Software Qual J (2013) 21:479–499
DOI 10.1007/s11219-012-9182-y

Monitoring the software development process using a short-run control chart

Chih-Wei Chang · Lee-Ing Tong

Published online: 22 July 2012

Abstract Techniques for statistical process control (SPC), such as using a control chart, have recently garnered considerable attention in the software industry. These techniques are applied to manage a project quantitatively and meet established quality and process-performance objectives. Although many studies have demonstrated the benefits of using a control chart to monitor software development processes (SDPs), some controversy exists regarding the suitability of employing conventional control charts to monitor SDPs. One major problem is that conventional control charts require a large amount of data from a homogeneous source of variation when constructing valid control limits. However, a large dataset is typically unavailable for SDPs. Aggregating data from projects with similar attributes to acquire the required number of observations may lead to wide control limits due to mixed multiple common causes when applying a conventional control chart. To overcome these problems, this study utilizes a *Q* chart for short-run manufacturing processes as an alternative technique for monitoring SDPs. The *Q* chart, which has early detection capability, real-time charting, and fixed control limits, allows software practitioners to monitor process performance using a small amount of data in early SDP stages. To assess the performance of the *Q* chart for monitoring SDPs, three examples are utilized to demonstrate *Q* chart effectiveness. Some recommendations for practical use of *Q* charts for SDPs are provided.

Keywords Software development process · Statistical process control · Control chart · Short production run · *Q* chart

C.-W. Chang (✉) · L.-I. Tong
Department of Industrial Engineering and Management, National Chiao Tung University, 1001, Daxue Rd., Hsinchu City 300, Taiwan, ROC
e-mail: scott.iem95g@nctu.edu.tw; scott.nctu@gmail.com

L.-I. Tong
e-mail: litong@cc.nctu.edu.tw

1 Introduction

Some process improvement models, such as the Capability Maturity Model (CMM), Software Process Improvement and Capability Determination model (ISO/IEC 15504), and Capability Maturity Model Integration (CMMI) model, have been widely employed by software organizations to improve the quality of their software development process (SDP). Quantitative management is commonly utilized to monitor and control process performance to reduce, eliminate, or prevent deficiencies in software quality. More specifically, a software product should be developed by a stable SDP that is quantitatively managed to meet or exceed customer expectations. Consequently, product quality, service quality, process performance, and other business objectives for a quantitatively managed process are controlled throughout the software development life cycle (Kulpa and Johnson 2008). However, variation always exists in an SDP, regardless of how carefully processes are managed. Software measurement data only provide quantitative information about process performance, not information about process stability. Without systematic methods for analyzing measurement data, interpreting measurement results for further decision making is typically difficult. Thus, an effective approach to analyze and control process variation in SDPs is indispensable for the software process improvement. Statistical process control (SPC), a powerful collection of process-improving tools, is an essential quantitative management technique for measuring, controlling, and reducing the magnitude of process variations. Shewhart control charts (Shewhart 1926) are the most commonly utilized SPC tools by the manufacturing industry to determine whether a process is stable. A control chart can be utilized to differentiate between an abnormal signal and noise (inherent or random variation) using measurement data for reducing process variation and achieving process stability. For example, $\overline{X}$ and R ($\overline{X} - R$) charts are typically used to monitor the mean and variability of a process, and the X and MR ($X - MR$) charts are commonly used to monitor individual measurements.

Control charts have recently garnered considerable attention in the software industry. Several studies have identified the benefits of using SPC methods to improve software quality and have demonstrated the success in using a control chart to monitor SDPs (Chang and Chu 2008; Diane and Stephen 2007; Florac and Carleton 1999; Jacob and Pillai 2003; Jalote and Saxena 2002; Komuro 2006; Weller 2000). However, some controversy exists regarding the suitability of using SPC techniques in the software industry. Particularly, challenges to the implementation of control charts in SDPs have been discussed in many studies (Baldassarre et al. 2004; Caivano 2005; Komuro 2006; Manlove and Kan 2007; Raczynski and Curtis 2008; Radice 1998; Sargut and Demirörs 2006; Tarhan and Demirörs 2006; Weller and Card 2008). Table 1 summaries the challenges identified in these studies when implementing SPC in SDPs. Accordingly, intensive human activities, diversity metrics, multiple common causes, and a small amount of process data are software industry characteristics that cannot be altered easily for specific quantitative techniques. Moreover, Shewhart control chart requires a large amount of data from a homogeneous source of variation when constructing valid control limits. But a large dataset is typically unavailable for SDPs. Therefore, rather than applying conventional control charts (i.e., the Shewhart control chart) directly to process data from the software industry, modified SPC approaches are needed to deal with such issues.

Job-shop and just-in-time (JIT) systems are used in the manufacturing industry in response to different customer demands and short product life cycles (Castillo et al. 1996; Castillo and Montgomery 1994). These production runs are called “short production runs”

Table 1 Summary of challenges identified in the literature when implementing SPC in SDPs

Authors	Description
Baldassarre et al. (2004)	Fundamental differences exist between software development and manufacturing processes Too many attributes and variables exist in the software life cycle
Caivano (2005)	Difficulty in choosing suitable indicators for implementing SPC in SDPs
Komuro (Komuro 2006)	Characteristics of SDP (e.g., human-intensive and process-centric) Multiple common causes of variation in SDPs. Difficulty obtaining a large set of homogeneous data
Manlove and Kan (2007)	Software is produced by people, not machines Too many sources of variability in the software development environment Multiple common causes of variation in SDPs Process indicator behavior varies with the development cycle
Raczynski and Curtis (2008)	Heterogeneous sources of variation Too few data points exist when constructing valid control limits Too many control charts must be maintained when disaggregating data by attributes
Sargut and Demirörs (2006)	Multiple common causes of variation in SDPs Selection of metrics for implementing SPC Trade-off between number of data points and depth of analysis
Tarhan and Demirörs (2006)	Rational sampling of process data Difficulty in choosing suitable indicators for implementing SPC in SDPs Lack of sufficient data
Weller and Card (2008)	Few data for constructing control charts Multiple common causes of variation in SDPs

or "short runs." With short runs, lot sizes are generally small and the amount of process data is limited. Consequently, an insufficient amount of data may bias process parameter estimations when applying a conventional control chart. Even when sufficient data required for estimating process parameters are available, charting at or very near the start of a short production run is desirable (Quesenberry 1991c). Several short-run control charts, such as deviation from nominal chart, the Q chart, Tukey's chart, and t-chart, are developed to overcome these issues associated with insufficient data (Alemi 2004; Bothe 1989; Castillo et al. 1996; Celano et al. 2011; Garjani et al. 2010; Quesenberry 1991c; Torng et al. 2009; Zhang et al. 2009). Most of these short-run control charts can be constructed with a limited amount of data. Among these short-run control charts, only the Q chart can monitor process performance at the start of a process with as few as three observations for real-time charting and has fixed control limits (i.e., $\pm$ 3) for plotting different performance measurements on the same chart. With the Q chart, problems in applying conventional control charts to situations lacking homogeneous data and maintaining many control charts can be resolved. Thus, the properties of the Q chart may be more appropriate and useful than conventional control charts when monitoring SDPs. While the Q chart has proven effective in short-run manufacturing, the Q chart has not been applied to the SDP. Therefore, the main objective of this study is to investigate the feasibility of using the Q chart as an alternative control chart for the SDP. To assess the performance of the Q chart in SDPs, three examples are utilized to demonstrate the effectiveness of the Q chart. Moreover, practical recommendations for utilizing Q charts are provided.

The remainder of this paper is organized as follows. Section 2 gives an overview of conventional and short-run control charts. Section 3 reviews work related to utilization of control charts by the software industry. Section 4 describes the application of Q charts for monitoring the SDP. Section 5 utilizes three examples to demonstrate the benefits of using Q charts in the SDP. Section 6 provides recommendations for utilizing Q charts. Finally, conclusions are given in Sect. 7.

