Lancaster University Management School

Lancaster University Management School:

Author Accepted Manuscript

This is an 'accepted manuscript' as required by HEFCE's Open Access policy for REF2021.

Please cite this paper as:
Williams, Richard A., (forthcoming), Conflict Propagation within large Technology and Software Engineering Programmes: A Multi-Partner Enterprise System Implementation as Case Study, IEEE Access

ACCEPTED FOR PUBLICATION | November 12, 2019

ORCID NUMBER: 0000-0001-6333-9448

DOI: [Insert DOI number if assigned on acceptance]

Dr Richard Alun Williams
Senior Lecturer in Management Science
Lancaster University Management School
Lancaster, LA1 4YX

TRIPLE-ACCREDITED, WORLD-RANKED

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2019.DOI

Conflict Propagation within Large Technology and Software Engineering Programmes: A Multi-Partner Enterprise System Implementation as Case Study

RICHARD A. WILLIAMS[1,2] (Senior Member, IEEE)
[1] Department of Management Science, Lancaster University, Lancaster, LA1 4YX, UK (e-mail: r.williams4@lancaster.ac.uk)
[2] Centre for Technological Futures, Lancaster University, Lancaster, LA1 4YW, UK

This work was part-funded by a SAMS-ESRC-UKCES Management and Business Development Fellowship under Grant ES/L002612/1.

ABSTRACT Enterprise system implementations are increasingly outsourced to multiple third-party service providers. These multi-partner technology and software engineering programmes are usually organized through project teams that align to the functional areas of the software. Cognitive, occupational and personal differences between members of project teams increases the potential for conflict, which in extreme cases may propagate throughout the entire social network of the programme. Using social network analysis and thematic coding analysis, within a single case study, new insights are provided into the development of conflict within and between individual project teams of large technology and software programmes, such as those seen within enterprise system implementations. A conceptual framework has been developed that builds on existing literature around conflict in groups, to explore how task, process and relationship conflict can develop in large enterprise system implementations. The conceptual framework illustrates how conflict, once developed, can propagate throughout the social network of the wider programme. Finally, we argue that high-conflict organizations, such as the temporary multi-partner implementation team that forms to deliver large technology and software programmes, have a tendency to contain competing networks, which actively promotes conflict. We conclude by setting the agenda for further research on how we may contain the spread of conflict once it has developed within technology and software engineering programme environments.

INDEX TERMS Conflict, Conceptual Framework, Enterprise System, Software Engineering Project, Project and Programme Management,

I. INTRODUCTION

IT is commonly known that the majority of failures within information systems development and implementation are not due to the technology, but are instead due to human and organizational reasons [1], [2]. This is compounded by the fact that the implementation and management of large software programmes often becomes the preserve of external service providers [3], whose employees may have different educational and training backgrounds, cognitive orientations, and social norms/cultures, with respect to the client's employees [4]. In fact, some of the larger software implementation programmes consist of hundreds of client and third-party team members, who may be located across multiple geographies and time zones, and structured into different project teams [5].

It has recently been argued that the increasing size and complexity of these technology and software engineering programmes, leads them to exhibit the behaviours and traits of complex systems [6]. The emergent behaviour generated within the system, may, to a large degree, be due to the complexity stemming from the large number of team members, and the growing trend, especially in the public sector, of using multiple vendors with back-to-back fixed-price contracts to implement the individual projects that combine to form the programme. Furthermore, the individual project teams may have competing priorities and objectives, which may lead to system-wide conflict, which emerges from the conflict within (intragroup) and between (intergroup) project teams.

There is a long history of applied research into technology and software engineering project management within the various IEEE Transactions and Conferences, that are focused

on software engineering (e.g. [7]), the project management required to successfully implement information technologies in organizations (e.g. [8], [9]), along with the social aspects of the implementation team environment (e.g. [10]–[12]). We build upon this expanding body of knowledge through focusing on the factors that contribute to conflict within the workplace social network of large multi-partner enterprise system implementations, which provides a contemporary example of a large technology and software engineering programme. A network lens to analyze projects has recently been advocated by Steen *et al.* [13], who suggest that a multi-method approach is required to investigate project network dynamics. We adopted the case study approach, and analyzed the results through qualitative data analysis, social network analysis, and diagrammatic modelling. Our findings suggest that the various project teams commence the programme using a shared vision and complementary aims and objectives, but that various external factors perturb the collegial working dynamics and *infect the system* with conflict.

In this article, we will start with a review of the literature on the social networks inherent to enterprise system implementations, the three different types of conflict (task, process, and relationship) and how they can develop within, and between, different teams, and the way that conflict can propagate throughout the social network of multi-partner enterprise system implementations. We then discuss the development and propagation of conflict within a case study that represents a large multi-partner technology and software engineering programme. We then propose an integrative conceptual framework that can serve as a means to develop a better understanding of the generation of conflict within the social network of such technology programmes. We conclude by setting the agenda for further research on how to mitigate the development of conflict, and how to dampen the conflict once it has developed in order to minimize propagation throughout the social network of team members.

II. RELATED WORK

This section provides an overview of the major concepts that contribute to the theory behind our study.

A. ENTERPRISE SYSTEMS AND THEIR IMPLEMENTATION

The term enterprise resource planning (ERP) was devised in 1990 by the Gartner Group [14] to define the high-level business processes that organizations used to manage their core administrative (back office) functions such as Human Resources, Payroll, Finance, etc. A number of software vendors, such as IBM, Oracle and SAP, offered ERP software packages to store the resulting data and automate a number of the business processes. With the expansion of Information Technology (IT) and Information Systems (IS) throughout the 1990s and 2000s, the software modules that were preconfigured to focus on back office functions were integrated together and also augmented with functionality for data mining, business intelligence, advanced planning, and automatically processing data from external supplier/customer relationships. These larger software systems were termed Enterprise Systems, and are now routinely used by organizations of all sizes, to help them manage their money, staff, products, customers and suppliers, more effectively [15].

Enterprise systems are designed to integrate the often-fragmented business processes and associated data relating to the separate organizational units of large organizations, into a single software system that facilitates the sharing of data and use of preconfigured (and standardized) software modules [16]. Therefore, enterprise system implementations should be viewed as organizational transformation projects that can lead to process improvement and innovation [17], and as such will impact all aspects of the organization, and not as large IT projects that can be implemented by the IT department in isolation [15]. Enterprise system implementations at large organizations are usually implemented by third-party service providers, with the largest implementations using the consultancy services of both the vendor (e.g. Oracle, SAP, etc) and professional services firms (e.g. Deloitte, Accenture, etc).

B. THE SOCIAL NETWORKS OF LARGE ENTERPRISE SYSTEM IMPLEMENTATIONS

Social Network is a term used to refer to a collection of people and the nature of the connections, relationships and interactions between them [18]. These relationships can represent any type of social behaviour, for example: friendship, cooperation, competition, aggression or hostility [19]. Taking inspiration from mathematical graph theory, the social network perspective views a social system as a set of interrelated nodes or actors (the people within the network) that are connected by one or more edges (the types of relationships/interactions between them) [20]. These networks are described as being either directed or undirected, depending on whether the connections between pairs of nodes are unidirectional or bidirectional, respectively [21].

Social networks promote communication between network members, and the network topology often facilitates the efficient and effective propagation of information [22]. A fundamental, and accepted principle, in network analysis is the presumption that nodes are not independent, but in fact influence each other through the relationships and interactions between them. This can occur through a variety of mechanisms, but the simplest and most common is that of direct transmission from one node to another node [23], and through successive transfers, can propagate within a network. Borgatti [24] conjectures that this propagation can be through either physical transfer of an item, or an imitative process of attitudes, behaviours or emotions. Crossley [25] defines social network analysis as a form of structural analysis that uses quantitative approaches (e.g. graphs, adjacency matrices, and descriptive statistics on the network topology) to provide a pragmatic means of recording and analyzing relational data that have been collected through qualitative methods.

As discussed by Barabasi *et al.* [26], most networks conform to a small world topology, which means that the average

number of intermediate connections between nodes (e.g. individual team members) is rather small. They further advise that networks within real-world systems display a degree of clustering that is higher than expected for random networks, which we believe is exemplified in large technology and software engineering programmes, such as enterprise system implementations. In general, enterprise system implementations are structured around functional project teams, which represent modules within the network and consist of team members from across the various organizations [27], with the core node within each module being the project manager for the particular project team. This complex interconnected network of individual team members that form the various project teams, gives rise to an overall programme-wide network structure that oftentimes maps onto the high-level functional modules within the enterprise software itself.

Guimera and Nunes Amaral [28] suggest that the modular structure of complex networks is the key to their success, which in the case of enterprise system implementations suggests that the structuring of project resources around functional modules plays a critical role in successful delivery of the programme. We agree with this in principle, but conjecture that although this small world topology with high connectivity within the individual clusters (within individual teams) and between the clusters (between the different teams) facilitates efficient delivery of the complete programme of work, it also incorporates fragility into the network because it allows conflict to propagate throughout the entire programme-wide network. Jackson [29] provides support to this argument by asserting that ideas and behaviours held by individuals within a social network can be transmitted from one person to another, and thus may propagate throughout the entire network. In addition, Gil [30] provides a conceptual framework around communication between team members, and how this affects the development of trust in the project manager in particular and the project more generally.

C. CONFLICT

Boulding [31] defined conflict within group situations as interpersonal incompatibilities (whether real or perceived) or the holding of discrepant opinions/outlooks between the parties involved, which may be individuals within a single group (intragroup) or between different groups (intergroup). It has been argued that conflict is an intrinsic part of group and organizational dynamics [32], and that it is strongly associated with individual group member's affective experiences [33], which can propagate throughout the network as a shared affect [34]. Conflict emerges from a variety of group-based situations [35], and in three main forms: task conflict, for example, differences in opinion between project team members on how tasks should be performed; process conflict, for example, an overly bureaucratic project environment with respect to policies/procedures; and relationship conflict, for example interpersonal issues between members of a project team [36]. The three types of conflict are expanded upon in more detail below.

1) Task Conflict

When conflict within project teams is functional, it is often task-focused [32], [37]. Task conflict has been defined as the awareness and subsequent behaviours that emerge between group members from differences in viewpoints and opinions of tasks that need to be performed within a group [38]. With particular emphasis on enterprise system implementations, this type of conflict may express itself within project teams as animated discussions and personal excitement from the individual team members [36]. Certain types of tasks have been shown to benefit from moderate levels of task conflict [39]. For example, it has been observed that during the design phase, if developers have been provided explicit requirements from the client, they often fixate on early solution ideas or the reuse of existing solutions, which can lead to suboptimal solutions [40], [41]. In addition, Shah and Jehn [42] have shown that project teams benefit from differences of opinion when faced with complex cognitive tasks. Within enterprise system implementations, this is frequently seen during the early design phase of projects, in particular around tasks associated with requirements analysis; or during the execution/delivery phase of the project when the team has encountered problems that require collective brainstorming to develop workarounds and mitigation actions. Schwenk [43] suggests that task conflict, when effectively managed, improves the quality of decision making because the consensus that emerges through synthesis of diverse perspectives, is superior to the perspective of any individual team member. In severe instances, it may also generate tension within the team [32], in particular if overly dominant team member(s) push their opinions in a forceful manner. This may inadvertently cause antagonism and unhappiness between other team members and themselves, which may emanate in an unwillingness to work together in the future (on other projects).

