IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

IMPACT OF GENERATIVE AI ON THE SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC)

[1]Ameya Shastri Pothukuchi, [2]Lakshmi Vasuda Kota, [3]Vinay Mallikarjunaradhya*

[1]Sr Product Manager, Microsoft, Redmond, USA
*[2]IT Risk Sr Analyst, Voya Services, Windsor, USA
*[3]Principal Product Manager, Thomson Reuters, Toronto, Canada

Abstract: The Software Development Lifecycle (SDLC) is a solid framework which has guided the broadly sequential nature of project delivery in the Tech industry. The way it has been implemented has changed over the years to a great extent in terms of the scale and complexity of the projects. Agile and other project management methodologies have shortened the time spent on each stage, because they are innately iterative and focus on shipping incremental software in quick intervals.

The recent development of an AI ecosystem with increasingly powerful generative AI technologies like ChatGPT has led to many questions. Foremost among these questions are how these developments will necessitate the software industry and development lifecycle itself to undergo a transformation, and whether it will impact the nature of work for millions of Tech professionals worldwide.

This paper focuses on the latter of the two questions above - and the authors would like to propose a new SDLC model based on how the different stages of the SDLC will likely undergo a fundamental shift catalyzed by recent advances made in generative AI.

Index Terms **- Generative AI, SDLC, ChatGPT, Software Development Life cycle, Bard**

I. INTRODUCTION

The traditional waterfall model has disparate stages that are sequential. The model follows a linear flow but suffers from the limitation of having to define all requirements at the beginning of the project. This leaves no room for accommodating changes and addressing the uncertainties in the project. Agile methodologies came into favour in the Tech industry starting in the 2010s and allowed for flexibility and planning in every phase of the project development. By involving stakeholders early on in the development, there is more scope to test the functionality at every iteration to get stakeholder feedback, rather than late in the process.

With the advent of AI - especially Generative AI, many of the respondents working in the Tech industry we surveyed were of the opinion that Generative AI would cause pace of work and output to increase and employee workforce requirements to decrease in the medium term future (5-8 years). As such, the aim of this study is to fill the gap in the literature by proposing a new SDLC model that reflects this new age of Generative AI-Assisted Software Development Life cycle (hereafter referred to as GAASD Lifecycle).

II. RESEARCH METHODOLOGY

2.1 Participant Sampling and Selection - The selection of interview and survey participants was based on purposive sampling.

A pool of participants was created by purposely selecting software engineers, UX Designers and PMs from different types of organizations with average work experience of 12.6 years.

2.2 Interview Response Coding - Regarding data analysis, we utilized an interpretivist approach. Our process involved the initial coding of data, followed by categorization and identification of thematic development.

The interviews and participant surveys were conducted in July 2023, with each interview lasting approximately 45-60 minutes and at least two researchers hosting it. Verbatim recordings were transcribed with participant consent, and Personally identifiable data (PII) was cleaned from transcription, and participants were referred to using pseudonyms.

A total of 30 Tech industry professionals whose experience was in the range of 3 to 28 years were interviewed.

Table 1: Interview Questions guide

1	Do you already use Generative AI (ChatGPT/Bard etc.) for your everyday tasks in your current role and to what extent?
2	How long from now do you think Generative AI models (like ChatGPT/Bard) will become reliable enough, that developers/technical PMs in the software industry can start depending on them to generate production-quality code and technical documents?
3	What are some of the effects Generative AI will have on the software (aka Tech) industry in terms of responsibility per employee and overall workforce requirements?
4	What do you think will be the effects of Generative AI on each phase of the SDLC?
5	Is there a lack of understanding currently about how Generative AI can realistically be used in the software industry, and what its limitations are?
6	Is there any additional information you've seen regarding adoption of Generative AI in the software industry that we didn't ask about, that you think is important for us to know?

III. MODELING AND ANALYSIS

For data analysis, we adopted an interpretivist approach. Our process involved the initial coding of data, followed by categorization and identification of thematic development. Additionally, to establish the credibility of our study, we evaluated our findings using the criteria described in Lincoln and Guba (1985). The 'Discussion' and Results sections below present the findings of our interviews with respondents about the impact of SDLC on AI.

To ensure the reliability of our findings, we conducted an external peer review of our results and process. The data and analysis process were made available to an external peer reviewer to scrutinize our interpretations and conclusions. This step served to strengthen the dependability of our results.

