

# What Programmers Aren't Taught in School

Petr Zemek

Lead Software Engineer at Gen™  
petr.zemek@gendigital.com  
<https://petrzemek.net>, @s3rvac



**AVG**



**Avast**

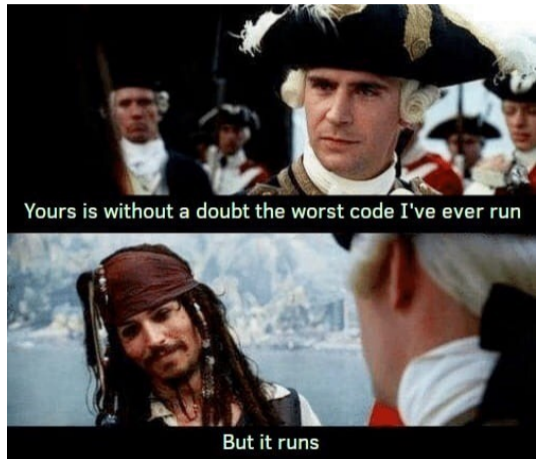
**Gen™**

Career in tech

Collaborating with others

Selected technical topics

Concluding remarks



([source](#))

## Career in tech

# Your first job – What to expect

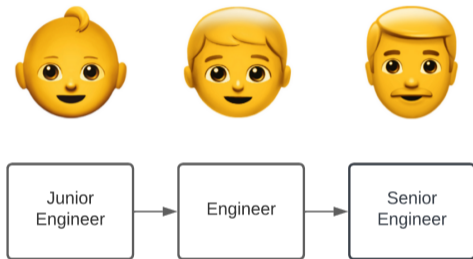
- Looking for a job
- Applying for a job, CV
- Hiring process, screening, interview(s)
- Onboarding, your first day/week/month



(source)

# Career paths – Starter pack

(source of images)

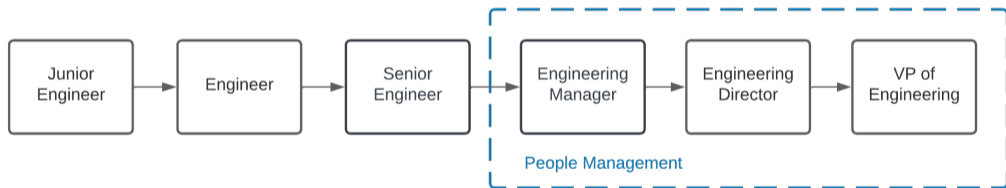


???

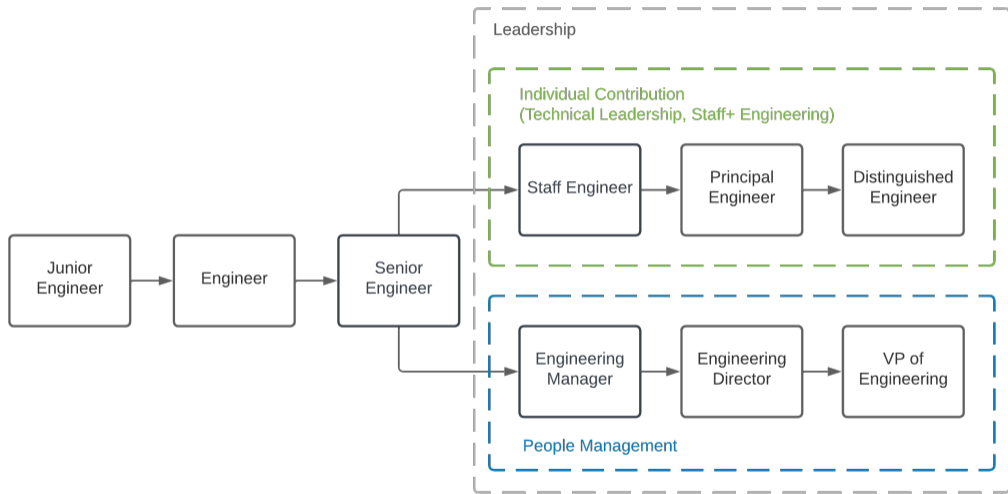


???

