ArcPilot

A bit about me…

- Co-Founded DREAMDEV TECHNOLOGIES LIMITED in 2020
  National Enterprise Awards – Overall Winner 2024
- 20+ years experience in Software Engineering
  - Freelance consultant for 10+ years in Ireland and the UK
- Co-Founder ArcPilot.ai, the AI Software Architecture Co-pilot

ArcPilot

ArcPilot

Bootstrapping & Behavioural Context

ArcPilot

ArcPilot

A Quick look at what we are working on...

Let’s set some context Context

ArcPilot

2024.10

The state of AI Development
Mid February 2025

If I connect an entire code repository to an LLM today, and ask it to implement something complex following my org's best practices..
I don't expect it to work… [1]
……..This is true as of (February 2025 😬)

[1] Lost in the Middle: How Language Models Use Long Contexts –Nelson F. Liu et al

LLM performance
Doesn’t scale

GitHub Copilot

CURSOR

Claude

“When the size and difficulty of a maths problem increases,
increases,
the probability of a correct answer decreases exponentially with the
exponentially with the number of steps involved” [1]

[1] GSM-Symbolic: Understanding the Limitations of Mathematical Reasoning in Large Language Models – Apple 2024

A look at the profile of the Augmented Engineer

- Not all engineers are equipped to exploit the capabilities of the LLM

- Highly knowledgeable
- Self motivated, self organizing
- Understands the technology intimately
- Understands the business problem…
  - Let’s call him Karl.. ☺

A look at the profile of the Augmented Engineer

- Unfortunately, we're not all Karl… Exceptional engineers are required to get exceptional results
- The Engineer must have an established internal "Programming Model"

GitHub Copilot

CURSOR

Claude

[1] Lost in the Middle: How Language Models Use Long Contexts –Nelson F. Liu et al

Product based Delivery

Feature Request

Development accounts for

10.5%

OF TOTAL LEAD TIME

GitHub Copilot improves developer productivity by 26%

Or in our example

2.73%

Reduction in Lead Time

https://devops.com/measuring-github-copilots-impact-on-engineering-productivity/

definition
- Prototyping

- Sprint Planning
- Tech Review

- Code Review
- Testing

- User Feedback
- Metric Analisis

- Monitoring
- Perf analysis

Total Avg lead time: 19 days | **Value added time:** 13 days | **Non Value added time** 6 days | **Total FTT:** 84%

Project Based Delivery

Project Request

Development accounts for

22%

OF TOTAL LEAD TIME

GitHub Copilot improves developer productivity by 26%

Or in our example

5.72%

Reduction in Lead Time

https://devops.com/measuring-github-copilots-impact-on-engineering-productivity/

- Business Case
- Requirements
- Stakeholder sign-off

- System Design
- Technical Spec
- Arc. Review

- Development
- Unit Testing
- Code Review

- System Testing
- Int Testing
- UAT

- Release Planning
- Prod. Deploy
- HOTO

Total Avg lead time: 159 days | **Value added time:** 112days | **Non Value added time** 47 days | **Total FTT:** 70%

How we make the most of AI, by playing to it's strengths

ArcPilot

2024.10

Technique number 1
Bootstrapping

- **Traditional code generation is 100% deterministic…**
- Boilerplate code makes up to 70% of a modern software solution…

- Writing a code templates are incredibly difficult and time consuming…. **Unless you are an AI are an AI**
- Encapsulate your project setup in a templates.
- Make it easy to standup entire backend services, UI components, Endpoints, Database migration Database migration scripts, Database Schema Migrations, Perf/ Unit/ Integration/Security Tests. Integration/Security Tests.
- Template CI/CD Pipelines and IaC

Technique number 2
File Relationship Mapping

File systems are complex, without intimate knowledge can be difficult to navigate.

An LLM with 2 simple prompts, run in batches over your Repo, will generate a really comprehensive architectural map of your code base…

Technique number 3
Behavioural Context

We can express any system in 2 ways…

1. As a representation of it's technical parts…
2. As a sum of it's behaviours

If we assume the YAGNI principal stands, every line of code should have a reason to exist.. As being exist.. As being part of

1. A behaviour
    1. Command, Event Triggered Command, Query
2. Data Type

- If we can categorise these reasons, we can tag the Behavioural context of your file base… base…

- This passed ot the LLM is a powerful way to provide instant context to the developer, tight context tight context for the AI, and simple pair programming over a branch

ArcPilot

Thank you!

2024.10