A Network Analysis of Online Forum Discussions on Executable Acceptance Test Driven Development

Shelly Park, Frank Maurer
University of Calgary, Department of Computer Science, Calgary, Alberta, Canada
{sshpark, fmaurer}@ucalgary.ca

Abstract

We performed a network analysis on an online discussion forum that discussed ideas for the next generation of testing tools for executive acceptance test driven development. We used a social network analysis to discover the underlying core concepts that in building their product vision. We used Edge-Betweenness algorithm to discover what consensus is reached and Degree Centrality algorithm to discover who are the central people facilitating the consensus in the product visioning process. We discovered three core concepts: Exploratory vs. Test automation, Business vs. Technology and *Communication. We also discovered that we can identify polarizing issues in the community by identifying the people with different degree centrality in the social network. Our research shows that a social network analysis a good way discover the characteristics of consensus reached during a product visioning process.*

1. Introduction

In Executable Acceptance Test Driven Development (EATDD), requirements are communicated using acceptance tests instead of using a natural language. The purpose of the executable acceptance tests is to facilitate better communication among all stakeholders including customers, developers and testers. Instead of an ambiguous requirements specification using a natural language, EATDD requires that requirements must be written in a testable form that can either succeed or fail. EATDD is becoming popular in the agile software engineering community and practitioners are trying to find tools that can facilitate EATDD better.

A distributed group of industry practitioners and academic researchers are collaborating online through discussion forums and blogs to envision what the next executable acceptance testing tool should behave like. This paper looks into the product visioning process for the next generation of executable acceptance testing tools and analyzes the consensus building process.

Unlike Test-Driven Development that is based on unit testing, which primarily impacts developers, EATDD must involve all stakeholders including customers, domain experts, developers and testers. People from different backgrounds and skills have different expectations about how one should create these tests and communicate the requirements to each other. Therefore, the issues involved in EATDD are much more complex than unit testing. While such collaboration between different people has a high potential for productive and innovative outcomes, chances for misunderstanding can also be very high.

Recently, the Agile Alliance organized two workshops to envision what EATDD testing tool should behave like [1, 2]. These loosely associated groups of volunteers are pursuing the discussion over an online discussion forum [3]. The online discussions are based on pure anecdotal evidences and single expert opinions. The benefit of driving innovation through online forums is the Wisdom of Crowds [4]. Science is a way to gain insights into the world and how it works. While the scientific approach of hypothesis generation and empirical validation is valid, it is also slow. The anecdotal evidence provided by expert groups is an alternative way to create insights. We then can use a social network analysis to extract this meaning and validate it in part by determining how consensus is reached. This approach can be faster in obtaining insights into the problem.

The purpose of this paper is to understand whether there is a structure to these discussions and whether there is consensus within this community. We wanted to discover if there are core concepts and issues that are fundamental to this product visioning process that are equally important to all stakeholders and bind this community together. We expect that there is a collection of issues or concepts that concerns all stakeholders in this product visioning process.

The paper is organized as follows. In Section 2, we describe the literature survey. In Section 3, we present our research hypothesis. In Section 4, we describe the research methodology, analysis and how we measured data. In Section 5, we present our data and the analysis. We present the implication of our research in Section 6. We conclude our findings in Section 7.

2. Literature Survey

Rittel and Webber defined a *wicked problem* as a problem where figuring out what the problem is the actual problem. Wicked problems have no stopping rule; solutions to wicked problems are not true or false, but good or bad; there is no immediate and no ultimate test of a solution to a wicked problem [5]. Figuring out the best software requirements for a new product is a wicked problem and it is a very difficult problem to solve.

The tool that this online group of people is trying to describe/specify is an executable acceptance testing tool or a story testing tool. The Agile Alliance has organized a few workshops [1,2] to organize a forum for a group of interested people to start envision the tool specification together. Recently, Andrea also published an article describing what her vision of such tool looks like [6].

