Data Science Approaches to Prevent Failure in Systems Engineering

Technical Report SERC-2019-RT-008

June 14, 2019

Principal Investigator: Dr. Karen Marais, Purdue University

Co-Principal Investigator: Dr. Bruno Ribeiro, Purdue University

Research Team:

Georgios Georgalis, Research Assistant, Purdue University

Leonardo de Abreu Cotta, Research Assistant, Purdue University

Sponsor: DASD(SE)

The Systems Engineering Research Center (SERC) is a federally funded University Affiliated Research Center managed by Stevens Institute of Technology.

This material is based upon work supported, in whole or in part, by the U.S. Department of Defense through the Office of the Assistant Secretary of Defense for Research and Engineering (ASD(R&E)) under Contract HQ0034-13-D-0004 (Task Order 0263, RT 206).

Any views, opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense nor ASD(R&E).

SUMMARY

This technical report documents progress under SERC RT-206 between June 15th, 2018 (task order start date) and June 14th, 2019 (task order completion date). The primary motivation for this research effort is a pressing need to identify ways of tracking project risk to prevent future systems engineering failures, while advances in data science approaches and neural network applications are the enablers.

Our work focuses on developing automated ways of tracking project risk based on two types of readily available information: enterprise software-derived data (*Company inputs*) and employee data collected via an app (*Crowd inputs*). The *Company inputs* carry risk information related to the daily operations of the organization (e.g., inventory data, number of failed parts, or financial data). We augment the database with *Crowd inputs* because we want to know what the people in the organization are doing to contribute to project risk. The underlying principle of our process is to collect these inputs continuously, frequently, and efficiently, and then process them using machine learning algorithms to predict failures. By predicting failures, we can make decision makers aware of the current risk of the projects in their organization, therefore giving them the opportunity to react before a failure occurs.

In this effort, we focused on developing the main functions of the failure prediction prototype and evaluating whether our approach is a valid process to measure risk. We did so by testing our prototype and process in engineering student teams at Purdue University. During the first year of support we have:

- Identified a set of potential causal or related factors that lead to failure and developed questions that aim to uncover the presence of these factors (*Crowd inputs*)
- Collected data for three semesters from design projects at Purdue University
- Completed statistical analyses to identify which *Crowd inputs* correlate with which types of failures
- Developed deep relational learning models that predict future project failures and failure causes

The report is organized as follows: First, we describe our process to identify factors that are associated with failures and to develop crowd signals that measure these factors. Second, we describe how we collected data from student design teams at Purdue University. Then, we present a series of mixed effects logistic regression models we trained using the collected data and their interpretation to identify which crowd signals correlate with increased probability of a failure or failure cause occurring during a project. The last section describes a deep learning approach to predict future failures and failure causes. We conclude the report with a summary of our completed work during the first year of the task and our plans for extending this work towards the goal of completing our prototype.

This Page Intentionally Left Blank

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF (TABLES, SEQUENCES)

INTRODUCTION

Anecdotes and statistics on the failures of systems engineering have become a sure-fire way of attracting attention and lamentation during presentations. No-one is immune to the failure disease and in particular past success is no guarantee of future performance—organizations that have succeeded spectacularly in one project may fail just as spectacularly in the next project.

For example, of 72 major United States defense programs in progress in 2008, only eleven of them were on time, on budget, and met performance criteria (Charette, 2008). The problems for U.S. aerospace and defense programs have only worsened since then: total cost overruns "have risen from 28 percent to 48 percent, from 2007 through 2015" (Lineberger and Hussain, 2016). In a recent assessment of U.S. Defense Acquisitions, the U.S. Government Accountability Office (GAO) found that these programs were "not yet fully following a knowledge-based acquisition approach", which will result in "cost growth or schedule delays" (GAO, 2017). The consumer goods sector also has many failures, such as the Xbox 360 "Red Rings of Death" or the Ford Explorer rollover problems (Takahashi, 2008; Bradsher, 2000).

In response to these dire statistics, new methods, processes, and tools are continuously proposed and implemented, including numerous new methods of risk identification, tracking, and management. Yet the frequency of failures shows no signs of decreasing, and, meanwhile, engineering creativity in large complex systems seems to be stifled. Why do these methods not help as much as we hoped? One possible reason, is the reliance on extensive data creation, collection, and tracking. When projects are under pressure, activities that are seen as non-essential to the core task will not be performed, or, worse, will be performed in a cursory compliance-oriented fashion, potentially leading to misleading data and erroneous conclusions about the state of risk.

Most systems engineering failures, even those in new, one-of-a-kind high-tech systems, do not involve previously unknown phenomena, or black swans (Sorenson and Marais, 2016). As appealing as the black swan metaphor is, the *real reasons* for most failures are, in fact, rather prosaic and predictable white swans. Sorenson and Marais identified a set of 22 "real reasons", ranging from "conducted poor requirements engineering" to "created inadequate procedures".

In complex development projects, neither traditional engineering management data nor big data analysis is able to consistently and accurately pinpoint issues. Failures, even though they may appear simple in retrospect, are often the result of a complex network of decisions, many of them locally and temporally rational. Modeling and predicting such complex events requires complex models; and complex models need large amounts of historical data to give accurate predictions and insights; cutting-edge projects do not have an abundance of historical data. More (and possibly better) data are needed.

In these complex scenarios, there is a potential to augment existing engineering management data with "wisdom of the crowd" information. Wisdom of the crowd (WoC) refers to the hypothesis that the collective opinion of a large number of non-experts (e.g., a novice engineer)

is a better signal to the health of a project than the opinion of a single expert (say, an experienced manager). For instance, employees give their best assessment of the timeline and budget of a project, given their knowledge of the system. These assessments are then combined with a machine learning algorithm to predict the probability that the project will be successful or the system will fail. Unfortunately, with bonuses and salaries depending on contracts, it is challenging to ensure employees truthfully report their project estimates.

Our effort leverages two main ideas: (1) risk assessment based on the "real reasons" for systems engineering failures, and (2) combining existing data with Wisdom of the Crowd (WoC) indicators to uncover the correlations between various (unreliable) traditional and crowd-derived measures and the measurable outcome (success, failure, or delay).

Figure 1 summarizes our vision of a tool to help organizations track and manage the risks of project failures. The tool is built around state-of-the-art relational deep learning (Meng et. al., 2018), together with contextual bandit techniques (Lin et al. 2010), using a combination of enterprise data, and "Wisdom of the Crowds" data that employees enter into a mobile device app. The prediction algorithm can be continually refined using each organization's own data.

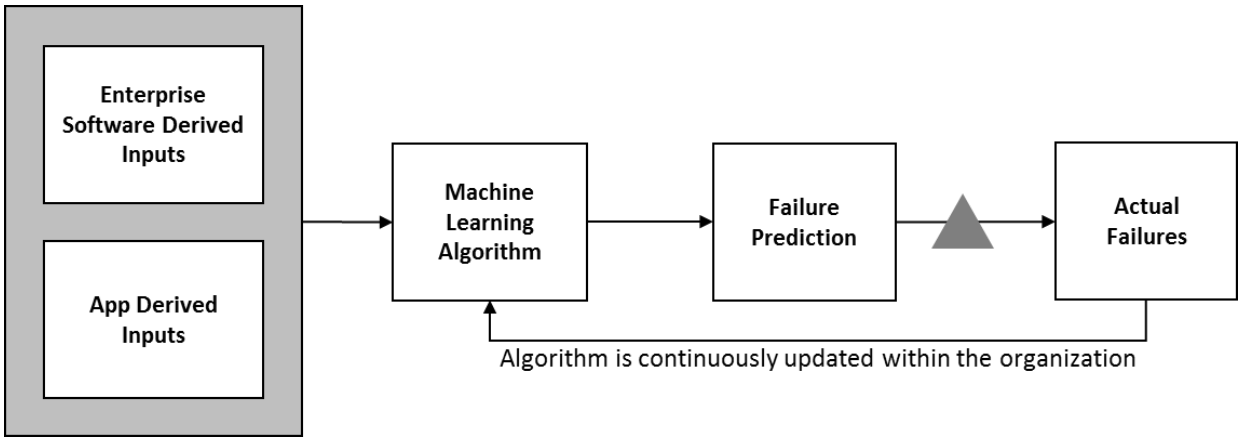

Figure 1: Envisioned Final Product: A prototype for risk assessment

In complex cutting-edge projects, neither traditional data tracking nor even Big Data analysis is able to consistently and accurately pinpoint issues. Failures, even though they may appear simple on the surface (the team should have had a contingency plan!), are often the result of a complex network of decisions, many of them locally and temporally rational. Modeling and predicting such complex events requires complex models; and complex models need large amounts of historical data to be able to produce accurate predictions and insights; cutting-edge projects do not have an abundance of historical data (and even when data is available, it may be hard to collect and put into appropriate formats).

In such complex scenarios that historical failure information is not available or attainable, augmenting traditional enterprise data with "wisdom of the crowd" information, may be sufficient for failure prediction. Wisdom-of-the-crowd (WoC) refers to the hypothesis that the collective opinion of a large number of non-experts (e.g., a novice engineer) is better than the

opinion of a single expert (say, an experienced manager). For instance, employees give their best assessment of the timeline and budget of a project, given their knowledge of the system. These assessments are then combined with a machine learning algorithm to predict the probability that the project will be successful or the system will fail.

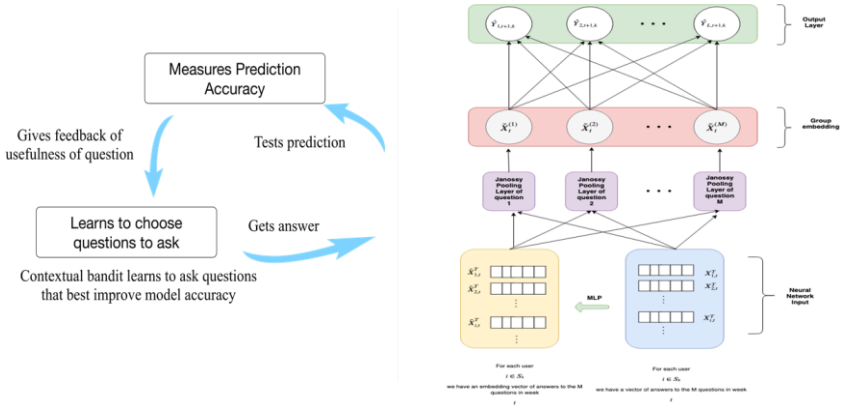

Figure 2: Neural network prediction model with contextual bandit algorithm for question selection.

Our interpretable neural network (Meng et al. 2018) learns which patterns of answers related WoC questions and measurable company data predict failures. The particular structure of our model requires orders of magnitude less data than traditional neural networks (tens of examples will be enough to train a first-generation model).

Hard-to-game WoC: Unfortunately, with bonuses and salaries depending on contracts, it is challenging to ensure employees truthfully report their project estimates (private information). Moreover, employees that have close relationships might give similar assessments and the collected data may be incomplete, further increasing the influence of these correlations. Correlations and biases must be accounted for in the final predictions. We address these limitations in two ways: (1) by asking some questions that are highly correlated with the predicted outcome but are hard to "game" (see Section 5.3), and (2) using new techniques developed by PI Ribeiro (Meng et al. 2018) to make predictions using complex relational patterns (Figure 3). The machine learning model can automatically learn which patterns in the WoC answers, combined with company data, tends to predict failure.

Initially, too many potential WoC questions: We show below how we can identify a large pool of potential questions. However, no team member is willing to answer hundreds of questions every week. We need to balance the number and frequency of promising questions with our need for information. Here, we will borrow from a widely successful technique used for online testing and advertisement: contextual bandits (Lin et al. 2010). A/B testing is a way for comparing the efficacy of two different versions of, say, a web page. The existing web page design is the null hypothesis, and the proposed design is the alternative hypothesis (similar to between-subject experimental design). So, the default ("A") webpage might have the shopping cart icon in the lower right corner, while the alternative webpage ("B") has the icon on the top right corner. The test then is to see which placement results in higher buying rates. However, when faced with hundreds of hypotheses to test (our questions), A/B testing is prohibitively expensive, as it requires too many

trials. Contextual bandits, on the other hand, integrate the learning algorithm (e.g., our relational deep learning method (Meng et al., 2018)) with a dynamic question-asking mechanism. This approach allows us to have a large pool of potentially useful questions, but dynamically predict the few questions that better help our model estimate outcomes (and, similarly, adapt the pool of questions for different organizations). In online advertisement, this boils down to showing online ads that will more likely result in purchases. In news organizations, this amounts to showing front-page news items that are more likely to be clicked. In our setting, this will result in deciding which questions we should ask. Our contextual bandit approach will use off-the-shelf Natural Language Processing tools to learn a model that translates the question sentences into the improvement in accuracy obtained when the question is asked.

Machine learning models are powerful but also data-hungry, so we will need a large set of signals. Generating this set of signals is the primary challenge of this work. We identified these inputs using three different approaches: (1) identifying the factors underlying the real reasons for failures, (2) using systems archetypes to identify dysfunctional cases of local rationality, and (3) using cognitive biases to identify potentially irrational and destructive actions.

