2-2011

Evolution of developer collaboration on the Jazz platform: A study of a large scale agile project

Subhajit DATTA
Singapore Management University, subhajitd@smu.edu.sg

Renuka SINDHGATTA

Bikram SENGUPTA

Citation

1

Evolution of Developer Collaboration on the Jazz Platform

A Study of a Large Scale Agile Project

Subhajit Datta
IBM India Research Lab
Embassy Golf Links, Block D
Bangalore 560071
subhajit.datta@in.ibm.com

Renuka Sindhgatta
IBM India Research Lab
Embassy Golf Links, Block D
Bangalore 560071
renuka.sr@in.ibm.com

Bikram Sengupta
IBM India Research Lab
Embassy Golf Links, Block D
Bangalore 560071
bsengupt@in.ibm.com

ABSTRACT

Collaboration is a key aspect of the agile philosophy of software development. As a software system matures over iterations, trends of developer collaboration can offer valuable insights into project dynamics. In this paper, we study evolution of developer collaboration for a large scale agile project on the Jazz platform. We construct networks of collaboration based on developer affiliations across comments on work items and file changes; and then compare parameters of such networks with established results from networks of scientific collaborations. The comparisons illuminate interesting facets of developer collaboration on the Jazz platform. Such perception helps deeper understanding of the role of interaction in agile projects, as well as more effective project governance.

Categories and Subject Descriptors

D.2.9 [**Software Engineering**]: Management—*life cycle, programming teams*; J.4 [**Social and Behavioural Sciences**]: Sociology

General Terms

Experimentation

Keywords

Jazz, agile development, collaboration, software teams, social network analysis

1. INTRODUCTION

Heraclitus' oft-quoted credo "Nothing endures but change", is as relevant for software development, as it is for our lives-at-large. There is much dynamism in a software project's path from conception to completion. This progression is especially interesting in agile development due its inherently *interactional* nature vis-a-vis the more *instructional* way of conventional, "non-agile" methodologies. The first credo of the *Manifesto for Agile Software Development* proclaims "Individuals and interactions over processes and tools", and one of the 12 guiding principles is "Business people and developers must work together daily throughout the project" [6]. This centrality of interaction in agile development is manifested in collaborations; developers coordinate and cooperate across multiple (and often ad hoc) channels of communication.

We posit that to understand the time dynamics of an agile development project, it is essential to study the variation of certain collaboration parameters over time. In the context of this paper, "evolution" will be taken to mean time variation of such collaboration parameters. Collaboration in agile software development is evinced in *connections* between developers at various levels. Two developers can be connected by their common expression of interest on a particular task or functional area, or by the incidence of both having changed the same file. Given that we have concrete evidence of such connections in project artefacts, a *network* across developers can be constructed to serve as a reliable model for the underlying collaboration. The network paradigm is widely established for studying collaboration between individuals in other areas [28] [2] [3] [32].

As a representative system for studying the variation of collaboration parameters over time, we have chosen a project on the Jazz platform[1] using agile development methodology, as described in detail in a later section. The suitability of our choice is supported by the observation "The Jazz project explicitly focuses on distributed collaboration and has adapted processes and tools to overcome known challenges." [42]. For our study, we define two types of *Jazz collaboration networks* (JCN). JCN-W denotes the network of developers affiliated by work items; JCN-F denotes the network of developers affiliated by files. ("Devleloper", "work item", "file", "affiliation", "JCN" are all explained in a later section.) In the remainder of the paper, whenever we refer to "Jazz development project" it is taken to mean a project on the Jazz platform, most often the project we are studying (as will be clear from the context); whenever we refer to "JCN", it signifies JCN-W and JCN-F collectively.

As a benchmark for comparing the parameters of JCN as they vary over time, we have chosen corresponding trends in the evolution of scientific collaboration networks; two scientists are connected if they have authored at least one paper together [5]. Scientific collaboration networks have been the subject of detailed scrutiny (see Related Work section), mainly due to the availability of reliable data in publication

[1] http://jazz.net

ISEC '11, February. 23-27, 2011, Thiruvananthapuram, Kerala, India

databases, that is essential for constructing the collaboration networks. We describe the benchmark networks in more detail in a later section.

The paper is organized as follows: After this introduction, the research question is presented, followed by an overview of our research contribution. The next section surveys related work. Subsequently, definitions and assumptions are presented which segue into a social network based view of Jazz collaboration. This is followed by description of the system being studied and our methodology. The next sections present results and discussions. We then highlight threats to validity, plans of future work and conclusions.

2. CONTRIBUTION OF THE RESEARCH

Based on the background discussion, we formally present the research question as:

Do time variations of certain parameters of Jazz collaboration networks for a representative project show different trends than corresponding parameters of scientific collaboration networks?

The remainder of this paper is focussed on addressing this question empirically, and offering some explanations for the results. The "certain parameters" that are used to compare the tends of JCN with the benchmarks are explained in detail in the Results section.

By addressing the research question, what is the contribution we seek to establish?

Due to the very nature of agile development, time dynamics of its collaboration are of much consequence to a project's outcome. As a project progresses, its ecosystem evolves in terms of the growing number of artefacts as well as the maturing perception of its stakeholders. There is much that is different between the first and the last releases of a multi-release project, but is not easy to trace the progression of this difference over time. A common approach is to measure how project artefacts, requirements or quality metrics have changed [10], [22], [38]. While these are important measures, they are essentially *symptomatic*. Code, requirements, or quality measures does not change by themselves, they are a manifestation of the changed understanding of the project's cosmos. As agile development is deeply interactive, it is reasonable to conjecture that such changed understanding is reflected in the collaboration characteristics of the most central category of stakeholders – the developers.

