# AI Coding Assistants - ANSSI, BSI

- Descrição do uso da IA
  - Geração de algoritmos
    - Simples
      - A study showed that current AI coding assistants are capable of correctly implementing such algorithms.
      - It was found that the proposed implementation suggestions did not only **have functional correctness** but also **optimal runtime performance** (Dakhel, et al., 2023).
    - The **success rate of the technology decreases for more complex tasks** but can still reach usable performance levels (Yeadon, et al., 2024).
    - AI coding assistants can be used for code completion, code adaption to specific contexts or refactoring. However, they are **limited in solving issues in real-world software engineering scenarios with large codebases** (Jimenez, et al., 2024).
  - Debugging
    - Using AI to support debugging could **reduce the workload in software development**.
    - For specific programming languages, such as JavaScript, it has already been shown that **AI models can be trained to automatically detect errors and function effectively** (Phadel, et al., 2018; Berabi, et al., 2021).
    - Furthermore, AI assistants not only detect but also **directly fix errors** (Allamanis, et al., 2021).
    - AI coding assistants have the potential to **reduce the workload related to correction and maintenance** of software, often referred to as "Technical Debt" (Popper, 2023).
  - Geração de testes
    - AI can automatically propose test cases and unit tests that also **cover edge cases** by analysing the existing code and understanding the program logic (Smith, 2024).
    - Additionally, applying test-driven development (TDD) could be simplified.
    - This way, the benefits of this software development paradigm (e.g. focus on design and functionality (Mayr, 2005), reduced debugging effort (Williams, et al., 2003) or better modularity (Madeyski, 2010))
  - Explicação do código
    - Reduce the effort required for familiarization by allowing developers to ask specific questions in natural language about the code (Bansal, et al., 2021).
  - Estilo, comentários e documentação
    - Apply **coding styles automatically** to improve the standardization, readability and maintainability of the code across larger teams.
    - Additionally, they can assist in generating automated comments and documentation blocks (Doerfeld, 2023).
  - Tradução de código
    - A study (Pan, et al., 2024) shows that **syntactic, semantic, dependency, and logical errors** occur in translations.
    - However, another study (Weisz, et al., 2022) demonstrates that **developers could also benefit from incorrect AI suggestions** when translating source code, albeit only for common languages.
- Recomendações para IA
  - AI coding assistants are **no substitute for experienced developers**.
  - A systematic risk analysis should be performed before introducing AI tools (including an assessment of the trustworthiness of providers and involved third-parties)
  - Gain in **productivity of development teams** must be compensated by appropriate **scaling measures in quality assurance teams** (AppSec, DevSecOps).
  - Generated source **code should generally be checked and reproduced** by the developers.
  - Use plain text to prompt AI coding assistants with the desired functionality of the code and relevant parameters such as input and output formats and the assistants automatically generate the required test cases.
  - The use of AI in programming tasks requires caution and awareness of the system boundaries (Kabir, et al., 2024).
- Riscos da IA
  - Dependência
    - Programmers relying too much on AI assisted tools might **loose over time the necessary skills to thoroughly evaluate and review source code**.
    - Studies show a **cognitive bias** when using AI coding assistants, as many developers perceive them as secure, although security vulnerabilities are regularly identified.
    - AI coding assistants can pose the risk of **excessive dependence and uncritical acceptance of generated code**, as even flawed solutions are well-worded (Dakhel, et al., 2023; Kabir, et al., 2024).
  - Segurança
    - An unrestrained use of the tools can have **severe security implications**.
    - It is possible that **sensitive code, login credentials, and API keys stored in public registries** could be fed into the **training** (Niu, et al., 2023).
    - Some developers even **bypass security guidelines** to use these assistants (Snyk Limited, 2023; Perry, et al., 2023).
    - New York University analysed the security of programs generated by coding assistants and found that about **40% of the programs had security vulnerabilities** (Pearce, et al., 2022).
    - One cause of these security flaws is the **use of outdated programs in the training data** of the AI models, leading to the suggestion of outdated and insecure best practices.
    - A study by Stanford University confirmed these security concerns and found that **insecure libraries were suggested even when their documentation flagged security concerns** (Perry, et al., 2023).
    - **Mitigation techniques** are getting proposed, these techniques are currently **limited in their capabilities** and **lack the generalization guarantees** that would allow them to be widely applicable (He, et al., 2023).
    - AI coding assistant generates code that references or **recommends non-existent packages or libraries**, it is known as package hallucinations. These hallucinations can lead to so-called **package confusion attacks** (Spracklen, et al., 2024).
  - Qualidade
    - Unclear prompts lead to **less useful results** (Dakhel, et al., 2023).
    - The output based on learned statistical distributions **can lead to illogical or impractical suggestions**, which only experienced developers can recognize and evaluate (Dakhel, et al., 2023; Chen, et al., 2021).
    - LLMs typically generate linguistically flawless and convincing text, which can foster an impression of human-like abilities and overconfidence in their statements (Automation Bias), **leading users to draw false conclusions** or accept unverified statements (BSI, 2024).
    - A study by Purdue University compared the solutions of a language model for coding tasks with answers from a developer forum and concluded that **the model provided incorrect answers 50% of the time** (Kabir, et al., 2024).
    - Explanations, comments or documentation generated by an AI assistant **can be incorrect or completely hallucinated**, which might lead to security issues and reduce code maintenability.
- Impressão pessoal
  - Developers **felt more productive** the more suggestions from the coding assistant they adopted (Ziegler, et al., 2022).
  - Mental **workload was reduced** using AI and employee **satisfaction increased** mainly through reducing repetitive tasks (Github, 2024).
  - No significant shortening of processing time for programming tasks through the use of an AI coding assistant but **participants generally preferred it over a traditional autocomplete** function (Vaithilingam, et al., 2022).
  - The **replacement of a laborious online search** was highlighted as an advantage by participants.
  - AI coding assistants can have a **positive impact on employee satisfaction**.
  - Currently, there is a **lack of conclusive evidence regarding productivity enhancement**.
- Acadêmico
  - Foco em assistentes de código de AI para desenvolvimento de **software profissional**
  - Developers dedicate more than 17 working hours per week to debugging (Stripe, 2018).
  - TDD
    - Benefícios
      - Focus on design and functionality (Mayr, 2005)
      - Reduced debugging effort (Williams, et al., 2003)
      - Better modularity (Madeyski, 2010)

