A Network Analysis of Stakeholders in Tool Visioning Process for Story Test Driven Development

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

Abstract

Participation from all stakeholders is important in a successful software development project, especially if the development project is complex and has many stakeholders. Identifying the key stakeholders is very difficult in a large community-based open source development project, because a lot of conflicting ideas exist in the community and not all of the necessary stakeholders are represented in the discussions. We analyzed the homogeneity of the stakeholders in the story-test driven development tool community and the diversity of the opinions represented by the stakeholders. We gathered opinions from the agile software engineering community on a list of desired features in a story testing tool. Then we categorize the community using a social network analysis to analyze the consensus building process. The network analysis reveals that the community has several key people with dominant degree centrality in the social network and the tool development community is remarkably homogeneous. Our research shows that a social network analysis is a good way to analyze the characteristics of consensus reached during a product visioning process.

1. Introduction

In Story Test Driven Development (STDD), requirements are communicated using tests. STDD is otherwise known as, executable acceptance test driven development, customer testing, specifications by example, example-driven development among many. STDD is an agile software development methodology of communicating requirements to the entire stakeholders using tests. The purpose of the story tests is to facilitate better communication among all stakeholders including customers, developers and testers. Instead of an ambiguous requirements specification using a natural language, STDD requires that requirements be written in a testable form that can either succeed or fail. STDD is becoming popular in the agile software engineering community and practitioners are trying to find tools that can facilitate STDD better.

Recently, the Agile Alliance organized workshops to envision what STDD testing tool should behave like [1, 2], because practitioners feel that the existing tools are inadequate for effectively facilitating STDD. Unlike Test-Driven Development that is based on unit testing, which primarily impacts developers, STDD 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 STDD 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.

Loosely associated groups of volunteers are pursuing the discussions over several face-to-face workshops and online discussion forums [3]. However, coming up with a good list of requirements for the future STDD tool is a very challenging task. We analyzed the discussions in the workshops and the online discussion forum on the next generation of STDD tool. Our analysis gathered about 300 features, issues, concerns and wish lists for the tool requirements. There are over 350 members who are following the discussions. From this huge list of features and stakeholders, we wanted to find out if there is a core set of concepts that are linking all of these opinions in these discussions and find out whether a social network analysis can provide a better insight into the online consensus building process. The analysis could provide a guideline as to which discussion topics are popular and which may be missing in the discussion.

The purpose of this paper is to understand whether there is consensus within the STDD tool building community. Due to a lot of discussions through online forums, there are many ideas flying around, but it is extremely difficult to get a big picture of what is being discussed. Identifying the key stakeholders is very

difficult in a large community-based open source development project, because a lot of conflicting ideas exist in the community and not all of the necessary stakeholders are represented in the discussions. In addition, many are opinions rather than a proposal for the tool requirements.

The motivation for our research is that these anecdotal evidences, opinions and their visions provided by the industry practitioners may provide interesting insights into a potential requirements gathering process. Our analysis could provide some ideas where the ad-hoc online discussions are heading as a community. The benefit of driving innovation through online forums is the Wisdom of Crowds [4]. The anecdotal evidence provided by expert groups is an alternative way to create insights. We can use a social network analysis to extract this meaning and validate it in part by determining how consensus is reached.

The research is qualitative and explorative in nature. Software engineering – particularly requirements engineering – is the problem of the domain. The problem that the requirements analysts must solve is often vague. Our research is important, because we look into the social aspect of deriving complex requirements, especially in a large open-source community based tool building project.

The paper is organized as follows. In Section 2, we describe the literature survey. In Section 3, we present the research methods. Section 4 presents our research hypothesis. In Section 5, we describe the research design. In Section 6, we present our results. We present the implication of our research in Section 7. We discuss the threats to validity of our research in Section 8 and conclude our findings in Section 9.

2. Literature Survey

2.1 Story Test Driven Development

Story Test Driven Development (STDD) or Executable Acceptance Test Driven Development (EATDD) is a way of communicating requirements through tests. In the Agile software engineering community, the concept is called by many names – customer tests [6], functional tests [6], story tests [7], example-driven development [8] and specifications by example [9] among many more. The terminology, Executable Acceptance Test Driven Development or just Acceptance Test Driven Development, seemed to be preferred in the agile research community [6, 8, 9].

