Assessing Individual Performance in Agile Undergraduate Software Engineering Teams

Rose F. Gamble
Tandy School of Computer Science
University of Tulsa
Tulsa, OK 74104
gamble@utulsa.edu

Matthew L. Hale
Tandy School of Computer Science
University of Tulsa
Tulsa, OK 74104
matt-hale@utulsa.edu

***Abstract*—The Agile Software Development (ASD) process is at the forefront of rapid product development driven by changing customer requirements and a trusted, self-organizing development team. Scrum has become a viable model of ASD focusing on determining immediate deliverables and structuring short timelines, called Sprints, for designing, implementing, and providing them for testing by the customer. While these practices are being adopted by organizations, there is significant difficulty in scaling them to the classroom. Once in place, it is a complex task to evaluate individual student performance based solely on the product outcome and Sprint grade. Thus, there is limited opportunity to catch performance problems that may lead to missing deliverable deadlines or decreasing team trust. In this paper, we impose ASD using Scrum on a senior software projects course in Computer Science. Using a collaborative environment that embeds a social network, project management modules, and event capture system, we perform broad data and event capture and analysis to investigate metrics that are relevant to assessing individual performance aspects related to functioning on an Agile team for software development. Our results suggest that predictive data is available after each Sprint to ascertain individual performance attributes and their relationship to product outcomes.**

Keywords—Agile Software Development; performance assessment; Scrum; undergraduate software engineering

I. INTRODUCTION

Many organizations have moved toward Agile Software Development (ASD) [1] for rapid deployment of software products, especially for web applications, web services, and cloud computing. A research survey conducted at Microsoft [2] found that around one-third of the respondents use ASD, with Scrum being the most used practice. Only recently has ASD and Scrum, specifically, been studied and reported on within a software engineering course setting. Many studies have analyzed the design of the curriculum covering the benefits and drawbacks of ASD and Scrum [3, 4]. For example, meeting frequency can be burdensome if it is required to be face-to-face every time. This observation parallels industry surveys where meeting frequency is viewed as a necessary drawback of Scrum [2]. Despite the drawbacks, the benefit of rapid delivery of product functionality from using ASD translates into student teams realizing non-trivial software products using three Scrum Sprints over a 16-week semester; something that is enormously advantageous to the graduating computer scientist.

When deploying Scrum in a course setting, product requirements and user stories are placed in an adaptive *Product Backlog*. Teams construct incremental, functional deliverables in the short timelines of the Sprints. A *Sprint Backlog* houses the tasks required to develop each deliverable designated for that Sprint. *Sprint Meetings* require team members to individually discuss their current progress on assigned tasks, their next task, and any problems they encountered or impediments they see in deliverable given their results and those of other team members.

While methods for integrating ASD into software engineering courses have solidified, assessing individual performance can be difficult because the Agile team's productivity and outcomes can mask individual contributions. Such assessment is needed to identify how an individual participated and to what extent. The assessment requires metrics that separate the individual's participation from the team results. Though research has been performed on individual assessment in a variety group projects, it often relies only on self-reporting by individuals on their efforts and their team members' efforts to produce a grading scheme. Virtual teams have been studied to determine if leadership or influence attributes can be assigned by studying social network interactions. Because we are studying ASD teams that are pre-formed in a pedagogical setting, we need specific metrics that track the Scrum process and its use by team members to deploy Sprint deliverables. The goal is to determine if predictive data exists after a Sprint that indicates individual performance so that individuals can be better trained to reach team goals. The study requires online tool support for capturing the discussion and logging product-related activities as they occur.

There is wide recognition in the tool support community [4-6] that specific tools must be made available, such as wikis or Google collaboration tools, forums or email, version control, and shared calendar applications to support project management, product development and interaction. However, substantial effort is needed to consolidate information captured from diverse tools into a single time series necessary for team or individual assessments.

In this paper, we define four performance metrics. These metrics are designed to characterize the depth and quality of individual performance as it relates to ASD team functions. *Contribution* measures the direct participation and quality of involvement of the student during Sprint Meetings. *Influence*

978-1-4673-5261-1/13/$31.00 ©2013 IEEE

measures if and how an individual directs the team's progress by being an engaged and active member of the team. *Impact* documents the causal relationships between what students say they will do, the actions the actually take, and the artifacts that result from their actions and, ideally, improve the Sprint deliverable's quality. Finally, *Impression* measures how well team members acknowledge the effort and performance that fellow team members make toward the successful completion of project deliverables.

