**TOPIC AREA PROPOSAL**

|  |
| --- |
| **Working title:**  **An Analysis of the use of AI within Software Engineering with the focus on SDLC**  **Investigating the intersection of Artificial Intelligence and Traditional SDLC**  **The Adoption of AI within the Software Development Life Cycle** |
| **Main theme** (A description of the general area):  Exploring the transformative potential and challenges in redefining software engineering and development practices, with a particular emphasis on its integration across each stage of the Software Development Life Cycle.  By investigating the integration of Artificial Intelligence within traditional and modern models – such as Waterfall, Agile, and DevOps. This research seeks to understand how AI can complement and enhance these frameworks without undermining established workflows.  The study will address critical challenges and proposes best practices for the seamless integration of AI, ensuring that software development processes remain efficient and effective. |
| **Outline of argument or position** (What is the case for doing this work?):  Present the case for integrating AI into established SDLC models (e.g., Waterfall, Agile, DevOps), emphasising how AI can enhance, rather than replace, existing processes. |
| **Justification of interest to others** (The topic is of importance and wide interest because…):   * Filling knowledge gaps by highlight the lack of comprehensive studies comparing AI’s effectiveness across different SDLC models and its full potential in coding, testing and deployment. * There are practical implications by conducting this research for software developers, project managers, and the industry as a whole – providing actionable insights and best practices for AI integration. * A potential for innovation and technological advancement by exploring new applications of AI in software development, potentially leading to the creation of novel tools, techniques, or methodologies. |
| **Possible research questions/issues to be explored:**   1. How can AI be integrated into traditional SDLC models (e.g., Waterfall, Agile, DevOps) without disrupting existing workflows? 2. How does the integration of AI technologies affect the efficiency and effectiveness of different SDLC models (e.g., Agile, Waterfall, DevOps)? 3. What are the best practices for integrating AI tools in the coding and development phase, and how do they affect developer productivity and software quality? 4. How do AI-based testing tools compare with traditional testing methods in identifying and addressing software bugs and vulnerabilities? 5. How does AI contribute to the maintenance phase of the SDLC, particularly in terms of predictive maintenance and automated problem resolution? 6. What are the economic implications fo adopting Ai-driven software development practices, including cost savings, ROI, and impact on market competitiveness? |
| **Objectives (SMART) that need to be completed to answer your potential questions:**  **Objective 1: Evaluate Ways to Add AI into Traditional SDLC Models**  **Specific:** Look at different ways to put AI into traditional SDLC models like Waterfall, Agile, and DevOps, to see how they can fit in without messing up what's already there.  **Measurable:** Score the level of disruption and how easy it is to add AI, from 1 to 10, based on what software development teams say.  **Achievable:** Look at stories from teams who've tried adding AI and ask them questions or get them to fill out surveys.  **Relevant:** This goal helps us understand how to mix AI into software making without causing trouble.  **Time-bound:** Finish this look-see within the first 6 months.  **Objective 2: Check How AI Changes SDLC Efficiency and Effectiveness**  **Specific:** Study the effect of adding AI on how well and efficiently different SDLC models work.  **Measurable:** Use performance indicators such as development time, bug rates, and deployment success rates before and after AI integration.  **Achievable:** Gather quantitative data from software projects that have incorporated AI and perform statistical analysis to evaluate impact.  **Relevant:** This goal is to quantify the benefits and potential drawbacks of AI in software development -- figuring out if AI really makes software development better and faster.  **Time-bound:** Aim to wrap this up within 9 months from the start.  **Objective 3: Find the Best Ways to Use AI in Coding and Development**  **Specific:** Determine best practices for integrating AI tools in the coding and development phases of the SDLC.  **Measurable:** Measure how productive developers are and the quality of the software before and after using AI tools, using things like how quick coding is done and how many bugs are found.  **Achievable:** Do case studies and talk to experts to compile a list of best practices.  **Relevant:** This goal addresses how AI tools can enhance coding and development processes.  **Time-bound:** Finish this within the first 12 months.  **Objective 4: Compare AI Testing Tools with Traditional Methods**  **Specific:** Compare AI testing tools side by side with traditional methods in terms of their effectiveness in identifying and resolving bugs and vulnerabilities.  **Measurable:** Check how many bugs are caught, the rate of false alarms, and how quickly bugs are fixed with both methods.  **Achievable:** Use a controlled study with several software projects to do the comparison.  **Relevant:** This goal is about seeing if AI can improve testing in making software.  **Time-bound:** Get this done within 12 months from starting.  **Objective 5: Evaluate AI's Role in SDLC Maintenance**  **Specific:** Explore AI's role in keeping software running smoothly, especially in predicting and fixing issues before they become big problems.  **Measurable:** Work out how much downtime and maintenance costs go down with AI help.  **Achievable:** Compare current data from projects using AI for maintenance with old data from before they started using AI.  **Relevant:** This goal focuses on AI's benefit in ongoing software maintenance.  **Time-bound:** Plan to finish within 12 months of kicking off.  **Objective 6: Study the Money Side of Using AI in Software Development**  **Specific:** Examine the economic implications of adopting AI-driven software development practices, focusing on cost savings, ROI, and market competitiveness.  **Measurable:** Calculate cost savings and ROI from AI adoption in case study projects and ask companies how it's changed their place in the market.  **Achievable:** Gather financial and business info from a sample of companies that have brought AI into their development work.  **Relevant:** This goal is to addresses the broader economic impact of AI in software development.  **Time-bound:** Aim to have this analysis done within 12months from the beginning. |
