A Model for Non-Intrusive Capture of Metrics for Early Project Estimation in Agile Environments

José Gamaliel Rivera Ibarra[1]

[1]Instituto Tecnológico de Sonora
Ciudad Obregón, Sonora, México

Abstract. *Early estimation in agile projects is essential for determining feasibility, costs, and planning. However, lacking detailed initial requirements introduces uncertainty and leads to inconsistent results. Although the literature proposes predictive models based on historical data to improve accuracy, the lack of databases tailored to each company's context prevents practitioners from applying these models effectively. This work presents a non-intrusive model for capturing key metrics (such as software size) in agile environments to facilitate the creation of historical databases. The model will be calibrated and validated through empirical studies to support the gradual adoption of a hybrid estimation approach that combines data-driven techniques with expert judgment. Preliminary results indicate that the scarcity of historical data in practice reduces estimation accuracy, highlighting the need for automated metric capture. This approach aims to enhance the reliability of early estimations in agile projects and provide objective support for decision-making in software project management.*

1. Introduction

Early estimation in agile development projects is fundamental for assessing feasibility, defining budgets, planning, and establishing a strategic foundation for decision-making during project execution. During the initiation and planning phases, such estimation provides stakeholders (investors, technical teams, and clients) with a clear view of feasibility, enabling them to anticipate risks, allocate resources efficiently, and align expectations from the outset [McConnell 2006].

Nonetheless, projects in early stages often yield imprecise estimates. The Cone of Uncertainty [Barry et al. 1981] stated that estimating based on vague and ambiguous requirements (characteristic of early stages and agile projects) can result in errors ranging between 60% and 160% of the actual final value [Cohn 2005], which may lead teams to fall short of their objectives [Prakash and Viswanathan 2017].

Researchers and practitioners have worked for decades to develop methods that yield more accurate estimates [Jadhav et al. 2022]. These methods primarily aim to determine the effort required for software development – that is, the amount of work necessary to complete a task satisfactorily [Sudarmaningtyas and Mohamed 2021]. Estimation approaches are broadly classified into three groups: expert judgment–based methods, which rely on the opinions of experienced professionals; data-driven (or quantitative) methods, which build predictive models from historical data to forecast effort; and hybrid methods, which combine data objectivity with professional expertise to enhance accuracy in early-stage estimation [Vera et al. 2018].

Agile development teams have predominantly relied on expert judgment for initial estimations. Although this approach allows for agility and flexibility, it generally produces inconsistent results due to the subjectivity and variability in interpreting requirement complexity [Zarour and Zein 2019]. In this context, using historical data provides an objective basis for enhancing estimation accuracy [Alsaadi and Saeedi 2022].

Researchers in agile project estimation advocate data-driven methods [Rivera et al. 2024]. They primarily employ regression analysis and machine learning techniques [Hameed et al. 2023, Rosa and Jardine 2023] to build predictive models that use software size as the primary independent variable or effort predictor [Ünlü et al. 2022]. However, these models require a large amount of historical data [Sudarmaningtyas and Mohamed 2021], forcing reliance on public sources that present various limitations:

- *Obsolete Records*: Many databases, such as ISBSG, NASA93, COCOMO81, or MAXWELL, contain old project records (from 1970 to 1999) [Beata et al. 2015], hindering their application in today's technological context.
- *Lack of Organizational Context*: These databases aggregate projects from various companies without including information about specific processes, tools, technologies, and methodologies, which limits model comparability and adaptability [Usman et al. 2014].
- *Non-Standardized Metrics*: Using subjective metrics, such as Story Points, prevents objective comparisons between projects [Hacaloğlu and Demirörs 2018].

Function Points (FP) provide a standardized metric for objectively measuring the functional size of software in completed projects [Hussain et al. 2013]. This measurement allows for calculating other relevant metrics, such as team velocity and defect density, and is fundamental for constructing historical databases that serve as input for predictive models [Hussain et al. 2013, Commeyne et al. 2016]. However, functional size measurement (FSM) techniques require considerable analytical effort [Sahab and Trudel 2020], which is often impractical in agile environments [Soubra et al. 2020], where teams prioritize activities that directly add value to the customer. Although some authors [Sahab and Trudel 2020, Özgesu Özen et al. 2020, Tenekeci et al. 2024] have proposed tools to perform FSM from source code, these have not yet achieved widespread adoption.