IDENTIFICATION AND DEVELOPMENT OF INPUT SIGNALS

To create the crowd signals, we considered multiple sources of literature. We included factors that previous research found related to project failures and team performance, systems archetypes to capture dysfunctional team practices, cognitive biases to identify potentially irrational and destructive individual actions, and a separate category of questions we suspect may be indirectly related to failure. We have so far used an initial set of 49 questions to collect the crowd signals (provided currently by the students, in the future by employees). Of the 21 failure causes identified by Sorenson and Marais (2016), we considered in this work the ten that apply to student projects, as shown in **Error! Not a valid bookmark self-reference.**. In the future, when we work with organizations, we will expand the set to include all 22 failure causes.

Table 1: The ten failure causes we consider in this work. Adapted from Sorenson and Marais, 2016.

Systems engineering failure causes	*Definition*
Failed to consider design aspect	Actor(s) in the organization failed to consider an aspect in the system design. In many cases, this causal action describes a design flaw, such as a single-point failure or component compatibility.
Used inadequate justification	Actor(s) in the organization used inadequate justification for a decision.
Failed to form a contingency plan	Actor(s) in the organization failed to form a contingency plan to implement if an unplanned event occurred.
Lacked experience	Actor(s)' lack of experience or knowledge led to the failure.
Kept poor records	Actor(s) in the organization did not review documentation or other work sufficiently to capture errors and deficiencies.

Systems engineering failure causes	*Definition*
Inadequately communicated	Actor(s) in the organization failed to communicate with each other such that personnel were confused with the information they were given, had to "fill in the gaps" in the information they were given, or not notified about important information at all.
Subjected to inadequate testing	One or more actors in the organization subjected a component or subsystem to inadequate testing. This causal action captures inadequate tests as well as adequate tests performed inadequately.
Managed risk poorly	Actor(s) in the organization failed to identify, assess, formulate, or implement a proper mitigation measure.
Violated procedures	Actor(s) in the organization violated a procedure pertaining to the system, such as a maintenance or operation procedure.
Did not allow system aspect to stabilize	Actor(s) in the organization did not allow a system aspect like personnel, design, or requirements to stabilize before moving forward with an action.

The crowd signals collect human-centric information (e.g., actions, behaviors, and habits) during the project that we know or suspect correlate to individual or team performance and therefore to project failures and failure causes. To arrive at a successful set of crowd signals, we start by surveying literature that includes factors that affect team, project, and individual performance. Each factor then leads to one or more student questions that applies specifically to the specialized context of student projects. When possible, we phrase the questions so they are hard to game, meaning they do not have obvious "correct" answers.

We note that we use the literature as a guide to define an initial set of questions and our questions are just one way of identifying the presence of a corresponding factor. For an initial set of factors, we included a wide range of literature from the following research areas in the search: human factors, systems engineering, project management, engineering education literature, psychology, and social sciences. Then, we used the definitions of ten common cognitive biases to capture individual actions that may contribute to failure and four systems archetypes that correlate with poor safety practices. Finally, we also included nine indirect questions that we suspected relate to how well a project is performing and how people perform while on a project, but have not been studied in previous research work. In summary, we have created questions from the following eight categories:

1. **9 *Performance* questions (Q1–Q9):** Factors that relate to team performance and/or project success, as identified by human factors, engineering education, and systems engineering literature
2. **5 *Critical Success Factors* questions (Q10–Q14):** Critical success factors that appear in successful projects as identified from project management literature
3. **5 *Individual Personality* questions (Q15–Q19)**: Individual personality characteristics that affect team performance as identified from social sciences and psychology literature
4. **6 *Student Estimation* questions (Q20-Q25)**: Student estimation of the project status
5. **4 *Safety Archetypes* questions (Q26–Q29)**: Organizational safety archetypes that relate to dysfunctional team practices that may lead to failures

6. **9 *Indirect signals* questions (Q30–Q38)**: Indirect phenomena or habits that may relate to project outcome
7. **2 *Risk Perception* questions (Q39–Q40)**: Current risk perception of the team members may relate to current project status
8. **9 *Individual Actions & Decisions* questions (Q41–Q49)**: Cognitive biases of the team members that may show as tendency to particular actions or decisions

To demonstrate our process, we describe two examples of hard to game questions. *Proactivity* is a factor that is associated with project performance, because proactive people are willing to take action to affect their environment, in contrast to non-proactive people who are less likely to take action (Kirkman and Rosen, 1999). Rather than asking students directly whether they think they are proactive (where the answer would most likely be "yes"), we ask "During the past week, how many times did you attempt to get involved with a project-related task that was outside your immediate responsibility?" (Q2).

The *bandwagon effect* is a cognitive bias where people do or believe things because many other people do or believe the same. Rather than asking members directly whether everyone does or believes the same, we ask "During the past week, did you have any arguments with your team about the next project actions/tasks?" (Q42). We note that our question is just one way of identifying the presence of the bandwagon effect.

Appendix C includes the complete list of all crowd signals together with definitions from literature.

Based on the types of possible answers, the questions provide different types of data: some are categorical (e.g., Q42: whether an individual had arguments or not), some are expressed on Likert-scale (e.g., Q13: ranging availability of resources from very low availability to very high availability), some are continuous percentages (e.g., Q21: the confidence in the project spending estimate), and some are integers (e.g., Q1: experience in number of previous projects).). So, each question requires a different coding before we use it as a predictor variable for our models. We give a detailed description of the coding scheme for all crowd signals when we describe our regression models in a following section.

Since we so far collect data from student projects, we use the instructors to collect data related to project performance. In an industry application, similar data can be collected from management or the enterprise software. The instructors provide answers to a total of 14 questions, as shown in Table 2. Three of the questions indicate failure in terms of three project metrics: budget, schedule, and technical performance. Ten questions indicate whether a student team showed signs of any of the ten failure causes we considered. Lastly, there is one question to rate each team's productivity for the week.

Table 2: The questions to the instructors. Three questions relate to observed project failures and ten questions relate to failure causes.

	Project Failures	
I1	Budget status	What is currently true about the project budget, compared to what you initially planned? (*Multiple choice: Under/On/Over budget*)
I2	Schedule status	What is currently true about the project schedule, compared to what you initially planned? (*Multiple choice: Ahead of/On/Behind schedule*)
I3	Technical requirements status	What is currently true about meeting the technical requirements for the project, compared to what you initially planned? (*Multiple choice: Meeting fewer/as planned/more requirements*)
	Productivity	
I4	Productivity	Rate each team's productivity. (*Likert scale answer: Not productive at all (1) to Extremely productive (5)*)
	***Failure causes [Sorenson and Marais, 2016]* (*Binary choice for each team: Occurrence/Not occurrence*)**	
I5	Indicate whether a team "Failed to consider an aspect in the system design" this past week.	
I6	Indicate whether a team "Made a decision or action that was not well justified" this past week.	
I7	Indicate whether a team "Did not consider redundant components or measures for their actions" this past week.	
I8	Indicate whether a team "Made a mistake because members lack experience" this past week.	
I9	Indicate whether a team "Did not properly document their progress" this past week.	
I10	Indicate whether a team "Run into communication issues" this past week.	
I11	Indicate whether a team "Did not run adequate tests for their equipment" this past week.	
I12	Indicate whether a team "Managed risk poorly" this past week.	
I13	Indicate whether a team "Violated rules or procedures" this past week.	
I14	Indicate whether a team "Rushed into action without fully understanding the impacts to the system" this past week.	

Regarding the budget and schedule metrics from the instructors, we classify any project that is not progressing as planned as a failure for that metric in the given week since there is a divergence from the initial project plan. Regarding the technical requirements metric, if a team is satisfying fewer requirements than planned, we consider it a failure.

We want to predict the three types of project failures (budget, schedule, technical requirements) and the ten failure causes with the models. We include the productivity of each team provided by the instructor (I4) as a predictor of the failures and failure causes, together with the crowd signals from the students. The next section includes a summary of the instructor responses, highlighting how the projects performed in terms of failure.

COLLECTED DATA FROM STUDENT DESIGN TEAMS

To investigate whether our approach of predicting future failures and failure causes from crowd signals is viable, we tested our prototype in student projects at Purdue University. Purdue offers many opportunities for students via the technical curriculum, to participate in smaller-scale engineering projects that include designing, manufacturing, testing, or operating engineering equipment. Through these phases, students are exposed to real-life situations and work in teams to solve problems, while making weekly progress with meetings and sessions. Others have also used student courses or teams to obtain useful information about how engineers work. For example, Bohlman and Bahill (2013) used data from systems engineering courses to identify mental mistakes engineers make while creating tradeoff studies. This approach offers several advantages: we have ready access to such teams, there are several potential teams with which we can work which reduces the risk of not getting enough data, and the teams are doing "real engineering."

We asked the students to answer our questions every week and their responses (the crowd signals) are the main inputs to train the logistic regression models. The course instructors provided three qualitative metrics of the status of each project, whether they witnessed signs of a failure cause in a given week, and information on how productive each team was. We want to predict the occurrence of a project failure with respect to a specific metric and the occurrence of a failure cause. We assume that the instructors' opinion on how the projects are performing is as close to the truth as we can get, since they are the primary stakeholders and closely monitor the student projects.

We collected data from 28 design project teams. Twelve of the projects provided data for 12 weeks (fall semester), six for 6 weeks (summer semester), and ten projects for 6 weeks (spring semester). Four of the projects continued across both the summer and fall semesters and six of the projects continued across both the fall and spring semesters, but the composition of the student teams was different. Defining an observation as a set of answers from a team member or instructor, we collected 240 observations from the two instructors (who responded each week), and 304 out of 750 possible observations from the 74 students (who were less reliable in responding).

As mentioned earlier, the instructors provided information on how productive each team was in a given week, how the projects were performing in terms of the three metrics (budget, schedule, and satisfying technical requirements), and whether they noticed any of the ten failure causes.

Figure 3, Figure 4, and Figure 5 summarize the data we collected from the instructors. Of the 6 projects that started in the first semester of our data collection, deviance from budget was the least frequent failure, only occurring for one of the projects for weeks 3, 4, and 6. Meeting the technical objectives was a frequent problem with 1 or 2 projects failing every week. Of the 12 projects we observed during the second semester of our data collection, remaining on schedule was the most challenging aspect for the projects. The projects during this second period appear to have performed a lot worse in terms of all metrics compared to the first collection period , and they also were closer to the statistics of real industry projects from the Project Management Institute. During the fall semester the teams were larger because of more students, and perhaps the instructors also had higher expectations. One of the courses we collected data from had more deadlines that the students had to meet during the fall semester, because the experiments needed to be ready for competitions or payload launches during the year. During the summer, student work is usually continued based on progress from previous semester and may be not as technically challenging as during a regular semester.

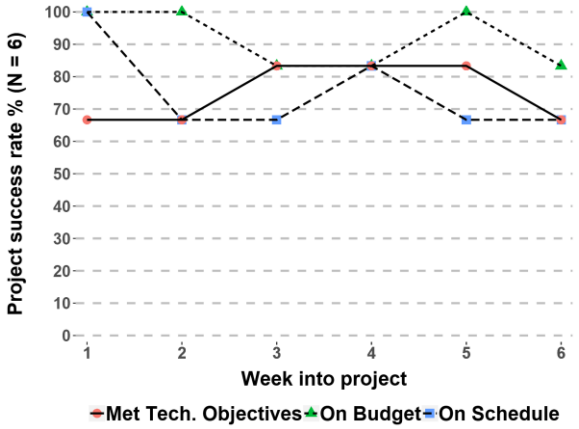

Figure 3: Success rates of the 6 projects for the first semester of data collection. Budget failure was the least frequent problem, while missing technical objectives occurred every week.

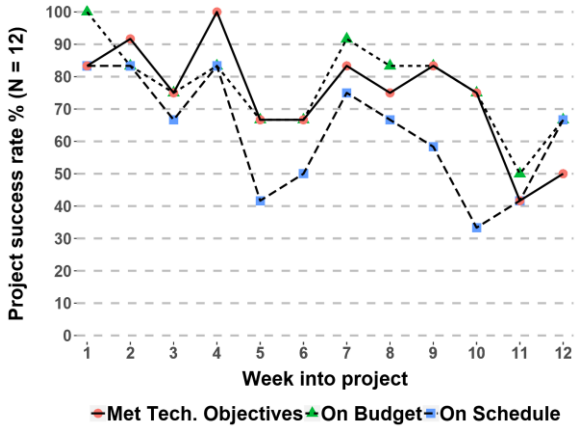

Figure 4: Success rates of the 12 projects for the second semester of data collection. Remaining on schedule was the biggest challenge for the student teams.

We also want to identify which of the ten failure causes were more frequent, and to do so, we define a detection measure $D_{i,Tot}$ for each of them. The detection measure is one way to communicate how often the instructors found evidence of failure cause i occurring in a student project. We first calculate a detection rate $D_{i,j}$ per project, taking into account how long the project lasted.

Detection rate of failure cause *i* for project *j*

$$D_{i,j} = \frac{\sum_{k=1}^{T_j} TRUE_{i,j,k}}{T_j} \ (\%) \qquad (1)$$

Where *i* is one of the ten failure causes, *j* is one of the 28 projects, T_j is the number of weeks project *j* lasted for, and $TRUE_{i,j,k}$ is a binary variable that is equal to 1 if failure cause *i* occurred for project *j* during week *k*, or 0 if not.

