ON APPLICATION OF STATISTICAL PROCESS CONTROL TO SOFTWARE ENGINEERING

Chakhunashvili Alexander
Skaraborg Hospital Group, Skövde, Sweden

Summary

While statistical methods for quality improvement have found broad application in traditional production industries (e.g. automotive, electronics, chemical and food-processing industries), their use in the software engineering field have been rather limited. There are, however, examples showing that the statistical quality improvement methods can equally well be applied to software engineering, or alternatively to larger IT application development. In this paper we review one important area of statistical quality improvement; namely, Statistical Process Control (SPC) and discuss its application in the software engineering context.

Key words: Quality, unwanted variation, Statistical Process Control (SPC), control chart, software engineering, software development process, software metrics.

1. Introduction

With the increased share of software in today's manufactured products demands on its quality and reliability are increasing as well. To meet these increasing demands it is not sufficient to focus on the software product alone, but also on the process facilitating the development of that software product i.e. the software development process; see e.g. Humphrey (1995) and Florac & Carleton (1999).

Software engineering, as a scientific discipline, has undergone substantial transformation from a manual machine code writing in the 1940s to object-oriented programming in the 1990s to today's managed code platforms such as Java, .NET and PHP. Furthermore, a number of software development models, including the Waterfall model (see e.g. Royce (1970)), incremental/iterative models (see e.g. Basili & Turner (1975)), the Spiral model (see e.g. Boehm (1988)), Rapid Application Development (see e.g. Martin (1991)) and the Rational Unified Process (see e.g. Kruchten (1998)) have evolved as a means to help develop software products in a structured and systematic way. Nevertheless, application of statistical quality improvement methods to the software development process has been rather limited. In the meantime, automotive, electronics, chemical and food-processing industries have benefited greatly from the use of statistical methods in their efforts to achieve high product quality and reliability.

The purpose of this paper is not to investigate the reasons behind the limited application of statistical quality improvement methods to software development processes. The purpose is rather to show some examples of application of Statistical Process Control (SPC), an important area of statistical quality improvement, to software development processes and discuss its application in the software engineering context.

2. Quality and Unwanted Variation

Since quality in industry is often viewed as inversely proportional to unwanted variation (see e.g. Montgomery (2001)), the attainment of high product quality depends largely on our ability to manage variation in various production and product development phases, including product planning, product/process development, serial production, sales, after-sales services and recycling. During this process, a thorough understanding of sources of variation, the means to detect, identify and eliminate them, and the actions required to minimize their impact on critical product, process or system characteristics is essential for delivering the product quality we promise to our customers. In addition to improved quality, reduced variability leads to fewer repairs, less re-work, and minimum waste, thus decreasing the total product realization cost; see Montgomery (2001) and Thornton (2004). A similar interpretation of quality is also found in Deming (1986), in which he compares the effect of quality improvement with productivity as a result of variation reduction. Furthermore, Deming describes this relationship as a chain reaction, leading to company's long-term sustainability and success in business.

Finding and eliminating sources of variation and minimizing the impact that they have on important product characteristics are two variation management strategies that companies deploy to cut down development, production, usage and recycling costs; yet deliver high quality and reliable products; see Abraham and Brajac (2001) and Thornton (2004). This thinking has also been reflected in different company-wide quality improvement initiatives, some later examples of which are Six Sigma, see e.g. Magnusson *et al*. (2003), Design for Six Sigma, see e.g. Watson (2005), and Lean Manufacturing, see e.g. Womack *et al*. (2007).

3. Statistical Process Control

Walter A. Shewhart was one of the first to understand the benefits of reducing variation in the manufacturing industry. He realized that although customers had varying needs and wants they would not appreciate variability in units of products manufactured to the same specifications. Thus, efforts should be directed to reducing variation between the units of the same type of product; see Shewhart (1931). However, this seems to be meaningless if all sources of variation are small relative to the total variation of the units. When no single source of variation is dominant, Shewhart says that the process is in a state of statistical control. He refers to these non-dominant sources of variation as "chance causes of variation". If, however, one or more sources of variation dominate over the others, the process is out of statistical control, and it may be economically feasible to find and eliminate the dominant sources of variation. Shewhart calls these dominant sources "assignable causes of variation". To detect assignable causes of variation, Shewhart proposed five criteria. These criteria, in simpler forms, are also found in different quality improvement techniques, such as the Japanese Seven Quality Control (7QC) tools; see Shewhart (1931) and Chakhunashvili (2006) for further details on

Shewhart's criteria for detecting the presence of assignable causes and Bergman & Klefsjö (2003) for a thorough review of the 7QC tools.