The lack of automated and nonintrusive mechanisms to capture metrics from ongoing and past (postmortem) projects poses an obstacle to creating the databases that would allow agile teams to adopt a hybrid estimation approach— that combines quantitative methods with expert judgment to improve early-stage estimation accuracy [Yogi and Chinthala 2013].

This work employs a Design Science Research framework to design and validate a non-intrusive model for capturing key metrics, such as functional size and team productivity, through an automated tool that facilitates the creation of historical databases from recent projects within the same company. Our approach implements an iterative process that involves artifact design, demonstration, and evaluation, which enables us to rigorously develop the model and seamlessly integrate it into existing agile workflows. Although our long-term vision is for this data to serve as input for a hybrid estimation approach, the scope of this work is limited to the design, development, and validation of the non-intrusive capture model.

The remainder of this article is organized as follows: Section 2 presents the research questions and objectives, clarifying the focus and expected contributions of our work. Section 3 details the research methodology, including the phases and activities undertaken to design, develop, and validate the proposed model. Section 4 outlines the preliminary results obtained so far, highlighting both the insights from the literature and the initial diagnostic of professional practice. In Section 5, we compare our approach with existing solutions, discussing key differences and potential advantages. Finally, Section 6 addresses the challenges that remain and outlines the next steps required to refine and fully implement our proposal.

2. Research Questions and Objectives

This research seeks to answer the following key questions:

- How can agile teams non-intrusively capture key project metrics using existing tools (Git, Jira, CI/CD) to build contextualized historical databases?
- What technical and methodological criteria enable the integration of non-intrusive metric capture with existing agile tools (e.g., Git, Jira, CI/CD)?
- How effective and applicable is the proposed non-intrusive model for automatically capturing metrics in agile environments, as evaluated through empirical studies in real-world settings?

The general objective is to develop and validate a non-intrusive model for capturing key agile project metrics that integrates with existing tools, enabling the creation of contextualized historical databases to support future estimation practices. The specific objectives include:

- Identify key metrics for early agile estimation and define standardized capture methods aligned with the Goal-Question-Metric (GQM) framework [Misra et al. 2005].
- Design a conceptual model for non-intrusive metric capture, including technical specifications for tool integration (APIs, automation scripts).
- Implement a functional prototype that automates metric extraction from agile tools (e.g., Jira, Git) and stores data with contextual metadata.
- Evaluate the model's accuracy and applicability in real-world agile environments.

3. Research Methodology

This research adopts a Design Science Research (DSR) framework [Alan et al. 2020], combined with a mixed-methods approach (qualitative and quantitative), and is structured into three main phases as illustrated in Figure 1.

Phase 1: Problem Identification and Solution Objectives

In this phase, we identify the core problem – the lack of reliable, context-specific historical databases in agile environments – and define the objectives of our solution. A systematic mapping study (SMS) is conducted following Kitchenham's methodology [Kitchenham and Charters 2007] to review current estimation techniques and their limitations. Concurrently, we diagnose professional practice through surveys and semi-structured interviews with agile practitioners to determine the key challenges in early estimation and data capture.

