

AI시대 SW 개발 및 접근방식의 변화

이건복

Digital & App Innovation Lead

마이크로소프트



변화하는 개발자 업무환경



하이브리드 근무 환경



엔지니어링 시스템 간의 마찰



급작스런 업무 중단

워크플로우의 강화(개선) 방안



클라우드



인공지능

작성할 수 있는 코드에 더 집중

클라우드 기반 및 AI지원



1. 개발자 경험에
최적화



2. 인텔리전트 앱
개발

새로운 페어프로그래밍 파트너로서의 AI

AI를 IDE에서 사용



속도 개선



설명문 작성



진단

```
20 {
21     for (int i = number; i <= 100; i++)
22     {
23         for (int j = 0; j <= 9; j++)
24         {
25             //Print the division table entry
26             Console.WriteLine($"{i} / {j} = {i / j}");
27
28             // Print the multiplication table entry
29             Console.WriteLine($"{i} * {j} = {i * j}");
30         }
31     }
32 }
33
34
35
36
37
38
```

GitHub Copilot

안녕하세요! 함께 빌드해 보겠습니다.

저는 AI로 작동되기 때문에 예상치 못한 실수를 할 수 있습니다. 생성된 코드 또는 제안을 확인하고 [피드백 공유](#)해야 합니다.

- 의도를 나타내기 위해 프롬프트의 시작 부분에 **슬래시 포함** 명령(예: /fix)이 있습니다.
- ********을 입력하여 포함할 코드를 참조하세요.
- **사용 Alt + /** 바로 가기를 사용하여 인라인 채팅을 열고 편집기에서 코드를 구체화합니다.

자세한 지침은 [/help](#)을(를) 사용하세요.

AI 제안은 정확하지 않을 수 있습니다.

Copilot에 물어보세요: 명령을 입력하려면 /를, 참조 코드를 입력하려면 #을 입력합니다.

/ #

출력

출력 보기 선택(S): GitHub Copilot

```
[INFO] [default] [2024-02-22T01:39:06.085Z] [fetchCompletions] engine https://copilot-proxy.gi
[INFO] [default] [2024-02-22T01:39:06.260Z] request.response: [https://copilot-proxy.githubuse
[INFO] [streamChoices] [2024-02-22T01:39:06.262Z] solution 0 returned. finish reason: [stop]
[INFO] [streamChoices] [2024-02-22T01:39:06.264Z] request done: headerRequestId: [43171bd:
[INFO] [CopilotProposalSourceProvider] Completion dismissed
```

GitHub Copilot 채팅 솔루션 탐색기 Git 변경 내용 팀 탐색기

속성



AI 기반 페어 코딩 파트너

```
JS demo.js 2 ●
C: > demo > JS demo.js
1 //Azure Cosmos DB에 연결하기 위한 함수 생성
2 ✨
3 function connectToCosmosDB() {
    const { CosmosClient } = require("@azure/cosmos");
    const endpoint = "https://<your-account-name>.documents.azure
    const key
```

```
Program.cs*
ConsoleApp1
14 else
15 {
16     Console.WriteLine("Invalid input. Please enter an integer.");
17 }
18
19 static void PrintMultiplicationTable(int number)
20 {
21     for (int i = number; i <= 100; i++)
22     {
23         for (int j = 0; j <= 9; j++)
24         {
25             //Print the division table entry
26             Console.WriteLine($"{i} / {j} = {i / j}");
27
28             // Print the multiplication table entry
29             Console.WriteLine($"{i} * {j} = {i * j}");
30         }
31     }
32 }
33
34
```

GitHub Copilot 채팅

/askvs 런타임중에 코드를 수정하는 방법

- 사용 Alt + / 바로 가기를 사용하여 인라인 채팅을 열고 편집기에서 코드를 구체화합니다.

자세한 지침은 /help을(를) 사용하세요.

/askvs 런타임중에 코드를 수정하는 방법

참조

GitHub Copilot

Visual Studio 2022에서는 "Hot Reload" 기능을 사용하여 애플리케이션을 중지하지 않고도 실행 중인 코드를 수정할 수 있습니다. 이 기능은 디버깅 세션 중에 코드를 실시간으로 수정하고 변경 사항을 즉시 반영할 수 있게 해줍니다.

