Bayesian Statistics Software Approach For Risk Control on Complex Software Projects*

Nadana Sundaram P V,
Research Scholar,
Madurai kamaraj University,
Madurai, Tamil Nadu.

Dr. K. Iyakutti,
Professor(Rtd),
Maduraikamaraj University,
Madurai, Tamil Nadu.

***Abstract* - Risk control in complex software projects is usually based on constructing a model through synchronization from many developers to automatically manage and prevent conflicts. However, the risk controlled with the aid of Computer Aided Software Engineering (CASE) Tools did not yield qualitative and quantitative based risk control factors during evaluation. Evaluating the risk by ranking the software and monitoring the important activities of software project manager is one solution to solve these limitations. But, these present research methodologies still need to control the risks on software projects with more advanced techniques to verify the effectiveness. The research work is focused on controlling the software risks with advanced features that influence the project success in software companies. Bayesian Statistics Software Process based Risk Control (BSS-RC) method is proposed in this paper to improve the quality of the software risk control. To start with, the software model's project complexity is reduced using Empirical Quantile Software Risk Control task. The Empirical Quantile equally divides the software project module into subsets for easy evaluation of risk control factor. The joint monitoring of software planning and project complexity uses cumulative distribution functions to produce an expected message alert per risk factor. Subsequently, BSS-RC method evaluates the risk after controlling the risk software module using JMP (Jump) Bayesian Statistics Advanced CASE tool to verify the effectiveness. Bayesian advanced CASE tool evaluates the technique qualitatively and quantitatively on multivariate software's. BSS-RC method works on the parametric factors such as mean score for risk control efficiency, risk control time and average run length on joint software monitoring.**

Keywords*: *Computer Aided Software Engineering, Bayesian Statistics Software Process, Risk Control, Cumulative Distribution Function, Empirical Quantile

1. INTRODUCTION

Early detection and providing measures for conflict and risk has come out as one of the key enablers for different types of software projects in software industries. In recent years the synchronization of activities by the programmers have been a means of providing measures for software risks for different users located at geographically distant places by extending the software maintenance risks using different methods. Several researchers have contributed towards software risk managements at early stage.

Motivated by this fact, Speculative Analysis (SA) Technique [1] was designed that efficiently used previously-unused information from different version control methods to accurately measure and diagnose different conflict classes. However, SA technique was not full proof towards qualitative and quantitative analysis. Stepwise Regression Analysis (SRA) Techniques Bayesian Statistics [2] introduced a novel statistical model that served as a risk management model using quantitative results. With this, the method was proven to be efficient to reduce software risk. However, the size of dataset was limited.

One of the key factors observed in organization is to address the risk management as it has significant impact on reducing the probability of software risk and extract the opportunities during the life cycle of the project. Knowledge-Based Risk Management (KBRM) [3] deployed Knowledge Management (KM) processes for enhancing the effectiveness and add up to the success rate. However, knowledge based managing of risk remained unaddressed. In [4] a knowledge based model was designed to minimize the risk involved during the design and implementation of software. But the cost involved in the design was high. A goal driven software development risk management modeling (GSRM) [5] was designed to reduce the production cost during the identification of risk in software. Lean construction principles [6] were introduced to analyze the risk involved in software design. The method was proved to be significant in terms of risk factors involved and was simple to be implemented.

In order to provide a scalable model for complex projects, Project Management Information Systems (PMIS) has gained concentric efforts over the recent years. An integrated approach was introduced in [7] to improve PMIS in software companies. However, planning and scheduling related to software projects remained unaddressed and hence was not a flexible model. To address this issue Project Portfolio Management (PPM) was introduced in [8] to achieve flexibility using several decision making approaches.

In this work, focus is made on controlling the software risks using Bayesian Statistics Software Process based Risk Control (BSS-RC) method. The contributions of BSS-RC method include the following:

1. To control the software risks using advanced features that greatly have an influential factor on the project success in software companies.
2. To improve the effectiveness of quality of the software risk control using Bayesian Statistics Software Process based Risk Control (BSS-RC) method.
3. To minimize the software model's project complexity using Empirical Quantile Software Risk Control task.
4. To evaluate the risk control factor using Empirical Quantile that equally segregates the software project module into subsets for measuring the risk control factor.
5. To produce an expected false message alert per risk factor by joint monitoring of software planning and project complexity using cumulative distribution functions.
6. To verify the effectiveness of BSS-RC method using JMP (Jump) Bayesian Statistics Advanced CASE tools.

