https://doi.org/10.26160/2474-5901-2024-42-42-49

ESTIMATION AND ASSESSMENT OF SOFTWARE TESTING EFFORTS

Shaikhulov E.
Kazan Innovative University named after V.G. Timiryasov, Kazan, Russia

Keywords: software testing, effort estimation, testing efforts, parametric models, empirical models, expert judgment, automation, agile methodologies.
Abstract. This article addresses issues of estimating and assessing testing efforts. Drawing on the experiences of companies such as Boeing, IBM, and Snapchat, various estimation models are analyzed, including parametric and empirical models, as well as expert judgment methods. Factors influencing labor efforts, such as project complexity and team qualifications, are discussed. The role of test automation in assessing potential costs is also explored.

ОЦЕНКА И АНАЛИЗ ЗАТРАТ НА ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Шайхулов Э.
Казанский инновационный университет имени В.Г. Тимирясова, Казань, Россия

Ключевые слова: тестирование программного обеспечения, оценка затрат, затраты на тестирование, параметрические модели, эмпирические модели, экспертное мнение, автоматизация, гибкие методологии.
Аннотация. В данной статье рассматриваются вопросы оценки и анализа затрат на тестирование. Опираясь на опыт таких компаний, как Boeing, IBM и Snapchat, анализируются различные модели оценки, включая параметрические и эмпирические модели, а также методы экспертной оценки. Обсуждаются факторы, влияющие на трудозатраты, такие как сложность проекта и квалификация команды. Также рассматривается роль автоматизации тестирования в оценке потенциальных затрат.

Introduction

Software testing (SWT) is an integral component of the software development lifecycle. The estimation of efforts required for testing remains a pivotal aspect of project management, influencing budget allocations, resource planning, and the overall project timeline. Accurate estimation of these efforts aids in mitigating risks associated with project delays and cost overruns, thereby enhancing project management efficacy.

The goal of this study is to elucidate various methodologies and models for estimating testing efforts (TE) within software projects.

Main part. Overview of testing methodologies

SWT is a systematic and crucial procedure in the software development lifecycle that involves evaluating and verifying the functionality, performance, and reliability of the program. The global SWT market is expected to grow to $68.01 billion by 2030 (fig. 1).

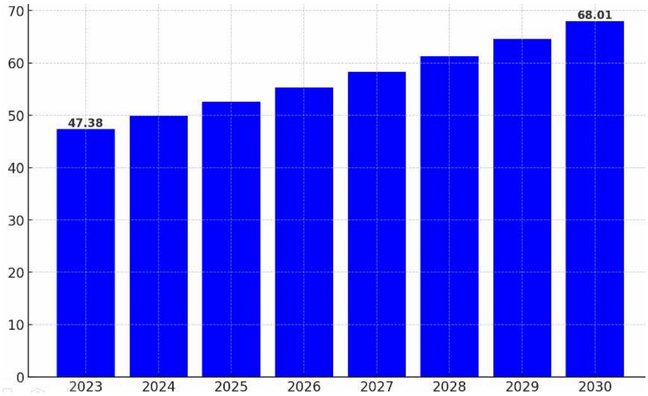

Fig. 1. Global SWT market size, billion dollars [1]

Testing methodologies in software development play a crucial role in defining how products are evaluated and refined. They are integral to managing quality and ensuring the final product meets all specified requirements. These methodologies can be broadly classified into traditional and agile methods, each catering to different project needs and offering distinct advantages and challenges in terms of effort estimation and project management:

– Traditional testing methodologies such as the **Waterfall** model advocate a systematic, sequential approach to SWT [2]. In the Waterfall model, testing commences only after the completion of the development phase, adhering to a rigid structure where each phase must be completed before the next begins. This method is highly structured and allows for extensive documentation and detailed planning, making it suitable for projects with clear, unchanging requirements.

The V-model, an extension of the Waterfall model, incorporates a more integrated approach to testing by associating each development stage with a specific testing phase [3]. Known as the Validation and Verification model, it enhances the focus on quality from the early stages of the software development lifecycle. This model is particularly effective in environments where failure can have serious safety or financial implications, as it systematically addresses quality at each step of development.

– Agile testing methodologies such as those implemented in **Agile** or **Scrum** frameworks emphasize flexibility and continuous testing throughout the development process [4]. Agile methodologies integrate testing as an ongoing activity, allowing for constant adjustments based on user feedback and evolving requirements. This iterative process involves repeated cycles or sprints, where development and testing occur simultaneously, promoting frequent reassessment and adaptation of both product and project scopes.