We outline a methodology to assign values to each metric. The values form a numeric *assessment profile* of an individual student within an ASD team using Scrum in a university capstone software engineering course (see Table 1 for example). We rely on our open source courseware, called SEREBRO, to capture and consolidate team event data. Subject matter experts (SMEs) perform content inspection to calculate the metric values. We state research questions relevant to the study that direct how the assessment profile can be used to understand individual performance on an ASD team.

In the next section, we discuss various studies on ASD. Section III outlines the capabilities of SEREBRO and prior studies conducted with using its data capture functions. Section IV details the methods for measuring Contribution, Influence, Impact, and Impression, states the research questions, examines the data collected, and discusses the implications of the results. Section V concludes the paper.

II. Related Work

Pushing ASD into an undergraduate or graduate software engineering class requires careful set up and management so that the students are not overwhelmed by the pace and activity of the process, given what is involved in a Scrum Sprint. Researchers have drafted such courses and reported, often anecdotally, on the findings [7]. Studies conducted in undergraduate classes have examined the social interaction with respect to ASD for the purpose of determining if students have more effective learning opportunities. One study [8] points to pair programming in ASD as a potential learning opportunity for software development. A Social Interaction Model of Pair Programming resulted from data collected in two forms for assessment: interviews to gauge participation and satisfaction, and self-reported documentation by the students about their perspective on project outcomes. These results indicated not only a feeling of more productivity, but also an increased skillset confidence and interest in IT.

In contrast to focusing on process, studies examined human behavior based on automatically data collected by IBM's collaboration tool JAZZ [9]. A social network was constructed by finding indirect interactions through developer build commits and artifact versioning as entered in a time-stamped software repository. The resulting *socio-technical network* was used to predict software quality based on whether or not successful incremental builds showed a high degree of connectivity, and hence, collaboration among developers [10]. These results emphasize the importance of collaboration in software engineering teams, further substantiating the use of collaboration metrics as performance measures [11].

Other studies examine the perception of productivity and satisfaction of using ASD [2, 4, 12]. Assessment is generally performed via surveys and personal interviews. Many claim a positive response to the practice, including a perception of increased productivity and product quality [13]. Others found that the main values attributed to ASD relate to communication and collaboration [14]. Studies of using ASD in graduate software engineering courses concur; also citing increased student team productivity and product quality [7].

A review of literature on assessing individuals in group projects does not provide adequate, reusable metrics for ASD teams. Existing measures heavily rely on self-reporting activity, detailed project evaluation rubrics, surveys and grades based on individual submissions related to the project [15-17]. This lack of metrics is especially an issue for classroom settings where feedback and individual grades should be provided after each Sprint to improve student activity quality and accountability on the project, in general, and within the development process and on a team, more specifically.

III. SEREBRO

SEREBRO 3.0 is used as courseware for three classes at the University of Tulsa: Software Engineering Projects I and II and Introduction to Psychology. SEREBRO features an *idea network*, which combines aspects of issue management and social networking, a set of *project management modules*, and an *event capture system* that logs all activities within the idea network and modules.

Fig. 1 is a screen shot of a SEREBRO idea network as a graphical forum for asynchronous postings of brainstorms (blue circles), agree and disagree nodes (green and orange triangles), and comments (bubbles). Multiple brainstorms may be used within a single topic thread and may be started by any team member. SEREBRO provides optional directed email alerts for posting and project activities to ensure team visibility. The idea network is implemented in jQuery and displays post content when a user hovers over a node icon. Clicking an icon in the network marks it on the left and brings up the corresponding post to the right of the tree for user response.

SEREBRO's project management modules include a Gantt chart, calendar, wiki, document upload area, Subversion (SVN) software repository, activity feed, and a spreadsheet. The wiki is most often used to build and structure project documentation. SEREBRO *uploads* typically consist of images, presentations, and other non-textual files relating to the project. The *SVN module* provides source and version control repositories for each team as well as a WebSVN UI, which allows team members to make manual changes from a web browser, and a commit feed that displays commit messages and allows SVN commits to be tagged using the tagging system. The SEREBRO *spreadsheet system* provides Excel-like functionality in a web-browser to allow students to store and process numerical or textual data. The Gantt Chart is used to assign and track tasks throughout development milestones.

