AN IN-DEPTH EXAMINATION OF USE CASE POINT ANALYSIS: ENHANCING SOFTWARE SIZING ACCURACY AND EFFICIENCY

Hulavale Tanuja Sambhaji[1*], Dr Priyanka Jhawar[2]

[1*] Research Scholar Department of Mechanical Engineering (Industrial Design), Sri Satya Sai University of Technology & Medical Sciences, Sehore M.P
[2]Research Guide, Department of Mechanical Engineering, Sri Satya Sai University of Technology & Medical Sciences, Sehore M.P

***Corresponding Author:** Hulavale Tanuja Sambhaji
[*] Research Scholar Department of Mechanical Engineering (Industrial Design), Sri Satya Sai University of Technology & Medical Sciences, Sehore M.P

ABSTRACT

To effectively manage projects, estimate costs, and allocate resources, software scaling must be precise and efficient. The Use Case Point (UCP) Analysis is a well-known method for determining the functional requirements-based software size. The study delves into UCP Analysis, including its methodology, applications, and advantages. A standardized method for estimating complexity and effort, UCP Analysis converts user interactions and system functions into measurable measurements. By analyzing real-world examples and statistical data, we show how UCP Analysis boosts software size estimates, lowers project risks, and improves planning in general. Furthermore, this study delves into the benefits of UCP Analysis compared to conventional size techniques, highlighting its function in encouraging software development results that are both predictable and successful.

Keywords: Software, Estimation, Size, Accuracy, Use case.

I. INTRODUCTION

Modern software development relies heavily on precise effort and project size estimations for planning and management purposes. Use Case Point (UCP) Analysis is one of the most well-known methods for determining software size estimates. In order to shed light on how to improve the precision and efficacy of software sizing, this article dives deeply into UCP Analysis, investigating its concepts, methodology, difficulties, and possible improvements.

The idea behind Use Case Point Analysis is to measure how well a software system works by looking at the use cases that represent user interactions. By shifting its emphasis from code-based metrics to functional requirements, UCP Analysis provides a broader picture of software size and complexity than its predecessors. A formal framework for software size estimation is provided by UCP Analysis, which involves assigning weights to various use case features including actors, transactions, and complexity variables.

As a first stage in UCP Analysis's methodology, defining and classifying use cases according to their functions and actors is essential. The amount of transactions and different complexity characteristics are determined by analyzing each use case that has been discovered. Software size estimation is obtained by giving weights to various elements and then computing the total UCP value. Although UCP Analysis follows a structured methodology, it still struggles to grasp the complexities of contemporary software systems.

Efficiently defining and classifying use cases is one of the main obstacles of UCP Analysis. The complexity of interdependent system features makes use case characterization difficult, which in turn increases the likelihood of ambiguity and inaccurate size assessment. Additionally, subjective judgment is often involved in evaluating complexity elements including technical difficulty and environmental considerations, which introduces uncertainty and unpredictability into the assessment process.

Several approaches can be used to overcome these obstacles and improve the precision and effectiveness of software sizing using UCP Analysis. The first step in reducing ambiguity and increasing coverage of all features is to improve the use case selection process through the application of domain-specific knowledge and stakeholder collaboration. To further simplify and lessen the impact of human bias in the estimating process, automated tools and algorithms can be utilized to evaluate use cases and complexity variables.

In addition, combining UCP Analysis with other sizing approaches, like Story Points or Function Point Analysis, can provide further insights and verify the estimation outcomes. Software sizing accuracy can be enhanced by comparing several metrics, which in turn provides a stronger basis for project planning and decision-making. The methodology's adaptability and dependability can be further improved through the ongoing validation and modification of UCP weights using empirical data and project feedback.

Ultimately, when it comes to software size estimation, Use Case Point Analysis is a great tool to have on hand. It provides a functional-oriented viewpoint that enhances the standard metrics based on code. The intricacies of contemporary software systems are difficult for UCP Analysis to capture, notwithstanding its usefulness. Better project management and resource allocation decisions are possible with UCP Analysis after some strategic upgrades, such as better use case identification, automated analysis tools, and integration with other sizing methodologies.

