and process models commonly found in many technology companies that implement structures and processes which specifically encourage specialisation and clearly defined segregation of duties. This specialisation is often intended to support knowledge management, ensure specialised functional expertise in senior leadership, support

career advancement for highly specialised roles, provide focus to meet deadlines, and to fit with adopted process models. These are all examples of the *as-machine* metaphor. Many senior leaders, within this study and outside of it, have an instinctive belief that productivity and innovation is maximised through specialisation, yet the evidence from this study strongly questions this. Rather, it shows that, for software product organisations, both productivity and innovation are impeded by these hardened, clearly defined, and specialised boundaries in organisational structure and process methodology.

In conclusion, I assert that the multi-disciplinary creativity and innovation necessary to create successful software products in a complex and uncertain world calls for a rethinking of the software product development organisation, both in structure and process. This rethinking would place a greater emphasis on facilitating the potential for product development teams to achieve a deep domain understanding and obtain their own collective context of the product requirements.

The rethinking needed is not simply to make changes to organisation structure and the methods of the product development process. In light of how product success is impacted by the degree to which the product development teams achieve understanding and context for the product, this rethinking is a mindset change, almost a paradigm shift. A mindset change fundamental enough to effect a softening of differences, ***blurring boundaries …*** the antithesis of the mechanistic *as-machine* metaphoric model way of thinking that is so common today in software product development organisations.

6.4 Future Work

This research shows that a CFPT's success towards ***blurring boundaries*** within the team, within the company, and between the team and its product domain generally defines the team's potential to collectively grok. The study shows that, at the very least, this potential to do so and do so *collectively* is influenced by the organisational and product planning process models in the company. However, I acknowledge that there may well be additional factors at play that call for further exploration. For example, I note that the general topic of non-co-located teams (and, teams of teams) has received, and continues to receive, significant research attention. Writing this in the midst of the COVID-19 crisis and being acutely aware that the topic of working location is more omnipresent than ever and appears to be a movement that will persist post-pandemic, it is clear that fully co-located vs partially distributed vs fully distributed CFPTs is a topic that calls for even further examination. Particularly, as it relates to the topics of this research, namely *collective* innovation, creativity, and domain grokking.

There is also further exploration needed to better understand the factors at play regarding how a cross-functional software product team defines its own success. I observed empowered teams being committed to the long-term success of the product while teams closer to the assembly of experts model tended to shrink the boundaries of ownership until they had reached a space for which they could claim ownership. It is unclear if this simply reflects a difference in the scope of permitted ownership or if these two team models define success using more fundamentally different considerations. It is also unclear whether there are significant differences in the sustainability of the sense of ownership between these types of teams.

My findings question the drive for deterministic structure and process when dealing with an opaque problem domain full of ambiguity, tacit knowledge, and rapid change. This research then raises questions regarding education and training. 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? Do current educational curricula adequately introduce and develop these skills?

Questions remain relating to the collective knowledge of software product development teams. For example, how does that which cannot be easily observed nor even expressed (e.g. tacit knowledge of the context of the requirements) be equally *understood, preserved,* and *transferred* within a team? Similarly, how does empathic appreciation of the context of a software solution translate across individuals, organisations, business domains, and cultures?

Several scope-related topics remain after this study. One, I believe that these findings may not be applicable where the software is being developed as a bespoke solution which is more transactional in nature and with a single customer's needs are to be satisfied rather than that of a market. Where the success of which is less dependent on the development team's understanding of the more tacit aspects of the requirements and their context, and where techniques of requirements validation and acceptance criteria may be more appropriate, even sufficient, for the *project* to meet the functional requirements/requests. Future research might explore this assertion.

Another applicability statement made is that the findings are likely to apply to technology-related product development other than software. The sampling strategy included software products only so this could be a topic for future research.

I found evidence that striving to be rational and deterministic is unquestioned in some firms. With statements from some senior executives in the study that the organisational model and inter-departmental mandates were beyond his/her individual influence, it appears that, in many organisations, the *as-machine* metaphor may be very deeply rooted and, therefore, not easily changed. De Alencar (2012) wrote that "despite an awareness of the need for creativity and innovation for organizational success, deep rooted tendencies to maintain the status quo prevailed, making it difficult to introduce changes in a direction of promising conditions for creativity". With creativity and innovation being imperatives for the success of software product companies, further research is required to explore how to create more awareness of the connection between creativity and innovation and the need for change in the organisational and planning process models.

Finally, while I observed teams using the ***broadening the lens*** mechanism towards ***blurring boundaries*** between the team and the product domain, there remains an important and challenging question, the question that started this quest at the beginning, which is why, *even in the same organisational, process, and leadership environment*, some software product teams observably achieve deeper grokking of the product domain than do others. I observed this phenomenon in my professional experience and peer discussions corroborated the awareness and importance of this question. I believe this remains a very important area of future inquiry for both academia and industry.

