Story Point Estimate Model: Project Development using Generative AI (GenAI) Tools

Ravi Kiran Mallidi
School of Computer Applications
Lovely Professional University
Punjab, India

Manmohan Sharma
School of Computer Applications
Lovely Professional University
Punjab, India

Yogeswara Prasad Paladugu
Associate Director
Cognizant Technologies Ltd.,
Hyderabad, India

ABSTRACT

Software development estimations by using Agile story points are crucial for predicting effort, cost, and scope. Currently, management is encouraged to explore the usage of AI tools for their practices to improve code productivity and debugging capabilities. In the past couple of years, Generative Artificial Intelligence (GenAI) tools have evolved and Organizations are using different tools in SDLC phases Architecture, Development, Testing, and Maintenance activities. GenAI usage in projects enhances productivity, eliminates bugs, and makes it more competitive for managing challenges. This paper discusses GenAI usage in software development and how Story Point estimates are derived, adjusted, and arrived at to complete the use case—conducted interviews with the development team where the Copilot and ChatGPT were used and identified the factors for considering the effort estimations. For the usage of the GenAI tools, the development cost may be reduced when compared with the traditional way of development which leads to a decrease in Agile Story Points. Presented the factors impacted while doing story point analysis for the usage of GenAI tools like ChatGPT, and Copilot. Coding assisting tools by GenAI provide code suggestions, tasks, code blocks, and standards. Educating project teams and shift of mindset and methodology required for usage of GenAI in project development and estimating the project.

Keywords

Effort Estimations, Agile, Gen AI Tools, Story Point, Software Development, Planning.

1. INTRODUCTION

GenAI is a subset of Artificial Intelligence for focuses on generating content creation, design and art, software development, language translation, health care, gaming, and finance. Due to the evolution of GenAI in the software market, Organizations start with a proof of concept (POC). Gartner predicted that by the end of 2025, 30% of GenAI projects will be stopped after POC. Gartner conducted a webinar pool with 2500 executives, 38% said customer experience is the primary focus, 26% indicates revenue growth, 17% indicates cost optimization, and 7% indicates business continuity for the adoption of GenAI in Organizations. The major challenges in adopting GenAI tools for development were substantial upfront costs for coding assistants, virtual assistants, and domain-level language models. Boston Consulting Group (BCG) predicted a 30% growth in upfront investment cost based on 330 IT leaders for the adoption of GenAI development tools for the next three years. GenAI tools like ChatGPT are trained on large amounts of data available in public domains. Most of the generated programs are not compliant with the General Data Protection Regulations (GDPR) and copyright. GenAI code generation for development projects has benefits and risks associated, below are some of the benefits:

- ✓ Tools like ChatGPT and Copilot accelerate the writing of code leading to a reduction in project turnover time.
- ✓ GenAI tools reduce the code review time by identifying the patterns related to bugs and vulnerabilities like cross-site scripting, cross-site request forgery, and redirect enablers.
- ✓ Streamlines the development processes and rapid development cycles.
- ✓ Scans the code base and predicts proactive bug identification.
- ✓ Automating for creating documentation, references, and guides from the code base.
- ✓ Time to market for faster releases.
- ✓ Coding similar patterns / Context-aware code suggestions, that improve code productivity within the project.
- ✓ Reduces the human-centric errors while coding and implementation.
- ✓ Multi-language support for generating code from GenAI tools.
- ✓ Optimizing resources based on data insights provided by the tool.
- ✓ Generating test scripts and automating the execution process.
- ✓ Bug deduction in the code repository and log the bug in with prioritization.
- ✓ Real-time data monitoring helps the project teams to resolve proactive issues resolutions.

Below are the challenges of using GenAI tools for project development.

- ✓ Adoption of GenAI tools in the projects takes time.
- ✓ Sometimes AI-generated code may vary in consistency and quality of code.
- ✓ Hidden issues may be caused by to lack of meticulousness of human expertise in code generation.
- ✓ The ability to reduce the coding skills of the developers.
- ✓ Developers spend time to validate the generated code for any vulnerabilities.
- ✓ Developers understand the limitations of GenAI tools for complex code generation and domain-specific requirements.
- ✓ Usage of code generation in applications may lead to vulnerabilities in adopting GDPR and copyright policies.
- ✓ The upfront investment of tooling costs, most of the vendors provide.

Estimating the development cost for GenAI tools utilizing is lesser in time for the developer to spend. In most of the scenarios, a 15 to 20% reduction in development time was observed while collecting data from the developer for analysis.

GenAI usage projects present unique challenges for estimations. Below are the Agile estimation techniques for GenAI usage projects

- ✓ Story mapping clearly defines where the application is using GenAI tools
- ✓ Adding some additional efforts for the initial stage of the project for understanding the tool by the development teams.
- ✓ Prepare pilot projects and implement
- ✓ Start with small component building and expand the activity with large component building.

