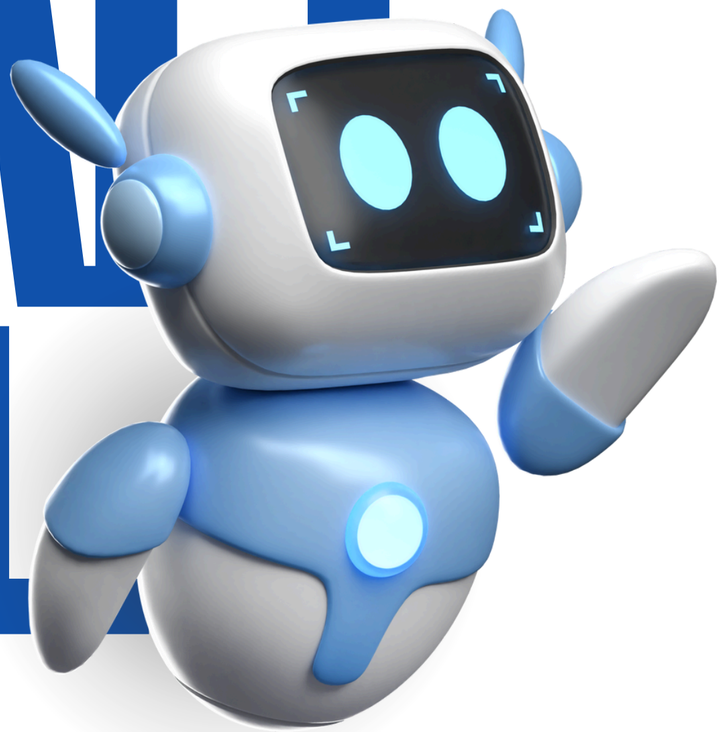
https://latief.reswandana.com

PRESENT PROPOSAL

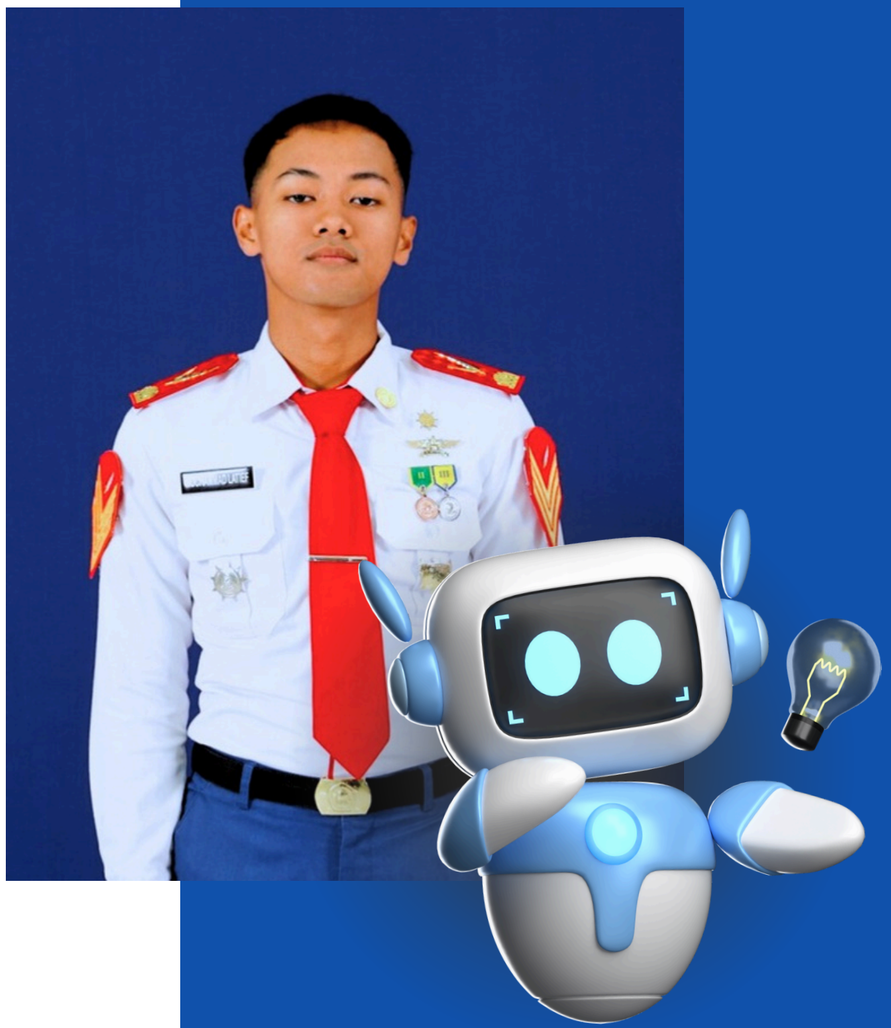


Pendekatan Sistem Kecerdasan Buatan Agenik pada Secure Software Development Life Cycle

@latief_reswandana

LATIEF RESWANDANA

222110181 2



III Rekayasa Perangkat Lunak Kripto

DAFTAR ISI

PROBLEM

LAMA

Secure Software Development Methodologies:
A Multivocal Literature Review

Arina Kudriavtseva, Olga Gadyatskaya

Accelerating Software Development Using Generative AI:
ChatGPT Case Study

Asha Rajbhoj
TCS Research, Tata Consultancy Services, Pune, I
asha.rajbhoj@tcs.com

Piyush Kulkarni
TCS Research, Tata Consultancy Services, Pune, I
piyush.kulkarni3@tcs.com

ABSTRACT

The Software Development Life Cycle (SDLC) comprises mu
phases, each requiring Subject Matter Experts (SMEs) with
specific skills. The efficacy and quality of deliverables of each
are skill dependent. In recent times, Generative AI techn
including Large-scale Language Models (LLMs) like GPT, ha
come significant players in software engineering. These m
trained on extensive text data, can offer valuable contributi
software development. Interacting with LLMs involves fe
prompts with the context information and guiding the gene
of textual responses. The quality of the response is dependa
the quality of the prompt given. This paper proposes a sy
atic prompting approach based on meta-model concepts for
phases. The approach is validated using ChatGPT for sma
complex business application development. We share the app
and our experience, learnings, benefits obtained, and the chall
encountered while applying the approach using ChatGPT
experience indicates that Generative AI techniques, such as
GPT, have the potential to reduce the skills barrier and acce
software development substantially.

CCS CONCEPTS

• Software and its engineering → Software creation and
agement; Software development process management; Soft
development methods.