A contribution of this paper is to empirically study the time dynamics of an agile project in terms of the collaboration characteristics of its developers, rather than the mutation of project artefacts. The significance of the perspective draws from the centrality of people and their interactions in agile development – "Agile has put the finger on the fact that we need highly motivated and competent people to be successful with software development. ... The focus on people is really what makes agile unique, and this is why agile originally broke through" [19]. To the best of our knowledge this is the first study of a large scale agile project based on evolving developer collaborations.

However, the emphasis on "highly motivated and competent people" provokes its own set of qualms. As early as 2002, in a conversation between DeMarco and Boehm, the former expressed his misgivings about the latter's comment on the need for "premium people" for agile's success. Evocatively, DeMarco asks "What are premium people ...? Are they Nietzsche's supermen? Are they the Alphas that Aldous Huxley wrote about in *Brave New World*?" [14]. These questions have to be addressed at depth if we are to understand the delicate balance between individual latitude and collective commitment in agile projects. In exploring this balance lies the greatest possibilities of agile development, as well as its gravest pitfalls. Collaboration on the Jazz platform provides a valuable opportunity for such exploration, due to the size and complexity of the system under development, as well as the diversity and geographic dispersion of the developers.

To the best of our knowledge, time dynamics of developer collaboration on the Jazz platform have not been studied at depth before; addressing this gap is another contribution of this paper. (Jazz development using constructs of social network analysis have been studied in [42] and [43]; however the objectives of these studies are different from ours.)

The next section places our research in the context of related work.

3. RELATED WORK

As we mentioned earlier, to compare JCN with other collaboration networks, we utilize established criteria of characterizing collaboration networks. The theory of networks has been explored at depth in [41], [21], [24] etc. The incidence of power laws in real world networks and the generation and detection mechanisms for power law behaviour have been investigated in [30], [12]. Key characteristics of social networks are discussed in [31].

The structure of scientific collaborations has been explored at depth by Newman; it is established that these collaboration networks form "small worlds" where pairs of randomly selected scientists are typically short distances away from one another and the networks show significant clustering [28]. Newman extends his enquiry of scientific collaboration networks in two subsequent papers, where the statistical properties of these networks are studied, along with the existence and size of a giant component, and other non-local characteristics such as closeness and betweenness [26], [27]. Newman's work illuminates how each discipline of scientific collaboration show subtly different patterns. These are manifested in the respective network characteristics and usually correlate well with the distinct mores of research in each field. Alberich et al. have studied the "Marvel Universe" collaboration structure (based on a densely populated cosmos of characters of a popular comics-book series) to understand whether a fictional network can closely mimic real world social networks [2].

The theory of networks have been applied in diverse fields. Repositories of open source data have been mined using social network analysis [25]. Puppin and Silvestri study the social network of Java classes and devise a mechanism for ranking classes based on relevance and acceptance [33]. Detection and resolution of bugs have been explored in [11], [20], [4]. Bird et al. have studied socio-technical networks to predict failures [7]. Our present work aligns with the context of socio-technical congruence outlined in this and other papers. Software team dynamics have been studied using an affiliation network based on the bug tracker of a development project in [13].

Interesting real world networks are dynamic – entities and relationships change with time. Evolution of social networks, such as networks of scientific collaborations are examined in [5], [29], [18], [44], [17] etc. Ahn et al. study the topological characteristics of large online social networking services in terms of their degree distribution, clustering coefficients, degree correlation, and evolution over time [1]. Extraction of social networks from academic communities and analysing their implications have been studied in [39], [34].

The evolution of agile teams have been studied in [37], [35], [36]. However, these papers do not focus on the collaborative aspect of agile development.

With the background of this related work, we next clarify the definitions and assumptions for our study.

4. DEFINITIONS

The Eclipse Way of collaboration which guides Jazz development, defines time bound iteration cycles [16]. The iteration plan for each team consists of task descriptions, which are recorded as *work items* [42]. Work items can be of different types, such as plan, user story, task, defect, enhancement, test case, etc. Each work item consists of a set of basic attributes that are useful for tracking it; such as name, unique identifier, description, iteration it has been planned for, creator (name of the team member who created the work item), owner (name of the team member who is responsible for successfully completing the work item), creation date, closure date, priority, estimated effort, actual effort and time spent. The real benefit of a work item based iteration plan, however, accrues from the links that may be established between these items and the corresponding development activity carried out. A work item can be linked to files stored in a configuration management system through the definition of one or more change sets. A change set is a collection of files grouped together by the developer in a manner that is meaningful for the project. The Jazz development platform also allows developers to post comments linked to work items. To define the Jazz collaboration networks (JCN), we are interested in details of *discussions*, *comments*, *developers*, and *files* around work items. Each of these are explained below:

- In Jazz development, work items are the atomic unit of tasks which are assignable and traceable. Work items are classified in different categories such as defects, enhancements, stories etc. "*Commenting on the work items is the main task-related communication channel* ... and they provide the context for communication and collaboration." (italics ours) [42]. So each work item is the epicentre of local collaborations, and a set of work items is the collective foci for a module of development activities. Each work item is owned by a developer who works on the development task.

- A discussion is associated with a work item, consisting of the fields: work item identifier, creator identifier and comment (the textual description of the discussion) and the date the comments was posted. Only one comment can be included in one discussion.

- Each developer is uniquely recognized by an identifier. For our study, a developer is an individual who either comments on at least one work item or changes at least one file. The fact that each developer and each work item in a Jazz development project is uniquely identifiable is very helpful for exploring its collaboration characteristics. In academic and other collaboration networks, often the same individual may be identified by slightly or significantly different names (due to use of initials or change of surnames etc [28], [2]). Profiling individuals accurately to build a reliable social network of academic researchers is an area of recent research [39]. Since there is no scope of such ambiguity in identifying developers and work items in Jazz development, a common threat to the validity of results in similar studies is removed in our case.

