

# Future of Software Engineering

**James Gress**

*AI Director*

*Emerging Technology AI Lead*

*Accenture*

 LinkedIn: - [jamesgress](#)

 GitHub: [jmgress](#)

 X.com: [@jmgress](#)

 [Tampa Bay Generative AI Meetup](#)



# The Evolution of Software Engineering in the Age of AI



# November 30, 2022: ChatGPT Launches

- OpenAI released ChatGPT to the public as a free research preview.
- Marked a significant milestone in AI, introducing a conversational interface for users.
- Sparked widespread interest and rapid adoption in various industries.

# 2023: AI-Powered Code Completion Becomes Mainstream

- Tools like GitHub Copilot, Tabnine, and Amazon CodeWhisperer gained popularity.
- Provided developers with real-time code suggestions within IDEs.
- Enhanced productivity and streamlined the coding process.

# 2024: Conversational Coding Interfaces Emerge

- Integration of AI chat interfaces into development environments.
- Tools like GitHub Copilot Chat allowed natural language interactions with code.
- Revolutionizing the way developers write and refine code.

# 2025: Agentic AI Systems Transform Development Workflows

- Emergence of agentic AI systems capable of autonomous task handling.
- Systems interpret high-level objectives and modify code across multiple files.
- Transformed software development by reducing manual intervention.

# 2025: Emerging Generative AI

- A2A - Agent to Agent Communications
- MCP - Model Context Protocols
- LOKA - Layered Orchestration for Knowledgeful Agents
- Trusted Agent Huddle

# Jensen Huang - CEO NVIDIA

"AI is not going to take your job. The person who uses AI is going to take your job."

"You probably recall over the course of the last 10, 15 years almost everybody who sits on a stage like this would tell you that it is vital that your children learn computer science. Everybody should learn how to program. And in fact, it's almost exactly the opposite."

"Software is eating the world, but AI is going to eat software."

— Various speeches, 2023–2024



# Satya Nadella - CEO Microsoft

*"AI is the runtime that will shape all of what we do"*

— **December 2024**

*"I think what'll happen is these crud I mean SAS applications are a crud database with a lot of business logic so the crud database will then get orchestrated outside of the business logic tier of just the SAS application"*

— **January 2025**

# Eric Schmidt - Former Google CEO

"We believe as an industry that in the next one year the vast majority of programmers will be replaced by AI programmers."

— August 2024

# Dario Amodei - CEO of Anthropic

"I think we'll be there in three to six months, where AI is writing 90% of the code. And then, in 12 months, we may be in a world where AI is writing essentially all of the code."

— March 2025

# LlamaCon 2025

## LIVE: Mark Zuckerberg chats with Microsoft CEO Satya Nadella



# Sam Altman – CEO of OpenAI

Questions was asked "What does my future look like?"

"There are sort of two views you can take.  
You can say, 'Oh man, it's doing everything I can do,'  
or you can say, 'Look at all the new things I can now do.'"

— April 2025

# Vibe Coding



**Andrej Karpathy** ✓

@karpathy



There's a new kind of coding I call "vibe coding", where you fully give in to the vibes, embrace exponentials, and forget that the code even exists. It's possible because the LLMs (e.g. Cursor Composer w Sonnet) are getting too good. Also I just talk to Composer with SuperWhisper so I barely even touch the keyboard. I ask for the dumbest things like "decrease the padding on the sidebar by half" because I'm too lazy to find it. I "Accept All" always, I don't read the diffs anymore. When I get error messages I just copy paste them in with no comment, usually that fixes it. The code grows beyond my usual comprehension, I'd have to really read through it for a while. Sometimes the LLMs can't fix a bug so I just work around it or ask for random changes until it goes away. It's not too bad for throwaway weekend projects, but still quite amusing. I'm building a project or webapp, but it's not really coding - I just see stuff, say stuff, run stuff, and copy paste stuff, and it mostly works.

6:17 PM · Feb 2, 2025 · **4M** Views



1.2K



3.9K



24K



12K



# Initial Concerns on Vibe Coding

- Do we care about the code?
- What if the code accesses sensitive data?
- Loss of architecture and design principles
- Erosion of team understanding and collaboration
- Testing becomes an afterthought
- Skill atrophy for developers

# Do We Really Care About the Code?

- Do we care about the code — or just the outcome?
- AI has already shifted part of our mindset — treating code as a temporary utility, not a permanent product.
- AI is already being inserted into code bases and we don't know even what it really does.