Previous studies have been performed using SEREBRO data captured in the CS and Psychology courses. These studies examined motivational techniques to increase creativity [18], including features to support teams [6],

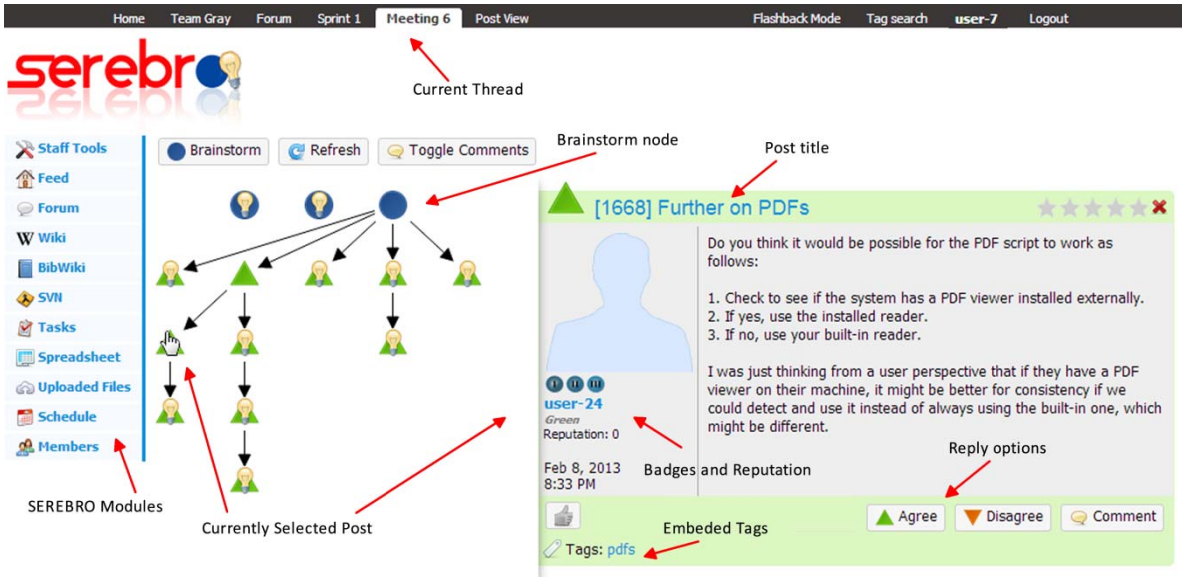

Fig 1. Sample SEREBRO Idea Network and Interface

developed event logging and automated scoring techniques for analyzing milestone performance within the Rational Unified Process [11], provided a tagging system for traceabilty across the artifacts [19], and proposed potential learning indicators evident in web-based collaboration [20]. Prior to Fall 2013, all Software Engineering Projects classes used a cross between the Waterfall development process and the Rational Unified Process to drive milestone formation and completion, which formed the basis for the prior studies. SEREBRO was tailored by class rubrics designed for each process milestone. Thus, posting in previous studies was very unstructured and conversational, informal, and sometimes completely off-topic. In addition, there was more freedom in how the various SEREBRO support modules were employed. The introduction of ASD, as discussed in this paper, required the teams to use the idea networks and project management modules in a much more structured way to facilitate Scrum practices such as attending Sprint Meetings, and maintaining the Product Backlog and the Sprint Backlog.

IV. Methodology and Assessment

In this study, we work within the ASD process using Scrum to the extent possible where there are frequent meetings with specified individual input requirements. The full set of SEREBRO modules is used to produce the various Scrum Sprint artifacts. The goal of the effort is twofold. First, we need to understand what is required to successfully scale Scrum to a class setting with one instructor and external customers, when students have additional classes and are preparing for graduation. Second, we want to get a clearer and more readily available picture of individual performance on an Agile team by obtaining predictive assessment metrics that identify how individuals perform during a Scrum Sprint.

Our study follows 16 students, across 4 teams in Software Engineering Projects II, working on non-trivial software products. The students were prepped on the use of SEREBRO, ASD, and Scrum as part of the previous fall semester software engineering immersion class (Software Engineering Projects I). During the fall course, students used SEREBRO for short project challenges to practice Sprints. For the Spring Sprints, the instructor serves as the Scrum Master and external customers serve as the product owners. Because pair programming in ASD has mixed reviews in both industry and academics [21], it was discussed as a potential strategy, but not required of the teams. We use the following artifacts generated within SEREBRO as part of a Scrum Sprint.

Meeting Check-ins: These are specific posts required in the idea network that are structured to simulate Sprint Meetings, though check-in times are generally 48-72 hours apart. In the studied Sprint, we analyze 10 meeting check ins that occurred over a 3 week period.

General Posts: These posts reside within the idea network and include face-to-face meeting minutes during customer meetings, general discussions on research, server configuration, technical problems, and conversations with the instructor. These posts have value as part of the general student engagement in the project.

Wiki: Generally, the wiki holds all documentation related work products. For the examined Sprint, the Wiki was required to hold an executive summary of the product, use cases, research on technology and competition, execution instructions, and a Sprint review or post mortem.