The structure of this paper is as follows. In Section 1, the basic problems related to software risk management in software engineering is described. Section 2 demonstrates the related work. In Section 3, an overall framework of Bayesian Statistics Software Process based Risk Control (BSS-RC) method is presented. Section 4 and 5 outline implementation results with parametric factors and present the result graph for with the help of table and graph values. Section 6 concludes the work.

2. RELATED WORKS

Efficient management of risk has been concentrated on diverse area of applications including surveillance, space shuttle, real estate builder financial institutions and so on. In [9], management of risk related to software projects in software industries were introduced with the aid of project risk management team. Despite, project quality was ensured but was not applied to real life data. To address this issue, knowledge management techniques [10] were used for analyzing the risk related to the software projects by efficient data collection methods. However the application of risk according to the life cycle was not performed. A good design of framework based on IEEE 1074 was designed in [11] for efficient functioning and monitoring of risks related to software projects in an intermittent manner.

In order to increase the success rate of software project, risk management is highly significant in software engineering. The design of expert system was constructed in [12] for efficient risk identification with the aid of lessons obtained from several of the unique and similar projects designed by software programmers. With this the bugs that may occur in the past was reduced considerably at a significant rate. Though efficient, it identified the risks based on the previous projects but not error prone to new projects. Awareness of formal risk management techniques that can address risk related to both old and new projects was designed in [13] resulting in handling the risks in a better way. Identification and management of risks related to control of supervisory and acquisition of data was designed in [14].

One of the most complex tasks during the design and analysis of software projects is decision making. In [15], an efficient decision making process was introduced to identify the risk related to the design of software projects with the aid of Bayesian methods. With this the uncertainty related to risk was reduced, but was not suited for dynamic web applications. Localization of faults in web applications was introduced in [16] using conditional and functional call statements. However, qualitative and quantitative factors were not analyzed.

3. BAYESIAN STATISTICS SOFTWARE PROCESS BASED RISK CONTROL ON SOFTWARE PROJECTS

In this section, an efficient method for controlling risks related to software projects using Bayesian statistics software process is explained with the help of a neat architecture diagram. The work starts with the designing empirical quantile by applying data quantile values to identify the risk module, followed by the evaluation of cumulative distribution function for efficient functioning of joint monitoring function to identify the message alert per risk factor using In Bayesian Statistics Software Process based Risk Control (BSS-RC) method.

In Bayesian Statistics Software Process based Risk Control (BSS-RC) method, the software risk factors are controlled and evaluated for effective software project development. To control the software risk, initially the task of Empirical Quantile Software Risk control is performed. The design graph for Empirical Quantile Software Risk is represented through Figure 1.

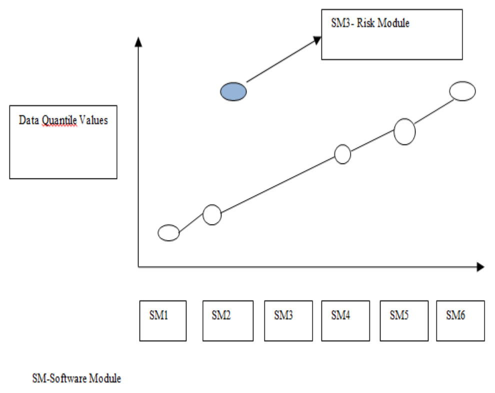

Figure 1 Empirical Quantile Software Risk Graph

Figure 1 illustrates the design of empirical quantile software risk graph. As depicted in the figure, SM denotes software module. For simulation purpose, the work includes six software modules namely, SM1, SM2, SM3, SM4, SM5 and SM6. The software risk factor occurred on SM3 where the risk module is shown in a separate curve. With the application of empirical quantile software, the risky software module is identified and limits the risk control. The job of Empirical Quantile takes the software modules at regular intervals to check the risk factor. The overall software project module is divided orderly into equally sized subsets to control the risk factor at higher rate using BSS-

RC method. With this, the cumulative distribution function is introduced that work with joint software planning and project complexity function monitoring.

