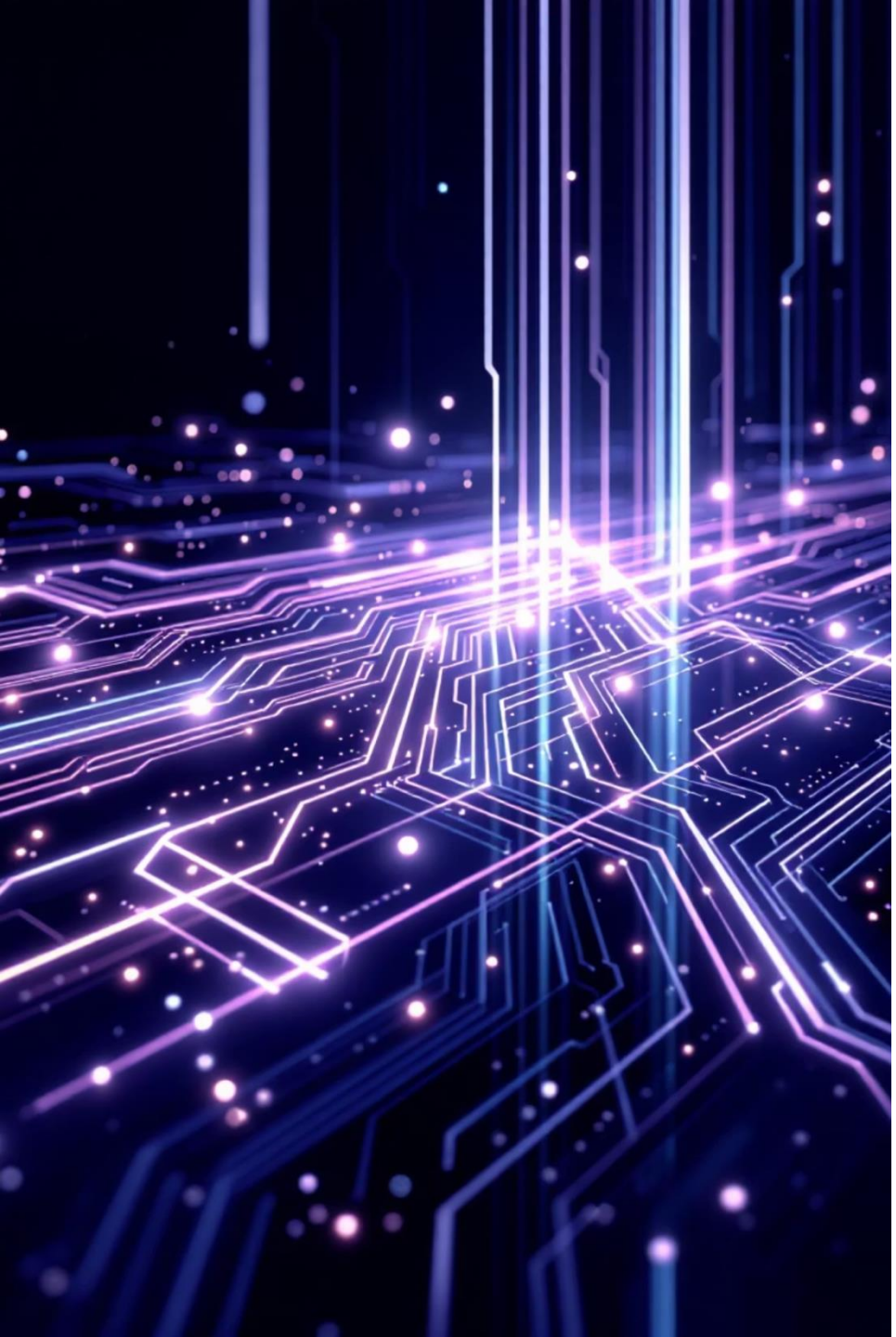


Security in AI Driven Development

Mohammad Mizanur Rahman

CTO, Brain Station 23

What is AI-Driven Software Development?

Accelerated Development

- Generates code snippets
- Automates refactoring & boilerplate boilerplate
- Reduces developer toil

Vibe vs. Agentic Coding

- **Vibe Coding:** AI co-pilot, human-controlled
- **Agentic Coding:** Autonomous AI agents agents

Transformative Impact

- Increased Speed & Throughput
- Enhanced Quality & Security
- Improved Accessibility & Skill Amplification
- Streamlined DevOps Integration

