The Academic College of Tel Aviv-Yaffo

Digital Project Management

72000901

Aggregated analysis of Software Effort Estimation Models and techniques

Submitted by:

Yoav Levinger

Elena Kirin

Grigory Shaulov

Instructor:
Dr. Eyal Rozen

Master's Degree in Information Systems
2024

Table of Contents

Abstract

This paper offers a detailed examination of the various methodologies used in software effort estimation (SEE), leveraging a wide array of sources including scientific studies, articles, white papers, and insights from management and version control tools like Jira and GitHub. Through the application of automated tools for data extraction, collection, and analysis, this study reviews multiple papers, selecting those that provide substantial information and results relevant to SEE. The analysis extracts these findings into a coherent overview of existing methods, proposes enhancements, and compares them across eight critical parameters. This thorough approach concludes in a comprehensive comparison table, shedding light on the efficacy of various methods in contributing to accurate SEE. The document aims to equip project managers and software development teams with a understanding of each method's strengths and limitations, thereby enabling the selection and tailoring of effort estimation techniques that best suit project-specific requirements.

Introduction

Accurate software effort estimation (SEE) is crucial to the successful planning, budgeting, and execution of software projects. Despite its critical importance, the challenge of estimating software development effort with precision continues, due to the inherent complexity and dynamic nature of software projects. Addressing this challenge, our inquiry reviews existing literature and multiple papers to identify those that offer significant insights into SEE methods and techniques. Using automated data analysis tools, we extract and analyze case studies, review methods, and suggest improvements, encapsulating our findings in a series of lists and explanations that cover reviewed and suggested methods. Furthermore, we devise a list of eight parameters for method comparison, leading to a detailed comparison table that highlights each method's benefits and drawbacks.

This document not only presents an aggregated analysis but also *proposes an innovative tool* designed to integrate with existing management platforms. This tool aims to streamline the application of selected SEE methods, enhancing project management and development processes by providing detailed task breakdowns, effort estimations, and technical guidance, thereby facilitating more accurate predictions and efficient resource allocation. The initiative underscores the importance of continuous research, methodological innovation, and the development of supportive tools to meet the evolving demands of software development effort estimation.

Methodology

We have reviewed a list of papers consisting of scientific research, articles and white papers dealing with software effort estimation (SEE) models, methods and techniques, some of which presented a case study, others processed relevant datasets taken from management tools like Jira or source control like GitHub.
Due to the amount of data and the complexity level we have used automated tools to extract, collect and analyze most of the data.

The stages of analysis performed as follows:

1. Reviewed more than 200 papers for relevancy to this analysis from which we have selected 47 relevant papers with adequate information.

2. We have extracted the following information: case studies, reviewed methods for SEE, suggested methods.
3. We have created a list of all the methods reviewed along with a brief explanation of each method.
4. We have created a list of suggested method(s) derived from the reviewed papers.
5. We have created a brief explanation of each suggested method.
6. We have devised a list of eight parameters for comparing the suggested methods.
7. Each suggested method was reviewed by the devised list of parameters.
8. We created a comparison table of the final list of suggested methods by the devised parameters.
9. We have highlighted the results in the table according to its contribution to SEE.

Software Effort Estimation Models

The papers reviewed various methods for estimating software development task efforts as a starting point, including traditional, agile, and advanced computational techniques. Here's a summarized list of the discussed methods within the referenced papers, all aggregated in *four major groups* along with a brief explanation of each software effort estimation model:

Group A: Traditional and Agile Methods

1. **Story Points**: An agile estimation method where tasks are rated based on complexity, effort, and risk, using an arbitrary scale (e.g., Fibonacci sequence).

2. **Ideal Days/Hours**: Estimates the amount of time a task would take in a perfect world, with no interruptions.

3. **Planning Poker:** An agile estimation technique where team members make estimates by playing numbered cards face-down to the table, avoiding the influence of other members.

4. **Wideband Delphi:** A consensus-based technique where experts estimate software development efforts in multiple rounds, refining their estimates until convergence is reached.

5. **Expert Judgment:** Relies on the experience and intuition of experts to estimate the effort required for software projects.

6. **Use Case Point:** An estimation method that calculates effort based on the complexity of use cases and technical and environmental factors.

7. **Velocity (Actual and Expected):** Measures the amount of work a team can complete in a sprint. Actual velocity is measured based on completed work, while expected velocity is projected based on past performance.

Group B: Computational and Machine Learning Techniques

1. **Bayesian Network Model:** Uses Bayesian networks to predict software development effort by modeling probabilistic relationships between variables.

2. **COCOMO I:** The Constructive Cost Model is an algorithmic software cost estimation model based on project size, complexity, and other factors.

3. **Function Point Analysis:** An estimation technique based on the functionality delivered, considering various elements like inputs, outputs, user interactions, and external interfaces.

4. **Top-down/Bottom-up Estimation:** Top-down involves estimating the project as a whole and then breaking down, while bottom-up involves estimating individual parts of the project and summing them up.

5. **Artificial Neural Networks:** Utilizes ANN to predict software effort based on training data from past projects.

6. **Support Vector Machine:** A machine learning model that can be used to predict software effort by finding the hyperplane that best separates data points of different categories.

7. **Pre-trained embedding models (e.g., BERT):** Leveraging pre-trained language models to understand project requirements and predict effort based on semantic understanding.

8. **Planning Poker combined with developer expertise:** Integrates the Planning Poker method with machine learning models that incorporate historical data and developer expertise.