Riding on the success of the test-driven development [10], many Agilists saw an opportunity to integrate entire stakeholders to participate in the test-driven development process. The idea behind STDD or EATDD is to write the requirements in a testable way to minimize miscommunication. The automation of these specifications into tests would ensure that the implementation is verified continuously. There are currently several tools to facilitate STDD/EATDD, but the most popular are Fit [11] and Fitnesse [12].

The concept of 'test' is different than the traditional sense of the word in Test Driven Development. In Test Driven Development, the tests are written by the developers to state beforehand what they are going to write in the code. In agile methods, software is built iteratively and the code may change many times. Therefore, the developers require a way to adapt to the high rate of code changes. The tests are a way of communicating meta-data about the code and alert the developers of unwanted changes in the behavior of the code due to later changes.

To loosely compare the differences between Test Driven Development and STDD, we can allude to white-box testing and black-box testing. If the unit tests in Test Driven Development ensure the design of the code – white-box testing, the STDD is black-box testing, because it specifies the behavior of software in terms of inputs and outputs. Therefore, some agile literature uses the word 'functional testing' to mean STDD.

Marick suggests that STDD tests are for exploration [8]. Therefore, he calls STDD as *Example-driven development* or *Business-facing tests* instead. The purpose of the tests is to create examples that will help all stakeholders understand the domain. Fowler also likes to call this process, 'Specification by Example' [9]. Fowler suggests that specification by examples mean highlighting only a few points and "you have to infer the generalizations yourself". Fowler suggests that the dominant idea with rigorous specification (and formal specifications) is that pre- and post- conditions must be explicitly stated in the requirements. However, Fowler found that pre-post conditions are very difficult to write in many situations. But asking for examples is much easier in some situations. Fowler stated that specification by examples is "less valuable in theory but more valuable in practice".

Many tools have been reportedly developed for STDD, but most of them are not available to the public. Some of the downloadable and freely available STDD tools include Fit [11], Fitnesse [12], GreenPepper [13], JAccept[14], TextTest [15], EasyAccept [16], AutAT [17], Robot Framework [18] and FitClipse [19]. People also claimed to have used Ruby, Selenium, Watir, Canoo WebTest and xUnit tests for acceptance testing.

3. Research Methods

We used a qualitative research method for analyzing the opinions presented by the participants in the discussion forum. Strauss and Corbin state that qualitative research is a "nonmathematical process of interpretation, carried out for the purpose of discovering concepts and relationships in raw data and then organizing these into a theoretical explanatory scheme". Qualitative findings can be done with three kinds of data collections: (1) open-ended interviews, (2) direct observation and (3) written documents. In this research, we are using written documents for our analysis. We used grounded theory and then organized our findings into a social network by people who reported the concepts.

3.1 Grounded Theory

One of the methods used for reduction of text to code is grounded theory [20]. 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 [20, 21] 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 have combined grounded theory and a network analysis before [22, 23]. 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.

3.2 Network Centrality

We applied network analysis on our coded data based on who reported the concepts [3]. We used Degree centrality and Betweenness centrality to obtain the network measures. *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 [24]. 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" [24], 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.

Clustering is a method of decomposing a set of entities into natural groups [24]. 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. 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 [25]. In this paper, we used the Betweenness Centrality metric for clustering analysis [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 [26]. 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.

4. Research Hypothesis

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 STDD tool building community currently is faced with a wicked problem, because they have many feature lists and many stakeholders but it is difficult to figure out what kind of consensus is reached through these discussions.

In an open source community, 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. Therefore, simply counting the frequency of topics does not provide a good indication of consensus reached by the community. We hypothesize that we can gain much better insight on issues using a social network analysis, because stakeholders with similar backgrounds could have similar wish lists. Additionally, people's wish list may be influenced by who they interact with more often.

