Title: Statistical Analysis of a Software Department Project Planning Process.

Course:
Year: 1994
Author(s): G.Tracy

Report No: P94067

Abstract: This project is a statistical study of the planning process of a software development department. The author describes the process used to plan software engineering projects. Next, he presents the actual results of projects. These are data acquired by analyzing the written records of software projects that followed Sequent's life cycle development model from 1989 until the present. He uses this data for a statistical analysis of the planning process, first developing a control chart to determine if the planning process is in a state of statistical control, then selecting a set of predictor variables and using linear regression analysis to test the null hypothesis that these variables are of no use in predicting variation in the schedule produced by the planning process. The software development planning process has many outputs including: schedule or predicted completion date, product content and functionality, detailed specification of who does what and when, and sales and return on investment analysis.

Statistical Analysis of a Software Department's Project Planning Process

Gary Tracy

EMP-P9467

9467

Statistical Analysis of a Software Department's Project Planning Process

by

Gary Tracy

Portland State University

EMGT 510

Dr. Dick Deckro

March 12, 1994

Introduction

This project is a statistical study of the planning process of a software development department. I will describe the process used to plan software engineering projects. Next I will present the actual results of projects. These results are data acquired by analyzing the written records of software projects that followed Sequent's life cycle development model from 1989 until the present. I will use this data for a statistical analysis of the planning process, first developing a control chart to determine if the planning process is in a state of statistical control, then selecting a set of predictor variables and using linear regression analysis to test the null hypothesis that these variables are of no use in predicting variation in the schedule produced by the planning process.

The software development planning process has many outputs including:

- schedule or predicted completion date
- product content and functionality
- detailed specification of who does what and when
- sales forecast and return on investment analysis

This study will focus on analysis of the planning process with respect to accuracy of the predicted completion date.

The Problem

Importance of Predictability In Software Engineering Projects

It has been stated that management is prediction [4, pg. 104] When we plan product development projects, an important output of the planning process is the predicted delivery date of the product.

When a project is late, a company's financial health is jeopardized. The most direct threat is loss of sales since the product is delayed in reaching the market, but there can be damage beyond loss of sales. The company's reputation can suffer. The financial community follows the progress of product development with interest, and provides regular progress reports to investors. A recent example is the NT operating system release from Microsoft. The Wall Street Journal followed the progress of this huge development closely, and reported on the extraordinary hours Microsoft engineers were putting in to bring the product out as quickly as possible.

Other damage is possible, for example the department most visibly responsible for project delay can be politically attacked by other departments. The late project invalidates the planning of other departments that depended on receiving output at a certain time. In the case of a software development project, this disrupts downstream operations in areas such as testing, manufacturing, product promotion, publicity, and formal introduction. All of these effects are costly.

Is There A Better Way?

First of all, what is late? What is the operational definition of late? The process described in this paper provides several schedule deliverables from planning teams, including a working delivery date and a contract delivery date. The idea behind these two dates is to provide project teams with aggressive delivery goals to shoot for during the implementation phase, as well as "90% reliable" contract dates that the team signs up to hit at all costs. This methodology aims to provide increased predictability in order to avoid the disastrous downstream effects described above.

This is a worthy goal, but by what method? How do planning teams make the call? This is where a statistical study might help. In addition to subject matter knowledge of how to design and build the software product, planning teams should understand the effectiveness of the planning process itself, in order to deliver reliable, credible schedules. This is the system in which they operate, and it should pay to understand the history and track record of that system.

The aim of this project is to investigate how statistical analysis might be used to provide information to project planning teams about the capability of the planning process that they are using. A further aim is to suggest how this information can be used to increase the effectiveness of a software department, and to suggest areas for further study.

