Forecasting from Defect Signals

Paul Below, QSM

Abstract. On large software development and acquisition Programs, testing phases typically extend over many months. It is important to forecast the quality of the software at that future time when the schedule calls for testing to be complete. Shewhart's Control Charts can be applied to this purpose, in order to detect a signal that indicates a significant change in the state of the software.

Introduction: Shewhart's Control Charts

Every process displays variation. Some processes display controlled variation and others display uncontrolled variation.

In the 1920s, Control Charts were invented by Walter A. Shewhart. In most of the rest of the 20th century the concepts were popularized by people such as W. Edwards Deming. The driver for this development as explained by Shewhart:

"The engineer desires to reduce the variability in quality to an economic minimum. In other words, he wants (a) a rational method of prediction that is subject to minimum error, and (b) a means of minimizing variability in the quality of a given produce at a given cost of production" [1].

Over several decades, we have found that software metrics commonly follow a Rayleigh curve [2]. This results in a very different situation from the typical use of control charts, where the process being measured is expected or desired to have a consistent output each time, every time.

In this paper, I describe the use of control charts during testing phases of software development projects. This use is not to determine if the testing is in control, nor is it in order to improve product quality (although that has also been done [3, 4]), but rather to determine when there has been a shift in quality. This is in order to improve mapping of project progress to forecast curves and thereby improve estimates.

Although some purists may raise objections to this application of control charts, I would quote Shewhart himself:

"…In other words, the fact that the criterion which we happen to use has a fine ancestry of highbrow statistical theorems does not justify its use. Such justification must come from empirical evidence that it works" [5].

Therefore, let us look at a typical example.

Using Control Charts to Detect Signals

In order to improve defect forecasts, I use Individuals and Moving Range charts (XmR). This is a type of control chart that is suitable for most real time situations, including the collection of periodic data such as defects detected in a given time period (such as week or month). The Individuals chart has each value plotted in time order. The Moving Range chart, on the other hand, plots the short-term variation from one period to the next.

While most signals appear on the individuals chart, it is good practice to look at the moving range chart as well, as some signals will only show up on it.

Control limits are based on the long-term average value as well as the average moving range value of one point to the next. It is important to calculate control limits correctly in order to not miss valid signals. The appropriate formulas can be found in select books [6, 7] and are also built into statistical tools such as SPSS, Minitab and SAS.

Control limits provide a signal of sporadic or chronic problems. For tracking defects, however, the signal we are looking for is a change in the underlying quality of the software product. Hopefully, this will be a signal of an improvement and not a signal of a problem!

Rules

There are a number of rules that are used to detect signals. The number of rules used and the definitions of the rules vary slightly from one source to another. However, the traditional use of control charts is best met by keeping the number of rules to a minimum, thereby reducing the chance of obtaining a false signal. In this application, however, it is more important to err on the side of obtaining a false signal rather than missing a true signal. All uses of control charts walk this decision line. Shewhart originally used 3 sigma limits because he wanted to minimize false signals, which would incur the unnecessary cost of researching a problem that didn't exist. In other words, when he saw a signal he wanted to be almost completely certain it was real.

In IBM SPSS 22, for example, there are 11 possible rules that can be turned on or off:

- One point above +3 Sigma, or one point below -3 Sigma
- 2 out of last 3 above +2 Sigma, or 2 of 3 below -2 Sigma
- 4 out of last 5 above +1 Sigma, or 4 out of 5 below -1 Sigma
- 8 points above center line, or 8 below center line
- 6 in a row trending up, or 6 trending down
- 14 in a row alternating up and down