- A file is a unit of code that is uniquely recognized by an identifier. Information on which package a particular file belongs to, which work item(s) include the file, which developer(s) have modified the file are also available.

- Given the above definitions, a Jazz collaboration network (JCN) is a network where the vertices (nodes) are developers and edges (undirected links) represent affiliations between developers based either on comments on work items or files changed. Formal definition of "affiliation", and specifications of JCN-W and JCN-F based on it are given in a later section.

- As we are interested in studying the time dynamics of JCN, we define a *time step* as a uniquely identifiable iteration of development of a particular release, for which a JCN can be constructed based on the available information. We have used ten time steps for our study, each time step represents one out of 10 iterations over a period of seven months for the system studied (the System Description section has more details).

5. ASSUMPTIONS

The following assumptions underlie our study:

- All details necessary for constructing the Jazz collaboration networks are available for access at the data repository of the Jazz platform. Often developers collaborate via telephone or face to face conversations. Naturally, the data repository will not have record of such contact. These off-record traces will not strongly affect the validity of our results as our chosen dataset has large number of geographically distributed developers.

- All developers are equal and none is more equal than others! In short, the hierarchy of the development team, whether based on skill, seniority, or some other clout, can be ignored. Although organizational structure may influence collaboration in agile projects, we do not attach any special weight to comments or file changes based on the perpetrator's position in the team hierarchy.

- From the previous assumption, it follows that all comments and all file changes are of equal importance. Some work items commented upon, or some files changed may be more significant from the project's point of view. JCN does not consider any such significance.

- While constructing JCN, no cognizance is taken of the semantics of developer comments or file changes. If we are able to parse every comment and every file change

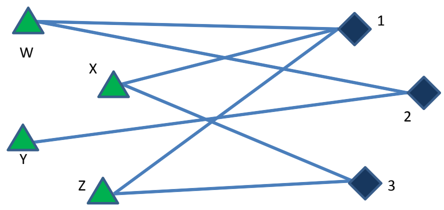

Figure 1: The Bipartite Graph of an Affiliation Network

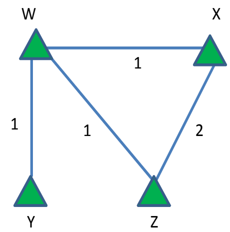

Figure 2: The Social Network Extracted from Figure 1

to extract their "meanings", we will surely know all that is interesting about collaboration dynamics. But the time and effort needed for such an enterprise is prohibitive for non-trivial projects. Our aim is thus to derive interesting insights without delving into such excruciating details.

6. A SOCIAL NETWORK BASED VIEW OF COLLABORATION

The basic idea behind *affiliation networks* is that two types of entities can be perceived in a social context: groups and members, and the two are related by *affiliations*. Affiliation networks are described through bipartite graphs, also known as two-mode networks [23]. The vertices of a bipartite graph are divided into two *disjoint* sets U and V, such that every edge connects a vertex in U to a vertex in V and there are no edges internally between the members of U and V. From the bipartite graph, a social network can obtained by substituting paths of length two among vertices in either set U or V by an edge. This single-mode network only contains vertices from either of set U or V and can be called a *social network* [13]. (We use "social network" in the common sense usage of the phrase, signifying connections between individuals in a social context, which may not fully conform to the rigorous characterization of social networks in [31].)

Figure 1 depicts the affiliation network in the form of a bipartite graph; where vertices on the left (W, X, Y, Z) are individuals and vertices on the right (1, 2, 3) are affiliations (such as membership of clubs, or participation in some joint enterprise etc.) and the links signify memberships. Figure 2 gives the social network arising from the affiliation network of Figure 1. (The weights of the links signify number of co-memberships.)

As mentioned, given an affiliation network in the form of a bipartite graph, we can generate *two* social networks from it: one each for the vertices of U and V. From Figure 1 we have chosen to extract the network where the vertices are individuals (W, X, Y, Z), as depicted in Figure 2. We could have also extracted a network where the vertices are the affiliations (1, 2, 3) [13].

Generating a social network from a bipartite "affiliation" graph is based on the observation that relationships between entities often develop due to some shared affiliation or common interest – such as going to the same club, writing a paper together, being interested in the same sport etc. The study of affiliation networks goes back several decades; Breiger had explored the duality of persons and groups in his eponymous paper in 1974 [8].

With this background, we are now positioned to formally define affiliations in the JCN context. The online *Oxford Dictionaries* define collaboration as "the action of working with someone to produce something". In that spirit, we recognize that collaboration in a Jazz project is manifested by two developers working together to fulfil the project's objectives. Indications of shared interest in a work item or mutual efforts at modifying a file serve as reliable evidence of such "working together". When two developers both comment on a work item or change the same file, their common concern can be recognized as an an affiliation that *connects* the two of them. Such affiliations are defined as:

- **Developer affiliation by work items**

  Let $D = \{d_1, d_2, ..., d_n\}$ denote the set of developers for the period of a project's life cycle being studied. For a developer d_i, let $W_i = \{w_1, w_2, ..., w_x\}$ denote the set of work items which have been commented on by d_i. Developers d_p and d_q are *affiliated* if $W_p \cap W_q \neq \phi$ and the weight of the affiliation A_{pq}, $weight(A_{pq}) = |W_p \cap W_q|$.

- **Developer affiliation by files**

  Let $D = \{d_1, d_2, ..., d_n\}$ denote the set of developers for the period of a project's life cycle being studied. For a work item d_i, let $F_i = \{f_1, f_2, ..., f_y\}$ denote the set of files changed by developer d_i. Developers d_p and d_q are *affiliated* if $F_p \cap F_q \neq \phi$ and the weight of the affiliation A_{pq}, $weight(A_{pq}) = |F_p \cap F_q|$.