KEYWORDS

AI in SDLC, Large Language Models, Generative AI, ChatGPT,

remote machines, leading
alware. Existing security
ime were inadequate [4],
erent approach to protect
are.
ally studying how to build

IEEE Access

Received 22 November 2024, accepted 9 December 2024, date of publication 17 December 2024,
date of current version 30 December 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3519423

RESEARCH ARTICLE

AI-Analyst: An AI-Assisted SDLC Analysis
Framework for Business Cost Optimization

NURUZZAMAN FARUQUI[1], PRIYABRATA THATOI[2], ROHIT CHOUDHARY[3],
IVANA RONCEVIC[4], HAMED ALQAHTANI[5], IQBAL H. SARKER[6], (Member, IEEE),
AND SHAPLA KHANAM[7]

[1]Department of Software Engineering, Daffodil International University, Daffodil Smart City, Dhaka 1216, Bangladesh
[2]Amazon, Chicago, IL 60606, USA
[3]Amazon, Dallas, TX 13455, USA
[4]Applied Linguistics Research Laboratory, Department of Linguistics and Translation, Prince Sultan University, Riyadh 11586, Saudi Arabia
[5]College of Computer Science, Informatics and Computer Systems Department, Center of Artificial Intelligence, King Khalid University, Abha 62521, Saudi Arabia
[6]Centre for Securing Digital Futures, School of Science, Edith Cowan University, Perth, WA 6027, Australia
[7]Faculty of Computing and Digital Technology, IHELP University, Kuala Lumpur 50490, Malaysia

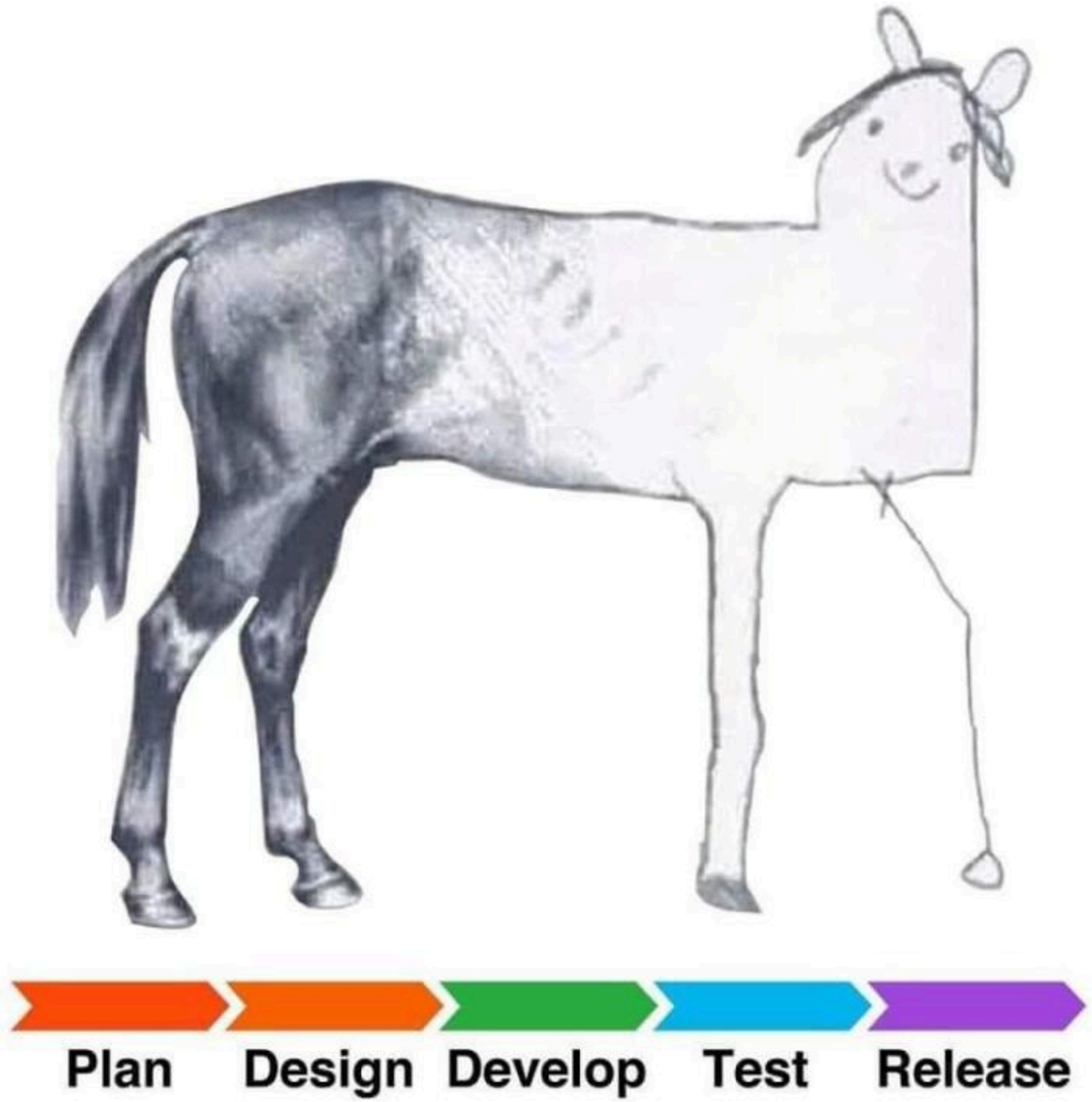
Corresponding authors: Shapla Khanam (shk.shapla@gmail.com) and Hamed Alqahtani (hsqahtani@kku.edu.sa)

This work was supported by the Deanship of Scientific Research at King Khalid University through a Large Group Research Project under Grant RGP2/267/45.

ABSTRACT Managing the System Development Lifecycle (SDLC) is a complex task because of its involvement in coordinating diverse activities, stakeholders, and resources while ensuring project goals are met efficiently. The complex nature of the SDLC process leaves plenty of scope for human error, which impacts the overall business cost. This paper introduces AI-Analyst, an AI-assisted framework developed

SSDLC merupakan proses yang kompleks dan memakan waktu :

- melibatkan banyak fase
- melibatkan banyak orang
- keahlian khusus



TIDAK KONSISTEN

On the Secure Software Development Process: CLASP and SDL Compared

Secure Software Development Methodologies:
A Multivocal Literature Review

Arina Kudriavtseva, Olga Gadyatskaya

to, Wouter Joosen
uven

Accelerating Software Development Using Generative AI:
ChatGPT Case Study

Asha Rajbhoj
TCS Research, Tata Consultancy Services, Pune, I
asha.rajbhoj@tcs.com

Piyush Kulkarni
TCS Research, Tata Consultancy Services, Pune, I
piyush.kulkarni3@tcs.com

ABSTRACT

The Software Development Life Cycle (SDLC) comprises mu
phases, each requiring Subject Matter Experts (SMEs) with
specific skills. The efficacy and quality of deliverables of each
are skill dependent. In recent times, Generative AI techn
including Large-scale Language Models (LLMs) like GPT, ha
come significant players in software engineering. These m
trained on extensive text data, can offer valuable contributi
software development. Interacting with LLMs involves fe
prompts with the context information and guiding the gener
of textual responses. The quality of the response is depende
the quality of the prompt given. This paper proposes a sy
atic prompting approach based on meta-model concepts for
phases. The approach is validated using ChatGPT for sma
complex business application development. We share the app
and our experience, learnings, benefits obtained, and the chall
encountered while applying the approach using ChatGPT
experience indicates that Generative AI techniques, such as
GPT, have the potential to reduce the skills barrier and acce
software development substantially.