# Career paths – Management



# Career paths – Technical leadership



## Selected other options:

- *Testing & Quality*: Tester, QA Engineer
- *Delivery*: Project Manager, Program Manager
- *Product & Design*: Product Manager, UI/UX Designer
- *Infrastructure*: Systems Engineer, Network Engineer
- *Data & Analytics*: Data Engineer, Data Analyst
- *Security*: Security Engineer, Security Analyst
- *Architecture*: Technical Architect, Solution Architect
- *Research*: AI Researcher, Threat Researcher
- *Other*: Technical Writer, IT Support / Helpdesk, ...


## Notes:

- Different roles / switching careers
- How to choose?

 **Bonsai farmer**  
My Bonsai Nursery · Self-employed  
Jun 2025 - Present · 1 yr  
Chehalis, Washington, United States · On-site



logo.png

 **Goose farmer**  
Self-employed · Self-employed  
Jul 2023 - Jun 2025 · 2 yrs  
On-site



**Microsoft**  
22 yrs 4 mos

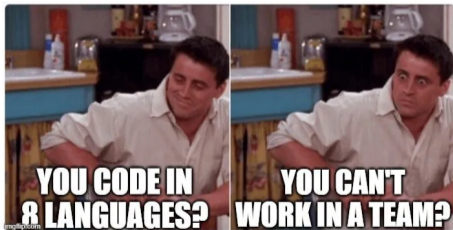
- **Principal Performance Architect**  
Full-time  
Jul 2022 - Jul 2023 · 1 yr 1 mo  
Chehalis, Washington, United States · Remote
- **Principal Software Development Engineer**  
Apr 2001 - Jul 2022 · 21 yrs 4 mos  
Redmond, WA

([source](#))

**Collaborating with others**

# Being a team player

- Being a nice person is more important than being technically excellent
- You need to be able to work with other people
- Disagree, then commit
- Respect other people's time
- Help other people succeed



([source](#))

- What are they?
  - Communication (written and verbal)
  - Collaboration
  - Asking for something
  - Giving and receiving feedback
  - Presenting your ideas and work
  - Empathy
  - Conflict resolution
  - Active listening
  - ...
- They matter, sometimes (much) more than technical skills
- People problems vs technical problems

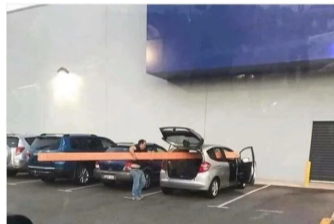


([source](#))

# Work split and tracking

- Jumping straight into code versus planning first
- Anything non-trivial requires some split and tracking
- Why tracking of work matters
- Common tools: Jira, GitHub Issues, Trello, Asana, ...
- Estimations

This guy is a software engineer, you can tell by his awesome estimation skills



(source)

# Leading projects

- Congratulation, you are a project lead now!
- Anyone\* can lead a project
- What do I even do?
  - Whatever is necessary to make the project succeed!
  - Ensure decisions are made
  - Break down the work to manageable chunks
  - Ensure that everyone knows what to work on
  - Ensure that work is being done
  - Eliminate blockers and unblock people
  - Identify and address risks
  - Communicate status and plans
  - ...



## **Selected technical topics**

NEWS 19 FEB 2025

## Hundreds of US Military and Defense Credentials Compromised

### Threat actors exploit a 0-day in exposed management consoles of Fortinet FortiGate firewalls

### 200 million social media records leaked in major X data breach

Protect yourself from phishing, identity theft with these cybersecurity tips

By Kurt Knutson, CyberGuy Report Fox News

Published April 9, 2025 10:00am EDT

### Meta Confirms WhatsApp Hack—Act Now To Stay Safe

By Davey Winder, Senior Contributor. Davey Winder is a veteran cybersecurity...

