

**Can generative AI understand software engineering tasks? Should you worry?**

**Java User Group Switzerland, 29.05.**

# Zu meiner Person...

```
{  
  name: "Iryna Dohndorf",  
  
  profession: "Software Engineer"  
  
  organisation: "Karakun AG",  
  
  education: "CS, Dr.-Ing. from TU Dortmund (2017)",  
  
  interests: "Speaker, Book Author, Machine Learning  
  Enthusiast"  
}
```

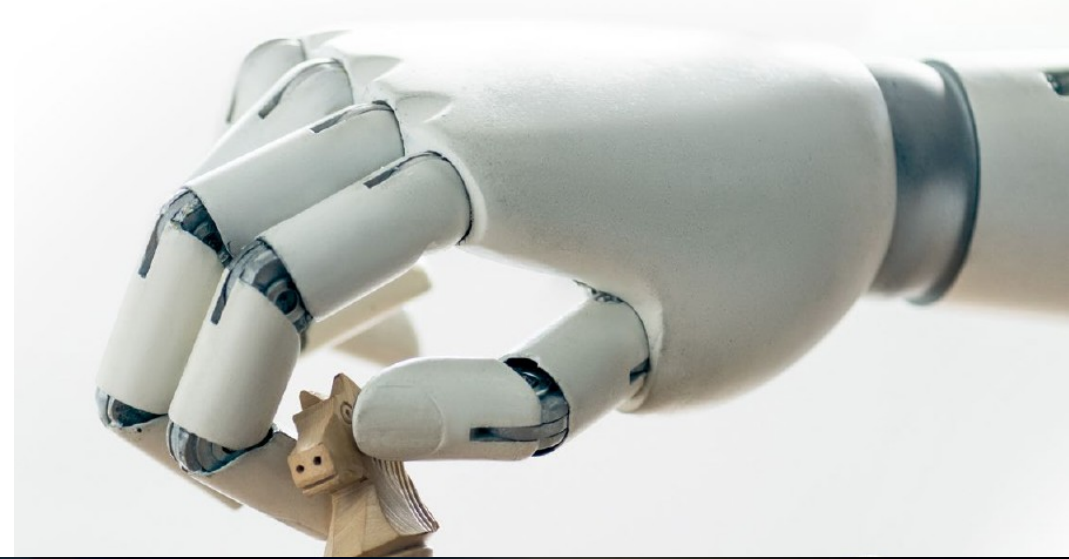


# Was Euch erwartet...

- Warum das Thema?
- Wie funktioniert „Generative KI“?
- Wie funktioniert das Lernen von Programmiererinnen?
- Was sind Einsatzmöglichkeiten für „Generative KI“
- Was muss bei dem Einsatz beachtet werden?



# Warum das Thema?



Default (GPT-3.5)

java implemente separate methods to add number of days, weeks and months in current day method should return LocalDate and take int as parameter

You can create separate methods to add a number of days, weeks, and months to the current date in Java using the `LocalDate` class from the `java.time` package. Here's a sample implementation:

```
java Copy code

import java.time.LocalDate;

public class DateManipulation {
    public static LocalDate addDays(int daysToAdd) {
        return LocalDate.now().plusDays(daysToAdd);
    }

    public static LocalDate addWeeks(int weeksToAdd) {
        return LocalDate.now().plusWeeks(weeksToAdd);
    }

    public static LocalDate addMonths(int monthsToAdd) {
        return LocalDate.now().plusMonths(monthsToAdd);
    }

    public static void main(String[] args) {
        int days = 5;
        int weeks = 2;
        int months = 3;
    }
}

Regenerate
```



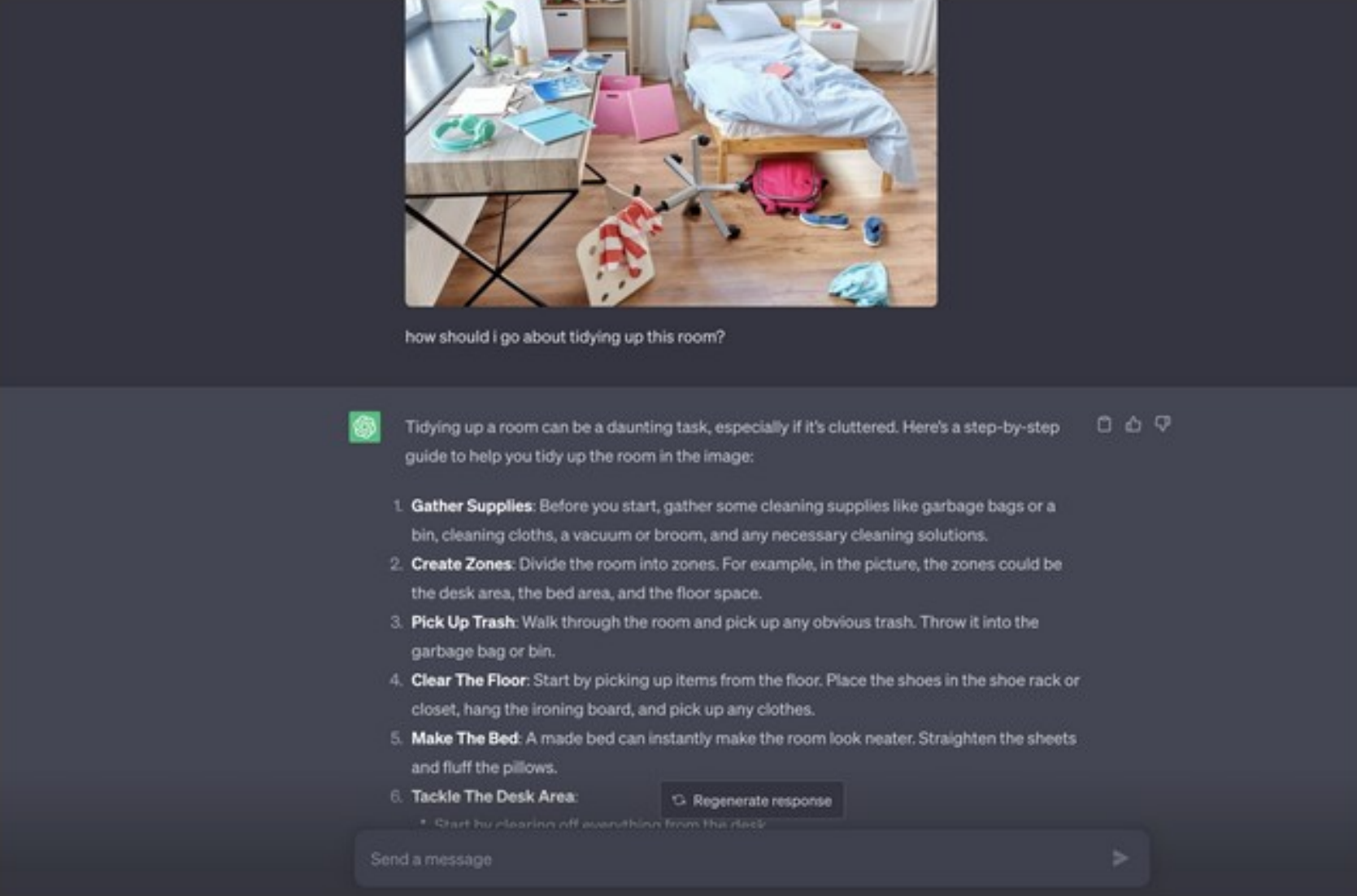
# Warum das Thema?

- KI hat sich in den letzten Jahren stark entwickelt und wird immer häufiger in der Softwareentwicklung eingesetzt
  - Ein Verständnis der Technologie ist enorm wichtig
- Technologie kann dazu beitragen den Entwicklungsprozess stark zu beschleunigen
- Kann die Qualität der Entwicklung verbessern
- Ermöglicht Entwicklung neuer Arten von Anwendungen, innovative Lösungen und hilft bei TDD etc.
- Kann sich auf Fähigkeiten von Softwareentwicklern auswirken
- Generative AI und LLMs bringen auch viele Risiken mit sich



# Generative KI

CodeGen,  
OpenAI Codex,  
Copilot,  
ChatGPT,  
Devin,  
Auto-GPT,  
CodeGPT,  
AlphaCode,  
Tabnine  
PolyCoder



how should i go about tidying up this room?