Example Control Charts

In Figure A, weekly defects detected are plotted. All the SPSS rules are turned on. If the defect detection rate has changed significantly, that would show up as a special cause signal in the control chart. In this example, the balance between testing and fixing has not remained constant. Five of the points show up as red, meaning they violated one of the rules (see Table 1).
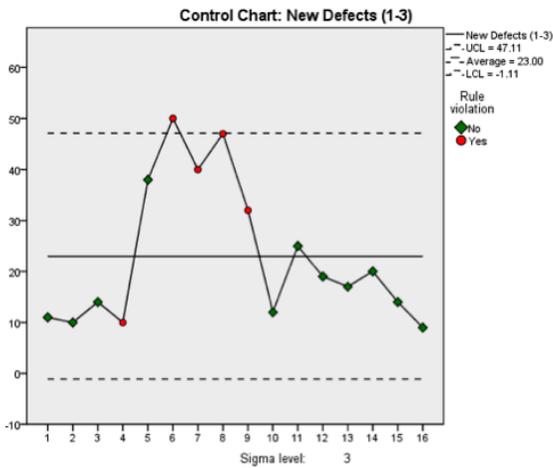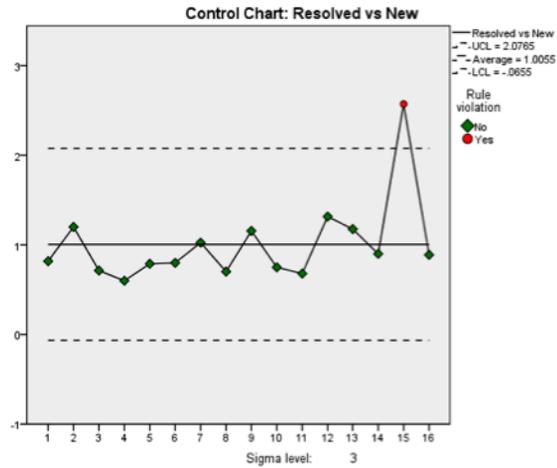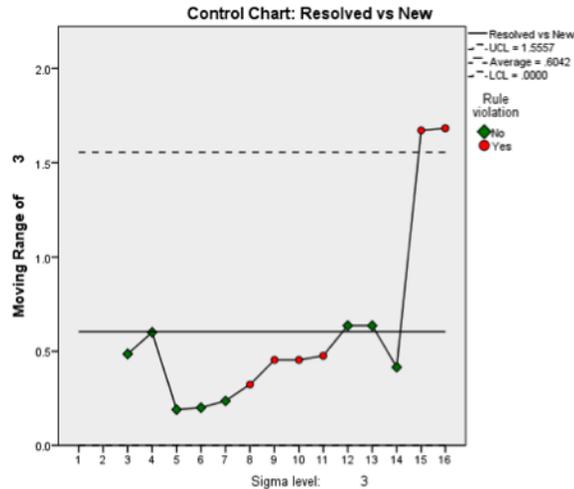
Figure A: Control Chart Example

Point #	Violations for Points
4	4 points out of the last 5 below -1 sigma
6	Greater than +3 sigma
6	2 points out of the last 3 above +2 sigma
7	2 points out of the last 3 above +2 sigma
8	4 points out of the last 5 above +1 sigma
9	4 points out of the last 5 above +1 sigma

Table 1: Rule violations

What can we surmise from this? These violations are not unusual. As mentioned previously, defect metrics commonly follow a Rayleigh distribution. In Figure B, actual defects detected monthly are overlain on a defect forecast based on the current project plan (a parametric SLIM Control forecast based on historical defect rates and project type, size, staff, and duration). We can see the peak detected as a set of rule violations falls in line with the peak of the Rayleigh curve.

One important question is whether the drop in defects in the last few weeks is a signal that a turning point on the project has been reached, as the Rayleigh curve suggests. Control charts can be used to help verify that the signal is real and not random noise.

Figure B: Rayleigh Example

Figures C and D: Individual and Moving Range Chart

Figures C and D are Individual and Moving Range charts for the ratio of defects discovered to defects resolved. This ratio measures the balance between defect detection in testing and defect repair. Values in the individuals chart greater than 1 indicate more defects were resolved than were detected during that week. Both charts show rule violations. Point 15 provides evidence that the balance has shifted, supporting the conclusion that the project is truly on the downslope of the Rayleigh Curve.