Agile testing is characterized by its collaborative nature, involving regular communication among cross-functional teams and stakeholders. This approach supports a dynamic development environment where requirements are anticipated to change and rapid product iterations are necessary. It is particularly suited to projects requiring frequent updates and quick adaptation to new information or user feedback.

The choice between traditional and agile testing methodologies significantly impacts the estimation of TE and the management of software development projects. Traditional methods provide a structured framework ideal for projects with fixed requirements and significant risk, whereas agile methods offer flexibility and adaptability, crucial for projects with dynamic scopes and the need for rapid development [5]. Understanding the distinct characteristics of each methodology allows project managers to align their testing strategies more effectively with project goals and stakeholder expectations, ensuring the timely delivery of a high-quality product.

Estimation models for TE

Accurately estimating TE is essential for the successful management of software development projects. Different estimation models provide frameworks and techniques to predict the resources and time required for testing, which helps in planning, budgeting, and resource allocation. These models can be broadly categorized into parametric models, empirical models, and expert judgment.

Parametric models use mathematical formulas to estimate the efforts based on a set of defined parameters. One of the most recognized parametric models is the **Constructive Cost Model** (COCOMO). COCOMO estimates the effort and cost of a software project by using historical data and current project parameters such as size of the software, complexity, required reliability, and the team's capability [6]. It provides a detailed and systematic approach to estimation, which can be tailored to fit various types of software projects.

Another parametric model is the Software Life Cycle Management (SLIM) model. SLIM uses a Rayleigh curve to model the staffing and time requirements across the lifecycle of the software development process. It is based on the relationship between time, effort, and the number of people involved, offering a more dynamic view of the entire development process.

Empirical models are based on analysis of data from completed projects to predict future efforts. By analyzing trends and outcomes from previous testing phases, empirical models can provide estimates that reflect the specific characteristics and performance of the testing team and environment.

For instance, the use of regression analysis in empirical models allows for the identification of key factors that significantly impact the testing effort. This method can adapt as more project data becomes available, continuously refining the accuracy of its predictions. Empirical models are inherently flexible and can be customized to the unique processes and metrics of a specific organization.

Expert judgment involves relying on the experience and intuition of seasoned professionals to estimate TE. This method is often used in conjunction with other models to fine-tune estimates based on the nuances of the project that might not be fully captured by parametric or empirical models.

Techniques such as the Delphi Method are employed to consolidate expert opinions. This method involves multiple rounds of anonymous feedback from a panel of experts, allowing consensus to be reached without the influence of dominant individuals. Expert judgment is invaluable, especially in new or highly innovative projects where historical data may be sparse or irrelevant.

Choosing the right model for estimating TE depends on various factors including the availability of data, the nature of the software project, and the organization's experience with similar projects. Parametric and empirical models offer structured approaches that can leverage historical data and mathematical formulas, providing a solid basis for estimates. However, expert judgment brings in the necessary flexibility and adaptability, particularly valuable in projects with unique or unprecedented elements. Combining these approaches often yields the best results, balancing data-driven precision with experienced insight to guide TE effectively. This integrated approach ensures that estimates are both realistic and tailored to the specific context of the project.

Factors influencing TE

Accurately estimating TE is critical for the effective management and execution of software projects. Several factors influence the time and resources required for testing, each varying in its impact based on the project's unique attributes. These factors include the complexity of the application, the experience of the testing team, and the methodologies employed. Understanding these influences is important for project managers to prepare realistic schedules and budgets. A list of these factors is presented in Table 1.

Tab. 1. Factors influencing TE [7, 8]

Factor category	Specific factors	Impact on TE
Project complexity	Software size, number of integrations, feature complexity	Larger and more complex projects typically demand more extensive and thorough TE
Technical challenges	Technology stack complexity, legacy system integration, requirement for specialized testing (e.g., security, load)	Advanced or unfamiliar technologies and legacy systems integration generally extend testing durations due to the higher risk of defects and issues.
Human factors	Team expertise, team size, communication efficiency	Highly skilled and larger teams might reduce testing time, whereas poor communication can lead to misunderstandings and rework, increasing efforts.
Methodological approach	Development methodology (Agile vs. Waterfall), testing tools and automation	Agile methods with iterative testing might speed up some processes but require continuous testing. The use of automated testing tools can significantly decrease manual testing time but requires upfront time investment for setup.
External factors	Regulatory compliance, market-driven deadlines	Regulatory requirements often necessitate additional testing to ensure compliance, while tight deadlines can pressure testing teams to increase efforts or adjust scope.

