A STOCHASTIC APPROACH TO SECURITY SOFTWARE QUALITY MANAGEMENT

Vojo Bubevski
Legal & General Account
TATA Consultancy Services Ltd
London, United Kingdom
vojo.bubevski@landg.com

ABSTRACT
The conventional approach to security software quality management specifically for ongoing projects has two major limits: (1) Six Sigma is not applied; and (2) analytic risk models are used. This paper proposes a stochastic method, which applies Six Sigma Define, Measure, Analyze, Improve and Control (DMAIC), Monte Carlo Simulation and Orthogonal Security Defect Classification (OSDC). DMAIC is tactically applied to assess and improve quality. Simulation predicts quality (reliability) and identifies and quantifies the quality risk. OSDC allows qualitative analysis. DMAIC is a verified structured methodology for systematic process and quality improvements. Simulation is superior to analytic risk models. OSDC offers qualitative improvements. This synergetic method eliminates observed deficiencies gaining important benefits including savings, quality and customer satisfaction. It is CMMI® (Capability Maturity Model Integration) compliant. The method is simplistically elaborated on a published third-party project.

Keywords: Six Sigma; DMAIC; Simulation; Security Software; Quality Management;

1. INTRODUCTION

Software quality is a multidimensional attribute including reliability, functionality, usability, performance, etc. It is a direct consequence of software processes. Software processes are inherently variable and uncertain, thus involving substantial risks. A key factor in Software Quality is *Software Reliability* as it is one of the the quality attributes most exposed to customer observation. In this paper, "reliability" and "quality" are used interchangeably.

Software quality and customer satisfaction are very important. Managing the software quality, particularly for security software, is a critical factor for software projects.

Six Sigma is used across industries for improving processes, quality and customer satisfaction. One of the principal Six Sigma methodologies is *DMAIC (Define, Measure, Analyze, Improve, Control)*. In Software Engineering, Six Sigma is compatible with *Capability Maturity Model Integration (CMMI®)*. Applications of Six Sigma methodologies in Software Development are discussed in published works (Tayntor 2002; Mandl 1985; Tatsumi 1987; Brownlie, Prowse and Phadke 1992; Bernstein and Yuhas 1993; Siviy, Penn and Stoddard 2007; Nanda and Robinson 2011).

Monte Carlo simulation is a methodology which iteratively evaluates a deterministic model by applying a distribution of random numbers as inputs, which allows to use probability and statistical tools to analyze the results. It is used for modeling phenomena with significant uncertainty, such as software development processes (Bratley, Fox, and Schrage 1983; Rubinstein and Kroese 2008). The term "simulation" is generically used in this paper to refer to "Monte Carlo simulation".

Software Reliability is a main subject in *Software Reliability Engineering (SRE)* (Lyu 1996). The software reliability analytic models have been available since the early 1970s (Lyu 1996; Kan 2002; Xie 1991). The need for a simulation approach to software reliability was recognized in 1993 by *Von Mayrhauser et al.* (Von Mayrhauser et al., 1993). Subsequently, substantial work on simulation was published (Gokhale, Lyu and Trivedi 1997; Gokhale, Lyu and Trivedi 1998; Tausworthe and Lyu 1996; Bubevski 2009; Bubevski 2010).

Six Sigma Software practitioners usually employ conventional analytic models. It has been reported that for Six Sigma in general, simulation models are superior to conventional analytic models (Ferrin, Miller and Muthler 2002).

The Orthogonal Security Defect Classification (OSDC) was established and used by *Hunny* to improve the quality of security software (Hunny 2012). OSDC is based on the Orthogonal Defect Classification (ODC), which was elaborated by *Chillarege* and implemented by IBM™ (Lyu 1996, Chapter 9). OSDC provides for applying qualitative analysis of security software.

1.1. Problem Statement and Proposal

The conventional approach to manage the quality of security software, specifically for an ongoing software projct, has two major limitations: (1) it doesn't apply Six Sigma methods on a current project, but uses the previous release's data to improve the quality of the next release; and (2) it uses analytic risk models.

The paper proposes a stochastic approach to Security Software Quality Management. This approach is based on the new method published by *Bubevski* (Bubevski, 2013). However, the approach presented herein applies the DMAIC

and Simulation methodologies specifically to Security Software by using OSDC.

The synergy of DMAIC, Simulation and OSDC eliminates the limitations identified above. By using this method, substantial savings and quality improvements can be achieved, increasing customer satisfaction.

1.2. Related Work

Hunny used OSDC in order to improve the quality of security software. He presented how the failure data collected for past releases of two software systems, which are OSDC-classified, can be used to improve the quality of the next/future release of the systems by applying analytic models (Hunny 2012).

Bubevski elaborated stochastic approaches to software quality management, which applied Six Sigma and Simulation. The methods were demonstrated and verified on real software projects using published data (Bubevski 2009; Bubevski 2010).

Bubevski also devised and elaborated a new approach to Software Quality Management of ongoing software projects by applying the Six Sigma DMAIC, Simulation and ODC methodologies. The nw method was proven in practice achieving savings, quality imrovments and high customer satisfaction (Bubevski, 2013).