Defect Prediction

The plan in Figure B was based on parametric estimating. It is possible to create very useful estimates of defects based on only a few key metrics. For example, I created a regression analysis to predict defects based on over 2000 recently completed software projects from the QSM database. This resulted in an adjusted R square of .537 using only the input variables the log of peak staff, the log of ESLOC, and the log of production rate (ESLOC per calendar month). The output variable is log of defects. (Why logs? For the explanation, see [8].) The standardized residuals are plotted on a histogram in Figure E. As can be seen, the residuals have a normal distribution with mean close to zero. The model is not skewed.
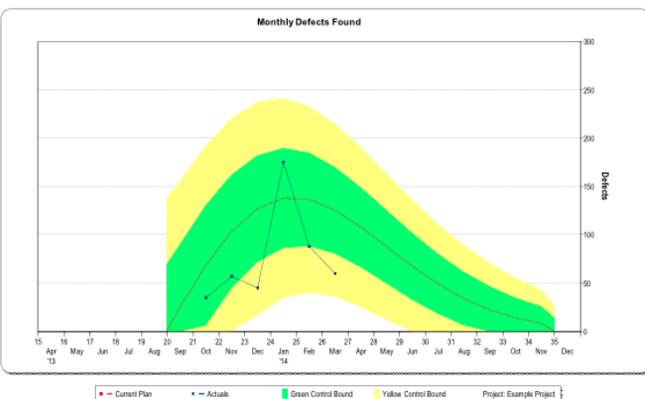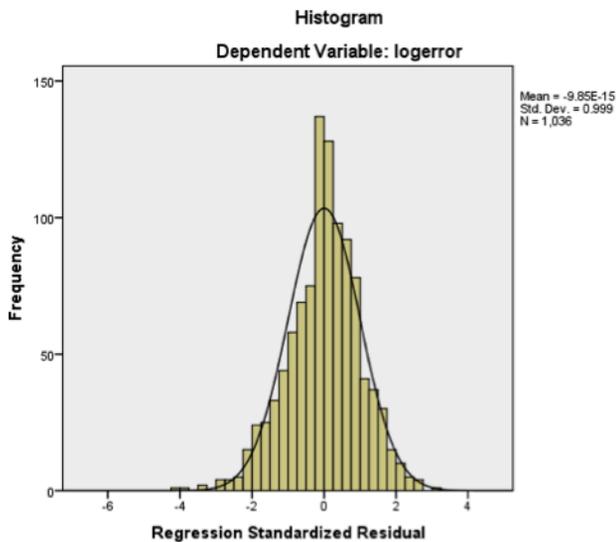
Figure E: Standardized Residuals

Large projects have multiple testing phases. Such models, with multiple control charts, can be used throughout. For example, with one Fortune 500 client, I found that merely using the number of prerelease defects was an excellent predictor of their go live release defects (R Square of over 0.7).

Summary

Control charts can be used to determine whether apparent changes in defect rates are significant. One use for this knowledge is to create and improve forecasts of Program completion, or software quality at key Program milestones. Shewhart gave us this thought regarding updating forecasts:

"…since we can make operationally verifiable predictions only in terms of future observations, it follows that with the acquisition of new data, not only may the magnitudes involved in any prediction change, but also our grounds for belief in it" [9].

ABOUT THE AUTHOR

Paul Below has over 30 years of experience in technology measurement, statistical analysis, estimating, Six Sigma, and data mining. He is a Principal Consultant with Quantitative Software Management, Inc. (QSM) where he provides clients with statistical analysis of operational performance, process improvement and predictability.

This is his second article for Crosstalk, and he is co-author of the IFPUG Guide to IT and Software Measurement (CRC Press, 2012). He has developed courses and been an instructor for estimating, Lean Six Sigma, metrics analysis, function point analysis, and also taught metrics for two years in the Masters of Software Engineering Program at Seattle University. He has presented papers at a dozen industry conferences.

Paul is a Certified SLIM Estimation Professional, and has been a Certified Software Quality Analysis and a Certified Function Point Analyst. He is a Six Sigma Black Belt, and has one US Patent. He is a member of IEEE, the American Statistical Association (ASA) and the National Defense Industrial Association (NDIA).

Quantitative Software Measurement, Inc. (QSM)
<http://www.qsm.com>
Phone: 800-424-6755, 703-790-0055
E-mail: paul.below@qsm.com

REFERENCES

1. Statistical Method from the Viewpoint of Quality Control, Walter A. Shewhart. Dover, 1986 edition, p.9.
2. Five Core Metrics: The Intelligence Behind Successful Software Management, Lawrence H. Putnam and Ware Myers. Dorset House, 2003, Chapter 13.
3. Why CMMI Maturity Level 5?, Michael Comps. Crosstalk, Jan-Feb, 2012, pp 15-18.
4. Do Not Get Out of Control: Achieving Real-time Quality and Performance, Craig Hale and Mike Rowe. Crosstalk, Jan-Feb 2012, pp. 4-8.
5. Economic Control of Quality of Manufactured Product, 50th Anniversary Commemorative Reissue, Walter A. Shewhart. ASQC, 1980, p. 18.
6. Understanding Statistical Process Control, Donald J. Wheeler. SPC Press, 2010.
7. Implementing Six Sigma: Smarter Solutions Using Statistical Methods, Second Edition, Forrest W. Breyfogle III. John Wiley & Sons, 2003.
8. The IFPUG Guide to IT and Software Measurement: A Comprehensive International Guide, IFPUG, ed. CRC Press, 2012. Chapter 17, Paul Below, pp. 319-333.
9. Statistical Method from the Viewpoint of Quality Control, Walter A. Shewhart. Dover, 1986 edition, p.104.