IV. RESULTS AND DISCUSSION

4.1 Discussion - From our discussions, the insights we gained regarding impact of AI in each phase of SDLC are summarized in detail below:

4.1.1 SDLC Stage 1- Planning and Requirements Gathering (i.e. Discovery):

Planning

In agile, planning is an overarching theme that is integrated throughout the development lifecycle and is not a separate phase in agile methodologies. In the waterfall SDLC model, the planning phase exists as the initial step.

After generative AI: With Generative AI, the planning phase can change considerably. Generative AI can analyze historical data from previous Agile projects, identify patterns, and generate more accurate estimates for future tasks and sprints. It can assist teams in better understanding their velocity and capacity for each iteration.

Requirements Gathering

In understanding the needs of the business stakeholders and users, AI can help streamline proposal writing and define the scope of project. AI tools like ChatGPT and Bard can build prototypes of applications reducing ambiguity about desired functionality. The requirements can be shaped from the text by utilizing Natural Language Processing (NLP) or by employing Machine Learning(ML) to study user behavior.

AI platforms can condense the time and effort taken by analysts to write user stories in translating business requirements. Business Analysts and Product Managers can distill the essence of the project as Generative AI tools can generate requirements that meet the scope of the project.

4.1.2 SDLC Stage 2 - Design

AI can guide the build of new system designs by rendering insights into selecting appropriate architecture patterns. It can generate system design from scratch based on the system definitions made by technical leads.

In addition, for existing designs – AI can aid in the below-mentioned ways:

- Analyze weaknesses in the designs,
- Suggest alternate design approaches by evaluation of design proposals ensuring uniformity in design
- Provide testing capabilities for performance and scalability **eg**: load management
- Documentation of system design
- UX (User-experience) mockups

By creating a blueprint for the software (front-end UX mockups, system architecture, data structure and algorithms inclusive), AI can assist in making informed decisions based on best practices.

4.1.3 SDLC Stage 3 - Development

Development (used interchangeably with 'Build' or 'Engineering') is the heart of the SDLC and most respected tech companies are developer or engineer driven. This phase is the most susceptible to change, of all the SDLC phases.

AI can convert system design into code in a short time by offering code suggestions that incorporate syntax checks, algorithmic improvements, code formatting and debugging. Generative AI can drive code optimization through improved performance and resource utilization by making efficient use of memory and organization of data structures.

Introducing an AI system's capability into software development is expected to not only help programmers (aka 'developers') do their tasks efficiently and faster, but also help organizations to reduce the workforce as employees can accomplish more tasks in the same amount of time.

According to a lead Senior Software Engineer we interviewed from one of the top MAAMA companies in Tech - "*Generative AI models are becoming increasingly powerful and accurate. Some experts even predict programmers will likely only review code written by AI models - in order to fine tune it to the requirements of project. This might likely lead to a reduction in workforce requirements for major corporations."*

4.1.4 SDLC Stage 4 - Testing and Quality Assurance

AI can help run tests on the developed code and features to make sure the requirements for functionality, performance and usability are met.For instance, UAT and Regression testing can utilize AI Capabilities and the results can be analyzed to fix errors in lower environment before pushing the code base to production.

AI tools can assist in generating test scripts and scenarios to ensure a well-tested software product is presented to end-users. It can aid data validation and debugging by identifying edge cases, executing automated cases, and logging defects or errors. Since AI can understand test frameworks, programming languages and methodologies, high-quality test cases can be created within a short turnaround time. Test Summary Reports (TSR) can be generated clearly and concisely to assess the readiness for deployment.

4.1.5 SDLC Stage 5 - Deployment and release

AI can help prepare for production release by organizing software components and laying out the schedule for deployment process. AI can automate the installation of software breaking down the release into individual tasks such as copying files, server configurations and finally restarting services. It also can monitor performance after deployment.

In the Post-deployment phase, AI tools can also offer insights and stats to improve the processes and boost the system's performance.

4.1.6 SDLC Stage 6 - Maintenance and Support:

In this phase, Generative AI tools are quite valuable for diagnosis and troubleshooting of production defects by triaging the issues and addressing the service tickets.

This helps improve resolution rates by monitoring system performance and – AI capabilities can be unleashed to see massive efficiencies.

AI can help :

- secure systems by deploying patches,
- analyze user behavior
- provide customized reports to applications owners and
- comply with regulatory requirements and security policies/standards.

Consequently, we **propose a new model of SDLC called GAASD (Generative AI assisted Software Development) Lifecycle** which incorporates the recent advances and characteristics of Generative AI. It differs from the two well-known SDLCs of Waterfall and Agile, in that human experts are largely expected to play the role of a reviewer to clean and customize the output (which could be code, UX mockup or a detailed requirements sheet) produced by Generative AI asper GAASD model given below.