This paper is organized as follows: Section II describes the Literature Review, Section III describes the Story Point estimation model for development projects using GenAI tools, Section IV describes case studies that utilize GenAI development tools and the time and cost saved by utilizing the AI for development, and Section V Conclusions and further work.

2. LITERATURE REVIEW

Agile has been used for the last twenty years for various projects in the IT industry. Software development models are changing day by day and adopting Generative AI tools for the development of projects.

Combining the results from the Delphi study along with GenAI and innovation management, established 10 themes and the themes can help to establish the management study [8]. Formulated 78 research questions in the software engineering area for using GenAI tools for development analyzed GenAI adoption in projects and concluded that GenAI brings significant changes in software engineering [10]. GenAI integration in requirement engineering provides significant opportunities. The conducted survey and results show that utilization of GenAI enhances the requirement engineering in the software life cycle with improved human AI models [4]. GenAI technology usage in low/no-code development enhances productivity and more accessible to non-technical users. Data modeling, design of user interface, code logic building, integration of external systems, testing, and maintenance are key stages for low-code development with agile principles [15]. Conducted a study in GenAI development within software engineering with critical pillars of People, Process, and Technology and proposed six essential actions for software engineering practice future to make strategic decisions by the management [3]. Replacing with GenAI tools project managers cannot empathize with the GenAI, Projects managers adopt GenAI results delivery value for Organizations, Employees, and Customers [13]. Addresses the behind characteristics of GenAI models with four families and opportunities for human-GenAI interactions [2]. Proposed four scenarios for the development of applications using generative AI and large-language models [14]. GenAI is capable of automating impractical applications, virtual assistance, and education services. GenAI is used by scholars and researchers in the Business & Information Systems Engineering industry [6]. The conducted study on ChatGPT addresses the pain area resolutions from the Agile development for training the various patterns resulting in Agile project management suggestions [12]. Proposed and used GitHub Copilot for development activities to increase software productivity with the measure of quality of code, defects, and developer satisfaction [16]. A systematic review conducted on agile estimations using data within in single Organization and multi-organizations shows that 29% of papers show accuracy metrics [5]. The research examines the usage of Artificial Nural Networks (ANN) for estimating development efforts thus giving accuracy in estimations for complex relationships projects. The research was carried out between the Case Base Reasoning model and the ANN model and showed better results by using the ANN model for effort estimations [7]. A systematic review was conducted for 17 papers and identified 10 key challenge barriers related to technology, organization, and environment for integrating AI in software planning [9]. The paper examines past and existing usage of Agile in suggests how Agile will transform in the future with an emphasis on the importance of Agile teaching in Organizations [1]. GenAI utilization in software architecture in the initial stages gives efficiency and observed challenges in software development. Proposed model for overcoming the challenges in the use of GenAI in software development resulting use of Retrieval Augmented Generative model [11].

3. STORY POINT ESTIMATION MODEL

Adoption of GenAI tools in an organization with proper planning and budgeting. A lot of tools are available in the market for generating the code. Below is the popular GenAI code generation tools list.

- ✓ ChatGPT – Code generation and explanation provided by the tools including C, C#, Go, PHP, Python, Java, and JavaScript. Seamless integration feature to connect to other systems.
- ✓ Copilot – Code generation for multiple languages Python, C, C#, Java, and JavaScript. Collaborated with different IDEs like Visual Studio, JetBrains, and Eclipse plugins.
- ✓ Alpha Code – Code generation tool with bug resolutions, suggestions, and optimizing code.
- ✓ Amazon Q – Generating code from small code snippets to the functional level. Supports for CLI and Q Developer from AWS.
- ✓ Google Gemini - Trained extensively for text and code generation with the support of programming languages C++, Go, Java, Python, and JavaScript.

Each tool has its respective components. The Organization has to buy the components according to their needs. Considered the factors for choosing the right tool as an important aspect of project success.

- ✓ Language support
- ✓ IDE integration support
- ✓ Features adoption and updates from LLM
- ✓ Security concerning copyright and licensing
- ✓ Pricing models

Team agility is calculated using the developer's velocity of the project. Boosting developer's velocity by using auto code generation tools, Low-code-No-code tools, and continuous integration and deployment. Nowadays GenAI tools are used for generating the code by providing a prompt by the developer. Based on the prompt the Large Language Model (LLM) generates code, and improvements in existing code, The GenAI assistance increases the velocity of developers by creating Rapid prototyping, converting concepts to functional code, generating documentation, and early bug deduction.