The network JCN-W has vertices as developers and its edges denote affiliations by work items; two developers share an edge if both of them have commented on at least one common work item. The network JCN-F has vertices as developers and its edges denote affiliations by files; two developers share an edge if both of them have changed at least one common file. The vertices of JCN-W and JCN-F are the same set of developers.

7. SYSTEM DESCRIPTION

We briefly describe the system and the project team. The system we have studied is a product developed on the Jazz platform using Java and JavaScript programming languages. The system has been developed over several years following the Scrum agile development method. Development using Scrum method progresses through short cycles of releases or iterations. In this study, we examined the development data over 10 time steps each representing one iteration, culminating in a major release. The development team was distributed across multiple countries and exclusively used the Jazz platform for collaboration. The number of developers, work items and files (across the whole set of iterations) was 106; 5,575; and 7,991 respectively.

8. METHODOLOGY

The following methodology was adopted for conducting the study:

1. Use the Jazz platform's Java client APIs used to connect to the project repository. Extract the attributes of a work item such as the owner, creator, iteration it has been planned for, duration, etc. For each work item, extract the files that are modified and the modification date. Extract the comments made by developers for each work item. Persist the data in a specifically designed MySQL database.

2. Compute affiliations as defined earlier between developers and generate JCN-W and JCN-F using a specially developed Java utility for each of the time steps; the networks are cumulative, that is, the network for the second time step include data from the first *and* second iteration, and so on. (To understand the possible influence of too frequently changed files – such as configuration and property files – on the structure of the network, JCN-F was generated for the entire set of files in an iteration, for the entire set of files minus the most frequently changed files, and only for the "functional" files such as Java and Java Server Pages. No appreciable influence on the general network characteristics was detected.)

3. Record the output of the previous step in a *.net file format defined by the open source network analysis tool Pajek [2].

4. For each time step, evaluate JCN parameters of interest (specified in the next section) using Pajek, NodeXL [3] and Gephi [4].

5. Compare time variations of JCN parameters with benchmarks. As mentioned earlier, benchmarks for comparing JCN time dynamics are taken as the co-authorship network of scientists generated from the electronic database of all relevant journals in mathematics and neuro-science for the period 1991-1998, as reported in [5]. Among other discussions, this paper [5] presents the time variation of the following parameters for the research collaboration networks: average separation, clustering coefficient, size of giant component, and average degree [5].

6. Analyse results and suggest plausible explanations for observed trends, and address the research question.

7. Identify threats to validity and scope of future work.

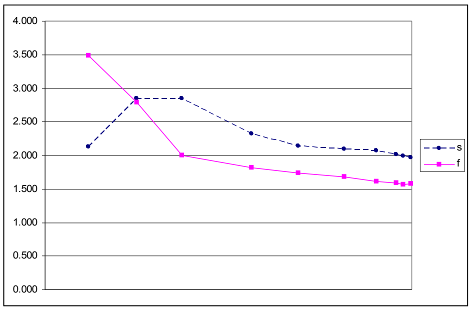

Figure 3: Variation of Average Separation (s) and Scaling Factor (f) across Iterations for JCN-W

9. RESULTS

In the following subsections we describe each parameter for JCN whose variation we are measuring over time and present corresponding results.

9.1 Average separation

In a network, the facility for two vertices i and j to communicate with each other depends on the length of the shortest path l_{ij} between them. The average of l_{ij} over all pairs of vertices in the network is expressed as "average separation", denoted by s [5].

Figure 3 and Figure 4 present trends of the variation of average separation for JCN-W and JCN-F respectively.(We also plot the scaling factor $f = log(N)/log(z)$ for JCN-W and JCN-F in each of the plots, whose implication is explained in the Discussions section; N denotes number of vertices and z denotes average degree, as defined later.) For these and all following plots, the x-axis is scaled by the duration of each iteration; thus the horizontal distance is proportional to the number of days of the respective iteration. As is apparent, the later iterations are of shorter durations. Evidently, for both JCN-W and JCN-F the value of s increases in the first few iterations and then starts coming down, stabilizing towards the latter iterations. *In the benchmark networks, the average separation has been reported to be monotonically decreasing with time* [5].

9.2 Clustering coefficient

It is usually observed in social networks that two vertices that are linked to a third are more likely to be themselves

[2]http://pajek.imfm.si/doku.php
[3]http://nodexl.codeplex.com
[4]http://gephi.org

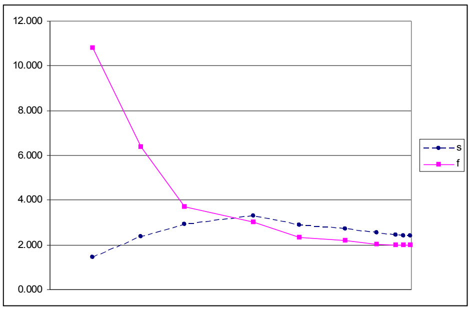

Figure 4: Variation of Average Separation (s) and Scaling Factor (f) across Iterations for JCN(F)

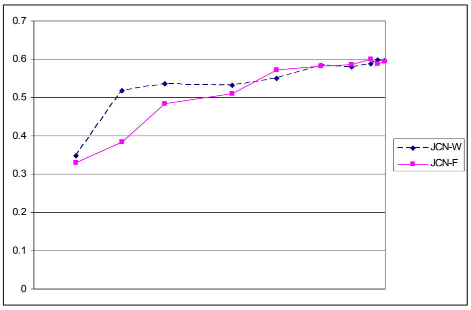

Figure 5: Variation of Clustering Coefficient (C) across Iterations for JCN-W and JCN-F

linked. Intuitively, two of one's friends have a higher probability of being friends themselves. This is measured by the *clustering coefficient*. For a vertex v with a degree k_v, there are k_v neighbours of v. If all of these k_v neighbours were linked, there would be k_v choose 2 or $k_v * (k_v - 1)/2$ links between them. Let N_v be the *actual* number of links between them. Then the clustering coefficient C_v of node v is defined as the ratio of the actual number of links and the maximum number of links between k_v neighbours of v, and is given by [2]:

$$C_v = \frac{2 * N_v}{k_v(k_v - 1)} \quad (1)$$

The clustering coefficient C for the whole network is given by the mean value of the clustering coefficient of all its vertices. It reflects the probability that two neighbours of an arbitrary vertex are directly linked [2].