Group C: Advanced Optimization and Estimation Models

1. **Differential Evolution (DE) for feature weight optimization:** Uses the DE algorithm to optimize the weights of features in estimation models to improve accuracy.

2. **Ensemble models combining various machine learning algorithms:** Combines predictions from different models to improve estimation accuracy.

3. **Harmony Search (HS):** An optimization algorithm inspired by the process of musical improvisation, used for optimizing estimation models.

4. **Satin Bowerbird Optimization (SBO) for optimizing ANFIS:** Uses SBO, an optimization technique inspired by the bowerbird's behavior, to optimize the parameters of Adaptive Neuro-Fuzzy Inference Systems for better estimation.

5. **Bayesian Belief Network optimized with Genetic Algorithm and Particle Swarm Optimization:** Combines Bayesian networks with optimization algorithms to fine-tune the network structure and parameters for improved effort estimation.

Group D: Hybrid and Novel Approaches

1. **Use of checklists to support expert judgment:** Enhances expert judgment with structured checklists to ensure all relevant factors are considered in the estimation process.

2. **Group Process Modeling using the Choquet Integral:** Applies the Choquet Integral in group decision-making to aggregate individual estimates into a collective estimate, considering the interdependencies of factors.

3. **Prediction-Based Techniques categorizing projects by complexity and developer expertise:** Uses prediction models to classify projects and tailor estimation techniques based on project complexity and the expertise of the development team.

4. **Effort Estimation using Bio-Inspired Algorithms for feature selection:** Utilizes algorithms inspired by biological processes, like genetic algorithms or particle swarm optimization, to select the most relevant features for effort estimation models.

Case studies

Some of the scientific papers present case studies regarding software effort estimation, these summaries highlight the diversity of methods and case studies across different contexts within software effort estimation research, demonstrating the application of traditional, machine learning, and bio-inspired algorithms tailored to specific project needs or data sources.

1. **Paper title:** "An Empirical Evaluation of Ensemble Adjustment Methods for Analogy-Based Effort Estimation" (Azzeh, M., Nassif, A.B. and Minku, L.L., 2015. *Journal of Systems and Software*, *103*, pp.36-52.)
   - **Case Study:** Large-scale comparison using eight datasets.
   - **Recommended Method:** Ensembles of linear adjustment methods.
   - **Explanation:** Combining strengths of individual methods improves accuracy and precision by leveraging diverse methodological insights.

2. **Paper title:** "Assessing the Effectiveness of Approximate Functional Sizing Approaches for Effort Estimation" (Di Martino, S., Ferrucci, F., Gravino, C. and Sarro, F., 2020. *Information and Software Technology*, *123*, p.106308.)
   - **Case Study:** Evaluates approximate functional sizing methods on 25 industrial software projects by an Italian company specializing in enterprise information systems (company unnamed).
   - **Recommended Method:** High Level FPA and COSMIC's Equal Size Bands approximation.
   - **Explanation:** These methods offer early size measures for effort prediction with comparable accuracy to standard methods.

3. **Paper title:** "Definition and Evaluation of a COSMIC Measurement Procedure for Sizing Web Applications in a Model-Driven Development Environment" (Abrahão, S., De Marco, L., Ferrucci, F., Gomez, J., Gravino, C. and Sarro, F., 2018. *Information and Software technology*, *104*, pp.144-161.)
   - **Case Study:** Evaluation with data from 30 Web applications developed using the OO-H method.
   - **Recommended Method:** OO-HCFP (Object-Oriented Hypermedia COSMIC Function Points).
   - **Explanation:** Provides accurate size and effort estimates by considering specific characteristics of Web applications.

4. **Paper title:** "Effort Estimation in Large-Scale Software Development: An Industrial Case Study" (Usman, M., Britto, R., Damm, L.O. and Börstler, J., 2018., *Information and Software technology*, *99*, pp.21-40.)
   - **Case Study:** Ericsson's large-scale distributed agile project involving 188 employees.
   - **Recommended Method:** A two-stage effort estimation process.
   - **Explanation:** Incorporates initial high-level quotes followed by detailed analysis estimates for improved accuracy.

5. **Paper title:** "Improving Software Effort Estimation Using Bio-Inspired Algorithms to Select Relevant Features: An Empirical Study"(Ali, A. and Gravino, C., 2021. *Science of Computer Programming*, *205*, p.102621.)
    - **Case Study:** Empirical analysis using eight publicly available datasets.
    - **Recommended Method:** Harmony Search (HS).
    - **Explanation:** Outperforms both bio- and non-bio-inspired algorithms in predicting effort across various datasets.
6. **Paper title:** "Satin Bowerbird Optimizer: A New Optimization Algorithm to Optimize ANFIS for Software Development Effort Estimation" (Moosavi, S.H.S. and Bardsiri, V.K., 2017. *Engineering Applications of Artificial Intelligence*, *60*, pp.1-15.)
    - **Case Study:** Optimization of the Adaptive Neuro-Fuzzy Inference System (ANFIS) for effort estimation.
    - **Recommended Method:** Satin Bowerbird Optimization (SBO).
    - **Explanation:** Utilizes behavior of satin bowerbirds to optimize ANFIS parameters, improving estimation accuracy.
7. **Paper title:** "Software Effort Estimation Based on Open-Source Projects: Case Study of GitHub" (Qi, F., Jing, X.Y., Zhu, X., Xie, X., Xu, B. and Ying, S., 2017. *Information and Software Technology*, *92*, pp.145-157.)
    - **Case Study:** Empirical study using GitHub data for collecting real-life effort data.
    - **Recommended Method:** Automated Function Point (AFP) approach combined with AdaBoost and Classification and Regression Tree (ABCART).
    - **Explanation:** Leverages diverse open-source project data for dynamic dataset expansion and accurate effort estimation.
8. **Paper title:** "Developing and Using Checklists to Improve Software Effort Estimation: A Multi-Case Study" (Usman, M., Petersen, K., Börstler, J. and Neto, P.S., 2018. *Journal of Systems and Software*, *146*, pp.286-309.)
    - **Case Study:** Infoway, Diyatech, and TSoft implementing customized checklists to enhance software effort estimation accuracy within agile environments.
    - **Recommended Method:** Customized checklists to support expert judgment.
    - **Explanation:** Aims to reduce underestimation bias and enhance estimation precision through systematic consideration of factors.
9. **Paper title:** "Bayesian Network Model for Task Effort Estimation in Agile Software Development". (Dragicevic, S., Celar, S. and Turic, M., 2017. *Journal of systems and software*, *127*, pp.109-119.)
    - **Case Study:** Agile projects in a software company (name undisclosed).
    - **Recommended Method:** Bayesian Network model.