From the author's perspective, the human factors detailed in Table 1, especially the expertise of the testing team, are crucial in optimizing the TE. This view is corroborated by the experience of AcmeTech Solutions, which implemented a comprehensive training program for its testing staff. By enhancing their team’s skills in automated testing tools and agile methodologies, AcmeTech noticed a significant reduction in their cycle times for testing. This shift not only improved the quality of their software products but also enabled the team to handle complex testing scenarios more efficiently, showcasing the profound impact of skilled human resources on testing efficacy.

Application of estimation models in practice

The effective use of estimation models in TE can significantly influence the success or failure of software development projects.

Boeing successfully implemented the COCOMO for estimating the effort required for their software development in avionics systems. This extensive project involved complex software integrations across multiple aircraft systems. Boeing opted for COCOMO because of its ability to factor in various project sizes and complexities by adjusting parameters such as the scale of the project and the experience of the team.

For Boeing, the key to the success of the COCOMO model was the availability of extensive historical data from previous projects, which allowed for accurate calibration of the model's parameters. This precise parameter adjustment ensured that the effort estimates were reliable, and the project adhered to its schedule and budget. Boeing's experience underscores the importance of detailed project records and data analytics in enhancing the accuracy of predictive models like COCOMO.

IBM has successfully integrated the COCOMO to estimate TE within its software development projects. By applying the COCOMO model, IBM can effectively forecast the resources required for testing each phase of development, allowing for strategic planning and allocation of testing resources. This parametric model has been particularly beneficial in scaling IBM’s larger, more complex projects, ensuring that TE are well-aligned with development timelines and project budgets.

Snapchat, in its fast-paced mobile app development environment, employs empirical models alongside expert judgment to estimate TE. By analyzing historical data from past projects, Snapchat's project managers are able to identify patterns and baselines that inform current testing effort estimations. This empirical approach is complemented by the Delphi method. This combination allows Snapchat to adapt to rapid iteration cycles and user-driven feature updates, ensuring that testing resources are accurately projected and efficiently utilized.

These case studies from Boeing, IBM and Snapchat illustrate the practical application of different estimation methods tailored to the needs of the organization and the nature of the projects. Boeing and IBM are use of the COCOMO model exemplifies how parametric estimation can provide robust, scalable frameworks for large-scale SWT. In contrast, Snapchat's integration of empirical models with expert judgment using the Delphi method showcases a flexible, adaptive approach suitable for the dynamic demands of mobile app development. Both strategies highlight the importance of selecting appropriate testing effort estimation techniques to enhance project management and ensure the successful delivery of software solutions.

The role of automation in improving effort estimations

The integration of automation into the testing process significantly enhances the precision of estimating the human efforts required for software projects. By utilizing specific automation tools and software, such as Selenium for web application testing, JMeter for performance testing, and Jenkins for continuous integration, the need for manual test execution is substantially reduced. These tools automate repetitive and time-consuming tasks, thus reducing the human effort involved and allowing testers to concentrate on more complex testing challenges that require nuanced judgment [9].

Selenium, for instance, automates web browsers, which allows teams to execute numerous tests concurrently across various browsers and platforms without direct human oversight. This capability decreases the time humans spend on routine test execution and increases the reliability of effort estimations by providing consistent test outputs. JMeter, on the other hand, simulates heavy loads on networks or servers to check performance under different conditions, which can be crucial for understanding the endurance and scalability of web applications without significant human intervention.

Jenkins automates aspects of software development related to building, testing, and deploying, enabling continuous integration and continuous delivery with minimal human effort. This automation ensures that development teams can detect problems early, thus reducing the time and human resources needed for troubleshooting and fixing issues.The predictability provided by these automation tools aids in more accurately estimating the human effort required for testing projects.

Conclusion

Accurate estimation of TE ensures efficient resource allocation and project management. Parametric models calculate effort based on software size, while empirical models adjust estimates using historical data. Expert judgment adds depth, addressing specific project nuances. Combining these approaches enhances the precision of testing effort estimates, crucial for meeting project timelines and managing workload effectively.

References

1. Software Testing Market Overview [Electronic resource]. – Access mode: https://www.statista.com/statistics/1401409/popular-ai-uses-in-development-workflow-globally/#:~:text=According%20to%20the%202023%20Stack,using%20it%20as%20of%202023.