Most network analyses are based on the Power Law [27]. The Power Law assumes that there is a strongly connected core in the network. It means there are several core concepts that connect most people. The other concepts are peripherals in the network. In our case, we suspect that the participants emphasize different issues based on what is more relevant to their current job. People with similar background and job functionality may think alike and group together more, because they tend to share similar experiences. Therefore, each of these groups may have a core idea that is different from other people.

Hypothesis: We hypothesize that there are multiple clusters of concepts, each with a core concept that is important to a particular group of stakeholders. People will naturally align themselves to these clusters of concepts by their job functions.

5. 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] and then meet once a year. 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 6. 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.

5.1 The Research Design for Degree Centrality

Our first 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 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. These 36 "experts" are chosen, because they frequently participate in the discussion. 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 [28]. 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.

5.2 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 [28]. 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.

6. Results

In this section, we present our results. We are going to first present our codes from the text analysis and then show the results for the degree centrality and the cluster analysis.

6.1 Coding Results

In this section, we are going to present 22 categories that were derived from the 300 features/issues 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:* How to involve all stakeholders into the STDD process including the developers, testers, project managers, business analysts and customers.
- *Adoption:* How to make people use the STDD tool
- *Test Maintenance:* How to maintain the story tests, especially in a large project with many tests.
- *Economic Value:* How to justify writing and maintaining tests, which can add up to a significant cost for a large project.
- *Regression Testing:* How to perform regression testing using STDD
- *Compatibility/Integration:* How to build an STDD tool that can work for as many IDEs, programming languages and various testing tools.
- *Usability:* How to make the STDD tool more usable by all stakeholders
- *Communication:* How to improve communication of the requirements between different types of stakeholders, especially when they do not hold the same kind of technical or domain knowledge
- *Business vs. Technology Problems:* Define what can be solved by the STDD tools and what should be solved by a better business analysis
- *Knowledge Representation:* How to represent the domain knowledge in tests
- *Notation/Language:* Research into a test specification language or a notation
- *Graphical Visualization:* How to represent the requirements in a visual way that is also testable
- *Architecture:* The tests should be able to test all parts of the architecture: data, model, user interface, etc.
- *Completeness:* How to know when the story tests are complete and how many story tests are required.
- *Distributed Tests:* How STDD tool should support distributed development teams
- *Different Perspectives/Skills:* The stakeholders have different skill sets and levels.
- *Exploratory vs. Test Automation:* How much should be automated
- *Workflow:* What is the STDD workflow?
- *Abstraction:* Be able to capture the knowledge using the tests at different knowledge abstraction levels.
- *Terminology:* What is the best terminology for STDD, because business customers do not understand STDD.
- *Reporting:* How to report the test results to the stakeholders.
- *Validation vs. Verification:* Story tests can be used for validation and verification. Which process should STDD support?

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 for each concept due to the limited spaces. The quotations in Table 1 represent opinions from the participants. Some are anecdotal evidences based on their experience or just a person's opinion on why he/she thinks the concept is important in the tool building process. We are not arguing for or against whether the opinions are correct.

Because this is a discussion forum with no restrictions on who participates, some topics had a very biased representation. For example, *Economic Value* was worded negatively only. They were suggesting the

difficulty of justifying STDD 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
Adoption	"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 Maintenance	"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
Communicat.	"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
Completeness	"I think that implying logical completeness is asking for trouble." #61
Distributed 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 Automation	"I do not think the skills [in TDD] are the same as traditional testing skills, nor the same as exploratory testing skills." #222
Workflow	"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
Abstraction	"This is all to do with the continuum between data, information, knowledge and potentially even wisdom." #198
Terminology	"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

6.2 Degree Centrality Analysis

The purpose of the degree centrality analysis is to discover how central a person is in the discussion if we form a network based on who proposes the similar concepts. Figure 1 shows the Degree Distribution Ranking graph that was graphed using JUNG [28]. As mentioned before, we wanted to find the people who are connected to the most people.

In the graph in Figure 2, the vertices are the "experts" and the edges are their interests. The bigger vertices mean two things: (1) they are connected to most people due to their vast breadth of interests; or (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 his/her ideas. As seen from the graph, this community is very tight with a lot of connections in the group. Due to the possible breach of privacy or possibly cause any discomfort by the people who participated in these discussions, we withheld their names. We identify these participants through numbers and by their initials.