Chillarege applied ODC and the Inflection S-shaped Software Reliability Growth Model for relative risk assessment of the final testing stage. The Inflection S-shaped Software Reliability Growth Model is analytic; it is used to predict the future course of the software reliability growth curve. This helped the project to reduce risk, meet the schedule and assure good field reliability gaining significant benefits (Lyu 1996, Sec. 9.5, Sec. 3.3.6).

Tausworthe & Lyu applied simulation for software reliability assessment on the Galileo spacecraft software project at the Jet Propulsion Laboratory™. The reliability simulation results were substantially better than the reliability predictions obtained by the analytic models such as Jelinski-Moranda, Musa-Okumoto and Littlewood-Verrall models (Tausworthe and Lyu 1996).

Gokhale, Lyu & Trivedi developed simulation models for failure behaviour of the most commonly used fault tolerance architectures. They demonstrated the ability to simulate very complex failure scenarios with various non-trivial dependences (Gokhale, Lyu and Trivedi 1997).

Gokhale, Lyu & Trivedi simulated the reliability of component based software. Discrete event simulation was applied to analyze complex systems, i.e. a terminating application, and a real time application with feedback control. The simulation models applied were superior to the conventional analytic models including Prevalent Markovian and Semi Markovian methods (Gokhale, Lyu and Trivedi 1998).

Simulation was applied by *Gokhale & Lyu* for structure-based analysis of software reliability. Simulation provided for tailoring the testing and repair strategies, and achieving the desired reliability cost-effectively (Gokhale and Lyu 2005).

Siviy, Penn and *Stoddard* used Six Sigma to reduce defects and improve quality. Conventional Six Sigma tools were used such as Rayleigh Fitted Histogram Defect Model and Cause-and-Effect Model, including Computer Aided Software Reliablity Estimation (Siviy, Penn and Stoddard 2007, Sec. 9.1).

Murugappan & Keeni combined Six Sigma with CMMI® to create a quality management system. The aim was to improve software processes and achieve CMMI® Level 4 compliance, which provides for quantitative and qualitative software quality management (Murugappan and Keeni 2003).

An application of CMMI® and Six Sigma in software processes improvement was elaborated by *Xiaosong et al.* The software process management was considered and Six Sigma and CMMI® integration was implemented achieving quality improvements (Xiaosong, Zhen, ZhangMin and Dainuan 2008).

Nanda and *Robinson* published a book including two case studies, which use DMAIC and conventional Six Sigma statistical tools for software defect reduction purposes. The book demonstrates how Six Sigma is applicable to the IT industry, with compelling success stories from today's leading IT companies (Nanda and Robinson 2011, Chapter 5).

Galinac & Car elaborated an application of Six Sigma in the continuous improvement of software verification process. Appling Six Sigma, change management, and statistical tools and techniques, solved the problem of fault slippage through the verification phases (Galinac and Car 2007).

Macke & Galinac presented experiences and results of applying Six Sigma for process improvements in a global software development organization including process definition, awareness for different levels of expectations in globally distributed teams, and introduction of regular scanning mechanisms. Success indicators were defined connecting process capability to business value in order to measure the improvement success (Macke and Galinac 2008).

A six sigma DMAIC approach to software quality improvement was presented by *Redzic & Baik*. Tactical changes were identified and established, which substantially increased the software quality of all software products (Redzic and Baik 2006).

Xiaosong et al used Six Sigma DMAIC and accomplished continuous quality improvement throughout the software development process for high-quality software product. . The software process deficiencies were identified

and eliminated to ensure the desired software quality (Xiaosong, Zhen, Fangfang and Shenqing 2008).

2. THE METHOD DEMONSTRATION (HYPOTHETICAL SCENARIO)

The elaboration is based on a real software project using published data (Lyu 1996, Dataset ODC4). The project is finished so this case is hypothetical. The failure data are available for the entire life cycle but only the data from the last 15 months are used (Table 1). To emulate the scenario of an ongoing project, the data from the first 12 months are used. The last three months' data are used to verify the results. We also pretend that the failure data are for a security software project. Thus, the original data use ODC, but they are mapped to OSDC (Hunny 2012, Table 3.1) because OSDC is specifically applicable to sequrity software. The OSDC Defect Types considered are: Security Functionality (SF), Security Logic (SL) and Miscellaneous (Misc.).

Table 1: Actual Failure Count Data

Month	SF	SL	Misc.
1	16	20	32
2	11	8	6
3	203	36	22
4	37	20	7
5	107	43	13
6	240	43	21
7	27	64	18
8	30	112	23
9	147	98	23
10	24	93	23
11	24	106	28
12	24	33	23
13	6	14	8
14	7	7	3
15	4	15	1

2.1. Assumptions

The method involves quantitative analysis of the metrics data, so the results are data driven. Consequently, the metrics data must be verified and reliable. Also, the organisation and the software project must have capabilities and experiences with quantitative analysis in order to use the method and provide for good and consistent results. Therefore, the fundamental assumption for the method feasibility is that the software organization and the software project are compliant with CMMI® Level 4.