6.5 Epilogue

"A human being should be able to change a diaper, plan an invasion, butcher a hog, conn a ship, design a building, write a sonnet, balance accounts, build a wall, set a bone, comfort the dying, take orders, give orders, cooperate, act alone, solve equations, analyse a new problem, pitch manure, program a computer, cook a tasty meal, fight efficiently, die gallantly. Specialization is for insects."

— *Robert A. Heinlein (1988, p. 248)*

References

Adolph, S., Hall, W., & Kruchten, P. (2011). Using grounded theory to study the experience of software development. *Empirical Software Engineering*, 16(4), 487–513.

Agile Alliance. (2001). *Manifesto for Agile Software Development*. http://www.agilemanifesto.org/

Allan, G. (2003). A critique of using grounded theory as a research method. *Electronic Journal of Business Research Methods*, *2*(1).

Amjera, S., & Maynard, C. (2015). *Remaining Competitive in the Technology Industry Report 132669d-G*. https://home.kpmg/content/dam/kpmg/pdf/2015/10/remaining-competitive.pdf

Ancona, D. G., & Caldwell, D. F. (2008). Demography and Design: Predictors of New Product Team Performance. *Organization Science*, 3(3), 321–341. https://doi.org/10.1287/orsc.3.3.321

Awa, K. I. (2016). Functional structure and operational issues: An examination of core challenges and remedies. *IOSR Journal of Business and Management Ver. III*. https://doi.org/10.9790/487X-18130104

Beck, K., & Andres, C. (2004). *Extreme Programming Explained: Embrace Change*. XP Series (2nd ed.). Boston: Addison-Wesley Professional.

Beck, K. (2007). Don't just DWTTY. *IEEE Software*, *24*, 63–64.

Berry, D. M. (1995). The importance of ignorance in requirements engineering. *The Journal of Systems and Software*, 28(2), 179–184. https://doi.org/10.1016/0164-1212(94)00054-Q

Boehm, B. (2000). Requirements that handle IKIWISI, COTS, and rapid change. *Computer*, 33(7), 99–102. https://doi.org/10.1109/2.869384

Charmaz, K. (1990). "Discovering" chronic illness: Using grounded theory. *Social Science and Medicine*, *30*(11). https://doi.org/10.1016/0277-9536(90)90256-R

Charmaz, K. (2000). Constructivist and objectivist grounded theory. In *Handbook of qualitative research (2nd ed.)*, 509–535.

Charmaz, K. (2014). *Constructing grounded theory.* (2nd ed.). London: SAGE Publications.

Cohn, M. (2004). *User Stories Applied: For Agile Software Development*. Boston: Addison-Wesley Professional. https://doi.org/10.1017/CBO9781107415324.004

Coplien, J. O., & Harrison, N. (2005). *Organizational Patterns of Agile Software Development*. Upper Saddle River, NJ: Pearson Prentice Hall.

Coghlan, D., & Brydon-Miller, M. (2014). *The SAGE Encyclopedia of Action Research*. *The SAGE Encyclopedia of Action Research*. London: SAGE Publications Ltd. https://doi.org/10.4135/9781446294406

Creswell, J. W. (2012). *Educational research: Planning, conducting, and evaluating quantitative and qualitative research*. *Educational Research* (4th ed., Vol. 4). Boston: Pearson. https://doi.org/10.1017/CBO9781107415324.004

de Alencar, Eunice M.L. Soriano. (2012). Creativity in Organizations: Facilitators and Inhibitors. *Handbook of Organizational Creativity*, Michael D. Mumford, Ed, London: Academic Press. 87-111. https://doi.org/10.1016/B978-0-12-374714-3.00005-7

Dong, Y., Dong, H., & Yuan, S. (2018). Empathy in design: A historical and cross-disciplinary perspective. In C. Baldwin (Ed.), *Advances in Intelligent Systems and Computing*. Vol. 586. 295–304. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-60642-2_28

Edmondson, A. C., & Lei, Z. (2014). Psychological Safety: The History, Renaissance, and Future of an Interpersonal Construct. *Annual Review of Organizational Psychology and Organizational Behavior*, 1(1), 23–43. https://doi.org/10.1146/annurev-orgpsych-031413-091305

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

Evernote [Computer Software]. (2018). https://www.evernote.com

Ferrari, A., Spoletini, P., & Gnesi, S. (2016). Ambiguity and tacit knowledge in requirements elicitation interviews. *Requirements Engineering*, *21*(3), 333–355. https://doi.org/10.1007/s00766-016-0249-3

Fernández, D. M., Wagner, S., Kalinowski, M., Felderer, M., Mafra, P., Vetrò, A., … Wieringa, R. (2017). Naming the pain in requirements engineering: Contemporary problems, causes, and effects in practice. *Empirical Software Engineering*, *22*(5), 2298–2338. https://doi.org/10.1007/s10664-016-9451-7

Fishburne, Tom (2008). Corporate Telephone. [Cartoon]. https://marketoonist.com/?s=telephone+game

