7

Demystifying Six Sigma Metrics in Software

Ajit Ashok Shenvi
Philips Innovation Campus
India

1. Introduction

Design for Six Sigma (DFSS) principles have been proved to be very successful in reducing defects and attaining very high quality standards in every field be it new product development or service delivery. These Six sigma concepts are very tightly coupled with the branch of mathematics i.e. statistics. The primary metric of success in Six sigma techniques is the Z-score and is based on the extent of "variation" or in other words the standard deviation. Many a times, statistics induces lot of fear and this becomes a hurdle for deploying the six sigma concepts especially in case of software development. One because the digital nature of software does not lend itself to have "inherent variation" i.e. the same software would have exactly the same behavior under the same environmental conditions and inputs. The other difficult endeavor is the paradigm of samples. When it comes to software, the sample size is almost always 1 as it is the same software code that transitions from development phase to maturity phase. With all this, the very concept of "statistics" and correspondingly the various fundamental DFSS metrics like the Z-score, etc start to become fuzzy in case of software.

It is difficult to imagine a product or service these days that does not have software at its core. The flexibility and differentiation made possible by software makes it the most essential element in any product or service offering. The base product or features of most of the manufactures/service providers is essentially the same. The differentiation is in the unique delighters, such as intuitive user interface, reliability, responsiveness etc i.e. the non-functional requirements and software is at the heart of such differentiation. Putting a mechanism to set up metrics for these non-functional requirements itself poses a lot of challenge. Even if one is able to define certain measurements for such requirements, the paradigm of defects itself changes. For e.g. just because a particular use case takes an additional second to perform than defined by the upper specification limit does not necessarily make the product defective.

Compared to other fields such as civil, electrical, mechanical etc, software industry is still in its infancy when it comes to concepts such as "process control". Breaking down a software process into controlled parameters (Xs) and setting targets for these parameters using "Transfer function" techniques is not a naturally occurring phenomenon in software development processes.

This raises fundamental questions like –

- How does one approach the definition of software Critical To Quality (CTQs) parameters from metrics perspective?
- Are all software related CTQs only discrete or are continuous CTQs also possible?
- What kind of statistical concepts/tools fit into the Six Sigma scheme of things?
- How does one apply the same concepts for process control?
- What does it mean to say a product / service process is six sigma? And so on …

This chapter is an attempt to answer these questions by re-iterating the fundamental statistical concepts in the purview of DFSS methodology. Sharing few examples of using these statistical tools can be guide to set up six sigma metrics mechanisms in software projects.

This chapter is divided into 4 parts --

1. Part-1 briefly introduces the DFSS metrics starting from type of data, the concept of variation, calculation of Z-score, DPMO (defects per million opportunities) etc
2. Part-2 gives the general set up for using "inferential statistics" – concepts of confidence intervals, setting up hypothesis, converting practical problems into statistical problems, use of transfer function techniques such as Regression analysis to drill down top level CTQ into lower level Xs, Design of experiments, Gage R&R analysis. Some cases from actual software projects are also mentioned as examples
3. Part-3 ties in all the concepts to conceptualize the big picture and gives a small case study for few non-functional elements e.g.Usability, Reliability, Responsiveness etc
4. The chapter concludes by mapping the DFSS concepts with the higher maturity practices of the SEI-CMMI[R] model

The Statistical tool Minitab[R] is used for demonstrating the examples, analysis etc

2. DfSS metrics

2.1 The data types and sample size

The primary consideration in the analysis of any metric is the *"type of data"*. The entire data world can be placed into two broad types - qualitative and quantitative which can be further classified into *"Continuous"* or *"Discrete"* as shown in the figure-1 below.

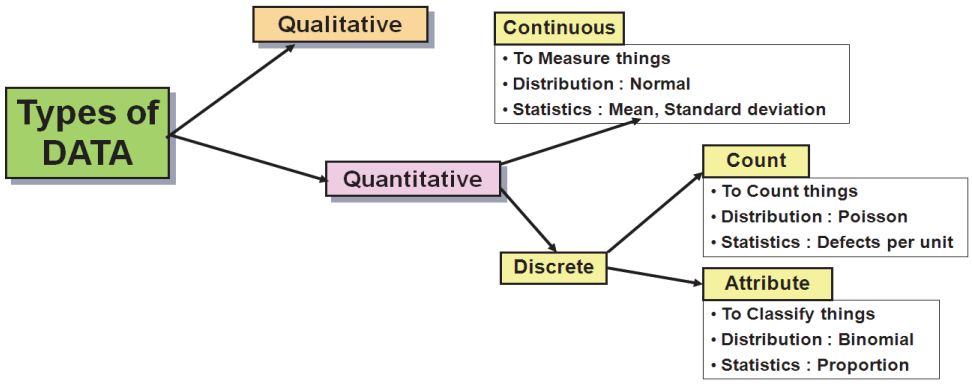


Fig. 1. The Different Data Types

The Continuous data type as the name suggests can take on any values in the spectrum and typically requires some kind of gage to measure. The Discrete data type is to do with counting/classifying something. It is essential to understand the type of data before getting into further steps because the kind of distribution and statistics associated vary based on the type of data as summarized in figure-1 above. Furthermore it has implications on the type of analysis, tools, statistical tests etc that would be used to make inferences/conclusions based on that data.

The next important consideration then relating to data is *"how much data is good enough"*. Typically higher the number of samples, the better is the confidence on the inference based on that data, but at the same time it is costly and time consuming to gather large number of data points.

One of the thumb rule used for Minimum Sample size (MSS) is as follows :-

- For Continuous data: $MSS = (2*Standard\ Deviation/\ Required\ Precision)^2$. The obvious issue at this stage is that the data itself is not available to compute the standard deviation. Hence an estimated value can be used based on historical range and dividing it by 5. Normally there are six standard deviations in the range of data for a typical normal distribution, so using 5 is a pessimistic over estimation.
- For Discrete-Attribute data : $MSS = (2/Required\ Precision)^2 *Proportion * (1-proportion)$. Again here the proportion is an estimated number based on historical data or domain knowledge. The sample size required in case of Attribute data is significantly higher than in case of Continuous data because of the lower resolution associated with that type of data.

In any case if the minimum sample size required exceeds the population then every data point needs to be measured.

2.2 The six sigma metrics

The word "Six-sigma" in itself indicates the concept of variation as *"Sigma"* is a measure of standard deviation in Statistics. The entire philosophy of Six Sigma metrics is based on the premise that *"Variation is an enemy of Quality"*. Too often we are worried only about "average" or mean however every human activity has variability. The figure-2 below shows the typical normal distribution and % of points that would lie between 1 sigma, 2 sigma and 3-sigma limits. Understanding variability with respect to *"Customer Specification"* is an essence of statistical thinking. The figure-3 below depicts the nature of variation in relation to the customer specification. Anything outside the customer specification limit is the ***"Defect"*** as per Six Sigma philosophy.

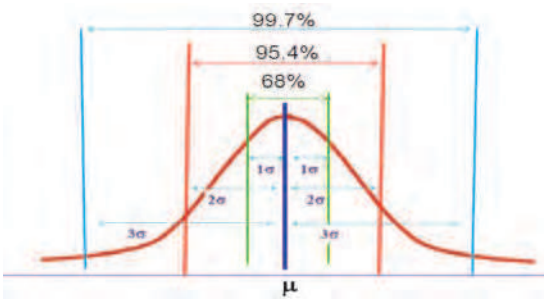


Fig. 2. Typical Normal Distribution

Fig. 3. Concept of Variation and Defects

2.2.1 The Z-score

Z-score is the most popular metric that is used in Six sigma projects and is defined as the *"number of standard deviations that can be fit between the mean and the customer specification limits"*. This is depicted pictorially in figure-4 below. Mathematically that can be computed as

$$Z = \frac{|CustomerSpecLimit - \mu|}{\sigma}$$

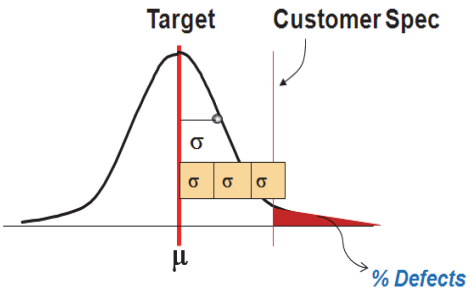


Fig. 4. The Z-score

So a *"3-Sigma"* process indicates 3 standard deviations can fit between mean and Specification limit. In other words if the process is centered (i.e. target and mean are equal) then a 3-sigma process has 6 standard deviations that can fit between the Upper Specification limit (USL) and Lower specification limit (LSL). This is important because anything outside the customer specification limit is considered a defect/defective. Correspondingly the Z-score indicates the area under the curve that lies outside Specification limits – in other words *"% of defects"*. Extrapolating the sample space to a million, the Z-score then illustrates the % of defects/defectives that can occur when a sample of million opportunities is taken. This number is called **DPMO** (Defects per million opportunities). Higher Z-value indicates lower standard deviation and corresponding lower probability of anything lying outside the specification limits and hence lower defects and vice-versa. This concept is represented by figure-5 below:

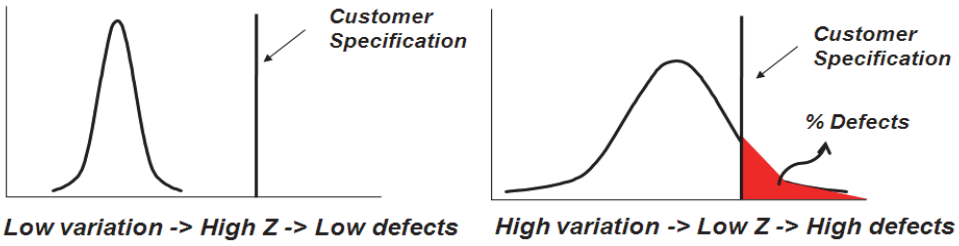


Fig. 5. Z-score and its relation to defects

By reducing variability, a robust product/process can be designed – the idea being with lower variation, even if the process shifts for whatever reasons, it would be still within the

customer specification and the defects would be as minimum as possible. The table-1 below depicts the different sigma level i.e. the Z scores and the corresponding DPMO with remarks indicating typical industry level benchmarks.