Tidying up a room can be a daunting task, especially if it's cluttered. Here's a step-by-step guide to help you tidy up the room in the image:

- Gather Supplies:** Before you start, gather some cleaning supplies like garbage bags or a bin, cleaning cloths, a vacuum or broom, and any necessary cleaning solutions.
- Create Zones:** Divide the room into zones. For example, in the picture, the zones could be the desk area, the bed area, and the floor space.
- Pick Up Trash:** Walk through the room and pick up any obvious trash. Throw it into the garbage bag or bin.
- Clear The Floor:** Start by picking up items from the floor. Place the shoes in the shoe rack or closet, hang the ironing board, and pick up any clothes.
- Make The Bed:** A made bed can instantly make the room look neater. Straighten the sheets and fluff the pillows.
- Tackle The Desk Area:**

Send a message

# Generative KI

**DALL-E** | KI kann nicht nur Texte erzeugen, sondern auch Bilder. Dieses »Kunstwerk« ist entstanden, indem man vorgab: »Ein Computer an einem Strand auf einem Liegestuhl unter einer Palme mit einem Cocktail.«



MANION BISCHOFF (ERSTELLT MIT DALL-E)

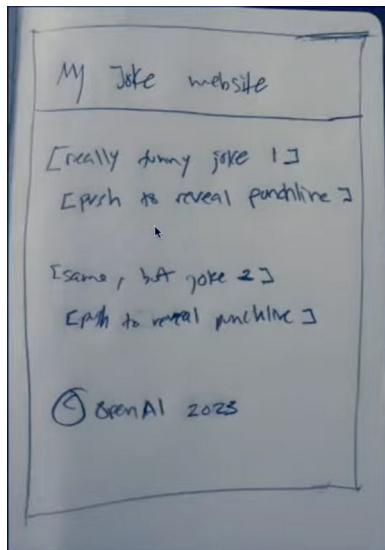
# Prototyp mit GPT-4

\*Multimodal

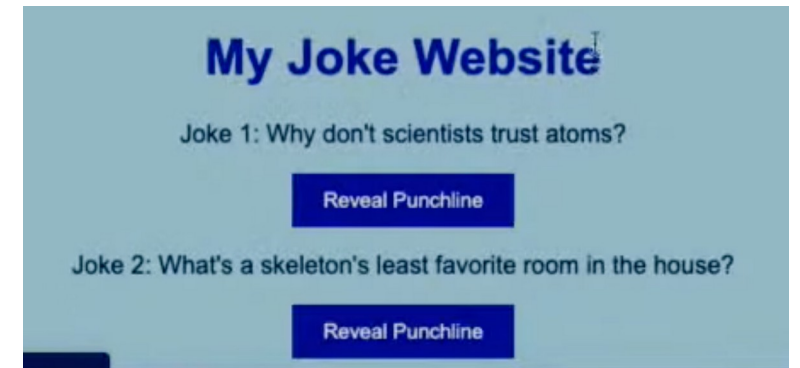


# PoC & Prototypen

- Generierung eines funktionsfähigen Prototypen aus Zeichnung



```
everything";  
    }  
    function revealJoke2() {  
        document.getElementById("jokeText").innerHTML = "Why did the chicken go to the seance? To get to the other side!";  
    }  
    </script>  
    </body>  
    </html>  
    Submitting: [{"role": "user", "content": ["Write brief HTML/JS to turn this mock-up into a colorful website, where the jokes are replaced by two real jokes.", {"image": "..."}]}]  
    Response: <DOCTYPE html>  
    <html lang="en">  
    <head>  
        <meta charset="UTF-8">  
        <meta name="viewport" content="width=device-width, initial-scale=1.0">  
        <title>My Joke Website</title>  
        <style>  
            body {
```



\* Multimodales GPT-4  
(Kostenloses ChatGPT ist GPT-3.5)

Model: GPT-4 with vision ALPHA

Can you help me write the code for this website?

Certainly! To create a simple website like the one depicted in the image, you can use HTML for the structure, CSS for styling, and JavaScript for the calculation logic.

Below is an example of how you can create the structure and functionality for this Compound Interest Calculator.

- HTML (index.html):** This is the main structure of your webpage.

html  Stop generating  Copy code

Send a message ..

ChatGPT may produce inaccurate information about people, places, or facts. [ChatGPT July 20 Version](#)

Model: GPT-4 with vision ALPHA

for the structure, CSS for styling, and JavaScript for the calculation logic.

Below is an example of how you can create the structure and functionality for this Compound Interest Calculator.

- HTML (index.html):** This is the main structure of your webpage.

```
html  Copy code
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="styles.css">
  <title>Compound Interest Calculator</title>
</head>
<body>
  <div class="calculator">
    <h1>Compound Interest Calculator</h1>
    <label for="principal">Initial principal ($):</label>
    <input type="number" id="principal" placeholder="1000">

    <label for="rate">Interest rate (%):</label>
    <input type="number" id="rate" placeholder="8.5">

    <label for="period">Period (years):</label>
    <input type="number" id="period" placeholder="10">

    <button onclick="ca [REDACTED] ton>
```

html  Stop generating  Copy code

Send a message ..

ChatGPT may produce inaccurate information about people, places, or facts. [ChatGPT July 20 Version](#)

Model: GPT-4 with vision ALPHA

**Star Data Table**

Star	Absolute Magnitude	Apparent Magnitude	Temperature (in degrees K)	Distance (in Light Years)
Sun	+5	-26.5	6,000	--
Spica	-3	+1	20,000	220
Sirius B	+12	+8	7,000	8.6
Aldebaran	-0.7	+0.9	4,500	68
Betelgeuse	-5.5	+0.8	3,300	650
Proxima Centauri	+15	+11	3,000	4.2

Transcribe the information from the 'Star Data Table' into a table and then explain how the diagram is helpful for astronomers.

T Here's the information from the 'Star Data Table' transcribed into a more readable table format:

Star	Absolute Magnitude	Apparent Magnitude	Temperature (K)	Distance (Light Years)
Sun	+5	☐ Stop generating	6,000	--

Send a message

ChatGPT may produce inaccurate information about people, places, or facts. ChatGPT July 20 Version

Here's the information from the 'Star Data Table' transcribed into a more readable table format:

Star	Absolute Magnitude	Apparent Magnitude	Temperature (K)	Distance (Light Years)
Sun	+5	-26.5	6,000	--
Spica	-3	+1	20,000	220
Sirius B	+12	+8	7,000	8.6
Aldebaran	-0.7	+0.9	4,500	68
Betelgeuse	-5.5	+0.8	3,300	650
Proxima Centauri	+15	+11	3,000	4.2