Literature Search

There has been a lot written about how software projects should be planned and executed. See for example, Jensen and Tonies, Youll, or Humphrey. [8] [14], [7] With respect to planning, the focus of these books is on improving accuracy of estimation by such methods as hiring the best, most experienced people; training them in the best practices of software development teams; motivating them through incentive programs or fear so that they will do their best; and then controlling the development process or execution of project plans as carefully as possible. The literature admits that projects are often late [7] The answer to the lateness problem, however, is renewed application of finding the best people, motivating them and controlling the development phase carefully. There is variety in how to accomplish the finding, motivating, and controlling. De Marco and Lister propose a lot of autonomy, and management that truly treats the people as the most valuable asset.[3] Youll emphasizes methods to remedy the inherent lack of visibility in software development. You can't see software come together in the same way that you see a bridge reach completion. [12]

There has been interest in feedback systems to improve the estimation process. Abdel-Hamid recommends a feedback system that puts continuous monitoring data into the hands of the project manager, but the focus remains on better execution, and continuous correction during the execution phase.[1] House and Price incorporate the idea of money and time into their feedback system so that project teams receive feedback to reinforce not only accuracy of estimation, but also rapid actual time to market. [6]

But, the software project planning process might be a real world process that can be studied, understood, and improved with the statistical tools used with success by other process managers. Production processes have been optimized for years using statistical techniques invented by Dr. Walter Shewhart at Bell Laboratories during the nineteen twenties. [10] Just in time manufacturing that enables companies to eliminate costly warehouses relies heavily on the ability of statistical analysis to reduce variation and increase predictability to target levels.

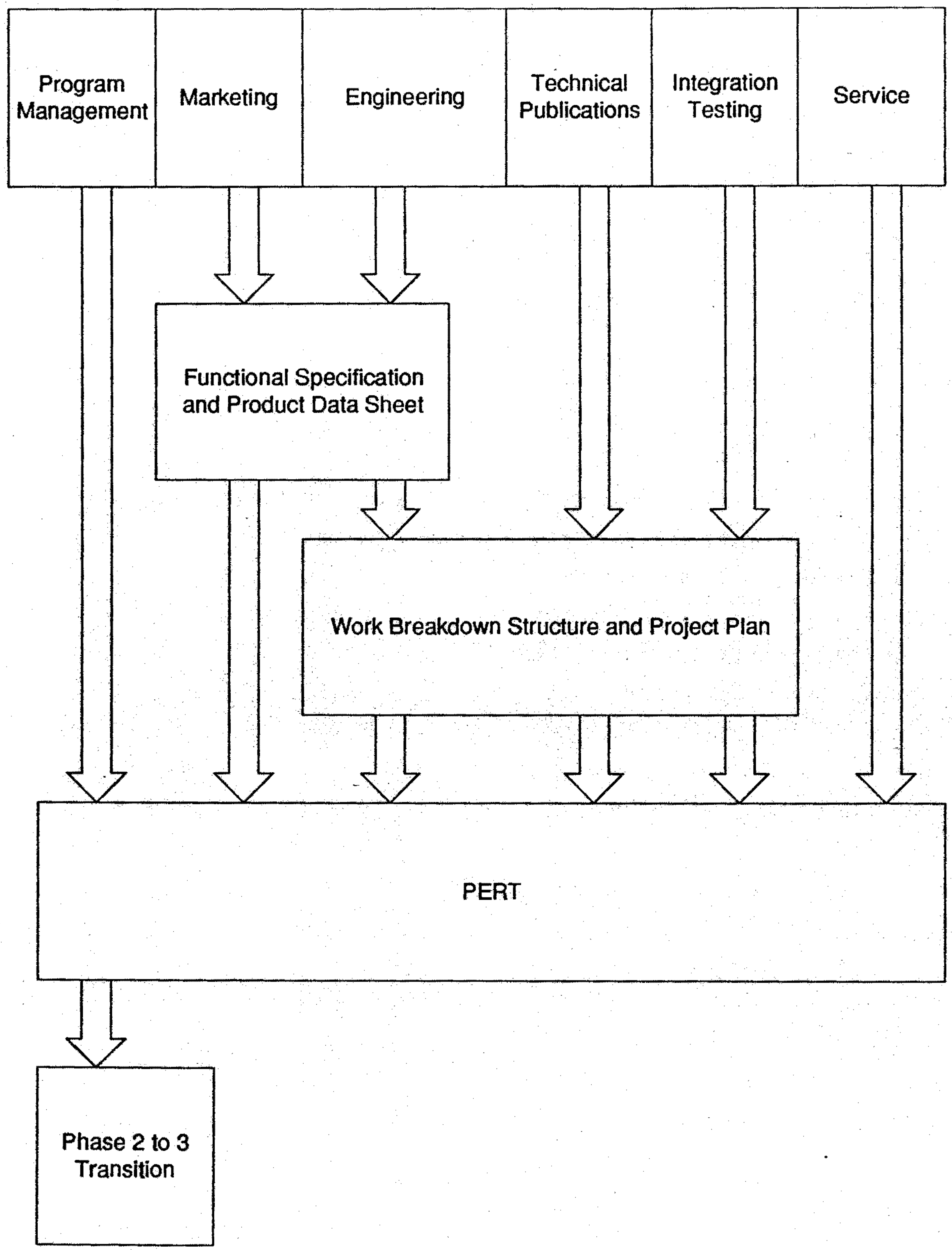
According to Wheeler and Chambers, statistical process control was invented by Dr. Walter Shewhart. [10, pg. 4] Statistical process control can be thought of as a method for listening to a process to determine the way things are, as opposed to the way things should be -- what you get as opposed to what you want. [10, pg. 37].