Fuller, R. C. (2019a). What T-shirt Are You Wearing? Towards the Collective Team Grokking of Product Requirements. *The Fifth International Conference on Advances and Trends in Software Engineering* (SOFTENG 2019), 37–40. Valencia, Spain. https://www.thinkmind.org/index.php?view=article&articleid=softeng_2019_3_10_90034

Fuller, R. C. (2019b). Functional organization of software groups considered harmful. *Proceedings - 2019 IEEE/ACM International Conference on Software and System Processes, ICSSP 2019.* 120–124. Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/ICSSP.2019.00024

Gladstein, D. L. (2006). Groups in Context: A Model of Task Group Effectiveness. *Administrative Science Quarterly*, 29(4), 499. https://doi.org/10.2307/2392936

Glaser, B. G., & Strauss, A. L. (1967). *The Discovery of Grounded Theory: Strategies for Qualitative Research*. *Observations* (Vol. 1). London: Aldine Transaction. https://doi.org/10.2307/2575405

Glaser, B. G. (1992). *Emergence Vs Forcing: Basics of Grounded Theory Analysis*. Sociology Press.

Goel, V. (1995). Sketches of Thought. In *Sketches of Thought*. MIT Press. https://doi.org/10.7551/mitpress/6270.001.0001

Google, *"re:Work Guide: Understanding team effectiveness*. (2020) https://rework.withgoogle.com/print/guides/5721312655835136

Gordon-Finlayson, A. (2010). QM2: Grounded Theory. In M. Forrester (Ed.), *Doing Qualitative Research in Psychology: A Practical Guide* (pp. 154–176). London, UK: Sage.

Grok. (1989). In *Oxford English Dictionary*. (2nd ed.). Oxford: Oxford University Press.

Guinan, P. J., Cooprider, J. G., & Faraj, S. (1998). Enabling Software Development Team Performance during Requirements Definition: A Behavioral Versus Technical Approach. *Information Systems Research*, 9(2), 101–125. https://doi.org/10.1287/isre.9.2.101

Hackman, J. R., & Oldham, G. R. (1980). Work Redesign. Reading, Mass.: Addison-Wesley.

Halliwell, S. (1988). *Plato: Republic X*. Liverpool: Liverpool University Press. https://doi:10.2307/j.ctv1228h5k

Heinlein, R. A. (1961). Stranger in a Strange Land. New York: Penguin Books.

Heinlein, R. A. (1988). *Time Enough for Love*. New York: Ace Books.

Highsmith, J. (2000). *Adaptive Software Development*. New York: Dorset House Publishing.

Highsmith, J. (2001). Order for Free: An Organic Model for Adaptation. In L. L. Constantine (Ed.), *Beyond Chaos: The Expert Edge in Managing Software Development* (pp. 251–257). Addison-Wesley. https://doi.org/10.1145/379391

Hoda, R., & Noble, J. (2017). Becoming Agile: A Grounded Theory of Agile Transitions in Practice. In *Proceedings - 2017 IEEE/ACM 39th International Conference on Software Engineering, ICSE 2017*. https://doi.org/10.1109/ICSE.2017.21

Hoegl, M., & Gemuenden, H. G. (2001). Teamwork Quality and the Success of Innovative Projects: A Theoretical Concept and Empirical Evidence. *Organization Science*, 12(4), 435–449. https://doi.org/10.1287/orsc.12.4.435.10635

Holston, D. (2011). *The Strategic Designer*. Cincinnati: HOW Books.

Hong, J. (1999). Structuring for organizational learning. *The Learning Organization*, *6*(4), 173–186. https://doi.org/10.1108/09696479910280631

Johnston, D. L. (1978). Scientists Become Managers-The 'T'-Shaped Man. *IEEE Engineering Management Review*, 6(3). https://doi.org/10.1109/EMR.1978.4306682

Jones, P. (2015). Sensemaking methodology: A liberation theory of communicative agency. https://www.epicpeople.org/sensemaking-methodology/

Kelly, W. (1972). *Pogo : we have met the enemy and he is us*. New York: Simon and Schuster.

Klein, G., Moon, B., Hoffman, R., & Associates, K. (2006). Making Sense of Sensemaking 2: A Macrocognitive Model. *IEEE Intelligent Systems*, 21(5), 88–92. https://doi.org/10.1109/MIS.2006.100

Kolko, J. (2010). Sensemaking and Framing: A Theoretical Reflection on Perspective in Design Synthesis. *2010 Design Research Society Conference*, 1–9. http://www.jonkolko.com/writingSensemaking.php

Kolko, J. (2014). *Well-Designed: How to create empathy to create products people love*. Harvard Business Review Press.

Koppen, E., & Meinel, C. (2012). Knowing People: The Empathetic Designer. *Design Philosophy Papers*, 10(1), 35–51. https://doi.org/10.2752/089279312X13968781797553

Kotter, J. P. (2012). Accelerate! In *Harvard Business Review* (Vol. 90, Issue 11). https://doi.org/10.2307/j.ctvj7wnss.11

Kouprie, M., & Visser, F. S. (2009). A framework for empathy in design: Stepping into and out of the user's life. *Journal of Engineering Design*, 20(5), 437–448. https://doi.org/10.1080/09544820902875033

Krznaric, R. (2014). *Empathy: why it matters, and how to get it*. New York: Penguin Random House.