II. REVIEW OF RELATED STUDIES

Rizvi, Mohammed. (2023). Cybersecurity now relies heavily on artificial intelligence (AI) because of its real-time threat assessment and action capabilities. AI is becoming more effective at detecting and preventing assaults that organizations rely on to stay ahead of the curve. In the realm of cybersecurity, AI mostly serves to detect and prevent threats. By utilizing machine learning techniques and extensive data analysis, artificial intelligence may identify patterns and irregularities in network traffic and user behavior that could suggest a possible cyberattack. In this way, security guards can anticipate and react swiftly to possible threats. One way AI can help stop assaults is by using predictive modeling. By analyzing previous attacks and identifying patterns, AI can also anticipate and prevent possible dangers. Another crucial role of AI in cybersecurity is the development of automated methods for responding to incidents. In order to minimize interruption and damage, these systems can analyze data, detect possible dangers, and then take action to either limit or lessen the impact of an attack. Safeguarding networks and sensitive data from evolving cyber threats requires businesses to utilize artificial intelligence in cybersecurity. In today's digital world, AI is quickly becoming an essential tool for effective cybersecurity due to its real-time data analysis capabilities and capacity to automate incident response. Artificial intelligence (AI) has several applications in cybersecurity, such as threat detection and prevention, which will be covered in this paper.

Sharma, Shikha. (2023). The discipline of precision agriculture is one of the fastest growing in the world. It uses data analytics and cutting-edge technology to maximize agricultural yields while minimizing waste. Recent developments, tools, and uses in precision agriculture are summarized below. Data collecting and analysis for informed decision-making is explored in relation to sensing technologies such ground-based sensors, GPS/GIS applications, and remote sensing. The essay delves further into the topic, exploring how variable rate methods like irrigation, nutrient application, and sowing affect input use and crop production. It goes on to talk about how the Internet of Things (IoT), remote sensing, and artificial intelligence are all part of precision crop management strategies that can improve farming. The potential advantages of precision agriculture are substantial, but there are still obstacles that must be overcome in the areas of affordability, availability, data administration, and training. In order to realize precision agriculture's full potential for efficient and sustainable food production, it is critical to conquer these obstacles. Improved crop yield, better resource management, and environmental sustainability can result from the broad adoption of precision agriculture, but only if farmers and stakeholders recognize and tackle these difficulties.

Said, Ayman & Karan, Dash. (2023). With an eye on improving intelligence and efficiency across a wide range of applications and sectors, this article investigates the ways in which cloud computing and artificial intelligence (AI) work together. A robust framework that can tackle complicated problems and open up new opportunities is the outcome of the combined efforts of AI and cloud systems. Key components of this integration, including analyzing data, optimizing resources, and deploying intelligent applications, are covered in the article. It shows how smarter decisions, more automation, and better system performance can result from AI-driven insights in cloud environments through examples and case studies.

Alrababa'h, Ne & Banimustafa, Ahmed. (2022). For software engineering projects to be successful, software estimate is crucial. The complexity, intangibility, diversity, and expertise/underlying technology required in software solutions make software effort prediction challenging. Estimating software development time in object-oriented software engineering projects is a common practice; this work seeks to improve software estimation accuracy by utilizing a data mining approach that combines Random Forests Regression with Use-Case Points analysis. Software effort estimation using our suggested method outperforms Use-Case Points estimation using R-Squared (R2) and other metrics like Mean Absolute Error (MAE) and Mean Squared Error (MSE) in terms of prediction accuracy.

Mahmood, Yasir et al., (2020). Software engineers and researchers have been working hard to improve the precision of software effort estimation (SEE) in recent years as a result of the field's rapid adoption of new technology and development approaches. There is an ongoing requirement for an accurate SEE since overestimation and underestimating are the two main obstacles to software development. Use case point (UCP) and expert judgment-based software development work estimation methodologies are discussed in this study, which presents a comprehensive assessment of studies related to best practices. Supporting researchers through a comprehensive review to facilitate other researchers' search for effort estimating studies is the primary purpose and contribution of this study. We conducted a state-of-the-art review from five different angles: (a) studies on UCPs and expert judgment-based effort estimation; (b) research contributions and future recommendations in various novelties; (c) dataset usage; (d) availability of accuracy metrics; and (e) study findings. Studies published between the years 2000 and 2019 were the subjects of our systematic review. In order to answer the research issues posed in this review, we combed through 34 original publications on UCP and estimating methods based on expert judgment. One of the most crucial aspects of expert judgment-based technique is the level of expertise within the team. Researchers use industrial datasets because they are readily available and easy to work with. There was an investigation into the fact that different evaluation accuracy criteria yield contradictory findings. Our research does not allow us to conclude that the UCP method produces the most promising outcomes when applied to algorithmic and machine learning techniques. However, datasets, methods, and evaluation criteria all have a role in how accurate SEEs are.