Z_{ST}	DPMO	Remarks
6	3.4	World-class
5	233	Significantly above average
4.2	3470	Above industry average
4	6210	Industry average
3	66800	Industry average
2	308500	Below industry average
1	691500	Not competitive

Table 1. The DPMO at various Z-values

Z-score can be a good indicator for business parameters and a consistent measurement for performance. The advantage of such a measure is that it can be abstracted to any industry, any discipline and any kind of operations. For e.g. on one hand it can be used to indicate performance of an "Order booking service" and at the same time it can represent the "Image quality" in a complex Medical imaging modality. It manifests itself well to indicate the quality level for a process parameter as well as for a product parameter, and can scale conveniently to represent a lower level Critical to Quality (CTQ) parameter or a higher level CTQ. The only catch is that the scale is not linear but an exponential one i.e. a 4-sigma process/product is not twice as better as 2-sigma process/product. In a software development case, the Kilo Lines of code developed (KLOC) is a typical base that is taken to represent most of the quality indicators. Although not precise and can be manipulated, for want of better measure, each Line of code can be considered an opportunity to make a defect. So if a project defect density value is 6 defects/KLOC, then it can be translated as 6000 DPMO and the development process quality can be said to operate at 4-sigma level.

Practical problem: "*Content feedback time*" is an important performance related CTQ for the DVD Recorder product measured from the time of insertion of DVD to the start of playback. The Upper limit for this is 15 seconds as per one study done on human irritation thresholds. The figure-6 below shows the Minitab menu options with sample data as input along with USL-LSL and the computed Z-score.

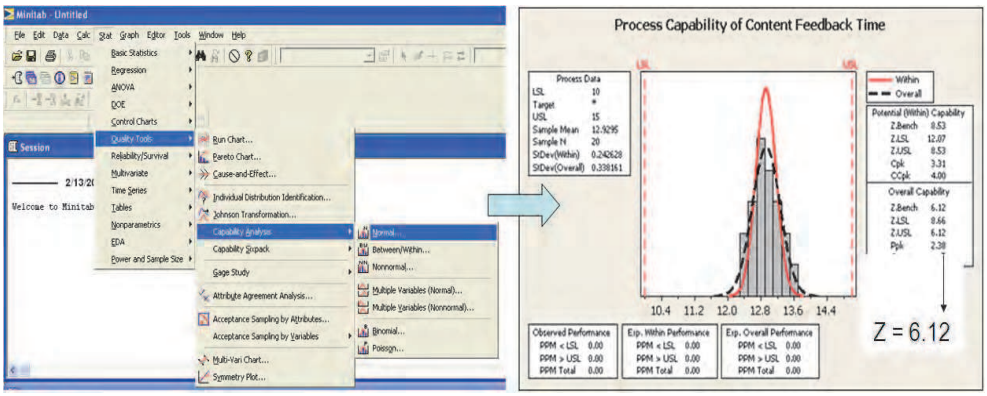


Fig. 6. Capability Analysis : Minitab menu options and Sample data

2.2.2 The capability index (Cp)

Capability index (Cp) is another popular indicator that is used in Six sigma projects to denote the relation between *"Voice of customer" to "Voice of process"*. Voice of customer (VOC) is what the process/product must do and Voice of process (VOP) is what the process/product can do i.e. the spread of the process.

$$Cp = VOC/VOP = (USL\text{-}LSL)/6\sigma$$

This relation is expressed pictorially by the figure-7 below

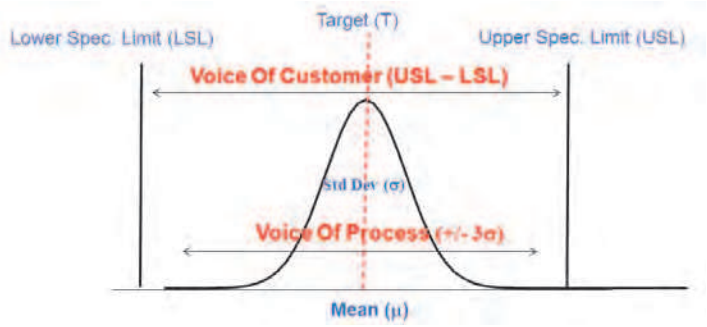


Fig. 7. Capability Index Definition

There is striking similarity between the definitions of Cp and the Z-score and for a centered normally distributed process the Z-score is 3 times that of Cp value. The table-2 below shows the mapping of the Z-score and Cp values with DPMO and the corresponding Yield.

Z_{ST}	DPMO	Cp	Yield
6	3.4	2	99.9997 %
5	233	1.67	99.977 %
4.2	3470	1.4	99.653 %
4	6210	1.33	99.38 %
3	66800	1	93.2 %
2	308500	0.67	69.1 %
1	691500	0.33	30.85 %

Table 2. Cp and its relation to Z-score

3. Inferential statistics

The *"statistics"* are valuable when the entire population is not available at our disposal and we take a sample from population to infer about the population. These set of mechanisms wherein we use data from a sample to conclude about the entire population are referred to as *"Inferential statistics"*.

3.1 Population and samples

"Population" is the entire group of objects under study and a *"Sample"* is a representative subset of the population. The various elements such as average/standard deviation

calculated using entire population are referred to as *"parameters"* and those calculated from sample are called *"statistics"* as depicted in figure-8 below.

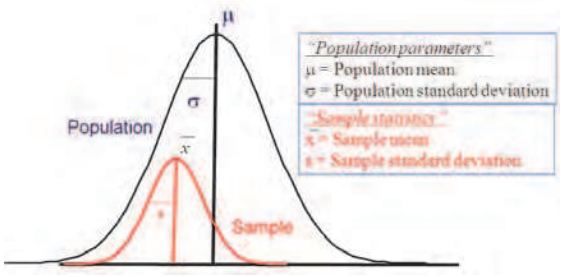


Fig. 8. Population and Samples

3.2 The confidence intervals

When a population parameter is being estimated from samples, it is possible that any of the sample A, B, C etc as shown in figure-9 below could have been chosen in the sampling process.

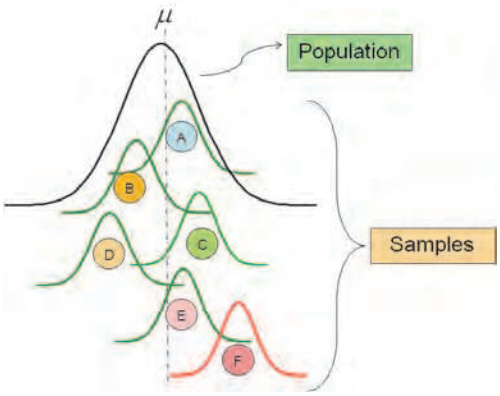


Fig. 9. Sampling impact on Population parameters

If the sample-A in figure-9 above was chosen then the estimate of population mean would be same as mean of sample-A, if sample B was chosen then it would have been the same as sample B and so on. This means depending on the sample chosen, our estimate of population mean would be varying and is left to chance based on the sample chosen. This is not an acceptable proposition.

From *"Central Limit theorem"* it has been found that for sufficiently large number of samples **n**, the *"means"* of the samples itself is normally distributed with mean at **μ** and standard deviation of **σ/sqrt (n)**.

Hence mathematically :

$$\mu = \bar{x} \pm z_{\alpha} s / \sqrt{n}$$

Where x is the sample mean, s is the sample standard deviation; α is the area under the normal curve outside the confidence interval area and z-value corresponding to α. This means that instead of a single number, the population mean is likely to be in a range with known level of confidence. Instead of assuming a statistics as absolutely accurate, "*Confidence Intervals*" can be used to provide a range within which the true process statistic is likely to be (with known level of confidence).

- All confidence intervals use samples to estimate a population parameter, such as the population mean, standard deviation, variance, proportion
- Typically the 95% confidence interval is used as an industry standard
- As the confidence is increased (i.e. 95% to 99%), the width of our upper and lower confidence limits will increase because to increase certainty, a wider region needs to be covered to be certain the population parameter lies within this region
- As we increase our sample size, the width of the confidence interval decreases based on the square root of the sample size: Increasing the sample size is like increasing magnification on a microscope.

Practical Problem: "*Integration & Testing*" is one of the Software development life cycle phases. Adequate effort needs to be planned for this phase, so for the project manager the 95% interval on the mean of % effort for this phase from historical data serves as a sound basis for estimating for future projects. The figure-10 below demonstrates the menu options in Minitab and the corresponding graphical summary for "*% Integration & Testing*" effort. Note that the confidence level can be configured in the tool to required value.