Software engineers meet frequently with the developed software modules to arrive at the qualitative and quantitative results. As depicted in Figure 1, the BSS-RC method with risk graph plotting is not ideally suited for planning and measuring the project complexity function. Therefore, cumulative distribution function is used for performing the joint monitoring of software planning and measuring the project complexity. As a result, the monitored module with risk is controlled by producing alarm rate.

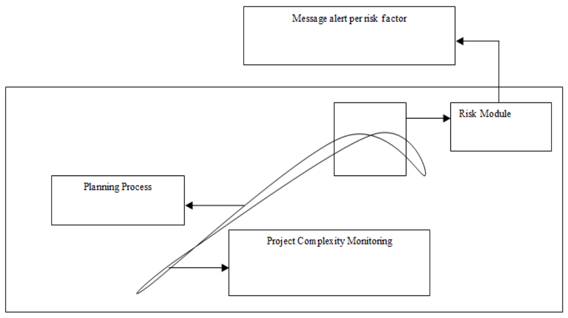

Figure 2 Cumulative Distribution Function on Joint Monitoring for planning and project complexity monitoring

The empirical or cumulative distribution function in BSS-RC method is associated with step wise software module monitoring function which jumps up to 1/n from 'n' data quantile values. The cumulative distribution in BSS-RC method with CASE tool estimates the true underlying software points with higher qualitative result.

The overall structural Diagram of BSS-RC is illustrated in Figure 3.

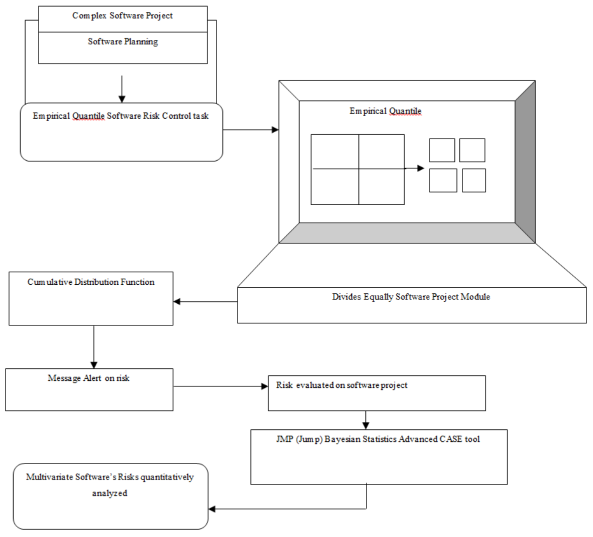
Complex Software Project
Software Planning
Empirical Quantile Software Risk Control task
Empirical Quantile
Cumulative Distribution Function
Divides Equally Software Project Module
Message Alert on risk
Risk evaluated on software project
JMP (Jump) Bayesian Statistics Advanced CASE tool
Multivariate Software's Risks quantitatively analyzed

Figure 3 Architecture Diagram of BSS-RC Method

As illustrated in Figure 3, the main work involved in the design of Bayesian Statistics Software Process based Risk Control (BSS-RC) method is to control the risk factor by evaluating different sets of software project module. BSS-RC initially takes the software planning procedure and then evaluates the complexity range of software projects for monitoring the risk factor. Empirical Quantile Software Risk Control task is employed that divide the software project module equally. The divided parts of the project module are then used to perform cumulative distribution function. The cumulative distribution function works with different complex project modules and produces the alert message whenever the risk is said to occur.

The software risk module is separated and evaluated using JMP (Jump) Bayesian Statistics Advanced CASE tool. The computer aided software engineering tool with JMP modeling software acts as an interface for easy evaluation of risk module. JMP presents the results after linking with each other risk software module and finally evaluate the risk factor. With the application of linking, multivariate's software risk evaluation is improved. The forthcoming subsections discusses in detail about the design of Bayesian statistics software process using empirical quantile software risk control and cumulative distribution function with the help of algorithmic description.

