Toward Increasing Collaboration Awareness in Software Engineering Teams

Shuddha Chowdhury
Tandy School of Computer Science
University of Tulsa
Tulsa, OK USA
shc422@utulsa.edu

Charles Walter
Tandy School of Computer Science
University of Tulsa
Tulsa, OK USA
charlie-walter@utulsa.edu

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

***Abstract*— This Research Full Paper investigates collaborative personality traits in undergraduate software engineering teams. Online tools, such as Slack.com, provide team engagement and project management. While metrics can be defined for team and individual performance, it is difficult to measure collaboration and its impacts. Specifically, the forms of collaboration that lead to a successful software product should have associated metrics that correlate with individual performance, peer assessments, and project outcomes. Given the difficulty of assembling teams that best exemplify collaborative personality traits, it may be more beneficial for team members to recognize these traits so that their positive aspects can be exploited toward a successful product outcome. We employ IBM Watson™ Personality Insights service to analyze team Slack.com posts collected from forty students across ten teams and two semesters of the capstone class. We correlate thirteen traits recognized for influencing collaboration with team grades, Sprint meeting check-in quality, and peer evaluations. The correlations show that each trait is related to at least one performance metric during at least one of the Sprints. We discuss the potential meaning of different traits emerging at different times during the semester and how that can inform strategies for software developer collaboration awareness and reward.**

Keywords—Collaboration, software engineering, capstone projects, personality traits

I. Introduction

Within the undergraduate computer science curriculum, there is often a team-based project to create a non-trivial software product. Learning objectives of the course may include project management, communicating with an external customer, producing periodic deliverables, performing modeling and analysis, and creating sufficient documentation. To complete the required work projects to meet the learning objectives, the project must be complex enough to require significant collaborative effort, which is unlikely to have been experienced by the students in prior coursework. Thus, team members should exhibit collaborative behaviors as well as recognize others' behaviors as having a positive effect on collaborative efforts. From a psychology perspective, the Big Five model [1, 2] defines categories of personalities that can be broken down into more detailed traits, some of which have been strongly characterized as improving the collaborative potential of a team [3, 4, 5]. Many of these traits have emerged from studies involving software development teams [6, 7, 8, 9].

The IBM Watson™ Personality Insights service uses a linguistic analyzer to derive personality traits from text-based conversations [10]. It returns the probability that a trait exists for a total of 52 personality traits of the Big Five model's main attributes. The Personality Insights service infers personality characteristics from textual information based on an open-vocabulary approach. This method reflects the latest trend in the research about personality inference [11, 12, 13].

For Agile software development, team members with broad skillsets are common because a particular skill can become unused as the product changes, leading to team members without development (or collaborative) roles [14]. Whitehead et.al [15] mentions that a general understanding of software engineering collaboration is still imperfect, requiring improvement in many areas. One of the fundamental obstacles noted is the lack of established metrics to quantitatively measure collaboration efforts in software teams.

In this paper, we examine how personality traits affect collaboration within 10 software engineering teams that together comprise 40 students across 2 semesters of the capstone course. We analyze the Slack.com posts of the teams across 3 sprints using the IBM Watson™ Personality Insights service to determine the strength of 13 personality traits that are found in the literature to be important for improving collaboration. The results are correlated against sprint grades, meeting check-in participation, and an assessment of each member. Certain collaborative personality traits emerge as more positive indicators of team performance at different points during the semester. The identification of such traits can lead to training throughout the semester to increase collaboration during Agile software development in the classroom and in industry. We conclude with a discussion of potential mechanisms for early identification of collaborative personality traits and for increasing and rewarding collaborative activity.

II. Related Work

Previous researchers have explored how to best create successful teams. Random creation can lead to teams full of underperforming students with only one high performing

978-1-5386-1174-6/18/$31.00 ©2018 IEEE

student doing much or all of the work [16]. Assigning teams by separating students into groups and creating teams by mixing the groups together may result in personality conflicts between team members that they are not trained to resolve [17]. Allowing self-creating of teams can result in some students not being chosen, which can be demoralizing [18]. In addition, companies have experimented with evaluating software developer personalities to create a good team balance that manifests different, creative ideas [19, 20, 21].

A. Using the IBM Watson™ Personality Insights Service

The Big Five personality model is highly respected model on which significant analyses rely [1, 2]. John and Srivastava [22] define the model as part of their taxonomy analysis. The high-level personality traits are agreeableness, conscientiousness, extraversion, emotional range, and openness (see Table 1). Agreeableness reflects seeing the good qualities of other people, lending things to people they know, and providing consolation to a person in need. Conscientiousness is represented by impulse control that facilitates task and goal directed behavior and can be exemplified by arriving early for appointments and double-checking work before submission. Extraverted people have an energetic response towards the social and material world. Emotional Range contrasts emotional stability and even-temperedness with negative emotionality such as feeling anxious, nervous, sad, or tense. Openness describes the breadth, depth, originality, and complexity of an individual's mental and experimental life, such as learning something for the joy of learning. Each of the personality traits in the Big Five model can be divided into other traits. In the next section we provide a survey of literature that points to personality traits within the Big Five model that are related to collaboration.

In this paper, we use IBM Watson™ Personality Insights service as a tool to measure the strength of targeted collaboration traits in that fall within the Big Five model given Slack posts. The Personality service assesses conversations to determine the strength of 52 personality traits [10], of which we only examine 13 traits. Using this tool is our preferred approach for three reasons. The first reason is that several studies have been conducted to validate the Personality Service's results [24, 25]. The second is that it is automated, allowing us to use the Slack posts required by the course. The automation eliminates the need to have the students take personality tests.