Exploring tools and AI-based technologies for adopting GenAI for development. AI-assist software development is a term used for the usage of AI and Machine Learning (ML) to enhance the SDLC process. In an SDLC process around 30-40% of time is spent on code development and unit testing activities. GenAI tools can help the developer community to decrease the effort spent on code development. For all SDLC stages, the GenAI plays a key role in Analysis, Design, Development, and Testing. Organizations start small and increase the capabilities of GenAI in their projects. Developers have to provide craft and clear prompts to the GenAI LLM engines for code generation. If the prompt is not proper the results are not up to the mark. The Developers have to spend more time fixing bugs / recreating the program if the prompt is not accurate. Sometimes AI-generated code may have bugs and errors due to the code being from the internet. Invert time to validate the correct vulnerabilities generated by the AI assistance. Not all the development teams are achieving the same productivity gain by using GenAI. Benefits depend on the maturity of the code, Use Case, technology stack used and experience with GenAI usage.

While using GenAI tools the developers have to rethink while providing the story points. The development effort may reduce drastically if the developer provides proper prompts, and other hand the development efforts might increase due to inappropriate prompts causing different results. Several experiments were conducted in different projects and observed that around 15-20% of savings for quality code generation, the case studies were described the Section 4. For this, the developers have to validate the efforts before proceeding to the scrum call. In this paper, we are presenting the below factors for determining the story point estimations while using GenAI for development and testing.

- ✓ How much code is generated – Entire function / complete application / prompt specific.
- ✓ How much bug detection the tool is capable
- ✓ How much documentation the tool is generating
- ✓ Is properly generating unit test cases
- ✓ Expertise level of developer

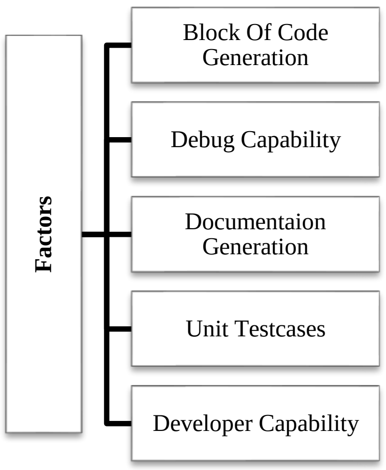

Fig 1: Factors Identification

Fig 1 shows the factors for story point estimation calculation. These factors are critical for estimation adjustment for the existing story point analysis. Each developer estimates the story points based on his experience and adds adjusted factors for the utilization of the GenAI tool for development activity. For each factor defined in Fig 1, the developer has to assess how much reduction/increase is required for the existing story point. The factor is in between 0-1 scale for calculation.

$$\text{Adjusted Factor } (\mathbf{AF}) = \sum_{i=1}^{n} (\text{Pi (Factors)} - 1)$$

For i = 1, 2…. n (n represents the total number of Factors defined in the project) and Pi represents each factor's value. In this context, the number "1" is often used as a baseline or neutral multiplier in effort estimation.

After deriving the factor analysis, the Story Point (SP) was calculated as below with the base of existing story points.

$$\mathbf{SP} = \boldsymbol{Estimated\ Story\ Points} * \boldsymbol{AF}$$

Return on Investment (ROI) has to be calculated while using the GenAI tool for development concerning existing costs spent. Organizations spend upfront costs for buying GenAI tools for use from the vendor. Due to the higher cost of the GenAI tool, the project team has to make proper decisions while choosing tool licenses based on resource usage/consumption-based usage. Total Cost of Ownership is calculated based on Initial cost, Ongoing cost, and hardware cost. Initial cost includes research, training, and development, and Ongoing cost. Ongoing cost includes additional plugin cost, LLM training costs, maintenance of product, and recurring subscription costs.

4. CASE STUDIES AND OBSERVATIONS

This section observed three project scenarios where the GenAI tools are implemented for the development phase. The metrics are not provided in this paper due to the customer's agreement not to publish the project data. The effort estimations are logged into the JIRA board and the project scrum master collects the story points estimates without using the GenAI tools in development and story points are estimated by using GenAI development tools used and compared. The final results are published in this by the percentage of benefit achieved by using GenAI development tools in the project. Below are the high-level details

Case Study 1*:*

Copilot is used to generate the Selenium scripts to suit the use case requirement for Leading Bank in the US. Copilot is used from the Visual Studio IDE environment and generates the Java and Selenium scripts for the test case of the project. Copilot generates the scripts (Selenium locators) for a given function as per the prompt provided by the user. Using Copilot for test case preparation with the help of context and prompting reduces the test case creation cycle by 20-30%, from a total of 40 story points to 30 story points (not following strict Fibonacci for story points estimates).

Case Study 2:

Application modernization from on-premise legacy to Azure cloud adopted Google SAAS and Copilot for migrating existing code to Azure serverless. The customer is having challenges in existing legacy applications like resource cost for the legacy stack, lack of documentation, on-premise data center, and managing large groups of coders and architects. By using the Copilot and maker checkers process we saw 30% productive improvements for migration, 50% improvement in unit test case development, and 30% code migration.