For the Project manager, the 95% confidence interval on the mean is of interest for planning for the current project. For the Quality engineer of this business, the 95% interval of standard deviation would be of interest to drill down into the data, stratify further if necessary and analyse the causes for the variation to make the process more predictable.

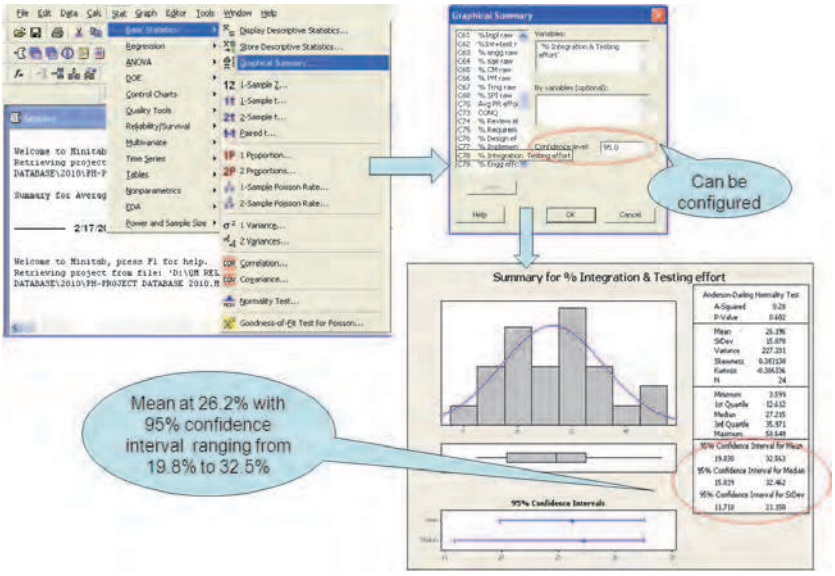


Fig. 10. Confidence Intervals : Minitab menu options and Sample Data

3.3 Hypothesis tests

From the undertsanding of Confidence Intervals, it follows that there always will be some error possible whenever we take any statistic. This means we cannot prove or disprove anything with 100% certainity on that statistic. We can be 99.99% certain but not 100%. *"Hypothesis tests"* is a mechanism that can help to set a level of certainity on the observations or a specific statement. By quantifying the certainity (or uncertainity) of the data, hypothesis testing can help to eliminate the subjectivity of the inference based on that data. In other words, this will indicate the *"confidence"* of our decision or the quantify risk of being wrong. The utility of hypothesis testing is primarily then to infer from the sample data as to whether there is a change in population parameter or not and if yes with what level of confidence. Putting it differently, hypothesis testing is a mechanism of minimizing the inherent risk of concluding that the population has changed when in reality the change may simply be a result of random sampling. Some terms that is used in context of hypothesis testing:

- **Null Hypothesis – H_o** : This is a statement of no change
- **Alternate Hypothesis** - **H_a** : This is the opposite of the Null Hypothesis. In other words there is a change which is statistically significant and not due to randomness of the sample chosen
- **α-risk** : This is risk of finding a difference when actually there is none. Rejecting H_o in a favor of H_a when in fact H_o is true, *a false positive*. It is also called as *Type-I error*
- **β-risk** : This is the risk of not finding a difference when indeed there is one. Not rejecting H_o in a favor of H_a when in fact H_a is true, *a false negative*. It is also called as *Type-II error*.

The figure-11 below explains the concept of hypothesis tests. Referring to the figure-11, the X-axis is the *Reality or the Truth* and Y-axis is the *Decision* that we take based on the data.

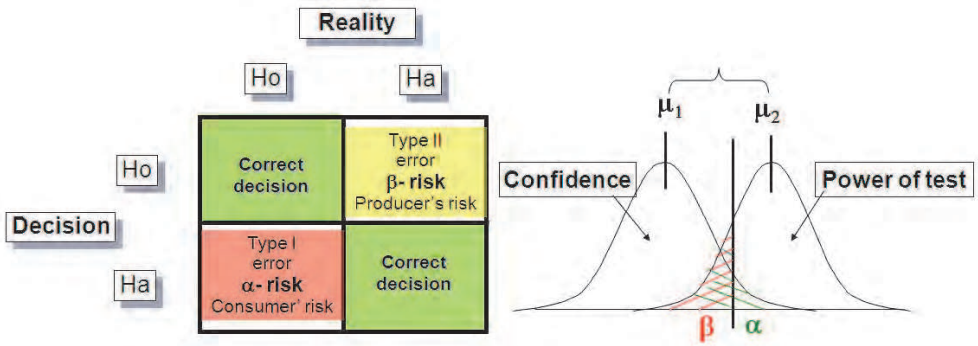


Fig. 11. Concept of Hypothesis Tests

If *"in reality"* there is no change (Ho) and the *"decision"* based on data also we infer that there is no change then it is a correct decision. Correspondingly *"in reality"* there is a change and we conclude also that way based on the data then again it is a correct decision. These are the boxes that are shown in green color (top-left & bottom-right) in the figure-11.

If *"in reality"* there is no change (H_o) and our *"decision"* based on data is that there is change(H_a), then we are taking a wrong decision which is called as *Type-I error*. The risk of

such an event is called as *α-risk* and it should be as low as possible. *(1-α)* is then the "*Confidence*" that we have on the decision. The industry typical value for α risk is 5%.
If "*in reality*" there is change (H_a) and our "*decision*" based on data is that there is no change (H_o), then again we are taking a wrong decision which is called a *Type-II error*. The risk of such an event is called as *β-risk*. This means that our test is not sensitive enough to detect the change; hence *(1-β)* is called as "*power of test*".
The right side of figure-11 depicts the old and the new population with corresponding α and β areas.
Hypothesis tests are very useful to prove/disprove the *statistically significant change* in the various parameters such as mean, proportion and standard deviation. The figure-12 below shows the various tests available in Minitab tool for testing with corresponding menu options list.

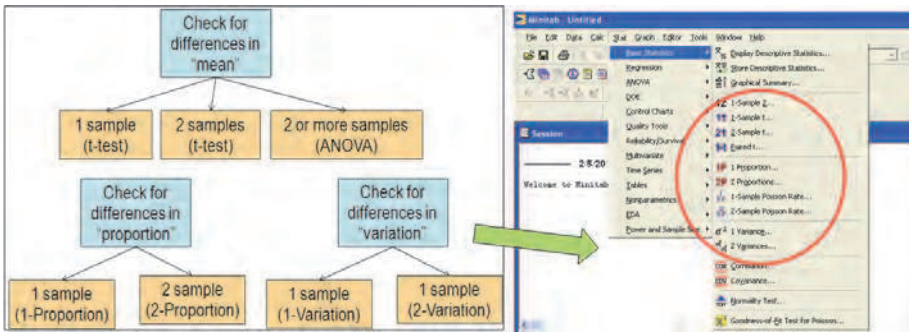


Fig. 12. The Various Hypothesis Tests and the Minitab Menu options

3.3.1 One-sample t-test

1-sample t-test is used when comparing a sample against a target mean. In this test, the null hypothesis is "*the sample mean and the target are the same*".
Practical problem: The "*File Transfer speed*" between the Hard disk and a USB (Universal Serial Bus) device connected to it is an important Critical to Quality (CTQ) parameter for the DVD Recorder product. The target time for a transfer of around 100 files of average 5 MB should not exceed 183 seconds.
This is a case of 1-Sample test as we are comparing a sample data to a specified target.
Statistical problem :
Null Hypothesis ---------- H_o: $\mu_a = 183$ sec
Alternate Hypothesis ------- H_a: $\mu_a > 183$ sec or H_a: $\mu_a < 183$ sec or H_a: $\mu_a \neq 183$ sec
Alpha risk ---------- $\alpha = 0.05$
The data is collected for atleast 10 samples using appropriate measurement methods such as stop-watch etc. The figure-13 below shows the menu options in Minitab to perform this test. After selecting 1-sample T-test, it is important to give the "*hypothesized mean*" value. This is the value that will be used for Null hypothesis. The "options" tab gives text box to input the Alternative hypothesis. Our H_a is H_a: $\mu_a > 183$ seconds. We select "*greater than*" because Minitab looks at the sample data first and then the value of 183 entered in the "Test Mean". It is important to know how Minitab handles the information to get the "Alternative hypothesis" correct.

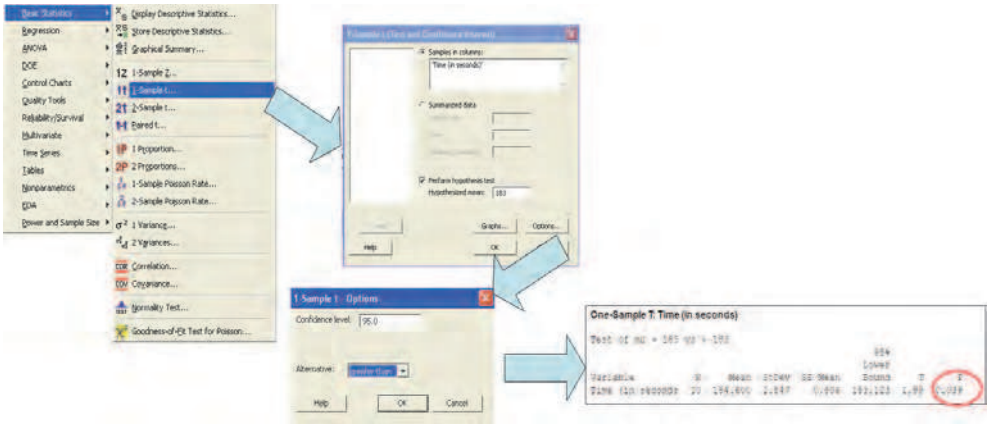


Fig. 13. 1-Sample t-test : Minitab menu options and Sample Results

- The test criteria was $\alpha = 0.05$, which means we were willing to take a 5% chance of being wrong if we rejected Ho in favor of Ha
- The Minitab results show the p-value which indicates there is only a 3.9% chance of being wrong if we reject Ho in favor of Ha
- 3.9% risk is less than 5%; therefore, we are willing to conclude Ha. The file-transfer, on average, is taking longer than 183 seconds between USB-Hard Disk

The same test would be performed again after the improvements were done to confirm the statistically significant improvement in the file-transfer performance is achieved.

