

The Small World Network Effect in Software Project Teams

Kevin Peterson

pete1968@umn.edu

Abstract

Team cohesion and the dynamics of team formation are important parts of any project, with software projects being no exception. An interesting aspect of team building is the relationships formed between the team members. Because of these relationships, representing software team members as a graph may be a natural way to explore team dynamics. As team members move between projects, these graphs become more and more connected as team members collaborate and form new relationships. We show that this connectivity, known as the “small world effect,” has a positive impact on team performance when the connectivity levels are moderate. Performance degrades, however, at both very high and very low levels of connectivity. This aligns with similar research findings of non-software teams.

Keywords. Small World, Project Management, Software

1 Introduction

A social network is a graph of people and the connections between them. The dynamics of social networks have proven to be an important research topic for many different areas of study, from academic publication[2], to the success of Broadway musicals[31]. In general, a social network is important to understanding how ideas and influence are spread[17]. Given that, we can begin to investigate the best conditions for optimal group performance within these networks.

A “small world” network is a social network characterized by high clustering of graph nodes paired with a short average path between them[33]. “Clustering” in this context means how closely nodes in the graph are related to each other, or “the friend of my friend is also my friend.” “Average path length”

is the shortest route between all possible nodes, averaged over the entire graph. For example, Milgram, in his seminal study, found that any two people are linked through a chain of friends and acquaintances on average of 6 people long[22], or in other words, the average path length between any two people is 6.

Uzzi [31] expanded on this by exploring how these small world networks correlate to collaboration and creativity. In his study, the network of artists involved in the making of Broadway musicals from 1945 to 1989 were analyzed. Performance was then measured based on financial success and critical acclaim, and statistically compared to the clustering and path length of the graph.

We follow Uzzi’s work, and expand it to software teams. We hypothesize that the qualities of a small world graph that foster collaboration, creativity, and the efficiency of how knowledge is transferred[18] will also apply to software project teams. In this work, we investigate the “small world” phenomenon quantitatively by exploring open source project data, and qualitatively by a series of interviews of subject matter experts, and look to compare Uzzi’s findings to ours.

1.1 Hypothesis

The formalization of the “small world” quality of a graph can be expressed by the *Small World Quotient*, or Q [32, 33]. The value Q is calculated by dividing the clustering factor of a graph by the average path length between nodes. Uzzi explored this value in graphs of Broadway productions[31]. His findings suggest that the success of Broadway productions is greatest when teams have a good mix of new and familiar members. Productions are less likely to be successful when team members are very familiar with each other. Also, the probability of success drops much the same way when there is little familiarity between team members. This indicates that there

is an optimal ratio of new and familiar members on a team – or, an optimal value of Q when related to performance. Thus, we formulate our hypothesis:

Software Project Teams will perform best at moderate levels of Q . This performance, measured by the amount of interest their projects generate, will increase as the level of Q increases for the contributor graph. This will continue up to an optimal Q value, after which performance will begin to degrade. Thus, the performance curve given Q will be inverse U-shaped, and match Uzzi’s findings[31].

2 Methods

2.1 Collection and Storage

To explore our hypothesis, software project data was collected and analyzed. This data was collected using the freely available FLOSSmole[14] datasets of free and open source projects. Two datasets were analyzed – data from the popular open source project site *Freecode*,¹ and another from the open source software repository *SourceForge*.² The Freecode dataset contained data up to September 2013, while the SourceForge dataset was slightly older with data gathered up to June 2009. Both datasets were chosen because they met two basic criteria: (1) they contained an exhaustive list of projects and contributors to those projects, and (2) they provided some sort of quantitative measure of project “popularity.” This popularity metric will be further described in sections below. Results were analyzed using the Python graph processing package NetworkX[11]. All source code pertaining to the collection and analysis can be found at <https://github.com/kevinpeterson/small-world-effect-research>.

2.2 Analysis

The collected data was organized into a graph structure, with the nodes representing the contributors and the edges representing one or more shared project between contributors. For our purposes, “contributors” includes any person that has been identified as being associated with the project by the project hosting site (Freecode and SourceForge). It is important to note that “contributor” does not always imply code contributions – as issue reporting and documentation, among other things, are valid contributions. Various

¹<http://freecode.com/>

²<http://sourceforge.net/>

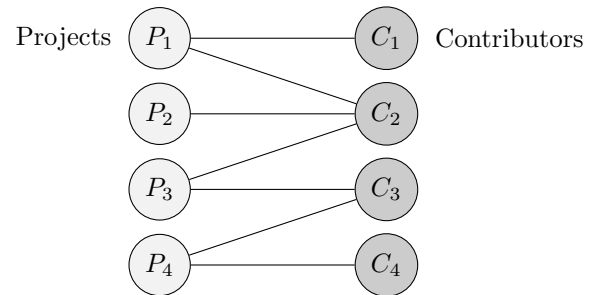


Figure 1: An example bipartite graph of Projects and Contributors

properties of the graph were analyzed, followed by a qualitative section to help link the quantitative analysis to subject matter expert experiences.