Kuhn, T. S. (2012). *The Structure of Scientific Revolutions* (4th ed.). University of Chicago Press.

Kurtz, C. F., & Snowden, D. (2003). The New Dynamics of Strategy: Sense-making in a Complex-Complicated World. *IBM Systems Journal*, 42(3), 462–483. https://doi.org/10.1109/EMR.2003.24944

Lakoff, G., & Johnon, M. (1980). *Metaphors We Live By*. Chicago: University of Chicago Press.

Mafra, P., Kalinowski, M., Fernández, D. M., Felderer, M., & Wagner, S. (2016). Towards Guidelines for Preventing Critical Requirements Engineering Problems. In *2016 42th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. https://doi.org/10.1109/SEAA.2016.50

March, J. G. (2010). *The Ambiguities of Experience*. *The Ambiguities of Experience*. Ithaca, NY: Cornell University Press. https://doi.org/10.7591/9780801459016

Maxwell, J. A. (2012). *Qualitative research design: An interactive approach*. Vol. 41. Thousand Oaks, Calif.: SAGE Publications.

Mendeley [Computer Software]. (2018). https://www.mendeley.com

Miles, M., & Huberman, M. (1994). *Qualitative Data Analysis, 2nd ed.* SAGE Publications.

Mitchell, R., Boyle, B., & Nicholas, S. (2009). The impact of goal structure in Team knowledge creation. *Group Processes and Intergroup Relations*, 12(5). https://doi.org/10.1177/1368430209340568

Mohanani, R., Ralph, P., & Shreeve, B. (2014). Requirements Fixation. In *Proceedings of the 36th International Conference on Software Engineering*, 895–906. New York, NY, USA: ACM. https://doi.org/10.1145/2568225.2568235

Morgan, G. (1997). *Imaginization: New Mindsets for Seeing, Organizing, and Managing*. Newbury Park, Calif.: SAGE Publications.

Morgan, G. (2006). *Images of Organization* (Updated ed). Thousand Oaks: SAGE Publications.

Morse, J. M. (2009). *Developing grounded theory. The second generation*. *Developing Qualitative Inquiry*. https://doi.org/10.1177/1049732307301611

NaPiRE: Naming the Pain in Requirements Engineering. (2020). http://napire.org

Naumer, C., Fisher, K., & Dervin, B. (2008). Sense-Making: a methodological perspective. *CHI2008 Workshop on SenseMaking*, Florence. https://doi.org/10.1002/asi.20690

NVivo [Computer Software]. (2018). https://www.qsrinternational.com/nvivo-qualitative-data-analysis-software/about/nvivo

Okely, J. (2010). Thinking through fieldwork. In *Analyzing qualitative data* (pp. 18–34). Taylor & Francis. https://doi.org/10.4324/9780203413081_chapter_1

Polanyi, M. (2009). *The tacit dimension*. Chicago: University of Chicago Press.

Postma, C., Zwartkruis-Pelgrim, E., Daemen, E., & Du, J. (2012). Challenges of Doing Empathic Design: Experiences from Industry. *International Journal of Design*; Vol 6, No 1 (2012), 59–70. http://www.ijdesign.org/ojs/index.php/IJDesign/article/view/1008/403

Ralph, P. (2013). The illusion of requirements in software development. *Requirements Engineering*, 18(3), 293–296. https://doi.org/10.1007/s00766-012-0161-4

Ralph, P., & Mohanani, R. (2015). Is Requirements Engineering Inherently Counterproductive? *Proceedings - 5th International Workshop on the Twin Peaks of Requirements and Architecture, TwinPeaks 2015*. https://doi.org/10.1109/TwinPeaks.2015.12

Reiter-Palmon, R., Wigert, B., & de Vreede, T. (2012). Team creativity and innovation: The effect of group composition, social processes, and cognition. *Handbook of Organizational Creativity*. 295–326. London: Academic Press [Imprint]. https://doi.org/10.1016/B978-0-12-374714-3.00013-6

Reiter-Palmon, R. & Harms, M. (2017). Team Creativity and Innovation. *Team Creativity and Innovation*. Oxford University Press. https://doi.org/10.1093/oso/9780190222093.001.0001

Ries, E. (2014). *The lean startup : how today's entrepreneurs use continuous innovation to create radically successful businesses*. New York: Crown Publishing Group.