2. Ogarkov A. Enhancing commercial effectiveness in pharmaceuticals: a case study on the implementation of successful sales and marketing strategies // Issues of management and economics: current state of current problems: collection. Art. based on materials from the LXXX International Scientific and Practical Conference. – M.: Publ. house "Internauka", 2024. – No. 2(71). – P. 81-88.
3. Zakharau A. The impact of fintech application implementation on improving consumer credit ratings // Sciences of Europe. 2024, vol. 138, pp. 14-16. DOI: 10.5281/zenodo.10957276.
4. Shravan P. A Comprehensive Research Analysis of Software Development Life Cycle (SDLC) Agile & Waterfall Model Advantages, Disadvantages, and Application Suitability in Software Quality Engineering // Software Quality Engineer. 2023, vol. 13(8), pp. 120-124. DOI: 10.29322/IJSRP.13.08.2023.p14015.
5. Kaliuta K. Personalizing the user experience in Salesforce using AI technologies // Computer-Integrated Technologies: Education, Science, Production. 2023, vol. 24(52), pp. 48-53.
6. Feizpour E., Tahayori H., Sami A. CoBRA without experts: New paradigm for software development effort estimation using COCOMO metrics // J Softw Evol Proc. 2023, vol. 35(12), p. 2569.
7. Kassaymeh S., Alweshah M., Al-Betar M.A., Hammouri A.I., Ma'aitah M.K. Software effort estimation modeling and fully connected artificial neural network optimization using soft computing techniques // Cluster Comput. 2024, vol. 27, pp. 737-760.
8. Bukhtueva I. The impact of AI technologies on business performance // Bulletin of Science. 2024, vol. 5, no. 3(72), pp. 467-476.
9. Bukhtueva I.A. Impact of AI-enabled software on organizational cost reduction // Universum: Economics and Law. 2024, vol. 4(114), pp. 41-45. URL: https://7universum.com/ru/economy/archive/item/17041.

Список литературы

1. Software Testing Market Overview [Электронный ресурс]. – Режим доступа: https://www.statista.com/statistics/1401409/popular-ai-uses-in-development-workflow-globally/#:~:text=According%20to%20the%202023%20Stack,using%20it%20as%20of%202023.
2. Ogarkov A. Enhancing commercial effectiveness in pharmaceuticals: a case study on the implementation of successful sales and marketing strategies // Вопросы управления и экономики: современное состояние актуальных проблем: сб. ст. по материалам LXXX Международной научно-практической конференции. – М.: Изд-во «Интернаука», 2024. – № 2(71). – С. 81-88.
3. Zakharau A. The impact of fintech application implementation on improving consumer credit ratings // Sciences of Europe. 2024, vol. 138, pp. 14-16. DOI: 10.5281/zenodo.10957276.
4. Shravan P. A Comprehensive Research Analysis of Software Development Life Cycle (SDLC) Agile & Waterfall Model Advantages, Disadvantages, and Application Suitability in Software Quality Engineering // Software Quality Engineer. 2023, vol. 13(8), pp. 120-124. DOI: 10.29322/IJSRP.13.08.2023.p14015.
5. Kaliuta K. Personalizing the user experience in Salesforce using AI technologies // Computer-Integrated Technologies: Education, Science, Production. 2023, vol. 24(52), pp. 48-53.

6. Feizpour E., Tahayori H., Sami A. CoBRA without experts: New paradigm for software development effort estimation using COCOMO metrics // J Softw Evol Proc. 2023, vol. 35(12), p. 2569.
7. Kassaymeh S., Alweshah M., Al-Betar M.A., Hammouri A.I., Ma'aitah M.K. Software effort estimation modeling and fully connected artificial neural network optimization using soft computing techniques // Cluster Comput. 2024, vol. 27, pp. 737-760.
8. Bukhtueva I. The impact of AI technologies on business performance // Вестник науки. – 2024. – Т. 5, № 3(72). – Р. 467-476.
9. Bukhtueva I.A. Impact of AI-enabled software on organizational cost reduction // Universum: Экономика и юриспруденция. – 2024. – Т. 4(114). – С. 41-45. – URL: https://7universum.com/ru/economy/archive/item/17041.

Шайхулов Эдуард – бакалавр	**Shaikhulov Eduard –** bachelor's degree
bringomun@rambler.ru	

Received 17.06.2024