B. Collaborative Personality Traits

From long-term observations of software development teams, McCuller [6] links adventurousness to productive software engineers as it relates to bringing new ideas to the project, experimenting with new tools and techniques, and collaborating with team members. Cautiousness was also believed to foster collaboration, because such a developer would be more likely to consider software reliability, security, and long-term maintenance as important to early product implementation. Bani-Salameh [26] analyzed programming activity for a simple programming exercise within a low-level, undergraduate class. The task required collaborative editing and debugging of code. Cautiousness, along with cooperation, reappeared as a collaborative trait that was an asset to team members who use their concern for project activities to assist the overall cohesion of the team.

Cooperation has been shown to play a significant role in collaboration project success. Cooperation results in team members not only exchanging product knowledge, but also knowledge about the business values and environment. Data collected from collaborative project management tools, self-reporting, and behavior observations was used in a study of six software development teams in a capstone course showed the positive effect of cooperation in creating, sharing, and integrating information [7]. Faraj and Sproull [27] examined the role of cooperation through a cross-sectional investigation of sixty-nine software development teams. They found that successful teams manage individual skillsets and knowledge interdependencies effectively, which can be performed through expertise and cooperation.

Cooperation has been coupled with trust as necessary for distributed and agile software development team performance [4]. For distributed teams, maintaining product repositories fosters cooperative attitudes, leading to improved collaboration. Trust allows for the knowledge exchange to be accepted by distributed team members. Weimer, et al. [28] suggest that trust is a critical factor for high quality collaborative software engineering teamwork. They extend a six-factor teamwork quality model that showed evidence of the relationship between teamwork quality and software quality [29]. They tested their model, incorporating trust, based on the data collected from 252 team members and stakeholders, showing that a high trust environment increases collaboration and improves team performance.

A study conducted by using sensor enabled badges indicated how communication affects team dynamics and performance [3]. Activity level was measured by the number and nature of exchanges among team members. It was noted that the activity level in a collaborative team is not static and may be based on changes to project requirements, team meeting times, and administrative details. The conducted studies showed that for projects demanding highly collaborative tasks, the most efficient work was performed by teams with people exhibiting high energy and engagement.

Chow and Cao [9] conducted a survey among Agile software developers, gathering data from 109 Agile projects from 25 countries across the world, identifying four prominent personality traits seen in other research: altruism, self-discipline, achievement striving and dutifulness. When developers helped each other, the Agile project had a higher chance of success. Self-disciplined developers attended daily meetings with a strong focus, took the appropriate technical training when required, and delivered the most important features early in the process. Successful teams were also comprised of achievement striving and dutiful members. They followed the Agile process as dictated by their company with regular software delivery and proper documentation.

Self-discipline and achievement striving have also been confirmed to enable collaboration, along with sympathy, based on a survey of 477 respondents from 71 Agile software development teams in 26 companies [30]. The study

investigated the effect of teamwork quality on team performance, learning, and work satisfaction in agile software teams and whether this effect differs from that of traditional software teams. The findings indicated that mutual sympathy between team members leads to better team cohesion and induces pride in the team's successes. This combination of personalities allows team members to collaborate to reach consensus on important issues, while respecting the contribution of each team member. Baumgart, et al. [8] conducted interviews with eleven Scrum team members from seven different companies to identify important characteristics for Agile software development based on practical experience. While self-discipline, along with achievement striving, and dutifulness, were noted as beneficial, altruism emerged as the most essential for collaboration for self-organizing teams studied. Self-disciplined members were more apt to meet goals stated during Sprint daily meetings. Achievement directly related to motivation for a successful product outcome without being deterred by change. Dutiful developers appeared to be morally bound to complete their tasks on time, as well as see that the team succeeds. Self-discipline, coupled with achievement striving, emerged in another study by Adkins [31] as critical factors in distinguishing between successful and unsuccessful team performance.

Authority challenging was noted as a collaborative trait by Hackman and Wageman [5]. In their study that found that collaborative activities in which team members work to achieve a common purpose may also require authority challenging team members. These team members seek to remove unnecessary constraints or adjust poor direction, so that the collaboration can have a positive impact on the team and project performance.

Emotional range as part of the Big Five model also factors into the collaborative effort experienced by a software development team. Graziotin et. al [32] surveyed 189 participants and identified 49 consequences of unhappiness during software development. They suggest that stress, anxiety, burnout, and depression all have adverse effects on software development teams, negatively impacting collaboration. Returning to the study by Baumgart et al. [8], their interviews with eleven Agile Scrum team members from seven different companies identified altruism and openness to change as the most important factors for team members.

III. Study Parameters and Methodology

In this section, we overview how the capstone computer science is organized around learning objectives related to software engineering. We also overview the metrics used to correlate with the collaborative personality assessment findings by IBM Watson™ Personality Insights service.

A. Class Organization and Expectations

The targeted class studies the Agile software development methodology and its application to their team project. The class is organized around three Agile Scrum Sprints across a single semester. Each Sprint lasts for approximately four weeks and requires a deliverable portion of the final product to be completed. Effort between Sprints and at the end of the semester includes time for presentation, customer meetings, and evaluation. The study includes teams from two different semesters of the class that have the same learning objectives and are organized in the same manner.