Russell, D., & Pirolli, P. (2009). An Overview of Sensemaking: A View from the Workshop CHI 2009. *Sensemaking Workshop at CHI*. http://sites.google.com/site/testdmr/chi-2009-sensemaking-submissions/3-SM-workshop-intro-Russell-Pirolli.pdf

Ryan, G. W., & Bernard, H. R. (2003). Techniques to Identify Themes in Qualitative Data. *Field Methods*, 15(1), 85–109. https://doi.org/10.1177/1525822X02239569

Saldaña, J. (2013). *The Coding Manual for Qualitative Researchers (2nd Ed.). SAGE Publications Inc.* (Second Ed.). London: SAGE Publications. https://doi.org/10.1017/CBO9781107415324.004

Santos, F. M., & Eisenhardt, K. M. (2005). Organizational Boundaries and Theories of Organization. *Organization Science*, *16*(5), 491–508. https://doi.org/10.1287/orsc.1050.0152

Schön, E. M., Winter, D., Escalona, M. J., & Thomaschewski, J. (2017). Key challenges in agile requirements engineering. *Lecture Notes in Business Information Processing*. https://doi.org/10.1007/978-3-319-57633-6_3

Schwaber, K. (2004). *Agile Project Management with Scrum.* Redmond, WA: Microsoft Press.

Sedano, T., Ralph, P., & Peraire, C. (2019). The Product Backlog. In *Proceedings - International Conference on Software Engineering*. https://doi.org/10.1109/ICSE.2019.00036

Senge, P. M. (2006). *The Fifth Discipline: The Art and Practice of the Learning Organization* (Revised Ed). New York: Crown Publishing Group.

Smith, E. R., Seger, C. R., & Mackie, D. M. (2007). Can Emotions Be Truly Group Level? Evidence Regarding Four Conceptual Criteria. *Journal of Personality and Social Psychology*. https://doi.org/10.1037/0022-3514.93.3.431

Stol, K.-J., Ralph, P., & Fitzgerald, B. (2016). Grounded theory in software engineering research: A critical review and guidelines. *Proceedings - International Conference on Software Engineering.* (pp. 120–131. https://doi.org/10.1145/2884781.2884833

Strauss, A., & Corbin, J. (2008). *Basics of Qualitative Research (3rd ed.): Techniques and Procedures for Developing Grounded Theory*. *Basics of Qualitative Research (3rd ed.): Techniques and Procedures for Developing Grounded Theory* (3rd ed.). Los Angeles: SAGE Publications. https://doi.org/10.4135/9781452230153

Stray, V., Sjøberg, D. I. K., & Dybå, T. (2016). The daily stand-up meeting: A grounded theory study. *Journal of Systems and Software*, 114. https://doi.org/10.1016/j.jss.2016.01.004

Tuckman, B. W. (1965). Developmental sequence in small groups. Psychological Bulletin, 63(6), 384–399. https://doi.org/10.1037/h0022100

van Rijn, H., Visser, F. S., Stappers, P. J., & Özakar, A. D. (2011). Achieving empathy with users: The effects of different sources of information. *CoDesign*, 7(2), 65–77. https://doi.org/10.1080/15710882.2011.609889

Wagner, S., Fernandez, D. M., Felderer, M., Vetrò, A., Kalinowski, M., Wieringa, R., Pfahl, D., Conte, T., Christiansson, M. T., Greer, D., Lassenius, C., Männistö, T., Nayebi, Oivo, M., Penzenstadler, B., Prikladnicki, R., Ruhe, G., Schekelmann, A., ...Winkler, D. (2019). Status quo in requirements engineering: A theory and a global family of surveys. *ACM Transactions on Software Engineering and Methodology*, 28(2). https://doi.org/10.1145/3306607

Wasserman, A. I. (2011). How the Internet transformed the software industry. *Journal of Internet Services and Applications*, 2(1), 11–22. https://doi.org/10.1007/s13174-011-0019-x

Weick, K. E., Sutcliffe, K. M., & Obstfeld, D. (2005). Organizing and the process of sensemaking. *Organization Science*, *16*(4), 409–421. https://doi.org/10.1287/orsc.1050.0133

Wieringa, R. (2014). *Design Science Methodology for Information Systems and Software Engineering*. Springer Berlin Heidelberg. https://doi.org/10.1145/1810295.1810446

Wolf, G. (1996, February). Steve Jobs: The Next Insanely Great Thing. *Wired Magazine*, (February), 1–10. http://www.wired.com/wired/archive//4.02/jobs_pr.html

Woodcock, A., McDonagh, D., Osmond, J., & Scott, W. (2018). Empathy, Design and Human Factors. *Advances in Usability and User Experience*. T. Ahram & C. Falcão (Eds.). 569–579. Cham: Springer International Publishing.

Woolery, E. (2011). *Design Thinking Handbook*. inVISION. https://www.designbetter.co/design-thinking

Woolley, A. W., Chabris, C. F., Pentland, A., Hashmi, N., & Malone, T. W. (2010). Evidence for a collective intelligence factor in the performance of human groups. *Science*, 330(6004), 686–688. https://doi.org/10.1126/science.1193147

Appendices

Appendix A - Organisation Consent Form

Organisation Consent Form

Inquiry into Requirements Understanding by Software Development Teams

I. Who is conducting the study?

Principal Investigator: Philippe Kruchten, PhD
Professor, Electrical and Computer Engineering
UBC
(604) xxx-xxxx

Co-Investigator: Rob Fuller
PhD candidate, Electrical and Computer Engineering
UBC
(604) xxx-xxxx
For partial fulfillment of PhD dissertation.