Figure 5 shows how the clustering coefficient for JCN-W and JCN-F vary with time. It is observed that both for JCN-W and JCN-F the C_v values monotonically increase with iterations, the rate of increase being damped in the latter iterations. *For benchmark networks C_v has been seen to decay with time.*

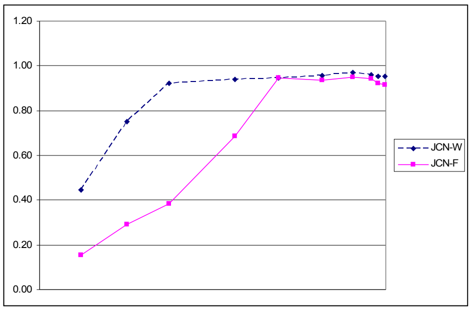

Figure 6: Variation of Size of the Giant Component (P_G) across Iterations for JCN-W and JCN-F

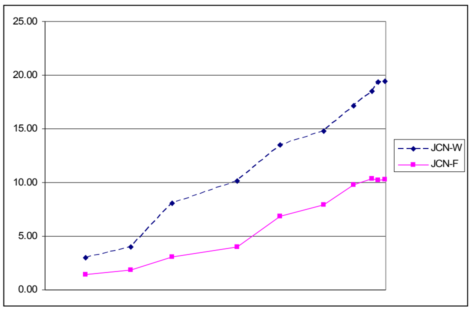

Figure 7: Variation of Average Degree (z) across Iterations for JCN-W and JCN-F

9.3 Size of giant component

In collaboration networks, usually there is evidence of a *giant component* – "large group of individuals who are all connected to one another by paths of intermediate acquaintances" [28]. The percent of vertices of such networks inside the giant (largest) component, denoted by P_G indicates whether a majority of the individuals are closely linked with one another and there are relatively few loosely connected outliers. The giant component is also referred to as the largest cluster.

From Figure 6, it is observed that the size of the giant component increases sharply in the early iterations and then becomes more or less stable close to the maximum value of 1 (on a relative scale) in the later iterations, both for JCN-W and JCN-F. *In the benchmark networks, size of the giant component shows a more gradual increase.*

9.4 Average degree

The *degree* of a vertex is the number of edges incident on it, and is denoted by k. The degree of a vertex in JCN indicates the number of other developers, a particular developer (represented by that vertex) has collaborated with. The average degree z across all vertices of the network thus indicates the mean number of collaborators per developer.

The variation of the average degree of JCN-W and JCN-

F with time is illustrated in Figure 7. As evident, for both JCN-W and JCN-F, z increases sharply, flattening out only during the very late iterations. *For the benchmark networks, the average degree is observed to increase monotonically, though with a lower gradient.*

Now, what are the implications of these time dynamics of average separation, clustering coefficient, size of giant component, and average degree of JCN-W and JCN-F vis-a-vis the corresponding benchmark characteristics as presented above? We address them in the next section.

10. DISCUSSIONS

10.1 The small world phenomenon

As observed in Figure 3 and Figure 4, the average separation for both JCN-W and JCN-F stays approximately around 2. This indicates the Jazz collaboration networks over time remain a "small-world", with roughly two degrees of separation. To establish this point further, we have plotted the ratios of the logarithm of the number of vertices and the logarithm of the average degree in Figure 3 and Figure 4 as the scaling factor (f). With progressive iterations, the plot for f and that of the average separation comes closer for both JCN-W and JCN-F. This signifies that the average degree scales logarithmically with the number of developers in the network by a factor approximately equal to the average separation over time, which is taken to be an evidence that the network is indeed a "growing small world" [18] [28].

Two degrees of separation point to a really small world. In the benchmark networks, the mathematics collaboration network starts with an average separation of almost 16, which comes down to around 9 at the last time period studied; the corresponding range is from 10 to around 5 for the neuro-science collaboration network [5]. It is not surprising that the average separation for JCN-W and JCN-F are significantly less than these values; the mathematics collaboration network was constructed from 70,975 authors across 70,901 papers, while the neuro-science collaboration network considered 209,293 authors across 210,750 papers. In comparison, the numbers of developers, work items and files considered in our study (as mentioned earlier) are several orders of magnitude smaller than the number of authors and papers in the benchmark networks.

10.2 Increasing collaboration

As evident from Figure 7, the average degree for both JCN-W and JCN-F show a progressively increasing trend, with the rate of increase diminishing towards the very late iterations. Increasing average degree, along with a growing giant component (Figure 6) signifies growing collaboration both in terms of commenting on work items as well as changing files. This points to the underlying dynamics of project progression, as each developer takes on new work, new connections with other developers add to his/her existing corpus of collaborations. If this explanation for increasing collaboration is valid, do we find a corresponding reflection in the trend of the clustering coefficient?

10.3 Probability of collaboration

We observe in Figure 5 that the clustering coefficient increases across the iterations both for JCN-W and JCN-F. With reference to the definition of clustering coefficient presented earlier, we notice that increase in its value signifies higher probabilities of collaboration between two developers – both of whom have separately collaborated with another particular developer – with progressive iterations. This seems to complement the explanation for increasing average degree; with each iteration the probability of developer collaboration goes up, and indeed developers collaborate more.