3.3.2 Two-sample t-test

2-sample t-test can be used to check for statistical significant differences in "means" between 2 samples. One can even specify the exact difference to test against. In this test, the null hypothesis is "*there is no difference in means between the samples*".

Practical problem : The "*Jpeg Recognition Time*" is another CTQ for the DVD recorder product. The system (hardware+software) was changed to improve this perfromance. From our perspective the reduction in average recognition time has be more than 0.5 sec to be considered significant enough from a practical perspective.

This is a case of 2-Sample test as we are comparing two independent samples.

Statistical problem :

Null Hypothessis ---------- H_o: $\mu_{Old} - \mu_{New} = 0.5$ sec

Alternate Hypothesis ------- H_a: $\mu_{Old} - \mu_{New} > 0.5$ sec

Alpha risk ---------- $\alpha = 0.05$

The data is collected for atleast 10 samples using appropriate measurement methods for the old and the new samples.

The figure-14 below shows the menu options in Minitab to perform this test. After selecting 2-sample T-test, either the summarized data of samples can be input or directly the sample data itself. The "options" tab gives box to indicate the Alternative hypothesis. Based on what we have indicated as sample-1 and sample-2, the corresponding option of "*greater than*" *or* "*less than*" can be chosen. It also allows to specify the "*test difference*" that we are looking for which is 0.5 seconds in this example.

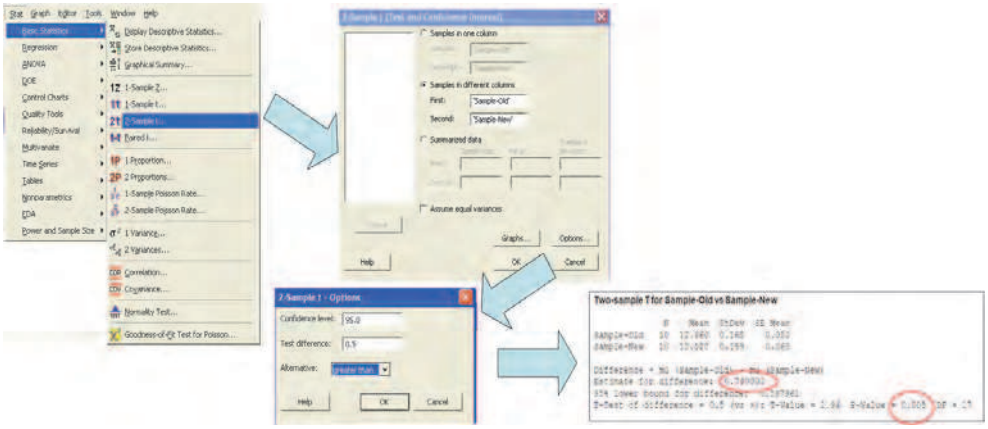


Fig. 14. 2-Sample t-test : Minitab menu options and Sample Results

- The criteria for this test was $\alpha = 0.05$, which means we were willing to take a 5% chance of being wrong if we rejected H_o in favor of H_a
- The Minitab results show the p-value which indicates there is only a 0.5% chance of being wrong if we reject H_o in favor of H_a
- 0.5% risk is less than 5%; therefore, we are willing to conclude H_a. The Sample-New has indeed improved the response time by more than 0.5 seconds
- The estimate for that difference is around 0.74 seconds

The above two sections has given some examples of setting up tests for checking differences in mean. The philosophy remains the same when testing for differences in "proportion" or "Variation". Only the statistic behind the check and the corresponding test changes as was shown in the figure-12 above.

3.4 Transfer functions

An important element of design phase in a Six sigma project is to break down the CTQs (Y) into lower level inputs (Xs) and a make a *"Transfer Function"*. The purpose of this transfer function is to identify the *"strength of correlation"* between the *"Inputs (Xs)"* and output (Y) so that we know where to focus the effort in order to optimise the CTQ. The purpose of this exercise also is to find those inputs that have an influence on the output but cannot be controlled. One such category of inputs is *"Constants or fixed variables (C)"*and other category is *"Noise parameters (N)"*. Both these categories of inputs impact the output but cannot be controlled. The only difference between the Constants and the Noise is the former has always a certain fixed value e.g. gravity and the latter is purely random in nature e.g. humidity on a given day etc.

There are various mechanisms to derive transfer functions such as regression analysis, Design of experiments or as simple as physical/mathematical equations. These are described in the below sections.

3.4.1 Physics/Geometry

Based on the domain knowledge it is possible to find out the relationship between the CTQ (Y) and the factor influencing it (Xs). Most of the timing/distance related CTQs fall under

this category where total time is simply an addition of its sub components. These are called as *"Loop equations"*. For e.g.
Service time(Y) = Receive order(x1) +Analyse order(x2) +Process order(x3) +Collect payment (x4)
Some part of the process can happen in parallel. In such cases
Service time(Y)=Receive order(x1)+Analyse order(x2)+Max(Process order(x3), Collect payment(x3))

Practical problem :

"Recording duration" (i.e. number of hours of recording possible) is one of the CTQs for the DVD recorder as dictated by the marketing conditions/competitor products. The size of hard disk is one of the factors influencing the duration. Each additional space comes at a cost hence it is important to optimise that as well. The transfer function in this case is the one that translates available memory space (in Gigabytes) into time (hours of recording). From domain knowledge this translation can be done using audio bit rate and video bit rate as follows:

$$b = ((video\_bitrate * 1024 * 1024)/8) + ((audio\_bitrate*1024)/8) \text{ bytes}$$

$$k = b/1024 \text{ kilobytes}$$

$$\text{no. of hrs of recording} = ((space\_in\ GB)*1024*1024)/(k*3600)$$

3.4.2 Regression analysis

"Regression Analysis" is a mechanism of deriving transfer function when historical data is available for both the Y and the Xs. Based on the scatter of points, regression analysis computes a best fit line that represents the relation of X to Y minimizing the *"residual error"*.

Practical Problem:

"Cost of Non-Quality (CONQ)" is a measure given to indicate the effort/cost that is spent on rework. If it was "right" the first time this effort could have been saved and maybe utilised for some other purpose. In a software development scenario, because there are bugs/issues lot of effort is spent on rework. Not only it is additional effort due to not being right the first time, but also modifying things after it is developed always poses risks due to regression effects. Hence CONQ is a measure of efficiency of the software development process as well as indirect measure for first-time-right quality. Treating it as CTQ (Y), the cause-effect diagram in figure-15 below shows the various factors (Xs) that impact this CONQ. This is not an exhaustive list of Xs and there could be many more based on the context of the project/business.

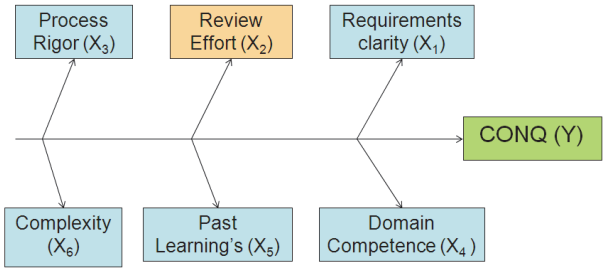


Fig. 15. Factors Impacting CONQ

Since lot of historical data of past projects is available, regression analysis would be a good mechanism to derive the transfer function with Continuous Y and Continuous Xs. Finding the relation between Y and multiple Xs is called *"Multiple Regression"* and that with single X is referred to as *"Simple Regression"*. It would be too complicated to do the analysis with all Xs at the same time; hence it was decided to choose one of the Xs in the list that has a higher impact, which can be directly controlled and most importantly which is "continuous" data for e.g. Review effort. The figure-16 below shows the Regression model for CONQ.