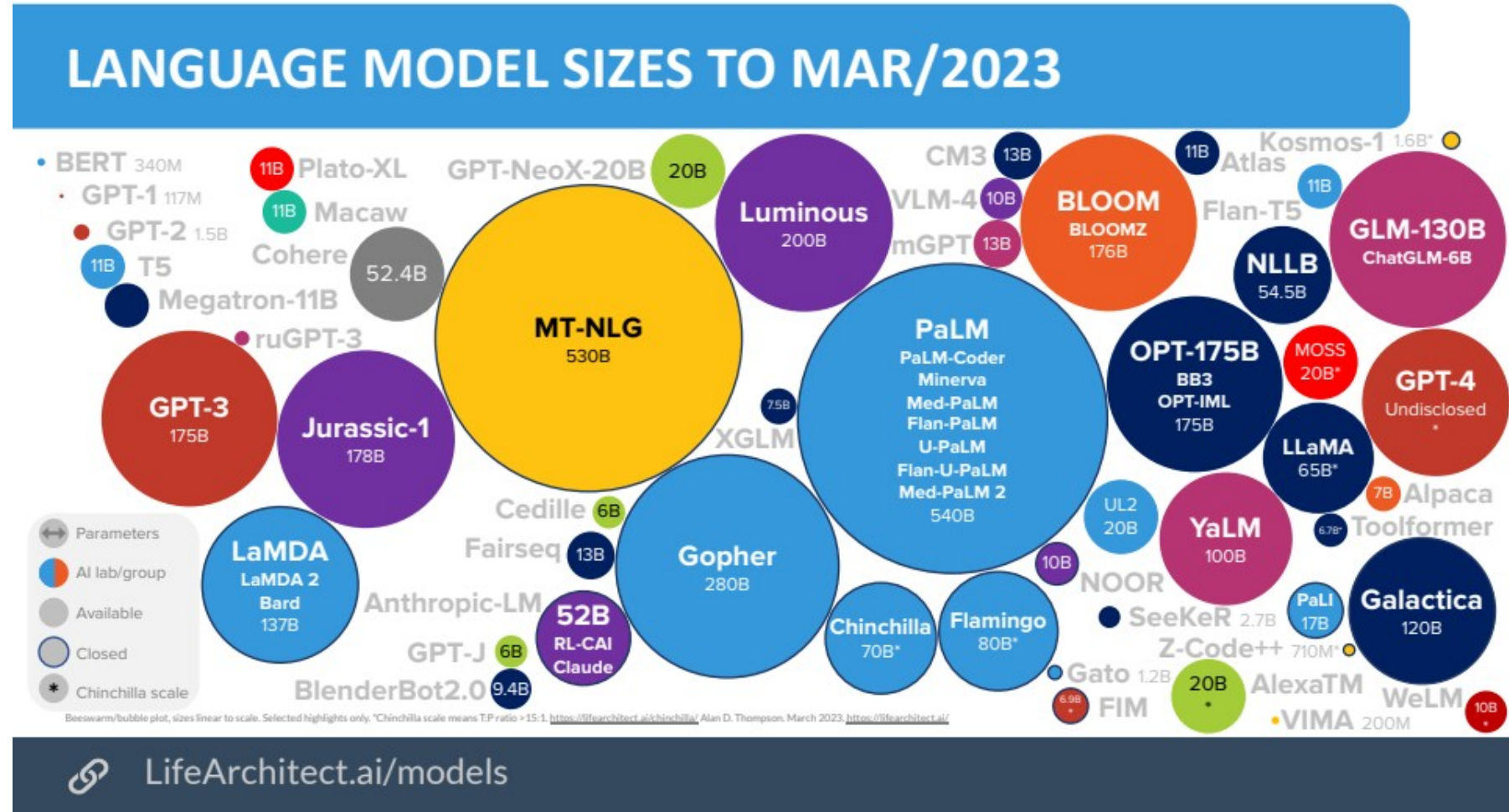
The diagram in the image is a Hertzsprung-Russell (H-R) diagram, which is extremely helpful for astronomers. Here's how:

- Star Classification:** The H-R diagram helps in classifying stars based on their luminosity, spectral type, color, temperature, and evolutionary stage. The main sequence, giants, supergiants, and white dwarfs are easily distinguishable.
- Understanding Stellar Evolution:** It provides insights into how stars evolve over time. For example, a star like the Sun will eventually move off the main sequence and become a red giant as it exhausts the hydrogen in its core.
- Estimating Distances:** By comparing a star's apparent magnitude with its absolute magnitude (as shown in the H-R diagram), astronomers can estimate its distance from Earth.
- Star Composition and Temperature:** The diagram shows the relationship between the temperature of stars and their luminosities. This helps astronomers infer the composition and age of the stars.

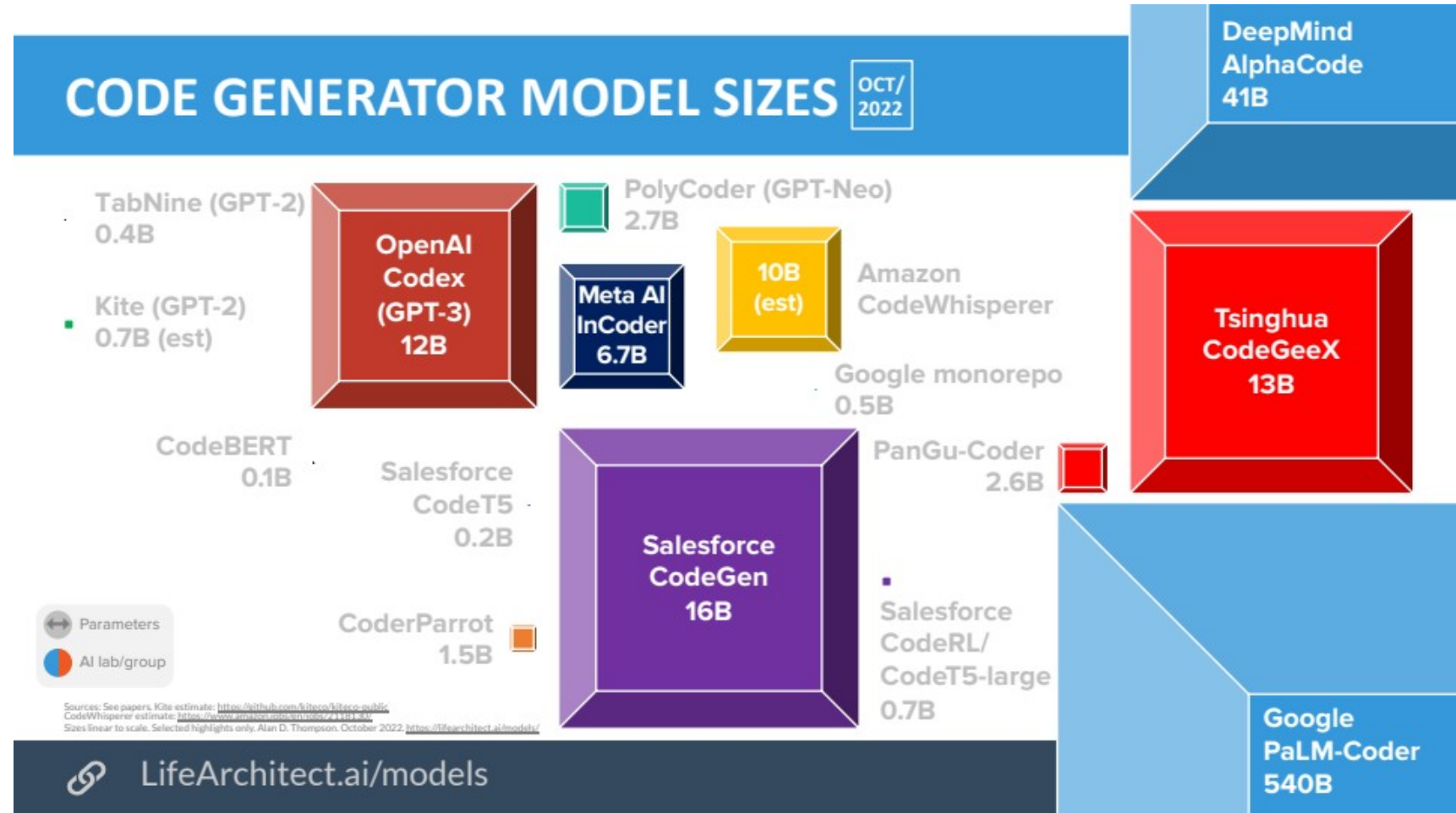
Send a message

Regenerate response

# AI at Scale



# AI at Scale



# GPT-4 Datenset

## DATASETS FOR LANGUAGE MODELS: SIMPLE VIEW

APR/  
2023

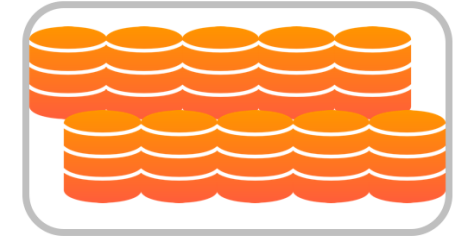
Close to scale. Selected highlights only. Using images from Flatiron.com. Alan D. Thompson, April 2023. <https://lifearchitect.ai>



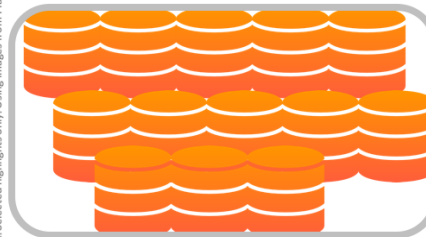
**GPT-3 dataset**  
499B tokens / 0.75TB



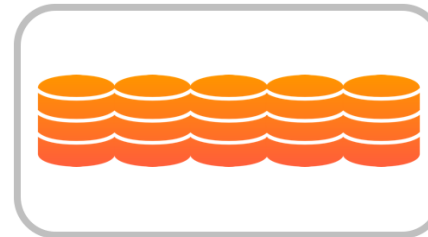
**The Pile dataset (GPT-Neo)**  
247B tokens / 0.8TB