Subversion Repository (SVN): The shared repository contains code commits and associated commit messages for version control during product development.

Gantt chart: A project management and task assignment tool houses the Sprint Backlog that is updated as tasks are assigned and completed during the Sprint or issues and impediments identified by the team are added and assigned to responsible parties.

Spreadsheet: The spreadsheet module in SEREBRO contains the Product Backlog that includes the product requirements and user stories.

Uploaded documents: SEREBRO provides a shared space for storing non-textual documents such as images and presentation files that supplement other module content.

Team Evaluations: After each Sprint, team members submit a confidential evaluation sheet on perceptions of their performance and their team members' performances during the Sprint. The evaluation consists of two parts. The first part is a questionnaire with 10 general evaluation questions in which every person evaluates his or her personal performance and the performance of each team member using a range of 0-10, with 10 being the highest. The second part of the form denotes a variety of roles and responsibilities, such as general manager and organizer, leader and inspirer, programmer, designer, analyst, and tester. In this part, each person denotes a percentage weight for a role or responsibility for each team member and then scores that performance from 0-10 at that role or responsibility.

Team Rankings: Each student ranks the amount of perceived engagement in intellectual input (product vision, understanding, and interpreting customer requirements), creative input (elegance in design, coding, aesthetics), and results achieved (tangible evidence of work and experience) for each team member. These rankings supplement the more detailed Team Evaluations with direct comparison.

Sprint Deliverable Grade: The deliverable for the Sprint includes a stand-alone functional component within a web application or integrated components that meet one or more essential requirements and encompass at least one user story. The deliverable is accompanied by project management artifacts that include the Product Backlog of requirements and user stories, the Sprint Backlog of tasks, impediments, and issues related to the Sprint, wiki documentation associated with the deliverable, the SVN repository, and the Sprint presentation and demonstration. A project grade is given to the team according to the quality of the deliverable overall. The Sprint Deliverable was worth a total of 800 points for which a complete set of rubrics was provided to the class.

A. Contribution Scores

Contribution assesses the direct participation and quality of involvement of the student in required Sprint Meetings, in the form of Meeting Check-ins. There were four separate check-in requirements: (a) being on time, (b) stating what was done since the last meeting, even if it was little to no activity, (c) stating what will be done for the next meeting, and (d) discussing a project issue, impediment, or activity found by the individual or a team member. Each requirement was worth 3 points with the Subject Matter Expert (SME) scoring as follows: 0 = not met, 1 = almost met, 2 = met, and 3 = exceeded requirement. The Total Meeting Points is the sum of the requirement points from 0-12 per individual for each meeting across all 10 meetings. The data in Table 1 consists of the users (column 1), the Sprint Deliverable Grade (column 2), the Total Meeting Points (column 3), and the final Contribution score (column 4) which is defined as the Total Meeting Points divided by the 120 maximum possible points. Contribution scores ranged from 31% to 90%, with a 54% average. Values for Impact, Impression, and Influence (columns 5, 6, and 7, respectively) in Table 1 and discussed in subsequent sections, comprise the numeric *assessment profile* for each student.

The first research question accounts for ASD and Scrum expectations, that short, frequent meetings where individuals are self-directed, accountable, and trusted produce better product outcomes. It assesses whether the amount and quality of meeting participation leads to a better deliverable.

RQ1: Does a team with high overall Contribution have better product outcomes?

To answer this question, we examine the linear dependence between the Contribution score and Sprint Deliverable Grade in Table 1 using the Pearson correlation coefficient (Pearson's r) [22] to determine if higher individual Contribution scores correlate with higher Sprint Deliverable Grade. Correlation values for this and subsequent research questions are calculated using sum of squares, as part of linear regressions between series of data for two metrics. All r values are calculated with 14 degrees of freedom (df) corresponding to the 16 users present on the 4 examined teams, i.e. df = 16-2 for two-tailed test. In this and the following sections, we use the standard critical values for r as follows:

- $|r| > 0.426$ corresponds to a significance level, denoted by a p value, of $p = 0.10$ (shown in correlation tables as purple),
- $|r| > 0.497$ represents $p = 0.05$ (red),
- $|r| > 0.574$ represents a high significance of $p = 0.02$ (yellow), and
- $|r| > 0.623$ is indicative of a very strong significance $p = 0.01$ (green).
- Any r values such that $|r| < 0.426$ are considered to be not indicative of a correlation (white).