Dhas, John.T.. (2020). The software development process's planning activities are mostly dictated by the software's size. A well-planned software development process is essential for its effective conclusion. Assessment of software size, estimation of effort in person hours or person months, calculation of cost and budget, preparation of a timetable, and allocation of resources and tasks are all steps in the planning process. When planning the development process, knowing the program size is crucial because The development process's effort, time, schedule, cost, and other variables are all affected by size. Size and size-based factors are causing problems with planning since the industry's sizing methodologies don't cover all parts of software. One of the most crucial aspects of project management is planning. This research article examines the application emphasis of sizing approaches and focuses on the importance of sizing.

Dhas, John.T. & Bharathi, C.R.. (2015). One effective method for estimating the time and energy required to construct software systems is software size estimation. Precise time, cost, and effort estimates are crucial to the completion of any software project. Reasonable goals for finishing the project can be defined with the help of estimation. To estimate anything, size is the most fundamental factor. Various sizing approaches are utilized by the software business. There is insufficient support for the use of lines of code, features, functions, use cases, objects, the internet, etc., to determine the size of E-Commerce systems. Imperfections, losses, and dissatisfied customers result from inaccurate estimations. In this paper, we'll look at the most popular size methods, the problems with them, and why a new strategy is needed for E-Commerce platforms.

Tunalı, Volkan. (2014). In order to successfully plan a software project, it is vital to have a decent idea of how big the software that needs to be produced is. The scope of software projects can be estimated using a variety of techniques. Function Point Analysis (FPA) is a popular and well-known technique. This research used the FPA technique to estimate the size of a software development project for mobile devices. The actual size of the project after development was compared to the estimate, and the findings were given.

III. PROPOSED METHODOLOGY

As part of our research, we analyzed ten separate systems, taking into account each one's unique technological requirements, environmental conditions, and the expertise of the development teams. Our estimates for the actor weight and use case weight are based on the use case diagrams of these systems. The number of use cases was determined by taking into account several technological and environmental variables of the system, as well as the unadjusted actor weight and use case weight. The following are displayed in table 1:

Table 1: Matrix of Use Case Point

Sr.No	**ProjID**	**UCP**				
		UUCW	**UAW**	**TCF**	**ECF**	**Count**
1	A	75	6	0.875	0.89	63.08
2	B	60	6	1.05	0.92	66.65
3	C	90	13	1.05	0.935	101.1
4	D	75	7	1.04	0.935	79.7
5	E	100	9	0.98	0.92	98.3
6	F	55	6	1.095	0.875	58.4
7	G	60	9	0.97	0.965	64.59
8	H	95	6	1.04	0.965	101.4
9	I	65	5	0.98	0.89	61.05
10	J	90	12	0.965	0.935	92.03

Based on this data, we have determined the development effort for various systems; the results may be seen in table 2.

Table 2: Matrix of Development Effort

Sr.No	**System**	**UCP**	
		System Size(Count)	**Effort (Person-months)**
1	A	63.08	7.01
2	B	66.65	7.41
3	C	101.1	11.23
4	D	79.7	8.86
5	E	98.30	10.92
6	F	57.90	6.43
7	G	64.59	7.18

8	H	101.10	11.23
9	I	61.05	6.78
10	J	92.03	10.23

IV. EXPERIMENTAL RESULTS

The impact of various environmental and technical factors on size estimation has been thoroughly examined. By applying use case point analysis, we may convert the unadjusted UCP to UCP by taking into account thirteen technical aspects and eight environmental factors. As shown in table 3, the following is the outcome of the correlation study conducted on both adjusted and unadjusted UCP:

Table 3: Correlation Between UUCP & UCP

	UUCP	UCP
UUCP	1	
UCP	0.968988	1

The impact of system attributes on project size estimation is demonstrated by the correlation analysis result. There is a highly significant correlation between the system attributes and the estimated project size. Figure 1 below shows a visual depiction of this impact.