2 Control charts

2.1 Conventional control charts

The control charts developed by Shewhart (1926) in the 1920 s are regarded as important tools for SPC, which uses statistical techniques to detect unusual sources of variation in a production process. Process variation can be classified as common-cause (or chance cause) variation and special-cause (or assignable cause) variation. Common-cause variation, characterized by a stable and consistent pattern of measured values over time, is the result of normal or inherent interactions among people, machines, materials, environment, and methods of a process (Florac and Carleton 1999). Special-cause variation, characterized by sudden or persistent abnormal changes to one or more process components, comprises events that are not part of the normal process. A typical control chart contains a center line (CL), which represents the average value of the process characteristic and two horizontal lines, namely the upper control limit (UCL) and lower control limit (LCL) (Fig. 1) (Montgomery 2009). If a data point falls outside the control limits, an assignable cause is assumed to exist and the process is suspected to be unstable or out of control; otherwise, the process is considered to be stable or in control. A number of Shewhart control charts exist, such as the $\overline{X} - R$, $\overline{X} - S$, and $X - MR$ charts for variable data, and the p, np, c, and u charts for attribute data. The $X - MR$ chart and u chart are the most commonly used control charts in the SDP (Baldassarre et al. 2007; Jacob and Pillai 2003; Kulpa and Johnson 2008). An $X - MR$ chart is a pair of charts; each data point in the X chart represents an individual value of a measurable characteristic and each point in the MR chart represents the moving range of two successive data points. The u chart is typically utilized to plot individual rates for nonconformities per unit.

Using Shewhart control charts involves two distinct phases, namely phases I and II. The primary task in phase I is to construct trial control limits to determine whether a process

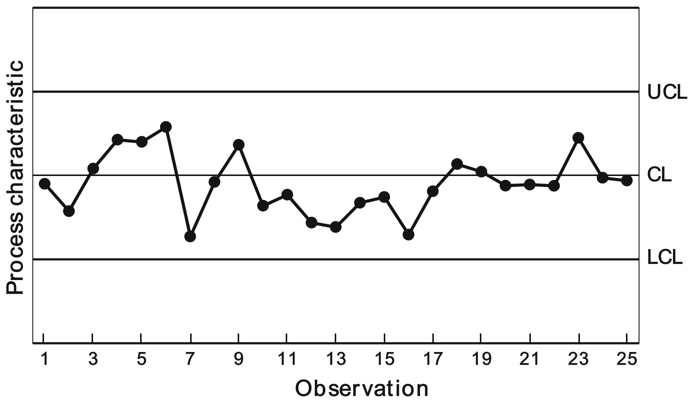

Fig. 1 A typical control chart (Montgomery 2009)

has been in control over the period in which process data were collected. Phase II is a process monitoring phase. Once control limits are determined in phase I, these control limits are applied to monitor future process performance in phase II. The Shewhart control chart requires a large number of observations to estimate the process mean and process variation to establish the control limits in phase I. Many studies have indicated that using 20–30 subgroups with 4 or 5 observations is adequate (Montgomery 2009; Quesenberry 1991c). For practical applications, Quesenberry suggested that 100 initial subgroups with at least 5 observations is reasonable for estimated control limits to perform as a chart with known parameters. (Quesenberry 1993; Tsai et al. 2004).

2.2 Short-run control charts

Mass production lot sizes are generally large and, therefore, constructing a control chart is not difficult (Castillo et al. 1996). However, a recent trend in manufacturing industries is to produce small lot sizes or use short production runs for flexible manufacturing using JIT or job-shop systems. Such manufacturing systems involving start-up processes and short runs are characterized by high varieties and low production volumes (Castillo et al. 1996; Quesenberry 1991c). Short production runs cause problems when constructing a Shewhart control chart, as the amount of data is insufficient for estimating process parameters to establish valid control limits. That is, operators typically encounter a situation in which a production run may end before they can determine whether the process was in or out of control. Therefore, some adjustments to conventional control charts are needed for SPC for short production runs.

2.2.1 DNOM charts

Cullen and Bothe (1989) introduced a control chart called the deviation from nominal method (DNOM) for short production runs. For instance, consider a particular part A, let M_i be the actual measurement for the ith sample and T_A be the nominal or target value of part A. Deviations from the nominal value for the ith sample can be expressed as

$$x_i = M_i - T_A \tag{1}$$

Thus, each x_i can be plotted in a time order on an $\overline{X} - R$ chart with its center line at zero. When using the DORM chart, process variances are assumed equal for all parts. When this assumption is violated, a standardized chart is typically employed (Montgomery 2009). Assuming σ is the process standard deviation for part A, let z_i be defined as follows:

$$z_i = \frac{M_i - T_A}{\sigma} \tag{2}$$

where z_i is the standardized value for the measurement of the ith sample. Thus, each z_i can be plotted in a time order on a standardized $\overline{X} - R$ chart with its center line at zero.

2.2.2 Q charts

With short production runs, process mean and variance cannot be known before the production run begins, and data for estimating process parameters to establish valid control limits are lacking. To deal with problems encountered while constructing a control chart for short production runs, Quesenberry developed a series of Q charts for cases in which process parameters are known or unknown (Quesenberry 1991c). Suppose a sequence of observations, $\{X_1, X_2, \cdots\}$, drawn from a process is normally distributed with mean μ and variance σ^2. Quesenberry (1991c) defined the following four Q statistics based on individual observations for cases in which μ and σ^2 are known or unknown:

Case I: Both $\mu = \mu_0$ and $\sigma = \sigma_0$ are known,

$$Q_r(X_r) = \frac{X_r - \mu_0}{\sigma_0} \quad r = 1, 2, \cdots \tag{3}$$

Case II: μ is unknown, and $\sigma = \sigma_0$ is known,

$$Q_r(X_r) = \left(\frac{r-1}{r}\right)^{\frac{1}{2}} \frac{(X_r - \overline{X}_{r-1})}{\sigma_0} \quad r = 2, 3, \cdots \tag{4}$$

Case III: $\mu = \mu_0$ is known, and σ^2 is unknown,

$$Q_r(X_r) = \Phi^{-1}\left\{G_{r-1}\left(\frac{X_r - \mu_0}{S_{0,r-1}}\right)\right\} \quad r = 2, 3, \cdots \tag{5}$$

where $S_{0,r}^2 = \frac{1}{r}\sum_{j=1}^{r}(X_j - \mu_0)^2$

Case IV: both μ and σ^2 are unknown,

$$Q_r(X_r) = \Phi^{-1}\left\{G_{r-2}\left[\left(\frac{r-1}{r}\right)^{\frac{1}{2}}\left(\frac{X_r - \overline{X}_{r-1}}{S_{r-1}}\right)\right]\right\} \quad r = 3, 4, \cdots \tag{6}$$

where $S_r^2 = \frac{1}{r-1}\sum_{j=1}^{r}(X_j - \overline{X}_r)^2$

where $\overline{X}_r$ in Eqs. (3)–(6) represents the sample mean estimated from the first r observations (i.e., $\overline{X}_r = \frac{1}{r}\sum_{j=1}^{r} X_j$); and $\Phi^{-1}(\cdot)$ and $G_v(\cdot)$ denote the inverse of the standard normal distribution function and the Student's t-distribution function with v degrees of freedom, respectively.

Unlike conventional control charts, Q charts are comprised of Q statistics, which are the transformations of original observations. The Q statistics for each of these four cases are normally distributed with mean $\mu = 0$ and variance $\sigma^2 = 1$. Thus, Q statistics can be plotted on an $X - MR$ chart with its center line at zero and control limits at $\pm$ 3. Other Q statistics are also developed for cases in which quality variables follow a binomial or Poisson distribution (Quesenberry 1991a, b, c).