Follow Author

Feb 03, 2025, 05:58am EST

### URGENT: Microsoft Patches 57 Security Flaws, Including 6 Actively Exploited Zero-Days

Mar 12, 2025 Ravie Lakshmanan

Patch Tuesday / Vulnerability

— Trending News

### Sensitive data of over one million Community Health Center patients exposed

January 31, 2025

### The Biggest Supply Chain Hack Of 2025: 6M Records Exfiltrated from Oracle Cloud affecting over 140k Tenants

CloudSEK uncovers a major breach targeting Oracle Cloud, with 6 million records exfiltrated via a suspected undisclosed vulnerability. Over 140,000 tenants are impacted, as the attacker demands ransom and markets sensitive data online. Learn the full scope, risks, and how to respond. Are you worried your organization might be affected? Check your exposure here - <https://exposure.cloudsek.com/oracle>

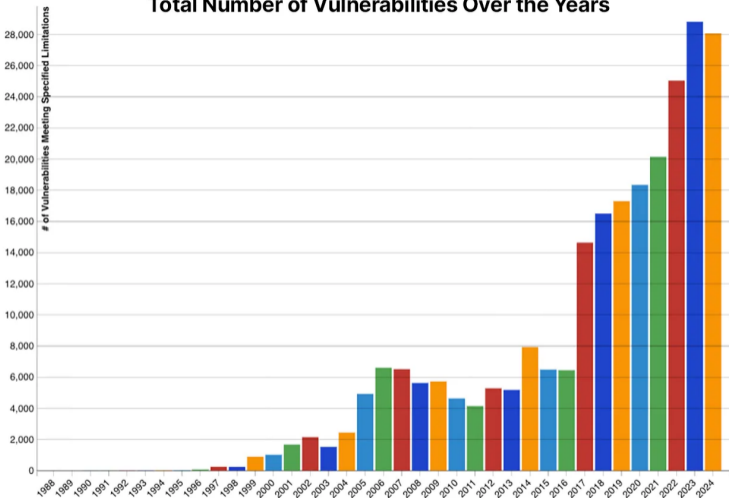


CloudSEK TRIAD

March 21, 2025

# Code safety and security: Motivation (Continued)

## Total Number of Vulnerabilities Over the Years



(source)

# Code safety and security

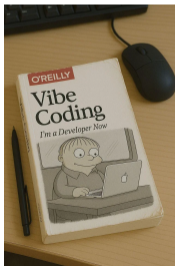
- Safety
  - What is it?
  - Measures
- Security
  - What is it?
  - Measures
- Can have immense consequences
- Now important more than ever



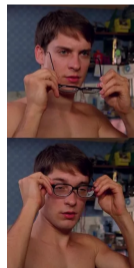
(source)



(source)



(source)



**VIBE CODING**

**VULNERABILITY AS A SERVICE**

(source)

# Code safety and security: Selected examples

ID: `1; DROP TABLE users`

- Insufficient error handling

```
int fclose(FILE *stream);
```

- Insufficient input validation

```
$id = $_GET['id'];  
$sql = "SELECT * FROM users WHERE id = $id";  
$result = $mysqli->query($sql);
```

- Memory-related issues
- Concurrency-related issues
- and more...

Petr Zemek: Safe and secure code (IVS 2025)

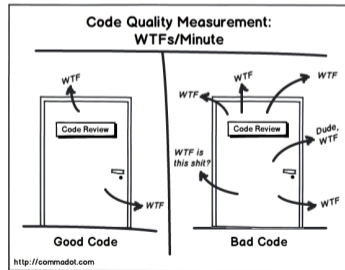


(source)



(source)

- What is a code review (CR)?
- Most commonly associated with pull requests (PR)
- How to evaluate code quality?
- What to focus on?
  - P1: Is the code correct?
  - P2: Is the code safe and secure?
  - P3: Does the code fit into the big picture?
  - P4: Is the code maintainable and efficient enough?
  - P5: Does the code conform to coding conventions?



(source)

# Reasons for creating PRs and doing CRs

- Finding bugs and other defects
- Learning something new
- Improving code-reading/understanding skills
- Increasing the sense of mutual responsibility within your team
- Finding a better solution
- Running automated checks before the code is merged
- Writing better code



([source](#))

# Code reviews anti-patterns

- “LGTM” without doing a proper review
- Focusing primarily on nitpicks
- Forcing subjective changes
- Not doing code reviews promptly
- “Bikeshedding”
- Back and forth (ping-pong) reviews
- Constantly bothering the author during the review

Petr Zemek: Pull requests and code reviews (IVS 2020)



I Am Developer  
@iamdeveloper

10 lines of code = 10 issues.