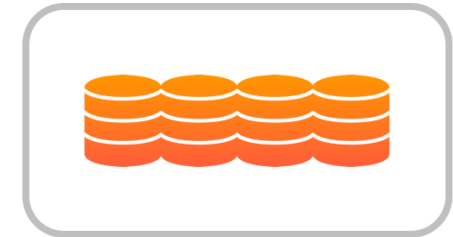
**MassiveText dataset (Chinchilla)**  
2.3T tokens / 10.5TB



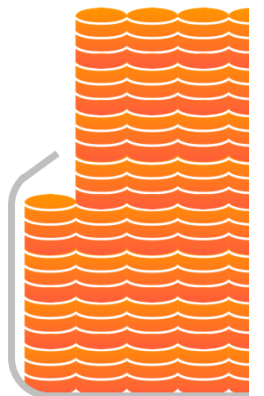
**Infiniset dataset (LaMDA)**  
2.8T tokens / 12.6TB



**Stability The Pile dataset**  
1.5T tokens / 5TB



**RedPajama dataset**  
1.2T tokens / 4TB



**GPT-4 (estimate)**  
20T tokens / 40TB



[LifeArchitect.ai/models](https://lifearchitect.ai/models)

Quelle: <https://lifearchitect.ai/models/>, Dr. Alan D. Thompson

# Generative KI

## Tasks

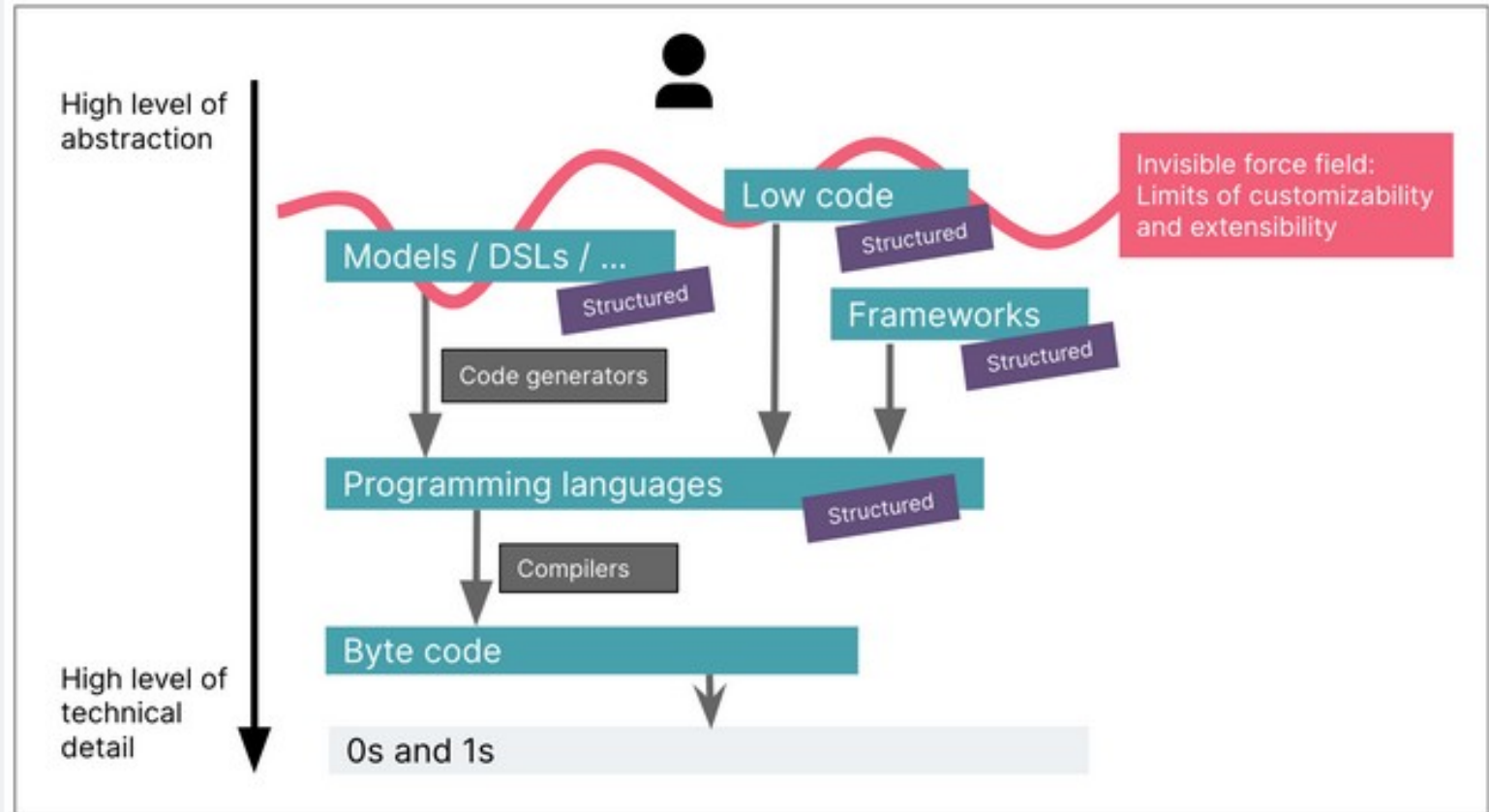
- Infos finden, schnell
- und im Kontext
- Code generieren
- Code erklären
- Code transformieren
- Tests schreiben
- Code optimieren
- generierten Code ausführen
- und debuggen



Tool	Tasks	Interaction	Prompt composition	Model	Origin / Hosting
<u>GitHub Copilot</u>	Code generation	In-line assistance	Composed by IDE extension	Trained with code, vulnerability filters	Commercial
<u>GitHub Copilot Chat</u>	All of them	Chat	Composed of user chat + open files	Trained with code	Commercial
<u>ChatGPT</u>	All of them	Chat	All done by user	Trained with code	Commercial
<u>GPT Engineer</u>	Code generation	CLI	Prompt composed based on user input	Choice of OpenAI models	Open Source, connecting to OpenAI API
<u>"Team AIs"</u>	All of them	Web UI	Prompt composed based on user input and use case	Most commonly with OpenAI's GPT models	Maintained by a team for their use cases, connecting to OpenAI APIs
<u>Meta's CodeCompose</u>	Code generation	In-line assistance	Composed by editor extension	Model fine-tuned on internal use cases and codebases	Self-hosted

# Unterschied zwischen Generativer AI und Code Generatoren

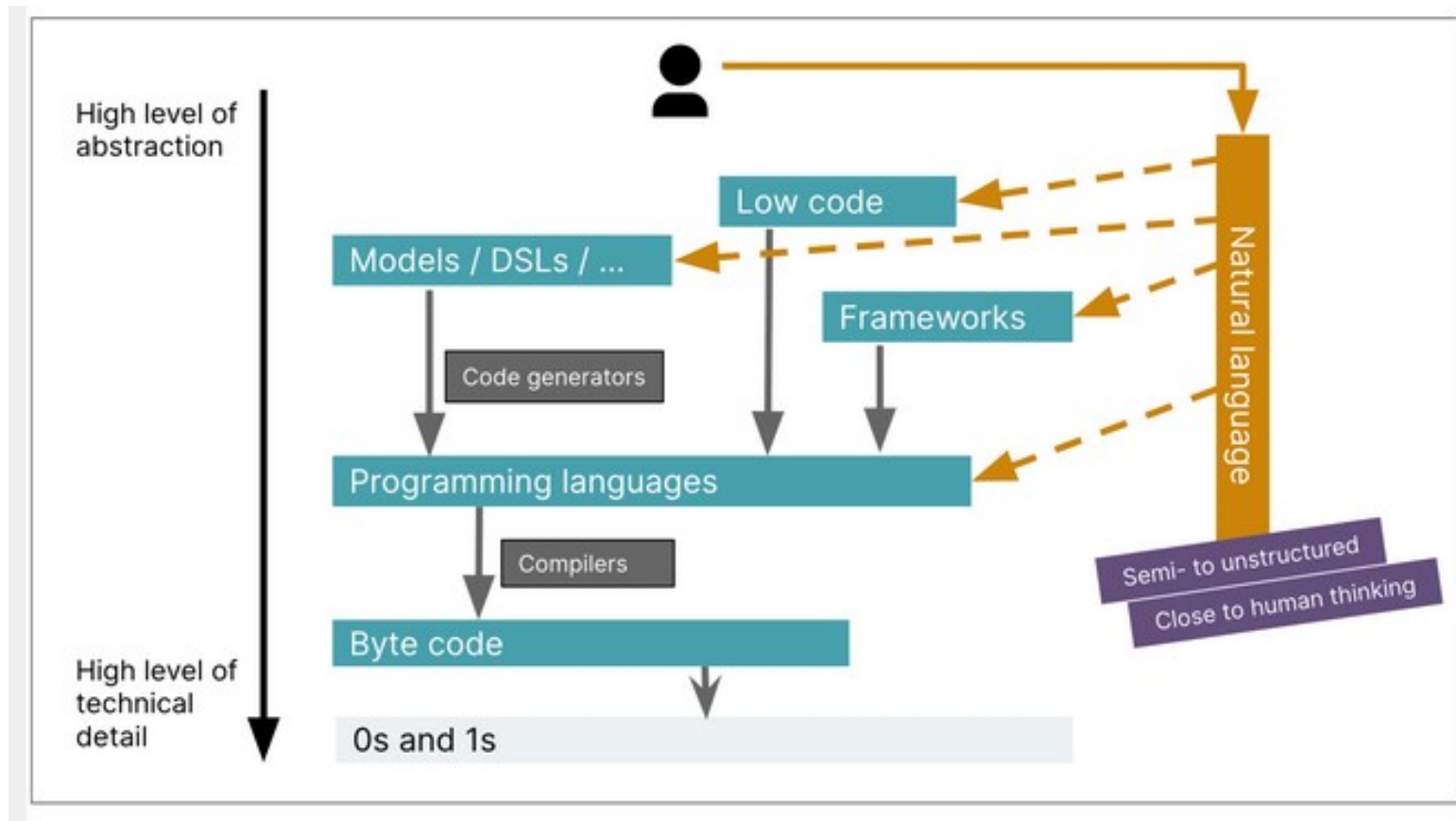
- Modelle dienen als Vertrag
- Modelle können auf unterschiedlichen Systemen für die Code-Generierung verwendet werden (Schnittstellenbeschreibung -> Client/Server)
- Modellsprache nicht intuitiv und ggf. schwer erlernbar
- Erzeugter Code kann nur über Modellanpassung modifiziert werden