3.1 *Bayesian Statistics Software Process*

Software project processing with Bayesian statistics is used to achieve the true state of degree (i.e., higher quality of software) more specifically. Bayesian statistics with number of Quantile Data interpretation 'QD' begins with informal software module descriptive.

$$\text{Prior Bayesian Distribution } p(\omega|(P,S)) = \frac{Prior(P|\omega)Prior(\omega|S)}{\int Prior\,(P|\omega)Prior(\omega|S)d\omega} \quad (1)$$

The formulation as described above in BSS-RC method provide advanced feature on software risk controlling by using of prior distribution function information 'P '. From (1), two parameters called as the prior distribution function for planning information 'P ' and complex project monitoring information 'S ' is used. 'ω' describes the available knowledge used for controlling the risk based on quantile data information. BSS-RC method with the CASE tool is used for controlling the risk factor.

3.1.1 *Empirical Quantile Software Risk Control task*

The Empirical Quantile risk control task in BSS-RC method with the natural estimator is defined as,

$$DQ(SM) = Empirical(\omega|DQ(\omega) \geq SM) \quad (2)$$

The Data Quantile 'DQ' of the software module 'SM ' measures whether the empirical value of available software module knowledge 'ω' is greater than the software module task or not. Followed by this, the empirical quantile value is measured and if the data quantile value for the available software module knowledge 'ω' is higher than the developed software project module SM , then the risk is said to be occurred on the software project. The higher rate of risk control factor improves the quality of software. The risk control factor in BSS-RC method is expressed through the risk control pointer function ‘I’,

$$DQ(\omega) = \frac{1}{\omega}\sum_{i=1}^{\omega} I \qquad (3)$$

The risk control pointer is used to measure the risk. If the risk control pointer function attains ‘1’then the measured risk is said to be controlled, on the other hand, the risk is not controlled on ‘0’ factor. The mean score for risk control efficiency MS_{rc} using BSS-RC method is the summation of software module task for an overall software project module SM_i at time interval $Time$. It is measured in terms of percentage (%).

$$MS_{rc} = \sum_{i=1}^{n} SM_i * Time \qquad (4)$$

3.2 *Cumulative Distribution Function*

The cumulative or empirical distribution function of complex software project is defined as,

$$cdf(DQ) = \frac{No.of\ SM\ in\ complex\ software\ project}{\omega} = \frac{1}{\omega}\sum_{i=1}^{\omega}\{1\} \qquad (5)$$

In order to measure the risk complexity on complex software projects, (5) is used with {1} factor indicating the occurrence of message alert. The overall information ‘ω ’ is used for monitoring the complex software project using cumulative distribution function ‘cdf. The complex software project success rate in BSS-RC method is achieved by combining (1) and (4) which is measured in terms of percentage (%).Higher is the software project success rate, higher is the ratio of risk control being observed. The risk control time RC_{time} in BSS-RC method is arrived at using the cumulative distributive function of Data Quantile ‘DQ’ $cdf(DQ)$ of software module SM . It is measured in terms of milliseconds.

$$RC_{time} = Time \int_{-\infty}^{x} cdf(DQ) \qquad (6)$$

The steps involved in cumulative distribution function is described as given below,

//Bayesian Statistical Software Cumulative Distribution Function

```
Step 1:  Define cdf (P,S)
  Step 2: Initial Count = 0
  Step 3: For Value in $\omega$ ,
  Step 4: If value $\omega$  <= SM
 Step 5:  Count+= 1
Step 6: Repeat step on every SM on complex
          Software Project
Step 7: Else Identify and Control Risk SM
Step 8: End For
```

The cdf function is developed effectively using BSS-RC method by taking into account not only the information related to planning but also the information related to software project monitoring. BSS-RC method with CASE tool technique expects the place the alert when the message per risk factor occurs. The risk factor on all SM is clearly examined and average run length on joint software monitoring is reduced by half a percent. The average run length or the average software module considered on joint software monitoring AR_{jsm} measures the average of step wise software module $\frac{SM_i}{n}$ from n to $\frac{1}{n}$.

$$AR_{jsm} = \sum_{i=n}^{\frac{1}{n}} \frac{SM_i}{n} \qquad (7)$$

Since the monitoring of the complex software project is combined, the running time is also reduced. The number of qualitative results in BSS-RC method exhibits the advance software risk control using cdf.