To accommodate the academic setting, the "Sprint daily check-in" is required by team members during a set time-period approximately twice a week. This meeting check-in requires individual posting on the team's Slack account on the public "general" channel. The check-in must include detailed postings on team member responsibilities regarding (1) what was done since the last check-in, (2) what will be done on the next check-in, and (3) brief comments, such as any impediments the individual is having with his/her tasks, the need to work with another team member, or the offer to help another team member. Face-to-face meeting results are generally discussed within the check-in or using additional Slack posts. This approach maintains the spirit of the "daily check-in" while accepting that college seniors can only devote about a quarter of their time per week to the project. Spacing out the check-ins and increasing the development time per Sprint allows for deliverable requirements to be met and accommodates interactions with external customers.

The Sprints have different work products associated with them to provide for broad education and training in software development. However, they rely on the same collaborative tools. User stories for the product and sprint backlogs, as well as use cases that expand on user story functionality, are part of all sprints. Sprint 1 focuses on establishing a product vision and coding the high-level interaction. The newly formed teams are introduced to Trello for project management with Kanban and to Lucidchart for software modeling. It is expected that Google Drive, GitLab, and Slack are familiar to the students since these are used regularly in earlier classes. These tools allow for joint editing. It should be noted that email and texting among team members is highly discouraged to better enable cooperation through the use of the class tools and tracking.

Sprint 2 focuses on implementation and analysis with teams now gaining an understanding of the capabilities of the team members with respect to language and platform expertise. Sprint 3 introduces packaging and transition requirements in addition to finishing the product to meet the expectations of the customer. Thus, the grade for each sprint encompasses additional work products relevant to that sprint or class training, such as user story expression, proper task description, story association, task distribution among team members, modeling and analysis results, and documentation.

B. Performance Metrics

We use three performance metrics directly associated with the class to correlate with the personality traits investigated in Section II.B. These are Team grade, Sprint participation, and Sprint impression.

1) Team Grade

Grades are the primary performance metric of each sprint. This metric represents the points obtained by the team for the sprint work products. It is not the percentage grade per student which relies on participation and impression (Sections III.B.2 and III.B.3). There are approximately 800 points per sprint.

The point accumulation is high so that a failure to complete one task does not heavily impact the overall project grade if other tasks are completed successfully. The point spread enables better quality consideration as it relates to the resulting deliverable and requirements satisfaction of the customer. Students are given a rubric that, in addition to the deliverable requirements, may include task breakdown, responsibility allocation, sprint backlog maintenance, user story and use case specification and refinement, modeling, coding practices, user interface design, and team presentation.

2) Sprint Participation

The sprint participation score is calculated by the quantitative and qualitative examination of an individual's check-in on Slack throughout each sprint. The check-in requirements maintain student engagement with teams throughout the semester. The participation score for a single check-in is based on the content details, addressing the three activities of the check-in, and timeliness. It may encompass multiple Slack posts. Because the check-in relates to responsibility, accountability, and project and team awareness, we use this as a performance measure with the assumption that the student who participates more actively is more likely to work toward a successful product outcome. In addition, the participation score is an individual score that is added to the team points for the overall sprint grade. We have previously shown that Team Grade, Participation, and Impression (discussed next) are highly correlated with each other [33].

3) Sprint Impression

The sprint impression score is based on self and team evaluations performed at the end of each sprint. Ratings are given for twenty questions that assess capabilities, engagement, and product knowledge. The sprint impression is the average across the scored questions for all team members. A question may not be scored if the team member is not assigned a task for which a rating can be determined. For example, if the question relates to modeling practices and the team member was not assigned a modeling task for the sprint, then no score would be given. Evaluation is taken very seriously by the teams. Besides providing a forum for expression regarding dynamics, the evaluation scores factor into an individual's sprint grade by weighting the team grade. For example, an impression score of 8.0 weights the team grade by 0.92, nearly dropping the awarded team points to an individual by a letter grade. A team member that does not submit an evaluation receives a 0 averaged into the impression scores of their team members. In addition to the impression score, we examine the average scores for each team member for specific evaluation questions that are most relevant to assessing collaboration. Table 1 shows the 20 survey questions for peer evaluation after each Sprint, from which the 6 questions assessed are taken. The 6 questions studies are in bold, italicized text in Table 1.

TABLE 1. PEER EVALUATION QUESTIONS

Q1	Participated in online and face-to-face meetings
Q2	Responded to team activity in a timely manner
Q3	***Contributed to project requirements gathering and/or review***
Q4	Contributed project ideas that were implemented
Q5	Contributed to document artifacts creation and/or review
Q6	***Contributed to project code and/or testing***
Q7	Set the standard for team performance
Q8	Independently learned something new to address project requirements
Q9	Completed all tasks assigned at agreed upon timeline
Q10	***Significantly aided the team or another team member during the semester***
Q11	Organized the user stories, Product Backlog, and Sprint Backlog
Q12	Communicated project and instructor expectations well
Q13	Performed design-based tasks (such as use cases and wireframe) effectively
Q14	Initiated and maintained required documents
Q15	***Worked with team to create overall product vision***
Q16	***Worked with team to ensure quality product interface***
Q17	Delegated tasks appropriately
Q18	Demonstrated exceptional programming ability
Q19	***Trained team members in new coding techniques***
Q20	Maintained open access to code for review

C. Slack.com Posts

A team Slack account is the primary mechanism for discussion outside of face-to-face meetings. Students are encouraged to keep the conversation on public channels only and are penalized if a Slack report shows a high percentage of private or direct message posts. Trello and GitLab are integrated into the account to maintain project awareness. Additional public channels can be enabled for task-specific communication. Because the collaborative tools, including Slack, are an integral part of the course, the students know that their data is captured during the course and can be used in cases to validate participation, such as when a team complains about a member not completing assigned tasks. As part of the study IRB protocol, students are asked to voluntarily consent to the use of their data for research purposes. They are made aware that their consent forms are not provided to the instructor until after grades are submitted and that their team is removed from any study unless all team members consent. The hope is that this practice reduces any bias that students have regarding what they post and leads to more organic exchanges with their teams.