| **key words or phrases for use in an online search:**  Keywords identified in the previous tutorial:   * AI in Software Engineering * AI Integration in SDLC * AI and Agile Methodology * AI in DevOps * AI for Code Generation * Automated Testing with AI * Predictive Maintenance AI * AI Impact on Software Quality * Economic Benefits of AI in Development * AI and Software Development Efficiency   New keywords:   * AI-Driven Development Practices * AI Tools for Coding Productivity * Best Practices for AI in SDLC * AI in Continuous Integration and Deployment * Machine Learning in Bug Detection * AI for Software Project Management * Cost-Effectiveness of AI Automation * ROI of AI in Software Projects * AI in Reducing Software Development Time * Case Studies on AI in Software Engineering * Comparative analysis of AI in Agile vs. Waterfall * Integrating AI with traditional software development workflows * Impact of AI on developer productivity and software quality * Effectiveness of AI-based vs. traditional testing methods * AI's role in predictive maintenance for software systems * Economic implications of adopting AI-driven software development * Challenges in AI integration into SDLC models * Success stories of AI in coding and development * Machine learning applications in software testing * Best practices for embedding AI in software development processes |
| **Required resources and issues of access (Hardware, software, target population and subjects):**  **Hardware:**   * **High-Performance Computers:** For running AI models and simulations, high-performance computing resources are essential. This may include access to servers or cloud-based computing platforms capable of handling large datasets and intensive computing tasks. * **Access Issues:** High costs associated with powerful computing hardware or cloud services fees. Solutions may include seeking university resources, grants, or partnerships with companies that can provide the necessary computing power.   **Software:**   * **AI and Machine Learning Libraries:** Tools and libraries such as TensorFlow, PyTorch, Scikit-learn for developing and testing AI models. * **AI Applications:** Applications such as ChatGPT-4, DALL-E, DeepCode for bug detection and code review, GitHub Copoilot and others for automated code generation and completion. As well as, Jira Software with Atlassian’s Project Central and ClickUp for project management and planning. * **Development Tools:** Integrated Development Environments (IDEs) like Visual Studio Code, JetBrains PyCharm, and others for coding. Additionally, software for version control (e.g., Git) and project management tools. * **Access Issues:** Licensing fees for professional versions of software or IDEs. However, many AI libraries and development tools have free, open-source versions that are widely used in academic research.   **Target Population and Subjects:**   * **Software Development Teams**: For surveys, interviews, and case studies to understand current practices, challenges, and perspectives on AI integration in SDLC. * **Software Projects:** Access to real-world software projects that have integrated AI at various stages of the SDLC for case studies and analysis. * **Access Issues:** Gaining access to software development teams and projects may be challenging due to confidentiality concerns, proprietary information, or lack of interest. Building networks through academic and professional contacts, attending industry conferences, and leveraging alumni networks can help overcome these barriers. |
| **Ethical considerations:**  **Bias and Fairness:**  **Mitigating Bias:** AI models can inherit or amplify biases present in their training data, leading to unfair outcomes or discrimination. It’s crucial to examine how bias can be identified and mitigated in AI applications used throughout the SDLC.  **Ensuring Fairness:** Strategies to ensure AI systems are fair and equitable, providing equal opportunities and outcomes for all users, regardless of their background.  **Privacy and Data Protection:**  **Data Privacy:** AI systems often require large datasets, raising concerns about user privacy and data protection. It's important to consider how personal data is collected, used, and stored, adhering to regulations like GDPR.  **Consent and Transparency:** Ensuring that data is collected with informed consent and that individuals understand how their data will be used by AI systems.  **Accountability and Responsibility:**  **Clear Accountability:** Determining who is responsible for the decisions made by AI systems, especially in critical phases of the SDLC where errors can have significant consequences.  **Error Handling and Redress:** Establishing mechanisms for identifying errors in AI-driven processes and providing avenues for redress for those affected by incorrect decisions.  **Security:**  **AI Security:** AI systems can be targets for malicious attacks that manipulate their behavior. Ensuring the security of AI applications within the SDLC is essential to prevent data breaches or compromised software integrity.  **Robustness and Reliability:** Ensuring AI systems are robust against manipulation and reliable in their operation, especially in critical applications.  **Transparency and Explainability:**  **Understanding AI Decisions:** AI systems, particularly those based on deep learning, can be opaque. Promoting transparency and explainability in AI decisions helps stakeholders understand how outcomes are derived.  **Documentation and Communication:** Providing clear documentation and communication about how AI systems operate, their limitations, and their decision-making processes.  **Impact on Employment:**  **Changing Job Landscape:** The integration of AI in software development could automate tasks traditionally performed by humans, impacting employment. Consideration should be given to how these changes are managed and communicated.  **Skill Development and Training:** Addressing the need for upskilling and training for software development professionals to work effectively with AI technologies.  **Ethical Development and Deployment:**  **Ethical Guidelines:** Adhering to ethical guidelines and standards for the development and deployment of AI systems, ensuring they are designed with societal values and norms in mind.  Stakeholder Engagement: Engaging with a broad range of stakeholders, including end-users, to understand their concerns and expectations regarding AI in software development. |