CMMI® Level 4 requires quantitative management of software processes and products within an organization. Thus, the criteria are as follows: (i) detailed measures of the software process and product quality are collected; and (ii) both the software process and products are quantitatively understood and controlled.

Also, there was no information about testing profile, defect relationship, fix (removal) rate or the rate of introducing new defects in the published data. Therefore, for the purpose of the demonstration only and to simplify the simulation models, at least until the experimental results are adequately verified, it is assumed that (i) testing operation profile is uniform, (ii) failures occur independently, (iii) defects are removed in the same time interval as they are encountered and (iv) no new defects are introduced with the fix. It should be considered that the demonstration simulation model's results were satisfactorily verified, so the assumptions were proven to be acceptable.

2.2. Software Quality Risk Management Using Six Sigma and Simulation

The method follows the DMAIC methodology as a tactical framework.

2.2.1. The Project Definition (Define)

We assume that the project is within the final testing stage at the end of Month 12 (TI(12)), which is three months from the targeted delivery date of the product.

Project Objective: Complete final test phase by the end of Month 15 (TI(15)) as planned and deliver the system on time, whilst achieving the quality goal. The delivery date is at the beginning of Month 16.

Project Quality Goal: The aim is to ensure that the system is stable and ready for delivery. All detected defects should be fixed and re-tested before the end of testing. Also, the final month of testing (Month 15) should have one defect per Defect-Type and three defects in total. Maximum two defects per Defect-Type and six defects in total are allowed.

Problem Statement: Assess and mitigate the risk to deliver the system on time, whilst achieving the quality goal. Critical to Quality (CTQ) for the project is the sequrity software reliability.

2.2.2. The Project Metrics (Measure)

In order to define the Failure Intensity Function (FIF) deterministic models by Defect-Type, which are required for simulation, we need to (1) transform the actual data by applying Rank Transformation (Conover and Iman 1981) to get the transformed FIF by Defect Type; and (2) approximate the transformed FIF by Defect Type.

Logarithmic and exponential approximations were tried. The R-square values by Defect Type were (1) Logarithmic: i) SF: $R^2 = 0.9254$; ii) SL: $R^2 = 0.8981$; iii) Misc.: $R^2 = 0.7385$; and iv) Total: $R^2 = 0.9604$; and (2) Exponential: i) SF: $R^2 = 0.8999$; ii) SL: $R^2 = 0.9276$; iii) Misc.: $R^2 = 0.7642$; and iv) Total: $R^2 = 0.9665$.

Comparing the R-square values, the exponential approximation is more accurate. Thus, the exponential approximation of the FIF is selected, which is used by the Musa's Basic Execution Time software reliability model (Lyu 19969, Sec. 3.3.4). The aproximations, i.e. the deterministic models of the FIF by Defect Type are as follows:

$$FIFf(k) = 262.33 \exp(-0.274\ k) \quad (1)$$
$$FIFl(k) = 179.17 \exp(-0.217\ k) \quad (2)$$
$$FIFm(k) = 41.138 \exp(-0.127\ k) \quad (3)$$
$$FIFt(k) = FIFf(k) + FIFa(k) + FIFm(k) \quad (4)$$

Where, FIFf, FIFl, FIFm and FIFt are the FIF for SF, SL & Misc. Defect-Type and the Total respectively, and k is the time interval (k = 1,2,…, n).

2.2.3. Six Sigma Process Simulation (Analyze)

To analyze the process, we simulate the FIFs for the future three months , i.e. from TI(13) to TI(15) inclusive. The simulation is based on the Musa's Basic Execution Time deterministic model (Lyu 19969, Sec. 3.3.4), which applies exponential FIFs. The Poisson distribution is used for the simulation.

To define the quality targets for Month 15 we use the Six Sigma *Target Value*, *Lower Specified Limit (LSL)* and *Upper Specified Limit (USL)*: a) Target Value is one for all defect types and three defects for the Total; b) USL is two for all defect types and six for the Total; b) LSL should be zero, but it will be set to a very small negative number to prevent an error in the Six Sigma metrics calculations, i.e. LSL is -0.0001 for all defect types including the Total. The Six Sigma process simulation results follows.

Figure 1 shows that the Total's distribution in Month 15 of testing totally deviates from the target specifications (i.e. LSL, Target Value and USL). Also, there is a 0.90 probability that the Total would be in the range 11 – 25; 0.05 probability that the Total would be more than 25; and 0.05 probability that the Total would be less than 11.

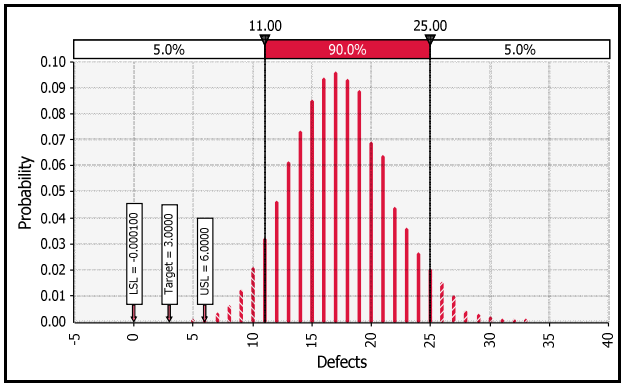

Figure 1: Total Defects Probability Distribution Month 15

Table 2 shows the predicted mean (μ), Standard Deviation (σ) and Minimum and Maximum Values for total number of defects by Defect-Type in the final month of testing TI(15) including the Total.