CCS CONCEPTS

• Software and its engineering → Software creation and
agement; Software development process management; Soft
development methods.

KEYWORDS

remote machines, leading
alware. Existing security
ime were inadequate [4],
rrent approach to protect
are.
lly studying how to build

ular interest in the context of this
ness is concerned, they both pro-
tivities covering a broad spectrum
cle. The processes also have un-
instance, Microsoft is using SDL
was con-
rity com-
SDL and
he former
SP, on the
small or-
nally, the
thorough

ating and
i) generic
n order to
thod; (ii)
ethod are
trong and
improve-
efit from,

ion 2 de-
opted for
sentation
heir char-
the com-
ses, on a
r of chal-

IEEE Access

Received 22 November 2024, accepted 9 December 2024, date of publication 17 December 2024,
date of current version 30 December 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3519423

RESEARCH ARTICLE

AI-Analyst: An AI-Assisted SDLC Analysis Framework for Business Cost Optimization

NURUZZAMAN FARUQUI[1], PRIYABRATA THATOI[2], ROHIT CHOUDHARY[3], IVANA RONCEVIC[4], HAMED ALQAHTANI[5], IQBAL H. SARKER[6], (Member, IEEE), AND SHAPLA KHANAM[7]

[1]Department of Software Engineering, Daffodil International University, Daffodil Smart City, Dhaka 1216, Bangladesh
[2]Amazon, Chicago, IL 60606, USA
[3]Amazon, Dallas, TX 13455, USA
[4]Applied Linguistics Research Laboratory, Department of Linguistics and Translation, Prince Sultan University, Riyadh 11586, Saudi Arabia
[5]College of Computer Science, Informatics and Computer Systems Department, Center of Artificial Intelligence, King Khalid University, Abha 62521, Saudi Arabia
[6]Centre for Evening Digital Futures, School of Science, Edith Cowan University, Perth, WA 6027, Australia
[7]Faculty of Computing and Digital Technology, IBELP University, Kuala Lumpur 50490, Malaysia

Corresponding authors: Shapla Khanam (dit.shapla@gmail.com) and Hamed Alqahtani (hsqahtani@kku.edu.sa)

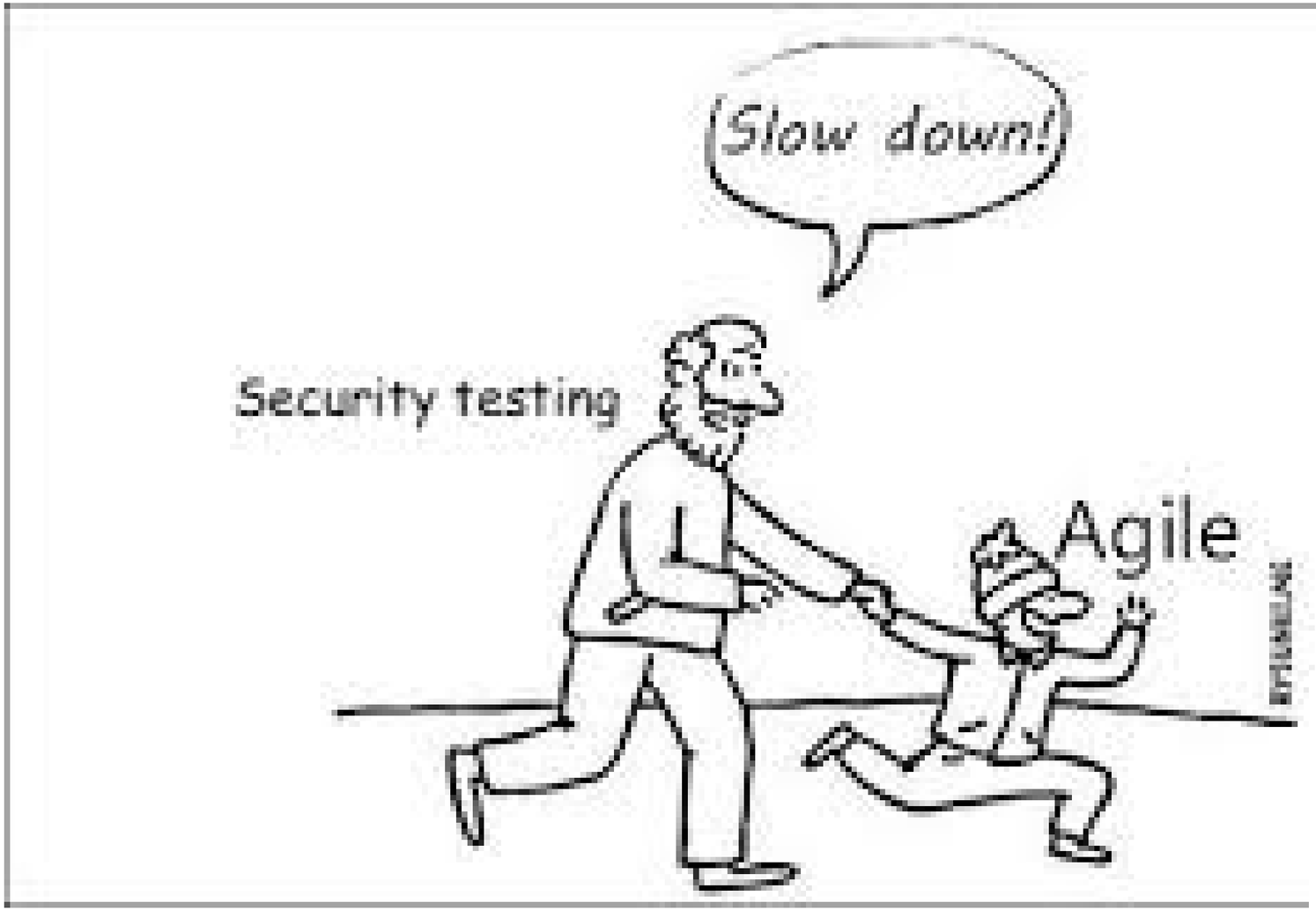
This work was supported by the Deanship of Scientific Research at King Khalid University through a Large Group Research Project under Grant RGP.2/267/45.

ABSTRACT Managing the System Development Lifecycle (SDLC) is a complex task because of its involvement in coordinating diverse activities, stakeholders, and resources while ensuring project goals are

SSDLC tidak konsisten di setiap tahap

- perbedaan metodologi
- perbedaan praktik keamanan
- kerangka kerja berbeda-beda
- fleksibilitas proses -> variasi dan non-determinisme dalam penerapannya.