2.2.3 *Other short-run control charts*

Alemi (2004) introduced an application of Tukey's control chart, a method for analyzing data based on the Boxplot developed by Tukey (1977), to calculate confidence intervals for medians to monitor individual observations. The control limits of Tukey's control chart can be established for a small number of observations by calculating the first quartile (Q_1) and third quartile (Q_3). The upper and lower control limits of Tukey's control chart are obtained, respectively, as follows:

$$\begin{aligned} UCL &= Q_3 + k \times IQR \\ LCL &= Q_1 - k \times IQR \end{aligned} \tag{7}$$

where IQR is the inter-quartile range (*i.e.*, $IQR = Q_3 - Q_1$), and k is a parameter for determining the width of control limits. The value of parameter k is usually set at 1.5. Tukey's control chart can be implemented easily by calculating quartiles from individual observations without estimating the process mean and variance. Torng et al. (2009), who evaluated the performance of Tukey's control chart in monitoring short-run processes, indicated that Tukey's control chart has good detectability in mean shift when process data follow a normal distribution. They also noted that a large k should be utilized (i.e., $k > 1.5$) for a non-normal distribution to reduce the type I error (Torng et al. 2009).

Zhang et al. (2009) developed a new chart, the t-chart, based on the Student's t statistic as an alternative to the Shewhart $\overline{X}$ chart for monitoring the process mean when process standard deviation σ s not estimated well or when process standard deviation varies. Suppose that the sequences of subgroup observations $\left\{X_{i,1}, X_{i,2}, \cdots, X_{i,n}\right\}$ from a process at time point $i = 1, 2, \cdots$ are normally distributed with a mean μ and variance σ^2, where n is subgroup size. The statistic t_i is defined as follows:

$$t_i = \frac{\overline{X}_i - \mu}{S_i/\sqrt{n}} \quad i = 1, 2, \cdots \tag{8}$$

where $\overline{X}_i = \frac{1}{n}\sum_{j=1}^{n} X_{i,j}$ and $S_i = \sqrt{\frac{1}{n-1}\sum_{j=1}^{n}(X_{i,j} - \overline{X}_i)^2}$ denote the subgroup mean and subgroup standard deviation, respectively. In Eq. (8), t_i follows a Student's t-distribution with n-1 degrees of freedom. The control limits of the t-chart can then be set as follows:

$$\begin{aligned} UCL_t &= G_{n-1}^{-1}\left(1 - \frac{\alpha}{2}\right) \\ LCL_t &= -UCL_t \end{aligned} \tag{9}$$

where $G_v^{-1}(\cdot)$ is the inverse distribution function of the Student's t-distribution with v degrees of freedom, and α is the false alarm rate and is often set at 0.0027. Thus, each statistic t_i can be plotted in a time order on a chart with its center line at zero and control limits at UCL_t and LCL_t. Notably, Eq. (9) does not require an estimation of process mean μ and standard deviation σ. That is, the control limits of the t-chart can be obtained as soon as subgroup data are collected, rather than estimating μ and σ using a sufficient amount of data. Celano et al. (2011) evaluated the suitability of using the t-chart in short-run manufacturing. They demonstrated that the t-chart can monitor a short-run process successfully.

3 Related work

3.1 Utilization of the conventional control chart in the software industry

Numerous studies have identified the benefits of using a control chart in the software industry. Florac and Carleton (1999) provided practical guidelines for SPC with the goal of improving the SDP. They provided extensive software process metrics and fully worked-out examples, demonstrating how to manage, control, predict, and improve process performance to achieve business and technical goals using control charts. Weller (2000) used an $X - MR$ chart of the code inspection rate to assess product quality during testing and used an u chart constructed with defect density data to predict post-shipping product quality for a major software release. They also indicated that applications of the SPC method provide a positive cost-benefit return.

Jacob and Pillai (2003) used an X chart for the code review process to monitor and control process-performance metrics such as preparation speed, review speed, and defect density. They summarized the benefits of applying SPC to the code review process and demonstrated that control charts can help manage, control, and improve coding and code review performance. Komuro (2006) shared experiences in applying SPC techniques to SDPs at Hitachi Software Engineering (HSE), Japan's largest software development company. Komuro also analyzed the characteristics of SDPs and their influence on SPC.

In CMMI, the control chart is used to establish process-performance baselines for a standard SDP of an organization. Moreover, the control chart is utilized to quantitatively manage the defined process of a project to achieve established quality and process-performance objectives (CMMI Product Team 2010; Sargut and Demirörs 2006).

3.2 Difficulties in applying a control chart to the software development process

Although several studies have assessed the effectiveness of using a control chart to monitor and control an SDP, some controversy surrounds the suitability of utilizing control charts directly in the software industry. In practice, software processes differ markedly from manufacturing processes (Baldassarre et al. 2007). The SDPs are more heavily dependent on human activities and require more creativity than manufacturing processes (Caivano 2005; Komuro 2006). For instance, software engineering inspections differ fundamentally from manufacturing inspections. The former is usually conducted by peers or experts, who use their knowledge and experience to identify errors in software products (e.g., documents or codes). The latter is generally examined by a machine, instruments, or operators according to standard operating procedures to identify defects. Moreover, unlike a manufacturing process, software process performance is often affected by many sources of variation such as software engineering tools, development models, coding language, skill of engineers, and the development environment (Florac and Carleton 1999). Consequently, multiple common causes are mixed when constructing a control chart for monitoring SDPs. Therefore, control limits are usually too wide to detect assignable causes in a software process (Raczynski and Curtis 2008; Sargut and Demirörs 2006; Weller and Card 2008).

Conventional control charts require data to be drawn independently from a homogeneous source of variation. However, data from a software process may easily violate this requirement due to a lack of rationally subgrouped measurements (Florac and Carleton 1999; Raczynski and Curtis 2008). To avoid this problem, software organizations often segregate or disaggregate data into groups with similar characteristics to reduce the effect of mixed multiple common causes and obtain a reasonable set of control charts, typically with narrow control limits. Tarhan and Demirörs (2006) applied a novel systematic approach to identify rational samples of a process as well as to select process metrics. However, segregating data into groups may generate another significant problem—insufficient data points to establish valid control limits at the organizational or project level (Raczynski and Curtis 2008; Tarhan and Demirörs 2006). To obtain additional data points, software organizations aggregate data to meet the requirements for constructing conventional control charts. Consequently, this may lead to wide control limits due to mixed multiple common causes. This has resulted in difficulties in applying a conventional control chart to SDPs. Sargut and Demirörs (2006) identified a trade-off between the number of data points and depth of analysis, making SPC difficult to apply to SDPs. Rather than using conventional control charts, an alternative control chart is needed to overcome this dilemma.

4 Applying Q charts to monitor the software development process

Quality engineers in the software industry usually need to produce high-quality software products within a limited period. Therefore, they must begin monitoring process performance using limited data in the early stage of the SDP. The properties of high variety and low volume (HV/LV) in short-run manufacturing are similar to the properties of SDPs. As mentioned in Sect. 2.2, short-run control charts (i.e., the Q chart, Tukey's chart, and t-chart) have early detection capability for monitoring process performance with a limited amount of data, and estimations of process parameters are not needed in phase I when constructing a control chart. However, among these short-run control charts, Tukey's chart does not perform well in detecting process shifts when a dataset is small and parameter k for control limits must be determined in advance according to data type (Torng and Lee 2008). The t-chart is often utilized to replace the $\overline{X}$ chart; however, application of a t-chart for individual measurements has not been examined in the short-run environment.