# Unterschied zwischen Generativer AI und Code Generatoren

- Natürlichsprachliche Anweisungen/Eingabe von Bildern
- Modelle können aber sehr mächtig sein. Nichtdeterminismus.
- Kein Erlernen zusätzlicher Modellsprache notwendig
- Es fehlt die Modellebene, die zwischen Entwicklern und Systemen austauschbar ist



# 10 Things Software Developers should learn about learning?

Papier von Neil C.C. Brown, Felienn F.J. Hermans, and Laurien E.Margulieux

## » key insights

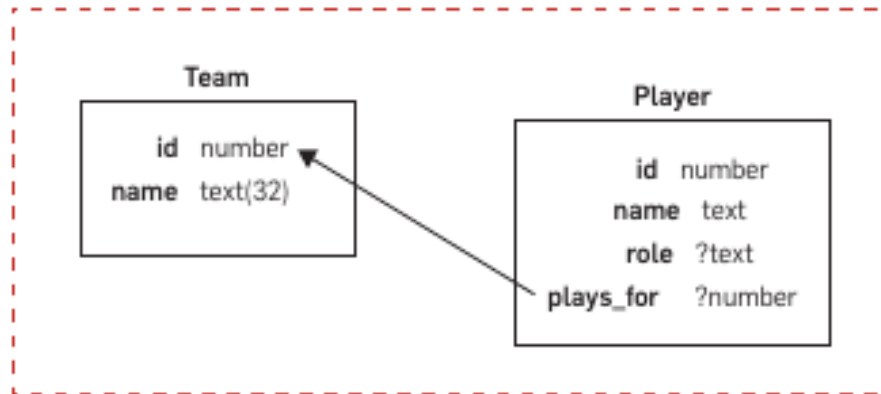
- Learning is vital for programmers, but the human mind works quite differently than a computer.
- Understanding how humans learn can help you learn more effectively.
- The Internet and LLMs have not made learning obsolete; learning is essential and takes time.
- Expertise changes how you think, letting you solve problems more easily but also potentially hindering your ability to teach.



# Human memory is not made of bits

Two ways of presenting the same database schema description with differing extraneous cognitive load

The dashed box on the left contains exactly the same information as the awkward textual description in the dashed box on the right. But if a developer only received one of the two to create an SQL database, they are likely to find the diagram easier than the text. We say that the text here has a higher extraneous cognitive load.

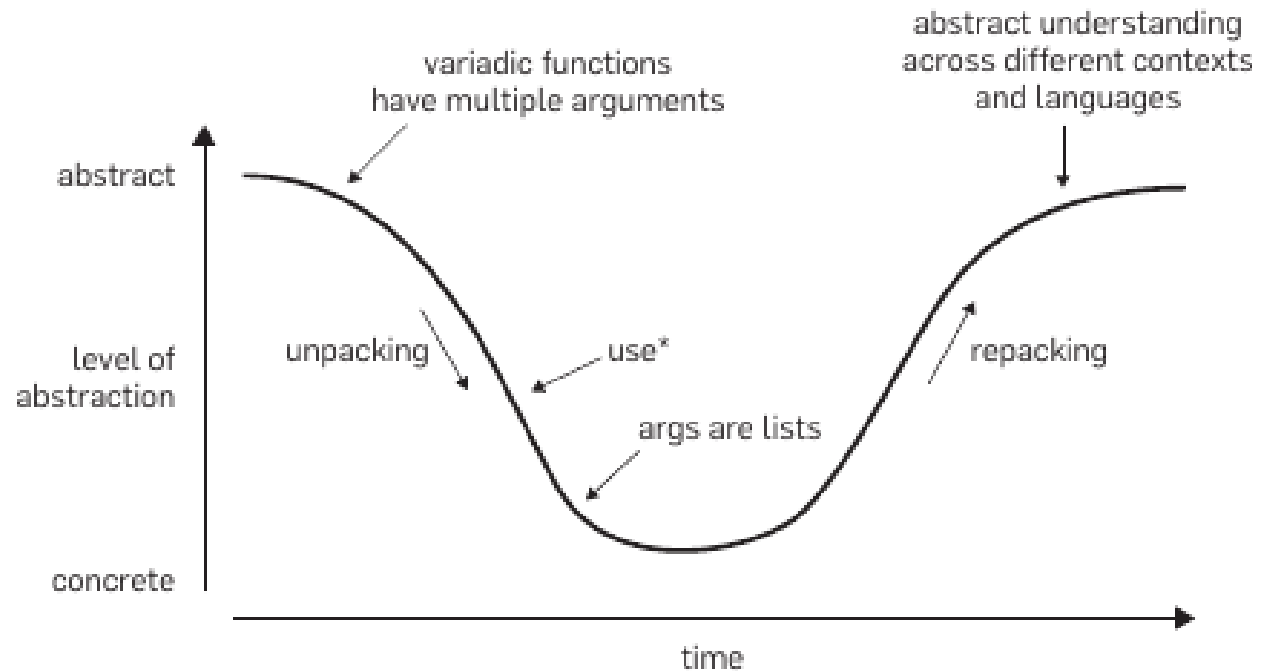


compared to

A team should have an id and a name. The name should be a text, the id should be numeric. The name should have a maximum length, which is 32. There are also players: a player should have an id (which, like teams, should be numeric), a name (that is text, but unlimited in length), and role (although the role can be missing), and a plays\_for which has the numeric id of their team. This link to the team can be missing.

# Expert developers can reason at a higher level by having memorized common patterns in program code, which frees up their cognition

Understanding a concept goes from abstract to concrete and back



# The Internet and LLMs have not made learning obsolete

Why is it worth learning details – or anything at all – if the knowledge is available from the Internet within seconds?