다음은 Hot Reload를 사용하는 방법입니다.

AI 제안은 정확하지 않을 수 있습니다.

Copilot에 물어보세요: 명령을 입력하려면 /를, 참조 코!

/ #

출력

출력 보기 선택(S): GitHub Copilot

```
[INFO] [default] [2024-02-22T01:39:06.085Z] [fetchCompletions] engine https://copilot-proxy.githubusercontent.com/v1/engines/copilot-codex
[INFO] [default] [2024-02-22T01:39:06.260Z] request.response: [https://copilot-proxy.githubusercontent.com/v1/engines/copilot-codex/completions] took 175 ms
[INFO] [streamChoices] [2024-02-22T01:39:06.262Z] solution 0 returned. finish reason: [stop]
[INFO] [streamChoices] [2024-02-22T01:39:06.264Z] request done: headerRequestId: [43171bda-30c4-49bb-a332-e462422865d5] model deployment ID: [y229cf7684e2f]
[INFO] [CopilotProposalSourceProvider] Completion dismissed
```

GitHub Copilot 채팅

솔루션 탐색기 Git 변경 내용 팀 탐색기

속성



Copilot을 사용한
개발자의 생산성
증가

87%

반복적인 작업에 대한
정신적 노력 감소

74%

좋아하는 작업에 더 집중

73%

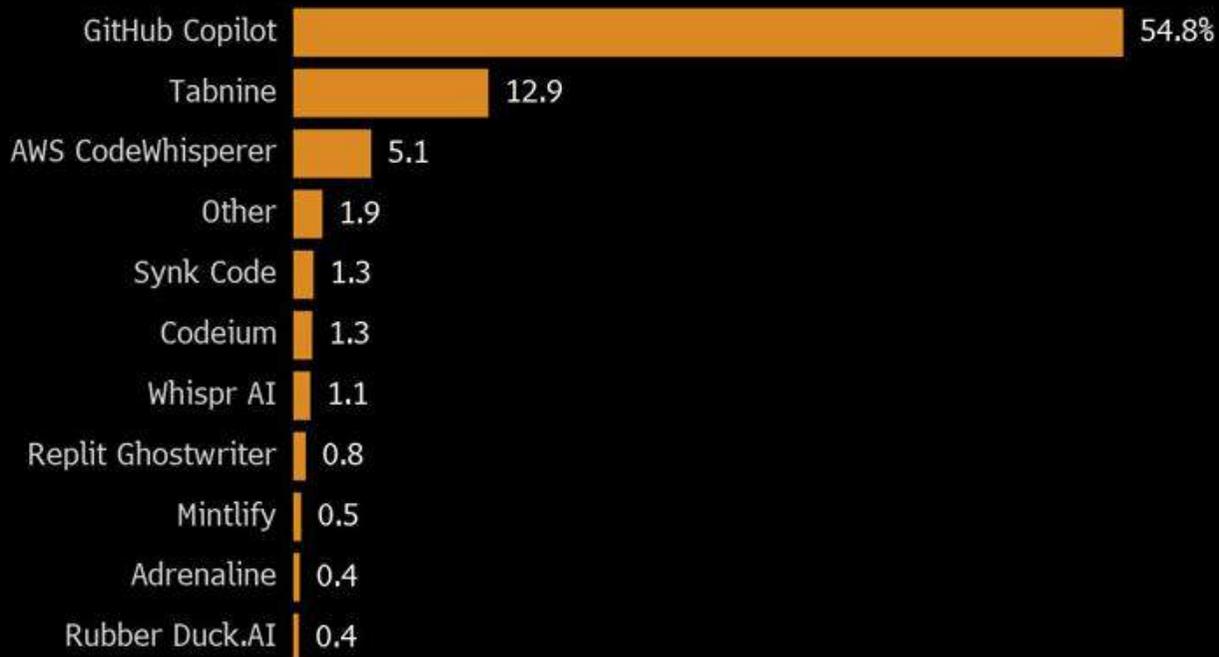
몰입할 수 있었다고
응답

소스: GitHub® <https://github.blog/2023-06-13-survey-reveals-ais-impact-on-the-developer-experience/>
<https://github.blog/2023-04-14-how-generative-ai-is-changing-the-way-developers-work/>.