TIDAK SEPADAN



On the Secure Software Development Process: CLASP and SDL Compared

Secure Software Development Methodologies:
A Multivocal Literature Review

Arina Kudriavtseva, Olga Gadyatskaya

to, Wouter Joosen
uven

Accelerating Software Development Using Generative AI:
ChatGPT Case Study

Asha Rajbhoj
TCS Research, Tata Consultancy Services, Pune, I
asha.rajbhoj@tcs.com

Piyush Kulkarni
TCS Research, Tata Consultancy Services, Pune, I
piyush.kulkarni2@tcs.com

remote machines, leading
alware. Existing security
ime were inadequate [4].
rent approach to protect
ure.
lly studying how to build

ular interest in the context of this
ness is concerned, they both pro-
tivities covering a broad spectrum
rcle. The processes also have un-
instance, Microsoft is using SDL
was con-
rity com-
SDL and
he former
us, mak-
SP, on the
small or-
nally, the
thorough

ABSTRACT

The Software Development Life Cycle (SDLC) comprises m
phases, each requiring Subject Matter Experts (SMEs) with
specific skills. The efficacy and quality of deliverables of each
are skill dependent. In recent times, Generative AI techn
including Large-scale Language Models (LLMs) like GPT, ha
come significant players in software engineering. These m
trained on extensive text data, can offer valuable contributi
software development. Interacting with LLMs involves fe
prompts with the context information and guiding the gener
of textual responses. The quality of the response is depende
the quality of the prompt given. This paper proposes a sy
atic prompting approach based on meta-model concepts for
phases. The approach is validated using ChatGPT for sma
complex business application development. We share the app
and our experience, learnings, benefits obtained, and the chall
encountered while applying the approach using ChatGPT
experience indicates that Generative AI techniques, such as
GPT, have the potential to reduce the skills barrier and acce
software development substantially.

IEEE Access

Received 22 November 2024, accepted 9 December 2024, date of publication 17 December 2024,
date of current version 30 December 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3519623

RESEARCH ARTICLE

AI-Analyst: An AI-Assisted SDLC Analysis Framework for Business Cost Optimization

NURUZZAMAN FARUQUI[1], PRIYABRATA THATOI[2], ROHIT CHOUDHARY[3], IVANA RONCEVIC[4], HAMED ALQAHTANI[5], IQBAL H. SARKER[6], (Member, IEEE), AND SHAPLA KHANAM[7]

[1]Department of Software Engineering, Daffodil International University, Daffodil Smart City, Dhaka 1216, Bangladesh
[2]Amazon, Chicago, IL 60606, USA
[3]Amazon, Dallas, TX 13455, USA
[4]Applied Linguistics Research Laboratory, Department of Linguistics and Translation, Prince Sultan University, Riyadh 11586, Saudi Arabia
[5]College of Computer Science, Information and Computer Systems Department, Centre of Artificial Intelligence, King Khalid University, Abha 62521, Saudi Arabia
[6]Centre for Securing Digital Futures, School of Science, Edith Cowan University, Perth, WA 6027, Australia
[7]Faculty of Computing and Digital Technology, HELP University, Kuala Lumpur 50490, Malaysia

Corresponding authors: Shapla Khanam (dsk.shapla@gmail.com) and Hamed Alqahtani (hsqahtani@kku.edu.sa)

This work was supported by the Deanship of Scientific Research at King Khalid University through a Large Group Research Project under Grant RGP2/26/45.

ABSTRACT Managing the System Development Lifecycle (SDLC) is a complex task because of its involvement in coordinating diverse activities, stakeholders, and resources while ensuring project goals are met efficiently. The complex nature of the SDLC process leaves plenty of scope for human error, which impacts the overall business cost. This paper introduces AI-Analyst, an AI-assisted framework developed

CCS CONCEPTS

• Software and its engineering → Software creation and
agement; Software development process management; Soft
development methods.

KEYWORDS

AI in SDLC, Large Language Models, Generative AI, ChatGPT,
automation, Automated Software Development

SSDLC dianggap tidak sepadan

- kompleksitas
- biaya tinggi,
- ketergantungan pada keahlian khusus
- keterbatasan struktur

mendorong pencarian pendekatan yang lebih ringan

RESEARCH ARTICLE

AI-Analyst: An AI-Assisted SDLC Analysis Framework for Business Cost Optimization

NURUZZAMAN FARUQUI[1], PRIYABRATA THATOI[2], ROHIT CHOUDHARY[3], IVANA RONCEVIC[4], HAMED ALQAHTANI[5], IQBAL H. SARKER[6], (Member, IEEE), AND SHAPLA KHANAM[7]

[1]Department of Software Engineering, Daffodil International University, Daffodil Smart City, Dhaka 1216, Bangladesh
[2]Amazon, Chicago, IL 60606, USA
[3]Amazon, Dallas, TX 13455, USA
[4]Applied Linguistics Research Laboratory, Department of Linguistics and Translation, Prince Sultan University, Riyadh 11586, Saudi Arabia
[5]College of Computer Science, Informatics and Computer Systems Department, Center of Artificial Intelligence, King Khalid University, Abha 62521, Saudi Arabia
[6]Centre for Securing Digital Futures, School of Science, Edith Cowan University, Perth, WA 6027, Australia
[7]Faculty of Computing and Digital Technology, HELP University, Kuala Lumpur 50490, Malaysia

Corresponding authors: Shapla Khanam (dsk.shapla@gmail.com) and Hamed Alqahtani (hsqahtani@kku.edu.sa)

This work was supported by the Deanship of Scientific Research at King Khalid University through a Large Group Research Project under Grant RGP2/267/45.

ABSTRACT Managing the System Development Lifecycle (SDLC) is a complex task because of its involvement in coordinating diverse activities, stakeholders, and resources while ensuring project goals are met efficiently. The complex nature of the SDLC process leaves plenty of scope for human error, which impacts the overall business cost. This paper introduces AI-Analyst, an AI-assisted framework developed using the transformer-based model with more than 150 million parameters to assist with SDLC management. It minimizes manual effort errors, optimizes resource allocation, and improves decision-making processes, resulting in substantial cost savings. The statistical analysis shows that it saves around 53.33% of costs in an experimental project. The transformer model has been trained with a uniquely prepared dataset tailored for SDLC through transfer learning. It achieved impressive results, with an accuracy of 91.5%, precision of 91.9%, recall of 91.3%, and an F1-score of 91.5%, demonstrating its high reliability and performance. The perplexity score of 15 further indicates the model's strong language understanding capabilities to retrieve relations from complex characteristics of Natural Language Processing (NLP). The AI-Analyst framework represents a significant advancement in integrating Large Language Models (LLMs) into SDLC, offering a scalable and cost-effective solution for optimizing business processes.