- **Explanation:** Emphasized for its simplicity, minimal data requirements, and suitability for agile methods due to its ability to incorporate expert judgment or empirical data.

10. **Paper title:** "Factors Affecting Duration and Effort Estimation Errors in Software Development Projects" (Morgenshtern, O., Raz, T., & Dvir, D. (2007). *Information and Software Technology*, *49*(8), 827-837.)
    - **Case Study:** The case study described in the paper involved the IT division of a large government organization in Israel, focusing on 43 internal software development projects executed during 2002 and completed by mid-2003.
    - **Recommended Method:** The paper recommended a method involving a comprehensive analysis of factors affecting estimation accuracy, particularly focusing on project uncertainty, estimation development and management processes, and estimator's experience.
    - **Explanation:** The method entails analyzing the impact of these factors on estimation errors through statistical analysis, including correlation analysis and regression models. It emphasizes reducing estimation errors by managing project uncertainty, improving estimation development and management processes, and leveraging estimator experience.

The proposed methods extracted from the articles

The documents review multiple studies, most recommending a method or methods for software development task effort estimation. Each method is evaluated based on factors such as accuracy, precision, complexity, flexibility, cost-effectiveness, scalability, usability, and integration capability with existing software development processes. The document highlights the diversity in effort estimation methods, suggesting that the choice of method depends on project specifics, available data, and organizational context.

Here's a summary of each of the recommended methods along with our notion of possible utilization by project managers:

1. **Method/Model:** Bayesian Network Model

   Description: Utilizes Bayesian networks for agile development, praised for simplicity and minimal data needs. Uses probabilistic graphical models to estimate efforts by considering dependencies between variables.
   Possible utilization by project managers: Project managers can input project factors (like size, complexity) into a software tool to get effort estimates.
   Analysis by parameters: High accuracy and precision, low complexity, very flexible, cost-effective, scalable to data size, highly usable early in projects, integrates well with agile processes.

2. **Method/Model:** Customized Checklists

   Description: Supports expert judgment in estimation, addressing common estimation issues. Checklists tailored to project specifics to assist in effort estimation.
   Possible utilization by project managers: Create a checklist of common tasks and adjust effort based on past projects to estimate new tasks.
   Analysis by parameters: Improved accuracy by reducing bias, enhances precision, low

complexity, highly flexible and customizable, cost-effective, scalable, user-friendly, integrates easily with agile methodologies.

3. **Method/Model:** Choquet Integral

   Description: Aggregates expert and non-expert estimates in project management, capturing interdependencies between factors. The Aggregation of individual estimates using fuzzy measures to consider interdependencies.
   Possible utilization by project managers: Gather estimates from team members on tasks, use software to combine them accounting for task complexities.
   Analysis by parameters: Enhances accuracy by capturing interdependencies, precise aggregation, moderate complexity, very flexible, potentially resource-intensive initially, scalable, sophisticated usability, integrates into project management frameworks.

4. **Method/Model:** Prediction-Based Technique

   Description: Categorizes projects by complexity and developer expertise for cost estimation to predict efforts.
   Possible utilization by project managers: Classify upcoming projects by size and assign based on team's past performance for accurate effort prediction.
   Analysis by parameters: Improved accuracy through categorization, enhanced precision, moderate complexity, highly flexible, potentially high cost-effectiveness, scalable, practical for agile environments, easily integrates into agile methodologies.

5. **Method/Model:** SE3M Model

   Description: Utilizes semantic analysis of requirements using pre-trained BERT models for effort estimation from textual requirements.
   Possible utilization by project managers: Analyze project requirements through AI tools to get effort estimates, requiring minimal manual input.
   Analysis by parameters: High accuracy with contextual understanding, precise due to fine-grained semantic analysis, higher complexity, flexible for various requirements, cost-effective through model reuse, scalable, requires machine learning expertise, integrates into estimation workflows.

6. **Method/Model:** Task Planning Model

   Description: Uses generative models for predicting and replicating tasks, considering project size and duration.
   Possible utilization by project managers: Use project planning tools that suggest tasks and efforts based on project scope and timeline.
   Analysis by parameters: Aims for high accuracy with task replication, enhanced precision, higher complexity, adaptable to project variations, improved cost-effectiveness through planning, scalable, requires understanding of task generation, compatible with various management approaches.