The first generation of executable acceptance testing tool is Fit [7]. Practicing EATDD in real development projects was much harder, because these tests needed to be read, written and maintained by a group of stakeholders who came from different backgrounds and skills. There are also many different names for this practice. It is also known as functional tests [8], customer tests [9] and specification by example [10] among many. Despite many names, the idea is to write the requirements in a testable form and hook up the software implementation to the tests using *fixtures* and execute the tests automatically to ensure regression testing.

2.2 Grounded Theory

One of the methods used for reduction of text to code is grounded theory [11]. In order to build our network graph, we need to generate a set of manageable core concepts from text available on the online forum [3]. We used grounded theory [12, 13] to analyze and to reduce the discussion text to code. Grounded theory is a bottom-up research process where we start with data and see what theories/concepts arise out of that data. There are three types of coding: Open coding, Axial coding and Selective Coding. Open coding is the process of developing categories of concepts and themes emerging from data. This phase is about exploring data. Axial coding is to build connections between categories. Selective coding is to refine coded data into structured relationships and categories. Our coded data is used along with the social network information to discover whether there is a core concept that is driving the community. A few researchers combined grounded theory and a network analysis before [14, 15]. Different disciplines use different methods for the network analysis. We decided to combine grounded theory with more rigorous network analysis based on graph theory for our purpose.

2.1 Network Centrality

We applied a network analysis on our coded data obtained from the message boards [3]. *Centrality* is an important concept that assigns “an order of importance on the vertices or edges of a graph by assigning real values to them”. The purpose of centrality indices is to quantify an intuitive feeling that some vertices or edges on a network are more central than others [16]. In Centrality analysis, we are trying to discover the *vertex central* from *vertex peripherals*. In order for a graph to be analyzed for centrality, the vertices must be reachable. *Reachability* is defined as “the number of neighbors or the cost it takes to reach all other vertices from it” [16], which is also called the *degree centrality*. It measures how many neighbors are connected to the vertex. For a graph $G=(V,E)$ with n vertices, the degree centrality $C_D(v)$ for vertex v is: $C_D(v)=\frac{\deg(v)}{n-1}$. The definition of centrality for a graph is: Let v be the node with the highest degree centrality in G. Let $G'=(V',E')$ be the n node connected graph that maximizes: $H=\sum_{j=1}^{|V'|} C_D(v')-C_D(v_j)$. Then the degree centrality of the graph G is $C_D(G)=\frac{\sum_{i=1}^{|V|} C_D(v)-C_D(v_i)}{H}$. We used degree centrality to find a group of *central* people who are facilitating the communication and influencing this community either as an idea leader or an idea radiator.

2.2 Cluster Analysis

Clustering is a method of decomposing a set of entities into natural groups [16]. Cluster analysis is an exploratory data technique that is used to explain scattered data, especially when one tries to explain mathematical model behind data derived from social or empirical data [16]. Cluster analysis is used when one is dealing with the types of problems where one wants to *explore* scattered data to discover whether a pattern of a structure exists in the data. Compared to other mathematical models that are used to discover such patterns such as *discriminant analysis, factor analysis, mixture resolving* or *dispersion analysis*, Cluster analysis allows the researcher to discover the patterns even with the most general problem statement and measurement techniques because its main aim is to reduce the "feature dimensionality" of a search space [17]. Clustering can be applied using many different techniques depending on which discipline it is used in. Some of the techniques used in computer science, especially in the field of Artificial Intelligence, include genetic algorithms and neural networks [117].

In this paper, we used the Betweenness Centrality metric [18]. For a graph $G = (V, E)$ with n vertices, betweenness $C_B(v)$ for vertex v is: $C_B(v) = \sum_{\substack{s \neq v \neq t \in V \\ s \neq t}} \frac{\sigma_{st}(v)}{\sigma_{st}}$.We used Edge-Betweenness algorithm, or also known as Girvan-Newman algorithm [18]. We used this algorithm because it is an algorithm that is used often in a social network analysis and serves our purpose. In this paper, we used a cluster analysis to discover a set of core concepts that are important to all stakeholders.