Shewhart's ideas on variation reduction formed the basis of SPC. As an important part of improvement work, the goal of SPC is to find assignable causes of variation and eliminate them as soon as possible; see Bergman & Klefsjö (2003). One of the most commonly used tools of SPC is the control chart. We illustrate the application of the Shewhart type of control chart in Section 3.1.

3.1 The Shewhart Control Charts

The idea of employing a control chart is to graphically monitor the variability of a product characteristic over time. Shewhart suggested the first control chart in 1924 when he studied variation in the measured qualities of products manufactured for Bell Systems; see Juran (1997). It was a control chart constructed to monitor the fraction of defective units. Following the Shewhart control chart numerous other control charts evolved over the decades. They can formally be grouped into two relatively larger subgroups depending on the type of data monitored. The first group includes control charts for variable data, also called continuous data measurements, and the second group includes control charts for attribute data, also called discrete data measurements. Furthermore, there are two measures usually monitored in a process, namely, the sample mean and the sample standard deviation (alternatively the sample range). The mean shows how close to the target the process performs while the standard deviation (alternatively the range) describes the dispersion of the process. Both changes in the process mean and increased dispersion may lead to undesired consequences. Therefore, it is often desirable to monitor not only the process mean, but also the process dispersion. Furthermore, some control charts may be more sensitive to changes in the process than others, and some control charts may treat observations individually while the other control charts might utilize smaller groups called samples. These and some other aspects of the charting technique are important to take into consideration when constructing a control chart.

A typical Shewhart control chart consists of a centerline (CL), normally corresponding to the process mean, and two control limits, the upper control limit (UCL) and lower control limit (LCL). A control chart should be designed so that it can quickly signal an alarm (the plotted measurement outside the control limits) if unusual behavior is detected in the observed characteristic. However, the number of false alarms, i.e. indications of assignable causes when they actually do not exist, should be as rare as possible. To meet these requirements control limits set at $\mu \pm 3\sigma$, where μ is the process mean and σ is the standard deviation, appear to be adequate. These limits are sometimes also referred to as the three-sigma limits. However, values other than three can also be considered depending on the objective of the monitoring procedure. Furthermore, the choice of three-sigma has economic reasons. According to Shewhart (1931), if the measured quantity of a quality characteristic deviates more than three-sigma times from its expected value, it is usually economically feasible to look for the causes of variation and try to eliminate them.

As an illustration of how the outcome of a typical control chart is interpreted, let us take a look at an $\bar{x}$ control chart shown in Figure 1. For details on how this control chart is constructed the reader is referred to Bergman & Klefsjö (2003), Wheeler & Chambers (1992) or Montgomery (2001), which provide a thorough explanation of the charting technique.