2) Process Conflict

When conflict within projects is focused on how tasks are successfully achieved, it is often labelled as process conflict [44]. Process conflict has been found within project teams that are experiencing poor productivity and showing low levels of group morale [36]. From the perspective of managing an enterprise system implementation, process conflict will predominantly relate to issues of duty, resource allocation to tasks, adherence to processes/procedures [45], excessive buildup of technical debt due to decisons [46], or technical dependencies that requires coordination between team members [47]. We therefore conjecture that within enterprise system implementations, disagreements between team members may lead to process conflict within the particular project team. For example, process conflict can develop following disagreements around who is responsible for completing a particular task, or around why decisions (that impact the wider team) made by an individual team member, have not been documented correctly. Although the conflict is confined to a subset of the team members, it has the potential to negatively affect the performance of the entire project team.

3) Relationship Conflict

When conflict within projects is focused on personal issues between individual team members, such as feelings of annoyance, frustration, and irritation, it is usually labelled as relationship conflict [36]. Relationship conflict is a form of affective conflict because individuals become aware of their incompatibilities through changes in their mood or through feeling tension and/or friction when they work together [48]. As discussed by Jackson [29], the people with whom we interact with on a regular basis are able to influence our motivations, decisions, and emotional wellbeing (including behaviour). Relationship conflict has been found to produce intolerance and mistrust regarding each other's intentions and behaviours [49]. This intolerance and mistrust causes considerable anxiety and emotional stress, which has been found to inhibit cognitive functioning in processing complex information [50], and ultimately leads to a reduction in performance [32]. Furthermore, when individual team members encounter interpersonal problems that result in relationship conflict, they may respond with a number of negative reactions, such as anger or frustration. Conversely, they may mentally decouple themselves from the situation because they experience anxiety and fear [51] or because they are uneasy with the feeling of disliking team members or being disliked by other team members [52]. As a consequence, they tend to work less effectively and may produce incomplete or poor quality project deliverables [53].

Within an enterprise system implementation, this means that fellow project team members from the same organization, or the client and/or third-party service provider, may affect the way an individual member views project aims and objectives, and the degree to which they are committed/motivated to perform project tasks and activities. In addition, implementations that utilize a multi-partner approach often encounter relationship conflict that is due to the competing corporate dynamics, and not just differences between individual team members. Kaiser and Bostrom [54] found that occupational and cognitive differences between the members of software project teams might increase the potential for conflict. Unless the existence of the conflict is both acknowledged and managed, it will invariably result in reduced productivity and satisfaction within the group [55]. In the most serious of cases relating to enterprise system implementations, this could result in failure of achieving project-level objectives by an individual project team, or failure of achieving the overall programme-level objectives due to conflict being propagated throughout the network.

D. PROPAGATION OF CONFLICT THROUGHOUT THE SOCIAL NETWORKS OF ENTERPRISE SYSTEM IMPLEMENTATIONS

The literature on conflict has focused almost exclusively on conflict generation at the level of individual team members or between two members. Here the individual team member collects and processes information from their environment, and uses their own individual cognition to perceive conflict, which in effect makes them only loosely coupled to the wider social system [56]. With respect to group conflict, Gelfand *et al.* [57] have developed a cultural transmission model of intergroup conflict, and Roberts *et al.* [58] have highlighted the relationship between task complexity (which may lead to task conflict) and the impact on communication patterns in IS project teams. Unfortunately, this model was focused on relationship conflict only, so did not cater for task and process conflict, and was also unrelated to enterprise system implementations. Their core assumption was that conflict is quite easily developed due to universal human traits, such as in-group preference and out-group hostility, and that once developed it can propagate within the group to manifest itself in conflict with another group.

We conjecture that the work of Gelfand *et al.* [57] has synergies to the way that relationship conflict can be developed and propagated within large multi-partner enterprise system implementations. For example, a project team made up of team members from a single organization (e.g. client, software vendor, or professional service provider) will have considerable in-group preference and if harm is detected towards one of their members by a member of another group, this harm will be felt by all in-group members, who will collectively punish the out-group [59].

III. RESEARCH METHODOLOGY

Our methodology is based on case study research [60] that used a combination of deductive inference and inductive research, through a multidisciplinary approach to empirical software engineering research [61]. Case study provides the ability to focus at the system-level behaviours and dynamics of a complex system, which in our case is the complex social system that develops during an enterprise system implementation with multiple partners. It also provided the ability for us to perform an in-depth investigation on the causes behind these system-level emergent behaviours [62]. Indeed, the choice of a case study allowed for the enterprise system implementation to be studied in its *real world* sociotechnical and spatio-temporal contexts [63]. The adoption of a case study approach allowed the utilization of both qualitative and quantitative data and mixed analytical method, to ensure a detailed analysis of the case along with elucidation of practical dimensions that needed to be considered in the conceptual framework, given the existing knowledge around the development of conflict within project teams (as discussed in Section 2).

Deductive reasoning was used following a review of the literature, to derive initial concepts of how conflict is developed within large technology and software engineering programmes, such as multi-partner enterprise system implementations. This was complemented by inductive reasoning, following participatory observation and the conducting of focus groups. Hanly [64] states that these two research strategies go hand-in-hand in scientific thinking, and facilitates the researcher to link theory to observable reality. By doing this, we aligned our methodological fit to the process advocated

by Miles and Huberman [65] for developing conceptual frameworks through case study.

The purpose of this study is to enhance our understanding of the causes behind conflict development within multi-partner enterprise system implementations (representing an example of large technology and software engineering programmes), and through development of a conceptual framework, can enhance our understanding of how conflict (once developed) can propagate throughout the social network of the wider programme. Two key definitions underpin the case: 1) conflict development, i.e. task, process or relationship conflict developing due to the interactions between individuals within a project team (intragroup), or between different project teams (intergroup); and 2) conflict propagation (following initial conflict development) throughout the wider social network of the enterprise system programme. Consequently, our case has several subunits of analysis focused on the hierarchy within the enterprise system implementation. The main unit was the programme as a whole (the "case"), and the smallest unit was the individual member. In addition to these two units, the case study also collected data about several intermediary units: the leaders among individual members; the project team to which specific individual members belonged; and the organization that employed individual members. Data came from a variety of sources, including observations, focus groups, and documentation shared with the research team.

A. DATA COLLECTION

Based on the multidisciplinary approach, data collection took a hybrid form over three phases. The first phase of data collection focused on documentary analysis, which allowed us to develop a comprehensive understanding of the overall case. Collected documentation related to: the programme aims, visions, and high-level scope; programme-level standards and procedures; programme-level roles and responsibilities; programme-level resource management plan; technical architecture; individual project management plans, scope and detailed requirements (functional and technical). This data provided the background context to understand the case, and to develop an initial conceptual framework.

The second phase of data collection focused on the need to develop an in-depth understanding of the hierarchical and resourcing structure of the programme. Consequently, an initial focus group was run with five project managers who were working on the programme, and represented: client, software vendor, and three professional service providers. The aim of the focus group was to: develop a list of members on the programme and their associated work-based demographic values; develop an adjacency matrix and resultant network map of the formal workplace relationships between the resources [21]. The focus group technique was selected because it provided the ability to gain a concensus on programme-level hierarchy and resourcing structure from senior resources who were employed by the different organizations involved in the enterprise system implementation, and we adopted the *realist strategy* [66] to empower the research participants to define the network structure, the interorganizational links, and the boundaries of the network for the case. In addition, the focus group technique allowed flexibility in the themes covered and questions asked of the participants, and also allowed for the participants to interact between themselves, which provided the ability to observe the dynamics between them (as representatives of the different organizations). In a similar way to semi-structured interviews, the focus group also allowed exploration of interesting themes in greater detail, to facilitate an increased understanding of the participant's background knowledge and sense making of experiences on the case [67].

TABLE 1: Composition of the focus groups

Focus Group	Code	Employer Type	Project Team
1	CustProg	Customer	PMO
	VenProg	Software Vendor	PMO
	PSP1Prog	Professional Service Provider 1	PMO
	PSP2Prog	Professional Service Provider 2	PMO
	PSP3Host	Professional Service Provider 3	Hosting
2	CustProg	Customer	PMO
	CustHR	Customer	HR
	CustPay	Customer	Payroll
	CustFin	Customer	Financials
	CustTech	Customer	Technical
	CustHost	Customer	Hosting
3	VenProg	Software Vendor	PMO
	VenHR	Software Vendor	HR
	VenPay	Software Vendor	Payroll
	VenFin	Software Vendor	Financials
	VenTrain	Software Vendor	Training
	VenTech	Software Vendor	Technical
4	PSP1Prog	Professional Service Provider 1	PMO
	PSP2Prog	Professional Service Provider 2	PMO
	PSP2HR	Professional Service Provider 2	HR
	PSP1Fin	Professional Service Provider 1	Financials
	PSP3Host	Professional Service Provider 3	Hosting

The third phase of data collection focused on the need to develop an in-depth understanding of the causes of conflict development within the case, and the causes of it's subsequent propagation through the social network of the wider programme. We performed three focus groups with project managers from the different organizations. The aim of these focus groups was to gain their perspectives on the working environment that they and their team members have encountered whilst working on the enterprise system implementation. Focus group participant's were purposively sampled [68] to ensure representation of project managers from the different organizations involved in the programme, and from the individual projects that make up the wider programme. The focus groups grouped project managers by their employer type (e.g. Client PMs, Software Vendor PMs, and Professional Service Provider PMs) to allow an open and frank discussion, and to mitigate any conflicts arising from such a discussion (see Table 1 for focus group composition). The focus groups lasted between 90-120min, were audio recorded, and were conducted with complete integration of ethical considerations. The audio recordings were later transcribed, anonymized, and analyzed using the framework described in the next subsection.

B. DATA ANALYSIS

The framework used for data analysis utilized three complementary strands that enabled explanatory case study data analysis. First, Social Network Analysis (SNA) was conducted on the programme structure and team member interaction data collected from focus group 1, to establish the network topology (network map) of the formal social interactions between resources on the enterprise system implementation, and to provide descriptive statistics of the network's characteristics. The purpose of SNA is to understand patterns of relationships among people within a system, to analyze the structure of these patterns, and to discover what their effects are on other people and organizations within the system [69]. However, SNA by itself does not allow us to develop a full understanding of the development and propagation of conflict, so needs complementing with other methods. Like Martínez et al [70], we consider that the principles of qualitative case study research [60] presents an opportunity to integrate SNA methods in the evaluation of conflict development and propagation within large multi-partner enterprise system implementations. Case studies performed under these situations are based on the analysis of the interactions of the programme members in the contexts where conflict is developed between an initial subset of the members and *potentially* spreads to other regions of the network.

The second strand of data analysis involved analysis of the qualitative data collected from focus groups 2-4, which allowed us to understand the causes of conflict development in the case, to provide in-depth explanations of the *local* effects of conflict (i.e. immediate intragroup or intergroup conflict), and to understand how conflict can propagate throughout the social network of the programme. In a similar way to the recent study of Ambituuni *et al.* [71], we used an inductive approach that was based on the strategy described in Braun and Clarke [72], through use of thematic analysis [73] to extract the major themes of relevance through coding of the data. This thematic coding assisted in establishing the causes of conflict development; the causes of conflict propagation; and the effects on social interactions following conflict development and propagation. Data familiarization was gained through repeated reading of the entire qualitative data set from the focus groups. On average, each transcript was read at least three times, with new themes emerging with each reading, or existing themes becoming more refined. The themes were considered in light of the overall topic of conflict and assigned initial codes that represent features of the data that appear interesting to the case, and were brought together to illustrate the analysis [74]. Subsequently, connections between themes and codes were established, and between themes and the various resources/roles within the programme. A degree of rationalization of themes was performed to ensure that data was reduced into meaningful categories.