Slack allows posts from all public channels to be exported into a JSON file, which can be processed for use by the IBM Watson™ Personality Insights service. We perform the export, remove posts by Trello and GitLab, and segregate the posts per user per sprint.

IV. RESULTS AND INTERPRETATION

This section shows the correlation of the personality assessment of Slack posts with the performance metrics presented in Section III.B. We filtered the Slack posts of 40 students across 10 teams by user and by Sprint. There was a total of 6210 posts, with 31% in Sprint 1, 36% in Sprint 2, and 33% in Sprint 3. We then processed them on IBM Watson™ Personality Insights service. We extracted the results for the 13 identified personalities with relevance to collaboration

activities, based on the literature surveyed and discussed in Section II.B. Tables 2 – 4 show these personality traits (row 2) according to their association with the Big Five model (row 1).

Using Pearson's correlation coefficient r, for a two-tailed test with 38 degrees of freedom, given 40 samples, the following critical values are computed, and are the same for each Sprint:

- $|r| > 0.402$ corresponds to a very strong correlation with $p = 0.01$ (Shown as bold and surrounded by a thicker border in the tables)
- $|r| > 0.312$ represents strong correlation with $p = 0.05$ (Bold only in the tables)
- $|r| > 0.263$ represents significant correlation with $p = 0.10$ (Not differentiated in the tables)

Tables 2-4 show the results for each sprint referencing the literature from which the collaborative personality trait was identified.

For Sprint 1 (Table 2), 6 traits in total correlate with team grades. Cooperation, cautiousness, and dutifulness have a very strong correlation. Sympathy has a strong correlation. Authority challenging and susceptible to stress have significant correlations. Surprisingly, none of these traits correlate to the quality of the check-ins (participation), team member assessment (impression), or the individual scores for questions related to collaboration. Adventurousness has no correlation with grades yet is correlated with participation and the impression scores. Because this was the first sprint on a new team, it is likely they awarded higher evaluation scores to those team members that quickly engaged with getting the project moving forward. One interpretation of sympathy's correlation in Sprint 1 is that early on, sympathetic students were understanding about skill set issues and encouraged teamwork to cover all of the work products. Openness to change has a strong positive correlation with impression and a positive correlation with Q15 and Q16 (see Table 2). This trait appears to present the most collaborative activity as evaluated by the team in Sprint 1, which is especially important as the team formulates their product vision and initial requirements, all while assessing either others' skillsets and work ethic.

Dutifulness negatively correlated with impression questions Q3, Q6, Q10 and Q15. One interpretation is that with it being the first sprint, there was more tolerance for students not completing a portion of their task on time. Some tasks could be pushed back to later sprints and team members may not recognize if a teammate is having trouble or slacking off in this early stage. Susceptible to stress has a positive correlation with grade but a negative correlation with impression. Susceptible to stress may have increased the work effort but was not rewarded by team members. Susceptible to tress also has a strong negative correlation with collaboration specific questions Q3, Q6, Q10 and Q16, which could mean that team members that are susceptible to stress may have either hid their work from the team or acted as if they were on their own.

For Sprint 2 (Table 3), only self-discipline positively correlated with grades. On interpretation is that after the first sprint, students have a clearer understanding of the project, customer, and team member capabilities and work habits. Those who have self-discipline are capable of completing most of the required tasks to achieve a high grade. Authority-challenging has a strong negative correlation with grades, yet a strong positive correlation with impression. This result may indicate that the authority challenging personality traits may be viewed by team members as embodying leadership qualities, but do not produce the desired product results. Several personality traits correlate with impression for Sprint 2. At the strongest level are cooperation, trust, and achievement-striving. Authority challenging, altruism, sympathy, cautiousness personalities and adventurousness all positively correlate with impression. From a team perspective, these traits may indicate that collaboration is beginning to be viewed positively by team members.

Interestingly, cooperation has a strong negative correlation with Q19 on training. As does achievement striving, dutifulness, and trust have strong negative correlations with Q19, is in contrast the literature. By Sprint 2, the team should be familiar with the coding language and tools they will be using and should not need additional training. If a team member does need training, then they are likely perceived as behind or slacking. Q3 has strong positive correlation with altruism, sympathy and activity level, which can mean that as requirements are being assessed and possibly changed, these collaborative traits are more appreciated. Similarly, sympathy has a significant correlation with both Q6 and Q10. Being open to change has a negative correlation with grade in Sprint 2 and a positive correlation with Q19. It is unclear why this relationship occurs with Grade, because Sprint 2 often sees a lot of product changes. However, team members with this trait may be less concerned about the change, as well as continually trying to help other team members that are not yet ready. We see this a lot with the use of GitLab, if a student was not fully exposed to its use.