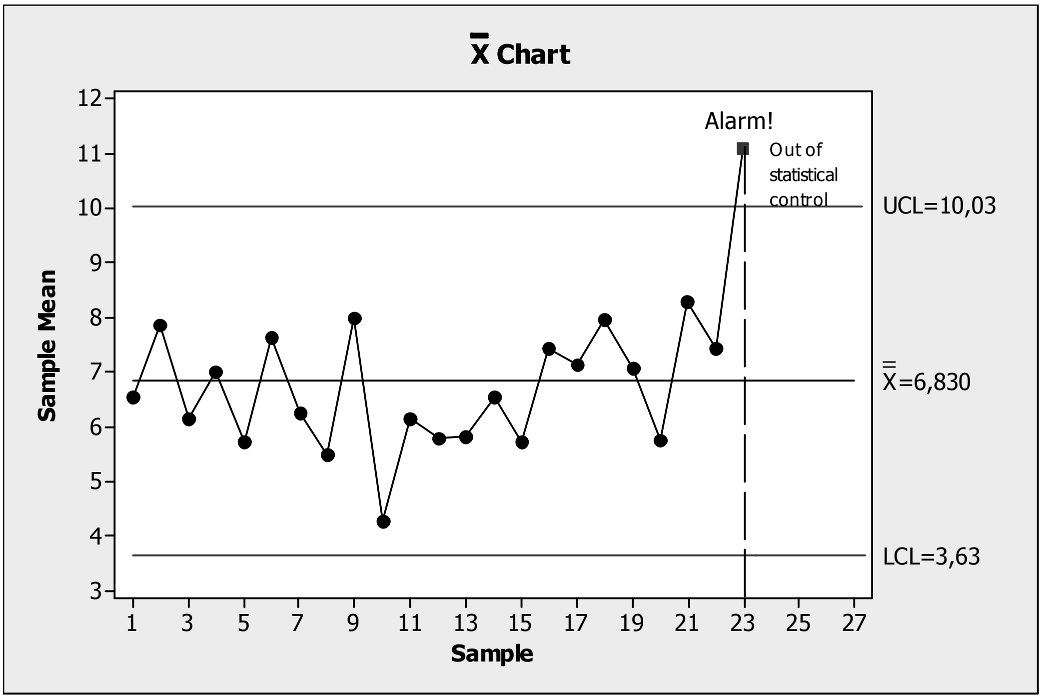

Figure 1. An $\bar{x}$ control chart

Although the initial observations are well within control limits, a point outside the UCL at sample 23 indicates that the process is influenced by assignable cause(s) of variation. Hence, the process is out of statistical control. The next step is to identify the assignable cause(s) and eliminate it/them. The elimination of an assignable cause considerably decreases the total process variation, thus improving the process stability and predictability.

While the judgment of the state of statistical control in the illustration above is based on a simple rule, a single observation outside the three-sigma limits, more systematic rules are needed in some situations to determine whether the process is or is not in statistical control. In this respect, it is worth mentioning the rules devised by Western Electric (1956). According to these rules, even two out of three consecutive points lying beyond the two-sigma limits, four out of five consecutive points lying beyond the one-sigma limit and eight consecutive points lying on one side of the target value would be considered as an indication of lack of statistical control. It should be kept in mind, however, that when applying these rules, the frequency of false alarms might increase.

One of the most important properties of a control chart is its ability to react to changes in the process. This property is also called the sensitivity of a control chart and can be assessed by the Average Run Length (ARL). The ARL shows how long it takes for the control chart to signal an alarm from the point at which a systematic change in the process has occurred. The ARL value depends on the size of the change and is normally measured in terms of the number of runs or time intervals; see Montgomery (2001) and Bergman & Klefsjö (2003). The ARL values can be computed using either probabilistic methods such as the Markov chain approach or by means of Monte Carlo simulations. While reliable ARL values using a probabilistic approach can be obtained analytically in relatively short time, Monte Carlo simulations offer other advantages including the possibility of varying the control chart parameters in a random manner.

3.2 The EWMA and CUSUM Control Charts

Some production processes are characterized by *stationary* variation, that is, variation observed about a fixed mean. Furthermore, variation in a process may be of different sizes. While the Shewhart control chart is effective in detecting larger shifts in order to promptly detect the presence of smaller variations (about 1.5 σ and less), some specific control charts may be required. Roberts (1959) formally introduced the Exponentially Weighted Moving Average (EWMA) control chart as an alterative to the Shewhart control chart, especially for monitoring processes with small variations. Its performance is often compared to that of the cumulative sum (CUSUM) control chart and is generally considered easy to implement and maintain; see e.g. Montgomery (2001). The CUSUM control chart is constructed by plotting the accumulated sum of the deviations from the target value; see Page (1961) and Evan (1963). Like the EWMA control chart the CUSUM control chart is also effective in detecting small changes in a process. The key point in differentiating EWMA and CUSUM control charts is the way each of them handles past data. The CUSUM places equal weights on all past observations whereas the EWMA gives greater weight to more recent observations and lesser weight to older ones. In this way, the EWMA quickly forgets the history and pays more attention to the nearest past. Finally, it should be mentioned that the Western Electric rules are not recommended to apply to the EWMA and CUSUM control charts since this might increase the false alarm rate due to the correlations observed between the consecutive EWMA and CUSUM values, respectively.