For the last strand of data analysis, we took inspiration from Software Engineering and Management Science through the use of diagrammatic modelling to visually depict the causes of conflict development within the programme. A number of approaches have been developed since the 1990s, which have borrowed standardized diagrammatic languages from software engineering. One such standard is the Unified Modelling Language (UML), which although originally developed to document requirements for the analysis and design of computer systems [75], has recently been used to model complex systems more generally (e.g. [76]). The diagrammatical modelling in this study was performed in an iterative manner following completion of the thematic coding of the qualitative data. We used UML (v2.4) [77] as the basis to semi-formally define the interactions between programme resources that are involved in conflict development. Along with UML notations, a less formal cartoon diagram was also used to ensure the complexity inherent to the development and propagation of conflict through the interactions between individual resources could be conveyed efficiently. This diagrammatic notation is called a *Rich Picture*, and is a diagrammatic notation used in the field of Management Science (specifically, the Soft Systems Methodology approach, after [78]). The result of the SNA, thematic coding and diagrammatic modelling was triangulated with documentation analysis to verify the data, to account for the longitudinal aspects of conflict [63] and to provide additional information.

IV. CASE STUDY: THE RESOURCE MANAGEMENT PROGRAMME

The case relates to a large United Kingdom (UK) based organization (the Client) that was engaged in a major strategic modernization programme. The wholesale replacement of legacy systems was impracticable, so they chose a pragmatic approach based upon a strategy of co-existence and interoperability between old and new IT and IS infrastructure. Resource Management was identified as the set of business processes that incurred the greatest costs, and therefore deemed the most likely route for significant cost savings. The client had scoped and defined a major business-driven IS/IT programme of work, which was underpinned by the ability to integrate new systems and existing legacy systems in order to successfully deliver significant cost savings and facilitate more efficient business processes.

The client was using a number of non-integrated systems to manage their Finance, Human Resources (HR), Procurement, Payroll and other resource management business functions. These systems had been developed over time as discrete, even standalone, solutions to satisfy specific business needs. Through the Resource Management Programme (RM Programme), the client aimed to develop a fully-integrated suite of IT-based RM facilities for rollout to back-office support functions across the UK. In addition, they aimed to implement a Commercial off-the-Shelf (COTS) software product to facilitate change in core RM business processes, with the objective of introducing best practice business processes to the wider employee base around the UK.

Through a competitive tendering process, the client employed (on a fixed-price basis) a global software organization that specialized in enterprise systems to act as the sole software vendor to install, configure and test the COTS software, and to play a leadership role in developing the technical architecture required to support the RM Programme. The client employed two different Professional Service Providers to act as Subject Matter Experts and trusted advisors regarding the configuration and extension of business processes that were provided by the COTS software. Finally, the client also employed a third Professional Service Provider who focused on the IT architecture for hosting the enterprise system and associated IS technology (e.g. security, network communications, connectivity to external organizations, backup, disaster recovery, etc), and for data extraction from the small number of legacy systems that would be decommissioned, along with transformation and loading of this data into the enterprise system. The software vendor structured the programme around the different functional modules within the enterprise system application (e.g. Financials, HR, Payroll, Technical Development, etc) and utilized their own proprietary project lifecycle and software development methodology, however the overall programme was managed using the client's in-house methodology.

The case therefore represents a large multi-partner enterprise system implementation that engages resources from multiple third-party organizations. In the RM Programme, these third-party resources were grouped together with client resources, and assigned to distinct project teams that map onto the functional structure of the enterprise system. As such, the resources from the multiple parties (client, software vendor and professional sevrice providers) work together within the programme through an interfirm network [79]. Each of the individual project teams on the RM Programme comprised of members from different employing organizations. As theoretically established in Section 2, this introduces the potential for conflict development right from the start of the programme, due to their different educational and cultural backgrounds, different professional identities and normalized behaviours, along with their different objectives and motivations for being on the project. Conflict within enterprise system implementations, can be categorized into three distinct types: task conflict, process conflict and relationship conflict; with the potential for each conflict type affecting the implementation in their own particular way. The aim of this case, is therefore to further our understanding of how the various types of conflict develop (and propagate) within large technology and software engineering programmes, in general, and multi-partner enterprise system implementations, in particular. Through conceptualization of this phenomena, we also aim to identify practical interventions that can be incorporated into technology and software engineering project management practice.

V. RESULTS AND DISCUSSION

A. THE ENTERPRISE SYSTEM IMPLEMENTATION SOCIAL NETWORK

An adjacency matrix was created to represent the undirected and unweighted relationships between the individual programme resources. The initial focus group indicated that the RM Programme contained 159 individual resources, which become the individual nodes in our network map, and also corresponds to the order of the graph(n). The relationships between the team members were consistent with Walker [80], and consisted of five types of task relationships: reporting, problem referral (escalation), dependence on other members for information, feedback on performance, and dependence on other members for extra resources. There were found to be 972 undirected workplace relationships (known as lines) between these resources, which corresponds to the size of the graph(m). These two graph properties (order and size of the graph) allow us to calculate the average number of connections between resources, which is 12.23, and corresponds to the average degree of graph(k). Finally, the maximum undirected lines possible within the network is n(n-1)/2, which equates to 12,561 lines. Through dividing the number of undirected workplace relationships by the maximum number of undirected lines possible, we can calculate the density of the graph, which in our case is 0.077. This indicates that the overall network connectivity (or proportion of unique ties) is only 7.7% of all possible connections, so is a relatively sparse network. Fig.1 depicts the network topology derived from the adjacency matrix, which has been structured to correspond to the programme structure, and the nodes colour-coded for employing organization. This represents a goal-directed network ([81], p91), where the sub-networks (individual project teams of the RM Programme) are highly structured around a set of leaders (PMs from the client, software vendor and relevant professional service provider) who articulate the goals. Through analysis of transcripts from the four focus groups, and documentary anaylsis of RM Programme documentation, it is evident that the social network comprises: Knots, Cut-Points, Cliques, Social Circles, and Structural Equivalence of certain nodes (terminology used as per [81]).

A number of knots have been identified, which represent the group of resources from the same employer who are within the same project team (e.g. the knot of client resources within the HR project team). The resources who make up these knots have either strong or weak ties. This is because the functional/technical resources who do the work, rely on each other considerably for assistance with design, development, and implementation tasks. As such, there are strong ties in each project team between resources from the same employer (e.g. client-client, software vendor-software vendor). There are also weak ties between resources from different employers (e.g. client-software vendor) as they need to work collaboratively to achieve project objectives, however, they have stronger loyalty to their fellow team members from their employer, than their fellow team members overall. The PMs and Programme Managers represent the cut points

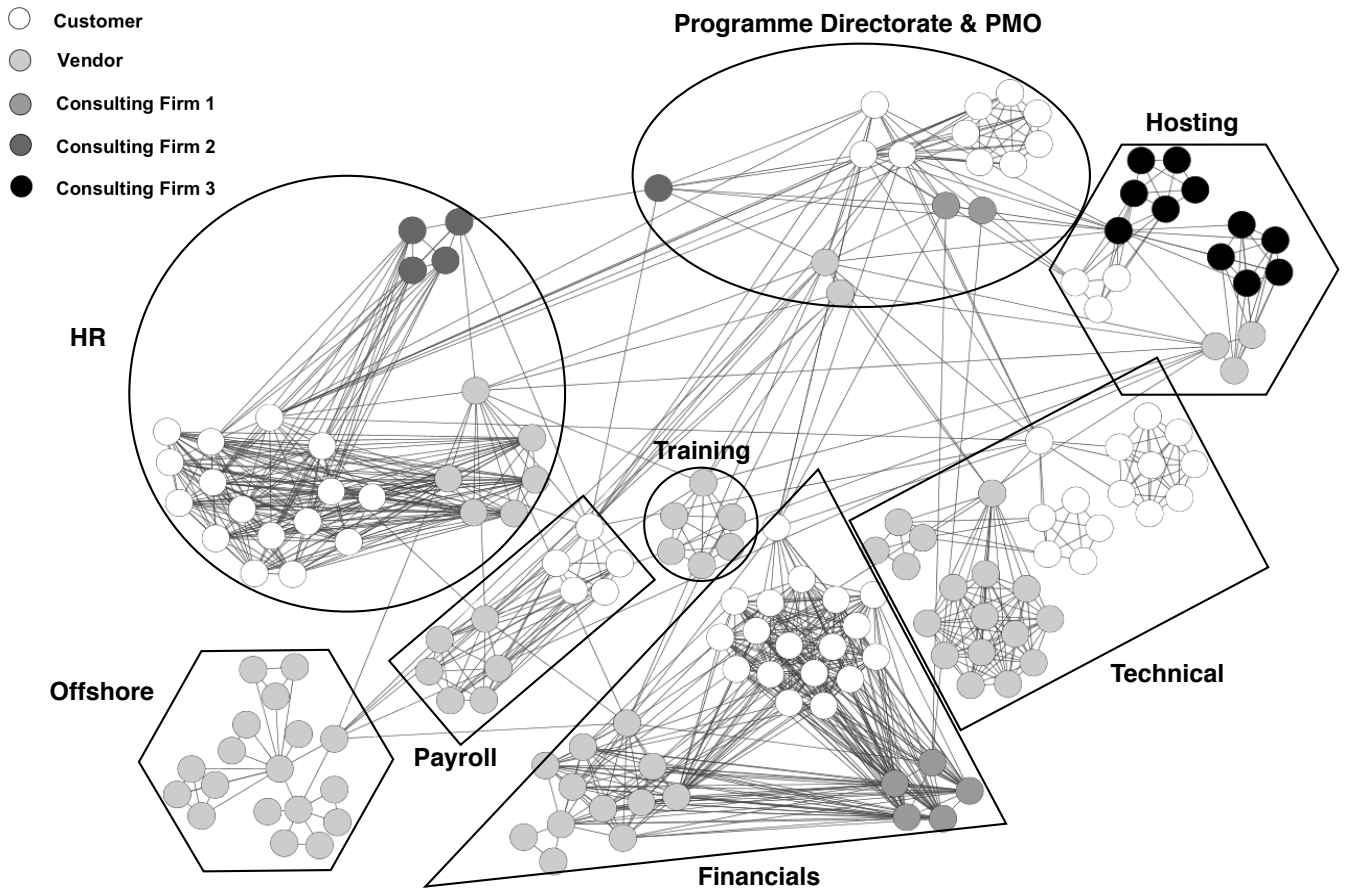

FIGURE 1: The Programme-Level Environment: Social network for the RM Programme. This social network corresponds to the work-based relationships of resources within the overall programme. Building upon the programme structure defined in Section 4, it can be seen that the programme-level social network is composed of sub-networks at the project team level, which implement specific modules from within the enterprise system software, and that there are specified resources who have *formal* working relationships with counterparts in other project teams.

within the network, and in effect act as the *bridgers* between the knots, along with acting as bridgers between the social circles, which within the programme-wide social network are synonymous with the individual project teams. In addition, 25 cliques were found within the social network, which related to the grouping of resources by project team and employer (e.g. client resources within the HR project team).