Conversely, the Q chart has been utilized for over a decade, such that several Q statistics have been proposed for cases in which process parameters (i.e., mean μ and variance σ^2) are known or unknown. Because process parameters are usually unknown before a process starts, Case IV of the Q statistic is recommended for constructing a Q chart to monitor SDPs. Notably, Eq. (6) for Case IV can be applied as few as three observations acquired. That is, the Q statistic can be plotted in a Q chart immediately after three observations are collected. This is useful to the software industry, because practitioners always desire to begin charting at or very near the start of a process to detect process shifts early. Moreover, because Q statistics follow a standard normal distribution, Q charts allow practitioners to plot different performance measurements on the same chart using the same control limits (i.e., $\pm$ 3). This reduces the number of control charts and simplifies maintenance efforts for an excessive number of control charts. Therefore, this study considers that the Q chart is more applicable than conventional control charts for monitoring the SDP. The benefits of Q charts for monitoring the SDP are summarized as follows (Quesenberry 1991c):

(1) A large dataset is not required to estimate process parameters. Conventional Shewhart charts require a sufficient number of data points (i.e., generally 20–30 subgroups with 4 or 5 observations) to estimate process parameters to establish valid control limits. However, this amount of process data is often unavailable in software development environments. Unlike conventional control charts, the upper and lower control limits of Q charts are set at $\pm$ 3, and Q statistics can be acquired and plotted on the Q chart in a time order without prior knowledge of process parameters.
(2) Real-time charting. Statistic Q_r in Eq. (6) is defined for $r \geq 3$, that is, Q charts can be constructed with only three observations. In other words, Q charts have an early detection capability to detect a process mean shift while using only the first few observations.

(3) Plotting different performance measurements on the same chart. All Q statistics proposed by Quesenberry (1991c) follow the standard normal distribution, meaning that the Q chart is a standardized chart with a center line at zero and control limits at $\pm$ 3. Therefore, process performance with different characteristics can be plotted on the same chart and practitioners can monitor different performance measurements simultaneously.

5 Examples of applying the Q chart to the SDP

To demonstrate the effectiveness of applying the Q chart to the SDP, three examples are presented, and Q chart results are compared with those of conventional control charts.

5.1 Example 1

A code review process is a formal and efficient way to examine a program in detail to identify errors (Fagan 1999). A software program can be checked and corrected through a review process before release. Several indicators, such as preparation rate, inspection rate, and total faults observed per thousand lines of code (KLOC), are commonly used to monitor the performance of the code review process (Barnard and Price 1994). In this example, control charts are utilized to monitor inspection rate. Inspection rate is defined as the number of lines of code (LOC) examined per hour during a review meeting. The individual moving range ($X - MR$) control chart is frequently utilized to monitor the review process (Florac and Carleton 1999; Weller 2000). Inspection rates are collected and plotted in a time order on an $X - MR$ chart to determine whether all data points are within control limits. When an inspection rate of a review meeting exceeds the upper control limit on the X chart (i.e., inspection rate is faster than expected), the ability to identify errors decreases and a re-inspection activity is needed. Conversely, when an inspection rate is lower than the lower control limit (i.e., inspection rate is slower than expected), too many errors exist in the code or a reviewer did not prepare well before a meeting. Some factors, such as reviewer knowledge, programming languages, and code complexities, impact inspection effectiveness (Barnard and Price 1994). Consequently, mean inspection rates differ among these factors. Therefore, to illustrate the code review process, a set of simulated inspection rates with three code complexity levels is used to demonstrate the difficulty in identifying process instabilities using the conventional control chart and the benefits of utilizing a Q chart.

Table 2 shows the simulated records for code inspection rate (LOC/h) and code complexity levels (i.e., high, medium, and low) from each code review meeting. The code inspection rates are simulated from normal distributions with various parameters (Table 3) for each complexity level. Notably, the inspection rate of 225 LOC/h for review sequence No. 24 is markedly larger (i.e., five standard deviation from the mean of 150) than others at the high complexity level. This data point is utilized to test the ability of control charts to detect an abnormal signal. Figure 2a shows an $X - MR$ chart constructed for 28 inspection rates in each review meeting. No out-of-control points exist in both the X chart and moving range chart. Thus, the process is stable or under control. Additionally, three X charts (Figs. 2b–d) are constructed for each of the three complexity levels, and analytical results demonstrate that the no instability exists at each complexity level in the review process. For comparison with X charts, three Q charts are constructed (Figs. 3b–d). The Q statistics in Q charts are obtained using Eq. (6), as both process mean and variance are unknown. The analytical results (Fig. 3b) reveal that an out-of-control signal exists in review sequence No. 9 for high complexity level (i.e., sequence No. 24 for all levels). Thus, the process is likely unstable.

In this example, an extremely high inspection rate is used to test the ability of the $X - MR$ chart and Q chart to detect an abnormal signal. When observations at all complexity levels are plotted in a time order, control limits are too wide (range, 63–351 LOC/h) to identify instabilities in process performance, such as the inspection rate of 225 LOC/h for the high complexity level (Fig. 2a). Thus, mixing data from different attributes on the same chart is inappropriate for monitoring process performance. The general solution is to construct a control chart for each complexity level to separate variation sources. The analytical results (Fig. 2b) indicate that the X chart cannot detect the extremely high value of observation (i.e., inspection rate of 225 LOC/h for the high complexity level). The reason is that when one constructs the $X - MR$ chart with a small number of observations,

Table 2 Code inspection rate (LOC/h) and code complexity level from each code review meeting

Review sequence	Inspection rate	Code complexity level
1	165	High
2	135	High
3	293	Low
4	167	High
5	240	Medium
6	223	Low
7	256	Low
8	277	Low
9	233	Medium
10	142	High
11	218	Low
12	282	Low
13	242	Low
14	209	Medium
15	200	Medium
16	244	Low
17	156	High
18	210	Medium
19	146	High
20	134	High
21	192	Medium
22	195	Low
23	150	High
24	225	High
25	280	Low
26	220	Medium
27	226	Medium
28	145	High

Table 3 Simulated parameters for inspection rate for each complexity level

	High level (excluding No. 24*)	Medium level	Low level
Observations	9	8	10
Mean (LOC/h)	150	200	250
Standard deviation (LOC/h)	15	20	25

* Inspection rate of 225 LOC/h for No. 24 is designed as five standard deviation from the mean of 150

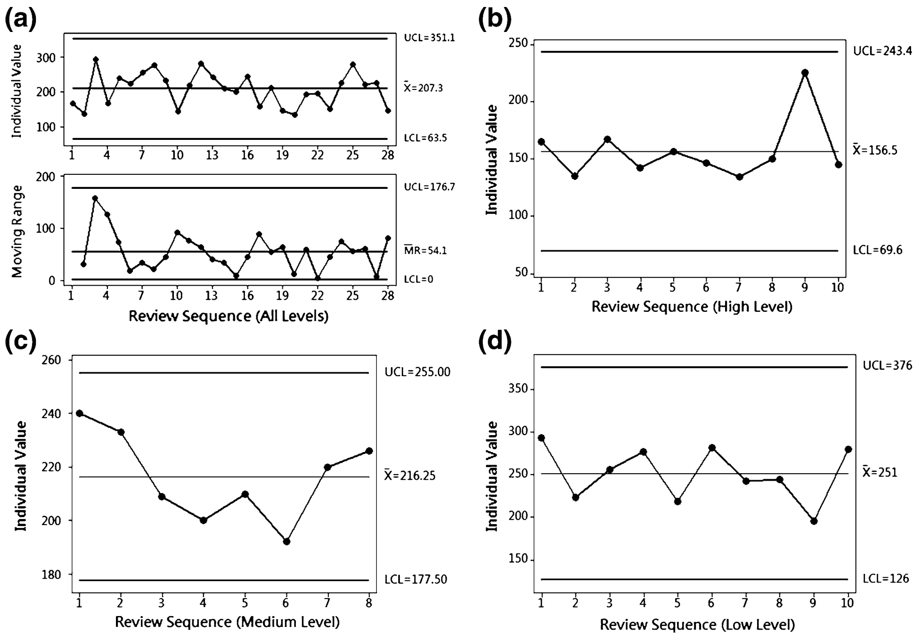

Fig. 2 Conventional control charts for inspection rate during a review process: **a** the $X - MR$ chart for all observations; **b** X chart for the high complexity level; **c** X chart for the medium complexity level; and **d** X chart for the low complexity level. Notably, no signs of instability exist in the moving range charts. To simplify this figure, moving range charts are not shown in **b**, **c**, and **d**

the extreme value(s) will inflate the average of moving range and bias estimations of process parameters. Consequently, practitioners obtain a wide set of control limits. Therefore, if control limits (Fig. 2b) are utilized in phase II for monitoring future data points, the X chart would typically fail to detect abnormal signals in the review process at the high complexity level.