3.3. Multivariate Control Charts

When several variables influence the process outcome it is often advantageous to monitor them with one single statistic instead of constructing a number of individual control charts, one for each variable. There are two major rationales in doing so. Firstly, it is easier to operate a single control chart than several of them and secondly, and perhaps more importantly, a multivariate control chart can detect systematic changes that an individual control chart cannot. This approach is especially useful when it comes to monitoring related variables. Thus, when designing a multivariate control chart, particular attention should be paid to the relationship between the observed variables. Furthermore, if the number of original variables is too large, numerous multivariate techniques, such as factor analysis, are often used to reduce the number of monitored variables. A detailed review of the multivariate control charts is beyond the scope of this paper. The interested readers are therefore referred to Hotelling (1947), Lowry *et al.* (1992) and Montgomery (2001).

4. Applying Statistical Process Control Tools to Software Engineering

To illustrate an application of a control chart, one of the most commonly used SPC tools, to a software process, let us take a look at the following example. A medium size software company, specializing in developing parsers[1], has decided to monitor its source code programming process. To do that the company has decided to look at the defect density defined as the number of defects (e.g. programming errors adversely affecting software functionality) in a software module divided by the module size in LOC (Lines of Code). Initially 20 software modules have been selected and defect densities have been plotted against the sequence of modules. The resulting control chart is shown in Figure 2. As we see, the process is not in statistical control as Module 13 and Module 20 have unusually high defect densities (the plotted points outside the UCL). Furthermore, the average defect density in a software module is about 2.4 percent, a certainly unacceptable figure for the company. As such, the company decides to investigate the causes for the high defect density and eliminate them as soon as possible.

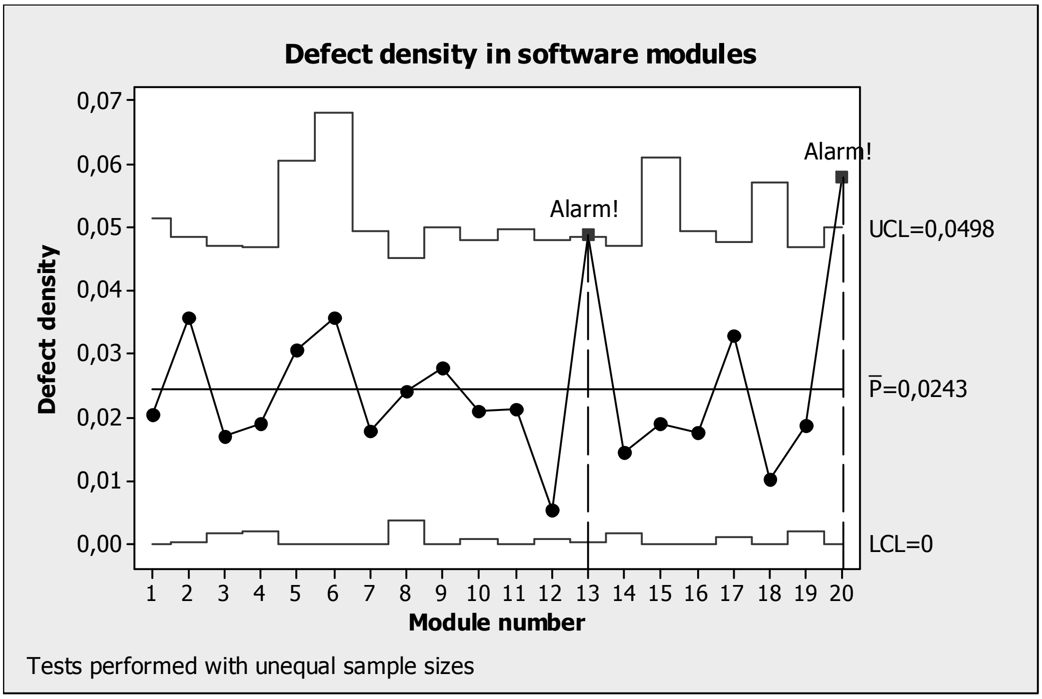

Figure 2. Defect density in software modules

As a step one they look at Module 13 and Module 20 and find that the programming teams responsible for those two modules have a number of newcomers, perhaps in the need of an introductory training. Instigating the newcomers' introduction-to-the-company training puts the source

[1] Syntactic data analyzers used in e.g. data interpreters and compliers.

code programming process back in a state of statistical control as well as decreases the defect density from 2.4 percent to 1.9 percent; see Figure 3.