However, because the information is available publicly, we don't feel that we had to anonymize their identity too much. The initials inside the brackets are the initials of their names.

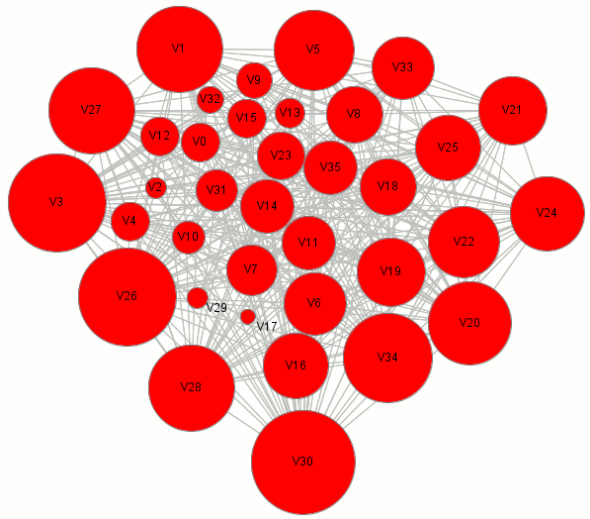

Figure 1: The Degree Distribution Ranking of the Participants

Highest Degree Centrality: As seen in Figure 1, there are about a half dozen people with a highest degree of centrality. They are V1 (A.B.), V3 (B.S.), V5 (B.M.), V20 (K.J.), V26 (N.J.), V27 (N.), V28 (P.L.), V30 (P.V.) and V34 (S.T.). These are top 25% of the population. Most people who are ranked at the top only participated in the discussions a few times and expressed a narrow set of interests in the discussion forum. For example, the following is the list of their interests for each of these participants. V28 (P.L.) only appeared in *Compatibility/Integration* category and V34 (S.T.) only appeared in *Test Automation.* V27 (N.) only appeared in two of the most highly discussed topics: *Compatibility/Integration* and *Different Perspectives/Skills*. Only V5 (B.M.) is unique from this list because he was the only person who had a vast breadth of interests and contributed frequently. For example, V5 (B.M.) appeared in 18 categories out of 22 categories. As a group, the people in this top tier of degree centrality measurement are interested in *Different Perspectives/Skills* and *Compatibility/Integration.* Some of these participants have already built STDD tools previously (A.B., B.M., P.L).

Middle of the Degree Centrality: The people who are ranked in the middle of the degree centrality are the *idea* leaders due to their frequent participation. They are V6 (E.H.), V8 (D.V.), V11 (E.P.), V14 (G.W.), V16 (J.M.), V18 (J.S.), V19 (J.A.), V21 (K.L.), V22 (L.C.), V24 (M.L.) and V25 (M.H.), V33 (W.C.) and V35 (M.S.). This group consists of about 40% of the population. 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.

Low Degree Centrality: 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. The rest of the population belongs in this category. Their ideas are shared by many people in the community or there is no strong conflict with their proposal so far. Most notably, V13 (G.M.) 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 on the test automation. V2 (A.M.) appeared in this category due to his vast breadth of interest in many topics. He has 14 interested categories. Unlike V5 (B.M.), V2 (A.M.) seemed to have many overlapping ideas with rest of the community. These people are likely to be in a good position to facilitate consensus in the community, because they don't have opposing influences in the network. However, there is no dominant concept in this ranking either, because there is no one who particularly stood out and championed for an idea.

Unlike our hypothesis, we find that degree ranking is influenced by one's number of interested topics in the discussions rather than by their job functionality. Based on the participation in these discussions, we find that the problem is getting the support from the middle-ranked participants who are in a deadlock due to their differing ideas. If the participant had a focused set of interests in the tool building discussion, they tend to rank higher. Some participants in the lower-ranked degree centrality (eg.V13) were also able to obtain a go-ahead from some people in the community although rare. It seems that consensus building is most likely to be led by tool builders at the top with a specific interest. However, the motivations and ideas tend to come from the people in the middle of the degree centrality ranking.