INDEX TERMS Transformer model, large language model, system development lifecycle, transfer learning, artificial intelligence, business cost optimization, project management automation, system analyst, LLM, SDLC, AI, PMP.

I. INTRODUCTION

SDLC management requires proper coordination of diverse activities, stakeholders, and resources while ensuring quality, compliance, and alignment with business objectives through- wastage, and many other issues [3]. Human error is a part of SDLC project management as it requires coordination among multiple interdependent static and dynamic entities [4]. The LLMs can reason from complex contexts

Paper ini menunjukkan bahwa SDLC itu kompleks, memakan waktu, dan rawan kesalahan manusia.

FARUQUI ET AL. (2024)

Accelerating Software Development Using Generative AI: ChatGPT Case Study

Asha Rajbhoj
TCS Research, Tata Consultancy Services, Pune, India
asha.rajbhoj@tcs.com

Akanksha Somase
TCS Research, Tata Consultancy Services, Pune, India
akanksha.somase@tcs.com

Piyush Kulkarni
TCS Research, Tata Consultancy Services, Pune, India
piyush.kulkarni3@tcs.com

Vinay Kulkarni
TCS Research, Tata Consultancy Services, Pune, India
vinay.vkulkarni@tcs.com

ABSTRACT

The Software Development Life Cycle (SDLC) comprises multiple phases, each requiring Subject Matter Experts (SMEs) with phase-specific skills. The efficacy and quality of deliverables of each phase are skill dependent. In recent times, Generative AI techniques, including Large-scale Language Models (LLMs) like GPT, have become significant players in software engineering. These models, trained on extensive text data, can offer valuable contributions to software development. Interacting with LLMs involves feeding prompts with the context information and guiding the generation of textual responses. The quality of the response is dependent on the quality of the prompt given. This paper proposes a systematic prompting approach based on meta-model concepts for SDLC phases. The approach is validated using ChatGPT for small but complex business application development. We share the approach and our experience, learnings, benefits obtained, and the challenges encountered while applying the approach using ChatGPT. Our experience indicates that Generative AI techniques, such as ChatGPT, have the potential to reduce the skills barrier and accelerate software development substantially.

CCS CONCEPTS

• **Software and its engineering** → Software creation and management; Software development process management; Software development methods.

KEYWORDS

AI in SDLC, Large Language Models, Generative AI, ChatGPT, SDLC automation, Automated Software Development

ACM Reference Format:
Asha Rajbhoj, Akanksha Somase, Piyush Kulkarni, and Vinay Kulkarni. 2024. Accelerating Software Development Using Generative AI: ChatGPT Case Study. In *17th Innovations in Software Engineering Conference (ISEC 2024), February 22–24, 2024, Bangalore, India.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3641399.3641403

1 INTRODUCTION

The Software Development Life Cycle (SDLC) consists of multiple phases. Each phase of the SDLC produces distinct engineering artifacts and requires Subject Matter Experts (SME) with skills relevant to each phase. The quality, as well as the efficiency of deliverables, is skill dependent. The recent progress in Generative AI techniques has significantly influenced software engineering and we believe it can lower this skills barrier by enabling domain SMEs to operate at the natural language level. Large-scale language models (LLMs), like OpenAI's Codex [1], and Generative Pre-trained Transformer (GPT) [2, 3] are increasingly adopted in AI-driven software engineering. These are trained on a large corpus of text data and have capabilities that make it a valuable tool for software development to enhance the efficiency and quality of the development process. This can save time and effort for the skilled development teams, allowing them to focus on higher-level tasks.

Interacting with LLMs in general involves feeding suitable prompts (natural language instructions) to provide a context and guide its generation of textual responses [4]. Many researchers have discussed the future of ChatGPT and other large language models having a significant effect on how we interact with technology [5, 6]. One may guide LLMs to generate desired responses in multiple ways. For instance, one may directly ask LLM to provide details. Another way is to dictate LLM to preempt how to follow the response generation. For instance, technology stack, design pattern, architecture, and so on are preempted prior to setting up the context for subsequent interaction for the code generation. We propose a systematic prompting approach to leverage LLMs for application development. The approach defines prompt templates for SDLC phases based on meta-model concepts. The prompting approach is validated for small yet complex business application development using ChatGPT. We share the approach and our experience, learnings, benefits obtained, and challenges encountered while applying the approach using ChatGPT. In summary, this paper makes the following contributions:

- Approach for accelerating software development by leveraging Generative AI.
- Generic prompt template for SDLC phases based on high-level meta-model concepts.
- Evaluation of approach using ChatGPT.
- Validation of the approach on a small yet complex enough business application.

Paper ini menunjukan keterlambatan dan hambatan keterampilan di SDLC, serta solusi percepatan dengan AI.

RAJBHOJ ET AL. (2024)

AI Governance in the System Development Life Cycle: Insights on Responsible Machine Learning Engineering

Samuli Laato
sadala@utu.fi
University of Turku
Turku, Finland

Teemu Birkstedt
teemu.birkstedt@utu.fi
University of Turku
Turku, Finland

Matti Mäntymäki
majuma@utu.fi
University of Turku
Turku, Finland

Matti Minkkinen
matti.minkkinen@utu.fi
University of Turku
Turku, Finland

Tommi Mikkonen
tommi.mikkonen@helsinki.fi
University of Helsinki
Helsinki, Finland

ABSTRACT

In this study we explore the incorporation of artificial intelligence (AI) governance to system development life cycle (SDLC) models. We conducted expert interviews among AI and SDLC professionals and analyzed the interview data using qualitative coding and clustering to extract AI governance concepts. Subsequently, we mapped these concepts onto three stages in the machine learning (ML) system development process: (1) design, (2) development, and (3) operation. We discovered 20 governance concepts, some of which are relevant to more than one of the three stages. Our analysis highlights AI governance as a complex process that involves multiple activities and stakeholders. As development projects are unique, the governance requirements and processes also vary. This study is a step towards understanding how AI governance is conceptually connected to ML systems' management processes through the project life cycle.

CCS CONCEPTS

• **Software and its engineering → Software creation and management.**

KEYWORDS

AI Governance, machine learning, software development life cycle, system development life cycle, MLOps, DevOps, software development

ACM Reference Format:
Samuli Laato, Teemu Birkstedt, Matti Mäntymäki, Matti Minkkinen, and Tommi Mikkonen. 2022. AI Governance in the System Development Life Cycle: Insights on Responsible Machine Learning Engineering. In *1st Conference on AI Engineering - Software Engineering for AI (CAIN'22), May 16–24, 2022, Pittsburgh, PA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3522664.3528598

CAIN'22, May 16–24, 2022, Pittsburgh, PA, USA

1 INTRODUCTION

The growing popularity of artificial intelligence (AI), particularly in the form of inscrutable machine learning (ML) models [2] across industry sectors has created the need for AI governance, i.e. to ensure that AI models operate in a planned and a desirable manner from development to initial roll-out, operation and eventual retirement [12]. The global push for AI governance is evident in developments such as the OECD recommendation on AI [36] and the European Union's proposed Artificial Intelligence Act which seeks to harmonise European AI rules [13]. Dafoe defines AI governance to consist of *"devising global norms, policies, and institutions to best ensure the beneficial development and use of advanced AI."* [9]. Other authors have complemented this sociopolitical view by focusing on organizational rules, practices and processes on AI technology [31, 41], and by identifying distinct levels of AI governance ranging from software development teams to organizational management and industry-level oversight [43]. In addition to the organizational AI governance [31], there is a need to involve governance aspects already at the stage of the AI system implementation. Accordingly, there is a need to connect the organizational level AI governance needs into AI implementation processes across the solution life cycle.