# Using AI-based coding assistants in practice: State of affairs, perceptions, and ways forward - ELSEVIER

- Acadêmico
- Descrição do uso da IA
  - Principais usos
    - 1. Escrita de artefatos em linguagem natural
    - 2. Escrever testes
    - 3. Refatoração
    - 4. Triagem de bugs
    - 5. Implementar novas funções
  - **Takeaway 1: The overall opinion regarding code provided by AI assistants differs, with the most positive aspect being its usability and the most negative being its security.**
- Ferramentas de IA
  - Fig. 1. The main tools used for the comparison.
  - Pesquisa das ferramentas mais utilizadas
    - 1. ChatGPT
    - 2. GitHub Copilot
    - 3. JetBrains AI Assistant
  - 84,2% do público utiliza
- Público
  - Pesquisa fechada
  - 48,9% dos desenvolvedores com 10 ou mais anos de experiência
  - 74.4% empregado incorporações e organizações
- Impressão pessoal
  - Fig. 2. The general opinion of respondents about each of the five main activities.
  - Atividades preferidas
  - **Takeaway 2. Among the main activities studied, implementing new features is the most enjoyable and the least likely to be delegated to an assistant, while writing tests and writing natural-language artifacts are the most unpleasant and the most likely to be delegated.**
  - Analisa em detalhe as impressões dos desenvolvedores quanto a diferentes usos da IA
  - Fig. 3. General opinions of respondents about code provided by AI assistants. NFR stands for non-functional requirements.