II. Why are we doing this study?

We want to learn more about the critical success factors that determine how well multi-disciplinary software product teams come to collectively understand the context for which their products are intended.
Your organisation has been invited to participate because of its current experience with software development teams and because we believe it can provide a perspective that will be very helpful for this study.

III. How is the study done?

Overall, the study will consist of direct interviews with individuals and direct observations of teams in action. The interviews will be individuals that have direct experience, in any one of a number of functional capacities, in software development teams. Consent for the interviews will be obtained by the individuals being interviewed. Direct observations of teams in action will be for the purpose of deepening the understanding of team dynamics, consent for which is being sought from you, as an authorised officer of your organisation.
The research method is very iterative with data gathered, analysed, more data gathered, analysed, etc.

IV. What happens if you agree?

Over a period of time, we will arrange for the co-investigator to passively observe several team meetings and interactions. These would typically be team meetings held for the purposes of product planning and/or feature prioritisation.

V. What happens to the study results?

The primary motivation for the study is to discover actionable insights so, once the dissertation is complete, the results will be published and also made available upon request to any that participated in the study. If you wish to be contacted directly when the results are complete, please provide your email or mailing address when you sign this consent form.

VI. Is there any way being in this study could be bad for the organisation?

No, we can't imagine that there is anything in this study that could negatively affect your organisation nor any individual.

VII. What are the benefits of participating?

Whether you experience any benefit from the participation itself is hard to predict and will be entirely a personal or organisational experience. The results of the study, however, could well be of benefit to yourself and others in the software development profession.

VIII. CONFIDENTIALITY

This is something we take very seriously. The anonymity of the organisation, its business activities and plans, as well as every individual will be respected. Information that discloses your identity, even indirectly, will not be released. All raw data (notes taken) will be stored on a single, highly secured computer that will have participants (organisation and individual) identified by code number only. Raw data will be kept for 5 years after the publication of the results, for auditability purposes, after which it will be electronically cleansed. There will be no secondary use of the data.

IX. Who can you contact if you have questions about the study

If you have any questions or concerns about what we are asking of you, please contact the study leader or one of the study staff. The names and telephone numbers are listed at the top of the first page of this form.

X. Who can you contact if you have complaints or concerns about the study

If you have any concerns or complaints about your rights as a research participant and/or your experiences while participating in this study, contact the Research Participant Complaint Line in the UBC Office of Research Ethics at 604-822-8598 or if long distance e-mail RSIL@ors.ubc.ca or call toll free 1-877-822-8598.

XI. ORGANISATION CONSENT AND SIGNATURE PAGE

Having your organisation take part in this study is entirely up to you. You have the right to refuse to participate in this study. If you decide to take part, you may choose to pull out of the study at any time without giving a reason and without any negative impact.

Your signature below indicates that you have received a copy of this consent form for your own records.

Your signature indicates that you consent to have your organisation participate in this study.

_____

Participant Signature Date

_____

Printed Name of the Participant signing above

_____

Title of the Participant signing above

_____

Organisation name

Appendix B - Individual Consent Form

Individual Consent Form

Inquiry into Requirements Understanding by Software Development Teams

I. Who is conducting the study?

Principal Investigator: Philippe Kruchten, PhD
Professor, Electrical and Computer Engineering
UBC
(604) xxx-xxxx

Co-Investigator: Rob Fuller
PhD candidate, Electrical and Computer Engineering
UBC
(604) xxx-xxxx

For partial fulfilment of PhD dissertation.

II. Why are we doing this study?

We want to learn more about the critical success factors that determine how well multi-disciplinary software product teams come to collectively understand the context for which their products are intended.
You have been invited to participate because of your current and/or past experience with software development teams and because we feel your perspective from that experience will be very helpful for this study.

III. How is the study done?

Overall, the study will consist of direct interviews with people such as yourself that have direct experience, in any one of a number of functional capacities, in software development teams. These interviews will be supplemented in select circumstances, with direct observations of

teams in action. Consent for the latter is obtained by authorised officers of the organisation selected.

The research method is very iterative with data gathered, analysed, more data gathered, analysed, etc. Based on the analysis, there could be occasions where you may be asked to participate in a brief follow-up interview for which you will be offered this same opportunity to agree or decline.

IV. What happens if you agree?

We will arrange to meet one-on-one for a casual discussion on this topic. This will be at a time and location suitable to you. This meeting would typically last for approximately one hour.

We will ask for your permission at the time to agree to the audio recording of the meeting. This will allow for a much more casual discussion by eliminating the need for extensive note-taking. Please note the commitment below in the Confidentiality section.

V. What happens to the study results?

The primary motivation for the study is to discover actionable insights so, once the dissertation is complete, the results will be published and also made available upon request to any that participated in the study. If you wish to be contacted directly when the results are complete, please provide your email or mailing address when you sign this consent form.

VI. Is there any way being in this study could be bad for you?

No, we can't imagine that there is anything in this study that could harm you or be bad for you.

