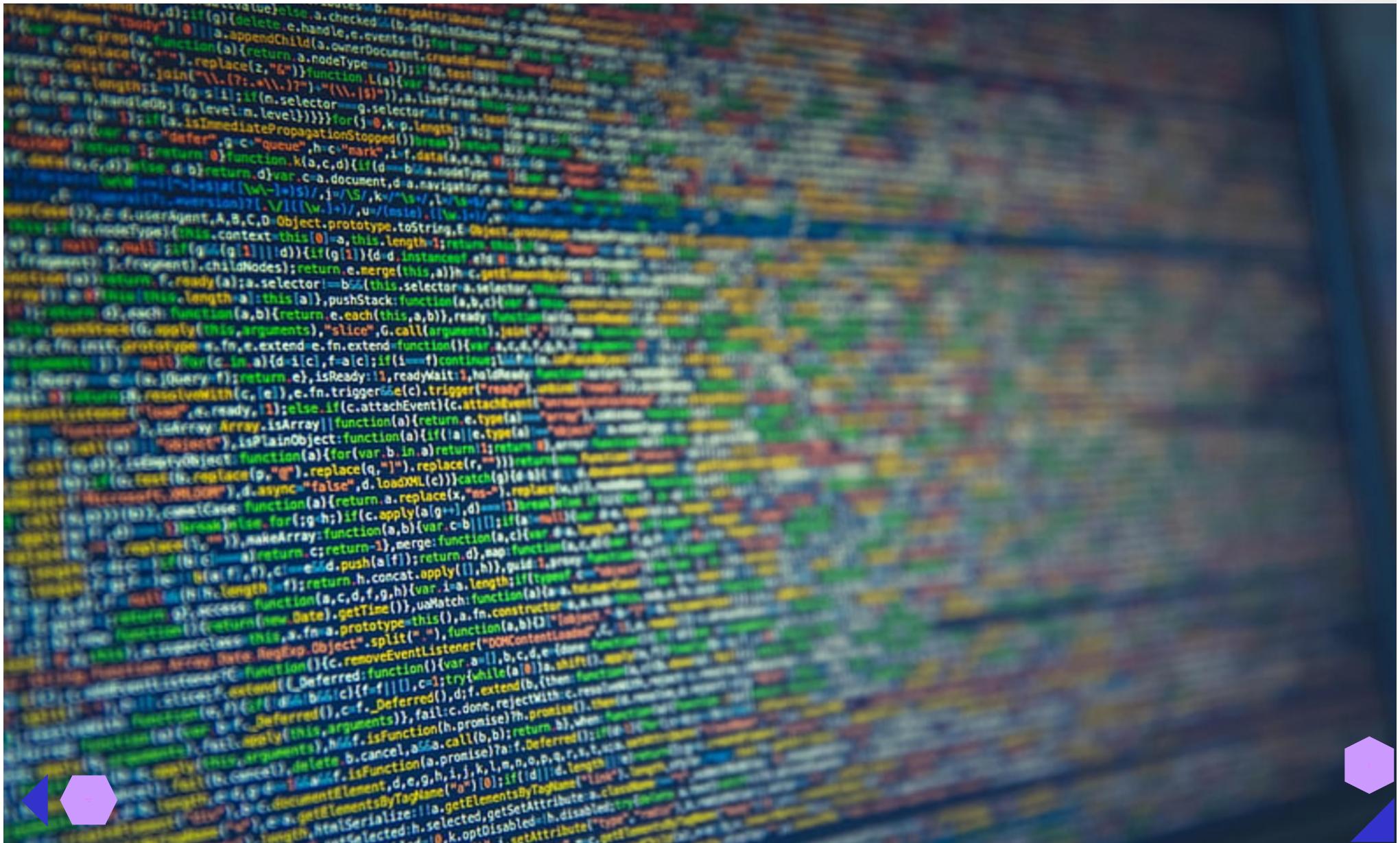


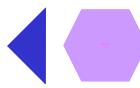
Introduction



Computer Science (TU Wien) – Focus: **Artificial Intelligence**
Economics (WU Wien) – Focus: **Modeling of Complex Systems**
Software Industry:



- **Software engineering roles:** requirements engineer, designer, architect, modeler, developer, tester, project manager, product manager, quality manager, defect manager, ...
- **Application domains:** banking, insurance, health, logistics, mobile telephony, landline telephony, culture event management, military, data forensics, public transport, waterway management, sales, customer relationship management,...
- **Elapsed time:** days to decades
- **Project sizes:** very small to > 1000 person years
- **Activity types:** initial development and maintenance
- **AI Projects:** ML for energy consumption, rollout AI Act, adaptive book proposal to customers, book proposal to libraries, several chatbot projects (finance, consulting, ...), ML-based cost estimation, ...



Software Engineering



What is Software Engineering?

Software Engineering is about **systematic, methodical** creation of software.
It covers the **whole** software development process, in particular:

Requirements Engineering

Software Architecture

Design, Development, Deployment

Testing and Evaluation

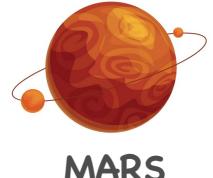
Software Engineering is not the same thing as Advanced Programming.

Still, you will see many code snippets in the presentations.

They are **reduced to the essentials** and need supplementing in order to be executable.



What Does NASA See as Software Engineering?



NASA Procedural Requirements

COMPLIANCE IS MANDATORY FOR NASA EMPLOYEES

NPR 7150.2D

Effective Date: March 08, 2022

Expiration Date: March 08, 2027

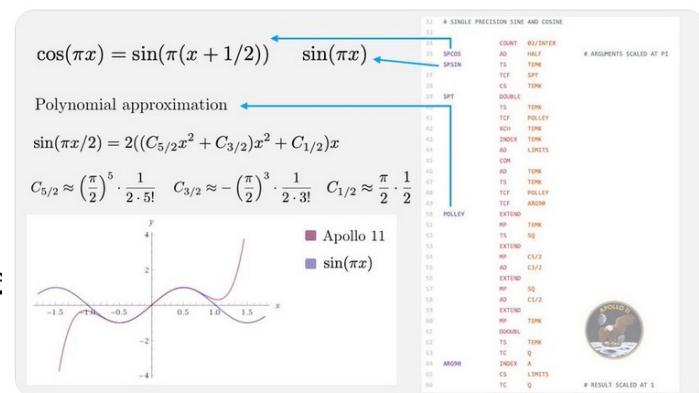
NASA Software Engineering Requirements

Chapter 4. Software Engineering

- 4.1 Software Requirements
- 4.2 Software Architecture
- 4.3 Software Design
- 4.4 Software Implementation
- 4.5 Software Testing
- 4.6 Software Operations, Maintenance, and Retirement



In 1969 Apollo 11, the spaceflight that first landed humans on the Moon, used these 30 lines of code to calculate transcendental functions like sine and cosine essential for navigation.



Sources: https://lws.larc.nasa.gov/vfmo/pdf_files/N_PR_7150_002D_.pdf, 2015,
<https://x.com/fermatslibrary/status/1963564035535777809?s=46&t=liJ6-bCACnb-WfoI5TZ2RA>, 2025



The Most Important Skills in Software Engineering



Santiago ✅
@svpino

Subscribe ⚡ ...

Everyone wants to learn how to code.

But coding is secondary.

Your ability to analyze a problem is the most important skill you can build.

Writing code is easy; most of the work is understanding what problem to solve and how to do it elegantly.

Software
Architecture is
needed.

Requirements
Engineering is
needed.



Logan Thorneloe ✅ @loganthoneloe · Jun 10

+1



1



153



Arpit Sharma ✅ @Arp_it1 · Jun 10

100%. Most bugs I've seen weren't bad code. They were bad thinking.



1



277



casslin ✅ @thulynnn · Jun 10

~~100~~ knowing the right problem to ask = almost all u need now.