Sprint 3 had less correlation between personality traits and performance metrics. For Sprint 3 (Table 4), altruism and sympathy have a strong correlation to grades, with trust showing some correlation. These three characteristics convey the kind of collaborative attitudes that would be expected at the end of the semester for a well-connected team, indicating that the use of the collaborative online tools may be fostering this interconnection. Sympathy correlates with impression, suggesting that those who are sympathetic to their team members needs at the end of the semester are seen as the leaders in the product completion. Authority challenging had a significant correlation with both Q6 and Q16, which both point to a team member that appears to want a quality product and may be pushing the team toward that end. Openness to change highly correlates with grade. In the final sprint, collaboration is needed to make quick changes to successfully produce the product that the customer wants. Altruism had a negative correlation with Q16, which makes sense that altruism as a collaborative trait may mean less quality during the fast-paced final sprint. Being open to change had a positive correlation with Q19, which should not be occurring at the end of the semester, yet those with this trait apparently were accepting of having to train.

TABLE 2. SPRINT 1 - CORRELATION OF PERSONALITY ASSESSMENT WITH PERFORMANCE METRICS

BIG FIVE	OPENNESS		AGREEABLENESS				CONSCIENTIOUSNESS				EXTRA-VERSION	EMOTIONAL RANGE	
	Adventurousness [6]	Authority challenging [5]	Altruism [8], [9]	Cooperation [4], [7], [26], [27]	Sympathy [30]	Trust [4], [28]	Cautiousness [6], [26]	Self-discipline [8], [9], [30], [31]	Achievement striving [8], [9], [30], [31]	Dutifulness [8], [9]	Activity level [3]	Susceptible to Stress [32]	Openness to change [8]
Grade	-0.182	0.282	0.251	**0.566**	**0.343**	0.052	**0.420**	0.004	0.048	**0.483**	-0.225	0.276	-0.027
Participation	**0.314**	0.104	0.101	0.043	0.103	0.084	0.006	-0.129	0.095	-0.014	0.117	-0.142	-0.026
Impression	**0.445**	-0.089	0.126	-0.192	-0.091	0.311	**-0.342**	0.062	-0.051	-0.233	0.049	**-0.384**	**0.327**
Q3: Reqs	**0.398**	-0.169	0.102	-0.249	-0.160	0.069	-0.242	0.114	0.101	**-0.324**	0.304	**-0.428**	0.310
Q6: Code	**0.384**	-0.094	0.109	-0.129	-0.069	0.118	-0.224	-0.040	-0.022	-0.274	0.118	**-0.317**	0.218
Q10: Aid	**0.441**	-0.073	0.167	-0.161	-0.012	0.143	-0.225	0.076	0.096	-0.287	0.235	**-0.452**	0.311
Q15: Vision	**0.335**	-0.036	0.215	-0.148	-0.046	0.191	-0.232	-0.073	-0.034	-0.267	0.200	-0.292	**0.363**
Q16: Interface	**0.486**	0.076	0.205	-0.084	-0.117	**0.314**	-0.136	0.086	0.107	-0.194	0.264	**-0.451**	**0.344**
Q19: Train	0.263	-0.018	0.247	0.116	0.036	0.006	-0.069	-0.051	-0.063	-0.052	-0.089	-0.152	0.302

For $p < 0.01$, $|r| > 0.402$; $p < 0.05$, $|r| > 0.312$; $p < 0.10$, $|r| > 0.263$

TABLE 3. SPRINT 2 - CORRELATION OF PERSONALITY ASSESSMENT WITH PERFORMANCE METRICS

BIG FIVE	OPENNESS		AGREEABLENESS				CONSCIENTIOUSNESS				EXTRA-VERSION	EMOTIONAL RANGE	
	Adventurousness [6]	Authority challenging [5]	Altruism [8], [9]	Cooperation [4], [7], [26], [27]	Sympathy [30]	Trust [4], [28]	Cautiousness [6], [26]	Self-discipline [8], [9], [30], [31]	Achievement striving [8], [9], [30], [31]	Dutifulness [8], [9]	Activity level [3]	Susceptible to Stress [32]	Openness to change [8]
Grade	0.005	**-0.365**	0.116	0.023	-0.022	-0.202	-0.014	**0.313**	0.132	0.237	0.072	-0.356	-0.309
Participation	0.229	0.221	0.237	**0.471**	**0.392**	0.194	0.172	0.121	0.191	0.189	0.102	-0.200	-0.192
Impression	0.298	**0.365**	**0.327**	**0.408**	**0.355**	**0.447**	**0.321**	0.222	**0.422**	0.248	0.212	-0.289	0.021
Q3: Reqs	0.128	-0.035	**0.329**	0.023	**0.334**	0.141	0.014	0.119	0.146	0.056	**0.328**	-0.143	0.086
Q6: Code	0.161	-0.022	0.230	0.011	0.294	0.003	0.099	0.036	0.068	-0.054	0.202	-0.111	0.041
Q10: Aid	0.086	-0.064	0.261	-0.038	0.274	0.003	0.039	0.014	0.013	-0.084	0.181	-0.062	0.119
Q15: Vision	0.053	0.002	0.053	-0.083	0.234	-0.090	0.023	-0.108	-0.045	-0.139	0.127	-0.063	0.064
Q16: Interface	0.139	0.087	-0.040	-0.033	0.088	0.025	0.081	0.009	-0.014	-0.043	0.208	-0.179	0.039
Q19: Train	0.073	-0.170	-0.090	**-0.415**	-0.170	-0.280	-0.206	-0.243	**-0.365**	**-0.371**	-0.172	-0.068	**0.357**