Furthermore, it was found that the majority of functional resources who were from the same employer within a project team, exhibit structural equivalence due to them being required to fulfill similar tasks and activities, and due to their completion of standardized training courses when initially recruited into the respective organization (e.g. core HR processes and procedures). There are of course exceptions, which result from resources who have niche functional skills, such as those involved with complex reporting requirements (e.g. activity-based costing for Financials) or those that require an interdisciplinary/interprofessional perspective (e.g. employment law for HR), but on the whole, it was found that a functional resource from a particular employer within an individual project team had structural equivalence with their colleagues from the same employer within the same team. A similar result was found for technical resources due to them fulfilling similar tasks and activities, and due to their similar training, experiences, and core technical skillsets (e.g. programming with Java, SQL, and XML; and software engineering), although exceptions were found for niche skills, such as database administration and technical architecture, which had no structurally equivalent nodes in the network. Conversely, although PMs were found to not have structural equivalence, those from the third parties were found to have role equivalence because they were able to be swapped around to other project teams, or replaced by other PMs from within their employers resource pool of PMs. This was evident from two examples, the first where the software vendor PM in the Financials project team was replaced due to lacking credibility with the client part-way through the programme; and the second example is where the software vendor PM in the HR project team was replaced five times, and each time at the request of the client HR PM due to a breakdown in working relationships, and the development of relationship conflict.

B. TASK CONFLICT WITHIN ENTERPRISE SYSTEM IMPLEMENTATIONS

As discussed above, the RM Programme consisted of a large number of personnel (n=159), who were employed by different organizations. As defined in Fig. 1, the RM Programme was structured around different project teams that aligned to the functional areas of the enterprise system application (e.g. HR, Payroll, Finance). Documentary analysis indicated that the majority, in particular the core functional consultants (from the software vendor and professional service providers) and the client functional resources, were assigned to the programme for its entirety (circa 3.5 years).

Analysis of transcripts from focus groups 2-4 indicate that the RM Programme exhibited considerable task conflict throughout. Not all task conflict was deemed negative however, because the animated discussions and personal excitement (a form of task conflict) from team members at the project initiation and design phases of the programme, were found to be beneficial to particular activities, such as problem solving, requirements gathering and analysis, and design of the enterprise systems configuration and extensions (where out-of-the-box functionality was not provided). This is confirmed by VenProg, who stated that *"If you're all aiming for the same thing, with a shared vision on programme aims and objectives, then task conflict between yours and other supplier resources can be a joy to watch [...]. There is a tendency for lively debate and the raising of voices, [...] but, positive task conflict, such as this, can help ensure the enterprise system is fit-for-purpose."*. Conversely, we also found that too much task conflict can interfere with consensus formation, and can also distract project team members from their individual tasks/objectives, which can lead to delays.

We also found that power imbalances between resources and the assignment of too much work to be significant reasons that task conflict may initially develop within a project team, with additional reasons relating to: differences of opinions during requirements gathering, requirements analysis, and design; difference of opinions with respect to problem solving, risk mitigation, and contingency planning; and the forcing of opinions onto others due to power imbalances or inflated ego (see Fig. 2). Additonally, we discovered that differences in corporate objectives were consistent themes within the focus groups, and also the triggers for a number of risks on the RM Programme centralized risk register. VenPay and VenProg discussed in-depth the development of task conflict through power imbalances and issues of ego between rival third-party organizations. To summarize, VenPay believed that the custumer appointed a consultancy firm to act on their behalf as the prime contractor in order to defer risks, and potentially to transfer accountability for any failure(s). He was very firm when stating that *"in my mind this is abdication of responsibility and accountability, but worse still, it leads to an overly dominant set of resources from the prime contractor who ensure that they win at all costs - the classic zero sum game, where to the victor comes the spoils, no matter the consequences to the other partners, or indeed the client."* [VenPay]. VenProg then built upon this analysis of the situation by advising that once the consultancy firm(s) start to *"play dirty and blame my team for mistakes"* that commitment issues from his resources rapidly ensues, and impacts his ability to manage the programme from a software vendor perspective. It also became evident that from his experience, there is always a phase within large enterprise system implementations where a convergence of activities (i.e. multiple work streams being performed in parallel), connects together deliverables from multiple suppliers, and that due to the contractual environment oftentimes being based around back-to-back fixed-price contracts, the suppliers have a tendency to blame each other for delays. Our analysis identified that such behaviours within the RM Programme created tensions between supplier teams, and if repeated, led to a change in individual supplier level behaviours, where they began to focus on activities that provided maximum benefit to them, and a lack of commitment to those that other suppliers were dependent upon.

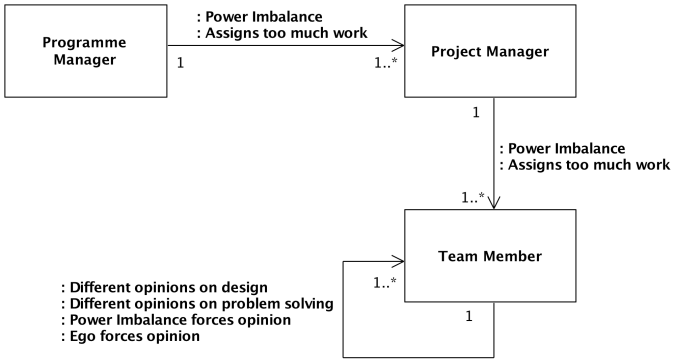

FIGURE 2: Development of task conflict in the RM Programme. UML Class Diagram depicting the development of task conflict in the enterprise system implementation. It can be seen that task conflict is directed, in that a Programme Manager can facilitate one or more Project Managers to experience task conflict; a Project Manager can facilitate one or more Team Members to experience task conflict; and an individual Team Member can facilitate one or more Team Members to experience task conflict.

Overall, our analysis indicated that within the RM Programme, task conflict was able to develop between members of a particular project team, (e.g. due to differences in their opinions on how to solve a particular problem the team faces, or due to power/ego imbalances), and that although the conflict was initially confined to a subset of team members, it had the potential to negatively affect the performance of the project team as a whole. In addition, task conflict was found to develop between two suppliers (e.g. due to lack of resources from one side; or between members of the different project teams, for instance due to differences in their opinions on how to solve a particular problem that affects both project team deliverables/objectives), and that once developed, this conflict had the potential to negatively affect the performance of both project teams.

C. PROCESS CONFLICT WITHIN ENTERPRISE SYSTEM IMPLEMENTATIONS

As discussed previously, when conflict within projects is focused on how tasks are successfully achieved, it is often labelled as process conflict. Analysis of transcripts from focus groups 2-4, along with documentary analysis, indicate that this was indeed the case throughout the RM Programme. We found that the RM Programme was routinely affected by disagreements that led to uncertainty, which is symptomatic of process conflict. PSP3Host provided a clear example of this through his animated discussion of the chaos that ensued to the RM Programme, and the generation of tension and conflict between third-party providers, when the client was either unwilling or incapable of making important decisions on design options that underpinned the enterprise system as a whole and therefore affected multiple functional modules, and thus multiple project teams. His comments reinforce those of VenPay above, but were made within a different focus group, so provide additional evidence of the client having deficiencies around accountability for the overall programme implementation, and that they are actively trying to delegate this to one of the third-party providers. Through documetary analysis, we conclude that this may have been a symptom of the contractual environment, where back-to-back fixed-price contracts between the client and the individual third-party providers, meant that none of them were contractually accountable for the end-to-end implementation.

CustHR similarly discussed in-depth how the uncertainty generated from the contractual environment led the third-party providers to prioritize fulfilling their own contractual obligations at the expense of the greater good for the RM Programme, which caused process conflict through disagreement between the various organizations on who is responsible for resolving emerging technical issues that affect the enterprise system as a whole and not just a particular functional module. In one particular instance, CustHR discussed how her team had only just averted a major disaster on the wider programme, due to the PSP1 team within the Financials project designing the Chart of Accounts in isolation (to other project teams), because that is where it technically sits within the software application, and as such was contractually their responsibility. She advised that it was only when a member of PSP2 who was independently overseeing the design of the HR and Payroll modules, brought it to her attention, that the programme management structures realized that the Chart of Accounts design that was being advocated, would impact on their ability to pay expenses or salary. It took 3 months of rework to Financials, HR, and Payroll module designs to develop a Chart of Accounts suitable for the entire organization. We consider this example to be consistent with the findings of Gil *et al.* [82] regarding external factors in engineering project designs having disproportionate consequences to dependent projects, which in this instance was the Chart of Accounts design in the Financials project team (the external design) impacting the design and implementation being performed by the HR project team.

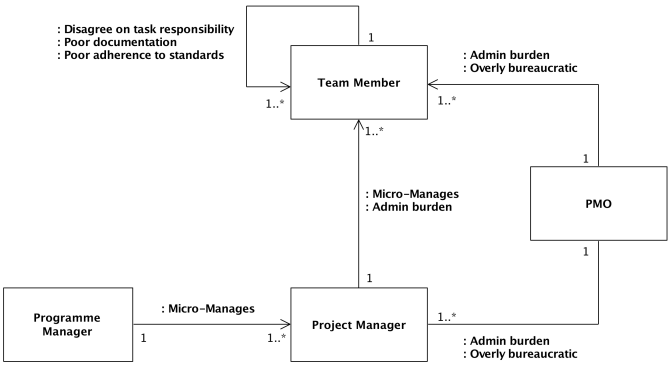

FIGURE 3: Development of process conflict in the RM Programme. UML Class Diagram depicting the development of process conflict in the enterprise system implementation. It can be seen that process conflict is directed, in that a Programme Manager can facilitate one or more Project Managers to experience process conflict through micro-management; a Project Manager can facilitate one or more Team Members to experience process conflict through micro-management or enforcing an excessive administrative burden; an individual Team Member can facilitate one or more Team Members to experience process conflict through disagreement on responsibilities, poor documentation of their functional/technical deliverables, or poor adherence to design/development/testing standards; and the PMO can facilitate one or more Project Managers or Team Members to experience process conflict through enforcing an excessive administrative burden or overly bureaucratic processes, policies and procedures.

A consistent theme throughout the focus groups with third-party providers was that the overly bureaucratic nature of the client led to a significant number of non-client resources not adhering to certain project management processes/procedures (such as formal reporting or scope management that are mandated by the PMO) because they were focused on implementation tasks, and believed the additional administrative tasks were an unnecessary burden. With specific reference to the software vendor, VenProg advised that this caused a degree of contempt to develop between his team and the client PMO, which led to a significant degree of process conflict. His exuberant discussions on this point were typified when he commented that there was too much emphasis within the RM Programme on the governance processes, which manifested itself in the generation of a mountain of documents that fulfilled no other purpose than forming an audit trail for when things went wrong. In his professional opinion, there was too much emphasis placed on auditability and not enough placed on deliverability. This was reinforced by PSP2Prog who suggests that there is no benefit to anybody in having a project that can be audited to show why it failed, but that there is huge benefit in having a project that does not need to be audited because it is a success, which we deduce

to mean that with the focus being disproportionately placed on document generation, the RM Programme inadvertently lost focus on actual programme delivery.

We also identified that the multi-partner environment within the RM Programme introduced significant risk of process conflict developing within the programme-wide social network due to disagreements around responsibilities for specific tasks, and the lack of trust when one supplier is dependent on the successful delivery of intermediate products/deliverables from one of the other suppliers (Fig. 3). These findings indicate that the propensity for process conflict is exacerbated when multiple organizations are needed to implement a large engineering or technology programme, and that the multi-partner alliances within the RM Programme introduced the risk of significant process conflict into the programme-wide social network.