VII. What are the benefits of participating?

Whether you experience any benefit from the participation itself is hard to predict and will be entirely a personal experience. The results of the study, however, could well be of benefit to yourself and others in the software development profession.

VIII. CONFIDENTIALITY

This is something we take very seriously. Your anonymity will be respected. Information that discloses your identity, even indirectly, will not be released. All raw data, including audio transcripts, will be stored on a single, highly secured computer that will have participants identified by code number only. Raw data will be kept for 5 years after the publication of the results, for auditability purposes, after which it will be electronically cleansed. There will be no secondary use of the data.

IX. Who can you contact if you have questions about the study

If you have any questions or concerns about what we are asking of you, please contact the study leader or one of the study staff. The names and telephone numbers are listed at the top of the first page of this form.

X. Who can you contact if you have complaints or concerns about the study

If you have any concerns or complaints about your rights as a research participant and/or your experiences while participating in this study, contact the Research Participant Complaint Line in the UBC Office of Research Ethics at 604-822-8598 or if long distance e-mail RSIL@ors.ubc.ca or call toll free 1-877-822-8598.

XI. PARTICIPANT CONSENT AND SIGNATURE PAGE

Taking part in this study is entirely up to you. You have the right to refuse to participate in this study. If you decide to take part, you may choose to pull out of the study at any time without giving a reason and without any negative impact.

Your signature below indicates that you have received a copy of this consent form for your own records.

Your signature indicates that you consent to participate in this study.

_____

Participant Signature Date

_____

Printed Name of the Participant signing above

Appendix C - Interview Guide

Interview Guide

Rob Fuller 06888713

Research Context

General research question – "how do empowered, multi-disciplinary product development teams develop and nurture a **collective** 'grokking' (empathic understanding) of the world for which their product/service is intended?"

Specific research question 1 – "how do multi-disciplinary software product development teams differentiate between what is expressed (asked for, wanted) versus what is truly needed?" **(requirements engineering 2.0)**

Specific research question 2 – "how do multi-disciplinary software product development teams assess their collective level of empathic understanding of their customer's world?" **(domain understanding & continual improvement)**

Specific research question 3 – "how does that which cannot be easily observed nor expressed be equally understood and preserved within a team?" (can there such a thing as **team memory** and, if so, what does it look like?)

Specific research question 4 – "how are technically-oriented people working in teams 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?" **(process reengineering)**

Focus for the research questions

- examine what empirical adaptations software development teams make to "traditional" requirements engineering practices and to common methods of user interaction design in order to support empathic-based software development towards a more complete and accurate understanding of the business needs.
- explore how these teams enlarge the definition of *user-experience* (UX) into an understanding of the entirety of their interaction with their working environment - the context their users operate within - including technology, culture, politics.
- examine how software development individuals and teams, who are trained and encouraged to apply their best judgement, suspend those judgements and opinions in order to connect with and exercise empathy for those for whom the solution is intended.

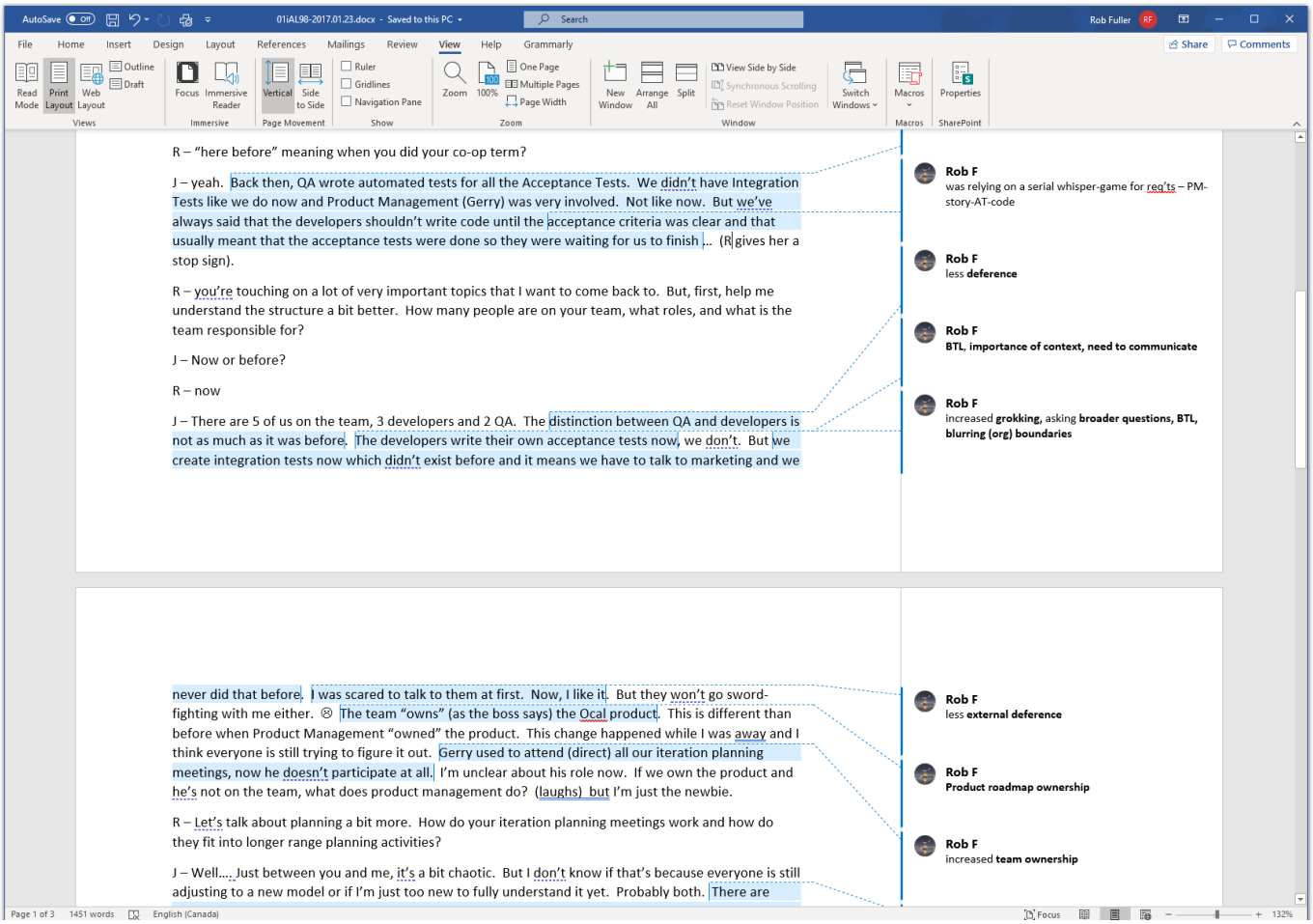
Interview Questions (to be adjusted when and as needed)

