

Mobile GenAI: Bridging Developer Aspirations and On-Device Realities

Rojin Vishkaie, Shantu Roy, and Laurence Moroney

Arm, Seattle, WA 98101, USA

ABSTRACT

The “Generative Renaissance” is rapidly expanding to mobile platforms, promising to redefine user experiences on Android. However, a significant gap exists between developer aspirations for on-device Generative AI (GenAI) and the perceived readiness of the current ecosystem. This paper presents findings from a study of 39 experienced Android developers surveyed at Droidcon NYC 2025. Our results reveal a critical “trust deficit” rooted in concerns over performance, reliability, and security. While developers are actively using GenAI for workflow productivity, they are hesitant to ship user-facing features. We find that developers prioritize robust, secure tooling and transparent performance benchmarks over novel capabilities alone. This study provides a framework for understanding the key barriers to adoption and offers a clear directive for platform and hardware vendors: building developer trust is the essential catalyst for unlocking the true potential of on-device GenAI on Android.

Keywords: Generative AI, Developer experience, Mobile graphics, Trust, Android

INTRODUCTION

Generative AI is poised to be the next major shift in mobile computing, with the potential to create highly personalized user experiences, dynamic game content, and intelligent, interactive applications. While cloud-based GenAI is now commonplace, the frontier of innovation lies in bringing these capabilities on-device to improve latency, enhance privacy, and enable real-time interaction.

However, despite the excitement, the integration of sophisticated GenAI into mainstream Android applications remains nascent. This raises a critical question: What are the primary barriers preventing experienced Android developers from fully embracing on-device GenAI? This paper investigates this question by analyzing the perceptions, aspirations, and concerns of the developer community. We explore the gap between the features developers want to build and what they believe is possible today, identifying trust as the central, unifying challenge.

METHODOLOGY

We conducted an in-person survey with 39 Android developers at the Droidcon NYC conference in June 2025. The participants represent a highly experienced cohort, with 56% (22 of 39) reporting over five years of

professional Android development experience. The majority (51%) focus on Productivity/Business applications, with others in Gaming, Social Media, and Education. The survey, designed for a 25–30 minute completion, combined multiple-choice questions, Likert scales, and open-ended prompts to capture both quantitative trends and qualitative insights. The data was anonymized and analyzed to identify dominant themes and patterns.

FINDINGS

Our analysis reveals four key findings that characterize the current state of on-device GenAI from the developer's perspective.

The Duality of Adoption: A Workflow Tool, Not Yet a Product Feature

There is a clear distinction in how developers use GenAI today. A significant majority (64%) use GenAI tools at least weekly in their development workflow. The most common uses are for productivity enhancement, such as code generation (26 mentions) and brainstorming ideas (21 mentions). This aligns with industry reports on the utility of AI assistants like GitHub Copilot (Goel, n.d.). However, this enthusiasm does not yet translate to shipping on-device GenAI features to end-users, indicating a barrier between personal use and product integration.

The Aspiration-Reality Gap

Developers see immense value in future on-device GenAI capabilities. Features enabling Personalized User Experiences and creating Intelligent Non-Player Characters (NPCs) were rated as highly valuable (4 or 5 out of 5) by 59% and 56% of respondents, respectively.

However, there is a stark contrast between this aspiration and their perception of current device capabilities. The most desired features are also seen as the least feasible. For example, “Sustained, complex GenAI workloads” and “Real-time generation of complex 3D assets” were rated as having low capability on current high-end devices. This gap is visualized in Table 1.

Table 1: Comparison of mean developer ratings for the value of future GenAI features versus the perceived capability of current high-end android devices (1 = Low, 5 = High). Negative gaps indicate high aspiration but low perceived feasibility.

GenAI Feature	Perceived Value (Avg. Score)	Perceived Capability (Avg. Score)	Aspiration-Reality Gap
Personalized UX	4.1	2.8	−1.3
Intelligent NPCs	3.9	2.5	−1.4
Real-Time Graphics	3.8	2.4	−1.4
Simple Image Filter	3.1	4.2	+1.1

The Trust Deficit: Performance, Predictability, and Privacy

The reluctance to ship on-device GenAI features is rooted in a significant “trust deficit.” When asked about their biggest concerns, developers’ top three answers were not about the novelty of the technology, but about fundamental implementation challenges:

- Insufficient Device Performance (22 mentions)
- Excessive Battery Consumption (21 mentions)
- Unpredictable/Inconsistent Outputs (Hallucinations) (20 mentions)

These concerns are compounded by deep-seated worries about security and privacy. “Security vulnerabilities” (19 mentions) and “Data privacy concerns” (16 mentions) were also top-tier issues.