3. Research Hypothesis

Most network analyses are based on the Power Law [19]. The Power Law assumes that there is a strongly connected core in the network. The participants of the online forum come from many different backgrounds: testing, development, business analysis, project management, teaching and research. We suspect that they emphasize different issues, especially when certain topics are much more relevant to their current job. These people also work and develop software for different industry domains. So our first hypothesis is that (1) there are clusters of concepts that are divided by people's roles in the software development process. (2) However, we also hypothesize that there is one core cluster of concepts that are shared by most participants no matter what their background are.

People do not always engage in all discussions and they do not openly reach consensus on what is important to everyone. Often, some people are simply silent about their opinions because they do not agree with the issues being discussed (or they could be simply too busy to care). Therefore, simply counting the frequency of topics does not provide a good indication of consensus reached by the community. Therefore, we need to analyze the social networks of the community to understand what issues are shared and agreed by the community in the product visioning process. We hypothesize that we can gain much better insight on issues using a social network analysis.

4. Research Design

Our research began when we participated in the first Agile Alliance Functional Testing Tool workshop [1]. This community keeps track of each other's progress mainly through a message board [3]. We started our data collection by going through the entries in the message board. The very first message starts on Sep 28, 2007. The data collection ended on December 2, 2008. At the time, there were a total of 536 messages.

First, we performed open coding on the message entries. We found that there were 226 articles that discussed important issues or concepts. The remaining articles were about announcements, workshop organizations and messages with no important discussions. The collection of these messages constituted over a thousand pages. Out of that list, we generated about 300 open codes to describe the contents. However, these 300 open codes were too granular and described too many details about the specific tool implementation features that we need to do further coding to reduce down to big concepts. Through axial and selective coding, we reduced the discussions down to 22 categories that can explain most of the contents discussed in the mailing list. The 22 categories are presented in Section 5. And then we assigned 226 articles into 22 discussion categories. One article may be assigned to more than one category. We decided to work with 22 broader categories, because we wanted to discover a general trend in the discussion rather than specific features that people proposed.

We called the people who proposed and discussed the 22 topics as "experts" in those categories and we discovered that there are 36 "experts". We use the term "expert" loosely. It simply means they are interested in the topic and they hold some kind of opinions on that topic. Some people appeared in more than one category, but nobody appeared in all of the categories.

4.1 The Research Design for Cluster Analysis

To obtain the core underlying concepts that are relevant to everyone in this community, we used 22 categories to form a network graph. 22 categories are used as vertices and the edges represent people's interest. Then we performed *Edge-Betweenness algorithm* on the graph. The tool we used is called JUNG [20]. This algorithm iteratively removes edges from the graph and reveals more strongly connected vertices. As we perform more iteration, we eliminated vertices that are not strongly connected to other vertices and eventually we are left with a set of vertices that are strongly connected to all other issues. Semantically, it means each time we apply the next iteration of the algorithm on the graph, we eliminate less interesting concepts. The final remaining clusters of vertices are referenced and cross-referenced by most of the participants in the community either directly or indirectly through other issues. Therefore, these final clusters are the concepts are relevant and interesting to everyone in the social network.

The aim of the cluster analysis is to figure out which of 22 categories are relevant to everyone in the online community. We want to discover the underlying concepts that are fundamental to all of the discussions in this community. If we find that there is more than one cluster of categories, then it means the community is separated by different interests and expertise. If there is only one core cluster, then it means most participants share similar ideas and interests.

4.2 The Research Design for Degree Centrality

Our second analysis is designed to figure out the person (or people) with the highest degree of centrality. The purpose of the degree centrality analysis is not necessarily to find the person with the most number of new and innovative ideas, but the person who has the most critical social connections to help communicate the ideas across different disciplines, or to find the "deal breaker" in the community. It is also equally possible that the people who are occupying the central position are simply well versed in many disciplines and share a lot of interests with many people. We wanted to see if we can use degree centrality to discover concepts that are more polarizing than others due to the division in peoples' opinions.