3.3 *JMP (Jump) Bayesian Statistics Advanced CASE tool*

Once the risk related to complex software project is measured, JMP Bayesian Statistics is applied in BSS-RC method to improve the quality of the project. JMP act as an interface with advanced Bayesian features to reach higher feasibility ratio. The computer aided software engineering tool with JMP modeling software acts as an interface for easy evaluation of risk module.

The JMP presents the results after linking the risk of the software module with each other to evaluate the risk factor. With this, the linking improves the multivariate’s software risk evaluation. The linking of the entire risks software module also improves the complex software project success rate. JMP used in BSS-RC method with CASE tool performs the advanced Bayesian statistical process.

4. EXPERIMENTAL EVALUATION

An experimental work on proposed Bayesian Statistics Software Process based Risk Control (BSS-RC) method uses JAVA. The JAVA code based implementation takes the Cylinder Bands Data Set from UCI repository to evaluate the proposed work. Cylinder Bands Data Set uses the Computed Aided Software Engineering (CASE) tools with advanced developed JMP software for effective result performance. CASE tools provide the significant measure for module assessment and controlling the risk factor. The statistical software process also provides the qualitative and quantitative result analysis.

BSS-RC method compares the result factor with existing Speculative Analysis (SA) Technique from version control operations which precisely diagnose important classes of conflicts and Stepwise Regression Analysis (SRA) Techniques. Risk control is evaluated on the parametric factors such as mean score for risk control efficiency, risk control time, complex software project success rate, and average run length on joint software monitoring.

To perform BSS-RC method for measuring the reducing the risk in software projects, several related parameters has to be tuned in order to obtain the best result. In this paper, these parameters were tuned based on JAVA setting conducted for the purpose of experimentation that considered software size in the range of 100 to 700 MB and software modules between SM1 and SM7.

5. RESULTS ANALYSIS OF BSS-RC

The Bayesian Statistics Software Process based Risk Control (BSS-RC) method is compared against the existing Speculative Analysis (SA) Technique [1] and Stepwise Regression Analysis (SRA) Techniques Bayesian Statistics [2]. The experimental results are measured using JAVA and are compared and analyzed with the help of the table values and graph representation given below. To support elaborate performance, in Table 1 we apply an efficient Empirical Quantile Software Risk Control to obtain the mean score for risk control efficiency and comparison is made with two other existing methods SA and SRA.

Table 1 Tabulation of mean score for risk control efficiency

Software size (MB)	Mean score for risk control efficiency (%)		
	BSS-RC	SA	SRA
100	58.35	46.31	40.21
200	64.25	52.22	46.12
300	66.55	54.52	48.42
400	62.15	50.11	44.01
500	68.45	56.43	50.33
600	72.35	60.33	54.23
700	78.55	66.54	62.44

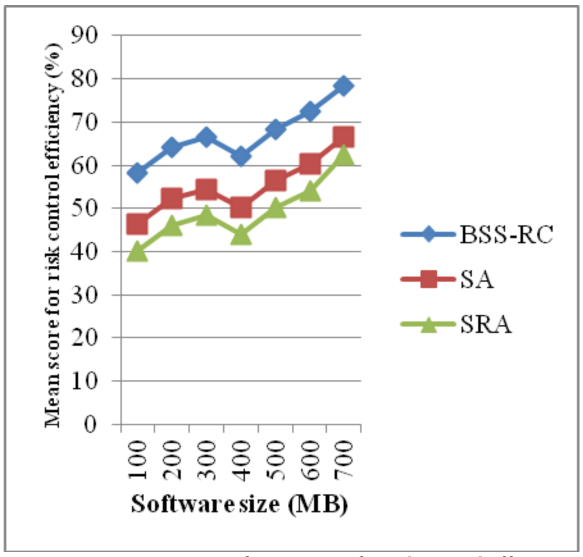

Figure 4Measure of mean score for risk control efficiency

Figure 4 show that the proposed BSS-RC method provides higher score for risk control efficiency when compared to SA [1] and SRA [2]. This is because of the application of Empirical Quantile Software Risk Control task that extracts the software modules at regular time interval resulting in the increased mean score for risk control efficiency by 15 – 20 % when compared to SA [1]. In addition to that by splitting the overall software project module into equally sized subsets that control the risk factor at higher rate and improves the mean score for risk control efficiency by 20 – 31 % compared to SRA [2].