Due to the increasing presence of ML models in software systems, data scientist and ML engineers have become prominent roles in software development teams. Thus, there is a need to integrate the work of data scientists and ML engineers into system development life cycle (SDLC) models [1, 20]. This is not a straightforward task, as the process of creating ML models is often experimental and takes place in uncharted territory, and is consequently, more difficult to predict than more mundane well established IT system development processes [20, 22]. Furthermore, due to the characteristic of ML models being inscrutable black boxes, issues such as ensuring explainability [2] and establishing audit trails for the ML models [9, 12] have to be accounted for. Hence, in addition to linking ML engineering with conceptual SDLC models, it is important to involve the aspect o AI governance in as well.

There is a long research tradition in SDLC models [40] with recent work partially taking place also under other terms such as DevOps [11] and MLOps [48]. Similarly AI research has a long history, and now that the field is experiencing a resurgence after

Paper ini menyoroti ketidakjelasan pemetaan tata kelola ke fase SDLC (ketidakonsistenan).

LAATO ET AL. (2022)

Kompleks, lambat, dan tidak konsisten...

butuh pendekatan baru!!

PROPOSED SOLUTION

AGENTIC AI

ALL THINGS IN A NUTSHELL

AGENTIC AI

APA ITU?

AI yang tidak sekadar reaktif, tapi proaktif, mampu mengambil inisiatif, dan berkoordinasi untuk menyelesaikan tugas kompleks secara mandiri

AGENTIC AI

AGEN-AGEN

dinamakan agentic karena dalam arsitekturnya berisi kumpulan ai agent yang berorkestrasi

SSDLC

SECURE SOFTWARE DEVELOPMENT LIFE CYCLE

mengintegrasikan keamanan sejak awal pengembangan perangkat lunak, namun praktiknya sering terkendala kompleksitas, kesalahan manual, dan keterbatasan sumber daya.

CLASP

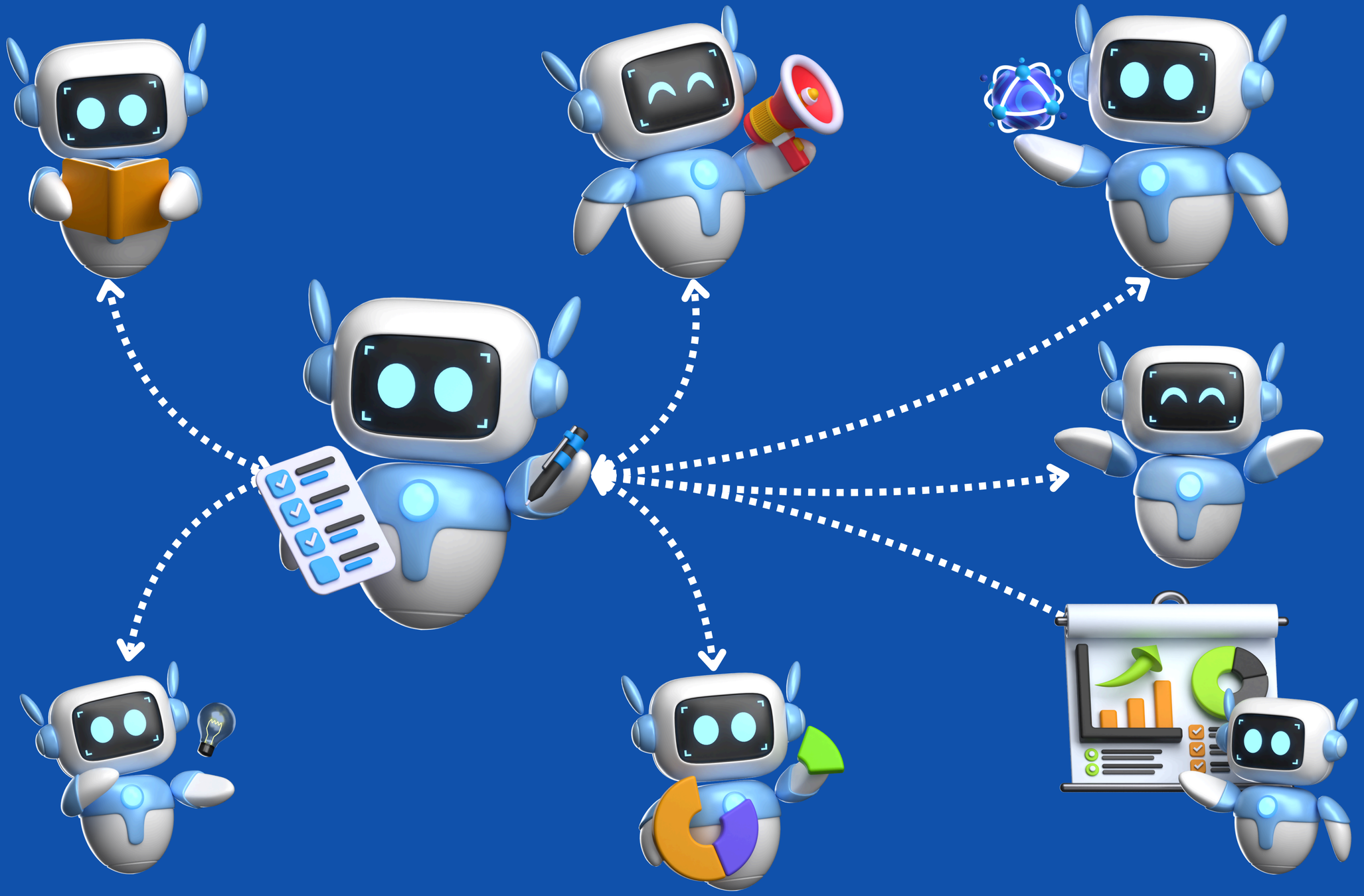
COMPREHENSIVE, LIGHTWEIGHT APPLICATION SECURITY PLAYBOOK

adalah kerangka kerja berbasis peran yang ringan dan komprehensif untuk SSDLC, tetapi masih rentan pada koordinasi rumit dan dokumentasi manual.