1



172



Kade ✅ @kademeta · Jun 11

Why do you think coding is still such a valued assessment as part of interviewing for SE positions?



1



21



Home ✅ @homeMetaX · Jun 10

Totally agree!! Problem solving trumps coding skills any day

Source: <https://x.com/svpino/status/1932335424720306525?s=46>, 2025

Goal of this Course



Most software failures aren't from bad code.

They're from bad decisions.

Common engineering mistakes:

1. Over-engineering – Complexity kills velocity. **Build what you need**, not what you imagine.
2. Ignoring the basics – Logs, monitoring, and error handling aren't optional.
3. **Skipping tests** – If you don't test it, your users will.
4. **Tightly coupling everything** – Future refactors shouldn't feel like surgery.
5. Chasing trends – The right tool > the latest tool.

Coding is not the most important part of SE.

Requirements Engineering is needed.

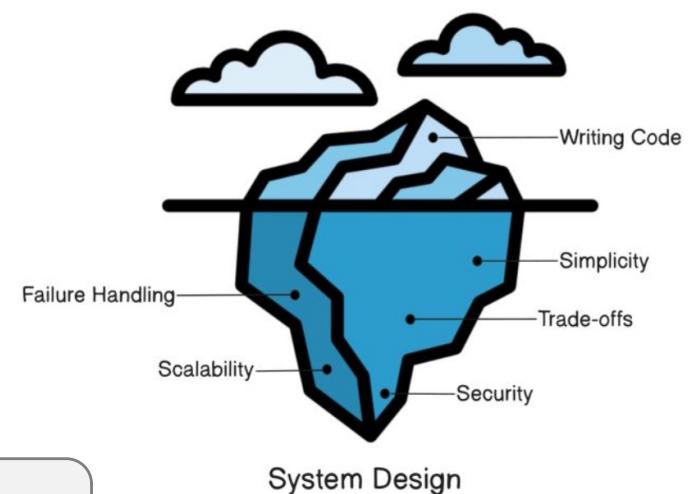
Testing is needed.

Software Architecture is needed.

Goal of this course:
learning to be a great software engineer.

Good engineers write code.
Great engineers make good decisions.

@rauljuncov



Good engineers write code. Great engineers make good decisions.



Source: <https://x.com/rauljuncov/status/1901607136741687564?s=46&t=liJ6-bCACnb-WfoI5TZ2RA>, 2025