Graph Structure

For each dataset, we first considered the dataset to be one large graph. It was observed that the contributor graph was “disconnected,” meaning that there was not a path between all possible nodes. From this large disconnected graph, we extracted a set of connected subgraphs, where each connected subgraph was analyzed. Metrics were then computed for each of these subgraphs. In order to correctly compute the metrics below, a minimum subgraph size was required. Specifically, many of the graph metrics of interest measured “triangles” of nodes and their connections. This implies a subgraph of ≥ 3 nodes is needed. Because of this, subgraphs of < 3 nodes were not considered for analysis.

The contributor/project data can be represented as a bipartite graph. A bipartite graph is a specialized type of graph containing two disjoint sets of vertices. This can be denoted by $G = (P, C, E)$, where $P \cap C = \emptyset$. For our purposes, let set P be the set of all projects, set C be the set of all contributors, and E the edges that connect them. Each project node, therefore, is only connected to contributor nodes, and vice versa, as show in figure 1. In other words, the sets P and C are disjoint. This bipartite representation is typical of what is found in social and collaboration networks[27].

A bipartite graph projection is necessary to further analyze the graph. A bipartite projection involves taking the disjoint node sets P and C , and representing the graph as relationships between only one of those sets of nodes. A projection onto the contributor nodes, or C , of figure 1 is show in figure 2. Here, contributors are connected directly, with edges occurring if a two contributors share one or more common projects. It is important to note that the projection

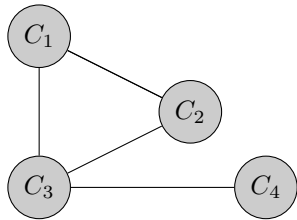


Figure 2: A contributor (C) projection of the figure 1 bipartite graph

could have been done in terms of the projects, instead of the contributors. This would have led to a graph with nodes of projects P , each being linked by sharing a common contributor. Projection is a common way of representing these bipartite graphs[23], although it is known that bipartite projections are lossy compared to the data represented in the original graph[35].

One of the main sources of information loss is multiple connections between contributors, or contributors that share multiple projects. When doing a bipartite projection based on contributors C , a single link could represent one shared project or many. This is a source of data loss in the projection. The simple approach is to treat one connection the same as many connections between nodes. This is straightforward but lossy[35, 10]. A better approach is to assign a *weight* to each edge representing repeated links – or in our case – multiple shared collaborators[34, 3].

Graph Analysis

To support our hypothesis, there are several key metrics to be studied in regards to the aforementioned graph. Many of these metrics follow the work of Boccaletti *et. al*[4], and other studies of similar graph types[18, 1].

Clustering Coefficient - Transitivity (C^Δ)

We calculate the *Clustering Coefficient* (C^Δ) as a measure of the proportion of closed triangles in a graph[25]:

$$C^\Delta = 3 \times \frac{\text{number of triangles}}{\text{number of connected triples}}$$

This metric shows us how closely related, or “clustered,” that the graph is. It is also the probability that two nodes will be connected if they share a common neighbor[24]. In our context, if contributors C_1 and C_2 both collaborate with a common contributor C_3 , this metric represents the probability that C_1 and C_2 will collaborate.

Clustering Coefficient - Average ($\overline{C^\lambda}$)

Another approach to clustering is to take the average of each node’s local clustering coefficient[33]. To calculate the local clustering coefficient of a node, we follow these steps:

Given our graph of nodes C connected by edges E , or $G = (C, E)$, for any given node we can determine its *neighborhood* (N_i), or the set of nodes directly connected to a given node.

$$N_i = \{v_j : e_{ij} \in E \wedge e_{ji} \in E \wedge v_j \in C\}$$

Given this *neighborhood*, we can then calculate how closely connected the nodes are. To do this, we take the number of actual connections between neighbors divided by the total possible number of connections, where k_i represents the count of the neighbors of a node, or $k_i = |N_i|$

$$C_i^\lambda = \frac{2|\{e_{jk} : v_j, v_k \in N_i, e_{jk} \in E\}|}{k_i(k_i - 1)}$$

From this *Local Clustering Coefficient* C_i^λ , we can then take the average over the entire set of nodes in the graph.

$$\overline{C^\lambda} = \frac{1}{|C|} \sum_{i \in C} C_i^\lambda$$

It is important to note that although the terminology is similar[31], $\overline{C^\lambda}$ and C^Δ are different measurements. Unless otherwise noted, the Transitivity version of this metric (C^Δ) will be used for all further calculations and metrics.