In the second experiment, the 36 "experts" are represented with vertices. Each time a person shares the same interest as another person, we connected two people with an undirected edge. We performed Degree Distribution Ranking on the graph [20]. This algorithm measures the strength of connections. It returns a local measure of the connectivity to its neighbors. The graph of Degree Distribution Ranking on our data is available in Section 5.

5. Results

5.1 Discussion Categories

In this section, we are going to present 22 categories that were derived from the coding process. These categories summarize the major issues that were discussed by the people in the online forum. They are *Team Involvement, Adoption, Test Maintenance, Economic Value, Regression Testing, Compatibility/Integration, Usability, Communication, Business vs. Technology Problems, Knowledge Representation, Notation/Language, Graphical Visualization, Architecture, Completeness, Distributed Tests, Different Perspectives/Skills, Exploratory vs. Test Automation, Workflow, Abstraction, Terminology, Reporting* and *Validation vs. Verification.* We suspect that some people might have guessed some of these categories, but no one would have guessed this entire list without the analysis. We included one quotation from each category in Table 1 to support why these categories are relevant to this community. We were only able to present one quotation each due to the page limit.

Because this is an online forum, some topics had a very biased representation. For example, *Economic Value* was worded negatively only. They were suggesting the difficulty of justifying EATDD to the team. No one gave a counter argument. However, some topics were given both sides of an argument. For example, *Exploratory vs. Test Automation* had a very heated discussion about what is test automation and how much should be automated. Some topics were proposed, but they were simply ignored by the community or misunderstood, such as *Validation vs. Verification.* The community quickly moved onto another topic before it received much recognition.

Table 1: A Sample of Quotations to Support the Relevance of Proposed Categories

Cat.	Quotations
Team Involvement	"How to get different parts of the organization - PM, devs, testers - engaged. And how I failed in this" #2
Adopt.	"Selling such a kind of tool is like attending to hit two balls on the same 'swing'. You have to sell the practices and sell the tool at the same time" #41
Test Maint.	"I think teams need to understand the importance of maintainability in both their product code and their test/fixture code." #247
Economic Value	"There were a couple of anti-patterns that tended to tip the ROI into negative

	territory.” #249
Regress. Testing	“You see the focusing benefit sooner - during the implementation of a story. Whereas the reference benefit comes after the story has been implemented.” #263
Compat. / Integr.	“A shared vision of the most important next steps is… Better IDE integration? More "productized" tools ([…] RubyFIT with Fitnesse on a Mac […])” #30
Usabilit.	“[I’m] a proponent of paper prototyping and wizard of oz testing on agile projects (code isn’t the only thing that can be tested!)” #391
Commu nicat.	“Communicate and Learn seems to me most important project goals and tools on the project should support them.” #169
Biz vs. Tech.	“I think it's important that acceptance tests be expressed in language, diagrams, whatever, that are independent of the technology.” #131
Knowl. Repr.	“There are two types of knowledge: you can ‘know how’ to act or you can ‘know that’ a fact is true. Computers deal in the latter; experts deal in the former” #5
Notat./ Lang.	“I’m heavily influenced by Brian’s use of dynamic language for testing.” #23
Graph. Visualiz	“We were trying to make the graphical specification more specific…and made it executable…” #58
Arch.	“It seems that we could run some parts of the ATs at the unit level, could be at the services level, could be at the GUI level. Each has their benefits and drawbacks.” #217
Complet eness	“I think that implying logical completeness is asking for trouble.” #61
Distr. Tests	“But the idea of product-quality seems to be very deep and distributed.” #302
Diff Persp. /Skills	“I really think it's a better perspective for looking at the problem. To see it from a requirement perspective, not a test perspective.” #79
Expl. vs. Test Auto.	“I do not think the skills [in TDD] are the same as traditional testing skills, nor the same as exploratory testing skills.” #222
Workf.	“We instead should focus on building tools that support a workflow. When faced with dilemma between making a tool more flexible or more simplistic, we choose path by asking ‘which support the Agile workflow better’? #131
Abstract .	“This is all to do with the continuum between data, information, knowledge and