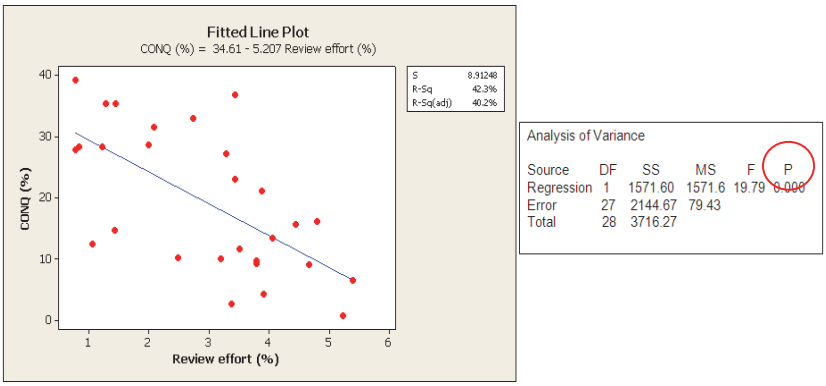


Fig. 16. The Regression Analysis for CONQ

When concluding the regression equation, there are 4 things that need to be considered:-

1. The p-value. The Null hypothesis is that *"there is no correlation between Y and X"*. So if p-value < α, then we can safely reject Null and accept the Alternate, which is that Y and X are correlated. In this case p-value is 0, this means that we can conclude that the regression equation is statistically significant
2. Once the p-value test is passed, the next value to look at is $R^2(adj)$. This signifies that the amount of variability of Y that is explained by the regression equation. Higher the R^2 better it is. Typical values are > 70%. In this case, R^2(adj) value is 40%. This indicates that only 40% of variability in CONQ is explained by the above regression equation. This may not be sufficient but in R&D kind of situation especially in software, where the number of variables are high, R^2(adj) value of 40% and above could be considered a reasonable starting point
3. The third thing is then to look at the residuals. A *Residual* is the error between the fitted line (regression equation) and the individual data points. For the regression line to be un-biased, the residuals themselves must be normally distributed (random). A visual inspection of the residual plots as shown in figure-17 below can confirm that e.g. a lognormal plot of residuals should follow a straight line on the "normal probability plot" and residuals should be either side of 0 in the "versus fits" plot. The "histogram" in the residual plot can also be good indication.
4. Once the above 3 tests pass, the regression equation can be considered statistically significant to predict the relations of X to Y. However one important point to note is the *"range of values for X"* under which this equation is applicable. For e.g. the above CONQ equation can be used only in the range of Review % from 0 to 6% as the regression analysis was done with that range.

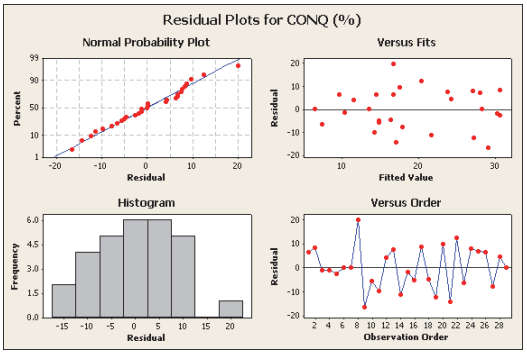


Fig. 17. Residual Analysis for CONQ

The project manager can now use the above regression equation to plan the % review effort in the project based on the target CONQ value. If there is more than 1 X impacting Y, then doing simple regression is not adequate as there could be lot of interaction effects of those Xs (X1, X2...) on Y. Hence it is advisable to do a *"Multiple Regression"* analysis in such cases. The philosophy remains the same for multiple regression, with only one change that p-value test now needs to be checked for each of the Xs in the regression summary.

3.4.3 Design of experiments (DOE)

Design of Experiments (DOE) is a concept of organizing a set of experiments where-in each individual X input is varied at its extreme points in a given spectrum keeping the other inputs constant. The effect on Y is observed for all the combinations and the transfer function is computed based on the same.

Practical Problem:

DVD-recorder has a USB port which can be used to connect digital cameras to view/copy the pictures. *"Jpg Recognition Time"* is a product CTQ which is crucial from a user perspective and the upper specification limit for which is 6 seconds. The Xs that impact the Jpg Recognition time CTQ from a brain storming exercise with domain experts are shown in figure-18 below.

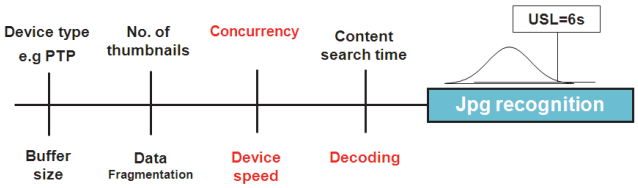


Fig. 18. The Factors Impacting JPG Recognition

Device speed in this case is the speed of USB device connected to the recorder and is then a discrete X which can take 4 values for e.g. USB 1.0 (lowest speed) to USB 2.0 device (highest speed).

Decoding is again a discrete X and can take 4 possible values – completely software, 70-30 software-hardware, 30-70 software-hardware, or completely hardware solution.

Concurrency is number of parallel operations that can be done at the same time and is also a discrete X. In this particular product up to 5 concurrencies are allowed.
"CPU Load" is another CTQ which is a critical for the reliable operation of the product. It is known from embedded software experience that a CPU load of > 65% makes the system unstable hence the USL is placed at 60%. A CPU load of <40% is not an efficient utilization of a costly resource such as CPU. Hence the LSL is defined to be 40%. The factors (Xs) that correlate to this CTQ i.e. CPU load are shown in the figure-19 below.

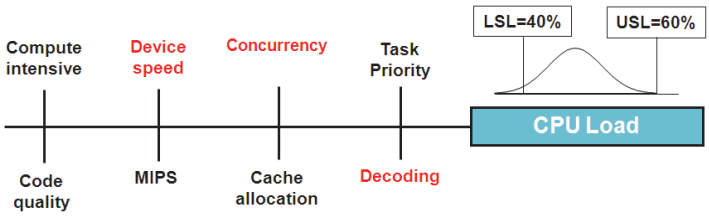


Fig. 19. The Factors Impacting CPU Load

It is interesting to note two things from figure-18 and figure-19 above:-

a. There are 3 factors (Xs) that are common to both the CTQs (Device speed, Decoding and Concurrency)
b. Some of the Xs are continuous such as Search time, buffer size, Cache etc and some others are Discrete such as Concurrency, Task priority etc. DOE is an excellent mechanism in these circumstances where there is a mix of discrete and continuous Xs. Also the focus now is not so much on the exact transfer function but more than "*Main effects plot*" (impact of individual Xs on Y) and "*Interaction Plots*" (impact of multiple Xs having a different impact on Y).

The figure-20 represents the DOE matrix for both these CTQs along with the various Xs and the range of values they can take.

C5	C6	C7	C8	C9
Concurrency	Decoding	Device speed	CPU Load	Jpg Recognition
1	1	1	30	5
5	1	1	60	10
1	4	1	20	3
5	4	1	40	6
1	1	4	60	4
5	1	4	90	7
1	4	4	40	1
5	4	4	60	10

Concurrency (numeric)
Low – minimum – 1
High – maximum - 5

Decoding (can be text or numeric)
Low – only software - 1
High – only hardware - 4

Device speed (text or numeric)
Low – minimum - 1
High – maximum - 4

Fig. 20. The DOE Matrix for CPU Load and JPG Recognition

The transfer function for both the CTQs from the Minitab DOE analysis are as below :-
CPU Load = 13.89 + 8.33*Concurrency – 1.39*Decoding + 11.11*Device-speed – 0.83*Concurrency*Decoding – 1.11*Decoding*Device-speed
Jpg Recognition = 4.08 + 1.8*Concurrency – 0.167*Decoding + 0.167*Device-Speed – 0.39*Concurrency*Decoding – 0.389*Concurrency*Device-Speed
Our aim is to achieve a "nominal" value for CPU load CTQ and "as low as possible" value for Jpg recognition CTQ. The transfer functions themselves are not important in this case as are the Main effects plots and Interaction plots as shown in figure-21 and figure-22 below

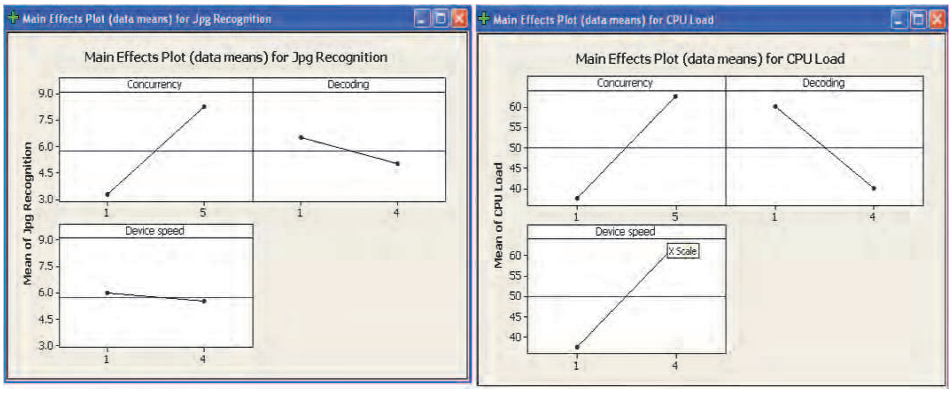


Fig. 21. Main Effects Plots for JPG Recognition and CPU Load

It is evident from the Main effects plots in the figure-21 above the impact of each of the Xs on the corresponding Ys. So a designer can optimise the corresponding Xs to get the best values for the respective Ys. However it is also interesting to note that some Xs have an opposite effect on the 2 CTQs. From figure-21 above – On one hand a Device speed of 4 (i.e. USB 2.0) is the best situation for Jpg recognition CTQ but it is worst case for CPU load CTQ on the other hand. In other words, the Device speed X impacts both the CTQs in a contradictory manner. The Interaction plots shown in figure-22 come in handy during such cases, where one can find a different X that interacts with this particular X in such a manner that the overall impact on Y is minimized or reduced i.e. "*X1 masks the impact of X2 on Y*".