DSRM + EKSPERIMEN

GABUNGAN DUA METODOLOGI

memungkinkan penciptaan artefak baru
sekaligus evaluasi efektivitasnya secara ilmiah
dan terukur.



COCOK → SSDLC

METHODOLOGY

DSRM + Eksperimen

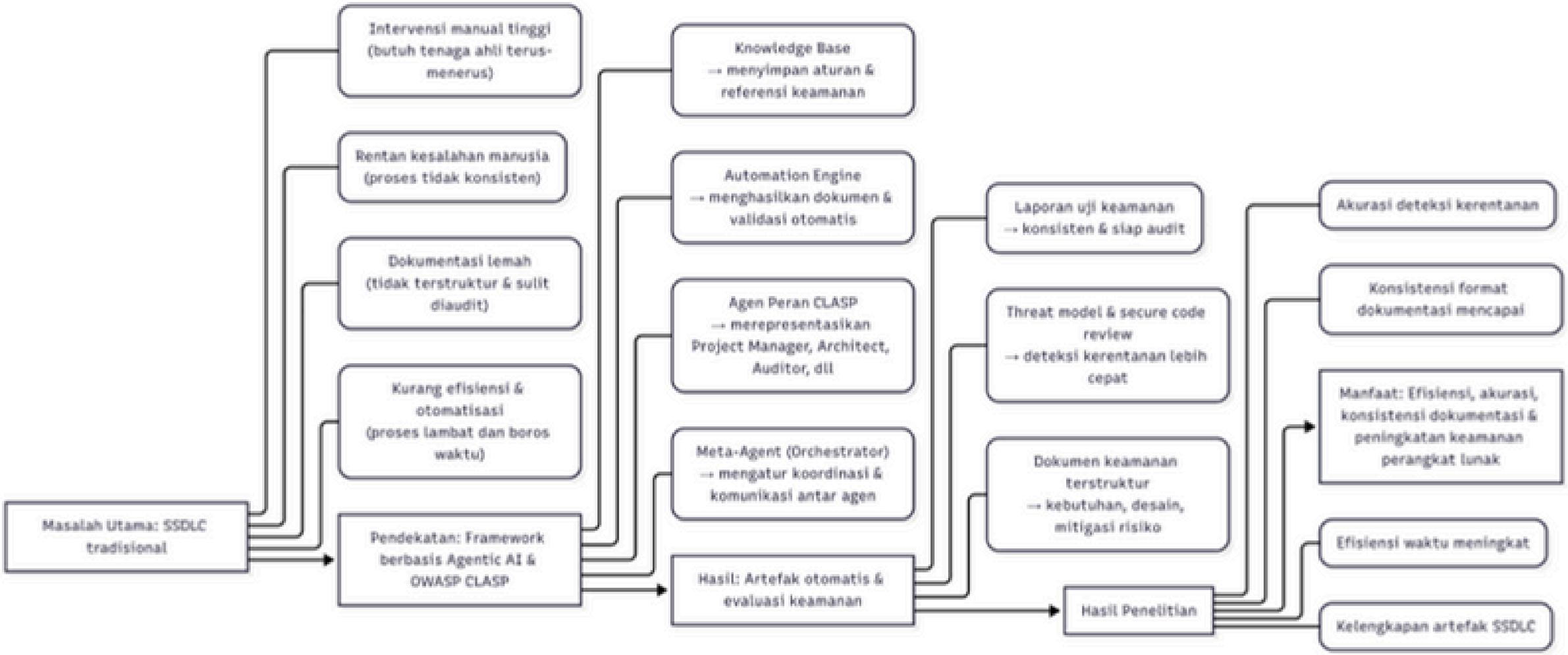
DSRM

memungkinkan penciptaan artefak inovatif yang secara langsung menjawab permasalahan dalam SSDLC

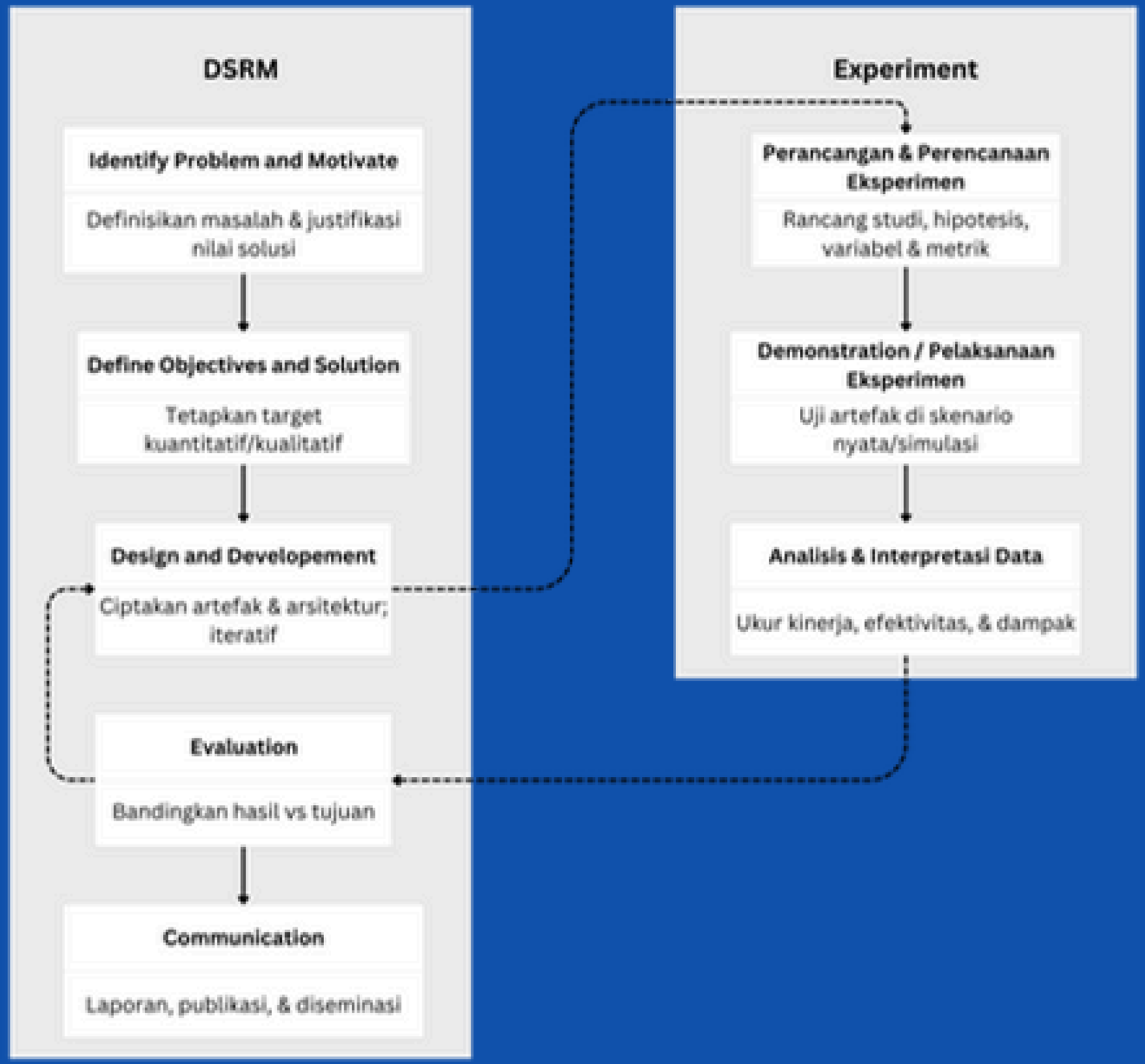
Eksperimen

memberikan evaluasi kuantitatif yang ketat untuk membuktikan efektivitas, efisiensi, dan kualitas artefak dibandingkan metode tradisional