	potentially even wisdom.” #198
Termin.	“On the other hand, we shouldn't eliminate the word 'test' from our vocabulary, because the 'executable examples' generally aren't sufficient to be considered a full test suite.” #196
Report.	“Difficulty ensuring sufficient visibility and repeatability of results across the organization - Inadequate reporting, meaningless failures,…, need for archival and comparison of historical test result….” #104
Valid. vs. Verific.	“System and Integration testing, however, are more concerned with the issue of 'Verification' than 'Validation'” #200

5.2 Cluster Graph Analysis

In this section, we are going to show consensus using a social network analysis. Some concepts are more important to some people than others and some people have more power to influence others. Therefore, we performed network analysis on our 22 categories.

First, we counted the number of times these concepts were discussed in the forum. The frequency is available in Table 2 under the “number of messages” column. *Different Perspectives/Skills* appeared the most frequently with 34 appearances. However, simply counting the number of occurrences may not provide deeper insights about the community consensus as not everyone may be participating in these discussions.

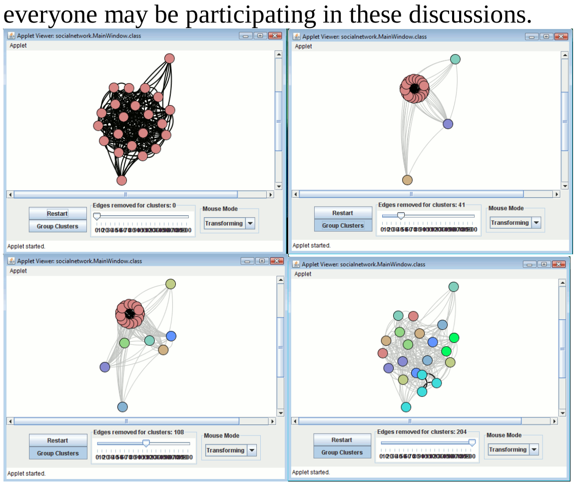

Figure 1: Four graphs showing how the graph was transformed after iterations of Edge Betweenness algorithm. The top left graph is the initial graph, and the bottom right graph is the final graph.

Therefore, we performed cluster analysis and graphed them using JUNG [20]. Figure 1 shows a few snapshots of the graph transformations after applying iterations of the Edge-Betweenness algorithm. Figure 1 is only meant to show visually the trend of the transformation. The detailed results are found in Table 2. As discussed in 4.1, the vertices represent the categories of concepts and edges represent people's interests. The graph starts iteration zero with one big core cluster of vertices (22 categories) that are strongly connected. As we remove more weak edges from the graph (which means removing less interesting concepts by the community), we can see that vertices leave the cluster one at a time. At the end, we are left with three vertices that are strongly connected together: Exploratory vs. Test Automation, Communication, and Business vs. Technology. It means these three final vertices are at the core of everyone's interest. Different colors mean they belong to different clusters. As you can see, there is only one cluster at any given point in time and no sub-clusters were formed.

Table 2: Ranked Order of Important Concepts Using Edge-Betweenness Algorithm

Rank	Rnk by Freq	Concept	# of Msg	# of Edges Removed
1	2	Expl. vs. Auto	23	204
1	4	Communication	19	204
1	18	Biz vs. Tech.	3	204
2	3	Usability	22	202
3	8	Abstraction	16	199
4	18	Distributed Tests	3	197
5	13	Graph. Visual.	8	192
6	1	Diff. Persp./skills	34	188
8	5	Adoption	17	179
9	10	Workflow	12	173
10	6	Compat./Integrat.	14	165
11	12	Architecture	8	154
12	16	Valid. vs. Verific.	5	141
13	9	Team Involvement	12	132
14	17	Reporting	4	119
15	6	Terminology	19	102
16	7	Economic Value	15	87
17	9	Completeness	14	72
18	15	Test Maintenance	5	57
19	8	Notation/Language	14	37
20	14	Regression Testing	6	19
21	11	Knowledge	11	9