Table 1: Predicted FIF for Month 15

Process	μ	σ	Min	Max
SF	4	2.06	0	14
SL	7	2.62	0	19
Misc.	6	2.47	0	19
Total	17	4.19	5	36

The predicted Total in TI(15) is 17, with Standard Deviation of 4.19 defects. This indicates that the product will not be stable for delivery at the end of Month 15.

The Six Sigma metrics used to measure the performance are: a) *Process Capability (Cp)* ; b) *Sigma Level*; and c) *Probability of Non-Compliance (PNC)*. The Sigma metrics by Defect-Type for Month 15 is given in Table 3. For example, the SF type has the lowest PNC equal to 0.8057, which is 80.57% deviation from the desired target range. The PNC for the Total is equal to 0.9988, which is 99.88% deviation from the specified target. All three Six Sigma metrics strongly suggest that the process would not perform well, so it would not deliver the desired quality at the end of Month 15.

Table 3: Process Six Sigma Metrics for Month 15

Process	Cp	PNC	Sigma Level
SF	0.1618	0.8057	0.2460
SL	0.1272	0.9687	0.0392
Misc.	0.1349	0.9461	0.0676
Total	0.2389	0.9988	0.0015

2.2.4. Six Sigma Simulation Sensitivity Analysis: CTQs Identification (Analyze)

The simulation sensitivity analysis is used to determine the influence of the change of a particular Defect-Type to the change of Total Defects for all Defect Types.

The correlation sensitivity shows that the most influential defect type, i.e. the top risk CTQ, is SL Defect-Type with correlation coefficient to the Total of 0.63. The Misc. and SF Defect-Type are less influential as their correlation coefficients are 0.58 and 0.49 respectively.

The regression sensitivity shows the quantitative parameters of the influence of the defect types to the total if they change by one Standard Deviation. That is, if the SL defects increase by one Standard Deviation, the Total will increase by 2.62 defects, which is the top risk defect type.

The regression coefficients for Misc. and SF defects are 2.47 and 2.06 defects respectively, so they are less influential. These results are consistent with the correlation sensitivity results.

2.2.5. Six Sigma Analysis Conclusions and Recommendation (Analyze)

The following are the conclusions from this Six Sigma analysis: a) The testing process would not perform well as shown by the considered Six Sigma metrics. Therefore, the system would not be ready for delivery as the quality goals would not be met at the beginning of Month 16 if the project maintains the current situation; and b) The CTQ to deliver the system is the software reliability, i.e. the predicted Total in TI(15) is 17 defects, versus the target value of three defects.

Analysis Recommendation: In order to deliver the system on time and achieve the quality goal, immediately undertake an improvement project to improve the process and enhance the software reliability, which is the CTQ.

2.2.6. Improvement Six Sigma Simulation (Improve)

The purpose of this Six Sigma simulation is to quantitatively determine the solution for improvement, i.e. to predict all the escaped defects (i.e. the defects that are believed to be in the system but they are not captured). Therefore, the software reliability for the future period will be simulated to predict when the reliability goal will be achieved.

It was analyzed and identified that this target could be met in Month 24. Thus, FIF by Defect-Type was simulated for the future period of 12 months, i.e. from Month 13 to Month 24. All the parameters for this simulation were exactly the same as for the previous simulation.

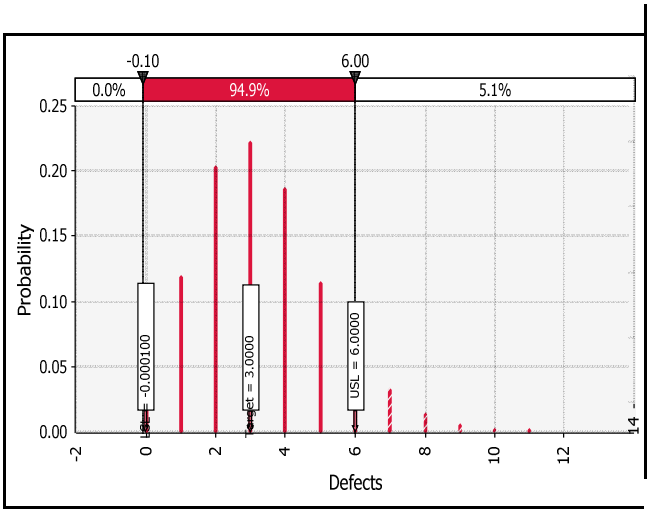

Figure 2: Total Defects Probability Distribution Month 24

As Figure 2 shows, the Total's distribution in Month 24 of testing fits in the process target specifications (LSL, Target Value and USL are marked on the graph). Also, there is a 0.949 (94.9%) probability that the Total in TI(24) would be in the specified target range 0-6 defects; and 0.051 (5.1%) probability that the Total would be more than six. The probability that there would be three defects in total is approximately 0.22 (22%).