TABLE 1. GRADE AND ASSESSMENT PROFILE

Student	Sprint Deliverable Grade	Total Meeting Points	Contribution	Impact	Impression	Influence
user-6	0.88	49	0.41	0.76	8.81	0.19
user-9	0.88	59	0.49	1.35	8.60	0.18
user-11	0.88	53	0.44	1.47	9.27	0.23
user-23	0.88	64	0.53	3.12	9.25	0.39
user-13	0.86	69	0.58	2.47	9.54	0.29
user-18	0.86	64	0.53	1.53	9.27	0.21
user-19	0.86	74	0.62	2.47	9.19	0.21
user-22	0.86	70	0.58	1.18	9.21	0.23
user-2	0.98	49	0.41	1.41	8.82	0.06
user-17	0.98	95	0.79	4.71	9.56	0.29
user-21	0.98	108	0.9	5.00	9.28	0.49
user-24	0.98	61	0.51	0.53	7.74	0.14
user-7	0.82	37	0.31	0.24	7.46	0.09
user-8	0.82	71	0.59	1.59	9.43	0.37
user-10	0.82	64	0.53	0.35	9.24	0.23
user-16	0.82	44	0.37	0.35	8.48	0.15

The p value is formally described as the probability of obtaining the observed result given that the null hypothesis is true. Our work uses a null hypothesis which can be stated as *there is no relation between metric X and metric Y*. Thus, a p-value of 0.01 would mean there is a 1% probability that one

could observe a correlation given there is actually no linear dependence between the two metrics. In other words, the lower the p-value, the higher the likelihood that the correlation result is significant of an actual dependency relationship between X and Y.

For RQ1, we found that higher Contribution scores correlated with higher Sprint Deliverable Grades (r=0.47, p=0.10). This suggests that, with 90% confidence, we can say that better individual contributions across the team lead to better final products. All correlations discussed henceforth will report the r and p values.

B. *Influence Scores*

Influence measures if and how an individual directs the team's progress by being an engaged and active member of the team. This activity may be in the form of intellectual and creative input or vision, results production, general project communication, and managing the team. We measure influence using the SEREBRO event system to obtain a raw count of the total number of posts, activities affecting artifacts, and the average events per day for an individual. Since each team has unique project requirements, energy, commitment, and skill sets, we weight the Influence score against the events of the team as a whole (see below). Given an individual's calculated Influence score, the second research question examines if an engaged and active team member, i.e. one with a high Influence score, drives the product progression. The third research question determines if an engaged and active team member is recognized as such by the team. The importance of these questions to the study is to understand how engagement and activity can and should be manifested so that an individual's participation is recognized as valuable.

$$\textbf{Influence score} = \frac{average(individual\ events\ per\ day)}{average(total\ team\ events\ per\ day)}$$

RQ2: Does an individual's Influence determine how much that person impacts the overall product?

For this question, we return to the students' numeric assessment profiles in Table 1. Influence and Impact (defined next) are highly correlated, (r=0.74, p=0.01) indicating there is a very strong relationship between the number of events an individual performs relative to the team and his/her impact on the final product.

RQ3: Does an individual's Influence score match the team's overall perception of the person's activities?

To answer RQ3, we use the Team Rankings to determine if a seemingly engaged and active person is ranked highly in intellectual input, creative input, and results achieved as perceived by their peers. Table 2 shows the results. Note that the negative numbers are inversely correlated because the higher the influence (i.e. the more events) the lower the rank should be, since students are ranked from best (1) to worst (4). For intellectual input, (Intellect with r=-0.58, p=0.02), the strength of the correlation indicates that highly influential people are recognized as providing vision and quality ideas to the development process and resulting product. This result implies that individuals who communicate more frequently in the idea network or create more artifacts (i.e, code, wiki documentation, spreadsheets, or uploads), are perceived to have provided intellect through that effort. Similarly, the results achieved (Results) strongly correlates to Influence (r= -0.65, p=0.02), indicating that the team acknowledges that engaged and active people produce actual work products. This correlation is expected because high result achievers will likely perform more artifact manipulation, increasing their Influence score. Creative input did not correlate with Influence. We infer from this result that either creativity is not well understood by the team members, or there is no perceived association between a person's creative input into a project and the activities performed. Though not shown, creative input also did not correlate with intellectual input (r = 0.42) indicating that individuals did not rank their teams using the same criteria for intellectual and creative rankings.