- The key distinction between a beginner who has never learned the details and thus lacks the memory connections, and an expert who has learned the deeper structure but searches for the forgotten fine details
- Searching the Internet and ChatGPT
  - the information was remembered less well
  - worse recall of the same information later, compared to first trying to think of the answer before resorting to searching
  - issue with cognitive load → switching between tasks, searching, and evaluating an AI-generated result

# Pairprogramming ist unersetzbar (noch)

## Benefits and purposes of Pair Programming

### 1+1 > 2

- Knowledge sharing
- Helps you get unstuck
- Onboarding
- Combine tactical and strategic modes of thinking

### Flow

- Keeping each other focussed
- Keep team's WIP (work in process") low

### Avoid waste

Pairing helps mitigate some of the "7 wastes of software development" (> M.&T. Poppendieck)

- Partially Done Work
- Extra Features
- Relearning
- Handoffs
- Delays
- Task Switching
- Defects

### Continuous Integration

(= developers integrate their work multiple times a day)

- Code review on-the-go and just-in-time
- Foster collective code ownership
- Make frequent integration practices like trunk-based development possible



"[working in pairs,] you can criticise each other, and find each other's errors, and use the best ideas" Jean Bartik, ENIAC programmer

### Practice skills needed on highly effective teams

Concentration

Time management

Empathy

Feedback

Task organisation

Communication

Benefits summarised from:

<https://www.youtube.com/watch?v=1C61GIBx6Yw>

<https://martinfowler.com/articles/on-pair-programming.html>

# Generative KI (LLM)

- Spezielle Architektur eines künstlichen neuronalen Netzes (Transformer)
  - Transformer bestehen aus Encodern und Decodern
    - Encoder verarbeitet Eingabedaten zu einer internen Repräsentation
    - Decoder nutzt Encoder Repräsentation um Ausgabe zu generieren
    - Beispiel: Sequence-to-Sequence Modell für die Übersetzung von Deutsch in Englisch

# Generative KI (LLM)

- GPT-Modelle bestehen nur aus Decodern \*
- Aufgabe: Was ist das nächstwahrscheinliche Wort (bzw. Token) \* in einem Kontext
- GPT-3 Vokabular Größe beträgt 50.257 Token

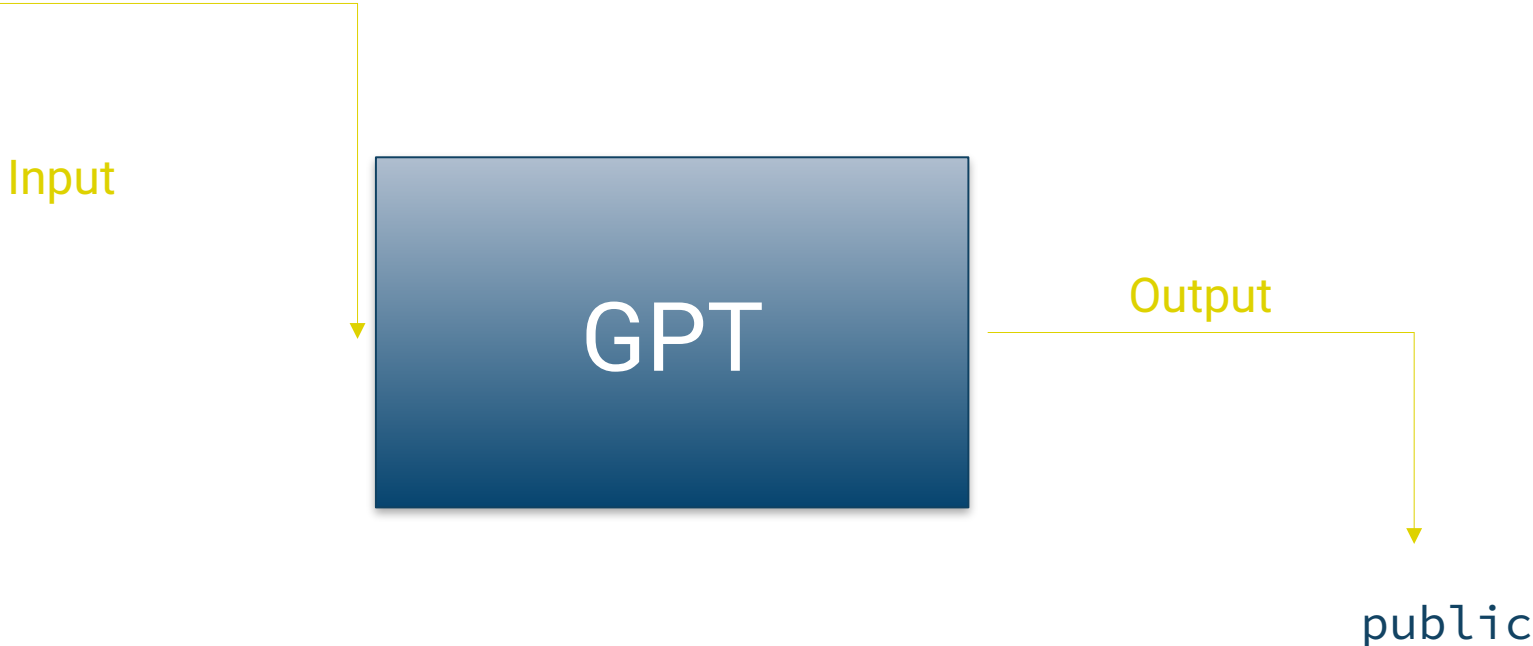


- \* Die Google-Suche ist ein Beispiel für eine Anwendung, die nur aus Encodern besteht (BERT)
- \* Token sind „Wortfetzen“ bzw. Silben, die mit einem Algorithmus berechnet wurden. Vor der Eingabe in GPT werden die Token in Zahlenwerte umgewandelt



# Generative KI

„Erstelle mir ein Hello World Programm in Java“



# Generative KI

„Erstelle mir ein Hello World Programm in Java“ `public`

Input



Output

`static`

# Generative KI

„Erstelle mir ein Hello World Programm in Java“ `public`

Input



Output

`static`

# Generative KI

„Erstelle mir ein Hello World Programm in Java“

```
public static void main(String args[]) { System.out.println(„Hello
```

Input



Output

World

# Einsatzgebiete

# Produktivität & Effizienz

- Unterstützung bei repetitiven Aufgaben (durch Automatisierung), Boilerplate code, übliche Muster (Mapping, Grouping, Sorting in Streams)
- Code completion
- Generierung von Code
- Debugging
- Erstellen von Proof-of-Concepts und Prototypen

Aber Vorsicht: Automation Bias, Sunk Cost Fallacy, Anchoring Effect

# Qualität & Sicherheit

- Überprüfung auf Code-Standards
- Code Dokumentation
- Code Reviews
  - Vermeidung von Syntaxfehlern
  - Identifikation von Sicherheitslücken
- Automatisierte Codeformatierung
- Generierung von Unit-Tests



# Individuell angepasste Trainings

- Analyse des „Skill-Sets“ und „Lerntyps“
- Erstellung eines individuellen Lernplans sowie kleinen Aufgaben
- Steigerung der Effizienz beim Lernen?



# Repetitive Aufgaben

# Repetitiven Aufgaben & Unit-Tests

“

*Kannst du mir ein Webformular erstellen, das ein Passwort speichert?*

*Wichtig ist, dass das Passwort eine **Mindestlänge von 8 Buchstaben** hat, mindestens **eine Ziffer** enthält, mindestens **ein Sonderzeichen** enthält sowie **Klein- und Großbuchstaben** enthalten sind? Entspricht das Passwort nicht den Vorgaben, so soll das Formular nicht absendbar sein.*

”

# Repetitiven Aufgaben & Unit-Tests

```
// Passwortvalidierungsfunktion (zum Testen)
function validatePassword(password) {
  if (password.length < 8) {
    return false;
  }
  if (!/\d/.test(password)) {
    return false;
  }
  if (!/[!@#%&*()_+[]\];;<>,.?~\|-]/.test(password)) {
    return false;
  }
  if (!/[a-z]/.test(password) || !/[A-Z]/.test(password)) {
    return false;
  }
  return true;
}
```

Mindestlänge von 8 Buchstaben

Mindestens eine Zahl

Mindestens ein Sonderzeichen

Klein- und Großbuchstaben

# Repetitiven Aufgaben & Unit-Tests

```
// Passwortvalidierungsfunktion (zum Testen)
function validatePassword(password) {
  if (password.length < 8) {
    return false;
  }
  if (!/^\d/.test(password)) {
    return false;
  }
  if (!/[A-Z]/.test(password)) {
    return false;
  }
  if (!/[a-z]/.test(password) || !/[!@#$%&*/.~?]/.test(password)) {
    return false;
  }
  return true;
}
```

Kannst du mir einen **Unit-Test** für die von dir vorgeschlagene Funktion "**validatePassword()**" erstellen?