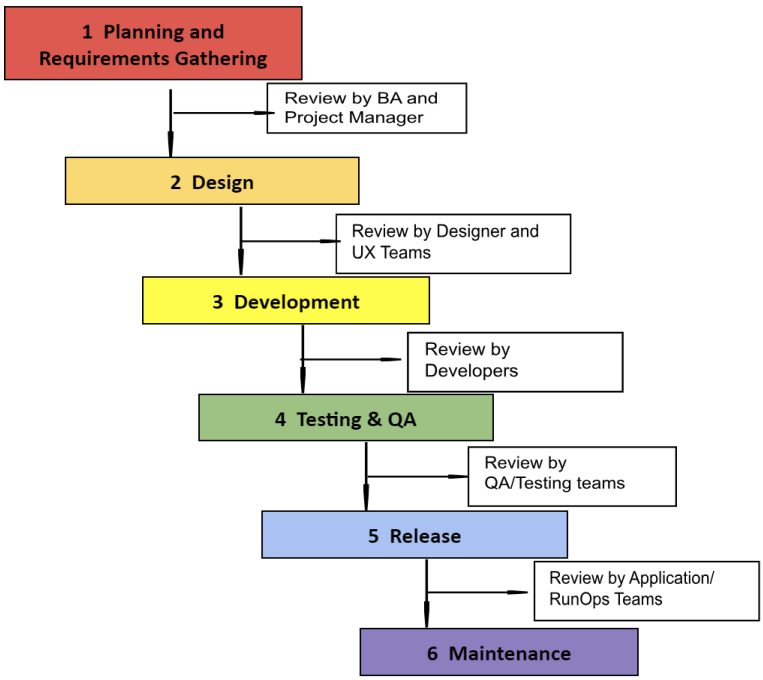

Fig (a) - Generative AI-Assisted Software Development Life cycle (GAASD Lifecycle).

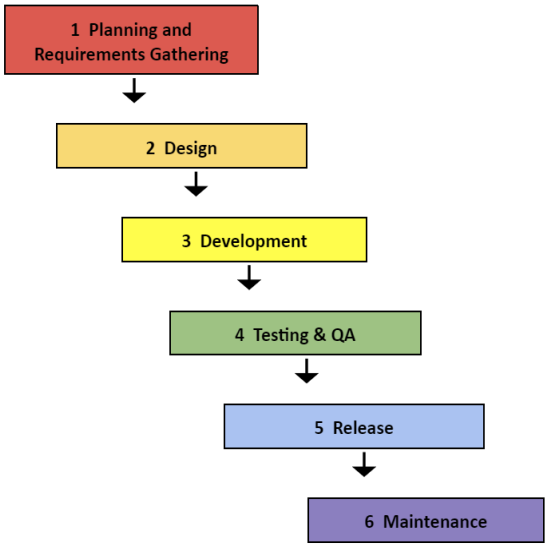

Fig (b) - Waterfall SDLC Methodology

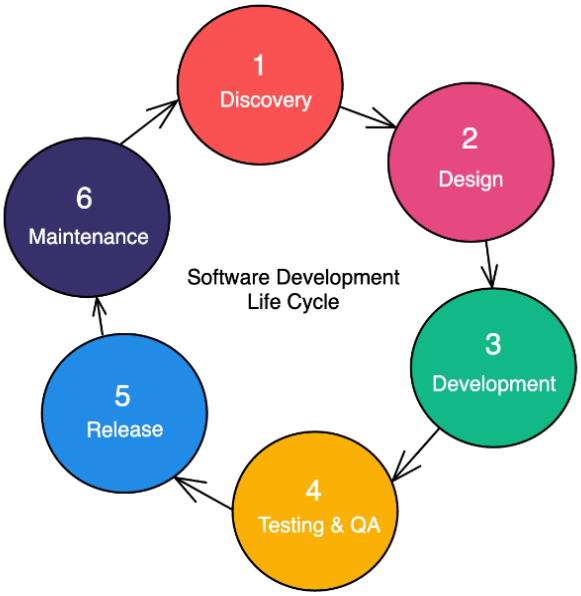

Fig (c) - Agile SDLC Methodology

4.2 Results - This research aimed to understand how the SDLC will need to adapt to the recent advances in Generative AI. After conducting an extensive review of the literature and an empirical study through our interviews, we have observed that Generative AI will have following impacts and characteristics:

1. **Speed**: Slightly more than half of our respondents interviewed or surveyed said it is 'Somewhat Likely' or 'Very likely' that in the next 5 years, the pace of software deployments in the industry would accelerate significantly as a direct result of Generative AI being incorporated into the Software Development Lifecycle (SDLC).
2. **Proliferation:** Generative AI is already being used by a majority of tech industry professionals for their everyday tasks. Only 20% of our 30 respondents said they don't use Generative AI at all in their daily tasks. The remaining 80% of Tech industry respondents indicated they used Generative AI frequently or minimally in their everyday tasks.
3. **Reliability:** 60% of our respondents believe that by 2028 (i.e. 5 years from now), Generative AI models (like ChatGPT/Bard) will become reliable enough that developers and technical PMs in the software industry can start depending on them to generate production-quality code and technical documents (like specifications docs, UX wireframes etc, test scenarios doc etc).
4. **Layoffs and Social effects**: 50% of our respondents were of the opinion that widespread use of Generative AI in the Tech industry will likely lead to a reduction in workforce requirements for major Tech corporations in the medium term future (i.e. the next 5-8 years).
5. **Reviewers:** Further, half our respondents also said it is possible that "With Generative AI doing most of the work for developers (engineers) and other folks like UX designers, technical roles in the Tech industry will be reduced to the role of reviewers (who review the AI's output before pushing it to prod).
6. **Limitations:** While AI is very resourceful in the development of SDLC, it has certain areas which can be improved.
    - **Bias** : AI tools are prone to reflect biases that arise from the data or the erroneous assumptions of the Machine Learning process.
    - **Hallucination** : The results and outcomes produced by AI can be false which deviate from facts and logic.
    - **Coherence** : AI tools strain to produce long strings of code without additional prompts. The software generated is not always usable as-is in most cases, especially when the application is complex and has a lot of components. The results generated by AI tools are brief and incomplete in some cases, requiring the user to provide additional context.
    - **Inaccuracy**: AI tool uses web-scraping to fetch data from various sources. For the reasons listed above, despite the multifold applications of AI- it cannot be trusted as the one-stop shop as it requires critical evaluation by humans or machines.
    - **Originality**: AI tools produce results that are similar in content and construction and the responses generated lack originality or novelty.
    - **Performance**: AI's current capabilities currently are assistive at best. It definitely enhances the efficiency of users but is not at a stage where the activities can be left to AI without manual intervention/supervision.

V. CONCLUSION

These findings have important implications for both researchers and practitioners. For researchers, this study highlights the need for Tech leaders to plan for the impacts of Generative AI on software development. Further, this study is meant to ideate new Software Development Lifecycles (SDLCs), in a way that is both sustainable and humane for employees and society.

While this study offers valuable insights into the issue of Generative AI's impact on SDLC, it is important to acknowledge its limitations. The relatively small sample size of this study may restrict the extent to which the findings can be generalized to a broader population or industries. Nevertheless, the results of this study can still provide significant implications for future research and practice in addressing the impacts of Generative AI on software development.

The researchers note that the use cases of AI are multifold and Generative AI is very resourceful in the implementation of SDLC irrespective of the size, scale and nature of the enterprises adopting it.

REFERENCES

[1] Laato, Birkstedt et al (1990). AI Governance in the System Development Life Cycle: Insights on Responsible Machine Learning Engineering

[2] M., & Strauss, A. (1990). Grounded theory research: Procedures, canons, and evaluative criteria.

[3] Lincoln, Y. S., & Guba, E. G. (1985). Naturalistic inquiry. sage.

[4] Mulgund, Pavankumar; Singh Raghvendra; Pothukuchi Ameya Shastri "A design science approach to the development of graduate IS course syllabus on Prompt Engineering", Submitted to, ISCAP 2023 conference.

[5] Sharma, Shreta and Pandey, S.K. 2013. Revisiting Requirements Elicitation Techniques. International Journal of Computer Applications (0975-8887). Vol. 12, pp. 35-39.

[6] Robles-Aguilar A, Ocharan-Hernandez J, Sanchez-Garcia A and Limon X. (2021). Software Design and Artificial Intelligence: A Systematic Mapping Study 2021 9th International Conference in Software Engineering Research and Innovation (CONISOFT). 10.1109/CONISOFT52520.2021.00028. 978-1-6654-4361-6. (132-141).

[7] Sharma, shreta and pandey, s.k. 2015: integrating ai techniques in sdlc: design phase perspective

[8] B. Namatherdhala, N. Mazher, G.K. Sriram : Artificial Intelligence in Product Management: systematic review Int Res J Mod Eng Technol Sci, 4 (7) (2022), pp. 2914-2917