Why Does Software Cost So Much? Initial Results from a Causal Search of Project Datasets

Michael Konrad
Software Engineering Institute
4500 Fifth Avenue
Pittsburgh, PA 15213
(412) 268-5813
mdk@sei.cmu.edu

Robert Stoddard
Software Engineering Institute
4500 Fifth Avenue
Pittsburgh, PA 15213
(412) 268-1121
rws@sei.cmu.edu

Sarah Sheard
Software Engineering Institute
4500 Fifth Avenue
Pittsburgh, PA 15213
(412) 268-7612
sheard@sei.cmu.edu

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM18-0550

Abstract

How can we achieve success with software-intensive systems? To control project outcomes we need to better understand which factors truly drive those outcomes versus those merely correlated with them. In this paper, we will share early results from the application of causal modeling tools to evaluate several potential causes of late delivery, cost overruns, and technical performance gap, such as the nature of the acquisition environment, number of requirements, team experience, existence of a known feasible architecture early in the project, and about 40 other factors.

To showcase the capability of causal modeling tools, the authors select two algorithms to analyze two datasets. Both datasets consist of survey results, one that was created for and analyzed in (Sheard, 2012) and the other from surveying software developers in a high-maturity government organization. These analyses give insight into the factors for project context, resources, stakeholder dynamics, and level of experience that cause particular project outcomes.

The authors conclude that causal modeling methods provide useful and usable insight for project management, extending the capabilities available using more traditional statistical methods toward achieving a more fundamental understanding: among the many options available to a project manager, which are more likely to have desirable effects on project outcomes? For example, if the project is not progressing well, what interventions should be taken (e.g., provide staff with additional training, reduce the number of difficult requirements or stakeholder decision makers) and what would the effects of those interventions likely be?

1. Introduction

Researchers investigating causality in software engineering avail themselves of a mixture of both observational data and limited, randomized control trials. Correlations in observational data, by themselves, cannot generally determine which relationships are causal (Cook, 2002); and while randomized control trials are the gold standard for establishing causality in other scientific domains, there are “practical and ethical considerations that limit the application of controlled experiments in many cases” (Spirtes, 2010). From a research perspective, what is needed is something to help bridge that gap, that is, to make causal and not just correlational inferences from whatever observational data and experimental data are available, and help target future research on specific sets of variables for further study to add to that base of knowledge. This is what Causal Inference is about.

In this paper, we describe the application of causal search algorithms (also called causal discovery algorithms) to two project datasets to investigate what causal relationships can be learned. The analyses of these and similar datasets are ongoing, so these should be considered to be early results.

There are many (at least several dozen) causal search algorithms (Center for Causal Discovery, 2017b) but in this paper, due to time and space limitations, and because the two project datasets are similar, consisting of a few dozen project variables composed of answers given to multiple-choice survey questionnaires, we treat our data as discrete and will focus on two different types of causal search algorithms that work with discrete data. More background on and sources for causal search and modeling can be found in (Hira, 2018a) and (Hira, 2018b).

Why should we care about obtaining causal vs. correlational results? Project prediction and control begins with an understanding of which factors cause project outcomes, and their causes, et cetera. Correlational results only indicate which pairs of variables tend to change in the same direction together (or in opposite directions together); and are thus not a basis for deciding what intervention to take. To give a simple example, the presence of yellow fingers in year Y may be correlated with having lung cancer in year Y+10; the two variables are correlated. But controlling whether one's fingers are yellow in year Y by wearing gloves or staining them yellow is unlikely to have any effect on contracting lung cancer by year Y+10. Rather, there is a common cause to having yellow fingers in year Y and lung cancer in year Y+10, and that common cause is cigarette smoking. Indeed, conditioning on cigarette smoking in years Y through Y+10 should cause the correlation between yellow fingers and lung cancer to disappear. Likewise, when we speak of obtaining a causal understanding for software project control, we're talking about finding the causes of those project outcomes, and not merely the variables that correlate with those project outcomes (as yellow fingers in year Y correlates with lung cancer in year Y+10). Perhaps a more compelling example of the limits of correlation is found in Simpson's paradox (Wikipedia, 2018).

Here's what a causal understanding of project outcomes and the variables that control them should enable program management and acquirers to have the knowledge to be able to do:

- Understand why reported costs, schedule, and quality for a program are what they are
- Control program costs, schedule, and quality throughout software development and sustainment lifecycles
- Inform "could/should cost" analysis and price negotiations
- Improve contract incentives for software intensive programs
- Increase competition using effective criteria related to software cost

So the above is the goal, and causal learning, which includes causal search and causal estimation, is the mechanism for determining the types of interventions that can improve project outcomes.

The rest of this paper is organized as follows: Section 2 describes the two types of causal search algorithms featured in this paper: constraint-based search and score-based search. In Section 3, we describe one algorithm from each causal search method type: PC-Stable and FGES, which are constraint-based and score-based, respectively. We employ two different search algorithms that work by different mechanisms to help reduce risk of over-interpreting search results. This risk of over-interpretation is especially evident when working with small datasets, so we will show in Section 4 the effects of having a small dataset on obtaining causal inference in Case

Study 1, which uses datasets having 42 variables and only 61 and 81 cases. We investigate a somewhat larger dataset in Section 5 and demonstrate how much more refined our view can be when looking at the same dataset (Case Study 2) through the lens of causal search rather than a correlation table. Finally, in Section 6 we come to our Conclusions. Section 7 is Next Steps, Section 8 is References, Section 9 is Author Bios, and we end with a description of the survey instrument use in Case Study 2 (Appendix A) as it is not available elsewhere. (The survey instrument used in Case Study 1 is fully described in (Sheard, 2012).)

2. Causal Learning Summary

The paper (Hira 2018b) contains a short, self-contained and yet relatively complete introduction to causal search algorithms and how they work, which the authors of this paper contributed. That introduction is liberally excerpted in the Section 2 subsections that follow.

2.1. Causal Discovery

Quoting from the section on causal search algorithms in (Hira 2018b):

> Causal search algorithms (also, called causal discovery algorithms) typically take a dataset and settings (or hyper-parameters) governing the search, and output a set of graphs whose nodes are the variables appearing in the dataset (and depending on the algorithm, may include latent variables) and whose edges indicate some kind of direct causal connection between the pair of nodes they join. (Optionally, the algorithms also take sets of required and prohibited direct causal relationships between pairs of variables, which the user can use to encode the results of experiments or elicited domain knowledge.) There are many variations on this simple theme among the dozens of search algorithms, but in terms of understanding how they function and thus something of their relative strengths and limitations, it will help to organize them into two broad categories: constraint-based and score-based search algorithms (Spirtes, 2010).
>
> For both categories of searches, pointwise-consistent convergence has been proven. In other words, with increasingly-large datasets drawn from the same population, the algorithm will eventually find the correct causal graph(s). Unfortunately, uniformly-consistent convergence has not been proven, which could provide the rate of convergence and level of confidence for particular causal relationships (Spirtes, 2010).

2.1.1. Constraint-Based Search Algorithms

Again quoting from the section on causal search algorithms in (Hira 2018b):

> One of the first practical constraint-based search algorithms developed was the PC search algorithm, which is the algorithm used in the previous paper by the authors (Hira, 2018a). In its simplest form, constraint-based search involves two stages: Adjacency Search and Edge Orientation. Starting with a complete undirected graph, edges are

iteratively removed by testing for the conditional independence of joined nodes given a subset of neighboring nodes. If conditional independence is found, the edge is removed and the conditioning set employed is noted for later use in the Edge Orientation stage. This process is continued until all edges have been evaluated in this way. The result of this first stage, Adjacency Search, is thus an undirected graph. Edge Orientation starts with an undirected graph and iteratively orients edges according to a few rules that make use of the conditioning sets noted during the Adjacency Search stage. The result is an equivalence class of graphs, called a Markov Equivalence Class (MEC), rather than a single graph, because it is often impossible to determine the orientation of all the edges in the undirected graph that is output from the Adjacency Search stage (Spirtes, 2010).

For example, suppose we have a dataset featuring three variables, X1, X2, and X3; and the only independence discovered among them is: X1 is independent of X3 conditioned on X2. We also suppose we have no additional knowledge to encode about X1, X2, and X3; only the dataset. Then the Adjacency Search stage will output the undirected graph X1 – X2 – X3 (as well as some kind of note that the conditioning set that made X1 and X3 independent is {X2}). Then, given that particular independence, it necessarily follows that during the Edge Orientation stage, the direction of orientations for the edges of this undirected graph will not be able to be determined uniquely. Indeed, any of the following three pairs of orientations are valid, constituting the MEC: { X1 → X2 → X3, X1 ← X2 ← X3, X1 ← X2 → X3 }. Note that the following sequence of orientations is not part of the MEC: X1 → X2 ← X3. This type of relationship among variables is referred to as a collider. In a collider, the independence conditioning set is the empty set, because X1 is independent of X3 unconditionally. Hence, if the only independence found among X1, X2, and X3 is that X1 and X3 are unconditionally independent, then the MEC would consist of exactly one graph: X1 → X2 ← X3. Thus, colliders provide important clues for orienting edges during the Edge Orientation Stage (Spirtes, 2010).

While the idea of a set of graphs being the output of a causal search may disappoint, it is important to note that all graphs in an MEC have the same set of colliders and are built on top of the same undirected graph. Thus all graphs in an MEC manifest the same set of correlations present in the dataset, but may vary as to the causal orientations for some edges.