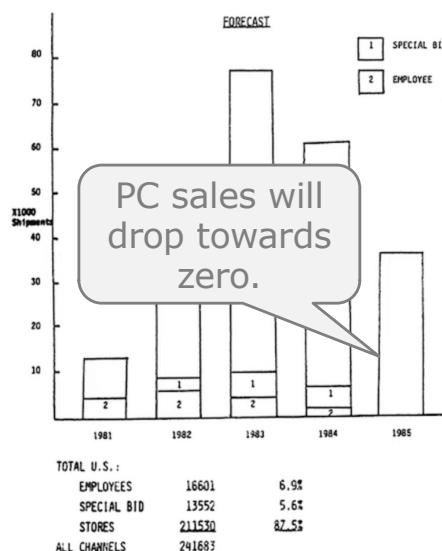


Artificial Intelligence meets Software Engineering

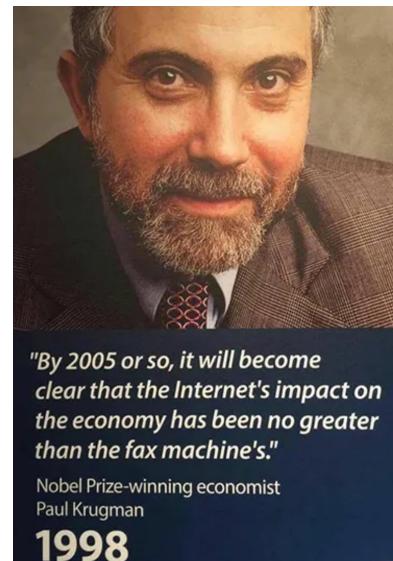


A Short History of Underestimating New Information Technologies

The PC



The Internet



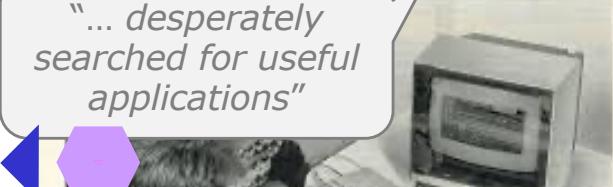
Smartphones



Die Enttäuschung ist vorprogrammiert

Kleine Denksportaufgabe: Man braucht es nicht und trotzdem wird es wie verrückt gekauft. Was ist das? Ganz einfach: ein Heimcomputer. Wir prüften sieben Modelle und suchten verzweifelt nach sinnvollen Einsatzmöglichkeiten. Unser Fazit: Wer auf die elektronische Auslastung seines Heimes

"... desperately searched for useful applications"



The Internet
will not become
a mass
medium.

NETZPOLITIK
"Internet wird kein Massenmedium"
Zukunftsrecherchen dämpfen Hoffnung auf weiteres Internet-Wachstum

2. März 2001, 1
Das Inter-
Zeit nich-
Fernseh-
Telefon o-
WWW m-
Angele-
Hörx die
Internets.
Jahre dämpft der Gründer des Hamburger
"Zukunftsinstituts". Der Anteil der Menschen, die das
weltweite Datennetz nutzen, werde zwar steigen, nicht
aber die Breitennutzung.

Large Language
Models are
impossible.

Whatever the other int-
approximation in this sense
be, it is clear
that it can shed no light on
problems of
grammar. There is no gener-
frequency of a string (or
its grammaticalness)
by considering suc-

(14) color

which is a gram-
ma-
fair to assume the
have occurred to
speakers of English
ordinary intonation
while he will read

(15) furious

with a falling intonation on each word, as in
the case of any ungrammatical string. Thus (14)
differs from (15) exactly as (1) differs from (2);
our tentative operational criterion for gram-
maticalness supports our intuitive feeling that
(14) is a grammatical sentence and that (15) is
not. We might state the problem of grammar, in
part, as that of explaining and reconstructing
the ability of an English speaker to recognise
(1), (14), etc., as grammatical, while rejecting
(2), (15), etc. But no order of approximation
model can distinguish (14) from (15) (or an
indefinite number of similar pairs). As n

Software Engineering and Artificial Intelligence

Software Engineering and Artificial Intelligence touch at two points:

- 1) How can we use AI as a tool in Software Engineering? (AI4SE)
- 2) What are the specific Software Engineering methods to use when engineering software that includes AI components (SE4AI)?



AI4SE



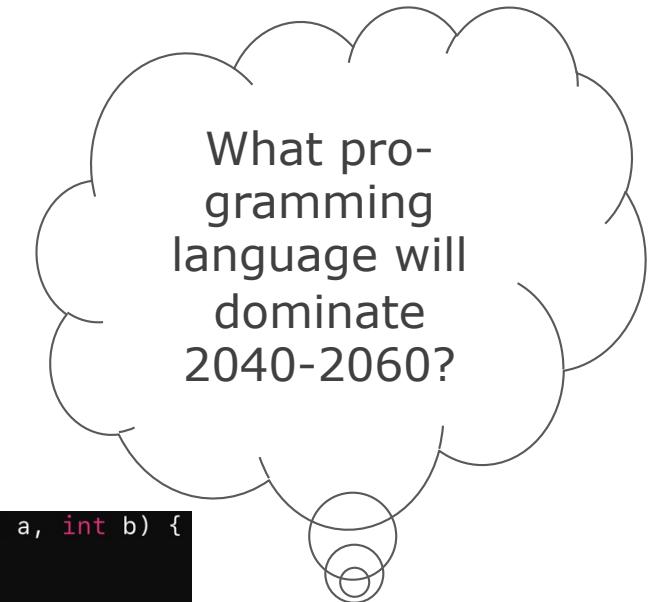
AI4SE - Coding



Programming Languages are Understood by Computers **and** Humans



```
int max(int a, int b) {  
    if (a > b) {  
        return a;  
    } else {  
        return b;  
    }  
}
```