D. RELATIONSHIP CONFLICT WITHIN ENTERPRISE SYSTEM IMPLEMENTATIONS

As discussed previously, when conflict within projects is focused on personal issues between individual team members, such as feelings of annoyance, mistrust, or differences in personal characteristics (e.g. education, family or professional backgrounds), or due to differences in corporate drivers, values, and cultures, it is usually labelled as relationship conflict. Analysis of transcripts from focus groups 2-4, along with documentary analysis of various project management artefacts produced during the implementation programme, indicated that this was indeed the case throughout the RM Programme (Fig. 4). At a high-level, this is succinctly put by VenTrain who states that *"[...] where you have a multi-partner model, with the prime consultancy firm acting on behalf of the client, there's all the opportunity in the world there for contention and attrition and ill feeling between the different third-parties."*

The RM Programme provided considerable instances of relationship conflict, however our analysis found the majority to be due to either a misalignment of organizational objectives (i.e. competing corporate dynamics, and not just differences between individual team members), or due to power imbalances or power grabs. VenTech discussed this in-depth with numerous examples around how relationship conflict developed from the behaviours of PSP1, who acted as the *prime contractor*, and whose Programme Director became seconded to the client part-way through the RM Progamme and therefore became part of the client Directorate, which in our view equates to the metaphorical *poacher, turned game keeper* scenario. VenTech's discussions explained how bad behaviour on the part of the prime contractor resulted in similarly bad behaviour developing within his team, in particular how mistrust developed when his team felt that they were being unduly blamed for technical issues and delivery delays. Our analysis suggests that the cause of this behaviour from the prime contractor and the resulting mistrust and relationship conflict, was ultimately due to the client's unwillingness to manage the programme for themselves, and the

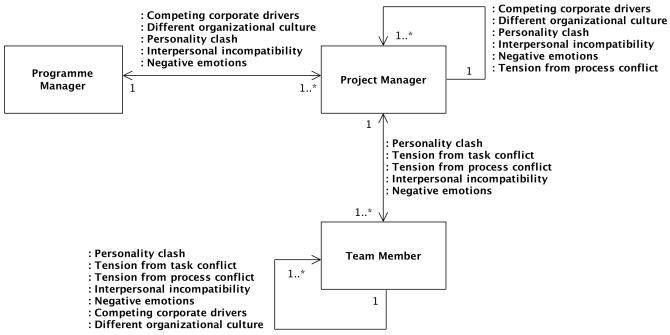

FIGURE 4: Development of relationship conflict in the RM Programme. UML Class Diagram depicting the development of relationship conflict in the enterprise system implementation. It can be seen that relationship conflict is bi-directional, in that conflict can develop between: a Programme Manager and one or more Project Managers through competing corporate drivers, different organizational cultures, personality clashes, interpersonal incompatibilities, or negative emotions; different Project Managers due to the same reasons, along with tension that arises from process conflict (transformation of process conflict to relationship conflict); a Project Manager and one or more Team Members due to personality clashes, tensions from task or process conflict, interpersonal compatibilities, or negative emotions; different Team Members due to the same reasons, along with competing corporate drivers and different organizational cultures.

appointment of the prime contractor to allow them to transfer accountability and responsibility, which links us back to our analysis on task and process conflict above, and the client's abdication of accountability and responsibility.

Through documentary analysis, we found that one of the major drivers for the RM Programme, was to streamline the client's back office functions. As a consequence, these efficiency gains would mean that fewer resources would be needed to fulfil the administrative functions, and if the individual resources could not be retrained or assigned elsewhere, it would lead to redundancies. Our analysis confirms that a number of client resources came to this realization part-way through their time on the RM Programme, and became demotivated and uncommitted to the aims and objectives of their respective project. This was discussed within the focus groups by PSP1Fin, PSP2HR, and VenPay, who all advised that there is a strange dynamic when a delivery team goes into a project to improve efficiency in the client's business, due to the client project team consisting of resources who have been removed from their daily jobs, and put under immense pressure to deliver a project for their employer, which is going to eliminate some people. PSP1Fin reinforced this by stating that it could be some of the project resources themselves, because they're now out of the daily cycle of the core business, and if the business were to survive the software implementation without them, then there is an indispensability question about them. To us, this is clear evidence of the unforeseen consequences of technology and engineering

programmes that lead to business efficiencies, inherently containing the potential for conflict due to the disconnect between an individual resource's personal objectives and that of the organization. Indeed, we believe it is understandable that resources may be unwilling to commit in a project team because they may think that the better the project does, the less chance they have of retaining their employment, or continuing to work with their friends/colleagues. This was evidenced when VenPay advised that following this realization by the client PM for HR, she became significantly aggressive towards the software vendor, so much so, that she went through five PMs from his organization in the first 18 months of the programme.

Alongside the examples discussed here, our results suggest that within enterprise system implementations, relationship conflict can develop between two or more members within the programme (either within an individual project team, or between project teams), and that this can be due to personality clashes, differences in demographics (e.g. nationality, race, ethnicity, religion, age, and gender), or differences in educational or family backgrounds (e.g. University attended, subject studied, class/wealth/caste of family). Although the examples of relationship conflict were initially confined to the subset of resources were it originated, we found there was the potential for it to negatively affect the performance of the project team(s) as a whole.

E. PROPAGATION OF CONFLICT WITHIN ENTERPRISE SYSTEM IMPLEMENTATIONS

Following on from the discussion of conflict generation above, and through further analysis of transcripts from focus groups 2-4 we have discovered that once conflict had developed within the RM Programme, it then had a propensity to propagate through the social network of either an individual project team, or indeed between project teams and thus *infect* the wider RM Programme. This is characterized by conflict propagating between team members of an individual project team, and also between team members that are situated within different project teams.

Analysis indicates that after conflict developed it had on a number of occasions propagated throughout the social network of the RM Programme. Instances had occurred either at a local level through affecting resources within an individual team, or more widely through conflict being spread via *bridgers* (predominantly PMs) outside of the initial project team where the conflict initially developed and over to other project teams that were connected to the *bridger*. We found that propagation could occur for all three conflict types (task, process and relationship), with the actual spread of the conflict being closely aligned to the physical structure of the underlying topology of the social network. For the RM Programme, this specifically meant that individual project team members might adopt conflict-related attitudes, beliefs, behaviour or even emotional states, from other members with whom they communicate.

In addition, we found that task and process conflict can evolve into relationship conflict if their causes and effects are not managed appropriately, i.e. task conflict that is caused through an imbalance in workload between team members within a project team, can evolve into relationship conflict between either the team member who is overloaded with work and their PM who assigns the work, or between the two team members who have differing workloads due to feelings of inequity and lack of parity within the project team. An example of conflict evolution and subsequent propagation relates to the HR project. It appears that the RM Programme when initially communicated to client personnel, was positioned as a technology project to replace legacy systems that did not communicate with each other effectively and efficiently, however upon progressing through project implementation it became apparent to more experienced and senior client personnel that alongside improvements to the technology, there would also be efficiency savings from implementing the business processes from the COTS software. CustHR became aware of this additional organizational benefit from a successful implementation of the RM Programme, and realized that efficiencies in business processes would allow for the reduction of headcount within the HR Department. Due to the location and culture inherent to the client organization, it became evident to us, that the majority of client employees were born and grew up in the local area, and that for them their colleagues were also deemed friends, and to some extent may have also been family members. As such, any restructuring of the organization to maximize benefit from the RM Programme, would have a significant impact on the employees, their families and the local economy. Upon realizing the wider consequences of a successful implementation, CustHR experienced significant process conflict because the consequences of her being involved with successful delivery of the RM Programme would negatively impact her colleagues who may also have been friends and family members. This process conflict (and to an extent we can conjecture a moral dilemma), led to CustHR becoming negative to the HR project and actively trying to subvert the implementation by withholding access to her team members and delaying approval of design and specification documents. This led the vendor HR team members to experience task conflict because they intimately relied upon the knowledge of CustHR's team members, and to the timely review and approval of designs and specifications. The delays incurred by the vendor HR team members resulted in the HR project falling behind schedule and the vendor HR PM to experience process conflict. Once the vendor HR PM had uncovered the cause of the delays, this resulted in the process conflict being experienced by them and CustHR to evolve into relationship conflict between each other. Finally, once the relationship between CustHR and the vendor HR PM had turned negative, the relationship conflict propagated throughout the team members within the HR project because the individual team members aligned with their employer PM. Unfortunately, the conflict was not confined to the HR project team, and through the two PMs acting as *bridgers* to other project teams,

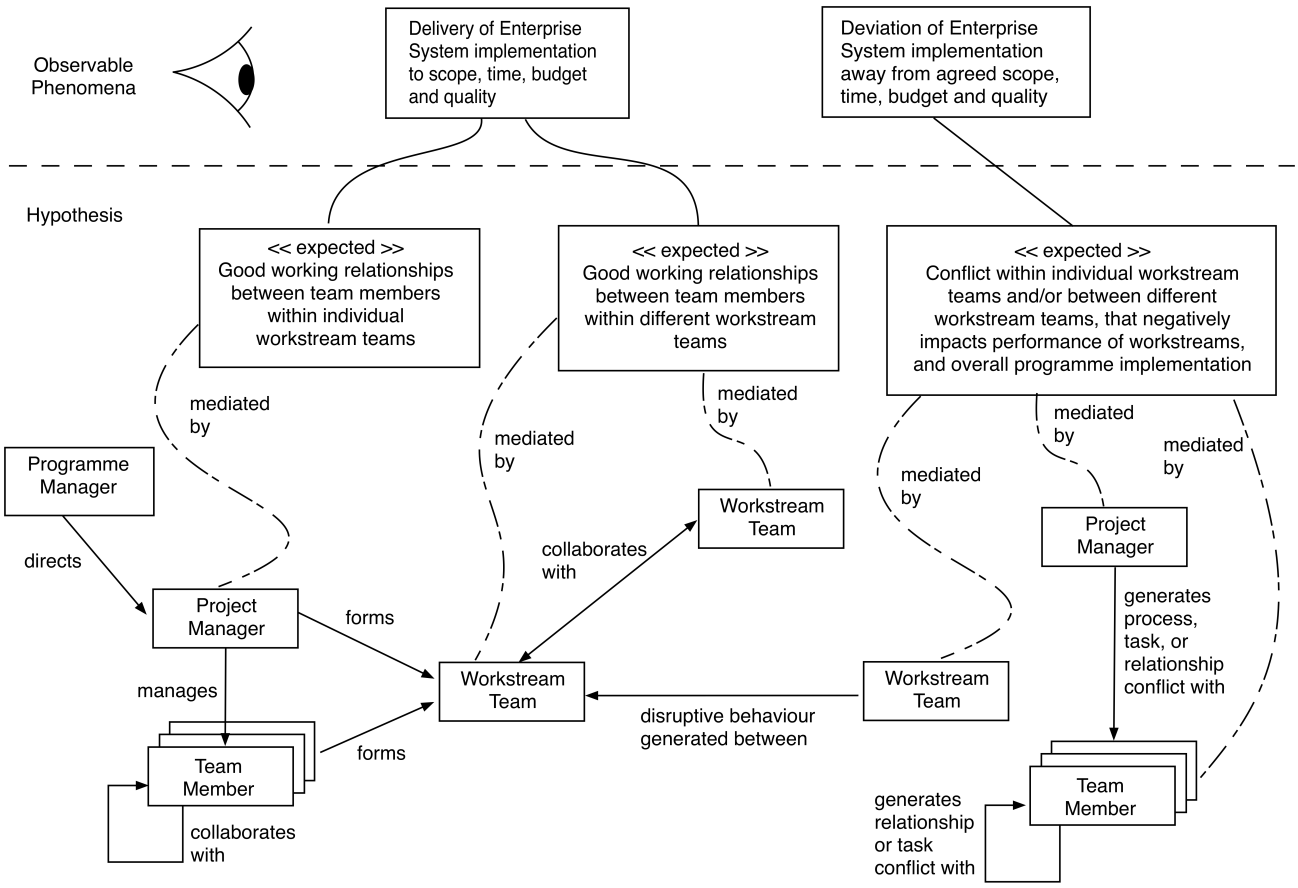

FIGURE 5: Rich picture: Observable phenomena that emerge from the interactions between individual team members. It is hypothesized that conflict can develop through a number of mechanisms, and that once formed, it may propagate throughout the social network of the wider programme. If any of these occur, they may be observed as deviations away from the agreed scope, time, budget and quality of the enterprise system implementation.

resulted in the relationship conflict propagating through the wider RM Programme social network. In this instance, the conflict was irrecoverable between the two PMs and the vendor PM was removed from the HR project, and indeed off the RM Programme. Focus group discussions allow us to infer that this was a sufficient gesture by the vendor to pacify the client team as a whole, but that the process conflict experienced by CustHR remained and proved to be a significant issue throughout the remainder of the RM Programme; it was eventually successfully implemented however.