7. **Method/Model:** "vpbench" Framework

   Description: Generates benchmarks for evaluating variant-rich software evolution, focusing on metadata precision. Benchmarks variant-rich software projects to estimate efforts based on metadata.
   Possible utilization by project managers: Evaluate past project data, focusing on variation, to

estimate effort for new projects with similar traits.
Analysis by parameters: Not directly applicable to task effort estimation but provides precise metadata for benchmarking, complex framework, flexible simulation capabilities, improves research and tool evaluation cost-effectiveness, scalable, specialized usability, integrates with software evolution methods.

8. **Method/Model:** Developer's Expertise-Based Estimation

   Description: Incorporates experienced developer input for project cost and time estimation, leveraging team expertise.
   Possible utilization by project managers: Consult with experienced team members on their effort estimates for tasks, adjusting based on project specifics.
   Analysis by parameters: High accuracy leveraging developer insights, improved precision, moderate complexity, highly adaptable, cost-effective with accurate estimates, scalable, user-friendly interface for inputs, integrates well with Scrum methodologies.

9. **Method/Model:** Ensemble Machine Learning Models

   Description: Combines several machine learning models to improve estimation accuracy: SVM, MLP-ANN, and GLM for effort and duration estimation, using an averaging approach.
   Possible utilization by project managers: Use machine learning tools that input project data and provide an averaged estimate from multiple models.
   Analysis by parameters: Highly accurate through algorithm strengths, precise with diverse models, complex but balanced, adaptable to various data, potentially cost-effective, scalable, requires machine learning knowledge, integrates into estimation processes.

10. **Method/Model:** Differential Evolution (DE)

    Description: Optimizes feature weights in analogy-based estimation, showing significant predictive performance improvement. An optimization technique that adjusts feature weights in estimation models for better accuracy.
    Possible utilization by project managers: Implement tools that use DE to fine-tune the estimation process based on project features.
    Analysis by parameters: Significant accuracy improvement, enhanced precision, moderate complexity, highly flexible, potentially cost-effective, scalable to dataset sizes, structured approach requires DE understanding, integrates into estimation frameworks.

11. **Method/Model:** Functional Stories and Issues

    Description: Combines agile methodologies with estimation techniques to improve accuracy, leveraging detailed project data.
    Possible utilization by project managers: Break down projects into stories and issues, estimating each separately for aggregated effort estimates.
    Analysis by parameters: Aims for enhanced accuracy with diverse data, tailored precision, moderate complexity, highly adaptable to agile projects, optimized cost estimation practices, applicable to various sizes, practical for government agency use, supports integration into DHS and DoD frameworks.

12. **Method/Model:** Multi-Layered Feed Forward ANN

    Description: Enhances neural network models for software effort estimation, focusing on adaptive learning capabilities. Artificial neural networks designed for adaptive learning to predict

software efforts.
Possible utilization by project managers: Utilize neural network-based tools to process project data for effort estimation, requiring historical data input.
Analysis by parameters: High potential accuracy, enhanced learning precision, relatively high complexity, flexible data handling, cost-effective over time, scalable to large datasets, technical setup required, integrates with effort estimation frameworks.

13. **Method/Model:** Productivity-Based UCP Models

    Description: Analyzes productivity as a factor in Use Case Points-based models, utilizing historical data for accuracy. Use Case Points method adjusted for productivity factors to estimate efforts.
    Possible utilization by project managers: Adjust Use Case Points estimations by incorporating team productivity metrics based on past projects.
    Analysis by parameters: Accuracy improved with historical data, enhanced precision, varied complexity, highly flexible, implied cost-effectiveness, scalable across projects, usability dependent on data availability, integrates with UCP-based estimation.

14. **Method/Model:** Harmony Search (HS)

    Description: An algorithm inspired by music improvisation to find optimal solutions for effort estimation. Recommended for future investigations due to superior performance in effort predictions across various datasets.
    Possible utilization by project managers: Apply HS optimization in estimation tools to find the best effort estimate based on project parameters.
    Analysis by parameters: better predictions across datasets, implied high precision, moderate complexity, highly adaptable, cost-effective through accuracy, scalable, requires bio-inspired algorithm knowledge, integrates with estimation techniques.

15. **Method/Model:** Satin Bowerbird Optimizer (SBO)

    Description: Optimization algorithm inspired by bowerbird behavior for parameter tuning in estimation models.
    Possible utilization by project managers: Use SBO-enhanced tools for effort estimation, focusing on fine-tuning model parameters for accuracy.
    Analysis by parameters: High expected accuracy, improved precision through tuning, moderate complexity, adaptable to project characteristics, potentially cost-effective, scalable, specialized knowledge required, integrates into effort estimation frameworks.

16. **Method/Model:** Automated Function Point (AFP) and ABCART

    Description: Combines function point analysis with machine learning for dynamic effort estimation. Combines AFP with AdaBoost and CART for dynamic dataset expansion and effort estimation, utilizing GitHub data.
    Possible utilization by project managers: Implement AFP tools that use data analysis and machine learning to adjust effort estimates in real-time.
    Analysis by parameters: High accuracy from diverse projects, enhanced precision, moderate complexity, highly adaptable using GitHub data, high potential cost-effectiveness, scalable, requires machine learning and data extraction knowledge, integrates well for data-scarce environments.

17. **Method/Model:** Optimal Bayesian Belief Network

Description: Bayesian networks optimized with genetic algorithms for effort estimation. Uses Bayesian networks optimized with GA and PSO, focusing on COCOMO components for accurate effort estimation.
Possible utilization by project managers: Use software that applies optimized Bayesian networks, inputting project specifics for accurate estimations.
Analysis by parameters: Enhanced accuracy with optimal adjustments, increased precision, moderately high complexity, adaptable optimization, potentially more cost-efficient, scalable to large projects, expertise required, enhances existing estimation processes.