Bloomberg - 개발자 코딩 지원도구

GitHub Copilot Top Pick For Coders

A survey asked developers which AI coding tool they used most in past year



Source: Stack Overflow 2023 Developer Survey

Bloomberg

물론 한계는 있습니다!

- 훈련된 데이터가 제안에 영향

- GitHub Copilot != 컴파일러, 보안검사

- 당신의 마음을 알지는 못합니다.



LLM을 사용하는 최적의 방법?

Semantic Kernel(SK)이란?

시맨틱 커널은 LLM을 네이티브 코드로
오케스트레이션할 수 있는 경량 오픈소스 SDK.

Semantic Kernel

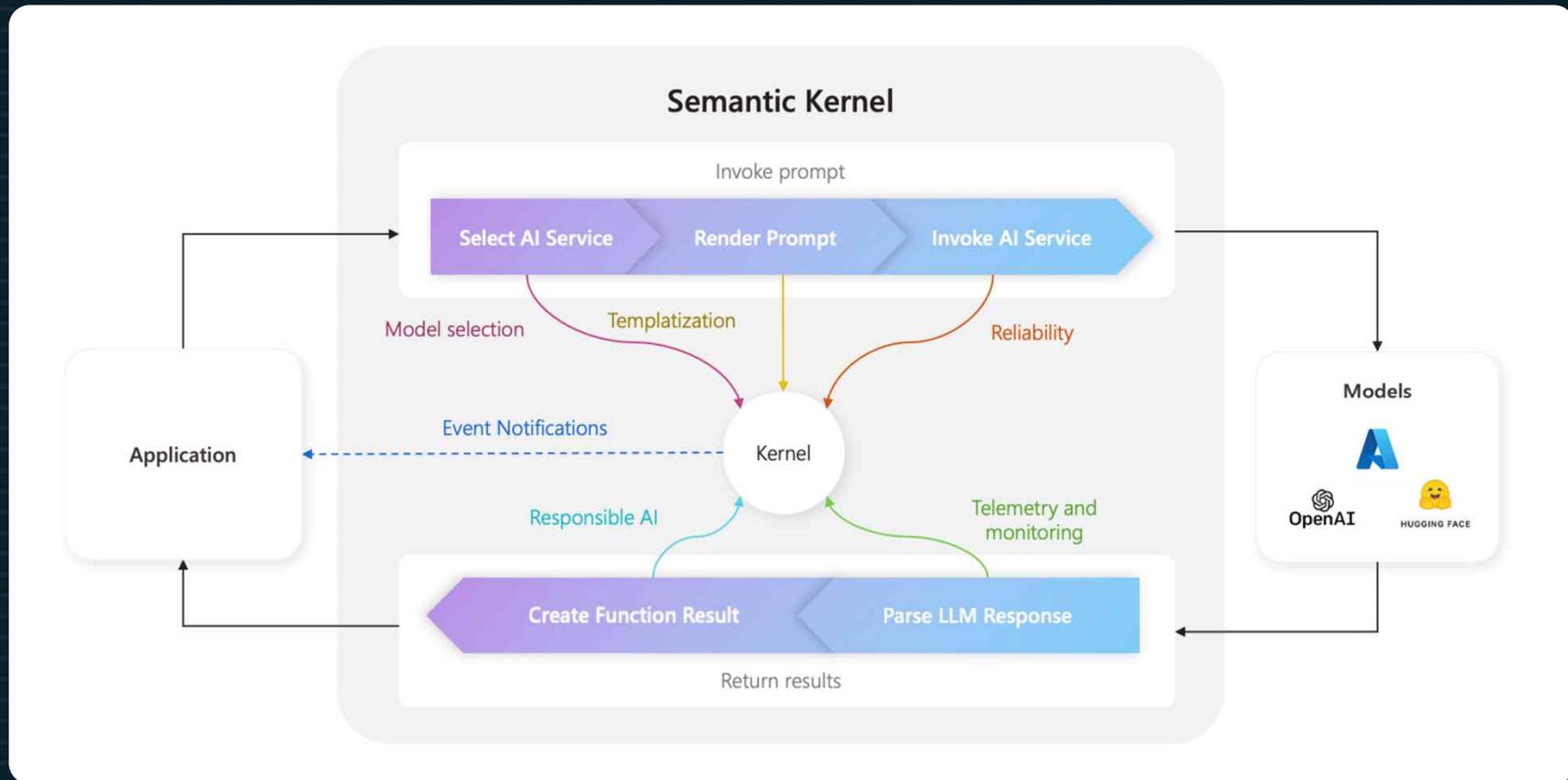
Semantic Kernel 지원언어...