Table 2 Tabulation of risk control time

Software size (MB)	Risk control time (ms)		
	BSS-RC	SA	SRA
100	0.124	0.145	0.177
200	0.158	0.179	0.201
300	0.423	0.444	0.476
400	0.456	0.477	0.507
500	0.488	0.509	0.538
600	0.358	0.379	0.408
700	0.586	0.607	0.639

The comparison of software risk control time is presented in table 2 with respect to different software of sizes in the range of 100 – 700 MB. With increase in the size of the software, the software risk control time is also increased, though not linear because of the fault occurred in the software differs at a varied rate.

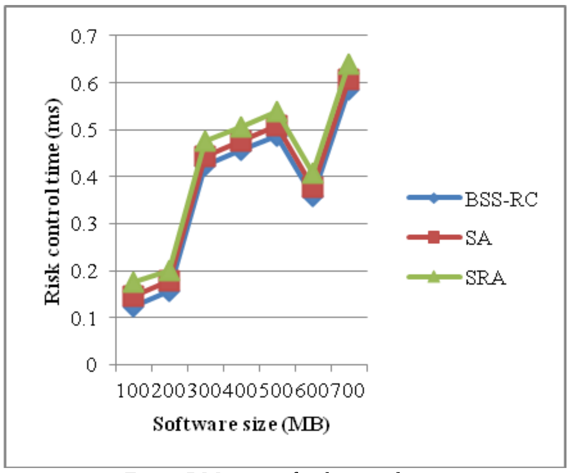

Figure 5 Measure of risk control time

To evaluate the performance of the risk control time, comparison is made with two other existing works Speculative Analysis (SA) Technique [1] and Stepwise Regression Analysis (SRA) Techniques [2]. In figure 5, the software sizes are varied between 100 and 700. From the figure it is illustrative that the risk control time is lesser using the proposed Bayesian Statistics Software Process based Risk Control (BSS-RC) method when compared to the two other existing works. This is because of the joint monitoring of software planning and measuring the project complexity using cumulative distributive function by minimizing the risk control time by 3 – 16 % compared to SA. Furthermore, with the aid of Data Quantile of the software module 'SM', the empirical value of available software module is compared with that of the developed software project module. With this, higher rate of risk control factor improves the quality of software improving the risk control time by 9 – 42 % compared to SRA.

Table 3 Tabulation of complex software project success rate

Software Modules (SM)	Complex software project success rate (%)		
	BSS-RC	**SA**	**SRA**
SM1	59.35	49.13	42.10
SM2	62.45	52.23	47.20
SM3	66.85	56.63	49.60
SM4	78.25	68.03	61.00
SM5	63.45	53.23	49.20
SM6	68.55	58.33	51.30
SM7	74.35	64.13	57.10

The value of complex software project success rate for Software Process based Risk Control (BSS-RC) method is elaborated in table 3. We consider the method with seven different modules of SM1 to SM7 for experimental purpose using JAVA.