The settings or hyper-parameters of a constraint-based search algorithm typically include but are not limited to:

1) Type of independence test used (e.g., Fisher Z Test, Conditional Correlation Test in the case of continuous data; Chi Square Independence Test in the case of discrete data)
2) Confidence level for conditional independence testing (a parameter called Alpha, which is used as the cutoff for p-values in the independence testing specified by previous item)
3) Maximum size of conditioning set (e.g., for the first Case Study, we use the value 2, given the small sample size—see Section 3.1.2 for a fuller explanation)

2.1.2. Score-Based Search Algorithms

Again quoting from the section on causal search algorithms in (Hira 2018b) but with revision to the paragraph contrasting constraint-based and score-based search algorithms:

> To those readers more familiar with machine learning, score-based search algorithms employ a familiar mechanism: a maximum likelihood-based score (such as, Bayesian information criterion (BIC)). Like constraint-based search, there are two stages, both are iterative, and in each iteration of each stage there is both a currently-considered MEC (see above section for an explanation of this term, but it is important to note that all graphs in a MEC share the same underlying undirected graph and same colliders) and a set of neighboring MECs, that either each possess an additional edge (first stage of search) or have one edge removed (second stage of search) (Spirtes, 2010).
>
> In each iteration of the first stage, from the currently-considered MEC, the algorithm scores all neighboring MECs that have one additional edge. The best-scoring neighboring MEC then becomes the currently-considered MEC in the next iteration. The algorithm continues to iterate, building graphs one edge at a time, until a better score cannot be attained. In the second stage, the algorithm proceeds similarly but in reverse, considering only those MECs having one edge removed. Again, the algorithm halts when no better score can be attained, and the resulting MEC is returned as the output (Spirtes, 2010).
>
> The relative pros and cons of score-based vs. constraint-based search algorithms are:
>
> - Score-based search algorithms:
> - Scale better and are more computationally efficient (Ramsey, 2017)
> - Scores closer to ground truth, especially for denser networks (Triantafillou, 2016)
> - Constraint-based search algorithms:
> - Are intuitive (Friedman, 1998)
> - Are modular: the choice of how to build graph (adjacencies and orientations) is orthogonal to choice of what independence test to utilize (Spirtes, 2010)

2.2. Causal Estimation

And a final quotation from the section on causal search algorithms in (Hira 2018b):

> When we wish to apply the results of our causal modeling to make predictions about the future values that variables are likely to attain arising as a result of hypothesized interventions, we need to quantify the result of our causal search. This is called causal estimation. Causal estimation involves parameterizing the variables and relationships appearing in the search graph resulting from causal search and then estimating what values to assign these parameters from the dataset. The resulting quantitative model can

then be evaluated for model fit. Causal estimation is not the focus of our paper, and thus we don't address that topic further in this paper and more can be found in (Spirtes, 2010).

3. Case Studies Involving Discrete Data

3.1.1. General Approach

Both project datasets consist of survey responses to multiple-choice items. Any additional information, for example, free-form comments to specific survey items, were not utilized in these analyses.

When working with response data from a survey whose items have an underlying ordinal structure such as a Likert scale (Case Study 1), the researcher has a choice to make: should algorithms designed to work with continuous variables be used, or algorithms designed to work with discrete variables (nominal data with either no order, a partial order, or a total order [ordinal]) be used? This is a more general question with some complicated considerations (Pasta, 2009); however, beyond the general answer of "it depends on your research question and your data," employing both types of algorithms seems to be a fair strategy, mindful that you're throwing away useful information in one case (when treating your data as nominal when it in fact has a total order; or when discretizing continuous data), and introducing incorrect information in the other (that an increment of one in an ordinal scale has the same meaning at every point in the scale).

In Case Study 1, almost all of the variables are Likert variables on a five-point scale, and so either category of algorithm could be used; whereas in Case Study 2, all the survey items are binary, and thus we should limit ourselves to only using algorithms designed for discrete data. Until recently, the causal search algorithms available through Tetrad assumed the entire dataset could be categorized one way or the other, though the situation has recently changed and algorithms addressing mixed data types are now also available (Center for Causal Discovery, 2017b). However, in both case studies, for the purposes of this article, the authors chose to investigate their datasets by employing only those causal search algorithms designed to work with discrete data types.

Beyond the option of discrete versus continuous, there is also the option, as mentioned earlier, of employing two different types of search algorithms that work by different mechanisms. Doing so will help reduce the risk of over-interpreting search results. In addition, by using two such different causal search algorithms, we can compensate for some of the weaknesses inherit in each causal search type—see Section 2.1.2 for a contrast. By using both constraint-based and score-based causal search algorithms, the authors are striving for an appropriate level of confidence in the causal search results, especially where the two resulting search graphs (graph sets) have common edges and orientations of edges. Nevertheless, it has been the authors' experience that different algorithms will provide different results, while having much consistency with each other. In particular, there seems to be value in slightly varying the

parameters and comparing the results using the Compare box in Tetrad that lists the specific adjacencies and orientations discovered in any of the search results, sorted by frequency of occurrence.

3.1.2. Causal Discovery Algorithms Employed

Here is a description of the search algorithms employed in this paper:

1) **PC Stable** is a variant of PC, which historically was among the first causal search algorithms to resolve the exponential time barrier for constraint-based causal search of a dataset of N variables. PC Stable addresses a problem with PC that what causal graphs it outputs depends on the order of the variables within the dataset (Colombo, 2014).

   Parameters:
   - Alpha: case studies 1 and 2 use values for the Alpha parameter befitting the sample size: a larger value for Alpha for Case Study 1 (Alpha = .10) and a smaller value for Case Study 2 (Alpha = .05).
   - Independence Test: Chi Square Test for Discrete data
   - Adjacency search: PC-Stable is specified
   - Collider conflicts: Orient bidirected is specified
   - Maximum size of conditioning set: for both case studies, we use the value 2, given the small sample sizes. When the sample size is small, it is very important to choose a small value for this parameter, because the expected values for the cells of a conditional probability table used to test conditional independence among many variables but only a small sample will almost all be very near zero, which means that we're nowhere near to achieving the asymptotic behavior of the chi-square statistic necessary to obtaining valid p-values from independence testing.
   - All other parameters were set to their default settings.

2) **FGES** (Fast Greedy Equivalent Search) is a score-based search algorithms and perhaps best qualifies as the causal-search data analyst's favorite "go to" search algorithm after PC (Center for Causal Discovery, 2017a).

   Parameters:
   - Scoring method: Discrete BIC Score
   - Penalty Discount: the default of 2 is often used, but for small dataset sample sizes: smaller values for the Penalty Discount are used, for example 1.0 or even 0.5. This is because although BIC is statistically consistent in model selection, it may overpenalize for model complexity on small samples. For Case Study 1, the Penalty Discount 1.0 was used; while for Case Study 2, the default 2.0 was used.
   - All other parameters were set at their default settings.

3.1.3. Confounders

We note that it is possible to have confounders (unobserved direct common causes of two variables in a system). However, in the causal discovery literature, properly dealing with confounders is still a challenging issue. For the data sets analyzed in the two case studies, researchers attempted to collect most, if not all, factors that might be relevant to the study. As a consequence, the researchers evaluate the risk of ignoring significant direct common causes which would have large effects is low. We therefore decided to adopt causal search methods that assume no confounders in order to benefit from the asymptotic correctness of the search results.

3.1.4. Tetrad

As part of a National Institutes of Health (NIH) Big Data initiative, the University of Pittsburgh, Carnegie Mellon University (CMU), and Pittsburgh Supercomputing Center serve as founding members of the Center for Causal Discovery (CCD). The CCD develops and maintains causal algorithms, software, and tools, including the Tetrad[1] program with its GUI, API, and command-line interfaces (referred to as Tetrad in this paper). Tetrad allows users to run causal search algorithms on a dataset as well as estimate and evaluate parametric models. For each case study (sections 4 and 5), the authors loaded the appropriate dataset and ran causal search algorithms (section 3.1.2) using Tetrad. Example screens, options, and results can be found in (Hira, 2018a), (Hira, 2018b), and in sections 4 and 5 of this paper.

4. Case Study 1

4.1. Problem

This case study focuses on re-examining the results from an earlier study on complexity drivers of systems and software development project success.

In 2012, Sarah Sheard completed her Ph.D. dissertation titled “Assessing the Impact of Complexity Attributes on System Development Project Outcomes” (Sheard, 2012). Here is an excerpt from the Abstract:

> The purpose of this study was to determine complexity variables that can be measured on a complex system development effort early or in the middle of the project and that have an impact on project outcomes of cost overrun, schedule delay, and performance shortfall. ... [This study used a] retrospective survey on 75 projects, mostly from the aerospace/defense domain. Surveys provided answers to over 50 questions about outcomes, demographics, and complexity of the system, the project, and the environment. Three of the complexity variables strongly predicted all outcomes. These variables were the number of difficult requirements, the amount of “cognitive fog” present in the project, and the relationships among stakeholders. About twenty variables total were usefully congruent with the outcomes. These variables can now be used in heuristics that suggest which kinds of complexity to reduce on what entities in order to increase the likelihood of positive project outcomes.

[1] https://github.com/cmu-phil/tetrad

The question that motivated this case study is whether the results reported above (cognitive fog, stakeholder relationships, number of difficult requirements predicting cost overruns, schedule delay, and performance shortfall) characterize causal relationships (and thus intervening on a cause will affect an outcome) or merely correlations. We intended to answer this question by analyzing the same survey results using the causal search methods described in Section 3.

4.2. Data

The data and how it was obtained is described in full detail in (Sheard, 2012). In summary, based on the results of a literature review, Sheard developed a taxonomy that identified possible sources of complexity (the system, the organization that will build it, the environment it is developed in, and cognitive factors such as learning curve); at what stages will such indicators of complexity (whether structural, dynamic, or sociopolitical) be available. From this taxonomy, Sheard constructed a survey was that queried those familiar with a particular project some facts regarding outcomes and the presence of complex (or complicating) factors. The resulting survey was the administered, collecting data from individuals on about 75 projects. That dataset was analyzed in (Sheard, 2012) and re-analyzed for this paper.