Consequently, the company is happy to be able to find and eliminate assignable causes of variation and thereby attain the process in statistical control. However, it is not satisfied with the fact that the defect density is still quite high (1.9 percent). To further improve the source code programming process the company has to consider making system changes as the remaining variability of the process can only be attributed to the chance causes of variation. Such a system change might include the introduction of the company-wide coding and design standards, regular design reviews and systematic follow ups on the deviations from the introduced standards.

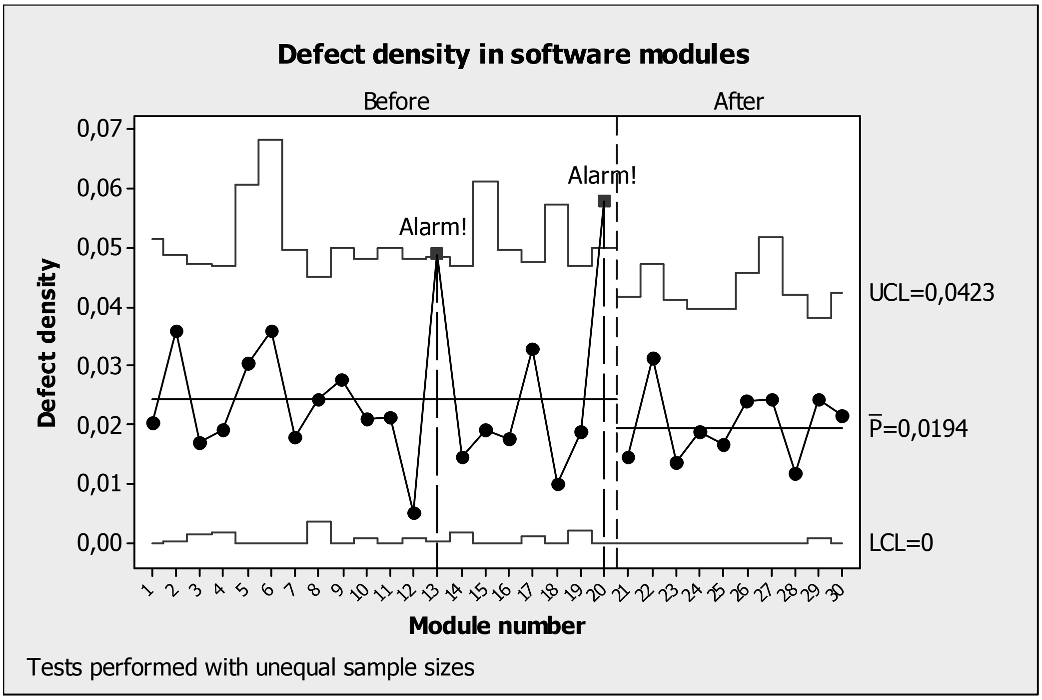

Figure 3. Defect density in software modules before and after instigating the introductory training

Although the example presented above is a simplification of reality, without taking into consideration all the complexity of the software development process, it shows how the control charting technique can be applied to a software process for improvement purposes. The detection of the assignable causes is usually done by the control chart. However, other tools including Cause and Effect Diagram, Five Whys and Stratification are often used to find and identify the true causes of variation.

Defect density, a quality indicator plotted on the control chart above, is just one of the so called indirect measures used to monitor a software process. Furthermore, a number of other indirect measures including Programmer Productivity defined as LOC produced divided by person month of effort, Defect Detection Efficiency defined as the number of defects detected divided by the total number of defects are also used to monitor different kinds of software processes. For further details about direct and indirect software measures see Fenton & Pfleeger (1997).

5. Discussion

There have been numerous efforts made to formalize and systematize the software development process over the past couple of decades including the development of Capability Maturity Model (CMM/CMMI) and its supporting processes, Personal Software Process (PSP) and Team software Process (TSP); see e.g. Humphrey (1995) and Humphrey (2000). These models and processes not only promote the use of statistical quality improvement tools in software engineering, but also make it easier to do so as they take the process view approach to software development. Both PSP and TSP provide numerous templates and guidelines to assist software practitioners in developing software products in a systematic and cost-effective way.

From a software quality standpoint, in addition to CMM/CMMI, it is interesting to take a look at the following software development models: Agile Software Development (see e.g. Shore & Warden (2007)) - employing a flexible software development model based on shorter and rather frequent iterations; Extreme Programming (see e.g. Beck (2000)) - another popular light-weight methodology promoting source code programming in pairs, writing unit tests before programming and keeping close contact with the customer all along the development process; Cleanroom Software Engineering (see e.g. Mills *et al.* (1987)) – a software development methodology challenging many of the traditional beliefs with regard to software engineering including the deterministic nature of software and thereby the limited role and use of statistical methods, unfeasibility of software fault prediction/avoidance as well as focus on extensive source code debugging.