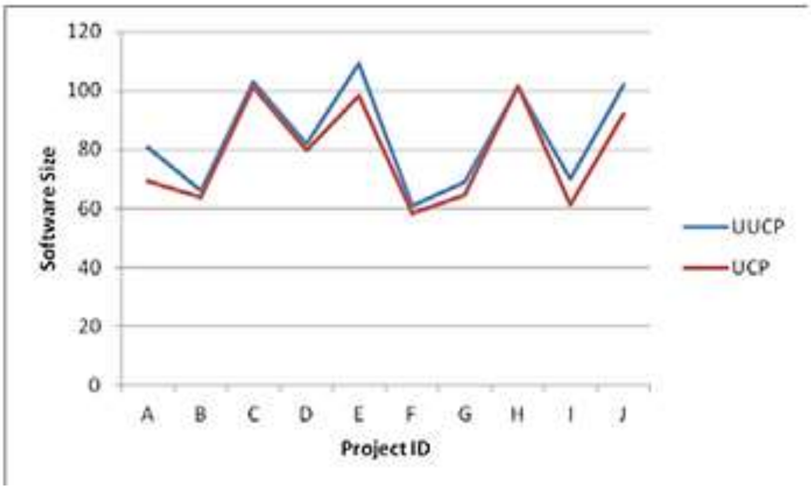

Figure 1: Graphical comparison of UUCP & UCP

You can see the impact of the various system parameters on the project size estimate in the graph. You can see the various project IDs on the X-axis, and the anticipated sizes of the projects using the use case point technique on the Y-axis. In addition, the graph displays the difference in project size while taking into account and not taking into account the different system characteristics.

V. CONCLUSION

The importance of Use Case Point (UCP) Analysis in improving software sizing accuracy and efficiency is highlighted by the evaluation of the method. This investigation has shed light on UCP Analysis's function in contemporary software development by delving into its basic ideas, approaches, difficulties, and possible improvements. UCP Analysis is a methodical approach to software size estimation by measuring a system's usefulness through use cases, which are encapsulations of user interactions. To provide a more complete picture of software size and complexity, UCP Analysis shifts the focus from standard code-based measurements to functional requirements. This approach allows for more sophisticated and precise size estimation by weighing several use case variables and computing the total UCP value.

To sum up, UCP Analysis does provide useful information about program sizing, but using it effectively requires fixing its flaws and making smart upgrades. More informed project management and resource allocation decisions can be made in the ever-changing software development landscape with the help of UCP Analysis, which improves software sizing accuracy and efficiency through methodology refinement, automation, integration with complementary metrics, and continuous improvement.

REFERENCES: -

1. Alrababa'h, Ne & Banimustafa, Ahmed. (2022). Random Forests for Predicting Software Effort Estimation Based on Use-Case Point Analysis.

2. Dhas, John.T. & Bharathi, C.R.. (2015). Relative analysis of sizing methods in the sense of e-commerce system. 10. 39808-39816.
3. Dhas, John.T.. (2020). IMPORTANCE OF SOFTWARE SIZING IN SOFTWARE PROJECT MANAGEMENT: A STUDY. Italian Journal of Pure and Applied Mathematics. 118. 269-273.
4. Ferchichi, J.P.Bourey, M.Bigand, M.Barron (2006) "Design System engineering of software products" implementation of a software estimation model", IMACS-2006, Beijing, China
5. Mahmood, Yasir & Kama, Nazri & Azmi, Azri. (2020). A systematic review of studies on use case points and expert-based estimation of software development effort. Journal of Software: Evolution and Process. 32. 10.1002/smr.2245.
6. Richard D. Stutzke (2005) "Estimating Software-Intensive systems", Addison Wesley.
7. Rizvi, Mohammed. (2023). Enhancing cybersecurity: The power of artificial intelligence in threat detection and prevention. International Journal of Advanced Engineering Research and Science. 10. 055-060. 10.22161/ijaers.105.8.
8. Robert t. Futrell, Donald F. Shafer, Linda I. Shafer, "Quality software project Said, Ayman & Karan, Dash. (2023). AI Integration in Cloud Systems: Enhancing Intelligence and Efficiency. 10.13140/RG.2.2.31220.12168.
9. Sharma, Shikha. (2023). Precision Agriculture: Reviewing the Advancements, Technologies, and Applications in Precision Agriculture for Improved Crop Productivity and Resource Management. 4. 41-45. 10.26480/rfna.02.2023.41.45.
10. Tunalı, Volkan. (2014). Software Size Estimation Using Function Point Analysis – A Case Study for a Mobile Application.