According to this prediction, the process could achieve the reliability goal in Month 24 if the project maintains the current situation.

Table 4: Predicted FIF for Month 24

Process	μ	σ	Min	Max
SF	0	0.61	0	4
SL	1	0.98	0	7
Misc.	2	1.39	0	8
Total	3	1.81	0	12

Table 4 shows that predicted number of defects for all types, including the Total, is within the specified target range. The Standard Deviation for all types including the Total, however, is relatively high.

The process Six Sigma metrics at the end of Month 24 are given in Table 5. For example, PNC for Misc. defects is 0.3112 (i.e. 31.12% deviation). The Total however shows only 5.11% deviation.

Table 5. Process Six Sigma Metrics for Month 24

Process	Cp	PNC	Sigma Level
SF	0.5468	0.0074	2.6783
SL	0.3397	0.0739	1.7872
Misc.	0.2391	0.3112	1.0127
Total	0.5510	0.0511	1.9506

All three Six Sigma metrics suggest that there are realistic chances that the process could perform and deliver the desired quality at the end of Month 24.

2.2.7. Improvement Recommendations (Improve)

The following defines and quantifies the solution for the improvement. The predicted total numbers of defects by Defect-Type including the Total for the future periods are shown in Table 6.

The predicted defects for Month 13 – 15 are expected to be detected and removed by the current project until the end of Month 15. The predicted defects expected to be found in the system from Month 16 to Month 24 are unaccounted for. These defects need to be detected and removed until the end of Month 15 in order to achieve the quality goal.

Table 6: Predicted Defects per Defect-Type for Future Periods

Time Period	SF	SL	Misc	Total
TI(13) – TI(15)	17	27	21	65
TI(16) – TI(24)	11	25	31	67
TI(13) – TI(24)	28	52	52	132

Therefore, the process improvement recommendation is: Immediately undertake an improvement project to deliver the system quality improvements as required to achieve the quality goals. The objectives of this project are:

1. Reanalyze the unstable defects applying *Casual Analysis and Resolution (CAR)*;
2. Determine the quality improvement action plan, establishing an additional tactical test plan;
3. Execute the tactical test plan to additionally test the system and detect and repair the escaped defects, i.e. the defects that is believed are in the system but have not been detected. According to the simulation above, there are 67 predicted escaped defects in total (TI(16) – TI(24), Table VI);
4. The additional testing, detection and correction of the escaped defects should be completed by the end of Month 15 to achieve the quality goal.

2.2.8. The Improvement Project and Employment of Additional Resources (Improve)

To undertake the improvement project, additional resources with special skills are needed. The current project is not behind schedule and is running according to plan. The problem is the quality of the product.

To minimize the Brooks' Law effect in employing the additional resources, a "surgical team" should be assigned to the project (Brooks 1995). The objective of the "surgical team" is to deliver the required quality improvement only. The current team working on the project should continue their work according to plan. The "surgical team" will not share any work with the current team.

2.2.9. Improvement Definition (Improve)

The process improvement is a new testing project, which is totally independent of the current testing in progress. There are only three months available to accomplish the improvement, as the quality goal needs to be met at the end of testing (i.e. at the ond of Month 15).

Keeping one month as the time interval for observation is not good because it provides for only two future check points. Thus, the time interval for observation will be reduced to one week. Thus, the proposed schedule for the testing improvement project during the next 13 weeks is: a) one week to start the project and appoint the staff; b) three weeks to complete the required analysis and test plans; and c) nine weeks of testing where the escaped defects will be detected and fixed.

The predicted distribution of the escaped defects by Defect-Type including the Total, which need to be detected and fixed during the testing period of nine weeks, i.e. TI(1) – TI(9), is: SF: 11 Defects; b) SL: 25 Defects; c) Misc.: 31 Defects; and d) Total: 67 Defects.

2.2.10. Six Sigma Simulation for Monitoring (Control)

It is imperative to establish continuous monitoring in order to discover any variances in the process performance, and determine and implement the appropriate corrective actions to eliminate the deviations. This will ultimately mitigate the risk and allow for the delivery of the product on time and the achievement of the quality goals.

In order to deliver the product on time and meet the quality goals, the control phase should be applied to both the current and the improvement testing process. It is recommended to create two additional Six Sigma simulation models and to apply them regularly on a weekly basis to both processes until the end of the projects.

A Six Sigma simulation model for monitoring of the improvement testing process will be demonstrated now. It is assumed that the improvement testing project is at the end of Week 3. An actual defect distribution by Defect-Type for the first three weeks of testing is also assumed, which is given in Table 7.

We need first to transform the assumed actual failures over the three weeks period, to determine the FIF. Also, we need to approximate the FIF by Defect-Type as required for the simulation. The logarithmic and exponential approximations were tried.