Mindestlänge von 8 Buchstaben

“

Mindestens eine Zahl

Mindestens ein Sonderzeichen

”

Klein- und Großbuchstaben

# Repetitiven Aufgaben & Unit-Tests

```
// Passwortvalidierungsfunktion (zum Testen)
function validatePassword(password) {
  if (password.length < 8) {
    return false;
  }
  if (!/\d/.test(password)) {
    return false;
  }
  if (!/[!@#%&*()_+[]\|:;<>.,?~\-\]/.test(password)) {
    return false;
  }
  if (!/[a-z]/.test(password) || !/[A-Z]/.test(password)) {
    return false;
  }
  return true;
}
```

```
// Jest-Testfall
describe('validatePassword', () => {
  it('should return true for a valid password', () => {
    const validPassword = "Passw0rd!";
    expect(validatePassword(validPassword)).toBe(true);
  });
  it('should return false for a password that is too short', () => {
    const shortPassword = "Sh0rt!";
    expect(validatePassword(shortPassword)).toBe(false);
  });
  it('should return false for a password with no digits', () => {
    const noDigitPassword = "NoDigit!";
    expect(validatePassword(noDigitPassword)).toBe(false);
  });
  it('should return false for a password with no special character', () => {
    const noSpecialCharPassword = "NoSpecialChar123";
    expect(validatePassword(noSpecialCharPassword)).toBe(false);
  });
  it('should return false for a password with no lowercase letters', () => {
    const noLowercasePassword = "NOLOWERCASE1!";
    expect(validatePassword(noLowercasePassword)).toBe(false);
  });
  it('should return false for a password with no uppercase letters', () => {
    const noUppercasePassword = "nouppercase1!";
    expect(validatePassword(noUppercasePassword)).toBe(false);
  });
});
```

\* Zusätzlich wurden weitere JS-Testframeworks wie „Mocha“ und „Jasmine“ vorgeschlagen.  
+ die vorgeschlagenen Testpasswörter sind echt cool

Jest

# Fehlerbehebung

# Syntax-Fehlerbehebung

- SpringBoot Anwendung für Hotelreservierungen
- Fehler 1: RequestMapping Annotation fehlt
- Fehler 2: PathVariable in GetMapping falsch definiert

# Fehler (1/2) - RequestMapping

“

Ich habe folgenden **Endpoint**, allerdings lässt dieser sich **nicht aufrufen**:

```
@RestController
public class HotelReservationController {
    @Autowired
    private HotelReservationService reservationService;

    @GetMapping("/{id}")
    public HotelReservation getReservation(@PathVariable Long id) {
        return reservationService.getReservation(id);
    }
}
```

”



# Fehler (1/2) - RequestMapping

\* Es scheint, dass Ihr HotelReservationController einen **Endpunkt für das Abrufen einer Hotelreservierung nach der ID definiert**<sup>1</sup>, aber es **fehlt der Pfad, auf dem dieser Endpunkt erreichbar sein soll**<sup>2</sup>. Um den Controller korrekt zu konfigurieren, sollten Sie den Pfad in der **@GetMapping-Annotation hinzufügen**.

1. ChatGPT versteht mein Vorhaben
2. Erkennt den Fehler

# Fehler (1/2) - RequestMapping

Hier ist eine aktualisierte Version des Controllers:

```
@RestController
@RequestMapping("/reservations") // Pfad für alle Endpunkte in diesem Controller
public class HotelReservationController {
    @Autowired
    private HotelReservationService reservationService;

    @GetMapping("/{id}")
    public HotelReservation getReservation(@PathVariable Long id) {
        return reservationService.getReservation(id);
    }
}
```



In diesem Beispiel ist **der Pfad für alle Endpunkte in diesem Controller /reservations**.

Dies bedeutet, dass der **getReservation-Endpunkt unter der URL /reservations/{id}** erreichbar sein wird.

Beachten Sie, dass der **{id}-Teil in der URL als Pfadvariable** behandelt wird und an die Methode **getReservation** übergeben wird.

# Fehler (2/2) - PathVariable

“

Ich habe folgenden **Endpoint**, allerdings lässt dieser sich **nicht aufrufen**:

```
@RestController
@RequestMapping("/reservations")
public class HotelReservationController {
    @Autowired
    private HotelReservationService reservationService;

    @GetMapping("/id")
    public HotelReservation getReservation(@PathVariable Long id) {
        return reservationService.getReservation(id);
    }
}
```

”

# Fehler (2/2) - PathVariable



Es sieht so aus, als ob Ihre **@GetMapping-Annotation** in Ihrem Controller den Pfad **fehlerhaft definiert**. Der Pfad sollte **{id}** in **geschweiften Klammern** haben, um zu kennzeichnen, dass es sich um eine Pfadvariable handelt.

# Fehler (2/2) - PathVariable

Hier ist eine aktualisierte Version des Controllers:

```
@RestController
@RequestMapping("/reservations")
public class HotelReservationController {
    @Autowired
    private HotelReservationService reservationService;

    @GetMapping("/{id}")
    public HotelReservation getReservation(@PathVariable Long id) {
        return reservationService.getReservation(id);
    }
}
```

- \* Mit dieser Änderung wird der **getReservation-Endpunkt** unter der URL **/reservations/{id}** erreichbar sein, wobei **{id}** eine **Pfadvariable** ist, die den Wert aus der URL extrahiert.

# Code Optimierung

# Code Optimierungen

- Fachlogik für Hotelreservierungen
- Optimierung 1:
  - Anzahl der Nächte
  - Rabatt für VIP-Kunden
- Optimierung 2:
  - Filtern von VIP-Buchungen
  - Sortieren nach Reservierungsdatum

# Code Optimierungen (1/2)

```
public double calculateTotalPrice(int basePrice, int numberOfNights, int discount, boolean isVip) {  
    double total = 0;  
    // Mehr als 3 Nächte  
    if (numberOfNights >= 3) {  
        total = basePrice * numberOfNights;  
        // VIP Rabatt  
        if (isVip) {  
            total -= discount;  
        }  
    }  
    // Weniger als 3 Nächte -> Preis um Faktor 1.5 erhöhen  
    } else {  
        total = basePrice * numberOfNights * 1.5;  
        // VIP Rabatt  
        if (isVip) {  
            total -= discount / 2;  
        }  
    }  
    }  
    return total;  
}
```



# Code Optimierungen (1/2)

```
public double calculateTotalPrice(int basePrice, int numberOfNights, int discount, boolean isVip) {  
    double total = 0;  
    // Mehr als 3 Nächte  
    if (numberOfNights >= 3) {  
        total = basePrice * numberOfNights;  
        // VIP Rabatt  
        if (isVip) {  
            total -= discount;  
        }  
    }  
    // Weniger als 3 Nächte -> Preis um Faktor 1.5 erhöhen  
    } else {  
        total = basePrice * numberOfNights * 1.5;  
        // VIP Rabatt  
        if (isVip) {  
            total -= discount / 2;  
        }  
    }  
    return total;  
}
```

## ChatGPT Optimierung

```
// Berechnung Grundpreis  
double total = basePrice * numberOfNights;  
// Weniger als 3 Nächte -> Preis erhöhen  
if (numberOfNights < 3) {  
    total *= 1.5; discount /= 2;  
}  
// VIP Rabatt  
if (isVip) {  
    total -= discount;  
}  
return total;
```

# Code Optimierungen (2/2)

```
// Init Liste mit VIP-Hotelreservierungen
List<HotelReservation> vipReservations = new ArrayList<>();

// Suchen nach VIP-Hotelreservierungen
for (HotelReservation reservation : reservations) {
    if (reservation.isVip()) {
        vipReservations.add(reservation);
    }
}
// Sortieren nach Reservierungsdatum
Collections.sort(vipReservations, (r1, r2) -> r1.getReservationDate().compareTo(r2.getReservationDate()));
```

# Code Optimierungen (2/2)

```
// Init Liste mit VIP-Hotelreservierungen
List<HotelReservation> vipReservations = new ArrayList<>();

// Suchen nach VIP-Hotelreservierungen
for (HotelReservation reservation : reservations) {
    if (reservation.isVip()) {
        vipReservations.add(reservation);
    }
}
// Sortieren nach Reservierungsdatum
Collections.sort(vipReservations, (r1, r2) -> r1.getReservationDate().compareTo(r2.getReservationDate()));
```

## ChatGPT Optimierung

```
// Anwendung von Stream API
List<HotelReservation> sortedVipReservations = reservations.stream()
    .filter(HotelReservation::isVip)
    .sorted(Comparator.comparing(HotelReservation::getReservationDate))
    .collect(Collectors.toList());
```