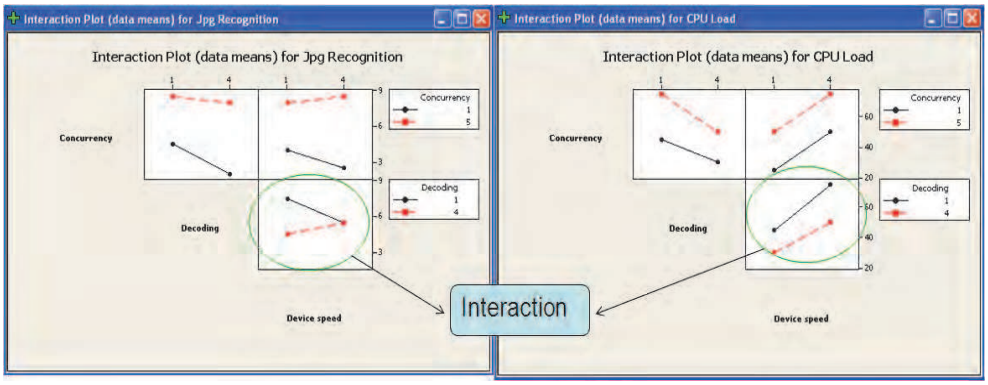


Fig. 22. Interaction Plots for JPG Recognition and CPU Load

From the figure-22 above it is seen that the Device speed X interacts strongly with Decoding X. Hence Device speed X can be optimised for Jpg recognition CTQ, and Decoding X can be used to mask the opposing effect of Device speed X on CPU load CTQ.

With "*Response optimizer*" option in Minitab, it is possible to play around with the Xs to get the optimum and desired values for the CTQs. Referring to Figure-23 below, with 3 concurrencies and medium device speed and hardware-software decoding, we are able to achieve CPU load between 30% and 50% and Jpg recognition time of 5.5s

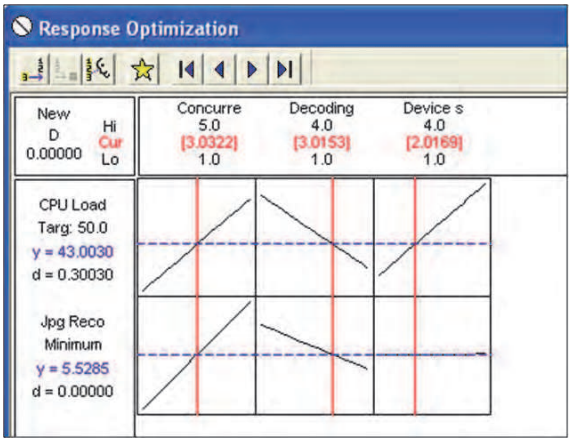


Fig. 23. The Response Optimiser for CPU Load and JPG Recognition

3.5 Statistical process control (SPC)

SPC is an *"Electrocardiogram"* for the process or product parameter. The parameter under consideration is measured in a time ordered sequence to detect shift or any unnatural event in the process. Any process has variation and the control limits (3-sigma from mean on both sides) determine the extent of *natural variation* that is inherent in the process. This is referred to as *"common cause of variation"*. Any point lying outside the control limits (UCL – upper control limit and LCL – lower control limit) indicates that the process is *"out of control/unstable"* and is due to some assignable cause that is referred to as the *"special cause of variation"*. The special cause necessitates a root cause analysis and action planning to bring back process back to control. The figure-24 below shows the SPC concept along with the original mean and the new mean after improvement. Once the improvement is done on the CTQ and the change is confirmed via the hypothesis test, it needs to be monitored via a SPC chart to ensure the *stability* of the same over a long term.

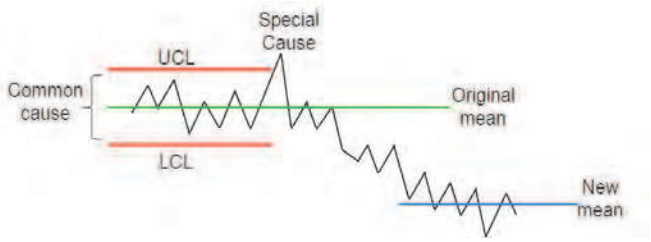


Fig. 24. SPC – Common Cause and Special Cause

It is important to understand that the Control limits are not the same as Specification limits. Control limits are computed based on historical data spread of the process/product performance whereas Specification limits come from Voice of customer. A process may be in control i.e. within control limits but not be capable to meet specification limits. The first step should be always bring the process *"in control"* by eliminating special cause of variation and then attain *"capability"*. It is not possible to achieve process capability (i.e. to be within specification limits) when the process itself is out of control.

Once the CTQ has attained the performance after the improvement is done, it is required to monitor the same via some appropriate SPC chart based on the type of data as indicated in the figure-25 below along with the corresponding Minitab menu options.

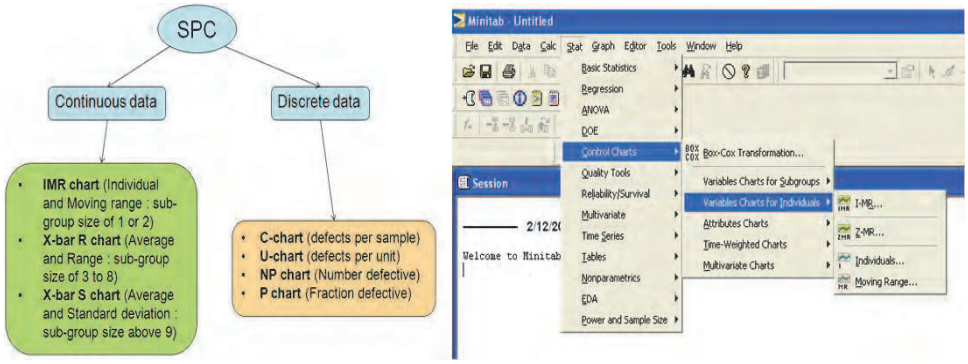


Fig. 25. The Various SPC harts and Minitab menu options

Practical Problem:

"Design Defect density" is a CTQ for a software development activity and number of improvements has been done to the design review process to increase design defect yield. So this CTQ can be monitored via an I-MR chart as depicted in figure-26 below. Any point outside the control limits would indicate an unnatural event in the design review process.

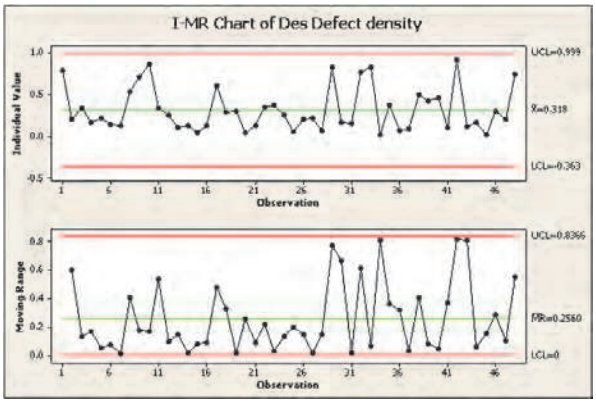


Fig. 26. The I-MR chart for defect density

3.6 Measurement system analysis

All decisions in a Six sigma project are based on data. Hence it is extremely crucial to ascertain that the measurement system that is used to measure the CTQs does not introduce error of its own. The measurement system here is not only the gage that is used to measure but also the interaction of inspectors and the gage together that forms the complete system. The study done to determine the health of the measurement system is called *"Gage Repeatability and Reproducibility (Gage R&R)"*. Repeatability refers to *"how repeatable are the*

measurements made by one inspector" and Reproducibility indicates *"how reproducible are the measurements made by several inspectors"*. Both repeatability and reproducibility introduces its own set of variation in the total variation. The figure-27 below depicts this relation.

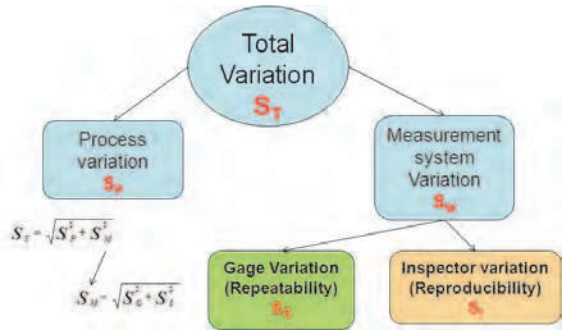


Fig. 27. The Measurement System Analysis : Variation

Since all the decisions are based on the data, it would be a futile attempt to work on a CTQ which has high variation when actually the majority of this is due to the measurement system itself. Hence there is a need to separate out the variation caused by the measurement system by doing an experiment of the measuring few already known standard samples with the gage and inspectors under purview. A metric that is computed as result is called *"%Tolerance GageR&R"* and is measured as *(6*S_M *100)/ (USL-LSL)*. This value should be less than 20% for the Gage to be considered acceptable.

Practical Problem:

There are many timing related CTQs in the Music Juke box player product and stop-watch is the gage used to do the measures. An experiment was set up with a stop watch and known standard use cases with set of inspectors. The results are analysed with Minitab Gage R&R option as shown in figure-28 along with the results.