500 lines of code = "looks fine."

Code reviews.

11:58 AM · Nov 5, 2013 · Tweetbot for iOS

[\(source\)](#)



[\(source\)](#)

# AI-assisted development

- The reality of how AI is used today
- What AI is good at
- What AI is bad at (and why it matters)
- Where AI cannot (currently) help you
- Use AI to learn better, not to skip learning
- AI is reshaping our jobs
- AI is not just about writing code
- Human in the loop
- Understand that LLM  $\neq$  AGI
- Agentic development
- Costs
- Safety & security (use [Sage](#))

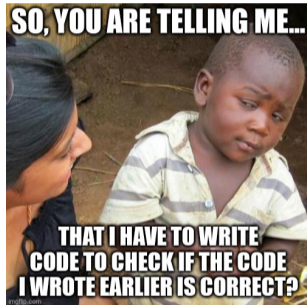


(source)

# AI-assisted development: 10 levels of using AI

- 1 (STORY MODE) No use
- 2 (STORY MODE) Copying and pasting from the web
- 3 (EASY) Autocomplete in your IDE/editor
- 4 (EASY) On-demand code generation in your IDE/editor
- 5 (MEDIUM) Interactive pair programming in a single session
- 6 (MEDIUM) Managing multiple local sessions for different tasks
- 7 (HARD) Offloading tasks to specific agents running remotely
- 8 (HARD) Autonomous agents working on your tickets (output: PRs)
- 9 (INSANE) Writing specifications, verifying behavior
- 10 (IMPOSSIBLE) Fully autonomous (setting strategy, AI does the rest)

- Why do we write tests?
- Tests must be automated
- Many test types available
- Beware of so-called *flaky tests*



(source)

# CI/CD – Automating the boring stuff

- Continuous Integration (CI) and Continuous Delivery (CD)
- Typical flow: push code → build → lint → tests (→ deploy)
- Why automate?
  - Never forgetting to run tests and checks
  - Catching issues early
  - Testing various environments
  - Reproducibility (“*it works on my machine*”)
  - Shorter delivery times
  - High-performance build infrastructure
  - Consistent deployments and rollbacks
- GitHub Actions, GitLab CI/CD, TeamCity, Jenkins, ...
- Status checks on GitHub, GitLab, ...
- Usually also set up for pull requests



(source)

✓	🔒	CI / tests (macos-15, pypy-3.10) (push)	Successful in 37s
✓	🔒	CI / tests (ubuntu-24.04, 3.10) (push)	Successful in 21s
✓	🔒	CI / tests (ubuntu-24.04, 3.14) (push)	Successful in 22s
✓	🔒	CI / tests (ubuntu-24.04, pypy-3.10) (push)	Successful in 32s
✓	🔒	CI / tests (windows-2022, 3.10) (push)	Successful in 28s

- What happens after the code is written?
  - Release and deploy (including infrastructure)
  - Monitoring
  - Handling incidents
  - Suggesting improvements
- Development (Dev) versus Operations (Ops)
- DevOps: *"You build it, you run it"*
- Release and deploy through CI/CD
- Infrastructure as Code (IaC)
- Monitoring (logs, metrics, alerts)



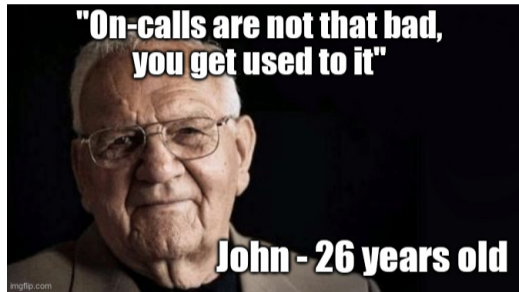
(source)



(source)

# Incident response and on-calls

- What is incident response?
- What are on-calls?
- The pros and cons of being on-call
- Tools: PagerDuty, Splunk, xMatters, ...
- On-call rotations and schedule
- Triggering an incident
- Time to engage, escalation policies
- Postmortems



([source](#))

# Technical writing and presenting

- Learn to become fully proficient in (professional) English
- Use every opportunity to practice presenting
- Focus on *why*
- Selected presentation anti-patterns
  - A wall full of text / reading slides aloud
  - Speedrunning
  - The jargon dump
  - Disregarding the audience
  - “Can everyone see this?”