Average Path Length (L)

Calculating the *Average Path Length* (L) allowed us to determine how far removed contributors in the graph are from one another. The path length is calculated as the average distance between any two node pairs in the graph:

$$L = \frac{1}{|C| \cdot (|C| - 1)} \sum_{i \neq j} \lambda(v_i, v_j)$$

Where given $v_i \in C$, $v_j \in C$, the function $\lambda(v_i, v_j)$ denotes the shortest path between the two nodes. We only considered path length for connected subgraphs, avoiding some complexities of this calculation on disconnected graphs[4].

Small World Quotient (Q)

The definition of a *small world* graph can be formalized by the following equations[15, 31]:

First, let γ_g^Δ equal the ratio of the clustering coefficient of a given graph and a random graph.

$$\gamma_g^\Delta = \frac{C_g^\Delta}{C_{rand}^\Delta}$$

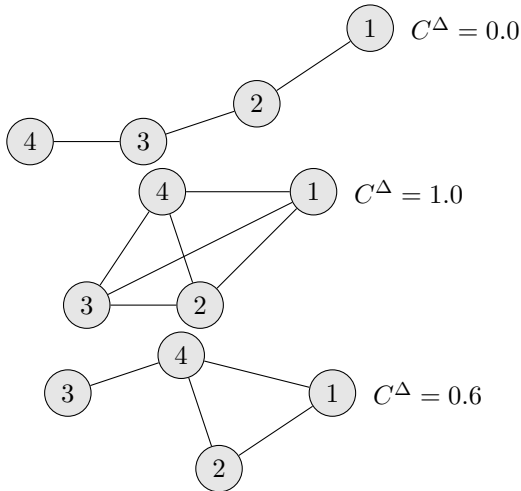


Figure 3: Example Clustering Coefficient (C^Δ) values for sample graphs

Next apply a similar pattern to the average path length λ_g .

$$\lambda_g = \frac{L_g}{L_{rand}}$$

Finally, the ratio of the above calculations yields the small world quotient, or Q , of the graph.

$$Q = \frac{\gamma_g^\Delta}{\lambda_g}$$

We assume throughout that any graph exhibiting $Q > 1$ is a small world graph.

Performance

To further explore the data, a measurement of project performance was needed. As the explored datasets consisted entirely of open source projects, research indicates that *quality*, *use*, *user satisfaction*, and *impact* are each possible metrics to measure performance or *success*[8].

For both datasets, we based our performance metric on *use*. For Freecode, we used the *Popularity Score* metric, which is calculated by the following formula:³

$$((record\ hits + URL\ hits) \cdot (followers + 1))^{1/2}$$

For the SourceForge dataset, performance was based on the repository's internal *rank* metric. This metric ranks each project from 1 to the total number of projects in SourceForge (P_{sf}), or $1 \leq rank \leq |P_{sf}|$. At the time of this writing, no

³<http://help.freecode.com/kb/statistics/how-do-you-measure-a-projects-popularity>

formula for the *rank* metric could be located for citation. Note that there was no attempt to normalize the two performance metrics between datasets. Because of this, no cross-dataset measurements can be made, as the performance metric is only valid in the context of the enclosing dataset.

Qualitative Analysis

So far we have explored, via data analysis, how closely project collaborators are connected to one another, and what correlations we can derive from this analysis. We have said little as to why these connections are made, how they are maintained, and what exactly a connection means at a personal level. Via qualitative analysis, we will attempt to enrich our data analysis by providing deeper context.

A connection between two people is much more than an edge between two nodes in a graph. It has been asserted that people tend to make connections with people with whom they share similarities[21]. This idea of homophily, however, needs to be further expanded before we can make assertions about correlations to performance. It has been shown that demographic homophily is complex to measure, correlates little to performance outcomes, and in fact correlates negatively[29, 19]. For our purposes, it is much more appropriate to examine how teams best leverage similarities of knowledge and information, as the flow of these assets can be a key driver to success[26].

In the same way we can explore the role of team familiarity. The two types of familiarity that we will explore are *member* familiarity and *task/project* familiarity[12]. These metrics, although not captured in data, could be important factors in *how* teams form, and thus help us to understand *why* teams perform the way that they do.