18. **Method/Model:** Consistent Fuzzy Analogy-Based Estimation (C-FASEE)

    Description: Integrates fuzzy set theory for handling imprecision and uncertainty, improving accuracy and managing uncertainty.
    Possible utilization by project managers: Employ analogy and fuzzy logic tools to estimate efforts, especially in uncertain or changing conditions.
    Analysis by parameters: High accuracy with uncertainty handling, enhanced precision, moderate complexity, adaptable to various scenarios, improved cost-effectiveness, scalable, requires fuzzy logic expertise, integrates into decision-making frameworks.

19. **Method/Model:** Hybrid Model from Use Case Points

    Description: Enhances UCP with clustering and machine learning for effort prediction. Combines bisecting k-medoids clustering, SVM, and RBFNN based on UCP for improved productivity and effort prediction.
    Possible utilization by project managers: Combine traditional UCP estimation with machine learning insights based on project data clustering.
    Analysis by parameters: Significant accuracy improvement, enhanced two-stage precision, higher complexity, adaptable to different datasets, improved cost-effectiveness, scalable, requires machine learning and UCP understanding, integrates into early project estimation.

Results:

We used these eight parameters to compare methods of estimating software development task efforts:

1. **Accuracy** of the estimates in terms of the closeness of the estimated effort to actual effort.
2. **Precision** in capturing the variability in software projects - the ability to accurately identify and measure the different factors that can change across various software development efforts.
3. **Complexity** of implementing the estimation method.
4. **Flexibility** to adapt to different types of projects and sizes.
5. **Cost-effectiveness** in terms of resources needed to implement and maintain the estimation process.
6. **Scalability** to handle projects of varying sizes and complexities.
7. **Usability** by project managers with varying levels of expertise.
8. **Integration** capability with existing project management tools and methodologies.

This table synthesizes the document's insights into each method's performance across the eight critical dimensions. It's designed to facilitate quick comparisons and informed decisions regarding the most suitable estimation techniques for different software development contexts. The values are colored by the Positive/Negative impact.

Method	**Accuracy**	**Precision**	**Complexity**	**Flexibility**	**Cost-effectiveness**	**Scalability**	**Usability**	**Integration**
Bayesian Network Model	High	High	Medium	High	High	Medium	High	High
Customized Checklists	Medium	Medium	Low	High	High	Low	High	Medium
Choquet Integral	High	High	High	Medium	Medium	Medium	Medium	Medium
Prediction-Based Technique	High	High	Medium	High	Medium	High	Medium	High
SE3M Model	High	High	High	Medium	Medium	High	Medium	Medium
Task Planning Model	Medium	Medium	Medium	High	Medium	Medium	Medium	High
"vpbench" Framework	High	High	High	Medium	Low	High	Low	High
Developer's Expertise-Based Estimation	High	Medium	Low	High	High	Medium	High	High
Ensemble Machine Learning Models	High	High	High	Medium	Medium	High	Medium	Medium
Differential Evolution (DE)	High	High	High	Low	Medium	Medium	Low	Low
Functional Stories and Issues	Medium	Medium	Medium	High	High	Medium	High	High
Multi-Layered Feed Forward ANN	High	High	High	Low	Low	High	Low	Medium

Productivity-Based UCP Models	High	Medium	Medium	High	Medium	Medium	Medium	High
Harmony Search (HS)	High	High	High	Low	Low	High	Low	Low
Satin Bowerbird Optimizer (SBO)	High	High	High	Low	Low	High	Low	Low
Automated Function Point (AFP) and ABCART	High	High	High	Medium	Medium	High	Medium	Medium
Optimal Bayesian Belief Network	High	High	High	Medium	Medium	High	Medium	Medium
Consistent Fuzzy Analogy-Based Estimation (C-FASEE)	High	High	Medium	High	High	Medium	High	High
Hybrid Model from Use Case Points	High	High	Medium	High	Medium	High	Medium	High

Discussion:

Using the derived information, let's review the influence of projects types and write a suggested practical guide for a project manager.

Different project types influence on selecting the most effective method

1. **Agile vs. Traditional Projects:** Agile methodologies favor SEE methods that are iterative and flexible, such as the Bayesian Network Model, which can incorporate expert judgment and empirical data effectively. In contrast, traditional projects might benefit more from structured approaches like the Constructive Cost Model (COCOMO) or Function Point Analysis, which provide a systematic way to estimate efforts based on project size and complexity.

2. **Project Complexity:** The complexity of a project significantly influences the choice of SEE method. For complex projects, methods that consider multiple variables and their interdependencies, such as Ensemble Machine Learning Models or Bayesian Networks, are preferable. Simpler projects might not require such sophisticated approaches, and expert judgment or analogy-based methods could suffice.

3. **Data Availability:** The availability and quality of data are critical factors in selecting an SEE method. Projects with extensive historical data can leverage machine learning models or data-driven

approaches like the SE3M Model, which utilizes semantic analysis. For projects with limited data, methods that rely more on expert opinion or simpler statistical models may be more appropriate.

4. **Team Expertise:** The expertise and experience of the project team play a crucial role in selecting an SEE method. Teams with strong analytical skills might lean towards computational or machine learning methods, while teams with less experience in data analysis might prefer expert judgment or analogy-based methods.