On the other hand, three Q charts are constructed for the three complexity levels. Compared with X charts (Figs. 2b–d) and Q charts (Fig. 3b–d), Q charts have patterns resembling those of X charts; however, the Q charts for the high complexity level (Fig. 3b) detects an unusual point in review sequence No. 9. The comparison results indicate that the Q chart detects abnormal signals better than the X chart when the sample size is small. Moreover, inspection rates for each complexity level can be plotted on the same chart (Fig. 3a). Therefore, the Q chart can reduce the number of control charts and simplify

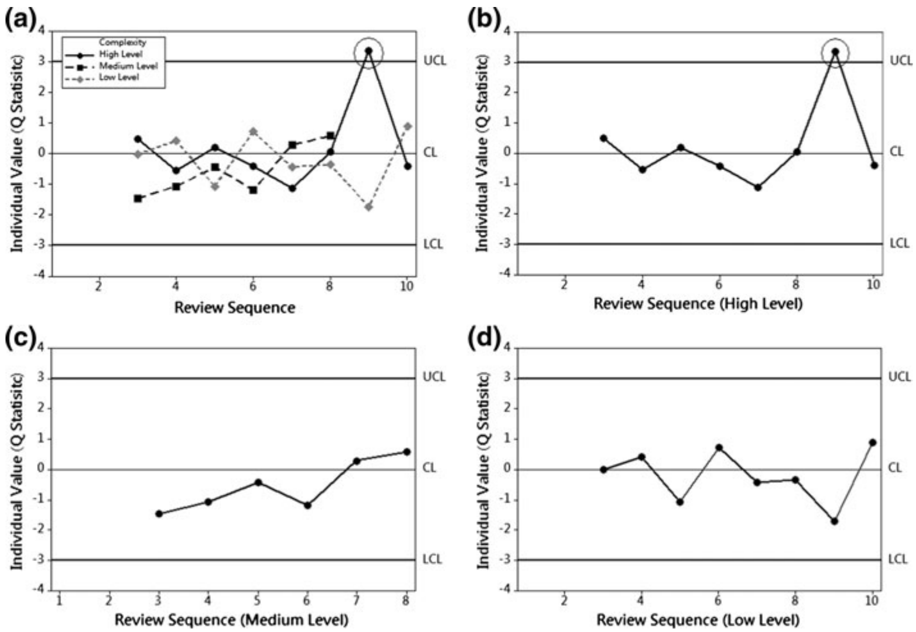

Fig. 3 The Q charts for inspection rate during a review process: **a** the Q chart for each complexity level; **b** Q chart for the high complexity level; **c** Q chart for the medium complexity level; and **d** Q chart for the low complexity level

maintenance efforts for an excessive number of control charts. Furthermore, the Q chart can be constructed as few as three observations collected in a time order. Figure 4 shows the process of using the Q chart in a review sequence at the high complexity level. That is, when each review meeting finishes and inspection data from the meeting are collected, the corresponding Q statistics can be plotted on the Q chart immediately. Once an out-of-control signal exists on the Q chart, investigative and corrective actions can be undertaken immediately to return the process to normal as early as possible.

5.2 Example 2

The second example illustrates the implementation of Q charts for decomposing process-performance data. In this example, the dataset of component defects used by Florac and Carleton (1999) on pp. 150–152 is considered. The dataset consists of the number of defects identified during the inspection process for each of the 21 components, and the defect data are classified based on eight defect types. Therefore, an $X - MR$ chart can be constructed by plotting the total number of defects aggregated from each defect type for each component to determine whether the process is in or out of control. The analytical results of the $X - MR$ chart obtained by Florac and Carleton (1999) on p. 151 indicate that all data points plotted on the $X - MR$ chart lie within control limits and reveal no assignable cause. However, causes of variation of each defect in each defect type are heterogeneous in most situations. Thus, when plotting aggregated data (i.e., total defects in this example) on a control chart, control limits lack the ability to detect unusual signals in the inspection process. Therefore, Florac and Carleton (1999) suggested that individual X charts be constructed for each of the eight defect types. Consequently, these X charts

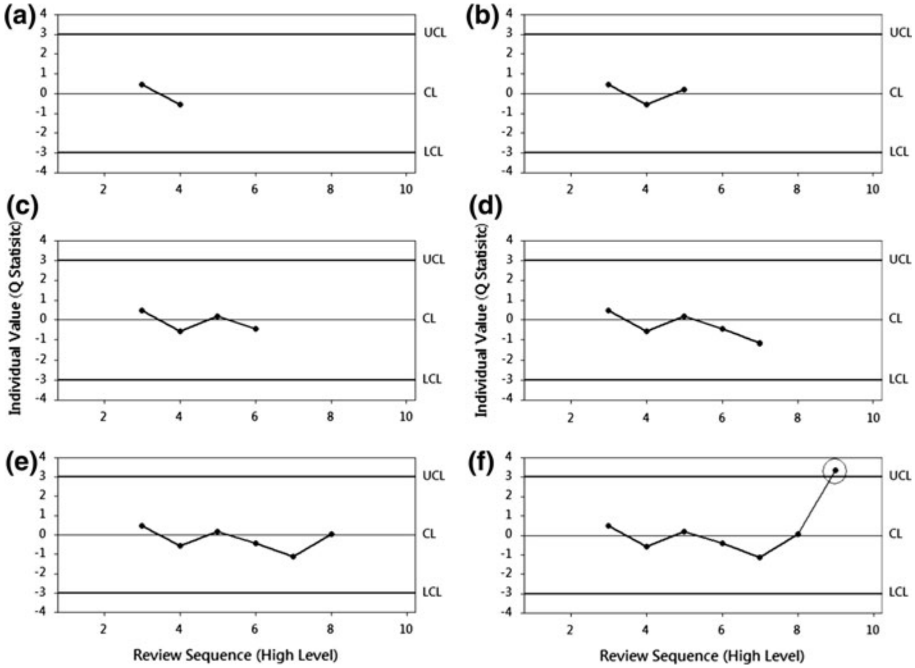

Fig. 4 The process of using the Q chart in a review sequence at the high complexity level

obtained by Florac and Carleton (1999) on p. 152 differ from the $X - MR$ chart for total number of defects: several points in the X charts exceed the upper limit, indicating that the process is unstable.

To compare X chart results, Q charts are constructed for each of the eight defect types (Fig. 5). The Q statistics in the Q charts are obtained using Eq. (6), as both process mean and variance are unknown. The Q chart results indicate that several data points lie outside control limits and have patterns resembling those in the X charts. The analytical results of Q charts also reveal that the process is out of control or unstable. Additionally, unlike the X chart—for which one must collect a sufficient number of observations to establish control limits—the Q chart can be constructed with as few as three observations, that is, to monitor process performance, practitioners can utilize the Q chart to begin identifying an unusual performance level at an early stage in the SDP. For instance, by constructing the Q chart from the start of a process (Fig. 5a), the out-of-control data point can be identified before component No. 9 is inspected. Therefore, corrective and preventive actions for the assignable cause can be initiated immediately.