To further study these factors, we arranged interviews with two current professionals in the field. Both have had experience with project management and are certified Project Management Professionals (PMP®). We proposed to them the following questions, with the intent of expanding further the questions posed above:

Q1: How do development teams collaborate and share ideas?

Q2: Do teams generally remain intact over multiple projects, or do members often shuffle and re-combine?

Q3: When assembling a team, which is preferred - members familiar with each other, or members

familiar with the task or project?

Q4: What are some characteristics of a high performing team?

Q5: In your experience, is it better to (A) keep an innovative team intact, (B) periodically bring in new members or re-assign existing members, or (C) split up and disperse the team to as many other projects as possible?

3 Results

3.1 Quantitative

Table 1: Freecode Statistics

Var	Mean	Med	Max	Min	Std
Q	1.36	1.0	6.38	0.85	0.72
C^Δ	0.91	1.0	1.0	0.18	0.19
$\overline{C^\lambda}$	0.85	1.0	1.0	0.14	0.25
L	1.12	1.0	2.32	1.0	0.26
N	4.06	3.0	16.0	3.0	2.12
E	5.34	3.0	78.0	2.0	7.48

P: 41121, P_c : 156, C: 21970

Table 2: SourceForge Statistics

Var	Mean	Med	Max	Min	Std
Q	1.44	1.0	8.04	0.78	0.69
C^Δ	0.95	1.0	1.0	0.19	0.12
$\overline{C^\lambda}$	0.91	1.0	1.0	0.08	0.19
L	1.09	1.0	3.84	1.0	0.22
N	5.38	4.0	194.0	3.0	5.27
E	15.88	6.0	8778.0	2.0	100.88

P: 163244, P_c : 9532, C: 222920

Summary statistics for the graph analysis are captured in tables 1 and 2. Statistics analyzed included:

Q - Small World Quotient

C^Δ - Clustering Coefficient - Transitivity

$\overline{C^\lambda}$ - Clustering Coefficient - Average

L - Average Path Length

N - Number of Nodes

E - Number of Edges

The results shown were as measured over the number of projects P , with the total number of contributors C . The number of connected subgraphs analyzed is shown as P_c . The connected subgraphs are the main

unit of analysis, and each statistical operation is applied to the set of subgraphs P_c .

A mean value of Q was observed to be > 1 for both data sets, which meets the criteria for a small world network[15]. The median, however, is 1.0 in both cases, which does not match the small world network criteria. A possible explanation of this is the power-law distribution of Q observed in both data sets (figures 6 and 7). This indicates that a large proportion of Q values are around 1.

C^Δ and $\overline{C^\lambda}$ both indicate values approaching 1. This observation indicates a high degree of clustering[33]. Exploring this phenomenon further, we can derive that most graphs conceptually resemble the middle or the bottom graph in figure 3, with relatively few resembling the top graph.

Note that P_c values are relatively small compared to P and C values. It would be intuitive to assume that $N \times P_c \approx C$ - in other words, if we split the entire graph into connected subgraphs, the number of subgraphs (P_c) multiplied by the average number of nodes (N) per subgraph should approximately equal the total number of contributors (C). This, however, is not the case. In fact, P_c is notably smaller than expected. Recall that during analysis, subgraphs of < 3 nodes were not considered. We can assume then that a large number of subgraphs are very small, with only one or two nodes. Further analysis is needed to fully understand the implications of this observation.

To validate the hypothesis, we expect to find that intermediate values of Q will be correlated with more popular projects. To explore this possibility, we compare the popularity given Q values as shown in figures 4 and 5. The Q values do display the inverse U-shaped characteristic as compared to popularity, which aligns with the findings of Uzzi[31], and supports our hypothesis.

Distribution of Q values are shown in figures 6 and 7. The values of Q generally follow a power-law distribution, with most values being at our around $Q = 1$. This indicates that many graphs are close to the delineation of being considered small world[15].