6.3 Cluster Graph Analysis

In this section, we are going to show what kind of consensus is reached in the social network analysis. 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. This is also the topic most discussed by the people in the high ranking degree centrality. 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.

Therefore, we performed cluster analysis and graphed them using JUNG [28]. Figure 2 shows a few snapshots of the graph transformations after applying iterations of the Edge-Betweenness algorithm. Figure 2 is only meant to show visually the *trend* of the transformation qualitatively. The results are found in Table 2.

The graph starts at 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. They are shown in the last graph (lower right graph in Figure 2). The three core clusters are shown in color cyan at the bottom.

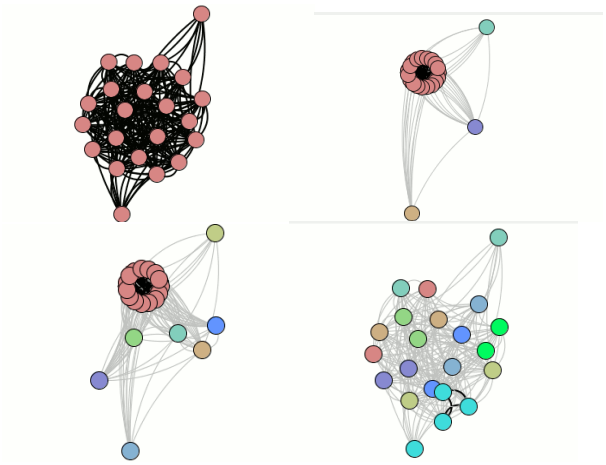

Figure 2: Four graphs showing how the graph was transformed after iterations of Edge Betweenness algorithm. The top left graph is the initial graph, and the bottom right graph is the final graph.

Notice that, however, only one cluster was formed. 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 like we hypothesized. It means the three final vertices are at the core of everyone's interest no matter what their background is. The community is not segregated into subgroups based on their job functionality.

As the vertices leave the graph, they do not form additional clusters or sub-clusters. Semantically, a lack of sub-clusters means the entire community is actually very homogeneous in terms of what they desire. Unlike our hypothesis where we assumed that people from different backgrounds will cluster around different concepts, the group shares the same visions.

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

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. An extremely homogeneous group means that the group should be able to come to consensus easily, but it also means the group lacks the diversity and no focus groups exist in this community to deal with sub-topics.

7. Implication

7.1 Degree Centrality Analysis

The purpose of the analysis is to find out the state of tool requirements discussions as they stand in the given time frame in a large open-source community. It can provide better insights into what type of people are joining the discussions, what kind of topics are being

discussed and provide some insights as to what or who may be missing in the discussions.

The purpose of the degree centrality analysis is to determine if there are people with more influence in the STDD tool visioning community and then find out what their message is. What was interesting is that the people who are the "idea leaders" appear in the middle of the degree centrality ranking. It is more important to have a focused interest in this kind of discussions to appear at the top degree centrality rather than participate in all discussions.

The people who appeared at the top ranking of the degree centrality built their tools or expressed only a focused interest in certain aspects of the tool. The result is suggesting that perhaps the best way to become the dominant participant in this kind of ad-hoc open source tool-visioning community is to build a tool and present to the community.

As our observation shows, being at the lowest tier of the social network also has an advantage. They seem to be the observers rather than debaters. As V13 has done, perhaps they are the people who are most prepared for the consensus building process, because their ideas are generally confronted least by the community.

The network analysis can show which ideas currently have champions. While participating in the discussions can generate many ideas as a community, the message becomes diluted by equally interesting stakeholders with conflicting ideas. The social network analysis can show which ideas have champions through the degree centrality analysis. In our result, the top tier groups clearly suggested that there were champions for Different Perspectives/Skills and Compatibility/Integration. As shown in Table 2, the lower extreme of the ranking shows that Knowledge is lacking champions. Our findings do *not* mean the requirements for the tool is identified, but rather the analysis identifies the gaps in the discussions because certain discussion topics are clearly left out of the community without any champions in the debate.