5.3 Example 3

Schedule and cost control are essential when evaluating the progress and performance of an SDP. Earned value management (EVM), which has been adopted by many software organizations, is a well-known system introduced by the U.S. Department of Defense for project monitoring. Earned value management uses two indicators, namely the Schedule Performance Index (SPI) and Cost Performance Index (CPI), which are directly related to project execution efficiency (Lipke and Vaughn 2000). The SPI represents the rate of

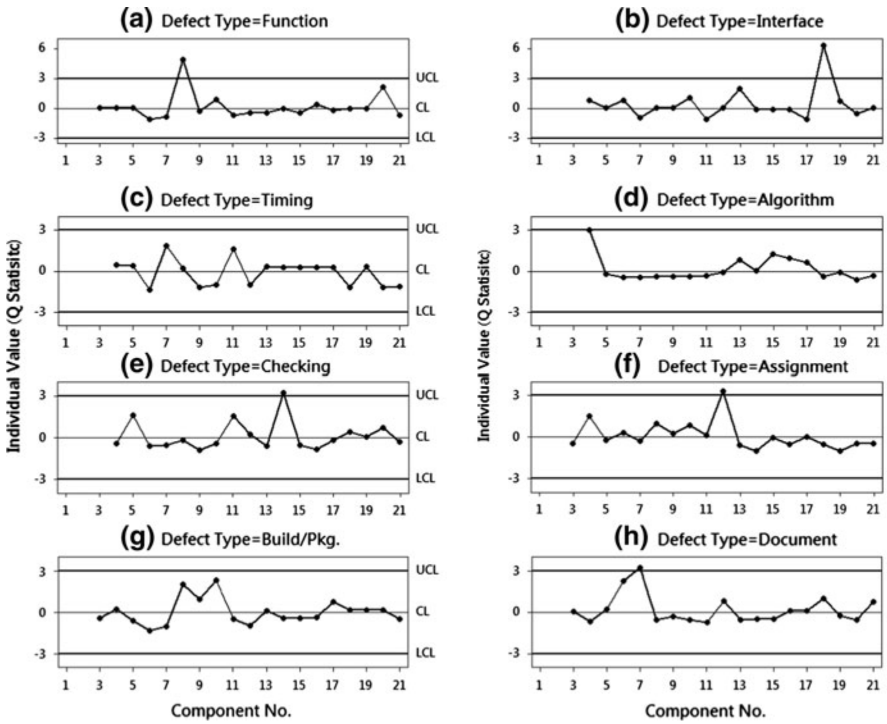

Fig. 5 The Q charts for each defect type (note: as the first two observations have the same value, the Q statistic for the third observation will be invalid. Therefore, Q charts in **b**, **c**, **d**, and **e** initiate from the fourth data point.)

achieving earned value with respect to the schedule baseline, and the CPI represents the rate of achieving earned value with respect to actual cost. If an SPI value is greater/less than 1.0, the project is ahead/behind schedule. Likewise, a CPI value greater/less than 1.0 indicates that the project is under/over budget. In practice, the SPI and CPI are collected once every 15 days or once a month during the development life cycle to assess project performance. A well-performing project should keep its SPI and CPI values as close to 1.0 as possible. However, variances between earned value and planned value (or actual value) are very common when executing a project in an SDP. Thus, defining the amount of deviation from 1.0 that is considered abnormal performance is difficult. Fortunately, statistical theory provides methods based on random variation in process performance to calculate the upper and lower limits for detecting abnormal performance. Consequently, several studies have applied conventional control charts (e.g., the $X - MR$ chart) to the SPI and CPI to monitor schedule and cost performance of software projects (Leu and Lin 2008; Lipke 2002; Lipke and Vaughn 2000; Wang et al. 2006).

When control charts are used to monitor the SPI and CPI, practitioners may encounter a general problem in that the number of data points of the SPI and CPI for constructing a control chart is insufficient in a short- or medium-term project. In this example, instead of utilizing conventional control charts, the Q chart is applied to a real case to monitor the SPI and CPI simultaneously. The case is a software development project with a maximum duration of 12 months. This case is provided by Information and Communications

Research Laboratories (ICL) of the Industrial Technology Research Institute (ITRI), Taiwan, and is a nonprofit software organization that achieved CMMI Level-4 in 2010. According to the ICL's policy of process-performance measures, the SPI and CPI (Table 4) are collected once every 15 days during a project. However, the number of SPI and CPI data points is too few for constructing an $X - MR$ chart within the first few months of this project. Consequently, the calculated control limits with small amounts of data are unreliable to detect abnormal performance in the early stages of this project. Therefore, the Q chart is utilized in this case to monitor the SPI and CPI. Figure 6 plots the SPI and CPI values on a Q chart every 15 days. Unlike the X chart, the SPI and CPI values can be plotted on the same Q chart, which helps project managers evaluate schedule and cost performance simultaneously. Additionally, project performance can be monitored using the Q chart in the early stages of this project from February 15. The analytical result of the Q chart indicates that cost performance is in control, but an unusual schedule performance is detected on June 15. Thus, the Causal Analysis and Resolution (CAR) process must be initiated immediately by project stakeholders to identify causes and then take actions to prevent future schedule delays.

6 Recommendations for utilizing Q charts

Section 5 discussed the benefits of using Q charts, that is, Q charts can be utilized to monitor effectively a SDP with a limited number of observations in the early stages of a project. However, like other conventional control charts (e.g., the $\overline{X} - R$ or $X - MR$ charts), some assumptions are needed for process data. When utilizing the Q chart and Eqs. (3)–(6) monitor SDPs, process observations or indicators are assumed to be independently normally distributed (Quesenberry 1991c). The limitation of this study is its inability to determine whether all metrics or indicators in SDPs satisfy Q chart assumptions. If these assumptions are severely violated when utilizing Q charts to monitor SDPs, the detection capability of the Q chart is reduced. In this situation, the data transformation method (e.g., logarithmic or square root transformations) should be applied to transform the original indicator into a new variable that is approximately normally distributed. Moreover, while

Table 4 The dataset of SPI and CPI provided by the Information and Communications Research Laboratories (ICL)

Date	SPI	CPI
Jan. 15	1.37	1.05
Jan. 31	1.46	1.02
Feb. 15	1.20	1.00
Feb. 28	1.09	0.99
Mar. 15	1.20	0.86
Mar. 31	1.20	0.93
Apr. 15	1.20	0.97
Apr. 30	1.20	0.90
May 15	1.20	1.05
May 31	1.28	1.03
June 15	0.77	1.00
June 30	0.65	0.96

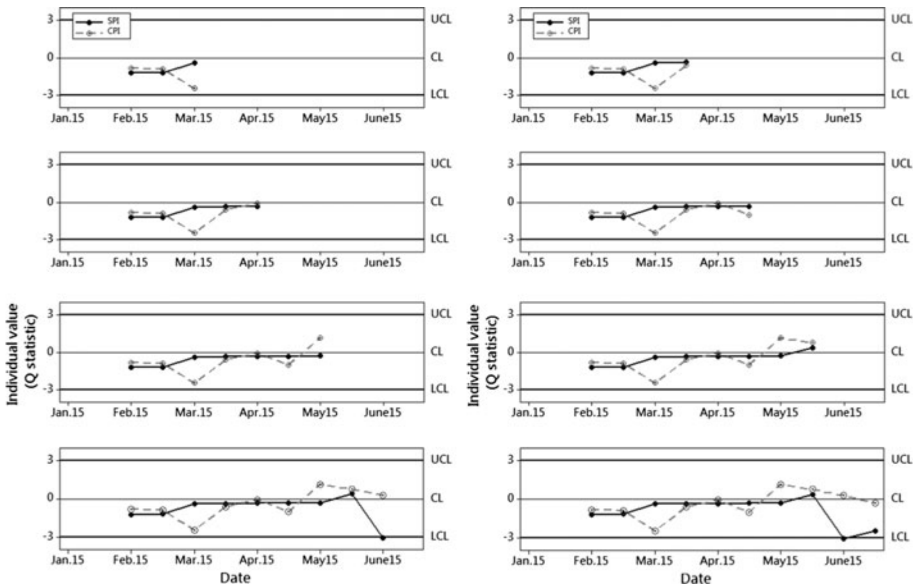

Fig. 6 The Q charts for the SPI and CPI

the Q chart has an early detection advantage over conventional control charts, some recommendations for utilizing Q charts in practice are summarized as follows.