With the detection rate $D_{i,j}$ for each failure cause and each project, we can calculate the total detection measure $D_{i,Tot}$ for each failure cause, by averaging across all projects:

Detection measure of failure cause *i*

$$D_{i,Tot} = 100\frac{\sum_{j=1}^{n} D_{i,j}}{n}\ (\%) \quad (2)$$

Where $D_{i,Tot}$ is the detection measure of failure cause *i*, averaged across all *n = 28* projects.

Figure 5 shows the detection measure (in percentage) for all failure causes. Based on the instructor responses and the aforementioned calculations, the most frequently detected failure causes were students lacking experience, using inadequate justification for their decisions, failing to consider a design aspect for their project, and running into communication issues. The least frequently detected failure causes were violating rules or procedures and failing to form contingency plans. The values of the detection measures may appear low, but some of these failure causes may be hard for the instructors to detect (e.g., students making a mistake and covering it up), or are not large issues in student projects (e.g., finding quick contingency plans may be easier to do in the context of a student project compared to an industry project).

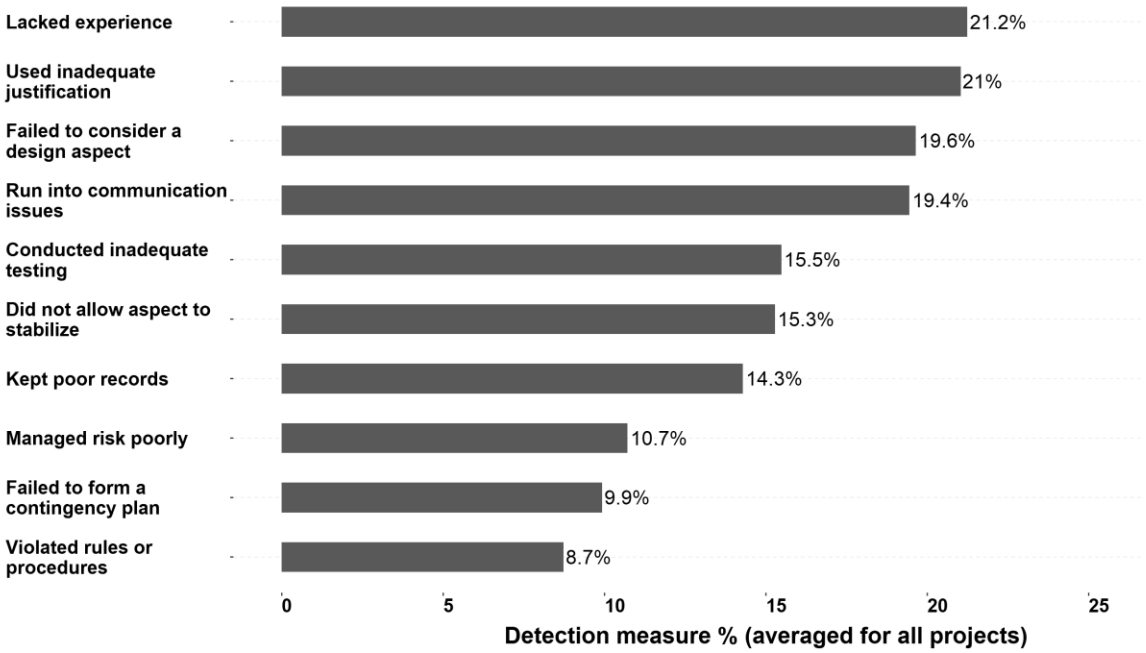

Figure 5: Detection measures in percentage (averaged across all projects) for the ten failure causes we considered in this work. Some of the failure causes may not be easy for the instructors to detect or have knowledge of.

SIGNALS THAT CORRELATE TO OCCURRENCES OF PROJECT FAILURES

This section describes our process to identify which of the crowd signals correlate with an increased likelihood of a failure or failure cause occurring during a project. We built mixed effects logistic regression models that correlate the crowd signals with the occurrence of project failures. We used mixed effects regression here because we collect data over time and we need to account for subject non-independence, as suggested by Harrison et al. (2018). The first part of this section

covers our process to setup the regression models and the coding schemes for the training data set. The second and third part show the resulting models and how we interpret the coefficients to identify the crowd signals that correlate to the occurrences of failures. The fourth and last part includes a cross validation process to find how well the regression models can predict the correct failure outcomes.

REGRESSION MODEL SETUP

To accomplish our research goal of predicting future failures and failure causes, we use the following information from the current project week *t*:

1. the individual student responses to the 49 questions (crowd signals),
2. the current state of the project from the instructor, and
3. the current productivity of the team from the instructor

as the independent predictors $X_{i,t}$ of a failure or failure cause *j* for the next week. The dependent variable $Y_{j,t+1}$ is binary such that when a failure or failure cause occurs, $Y_{j,t+1} = 1$, and when they do not, $Y_{j,t+1} = 0$. We train logistic regression models to predict the probability of occurrence, $\hat{p}(Y_{j,t+1} = 1)$, for each of the three possible failures (budget, schedule, and technical performance). Based on the observations, the model learns a function that maps the predictors $X_{i,t}$ to a predicted binary outcome $\hat{Y}_{j,t+1}$. Table 3 summarizes the variables for the three failure prediction models.

Table 3: Predictors and dependent variables for failure prediction. We build three models (one for each failure: budget, schedule, and technical requirements), from 47 predictors.

Independent variables (predictors) $X_{i,t}$ at week t			
(1)	Crowd signals	$X_{1-45,t}$	Come from the student responses to Q1–Q49. Depending on the metric, we use 2 out of 6 questions from the *Student Estimation* category, the two that apply to that particular metric. Therefore, there are a total of 45 predictors from the crowd signals.
(2)	Current state of the project	$X_{46,t}$	Comes from the instructors' response to I1–I3, depending on the metric we are predicting.
(3)	Productivity of the team	$X_{47,t}$	Comes from the instructors' response to I4.
Dependent variable $Y_{j,t+1}$ at week t+1			

Predicting failure in terms of metric *j* at week *t+1*	$Y_{j,t+1}$	Comes from the instructors' response to I1–I4 from the following week. *j = 1* corresponds to the budget metric *j = 2* corresponds to the schedule metric *j = 3* corresponds to the technical performance metric

For these classification problems (i.e., binary dependent variables: occurrence or not), we use logistic regression. When using regression, we must consider and attend to some of the logistic regression assumptions. The data from the crowd signals is in panel form, that is, it includes repeated measurements from the same individuals over time. Regression models are built on the assumption that observations are independent, which does not hold here, as the responses from the same individual at different times are not independent. One common way to account for non-independence of panel observations in linear models is to include random effects (Harrison et al. 2018). Random effects account for non-independence of the multiple responses coming from a single subject and allow estimation of variance between different subjects. With random effects, each person has their own intercept term in the model. Models with random effects assume that uncontrolled person-specific effects (e.g., age or gender) are not correlated with the predictors. If this assumption is not true, the parameter estimation will include omitted variable bias, and the model will not be reliable. We tested the assumption using Mundlak's auxiliary regression approach (Mundlak 1978, Schmidheiny 2011). There is no evidence of correlation between the time-invariant unobservable variables and the predictors. The random effects take a different value for each student *i* and appear in our model as c_i. Based on these considerations, our models will be of the following form when predicting the three types of failure ($j = 1, 2, 3$):

$$\hat{Y}_{j,t+1} = \log\left(\frac{\hat{p}_{j,t+1}}{1-\hat{p}_{j,t+1}}\right) = a + bX_{i,t}{}^{T} + c_i + \varepsilon_{it} \qquad (3)$$

Where a is the intercept constant, b is a column vector of slopes for each predictor, $X_{\mathrm{i},t}{}^{T}$ is a row vector of the 47 predictors at week t, $c_{\mathrm{i}} \sim N(0, \sigma_i{}^2)$ are individual random effects, and $\varepsilon_{it} \sim N(0, \sigma_\varepsilon{}^2)$ is the observation-specific random error.

Since all the predictors $X_{i,t}$ come from either instructor or student responses, we code them before building the models, depending on the types of data they include. Table 4 summarizes the coding process for all the predictors.

Table 4: Coding schemes for instructor and student responses, dependent on data type.

Data type	*Applicable questions*	*Coding scheme*
5-point full Likert scale	Q3, Q4, Q5, Q7, Q11–Q19, Q30, I4	Coded as integer 1–5 for each level
Integer	Q1, Q2, Q35, Q36, Q38	Not coded, treated as integer
Continuous percentage	Q6, Q8, Q10, Q21, Q23, Q25, Q34, Q49	Not coded, treated as continuous
Categorical	Q20, Q22, Q24, Q26–Q29, Q31–Q33, Q37, Q39–Q48, I1–I3, I5–I14	Coded as categorical (using one-hot encoding)
Character multiple answer	Q9	First, we calculate a sum where adjectives associated with creative designs count as +1, and their opposite adjectives as -1. The sum value is then in the range: $-6 \leq \sum adj \leq 6$. We then code the sum as categorical with the balanced scheme: $Creativity = \begin{cases} \text{Low}, \sum adj \leq -2, \\ \text{High}, \sum adj \geq 2 \\ \text{Moderate}, otherwise \end{cases}$

INTERPRETATION OF REGRESSION MODELS

Table 5, Table 6, and Table 7 show the resulting models for the three project metrics. The coefficients of the predictor variables from the logistic regression models are interpreted in terms of the log-odds of the outcome (occurrence of failure in this case). For example, we interpret the coefficient for experience b_1=*-0.05* for the budget model as the expected change in the log-odds of having a budget failure for a one-unit increase in experience, while keeping all other predictors at fixed values. Equivalently, the odds ratio can be calculated by exponentiating the coefficient value to get 0.95 which means we expect to see about 5% decrease in the odds of having a budget failure, for a one-unit increase in experience, while keeping all other variables at fixed values. For the scope of this work, we do not focus on the numerical values of the change in odds, but rather resort to a qualitative interpretation of the coefficients based on their sign. For providing feedback (described in chapter 3), we have interest in the coefficients with p-values less than 0.1, meaning that there is enough evidence of their respective questions to be good predictors of failure. When interpreting the coefficients for categorical variables, the change in odds of a failure occurring is calculated from the reference category. For example, the positive coefficient

b_9 in the budget model that corresponds to "low creativity" indicates that while holding all other predictors at fixed values, the odds of a budget failure will increase when creativity is low compared to the reference category ("high creativity" in this case). Increasing log-odds (positive coefficient) in logistic regression implies increasing probability of failure.

The model that predicts budget failure indicates that increasing productivity (I4), increasing freedom to students (Q7), not having previous problems resurface (Q28), and going with their first idea (Q48) reduce the likelihood of a budget failure. In contrast, exercising more than 4 times a week (compared to 1-2 times), having a budget failure the previous week (I1), and showing preference towards a schedule failure (compared to a cost mishap) all increase the likelihood of a budget failure. The budget model shows zero variance in the random effects, which indicates that although we expect some variation between students, the extent of this variation can be fully described by the residual term ε_{it} alone. In this case, the mixed effects model performs, in terms of accuracy, close to the generalized linear model without random effects.

Table 5: Mixed-effects logistic regression model for prediction of budget failure.

Coefficient	*Estimate (error)*	*Coefficient*	*Estimate (error)*	*Coefficient*	*Estimate (error)*
a	-3.222 (2.383)	b_{26}(Q27 = No)	-0.798 (1.028)	b_{52}(Q40 = Schedule)	0.582 (0.598)
b_1(Q1)	-0.05 (0.223)	b_{27}(Q27 = Yes)	-0.913 (0.984)	**b_{53}(Q41 = No)**	**1.865 (1.116)^**
b_2(Q2)	-0.302 (0.392)	**b_{28}(Q28 = No)**	**-2.829 (1.151)***	**b_{54}(Q41 = Yes)**	**2.422 (1.246)^**
b_3(Q3)	0.313 (0.257)	**b_{29}(Q28 = Yes)**	**-2.027 (1.183)^**	b_{55}(Q42 = No)	-0.179 (1.283)
b_4(Q4)	-0.389 (0.264)	b_{30}(Q29 = No)	0.144 (1.231)	b_{56}(Q42 = Yes)	-0.095 (1.388)
b_5(Q5)	-0.126 (0.269)	b_{31}(Q29 = Yes)	1.743 (1.326)	b_{57}(Q43 = No)	1.137 (1.593)
b_6(Q6)	-0.093 (0.253)	b_{32}(30)	0.381 (0.266)	b_{58}(Q43 = Yes)	2.229 (1.595)
b_7(Q7)	**-0.733 (0.266)****	b_{33}(Q31 = 2-3h)	-0.229 (0.678)	b_{59}(Q44 = No)	-1.626 (1.695)
b_8(Q8)	-0.182 (0.275)	b_{34}(Q31 = 3-4h)	-0.545 (1.229)	b_{60}(Q44 = Yes)	-1.591 (1.718)
b_9(Q9 = Low)	2.264 (1.578)	**b_{35}(Q31 = <1h)**	**-1.163 (0.663)^**	b_{61}(Q45 = No)	-1.832 (2.291)
b_{10}(Q9 = Moderate)	0.612 (0.494)	b_{36}(Q31 = >4h)	-0.096 (1.096)	b_{62}(Q45 = Yes)	-2.677 (2.281)
b_{11}(Q10)	-0.428 (0.274)	b_{37}(Q32 = Dining hall)	1.039 (1.1)	b_{63}(Q46 = No)	1.31 (1.243)
b_{12}(Q11)	0.033 (0.244)	b_{38}(Q32 = Restaurants)	1.415 (1.127)	b_{64}(Q46 = Yes)	0.842 (1.226)
b_{13}(Q12)	0.053 (0.267)	b_{39}(Q32 = Home-prepared)	0.124 (0.914)	b_{65}(Q47 = No)	0.28 (1.752)
b_{14}(Q13)	0.014 (0.235)	b_{40}(Q33 = No)	0.499 (0.662)	b_{66}(Q47 = Yes)	-0.146 (1.755)
b_{15}(Q14)	0.361 (0.267)	b_{41}(Q33 = Some)	0.146 (0.644)	**b_{67}(Q48 = First thought)**	**-1.797 (0.908)***
b_{16}(Q15)	-0.361 (0.293)	b_{42}(Q34)	-0.325 (0.253)	b_{68}(Q48 = Think through)	-0.709 (0.73)
b_{17}(Q16)	-0.303 (0.252)	b_{43}(Q35)	-0.295 (0.29)	b_{69}(Q49)	0.346 (0.267)
b_{18}(Q17)	0.221 (0.268)	b_{44}(Q36)	0.037 (0.239)	**b_{70}(I1= Failure)**	**1.885 (0.577)****
b_{19}(Q18)	-0.296 (0.259)	b_{45}(Q37 = 3-4 times)	0.084 (0.725)	**b_{71}(I4)**	**-0.565 (0.256)***
b_{20}(Q19)	0.086 (0.258)	**b_{46}(Q37 =More than 4 times)**	**1.506 (0.765)***		
b_{21}(Q20 = Over budget)	**-3.313 (1.063)****	**b_{47}(Q37 = No)**	**1.179 (0.687)^**		
b_{22}(Q20 = Under budget)	-0.236 (0.563)	b_{48}(Q38)	-0.392 (0.269)		