ChatGPT 4o ▾



Share



How many days has it been since ChatGPT was released?



✓ As of today, it has been **880 days** since ChatGPT was released on November 30, 2022.

Let me know if you want me to also show you how to calculate this automatically anytime in the future! [\[>-\]](#)



How many days has it been since ChatGPT was released?

### Analysis



python

Always show details ☐

Copy

```
from datetime import datetime
```

```
# Define the release date
```

```
release_date = datetime(2022, 11, 30)
```

```
# Get today's real date
```

```
today_date = datetime.now()
```

```
# Calculate the difference
```

```
days_since_release = (today_date - release_date).days
```

```
days_since_release
```

Result

880

create a python script that will download all pdf's from a given website



Sure! Here's a simple but solid Python script that **downloads all PDF files** from a given website URL. It uses `requests`, `BeautifulSoup`, and `urllib.parse` to find and download the PDFs. I'll include some good error handling too:

[Download Pdfs](#)

Answer in chat instead >

This script will:

- Ask for a website URL.
- Find all links that end in `.pdf`.
- Download them to a local folder called `pdfs/`.

You can easily extend it to **recursively crawl sub-pages** if needed too.

Would you like me to show you a version that can **crawl the entire website** (not just one page)? 🚀



Ask anything

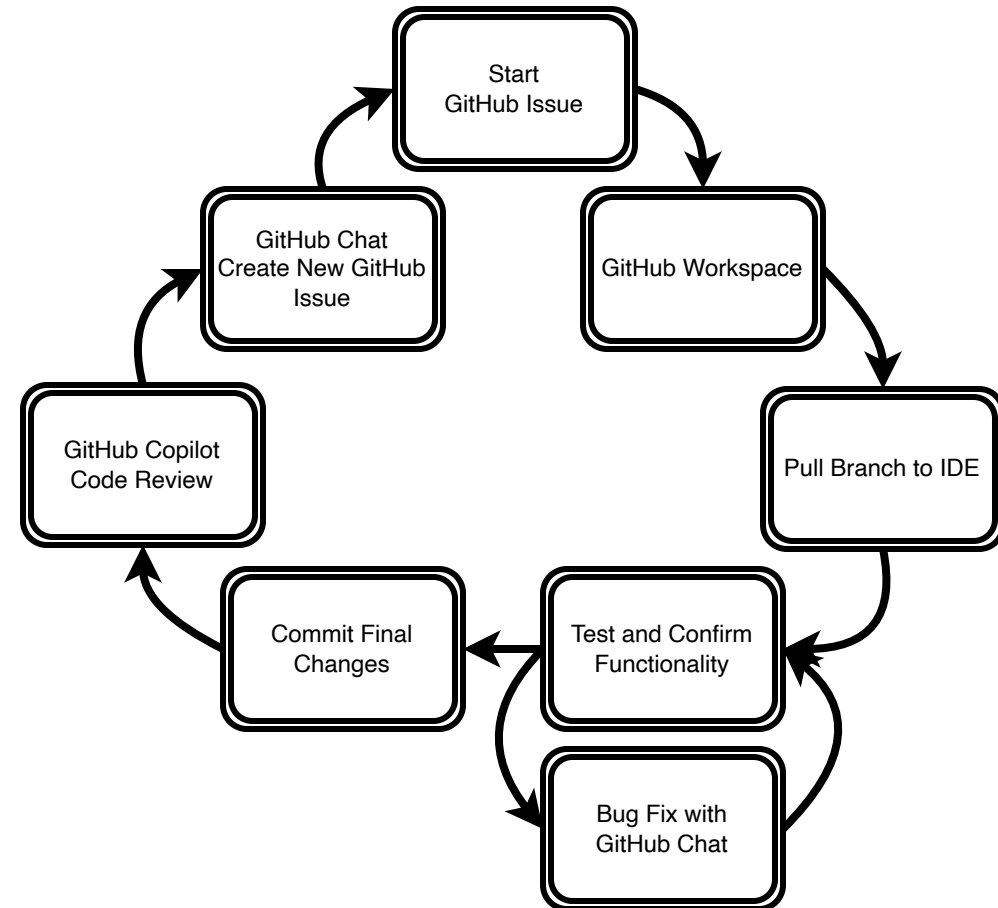


```

1  import os
2  import requests
3  from bs4 import BeautifulSoup
4  from urllib.parse import urljoin
5
6  def download_pdfs_from_website(url, download_folder="pdfs"):
7      # Create a folder to save PDFs
8      if not os.path.exists(download_folder):
9          os.makedirs(download_folder)
10
11     try:
12         response = requests.get(url)
13         response.raise_for_status()
14     except requests.RequestException as e:
15         print(f"Error fetching the website: {e}")
16         return
17
18     soup = BeautifulSoup(response.text, "html.parser")
19     pdf_links = []
20
21     # Find all anchor tags with href ending in .pdf
22     for link in soup.find_all("a", href=True):
23         href = link['href']
24         if href.lower().endswith(".pdf"):
25             full_url = urljoin(url, href)
26             pdf_links.append(full_url)
27
28     if not pdf_links:
29         print("No PDFs found on the page.")
30         return
31
32     print(f"Found {len(pdf_links)} PDF(s). Starting download...")
33
34     for pdf_url in pdf_links:
35         try:
36             pdf_response = requests.get(pdf_url)
37             pdf_response.raise_for_status()
38             filename = os.path.join(download_folder, os.path.basename(pdf_url))
39
40             with open(filename, 'wb') as f:
41                 f.write(pdf_response.content)
42
43             print(f"Downloaded: {filename}")
44         except requests.RequestException as e:
45             print(f"Failed to download {pdf_url}: {e}")
46
47     if __name__ == "__main__":
48         website_url = input("Enter the website URL: ")
49         download_pdfs_from_website(website_url)
50