# Categorias

- Recomendações para IA
- Impressão pessoal
- Riscos da IA
- Uso histórico
- Público
- Ferramentas de IA
- Descrição do uso da IA
- Métricas de código
- Pessoal
- Acadêmico
- Empresarial

# AI Copilot Code Quality - GitClear

- Uso histórico
- Empresarial
- Acadêmico
  - Código duplicado é propenso a bugs
    - "57.1% of all co-changed clones are involved in bugs"
  - DORA - Artificial intelligence: Adoption and attitudes
- Métricas de código
  - 2024
    - linhas copiadas > linhas movidas
    - 8x mais blocos de código com 5+ linhas duplicadas
    - Redução dos indicadores de refactor (linhas movidas)

| | Added | Deleted | Updated | Moved | Copy/pasted | Find/replaced | Churn |
|---|---|---|---|---|---|---|---|
| 2020 | 39.2% | 19.1% | 5.2% | 24.1% | 8.3% | 2.9% | 3.1% |
| 2021 | 39.5% | 19.3% | 5.0% | 24.8% | 8.4% | 3.4% | 3.3% |
| 2022 | 40.9% | 19.8% | 5.2% | 20.8% | 9.4% | 3.7% | 3.3% |
| 2023 | 42.3% | 21.1% | 5.6% | 16.8% | 10.6% | 3.8% | 4.1% |
| 2024 Projected | 43.6% | 22.1% | 5.8% | 13.4% | 11.6% | 3.8% | 7.1% |
| **2024 Actual** | **46.2%** | **21.9%** | **5.9%** | **9.5%** | **12.3%** | **4.2%** | **5.7%** |
| **YoY change** | +9.2% | +3.8% | +7.2 | **-38.9%** | **+17.1%** | +16.7% | **+39%** |

| Year | Commits scanned | Total dupe blocks found | Commits containing dupe block | Duplicate block % | Median dupe block size |
|---|---|---|---|---|---|
| 2020 | 19,805 | 9,227 | 139 | 0.70% | 10 |
| 2021 | 29,912 | 9,295 | 143 | 0.48% | 11 |
| 2022 | 40,010 | 10,685 | 182 | 0.45% | 11 |
| 2023 | 41,561 | 20,448 | 747 | 1.80% | 10 |
| 2024 | 56,495 | 63,566 | 3,764 | 6.66% | 10 |

  - Aumento da consistência de estilo do código
- Recomendações para IA
  - Estabelecer métricas que avaliem e limitem o custo de manutenção de longo prazo do código
  - Garantir que o uso de IA não resulte em sistemas mais complexos e difíceis de sustentar com o tempo.
- Riscos da IA
  - Google DORA avalia que o crescimento da adoção do uso de IA em 25%
    - -7.2% delivery stability
    - -1.5% delivery throughput
  - Ampliar a quantidade de linhas que precisarão de manutenção contínua.
  - Janela de contexto insuficiente para reaproveitar código
- Impressão pessoal
  - Perceptions of productivity changes due to AI
  - Extremely increased my productivity
  - Moderately increased my productivity
  - Slightly increased my productivity
  - No impact on my productivity
  - Slightly decreased my productivity
  - Moderately decreased my productivity
  - Extremely decreased my productivity
  - Percentage of respondents
  - Error bar represents 95% uncertainty interval
  - Figure 4. Respondents' perceptions of AI's impacts on their productivity.
  - DORA 2025 — 22 — Artificial intelligence: Adoption and attitudes