Figure 1 describes the various project outcomes considered in (Sheard, 2012):

Delivered	At the finish: (1) Project delivered a product; (2) Project was cancelled without delivering a product
OverCost	At the point of finishing, how much did the project cost, compared to the initially predicted cost for delivery? (1) Under cost; (2) At cost +/- 5%; (3) 5-20% over plan; (4) 20-50% over; (5) 50-100% over; (6) more than 100% over plan
Late	At the point of finishing, how long had the project taken, compared to the initially scheduled development time? (1) Ahead of schedule; (2) On time within 5%; (3) 5-20% late; (4) 20-50% late; (5) 50-100% late; (6) More than 100% late
PerfGap	At the point of finishing, how was the project performance, compared to the initially specified performance? (Please consider the average performance of *mission critical* features, and add any qualifiers in Notes.) (1) Performance was higher than specified; (2) Performance was the same as specified, within 5%; (3) Performance was low by 5-20% (in terms of fewer features or waived requirements); (4) Performance was low by 20-50%; (5) Performance was low by more than 50%, or the project was cancelled
Success	Using whatever criteria are appropriate for this project, please describe how successful the project was, and what made the project successful or unsuccessful: (1) A great failure; (2) A mild or moderate failure; (3) Neutral; (4) A mild or moderate success; (5) A great success

EvolOp	How much did the system evolve during its operational lifetime? (1) System was never in operation; (2) System was in use, but it shut down fairly soon afterward; (3) System completed (or is in the process of completing) its intended operational lifetime essentially as delivered; (4) System was changed somewhat during its operational lifetime; (5) System evolved to be essentially a different system during its operational lifetime
GoodEst	"On the project, estimates generally turned out to be right." Do you agree with this statement? (1) Strongly Agree; (2) Agree; (3) Neutral; (4) Disagree; (5) Strongly Disagree

Figure 1 Project outcome variables in Case Study 1 (characterizations of project success)

Figure 2 describes some of the variables that could be collected earlier in a project that were reported in (Sheard, 2012) as predictors of project success.

Req-Diff	Difficult requirements are considered difficult to implement or engineer, are hard to trace to source, and have a high degree of overlap with other requirements. How many system requirements were there that were Difficult? (1) 1-10; (2) 10-100; (3) 100-1000; (4) 1000-10,000; (5) Over 10,000
CogFog	"The project frequently found itself in a fog of conflicting data and cognitive overload". Do you agree with this statement? (1) Strongly Agree; (2) Agree; (3) Neutral; (4) Disagree; (5) Strongly Disagree
StakeRelnship	Where did your project fit in the following eight attributes, on a scale of (1) Traditional, (2) Transitional, or (3) Messy Frontier? Stakeholder relationships: (1) Relationships stable; (2) New relationships; (3) Resistance to changing relationships

Figure 2 Project factors that predict project outcomes in Case Study 1 according to (Sheard, 2012)

Our purpose here is not to give a complete description of all factors but to see how far we can recreate the findings in (Sheard, 2012). Our focus above is on describing just these factors in greater detail.

4.3. Discovery Results

In Sheard's dissertation, the relationships among complexity drivers and project outcomes is displayed in a color-coded table (see Figure 29, page 159 in (Sheard, 2012)). The table depicts the results of an effort to understand causality using a traditional statistical approach to evaluate the relationships among these variables. As we seek to make a contrast between what causal search offers over a more traditional approach, we provide an extended excerpt from (Sheard, 2012) describing this color-coded table and its interpretation:

> …created by placing all the p-values from all the t-tests in one chart, with the split variable on the left and the p-values for all other variables in the row. This results in an

N-squared chart in that the same variables appear on the left on the top. …the split variables appear down the left and t-test variables appear across the top.

The p-value in each cell was then replaced by ** if the means difference was very significant ($p<0.001$) or * if the means difference was significant ($p<0.05$ but not <0.001). Cells without text in them had p-values above 0.05.

Colors … indicate congruence. If the higher-complexity group (as split by the variable on the left) had a higher-complexity mean for the variable on the top, the two variables were considered congruent and the box was colored green. If the higher-complexity group had a lower complexity mean, the box was colored red. If the means were essentially equivalent (t-test showed 0.5 or more likelihood of the same population) then the box was colored yellow. Grey boxes denote the diagonal where the variables on the top and left are the same, and t-tests are meaningless.

Note that most of the relationships with … * or ** … are green. This suggests that, with the exception of the variables in the third group (start and finish year and project management techniques), projects that are more complex are more complex globally rather than in only a few variables. Projects do not get more complex in some variables (say, socio-political variables) while simultaneously getting simpler in others (say, technical variables).

…To make conclusions about which variables best align with improved outcomes, this chart should be ordered to make the most aligned variables on the top and left. This was accomplished by the following steps.

For all variables except project management and year variables, the number of *'s in a row and the number of **'s were counted and summed these via the formula:

sum = * + 2 **.

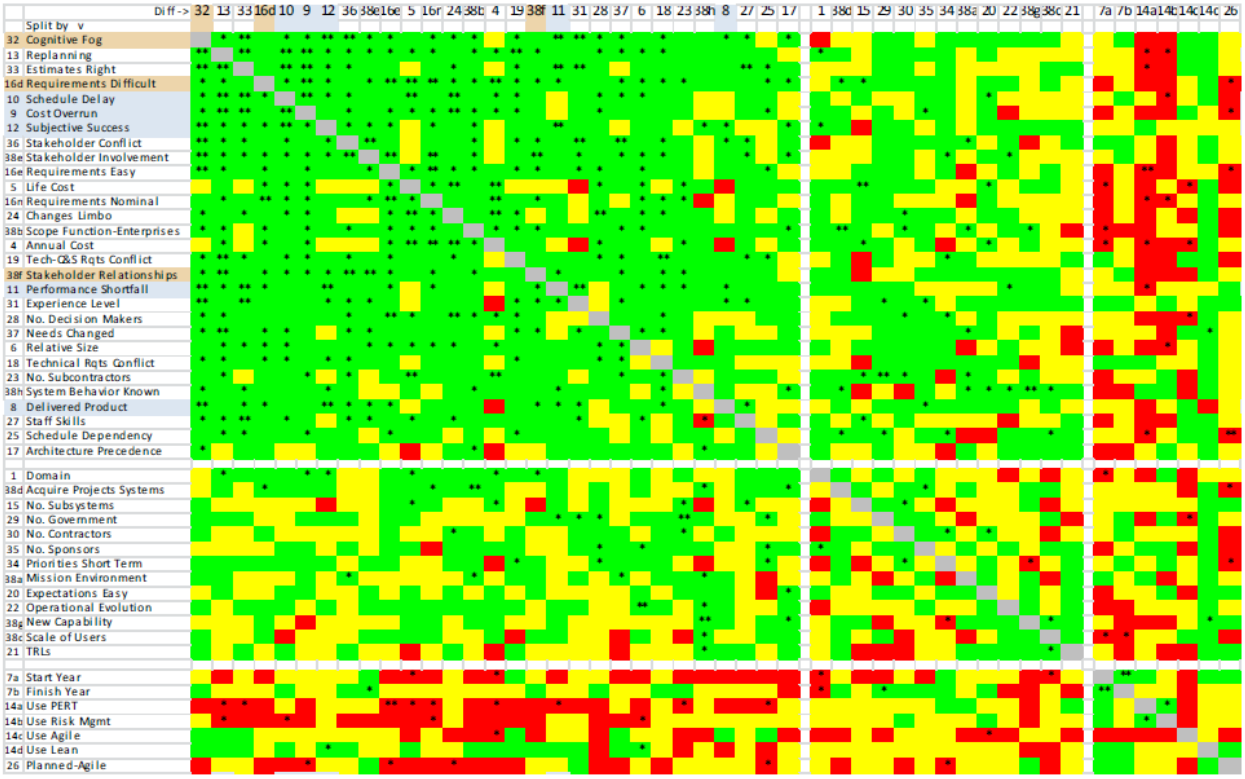
Similarly, the number of *'s and **'s in the same variable's column were counted and summed, again using sum = * + 2 **. Then the two sums were added to create a variable congruence number.

The spreadsheet was then sorted (rows and columns) by variable congruence, highest to lowest, with the project management [techniques] and year variables placed at the bottom. A blank line was added between the top nearly all-green group and the middle group, which had fewer cases of significance association with the top variables. …

The variable with the highest congruence value was Q32—Cognitive Fog, a middle-of-the project variable, which produces 20 significant variables when used as the split variable, and shows significant differences when 22 other variables are used as the split variable. Other high congruence variables are Q13—Replanning and Q33—Estimates right, which are hybrids of complexity and outcome variables and not terribly useful, as explained earlier in this chapter, and then Q16d—Requirements Difficult, another of the hypothesis variables. The third hypothesis variable, Q38f—Stakeholder Relationships, appears below others such as cost, changes in limbo, and conflict either among

stakeholders or between technical requirements and cost/schedule constraints. The reason Q38f—Stakeholder Relationships is a hypothesis variable and those others are not is because Q38f has significant relationships with all three outcome variables (Cost overrun, schedule delay, and performance shortfall) and the others are missing significant relationships with at least one of the three outcomes.

The table referred to in the above excerpt is shown as Figure 3 below.



*Significant, p<0.05; **Significant, p<0.001. Green: variable complexity rises together; Red: opposite; Yellow: neither. Blue=outcome variables; Beige=hypothesis variables.

Figure 3 Congruence among complexity drivers and relationships with project outcomes (from (Sheard, 2012)).

Intriguingly, Sheard also constructs a hypothetical causal model (see Figure 33, page 251 in (Sheard, 2012)), which is shown in Figure 4. The intended use for such a diagram is to identify what factors to change to help ensure a project stays on track toward achieving success (or to minimize particular departures from success). Of course, that's also the intended use of results from a causal search, but the results of a causal search are based on the conditional independences identified within the dataset (or the log likelihood of the data given a particular causal model) and are thus more empirically grounded.