Vibe Coding: Human + AI Collaboration

→ Collaborative AI Model

→ Prompt-Driven Synthesis

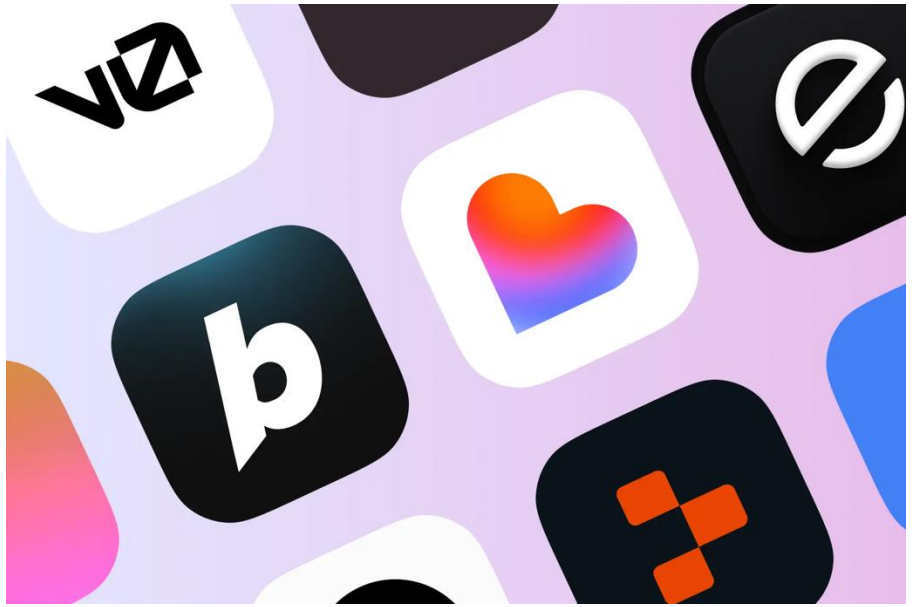
→ AI Outputs: Code completions, Refactoring, Boilerplate, Bug hints

Accelerated Prototyping

- Generates scaffold code
- Defines APIs
- Facilitates PoC development

Human Oversight & Validation

- Engineers retain control
- Ensures coding standards, security
- Integrated in Git & CI/CD



Prominent Platform & Tools

Lovable

Replit

Bolt

V0

Agentic Coding: Autonomous AI Developers

The Next Frontier in Automation

Definition

- Orchestrated LLMs and AI agents
- Autonomously plans, writes, tests, debugs, and deploys software software
- Minimal human intervention based on high-level requirements requirements

Key Components

- Planning modules
- Code generation
- Automated testing
- Static/dynamic analysis
- Seamless CI/CD deployment

Multi-Agent Orchestration

- Agents decompose goals into sub-tasks
- Interact with external tools (Git, IDEs, IDEs, cloud APIs)
- Enables dynamic problem-solving

Comprehensive SDLC Automation

- Automates entire SDLC, from design to operations
- Generates code, refactors, implements features, optimizes
- Reduces Time-To-Market and operational overhead

Human-in-the-Loop Governance

- Human oversight critical for quality, security, and ethics
- Human roles: architecture, code reviews, security auditing
- Strategies: formal verification, adversarial testing, red-teaming

Security Vulnerabilities in AI-Generated Code

AI Code Security Challenges

Critical flaws detected in AI-generated code.

Root Cause: Training Data

Vulnerabilities from biased/low-quality training data.

69%

Pervasive Security Flaws

69% of AI code (without security prompts) has detectable flaws.

flaws.

86%

Input Validation Gaps

86% of AI models fail input validation (CWE-80, CWE-77).

20%

SQL Injection Risk

20% of AI database code introduces SQL Injection (CWE-89).

89).

Common Weaknesses from AI Codegen

Cryptographic Failures (CWE-327)

Insecure cryptographic primitives.

Log Injection (CWE-117)

Insecure logging, unsanitized user data.

Insecure Direct Object References (CWE-284)

Missing authorization checks.

Hardcoded Credentials (CWE-798)

Hardcoded secrets in source code.

Insufficient Session Management (CWE-613)

Weak session logic.

Unique Risks in AI-Generated Code

Cognitive Over-Reliance

- Developers over-rely
- Reduced scrutiny

Outdated Patterns

- AI propagates insecure patterns
- Introduces weak crypto/APIs

Missing Security Context

- AI lacks threat model understanding
- Omits critical security controls

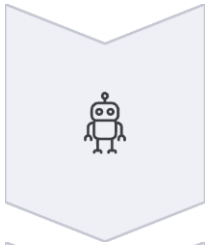
Adversarial Prompting

- Prompt injection to coerce AI
- Generates malicious code/backdoors

Gartner Recommendations:

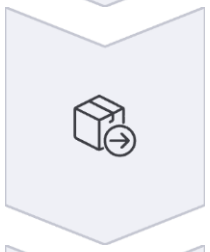
- Strict isolation & containerization
- Network segmentation & sandboxes
- Robust CI/CD with advanced security testing
- Adherence to NIST AI RMF

Unique Risks in Agentic Coding



Unpredictable Code Generation & Opacity

- Non-deterministic, "black-box" code challenging security testing and auditing.



