

AI Impact on Software Engineering Productivity

Executive Summary

Amirhossein Jofreh

November 2025

Abstract

This report synthesizes peer-reviewed research from IEEE and ACM publications to quantify AI's impact on software engineering tasks. Analysis reveals **efficiency gains of 10%–110%** depending on task type, with median improvements of **25–50%** across most development activities. Code generation, documentation, and testing show the highest and most consistent benefits.

Key Findings

- **Code Writing:** Developers complete coding tasks **26–55% faster** with AI assistance
- **Documentation:** Time reduced by **50%** with 7.5% quality improvement
- **Testing:** Coverage improvements of **20–110%**; test generation time reduced from 20+ minutes to seconds
- **Junior developers** benefit most from routine tasks; **senior developers** gain more in complex reviews
- Full productivity realization takes approximately **11 weeks** of tool adoption

AI Efficiency Gains by Task

Task	Min Gain	Max Gain	Median
Code Writing	10%	55.8%	26%
Code Review	15%	40%	25%
Testing	20%	110%	35%
Debugging	15%	45%	25%
Documentation	30%	60%	50%
Refactoring	25%	67%	35%
Maintenance	20%	40%	30%
Planning	5%	20%	12%

Table 1: Comprehensive AI Efficiency Gains Across Software Engineering Tasks

Visual Summary

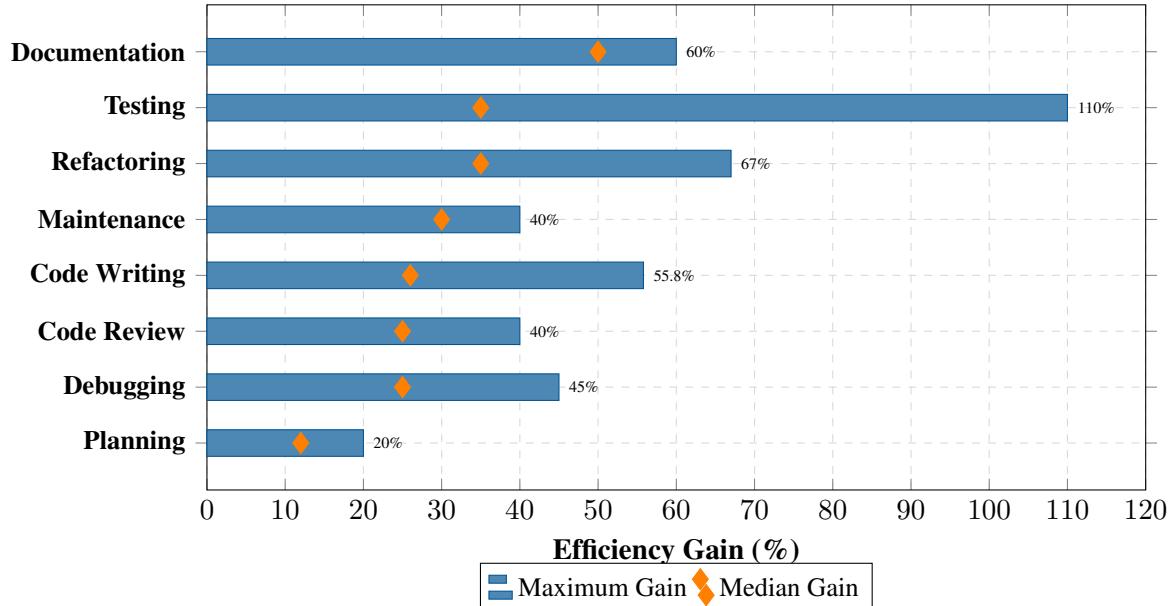


Figure 1: AI Efficiency Gains: Maximum vs. Median by Task Category

Impact by Experience Level

Task	Junior (0-2 yrs)	Mid (3-5 yrs)	Senior (6+ yrs)
Code Writing	45–55%	30–40%	20–30%
Code Review	25–35%	25–35%	30–40%
Debugging	35–45%	25–35%	15–25%
Testing	40–50%	35–45%	25–35%

Table 2: AI Efficiency Gains by Developer Experience Level

Recommendations

- **Prioritize AI deployment** for documentation, testing, and code generation (highest ROI)
- **Invest in training** — full benefits require 11 weeks of adoption
- **Maintain human oversight** — AI suggestions require verification
- **Tailor approach by experience** — junior devs need different support than seniors

Conclusion

AI tools deliver **10–110% efficiency gains** across software engineering tasks, with median improvements of **25–50%**. Documentation and testing show the most consistent benefits. Strategic deployment with proper training and human oversight positions organizations to maximize productivity while maintaining software quality.

Key Sources

- Peng et al. (2023) — GitHub Copilot Study, *arXiv*
- Meyer et al. (2021) — Developer Workdays, *IEEE TSE*
- McKinsey & Company (2023) — Generative AI Productivity Report
- DORA Report (2024) — State of DevOps
- Google RCT (2024) — Internal AI Productivity Study