3.2 Qualitative

The research interviews provided a look into how some of these quantitative findings translate into actual industry practice. They also helped us to obtain a deeper understanding of what we had found in the data analysis, as we could begin to build some workplace context into our results. There were several re-occurring themes from which we will base our analysis[30, 28]:

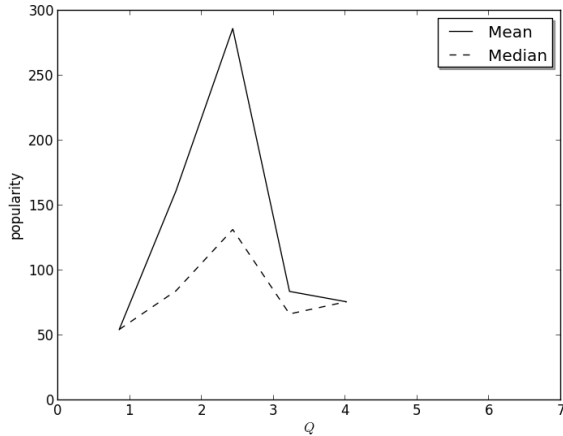


Figure 4: Freecode Q and Project Popularity

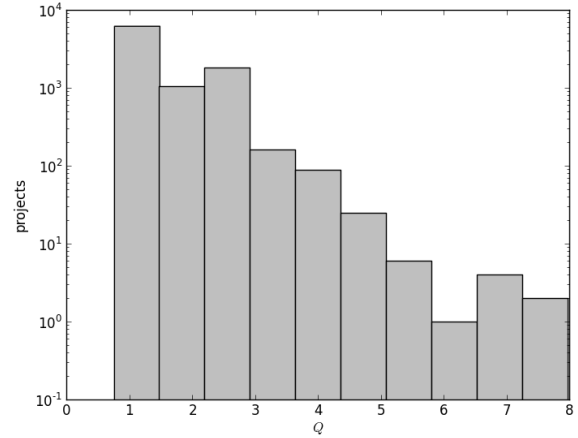


Figure 7: SourceForge Q Distribution

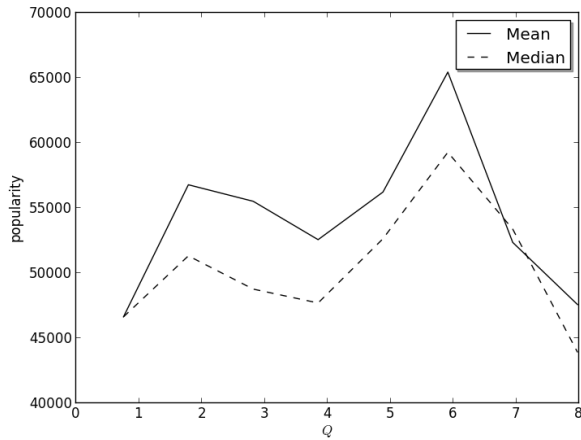


Figure 5: SourceForge Q and Project Popularity

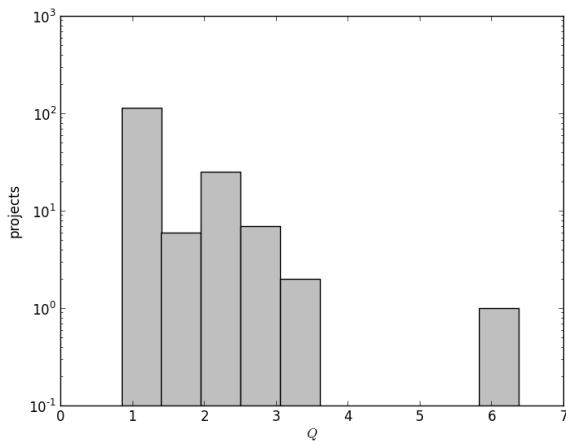


Figure 6: Freecode Q Distribution

New contributors are good, as long as the core team remains intact.

Both participants mentioned the idea of keeping a “core” team intact, and it was an often cited concept in both interviews. This concept appears to align with the idea of a *core/periphery*[5], where the graph under study contains one (or many) tightly interconnected subgraphs, with less tightly connected nodes radiating out from this core.

Although our data analysis does not include any examination of a core/periphery, it is an interesting direction for future research. Notably, has been suggested that individual performance in a group is highest if that individual is not in the core or the periphery, but somewhere in between[6]. This is especially interesting as it aligns in principle with our hypothesis that intermediate values of Q are optimal.

Teams can stagnate without new members.

Both interview participants indicated that new members were essential in bringing new ideas to the team. A highly static team (or, a team with a high Q value), runs the risk of falling behind the technology trends. Occasional new members (or, intermediate values of Q) were perceived as a positive in terms of information flow.

Collaboration is key to performance.

Both interviewees suggested that team collaboration had a direct impact on performance. Teams that showed good collaboration both intra-team and inter-team were noted for their positive performance characteristics. Further exploration of this topic is needed to determine more fully the nature of this collab-

oration. Specifically, we know that high conflict is detrimental to team performance, especially in complex work environments[9]. A further research direction may be whether or not good team collaboration/communication promotes conflict avoidance, conflict resolution, or otherwise.