Not much is written about this approach to understanding the software planning process, but Richard Zultner, a consultant who specializes in applying Deming methodology to the software industry says he has used statistical process control in this manner with success [13], [14]

The Process

Sequent uses a life cycle project model with formal phase transitions. Projects are conceived, justified, and resources for planning are allocated in phase one. Detailed project planning occurs in phase two. Execution of the detailed plan happens in phase 3. The product is introduced to a controlled group of customers in phase 4, and the product is manufactured and made generally available to customers in phase 5. The following diagram shows phase two, the planning process. Time moves from top to bottom in the diagram, and the blocks in the top row show functional areas. The arrows indicate which functional areas have roles and responsibilities with respect to producing the deliverables described in the lower boxes.

Process Chart



The process is conceptually simple. Marketing and engineering collaborate to develop the functional requirements of the product. These are communicated through a functional specification document, and a product data sheet. The product data sheet contains an executive summary of the functional specification. Engineering, technical publications, and integration testing then work together to produce a work breakdown structure and detailed project plan. The work breakdown structure is a collection of the tasks that make up the project, together with an estimate of resources required to complete the tasks. The project plan specifies who does what so that the tasks will be completed. At this point there is sufficient information to construct a pert chart. The pert chart lays out the tasks on the project plan graphically, and shows task dependencies. Based on these dependencies, the pert shows not only who does what, but when it will be done. Service contributes to the pert, since the project timing is now understood, and the service organization has tasks to perform at the next phase transition. Program management contributes to the pert, since their role is to orchestrate the entire life cycle of the project. Finally, program management pulls all of the information together and presents the plan at a formal phase transition. One technique that Sequent uses to achieve reliable estimates is the concept of multiple predicted completion dates. The planning team estimates the completion date, and then establishes a so-called 50% date, a 90% date, and a contract date. As the names imply, the 50% date is an aggressive delivery date that the team feels 50% confident of hitting, and the 90% date allows more time for the project, so that the team has higher confidence that they can complete the project by this time. The contract date is the actual date that the team signs up to deliver on and be measured by, and it can be either or neither of the other dates. The idea is that the team will base the contract date on environmental factors such as time to market needs of the company, while the 50% and 90% dates will provide management with calibration parameters to help monitor and control the execution phase. Sequent works hard at finding the best people, motivating them and controlling the development phase carefully -- and projects are late.

The Data

The engineering organization studied here has been using the formal product life cycle process described above since late in 1989. Since the transitions are formal, and we keep records of dates, I was able to examine the notebooks full of phase transition data, and analyze that data from all software projects which have run under this process. For each project I collected or calculated the following information:

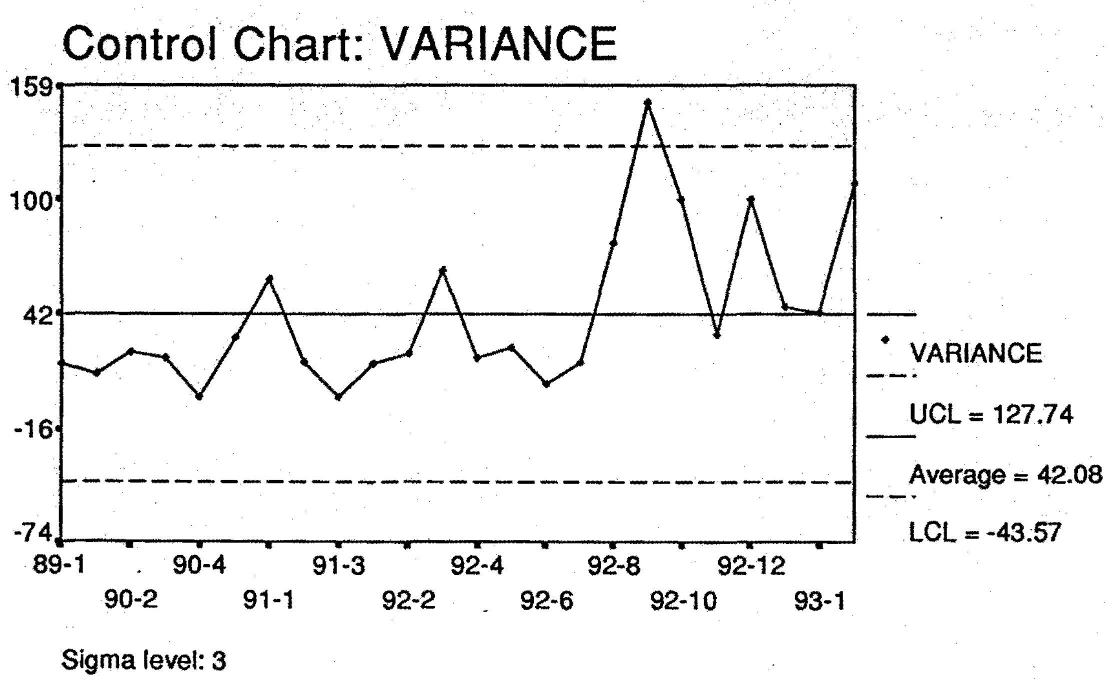
- project name
- start date of the project execution phase
- planned project completion date

- actual project completion date
- planned project size in staff months
- duration in calendar days of the project execution phase
- size of the project team
- percentage of the project spent in planning phase

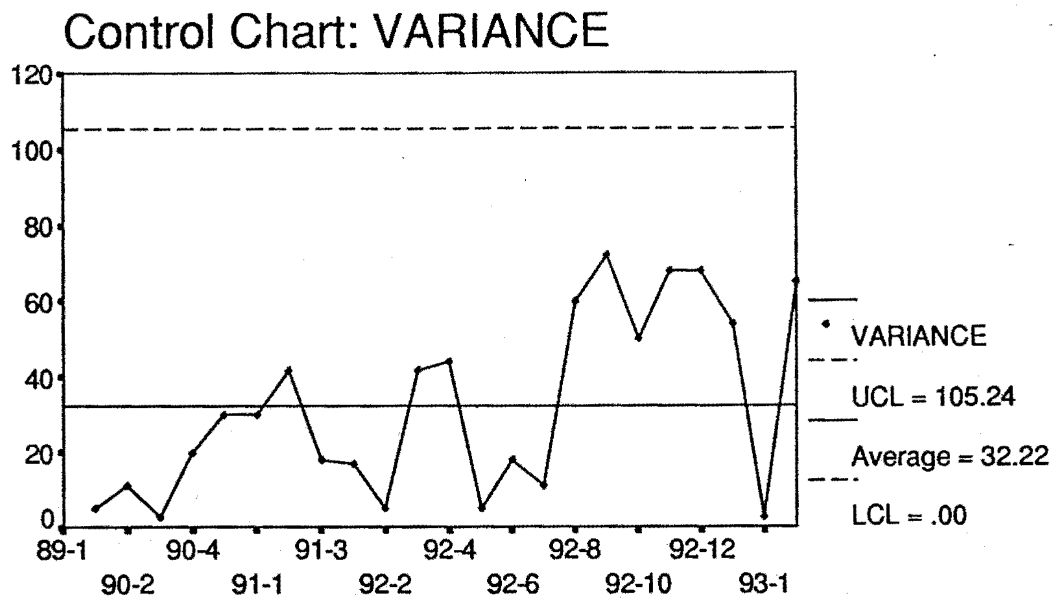
Since we are analyzing a process, the data that I have collected represents a sample from the population of possible output from this process. Any process conceptually produces an arbitrarily large population, so any set of results can be considered a sample.

