Fuzzy Systems and Soft Computing
ISSN : 1819-4362

IMPLEMENTATION OF REQUIREMENT PRIORITIZATION THROUGH CENTRAL LIMIT THEOREM FOR HASSLE-FREE IN SOFTWARE PROJECT

Dr.M.KARTHIKA, Assistant Professor and Head, Department of Information Technology, Mannar Thirumalai Naicker College, Madurai, Tamilnadu
Dr.T.SUJITHRA, Assistant Professor, Department of Information Technology, Mannar Thirumalai Naicker College, Madurai, Tamilnadu

Abstract
Software Engineering has improved to a remarkable level these days. Nowadays need of software engineering arises because of higher rate of change in user requirements and environment on which the software is working. Software Industry focused on customer need and quality. Software industry facing, involved in developing more complex projects. To maintain the quality of a project the initial stage is validating the requirements. More than the development procedures for building a software system there are more expectations from customer. Customer requirements processing are playing the vital role on requirement prioritization. The goal of Requirement prioritization is to evolve effective, simple and scalable of the software products in which it undergoes continuous changes and releases. The requirement prioritization can be based on the customer need and the current trend and also rank through probability or priority values assigned for a particular environment. To enhance the requirements priority the efficient methods to enhance the priority scheduling is the implementation of central limit theorem. So by doing such prioritization we can assimilate the needs based on urgency, relevance and business needs. Central limit theorem applies to the benefit of prioritization and the prime aim of this paper is to categorize the number of people and number of days of a software process thereby deducing an optimal solution for the process of choosing the number of people and number of days which can be satisfied by the implementation of this project**.**
Keywords: Requirement prioritization, normal distribution, central limit theorem.

Introduction
Software product commences with collection and recognition of specific requirements by the customer and of the project cycles through need protocols and how these requirements have to be met in the product arena. Requirements engineering in software development includes, among other activities, the process of evaluating, prioritizing and selecting the requirements which will be implemented in the next release of a software product. In comparison with other relevant approaches in the literature, the proposed model is more enhanced since it includes structural elements, such as different service rates for servers (i.e., analysts may have different productivity rates), re-entrant lines and garbage collectors for requirements as well as the option that the initial classification of a requirement might be re-evaluated and altered during the process.

- The mean of the sampling distribution of X equals the population mean, regardless of the sample size or the population distribution.
- The standard deviation of the sampling distribution of X equals the population standard deviation divided by the square root of the sample size, regardless of the population distribution.
- The shape of the sampling distribution of X is approximately normal for large sample sizes, regardless of the population distribution, and it is normal for any sample size when the population distribution is normal[5].