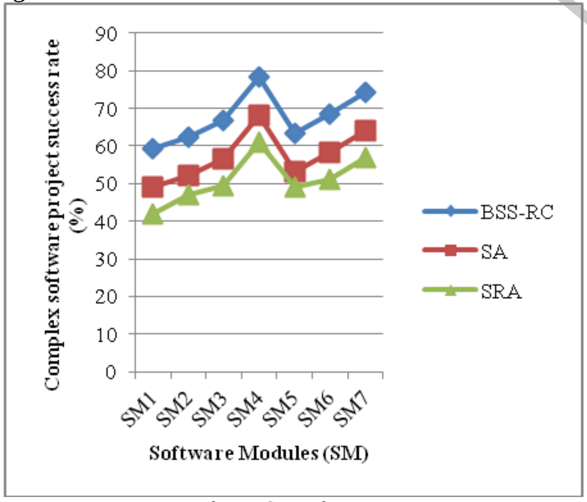

Figure 6 Measure of complex software project success rate

In figure 6, the value for complex software project success rate is depicted using seven different software modules namely SM1 through SM7 for the purpose of experiment. From the figure, the value of complex software project success rate using the proposed Bayesian Statistics Software Process based Risk Control (BSS-RC) method is higher when compared to two other existing works Speculative Analysis (SA) Technique [1] and Stepwise Regression Analysis (SRA) Techniques [2]. Besides we can also observe that by varying the software modules which consist of different sizes, the success rate is increased using all the methods. But comparatively, it is higher in Bayesian Statistics Software Process based Risk Control (BSS-RC) method because with the application of Bayesian statistics, the software project processing achieves higher complex software project success rate using prior distribution function information for planning and project monitoring information for measuring the project complexity. With this, even the complex software project success rate is improved. Furthermore, with the application of JMP Bayesian Statistics, the software quality of the project is improved to reach higher feasibility ratio and therefore improves the complex software project success rate improved by 13 – 17 % when compared to SA and 23 – 29 % than the SRA.

Table 4 Tabulation of average run length on software monitoring

Software Modules (SM)	Average run length on software monitoring (%)		
	BSS-RC	SA	SRA
SM1	50.40	60.60	64.72
SM2	52.07	65.27	68.39
SM3	54.04	67.14	71.26
SM4	55.13	69.33	73.45
SM5	62.13	75.33	78.45
SM6	54.00	68.20	72.32
SM7	53.13	66.33	70.45

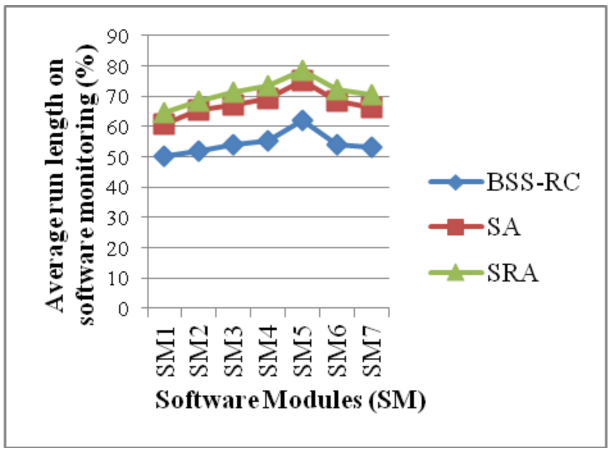

Figure 7 Measure of average run length on software monitoring

Table 4 and Figure 7 illustrate the average run length on software monitoring versus different software modules in terms of MB for experimental purpose conducted using JAVA. From the figure we can note that the average run length on software monitoring value attains 25.35 % improved for a software module of type SM2 when compared to SA [1] and 31.34 % improved when compared to SRA [2] which shows that there is a significant improvement using the proposed Bayesian Statistics Software Process based Risk Control (BSS-RC) method. This is because, the empirical distribution function in BSS-

RC method associated with step wise software module monitoring function jumps efficiently from n to 1/n from data quantile values. With this, the cumulative distribution in BSS-RC method using CASE tool estimates the true underlying software points with higher qualitative result improving the average run length on joint software monitoring by 20 – 26 % compared to SA and 26 – 33 % when compared to SRA.

6. CONCLUSION

A Bayesian Statistics Software Process based Risk Control (BSS-RC) method with software project of different sizes, have been designed to improve the quality of software risk control. We adopt Bayesian statistical software cumulative distribution algorithm, design prior distribution function and complex project monitoring information, cumulative distribution function and proposed a Bayesian Statistics Software Process based Risk Control (BSS-RC) method which attains improved risk control efficiency in software companies. The proposed Empirical Quantile risk control task divides equally software project module that in a way efficiently controls the risk factor in software module. In addition, the cumulative or empirical distribution function of the complex software project takes not only planning information but also the information related to software project monitoring to improve the average run length on software monitoring. Finally, the computer aided software engineering tool integrated with JMP modeling software provides a means for easy evaluation of the risk module. Experimental evaluation is conducted with the Cylinder Bands Data Set extracted from UCI repository to analyze the software risk on projects and measured the performance in terms of mean score for risk control efficiency, software risk control time, average run length and software project success rate. Performances results reveal that the proposed Bayesian Statistics Software Process based Risk Control (BSS-RC) method provides higher level of software project success rate and risk control efficiency at relatively lesser amount of risk control time in software companies. Compared to the existing software risk control methods, the proposed Bayesian Statistics Software Process based Risk Control method achieves 29 % improved software project success rate in identifying the software risks with an average run length reduced to 33 percent and risk control time by 42 percent compared to state-of-art works.