5. **Organizational Context:** The broader organizational context, including the industry sector, organizational maturity in project management, and the specific goals of the project, also influences the choice of SEE method. For instance, in highly innovative sectors, methods that allow for rapid adjustments and incorporate new information, like agile-based SEE methods, might be preferred.

6. **Integration with Existing Tools:** The ability to integrate the SEE method with existing project management and development tools is another critical consideration. Methods that offer seamless integration with tools like JIRA or Trello can enhance the efficiency of the estimation process and ensure consistency in project management practices.

Practical guidelines for a project manager to utilize software effort estimation (SEE) methods effectively

1. **Understand the Project's Context:** Before selecting an SEE method, thoroughly understand the project's requirements, complexity, and available data. This understanding is crucial for choosing the most suitable estimation technique.

2. **Evaluate Data Availability:** Assess the amount and quality of historical data and current project data. For projects with rich historical data, machine learning models or advanced statistical methods may be more appropriate. In contrast, for projects with limited data, simpler methods like expert judgment or analogies might be more effective.

3. **Select the Appropriate SEE Method:** Choose an SEE method that aligns with the project's characteristics and the organization's capabilities. For example, if the project is agile and data-driven, Bayesian Network Models or Machine Learning Models could be ideal. Conversely, for more traditional projects or when detailed data is not available, methods like Expert Judgment or Analogies may be preferable.

4. **Customize the SEE Method:** Tailor the chosen SEE method to the specific needs of the project. This may involve adjusting parameters, integrating domain-specific knowledge, or combining different methods to improve accuracy.

5. **Use Tools to Support SEE:** Leverage software tools that can automate and facilitate the SEE process. These tools can help in data collection, analysis, and visualization, making the estimation process more efficient and reliable.

6. **Iteratively Refine Estimates:** Treat initial estimates as starting points. Refine these estimates as the project progresses and more information becomes available. This iterative approach helps in accommodating changes and refining accuracy.

7. **Incorporate Team Feedback:** Engage the project team in the estimation process. Their insights and feedback can provide valuable perspectives that enhance the accuracy of the estimates.

8. **Document and Review Estimates:** Maintain a record of estimation results, methodologies used, and the rationale behind the estimates. This documentation is valuable for future projects and can help in refining the estimation process over time.

9. **Continuously Learn and Adapt:** SEE is an evolving field. Stay updated with the latest research, tools, and techniques. Learning from past projects and adapting the estimation process accordingly can lead to continuous improvement in estimation accuracy.

Aligning estimation

The exploration and comparison of various software development task effort estimation methods, reveal a sophisticated landscape where different approaches offer unique advantages and challenges. The discussion around these methods underscores the criticality of aligning estimation techniques with specific project requirements, organizational contexts, and available datasets. This alignment is essential for enhancing estimation accuracy, precision, and overall project management efficiency.

Implications of Method Diversity

The diversity in recommended methods, from Bayesian Network Models and Customized Checklists to advanced algorithms like Differential Evolution and Satin Bowerbird Optimizer, illustrates the multifaceted nature of effort estimation. Each method's distinct analysis parameters, such as accuracy, precision, complexity, and scalability, highlight a critical insight: there is no one-size-fits-all solution. The implication here is profound for project managers and development teams. It suggests that the selection of an effort estimation method must be a deliberate decision, informed by an understanding of the project's specific characteristics, the team's expertise, and the organizational goals.

Relevance of Method Attributes

The relevance of each method's attributes cannot be overstated. Factors such as cost-effectiveness, usability, and integration capability with existing software development processes are pivotal. For instance, methods like the Bayesian Network Model and the SE3M Model, which offer high accuracy and precision with varying degrees of complexity and cost-effectiveness, underscore the trade-offs that organizations must consider. In environments where flexibility and adaptability are crucial, methods offering high scalability and integration, such as Customized Checklists and Developer's Expertise-Based Estimation, become invaluable.

Organizational Context and Data Availability

The organizational context and data availability is critical for the choice of effort estimation method. For organizations with rich historical data and advanced analytics capabilities, methods using machine learning models or semantic analysis, like the Ensemble Machine Learning Models or the SE3M Model, offer a promising direction. On the other hand, for organizations at an earlier stage of data maturity or with limited analytics capabilities, simpler, more intuitive methods like Customized

Checklists or Task Planning Models may provide a more practical and immediately beneficial approach.

Future Directions and Integration Challenges

Looking ahead, the continuous evolution of software development methodologies and the increasing complexity of projects suggest that effort estimation methods will need to become even more sophisticated and adaptable. The integration of these methods into agile and traditional project management frameworks remains a challenge but also an ***opportunity for innovation***. Tools and platforms that can seamlessly incorporate various estimation methods, allowing project managers to switch between them based on project phase or changing requirements, could significantly enhance project planning and execution efficiency.

Conclusions

In conclusion, the subtle understanding of each method's strengths and limitations, offers a valuable resource for project managers and software development teams. By carefully considering the project-specific, organizational, and contextual factors, practitioners can select and tailor effort estimation methods that best fit their needs, ultimately leading to more accurate predictions, efficient resource allocation, and successful project outcomes. The discussion underscores the importance of ongoing research, methodological innovation, and the development of tools that support the diverse and dynamic nature of software development effort estimation.

In order to implement the selected methods, it requires quite a bit of effort from the project manager and the development teams, we suggest an innovation that the usage of these models and methods would occur using a designated tool, enriched with AI capabilities, incorporated (plugin) into existing management tools such as Jira, Project, Azure DevOps, etc.

