GitHub Copilot Flight Plan

Our agile approach to Enterprise GitHub Copilot Adoption

The Challenge

Enterprise adoption of GitHub Copilot is more than a licensing exercise. To deliver meaningful impact, it requires coordinated alignment across engineering, security, compliance, and leadership. Realizing the full value of Copilot, introducing DevSecOps workflows and transition toward spec-driven development, demands a deliberate, structured, and methodical approach. This ensures adoption is secure, scalable, and tightly integrated into existing coding and delivery practices, while enabling teams to move faster with confidence. Without a structured approach teams risk inconsistent adoption, security exposure, unclear ROI, and underutilized capabilities.

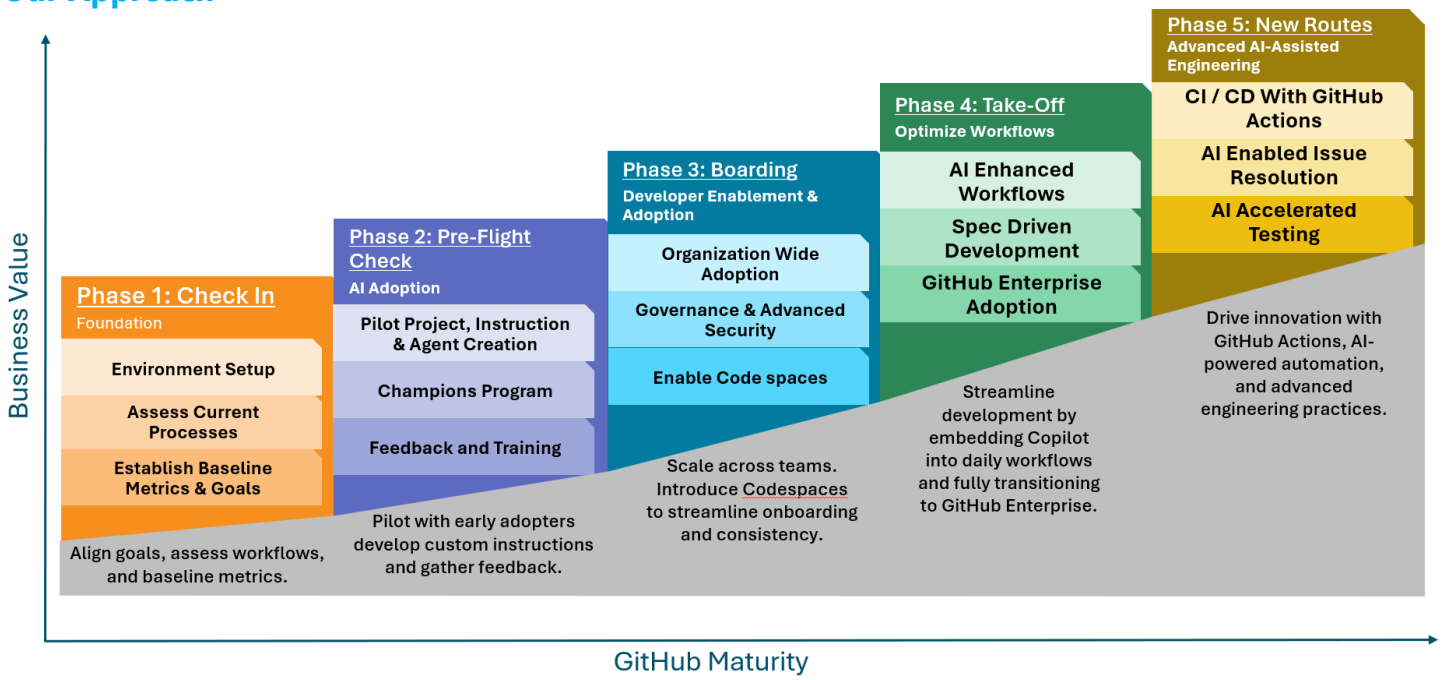
The Solution

Introducing the Spyglass GitHub Copilot Flight Plan. Our five-phase, structured approach helps organizations confidently deploy GitHub Copilot while balancing developer velocity, security, governance, and measurable business value. From vision and readiness through scaled adoption and advanced engineering use cases, we guide teams every step of the journey to increase Copilot maturity and maximize impact.

GitHub Copilot & Enterprise Offerings

- GitHub Copilot Readiness & Security Assessment
- GitHub Copilot Developer Enablement
- GitHub Enterprise Adoption
- GitHub Enterprise For Advanced Engineering & Extensibility

Our Approach



Phase 1: Check In (Foundation)

During Check-In, we establish a clear foundation for GitHub Copilot success. We align stakeholders on goals, assess engineering workflows, and define what success looks like for AI-assisted development.

Key activities include:

- Define GitHub Copilot goals and success criteria
- Align on supported languages, frameworks, and repositories
- Analyze developer workflows and personas
- Establish baseline productivity and quality metrics

This phase ensures GitHub Copilot is introduced with intention — not experimentation without direction.

Phase 2: Pre-Flight Check (AI Adoption)

In Pre-Flight Check, we evaluate whether your environment is technically and organizationally ready for GitHub Copilot at scale. We focus on security, governance, and responsible AI usage.

Key activities include:

- Pilot Project with a small project or team to demonstrate quick wins and minimize risk
- Identify enthusiastic developers to become Copilot "champions".
- Provide Copilot training and develop client specific Copilot Instruction files

This phase ensures GitHub Copilot is used with custom instruction files to guide code generation.

Phase 3: Boarding (Developer Enablement & Adoption)

Boarding is where GitHub Copilot moves from enabled to effective. We focus on driving meaningful adoption through targeted enablement and best practices.

Key activities include:

- Enable Copilot for additional teams and repositories
- Deliver role-based developer training
- Enable Copilot for commit messages
- Establish prompt engineering and AI-assisted coding best practices
- Introduce Codespaces for cloud enabled development and Copilot Automation

This phase accelerates adoption while ensuring developers use GitHub Copilot confidently and consistently.

Phase 4: Take-Off (Optimize Workflows)

In Pre-Flight Check, we optimize engineering workflows by embedding Copilot into daily development, embracing spec-driven development, and fully transitioning to GitHub Enterprise capabilities.

Key activities include:

- Integrate Copilot into daily development processes such as code reviews, bug triage, minor bug fixes
- Move towards Spec Driven Development to maximize Copilot productivity
- Develop additional Copilot developer instructions agents and skills
- Adopt GitHub Enterprise for Source Code Control

This phase ensures GitHub Copilot is used to maximize developer productivity across the Enterprise.

Phase 5: New Routes (Advanced AI-Assisted Engineering)

In New Routes, organizations move beyond core GitHub Copilot usage to advanced AI-assisted engineering scenarios.

Key activities include:

- Implement GitHub Actions to integrate build and release pipelines for seamless CI/CD automation
- Implement bug assignment to Copilot for fully automated bug fixing
- Implement unit code generation, automated testing, and issue creation for automated testing
- Continuously evolve AI-assisted development practices

This phase enables ongoing innovation and long-term competitive advantage.

Spyglass MTG Can Help!

Spyglass MTG enables successful GitHub Copilot rollouts by aligning engineering teams, security, and leadership around a clear, phased path to adoption, scale, and long-term impact. Contact us today!

SPYGLASSMTG

Microsoft Solutions Partner

Infrastructure Azure
Data & AI Azure
Digital & App Innovation Azure
Modern Work
Security