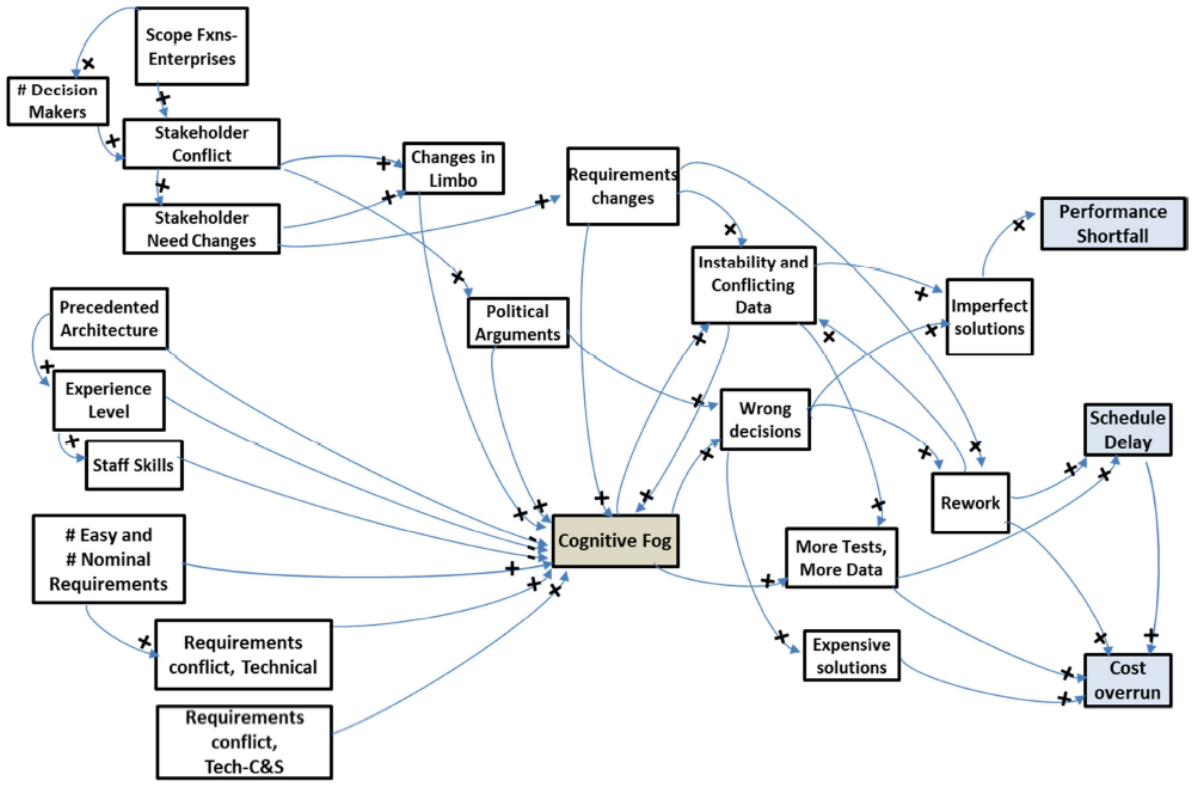


Figure 4 Relationship of Cognitive Fog to Other Variables (repeat of Figure 33 from (Sheard, 2012))

The next few paragraphs describe how we re-analyzed Sheard's dataset using causal search.

A typical preparation step in causal search is to organize the variables according to known precedence, therefore identifying what could not have caused what (with curious exceptions, later state or events cannot have caused earlier state or events). Providing such information may help disambiguate causal edge orientations.

In her original study, Sheard studied 52 variables potentially impacting project outcomes. After discussion with Sheard, ten of these variables were dropped from our causal analysis: the eight management technique variables (use of PERT, Agile, etc.) and two start/end date (date range) variables. The two start/end date survey items were dropped primarily because repeated causal testing did not find any causal role for the start/end date variables. The eight management technique variables were for a different reason: the part of the survey covering management techniques had confusing logic and only asked that the respondent to check "yes" if the technique was used on the program, and thus introduced uncertainty in the responses for these eight items. Removing these 10 variables left 42 variables for our study.

The remaining 42 variables were organized into three tiers in (Sheard, 2012) according to when they would likely be available for collection and thus serve as candidate early indicators of a project failing to meet desired outcomes. These tiers and two additional tiers are shown in Figure 5 below. Tiers 2 and 4 were inserted between Sheard's first and second tiers, and second and third tiers, respectively, to accommodate variables that may need to be collected in more

than one project stage in order to obtain an accurate evaluation of the project variable. These tiers are notional, but Tier 1 corresponds to what should be available or estimable near the beginning of a software project. Tier 3 corresponds to a stage of the project where the project is hitting its stride and if some issue is likely to be persistent for the remainder of the project, it is often manifest at this time to some stakeholders. Tier 5 corresponds to project outcomes. The figure is a screen grab from a Knowledge box in Tetrad where the user specifies how the variables appearing in a dataset should be partitioned into tiers, which is one way in which domain knowledge can be introduced into and inform a causal search.

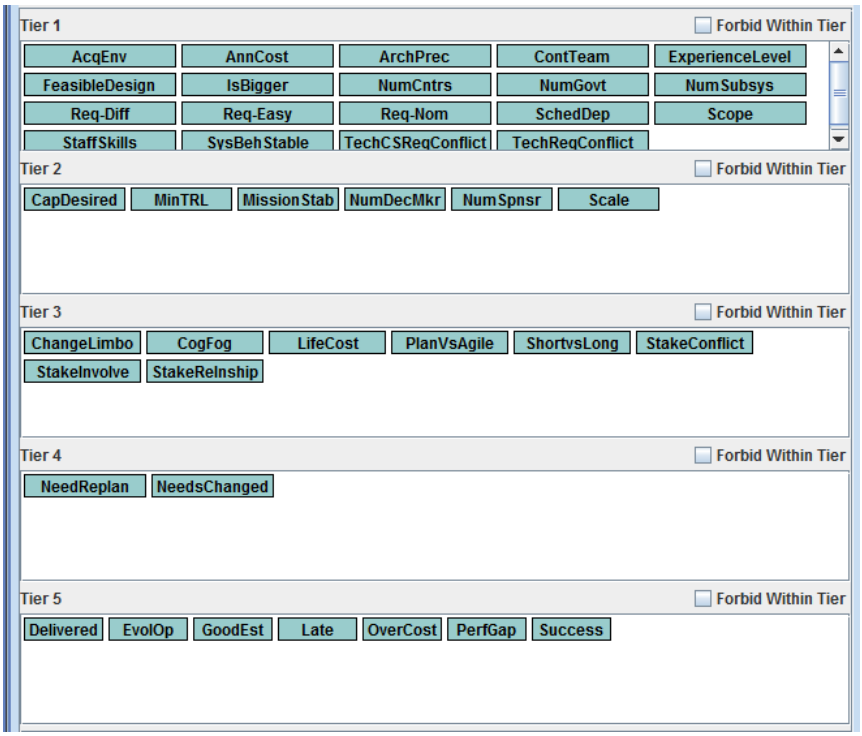


Figure 5 Organizing the 42 complexity and project outcome factors into 5 tiers approximating when they might become available during the life of a project

PC-Stable and FGES algorithms were then both applied to both the more limited dataset of Aerospace/Defense projects (61 in number) and the more full dataset of 81 projects (including Civil, Commercial, and Other). The result is four causal search graphs. An example graph from one of the searches (PC-Stable, full dataset) is shown in Figure 6 below.

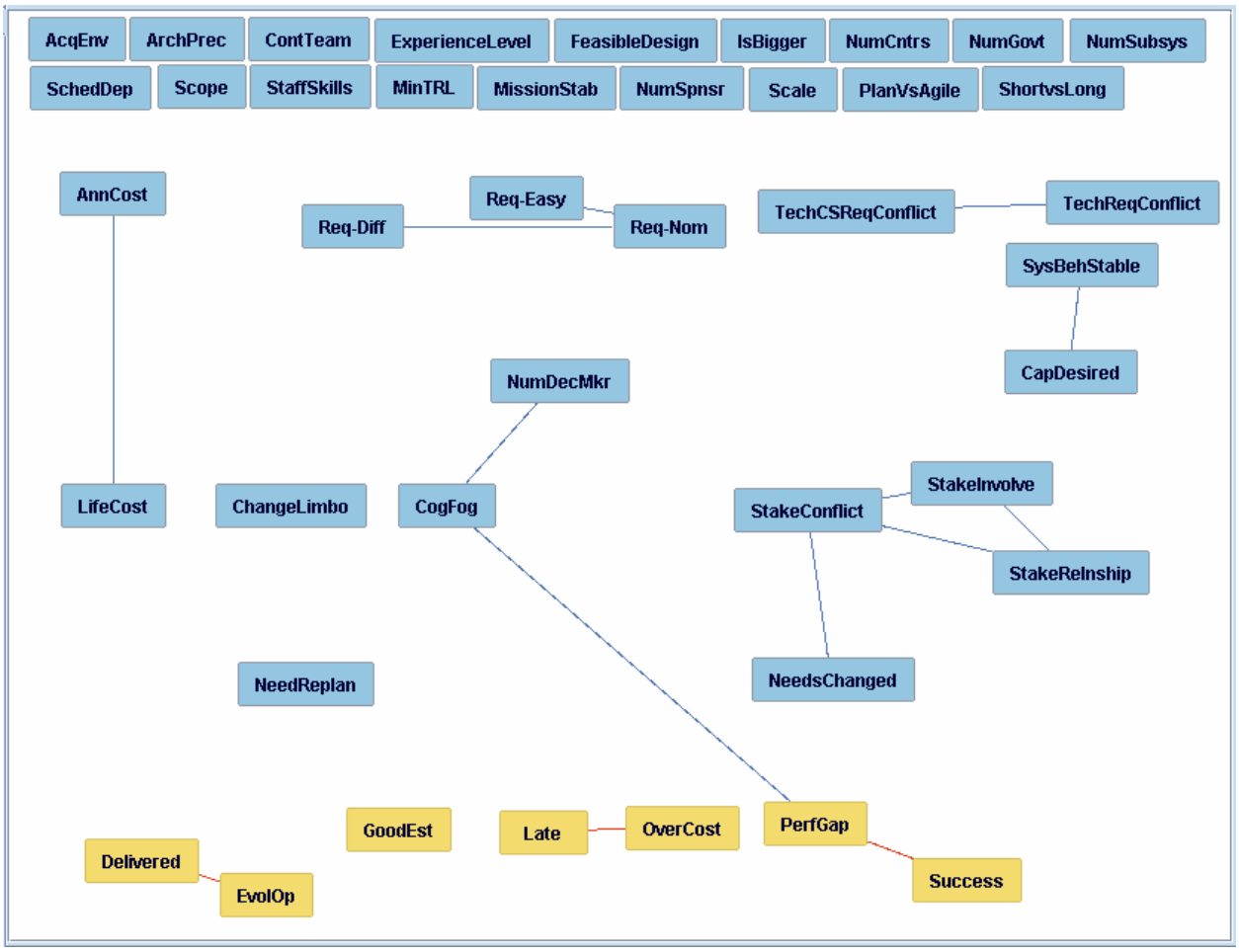


Figure 6 Result from applying PC-Stable (Alpha=.10) to the full dataset. Outcome (Tier 5) variables are highlighted in yellow. Variables without causal relationships appear at very top.