Table 7: Assumed Actual Failure Count Data

TI (Week)	SF	SL	Misc.	Total
1	2	3	5	10
2	3	3	4	10
3	2	4	4	10
Total:	7	10	13	30

The R-square values are:

(1) Logarithmic: i) SF: $R^2 = 0.8668$; ii) SL: $R^2 = 0.8668$; iii) Misc.: $R^2 = 0.8668$; and iv) Total: $R^2 = 0.8668$.

(2) Exponential: i) SF: $R^2 = 0.75$; ii) SL: $R^2 = 0.75$; iii) Misc.: $R^2 = 0.75$; and iv) Total: $R^2 = 0.75$.

Comparing the R-square values, the logarithmic approximation is more precise. Thus, we will select the Logarithmic Poisson Reliability Model for the simulation. The aproximation of the FIF by Defect Type is as follows:

$$FIFf(k) = -0.968 \ln(k) + 2.9112 \quad (5)$$

$$FIFl(k) = -0.968 \ln(k) + 3.9112 \quad (6)$$

$$FIFm(k) = -0.968 \ln(k) + 4.9112 \quad (7)$$

$$FIFt(k) = FIFf(k) + FIFa(k) + FIFm(k) \quad (8)$$

Where, FIFf, FIFl, FIFm and FIFt are the FIF for SF, SL & Misc. Defect-Type and the Total respectively, and k is the time interval (k = 1,2,…, n).

The software reliability for the future period of six weeks will be predicted (simulated), i.e. from TI(4) to TI(9) inclusive. For this prediction, the discrete event simulation is used applying the Poisson distribution on the formulas above (5 - 8).

The major objective of the improvement project is to capture and fix the escaped defects. The escaped defect distibution by Defect-Type is a) SF: 11 Defects; b) SL: 25 Defects; c) Misc.: 31 Defects; and d) Total: 67 Defects. Thus, the Six Sigma Target Value, LSL and USL are: a) SF: Target Value is 11, LSL is 9 and USL is 13; b) SL: Target Value is 25, LSL is 22 and USL is 28; c) Misc.: Target Value is 31, LSL is 28 and USL is 35; and d) Total: Target Value is 67, LSL is 60 and USL is 74.

The process Six Sigma simulation results (Figure 3) show that the Total's distribution for the final week of testing TI(9) fits well within the process target specifications. For example, there is a 0.742 (74.2%) probability that the Total in Week 9 would be in the specified target range 60-74 defects. This indicates that the improvement project could achieve the quality target.

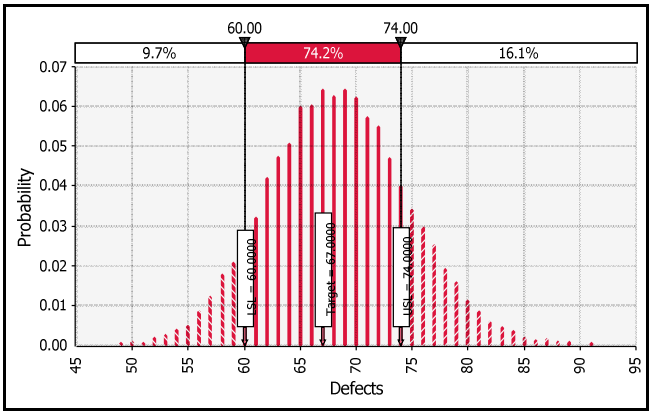

Figure 3: Total Defects Probability Distribution for Week 9

Table 8: Predicted FIF for Week 9

Process	μ	σ	Min	Max
SF	14	2.61	7	26
SL	23	3.56	12	39
Misc.	32	4.32	18	50
Total	69	6.16	48	91

Also, for Week 9 (Table 8) the predicted Total is 69 defects with Standard Deviation of 6.16 defects (8.93%), which is acceptable.

The process Six Sigma metrics (Table IX) shows that for the Total, the PNC metric is 0.2582, i.e. 25.82% deviation from the desired target range, which is acceptable. Therefore, the chances that the improvement testing process could perform as expected are high.

Table IX. Improvement Testing Process Six Sigma Metrics

Process	Cp	PNC	Sigma Level
SF	0.2554	0.5609	0.5815
SL	0.2812	0.5520	0.5948
Misc.	0.2700	0.4230	0.8012
Total	0.3789	0.2582	1.1307

All three Six Sigma metrics strongly suggest that the improvement testing process performed well during the first three weeks of testing. Therefore, there is no need for any corrective action as at the end of Week 3. However, it is required to continue to analyze the process performance by applying the above DMAIC-Simulation analysis regularly, i.e. at the end of every week, until the end of the project.

Similarly, a Six Sigma simulation model can be easily created to monitor the current testing process regularly on a weekly basis until the end of the project. For this purpose, the predicted defect distribution for the period TI(13) – TI(15) should be transformed in a desired weekly defect distribution.

2.2.11. Verification of Results

The experimental results, i.e. the predictions, are compared with the actual available data for verification. It should be underlined that there are no data available from System's Operation. Thus, it is impossible to verify the predictions for improvments and predictions for control.