To fully use the recommendations aggregated in this document, the designated tool would, for example, perform the following automated steps:

1. Query for adequate details regarding the requested software task, a questionnaire.
2. Hold indexed data of software development tasks collected from relevant datasets like GitHub.
3. Collect information taken from the current company's source control and task management tools.
4. Collect development teams members qualifications data.
5. Read documents relevant for the current project being developed.
6. Have a connection to online resources and tools.
7. Implement multiple methodologies presented in this document according to analyzed data.
8. The tool's analysis should reply with an educated information:
    a. A breakdown of the requested task.
    b. A document suggesting development phases and relevant generated code samples sections for each sub-task.
    c. An estimated effort for each sub-task.
    d. List of concerns for the required task, such as security concerns, required hardware, licenses, regulation to consider, integration efforts, etc.
    e. An overall effort estimation for this task, that can be formed as a range depending on all relevant information.
    f. The generated data will be gathered as an instructions document.

This document would eventually be of use for both the project manager in SEE, and will eventually direct and focus the development team in the actual development and as type of phase-maker or suggested technical guide.

We have the confidence that utilizing such a tool by project managers and development teams, would present a significant visible ROI.

References

1. Dragicevic, S., Celar, S. and Turic, M., 2017. Bayesian network model for task effort estimation in agile software development. *Journal of systems and software*, *127*, pp.109-119.
2. Usman, M., Petersen, K., Börstler, J. and Neto, P.S., 2018. Developing and using checklists to improve software effort estimation: A multi-case study. *Journal of Systems and Software*, *146*, pp.286-309.
3. Butt, S.A., Ercan, T., Binsawad, M., Ariza-Colpas, P.P., Diaz-Martinez, J., Pineres-Espitia, G., De-La-Hoz-Franco, E., Melo, M.A.P., Ortega, R.M. and De-La-Hoz-Hernandez, J.D., 2023. Prediction based cost estimation technique in agile development. *Advances in Engineering Software*, *175*, p.103329.
4. Fávero, E.M.D.B., Casanova, D. and Pimentel, A.R., 2022. SE3M: A model for software effort estimation using pre-trained embedding models. *Information and Software Technology*, *147*, p.106886.
5. Wysocki, W., 2023. Task Planning Model of Software Process. *Procedia Computer Science*, *225*, pp.736-745.
6. Derks, C., Strüber, D. and Berger, T., 2023. A benchmark generator framework for evolving variant-rich software. *Journal of Systems and Software*, *203*, p.111736.
7. Butt, S.A., Khalid, A., Ercan, T., Ariza-Colpas, P.P., Melisa, A.C., Pineres-Espitia, G., De-La-Hoz-Franco, E., Melo, M.A.P. and Ortega, R.M., 2022. A software-based cost estimation technique in scrum using a developer's expertise. *Advances in Engineering Software*, *171*, p.103159.
8. Kumar, P.S., Behera, H.S., Kumari, A., Nayak, J. and Naik, B., 2020. Advancement from neural networks to deep learning in software effort estimation: Perspective of two decades. *Computer Science Review*, *38*, p.100288.
9. Pospieszny, P., Czarnacka-Chrobot, B. and Kobylinski, A., 2018. An effective approach for software project effort and duration estimation with machine learning algorithms. *Journal of Systems and Software*, *137*, pp.184-196.
10. Benala, T.R. and Mall, R., 2018. DABE: Differential evolution in analogy-based software development effort estimation. *Swarm and Evolutionary Computation*, *38*, pp.158-172.
11. Rosa, W. and Jardine, S., 2023. Data-driven agile software cost estimation models for DHS and DoD. *Journal of Systems and Software*, *203*, p.111739.
12. Rijwani, P. and Jain, S., 2016. Enhanced software effort estimation using multi layered feed forward artificial neural network technique. *Procedia Computer Science*, *89*, pp.307-312.
13. Azzeh, M., Nassif, A.B., Elsheikh, Y. and Angelis, L., 2022. On the value of project productivity for early effort estimation. *Science of Computer Programming*, *219*, p.102819.
14. Jørgensen, M. and Halkjelsvik, T., 2020. Sequence effects in the estimation of software development effort. *Journal of Systems and Software*, *159*, p.110448.
15. Singh, T., Singh, R. and Mishra, K.K., 2018. Software cost estimation using environmental adaptation method. *Procedia computer science*, *143*, pp.325-332.
16. Azzeh, M., Nassif, A.B. and Minku, L.L., 2015. An empirical evaluation of ensemble adjustment methods for analogy-based effort estimation. *Journal of Systems and Software*, *103*, pp.36-52.
17. Di Martino, S., Ferrucci, F., Gravino, C. and Sarro, F., 2020. Assessing the effectiveness of approximate functional sizing approaches for effort estimation. *Information and Software Technology*, *123*, p.106308.
18. Abrahão, S., De Marco, L., Ferrucci, F., Gomez, J., Gravino, C. and Sarro, F., 2018. Definition and evaluation of a COSMIC measurement procedure for sizing Web applications in a model-driven development environment. *Information and Software technology*, *104*, pp.144-161.
19. Usman, M., Britto, R., Damm, L.O. and Börstler, J., 2018. Effort estimation in large-scale software development: An industrial case study. *Information and Software technology*, *99*, pp.21-40.
20. Ali, A. and Gravino, C., 2021. Improving software effort estimation using bio-inspired algorithms to select relevant features: An empirical study. *Science of Computer Programming*, *205*, p.102621.
21. Moosavi, S.H.S. and Bardsiri, V.K., 2017. Satin bowerbird optimizer: A new optimization algorithm to optimize ANFIS for software development effort estimation. *Engineering Applications of Artificial Intelligence*, *60*, pp.1-15.
22. Qi, F., Jing, X.Y., Zhu, X., Xie, X., Xu, B. and Ying, S., 2017. Software effort estimation based on open source projects: Case study of Github. *Information and Software Technology*, *92*, pp.145-157.
23. Zare, F., Zare, H.K. and Fallahnezhad, M.S., 2016. Software effort estimation based on the optimal Bayesian belief network. *Applied Soft Computing*, *49*, pp.968-980.