(1) Compared with conventional control charts, a sufficiently large number of observations in phase I for estimating process parameters are not needed when constructing Q charts and control limits of Q charts do not need to be recalculated. While the control limits of the Q chart do not need to be updated or revised, we suggest that a new Q chart be constructed when processes change. Once processes have changed, the degree of process variation may be altered accordingly. Consequently, data collected previously are no longer applicable for an ongoing Q chart. For instance, if a process has been improved due to the effect of corrective actions, practitioners should start a new Q chart with new observations collected from the improved process to reflect the current state of the process.

(2) When an out-of-control point exists in the Q chart, the process is likely unstable and efforts should be made to determine the cause of this unusual data point. Additionally, when utilizing Q charts, we do not recommend that one postpones investigative actions until a second signal is received, because the out-of-control value will affect calculations of subsequent Q statistics. When the abnormal observation is included in the calculations, the updated S_{r-1} in Eq. (6) will be inflated. Consequently, the calculated subsequent Q statistics is pulled toward zero and the abnormal signal is unable to be detected (Zantek and Nestler 2009). Therefore, once assignable causes have been identified for the out-of-control point, this point should be removed from the data sequence and not used in subsequent computations. For instance, if the out-of-control signal at the seventh data point (Fig. 5h) has assignable causes after investigation, this point should be removed from the data sequence before plotting the eighth point. The subsequent Q statistics can then be calculated and plotted on the chart to reflect the most recent process state. Figure 7 shows the

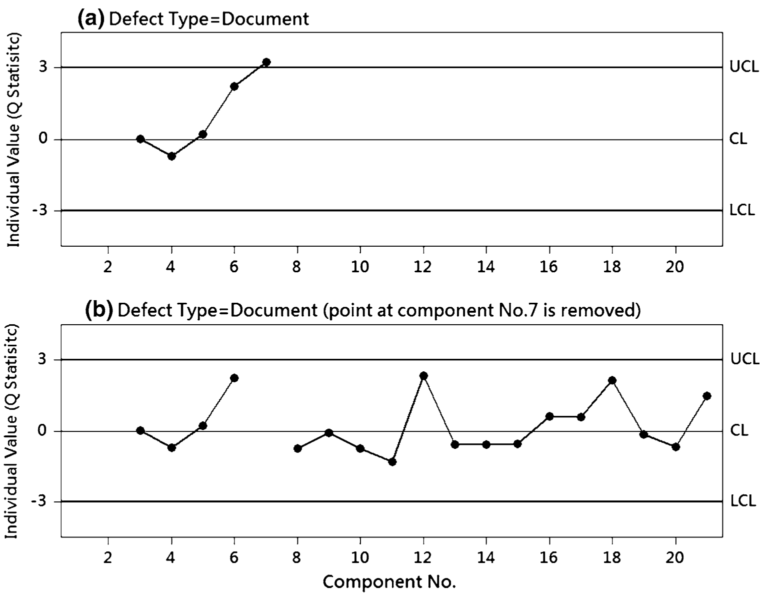

Fig. 7 The *Q* charts for documentation defect type in example 2: **a** occurrence of assignable cause; and **b** removal of the point at component No. 7

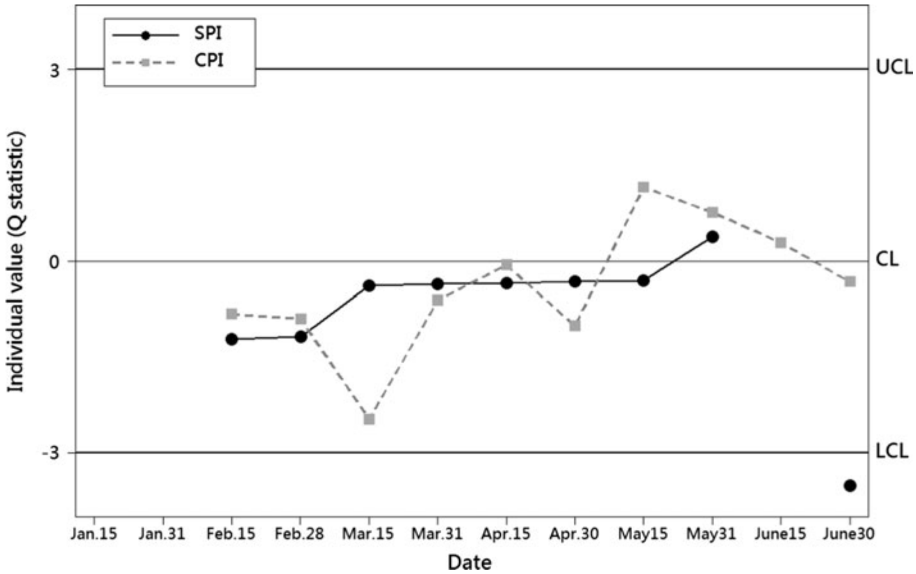

Fig. 8 The revised *Q* chart of the SPI and CPI for June 30 after removing the abnormal point on June 15

outcome of removing an out-of-control signal from the *Q* chart for documentation defect type in example 2. Likewise, in example 3, while an abnormal performance of SPI has been found on June 15, the point should be removed before plotting subsequent points. The *Q* chart of the SPI and CPI for June 30 is revised in Fig. 8.

(3) Since all observations are transformed into Q statistics, data points on Q charts no longer have the original measurement scale. Therefore, when interpreting out-of-control signals on Q charts, we suggest that one refers to the original observation value for additional information. For instance, the out-of-limit value of 4.9 at the eighth data point (Fig. 5a) provides less information than the corresponding measurement of 20 defects when investigating the root cause.

7 Conclusions

The Shewhart control chart is very useful for monitoring process variations over time and can help practitioners manage process performance quantitatively. However, the conventional control chart requires a large number of observations to estimate process parameters accurately and establish valid control limits; otherwise, type I or type II errors may occur in statistical reasoning when utilizing a control chart. However, a large amount of data is unavailable in SDPs. Therefore, in this study, the Q chart, which is adopted for short-run manufacturing, is used as an alternative control chart to overcome such an issue for the SDP. Additionally, the early detection capability, real-time process monitoring, and fixed control limits are advantages of applying the Q chart to monitor SDPs. Software practitioners can begin monitoring performance with a limited amount of data early in a process to produce high-quality software products within a short period. Thus, we conclude that the Q chart is more applicable than the conventional control chart for monitoring the SDP.

The Q chart for a normal process with unknown parameters for monitoring the process mean is considered in this study. The corresponding Q charts for data characterized by a binomial or Poisson distribution can also be applied in a similar manner. Thus, we suggest that further research focus on the application of Q charts for attribute data, such as nonconformities, for various metrics collected in a software environment.

Acknowledgments The authors would like to thank the National Science Council of the Republic of China, Taiwan, for financially supporting this research under Contract No. NSC 97-2221-E-009-111-MY3. The authors also would like to thank the Information and Communications Research Laboratories (ICL) of the Industrial Technology Research Institute (ITRI), ROC, Taiwan, for providing actual project data to enrich the application of this study.

References

Alemi, F. (2004). Tukey's control chart. *Quality Management in Healthcare, 13*(4), 216.

Baldassarre, T., Boffoli, N., Caivano, D., & Visaggio, G. (2004). Managing software process improvement (SPI) through statistical process control (SPC). In *Proceedings of the 5th International Conference on Product Focused Software Process Improvement*. Kansai Science City, Japan.