REFERENCES

1. Yuriy Brun., Reid Holmes, Michael D. Ernst, and David Notkin., Early Detection of Collaboration Conflicts and Risks," IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, 2013
2. Abdelrafe Elzamly., Burairah Hussin., "Mitigating Software Maintenance Project Risks with Stepwise Regression Analysis Techniques," Journal of Modern Mathematics Frontier Volume 3 Issue 2, doi: 10.14355/jmmf.2014.0302.02, June 2014
3. Samer Alhawaria, Louay Karadshehb, Amine Nehari Taletc, Ebrahim Mansoura," Knowledge-Based Risk Management framework for Information Technology Project", International Journal of Information Management, Elsevier, August 2012
4. Edzreena Edza Odzaly, Des Greer, Paul Sage," Software Risk Management Barriers: an Empirical Study", IEEE, Third International Symposiumm on Empirical Software Engineering and Measurement, Aug 2009
5. Shareeful Islam, Md. Mahbubul Alam Joarder and Siv Hilde Houmb," Goal and Risk Factors in Offshore Outsourced Software Development From Vendor's Viewpoint", IEEE, Fourth IEEE International Conference on Global Software Engineering, Aug 2009
6. Usama Hamed Issa," Implementation of lean construction techniques for minimizing the risks effect on project construction time", Alexandria Engineering Journal, Elsevier, Aug 2013
7. M. Braglia, M. Frosolini," An integrated approach to implement Project Management Information Systems within the Extended Enterprise", International Journal of Project Management, Elsevier, Dec 2012
8. Ernesto Gutiérrez, Mats Magnusson," Dealing with legitimacy: A key challenge for Project Portfolio Management decision makers", International Journal of Project Management, Jan 2013
9. Lazaros Sarigiannidis, Prodromos D. Chatzoglou," Software Development Project Risk Management: A New Conceptual Framework", Journal of Software Engineering and Applications, Aug 2011
10. Sandra Miranda Neves, Carlos Eduardo Sanches da Silva, Valério Antonio Pamplona Salomon, Aneirson Francisco da Silva, Bárbara Elizabeth Pereira Sotomonte," Risk management in software projects through Knowledge Management techniques: Cases in Brazilian Incubated Technology-Based Firms", International Journal of Project Management, May 2014
11. Cristina Lópeza, Jose L. Salmerona," Monitoring software maintenance project risks", CENTERIS 2012 - Conference on ENTERprise Information Systems, Elsevier, June 2012
12. Julia Botan Machado, Silvio do Lago Pereira," Automatic Risk Identification in Software Projects: An Approach Based on Inductive Learning", Intelligent Information Management, Oct 2012
13. Noela Jemutai Kipyegen, Waweru Mwangi and Stephen Kimani," Risk Management Adoption Framework for Software Projects: A Case Study for Kenyan Software Project Managers and Developers", IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 3, No 3, May 2012
14. Abdelghafar M. Elhady, Ahmed Abou Elfetouh S. &Hazem M. El-bakry, A. E. Hassan ,"
Generic Software Risk Management Framework for SCADA System", International Journal of Computer Applications (0975 – 8887) Volume 70– No.3, May 2013
15. Ayse Tosun Misirli, Member, IEEE, and Ayse Basar Bener,," Bayesian Networks For Evidence-Based Decision-Making in Software Engineering", IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 40, NO. 6, JUNE 2014
16. Shay Artzi, Julian Dolby, Frank Tip and Marco Pistoia," Fault Localization for Dynamic Web Applications",IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 38, NO. 2, MARCH/APRIL 2012