For $p < 0.01$, $|r| > 0.402$; $p < 0.05$, $|r| > 0.312$; $p < 0.10$, $|r| > 0.263$

TABLE 4. SPRINT 3 - CORRELATION OF PERSONALITY ASSESSMENT WITH PERFORMANCE METRICS

BIG FIVE	OPENNESS		AGREEABLENESS				CONSCIENTIOUSNESS				EXTRA-VERSION	EMOTIONAL RANGE	
	Adventurousness [6]	Authority challenging [5]	Altruism [8], [9]	Cooperation [4], [7], [26], [27]	Sympathy [30]	Trust [4], [28]	Cautiousness [6], [26]	Self-discipline [8], [9], [30], [31]	Achievement striving [8], [9], [30], [31]	Dutifulness [8], [9]	Activity level [3]	Susceptible to Stress [32]	Openness to change [8]
Grade	0.062	0.043	**0.342**	0.130	**0.380**	0.282	-0.052	0.177	0.241	-0.026	0.262	0.191	0.273
Participation	0.067	0.163	0.109	0.178	0.215	0.146	0.182	0.221	0.182	0.244	0.131	-0.042	0.025
Impression	0.050	0.186	0.058	0.145	0.300	0.044	-0.022	-0.027	0.023	-0.054	0.052	0.261	0.090
Q3: Reqs	0.006	0.189	-0.066	0.015	0.118	0.156	-0.040	-0.058	0.060	0.018	0.090	0.193	0.068
Q6: Code	-0.008	0.270	-0.129	0.085	0.045	0.139	0.031	-0.092	0.019	0.064	0.041	0.165	-0.035
Q10: Aid	-0.025	0.218	-0.099	0.045	0.045	0.217	0.012	-0.013	0.064	0.089	0.123	0.129	0.079
Q15: Vision	-0.023	0.145	-0.025	-0.130	0.078	0.108	-0.129	-0.025	0.059	-0.059	0.073	0.130	0.177
Q16: Interface	-0.066	0.284	-0.293	0.071	-0.123	0.115	0.111	-0.185	-0.050	-0.052	-0.045	0.120	-0.034
Q19: Train	-0.144	-0.090	0.174	-0.222	0.128	0.150	-0.204	-0.022	0.069	-0.049	0.130	0.120	0.292

For $p < 0.01$, $|r| > 0.402$; $p < 0.05$, $|r| > 0.312$; $p < 0.10$, $|r| > 0.263$

The sharp drop in number of correlating collaborative personality traits between Sprint 1 and Sprint 2 may be related to the teams becoming more familiar with each other. Sprint 1 focused on getting the product started, requiring a deliverable that showed what could be done and worked to some degree. In Sprint 1, the help that would be needed was not well-understood, leading to a small number of correlations for sympathy and altruism. Sprint 2, on the other hand, required the team to make changes to their original product based on customer feedback while also pushing forward with substantial development. Additionally, midterms and spring break always occur during Sprint 2, leading to delays and missed deadlines as students prioritize studying to working on the project or preparing to have the week off. Traits like sympathy and altruism become more important as changes are requested and delays occur. This outcome likely led to the higher impression correlation and correlations with the survey questions.

There is also a drop in the overall number of correlations between Sprint 2 and Sprint 3. Sprint 3 focuses on students completing documentation and ensuring all requested functionality is implemented. Bug fixes are common in this sprint. It is important for team members to be altruistic in taking on more work to ensure completion, sympathetic to each other's concerns, such as graduation, finals, and other stressful situations, and trusting in each other to complete the tasks.

It is unclear if (1) students alter their conversation and posts depending on the task at hand or (2) students with the needed personality traits for that sprint at that time carry the team toward success and team members lacking those traits may be less successful or (3) comfortability with the team or, at least, better knowledge of the personalities among the team alter a team member's post contents if they are trying to maintain a collaborative team to reach a successful project conclusion for the particular sprint.

The research overviewed in Section 2 suggest that the 13 personality traits in Tables 2-4 positively supported collaboration. Our findings are somewhat different for undergraduate students studied. If we focus on grades as the primary performance metric, altruism, cooperation, sympathy, cautiousness, self-discipline, and dutifulness have a positive relationship at some time during the semester. Sympathy is the only trait that appears with a positive correlation to grades in more than one sprint. Authority challenging is the only trait with a negative correlation to grades within a single sprint. The remaining traits appear to not have a relationship with grades.

V. PROMOTING COLLABORATION

Though the capstone course relies on several collaborative tools, students are not explicitly made aware of how they can improve their collaborative activities without taking on more than their share of the workload. Incentive is one mechanism. It has been suggested that paying for performance increases intrinsic and self-deterministic motivation [34, 35]. Translating this to a software development team means rewarding collaborative behaviors. The challenge is how to identify these behaviors in a way that will provide timely feedback in terms of short and long-term rewards. The findings from the study, as discussed in Section IV, indicate that different collaborative behaviors are more effective depending on the sprint. Thus, rewarding adventurousness in Sprint 1 by giving credit for asking the customer detailed questions or stating new ideas for the project to meet customer

requirements in team meetings could promote collaboration toward positive outcomes.