Two comparisons are performed as presented below: a) Partial Data Comparison; and b) Overall Data Comparison.

Partial Data Comparison:

Table 10: Partial Data Comparison

Process	Defects		
	Actual	Pred.	Error %
SF	17	17	0
SL	36	27	-25
Misc.	12	21	75
Total	65	65	0

The results (Table 10) are verified by comparing the predicted total number of defects by Defect-Type including the Total for the three months period TI(13) – TI(15), versus the corresponding actual defects.

The SF defects and the Total are accurately predicted. The SL defects are underestimatedand Misc. defects are overestimated. These prediction results are acceptable.

Overall Data Comparison:

The overall data comparison is shown in Table XI.

Table 11: Overall Data Comparison

Process	Defects		
	Actual	Pred.	Error %
SF	907	907	0
SL	712	703	-1.2640
Misc.	251	260	3.5857
Total	1870	1870	0

The results are verified by comparing the actual and predicted total number of defects by Defect-Type including the Total for the entire period TI(1) – TI(15), with the corresponding actual defects. Again, the SF defects and the Total are accurately predicted. The SL defects are underestimated with a minimal error. The Misc. defects are slightly overestimated. Thus, these prediction results are very good.

Considering the calculated errors in Table 10 and Table 11, the experimental results are satisfactorily verified.

3. CONCLUSION

The conventional security software quality management of ongoing projects has two major weaknesses: i) analytic risk models are used; and ii) structured methodologies for process and quality improvements are not systematically applied. The proposed novel practical method applies Six Sigma DMAIC, Monte Carlo Simulation and OSDC methodologies. Simulation is superior to analytic risk models and DMAIC is a proven and recognized methodology for systematic process and quality improvements. OSDC provides for qualitative analysis offering qualitative improvements. This synergetic method eliminates the observed limitations of the conventional approach.

The method fully follows the DMAIC framework including the five phases: define, control, analyse, improve and control. It is compatible with CMMI® and can substantially help software projects to deliver the product on time and achieve the quality goals.

The method tactically uses the synergy of the three applied methodologies, i.e. Six Sigma DMAIC, Monte Carlo Simulation and OSDC, which provides for strong performance-driven software process improvements and achieves important benefits including savings, quality and customer satisfaction.

In comparison with the conventional methods, the stochastic approach is more reliable and comprehensive as the inherent variability and uncertainty are accounted for, allowing for probability analysis of the risk. Therefore, the confidence in the method's decision support is substantial, which is of mission-critical importance for software projects.

The simulation models used to demonstrate the method are simple for practical reasons in order to facilitate the elaboration. The models could be easily enhanced to provide for more complex analysis of the ongoing software projects.

ACKNOWLEDGMENTS

I acknowledge *Lyu* (Lyu 1996) for published data used in this work. Also, I would like to thank my daughter, Ivana Bubevska, for reviewing the paper and suggesting relevant improvements.

REFERENCES

Tayntor, C.B., 2002. *Six Sigma Software Development.* Auerbach: Boca Raton, Florida, US.

Mandl, R., 1985. Orthogonal Latin Squares: An Application of Experiment Design to Compiler Testing. *Communications of the ACM,* Vol. 128, No. 10, pp. 1054-1058.

Tatsumi, K., 1987. Test Case Design Support System. *Proceedings of ICQC,* Tokyo.

Brownlie, R., Prowse, J., and Phadke, M.S., 1992. Robust Testing of AT&T PMX/StarMAIL Using OATS. *AT&T Technical Journal,* Vol. 71. No. 3, pp. 41- 47.

Bernstein, L., and Yuhas, C. M., 1993. Testing Network Management Software. *Journal of Network and System Management*, Vol. 1, No. 1.

Siviy, J.M., Penn, L.M., and Stoddard, R.W., 2007. *CMMI® and Six Sigma: Partners in Process Improvement (SEI Series in Software Engineering).* Addison-Wesley Professional: Boston, Massachusetts, US.

Bratley, P., Fox, B.L., and Schrage, L.E., 1983. *A Guide to Simulation.* Springer-Verlag: New York.

Rubinstein, R.Y., and Kroese, D.P., 2008. *Simulation and the Monte Carlo Method.* John Wiley & Sons: New Jersey.

Lyu, M.R., 1996. *Handbook of Software Reliability Engineering*. IEEE Computer Society Press: Los Alamitos, CA, US.

Kan, S.H., 2002. *Metrics and Models in Software Quality Engineering*. Addison-Wesley Professional: Los Alamitos, CA, US.

Von Mayrhauser, A., et al., 1993. *On the need for simulation for better characterization of software reliability*. Proceedings of Fourth International Symposium on Software Reliability Engineering, pp. 264-273. Denver, Colorado, US.

Gokhale, S.S., Lyu, M.R., and Trivedi, K.S., 1997. Reliability Simulation of Fault-Tolerant Software and Systems. *Proceedings of Pacific Rim International Symposium on Fault-Tolerant Systems*, Taipei, Taiwan.