Note that almost half of the variables do not appear in any causal relationship. This would seem to be attributable to the relatively small sample size. The top two rows of variables consist of variables that do not appear in any causal relationship in any of the four searches (they were manually moved to those positions). The rest of the graph is where we've moved a lot of the variables that appear in one or more causal relationships among the four searches. Note:

- NumDecMkr is adjacent to CogFog, which is adjacent to PerfGap, which is adjacent to Success. Because the first three of these variables belong to different tiers, the first two of these adjacencies are direct causal relationships. The valence of almost all of the variables in the dataset is generally from good to bad (the underlying ordinal scale runs from good condition to bad condition), so one way to interpret this causal path is that a lot of decision makers causes a lot of cognitive fog, which in turn causes a big performance gap in project outcome, and that this causes or is caused by a lack of overall project success. (We can't determine causal direction for PerfGap and Success because, in part, both are assigned to the same tier.)
- Note the three requirements variables causally relate to each other (Req-Easy to Req-Nom to Req-Diff) but not to anything else, whereas (Sheard, 2012) discovers they had predictive value. This is not necessarily a contradiction, but instead reflects the inability

of a small sample to provide adequate power to identify any additional causal relationships involving these variables.

- The three stakeholder variables (StakeConflict, StakeRelnship, StakeInvolve) causally relate to each other and StakeConflict is adjacent to NeedsChanged (that the project needs did change), though there's not a direct path to any project outcome.
- Late and OverCost are also causally related to each other, which is not so surprising.

A word of caution in interpretation: whenever we say a variable directly causes another, this is relative to the set of other variables included in the causal search. Had additional variables been included in the causal search, it is possible that one or more of them might have mediated the relationship between the two variables; and thus what was a connection in the search graph might be replaced by a more nuanced network of causal relationships.

How do the results of the four causal searches (from applying PC-Stable and FGES applied to both datasets) compare? In Tetrad, you can answer this question by feeding the results of the four search boxes are fed into a Tetrad Compare box. As a result, you obtain a report of with what frequency different adjacencies and directed edges appear in the four search graphs:

In all 4 graphs, the following adjacencies (direct causal connections) were found (under each stated frequency, the order of rows and of variables in a row is not significant).

1. CapDesired --- SysBehStable
2. LifeCost --- AnnCost
3. StakeRelnship --- StakeInvolve

In 3 graphs...

1. Delivered --- EvolOp
2. Late --- OverCost

In 2 graphs...

1. PerfGap --- CogFog
2. PerfGap --- Success
3. Req-Diff --- Req-Nom
4. Req-Nom --- Req-Easy
5. StakeRelnship --- StakeConflict
6. Success --- Delivered

In 1 graph...

1. CogFog --- NumDecMkr
2. NeedsChanged --- StakeConflict
3. Scope --- AcqEnv

4. StakeConflict --- SysBehStable
5. StakeInvolve --- StakeConflict
6. TechCSReqConflict --- TechReqConflict

Direct causal connections uncontradicted in 2 graphs (note the edge is now oriented)...

1. AnnCost --> LifeCost
2. SysBehStable --> CapDesired

Uncontradicted in 1 graph...

1. CogFog --> PerfGap
2. Success --> PerfGap
3. SysBehStable --> StakeConflict

Contradicted (where the search results of the four searches were not consistent):

(empty)

By setting Alpha equal to 0.10, we are admitting more false positives in the search results; however, where agreement is found among the results of multiple searches, we're more likely to have found true positives, especially if different types of causal search algorithms are used, as we have done here (constraint-based vs score-based causal search).

As mentioned, we could have also tried to search the dataset treating the data as Continuous rather than Discrete. This turns up additional causal connections but for this paper we focus on a treatment of the data as Discrete.

4.4. Case Study 1 Conclusions

Sarah Sheard's dissertation suggested that by addressing causes of cognitive fog and selected other complexity drivers, project outcomes could be improved.

Using PC-Stable and FGES algorithms and applying them to both the more limited dataset of Aerospace/Defense projects (61 in number) and the more full dataset of 81 projects (including Civil, Commercial, and Other) we discovered consistent evidence that indeed Cognitive Fog is a direct cause of some project outcomes, in particular, Performance Gap (appearing as PerfGap in Figure 1). There was some limited evidence for Stakeholder Relationships (appearing as StakeRelnship in Figure 2) being a direct cause of some project outcomes as well. Finally, there was almost no evidence for the number of difficult requirements being a direct cause of project success unless we tuned the causal search parameters in a way that also further increased the probability of false positives; however, recognizing that the sample size was small, the authors chose to exercise some level of conservatism in the parameter settings employed in the causal searches to keep the level of false positives relatively low.

Intriguingly, the causal search teased apart some of the causal influences. For example, specifically for Cognitive Fog, a direct cause is suggested: Number of Decision Makers. This result suggests that a program manager can take action to improving project outcomes: scrutinizing the organization of the program and its stakeholders to reduce the number of decision makers.

In any case, the bottom line is that even with a relatively small dataset with a number of cases only slightly larger than the number of variables, direct causal relationships could be found that can guide a program manager to make constructive changes to the direction of a program toward improved project outcomes.

5. Case Study 2

5.1. Problem

Several years ago, an SEI client from the U.S. Department of Defense (DoD) desired to further study software team dynamics across 100+ active maintenance projects with the aim to better understand what drives software team performance. Although a litany of published research existed for teams in general and for software teams specifically, there appeared to be a major gap in any systematic causal research of software team performance factors. Given this situation, the SEI partnered with the client organization to construct a set of surveys that covered a lengthy list of factors mined from Watts Humphrey's publication on leading Team Software Process (TSP) teams (Humphrey, 2006). More than 120 factors were identified in the publication as potential reasons for software teams performing differently, although only a few were embodied in official TSP software measures. The SEI and client felt that the factors could be measured subjectively through a set of periodic, random surveys to members of the client's software projects. The factors were grouped according to the time periods in which change would be expected and managed. As such, a set of factors based on a weekly team survey, represented in Appendix A, comprise the causal learning demonstration for this case study. This case study does not purport to cover a complete analysis of this data but seeks to show how traditional correlation of data can be misleading to decision makers who wish to act on the analysis and intervene with process improvements expected to increase team performance. For the scope of this paper, the authors only focus on causal discovery of relationships of the independent factors and three outcome factors related to cost, schedule and quality. The authors purposely will not discuss causal discovery from a time series standpoint, although that would equally make an interesting demonstration.

5.2. Data

Thirty-four factors were identified for the weekly survey which included identity of the organizational entity (e.g. squadron) and thirty-three subjective binary assessments by randomly-chosen individuals across the entities related to their software team operation. Although the original design for the survey included continuous and ordinal scales for the survey questions, the client felt strongly that individuals would not take the survey if it was not simple and quick. A series of experiments concluded that a binary set of questions would be

best initially to ensure desirable response rates. Individuals received communications from senior leadership encouraging participation knowing that all responses would be anonymous and analyzed by a third party, i.e., the SEI. The SEI took great pain to minimize required scrolling during survey administration and to enable point and click in rapid fashion. The resulting response rates varied week by week but were near 90%. The client invested time to ensure random sampling was representative across the entities but also did not require any one individual taking more than 3 surveys in a given calendar year. The SEI and client reasoned that analyses of this initial survey would motivate more detailed research on a subset of the 120+ factors. A total of 418 weekly survey responses made up the data set for this case study.

5.3. Discovery Results

One motivation for this paper is to contrast the traditional use of statistical correlation with what can be determined through application of causal search algorithms to these change drivers and team performance outcomes. In Figure 7, traditional correlation analysis of the independent binary factors against the three ordinal outcome factors may be seen. A number of correlations were used for corroboration including Kendall tau-b, Kendall tau-c, Gamma and Spearman's correlation. All four correlation measures were in agreement using the 0.05 cutoff for significance. Significant correlations are marked by the shaded blue cells. As shown, eighteen factors were significantly correlated with the Quality outcome, five factors were significantly correlated with the Cost outcome and twenty-one factors were significantly correlated with the Schedule outcome.

Although not shown, one could also review the correlation among the set of 33 independent factors themselves to observe the degree of multicollinearity. Depending on the next step of modeling, the multicollinearity may be desired (e.g., for Covariance Based Structural Equation Modeling) or not (e.g., logistic regression on the ordinal outcome factors). The authors continued performing ordinal logistic regression on each of the outcomes: cost, schedule and quality. As may be seen in Figure 8, subsets of factors remained significant when participating in ordinal logistic regression exercises of the three outcome factors. Although all three models performed poorly in context of the McFadden score (McFadden, 1974), specific factors for each outcome remained statistically significant. Three factors remained significant predicting Quality: 1) Team Consensus, 2) NeedUnplannedHelp, and 3) OpenClimateIdeas2. Three factors remained significant predicting Cost: 1) MissedLateDecisions, 2) TeamConsensus, and 3) ProcessProbResolved. Seven factors remained significant predicting Schedule: 1) LeaderDealPerfProblems, 2) PrioritizedWork, 3) MissedLateDecisions, 4) GoodImproveData, 5) GoodTeamCommunication, 6) StressOvertime, and 7) OpenClimateIdeas2. As a result, whether correlation of factors to outcomes is used or whether ordinal logistic regression is used, a decision maker would now be facing a dilemma in terms of which factors to address to improve any one of the outcome factors.