TABLE 2. INFLUENCE CORRELATED WITH INTELLECTUAL, CREATIVE, AND RESULT RANKINGS

	Intellect	Creative	Results
Influence	-0.58	-0.01	-0.65

C. *Impact Scores*

Impact documents the causal relationships between posts and activities that subsequently alter artifacts. Impact scores are based on three forms of self-regulated learning strategies [23]. An individual posting that he or she will perform a task is a form of planning. Following that statement with an action constitutes a preplanned activity. If a user performs an action and then posts, any request for evaluation is a third form of self-regulated learning.

To measure an individual's project Impact, we link communication posts with artifact creation activities. We automatically filter posts according to phrases and words indicative of the self-regulated learning strategies, such as "will," "done," "I'll," "I can," "I have," "doing," "I did," and "I've." A manual review of the filtered posts is performed to remove any spurious posts. The remaining posts contain one or more *goal statements* that relate to the creation or advancement of the Sprint Deliverable. The SME compares the posts against the set of product deliverables to determine if the goal statements relate to actionable events in SEREBRO. An actionable event is defined as the creation or modification of an artifact within a SEREBRO module. Actionable events include SVN code commits, file uploads, updating the Product or Sprint Backlogs, and documentation wiki edits. The SME records the number of *links* that a goal statement post has to an actionable event and assigns a quality value to the event's effect on the deliverable. These values are

- 0 = Poor quality work – event had minimal effect on deliverable completion.
- 1 = Low quality work – event had little effect on deliverable completion.
- 2 = Good quality work – event had direct effect on deliverable completion.
- 3 = High quality work – event had significant effect on deliverable completion.

Some example quality assessments include:

- poor quality code commit (uncommented, no commit messages, few lines or single characters)

- high quality code commit (well commented, commit messages, significant number of lines added or changed)
- poor quality use case diagram uploaded file (lots of errors, carelessness is evident)
- high quality use case diagram (no or low errors, thoughtful, complete)
- poor quality wiki change (spelling errors, broken links, incomplete, and non-descriptive, with no change messages provided)
- high quality wiki change (no errors, no broken links, complete and descriptive, change messages provided)

We calculate a single Impact score using the formula:

Impact score =

$$\frac{\text{Scored Posts}}{\text{Total Scored Posts for Team}} * \text{Average\#Links} * \text{Average Quality Assigned}$$

where Average#Links is the average number of links the individual has across all filtered and scored posts, Average Quality Assigned is the average assessment value of all of the links found, and (Scored Posts / Total Scored Posts for Team) normalizes the Impact score given a particular team.

RQ2 shows that an individual's Impact is related to his or her engagement and activity (i.e., Influence). We introduce a fourth research question to determine if an individual's Impact is indicative of overall product outcome. This question studies whether an individual's planning of an activity, followed by its performance leads to a better overall deliverable.

RQ4: Do individuals with high Impact Scores have better product outcomes?

We answer this by determining if Impact scores correlate with higher grades given the student assessment profile in Table 1, which shows a strong relationship (r=0.54, p=0.05).

D. Impression Scores

Impression examines how a team member acknowledges the effort and performance that another team member has made toward completing a successful Sprint Deliverable. A poor evaluation by the whole team can cost a member a letter grade for the Sprint Deliverable, so they are taken very seriously. How individuals view and value their team members should qualify the extent of the high trust environment on an ASD team. If an individual is a 'social loafer', then the team may continue the project without input from him or her, and provide low evaluation scores. In contrast, if everyone's impression of a person's role and responsibilities and the effort put into them is similar, then it can be assumed that the person can be trusted at some level of proficiency, competency, and work ethic. Thus, individuals that are actively involved with the team with tangible effort toward a successful Sprint Deliverable, as measured by Contribution, Impact, and Influence scores, should be highly valued.

Using the Team Evaluations, we form Impression score as:

Impression score =

$$\frac{\text{Avg(G.Q1…G.Q10)} + \text{Avg(R.Q1…R.Q10)}}{20} \_ \frac{\text{G.Avg+R.Avg}}{2}$$

where G.Q1…G.Q10 are the 10 general survey questions taken from the Team Evaluations and R.Q1…R.Q10 are the 10 weighted role-based survey questions. The metric G.Avg is the average across the 10 general questions, while R.Avg is the average across the 10 role-based questions. The next research question (RQ5) asks if team member impressions are indicative of overall performance.

RQ5: Do group impressions of team members accurately reflect the team member's performance towards completion of a Sprint Deliverable?

To answer this question, we examined group impressions at the aggregate level using G.Avg, R.Avg, and the combined Impression score, and also at a finer granularity by examining individual question responses of the Team Evaluations. These values are correlated against Impact, Contribution, and Influence. We correlated these values with measures of specific activities. We compare the Team Rankings on intellectual input, creative input, and results achieved and Impression against the other metrics to determine if ranking mirrors the Impression score.

