

## Statistical Behaviour of Software Process and its Impact on Process Performance

B R Sharma

Department of Computer Science  
HMR Institute of Technology and Management  
Indraprastha University New Delhi, Hamidpur  
New Delhi Zip 110 038 INDIA

**Abstract:** Tools, methods, type of software, and the environment of development are the factors that can affect the performance of Software processes. The outcome of a software process can be controlled by application of SPC techniques. SPC is important in software processes because it can guide on how to stabilize and improve processes. SPC can also be used to demonstrate the improvement effects. Software development cannot be classified as pure SPC controlled process. Multiple issues need to be addressed by software development processes teams including Team Management, working with limited data, aligning business goals with development goals of project is very important for success of development project.

**Keywords** Process: Performance, SPC, Software Development Models

### I. Introduction

The optimal outcome of a process is normally ensured by strict monitoring and control based on techniques of Statistical Process Control (SPC). These can be applied to any process, like a tool, during its development/manufacturing stage, to measure the outcome of a product design specifications. SPC makes use of control charts, and monitors continuous improvement and on setting specialized experimental methods (DOE), in case required. Modern SPC systems have real time access to data and ability to respond to incoming data online.

The SPC is applied mainly in a process to following activities:

1. Understand a process and its specification limits.
2. To make a process stable, Eliminate assignable (special) sources of variation.
3. Use Control charts to monitor the ongoing processes, to detect any significant changes of mean or variation.

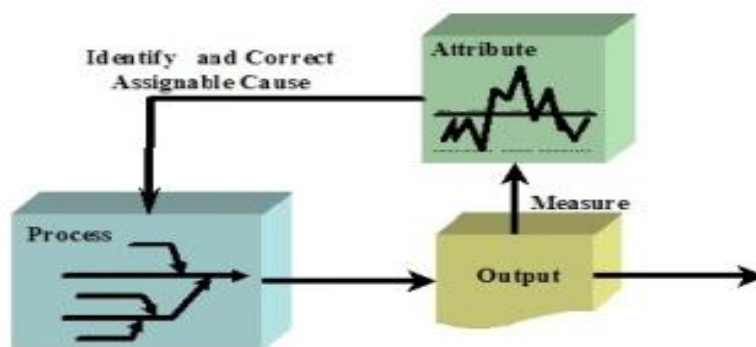


**Fig.1- Attributes of a Process monitored continuously**

SPC optimizes the information needed for a process measurement, which helps making management decisions. Statistical technique enables to work out the business baselines, support process improvements, and provide results of processes. SPC also helps in real time analysis to establish controllable process baselines; dynamic improvement of process capabilities; and focus on areas of improvement. SPC enables this decision making based on data analysis and not on opinion.

### II. Software Development Processes – Main Characteristics

A software development process in general has the following characteristics:

- (1) Software development processes are based more on thinking and are creative and unlike manufacturing. Results of software processes measurement can easily vary, and that makes a process difficult to control and minimize.

- (2) Multiple common causes: Tools, methods, type of software, and the environment of development are the factors that can affect the performance of these processes. Thus, the data obtained will be a mixed set of common cause results.
- (3) It is difficult to get a large set of homogeneous data, as can be gathered from manufacturing processes where daily production levels are high. This is due to each process having its own set of unique activities that need to be executed.
- (4) Software development follows “phase” and “cycle” concept which is quite different from continuous operations.
- (5) Process Capability is not clearly measurable as in Software Development because process limits and level of end quality not directly related.

SPC is important in software processes because it can guide on how to stabilize and improve processes, despite above stated constraints. SPC can also be used to demonstrate the improvement effects. These difficulties can be overcome by putting more emphasis on processes rather than products. Thus using product data and process data can increase the available data significantly. As the definition of software process involves people, tools, and work environment, SPC can bring stability or normalization to human activities. Multiple Common causes can be identified and separated using Statistical Process Control techniques.

### III. SPC and CMM Models

SPC can be applied to any process development requiring statistical control including software development processes. A process has multiple outputs as identifiable causes, as shown in Fig.1. In the CMMI model, the level 4 and 5 practices of the model use the SPC for process control. Whereas, SPC uses tools to monitor the performance of the production process to detect any significant variations to enable checking of production of a sub-standard article. The process output variation is categorized as follows:

- 1) "Common Causes" - is referred to as non-assignable or common, normal sources of variation. Such variations in any process are normally there and to be anticipated. This type of causes produces a stable and repeatable distribution over time.
- 2) "Special Causes" - is referred to as assignable sources of variation. These refer to any factor causing variation that affects only some of the process output. They are often intermittent and unpredictable.

These outputs, in turn, have measurable attributes. A process is under SPC if the observed process variation of process attributes is in the range of variation that of natural causes. When that variability of the process exceeds the range to be expected, the practitioner of SPC then identifies and corrects assignable causes.