The question now arises whether causal learning comprised of causal discovery (e.g. search) and causal estimation, can shed further light on the real causal drivers of the outcome factors. At this point, the authors initiated two causal search journeys that, although not comprehensive, do serve to show the value add of such causal technology.

	QualityOutcome	CostOutcome	ScheduleOutcome
ID			
WeekNumber			
Squadron			
IndivUnclearGoals			
IndivMotivateByLeader	■		■
LeaderDealPerfProblems			■
TeamConflictNotResolved	■	■	
PerfMeasured	■		■
PrioritizedWork			■
ChangeDirection	■		■
QualitySuffer	■		■
IndivUnhappyTasks	■		■
MissedLateDecisions		■	■
IndivSatisRole			
GoodMeetings	■		
ProcessNonCompliance	■		■
TeamConsensus	■	■	■
LackConsensusImpacts	■	■	■
GoodProgressReviews			
GoodImproveData	■		■
OpenClimateIdeas			■
ExternalFeedback	■		■
TeamLoadBalanced			■
ReqtsNotAnalyzed	■		■
NeedUnplannedHelp	■		■
CustomerInvolved			
ProcessGuidanceUsed	■		
ProcessProbResolved			■
IndivQualityData		■	
IndivTaskDisatisfaction	■		■
GoodTeamCommunication			
StressOvertime	■		■
OpenClimateIdeas2	■		
OpenTeamDiscussion			
InternalTeamCooperation			■
F2FwithLeader			

Figure 7 Factors Significantly Correlated with Outcomes

	QualityOutcome	CostOutcome	ScheduleOutcome
LeaderDealPerfProblems			
PrioritizedWork			
MissedLateDecisions			
TeamConsensus			
GoodImproveData			
NeedUnplannedHelp			
ProcessProbResolved			
GoodTeamCommunication			
StressOvertime			
OpenClimateIdeas2			

Figure 8 Factors with Red Borders deemed significant in Ordinal Logistic Regression

Figure 9 displays the Knowledge Box within the Tetrad tool used for both discovery searches discussed in this case study (PC-Stable and FGES). One may observe that the authors used a 3 tier approach to add constraints for what causal relationships are not allowed in the analysis based on time precedence. In this instance, the three factors in Tier 1 are purely exogenous and cannot be caused by any other factors in the model. The three factors in Tier 3 represent true outcomes of interest and cannot serve as causal drivers of factors in the Tiers 1 and 2.

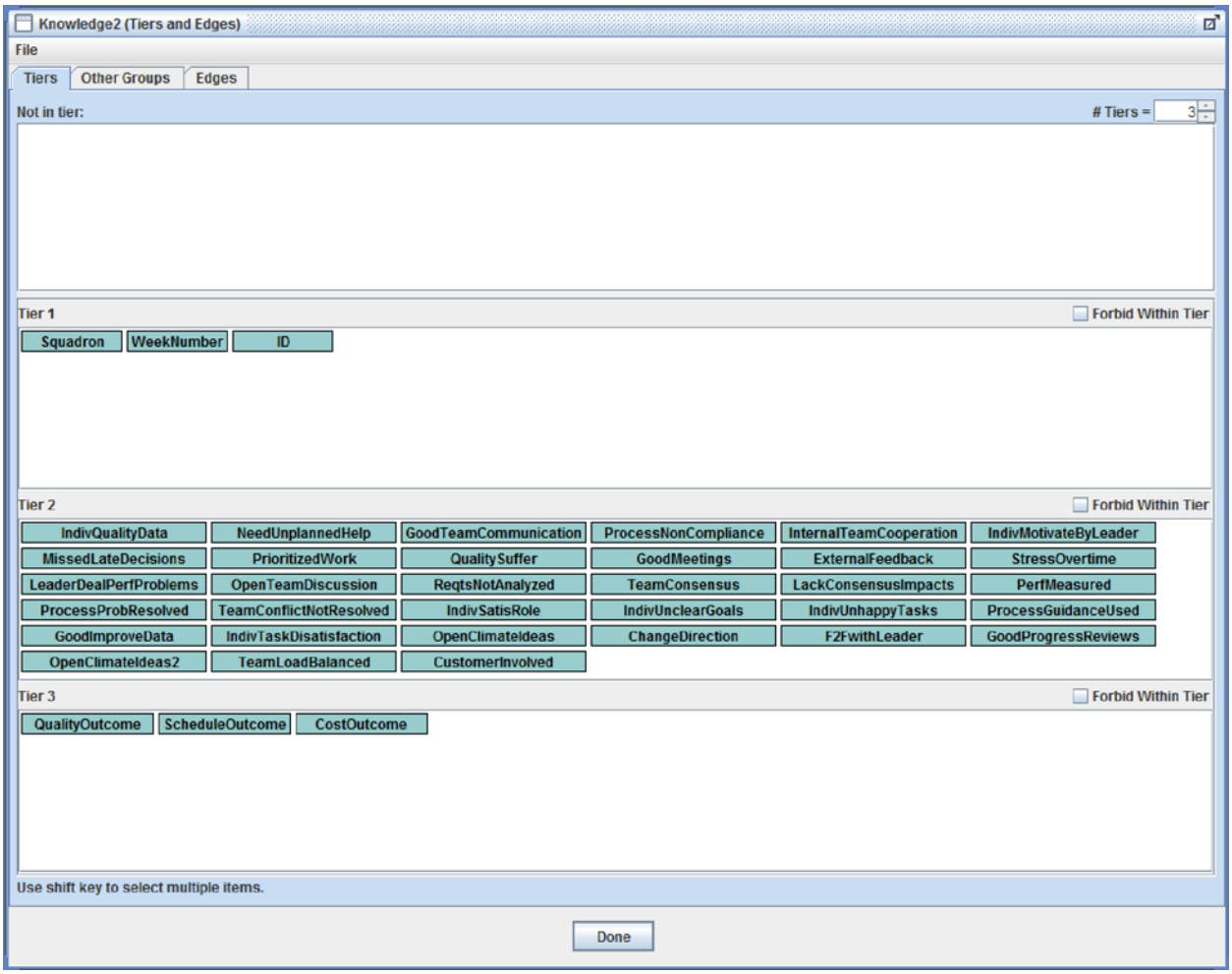


Figure 9 Tetrad Knowledge Box used in Discovery for PC-Stable and FGES

Figure 10 shows one of the output displays from Tetrad, the causal structure seen in the Search box of Tetrad. This initial search journey employed the PC-Stable algorithm using an alpha of 0.0001 and set the maximum size of the conditioning sets of 2. The causal structure in its current form is not easily readable but does show five factors unconnected to the remainder of the factors. The three outcome factors are highlighted in yellow to distinguish their placement. Such a causal structure from a Search box may include undirected edges, directed edges and bi-directed edges. Although this graph may provide causal answers for any of the possible relationships among factors, the authors next focus on the Markov blankets surrounding key factors of interest.

Figure 11 shows the Markov blanket for the set of three outcome factors using the PC-Stable search algorithm. As may be seen, only two other factors comprise the Markov blanket: 1) Stress from Overtime and 2) Good Improvement Data collected within the team. Thus, all influences on the three outcome factors must come through these two factors. As such, the three outcome factors are independent of all the other factors not in the Markov blanket, conditioned on these two factors.