Control Chart

My first step in analyzing this data is to construct an X-MR control chart using SPSS. I selected this type of control chart because according to Wheeler and Chambers, it is the most sensitive of the charts, and is appropriate for my data since my data falls into the periodically collected category. [10, pg. 217] Here is the first control chart:



Here is the moving range chart for the data:

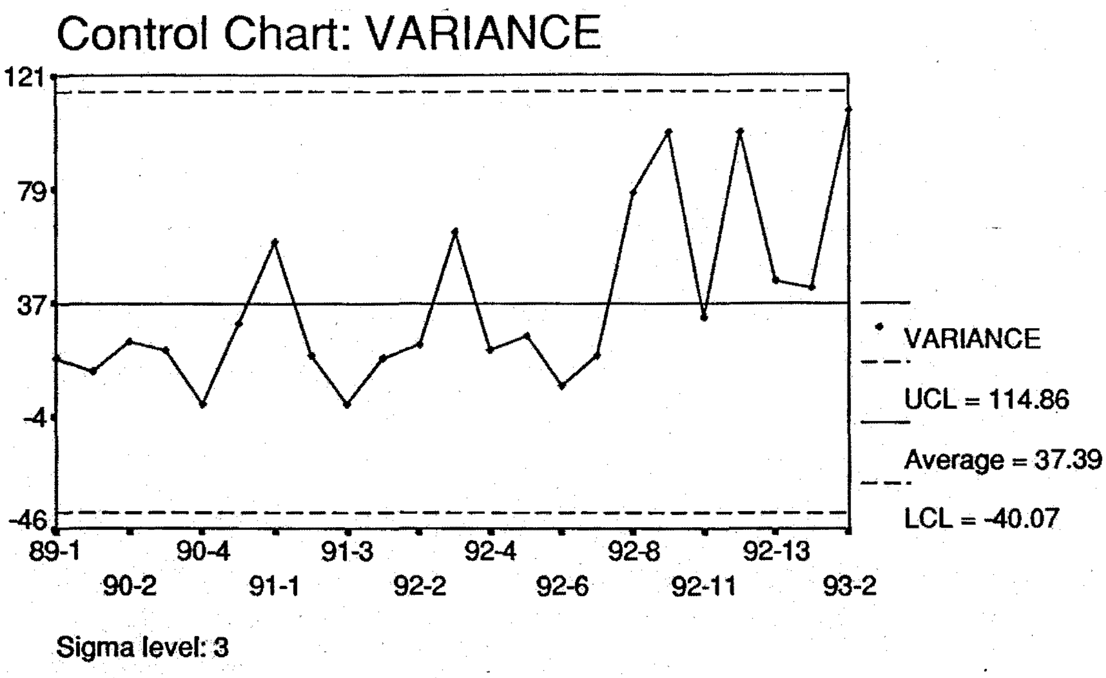


Sigma level: 3

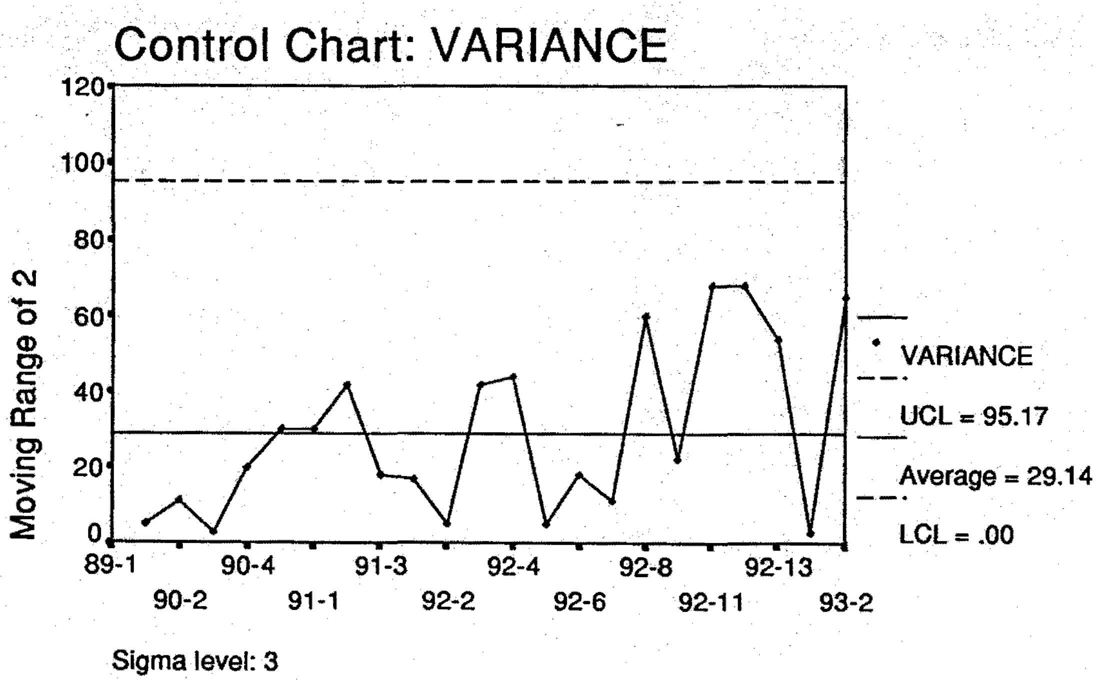
The chart tells us that the planning process is not in a state of statistical control due to project 92-9 that is outside the control limit on the high side. When a control limit is exceeded, the result that exceeds the limit is said to be affected by special or assignable cause. This means the chart has detected variation that is outside of the cause system inherent in the process. [5, pg. 312] The chart can detect the variation, but not the cause, so we must examine project 92-9. Project 92-9 is described in the table in the appendix. From there we can see that project 92-9 entered the planning phase in September of 1991, and stayed there until August of 1992, then it completed in 28 days. So we see that Project 92-9 is unusual in ways other than the large schedule variation. Project 92-9 is the shortest duration project in the set, but was in the planning phase for the longest time. I believe that the extreme amount of time between the start of planning and the start of execution, coupled with the very short execution phase indicates a legitimate special cause of variation, and so this project can be excluded without invalidating analysis of the process by analyzing the remaining projects. In order to improve Sequent's planning process based on this first control chart, the project 92-9 team should be consulted to find out more about the unusual behavior, and maybe management action or policy change can eliminate this potential special cause from the system.

The next step is to build a new control chart without project 92-9, and see if the planning process is in statistical control.

The following is a new control chart with 92-9 removed:



The following is the Moving Range chart:



Discussion

Now the process appears to be in statistical control. According to control chart theory, all of the variation in the schedule can be attributed to common cause, or natural variation in the process itself. [5] All values in the moving range chart are also within the control limits indicating that the system is free from discontinuities in the time series. [10, pg. 219] The process is in a state of statistical control which means that improvement requires changes to the process itself, but what steps can be taken to increase the process capability? The approach recommended by Brassard is to form a cross-functional team to analyze the process cause system with the aim of improving process capability. [Brassard] A cross functional team could develop a cause and effect diagram to try to identify and improve the cause system.

Development of Hypothesis for Multiple Regression

For regression analysis I designed a model with schedule variation as the dependent variable. Independent variables are planned project size in staff-months, actual project duration in days, percentage of time spent planning the project, and project team size. My null hypothesis is that with 95% confidence, no prediction is possible from my model containing these four predictor variables.

The SPSS output from this regression is included in the appendix. The variable called PROJ LN is project duration in days. PLAN TM is the percentage of project time devoted to planning the project, TM SZ is the project team size, and PROJ CST is the size of the project in staff months.