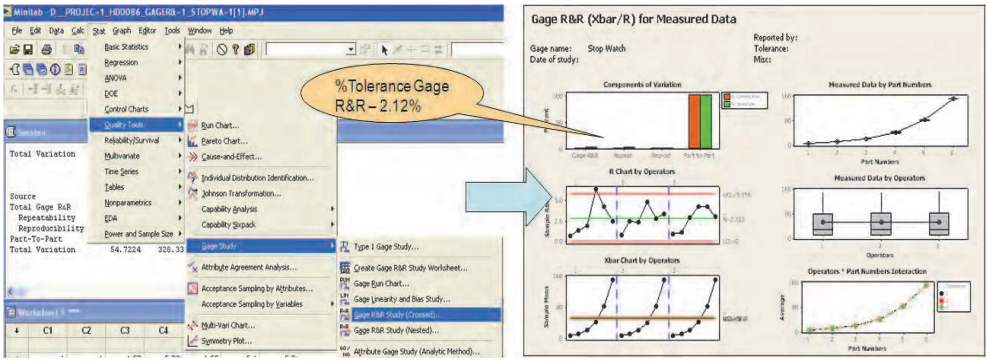


Fig. 28. Gage R&R Analysis : Minitab menu options and Sample results

The Gage R&R gives the total Measurement system variation as well as Repeatability and Reproducibility component of the total variation.

4. Tying It together – the big picture

In the previous sections we have seen number of statistical concepts with number of examples explaining those concepts. The overall big picture of a typical Six sigma project with these statistical concepts can be summarised as depicted in the figure-29 below.

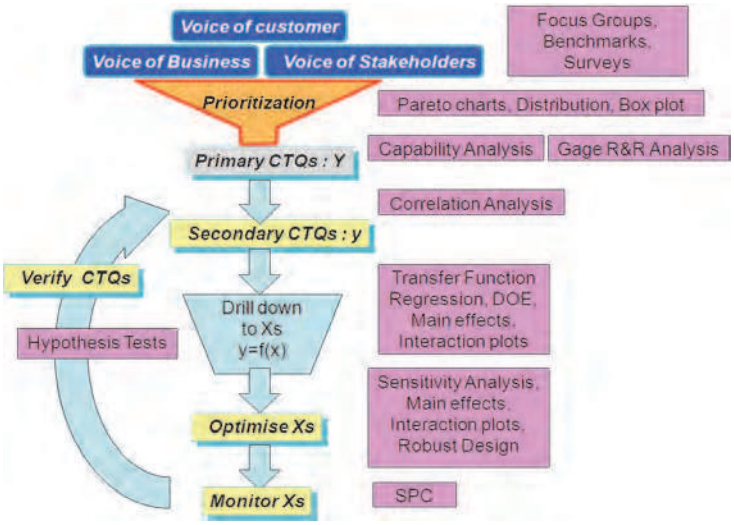


Fig. 29. Snapshot of Statistical Mechanisms in a DFSS project

The Starting point is the always the *"Voice of customer or Voice of Business or Voice of stakeholders"*. Concepts like Focus groups interviews, Surveys, Benchmarking etc can be used to listen and conceptualize this "Voice". It is important to understand this "Voice" correctly otherwise all the further steps become futile.

Next this *"Voice of customer"* i.e. the customer needs have to be prioritised and translated into specific measurable indicators i.e. the *"Primary CTQs (Y)"*. Tools like Frequency distributions, Box plots, Pareto charts can be some of the techniques to do the prioritisation. Capability analysis can indicate the current capability in terms of *Z-score/Cp* numbers and also help set targets for the six sigma project. This is the right time to do a measurement system analysis using Gage R&R techniques.

The lower level CTQs i.e. the *"Secondary CTQs (y)"* can then be identified from Primary CTQs using techniques such as Correlation analysis. This exercise will help focus on the few vital factors and eliminate the other irrelevant factors.

Next step is to identify the Xs and find mathematical *"Transfer function"* relating the Xs to the CTQs (y). Regression Analysis, DOEs are some of the ways of doing this. In many cases especially software, often the transfer function itself may not be that useful, but rather the *"Main effects and Interaction plots"* would be of more utility to select the Xs to optimise.

"Sensitivity Analysis" is the next step which helps distribute the goals (mean, standard deviation) of Y to the Xs thus setting targets for Xs. Certain Xs would be noise parameters and cannot be controlled. Using *"Robust Design Techniques"*, the design can be made insensitive to those noise conditions.

Once the Xs are optimised, *"SPC charts"* can be used to monitor them to ensure that they are stable. Finally the improvement in the overall CTQ needs to be verified using *"Hypothesis tests"*.

4.1 The case study

DVD-Hard disk recorder is a product that plays and records various formats such as DVD, VCD and many other formats. It has an inbuilt hard disk that can store pictures, video, audio, pause the live-TV and resume it later from the point it was paused etc. The product is packed with more than 50 features with many use cases in parallel making it very complicated. Also because of the complexity, the intuitiveness of user-interface assumes enormous importance. There are many "Voices of customer" for this product – Reliability, Responsiveness and Usability to name a few.

4.1.1 Reliability

One way to determine software reliability would be in terms of its robustness. We tried to define *Robustness* as CTQ for this product and measured it in terms of *"Number of Hangs/crashes"* in normal use-case scenarios as well as stressed situations with target as 0.
The lower level factors (X's) affecting the CTQ robustness were then identified as:

- Null pointers, Memory leaks
- CPU loading, Exceptions/Error handling
- Coding errors

Robustness = f (Null pointers, Memory leaks, CPU load, Exceptions, Coding errors)
The exact transfer function in this case is irrelevant as all the factors are equally important and need to be optimized.

4.1.2 Responsiveness

The CTQs that would be directly associated with *"Responsiveness"* voice are the Timing related parameters. For such CTQs, the actual transfer functions really make sense as they are linear in nature. One can easily decide from the values itself the Xs that need to be optimized and by how much. For e.g.
Start-up time(y) = drive initialization(x1) + software initialization(x2) + diagnostic check time(x3)

4.1.3 Usability

Usability is very subjective parameter to measure and very easily starts becoming a discrete parameter. It is important that we treat it as a continuous CTQ and spend enough time to really quantify it in order to be able to control its improvement.
A small questionnaire was prepared based on few critical and commonly used features and weightage was assigned to them. A consumer experience test was conducted with a prototype version of product. Users with different age groups, nationality, gender, educational background were selected to run the user tests. These tests were conducted in home-like environment set-up so that the actual user behaviour could be observed.
The ordinal data of user satisfaction was then converted into a measurable CTQ based on the weightage and the user score. This CTQ was called as *"Usability Index"*. The Xs impacting this case are the factors such as Age, Gender etc. The interaction plot shown in the figure-30 below helped to figure out and correct a lot of issues at a design stage itself.

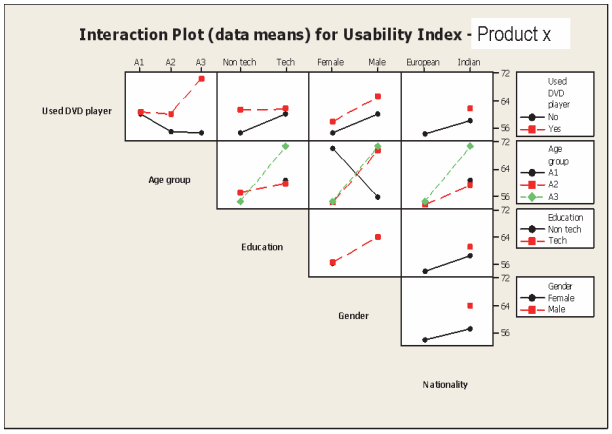


Fig. 30. Interaction Plot for Usability

5. Linkage to SEI-CMMI[R]

Level-4 and Level-5 are the higher maturity process areas of CMMI model and are heavily founded on statistical principles. Level 4 is the *"Quantitatively Managed"* maturity level which targets "special causes of variation" in making the process performance stable/predictable. Quantitative objectives are established and process performance is managed use these objectives as a criteria. At Level 5 called as *"Optimizing"* maturity level, the organization focuses on "common causes of variation" in continually improving its process performance to achieve the quantitative process improvement objectives. The process areas at Level-4 and Level-5 which can be linked to six sigma concepts are depicted in figure-31 below with the text of the specific goals from the SEI documentation