Priority, focus, and timeline are strong drivers.

Focus was specifically called out as an indicator of performance in one interview. Clear goals and a common objective were positive influencers, while continuously shifting focus was a negative influence. Research in this area suggests that creative teams have “goals that are clear and compelling, but also open and challenging.”[16]

Timeline was discussed as a strong factor in team success. Team building could be thought of as function of the timeline – meaning that as the timeline of the project increased, more time could be devoted to team building activities.

It was also indicated that priority was a major factor. Different project priorities, and the resultant shifting of resources, had the possible effect of weakening a strong team’s core members. This can be related to focus as well, as changing priorities can cause resource realignment and decreased overall focus.

4 Conclusion

Our hypothesis theorizes that medium values of clustering between collaborators on projects is optimal for project performance. The data analysis, specifically figures 4 and 5, seem to support this claim. Also, our qualitative research tends to support the hypothesis as well. Our interviews revealed that from experience, teams perform best when a stable core is periodically augmented with new members. This would be a qualitative representation of an intermediate Q value – significant clustering but not to the extent of isolation. Given both the quantitative and qualitative analysis, we can assert to align with Uzzi’s findings[31] and show that intermediate values of Q are correlated with increased project performance.

5 Future Research

Social networks, interpersonal relationships, and software development are all complex topics. We have explored a narrow slice of these in context, and there are some important areas of future exploration. First, the contributor graph contains a large amount of sub-

graphs, and as a whole appears less clustered than some other observed social network graphs – consistent with Madey’s findings[20] on SourceForge data. As noted in our analysis, a large number of these subgraphs have only one or two contributors. This implies that a large number of contributors work alone and on a single project. We do not, with the current analysis, know how (or if) this degrades the quality of our findings.

A further area of study would be how our findings relate to previously defined *success factors* in projects[7]. Also, we do not know directly *how* intermediate values of Q influence performance – we can only observe a quantitative and qualitative correlation. We make a tacit equivalence to clustering and knowledge sharing, but we have not directly observed that in our analysis. Research does suggest that social factors are knowledge sharing motivators[13], but how that relates specifically to our work is unexplored here.

6 Acknowledgments

We thank Praveen Mittal and John Kaman for their insight, guidance, and instruction, M Iftekhar (Ifte) Rana and Traci St. Martin for their time and expertise, and Rick Kiefer for his review.

References

- [1] Lada A Adamic. The small world web. In *Research and Advanced Technology for Digital Libraries*, pages 443–452. Springer, 1999.
- [2] Albert-Laszlo Barabási, Hawoong Jeong, Zoltan Nédá, Erzsebet Ravasz, Andras Schubert, and Tamas Vicsek. Evolution of the social network of scientific collaborations. *Physica A: Statistical Mechanics and its Applications*, 311(3):590–614, 2002.
- [3] Alain Barrat, Marc Barthelemy, Romualdo Pastor-Satorras, and Alessandro Vespignani. The architecture of complex weighted networks. *Proceedings of the National Academy of Sciences of the United States of America*, 101(11):3747–3752, 2004.
- [4] Stefano Boccaletti, Vito Latora, Yamir Moreno, Martin Chavez, and D-U Hwang. Complex networks: Structure and dynamics. *Physics reports*, 424(4):175–308, 2006.