```

# Vibe Coding for the Enterprise



# How do we embrace AI for Software Engineering?

# Calibrated Trust Between Humans and Machines

- Trust isn't binary—it's a spectrum
- We must know when to rely on AI and when to verify
- Align confidence with competence
- Design workflows that include human-in-the-loop oversight

*"We don't need blind trust—we need calibrated collaboration."*

# AI Writing Code $\neq$ Ignoring DevOps

"When we say AI will write the code, we don't mean we're skipping the discipline."

AI is transforming how code is created —  
but **code quality, security, and delivery pipelines still matter.**

- DevOps isn't going away
- Guardrails are still essential
- Testing, observability, and governance remain critical



We're not removing the need for rigor —  
we're **elevating it to meet the speed of AI.**

## Don't Just Automate the Past—Reimagine the Future

"It's like putting a jet engine on a horse-drawn carriage. Sure, it moves faster, but you're missing the point—the whole vehicle can be reinvented. Generative AI gives us a chance to reimagine how we deliver, not just speed up what we already do."

# AI as a Teammate, Not a Tool

- AI isn't replacing your team—it's joining it
- Like a great teammate, it doesn't wait for instructions
- It watches, understands, and suggests—proactively
- But like any teammate, it still needs guidance

*"Think of AI as the pair programmer that never sleeps."*

# From Asking to Advising: A New AI Interaction Paradigm

- Stop building chatbots. Start building outcomes.
- Don't make users *ask*—let AI *advise*
- Systems should anticipate needs, not just respond
- The future UX is ambient, embedded, and contextual

# Preparing for the Future

- Learn agentic systems, system thinking, and architecture
- Embrace change: experimentation and transparency
- Grow internal champions for AI adoption

# The End of "Tech for Tech's Sake"

- Old mentality: "I just code, not business"
- AI needs business context to generate useful output
- Engineers must understand goals, not just syntax

*"In the future, the best engineers won't just know how to code — they'll know why the code matters."*

# Closing: A Call to Lead

"AI is the runtime that is going to shape all of what we do..." — Satya Nadella

- Will you be a passenger — or help engineer the future?
- Start learning. Start building. Start leading.

*"The next chapter of software engineering is being written now — and we're all holding the pen."*

# Questions?