24. Ezghari, S. and Zahi, A., 2018. Uncertainty management in software effort estimation using a consistent fuzzy analogy-based method. *Applied Soft Computing*, *67*, pp.540-557.
25. Kirmani, M.M. and Wahid, A., 2015. Use case point method of software effort estimation: a review. *International Journal of Computer Applications*, *116*(15).
26. Azzeh, M. and Nassif, A.B., 2016. A hybrid model for estimating software project effort from Use Case Points. *Applied Soft Computing*, *49*, pp.981-989.
27. Mensah, S., Keung, J., Bosu, M.F. and Bennin, K.E., 2018. Duplex output software effort estimation model with self-guided interpretation. *Information and Software Technology*, *94*, pp.1-13.
28. Govil, N. and Sharma, A., 2022. Estimation of cost and development effort in Scrum-based software projects considering dimensional success factors. *Advances in Engineering Software*, *172*, p.103209.
29. Singal, P., Kumari, A.C. and Sharma, P., 2020. Estimation of software development effort: A Differential Evolution Approach. *Procedia Computer Science*, *167*, pp.2643-2652.
30. Idri, A., Hosni, M. and Abran, A., 2016. Improved estimation of software development effort using classical and fuzzy analogy ensembles. *Applied Soft Computing*, *49*, pp.990-1019.
31. Alqasrawi, Y., Azzeh, M. and Elsheikh, Y., 2022. Locally weighted regression with different kernel smoothers for software effort estimation. *Science of Computer Programming*, *214*, p.102744.
32. Idri, A., Abnane, I. and Abran, A., 2016. Missing data techniques in analogy-based software development effort estimation. *Journal of Systems and Software*, *117*, pp.595-611.
33. Phannachitta, P., 2020. On an optimal analogy-based software effort estimation. *Information and Software Technology*, *125*, p.106330.
34. García-Floriano, A., López-Martín, C., Yáñez-Márquez, C. and Abran, A., 2018. Support vector regression for predicting software enhancement effort. *Information and Software Technology*, *97*, pp.99-109.
35. González-Ladrón-de-Guevara, F., Fernández-Diego, M. and Lokan, C., 2016. The usage of ISBSG data fields in software effort estimation: A systematic mapping study. *Journal of Systems and Software*, *113*, pp.188-215.
36. Alatawi, M.N., Alyahyan, S., Hussain, S., Alshammari, A., Aldaeej, A.A., Alali, I.K. and Alwageed, H.S., 2023. A Data-Driven Artificial Neural Network Approach to Software Project Risk Assessment. *IET Software*, *2023*.
37. Saqlain, M., Abid, M., Awais, M. and Stević, Ž., 2023. Analysis of Software Effort Estimation by Machine Learning Techniques. *Ingénierie des Systèmes d'Information*, *28*(6).
38. Sousa, A.O., Veloso, D.T., Gonçalves, H.M., Faria, J.P., Mendes-Moreira, J., Graça, R., Gomes, D., Castro, R.N. and Henriques, P.C., 2023. Applying Machine Learning to Estimate the Effort and Duration of Individual Tasks in Software Projects. *IEEE Access*.
39. Wysocki, W., Miciuła, I. and Mastalerz, M., 2022. Classification of Task Types in Software Development Projects. *Electronics*, *11*(22), p.3827.
40. Latif, A., Fitriana, L.A. and Firdaus, M.R., 2021. Comparative analysis of software effort estimation using data mining technique and feature selection. *JITK (Jurnal Ilmu Pengetahuan dan Teknologi Komputer)*, *6*(2), pp.167-174.
41. Alsheikh, N.M. and Munassar, N.M., 2023. Improving Software Effort Estimation Models Using Grey Wolf Optimization Algorithm. *IEEE Access*.
42. Thant, Khin & Khaung Tin, Hlaing Htake. (2023). Learning The Efficient Estimation Techniques for Successful Software Project Management. 11. 4-8. 10.22159/ijet.2023v11i3.47605.
43. Azzeh, M., Nassif, A.B. and Attili, I.B., 2021. Predicting software effort from use case points: A systematic review. *Science of Computer Programming*, *204*, p.102596.
44. Morgenshtern, O., Raz, T., & Dvir, D. (2007). Factors affecting duration and effort estimation errors in software development projects. *Information and Software Technology*, *49*(8), 827-837.
45. Bonetti, A., Bortot, S., Fedrizzi, M., Pereira, R. M., & Molinari, A. (2012). Modeling group processes and effort estimation in project management using the Choquet integral: An MCDM approach. *Expert Systems with Applications*, *39*(18), 13366-13375.
46. Md. Tanziar Rahman, Md. Motaharul Islam, and Ummay Salma Shorna, “A Comparative Analysis of Regression Models for Software Effort Estimation”, IJITC, vol. 3, no. 06, pp. 26–42, Oct. 2023.
47. Timothy M. Persons, Ph.D. (2020). COST ESTIMATING AND ASSESSMENT GUIDE - Best Practices for Developing and Managing Program Cost. White paper by: GAU, U.S. Government Accountability Office.