TABLE 3. CORRELATIONS BETWEEN IMPRESSION METRICS AND IMPACT, CONTRIBUTION, INFLUENCE, AND PERFORMANCE METRICS

	Impact	Contrib.	Influence	Wiki	Upload	SB	Commit	Artifacts	Thread	Ideas
G.Q5	0.16	0.25	0.54	0.67	0.72	0.62	0.10	0.59	0.32	0.44
G.Q9	0.48	0.35	0.23	-0.13	0.06	-0.12	0.28	0.29	0.03	0.16
G.Avg	0.54	0.60	0.67	0.14	0.29	0.17	0.57	0.63	0.61	0.57
R.Q1	-0.14	0.04	0.31	0.57	0.66	0.70	-0.31	0.29	0.17	0.27
R.Q7	-0.25	-0.17	0.01	-0.09	-0.04	-0.02	0.03	0.14	-0.16	-0.10
R.Q8	0.54	0.46	0.45	-0.20	-0.09	-0.28	0.69	0.32	0.49	0.50
R.Avg	0.59	0.54	0.51	0.20	0.33	0.22	0.40	0.60	0.27	0.36
Impress.	0.59	0.60	0.64	0.17	0.32	0.19	0.53	0.65	0.50	0.51
Intellect	-0.50	-0.46	-0.58	0.17	-0.07	0.13	-0.67	-0.48	-0.60	-0.55
Creative	-0.10	0.01	-0.01	0.34	0.14	0.05	-0.08	-0.08	0.02	0.06
Results	-0.52	-0.61	-0.65	0.05	-0.08	0.00	-0.70	-0.61	-0.56	-0.57

Table 3 shows these correlations along with a sampling of five individual questions (G.Q5, G.Q9, R.Q1, R.Q7, and R.Q8) taken from the Team Evaluations. In G.Q5, individuals are asked to score how well a team member "Contributed to document artifact creation and/or review." We found that individuals with high G.Q5 evaluation scores had higher percentages of Wiki changes, Uploads, and Sprint Backlog (SB) updates relative to their peers as well as higher overall numbers of ideas, artifacts and thus higher Influence scores. R.Q1 asks team members to weight the specific responsibility of "Organizing the requirements, user stories, and Sprint Backlog" and score that effort. In SEREBRO, these activities are recorded in the wiki, the Sprint Backlog, and uploaded document files, so it is not surprising that there are correlations to each fine-grained activity. Another question R.Q8 grades team members responsible for "Demonstrating exceptional programming ability." Table 3 shows that highly rated R.Q8 individuals also had higher Impact, Contribution, Influence and Code Commits relative to lower rated peers. On the other hand G.Q9 asked if, in general, a user "Completed all tasks assigned at agreed upon timeline" and R.Q7 which grades those users responsible for "Delegating tasks appropriately" both correlated very weakly across the board suggesting that either the team members did not have significant information to form accurate impressions about these topics or that the performance metrics in the table were not representative of the team evaluation questions.

Overall, the correlations shown in Table 3 suggest that team member impressions reflect performance reality during the use of ASD and Scrum within a class setting. The communication structure and notification system provided by SEREBRO appears to aid the ability of team members to accurately assess their teammates by providing better visibility and transparency across the board. That is to say, such a tool makes it easier for team members to detect the extent to which their teammates are performing their assigned tasks.

V. Discussion and Conclusion

In this paper, we define and study four metrics to quantify and qualify various individual attributes with an ASD team. The metrics, Contribution, Influence, Impact, and Impression, provide broad characteristics of the level of engagement, activity, and product related results of an individual on a team. We find that teams with higher individual levels of contribution and impact had better final product outcomes. In addition, team members are able to form accurate appraisals of each other's contributions, influence, and impact on the project given high visibility of the development process. The study provides a foundation for direct and objective feedback to individuals regarding more detailed qualities regarding performance after each Sprint.

Our results suggest that using collaborative environments to obtain performance and collaboration metrics allows for extensive individual evaluation and assessment. They also suggest that that high levels of transparency, as expected within the ASD process, contributes to a high trust environment, since individual efforts are visible to the team in tangible ways. This visibility relates directly to project outcome. The challenge is to craft a supportive environment for ASD process facilitation and collaboration for a class. While these environments are becoming more readily available, such as IBM's Jazz, automated methods for detecting impact links, analyzing conversations and code commits, and assessing collaborative activity are lacking. Without environmental support, it is difficult to attach an objective value to an individual's detailed project performance. This dilemma is more notable at the university level when training in software engineering should result in a grade that shows where improvements can be made. Unless event and activity tracking can be translated into the performance metrics discussed, Impression must be heavily relied on for performance evaluation.