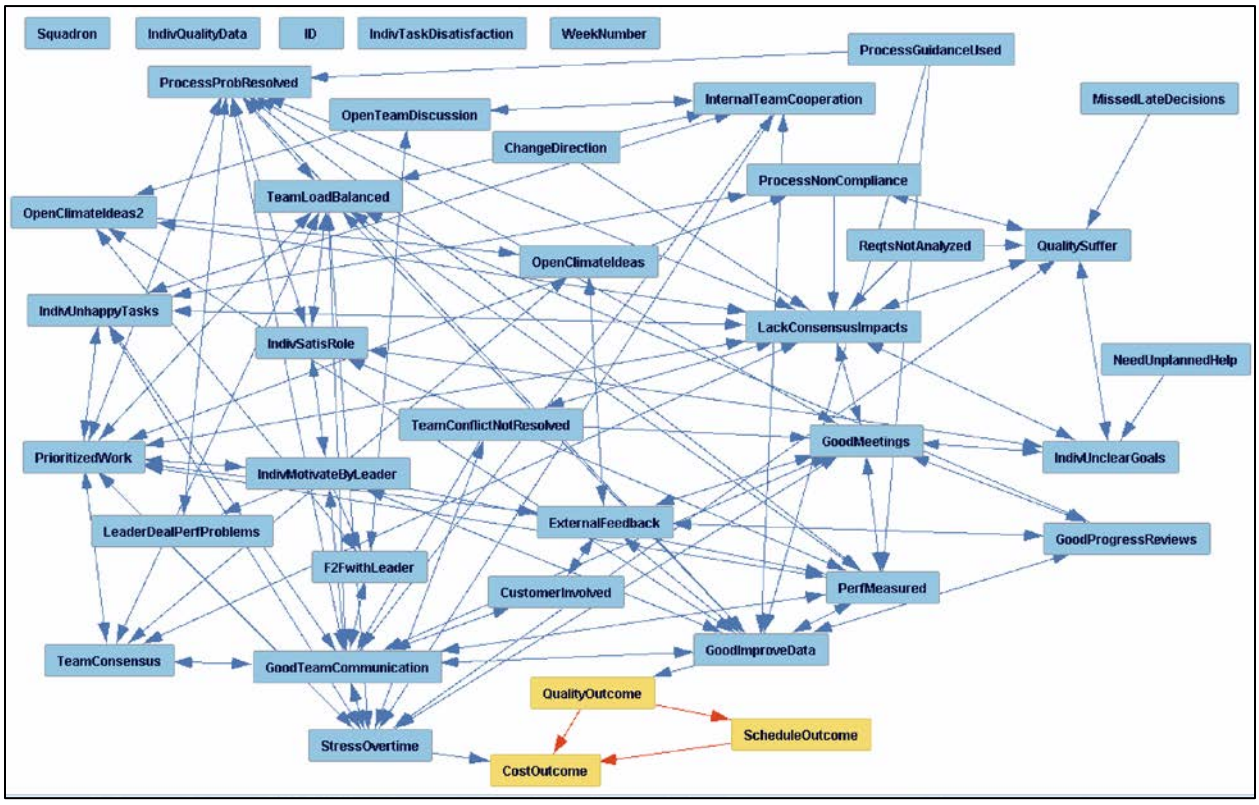


Figure 10 Overall Causal Structure from PC-Stable

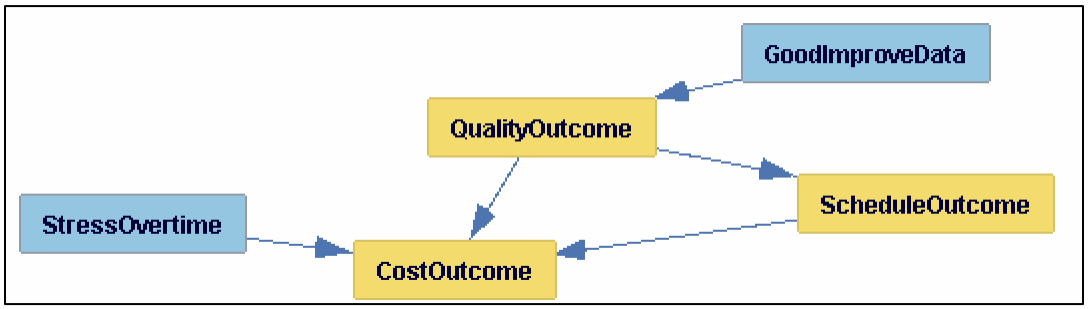


Figure 11 Markov Blanket for Set of Outcome Factors using PC-Stable

Figure 12 shows that one may also analyze the Markov blanket of any other factor in the causal structure. Here, one may see the Markov blanket for the factor, Quality Suffers, using the PC-Stable search algorithm. Hence, one can learn more about how the causal influences reach the factor, Quality Suffers. In this case, Quality Suffers can only influence factors outside of the Markov blanket through the factors in the Markov blanket.

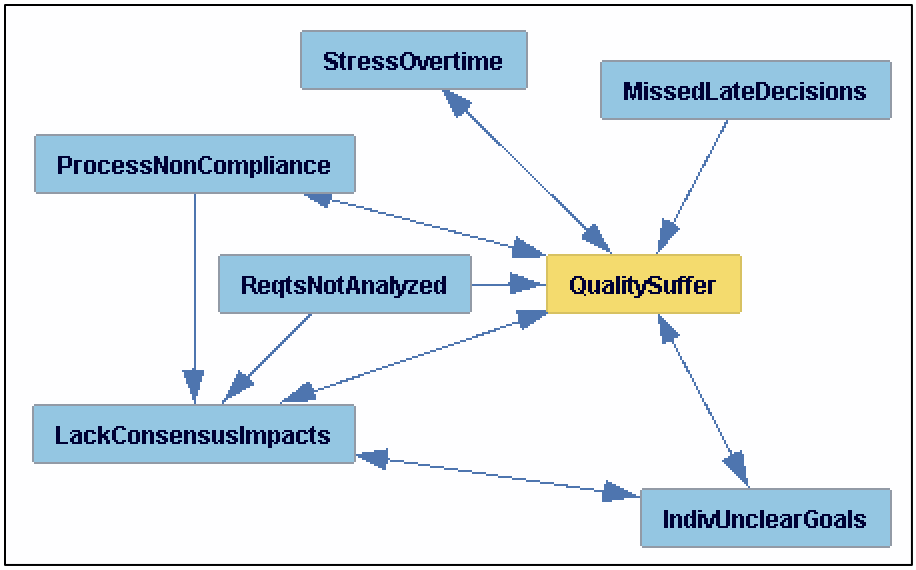


Figure 12 Markov Blanket for "Quality Suffers" Factor using PC-Stable

Figure 13 and Figure 14 depict the same results for the Markov blanket of the three outcome factors but using the FGES score-based search algorithm instead of the PC-Stable constraint-based search algorithm. There is agreement in the FGES and PC-Stable results of the Markov blanket for the three outcome factors except that PC-Stable additionally identifies a causal relationship between Quality to Cost and a causal relationship from Stress from Overtime to Cost. The Markov blanket for Quality Suffers agrees between both search algorithms with the exception of the role of two factors: LackConsensusImpacts and IndivUnhappyTasks. The authors do not necessarily expect to see unanimity across search algorithms but treat the collective results as informative on the causal structure and as motivation for further study.

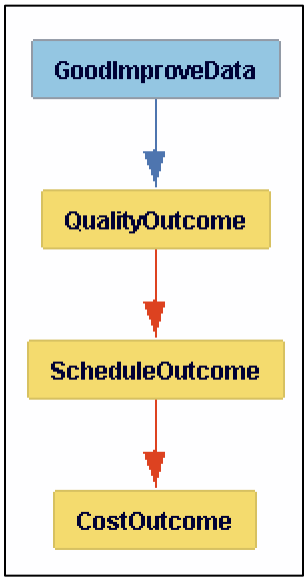


Figure 13 Markov Blanket for Set of Outcome Factors using FGES

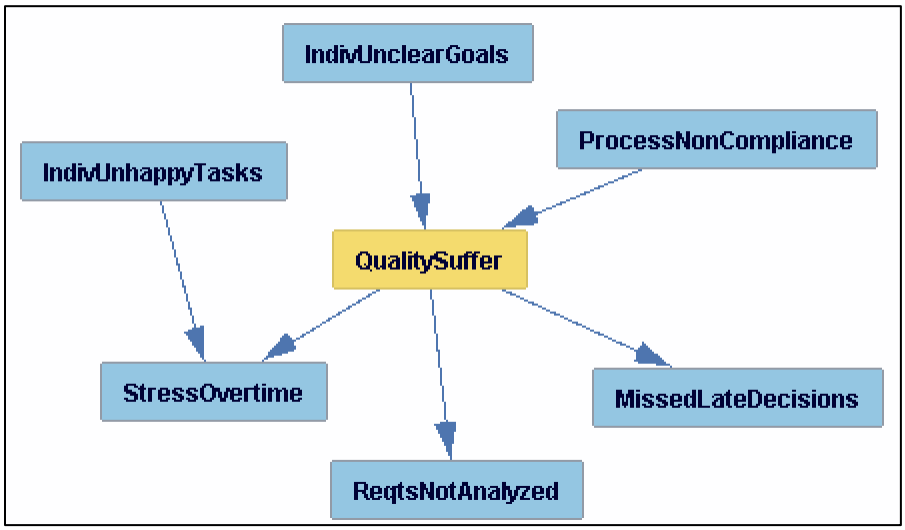


Figure 14 Markov Blanket for "Quality Suffers" Factor using FGES

5.4. Case Study 2 Conclusions

It remains interesting that both causal searches identified the causal structure among the three outcome factors to be: Quality causes Schedule causes Cost. In the FGES case, Quality also had a direct causal influence on Cost. Also of interest is that only two factors directly cause the three outcomes, namely GoodImprovementData and StressOvertime. Thus, outside intervention on the process to improve Cost, Schedule and Quality should focus on improving the quality of improvement data collected and used, as well as, taking action to reduce individual stress concerning working required overtime. Each of these two factors can, in turn, be analyzed by their Markov blankets to identify the direct causes to be manipulated to change them as well. This is in stark contrast to conclusions to be made from the traditional correlation analysis alone.

6. Conclusions

In both case, studies, interesting causal relationships were identified though the total number of projects or subjects involved was small (especially for Case Study 1). Both PC-Stable and FGES saw somewhat different but complementary, and mostly consistent relationships.

We believe this paper makes a case for other systems and software researchers to begin employing causal search algorithms in their research. In addition to providing a few simple predictive relationships, the causal graphs that are produced establish a much more nuanced causal context for interpreting variable relationships and predictors of project outcomes than can be found in most uses of regression. (However, properly applied, regression is a special case of causal search and estimation that produces results entirely consistent with casual search and

estimation (Spirtes, 2010).) Advancing beyond prediction, causal search enables the researcher to begin prescribing the nature of interventions that accurately define the expected changes in outcome factors.

7. Next Steps

7.1. Related to the case studies

In both of the case studies, additional causal search activity remains to be completed including the use of additional causal search algorithms to corroborate results and the sensitivity analysis of the various inputs to the search algorithms. To further validate the causal search findings, the authors would like to engage organizations to intervene and test the causal search findings for expected changes in the outcomes.

7.2. Related to desired research collaboration with others

The authors would enjoy additional research collaboration from others. The collaboration may occur in different forms depending on the situation related to the data and collaborator interests and availability. Several example collaboration approaches are detailed as follows:

1) Collaborators provide the authors with access to research data and the researchers provide causal search results as a service,
2) Collaborators provide the authors with access to research data and then receive training from the researchers and conduct causal search as a partnership with the researchers, and
3) Collaborators, for proprietary and other reason, receive causal search training from the researchers and then perform the causal search themselves, with long distance coaching from the researchers, thereby respecting the privacy or confidentiality of the research data. In this case, the researchers gain sanitized research results which may be published as sanitized causal structures.

7.3. Further deployment and adoption by cost estimation community

The authors would like to progress the adoption of causal search and estimation within the cost estimation community. We believe that cost estimation modeling would become more valuable with the capability to become prescriptive in nature, thereby guiding interventions in active programs and in contract negotiations. We believe cost estimation should progress far beyond prediction and offer greater value to cost stakeholders.

8. Acknowledgment

This material is based upon work supported in part by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The authors would like to thank David Zubrow (SEI) for his encouragement, support, and insights for the work in this paper. Additionally, the authors thank David Danks, Kun Zhang, Madelyn Glymour, and Joe Ramsey for their help in understanding causal search, the search algorithms, and the Tetrad tool.