Our findings from the RM Programme allow us to conceptualize the propagation of conflict within large multi-partner enterprise system implementations. This is characterized by conflict propagating between team members of an individual project team, and also between team members that are situated within different project teams. We take inspiration from Gamero *et al.* [34], who discovered that the type of conflict being experienced can transition over time due its dynamic nature, with task or process conflict being able to transition to relationship conflict, and relationship conflict being directly related to shared affect, and thus the most common form of conflict to propagate through a social network. This propagation of conflict is visualized in Fig. 5 through the use of a rich picture diagram, which depicts the observable phenomena of enterprise system implementations; the behaviours that are hypothesized to be responsible for these phenomena; and at an abstracted level, the components of the complex system that are believed to be responsible for the development of these emergent behaviours. At the highest level of the complex dynamical social system, delivery of the enterprise system implementation either proceeds to the project plan for the core measurable aspects of scope, time, budget and quality, or deviates from the agreed values. It is hypothesized that these phenomena (delivery to plan, or deviation from agreed values) occur through three sets of interactions throughout the social network of the enterprise system implementation. These are: 1) good working relationships within individual teams that are mediated by the PM and the collaboration and collegial approach of individual team members; 2) good working relationships between different teams that are mediated by the collaborative approach between the individual team members from the respective teams; 3) conflict develops within an individual team, or between teams, that negatively impacts performance of in-

dividual teams and may also impact the overall programme implementation if the conflict propagates.

In summary, our findings from the RM Programme indicate that once conflict has developed within, or between, project teams on large multi-partner technology or engineering programmes, it can propagate throughout the social network of team members, which may negatively affect the performance of the wider programme as a whole. This is consistent with the notion of in-group and out-group dynamics, where through their normal day-to-day communication with other team/organizational members, allegiances and solidarity can form with those like themselves (e.g. project team or organization). This facilitates the propagation of conflict, so that members within each individual project team or from a particular organization, act in solidarity, but are in conflict with members from other project teams or organizations. As such, the conflict, which initially developed between a subset of team members from two different project teams, has the potential to propagate throughout the social network of the wider enterprise system structure, and may negatively affect implementation of the enterprise system as a whole.

VI. IMPLICATIONS FOR RESEARCH

Our conceptual framework has provided insight into how conflict can develop and propagate within large multi-partner enterprise system implementations. It is clear that every individual within the social network of these large technology and software engineering programmes can be perceived as a representative of a particular group, which we have categorized as the project teams that implement a specific functional module of the enterprise system or develop new technological workarounds to problems.

Our findings indicate that the small average distances in enterprise system programme team networks facilitate diffusion of ideas and emotions, with the former often being associated with the positive side of conflict such as task conflict when designing the system or identifying solutions to problems/issues, and the latter being associated with the negative side of conflict through the propagation of negative emotions and behaviours, which are often associated with process or relationship conflict. We believe that the network structure can be both the cause of conflict, through the generation of conflict at the interpersonal relationship level, and the effect of conflict once established, as the initial conflict is propagated throughout the wider social network. Our findings indicate that this may be due to the interpersonal relationships between individuals, acting to configure and adjust the informal social structure of the temporary organization (e.g. the RM Programme), and that the overall social structure of the organization will effect the capacity of individuals to develop interpersonal relationships, and thus the potential for initial conflict development and subsequent propagation.

The nature and frequency of these interpersonal interactions may affect the perceptions of project team dynamics, and whether there is the development of intra- or inter- team conflict. Similarly, if intra- or inter- team conflict is perceived to be developing, then this may act as a feedback loop to affect the nature and frequency of interpersonal interactions. We believe that this can also feed-forward by affecting the perceptions of other interpersonal interactions within the social network, that were not involved in the initial conflict generating circumstances. To this end, we believe that destructive conflict, such as the relationship conflict that can arise through interpersonal interactions, can spread through the immediate networks of the initial protagonists, and then reach a tipping point, where the conflict becomes independent of the initial cause, and escalates throughout the wider network without reference to the original circumstances of conflict development. We propose this to be similar to the spread of contagion or disease that is often the focus within the scientific disciplines of epidemiology or immunology.

Our focus in developing this conceptual framework was to initiate a dialogue about the development and propagation of conflict within the social networks of large technology or software engineering programmes that use a multi-partner environment, and in particular enterprise system implementations, due to their influence over the organization-wide business processes/operations used by the client. This complements the recent work of Adami and Verschoore [83] who focused on project governance within a network of interacting organizations, along with that of Rivard and Lapointe [84] who used a cybernetic approach for investigating user resistance to IT implementation. Through leveraging our rich picture (see Fig. 5) and UML diagrams of conflict development (Figs. 2-4), we believe that parsimonious models of causal relationships between specific predictors and outcomes could be developed, which would subsequently facilitate the generation of hypotheses to be tested through future research.

Furthermore, we believe that the existing, and indeed extensive, body of knowledge surrounding organizational conflict, is ready to be augmented through the use of new computational approaches. For instance, we strongly believe that there is opportunity for future theoretical and empirical research into this area, and particularly advocate the use of social network analysis using computational visualizations, along with computational modelling and simulation, to better understand the causes and dynamics (e.g. feedback loops) of conflict propagation within this example of a technology or software engineering programme. Our conceptual framework is designed as a starting point for this conversation and we hope it will initiate a number of lines for future research.

The objective of social network analysis is to investigate the topological pattern and qualitative characteristics of the interactions that occur between the different members of the social network. As such, we believe that new insights can be gained from this approach to enhance our conceptual framework, and to develop hypotheses on the causes for conflict development between a small subset of members within the network, and how this initial conflict can ultimately lead to network-wide *symptoms* through the conflict acting as a *con-*

tagion and potentially *infecting* the wider network as a whole. The objective of computational modelling and simulation in this case, would be to use the computational model (such as an agent-based model) as a scientific instrument, which would allow experimentation to be performed within the computer to test various hypotheses around the development and propagation of conflict [85]. We advocate three main strands of computational experimentation. The first would be to take inspiration from immunology and public health, to better understand the propagation of conflict as a contagion within the social networks of large multi-partner enterprise system implementations. The second would be to develop computational models of the complex social system, to allow us to gain a better understanding of the underlying social relationships and dynamics that could be used to resolve conflict; in essence developing hypotheses on how to generate a robust and appropriate dampening response, which could act as a negative feedback loop to system dynamics around conflict. The third would be to gain insight into the robustness and fragility of the social networks within and between project teams, which would allow us to develop new hypotheses on how PMs may mitigate the risk of conflict propagating outside of their immediate team members, and in effect *quarantine* their team, to protect the wider social system.

VII. IMPLICATIONS FOR PRACTICE

Conflict within a technology and software engineering programme, such as the RM Programme, is practically intrinsic to the life and dynamics of the constituent project teams. As such, it can be considered as a universal social process through which differences of opinion between individual team members (interpersonal conflict), between members of a team (intragroup conflict), or between different teams (intergroup conflict), are confronted and occasionally resolved.

The Standish Chaos Report [86] argues anecdotally, that the majority of software engineering and IT projects are implemented over both time and budget. We believe the 'human' element is a major reason for this, and hypothesize that conflict within the social network of the implementation team(s) is to blame. In particular, we believe that this conflict within large multi-partner technology or software engineering programmes, such as the RM Programme in this study, may develop through three main causes: communication issues, lack of trust between vendor and client resources, and lack of commitment (by individual team members) to the project. Furthermore, De Dreu and Weingart [87] advise that both task and relationship conflict are consistently linked with poor performance of project teams, in particular when they are engaged in highly complex tasks, such as those involved in technology or software engineering programmes.

Our findings indicate that the RM Programme, which was an example of a temporary organization that was formed through the integration of multiple third-party providers along with client resources, contained competing networks through multiple organizational groups (i.e. project teams composed of members from a vendor or professional service provider) making ties to the same third-party (i.e. the client). Indeed, the findings suggested a tendency for competition to arise between two or more teams from the different professional service providers, which we conjecture may have been due to the economic or political environment of their respective employer organizations, who may have requested them to extract additional revenues from the client. In addition, and more sinisterly, the team(s) may also be requested from senior management within their employer organization to actively damage the reputation of a competitor, and thus poach the contract(s) that they have with the client - in effect, the client is coerced into removing one of the third-parties from the multi-partner programme.

We believe that our conceptual framework along with future research that results from it, will benefit professional practice in four main ways. First, it will help Project and Programme Managers to recognize and understand that conflict, once developed, can propagate throughout the wider social network of large multi-partner technology or software engineering programmes, and not only cause significant impact to their project, but may also act as a contagion to impact other projects within the wider programme, with potentially catastrophic effects. Second, it will help Programme Managers and members of organizational senior leadership teams (such as CIO, CTO, and Project Sponsor) to recognize and understand the impact that multi-partner relationships can have on the causes of conflict development, and the need to mitigate this through developing collaborative relationships between all parties, potentially through pre-project partnering activities. Third, we hope it will facilitate a renewed interest into PM methodologies and project lifecycles for managing technology or software engineering projects that exhibit complexity, and could also be integrated into Project Management Maturity Models, which have recently received a renewed interest [88]. Fourth, due to many Western countries rapidly becoming knowledge-based service economies, we believe the efficient management and implementation of technology or software engineering projects, in particular within the Public Sector, is of paramount importance to facilitating a healthy economy.

This study provides evidence of the paramount importance of fostering an environment where open and frank discussions can take place between all organizations, so that mistakes can be raised and discussed in a positive manner. Specifically, it's about developing a mutual trust relationship, where the third-parties can tell the client that they're wrong, and also where the client can tell the third-parties that they made a mistake in the contracting or design phase, and would like to move forward in a positive manner to ensure the best outcome for the implementation, without being financially penalized unduly. Unfortunately, large software vendors or professional service providers are oftentimes only interested in steadfast adherence to the contract, which generates conflict in the technology or software engineering programme. With this in mind, the environment should never be allowed

to degenerate to a situation where organizations become defensive. Furthermore, the relationships should never be allowed to deteriorate so that the client tries to punish the third-party providers commercially, or that the individual third-party providers try and extract unwarranted revenues from the client or act as predators against each other.