The variation of clustering coefficient with time shows a different trend for JCN vis-a-vis the benchmark networks. For both the mathematics and neuro-science collaboration networks, clustering coefficient decreases over time. This can be explained by the drivers of scientific collaboration. Scientist have specific areas of expertise and they collaborate with other scientist when there is a match of research interests. (Match of research interest may not be the *only* reason why scientists collaborate, but it is certainly a primary reason.) Thus it is unlikely that the clustering coefficient will increase in a growing network of scientific collaboration; merely the fact that there are more scientists and more collaboration does not enhance the probability of two scientists collaborating, even if both of them have a common collaborator. Scientific collaboration is ultimately bound by expertise and interest, both of which are limited at the individual level. And in the era of specialization, scientific research fields are increasingly fragmented, with researchers adhering closely to specific areas. But developers in JCN-W and JCN-F are not collaborating on abstract problems; all of them are working to collectively fulfil a related set of functionalities for a common system. As the system matures over iterations, it becomes more likely that two developers will themselves collaborate, if they have separately collaborated with a third developer; after all, every developer is aligned to the same project's charter. This basic difference between the dynamics of scientific collaboration and collaboration on the Jazz platform, manifests in the distinct time variations of the clustering coefficients of the respective collaboration networks.

10.4 Stability after initial iterations

For all the parameters discussed above, it can be observed from the respective figures that the values appear to stabilize after the fifth or sixth iteration. What is the implication of this trend? As mentioned earlier, we are studying 10 iterations towards a major release of the system. The functionality of the release becomes clearly defined during the middle of the release cycle, no new functionality is subsequently added, and testing and bug fixing commences from then onwards. It is thus expected that the rate of variation of the parameters would also start decreasing around this time.

10.5 Talk versus work

As defined, JCN-W can be seen as a "talk" network (developers *commenting* on work items), and JCN-F as "work" network (developers *changing* files). In a project, it is always interesting to know how the dynamics of talk versus work plays out. Collectively, do developers collaborate more via talk or work? At the individual level, does more collaboration via talk indicate more collaboration via work, or is the opposite true (as often suspected)? Analysing JCN-W and JCN-F across the iterations offers some key insights.

Figure 8 shows the increase in the number of vertices, and

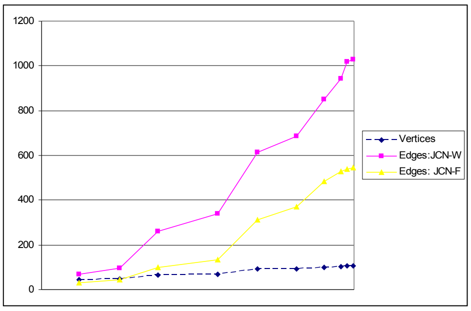

Figure 8: Vertices and Edges across Iterations for JCN-W and JCN-F

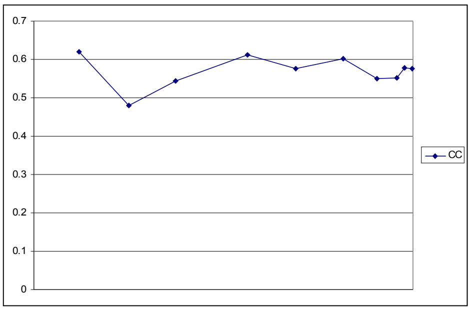

Figure 9: Pearson Correlation Coefficient between the Degrees of the Vertices of JCN-W and JCN-F across Iterations

the edges of of JCN-W and JCN-F across iterations. Between the first and last time step, the number of vertices increases by a factor of more than 2 (45 to 106), the edges of JCN-W increase by a factor of more than 15 (67 to 1029), the edges of JCN-F increase by a factor of around 17 (32 to 546). Also, on average there is close to twice as many edges in JCN-W as in JCN-F across the iterations. An edge in JCN-W denotes collaboration between two developers as manifested in comment(s) on a common work item; an edge in JCN-F denotes collaboration between two developers as manifested in change(s) to a common file. So, developers collaborate approximately twice as many times through talk as they do through work. This factor of difference is influenced to some extent by the limited definition of "work" in this study; we recognize as work only the changing of files. Given this definition, it is thus not unexpected that developers will reach out to other developers more through talk than through work.

In Figure 9 we have plotted the Pearson correlation coefficient between the degrees of the vertices of JCN-W and JCN-F for each iteration. The coefficient varies between 0.619 and 0.479 with an average of 0.569. Thus collaboration via talk and collaboration via work show a moderate positive correlation.

10.6 Implications for project governance

How are the trends discussed above significant from a project governance point of view?

A small-world network with a degree close to 2 implies that on average all the developers are closely connected to one another. This is likely to facilitate low communication overheads, and may in some cases mitigate the influences of Brooks' oft-quoted thesis in his *Mythical Man Month* essay [9] and its more recent examinations [15]. Increasing collaboration amongst developers with progressive iterations is an indication that as the release deadline comes closer, the development team gets more closely aligned to collective objectives of the project. Increasing average degree and clustering coefficient, and growth of the giant component all point to growing collaboration as the release cycle moves forward. Absence of similar trends in some project can be indicative of special circumstances requiring management attention. Similarly, if the parameters do not stabilize even after half the iterations planned for a release have been completed (unlike JCN-W and JCN-F of the project studied in this paper), it can be symptomatic of significant oscillations in the system's functionality as well as the team's collaboration structure. The observations of more collaboration by way of talk than work, and positive correlations between collaborations by talk and work, may have specific implications for particular projects. However, these general trends are reflective of the deeply interactional nature of agile development and any wide variations in a given project may indicate the need for corrective action.