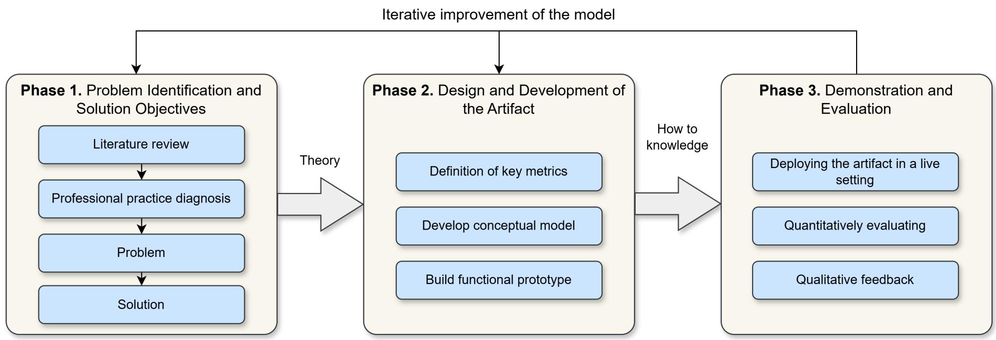

Figure 1. Research methodology based on the DSR framework [Alan et al. 2020]

Phase 2: Design and Development of the Artifact
Based on the findings from Phase 1, we design the conceptual model for non-intrusively capturing key metrics (e.g., functional size, team productivity). This phase involves:

- Defining key metrics using the GQM framework to align with estimation objectives.
- Developing the theoretical framework and data flow processes for constructing context-specific historical databases.
- Building a functional prototype that integrates with agile tools (Jira, Git, CI/CD, SonarQube) via APIs and automation scripts to capture metrics non-intrusively. For example: (a) Extracting task progress from Jira (e.g., Sprint Burndown), (b) deriving code activity from Git commits, (c) logging build/deployment metrics from CI/CD pipelines; and (d) parsing code quality data from SonarQube. This ensures that metric capture leverages existing workflows without manual intervention or disruption.

Phase 3: Demonstration and Evaluation
We validate the proposed model through empirical experiments in a real-world agile development environment in this phase. Key activities include:

- Deploying the artifact in a live setting to capture metrics from current and post-mortem projects.
- Quantitatively evaluate the accuracy of the captured metrics against manual measurements using performance indicators (e.g., MAE).
- Collecting qualitative feedback from agile teams regarding the model's non-intrusive integration and practical utility.
- Incorporating a feedback loop for iterative improvement of the model.

This DSR-based methodology ensures that the proposed model is rigorously designed, developed, and validated, addressing the key challenges of early estimation in agile environments.

4. Preliminary Results

Preliminary findings from the SMS [Rivera et al. 2024] and the professional practice diagnosis have revealed the following:

- Most early estimation methods proposed in the literature use data-driven approaches, primarily based on regression analysis and machine learning techniques [Hameed et al. 2023, Rosa and Jardine 2023].
- There is a marked dependence on expert judgment in agile practice, which results in inconsistent estimations.
- The lack of historical databases limits the adoption of quantitative methods, as automated mechanisms for non-intrusively capturing metrics are absent.

In response to these limitations, this work proposes the development of an automated tool that:

- Non-intrusively captures key metrics, integrating with agile environments' software development and project management tools.
- Enables the construction of a contextualized historical database that is standardized (e.g., using Function Points) [Commeyne et al. 2016].
- Facilitates, in future stages, the integration of hybrid estimation models that combine objective data with expert judgment, allowing for continuous feedback and improvement of the estimation process.

With this proposal, our vision is to build a hybrid early estimation model for agile projects based on the following pillars (however, this work focuses on a non-intrusive metric capture model):

1. *Hybrid Approach*: Since no single estimation technique has proven superior in all contexts, combining expert judgment with data-driven methods offers greater accuracy and adaptability [Usman et al. 2014]. This approach allows professionals to select, combine, and prioritize different estimation methods according to the project's context and needs.
2. *Nonintrusive Tools*: Metric capture should not interrupt the workflow of agile teams. Therefore, we propose automated tools that extract key metrics (e.g., functional size) from current and past projects. These tools integrate with CI/CD, Git, SonarQube, and Jira, minimizing manual intervention.
3. *Database*: Results from previous projects support estimations. Thus, the model aggregates data from dispersed sources (Jira, Git, CI/CD) into a unified repository, creating a historical database that includes recent projects from the company, with standardized metrics (e.g., Function Points for size, velocity for productivity) and contextual metadata (team configuration, technologies, agile practices). This structure enables objective comparison while preserving project-specific context.
4. *Feedback*: The estimation model must be continuously adjusted and improved based on metrics obtained during project execution. Nonintrusive metric capture provides real data that allows for updating and refining predictive models, while experts can identify new predictors from the feedback, generating a framework for continuous improvement.