Similarly, for Sprint 2, rewarding cooperation may include giving credit for team members that work with all of the online tools, instead of focusing on just GitLab for coding or Google Drive for documentation. If team members interact across all tools, then there will be additional collaborative experiences. With Sprint 3, altruism and sympathy can be rewarded by given credit to team members that directly support another team member, such as through code review, pair programming, and unit testing. This style of reinforcement learning to promote increase collaboration will be deployed in the course for the next two years and studied.

VI. Limitations

We acknowledge that our work has some limitations. First, we use data from only 40 students in 10 teams over 2 semesters. A larger study, to be conducted in the future, may provide clearer insight. We acknowledge that, though there are penalties for unauthorized project discussion that is not face-to-face or on Slack, some teams pursue other discussion venues like Discord and can keep it hidden. The average number of posts on Slack per individual in this study during a semester is 150 posts, which is a little over 10 posts per week. This number provides sufficient data for analysis. Teams without sufficient conversation data are excluded from study.

Another limitation of this study is that the requirements, activities, and timelines are different for each sprint. The number of comments in Slack and the roles and responsibilities shared or individualized among team members is also different. While Sprint 1 can claim significant attention by the team, Sprint 2 competes with midterms, job and graduate school interviewing, and spring break. Sprint 3 competes with graduation preparation and finals. Though the IBM Watson™ Personality Insights service performs a reasonable analysis given sufficient comments to analyze, it is less accurate for students who might post only the minimum amount required for meeting check-in. This analysis must be taken into account when examining the personality traits relationship with performance metrics.

VII. Conclusions

In this paper, we surveyed papers linking personality traits to team collaboration to identify 13 traits that specifically related for software development teams. We exported Slack.com posts from 40 students from a capstone software development course across 3 sprints to study student personality traits using the IBM Watson™ Personality Insights service. The results suggest a diversity in the roles and timing of collaborative trait usefulness for each Sprint.

Because it is a team project, understanding the personality traits exhibited by teams and the relationship to performance metrics and product outcome would provide a difference perspective on collaboration. Part of the future work will be to determine an assessment model for team personality. In addition, to better collaborate, team members should be trained to recognize and encourage these traits in themselves and their teammates and exploit them at proper times during the Agile development process. Our future work will experiment with methods of training students in the discovery, tolerance, and usefulness of collaborative personality to improve team performance as it relates to their capstone project and later software development. We plan to replicate the study across multiple engineering departments with capstone courses that include team projects and use Slack for project management.

An outcome of this type of research can lead to collaborative tool developers creating online tools that promote better collaborative activity. For example, Google Drive could include “agree/disagree/neutral” buttons related to collaborative document editing with supporting comments, if applicable, to let team members be aware of other team members’ reviews. Quick buttons could then generate more discussion and engagement in the activity to bring it to a positive conclusion. Additional research is also needed to map individuals with collaborative personality traits as expressed by the IBM Watson™ Personality Insights service based on their posts to quantitative and qualitative individual performance metrics. These metrics can be obtained by the type and quantity of events using the other cooperative tools that are part of the course. For example, mapping the personality traits to the number of Git commits in GitLab and the size of these commits would provide quantitative information about the work completed as it relates to a particular personality trait. We will examine these personality traits across access and direct interaction with all of the collaboration tools used in the capstone course.

We also plan to further investigate methods to encourage collaborative behaviors, such as explicitly increasing the value of collaborative grading criteria or requiring a certain level of input into each of the collaborative tools. Additionally, each meeting check-in could be made to more clearly express collaborative activities, requiring each team member to state how and with whom they collaborated since the last check-in. This concept could also be expanded to include the identification and encouragement of personality traits or activities which we have shown here correlate to improved grade for each sprint.

REFERENCES

[1] W. Norman, “Toward an adequate taxonomy of personality attributes: Replicated factor structure in peer nomination personality rating,” J. of Abnormal and Social Psychology, 66:6(574-583), 1963.

[2] P. Costa and R. McCrae, “Revised NEO Personality Inventory (NEO-PI-R) and NEO Five-Factor Inventory (NEO-FFI) Manual,” Odessa, FL: Psychological Assessment Resources, 1992.

[3] A. Pentland, “The New Science of Building Great Teams,” Harvard Business Review, 90:4(60-70), 2012.

[4] B. Ramesh, L. Cao, K. Mohan, and P. Xu, “Can distributed software development be agile?,” Commun. ACM, 49:10(41-46), 2006.

[5] R. Hackman and R. Wageman, “When and how team leaders matter,” Research in Organizational Behavior, 26(37–74), 2004.

[6] P. McCuller, “How to Recruit and Hire Great Software Engineers: Building a Crack Development Team,” Apress, Berkely, CA, 2012.

[7] F. Lanubile, “Collaboration in Distributed Software Development,” In Software Engineering, A. Lucia and F. Ferrucci (Eds.). Lecture Notes in Computer Science, 5413(174-193), Springer-Verlag, 2009.

[8] R. Baumgart, M. Hummel, and R. Holten, “Personality Traits of Scrum Roles in Agile Software Development Teams-A Qualitative Analysis,” European Conf.on Information Systems, 2015.

[9] T. Chow and D. Cao, “A survey study of critical success factors in agile software projects,” J. Syst. Softw. 81, 6 , June 2008, 961-971.

[10] T. Davenport and J. Kirby, "Just How Smart Are Smart Machines?," MIT Sloan Management Review, Cambridge Vol. 57, Iss. 3, 2016.