Expanded Software Supply Chain & API Attack Surface

- Increased attack surface from AI APIs and automated dependency risks.

Shadow AI Deployments & Governance Bypass

- "Shadow AI" bypasses governance, creating unmanaged attack vectors and compliance risks.

Advanced Model Manipulation & Integrity Compromise

- Susceptible to data poisoning, adversarial attacks, and sensitive data leaks.

Addressing AI-Driven Development Security Risks

Human-in-the-Loop Governance

- Manual code review & threat modeling
- Security sign-off for PRs
- Adherence to secure coding standards

Specialized AI/MLSec Tools

- AI-specific SAST/DAST
- IAST & SCA for third-party risks
- Fuzzing for adversarial attacks

Containerized Environments

- Sandboxed environments (Kubernetes)
- Network segmentation
- Container runtime security & artifact scanning

Robust Security Primitives

- Strict input validation & output sanitization
- Modern cryptographic standards
- Secure key management (HSMs, vaults)

Continuous Monitoring

Real-time threat detection (SIEM, SOAR)

Developer Training

AI/ML security best practices & adversarial ML

Regulatory Compliance

Adherence to data privacy regulations

Case Study: Mitigating AI Code Risks at Scale

Integrated DevSecOps Toolchain

- Holistic DevSecOps approach
- Early detection of AI code vulnerabilities
- 50% reduction in AI-related security incidents
- Addresses OWASP Top 10 for LLM Applications Applications

Automated CI/CD Security

- Automated AI code scanning in CI/CD
- Detects 80% of AI security flaws pre-deployment deployment
- Identifies prompt injection, deserialization, IDOR IDOR
- Policy-as-code (OPA) for compliance enforcement

AI Governance & Risk Frameworks

- Aligned with NIST AI RMF and ISO 42001
- Defines shared responsibility & security gates
- Mandates human-in-the-loop for critical code
- Includes data provenance and model validation

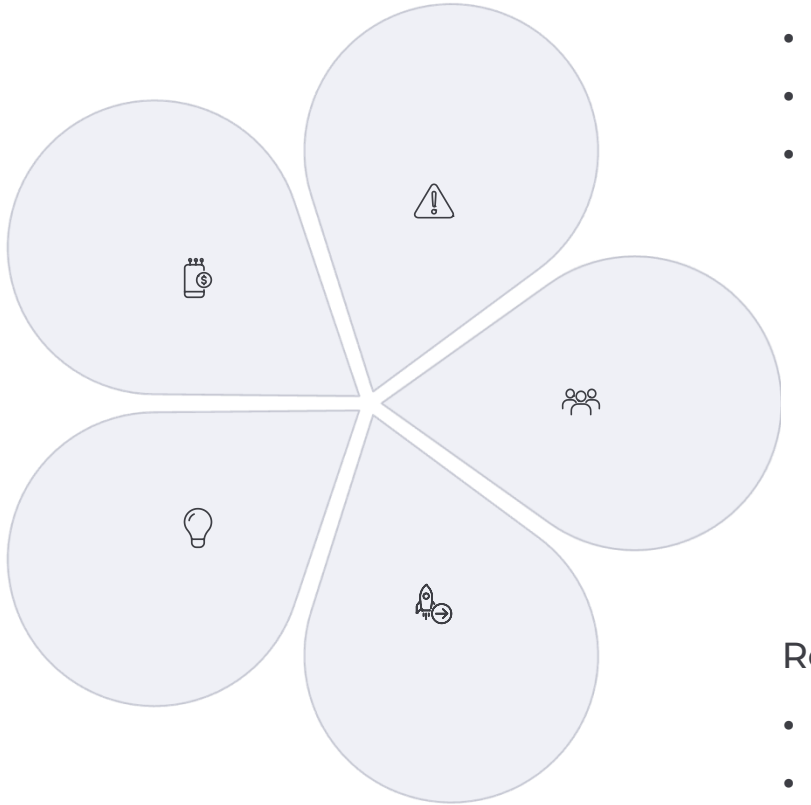
Conclusion: Harnessing AI Safely with Technical Precision

Productivity Gains

- Boost developer velocity.
- Automate code generation and completion.
- Optimize CI workflows.

Intelligent AI Adoption

- Leverage AI with explicit security measures.
- Enable automated compliance & real-time threat detection.
- Monitor AI systems for anomalies.



Mitigate AI Security Risks

- Address AI-generated code flaws.
- Implement model-specific defenses.
- Guard against prompt injection, data leakage, and attacks. attacks.

Enhance DevSecOps & MLOps

- Integrate human oversight & security best practices.
- Automate AI-specific security testing.
- Adhere to NIST AI RMF & OWASP Top 10 for LLMs.

Redefine Software Engineering

- Shift focus to strategic tasks.
- Emphasize prompt engineering & AI orchestration.
- Prioritize architectural design & ethical AI.

LinkedIn

THANK YOU