7.2 Cluster Analysis

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 Table2). Semantically, it means this is a very homogenous group and there is not enough diversity in the community. Or everyone in this community believes in the same thing. Despite a large number of participants, many stay silent. What our analysis is suggesting is that despite a large diversity of individual ideas, the community tends to steer the discussion into a common theme over time based which ideas get champions. The diversity of ideas seems to get silenced over time.

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. It means that online forums can attract like-minded people together to collaborate online. It also shows that a lot of ideas can be generated from these workshop/online discussions (over 300 ideas) and surprisingly everyone has a common goal (3 common concepts).

However, we didn't expect this community to be homogenous, which may mean outside viewpoints are not well represented in the community. The community doesn't have many domain knowledge experts, which explains why *knowledge* is represented least in the product visioning discussion (See Table 2).

The result from the research can be used to think about the tool requirements by analyzing what a selected group of industry practitioners suggested to be their idea of story testing tool would look like. Also, the research analyzes how effective online discussion forum is in identifying people's wishes. We were interested in the nature of the community and the kind of consensus that can be built by using open discussion forums. It wasn't surprising that there were a lot of conflicting ideas, but that the community is extremely homogeneous was surprising because we originally had about 300 codes to begin with and over 350 members. Our analysis shows that a social network analysis can show interesting insights into a very ad-hoc looking set of wish lists. Online requirements gathering process can work when you want to gather like-minded people together. However, initiating an action seems difficult in this kind of community, because most ideas are without identifiable champions. The findings can perhaps spur some interest in scientific research about the tool requirements for STDD.

8. Threats to Validity

In this section, we are going to discuss the validity of our findings in respect to internal and external validity. The study was performed on an online discussion forums organized by Agile Alliance. Therefore, there is a risk of single group threats, which applies when the result looks at a single group. More empirical studies are needed to generalize our result with other similar discussion forums. The research also looks into one type of *qualitative* analysis: written documents. The written documents may not express

the participant's desires very well because some people may not have participated fully in the discussions due to their busy lives. Therefore, we cannot generalize what people have written on the forum as their final words. In addition, the discussion forum tends to attract certain types of people - in this case, testers and developers. The self-selection may lead to a biased view of the software requirements.

We used our coding in consistent manner, but other researchers may derive different codes. As Strauss and Corbin suggest, qualitative analysis is an analysis of the interpretation, but a systematic one. Therefore, we will not generalize our findings beyond what the qualitative analysis can provide to us: insights. For conclusion validity, we have shown that our network analysis has shown interesting trends as shown by the graphs and tables. However, as this was not a quantitative experiment, we present our result only as an explorative insight into the current state of STDD tool visioning process in the community. A qualitative analysis is important, because it can provide a bigger picture for phenomena. As online collaboration grows, we may be able to make better use of the online community for gathering requirements and we propose that social network analysis may be one of the methods for analysis.

Basili et al. stated that drawing a conclusion from one empirical study in software engineering is very difficult, because any number of context variables could have influenced the result [38]. For this reason, we cannot assume that results from this forum can be generalized into an ideal STDD tool. One criticism of empirical studies is that the result may seem obvious after the fact, but this is a misguided belief as some important facts are discovered through the evidence collected.

9. 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 interview and survey the community to gather opinions from people who do not actively engage in the discussions and find out if their opinions are different.

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

[7] Kerievsky, J. Storytesting, http://industrialxp.org/storytesting.html

[8] Marick, B., Example-Driven Development, http://www.exampler.com

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

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

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

[12] Fitnesse, fitnesse.org

[13] Greenpepper, www.greenpeppersoftware.com

[14] Miller, R., Collins, C. Acceptance Testing. Proc. XPUniverse 2001, July, 2001

[15] TextTest, http://texttest.carmen.se/

[16] EasyAccept, http://easyaccept.sourceforge.net

[17] AutAT, http://boss.bekk.no/boss/autat/

[18] Robot Framework, http://code.google.com/p/robotframework/

[19] FitClipse, http://sourceforge.net/projects/fitclipse/

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

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

[22] 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

[23] 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

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

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

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

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

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