Coefficient	*Estimate (error)*	*Coefficient*	*Estimate (error)*	*Coefficient*	*Estimate (error)*
b_{23}(Q21)	**0.625 (0.301)***	b_{49}(Q39 = Requirements)	0.361 (0.709)	^ p < .01 * p < .05 ** p < .01 *** p < .001	
b_{24}(Q26 = No)	0.91 (1.273)	**b_{50}(Q39 = Schedule)**	**1.638 (0.606)****		Random effects $c_i \sim N(0, 0.e^{-9^2})$
b_{25}(Q26 = Yes)	**2.298 (1.196)^**	b_{51}(Q40 = Requirements)	-0.654 (1.432)		

The model that predicts schedule failure indicates that increasing frequency of team members discussing matters about their lives (Q19) and turning down activities that they consider more fun (Q38) reduce the likelihood of a schedule failure. On the contrary, with increasing student confidence in their success without oversight (Q8), having previous problems resurface due to poor previous solutions (Q28), showing preference towards a schedule failure (compared to a cost mishap), having a schedule failure the previous week (I2), and having increasing confidence in the truthfulness of their responses (Q49) all increase the likelihood of a schedule failure. For Q46, both the "yes" and "no" answers correlate to higher likelihood of schedule failure compared to a scenario where the questions do not apply (i.e., the students did not respond or did not have to make any new decisions). Not learning anything new (Q46 = No) is the worse option. For Q48, again both "yes" and "no" options correlate to lower likelihood of a schedule failure compared to the "Do not apply" option. Teams that went with their first idea were less likely to have a schedule failure the following week, possibly because going with their first idea allowed teams to make progress rather than spending time thinking through different solutions. Similar to the budget model, the schedule model also shows zero variance in the random effects.

Table 6: Mixed-effects logistic regression model for prediction of schedule failure.

Coefficient	*Estimate (error)*	*Coefficient*	*Estimate (error)*	*Coefficient*	*Estimate (error)*
a	-0.957 (2.464)	b_{26}(Q27 = No)	0.58 (1.03)	b_{52}(Q40 = Schedule)	-0.82 (0.611)
b_1(Q1)	-0.064 (0.193)	b_{27}(Q27 = Yes)	1.055 (0.938)	b_{53}(Q41 = No)	-1.389 (0.885)
b_2(Q2)	0.006 (0.201)	b_{28}(Q28 = No)	1.405 (1.08)	b_{54}(Q41 = Yes)	-0.954 (0.949)
b_3(Q3)	0.164 (0.229)	**b_{29}(Q28 = Yes)**	**2.362 (1.14)***	b_{55}(Q42 = No)	-0.631 (1.249)
b_4(Q4)	-0.248 (0.235)	b_{30}(Q29 = No)	-1.631 (1.325)	b_{56}(Q42 = Yes)	-1.647 (1.268)
b_5(Q5)	-0.052 (0.216)	b_{31}(Q29 = Yes)	-1.046 (1.299)	b_{57}(Q43 = No)	-0.914 (1.322)
b_6(Q6)	-0.082 (0.214)	b_{32}(30)	0.173 (0.231)	b_{58}(Q43 = Yes)	-2.186 (1.329)
b_7(Q7)	-0.089 (0.222)	b_{33}(Q31 = 2-3h)	0.685 (0.601)	b_{59}(Q44 = No)	-1.508 (1.77)
b_8(Q8)	**0.715 (0.268)****	b_{34}(Q31 = 3-4h)	-0.049 (0.95)	b_{60}(Q44 = Yes)	-1.121 (1.764)
b_9(Q9 = Low)	1.206 (1.449)	b_{35}(Q31 = <1h)	0.12 (0.58)	b_{61}(Q45 = No)	0.718 (1.958)
b_{10}(Q9 = Moderate)	-0.021 (0.45)	b_{36}(Q31 = >4h)	1.49 (1.006)	b_{62}(Q45 = Yes)	1.38 (1.955)
b_{11}(Q10)	0.079 (0.22)	b_{37}(Q32 = Dining hall)	1.583 (1.193)	**b_{63}(Q46 = No)**	**4.177 (1.309)****
b_{12}(Q11)	0.038 (0.239)	b_{38}(Q32 = Restaurants)	0.263 (1.107)	**b_{64}(Q46 = Yes)**	**2.931 (1.217)***
b_{13}(Q12)	0.035 (0.233)	b_{39}(Q32 = Home-prepared)	1.569 (0.954)	b_{65}(Q47 = No)	-1.929 (1.37)
b_{14}(Q13)	0.23 (0.228)	b_{40}(Q33 = No)	0.114 (0.539)	b_{66}(Q47 = Yes)	-0.82 (1.333)
b_{15}(Q14)	0.226 (0.24)	b_{41}(Q33 = Some)	-0.553 (0.555)	**b_{67}(Q48 = First thought)**	**-2.905 (0.829)*****
b_{16}(Q15)	-0.423 (0.279)	b_{42}(Q34)	-0.192 (0.239)	**b_{68}(Q48 = Think through)**	**-1.923 (0.637)****

Coefficient	*Estimate (error)*	*Coefficient*	*Estimate (error)*	*Coefficient*	*Estimate (error)*
b_{17}(Q16)	**0.408 (0.215)^**	b_{43}(Q35)	-0.214 (0.252)	**b_{69}(Q49)**	**0.541 (0.258)***
b_{18}(Q17)	0.317 (0.263)	b_{44}(Q36)	0.14 (0.213)	**b_{70}(I2= Failure)**	**1.021 (0.469)***
b_{19}(Q18)	0.292 (0.236)	b_{45}(Q37 = 3-4 times)	0.08 (0.578)	b_{71}(I4)	-0.312 (0.226)
b_{20}(Q19)	**-0.493 (0.238)***	b_{46}(Q37 =More than 4 times)	-0.153 (0.628)		
b_{21}(Q22 = Behind)	-0.876 (0.967)	b_{47}(Q37 = No)	-0.293 (0.581)		
b_{22}(Q22 = On)	-0.543 (0.91)	**b_{48}(Q38)**	**-0.808 (0.27)****		
b_{23}(Q23)	0.112 (0.22)	b_{49}(Q39 = Requirements)	0.46 (0.644)	^ p < .01 * p < .05 ** p < .01 *** p < .001	
b_{24}(Q26 = No)	0.771 (0.967)	**b_{50}(Q39 = Schedule)**	**1.595 (0.595)****		Random effects $c_i \sim N(0, 0.e^{-9^2})$
b_{25}(Q26 = Yes)	0.941 (0.861)	b_{51}(Q40 = Requirements)	1.502 (1.038)		

The model that predicts failure regarding the technical requirements indicates that increasing frequency of students thinking they made meaningful progress (Q5), increasing number of project outputs (Q30), not discussing trivial matters (Q47), and increasing number of unscheduled team meetings outside regular class time (Q35) all reduce the likelihood of a failure regarding the technical requirements. In contrast, increasing frequency of students noticing a "silent room" (Q14), having students say they think they are satisfying fewer requirements (Q24, compared to saying they are doing as planned), not learning anything new (Q46), and having a requirements failure the previous week (I3) all increase the likelihood of a future requirements failure. For Q31, spending 2–3 or 3–4 hours on social media both correlate with increased likelihood of failure (compared to spending 1–2 hours). For Q28, again both the "yes" and "no" answers correlate to higher likelihood of requirements failure compared to a scenario where the questions do not apply. Having previous problems surface (Q28 = Yes) is the worse option.

Table 7: Mixed-effects logistic regression model for prediction of technical performance failure.

Coefficient	*Estimate (error)*	*Coefficient*	*Estimate (error)*	*Coefficient*	*Estimate (error)*
a	**-10.909 (3.74)****	**b_{26}(Q27 = No)**	**-3.343 (1.498)***	b_{52}(Q40 = Schedule)	0.281 (0.778)
b_1(Q1)	0.228 (0.274)	**b_{27}(Q27 = Yes)**	**-3.051 (1.335)***	b_{53}(Q41 = No)	-0.602 (1.168)
b_2(Q2)	0.432 (0.332)	**b_{28}(Q28 = No)**	**3.065 (1.545)***	b_{54}(Q41 = Yes)	-0.944 (1.271)
b_3(Q3)	0.457 (0.322)	**b_{29}(Q28 = Yes)**	**4.041 (1.782)***	b_{55}(Q42 = No)	1.915 (1.796)
b_4(Q4)	-0.359 (0.314)	b_{30}(Q29 = No)	-0.65 (1.635)	b_{56}(Q42 = Yes)	0.282 (1.675)
b_5(Q5)	**-0.819 (0.36)***	b_{31}(Q29 = Yes)	-0.421 (1.684)	b_{57}(Q43 = No)	-1.148 (1.921)
b_6(Q6)	-0.021 (0.276)	**b_{32}(30)**	**-0.653 (0.324)***	b_{58}(Q43 = Yes)	-3.165 (2.008)
b_7(Q7)	-0.284 (0.353)	**b_{33}(Q31 = 2-3h)**	**1.838 (0.831)***	b_{59}(Q44 = No)	1.707 (3.973)
b_8(Q8)	0.594 (0.373)	**b_{34}(Q31 = 3-4h)**	**3.174 (1.275)***	b_{60}(Q44 = Yes)	1.88 (4.033)
b_9(Q9 = Low)	2.336 (1.807)	b_{35}(Q31 = <1h)	1.072 (0.832)	b_{61}(Q45 = No)	4.809 (5.108)
b_{10}(Q9 = Moderate)	-0.283 (0.597)	b_{36}(Q31 = >4h)	0.668 (1.267)	b_{62}(Q45 = Yes)	6.19 (5.178)
b_{11}(Q10)	0.017 (0.277)	b_{37}(Q32 = Dining hall)	1.009 (1.294)	**b_{63}(Q46 = No)**	**3.985 (1.717)***
b_{12}(Q11)	-0.079 (0.299)	b_{38}(Q32 = Restaurants)	2.067 (1.527)	b_{64}(Q46 = Yes)	1.416 (1.614)
b_{13}(Q12)	-0.213 (0.3)	**b_{39}(Q32 = Home-prepared)**	**1.851 (1.112)^**	**b_{65}(Q47 = No)**	**-4.223 (1.801)***
b_{14}(Q13)	-0.283 (0.297)	b_{40}(Q33 = No)	-0.44 (0.743)	b_{66}(Q47 = Yes)	-2.43 (1.828)

Coefficient	*Estimate (error)*	*Coefficient*	*Estimate (error)*	*Coefficient*	*Estimate (error)*
$\boldsymbol{b_{15}}$**(Q14)**	**0.843 (0.339)***	b_{41}(Q33 = Some)	-0.9 (0.98)	b_{67}(Q48 = First thought)	-1.276 (0.964)
$\boldsymbol{b_{16}}$**(Q15)**	**-0.751 (0.391)^**	b_{42}(Q34)	-0.016 (0.312)	b_{68}(Q48 = Think through)	-0.843 (0.789)
b_{17}(Q16)	0.514 (0.318)	$\boldsymbol{b_{43}}$**(Q35)**	**-0.776 (0.365)***	b_{69}(Q49)	-0.306 (0.363)
$\boldsymbol{b_{18}}$**(Q17)**	**0.694 (0.377)^**	b_{44}(Q36)	-0.537 (0.433)	$\boldsymbol{b_{70}}$**(I3= Failure)**	**2.586 (0.725)*****
b_{19}(Q18)	-0.338 (0.317)	b_{45}(Q37 = 3-4 times)	-0.428 (0.757)	b_{71}(I4)	-0.411 (0.284)
b_{20}(Q19)	-0.393 (0.349)	b_{46}(Q37 =More than 4 times)	0.155 (0.837)		
$\boldsymbol{b_{21}}$**(Q24=Fewer)**	**1.681 (0.815)***	b_{47}(Q37 = No)	-1.182 (0.843)		
b_{22}(Q24=More)	0.83 (0.794)	b_{48}(Q38)	-0.392 (0.348)		
b_{23}(Q25)	0.667 (0.424)	b_{49}(Q39 = Requirements)	0.147 (0.776)	^ $p < .01$	
b_{24}(Q26 = No)	1.017 (1.313)	b_{50}(Q39 = Schedule)	-0.083 (0.685)	* $p < .05$	Random effects $c_i \sim N(0, 0.483^2)$
$\boldsymbol{b_{25}}$**(Q26 = Yes)**	**2.637 (1.296)***	b_{51}(Q40 = Requirements)	-2.307 (1.658)	** $p < .01$ *** $p < .001$	

REGRESSION MODEL VALIDATION

To investigate the ability of our models to make accurate predictions of failure outcomes, we used *k*-cross validation (Arlot and Celisse, 2010) with *k = 10* folds. Cross-validation is a technique to evaluate the ability of the model to generalize, that is, make accurate predictions from unknown data. To complete the validation process, we split the dataset into 10 folds. We use 9 of the folds as the training set to build a logistic regression model, we run the model using the last fold as the testing set, and record the number of correct outcome predictions in that last fold. We repeat the process 10 times, having all folds be the testing set, as shown in Figure 6.