If the dominant sources of variations are known, then appropriate change can be planned on these resources. It may be possible to remove these known sources and once removed, the process is said to be "stable". A stable process is one that has its variation within a known set of limits, until another assignable source of variation occurs.

SPC involves implementation of following key steps:

- Identification of defined processes
- Identification of measurable attributes of these process
- Understand natural variation of these attributes and characterize these.
- Process variation tracking

Maturity Level 4 of CMMI i.e. “Quantitatively Managed Process” involves Quantitative management is applied on the set of processes that produces a product. The significant contributors of these sub process, those impact overall process performance, are managed statistically. The results of process performance are analyzed statistically for the selected sub-processes. Thus, real time control charts like p-chart and c-chart can be used in Design phase of Software Development. Process uses Polynomial charts used in Coding phase, z-chart in testing phase and Early Customer Quality Index(ECQI) is used in beta phase. Any special causes that impact process variation are identified and appropriate correction is applied to the source of a special cause, where possible, and is corrected to prevent its recurrence.

Maturity Level 5 of CMMI that is, “Optimizing Processes”, the organization attempts improvement in its processes on continuous basis based on a quantitative understanding of the common causes of inherent variations in processes. These common causes of process variation are optimized by change in design of process. This change can be shift in the mean of the process. This can also be achieved as and when the process is re-stabilized by tuning variation. This leads to better process performance and also achieves desired process improvement objectives.”

Quantitative Project Management (QPM) process in the CMMI, (Goal #2) supports Statistical Managed Sub-process Performance with an objective to Understand process Variation.

CMMI uses statistical techniques to enable processes analysis, identify special causes of any variation in processes. This helps ensure process performance remains within well defined limits.” In addition, CMMI

Generic Practice #4.2 states "Stabilize Sub-process Performance" and explains "A stable sub-process does not exhibits significant indication in special causes of process variation. Stable sub-processes are predictable to the extent, the limits of which are established by the natural bounds of the sub process."

The emphasis placed upon SPC within the CMMI is such that, an organization cannot exceed a CMMI process maturity level rating of three. The degree of emphasis on SPC in CMMI process is more than any other measurement and analysis technique.

#### IV. Conclusion

SPC helps in understanding and improvement of processes like manufacturing and refineries, where the process is highly repetitive, it works very well. This is because these processes have a fairly consistent about their inputs and processing methods. The usefulness of applying SPC to human/knowledge intensive processes like engineering/training is not very clear and is questioned by intellectuals.

The human intensive processes like software development have performances which can vary widely from each project and have unique characteristics. Implementation of SPC in CMMI process of software development requires that 21 out of 25 process areas in CMMI, which are categorized in the maturity level 2 or 3 be applied first. SPC is applied only in remaining four process areas at level 4 or 5. This bottom up approach has impact on overall software development improvement. Nevertheless, software development cannot be classified as pure SPC controlled process.

Multiple issues needs to be addressed by software development processes requires that development teams need to focus on following:

- (i) Team Management and creativity
- (ii) Multiple issues which crop up unaddressed during development
- (iii) Working with limited available data and analysis of this data.

Nevertheless it can be shown that SPC based processes at level 4 and 5 have definite positive impact on the quality, productivity and cost reduction. This suggests that software development is required to focus more on process rather be product centric. Software development processes are team effort and creativity of individual is involved, team leader also need to take into consideration motivation factor of the team. In addition, alignment of development process with in the business goals is very important for the success of development project/product. SPC can helps in guiding as to how to improve processes in certain defined approach.

#### References

- [1]. Donald J. Wheeler, David S. Chambers. Understanding Statistical Process Control, Second Edition. SPC Press, 1992.
- [2]. William A. Florac, Anita D. Carelton, Measuring the software Process : Statistical Process Control Improvement 1999.
- [3]. Department of Defense and US Army. Practical Software and Systems Measurement. Version 4.0b. October 2000.
- [4]. Jonathan Beckham, Six Sigma and CMMI : Showdown in Software Process Engineering, 2000
- [5]. Anita Carleton. Statistical Process Control for Software.
- [6]. Stephen H. Kan. Metrics and Models in Software Quality Engineering, Second Edition. Addison Wesley, 2003
- [7]. Ian Sommerville. Software Engineering, Seventh Edition. Addison Wesley, 2004. Quality Management Systems - Requirements, Third Edition, ISO 9001:2000(E) -2004
- [8]. Software Engineering Institute (SEI). Capability maturity model integrated (CMMI) for Development, Version 1.2. CMMI-DEV, V1.2, CMU/SEI-2006-TR-008, ESC-TR-2006-008. August 2006.
- [9]. Lawrence H. Putnam, Ware Myers. Measures For Excellence, Prentice Hall PTR, 1992.
- [10]. Fagan, M.E., Design and Code inspections to reduce errors in program development, IBM Systems Journal, 1976, Vol. 15, No 3, pp. 182-211
- [11]. Bob Raczynski and Bill Curtis. Software Data Violate SPC's Underlying Assumptions, IEEE Software, May/June 2008