The application of the algorithm on our data shows that there are three fundamental core concepts that interest everyone in the community: Exploratory vs. Test Automation, Communication, and Business vs. Technology. As the vertices leave the graph, they do not form additional clusters or sub-clusters. Semantically, a lack of sub-clusters means that the entire community is actually more homogenous despite their differences.

Because we observed only one cluster, we conclude that the participants share similar interests and this is surprisingly a very homogeneous community. It is also interesting to note that these vertices left the core cluster one at a time as we applied subsequent iterations of the algorithm. It means there is a clear ranking of "interestingness" to thee participants in a community. The lack of sub-clusters in our graph shows that there are no strongly divided sub-groups of individuals who are interested in specialized topics. It also means that there is a core group of individuals who are leading this group (or a core set of idea leaders), which leads to our second analysis on who these people.

5.4 Degree Centrality

The purpose of the second analysis is to discover the degree centrality of the people who are involved in the discussion. Figure 2 shows the Degree Distribution Ranking graph that was graphed using JUNG [20]. As mentioned before, we wanted to find the people who are connected to the most number of people.

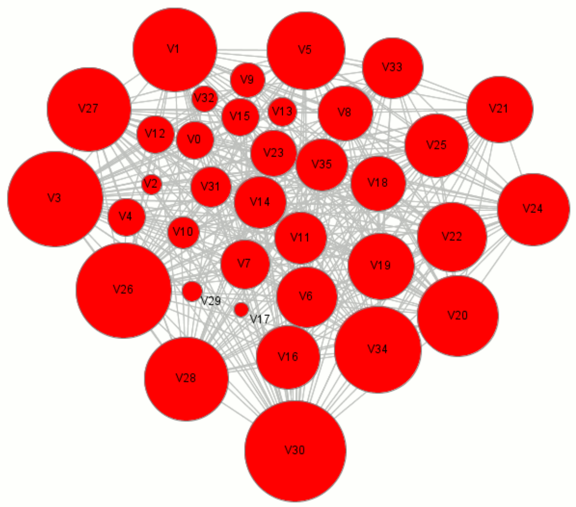

Figure 2: The Degree Distribution Ranking of the Participants

In the graph in Figure 2, the vertices are the "experts" and the edges are their interests. Bigger vertices mean two things: (1) they are connected to most number of people due to their vast breadth of interests; (2) they are critical in spreading ideas because of their highly focused and specialized interest. Therefore, this graph is not measuring the persons' innovativeness of idea they suggested. Due to the page limitation, we could not include a table of all of the participants' interests and opinions in this paper, so we will only present interests for some of the people.

As seen in Figure 2, there are about a half dozen people with a highest degree of centrality. They are V1 (Andre Brissette), V3 (Ben Simo), V5 (Brian Marick), V20 (Kay Johansen), V26 (Naresh Jain), V27 (Neil), V28 (Pekka Laukkanen), V30 (Pierre Veragen) and V34 (Stan Taylor). Most people who are ranked at the top only participated in the discussion a few times and have expressed a narrow set of interests in the discussion forum. V5 (Brian Marick) is unique from this list because he was the only person who had a vast breadth of interests and contributed frequently. For example, V5 (Brian Marick) appeared in 18 categories out of 22 categories. He had unique viewpoints in many of these categories. The other six people are positioned at the top, because they occupy a unique position in the social network graph due to their very narrow spoken interest. Therefore, the algorithm has chosen these people to be the bridge between two groups of people. For example, P28 (Pekka Laukkanen) only appeared in *Compatibility/Integration* category and V34 (Stan Taylor) only appeared in *Test Automation.* V27 (Neil) only appeared in two of the most highly discussed topics: *Compatibility/Integration* and *Different Perspectives/Skills*. These people appear infrequently, but their interest lies between a large cross section of people. Therefore, we propose that the topics they appear are some of the controversial or most stimulating topics in the community. The most dominant topics of interest in this ranking are *Different Perspectives/Skills* and *Compatibility/Integration.*