# Code Dokumentation

# Code Dokumentation (1/2)



Kannst du mir folgenden **Code dokumentieren**:

```
public double calculateTotalPrice(int basePrice, int numberOfNights, int discount, boolean isVip) {
    double total = basePrice * numberOfNights;
    if (numberOfNights < 3) {
        total *= 1.5; discount /= 2;
    }
    if (isVip) {
        total -= discount;
    }
    return total;
}
```



# Code Dokumentation (1/2)

```
/**
 * Diese Methode berechnet den Gesamtpreis einer Hotelbuchung auf der Grundlage
 * des Grundpreises, der Anzahl der Übernachtungen, eines Rabatts und des VIP-Status.
 *
 * @param basePrice      Der Grundpreis pro Übernachtung.
 * @param numberOfNights Die Anzahl der Übernachtungen in der Buchung.
 * @param discount       Der Rabatt, der auf den Gesamtpreis angewendet wird.
 * @param isVip         Ein Flag, das anzeigt, ob der Gast ein VIP ist.
 * @return              Der berechnete Gesamtpreis der Buchung.
 */
public double calculateTotalPrice(int basePrice, int numberOfNights, int discount, boolean isVip) {
    // Der Gesamtpreis wird zunächst auf den Grundpreis multipliziert.
    double total = basePrice * numberOfNights;

    // Wenn die Anzahl der Übernachtungen weniger als 3 beträgt, wird der Gesamtpreis um den Faktor 1,5 erhöht,
    // und der Rabatt wird halbiert. Dies entspricht einer Sonderregelung für kürzere Aufenthalte.
    if (numberOfNights < 3) {
        total *= 1.5;
        discount /= 2;
    }

    // Wenn der Gast ein VIP ist, wird der gewährte Rabatt vom Gesamtpreis abgezogen.
    if (isVip) {
        total -= discount;
    }

    // Der berechnete Gesamtpreis wird zurückgegeben.
    return total;
}
```



# Code Dokumentation (2/2)

“

Kannst du mir folgenden **Code dokumentieren**:

```
public HotelReservation updateReservation(Long id, HotelReservation newReservation) {  
    HotelReservation existingReservation = getReservation(id);  
  
    existingReservation.setGuestName(newReservation.getGuestName());  
    existingReservation.setCheckInDate(newReservation.getCheckInDate());  
    existingReservation.setCheckOutDate(newReservation.getCheckOutDate());  
  
    return reservationRepository.save(existingReservation);  
}
```

”

# Code Dokumentation (2/2)

```
/**
 * Aktualisiert eine vorhandene Hotelreservierung anhand ihrer eindeutigen ID mit den Informationen
 * aus der neuen Reservierung.
 *
 * @param id           Die eindeutige ID der zu aktualisierenden Reservierung.
 * @param newReservation Eine Instanz der HotelReservierung, die die aktualisierten Informationen enthält.
 * @return            Die aktualisierte Hotelreservierung, wie sie in der Datenbank gespeichert wurde.
 * @throws NotFoundException Wenn keine Reservierung mit der angegebenen ID gefunden wurde.
 */
public HotelReservation updateReservation(Long id, HotelReservation newReservation) {
    // Suchen Sie die vorhandene Reservierung anhand ihrer eindeutigen ID.
    HotelReservation existingReservation = getReservation(id);

    // Aktualisieren Sie die Informationen der vorhandenen Reservierung mit den Werten aus der neuen Reservierung.
    existingReservation.setGuestName(newReservation.getGuestName());
    existingReservation.setCheckInDate(newReservation.getCheckInDate());
    existingReservation.setCheckOutDate(newReservation.getCheckOutDate());

    // Speichern Sie die aktualisierte Reservierung in der Datenbank mithilfe des Repositorys.
    return reservationRepository.save(existingReservation);
}
```





# User Stories, Epics & Tasks

# User Stories, Epics & Tasks

Du bist **Product Owner** und sollst **Epics, Akzeptanzkriterien, User Stories** sowie **Tasks** für die **Stornierungsfunktionalität meiner Hotelreservierungen** Anwendung definieren.

# User Stories, Epics & Tasks

**Benutzerstory 1:** Als Benutzer möchte ich eine Hotelreservierung stornieren können

## Tasks

- Implementiere eine **Schnittstelle** in der Präsentationsschicht, um die **Stornierung anzufordern**.
- Implementiere eine **Methode** in der Anwendungsschicht, um die **Stornierung durchzuführen**.
- Validiere, ob der **Benutzer die Berechtigung zur Stornierung** hat.
- Validiere, ob die **Reservierung in der Zukunft** liegt.
- **Aktualisiere die Datenbank**, um die Stornierung zu reflektieren.
- Zeige dem Benutzer eine **Bestätigungsnachricht** an.

# User Stories, Epics & Tasks

**Benutzerstory 1:** Als Benutzer möchte ich eine Hotelreservierung stornieren können

## Akzeptanzkriterien

1. Der Benutzer hat die Möglichkeit, eine **Reservierung zu stornieren**, indem er auf die **Stornierungsoption auf der Buchungsseite klickt**.
2. Die Benutzeroberfläche zeigt eine **Bestätigungsnachricht** an, die den Benutzer über die erfolgreiche Stornierung informiert.
3. Die **Stornierung** kann nur von dem **Benutzer** durchgeführt werden, der die **ursprüngliche Buchung vorgenommen** hat.
4. Die Stornierung kann **nur für Reservierungen in der Zukunft** durchgeführt werden. Stornierungen für vergangene Aufenthalte sind nicht zulässig.

# User Stories, Epics & Tasks

**Benutzerstory 2:** Als Benutzer möchte ich den Gesamtpreis meiner Reservierung nach der Stornierung sehen.

- Implementiere eine Methode zur **Aktualisierung des Gesamtpreises nach der Stornierung**.
- Validiere, ob **Rabatte oder Sonderangebote betroffen** sind.
- **Aktualisiere die Datenbank** mit dem neuen Gesamtpreis.

## Akzeptanzkriterien

1. Nach der Stornierung wird der Benutzer auf der **Buchungsseite den aktualisierten Gesamtpreis** seiner Reservierung sehen.
2. Wenn **Rabatte** oder **Sonderangebote** auf die ursprüngliche Buchung angewendet wurden, sollte der **aktualisierte Preis** diese Änderungen berücksichtigen.

# User Stories, Epics & Tasks

**Benutzerstory 3:** Als Benutzer möchte ich nach einer Stornierung ein Zimmer wieder buchen können.

- **Aktualisiere den Status des stornierten Zimmers** in der Datenbank, um es wieder buchbar zu machen.

## Akzeptanzkriterien

1. **Nach der Stornierung** wird das stornierte Zimmer **wieder in der Liste der verfügbaren Zimmer** auf der Buchungsseite angezeigt.
2. Der Benutzer kann das zuvor **stornierte Zimmer erneut buchen**, wenn es verfügbar ist.

# User Stories, Epics & Tasks

**Benutzerstory 4:** Als Administrator möchte ich Zugriff auf Stornierungsprotokolle haben.

- Implementiere eine Protokollierung der Stornierungen für Administratoren.

## Akzeptanzkriterien

1. Der Administrator hat Zugriff auf eine **Übersichtsseite** oder ein Dashboard, auf dem **die Protokolle der Stornierungen angezeigt** werden.
2. Die **Stornierungsprotokolle** enthalten **Informationen wie Benutzername, Reservierungsdaten, Datum und Uhrzeit der Stornierung**.
3. Die Protokolle sind für den Administrator leicht **durchsuchbar** und **filterbar**, um spezifische Informationen zu finden.
4. Es gibt **Sicherheitsmaßnahmen**, um sicherzustellen, dass **nur autorisierte Administratoren Zugriff** auf die Stornierungsprotokolle haben.

# Herausforderungen & Fazit



# Herausforderungen & Fazit

- KI kann Entwicklungsprozess deutlich beschleunigen
  - Codegenerierung durch natürliche Sprache
- Bringt eine neue Sicht auf Aufgaben, Ideen zum Design von Tests
  - Schlägt neue Technologien vor
- Dient als Knowledge Base
- **Versteht eigentlich nicht was es macht und auch nicht den eigentlichen Dialog**
- Antworten hängen stark von Datensatz und Datenaktualität ab
- Datenschutz