This aligns with broader industry discussions about the risks of AI-generated code and the “black box” nature of many models (Dius, n.d.; TalentElgia, n.d.). This focus on fundamentals is further reinforced by the paramount importance of data control. A striking 92% of respondents rated “Data Sovereignty” (keeping user data on-device) as “Very” or “Extremely” important.

The Path to Trust: A Call for Better Tools and Transparency

When asked what would most increase their trust, developers did not prioritize more powerful AI models. Instead, they overwhelmingly pointed to the need for a stronger support ecosystem. The top trust-building factors were:

- Stronger, well-documented security assurances (23 mentions)
- More transparency into how models work (19 mentions)
- Robust and easy-to-use SDKs/tools (18 mentions)

This desire for better tooling was echoed in their preference for a hybrid integration model, with 95% finding GenAI features in their IDE (e.g., Android Studio) helpful. This indicates a strong demand for solutions that are not just powerful, but also reliable, understandable, and seamlessly integrated into their existing workflows.

DISCUSSIONS AND IMPLICATIONS

The findings from our study suggest that for the “Generative Renaissance” to fully flourish on Android, the focus must shift from demonstrating novel capabilities to building a foundation of developer trust. Developers, particularly the experienced professionals in our sample, are pragmatic. They embrace GenAI for productivity but are cautious gatekeepers of the user experience, unwilling to ship features that are slow, power-hungry, unpredictable, or insecure (Appventurez, n.d.).

This presents a clear call to action for platform owners and hardware designers like Google and Arm. The “want vs. possible” gap in areas like real-time graphics and intelligent agents highlights a significant market

opportunity. However, to capture it, the primary developer concerns must be addressed. This involves:

- **Engineering for Trust:** Prioritizing power-efficient performance, predictable model behavior (PYMNTS.com, n.d.), and verifiable security is paramount.
- **Enabling Through Tooling:** Developers need more than just APIs; they need robust, well-documented SDKs, transparent performance benchmarks, and deep integration into tools like Android Studio (Arm, n.d.).
- **Ecosystem Advocacy:** When asked what companies' message should carry to platform owners, a dominant theme emerged: the need for standardized, low-level APIs that provide more direct control over hardware schedulers for predictable performance and power management of AI workloads.

CONCLUSION

Android developers are not resistant to Generative AI; they are waiting for it to become trustworthy and practical for on-device deployment. Their current adoption for workflow tasks shows a clear appetite, but their concerns about performance, reliability, and security are significant barriers to creating the next generation of intelligent mobile applications. The path forward lies not in a race for the most complex models, but in a collaborative effort by hardware and software leaders to provide a transparent, secure, and efficient foundation upon which developers can confidently build. By addressing this trust deficit, we can bridge the gap between aspiration and reality, truly ushering in the era of the Generative Renaissance on mobile.

ACKNOWLEDGMENT

We thank Joe Alderson and Sharbani Roy for their invaluable advice and support.

REFERENCES

- Appventurez. (n.d.). *10 Challenges in Android App Development*. [Online]. Available at: <https://appventurez.com>.
- Arm. (n.d.). *Arm Performance Studio for Mobile Optimization*. [Online]. Available at: <https://www.arm.com>.
- Dius. (n.d.). *Designing for trustworthy generative AI*. [Online]. Available at: <https://dius.com.au>.
- Goel, N. (n.d.). *AI coding tools are simultaneously terrible and necessary*. [Online]. Available at: <https://blog.giga.ai/ai-coding-tools>.
- PYMNTS.com. (n.d.). *Can Arm's Mobile Lead Translate to AI? Chip Designer Bets on Efficiency*. [Online]. Available at: <https://www.pymnts.com>.
- TalentElgia. (n.d.). *Generative AI Challenges and Their Solutions*. [Online]. Available at: <https://talentelgia.com>.