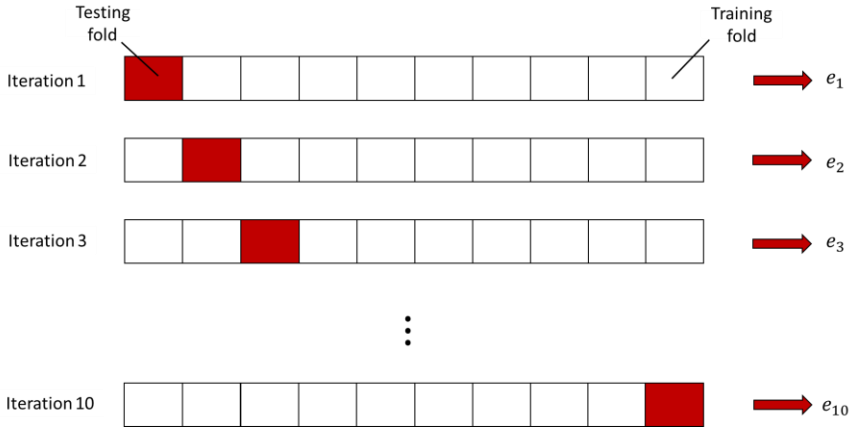

Figure 6: 10-fold cross validation process. Our dataset is split in 10 folds of equal data points. At each iteration, we use 9 folds as the training set for the logistic regression model and then run the model on the remaining fold (testing fold). We record how many correct predictions the algorithm correctly identified in the testing fold. We repeat the process for all folds.

Because we know the true outcomes from our dataset, we evaluate an accuracy measure e_i in each iteration. If the model returns a predicted probability of failure occurring greater than 50%,

then we classify that as a failure. For each of the training folds, we can create a confusion matrix with the predicted and actual outcomes (Table 8):

Table 8: Generic confusion matrix for logistic regression models.

	Predicted: Failure	*Predicted: Not failure*
Actual: Failure	n_1	n_2
Actual: Not failure	n_3	n_4

The accuracy measure is the ratio of correct outcomes identified by the model in the particular testing set, over the total outcomes:

$$e_i = \frac{n_{correct}}{n_{total}} = \frac{n_1 + n_4}{\sum_{i=1}^{4} n_i} \tag{4}$$

Figure 7 shows the results of the model validation process, that is, the percentage of correct predictions for each model and iteration. The budget and schedule models predict correctly, on average, 54% of outcomes. The technical requirements model predicts 60% of outcomes correctly on average.

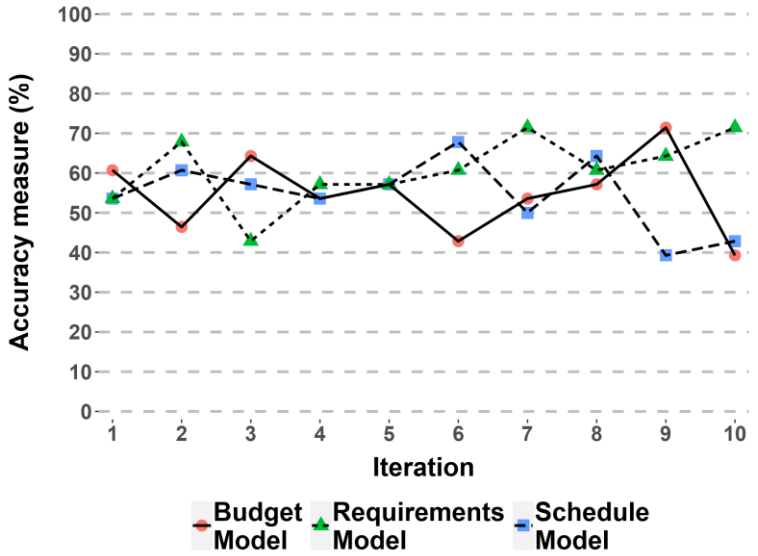

Figure 7: All three failure prediction models correctly predicted between 40–70% of outcomes of unknown data. Logistic regression is a classification approach with many assumptions, and we expect more advanced methods to perform better.

DEEP LEARNING FOR FAILURE PREDICTION

In the previous sections we considered the logistic regression model with random effects. Logistic regression belongs to a larger class of generalized linear models. We also discussed the assumptions behind the mixed effects logistic regression model used in our previous analysis.

Here we revisit this discussion as it is important in understanding the role of our new machine learning methods.

The easiest way to identify the internal assumptions of any statistical model is to analyze the equation it optimizes to obtain its model parameters. In our random effects model, each student i appears with a variable effect c_i assumed to be normally distributed. Given training data $\{(Y_{j,t+1}, X_{i,t})\}_{j=1,2,3;t=1,\ldots,t_{\max};i=1,\ldots,i_{\max}}$ for all three types of failure $j = 1,2,3$, all students $i = 1, \ldots, i_{\max}$, and all weeks $t = 1, \ldots, t_{max}$, our mixed effects logistic regression model finds parameters $\hat{a}, \hat{\boldsymbol{b}}, \hat{\boldsymbol{c}}$ that maximize the likelihood:

$$\hat{a}, \hat{\boldsymbol{b}}, \hat{\boldsymbol{c}} = \text{argmax}_{a,\boldsymbol{b},\boldsymbol{c}} \prod_{t=1}^{t_{\max}-1} \prod_{i=1}^{i_{\max}} P(Y_{j,t+1,i} = 1 | X_{i,t}; a, \boldsymbol{b}, c_i) \quad (5)$$

where $\boldsymbol{c}$ is a vector of the students' mixed effects (c_i is an element of the vector), $Y_{j,t+1,i}$ is the j-th binary failure signal (given by the instructor) at week $t+1$ for the team of student i, and argmax returns the set of parameters that maximize the function, with

$$P(Y_{j,t+1,i} = 1 | X_{i,t}; a, \boldsymbol{b}, c_i) \propto \exp\left(a + \boldsymbol{b} {X_{i,t}}^T + c_i\right) \quad (6)$$

where $\propto$ is the symbol for *proportional to*, used to simplify the notation since the probabilities needs to sum to one, i.e.,

$$P(Y_{j,t+1,i} = 1 | X_{i,t}; a, \boldsymbol{b}, c_i) + P(Y_{j,t+1,i} = 0 | X_{i,t}; a, \boldsymbol{b}, c_i) = 1 \quad (7)$$

Any multiplication in the likelihood function comes from an implicit independence assumption between the variables. Note that

$$P(Y_{j,t+1,i} = 1 | X_{i,t}; a, \boldsymbol{b}, c_i) \propto \exp(a) \exp\left(\boldsymbol{b} {X_{i,t}}^T\right) \exp(c_i) \quad (8)$$

which means that for each student, each of the factors in $X_{i,t}$ are also assumed independent.

These independence assumptions needed for the random effects model can weaken its predictive performance. For instance, the responses of students in the same team are likely dependent. The responses to different questions by the same student are also correlated, but the random effects model can only capture very weak correlations between them. Such correlations are captured through parameter c_i, an average of the questions' linear interactions. These and other independences in the random effects model create odd situations, such as predicting that student i will fail but student j will succeed, even though students i and j are in the same team --- i.e., they should fail and succeed together. In what follows we develop a significantly more powerful model, albeit less interpretable.

*What is the **most expressive** statistical model for our problem?* Here we are looking to extract all possible dependencies between the variables (linear and non-linear dependencies). Our approach, Janossy Pooling, (Murphy et al., 2019) recently published at the International Conference for Learning Representations (ICLR), provides the **first** machine learning method that is *probably* able to express any dependencies between the variables in a team of respondents.

In order to understand our approach, we first need to introduce the concept of *exchangeability*. Let S_k be set of students in the *k*-th team. Consider jointly predicting the failure metrics $j = 1,2,3$ of team k at week t:

$$P^*(Y_{1,t+1,k} = y_{1,t+1,k}, Y_{2,t+1,k} = y_{2,t+1,k}, Y_{3,t+1,k} = y_{3,t+1,k} | \{X_{i,t}\}_{i \in S_k}) \tag{9}$$

where the conditional is over the set of answers of all students $\{X_{i,t}\}_{i \in S_k}$. Note that the above equation makes no other independence assumptions except independence over time. Unfortunately, P^* in the above equation is not a proper mathematical formulation of the probability function. A probability function cannot take sets of answers as inputs, only ordered variables. The mathematically consistent way to define P^* is then though symmetric functions, say, if $S_k = \{\text{Alice,Bob}\}$, then

$$\begin{aligned} P^*(Y_{1,t+1,k} = y_{1,t+1,k}, Y_{2,t+1,k} = y_{2,t+1,k}, Y_{3,t+1,k} = y_{3,t+1,k} | X_{\text{Alice},t}, X_{\text{Bob},t}) = P^*(Y_{1,t+1,k} \\ = y_{1,t+1,k}, Y_{2,t+1,k} = y_{2,t+1,k}, Y_{3,t+1,k} = y_{3,t+1,k} | X_{\text{Bob},t}, X_{\text{Alice},t}) \end{aligned} \tag{10}$$

Now, P^* is a proper probability function. For larger teams of students, P^* must give the same probability for all possible permutation of the students. Hierarchical Bayesian models will fake these dependencies by evoking conditional independencies, but these are still fundamentally underpowered (Diaconis 1977).

Janossy pooling is a representation learning method that relies on the power of deep neural networks and on the flexibility of stochastic optimization methods to achieve its goals. Pooling is a fundamental operation in deep learning architectures (Le Cun et al., 2015). The role of pooling is to merge a collection of related features into a single, possibly vector-valued, summary feature. A prototypical example is in convolutional neural networks (CNNs) (Le Cun et al., 1995), where linear activations of features in neighborhoods of image locations are pooled together to construct more abstract features. A more modern example is in neural networks for graphs, where each layer pools together embeddings of neighbors of a vertex to form a new embedding for that vertex, see for instance, (Kipf & Welling, 2016; Duvenaud et al., 2015; Xu et al., 2019).

A common requirement of a pooling operator is invariance to the ordering of the input features. In CNNs for images, pooling allows invariance to translations and rotations, while for graphs, it allows invariance to graph isomorphisms. Existing pooling operators are mostly limited to pre-defined heuristics such as max-pool, min-pool, sum, or average. Another desirable characteristic of pooling layers is the ability to take variable-size inputs, such as a variable-size number of team

members. Our goal is to design flexible and learnable pooling operators satisfying these two desiderata.

Abstractly, we will view pooling as a permutation-invariant (a.k.a. symmetric) function acting on finite but arbitrary length sequences $(X_{i,t})_{i\in S_k}$. All elements $X_{i,t}$ of the sequences are features lying in some space $\mathcal{H}$ (which itself could be a high-dimensional Euclidean space $\mathbb{R}^d$ or some subset thereof, $d \geq 1$). The sequences $(X_{i,t})_{i\in S_k}$ are themselves elements of the union of products of the $\mathcal{H}$-space: $(X_{i,t})_i \in \bigcup_{m=0}^{\infty} \mathcal{H}^m$.