Finally, it should be noted that although in this paper we mainly have focused on SPC, there are numerous other quality methods, of both quantitative and qualitative nature, applicable to software engineering. These methods include Design of Experiments (see e.g. Box *et al.* (1978)) - especially useful in the context of software testing, Quality Function Deployment (see e.g. Hauser & Clausing (1988)) – a methodology aimed at systematic collection of customer data and their translation into product or process design parameters, Failure Mode and Effect Analysis (see e.g. Stamatis (1994)) – used to identify and assess the risks related to products and processes subject to improvement.

References

1. Abraham, B. & Brajac, M. (2001). "Variation Reduction and Robust Design." Commun. Statist.-Theory Meth. 30(8-9): 1951-1962.
2. Basili, V. & Turner, A. (1975). “Iterative Enhancement: A Practical Technique for Software Development”. IEEE Transactions on Software Engineering. Vol. 1, No. 4, pp. 390-396.
3. Beck, K. (2000). *Extreme Programming Explained. Embrace Change*. Addison-Wesley.
4. Bergman, B. & Klefsjö, B. (2003). *Quality from Customer Needs to Customer Satisfaction.* Lund: Studentliteratur.

5. Boehm, B. W. (1988). “A spiral model of software development and enhancement”. Computer. Vol. 21, Issue 5, May 1988 Page(s):61 – 72.
6. Box, G.E.P., Hunter, W.G. & Hunter, J.S. (1978). *Statistics for Experimenters: An Introduction to Design, Data Analysis, and Model Building*. John Wiley & Sons, Inc.
7. Chakhunashvili, A. (2006). *Detecting, Identifying and Managing Sources of Variation in Production and Product Development*. PhD Thesis. Chalmers University of Technology. Gothenburg, Sweden. ISSN 0346-718X.
8. Deming, E. W. (1986). *Out of the Crisis*, The MIT Press, Cambridge, Massachusetts.
9. Evan, W. D. (1963). "When and How to Use CuSum Charts." Technometrics 5.
10. Fenton, E. N. & Pfleeger, S. L. (1997). *Software Metrics. A Rigorous & Practical Approach*. PWS Publishing Company.
11. Florac, A.W. & Carleton, A. D. (1999). *Measuring the Software Process*. SEI Series in Software Engineering. Addison-Wesley.
12. Hauser, J.R. & Clausing, D. (1988). “The House of Quality”. The Harvard Business Review. May-June, 66-73.
13. Hotelling, H. (1947). "Multivariate Quality Control--Illustrated by the Air Testing of Sample Bombsights." Techniques of Statistical Analysis, eds. C. Eisenhart, M. W. Hastay, and W. A. Wallis, New York: McGraw-Hill: 111-184.
14. Humphrey, W.S. (1995). *A Discipline for Software Engineering*. SEI Series in Software Engineering. Addison-Wesley.
15. Humphrey, W.S. (2000). *Introduction to the Team Software Process*. SEI Series in Software Engineering. Addison-Wesley.
16. Juran, J. M. (1997). "Early SQC: A historical Supplement." Quality Progress (September).
17. Kruchten, P. (1998). *The Rational Unified Process: An Introduction*. Addison-Wesley Longman, Inc.
18. Lowry, C. A., Woodall, W. H., Champ, C. W. & Rigdon, S. E. (1992). "Multivariate exponentially weighted moving average control chart." Technometrics 34(1): 46.
19. Magnusson, K., Kroslid, D. & Bergman, B. (2003). *Six Sigma, The Pragmatic Approach*. Studentlitteratur.
20. Martin, J. (1991). *Rapid Application Development*, Macmillan.
21. Mills, H.D., Dyer, M. & Linger R. (1987). “Cleanroom Software Engineering”. IEEE Software, 4 (5): 19–25.
22. Montgomery, D. C. (2001). *Introduction to Statistical Quality Control*, John Wiley & Sons, Inc.
23. Page, E. S. (1961). "Cumulative Sum Control Charts." Technometrics 3.
24. Roberts, S. W. (1959). "Control Chart Tests Based on Geometric Moving Averages." Technometrics 1.
25. Royce, W.W. (1970). "Managing the Development of Large Software Systems", Proceedings of IEEE WESCON 26 (August): 1-9.
26. Shewhart, W. A. (1931). *Economic Control of Quality of Manufactured Product*. New York, Van Nostrand Company, Inc.
27. Shore, J. & Warden, S. (2007). *The Art of Agile Development*. O'Reilly.
28. Stamatis, D.H. (1994) *Failure Mode and Effect Analysis: FMEA from Theory to Execution*. ASQ Quality Press, Milwaukee.
29. Thornton, A. C. (2004). *Variation Risk Management: Focusing Quality Improvements in Product Development and Production*, Wiley.
30. Watson, G.H. (2005). *Design for Six Sigma: Innovation for Enhanced Competitiveness*. GOAL/QPC, Publisher.
31. Western Electric (1956). *Statistical Quality Control Handbook*. Western Electric Corporation Indianapolis, IN, USA.
32. Wheeler, D. J. & Chambers, D. S. (1992). *Understanding Statistical Process Control*, SPC Press.
33. Womack, J.P., Jones D. T. & Roos D. (2007). *The Machine that Changed the World*. Free Press. New York, NY, USA.