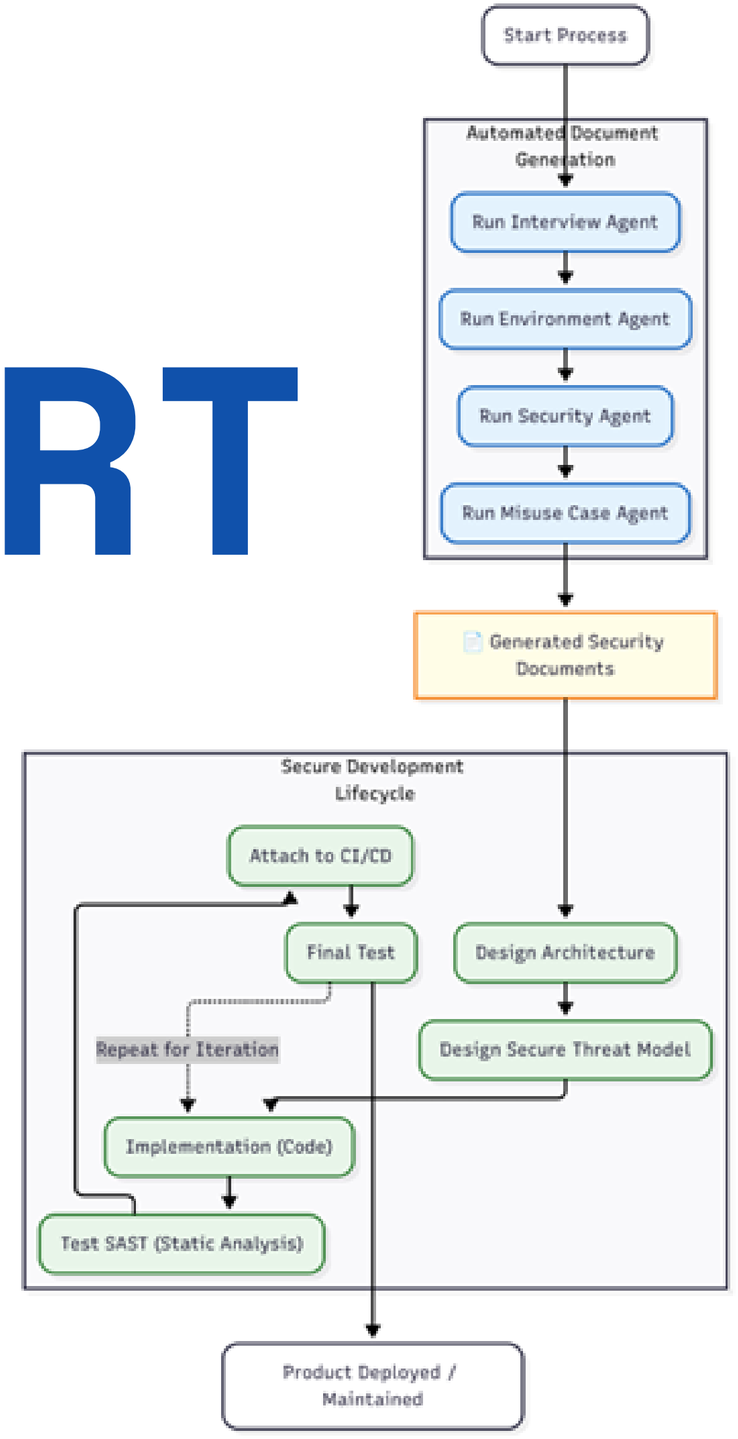
KERANGKA KONSEPTUAL



DESAIN PENELITIAN

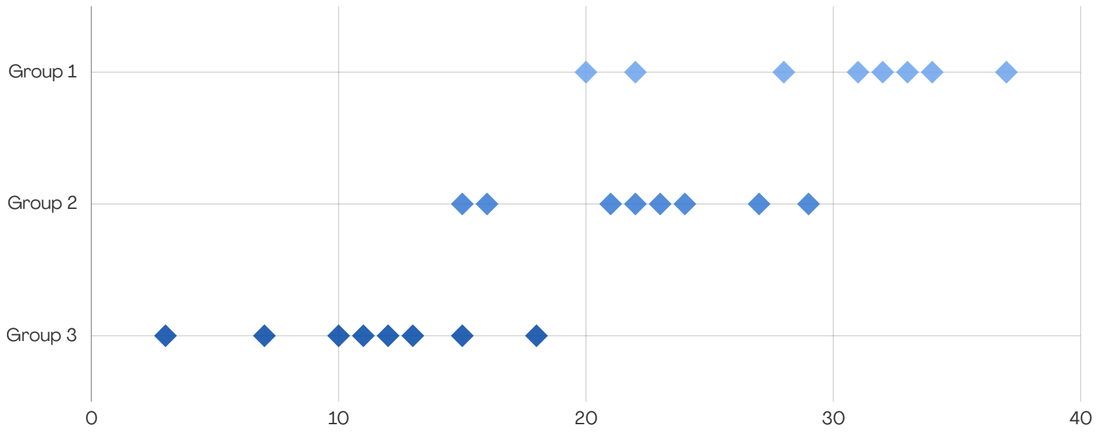


FLOWCHART


Start Process
Automated Document Generation
Run Interview Agent
Run Environment Agent
Run Security Agent
Run Misuse Case Agent
📄 Generated Security Documents
Secure Development Lifecycle
Attach to CI/CD
Final Test
Design Architecture
Repeat for Iteration
Design Secure Threat Model
Implementation (Code)
Test SAST (Static Analysis)
Product Deployed / Maintained

DEMO

SUMMARY



penelitian ini diharapkan mampu membantu **mengotomatisasi** pengembangan perangkat lunak yang lebih **terukur**, **jelas**, **aman**, dan **cepat** tanpa menambah beban kerja manusia.

THANKS YOU