Traditionally, large technology or software engineering programmes that use multiple third-parties, such as the RM Programme, have consisted of contractual arrangements from the client to each individual third-party organization - akin to a hub and spoke on a bicycle, which has resulted in no incentive for individual third-party organizations to work with each other. Moving forward, this needs to be changed to foster a more collegial environment. Although there will still be a need for individual agreements between the client and the various third-party organizations, this would present an opportunity to introduce a *play it nice clause* and a *problem solving budget*, which means that they get incentivized if they run towards problems and proactively work together to fix them, rather than run away from problems and blame each other. We hypothesise that this would change behaviours and facilitate a collaborative environment because the organizations that help resolve problems will get a share of the *reward*. In addition, a crucial aspect of even being awarded the contract could be for them to declare that they will work with any other third-party organization, even those that they may have a competitive commercial relationship with at other clients.

Finally, we believe that there is a need to develop trust and foster good interorganizational relationships within large multi-partner technology or software engineering programmes, in order to set the direction, develop a shared vision, be clear about what each organization wants, and be clear on what success looks like, for each organization. It is paramount that each organization has *permission* to be an equal in the delivery, and to ensure that the various organizations work together and work closely. We believe that it is also important for the senior members of each organization (e.g. Programme Manager, Project Manager, or Team Leads) to have the ability to pick the phone up or have a face-to-face discussion to initially discuss and resolve problems, and not resort to email straight away to berate the other organization and to act as an audit trail of their displeasure. With that in mind, we believe that the only way for trust to be earned in such complex social and technical environments, is by developing a shared vision and set of objectives, and to ensure that the joint mission is for a successful implementation for all organizations.

VIII. CONCLUSION

Theory about the propagation of behaviours within social networks of interpersonal relationships within a workplace setting has a substantial history, which we believe dates back to Simmel's analysis of triads [89] that was fundamental to early research into group dynamics and social network analysis. With respect to technology or software engineering programmes, such as the RM Programme in this study, the conflict that may result between individuals due to their differing roles and employers, and the diverse nature of professional and social relationships that form in these large programmes, seems to underlie many of the multi-partner alliance difficulties. Although there has been considerable emphasis on investigating collaboration and group structures within multi-party business arrangements, we believe that little attention has been paid to the way in which conflict can propagate throughout the large-scale social network that develops from such multi-partner environments, in particular with respect to large multi-partner technology or software engineering programmes.

Our findings indicate that the RM Programme had multiple occurences of conflict development, and that some of these were significant enough to propagate outside of the initial setting (i.e. initial resources at the centre of conflict development) and *infect* other resources within the wider programme. We would like to draw a parallel of this situation to the discussion by Kilduff and Tsai [81] around goal-directed networks, which they suggest are likely to be highly connected, and thus lead to a tightly-coupled central structure, which for the RM Programme corresponds to the social circles that are represented by the individual project teams. As such, any conflict generated through the actions of one team member may affect the whole of the central core. In goal-directed networks, the potential for task conflict is likely to arise around agreement with and accomplishment of goals, which if not managed can lead to the development and then propagation of relationship conflict. This was indeed found to be the case within the RM Programme.

It is our hope that the conceptual framework developed in this study, can assist in developing intervention strategies for use by Project and Programme Managers working in large multi-partner technology or software engineering programmes. Of particular importance is the fact that Team Members who are consistently involved in conflict, either have a problem with the programme/personnel, or are the problem [90]. Unfortunately, we know from our prior experience in project management within IT consulting, that it takes time and effort to distinguish the *have a problem* from the *is a problem* resource. As such, if a relationship is broken, or if individual resources can't interact constructively with team members, disrupt meetings, or can't collaborate, then the pragmatic solution is to remove them. We envisage that the conceptual framework will initiate a renewed interest into conflict within teams, and that it, along with the results from future research, will provide insight into the robustness and fragility of project management of technology or software engineering programmes in general, and large enterprise system implementations in particular.

We believe that our conceptualization of the propagation of conflict may facilitate future research that provides an enhanced understanding of the reasons why large IT/IS programmes are not implemented within the initially agreed scope, time and cost parameters. These findings may then

lead to the design and development of new approaches for project lifecycles and project management methodologies (in particular around Risk Management) that are focused on the project social network, and how the network topology can facilitate propagation and amplification of social behaviours. This is of crucial importance, because knowledge of the amplification mechanisms within the network are essential prerequisites for developing approaches for mitigating the risk of conflict arising, or dampening conflict once it has occurred. In fact, we venture that once approaches for dampening conflict are developed, it will then become possible to propagate alternate social behaviours along the network, such as commitment and trust between members of the social network.

While some authors call for more research into success and failure factors of software engineering project management, based on the traditional project management triple constraint (i.e. Scope, Cost, and Time, which all impact upon Quality), our main interest is in understanding and describing the underlying causes in which conflict may affect successful implementation of large technology or software engineering programmes. Our conceptualization offers some implications for technology or software engineering management of large multi-partner programmes, in recommending: (a) that task and process conflict are actively managed by the PM so that they do not evolve into relationship conflict; (b) that the Programme-level Management Team do not just focus on the technical progress of the individual project teams, but also actively focus on intrateam and interteam dynamics and relationships; and (c) that conflict, once developed can propagate throughout the wider programme, and that preemptive action will be required, because we understand from our own experience that it requires less time and effort to proactively mitigate conflict propagation, than to reactively dampen down the response once it has begun.

REFERENCES

[1] T. L. Griffith and G. B. Northcraft, "Cognitive elements in the implementation of new technology: Can less information provide more benefits?" MIS Quarterly, vol. 20, no. 1, pp. 99–110, 1996.

[2] C. C. Chen, C. C. H. Law, and S. C. Yang, "Managing ERP implementation failure: A project management perspective," IEEE Transactions on Engineering Management, vol. 56, no. 1, pp. 157–170, 2009.

[3] M. J. Liberatore and W. Luo, "Coordination in consultant-assisted IS project: An agency theory perspective," IEEE Transactions on Engineering Management, vol. 57, no. 2, pp. 255–269, 2010.

[4] D. Robey, D. L. Farrow, and C. R. Franz, "Group process and conflict in system development," Management Science, vol. 35, no. 10, pp. 1172–1191, 1989.

[5] R. Hekkala and C. Urquhart, "Everyday power struggles: Living in an IOIS project," European Journal of Information Systems, vol. 22, no. 1, pp. 76–94, 2013.

[6] S. J. Whitty and H. Maylor, "And then came complex project management (revised)," International Journal of Project Management, vol. 27, no. 3, pp. 304–310, 2009.

[7] R. H. Thayer, A. B. Pyster, and R. C. Wood, "Major issues in software engineering project management," IEEE Transactions on Software Engineering, vol. SE-7, no. 4, pp. 333–342, 1981.

[8] D. I. Cleland, "Organizational dynamics of project management," IEEE Transactions on Engineering Management, vol. EM-13, no. 4, pp. 201–205, 1966.

[9] A. J. Shenhar, "From theory to practice: Toward a typology of project-management styles," IEEE Transactions on Engineering Management, vol. 45, no. 1, pp. 33–48, 1998.

[10] H. J. Thamhain and D. L. Wilemon, "Building high performing engineering project teams," IEEE Transactions on Engineering Management, vol. EM-34, no. 3, pp. 130–137, 1987.

[11] T. O'Leary and T. Williams, "Managing the social trajectory: A practice perspective on project management," IEEE Transactions on Engineering Management, vol. 60, no. 3, pp. 566–580, 2013.

[12] B. J. Galli, "The future of economic decision making in project management," IEEE Transactions on Engineering Management, vol. Advanced Online Publication, p. DOI: 10.1109/TEM.2018.2875931, 2018.

[13] J. Steen, R. DeFillippi, J. Sydow, S. Pryke, and I. Michelfelder, "Projects and networks: Understanding resource flows and governance of temporary organizations with quantitative and qualitative research methods," Project Management Journal, vol. 49, no. 2, pp. 3–17, 2018.

[14] L. Wylie, "ERP: A vision of the next-generation MRP II," Gartner Group, Scenario S-300-339, April 1990.

[15] S. F. King and T. F. Burgess, "Beyond critical success factors: A dynamic model of enterprise system innovation," International Journal of Information Management, vol. 26, no. 1, pp. 86–97, 2006.

[16] T. H. Davenport, "Putting the enterprise into the enterprise system," Harvard Business Review, vol. 76, no. 4, pp. 121–129, 1998.

[17] J. Frishammar, M. Kurkkio, L. Abrahamsson, and U. Lichtenthaler, "Antecedents and consequences of firms' process innovation capability: A literature review and a conceptual framework," IEEE Transactions on Engineering Management, vol. 59, no. 4, pp. 519–529, 2012.

[18] B. Martinez-Lopez, A. M. Perez, and J. M. Sanchez-Vizcaino, "Social network analysis: Review of general concepts and use in preventive veterinary medicine," Transboundary and Emerging Diseases, vol. 56, pp. 109–120, 2009.

[19] J. Krause, D. P. Croft, and R. James, "Social network theory in the behavioural sciences: Potential applications," Behavioral Ecology and Sociobiology, vol. 62, no. 1, pp. 15–27, 2007.

[20] A. Marin and B. Wellman, Handbook of Social Network Analysis. London, UK: Sage, 2010, ch. Social Network Analysis: An Introduction.

[21] J. Scott, Social Network Analysis, 3rd ed. London, UK: Sage, 2013.

[22] G. Tong, S. Li, W. Wu, and D. Z. Du, "Effector detection in social networks," IEEE Transactions on Computational Social Systems, vol. 3, no. 4, pp. 151–163, 2016.

[23] S. P. Borgatti and X. Li, "On social network analysis in a supply chain context," Journal of Supply Chain Management, vol. 45, no. 2, pp. 5–22, 2009.

[24] S. P. Borgatti, A. Mehra, D. J. Brass, and G. Labianca, "Network analysis in the social sciences," Science, vol. 323, pp. 892–895, 2009.

[25] N. Crossley, "The social world of the network: Combining qualitative and quantitative elements in social network analysis," Sociologica, p. doi:10.2383/32049, 2010.

[26] A. L. Barabasi, H. Jeong, Z. Neda, E. Ravasz, A. Schubert, and T. Vicsek, "Evolution of the social network of scientific collaborations," Physica A: Statistical Mechanics and its Applications, vol. 311, no. 3, pp. 590–614, 2002.

[27] S. L. Lim and A. Finkelstein, "StakeRare: Using social networks and collaborative filtering for large-scale requirements elicitation," IEEE Transactions on Software Engineering, vol. 38, no. 3, pp. 707–735, 2012.

[28] R. Guimera and L. A. Nunes Amaral, "Functional cartography of complex metabolic networks," Nature, vol. 433, pp. 895–900, 2005.

[29] M. O. Jackson, The handbook of social economics. Amsterdam, Holland: North-Holland, 2010, ch. An overview of social networks and economic applications, pp. 511–585.

[30] N. A. Gil, "Language as a resource in project management: A case study and a conceptual framework," IEEE Transactions on Engineering Management, vol. 57, no. 3, pp. 450–462, 2010.

[31] K. Boulding, Conflict and Defense. New York, NY, USA: Harper and Row, 1963.

[32] K. A. Jehn, "A multimethod examination of the benefits and detriments of intragroup conflict," Administrative Science Quarterly, vol. 40, no. 2, pp. 256–282, 1995.

[33] B. M. Gayle and R. W. Preiss, "Assessing emotionality in organizational conflicts," Management Communication Quarterly, vol. 12, no. 2, pp. 280–302, 1998.