References

[1] A. Cockburn, *Agile Software Development*. Reading, Massachusetts: Addison Wesley Longman, 2001.

[2] A. Begel, and N. Nagappan, "Usage and Perceptions of Agile Software Development in an Industrial Context: An Exploratory Study," in *First International Symposium on Empirical Software Engineering and Measurement*, 2007.

[3] D. Knudson, and A. Radermacher, "Updating CS Capstone Projects to Incorporate New Agile Methodologies used in Industry," in *24th IEEE Conference on Software Engineering Education and Training*, 2011.

[4] F. Meawad, "The Virtual Agile Enterprise: Making the Most of a Software Engineering Course," in *24th IEEE Conference on Software Engineering Education and Training*, 2011.

[5] B. Bruegge, M. Reiss, and J. Schiller, "Agile Principles in Academic Education: A Case Study," in *Sixth International Conference on Information Technology: New Generations*, 2009.

[6] N. M. Jorgenson, M. Hale, and R. Gamble, "SEREBRO: Facilitating Student Project Team Collaboration," in *International Conference on Software Engineering,* 2011.

[7] D. F. Rico, and H. H. Sayani, "Use of Agile Methods in Software Engineering Education," in *Agile*, 2009.

[8] K. M. Slaten, M. Droujkova, S. B. Berenson, L. Williams, and L. Layman, "Undergraduate Student Perceptions of Pair Programming and Agile Software Methodologies: Verifying a Model of Social Interaction," in *Proc. of the Agile Development Conference*, 2005.

[9] A. Schroter, "Predicting Build Outcome with Developer Interaction in Jazz," in *International Conference on Software Engineering*, 2010.

[10] M. Cataldo, P.A. Wagstrom, J.D. Herbsleb, and K.M. Carley, "Identification of Coordination Requirements: Implications for the Design of Collaboration and Awareness Tools," in *20th Conference on Computer Supported Cooperative Work*, 2006

[11] M. Hale, N. Jorgenson, and R. Gamble, "Predicting Individual Performance in Student Project Teams," in *Proceedings of the 24th Conference on Software Engineering Education and Training*, 2011.

[12] G. Melnik, and F. Maurer, "Introducing Agile Methods: Three Years of Experience," in *Proc. of the 30th EUROMICRO Conference*, 2004.

[13] S. Overhage, and S. Schlauderer, "Investigating the Long-Term Acceptance of Agile Methodologies: An Empirical Study of Developer Perceptions in Scrum Projects," in *45th Hawaii International Conference on System Sciences*, 2012.

[14] C. Patel, M. Lycett, R. Macredie, and S. Cesare, "Perceptions of Agility and Collaboration in Software Development Practice," in *Proceedings of the 39th Hawaii International Conference on System Sciences*, 2006.

[15] W. Cooley, "Individual Student Assessment in Team-Based Capstone Design Projects," in *34th Frontiers in Education Conf.*, 2004.

[16] J. Hayes, T. Lethbridge, and D. Port, "Evaluating Individual Contribution Toward Group Software Engineering Projects," in *25th Int'l Conf. on Software Engineering*, 2003.

[17] K. Swigger, et al., "A Comparison of Team Performance Measures for Global Software Development Student Teams," in *4th Int'l Conf. on Global Software Engineering*, 2009.

[18] D. F. Grove, N. Jorgenson, R. Gamble, S. Sen, and B. Brummel, "Adapting rewards to encourage creativity," in *M.A.S. for Education & Interactive Entertainment: Design, Use, & Experience*, 2010.

[19] M. Hale, N. Jorgenson, and R. Gamble, "Analyzing the Role of Tags as Lightweight Traceability Links," in *Proc. 6th International Workshop on Traceability in Emerging Forms of Software Engineering* 2011.

[20] M. Hale, R. F. Gamble, K. S. Wilson, and A. Narayan, "Collaborative Learning in Software Engineering Teams," in *17th Americas Conference on Information Systems*, 2011.

[21] T. Dingsøyr, T. Dybå, and P. Abrahamsson, "A Preliminary Roadmap for Empirical Research on Agile Software Development," *Agile*, 2008.

[22] S. Stigler, "Francis Galton's Account of the Invention of Correlation," *Statistical Science 4 (2): 73–79,* 1989.

[23] P. H. Winne, "Improving measurements of self-regulated learning," *Educational Psychology,* vol. 45, pp. 267-276, 2010.