C# (V1.6)

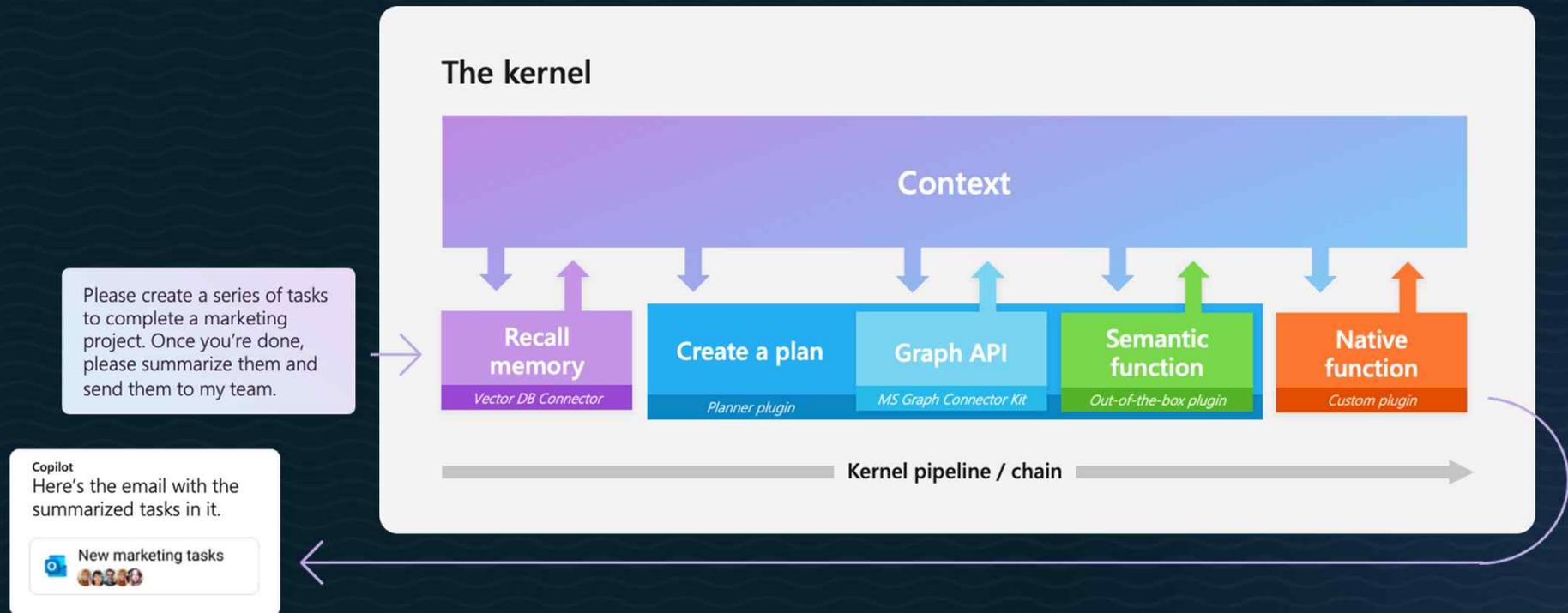
Python (Beta)

Java (Release Candidate)

커널은 이 모든 것의 중심



Semantic Kernel 아키텍처



Semantic Kernel 은 확장가능



Semantic Kernel의 구성

1

Plugins

2

Planners

3

Personas

요약

- 개발자의 코드를 학습하지 않습니다.
- AI 에이전트와 같이 일하는 방법.
- 중요한 코드에 더 집중하세요.
- 새로운 개발자의 삶이 어떤 것인지 보여주세요!

Microsoft ❤️ developers