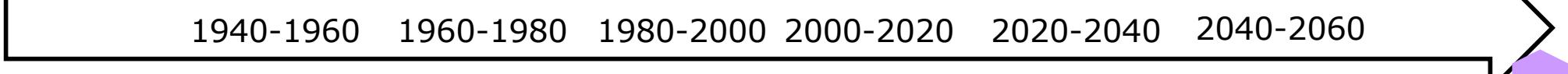
?

```
MOV AX, 5  
MOV BX, 10  
CMP AX, BX  
JG greater  
MOV CX, AX  
JMP end  
greater:  
MOV CX, BX  
end:
```

```
FUNCTION max_of_two(a, b)  
REAL :: a, b  
REAL :: max_of_two  
  
IF (a > b) THEN  
    max_of_two = a  
ELSE  
    max_of_two = b  
END IF  
END FUNCTION max_of_two
```

```
public static int max(int a, int b) {  
    if (a > b) {  
        return a;  
    } else {  
        return b;  
    }  
}
```

```
def max(a, b):  
    return a if a > b else b
```



Note that the languages are getting more and more readable to humans.

Now Humans and Machines Understand Natural Languages → Vibe Coding

Angeheftet



Andrej Karpathy ✅ @karpathy · 24. Jan. 2023

The hottest new programming language is English

910 5.744 35.819 5 Mio. ⌂ ⌁



Russell Kaplan ✅ @russelljkap... · 05.08.24

4/ As a result, software engineering will look radically different in a few years. True coding agents, which do tasks end to end, will complement today's AI copilots. The experience will look something like giving every engineer an army of interns.

7

46

644

98K



At least one month of requirements engineering is **part of the curriculum**



Russell Kaplan ✅ @russelljkap... · 05.08.24

5/ In this new world, every engineer becomes an engineering manager. You will delegate basic tasks to coding agents, and spend more time on the higher level parts of coding: understanding the requirements, architecting systems, and deciding what to build.

At least one month of software architecture is **part of the curriculum**

This is taught in Application Lifecycle Management, which is **part of the curriculum**.



Is This The Future of Programming?



Florian Gallwitz ✅
@FlorianGallwitz

🔗 Translate post

Wenn man ein wenig mit GPT-5 Codex herumgespielt hat, fällt es zunehmend schwer, sich vorzustellen, dass irgendjemand in einem Jahr noch von Hand programmiert.



Florian Gallwitz ✅ @FlorianGallwitz · 2h

Die Annahme, dass Doktoren der Informatik diesem Ding in Punkt Programmier-Skills noch überlegen seien (PhD-level skills), halte ich für überaus gewagt. Ein einstelliger Prozentsatz davon vielleicht.



*"It's hard to imagine that in a year's time, anyone will still be programming by hand. Programming will certainly be done by AI in the future. Only top people who can use AI4SE will survive. **People should spend every free minute learning about AI**".*



Florian Fromm ✅ @florianfromm1 · 1h

Programmieren wird sicher zukünftig von AI erledigt. Aber die eigentliche Herausforderung im Software Engineering war nie das coden, sondern ein reales Problem korrekt zu abstrahieren und zu wissen welche technischen Rahmenbedingungen vorliegen.
Ich denke das wird in Zukunft die zentrale Aufgabe von ITlern sein und natürlich die Ergebnisse der AI zu verifizieren. Das vielleicht auch mit AI ... also im Endeffekt werden Entwickler um einen riesigen Faktor produktiver!