ხარისხის მართვის სატატისტიკური მეთოდიკის გამოყენება პროგრამული უზრუნველყოფის შექმნის პროცესში

ალექსანდრე ჩახუნაშვილი
სკარაბორგის ჰოსპიტალების ჯგუფი, სკოვდე,
ჩალმერსის ტექნოლოგიური უნივერსიტეტი, გოტენბურგი (შვედეთი)

რეზიუმე

მიუხედავად იმისა, რომ ხარისხის მართვის სტატისტიკური მეთოდები ფართოდ გამოიყენება წარმოების ტრადიციულ დარგებში (მაგალითად, საავტომობილო, ელექტრონიკის, ქიმიური და კვების მრეწველობის დარგები), მათი გამოყენება პროგრამული უზრუნველყოფის ხარისხის გასაუმჯობესებლად საკმაოდ მწირია. თუმცა არსებობს მაგალითები, რომლებიც ცხადყოფს, რომ ხარისხის მართვის სატატისტიკური მეთოდები შესაძლებელია საკმაოდ წარმატებულად იქნას გამოყენებული პროგრამული უზრუნველყოფის შემუშავების პროცესში. ამ სტატიაში ჩვენ მიმოვიხილავთ ხარისხის სტატისტიკური მართვის ერთ-ერთ მნიშვნელოვან სფეროს, კერძოდ, პროცესის სტატისტიკურ მართვას და განვიხილავთ მისი გამოყენების შესაძლებლობებს პროგრამული უზრუნველყოფის შექმნის კონტექსტში.

ИСПОЛЬЗОВАНИЕ СТАТИСТИЧЕСКОЙ МЕТОДИКИ УПРАВЛЕНИЯ КАЧЕСТВОМ В ПРОЦЕССЕ СОЗДАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Чахунашвили А.
Группа больниц Скараборга, Сковде (Швеция),
Технологический Университет Чальмерса, Готенбург

Резюме

Несмотря на то, что статистические методы управления качеством широко используются в традиционных отраслях производства (например, отрасли транспорта, электроники, химической и пищевой промышленности и т.д.), их применение с целью совершенствования качества программного обеспечения довольно ограничено. Однако имеются примеры, которые показывают, что статистические методы управления качеством программного обеспечения можно довольно эффективно использовать в процессе их создания. В данной работе рассматривается одна из важнейших сфер статистического управления качеством, в частности статистическое управление процессами и обсуждаются вопросы возможности их применения в контексте разработки программного обеспечения.

Brief biographical notes about author:

Alexander Chakhunashvili is a Master Black Belt at the Skaraborg Hospital Group in Skövde, Sweden. He earned his Ph.D. degree at the Division of Quality Sciences at Chalmers University of Technology in Gothenburg, Sweden, in 2006. He also holds a M.Sc. in management of production from Chalmers University of Technology and a B.Sc. in Automated Control Systems from Georgian Technical University in Tbilisi, Georgia. His research interests are statistical process control/improvement, variation management and robust design. The ideas presented in this paper are based on his research at Chalmers University of Technology.