[11] P. Arnoux, A. Xu, N. Boyette, J. Mahmud, R. Akkirajuand, and V. Sinha, “25 Tweets to Know You: A New Model to Predict Personality with Social Media,” Int’l AAAI Conf. on Web and Social Media, 2017.

[12] H. A. Schwartz, J. C. Eichstaedt, M. L. Kern, L. Dziurzynski, S. M. Ramones, and M. Agrawal, “Personality, gender, and age in the language of social media: The open-vocabulary approach,” PLoS ONE, 8 e73791, 2013.

[13] B. Plank and D. Hovy, “Personality Traits on Twitteror or How to Get 1,500 Personality Tests in a Week,” Proc. of the 6th Wksp on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, 2015.

[14] K. Schwaber, “Agile Project Management with Scrum,” Redmond,WA:Microsoft Press, 2004.

[15] J. Whitehead, I. Mistrík, J. Grundy, and A. V. Hoek,“Collaborative software engineering: concepts and techniques,”In Collaborative Software Engineering. Springer, 1—30,2010.

[16] M. Bastarrica, D. Perovich, and M. Samary, “What can students get from a software engineering capstone course?,”In Proc. of the 39th Int'l Conf. on Software Engineering: Software Engineering and Education Track . IEEE Press, Piscataway, NJ, USA, 137-145, 2017.

[17] M. Missiroli, D.Russo, and P. Ciancarini, “Learning Agile software development in high school: an investigation,” In Proc. of the 38th Int'l Conf. on Software Engineering Companion. ACM, New York, NY, USA, 293-302, 2016.

[18] T. M. Vitolo, K. E. Hersch and B. J. Brinkman, "Making the connection: Successful cross campus collaboration among students," 2016 IEEE Frontiers in Education Conf., Erie, PA, USA, 2016, pp. 1-7.

[19] J. W. Moy and K. F. Lam, “Selection criteria and the impact of personality on getting hired,” Personnel Review, 33:5(521–535), 2004.

[20] H. Bradley and F. J. Hebert, “The effect of personality type on team performance,” J. of Management Development, 16:5(337–353), 1997.

[21] L. Capretz and F. Ahmed, “Why do we need personality diversity in software engineering?,” SIGSOFT Softw. Eng. Notes, 35:2(1-11), 2010.

[22] O. P. John and S.Srivastava, “The big five trait taxonomy: History, measurement and theoretical perspectives,” In Handbook of Personality: Theory and Research: 2nd Ed., pp. 102--138. The Guilford Press, 1999.

[23] J. Penningn, R. Socher, and C.D. Manning, “Glove: global vectors for word representation,” Proc. of the Empiricial Methods in Natural Language Processing, pp. 1-12, 2014.

[24] K. Lee, J. Mahmud, J. Chen, M. Zhou, and J. Nichols, “Who Will Retweet This? Detecting Strangers from Twitter to Retweet Information,” ACM Transactions on Intelligent Systems and Technology, vol. 6, no. 3, pp. 1–25, Apr. 2015.

[25] Z. Liu, Y. Wang, J. Mahmud, R. Akkiraju, J. Schoudt, A. Xu, and B. Donovan, “To Buy or Not to Buy? Understanding the Role of Personality Traits in Predicting Consumer Behaviors,” Social Informatics, pp. 337–346, 2016.

[26] H. Bani-Salameh and C. Jeffery, “Collaborative and social development environments: a literature review,” Int. J. Comput. Appl. Technol. 49:2(89-103), 2014.

[27] S. Faraj and L. Sproull, “Coordinating Expertise in Software Development Teams,” Manage. Sci., 46:12, December 2000.

[28] E. Weimar, A. Nugroho, J. Visser, A. Plaat, M. Goudbeek, and A. P. Schouten, “The influence of teamwork quality on software team performance,” arXiv preprint arXiv:1701.06146, 2017.

[29] M. Hoegl and H. G. Gemuenden, “Teamwork Quality and the Success of Innovative Projects: A Theoretical Concept and Empirical Evidence,” Organization Science 12, 4, July 2001, 435-449.

[30] Y. Lindsjørn, D.Sjøberg, T. Dingsøyr, G.Bergersen, and T. Dybå, “Teamwork quality and project success in software development,” J. Syst. Softw. 122, December 2016.

[31] L. Adkins, “Coaching Agile Teams: A Companion for Scrummasters, Agile Coaches, and Project Managers in Transition,” ,1st ed., Addison-Wesley Professional, 2010.

[32] D. Graziotin, F. Fagerholm, X. Wang, and P. Abrahamsson, “Consequences of unhappiness while developing software,” In Proc. of the 2nd Int'l Wksp on Emotion Awareness in Software Engineering. IEEE Press, pp. 42-47, 2017.

[33] R. F. Gamble and M. L. Hale, “Assessing individual performance in Agile undergraduate software engineering teams,” 2013 IEEE Frontiers in Education Conference (FIE), Oct. 2013.

[34] P. Moisés and C.Russell, A. & V.-Gupta, “Effects of Motivation on Educational Attainment: Ethnic and Developmental Differences Among First-Generation Students,” J. of Hispanic Higher Education, 2012.

[35] A. Young, J.G., Arthur, H. Hawthorne,“Cultural and socioeconomic differences in academic motivation and achievement: A self-deterministic approach,” Journal of Border Educational Research, 9(1), 2013.