5. Differences with Previous Approaches

Various studies have developed tools to measure the functional size of software from source code [Sahab and Trudel 2020, Özgesu Özen et al. 2020, Tenekeci et al. 2024]; however, these solutions have not addressed two critical aspects:

- *Nonintrusive Integration*: Existing tools have not investigated how to capture metrics without disrupting the workflow of agile teams.
- *Construction of Historical Databases*: These tools' primary focus has been on calculating functional size without structuring and storing the data in a way that can be used to model future estimations.

Our proposal differs by prioritizing the automated and nonintrusive capture of key metrics, enabling the construction of a historical database that serves as input for the progressive application of hybrid estimation models. These models combine data-driven methods—especially those based on machine learning—with expert judgment. The following table compares approaches based on expert judgment, data-driven methods, and the hybrid approach (the basis of our proposal).

Table 1. Comparison between estimation approaches for early stages of agile projects

Feature	Expert-based	Data-driven	Hybrid (proposal)
Source of data and information	Subjective experience of experts	Databases of previous projects	Combine both sources
Precision	Variable, depends on the experience of the experts	Variable, high when there is enough historical data	It is constantly improving with data from new projects
Applicability in agile environments	High, but prone to bias and human error	Low, lack of historical data is an obstacle	High, non-intrusive tool builds database
Adaptability	Depends on experts to adjust to new projects	Depends on updating the models with new data	Dynamic, data is automatically captured and fed back to the models

6. Challenges and Next Steps

Although the proposal presents a promising approach, we have identified several challenges:

- *Integration of Estimation Approaches*: An important challenge is to establish a set of criteria that enables the coherent combination of different estimation methods, thereby implementing a hybrid approach adaptable to various scenarios.
- *Incorporation of advanced methods*: To apply machine learning-based techniques, collaboration with experts and the availability of specialized tools are required, which might not be present in current teams and environments.

The following steps in this research are:

- *Model Design*: Determine the key metrics for early-stage estimation, specify the processes and data flow required to build historical databases, and detail criteria for non-intrusive integration.
- *Model Extension*: Investigate the possibility of expanding the model to integrate additional sources of information, such as physiological or interaction metrics (e.g., messaging and email applications), which could enhance future predictive estimation models.

- *Implementation and Calibration*: Develop a functional prototype and use post-mortem projects to verify its ability to automatically extract metrics, comparing these results with manual measurements performed by experts.
- *Experimental Evaluation*: Design and execute an experiment in an agile development environment to assess the model's effectiveness in obtaining metrics and collect team feedback on its utility and applicability.

References

Alan, vom Brocke Jan, M. A., and Hevner (2020). *Introduction to Design Science Research*, pages 1–13. Springer International Publishing.

Alsaadi, B. and Saeedi, K. (2022). Data-driven effort estimation techniques of agile user stories: a systematic literature review. *Artificial Intelligence Review*, 55:5485–5516.

Barry, B. et al. (1981). Software engineering economics. *New York*, 197:40.

Beata, Przemysław, K. A. P., and Czarnacka-Chrobot (2015). Application of function points and data mining techniques for software estimation - a combined approach. In Beata, Świerczek Jaroslaw Kobyliński Andrzej, and Czarnacka-Chrobot, editors, *Software Measurement*, pages 96–113. Springer International Publishing.

Cohn, M. (2005). *Agile Estimating and Planning*. Prentice Hall, Upper Saddle River, NJ, USA.

Commeyne, C., Abran, A., and Djouab, R. (2016). Effort estimation with story points and cosmic function points - an industry case study. In *Software Measurement News*.

Hacaloğlu, T. and Demirörs, O. (2018). Challenges of using software size in agile software development: A systematic literature review. *Academic Papers at IWSM Mensura 2018*.