Analysis

I used the 'Enter' method of multiple regression, and asked for descriptive information about the data. This information is shown on SPSS output page 2. The mean value of the planning time variable is 46.7 which says that Sequent software project teams spend an average of nearly half of project time planning the project. The mean value of the project cost variable is 30.73 meaning that the average software project over the last four and one half years has been estimated to require about 31 staff months of effort. The mean value of the project length variable is 144.1. this represents the average actual length of Sequent software projects during the time period under study -- about five months. The mean value of the team size variable is 5.565, so Sequent project teams have an average size of about 6 people.

The N of cases line on page 2 tells us that there are 23 cases in my sample. This is acceptable, but on the low side of the recommended range of at least twenty to twenty-five. [2, pg. 294]

The correlation matrix on page two of the SPSS output indicates that project cost and project length are highly correlated. Project cost and team size are also highly correlated. This makes sense. A linear association is intuitively likely for these variables. A long project tends to be expensive and vice versa, and a project with a big team also tends to be expensive.

On page three of the SPSS output we find the R square and adjusted R square. The R squared represents the proportion of variability in the sample which is explained by the model. The .36184 value is low. F represents the null hypothesis test statistic. From the SPSS output we see from the Regression row, DF column that the numerator has 3 degrees of freedom. The numerator represents the number of questions asked in the null hypothesis. The Residual row, DF column shows that the denominator has 18 degrees of freedom. This number represents the number of observations minus the number of parameters estimated. Now I have enough information to test the null hypothesis. I look up F in the alpha equals .05 (95% confidence) table for numerator of 3 degrees of freedom and denominator of 18 degrees of freedom, and find the value 3.13. The calculated F for my model is 2.55150, and that is too small to support rejection of the null hypothesis.

Page four of the SPSS output is no longer of interest. The regression coefficients and constant from the variables in the equation table indicate that

variation = .07(project cost) - .2(planning time) - .25(project length) + 3.65(team size) + 59.7

But this model is not useful since the null hypothesis cannot be rejected.

Logically, this makes sense based on the control chart analysis of the process. Since the chart indicates that the process is in a state of statistical control, each of the schedule variance results can be attributed to variation in the process itself. The cause system for this variance is unknown so a search for predictor variables will be difficult. My simple model based on the available data does not provide subtle enough explanatory information.

Conclusions

The regression analysis demonstrated an absence of predictive capability for the model, but the control chart supports the theory that the planning process is not riddled with special causes, and so I might expect difficulty in developing a good set of independent variables. But the information provided by the control chart has potential value. This information indicates that Sequent's planning process will continue to deliver schedules which are missed by an average of about 37%, and with considerable variance, unless a change is made to the planning process itself. Management intervention is common during the execution phase when a project is observed to be behind schedule by 20% or so, but the control chart says the intervention is too late. The control chart provides an operational definition of late, and management intervention isn't appropriate unless a project exceeds the upper control limit of about 115%. Management action should be directed toward improving the planning process to reduce variation and lower the center line so that future projects will be more on target. [10, pg. xix]

Equally controversial is the implication that the information indicates equality of result for all projects that are within the control limits. The project 90-4 team that hit their schedule dead on was no better or worse than the project 92-10 team that missed their estimated completion date by 100%. Can we reach that conclusion for the planning process? The advantage of this conclusion is reduction in the number of occurrences of what Deming and Shewhart call Mistake number 1 -- attributing to special cause a result that is due to common cause.[4, pg. 102] The resources expended for intervention for a project that is behind schedule, but not outside the control limits, could be used to improve the planning process itself. The disadvantage of reaching this conclusion is the perception that management is not doing everything possible to deliver products on time. Management intervention during the execution phase of a project which falls behind schedule is common. Expensive and disruptive interventions include such actions as reorganization and reassignment

In my opinion, study such as this and the controversial information that it provides can help project teams improve, and so should be collected, maintained, and shared at all levels.

Future Work

It would be interesting to design some experiments. The problem is that each project is important so any experimental impact on the project itself would be hard to justify. Projects are also relatively long term, so the results of experiments will take a long time to acquire. There is not much opportunity for experimentation and quick feedback. Some managers consider analysis of the type done in this project potentially dangerous. If project planners see schedule variance being measured and studied then they might become too conservative in their estimates. This would bias analysis, but more importantly, it could slow the delivery of new products. The goal is not for planners to become more conservative, the goal is for planners to improve their process. There is an analogy to just in time manufacturing. "The lazy way is to build that warehouse and hope your competition is doing the same... of course he is not." [11]

The sample size is not very large. It would be useful to expand the study to include other software development environments if I could find others that use the same development process.