The people who are ranked in the middle of the centrality are the *idea* leaders due to their frequent participation. They are V6 (Elisabeth Henrickson), V8 (David Vydra), V11 (Erik Peterson), V14 (Greg Wilson), V16 (Jean Mcauliffe), V18 (Jim Shore), V19 (Jennitta Andrea), V21 (Kevin Lawrence), V22 (Lisa Crispin), V24 (Mark Levison) and V25 (Matt Heuser), V33 (Ward Cunningham) and V35 (Michael Stockdale). This is a very large list of people and they together have a large range of influence in the community. Most of these people are very vocal about their opinions and they participate often. However, their influence is often counter balanced with another strong idea leader. The competing interest with another contender puts them in the middle of the degree centrality. We couldn't find dominant concepts in this ranking, because there are too many ideas.

The people who are grouped in the lower degree centrality are due to (1) their lack of participation or (2) a lot people already share the same view. Their ideas are shared by many people in the community or there is no strong conflict with their proposal so far. Most notably, V13 (Gerard Meszaros) is categorized in this category. If you look at his posting in [3, #460], he was already able to get consensus for his ideas by the community. V2 (Antony Marcano) also appeared in this category due to his vast breadth of interest in many topics. He has 14 interested categories. Unlike V5 (Brian Marick), V2 (Antony Marcano) seemed to have too many overlaps with rest of the community. These people are likely to be in a good position to facilitate consensus in the community.

6. Implication

6.1 Cluster Analysis

In Section 3, we hypothesized that there will be clusters of concepts that define this community due to the diversity of stakeholders. However, our results show otherwise. There is only one core cluster with three highly ranked concepts (See Figure 1). Semantically, it means this is a very homogenous group and there is not enough diversity in the community.

The cluster analysis reveals interesting phenomena. First, the people in this community share similar "expertise" and interests to the point where their degree of interest can be ranked (See Table 2 for the ranking), which is certainly an unexpected result. We didn't expect this community to be so homogenous. It means that there is the least amount of expertise in the last ranked concept, *Knowledge*. We define *Knowledge* as how one articulates, communicates and transfers the necessary domain knowledge required to develop software. The community doesn't have many domain knowledge experts, which explains why this concept is represented the least in the product visioning discussion. We cautiously conclude that perhaps the tools developed from this community may not be accepted by the domain knowledge experts from various industries due to their lack of representation in this community.

Based on the results from the cluster graph analysis, we conclude that there are three core concepts: Exploratory vs. Test Automation, Communication, and Business vs. Technology. We define *Exploratory vs. Test Automation* as how one should facilitate test automation and how much should be automated. The *Communication* states that the purpose of EATDD is to facilitate better requirements elicitation rather than to facilitate mere test automation. The participants said the tool should strive to become a lightweight requirements communication tool. The *Business vs. Technology Problems* states that there are two types of problems in EATDD and that we should first categorize the problems into a business problem or a technology problem and then come up with the appropriate solutions. Because these three concepts are so strongly connected to the remaining concepts, we conclude that perhaps these three core concepts are the underlying visions for all stakeholders and the remaining 19 concepts are the specifics of how such vision can come true.

The next observation is with the concept, *Business vs. Technology Problems,* which comes as a highly surprising finding. This concept was not discussed much (ranked 18 in frequency). However, it is ranked as first in the cluster graph analysis when we take the social network into account. We discovered that this concept connects all other concepts into the core cluster. If we didn't perform the social network analysis, we wouldn't have discovered how fundamental this concept is to all of the people in this community.

