A HYPOTHESIS-DRIVEN AND MODEL-BASED INVESTIGATION IN THE INDIAN SDLC CONTEXT

Santhi Theja

Head of Department BCA, SEA College of Science, Commerce and Arts

Shitij Chatterjee

6th Sem BCA, SEA College of Science, Commerce and Arts

DOI: doi.org/10.34293/seamjr-v1i4-003

Abstract

The quick addition of Generative Artificial Intelligence into software engineering has changed the way we develop software today by automating tasks like writing code making documents and testing. These abilities make developers work efficiently. However, because we mostly use these tools we have found some problems with software quality, governance and understanding how things work. also people are not using their judgment as much as they should. These issues are especially bad in India where a lot of people work on software development companies focus on providing services. The government is watching more closely. This paper talks about Collaborative Intelligence, on what way for humans and Artificial Intelligence systems to work together to make decisions and create things throughout the entire process of making software. The study makes a framework for Collaborative Intelligence based on theory. It looks at things like how much humans and Artificial Intelligence work together how much humans are mature the governance is and how much developers trust the system. It also makes some guesses that can be tested about how these things affect how much work gets done, the quality of the software and how well the system is adopted. Generative Artificial Intelligence helps with tasks. Collaborative Intelligence helps humans and Artificial Intelligence systems work together. This is a change in the way we make software, and it will help us make better software in the future with Generative Artificial Intelligence and Collaborative Intelligence.A mixed-methods longitudinal research design is proposed to empirically evaluate the framework within Indian software organisations. By repositioning AI from an automation tool to a collaborative partner, this work provides a governance-aware and ethically grounded pathway for sustainable AI-enabled software engineering.

Keywords: *Collaborative Intelligence, Generative AI, Software Development Life Cycle, Human–AI Collaboration, AI Governance, Indian Software Industry.*

Why we Need to Rethink AI in Software Teams

Generative AI is changing the way we do things in software engineering fast. It is doing tasks that we used to do by hand like writing code finding mistakes and making tests faster than we can. In India big companies and small startups are all using Generative AI. They are doing this because they need to stay competitive keep their costs low and meet the expectations of their clients. But there is a problem: most teams are just using Generative AI without thinking about how it will work. This is starting to cause problems.

Now we are using Generative AI like a helper that makes our work easier. We are not really treating it like a member of our team. This might sound okay. It is causing some issues. We are not sure who is responsible when things go wrong we are trusting Generative AI much we are not being transparent and we are taking some big risks. This is especially true for areas like money, healthcare and government projects. Indian companies are making software with Generative AI for these areas, so these are not just problems they are real issues that we face every day. Generative AI is being used to help with these tasks. We need to be careful and think about how we are using Generative AI

Our argument is simple: you can't get lasting value from AI by just automating bits and pieces. We need a bigger shift—toward what we call Collaborative Intelligence (CI). That means designing structured, thoughtful collaboration between humans and AI into every part of the software development life cycle (SDLC). In this paper, we aim to: define CI as a formal way of thinking about human–AI teamwork in software, propose a CI-driven SDLC framework that fits the Indian context, and build testable models to see how CI affects productivity, quality, and governance.

What's Already Been Done (and what's missing)

Most research on Artificial Intelligence, in software engineering focuses on tasks. These tasks include code completion. They also include predicting bugs and auto-generating tests. Studies do show that using AI in software engineering can increase productivity. But they also having real problems like ("hallucinations"), developers losing deep understanding of their code, and people trusting AI suggestions.

Research on human–AI interaction points to something called *automation bias*-basically, we humans tend to trust algorithms even when they're clearly wrong. The good news is that keeping a human in the loop helps avoid these traps, especially in high-risk situations, because it preserves human judgment and accountability. But most studies stop at single tasks. They don't look at redesigning the entire software development process from end to end.

In India people have talked a lot about how the workforce's changing and how we need to retrain people and what this means for the economy. We do not have any systems in place that bring together people working together good governance and trust when it comes to using Artificial Intelligence in the way we develop software. The thing that is missing is something that combines all these things that is what we are trying to do. We are trying to fill this gap by talking about Collaborative Intelligence as a way to bring everything together a way that's good for everyone and lasts a long time a sustainable approach, to Collaborative Intelligence.

What is Collaborative Intelligence?

Here's How we Define CI for this Study

A way of working where humans and AI agents jointly create, review, and manage software artifacts—using shared context, clear role boundaries, continuous feedback loops, and traceable decisions.

Unlike approaches that focus on replacing humans with AI, CI is about complementary. It recognises that good software outcomes emerge from the interaction between technical tools and human expertise.

Core Principles of Collaborative Intelligence

First, role complementary – humans keep final say on judgment-heavy, ethical, and strategic calls, while AI handles scaling, pattern finding, and synthesis. Second, context persistence – AI works within real-world constraints by tapping into shared knowledge and project memory. Third, provenance and easy understanding- every AI-generated artifact should be traceable back to its source, the model used, and who approved it. Finally, continuous learning – insights from deployments and post-incident reviews feed back into both how the team works and how the models are improved.