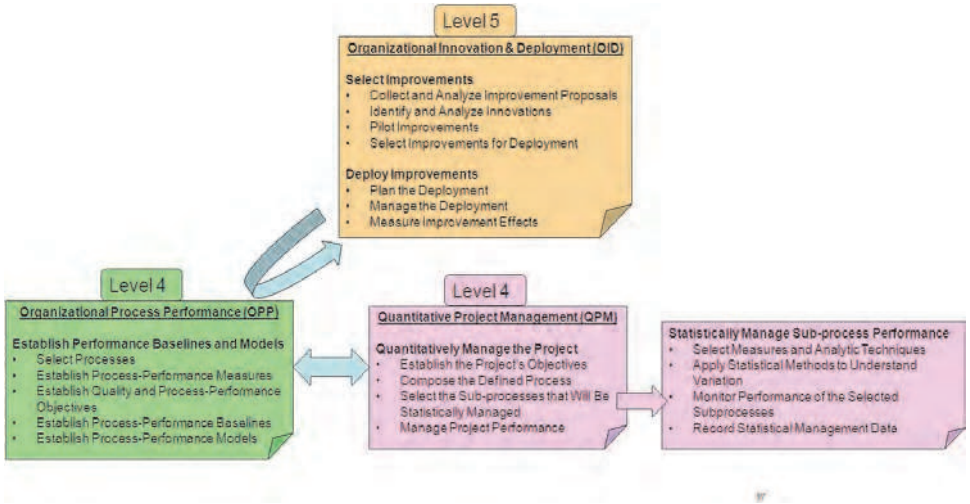


Fig. 31. The CMMI Higher Maturity Process areas

A typical example of the linkage and use of various statistical concepts for OPP, QPM and OID process areas of CMMI is pictorially represented in figure-32 below. In each of the process areas, the corresponding statistical concepts used are also mentioned.
One of the top-level Business CTQ (Y) is the "*Customer Feedback*" score which is computed based on a number of satisfaction questions around cost, quality, timeliness that is solicited via a survey mechanism. This is collected from each project and rolled up to business level. As shown in the figure-32 below, the mean value was 8 on a scale of 1-10 with a range from 7.5 to 8.8. The capability analysis is used here to get the 95% confidence range and a Z-score. The increase in feedback score represents increase in satisfaction and correspondingly more business. Hence as an improvement goal, the desired feedback was set to 8.2. This is part of OID part as depicted in figure-32 below.
Flowing down this CTQ, we know that "*Quality and Timeliness"* are the 2 important drivers that influence the score directly; hence they are lower level CTQs (y) that need to be targeted if we need to increase the satisfaction levels.
Quality in software projects is typically the *Post Release defect density* measured in terms of defects/KLOC. Regression analysis confirms the negative correlation of post release defect density to the customer feedback score i.e. lower the density, higher is the satisfaction.
The statistically significant regression equation is
Cust F/b = 8.6 – 0.522*Post Release Defect Density.
Every 1 unit reduction in defect density can increase the satisfaction by 0.5 units. So to achieve customer feedback of 8.2 and above the post release defect density needs to be contained within 0.75 defects/KLOC. This becomes the Upper spec limit for the CTQ (y) Post release defect density. The current value of this CTQ is 0.9 defects/KLOC. From OPP perspective it is also necessary to further break down this CTQ into lower level Xs and the corresponding sub-processes to control statistically to achieve the CTQ y.

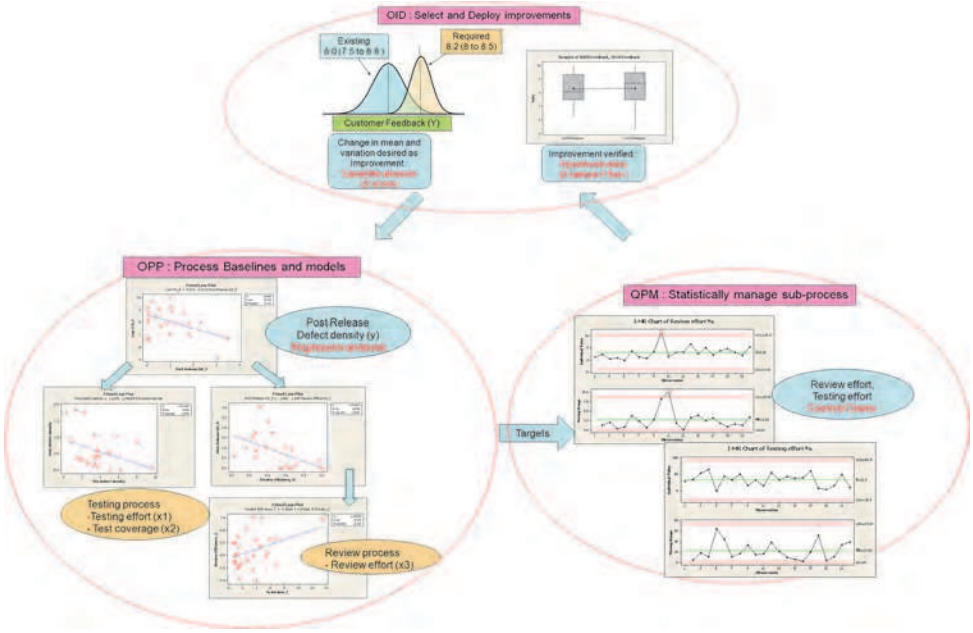


Fig. 32. Linkage of Statistical concepts to CMMI process areas

Further regression analysis shows two parameters that impact post-release defect density:

1. Pre-release defect density influenced by the Testing sub-process
   Regression equation : *Post Defect density = 0.93 – 0.093 * Pre-release Defect density*. To contain the post release defect density within 0.75 defects/KLOC, the pre-release defect density has to be more than 2 defects/KLOC. This means the testing process needs to be improved to catch atleast 2 defects/KLOC. Testing effort and Test coverage are the further lower level Xs that could be improved/controlled to achieve this.
2. *Review efficiency* influenced by the *Review* sub-process

Regression equation : *Post Defect density = 1.66 – 1.658 * Review efficiency*. To contain the post release defect density within 0.75 defects/KLOC, review efficiency has to be more than 55%. This means that review process needs to be improved to catch atleast 55% of defects. Review efficiency is lagging indicator as the value would be known only at the end and is not a directly controllable X. This needs to be further broken down to lower level X that can be tweaked to achieve the desired review efficiency. *Review effort* is one such X. Regression equation : *Review efficiency = 0.34 + 0.038 * Review effort.* To achieve a Review efficiency of 55% and more, a review effort in excess of 5.2% needs to be spent.

The above modeling exercise is part of OPP. Setting objectives at project level and selecting the sub-process to control is then an activity under QPM process area. Based on the business goal (Y) and overall objective (y), the project manager can select the appropriate sub-process to manage and control by assigning targets to them coming from the regression model. As shown in figure-32, the SPC chart for Review effort and Testing effort are used to control those processes. Once the improvement is achieved on the Y and y, hypothesis tests such as 2-sample T tests can be used to confirm a statistical significant change in the CTQ (Y).

6. Conclusion – software specific learning points

Using statistical concepts in software makes it challenging because of 2 primary reasons:-

Most of the Y's and X's in software are discrete in nature as they belong to Yes/No, Pass/Fail, Count category. And many of the statistical concepts are not amenable for discrete data

- The sample size in software is often 1 – the same piece of code evolves throughout

Few points to be kept in mind when approaching with statistics for software :-.

- Challenge each CTQ to see if it can be associated with some numbers rather than simply stating it in a digital manner. Even conceptual elements like Usability, Reliability, Customer satisfaction etc can be quantified. Every attempt should be to made to see if this can be made continuous data as much as possible
- For software CTQs, the specification limits in many of the cases may not be hard targets. For e.g. just because the start-up takes 1 second more than the USL does not render the product defective. So computing Z-scores/Cp numbers may pose a real struggle in such circumstances. The approach should be to see a change in the Z-scores/Cp vales instead of the absolute numbers itself
- Many of the Design of experiments in software would happen with discrete Xs due to nature of software. So often the purpose of doing these is not with the intent of generating a transfer function but more with a need to understand which "Xs" impact the Y the most – the cause and effect. So the Main effects plot and Interaction plots have high utility in such scenarios

- Statistical Capability analysis to understand the variation on many of the CTQs in simulated environments as well as actual hardware can be a good starting point to design in robustness in the software system.
- All Statistical concepts can be applied for the software "Continuous CTQs"

7. References

Ken Black.(2004). Business Statistics for Contemporary Decision Making, Fourth Edition

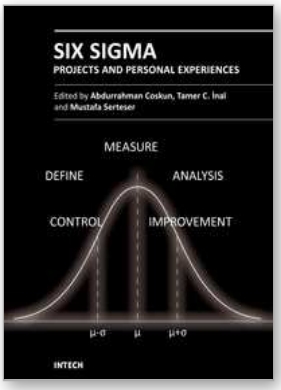
Quentin Brook. Six Sigma and Minitab, A toolbox Guide for Managers, Black Belts and Green Belts, QSB consulting, www.QSBC.co.uk

Jeannine M. Siviy (SEI), Dave Halowell (Six Sigma advantage). 2005. Bridging the gap between CMMi & Six Sigma Training. Carnegie Mellon Sw Engineering Institute

Minitab tool v15– Statistical tool. http://www.minitab.com

Philips DFSS training material for Philips. 2005. SigMax Solutions LLC, USA

Ajit Ashok Shenvi, (August 2010). Design for Six Sigma in software, In: Quality Management and Six Sigma, Abdurrahman Coskun (Ed), ISBN 978-953-307-130-5, Sciyo, Available from http://www.intechweb.org/books/show/title/quality-management-and-six-sigma



Six Sigma Projects and Personal Experiences

Edited by Prof. Abdurrahman Coskun

ISBN 978-953-307-370-5
Hard cover, 184 pages
Publisher InTech
Published online 14, July, 2011
Published in print edition July, 2011

In the new millennium the increasing expectation of customers and products complexity has forced companies to find new solutions and better alternatives to improve the quality of their products. Lean and Six Sigma methodology provides the best solutions to many problems and can be used as an accelerator in industry, business and even health care sectors. Due to its flexible nature, the Lean and Six Sigma methodology was rapidly adopted by many top and even small companies. This book provides the necessary guidance for selecting, performing and evaluating various procedures of Lean and Six Sigma. In the book you will find personal experiences in the field of Lean and Six Sigma projects in business, industry and health sectors.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Ajit Ashok Shenvi (2011). Demystifying Six Sigma Metrics in Software, Six Sigma Projects and Personal Experiences, Prof. Abdurrahman Coskun (Ed.), ISBN: 978-953-307-370-5, InTech, Available from: http://www.intechopen.com/books/six-sigma-projects-and-personal-experiences/demystifying-six-sigma-metrics-in-software

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821