The Janossy pooling function $\bar{\bar{f}}$ is a permutation-insensitive function that starts with a permutation-sensitive function $\vec{f}$, parameterized by θ, which can take any variable-size sequence as input (from $\bigcup_{m=0}^{\infty} \mathcal{H}^m$) and outputs a real-valued vector in $\mathbb{R}^d$. In practice, we implement $\vec{f}$ with a neural network. The Janossy pooling function $\bar{\bar{f}}$ is an average of the value of $\vec{f}$ evaluated over all permutations of the input, that is,

$$\bar{\bar{f}}\left(\left(X_{i,t}\right)_{i\in S_k}; \theta\right) = \frac{1}{|S_k|!}\sum_{\pi\in\Pi} \vec{f}\left(\left(X_{i,t}\right)_{i\in\pi(S_k)}; \theta\right) \tag{11}$$

where $\pi(S_k)$ is a permutation of the students of team S_k and Π is the set of all such permutations. The output of $\bar{\bar{f}}$ is then a real-valued vector in $\mathbb{R}^d$.

Assume we know the optimal parameters $\theta^\star$ that define $\bar{\bar{f}}$ in order to best predict the outputs

$$P^*(Y_{1,t+1,k} = y_{1,t+1,k}, Y_{2,t+1,k} = y_{2,t+1,k}, Y_{3,t+1,k} = y_{3,t+1,k} | \bar{\bar{f}}\left(\left(X_{i,t}\right)_{i\in S_k}; \theta^\star\right)) \tag{12}$$

and because $\bar{\bar{f}}$ is by definition permutation-insensitive, P^* is a proper probability function. We can now define a learnable permutation-sensitive function using another neural network to obtain P^*, as detailed next.

Once all the neural networks are connected into a deep neural network, we can perform what is known as end-to-end learning,

$$\begin{aligned}\widehat{\boldsymbol{W}}, \hat{\theta} = argmax_{\boldsymbol{W},\theta} \prod_{t=1}^{t_{max}-1}\prod_{k=1}^{k_{max}} P^*(Y_{1,t+1,k} = y_{1,t+1,k}, Y_{2,t+1,k} = y_{2,t+1,k}, Y_{3,t+1,k} \\ = y_{3,t+1,k} | \bar{\bar{f}}\left(\left(X_{i,t}\right)_{i\in S_k}; \theta\right); \boldsymbol{W})\end{aligned} \tag{13}$$

where $P^*[;\boldsymbol{W}]$ is a multilayer perceptron parameterized by $\boldsymbol{W}$. If the number of students in a team is large, we can use a stochastic optimization approach, detailed in our work (Murphy et al., 2019), denoted π-SGD.

THE NEURAL NETWORK ARCHITECTURE

In our experiments, P^* is a multilayer perceptron with one hidden layer, while $\bar{\bar{f}}$ is a unary Janossy pooling tractable approximation (see Murphy et al., (2019) for details) with an extra multilayer perceptron at the output to recover some of the representation power lost by the approximation.

We chose this simpler approximation procedure due to the limited amount of data of our experiments. Having more data allows a more expressive version of $\bar{\bar{f}}$.

Our neural network architecture first creates a representation of all questions of a given student i, denoted

$$\tilde{X}_{i,t} = \sigma\left(\boldsymbol{W_1} X_{i,t}^T\right)^T \tag{14}$$

where $X_{i,t} = \left(x_{i,t,m}\right)_{m=1}^{M}$ are the answers to the M=43 questions from student i at week t, $\boldsymbol{W_1} \in \mathbb{R}^{d \times M}$, where $d \geq 1$ is the number of neurons in the hidden layer and $\sigma(.)$ is the sigmoid function applied element-wise (applied to every element of the resulting vector). Then, the overall hidden representation of each question m is given by our Janossy pooling function

$$h_t^{(m)} = \bar{\bar{f}}\left(\left(\tilde{X}_{i,t}\right)_{i \in S_k}, \left(x_{i,t,m}\right)_{i \in S_k}; \theta_m\right) \tag{15}$$

whose output is then fed into a single-layer feedforward network. The unary Janossy pooling with respect to question m is given by:

$$\bar{\bar{f}}\left(\left(\tilde{X}_{i,t}\right)_{i \in S_k}, \left(x_{i,t,m}\right)_{i \in S_k}; \theta_m\right) = \sigma(\boldsymbol{W}_2^{(\mathrm{m})}(\sum_{\mathrm{i} \in \mathrm{S_k}} \sigma(\boldsymbol{W_1}^{(\mathrm{m})} \mathrm{CONCAT}\left(x_{i,t,m} \tilde{X}_{i,t}\right)^T))^T)^T \tag{16}$$

where $\theta_m = \boldsymbol{W}_2^{(m)} \in \mathbb{R}^{d_m \times d}$, $\boldsymbol{W_1}^{(m)} \in \mathbb{R}^{d \times (d + l_m)}$ such that d is the hidden layer dimension, d_m is the question m embedding dimension and $CONCAT(.)$ is a function that concatenates the input vectors. Finally, concatenated, the question embeddings compose an embedding for group k in week t

$$h_{k,t} = \mathrm{CONCAT}(\mathrm{h}_{i,t}^{(1)}, \ldots, \mathrm{h}_{i,t}^{(M)}) \tag{17}$$

This embedding, or hidden representation is now used to optimize the network weights in an end-to-end supervised learning task (Goodfellow et. al 2016, pp. 200) for jointly predicting the three failure types $y_{1,t+1,k}, y_{2,t+1,k}, y_{3,t+1,k}$, that is, we try to maximize the joint probability

$$P^*(Y_{1,t+1,k} = y_{1,t+1,k}, Y_{2,t+1,k} = y_{2,t+1,k}, Y_{3,t+1,k} = y_{3,t+1,k} | h_{k,t}) \tag{18}$$

The above model is also known as multi-task learning (Zhang et. al 2018). A single-task learning setting was used in our mixed effects logistic model, where it learns the marginal probability

$$P^*(Y_{j,t+1,k} = y_{j,t+1,k}|h_{k,t}) \tag{19}$$

for each $j = 1,2,3$, task, *i.e.* we have three different models, one for each prediction task.

Since a powerful-enough neural network can learn to marginalize the predictions, transforming a multi-task architecture into a single-task architecture, the multi-task setting is indeed more powerful than the single-task one. However, in practice we can observe difficulties in training multi-task models, due to noise imbalance in the classes (Kendall et. al 2018), *i.e.* a subset of tasks can have more noise (harder to learn) and dominate the loss during training. Thus, we need to test the effectiveness of both approaches.

The neural network takes advantage of multi-task learning when the prediction tasks (different failures) are correlated. Say, if we were to predict the intensity of raining and the delays in traffic in a given day. Clearly, learning whether it rains helps to decide the amount of traffic. Having a single model allows *knowledge sharing* between the two random variables and the representation learned by the neural network. In our setting, since we are modeling different types of failure, they are also likely related and, thus, multi-task learning was expected to show more accurate predictions. And indeed, our empirical results show that multi-task learning tends to give more accurate predictions than learning the tasks independently.

We illustrate the final neural network architecture in Figure 8.

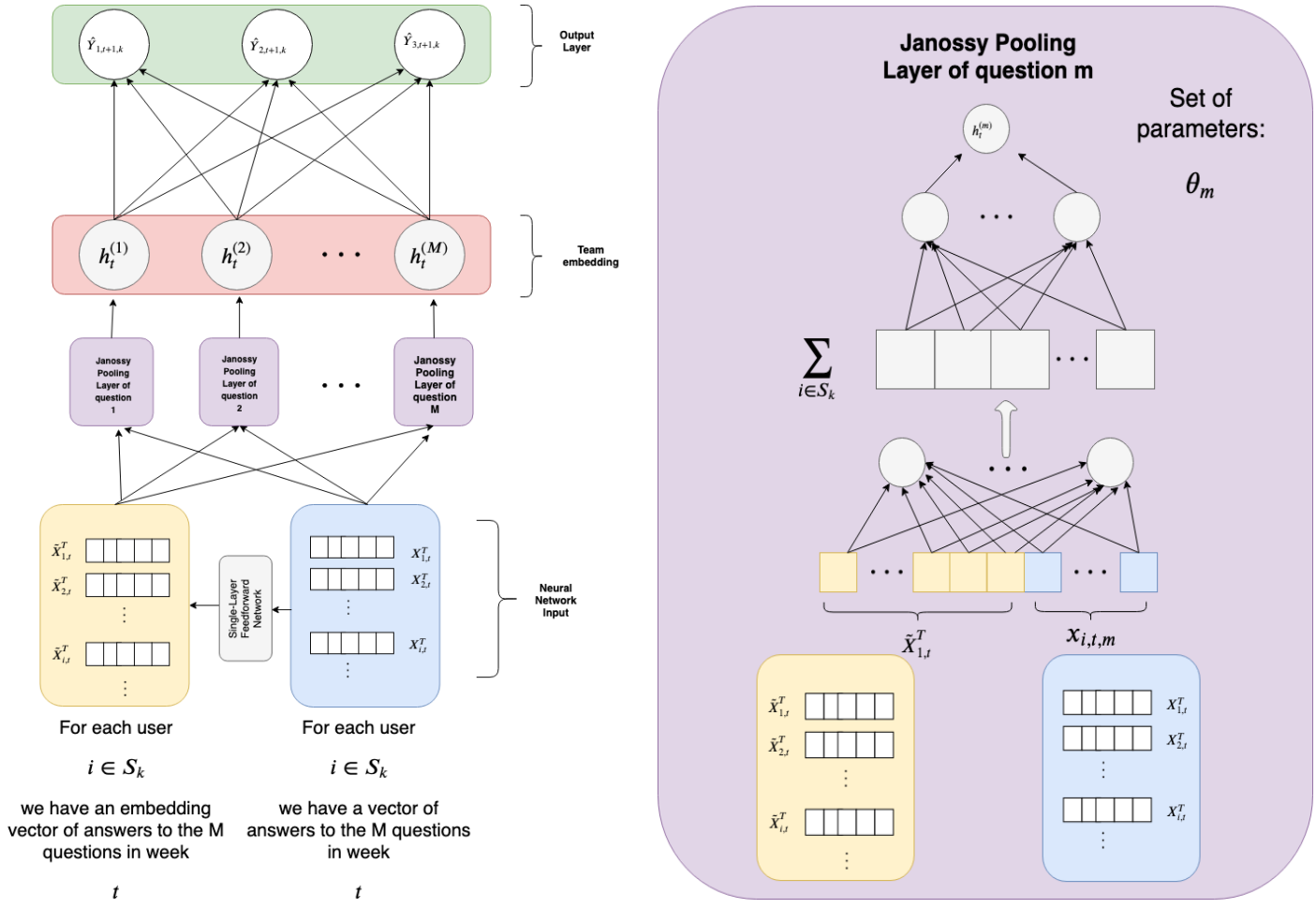

Figure 8: On the right, we show the Janossy pooling layer for an specific question m, note how the full architecture on the left uses M different layers of this type (each with its own set of parameters θ_m). On the left, we show the whole architecture, we omit details from standard layers such as a simple fully connected neural network (denoted by MLP). The central idea is to learn a team representation through a Janossy pooling representation that is then used in a multi-task neural network for supervised learning (learning knowing the outputs).

RESULTS

In what follows, we evaluate the proposed neural network architecture in the dataset collected during this project and described in the first three sections. Here, we consider two different multi-task settings, one over project failures with 3 types of failures and another over failure causes with 10 types of failure causes. The dataset we use here contains 174 examples spanning all teams introduced in Section 3. The number of failures (or failure causes) across all the 13 predicted variables vary from 0.133 to 0.373.

Project Failures. First, we look at predicting how a team will perform next week in terms of budget, schedule and technical objectives, as three binary random variables, each encoding whether the team is on budget or not, on schedule or not, and meeting the planned objectives

or not. We will refer to this set of tasks as project failures tasks. For this set of tasks, we use the label from the previous week together with the embedding $h_{k,t}$ in the last layer.

Failure Causes. Second, we look at predicting the ten different types of failure causes described in Table 1, as ten different binary random variables, each encoding whether or not the failure cause was detected in the team during the following week. We refer to this set of tasks as failure causes.

In order to evaluate our model, we consider two multi-task learning settings, one where we learn jointly the project failures and one where we learn jointly the failure causes. Alternatively, we learn a marginal model (single-task) for each binary random variable in both settings to compare its predictive performance against multi-task learning. We also want to verify whether our pooling operator and overall architecture improves upon an off-the-shelf logistic regression model. Unlike our previous logistic regression, we will make it a team prediction model by adding the binary answers of all students in the team. The standard logistic regression uses a single-task setting. We evaluate these models in with 5-fold cross validation, where we use three folds to train, one to validate and one to test. Please, refer to the next section for implementation details.

Table 9 and Table 10 show the prediction accuracy of the different models. We note how the simple logistic regression model does worse than our proposed architecture (in single-task) in all but one task and is never better than the multi-task version of our proposed neural network model. This shows how although the neural network model is more complex, it is able to capture the necessary dependencies in the data and generalize well to the test data. These results validate our hypothesis that failure causes and project failures can present dependencies, since the multi-task setting often gives better predictions than the single-task models in most tasks.

Table 9: Mean accuracy and standard deviation of models in project failure tasks in a 5-fold cross validation. In bold, we show the model which achieved the highest mean accuracy

Project Failure	*Logistic Regression (Single Task)*	*Our Model (Single Task)*	*Our Model (Multi-task)*
Budget	0.642 ± 0.080	0.689 ± 0.09	$\mathbf{0.729 \pm 0.068}$
Schedule	0.523 ± 0.072	$\mathbf{0.586 \pm 0.068}$	0.580 ± 0.041
Technical Requirements	0.580 ± 0.062	0.643 ± 0.035	$\mathbf{0.688 \pm 0.088}$

Table 10: Mean accuracy and standard deviation of models in failure causes tasks in a 5-fold cross validation. In bold, we show the model which achieved the highest mean accuracy .

