



Hochschule Aalen
UNIVERSITY OF APPLIED SCIENCES

Welcome

AI IN SOFTWARE DEVELOPMENT

AI Milestones Timeline

Let's explore the full post-ChatGPT story together:

From AlexNet (2012) to autonomous AI agents (2026) — every model, every breakthrough, every training cost.

- Filterable by category
- Detailed descriptions and paper links
- Training cost comparisons

Open the AI Milestones Timeline

Find the milestone that surprises you most. Be ready to share!

The LLM Provider Landscape (2026)

Closed-Source:

- **OpenAI** — GPT-4o/o3, 128K ctx, Copilot
- **Anthropic** — Claude Opus/Sonnet, 200K ctx, Claude Code
- **Google** — Gemini Ultra/Pro, 1M+ ctx, Code Assist

Open-Weight:

- **Meta** — Llama 4, 128K ctx, free
- **DeepSeek** — R1/V4, 128K ctx, cost-efficient
- **Mistral** — Large, 128K ctx, EU data sovereignty

Plus: Microsoft (Azure OpenAI), Amazon (Bedrock, Q Developer), xAI (Grok)

The Overwhelming Choice Problem

8+ providers × multiple models × different pricing = **paralysis**

You need a systematic evaluation based on:

- **Task type** — generation, review, documentation, testing?
- **Data privacy** — GDPR, on-premises requirements?
- **Budget** — API costs scale quickly
- **Reproducibility** — open vs. proprietary trade-offs
- **Integrations** — IDE, CI/CD, code review toolchain?

Discussion: Which AI coding tool have you already tried? What was your experience?

Why VS Code + GitHub Copilot?

1. **Free for students** — [GitHub Education](#) program
2. **Best IDE integration** — built specifically for VS Code
3. **Industry standard** — 70% of developers use VS Code, most popular AI tool
4. **Model-agnostic** — supports Claude, Gemini as backends (since 2025)
5. **Full spectrum** — inline completions → chat → agent mode

We'll also explore: direct API access (Week 6), Claude Code (Week 9), Cursor, and multiple models for comparison (Week 7).

GitHub Copilot Setup

Step-by-step:

1. **GitHub Education** — Sign up at education.github.com with your `@studmail.htw-aa1en.de` email
2. **Install VS Code** — Download from code.visualstudio.com
3. **Install extensions** — "GitHub Copilot" + "GitHub Copilot Chat"
4. **Sign in** — Use your GitHub account (Education pack auto-activates)
5. **Verify** — Create a `.py` file and watch for ghost text suggestions
6. **Python extension** — Install the Microsoft Python extension

Hugging Face Setup

Why? Unified API, thousands of models, not locked to a single provider.