11. THREATS TO VALIDITY AND FUTURE WORK

In our study, we have compared both JCN-W and JCN-F with benchmark research collaboration networks from mathematics and neuro-science. JCN-W and JCN-F, though both being collaboration networks of same developers for the same project on the Jazz platform, are subtly different in their nature. The foundation for JCN-F is doing something together, and is congruent to the joint authoring of a research paper. On the author hand, JCN-W is more akin to the network of participants in an online discussion forum. Thus a more fair benchmark for comparing JCN-W would have been with the participant networks of such a group where individuals are affiliated by their interest in common topics. To the best of our knowledge, there is no established study of such a network. While computing the parameters of comparing networks, weights of edges have not been considered. (For JCN-F the weight of an edge would be the *number* of files the two developers sharing the edge have both worked on; and correspondingly for JCN-W.) This was motivated by the point of view that significant insights from networks can be gleaned by considering undirected, unweighed edges [40]. Consideration of weights are unlikely to change the values of the parameters we analysed, other than perhaps the giant component.

In our future work we plan to extend the definition of JCN-F to address a larger scope of collaboration between developers through work. With the extended definition, we expect to be better positioned to compare collaborations through talk vis-a-vis through work, by comparing the characteristics of the corresponding networks. Additionally, we are interested in building and validating models for the evolution of JCN-W and JCN-F, which would help establish a theoretical basis for the empirical results presented in this paper.

12. CONCLUSIONS

In this paper we have studied the evolution of developer collaboration for an agile project on the Jazz platform across 10 iterations of development culminating in a major release. We constructed networks of developers collaborating by commenting on work items (JCN-W) and developers collaborating by changing files (JCN-F) and compared parameters of JCN-W and JCN-F such as average separation, clustering coefficient, size of giant component, and average degree with networks of scientific collaborations as studied in [5]. Based on our results, the research question can be answered as: Jazz collaboration networks are similar in some parameters while dissimilar in other parameters, when compared to benchmark scientific collaboration networks. The similarities as well as differences were explained in terms of the distinct genesis and dynamics of JCN versus the scientific collaboration networks. In summary, Jazz collaboration networks are "small-worlds" with approximately two degrees of separation, developers are likely to collaborate more as iterations proceed towards a release, parameters of collaboration change rapidly in earlier iterations than later ones, collaboration via talk is nearly twice more frequent than collaboration via work, and the former collaboration is positively correlated to the latter. Our study illuminates interesting aspects of the interactional nature of agile development which have notable implications for governance of software development projects.

13. REFERENCES

[1] Ahn, Y., Han, S., Kwak, H., Moon, S., and Jeong, H. Analysis of topological characteristics of huge online social networking services. In *Proceedings of the 16th international conference on World Wide Web* (Banff, Alberta, Canada, 2007), ACM, pp. 835–844.

[2] Alberich, R., Miro-Julia, J., and Rossello, F. Marvel universe looks almost like a real social network. *cond-mat/0202174* (Feb. 2002).

[3] Amaral, L. A. N., Scala, A., Barthelemy, M., and Stanley, H. E. Classes of behavior of small-world networks. *cond-mat/0001458* (Jan. 2000).

[4] Aranda, J., and Venolia, G. The secret life of bugs: Going past the errors and omissions in software repositories. In *Proceedings of the 2009 IEEE 31st International Conference on Software Engineering* (2009), IEEE Computer Society, pp. 298–308.

[5] Barabasi, A. L., Jeong, H., Neda, Z., Ravasz, E., Schubert, A., and Vicsek, T. Evolution of the social network of scientific collaborations. *cond-mat/0104162* (Apr. 2001). Physica A 311, (3-4) (2002), pp. 590-614.

[6] Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., aAndrew Hunt, Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., and Thomas, D. Manifesto for agile software development. `http://agilemanifesto.org/principles.html` Last accessed: July 26, 2010, 2001.

[7] Bird, C., Nagappan, N., Devanbu, P., Gall, H., and Murphy, B. Putting it All Together: Using Socio-Technical Networks to Predict Failures. In *Proceedings of the 17th International Symposium on Software Reliability Engineering* (2009), IEEE Computer Society.

[8] Breiger, R. The duality of persons and groups. *Social Forces 53*, 2 (1974), 190, 181.

[9] Brooks, F. P. *The Mythical Man-Month: Essays on Software Engineering, 20th Anniversary Edition.* Addison-Wesley, 1995.

[10] Capiluppi, A., Fernandez-Ramil, J., Higman, J., Sharp, H. C., and Smith, N. An empirical study of the evolution of an Agile-Developed software system. In *Proceedings of the 29th international conference on Software Engineering* (2007), IEEE Computer Society, pp. 511–518.

[11] Chen, I., Yang, C., Lu, T., and Jaygarl, H. Implicit social network model for predicting and tracking the location of faults. In *Proceedings of the 2008 32nd Annual IEEE International Computer Software and Applications Conference* (2008), IEEE Computer Society, pp. 136–143.

[12] Clauset, A., Shalizi, C. R., and Newman, M. E. J. Power-law distributions in empirical data. *0706.1062* (June 2007). SIAM Review 51, 661-703 (2009).

[13] Datta, S., Kaulgud, V., Sharma, V. S., and Kumar, N. A social network based study of software team dynamics. In *ISEC '10: Proceedings of the 3rd India software engineering conference* (New York, NY, USA, 2010), ACM, pp. 33–42.

[14] DeMarco, T., and Boehm, B. The agile methods fray. *Computer 35*, 6 (2002), 90–92.

[15] di Penta, M., Harman, M., Antoniol, G., and Qureshi, F. The effect of communication overhead on software maintenance project staffing: a Search-Based approach. In *2007 IEEE International Conference on Software Maintenance* (Paris, France, 2007), pp. 315–324.

[16] Frost, R. Jazz and the eclipse way of collaboration. *IEEE Softw. 24*, 6 (2007), 114–117.