Junior devs writing comments:



(source)



(M365 Copilot)

# Code quality and technical debt

- What is technical debt and why it happens?
- It is not always bad (trade-offs)
- Code gets written once but read many times
- Code quality is important

Petr Zemek: High-quality code (IVS 2023)



([source](#))

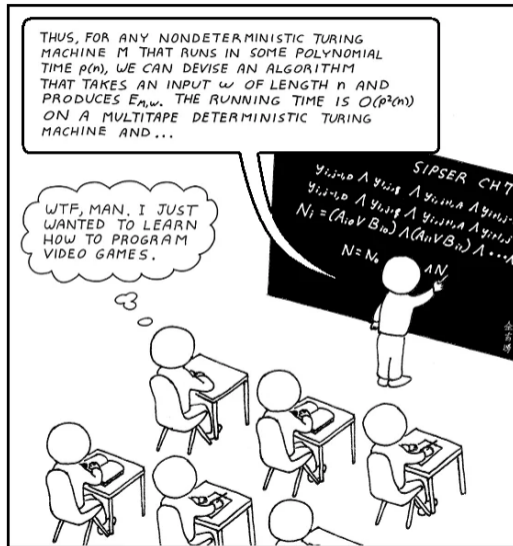
When you're trying to look at the code you wrote a month ago



([source](#))

## Concluding remarks

# Core topics vs hot topics



(source)

# Core topics vs hot topics: Examples

- Hot topic: programming in *⟨insert a trendy programming language here⟩*
- Core topics: programming language theory, compilers
  
- Hot topic: cloud computing
- Core topics: distributed systems, distributed algorithms, operating systems
  
- Hot topic: writing video games
- Core topics: graphics, linear algebra, digital image processing
  
- Hot topic: AI
- Core topics: ML, natural language processing, human-computer interaction

# My other talks

- Understanding engineering career ladders (Gen, 2025-12-11)
- To manage or not to manage? That is the question... (Gen, 2025-06-05)
- Staff+ engineering: Leadership beyond management (Gen, 2025-01-29)
  
- Safe and secure code (FIT - IVS, 2025-05-07)
- High-quality code (FIT - IVS, 2023-05-03)
- Introduction to Python (FIT - IPP, 2021-03-08)
- Pull requests and code reviews (FIT - IVS, 2020-04-28)

...and more

→ <https://petrzemek.net/talks>



**Petr Zemek**

@petrzemek · 129 subscribers · 33 videos

Hi! I am Petr Zemek (AKA "s3rvac"), a software engineer from the Czech Republic.

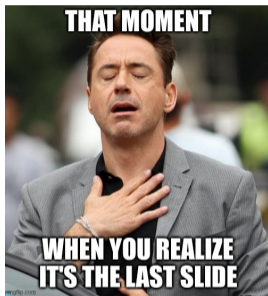
[petrzemek.net](https://petrzemek.net) and 1 more link

Subscribe

<https://www.youtube.com/@petrzemek>

# Summary

- There are many career options in tech
- Being a team player and having good soft skills is very important
- Real-world projects need to be split into manageable chunks, and someone needs to lead them
- Code safety and security is more important than ever
- All changes have to be reviewed by someone else, and tested before being released
- AI-assisted development is becoming commonplace
- Automation of the boring stuff through CI/CD
- DevOps, incident response, and on-calls are critical parts of software development
- Focus on core topics (e.g. machine learning) rather than hot topics (e.g. AI tools)



# Gen<sup>TM</sup>

<https://www.gendigital.com/>

Our mission is to create innovative and easy-to-use technology solutions that help people grow, manage, and secure their digital and financial lives.