Case Study 3:

Low Code No Code (LCNC) development platforms along with GenAI give faster development in UI generation and prototyping. Chatbot solutions have been provided to the Leading Bank in India to streamline the process the regulations are informed to the bank staff. Develop the chatbot by using LCNC with GenAI without writing the code from the developers. Chatbot has been developed and exposed to internal employees to search and retrieve documents and adhere to bank regulations. Web scraping capabilities helped the bank employees with updates and changes.

5. CONCLUSION

Organizations are increasing their spending on the adoption of GenAI tools for different activities including development and testing activities. Firms are more cautious about security vulnerabilities and copyright issues related to GenAI-generated code. Project teams are expecting to resolve these issues soon from the vendors to mitigate legal concerns. Project teams have seen significant savings on development activities while using GenAI tools like ChatGPT and Copilot. Barriers have been seen from developers' insufficient ability. Educate and continuing training provided to the development teams for the usage of GenAI tools and how to estimate the story point helps the Organizations to minimize the insufficient capabilities. Due to code generation, bug identification, and unit test case preparation with agility the development efforts are reduced to 15-20% after studying the case studies in Section 4. Expecting there may be a lot of improvement in GenAI tools for generating code like APIs, and Microservices structures in coming years. Further study has to continue on other SDLC processes for the usage of the GenAI development tools in the project due to the usage of GenAI tools adoption is new to the Organization. Along with this, perform the study and comparison of effort estimations between traditional way with GenAI development tools usage in the project development with various factors.

6. REFERENCES

[1] Beard, J. W., Storey, V. C., Samuel, B. M., Lukyanenko, R., Wiedemann, A., Schuff, D., ... & Islamzada, F. (2024). Agile Development: The Promise, the Reality, the Opportunity. In Agil-ISE@ CAiSE (pp. 7-17)

[2] Bordas, A., Le Masson, P., Thomas, M., & Weil, B. (2024). What is generative in generative artificial intelligence? A design-based perspective. *Research in Engineering Design*, 1-17.

[3] Bruhin, O., Ebel, P., Müller, L., & Li, M. M. GenAI, and Software Engineering: Strategies for Shaping the Core of Tomorrow’s Software Engineering Practice.

[4] Cheng, H., Husen, J. H., Peralta, S. R., Jiang, B., Yoshioka, N., Ubayashi, N., & Washizaki, H. (2024). Generative AI for Requirements Engineering: A Systematic Literature Review. *arXiv preprint arXiv:2409.06741*.

[5] Fernández-Diego, M., Méndez, E. R., González-Ladrón-De-Guevara, F., Abrahão, S., & Insfran, E. (2020). An update on effort estimation in agile software development: A systematic literature review. *IEEE Access*, *8*, 166768-166800.

[6] Feuerriegel, S., Hartmann, J., Janiesch, C., & Zschech, P. (2024). Generative ai. *Business & Information Systems Engineering*, *66*(1), 111-126.

[7] Finnie, G. R., & Wittig, G. E. (1996, January). AI tools for software development effort estimation. In *Proceedings 1996 International Conference Software Engineering: Education and Practice* (pp. 346-353). IEEE.

[8] Mariani, M., & Dwivedi, Y. K. (2024). Generative artificial intelligence in innovation management: A preview of future research developments. *Journal of Business Research*, *175*, 114542.

[9] Mohammad, A., & Chirchir, B. (2024). Challenges of Integrating Artificial Intelligence in Software Project Planning: A Systematic Literature Review. *Digital*, *4*(3), 555-571.

[10] Nguyen-Duc, A., Cabrero-Daniel, B., Przybylek, A., Arora, C., Khanna, D., Herda, T., ... & Abrahamsson, P. (2023). Generative Artificial Intelligence for Software Engineering--A Research Agenda. *arXiv preprint arXiv:2310.18648*.

[11] Rivera Hernández, B. M., Santos Ayala, J. M., & Méndez Melo, J. A. (2024). Generative AI for software architecture.

[12] Sainio, K. (2023). *Generative Artificial Intelligence Assisting in Agile Project Pain Points* (Doctoral dissertation, Master’s Thesis, Faculty of Management and Business, Tampere University, Finland).

[13] Saunders, G. (2024). Who is the Better Project Manager: Artificial Intelligence or Human Intelligence?

[14] Sauvola, J., Tarkoma, S., Klemettinen, M., Riekki, J., & Doermann, D. (2024). Future of software development with generative AI. *Automated Software Engineering*, *31*(1), 26.

[15] Sido, N., & Emon, E. A. (2024). Low/No Code Development and Generative AI.

[16] Smit, D., Smuts, H., Louw, P., Pielmeier, J., & Eidelloth, C. (2024). The impact of GitHub Copilot on developer productivity from a software engineering body of knowledge perspective.