[34] N. Gamero, V. Gonzalez-Roma, and J. M. Peiro, "The influence of intra-team conflict on work teams' affective climate: A longitudinal study,"

Journal of Occupational and Organizational Psychology, vol. 81, no. 1, pp. 47–69, 2008.
[35] A. Zornova, P. Ripoll, and J. M. Peiro, "Conflict management in groups that work in two different communication contexts," Small Group Research, vol. 33, no. 5, pp. 481–508, 2002.
[36] K. A. Jehn and E. A. Mannix, "The dynamic nature of conflict: A longitudinal study of intragroup conflict and group performance," Academy of Management Journal, vol. 44, no. 2, pp. 238–251, 2001.
[37] R. Cosier and G. Rose, "Cognitive conflict and goal conflict effects on task performance," Organizational Behavior and Human Performance, vol. 19, no. 2, pp. 378–391, 1977.
[38] A. Amason and H. Sapienza, "The effects of top management team size and interaction norms on cognitive and affective conflict," Journal of Management, vol. 23, no. 4, pp. 496–516, 1997.
[39] K. A. Jehn and P. Shah, "Interpersonal relationships and task performance: An examination of mediating processes in friendship and acquaintance groups," Journal of Personality and Social Psychology, vol. 72, no. 4, pp. 775–790, 1997.
[40] P. Ralph, "Software engineering process theory: A multi-method comparison of sensemaking-coevolution-implementation theory and function-behavior-structure theory," Information and Software Technology, vol. 70, pp. 232–250, 2016.
[41] R. Mohanani, B. Turhan, and P. Ralph, "Requirements framing affects design creativity," IEEE Transactions on Software Engineering, vol. doi:10.1109/TSE.2019.2909033, Advanced Online Publication.
[42] P. Shah and K. A. Jehn, "Do friends perform better than acquaintances? the interaction of friendship, conflict and task," Group Decision and Negotiation, vol. 2, no. 2, pp. 149–166, 1993.
[43] C. Schwenk, "Conflict in organizational decision making: An exploratory study of its effects in for-profit and not-for-profit organizations," Management Science, vol. 36, no. 4, pp. 436–448, 1990.
[44] K. A. Jehn, "A qualitative analysis of conflict types and dimension in organizational groups," Administrative Science Quarterly, vol. 42, no. 3, pp. 530–557, 1997.
[45] D. Damian and J. Chisan, "An empirical study of the complex relationships between requirements engineering processes and other processes that lead to payoffs in productivity, quality and risk management," IEEE Transactions on Software Engineering, vol. 32, no. 7, pp. 433–453, 2006.
[46] N. Ramasubbu and C. F. Kemerer, "Managing technical debt in enterprise software packages," IEEE Transactions on Software Engineering, vol. 40, no. 8, pp. 758–772, 2014.
[47] K. Blincoe, G. Valetto, and D. Damian, "Facilitating coordination between software developers: A study and techniques for timely and efficient recommendations," IEEE Transactions on Software Engineering, vol. 41, no. 10, pp. 969–985, 2015.
[48] A. Amason, "Distinguishing effects of functional and dysfunctional conflict in strategic decision making: Resolving a paradox for top management teams," Academy of Management Journal, vol. 39, no. 1, pp. 123–148, 1996.
[49] R. A. Baron, "Countering the effects of destructive criticism: The relative efficacy of four interventions," Journal of Applied Psychology, vol. 75, no. 3, pp. 235–245, 1990.
[50] B. Staw, L. Sandelands, and J. Dutton, "Threat-rigidity effects in organizational behavior: A multi-level analysis," Administrative Science Quarterly, vol. 26, no. 4, pp. 501–524, 1981.
[51] R. S. Ross, Small Groups in Organizational Settings. Englewood Cliffs, NJ, USA: Prentice Hall, 1989, ch. Conflict, pp. 139–178.
[52] R. E. Walton and J. M. Dutton, "The management of interdepartmental conflict: A model and review," Administrative Science Quarterly, vol. 14, no. 1, pp. 73–84, 1969.
[53] D. C. Wilson, R. J. Butler, D. Cray, D. J. Hickson, and G. R. Mallory, "Breaking the bounds of organization in strategic decision making," Human Relations, vol. 39, no. 4, pp. 309–332, 1986.
[54] K. M. Kaiser and R. P. Bostrom, "Personality characteristics of MIS project teams: An empirical study and action research design," MIS Quarterly, vol. 6, no. 4, pp. 43–60, 1982.
[55] V. Wall and L. Nolan, "Perceptions of inequity, satisfaction, and conflict in task-oriented groups," Human Relations, vol. 39, no. 11, pp. 1033–1052, 1986.
[56] C. Lee, Alternatives to cognition: A new look at explaining human social behaviour. Mahwah, NJ, USA: Lawrence Erlbaum Associates Inc, 1998.
[57] M. Gelfand, G. Shteynberg, T. Lee, J. Luns, S. Lyons, C. Bell, J. Y. Chiao, C. B. Bruss, M. Al Dabbagh, Z. Aycan, A. H. Abdel-Latif, M. Dagher, H. Khashan, and N. Soomro, "The cultural contagion of conflict," Philosophical Transactions of The Royal Society B, vol. 367, pp. 692–703, 2012.
[58] T. L. Roberts, P. H. Cheney, and P. D. Sweeney, "Project characteristics and group communication: An investigation," IEEE Transactions on Professional Communication, vol. 45, no. 2, pp. 84–98, 2002.
[59] D. M. Strenstrom, B. Lickel, T. F. Denson, and N. Miller, "The roles of ingroup identification and outgroup entitativity in intergroup retribution," Personality and Social Psychology Bulletin, vol. 34, no. 11, pp. 1570–1582, 2008.
[60] R. Yin, Case Study Research and Applications: Design and Methods, 6th ed. Los Angeles, USA: Sage, 2018.
[61] S. E. Sim, J. Singer, and M. A. Storey, "Beg, borrow, or steal: Using multidisciplinary approaches in empirical software engineering research, an ICSE 2000 workshop report," Empirical Software Engineering, vol. 6, no. 1, pp. 85–93, 2001.
[62] H. Simons, "The paradox of case study," Cambridge Journal of Education, vol. 26, no. 2, pp. 225–240, 1996.
[63] A. M. Pettigrew, "Longitudinal field research in change: Theory and practice," Organization Science, vol. 1, no. 3, pp. 267–292, 1990.
[64] C. Hanly, "The interplay of deductive and inductive reasoning in psychoanalytic theorizing," The Psychoanalytic Quarterly, vol. 83, no. 4, pp. 897–915, 2014.
[65] M. B. Miles and A. M. Huberman, Qualitative Data Analysis: An expanded sourcebook, 2nd ed. Thousand Oaks, CA, USA: Sage, 1994.
[66] P. Kenis and L. Oerlemans, The Oxford Handbook of Inter-Organizational Relations. Oxford, UK: Oxford University Press, 2010, ch. The Social Network Perspective: Understanding the structure of cooperation, pp. 289–312.
[67] R. A. Krueger and M. A. Casey, Focus Groups: A Practical Guide for Applied Research, 5th ed. Thousand Oaks, CA, USA: Sage, 2015.
[68] C. Robson, Real World Research, 4th ed. Chichester, UK: Wiley, 2015.
[69] S. Wasserman and K. Faust, Social Network Analysis: Methods and Applications. Cambridge, UK: Cambridge University Press, 1994.
[70] A. Martínez, Y. Dimitriadis, B. Rubia, E. Gómez, and P. de la Fuente, "Combining qualitative evaluation and social network analysis for the study of classroom social interactions," Computers and Education, vol. 41, pp. 353–368, 2003.
[71] A. Ambituuni, E. Ochieng, and J. M. Amezaga, "Optimizing the integrity of safety critical petroleum assets: A project conceptualization approach," IEEE Transactions on Engineering Management, vol. 66, no. 2, pp. 208–223, 2018.
[72] V. Braun and V. Clarke, "Using thematic analysis in psychology," Qualitative Research in Psychology, vol. 3, no. 2, pp. 77–101, 2006.
[73] C. Grbich, Qualitative Data Analysis: An Introduction. Newbury Park, CA, USA: Sage, 2012.
[74] J. Attride-Stirling, "Thematic networks: an analytical tool for qualitative research," Qualitative Research, vol. 1, no. 3, pp. 385–405, 2001.
[75] M. Fowler, UML Distilled: A Brief Guide to the Standard Object Modeling Language, 3rd ed. Boston, USA: Addison Wesley, 2004.
[76] R. A. Williams, J. Timmis, and E. E. Qwarnstrom, "Statistical techniques complement UML when developing domain models of complex dynamical biosystems," PLoS ONE, vol. 11, no. 8, p. e0160834, 2016.
[77] Object Management Group, "Unified modeling language superstructure specification v2.4." [Online]. Available: https://www.omg.org/spec/UML/2.4/About-UML/
[78] P. Checkland, Systems Thinking, Systems Practice, 2nd ed. Chichester, UK: John Wiley & Sons Ltd, 1999.
[79] C. Jones, W. S. Hesterly, and S. P. Borgatti, "A general theory of network governance: Exchange conditions and social mechanisms," Academy of Management Review, vol. 22, pp. 911–945, 1997.
[80] G. Walker, "Network position and cognition in a computer software firm," Administrative Science Quarterly, vol. 30, no. 1, pp. 103–130, 1985.
[81] M. Kilduff and W. Tsai, Social Networks and Organizations, 1st ed. Los Angeles, USA: Sage, 2003.
[82] N. Gil, I. D. Tommelein, and L. W. Schruben, "External change in large engineering design projects: The role of the client," IEEE Transactions on Engineering Management, vol. 53, no. 3, pp. 426–439, 2006.
[83] V. V. Adami and J. R. Verschoore, "Implications of network relations for the governance of complex projects," Project Management Journal, vol. 49, no. 2, pp. 71–88, 2018.
[84] S. Rivard and L. Lapointe, "A cybernetic theory of the impact of implementers' actions on user resistance to information technology implementation," in Proceedings of the 43rd Hawaii International Conference

on System Sciences. Honolulu, HI, USA: IEEE Computer Society, 5-8 January 2010.
[85] R. A. Williams, "Modelling conflict within the social networks of large multi-vendor software projects using communicating stream X-Machines," in Proceedings of the European Conference on Artificial Life. Cambridge, MA, USA: MIT Press, July 2015, p. 79.
[86] Standish Group, "Chaos report," Standish Group, Boston, MA, USA, Tech. Rep., 2015.
[87] C. K. W. De Dreu and L. R. Weingart, "Task versus relationship conflict and team effectiveness: A meta-analysis," Journal of Applied Psychology, vol. 88, no. 4, pp. 741–749, 2003.
[88] M. Irfan, M. Hassan, and N. Hassan, "The effect of project management capabilities on project success in pakistan: An empirical investigation," IEEE Access, vol. 7, pp. 39 417–39 431, 2019.
[89] G. Simmel, The Sociology of Georg Simmel, Wolff translation ed. Glencoe, IL, USA: Free Press, 1950.
[90] K. M. L. abd K C C Chan, "Rescuing troubled software projects by team transformation: A case study with an ERP project," IEEE Transactions on Engineering Management, vol. 55, no. 1, pp. 171–184, 2008.

RICHARD A. WILLIAMS (M'16–SM'16) received the Ph.D. degree in computer science (2015) from the University of York, UK. He is an Associate Professor in management science at Lancaster University, UK. He was previously an Applications Technology Consultant and then Project Manager for Oracle Corporation in the UK. His research focuses on complex systems science, with particular emphasis on cybernetics and agent-based modelling and simulation to further our understanding of complex dynamical social systems.

Dr Williams is also a Chartered Professional Member of the British Computer Society, a certified Project Management Professional of the Project Management Institute, and a Fellow of the Royal Society for the Encouragement of Arts, Manufactures and Commerce.