The Tetrad program is released under the GNU GPL v. 2 license and may be freely downloaded and used without permission of copyright holders, who reserve the right to alter the program at any time without notification. Executable and Source code for all versions of Tetrad V are copyrighted, 2015, by Clark Glymour, Richard Scheines, Peter Spirtes and Joseph Ramsey. The Tetrad codebase is publically available on GitHub. The programmer's website can be found here (https://www.andrew.cmu.edu/user/jdramsey/).

9. References

Boehm, B. W., Madachy, R., & Steece, B. (2000). Software cost estimation with Cocomo II with Cdrom. Prentice Hall PTR.

Center for Causal Discovery, "Fast Greedy Equivalence Search (FGES) Algorithm for Continuous Variables."2017. http://www.ccd.pitt.edu/wiki/index.php?title=Fast_Greedy_Equivalence_Search_(FGES)_Algorithm_for_Continuous_Variables

Center for Causal Discovery. *Tetrad Manual*, June 2017. http://www.phil.cmu.edu/tetrad/

Colombo, D., & Maathuis, M. H. "Order-independent constraint-based causal structure learning." Journal of Machine Learning Research, 15 (1), 3741-3782. 2014.

Cook, T. D., Campbell, D. T., & Shadish, W. (2002). Experimental and quasi-experimental designs for generalized causal inference. Boston: Houghton Mifflin.

Elwert, F. (2013). Graphical causal models. In Handbook of causal analysis for social research (pp. 245-273). Springer, Dordrecht.

Friedman, N., and Goldszmidt, M. (1998). Learning Bayesian Networks from Data. AAAI Tutorial Program, 1998. http://ai.stanford.edu/~moises/tutorial

Hira, A., Sharma, S., & Boehm, B. (2016, May). Calibrating COCOMO® II for projects with high personnel turnover. In Proceedings of the International Conference on Software and Systems Process (pp. 51-55). ACM.

Hira, A., Boehm, B., Stoddard, R., & Konrad, M. (2018, February). Preliminary Causal Discovery Results with Software Effort Estimation Data. In *Proceedings of the 11th Innovations in Software Engineering Conference* (p. 6). ACM.

Hira, A., Boehm, B., Stoddard, R., & Konrad, M. (2018, TBD). Further Causal Search Analyses with UCC's Effort Estimation Data. Acquisition Research Symposium, 2018. http://www.researchsymposium.org

Humphrey, W. (2006). TSP(sm): Leading a Development Team. Pearson Education.

McCabe, T. J. (1976). A complexity measure. IEEE Transactions on software Engineering, (4), 308-320.

McFadden D. 1974. Conditional logit analysis of qualitative choice behavior. In: Zarembka P (Eds). Frontiers in econometrics (pp. 105-142). New York, Academic Press.

Nguyen, V. (2010, September). Improved size and effort estimation models for software maintenance. In Software Maintenance (ICSM), 2010 IEEE International Conference on (pp. 1-2). IEEE.

Park, R. E. (1992). Software size measurement: A framework for counting source statements (No. CMU/SEI/92-TR-20). Carnegie-Mellon Univ Pittsburgh PA Software Engineering Inst.

Pasta, D. J. Learning When to Be Discrete: Continuous vs. Categorical Predictors. Paper 248-2009, SAS Global Forum 2009. http://support.sas.com/resources/papers/proceedings09/248-2009.pdf

Pearl, J. (2001). Causal inference in the health sciences: a conceptual introduction. Health services and outcomes research methodology, 2(3-4), 189-220.

Ramsey, J., Glymour, M., Sanchez-Romero, R., & Glymour, C. (2017). A million variables and more: the Fast Greedy Equivalence Search algorithm for learning high-dimensional graphical causal models, with an application to functional magnetic resonance images. *International Journal of Data Science and Analytics*, *3*(2), 121–129. http://doi.org/10.1007/s41060-016-0032-z

Ruben Sanchez-Romero, Joseph D. Ramsey, Kun Zhang, Madelyn R. K. Glymour, Biwei Huang, Clark Glymour, "Causal Discovery of Feedback Networks with Functional Magnetic Resonance Imaging." 2018. https://www.biorxiv.org/content/early/2018/01/10/245936

Sheard, Sarah. *Assessing the Impact of Complexity Attributes on System Development Project Outcomes*, PhD diss., Stevens Institute of Technology, 2012.

Spirtes, P. (2010). Introduction to causal inference. Journal of Machine Learning Research, 11(May), 1643-1662.

Triantafillou, S., & Tsamardinos, I. (2016). "Score based vs constraint based causal learning in the presence of confounders." Causation: Foundation to Application Workshop, UAI 2016. http://people.hss.caltech.edu/~fde/UAI2016WS/papers/Triantafillou.pdf

Wikipedia contributors. (2018, June 14). Simpson's paradox. In *Wikipedia, The Free Encyclopedia*. Retrieved 16:49, June 22, 2018, from https://en.wikipedia.org/w/index.php?title=Simpson%27s_paradox&oldid=845889094

10. About the Authors

Michael Konrad is a Principal Researcher at the SEI providing analytic support to various projects using statistics, machine learning, and most recently, causal learning. Since 2013, Konrad has contributed to research in requirements engineering, software architecture, and system complexity measurement. From 1998 to 2013, he contributed to CMMI in many technical leadership roles. Prior to 1998, Konrad was a member of the teams that developed the original Software CMM and ISO 15504. He is coauthor of the main Capability Maturity Model Integration for Development (CMMI-DEV) books. Konrad received his PhD in mathematics from Ohio University in 1978.

Robert Stoddard is a Software Engineering Institute Principal Researcher at Carnegie Mellon University. His research includes machine/causal learning, applied statistics, Bayesian probabilistic modeling, Six Sigma, and quality/reliability engineering. Robert achieved an MS in Systems Management and significant doctoral progress in reliability and quality management. Robert is a Fellow of the American Society for Quality and senior member of the IEEE. Robert holds five ASQ certifications and is a Motorola-certified Six Sigma Master Black Belt.

Sarah Sheard is a Principal Engineer at CMU's Software Engineering Institute. She has over 20 years of experience in systems engineering, software and systems process improvement, and complexity science. A Founder's Award winner and Fellow of INCOSE, she wrote several well-known papers on systems and software engineering including, "Principles of complex systems for systems engineering" (2009), "Evolution of the frameworks quagmire," (2001), and "Twelve systems engineering roles," (1996). Her three "best papers" included the 2009 paper, "Capturing the Systems Engineering Process" (1992), and "Change Agency for Systems Engineers" (2015, with co-authors Dorothy McKinney and Eileen Arnold.) Since her 2012 Ph.D. dissertation on complexity and systems engineering, she has been researching complexity and safety, systems and software engineering during sustainment, and systems and software architecture interfaces. She also consults with government clients.

Appendix A: Binary Questions of Random Weekly Team Surveys

Please read each question carefully and answer with regards to your team's leadership during the past week:

	Yes (1)	No (2)
Were you unclear about any of the primary team goals? (3)	❍	❍
Were you motivated by your team leader to accomplish the task at hand? (4)	❍	❍
Did your team leader properly deal with any poor performers on your team? (5)	❍	❍
Were you aware of team conflicts that were not resolved in a timely manner? (6)	❍	❍
Were you satisfied with how performance was measured within your team? (7)	❍	❍
Were you satisfied with how your team prioritized work? (8)	❍	❍
Did your team leader need to unexpectedly change direction of the team? (9)	❍	❍
Were you aware of quality suffering to meet other goals? (10)	❍	❍
Were you aware of team members unhappy with their assigned work? (11)	❍	❍
Were you aware of important team decisions that were missed or late? (12)	❍	❍
Were you satisfied with your role as defined on the team? (13)	❍	❍

Please read each question carefully and answer with regards to your team's operation during the past week:

	Yes (1)	No (2)
Were your team meetings well-organized and well-run? (1)	❍	❍
Were you aware of team members not following the team plan or processes? (2)	❍	❍
Did your team reach consensus when needed on key team decisions? (5)	❍	❍
Did you team experience negative impacts from a lack of team consensus? (6)	❍	❍
Did you feel sufficient progress reviews were held within your team? (7)	❍	❍
Were you satisfied with the quality and timeliness of team improvement data collected? (8)	❍	❍
Was there an open climate to submit ideas for improvement? (20)	❍	❍
Were you satisfied with the feedback provided to the team from external stakeholders? (21)	❍	❍
Were you satisfied with the load balancing within your team? (22)	❍	❍
Were you aware of team requirements changes not accompanied by impact analysis? (23)	❍	❍

Please read each question carefully and answer with regards to your team's operation during the past week:

	Yes (1)	No (2)
Did your team need to ask for unplanned help from outside the team? (24)	❍	❍
Were you satisfied with the amount of customer involvement, via face time, meeting time, or telecons? (25)	❍	❍
Were you satisfied with the degree to which process guidance and checklists were used? (26)	❍	❍
Were you satisfied with the degree to which process problems were tracked, handled and resolved in a timely fashion? (27)	❍	❍
Did you collect your own personal quality data? (28)	❍	❍
Were you dissatisfied with the number of tasks you handled or the number of changed tasks? (29)	❍	❍
Were you satisfied with the frequency and nature of team communications? (30)	❍	❍
Did you observe team members under unusual stress or working excessive overtime? (31)	❍	❍

Please read each question carefully and answer with regards to your team's culture during the past week:

	Yes (1)	No (2)
Was there an open climate to submit ideas for improvement? (1)	❍	❍
Was open discussion and individual commitment demonstrated within your team? (2)	❍	❍
Were you satisfied with the degree of internal team cooperation? (3)	❍	❍
Were you satisfied with the degree of one-to-one face time you experienced with your team leader? (4)	❍	❍

Describe your team performance at this point in time.

	Worse than Plan (1)	On Plan (2)	Better than Plan (3)	Don't Know (4)
Quality (1)	❍	❍	❍	❍
Cost (2)	❍	❍	❍	❍
Schedule (3)	❍	❍	❍	❍