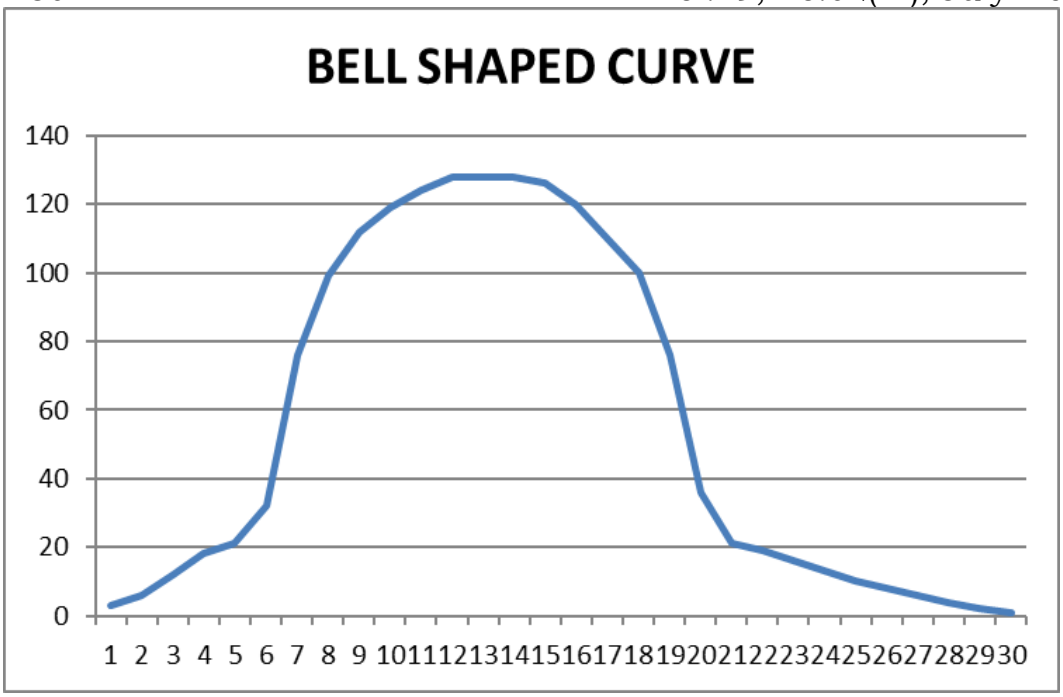
Central Limit Theorem
Central Limit Theorem is a mathematical concept which emphasizes a sufficiently large space from a random pick with a finite level of deviation, the average of all these trials from the random group will be more or less equal to the average of the random group. The Central Limit Theorem may be stated as follows:
Let X1,....x_n, be a sequence of independent random variables each having the same distribution with finite mean μ and finite variance of σ^2. If x_n—**is** the mean of x1,..., x_n, then the distribution of the

standardized variable $Z_n = \frac{X-\mu}{\sigma/\sqrt{n}}$ converges to the Normal (**0,** 1) distribution as n $\rightarrow\infty$ .The point of the theorem is that no matter what the original distribution, the mean of a large enough sample will have a nearly normal distribution. Note that the population from which we are sampling could have a distribution that is uniform, skewed, or non-normal in some other way. It is the sample mean that behaves as a random variable with an approximately normal distribution. This theoretical sampling distribution has a mean equal to μ**,** the mean of the population and a standard deviation equal to / n. This powerful result may be used to explain why, for example, the observations in many engineering, physical and psychological processes follow the characteristic bell-shaped curve of the normal distribution [4].

Input and Output Process

When the number of people is increased we may blatantly observe that the number of days required to complete a task is subsequently decreased and on plotting these values on a Cartesian plane we obtain a bell shaped normalization curve from which the various priorities of task can be streamlined based on the necessity of the task which is indicated by priority value through central limit theorem.

People	No. of days
1	3
2	6
3	12
4	18
5	21
6	32
7	76
8	99
9	112
10	119
11	124
12	128
13	128
14	128
15	126
16	120
17	110
18	100
19	76
20	36
21	21
22	19
23	16
24	13
25	10
26	8
27	6
28	4
29	2
30	12



Conclusion

The implementation of requirement prioritization through central limit theorem could be one of the most feasible and viable technique when a software engineer has gathered all existing solution requirements. This project revolves around prioritization of client needs based on central limit theorem which perhaps also aims at a business value. More the business value of a requirement, greater the priority stakeholders may choose to assign it. Some requirements pose a significant risk of project failure if not implemented successfully [2].The software analyst may assign a high priority to this category of requirements so that if the project does fail the least amount of effort would have been spent. Implementation difficulties are common and much pronounced in requirement prioritization. The requirements fall into two strata namely easier requirements and difficult requirements. Easier requirements are the straightforward needs that may lead to quick wins and provide an opportunity to familiarize themselves with rest of the elements of the project before they are about to handle a complex requirement. There is an equal probability of getting a success or failure in processing a requirement but still the ultimate result of the project lies in successful requirement prioritization. In lew of it as a part of enhancing the success rates the requirements which have higher probability of success will be given higher priority, there by achieving success. There is a regulatory compliance for every project. Adherence to these regulatory protocols becomes necessary for a programmer analyst and so such requirements of this sort needs to be implemented mandatorily [3] within a period of time besides their priority value. The ultimate aim of a project is to enhance to a business and achieve profit. When a programmer comes across a high value business requirement out of the blue he has to assign a high priority for this business value requirement aiming profit. In this requirement session a requirement will not be implemented or a requirement which is on a quest or which is under discrepancy or which is in a stake or in a state of disagreement will be assigned a lowest priority by assimilating all these techniques we can extend this project in future for various software requirements based simulations.