A Collaborative Intelligence–Driven SDLC

Our proposed CI-driven SDLC rethinks responsibilities across six interconnected phases:

Requirements Engineering

AI helps pull together stakeholder inputs, spot inconsistencies, and suggest alternative interpretations. But humans still validate business intent, ethics, and feasibility.

Architecture and Design

AI tools generate design options and trade-off analyses (cost vs. performance vs. scalability). Human architects make the final call and ensure security and regulatory compliance.

Implementation

Developers and AI work side by side—AI handles boilerplate code, refactoring

suggestions, and static analysis. Any safety critical or domain sensitive code gets mandatory human review.

Testing and Quality Assurance

AI generates thorough test suites, including edge cases and adversarial tests. Human testers focus on exploratory testing, usability, and system level behavior.

Deployment and Operations

AI assists with release analysis, monitoring, and anomaly detection. Human operators approve deployments, manage incidents, and own operational accountability.

Continuous Learning

Post mortems combine AI generated root cause hypotheses with human led organizational learning, so both processes and models keep getting better.

Our Research Model and Hypotheses

Key Concepts we'll Measure

- CI Intensity (CII) – how deeply structured human–AI collaboration is embedded
- Human Oversight Ratio (HOR) – what proportion of AI outputs gets reviewed by humans.
- Governance Maturity (GM) – strength of traceability, auditability, and controls
- Developer Trust (DT) – how much developers trust that the AI is reliable and appropriate
- SDLC Productivity (SP) – effective output per unit of time
- Software Quality (SQ) – how well defects are caught before release

What We Expect to Find (hypotheses)

- CI driven SDLCs will be more productive than those using GenAI alone.
- Beyond a certain point, adding more CI intensity yields smaller productivity gains (diminishing returns).
- CI driven SDLCs will have lower defect escape rates.
- Human oversight changes the relationship between AI usage and software quality.
- Stronger governance maturity leads to higher developer trust.
- Developer trust is a key factor that keeps teams using AI over the long term.
- If you automate too much without enough oversight, defect risk follows a U shaped curve.

How we'll Analyze the Data

We model SDLC productivity as a non linear function of CI intensity, oversight, and governance:

$$SP = \alpha + \beta_1 \cdot CII + \beta_2 \cdot CII^2 + \beta_3 \cdot HOR + \beta_4 \cdot GM + \varepsilon$$

The probability of a defect escaping to production follows a logistic curve, factoring in automation level, oversight, and governance:

P(D_escape) = 1 / (1 + e^–(γ□ + γ□·AI – γ□·HOR – γ□·GM))

Developer trust will be treated as a mediating variable using structural equation modelling.

How we'll Run the Study

We're using a mixed-methods, longitudinal design across several Indian software organisations—startups, mid-sized companies, and large enterprises. Quantitative data will come from code repositories, issue trackers, and quality metrics. Qualitative insights will come from semi-structured interviews and analysis of actual work artifacts.

What we Expect to Contribute

On the theory side, this work aims to put Collaborative Intelligence on the map as a foundational concept in software engineering research, and to introduce non-linear models of automation risk. On the practical side, we'll offer evidence-based guidance for redesigning SDLCs and integrating AI with governance in mind—especially useful for regulated Indian industries like banking, insurance, and healthcare.

Limitations

Our initial validation will be on a pilot scale. To generalize broadly across different domains and geographies, we'll need longer-term, larger-scale studies.

Conclusion

Collaborative Intelligence is not a small change. It is a major change in the way we include AI in software engineering. Changing from automation to organized teamwork helps Indian software companies get long lasting improvements in productivity without losing quality, trust or responsibility. This study shows a theory and real world proof, for the next level of software development powered by AI. One where people and AI really work together as partners.

Acknowledgement

This paper was presented at the Two-Day National Level Seminar on "Realising Sustainable Development Goals in the Indian Context" held at S.E.A. College of Commerce and Arts on 5th -6th March 2026.

References

A. Agrawal, J. Gans, and A. Goldfarb, *Prediction Machines*, 2018.

D. Amershi et al., "Guidelines for Human–AI Interaction," CHI 2019.

L. Bao, M. Li, and A. van Deursen, "Human–AI Collaboration in Software Engineering," *ACM Computing Surveys*, 2023.

T. M. Mitchell, *Machine Learning*, 1997.

B. Friedman and H. Nissenbaum, "Bias in Computer Systems," *ACM TOIS*, 1996.

M. C. Paulk, “A History of the Capability Maturity Model,” *Software Quality Professional*, 2009.

S. S. Shapiro, “Human-in-the-Loop Machine Learning,” *AI Magazine*, 2020.

N. J. Nilsson, *The Quest for Artificial Intelligence*, 2010.

D. Kahneman, *Thinking, Fast and Slow*, 2011.

I. Sommerville, *Software Engineering*, 10th ed., 2016.