Gokhale, S.S., Lyu, M.R., and Trivedi, K.S., 1998. Reliability Simulation of Component-Based Software Systems. *Proceedings of Ninth International Symposium on Software Reliability Engineering*, Paderborn, Germany.

Tausworthe, R.C., and Lyu, M.R., 1996. Software Reliability Simulation. In: Lyu, M.R., ed. *Handbook of Software Reliability Engineering,* Chapter 16. IEEE Computer Society Press: Los Alamitos, CA, US.

Bubevski, V., 2009. A Simulation Approach to Six Sigma in Software Development. *Proceedings of the 2009 Summer Computer Simulation Conference*. pp. 125-132. Istanbul, Turkey.

Bubevski, V., 2010. An Application of Six Sigma and Simulation in Software Testing Risk Assessment. *Proceedings of the 2010 Third International Conference on Software Testing, Verification and Validation*. pp. 295-302. Paris, France.

Ferrin, D.M., Miller, M.J., and Muthler, D., 2002. Six Sigma and simulation, so what's the correlation?. *Proceedings of the 2002 Winter Simulation Conference*, December 2002, San Diego, California, US.

Lakey, P.B., 2002. Software Reliability Prediction is not a Science… Yet. Cognitive Concepts, St. Louis, US.

Brooks, F.P. Jr., 1995. The Mythical Man-Month (Essays on Software Engineering, Anniversary Edition). Addison-Wesley: Boston, Massachusetts, US.

Nanda, V., and Robinson, J.A., 2011. *Six Sigma Software Quality Imrovment.* McGraw-Hill Professional: New York City , NY, US.

Xie, M., 1991. *Software Reliability Modelling*, World Scientific: Singapore.

Conover, W.J., and Iman, R.L., 1981. Rank Transformations as a Bridge Parametric and Nonparametric Statistics. *The American Statistician*, Vol. 35. No. 3, August 1981, pp. 124.

Gokhale, S.S., and Lyu, M.R, 2005. A simulation approach to structure-based software reliability analysis. *Software Engineering, IEEE Transactions on*, Vol. 31, Issue 8, August 2005, pp. 643-656.

Murugappan, M., and Keeni, G., 2003. Blending CMM and Six Sigma to meet business goals. *Software, IEEE*, Vol. 20, Issue 2, Mar/Apr 2003, pp. 42 – 48.

Xiaosong, Z., Zhen, H., ZhangMin, Y.W., and Dainuan, Y., 2008. Process integration of six sigma and CMMI. *Proceedings of 6th International Conference on Industrial Informatics (INDIN*), pp. 1650-1653. 2008, Daejeon, Korea.

Galinac, T, and Car, Z., 2007. Software verification improvement proposal using Six Sigma. *LNCS*, Vol. 4589, pp. 51-64.

Macke, D., and Galinac, T., 2008. Optimized software process for fault handling in global software development. *LNCS*, Vol. 5007, pp. 395-406.

Redzic, C., and Baik, J., 2006. Six Sigma approach in software quality improvement. *Proceedings of 4th International Conference on Software Engineering Research, Management and Applications (SERA),* pp. 396-406. 2006, Seattle, Washington, US.

Xiaosong, Z., Zhen, H., Fangfang, G., and Shenqing, Z., 2008. Research on the application of six sigma in software process improvement. *Proceedings of 4th International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP),* pp. 937-940. 2008, Harbin, China.

Hunny, U., 2012. *Orthogonal Security Defect Classification for Secure Software Development*. Thesis (PhD) Queen's University, Kingston, Ontario, Canada.

Bubevski, V., 2013. A Novel Approach to Software Quality Risk Management", *Software Testing, Verification & Reliability – STVR.* In Early View.

AUTHORS BIOGRAPHY

Vojo Bubevski comes from Berovo, Macedonia. He graduated from the University of Zagreb, Croatia in 1977, with a degree in Electrical Engineering - Computer Science. He started his professional career in 1978 as an Analyst Programmer in Alkaloid Pharmaceuticals, Skopje, Macedonia. At Alkaloid, he worked on applying Operations Research methods to solve commercial and pharmaceutical technology problems from 1982 to 1986.

In 1987 Vojo immigrated to Australia. He worked for IBM™ Australia from 1988 to 1997. For the first five years he worked in IBM™ Australia Programming Center developing systems software. The rest of his IBM™ career was spent working in IBM™ Core Banking Solution Centre.

In 1997, he immigrated to the United Kingdom where his IT consulting career started. As an IT consultant, Vojo has worked for Lloyds TSB Bank in London, Svenska Handelsbanken in Stockholm, and Legal & General Insurance in London. In June 2008, he joined TATA Consultancy Services Ltd.

Vojo has a very strong background in Mathematics, Operations Research, Modeling and Simulation, Risk & Decision Analysis, Six Sigma and Software Engineering, and a proven track record of delivered solutions applying these methodologies in practice. He is also a specialist in Business Systems Analysis & Design (Banking & Insurance) and has delivered major business solutions across several organizations. He has received several formal awards and published a number of written works, including a couple of textbooks. Vojo has also been featured as a guest speaker at several prominent conferences internationally.