- [5] Stephen P Borgatti and Martin G Everett. Models of core/periphery structures. *Social networks*, 21(4):375–395, 2000.
- [6] Gino Cattani and Simone Ferriani. A core/periphery perspective on individual creative performance: Social networks and cinematic achievements in the hollywood film industry. *Organization Science*, 19(6):824–844, 2008.
- [7] Terry Cooke-Davies. The real success factors on projects. *International journal of project management*, 20(3):185–190, 2002.
- [8] Kevin Crowston, Hala Annabi, and James Howison. Defining open source software project success. In *in Proceedings of the 24th International Conference on Information Systems (ICIS 2003)*, pages 327–340, 2003.
- [9] Carsten KW De Dreu and Laurie R Weingart. Task versus relationship conflict, team performance, and team member satisfaction: a meta-analysis. *Journal of applied Psychology*, 88(4):741, 2003.
- [10] Jerrold W Grossman and Patrick DF Ion. On a portion of the well-known collaboration graph. *Congressus Numerantium*, pages 129–132, 1995.
- [11] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using NetworkX. In *Proceedings of the 7th Python in Science Conference (SciPy2008)*, pages 11–15, Pasadena, CA USA, August 2008.
- [12] David A Harrison, Susan Mohammed, Joseph E McGrath, Anna T Florey, and Scott W Vanderstoep. Time matters in team performance: Effects of member familiarity, entrainment, and task discontinuity on speed and quality. *Personnel Psychology*, 56(3):633–669, 2003.
- [13] Paul Hendriks. Why share knowledge? the influence of ict on the motivation for knowledge sharing. *Knowledge and process management*, 6(2):91–100, 1999.
- [14] James Howison, Megan Conklin, and Kevin Crowston. Flossmole: A collaborative repository for floss research data and analyses. *International Journal of Information Technology and Web Engineering*, 1(3):17–26, 2006.
- [15] Mark D Humphries and Kevin Gurney. Network small-world-ness: a quantitative method for determining canonical network equivalence. *PLoS One*, 3(4):e0002051, 2008.
- [16] Scott G Isaksen and Kenneth J Lauer. The climate for creativity and change in teams. *Creativity and Innovation Management*, 11(1):74–86, 2002.
- [17] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146. ACM, 2003.
- [18] Vito Latora and Massimo Marchiori. Efficient behavior of small-world networks. *Physical review letters*, 87(19):198701, 2001.
- [19] Barbara S Lawrence. Perspective the black box of organizational demography. *Organization Science*, 8(1):1–22, 1997.
- [20] Greg Madey, Vincent Freeh, and Renee Tynan. The open source software development phenomenon: An analysis based on social network theory. In *Americas conf. on Information Systems (AMCIS2002)*, pages 1806–1813, 2002.
- [21] Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, pages 415–444, 2001.
- [22] Stanley Milgram. The small world problem. *Psychology today*, 2(1):60–67, 1967.
- [23] Mark EJ Newman. Scientific collaboration networks. i. network construction and fundamental results. *Physical review E*, 64(1):016131, 2001.
- [24] Mark EJ Newman. Properties of highly clustered networks. *Physical Review E*, 68(2):026121, 2003.
- [25] Mark EJ Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003.
- [26] Mark E Nissen. An extended model of knowledge-flow dynamics. *Communications of the Association for Information Systems*, 8(18):251–266, 2002.
- [27] José J Ramasco, Sergey N Dorogovtsev, and Romualdo Pastor-Satorras. Self-organization of collaboration networks. *Physical review E*, 70(3):036106, 2004.
- [28] Iftexhar Rana. interview, Nov. 27 2013.

- [29] Ray Reagans, Ezra Zuckerman, and Bill McEvily. How to make the team: Social networks vs. demography as criteria for designing effective teams. *Administrative Science Quarterly*, 49(1):101–133, 2004.
- [30] Traci St. Martin. interview, Nov. 21 2013.
- [31] Brian Uzzi and Jarrett Spiro. Collaboration and creativity: The small world problem. *American journal of sociology*, 111(2):447–504, 2005.
- [32] Duncan J Watts. *Small worlds: the dynamics of networks between order and randomness*. Princeton university press, 1999.
- [33] Duncan J Watts and Steven H Strogatz. Collective dynamics of small-worldnetworks. *nature*, 393(6684):440–442, 1998.
- [34] Hongyuan Zha, Xiaofeng He, Chris Ding, Horst Simon, and Ming Gu. Bipartite graph partitioning and data clustering. In *Proceedings of the tenth international conference on Information and knowledge management*, pages 25–32. ACM, 2001.
- [35] Tao Zhou, Jie Ren, Matúš Medo, and Yi-Cheng Zhang. Bipartite network projection and personal recommendation. *Physical Review E*, 76(4):046115, 2007.

Appendix A: Interview Transcript

Interview with Traci St. Martin, PMP®, Mayo Clinic, Nov 21, 2013.

Q1: How do development teams collaborate and share ideas?

Co-location is important for teams, as it really fosters communication and collaboration. Being able to quickly talk with someone and resolve a problem is very important. In some cases, having the team in the same room is beneficial, especially if the project is going badly. That way, ideas can get discussed and problems resolved very quickly. Regular meetings are also important with large teams. With these meetings, it is important to have the correct tools to facilitate communication. These meetings help projects have visibility into each other. Management support is essential to this communication process. Meetings between teams can consist of all members of the teams if the teams are small enough, but most of the time should just be a representative.

Q2: Do teams generally remain intact over multiple projects, or do members often shuffle and re-combine?

