PakTech Today | Pakistan's Premier Technology Magazine
Vol. 01, Issue 08 • June 2026

TECHNOLOGY FEATURE Industrial News / Technology Report / Research Communication

The Aimless Dive Problem: Why Surface-Level Research Habits Undermine Innovation in Software Engineering

When an idea strikes, students aimlessly dive. They rush to open the IDE, paste AI-generated code, and ship the project, never pausing to ask: how does this work, and why? When someone finally does ask, the engineer is exposed—confident on the surface, but hollow in depth. In Pakistan's software engineering industry, this aimless dive is silently costing us the one thing we cannot afford to lose: innovation.

By **Haidar Ali** | University of Agriculture, Faisalabad | Published: June, 2026
mughalhaider08@gmail.com | Position: Software Engineering Student

Introduction

The pattern across our universities is the same. An assignment comes, a deadline approaches, and the first instinct is to paste the question directly into ChatGPT or Gemini, without understanding the real problem, and submit the result without even reviewing it. This habit is so normalized that students, and even some teachers, do not recognize it as the core threat to a young innovation mindset.

Why does this happen? Because students have shaped their thinking around one dangerous assumption: why spend hours understanding a problem when AI explains it in seconds? Why read documentation when AI generates code from a single idea? Why research deeply to understand what already exists and what gaps remain? The tool was built to assist engineers and sharpen their thinking, not to replace their thinking entirely.

The damage is not visible on a resume, a portfolio, or in projects built through these practices. It shows up later ,in the startup that cannot scale its own product, in the developer who freezes when AI gives a wrong answer and has no foundation to catch it, in the innovation that never happens because no one was willing to research deeply enough. Pakistan's tech industry is growing rapidly — but real growth cannot be built on a hollow research culture.

> ***" True innovation in software engineering is not born at the keyboard. It is born in the hours of deep inquiry that happen before the keyboard is ever touched."***

Technology Overview

Surface-level research looks like an effort-based research, but this effort pointed in the wrong direction. When students use LLMS to generate the code without evaluating underlying architecture impact. They just focus on shortcuts ignoring edge cases, validation or security.

While deep research in Software engineering is a most crucial part that formulated the investigation the systematic approach, understanding the core problem and propose a sophisticated engineering solution which makes to grind on problem solving skill, which is the actual and most fundamental skill many beginners ignored. A student wants to build a login system in react. He puts his goal in CHATGPT to build me the login system with code, he copies the code and it works, that's it. He is not inquiring what's behind the system? What security consequences we face with this approach? What the actual documentation actually recommend? How do we scale it in future. Although, he searched but Engineering mindset is lost.

Another example from previous semester project. We were asked to build Sentiment Analysis Project .we had a document having instructions written ,just copied the document and paste into AI model to develop the project with code, then pasted it into Google Colab ,by trial and error , prompting Chatgpt repeatedly until the code ran and submitted online. In the whole scenario, I observed, there was no understanding because of deadline, students wanted to build quickly without understanding what code actually did. The code runs but understanding zero.

Contrast with true engineering mindset one who actually reads the original BERT research paper, understand why transformer-based models outperformed older approaches, asks what training data their chosen model was built on and then decide whether this fits their usecase or something better available that builds by another Engineer.

AI makes the surface level research fast. For a senior engineer the practice works, because he is familiar with deep context, verifies every output against official document, validates with research papers, and challenges the models assumption by counter questioning.

But for the student build this instinct from the beginning, AI does the opposite. The code runs the feature is implemented successfully but there is no verification, there is no questioning and no understanding at all.

The tool was built to improve the thinking,not to replace the awareness that thinking needs to happen at all.

Key Findings & Impact

The numbers are harder to ignore. A 2023 GitHub survey found that while 92% of developers are already using AI coding tools inside and outside of work, over 70% admitted they do not fully verify AI-generated code before shipping it. **[2]** these stats do not show the productivity intent instead this is the risk factor, specially in Pakistan, where fresh graduates enter the industry already blindly trust on AI's output without challenging it because they do not know what actually behind the scene.

The Stack Overflow Developer Survey 2024 reported that developers who rely heavily on AI assistance without foundational research depth are significantly more likely to introduce security vulnerabilities and architectural flaws in production systems. **[3]** that is the key when enterprise system fails under load, so they can't scale the startup and the security breaches exposes what no-body thought to inquire during development.

The impact of this problem extends well beyond academics. For Pakistan specifically, the implications cut deeper than individual careers. PSEB reported that Pakistan's IT exports crossed 2.6 billion dollars in 2023, with ambitions to reach 5 billion by 2026. **[1]** Software engineers requires t build the system that last --not just features that demo well. Every developer who ships the AI-generated code without deep understanding ,It is not just killing their own growth .they are quietly lowering the standards on what Pakistan's tech Industry actually produce and export globally.

What's Next?

The solution does not start in the final years. It starts on the very first day of the university. Students must be aware the facts, what is going on behind the screen, before writing a single line of code. They should first brainstorm the problem. What is actually being asked? Why does this problem come and why should solutions exist? What has already been made before, if it exists, then how can it improve?

There is a study that suggests three steps before going right into IDE. First, describe the problem in your own writing without touching any AI-trained model. Second, spend some time to real investigation what already exists through technical research papers and real documentation. Third, find the gap, what does your proposed solution exactly do that the existing do not. This makes the mindset of the student align with the engineering mindset that actually builds the things what the industry needs **[4]**

It should be the student's habit to read the documentation, study research papers, evaluate existing solutions, and go deep to understand others' code. This is not an academic formality; it is the actual beginning of innovation in software engineering that lacks through weak practices. True innovation is not something entirely new. It is overlapping ideas that exist before in a structured way, combined by an engineer who understood the underlying details that others missed. The industry is talking about shifting from valuing lines of code to demanding clarity of plan [6]. Engineers should be curious enough to challenge the world, not just focused on the lines of code, but the real systems and clarity behind them.

The next industrial revolution in Software will not be ignited by the lines of code.it will be ignited by the engineers who finally learned to ask why before they asked how.

About the Author

Haidar Ali is a software engineering student with keen interest in software engineering culture, research, and move forward to Pakistan's growing tech industry. He can be reached at mughalhaider08@gmail.com

Keywords: Artificial Intelligence; Surface-level research; Software Engineering; Deep Research, Innovation in Software engineering, Pakistan Tech

References

[1] Pakistan Software Export Board (PSEB) (2023) *IT Industry Survey and Export Statistics.* Government of Pakistan. Available at: https://pseb.org.pk

[2] GitHub (2023) *The State of the Octoverse 2023.* GitHub Inc. Available at: https://octoverse.github.com

[3] Stack Overflow (2024) *Developer Survey 2024.* Stack Overflow. Available at: https://survey.stackoverflow.co/2024

[4] Prather, J., Reeves, B.N., Denny, P., Becker, B.A., Leinonen, J., Luxton-Reilly, A. and Finnie-Ansley, J. (2024) 'The Widening Gap: The Benefits and Harms of Generative AI for Novice Programmers', *Proceedings of the ACM Conference on International Computing Education Research (ICER 2024),* ACM. DOI: 10.1145/3632620.3671116 https://dl.acm.org/doi/pdf/10.1145/3632620.3671116