I would like to develop more predictor variables for another model. I believe that the following have potential predictive capability:

- project leader years of experience
- project leader number of previous similar projects
- average experience of project members
- number of projects the team has worked on together
- a measure of corporate visibility of the project
- time to market pressure during the planning phase
- degree of adherence to the process

I could collect this data by interviewing team members and leaders.

Works Cited

[1] Tarek K. Abdel-Hamid, "Adapting, Correcting, and Perfecting Software Estimates: A Maintenance Metaphor," *Computer*, pp. 20-29, Vol. 26, No 3, March 1993

[2] Michael Brassard. *The Memory Jogger Plus+*, Methuen, MA.: GOAL/QPC, 1989

[3] Tom DeMarco and Timothy Lister, *Peopleware, Productive Projects and Teams*, New York, NY. Dorset House Publishing Co., 1987

[4] W. Edwards Deming, *The New Economics*, Cambridge, Mass: MIT Center for Advanced Engineering Study, 1993

[5] W. Edwards Deming, *Out of the Crisis*, Cambridge, Mass: MIT Center for Advanced Engineering Study, 1986

[6] Charles H. House and Raymond L. Price, "The Return Map: Tracking Product Teams," *Harvard Business Review*, pp. 92-100, January-February, 1991

[7] Watts S. Humphrey, *Managing the Software Process*, Menlo Park, Ca.: Addison-Wesley Publishing Company, Inc., 1990

[8] Randall W. Jensen and Charles C. Tonies, *Software Engineering*, Englewood Cliffs, New Jersey: Prentice-Hall Inc., 1979

[9] Western Electric Co., Inc. *Statistical Quality Control Handbook*, Charlotte, North Carolina: Delmar Printing Co., 1956, 11th Printing May 1985

[10] Donald J. Wheeler and David S. Chambers, *Understanding Statistical Process Control*, Knoxville, Tenn.: SPC Press, 1992

[11] Donald J. Wheeler. *Short Run SPC*, Knoxville, Tenn.: SPC Press, 1991

[12] David P. Youll, *Making Software Development Visible, Effective Project Control*, New York, NY: John Wiley and Sons Inc, 1990

[13] Richard E. Zultner, "TQM for Technical Teams," *Communications of the ACM.*, pp. 79-91, Vol. 36, No. 10, October 1993

[14] Richard E. Zultner, personal correspondence, February, 1994

Selected List of Works Consulted

Bruce F. Baird. *Managerial Decisions Under Uncertainty*, New York, NY.: John Wiley and Sons Inc., 1989

W. B. Chew, "No-nonsense guide to measuring productivity," *Harvard Bus. Rev.*, pp. 110-118, Jan.-Feb. 1988

William Edwards Deming. *Some Theory of Sampling*, New York, NY. Dover Publications Inc., 1950

Lloyd Dobyns and Clare Crawford-Mason. *Quality or Else*, Boston, Mass: Houghton Mifflin Co. 1991

Andrew S. Grove, *High Output Management*, New York, NY: Random House Inc., 1983

Joseph F. Hair, Jr., Rolph E. Anderson, Ronald L. Tatham, and William C. Black. *Multivariate Data Analysis with Readings*, New York: Macmillan Publishing Co. 1992

John P. Kotter and James L. Heskett. *Corporate Culture and Performance*, New York, NY. The Free Press. 1991

David I. Levine, "Participation, productivity, and the firm's environment", *California Management Review*, Vol. 332, Summer 1990, pp. 86-100

Joel E. Ross, *Total Quality Management*, Delray Beach, Fla.: St. Lucie Press, 1993

Robert S. Schulman, *Statistics in Plain English*, New York, NY.: Van Nostrand Reinhold, 1992

Daniel A. Thomann and Donald E. Strickland, "Managing collaborative organizations in the 90s", *IM*, pp. 26-29, July/August 1992

M. Tribus, "Applying quality management principles," *Res. Manag*., pp. 11-21, Nov.-Dec. 1987