6.1 Degree Centrality Analysis

Our data from the degree centrality analysis shows that sometimes the people who are positioned in the centrality position is not necessarily an idea leader or the most active participant, but a person who bridges a gap between two or more different groups of people in partly due to their highly specialized interest or a unique view of the situation.

We noticed that most of the idea leaders are categorized in the middle tier of the social network, because they are all equally trying to influence and shape the product vision. As our observation shows, being at the lowest tier of the social network also has an advantage. Their ideas are mostly accepted by the community. Perhaps they are the people who are most prepared for the consensus building process.

We found that identifying a person's position in the social network can better identify the polarizing issues in the community and what each person's role may be in the community. We found that the degree centrality analysis using social network is better at putting the findings into social context of the community.

8. Conclusion and Future Work

In this paper, we have shown the importance of analyzing the social network in product visioning process. Our research shows that social network analysis is crucial in discovering consensus within a community. We found that *who* proposed the ideas for the product features is as important as *what* is proposed, because we need to put our findings into the social context and social influences in the community. For our future work, we intend to broaden our data set into the blogs and other articles that were recommended in the message forums and see if broadening the community will change our results. It may include much more diverse group of communities outside of this particular discussion forum.

10. References

[1] Agile Alliance Functional Testing Tools Visioning Workshop, Oct 2007, Portland, Oregon

[2] Agile Alliance Functional Testing Tools Visioning Workshop, Agile 2008, Toronto, Canada

[3] Agile Alliance Functional Testing Tools Discussion Forum: http://tech.groups.yahoo.com/group/aa-ftt

[4] Wurowski, J., *The Wisdom of Crowd,* Random House, 2004

[5] Rittel, H., Webber, M. "Dilemmas in a General Theory of Planning", *Policy Sciences*, 4, 155-169, 1973

[6] Andrea, J., "Envisioning the Next Generation Testing Tools", *IEEE Software*, May 2007

[7] Fit, http://fit.c2.com

[8] Beck, K. Extreme Programming Explained: Embrace Change, 1/e. Addison-Wesley, Boston, MA, 1999

[9] Meszaros, G., *xUnit Test Patterns: Refactoring Test Code*, Addison-Wesley, 2007

[10] Fowler, M. "Specification by Example". Online: http://www.martinfowler.com/bliki/SpecificationByExample.html

[11] Denzin, N., Lincoln, Y., *Handbook of Qualitative Research, 2nd Ed.,* Sage Publications, 2000

[12] Glaser, B., Strauss, A., *The discovery of grounded theory: Strategies of qualitative research*. London: Wiedenfeld and Nicholson, 1967

[13] Strauss,A., *Qualitative Analysis for Social Scientists.* Cambridge, England: Cambridge University Press, 1987

[14] Sailer, K., Penn, A., The Performance of Space – Exploring Social Spatial Phenomena of Interaction Patterns in an Organization, *Architecture and Phenomenology Conference*, May 13-17, 2007, Haifa, Israel

[15] Abrams, S., Deflorio, P., More than Videoconferencing: Trials of a Sidebar Voice System for Distributed Studies, 2nd Concurrent Engineering Workshop for Space Applications, European Space Agency, ESTEC; Noordwijk, The Netherlands, Oct 19-20, 2006

[16] Brandes, U., Erlebach, T., *Network Analysis: Methodology Foundations,* LNCS 3418, Springer, 2005

[17] Dongen, S., Graph Analysis and Graph Clustering. In *Clustering by Flow Simulation*, Chapter 2, Wiskunde en Informatica Proefschriften, 2000, pp. 17-24

[18] Newman, M., Girvan, M., Finding and Evaluating Community Structure in Networks, *Physical Review, 69,* 026113, 2004

[19] Barabasi, A., *Linked: How Everything is Connected to Everything Else.* Perseus Publishing, Cambridge, MA, 2002

[20] JUNG, http://jung.sourceforge.net/doc/index.html