[17] Horn, D. B., Finholt, T. A., Birnholtz, J. P., Motwani, D., and Jayaraman, S. Six degrees of jonathan grudin: a social network analysis of the evolution and impact of CSCW research. In *Proceedings of the 2004 ACM conference on Computer supported cooperative work* (Chicago, Illinois, USA, 2004), ACM, pp. 582–591.

[18] Huang, J., Zhuang, Z., Li, J., and Giles, C. L. Collaboration over time: characterizing and modeling network evolution. In *Proceedings of the international conference on Web search and web data mining* (Palo Alto, California, USA, 2008), ACM, pp. 107–116.

[19] Jacobson, I. Everyone wants to be agile. `http://blog.ivarjacobson.com/everyone-wants-to-be-agile/` Last accessed: July 28, 2010, 2008.

[20] Jeong, G., Kim, S., and Zimmermann, T. Improving bug triage with bug tossing graphs. In

Proceedings of the 7th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering on European software engineering conference and foundations of software engineering symposium (Amsterdam, The Netherlands, 2009), ACM, pp. 111–120.

[21] Kleinberg, J. M. Navigation in a small world. *Nature 406*, 6798 (2000), 845.

[22] Kumar, M., Ajmeri, N., and Ghaisas, S. Towards knowledge assisted agile requirements evolution. In *Proceedings of the 2nd International Workshop on Recommendation Systems for Software Engineering* (Cape Town, South Africa, 2010), ACM, pp. 16–20.

[23] Lattanzi, S., and Sivakumar, D. Affiliation networks. In *Proceedings of the 41st annual ACM symposium on Theory of computing* (Bethesda, MD, USA, 2009), ACM, pp. 427–434.

[24] Leskovec, J., Kleinberg, J., and Faloutsos, C. Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining* (Chicago, Illinois, USA, 2005), ACM, pp. 177–187.

[25] Lopez-Fernandez, L., Barahona, G., and Robles, G. Applying social network analysis to the information in CVS repositories. In *Proceedings of the Mining Software Repositories Workshop. 26th International Conference on Software Engineering* (Edinburgh, Scotland, 2004).

[26] Newman, M. Scientific collaboration networks. I. network construction and fundamental results. *Physical Review E 64*, 1 (2001), 016131.

[27] Newman, M. Scientific collaboration networks. II. shortest paths, weighted networks, and centrality. *Physical Review E 64*, 1 (2001), 016132.

[28] Newman, M. E. J. The structure of scientific collaboration networks. *cond-mat/0007214* (July 2000). Proc. Natl. Acad. Sci. USA 98, 404-409 (2001).

[29] Newman, M. E. J. Clustering and preferential attachment in growing networks. *cond-mat/0104209* (Apr. 2001). Phys. Rev. E 64, 025102 (2001).

[30] Newman, M. E. J. Power laws, pareto distributions and zipf's law. *cond-mat/0412004* (Nov. 2004). Contemporary Physics 46, 323-351 (2005).

[31] Newman, M. E. J., and Park, J. Why social networks are different from other types of networks. *cond-mat/0305612* (May 2003). Phys. Rev. E 68, 036122 (2003).

[32] Newman, M. E. J., Strogatz, S. H., and Watts, D. J. Random graphs with arbitrary degree distributions and their applications. *cond-mat/0007235* (July 2000). Phys. Rev. E 64, 026118 (2001).

[33] Puppin, D., and Silvestri, F. The social network of java classes. In *Proceedings of the 2006 ACM symposium on Applied computing* (Dijon, France, 2006), ACM, pp. 1409–1413.

[34] Reijers, H. A., Song, M., Romero, H., Dayal, U., Eder, J., and Koehler, J. A collaboration and productiveness analysis of the BPM community. In *Proceedings of the 7th International Conference on Business Process Management* (Ulm, Germany, 2009), Springer-Verlag, pp. 1–14.

[35] Robles, G., Gonzalez-Barahona, J., and Herraiz, I. Evolution of the core team of developers in libre software projects. In *Mining Software Repositories, 2009. MSR '09. 6th IEEE International Working Conference on* (2009), pp. 167–170.

[36] Rowley, D., and Lange, M. Forming to performing: The evolution of an agile team. In *AGILE 2007* (2007), pp. 408–414.

[37] Ruhnow, A. Consciously evolving an agile team. In *AGILE 2007* (2007), pp. 130–135.

[38] Sato, D., Goldman, A., and Kon, F. Tracking the evolution of object-oriented quality metrics on agile projects. In *Proceedings of the 8th international conference on Agile processes in software engineering and extreme programming* (Como, Italy, 2007), Springer-Verlag, pp. 84–92.

[39] Tang, J., Zhang, D., and Yao, L. Social network extraction of academic researchers. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on* (28-31 2007), pp. 292 –301.

[40] Watts, D. Networks, dynamics, and the Small-World phenomenon. *The American Journal of Sociology 105*, 2 (1999), 527, 493.

[41] Watts, D. J., and Strogatz, S. H. Collective dynamics of /'small-world/' networks. *Nature 393*, 6684 (June 1998), 440–442.

[42] Wolf, T., Nguyen, T., and Damian, D. Does distance still matter? *Softw. Process 13*, 6 (2008), 493–510.

[43] Wolf, T., Schroter, A., Damian, D., and Nguyen, T. Predicting build failures using social network analysis on developer communication. In *Proceedings of the 31st International Conference on Software Engineering* (2009), IEEE Computer Society, pp. 1–11.

[44] Wu, B., Zhao, F., Yang, S., Suo, L., and Tian, H. Characterizing the evolution of collaboration network. In *Proceeding of the 2nd ACM workshop on Social web search and mining* (Hong Kong, China, 2009), ACM, pp. 33–40.