Failure Causes	*Logistic Regression (Single Task)*	*Our Model (Single Task)*	*Our Model (Multi-task)*
Failed to consider design aspect	0.597 ± 0.092	0.787 ± 0.054	$\mathbf{0.793 \pm 0.062}$
Used inadequate justification	0.568 ± 0.018	$\mathbf{0.735 \pm 0.041}$	0.724 ± 0.051
Failed to form a contingency plan	0.684 ± 0.058	0.821 ± 0.073	$\mathbf{0.867 \pm 0.043}$
Lacked experience	0.671 ± 0.073	0.769 ± 0.054	$\mathbf{0.804 \pm 0.049}$
Kept poor records	0.717 ± 0.056	0.781 ± 0.046	$\mathbf{0.838 \pm 0.053}$
Inadequately communicated	0.585 ± 0.056	0.729 ± 0.067	$\mathbf{0.735 \pm 0.059}$
Subjected to inadequate testing	0.666 ± 0.068	0.827 ± 0.056	$\mathbf{0.844 \pm 0.093}$
Managed risk poorly	0.833 ± 0.070	0.781 ± 0.049	$\mathbf{0.862 \pm 0.055}$
Violated procedures	0.787 ± 0.047	0.868 ± 0.052	$\mathbf{0.885 \pm 0.057}$
Did not allow system aspect to stabilize	0.666 ± 0.066	0.810 ± 0.049	$\mathbf{0.827 \pm 0.047}$

IMPLEMENTATION DETAILS

We set the last hidden layer to *M* neurons, that is, the number of questions. The dimension of every other hidden layer is set to 16. We train the model for 200 epochs with Adam. The early stopping strategy was to pick the best model in the validation data over the 200 training epochs. The best learning rate found was 0.01 and no weight decay (L2 regularization) used. The logistic regression we uses the default implementation from the Sklearn (Pedregosa et. al 2007) package.

CONCLUSION AND FUTURE WORK

During our first year we completed five peer-reviewed conference publications and are in the process of writing three journal articles. Our first publication will appear at the ASEE annual conference in June 2019 and was selected as the best overall PIC paper (PICs are groups of research divisions in the conference). The paper showcases parts of our statistical analyses that identify which *Crowd inputs* correlate with ten common causes of systems engineering failures. Our second publication will appear at the INCOSE annual symposium in July 2019 and includes a preliminary study where we found that the concept of Wisdom-of-the-Crowd consistently applied when estimating cost risk of a project (i.e., processing the team members' opinions about the project cost as a crowd provides better estimates than processing their opinions individually). Our third publication is that of Janossy pooling, which uses neural networks to give the most-expressive representation of sets for machine learning tasks, published at the 2019 International Conference for Learning Representations (ICLR), our forth publication is an extension of Janossy pooling to graphs and tensor inputs, published at the 2019 International Conference on Machine Learning (ICML), and the fifth publication considers the problem of extending Janossy pooling to represent sets of sets, published at the 2019 ACM SIGKDD Conference.

With the completion of the first year, we have shown that our approach works in student teams and we now have enough knowledge to move forward with an industrial application of our prototype. For the second year of support, we would like to partner with industry or other (e.g., DoD, NGO) organizations. We will start by engaging potential collaborators at the conferences where we present our research, use contacts from our school's Industrial Advisory Council, and contact other organizations independently. We propose to improve on our approach in two ways: 1) by adding the *Company inputs* in our models and 2) by testing our prototype on industry projects. Organizations collect a wide range of information (e.g., financial, product, and market data) that could further improve the prediction capabilities of our prototype. In the first part of our planned future work, we will expand the input data set, to reflect the broader range of data available. Then, we will deploy our app in our partner organization(s) to collect *Company* and *Crowd inputs*, based on which we will predict failures at the organization(s). Because we use a specific organization's data for our process and training of the predictive models, our prototype is tailored to that particular organization.

Our work over the second year, when completed, can provide value to our industry partners and to the Department of Defense. To our industry partners we will give value by leveraging the predictive capability of our prototype. Our process gives us the opportunity to provide feedback to decision makers, alerting them of upcoming failures, and suggesting corrective actions. Through our analyses we know which human actions (captured by the *Crowd inputs)* correlate to the occurrences of failures (e.g., employees not being proactive). Similarly, via the *Company inputs* we will know which organizational processes lead to failure (e.g., low marketing budget). With such knowledge, our feedback is targeted, suggesting changes to specific human behaviors or company processes. Also, our process is tailored to the particular organization as we will use their own data to train our predictive models.

After the second year of our effort, we will have completed testing of our prototype and collected valuable feedback from the testing phase at organizations. In the long term, our plan is to arrive at a final product: a complete risk assessment prototype that takes daily inputs and predicts the occurrences of failures, suggesting corrective actions. The idea that we use an organization-specific product instead of a generic framework may make our idea more easily marketable. We know from historical data and literature that defense programs often face challenges to be on time, on budget, and satisfy technical requirements as planned. Our final product may assist in reducing occurrences of such failures in large defense projects, which would increase the value of engineered systems.

APPENDIX A: RESULTING PUBLICATIONS FROM THE RESEARCH GRANT

1 Georgalis, G and Marais, K 2019, "Can we use Wisdom-of-the-Crowd to Assess Risk of Systems Engineering Failures?" *INCOSE 2019 International Symposium*, Orlando, FL, July 2019.

2 Georgalis, G and Marais, K 2019, "Assessment of Project-Based Learning Courses using Crowd Signals." *ASEE 2019 Annual Conference & Exposition*, Tampa, FL, June 2019. Selected as the ASEE Best Overall PIC Paper

3 Murphy, R, Srinivasan, B, Rao, V and Ribeiro, B 2019, "Janossy pooling: Learning deep permutation-invariant functions for variable-size inputs," International Conference on Learning Representations (ICLR), 2019, New Orleans, LA, May 2019.

4 Murphy, R, Srinivasan, B, Rao, V and Ribeiro, B 2019, "Relational Pooling for Graph Representations," *International Conference on Machine Learning (ICML*), Long Beach, CA, June 2019.

5 Meng, C, Yang, J, Ribeiro, B and Neville, J 2019, "HATS: A Hierarchical Sequence-Attention Framework for Inductive Set-of-Sets Embeddings" *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Anchorage, AK, August 2019.

APPENDIX B: CITED REFERENCES

Arlot, S and Celisse, A 2010, 'A survey of cross-validation procedures for model selection', *Statistics surveys*, Vol. 4, pp. 40-79.

Baybutt, P 2018, 'The validity of engineering judgment and expert opinion in hazard and risk analysis: The influence of cognitive biases', *Process Safety Progress*, Vol. 37, No. 2, pp. 205-210.

Bloem-Reddy, B and Teh, YW 2019, 'Probabilistic symmetry and invariant neural networks', *arXiv preprint arXiv:1901.06082*.

Bohlman, J and Bahill, AT 2013, 'Examples of mental mistakes made by systems engineers while creating tradeoff studies', Studies in Engineering and Technology, Vol. 1, No. 1, pp. 22–43.

Charette, RN 2008, "What's wrong with weapons acquisitions?", *IEEE Spectrum*, Vol. 45, No. 11, pp. 33-39.

Chua, DKH, Kog, YC and Loh, PK 1999, 'Critical success factors for different project objectives', *Journal of construction engineering and management*, Vol. 125, No. 3, pp. 142-150.

Cordery, JL, Morrison, D, Wright, BM and Wall, TD 2010, 'The impact of autonomy and task uncertainty on team performance: A longitudinal field study', *Journal of Organizational Behavior*, Vol. 31, No. 2–3, pp. 240-258.

Diaconis, P 1977, 'Finite forms of de Finetti's theorem on exchangeability', *Synthese*, Vol. 36, No.2, pp. 271-281.

Dietz, T 2017, 'Drivers of human stress on the environment in the twenty-first century', *Annual Review of Environment and Resources*, Vol. 42, pp.189-213.

Dorst, K and Cross, N 2001, 'Creativity in the design process: co-evolution of problem–solution', *Design studies*, Vol. 22, No. 5, pp. 425-437.

GAO 2017, 'Defense Acquisitions; Assessments of Selected Weapon Programs', United States Government Accountability Office, Washington, D.C.

Gilson, LL, Mathieu, JE, Shalley, CE & Ruddy, TM 2005, 'Creativity and standardization: complementary or conflicting drivers of team effectiveness?', *Academy of Management journal*, Vol. 48, No. 3, pp. 521-531.

Hamilton, BH, Nickerson, JA and Owan, H 2003, 'Team incentives and worker heterogeneity: An empirical analysis of the impact of teams on productivity and participation', *Journal of political Economy*, Vol. 111, No. 3, pp. 465-497.

Harrison, XA, Donaldson, L, Correa-Cano, ME, Evans, J, Fisher, DN, Goodwin, CE, Robinson, BS, Hodgson, DJ and Inger, R 2018, 'A brief introduction to mixed effects modelling and multi-model inference in ecology', *PeerJ*, Vol. 6, p. e4794, viewed 13 November 2018, <https://peerj.com/articles/4794>.

Judge, TA and Bono, JE 2000, 'Five-factor model of personality and transformational leadership', *Journal of applied psychology*, Vol. 85, No. 5, pp. 751.

Kendall, A. Gal, Y and Cipolla, R 2018, 'Multi-task learning using uncertainty to weigh losses for scene geometry and semantics', *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* pp. 7482-7491.

Kipf, TN and Welling, M 2016, 'Semi-supervised classification with graph convolutional networks', *International Conference on Learning Representations,* April 24-26 2017, Toulon,

France.
Kirkman, BL & Rosen, B 1999, 'Beyond self-management: Antecedents and consequences of team empowerment', *Academy of Management journal*, Vol. 42, No. 1, pp. 58-74.
Lehner, P, Seyed-Solorforough, MM, O'Connor, MF, Sak, S and Mullin, T 1997, 'Cognitive biases and time stress in team decision making', *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, Vol. 27, No. 5, pp. 698-703.
Lineberger, RS and Hussain, A 2016, "Program management in aerospace and defense: Still late and over budget", *Deloitte Development LLC*.
Marais, K, Saleh, JH and Leveson, NG 2006, 'Archetypes for organizational safety', *Safety Science*, Vol. 44, No. 7, pp. 565-582.
Meng, J, Chandra Mouli, S, Ribeiro, B, and Neville, J 2018, "Predicting Subgraph Evolution in Heterogeneous Dynamic Networks", *AAAI Conference on Artificial Intelligence*.
Montibeller, G and Von Winterfeldt, D 2015, 'Cognitive and motivational biases in decision and risk analysis. *Risk Analysis*, Vol. 35, No. 7, pp. 1230-1251.
Mundlak, Y 1978, 'On the pooling of time series and cross section data', *Econometrica: Journal of the Econometric Society*, Vol. 46, No. 1, pp. 69-85.
Murphy, RL, Srinivasan, B, Rao, V and Ribeiro, B 2018,' Janossy pooling: Learning deep permutation-invariant functions for variable-size inputs, *International Conference on Learning Representations,* May 6-9 2019, New Orleans, LA.
Nolan, A, Pickard, AC, Nolan, J, Beasley, R and Pruitt, TC 2018, 'How Many Systems Engineers Does It Take To Change a Light Bulb?', *In INCOSE International Symposium*, Vol. 28, No. 1, pp. 777-790.
Pedregosa, F, Varoquaux, G, Gramfort, A, Michel, V, Thirion, B, Grisel, O, Blondel, M, Prettenhofer, P, Weiss, R, Dubourg, V and Vanderplas, J 2011, 'Scikit-learn: Machine learning in Python', *Journal of machine learning research*, Vol. 12, pp. 2825-2830.
Peeters, MA, Van Tuijl, HF, Rutte, CG and Reymen, IM 2006, 'Personality and team performance: a meta-analysis', *European Journal of Personality: Published for the European Association of Personality Psychology*, Vol. 20, No. 5, pp. 377-396.
Pinto, JK and Slevin, DP 1987, 'Critical factors in successful project implementation', *IEEE transactions on engineering management*, Vol. 1, pp. 22-27.
Reagans, R, Argote, L and Brooks, D 2005, 'Individual experience and experience working together: Predicting learning rates from knowing who knows what and knowing how to work together', *Management Science*, Vol. 51, No. 6, pp. 869-881.
Rockenbach, B, Sadrieh, A and Mathauschek, B 2007, 'Teams take the better risks', *Journal of Economic Behavior & Organization*, Vol. 63, No. 3, pp. 412-422.
Salas, E, Cooke, NJ and Rosen, MA 2008, 'On teams, teamwork, and team performance: Discoveries and developments', *Human factors*, Vol. 50, No. 3, pp. 540-547.
Schmidheiny, K 2016 'Panel data: fixed and random effects', *Short Guides to Microeconometrics*, University of Basel, viewed 13 November 2018.
Sjöberg, L 2000, 'Factors in risk perception', *Risk analysis*, Vol. 20, No. 1, pp. 1-12.
Sorenson, D and Marais, K 2016, 'Patterns of causation in accidents and other systems engineering failures', In IEEE Systems Conference (SysCon), *Annual IEEE* (pp. 1-8).
van Mierlo, H, Rutte, CV, Vermunt, JK, Kompier, MAJ and Doorewaard, JAMC 2006, 'Individual autonomy in work teams: The role of team autonomy, self-efficacy, and social support',

European Journal of Work and Organizational Psychology, Vol. 15, No. 3, pp. 281-299.
Vîrgă, D, CurŞeu, PL, Maricuţoiu, L, Sava, FA, Macsinga, I and Măgurean, S 2014, 'Personality, relationship conflict, and teamwork-related mental models', *PloS one*, Vol. 9, No. 11, pp.e110223.
Xu, K, Hu, W, Leskovec, J and Jegelka, S 2019, 'How Powerful are Graph Neural Networks?', *International Conference on Learning Representations,* New Orleans, LA, May 6-9 2019.