Future work

Requirement prioritization is an effective methodology of queuing process aims in making out big bugs and a refined business atmosphere. In the field of software engineering by assigning priorities to the requirements, requirements are fulfilled. The future of these project lies in the application of different techniques of assigning priority values to the given set of requirements logically, mathematically and commercially on may attain success. These priority assigning techniques by central limit theorem can be extended by the implication of central limit theorem by bernoullis trials from which we can extract the immediate and absolute necessity of a requirement

with accuracy and certainity.subtle implication of various statistical probability distribution in the place of central limit theorem will open doors for a high class requirement prioritization and a high profile business ambience. So, as stated above hereby this project wipes off all the preset notions and opinions of assigning a priority based on superficial factors alone and in real time this project gets into the skin of a requirement processing needs considering important factors and also every other non-trivial factors as well which when prioritized based on the probability of success, urgency, promulgator and client needs will be the best possible solution for a programmer analyst in future making it a cake walk so that all the relevant requirements are processed at first based on client needs by applying various theorems and lemmas other than central limit theorem with a white spreading oceanic purpose business value in future.

References

1. Lehtola, Laura, Marjo Kauppinen, and Sari Kujala. "Requirements prioritization challenges in practice." Product focused software process improvement. Springer Berlin Heidelberg, 2004. 497-508.
2. Berander, Patrik, and Anneliese Andrews. "Requirements prioritization." Engineering and managing software requirements. Springer Berlin Heidelberg, 2005. 69-94.
3. Karlsson, J. & Ryan, K. (1997). A Cost-Value Approach for Prioritizing Requirements, IEEE Software September/October 1997, 67-74.
4. Hogg & Tanis, 1977, p. 155; Mood, Graybitt E Gas**,** 1974, p. 195.
5. CLT Calculations.pdf
6. Bauer (2001, Theorem 30.13, p.199)
7. Billingsley (1995, p.362)
8. Van der Vaart, A. W. (1998). Asymptotic statistics. New York: Cambridge University Press. ISBN 978-0-521-49603-2. LCCN 98015176.
9. Voit, Johannes (2003). "Section 5.4.3". The Statistical Mechanics of Financial Markets. Texts and Monographs in Physics. Springer-Verlag. ISBN 3-540-00978-7
10. Gnedenko, B. V.; Kolmogorov, A. N. (1954). Limit distributions for sums of independent random variables. Cambridge: Addison-Wesley.
11. Uchaikin, Vladimir V.; Zolotarev, V. M. (1999). Chance and stability: stable distributions and their applications. VSP. pp. 61–62. ISBN 90-6764-301-7.
12. Billingsley (1995, Theorem 27.4)
13. Durrett (2004, Sect. 7.7(c), Theorem 7.8)
14. Durrett (2004, Sect. 7.7, Theorem 7.4)

15. Billingsley (1995, Theorem 35.12)
16. "An Introduction to Stochastic Processes in Physics" jhupbooks.press.jhu.edu. Retrieved 2016-08-11.
17. Stein, C. (1972). "A bound for the error in the normal approximation to the distribution of a sum of dependent random variables". Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and Probability: 583–602. MR 402873. Zbl 0278.60026.
18. Chen, L. H. Y.; Goldstein, L.; Shao, Q. M. (2011). Normal approximation by Stein's method. Springer. ISBN 978-3-642-15006-7.