# Magic Quadrant for AI Code Assistants - Gartner

- Empresarial
  - Previsões de uso da IA
    - 5% de participação da IA no SDLC em 2024 -> 40% em 2027
    - 80% das empresas usarão ferramentas de teste com IA em 2027
    - 90% das empresas de software usarão AI code assistants em 2028
    - 25% dos bugs em produção virão de código de IA não revisado por humanos
  - Otimismo quanto à
    - Eficiência
    - Velocidade
    - Sobrecarga cognitiva
    - Resolução de problemas
    - Aprendizado
    - Criatividade
- Ferramentas de IA
  - Descrição do uso da IA
    - Explicação do código
      - Desenvolvedores usarão os assistentes para compreender dependências complexas pelo programa para reduzir a dívida técnica e modernizar o código
    - Modernização de código
    - Depuração de código
    - Geração de código
    - Criação de artefatos e testes
  - Necessidades de mercado
    - Chat de conversa integrado ao ambiente de desenvolvimento
    - Garantir que o modelo não será treinado com o código ou documentação do cliente
    - Possibilidade de usar o assistente de código para diferentes ecossistemas dos clientes
    - Completar códigos com contextos multilinha
    - Completar código usando linguagem natural
  - Principais ferramentas
    - GitLab
    - Amazon Web Services
    - Google Cloud
    - GitHub

# AI 2024 Developer Survey - SO

- Pessoal
- Uso histórico
  - Paralelo com pesquisas anteriores
    - Mostrar mudança na percepção
    - Acompanhar mudança de uso
- Impressão pessoal
  - AI tool sentiment
    - 72% é favorável ou muito favorável ao uso de IA
  - Accuracy of AI tools
    - 43% confia na precisão da IA e 30,4% não confia
  - AI tools' ability to handle complex tasks
    - 43,2% acredita que a IA é ruim ou péssima lidando com tarefas complexas
  - Are AI tools a threat to your job
    - 68,3% acredita que a IA não é uma ameaça a seus trabalhos
  - AI Responsibility — Professional Developers — Learning to Code — Other Coders
    - Avaliar por recortes de experiência
  - AI tools next year
- Ferramentas de IA
  - Benefícios
  - Principais usos
  - AI tools in the development process
    - 61,8% do público utiliza e 13,8% planeja utilizar
- Descrição do uso
- Público
  - Pesquisa aberta
  - Desenvolvedores de diferentes níveis de experiência
  - Estudantes de código

# Principais Tendências Tecnológicas Estratégicas Para 2025: IA Agêntica - Gartner

- Empresarial
  - IA agêntica
    - Agir x Pensar
    - Otimismo quanto a
      - Aumento de escala
      - Redução no tempo de ação
      - Resolução de problemas
      - Simplificação de modelagens de dados
    - Previsões de uso da IA
      - 1% (2024) -> 33% dos aplicativos de softwares empresariais incluirão IA agêntica (2028)
      - Clientes-máquina de IA substituirão 20% das interações em e-commerces para humanos
    - Figura 1: A lacuna na agência de IA é preocupante
    - Gartner
- Recomendações para IA
  - Mapear decisões e ações que podem ser automatizadas
  - Definir níveis de agência da IA em cada ambiente e fluxo de trabalho
  - Integrar a IA no planejamento estratégico
  - Criar barreiras operacionais para a IA agêntica
  - Criar diretrizes claras, legais e éticas para seu uso
  - Adicionar IA em fluxos de trabalho em que haja demanda por escala e eficiência
  - Pensar agentes como colegas digitais de Nível 1
- Riscos da IA
  - A baixa qualidade pode ser um ciclo vicioso entre conteúdo gerado e consumido pela IA
  - Pode ser alimentada com dados de baixa qualidade
  - Pode não atender os padrões de segurança e qualidade
  - Inserções de prompts, jailbreaks, ataques de segurança de dados e ataques cibernéticos, criados e executados por agentes de IA
  - Malwares inteligentes
  - Alienar o cliente do uso dos sistemas
- Impressão pessoal
  - Principais tipos de valor de negócios com a aplicação da IA generativa
  - Múltiplas respostas permitidas
  - n = 78 CIOs, com incluir "não tenho certeza"
- Descrição do uso da IA
  - Autonomia para
    - Gerar código
    - Gerenciar projetos
    - Gerenciar processos
    - Gerar modelos
    - Resolver problemas
- Público
  - 83 CIOs membros do Research Circle da Gartner