Baldassarre, T., Caivano, D., Kitchenham, B., & Visaggio, G. (2007). Systematic review of statistical process control: An experience report. In *Proceedings of the 11th International Conference on Evaluation and Assessment in Software Engineering (EASE)*. Keele, UK.

Barnard, J., & Price, A. (1994). Managing code inspection information. *IEEE Software, 11*(2), 59–69.

Bothe, D. R. (1989). A powerful new control chart for job shops. In *Proceedings of the 43rd Annual Quality Congress Transaction*. Toronto, Ontario, Canada.

Caivano, D. (2005). Continuous software process improvement through statistical process control. In *Proceedings of the Ninth European Conference on Software Maintenance and Reengineering (CSMR'05)*. Manchester, UK.

Castillo, E. D., & Montgomery, D. C. (1994). Short run statistical process control: Q chart enhancements and alternative methods. *Quality and Reliability Engineering International, 10*(2), 87–97.

Castillo, E. D., Grayson, J. M., Montgomery, D. C., & Runger, G. C. (1996). A review of statistical process control techniques for short run manufacturing systems. *Communications in Statistics—Theory and Methods, 25*(11), 2723–2737.

Celano, G., Castagliola, P., Trovato, E., & Fichera, S. (2011). Shewhart and EWMA t control charts for short production runs. *Quality and Reliability Engineering International, 27*(3), 313–326.

Chang, C.-P., & Chu, C.-P. (2008). Improvement of causal analysis using multivariate statistical process control. *Software Quality Journal, 16*(3), 377.

Cullen, C. C., & Bothe, D. R. (1989). *SPC for short production runs*. Dayton, OH: In Proceedings of the National Aerospace and Electronic Conference.

Diane, M., & Stephen, H. K. (2007). Practical statistical process control for software metrics. *Software Quality Professional, 9*(4), 15.

Fagan, M. E. (1999). Design and code inspections to reduce errors in program development. *IBM Systems Journal, 38*(2.3), 258-287.

Florac, W. A., & Carleton, A. D. (1999). *Measuring the software process: statistical process control for software process improvement*: Addison-Wesley.

Garjani, M., Noorossana, R., & Saghaei, A. (2010). A neural network-based control scheme for monitoring start-up processes and short runs. *International Journal of Advanced Manufacturing Technology, 51*(9–12), 1023–1032.

Jacob, A. L., & Pillai, S. K. (2003). Statistical process control to improve coding and code review. *IEEE Software, 20*(3), 50.

Jalote, P., & Saxena, A. (2002). Optimum control limits for employing statistical process control in software process. *IEEE Transactions on Software Engineering, 28*(12), 1126–1134.

Komuro, M. (2006). Experiences of applying SPC techniques to software development processes. In *Proceedings of the 28th international conference on Software engineering*. Shanghai, China.

Kulpa, M. K., & Johnson, K. A. (2008). *Interpreting the CMMI: A process improvement approach*. Boca Raton: CRC Press.

Leu, S.-S., & Lin, Y.-C. (2008). Project performance evaluation based on statistical process control techniques. *Journal of Construction Engineering & Management, 134*(10), 813–819.

Lipke, W. (2002). Statistical process control of project performance. *CrossTalk, The Journal of Defense Software Engineering, 15*(3), 15–18.

Lipke, W., & Vaughn, J. (2000). Statistical process control meets earned value. *CrossTalk, The Journal of Defense Software Engineering, 13*(6), 16–20.

Manlove, D., & Kan, S. H. (2007). Practical statistical process control for software metrics. *Software Quality Professional, 9*(4), 15.

Montgomery, D. C. (2009). *Introduction to statistical quality control*. Hoboken, N.J.: John Wiley.

Quesenberry, C. P. (1991a). SPC Q charts for a binomial parameter p: Short or long runs. *Journal of Quality Technology, 23*(3), 239–246.

Quesenberry, C. P. (1991b). SPC Q charts for a Poisson parameter lambda: Short or long runs. *Journal of Quality Technology, 23*(4), 296–303.

Quesenberry, C. P. (1991c). SPC Q charts for start-up processes and short or long runs. *Journal of Quality Technology, 23*(3), 213–224.

Quesenberry, C. P. (1993). The effect of sample size on estimated limits for Xbar and X control charts. *Journal of Quality Technology, 25*(4), 237–247.

Raczynski, B., & Curtis, B. (2008). Software data violate SPC's underlying assumptions. *IEEE Software, 25*(3), 48–50.

Radice, R. (1998). Statistical process control for software projects. In *Proceedings of the 10th Software Engineering Process Group Conference.* Chicago, Illinois, USA.

Sargut, K. U., & Demirörs, O. (2006). Utilization of statistical process control (SPC) in emergent software organizations: Pitfalls and suggestions. *Software Quality Journal, 14*(2), 135.

Shewhart, W. A. (1926). Quality control charts. *Bell System Technical Journal, 5*, 593–603.

Tarhan, A., & Demirörs, O. (2006). Investigating suitability of software process and metrics for statistical process control. In I. Richardson, P. Runeson, & R. Messnarz (Eds.), *Software Process Improvement* (Vol. 4257, pp. 88-99, Lecture Notes in Computer Science): Springer Berlin/Heidelberg.

CMMI Product Team (2010). *CMMI® for Development, Version 1.3*. Software Engineering Institute, Carnegie Mellon University.

Torng, C. C., & Lee, P. H. (2008). ARL performance of the Tukey's control chart. *Communications in Statistics-Simulation and Computation, 37*(9), 1904–1913.

Torng, C. C., Liao, H. N., Lee, P. H., & Wu, J. C. (2009). *Performance evaluation of a Tukey's control chart in monitoring gamma distribution and short run processes*. Hong Kong: In Proceedings of the International MultiConference of Engineers and Computer Scientists.

Tsai, T. R., Wu, S. J., & Lin, H. C. (2004). An alternative control chart approach based on small number of subgroups. *International Journal of Information and Management Sciences, 15*(4), 61–73.

Tukey, J. W. (1977). *Exploratory data analysis*. Reading, Mass: Addison-Wesley Pub. Co.

Wang, Q., Jiang, N., Gou, L., Che, M. R., & Zhang, R. H. (2006). Practical experiences of cost/schedule measure through earned value management and statistical process control. In *Software Process Change* (Vol. 3966, pp. 348–354, Lecture Notes in Computer Science).

Weller, E. (2000). Practical applications of statistical process control. *IEEE Software, 17*(3), 48–55.

Weller, E., & Card, D. (2008). Applying SPC to software development where and why. *IEEE Software, 25*(3), 48–48, 50.

Zantek, P. F., & Nestler, S. T. (2009). Performance and properties of *Q*-statistic monitoring schemes. *Naval Research Logistics, 56*(3), 279–292.

Zhang, L., Chen, G., & Castagliola, P. (2009). On t and EWMA t charts for monitoring changes in the process mean. *Quality and Reliability Engineering International, 25*(8), 933–945.

Author Biographies

Chih-Wei Chang is currently a Ph.D. student in the Department of Industrial Engineering and Management, National Chiao Tung University, Hsinchu, Taiwan. He received his B.S. in 1999 and M.S. in 2001 from the same department. He worked as an assistant researcher of software quality assurance at National Space Organization from 2001–2006 and worked as a quality engineer at Silicon Touch Technology Inc. from 2009–2011. His current research activities include software quality assurance, statistical process control, and data mining in industrial applications.

Lee-Ing Tong received the B.S. from National Cheng Kung University, Taiwan, and M.S. and Ph.D. from University of Kentucky, USA, all in Statistics. She is Professor in the Department of Industrial Engineering and Management, National Chiao Tung University, Hsinchu, Taiwan. She was an associate professor in the Department of Mathematics at University of Massachusetts-Dartmouth, USA. Her current research activities include quality engineering and management, and statistics in industrial applications.