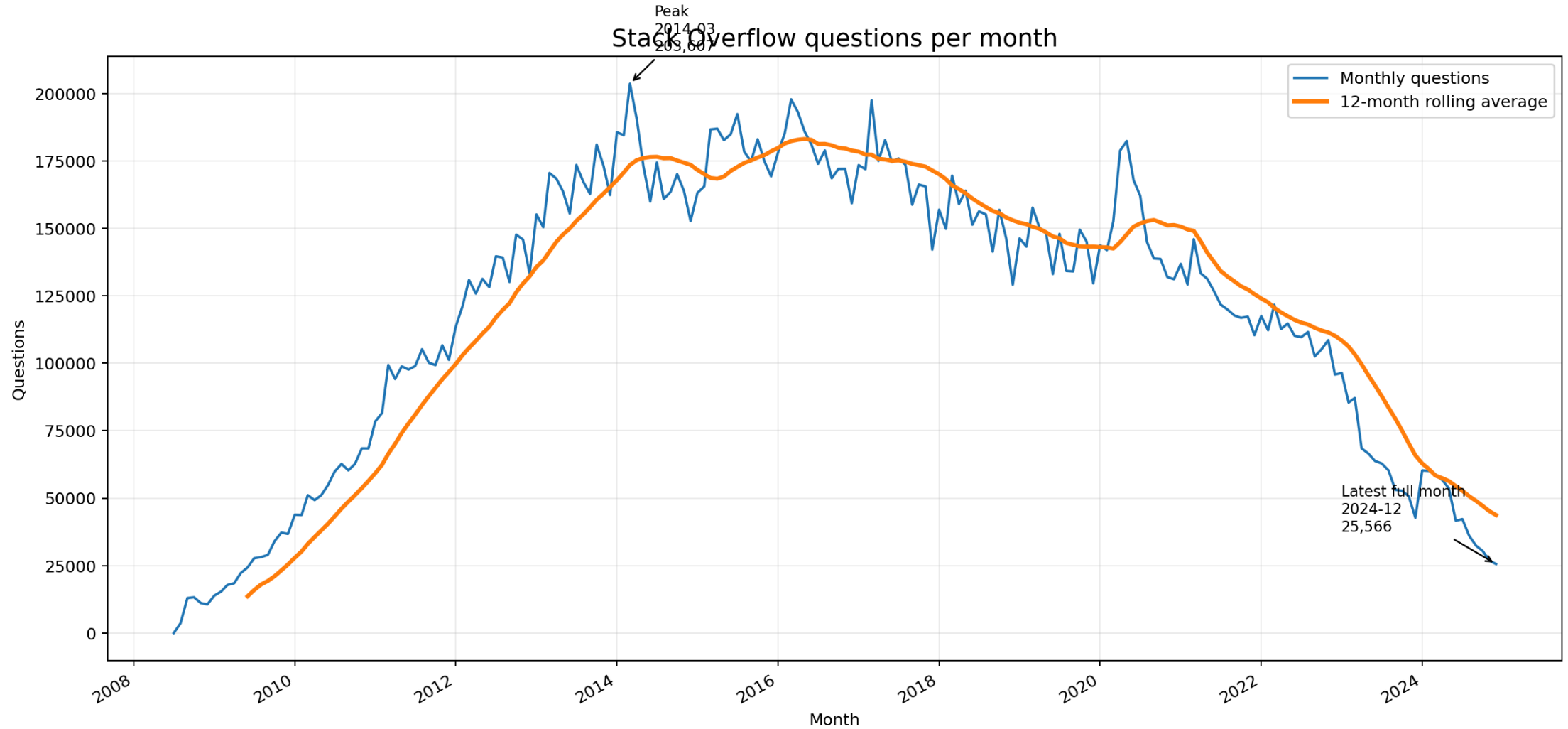
1. **Create account** at huggingface.co → claim student benefits
2. **Generate API token** (read-only) at huggingface.co/settings/tokens
3. **Clone starter repo:**

```
git clone https://github.com/L3GJ0N/hf-llm-uv-starter
```

4. **Create** `.env` **file** with your `HF_TOKEN=hf_...`

5. **Try it out:**

- `uv run hf-llm list-models`
- `uv run hf-llm ask --model [name] --prompt "Explain recursion"`
- `uv run hf-llm chat --model [name]`



The Challenge

Your mission: Recreate the Stack Overflow plot using Python + AI tools.

- Find the data **programmatically** (Stack Overflow Data Explorer, Stack Exchange API, ...)
- Reproduce the chart: monthly question counts + rolling average
- **Extend it:** Overlay markers for key AI breakthroughs from the timeline

Two observations you will make:

1. This is surprisingly doable with AI assistance
2. You **will** hit real obstacles (data access, parsing, bugs)

Pre-Challenge Survey

Before you start — answer honestly:

- How difficult do you think this is? (1-5)
- Time estimate for the script only?
- Time estimate for full SW dev cycle (structure, tests, CI/CD, docs)?
- Confidence level? (1-5)

Take the survey now:

<https://forms.gle/XqmQDD1KvqYXfWfs5>

The gap between your estimates and reality = a core lesson of this course.

Challenge Rules

- **Teams:** 2-3 people
- **Tools:** Must use an AI coding assistant (Copilot recommended)
- **Deliverable:** GitHub repository with:
 - Python code reproducing the chart + AI milestone markers
 - Record of **every prompt** you used

Why record prompts?

Prompts are design decisions. They deserve version control just like code.

The AI Diary — A Preview

For every AI interaction, capture:

What to record:

- What you **asked** (prompts)
- What you **got** (responses)
- What you **changed** (iterations)
- What you **learned** (patterns)

Why it matters:

- Reproducibility
- Learning from patterns
- Team knowledge sharing
- Professional accountability

For now, find your own natural approach. We'll formalize this later.



Hochschule Aalen
UNIVERSITY OF APPLIED SCIENCES

Questions?

THANK YOU

Your feedback is welcome.