APPENDIX C: COMPLETE LIST OF CROWD INPUT SIGNALS

Table 11 shows the complete list of the 49 crowd signals and their definitions from literature when applicable, organized by category.

Table 11: The questions that collect the crowd signals from the students. Each question is generated based on the definitions of corresponding literature.

		Performance	
Q1	Individual Experience	The level of proficiency of employees as well as the collective ability to exchange knowledge [Reagans et al. 2005].	How many engineering projects have you participated in so far? Include all engineering projects from coursework, internships, or extracurricular activities. (*Integer answer*)
Q2	Proactivity	Proactive individuals show initiative, are willing to take action and affect their environment, and show perseverance [Kirkman and Rosen 1999].	During the past week, how many times did you attempt to get involved with a project-related task that was outside your immediate responsibility? (*Integer answer*)
Q3	Stress level	High level of stress is associated with increased anxiety, negative emotions, distraction, conflict, and loss of team orientation [Dietz et al. 2017].	During the past week, how often were you unable to focus on this project? (*Likert scale answer: Never (1) to Always (5)*)
Q4	Coordination (1)	The unification, integration, synchronization of the efforts of group members to provide unity of action in the pursuit of common goals [Salas et al. 2008].	During the past week, how often did you interact with your team members while completing separate project tasks? (*Likert scale answer: Never (1) to Always (5)*)
Q5	Team Impact	Teams have been shown to impact the productivity and performance of a project [Hamilton et al. 2003].	During this past week, how often did you think that your team made progress that was meaningful for the success of this project? (*Likert scale answer: Never (1) to Always (5)*)
Q6	Coordination (2)	The unification, integration, synchronization of the efforts of group members to provide unity of action in the pursuit of common goals [Salas et al. 2008].	During the past week, for roughly what percentage of your team do you know exactly what they worked on? (*Continuous percentage answer*)
Q7	Standardized work	Standardized work practices detail how work should be performed [Gilson et al. 2005].	During the past week, rate the level of freedom you felt you had on how to complete your project tasks. (*Likert scale answer: No freedom (1) to Complete freedom (5)*)

Q8	Team Autonomy	High team autonomy has been linked to increased productivity, quality of performance, innovativeness, job satisfaction, decreased turnover, and fewer accidents [van Mierlo et al. 2006, Cordery et al. 2010].	Assume that the course instructor is unavailable for the remaining of the semester. What do you think is the chance your team will successfully complete all the assigned tasks without any oversight for the rest of the semester? (C*ontinuous percentage answer*)
Q9	Creativity	Teams that explore alternative ways to accomplish their work also should be better able to meet the needs of their customers [Dorst and Cross 2001, Gilson et al. 2005].	During the past week, which of the following attributes/adjectives relating to creativity do you feel apply to your team's project work? (M*ultiple answer between 6 adjectives that relate to creativity and 6 that do not*)
CSF (Critical Success Factors) [Pinto and Slevin 1987, Chua et al. 1999]			
Q10	Modularization	Modular design, or "modularity in design", is a design approach that subdivides a system into smaller parts called modules or skids, that can be independently created and then used in different systems.	During the past week, roughly what percentage of the tasks you performed could be done independently of the rest of the project? (*Continuous percentage answer*)
Q11	Clear objectives	To have effective tasks, then it is important to plan and pen clearly defined objectives that can deliver desired results.	During the past week, how clearly defined were your team's objectives? (*Likert scale answer: Not clear at all (1) to Completely clear (5)*)
Q12	Commitment	The state of being dedicated to a cause.	If your team announced to you today that they all quit, would you be willing to continue working on the project with a completely new team? (*Likert scale answer: Definitely not (1) to Definitely yes (5)*)
Q13	Availability of resources	Availability means capable of being used or the extent to which resources are available to meet the project's needs.	During the past week, rate your team's availability of resources (tools/space/software/funds) for you to use. (*Likert scale answer: Very low availability (1) to Very high availability (5)*)
Q14	Communication	Communication is the act of conveying intended meanings from one entity or group to another through the use of mutually understood signs and semiotic rules.	During the past week, how often did you notice a "silent room" while you were working with your team? (*Likert scale answer: Never (1) to Always (5)*)
Individual Personality [Judge and Bono 2000, Vîrgă, D et al. 2014, Peeters et al. 2006]			

Q15	Neuroticism	Neurotic individuals are associated with low emotional stability, experience frustration, anxiety, depression, and negative emotions.	During the past week, how often did you feel frustrated by your team members or your team's performance? (*Likert scale answer: Never (1) to Always (5)*)
Q16	Openness to experience	Openness reflects the degree of intellectual curiosity, creativity and a preference for novelty and variety a person has.	During the past week, how often did you come up with or agree to a new idea for your project? (*Likert scale answer: Never (1) to Always (5)*)
Q17	Conscientiousness	Conscientiousness implies a desire to do a task well, and to take obligations to others seriously. Conscientious people tend to be efficient and organized as opposed to easy-going and disorderly.	During the past week, how often did you skip, delay, postpone, or cancel a task/activity/obligation you were required to do/attend? (*Likert scale answer: Never (1) to Always (5)*)
Q18	Extraversion	Indicates how outgoing and social a person is.	During the past week, how often did you find yourself being the center of the attention of your team? (*Likert scale answer: Never (1) to Always (5)*)
Q19	Agreeableness	Agreeableness manifests itself in individual behavioral characteristics that are perceived as kind, sympathetic, cooperative, warm, and considerate.	During the past week, how often did your team members share detailed about their life with you? (*Likert scale answer: Never (1) to Always (5)*)
		Student Estimation [Nolan et al. 2018]	
Q20	Project spending estimate	Students give a qualitative estimate of how much they are spending.	Which of the following reflects your current estimate about your project spending? (*Multiple choice: Under/Over/On budget*)
Q21	Confidence in project spending estimate	Confidence in the spending estimate	How confident are you in your estimate? (*Continuous percentage answer*)
Q22	Project timeline estimate	Students give a qualitative estimate of whether they are staying on schedule.	Which of the following reflects your current estimate about your project's timeline? (*Multiple choice: Ahead of/Behind/On schedule*)

Q23	Confidence in project timeline estimate	Confidence in the timeline estimate	How confident are you in your estimate? (*Continuous percentage answer*)
Q24	Project technical performance estimate	Students give a qualitative estimate of whether they are satisfying their requirements.	Which of the following reflects your current estimate about your project's technical performance? (*Multiple choice: satisfying fewer/more/as planned requirements*)
Q25	Confidence in project technical performance estimate	Confidence in technical performance estimate	How confident are you in your estimate? (*Continuous percentage answer*)
		Team Actions & Archetypes [Marais et al. 2006]	
Q26	Unintended side effects of fixes	Poorly thought out fixes may have unintended side effects.	If new problems occurred this week, do you think they were handled appropriately? (*Multiple choice: Yes/No/Does not apply*)
Q27	Stagnant risk management	When technological advances are not accompanied by concomitant understanding of the associated risks, risk may increase.	During the past week, did your team consider new potential risks as a result of any new project tasks or updates? (*Multiple choice: Yes/No/Does not apply*)
Q28	Fixing symptoms rather than root causes	Fixes to problems that only address the symptoms may worsen or prolong the original problem.	During the past week, were you disappointed because a problem that your team thought had been fixed, had instead continued or gotten worse? (*Multiple choice: Yes/No/Does not apply*)
Q29	The vicious cycle of bureaucracy	When organizations respond to problems with more rules and bureaucracy, employees may become apathetic or alienated.	During the past week, were you frustrated about any rule or bureaucracy that was out of your control? (*Multiple choice: Yes/No/Does not apply*)
		Indirect Signals	
Q30	Number of material outputs	An increase or decrease in hardcopy or electronic files may indicate how much progress the team is making and therefore relate to project performance.	During the past week, did you notice a change in project outputs (hardcopy documents, electronic files, scrap paper to sketch ideas etc.) from your team? (*Likert scale answer: Large decrease (1) to Large increase (5)*)

Q31	Social media engagement	Time spent on social media may be related to distracted individuals are while working on a project.	During the past week, how much time on average per day did you spend on social media platforms? (*Multiple choice: <1/ 1-2/ 2-3/ 3-4/ >4 hours*)
Q32	Eating habits(1)	Eating habits impact overall individual health and therefore may relate to how individuals perform.	During the past week, which of the following statements best describes your eating habits this week? (*Multiple choice: Fast food/ Restaurants/ Home/ Dining Halls*)
Q33	Eating habits(2)	Eating habits impact overall individual health and therefore may relate to how individuals perform.	During the past week, did you have breakfast before coming in for class? (*Multiple choice: No/Before some/ Before all class times*)
Q34	Time spent thinking the project	How long an individual spends thinking about the project may be correlated to much they contribute to the project.	During the past week, what percent of your working time did you spent thinking about this project or working on this project? (*Continuous percentage answer*)
Q35	Unscheduled team meetings	Unscheduled team meetings may indicate team effort to meet performance requirements during crunch times.	During the past week, how many times did you and other members of your team arrange to meet and work on the project outside the regular class time? (*Integer answer*)
Q36	New equipment	Ordering new supplies may be related to how a project is progressing and are related to project spend.	During the past week, how many items (tools/supplies/project equipment) did your team order? (*Integer answer*)
Q37	Exercising habits	Exercising habits are related to overall individual health and may be related to how individuals perform on a project.	During the past week, how often did you physically exercise? (*Multiple choice: 0/ 1-2/ 3-4/>4 times*)
Q38	Financial pressure	Financial pressure arises from any situation where money worries are causing stress, which may relate to lack of the individual's focus on a project.	During the past week, how often did you turn down a fun activity because you thought it was too expensive? (*Integer answer*)
		Risk perception [Sjöberg 1999, Rockenbach et al. 2007]	

Q39	Risk perception	Students rank three hypothetical scenarios from the one they consider the highest risk to the one they consider the lowest risk. The scenarios are related to a cost, schedule, or requirements mishap.	Which of the following events do you consider the highest risk for your project's overall success? (*Ranking between a cost/schedule/requirements risk*)
Q40	Outcome preference	In the scenario that a failure is bound to happen, students provide the one they think would have the lowest impact.	If you had to choose one of the following failures for your project at the end of the semester, which would have the lowest impact? (*Multiple choice cost/schedule/requirements failure*)
Individual Actions & Decisions [Lehner et al. 1997, Montibeller & Winterfeldt 2015, Baybutt 2018]			
Q41	Ambiguity effect	The tendency to avoid options for which missing information makes the probability seem "unknown".	During the past week, did you disagree with an idea or decision because you thought you did not understand all potential implications? (*Multiple choice: Yes/No/Does not apply*)
Q42	Bandwagon effect	Tendency to do or believe what others do or believe. As more people come to believe in something, others do too, regardless of the underlying evidence.	During the past week, did you have any arguments with your team about the next project actions/tasks? (*Multiple choice: Yes/No/Does not apply*)
Q43	Focusing effect	The tendency to place too much importance on one aspect of an event.	During the past week, can you single out one project decision by your team as the most important? (*Multiple choice: Yes/No/Does not apply*)
Q44	Normalcy bias	The refusal to plan for, or react to, a disaster which has never happened before.	During the past week, did you spend any time thinking about how things might go wrong for this project? (*Multiple choice: Yes/No/Does not apply*)
Q45	Not invented here	Aversion to contact with or use of products, research, standards, or knowledge developed outside a group.	During the past week, did you get any new ideas about your project from other teams or people? (*Multiple choice: Yes/No/Does not apply*)
Q46	Confirmation bias	The tendency to search for, interpret, focus on and remember information in a way that confirms one's preconceptions.	During the past week, did you learn any new things that surprised you, because of your involvement with this project? (*Multiple choice: Yes/No/Does not apply*)

Q47	Parkinson's Law of Triviality	The tendency to give disproportionate weight to trivial issues.	During the past week, did your team spend significant time discussing what you thought as trivial matters about the project? (*Multiple choice: Yes/No/Does not apply*)
Q48	Anchoring	The tendency to rely too heavily, or "anchor", on one trait or piece of information when making decisions (usually the first piece of information acquired on that subject).	For any new project decisions that you had to make this week, did you think through all viable solutions or go with the one that you thought of first? (*Multiple choice: Think through/First thought/Does not apply*)
Q49	Overconfidence effect	Excessive confidence in one's own answers to questions.	How confident do you feel about the accuracy of your answers to this questionnaire? (*Continuous percentage answer*)