Hameed, S., Elsheikh, Y., and Azzeh, M. (2023). An optimized case-based software project effort estimation using genetic algorithm. *Information and Software Technology*, 153:107088.

Hussain, I., Kosseim, L., and Ormandjieva, O. (2013). Approximation of cosmic functional size to support early effort estimation in agile. *Data & Knowledge Engineering*, 85:2–14. Natural Language for Information Systems: Communicating with Anything, Anywhere in Natural Language.

Jadhav, A., Kaur, M., and Akter, F. (2022). Evolution of software development effort and cost estimation techniques: Five decades study using automated text mining approach. *Mathematical Problems in Engineering*, 2022:5782587.

Kitchenham, B. and Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering. Technical report, Technical report, EBSE Technical Report EBSE-2007-01.

McConnell, S. (2006). *Software Estimation: Demystifying the Black Art*. Microsoft Press, USA.

Misra, S., Kumar, V., and Kumar, U. (2005). Goal-driven measurement framework for software innovation processes. volume 2, pages 710–716.

Prakash, B. and Viswanathan, V. (2017). A survey on software estimation techniques in traditional and agile development models. *Indonesian Journal of Electrical Engineering and Computer Science*, 7:867–876.

Rivera, J. G., Borrego, G., and Palacio, R. R. (2024). Early estimation in agile software development projects: A systematic mapping study. *Informatics*, 11.

Rosa, W. and Jardine, S. (2023). Data-driven agile software cost estimation models for dhs and dod. *Journal of Systems and Software*, 203:111739.

Sahab, A. and Trudel, S. (2020). Cosmic functional size automation of java web applications using the spring mvc framework. In *IWSM-Mensura*.

Soubra, H., Abufrikha, Y., Abran, A., et al. (2020). Towards universal cosmic size measurement automation. In *Joint Proceedings of the 30th International Workshop on Software Measurementand the 15th International Conference on Software Process and ProductMeasurement (IWSM Mensura 2020), Mexico City, Mexico, October 29-30,2020*.

Sudarmaningtyas, P. and Mohamed, R. B. (2021). A review article on software effort estimation in agile methodology. *Pertanika Journal of Science & Technology*, 29.

Tenekeci, S., Ünlü, H., Dikenelli, E., Selçuk, U., Soylu, G. K., and Demirörs, O. (2024). Predicting software size and effort from code using natural language processing. In Trudel, S., Demirörs, O., Moulla, D. K., and Hacaloglu, T., editors, *Joint Proceedings of the 33rd International Workshop on Software Measurement and the 18th International Conference on Software Process and Product Measurement (IWSM-MENSURA 2024), Montréal, Canada, September 30 - October 4, 2024*, volume 3852. CEUR-WS.org.

Usman, M., Mendes, E., Weidt, F., and Britto, R. (2014). Effort estimation in agile software development: A systematic literature review. In *Proceedings of the 10th International Conference on Predictive Models in Software Engineering*, pages 82–91. Association for Computing Machinery.

Vera, T., Ochoa, S., and Perovich, D. (2018). Survey of software development effort estimation taxonomies.

Yogi, M. K. and Chinthala, V. (2013). A hybrid approach to sizing problem in software project estimation. *International Journal for Scientific Research and Development*, 1(1):29–34.

Zarour, A. and Zein, S. (2019). Software development estimation techniques in industrial contexts: An exploratory multiple case-study. *International Journal of Technology in Education and Science*, 3:72–84.

Özgesu Özen, Özsoy, B., Aktılav, B., Güleç, E. C., and Demirörs, O. (2020). Automated estimation of functional size from code. In *2020 Turkish National Software Engineering Symposium (UYMS)*, pages 1–7.

Ünlü, H., Hacaloglu, T., Büber, F., Berrak, K., Leblebici, O., and Demirörs, O. (2022). Utilization of three software size measures for effort estimation in agile world: A case study. In *2022 48th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pages 239–246.