Set the scene before starting in on the interview questions. Establish whatever context is needed – address questions/concerns the participant may have at the outset, reiterate the research focus and areas the interview discussion will explore. Also, establish the background, training, and tenure of the participant. Initial questions create context then directed towards more detail on the development team planning process.

1. "Tell me a bit about your team, your role on that team, etc. Mostly I'm trying to get an organisational structure picture first and we can discuss the team planning and analyses processes".
2. "How many people are on your team, what roles are there on the team, and what is the team accountable for?"
3. "Let's talk about planning a bit. How do your iteration planning meetings work and how do they fit into longer range planning activities?"
4. "Describe an iteration planning session for me."
5. "How does the team determine what the general priority is for the coming iteration period, before you actually do the detailed planning?"
6. "What more can you tell me about the team as a 'team' … cohesion, sense of common purpose, etc.?"
7. "Please describe your entry to the team, the condition of the team at that time, and your 'ramp' to understand the world for which the product is targetted (the domain and its broader context)."
8. "Please describe how the longer-term product roadmap is created, maintained, communicated and how your team participates in any of those activities?"
9. "How does the team determine to what degree its product has met the true needs of the intended user/customer/stakeholder?"

"can you think of a time when?" is better than "give me an example"

Appendix D - Extract from Annotated Interview Transcription



R – "here before" meaning when you did your co-op term?

J – yeah. Back then, QA wrote automated tests for all the Acceptance Tests. We didn't have Integration Tests like we do now and Product Management (Gerry) was very involved. Not like now. But we've always said that the developers shouldn't write code until the acceptance criteria was clear and that usually meant that the acceptance tests were done so they were waiting for us to finish … (R gives her a stop sign).

R – you're touching on a lot of very important topics that I want to come back to. But, first, help me understand the structure a bit better. How many people are on your team, what roles, and what is the team responsible for?

J – Now or before?

R – now

J – There are 5 of us on the team, 3 developers and 2 QA. The distinction between QA and developers is not as much as it was before. The developers write their own acceptance tests now, we don't. But we create integration tests now which didn't exist before and it means we have to talk to marketing and we

Rob F
was relying on a serial whisper-game for req'ts – PM-story-AT-code

Rob F
less **deference**

Rob F
BTL, importance of context, need to communicate

Rob F
increased **grokking**, asking **broader questions, BTL, blurring (org) boundaries**

never did that before. I was scared to talk to them at first. Now, I like it. But they won't go sword-fighting with me either. ☹ The team "owns" (as the boss says) the Ocal product. This is different than before when Product Management "owned" the product. This change happened while I was away and I think everyone is still trying to figure it out. Gerry used to attend (direct) all our iteration planning meetings, now he doesn't participate at all. I'm unclear about his role now. If we own the product and he's not on the team, what does product management do? (laughs) but I'm just the newbie.

R – Let's talk about planning a bit more. How do your iteration planning meetings work and how do they fit into longer range planning activities?

J – Well…. Just between you and me, it's a bit chaotic. But I don't know if that's because everyone is still adjusting to a new model or if I'm just too new to fully understand it yet. Probably both. There are

Rob F
less **external deference**

Rob F
Product roadmap ownership

Rob F
increased **team ownership**