I have had both. It really depends on the projects and the situation at the time. It can be nice to have new people brought in to make things different, and to bring some new thoughts and ideas to the team. Even one person can change the dynamics of a team, either positively or negatively. As a project manager, I have to be watchful of this, as negativity can spread quickly. I have also noticed that as people move around to different projects, they are generally brought up to the level of the team. For example, if somebody joins a high performing team, they tend to elevate their skill to match that of the group, which is a good thing.

Q3: When assembling a team, which is preferred – members familiar with each other, or members familiar with the task or project?

Familiarity is good, but you have to account for burnout if people work on the same project for too long. This really depends on project dynamics, special skills needed, etc. It is important to note, however, that team members can be familiar with each

other but not get along well, or not work well with each other. I have had projects where team members were very familiar with each other, but did not function well as a team. Clashing personalities and other team problems can cause a toxic environment. Most importantly, whether the team is familiar with each other or not, is that they can share knowledge and skills.

Q4: What are some characteristics of a high performing team?

Most teams I notice go through the Forming, Storming, Norming and Performing stages. Good teams usually have a strong leader, a good vision, and good management support. Of course, they also possess the necessary skills for the project (or, if not, are willing to learn). Good teams have respect for each other, their leadership, and their management. Professionalism is also an indicator of a high performing team. Teams that truly care about their job, the team, and the project are generally more successful. Teams are generally more successful if the individuals themselves are skilled and talented. Being able to react to change is important, and good teams generally have good change management procedures.

Q5: In your experience, is it better to (A) keep an innovative team intact, (B) periodically bring in new members or re-assign existing members, or (C) split up and disperse the team to as many other projects as possible?

In general, I like to keep good teams intact. It is, however, nice to bring in or add a new member to a team. One strategy I like to employ is to bring in a new member, train them and let them learn from the high performing team, and move them out of the project and into a new one. This keeps the core team intact, which is important. Priorities can greatly influence this, and sometimes personnel moves are driven from higher management priorities. If experts are removed from a high performing team, the team may suffer. There is, however, a risk that knowledge becomes concentrated. High priority/high visibility teams have a tendency to get the highest performing individuals, which can cause other projects to suffer.

Appendix B: Interview Transcript

Interview with M Iftekhar (Ifte) Rana, PMP®, Mayo Clinic, Nov 26, 2013.

Q1: How do development teams collaborate and share ideas?

In my previous experiences, there were tight dependencies between teams and changes could have large impacts. Because of this, team collaboration was important and was stressed. Because of the tight dependencies, teams were involved in each other's planning phases. We found that bringing teams together during planning helped collaboration and communication. Also, we noticed that the quality of communication and collaboration between teams had a direct impact how teams performed, and how the project fared as a whole.

Q2: Do teams generally remain intact over multiple projects, or do members often shuffle and re-combine?

It is more of a problem when focus changes rapidly. If there is a stable focus for the project, and the goals of the project remain relatively intact, shifting of personnel has less impact. Continuously changing focus definitely has a negative impact on team performance. Also, when focus is changing rapidly, it is hard to maintain a stable team core, as they tend to get re-assigned or loaned out to different projects.

Q3: When assembling a team, which is preferred – members familiar with each other, or members familiar with the task or project?

This really depends on the time-line of the project. Short-term projects don't have as much time for team bonding – it is more important to get the people with the right skill set. For longer term projects, you have to be much more careful about people getting along with each other. For these types of projects, you can afford to invest some time into getting the team right and doing some team building. Also, with long term projects, you have more of a chance of members "learning on the job," so even if the technical skills aren't there right away, you can factor in some learning time if they are a good fit personality-wise. It is the same reason why we ask a lot of behavioral questions for new full-time employees, and less so with

contractors. It all depends on the time commitment and how much team building you can plan for.

Q4: What are some characteristics of a high performing team?

The number one most important factor for a high performing team is good communication skills. Good teams that I've worked are highly connected, and all members of the team communication with each other. This leads to feedback coming from many different angles, which helps teams get better. Team performance is usually tied directly to how well a team communicates and shares knowledge with each other.

Q5: In your experience, is it better to (A) keep an innovative team intact, (B) periodically bring in new members or re-assign existing members, or (C) split up and disperse the team to as many other projects as possible?

It is usually good to keep teams intact if they are performing well. That said, if you do not keep growing the team, they will quickly lose touch with changes in technology. New members help to bring this into the team, and if a team is isolated they often don't keep up. Technology trends move very quickly, so it is essential to have fast communication channels so that information can spread quickly. Even if new/different members aren't coming into the team, there still should be communication channels open. Communication between teams should happen whether or not teams are actually working together. In general, it is best to periodically bring in new members, but keep the core team intact if possible.