Florian Gallwitz ✅ @FlorianGallwitz · 1h

Sie werden deutlich produktiver, d.h. weniger Entwickler leisten die gleiche Arbeit, die Kosten für Software fallen. Werden wir weniger Entwickler brauchen oder wird die Nachfrage im gleichen Maße steigen? Ich fürchte, die Zahl der Entwickler wird sinken, und nur Top-Leute mit der Fähigkeit, sinnvolle Aufgaben an zahllose KI-Agenten zu verteilen, werden überdauern.



Artalb @ArtAlb03 · 2h

Das ist für Menschen, die ein Informatik-Studium beginnen möchten, sehr demotivierend. Was sagen Sie denn Ihren Studenten?



Florian Gallwitz ✅ @FlorianGallwitz · 2h

Ich bin da sehr offen und rate Ihnen, sich in jeder freien Minute mit der Nutzung dieser Technologie vertraut zu machen.



WEGEN KI

Die Zahl arbeitsloser Programmierer hat sich beinahe verdoppelt

Arbeitslosigkeit steigt laut Daten des AMS in Österreich deutlich an,
hauptverantwortlich seien dafür ChatGPT & Co. KI-Software programmierte



Rohit Mittal

OpenAI has been paying \$500k+ base salaries to Member of Technical Staff since 2023, highest being \$650k.

With stocks, these people would be making like \$1.5M per year.

The fight for AI talent is intense.

EMPLOYER	JOB TITLE	BASE SALARY	LOCATION	SUBMIT DATE	START DATE
OPENAI OPCO LLC	MEMBER OF TECHNICAL STAFF	650,000	SAN FRANCISCO, CA	09/01/2023	10/03/2023
OPENAI OPCO LLC	MEMBER OF TECHNICAL STAFF	530,000	SAN FRANCISCO, CA	04/18/2024	09/01/2024
OPENAI OPCO LLC	MEMBER OF TECHNICAL STAFF	477,000	SAN FRANCISCO, CA	04/11/2024	10/01/2024



...



Deedy

OpenAI paid an average of \$733k/yr across ~6000 employees in stock, nearly THREE times every single other public co.

Subscribe ...



Deedy

BREAKING: Detailed list of all 44 people in Meta's Superintelligence team.

Each of these people are likely getting paid \$10-\$100M/yr.

Source: <https://www.kleinezeitung.at/wirtschaft/2015553/die-zahl-arbeitsloser-programmierer-hat-sich-beinahe-verdoppelt>,

Sep 30th, 2025, <https://x.com/rohitdotmittal/status/1855756154682376347?s=46&t=liJ6-bCACnb-WfoI5TZ2RA>,
<https://x.com/deedydas/status/1946597162068091177?s=46&t=liJ6-bCACnb-WfoI5TZ2RA>,
<https://x.com/deedydas/status/1942612745725304954?s=46&t=liJ6-bCACnb-WfoI5TZ2RA>, 2025



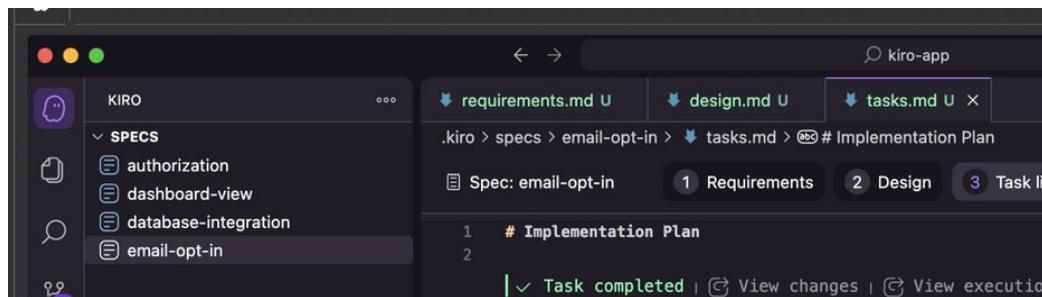
AI4SE – Software Engineering Activities Other Than Coding



Example: Spec Driven Development

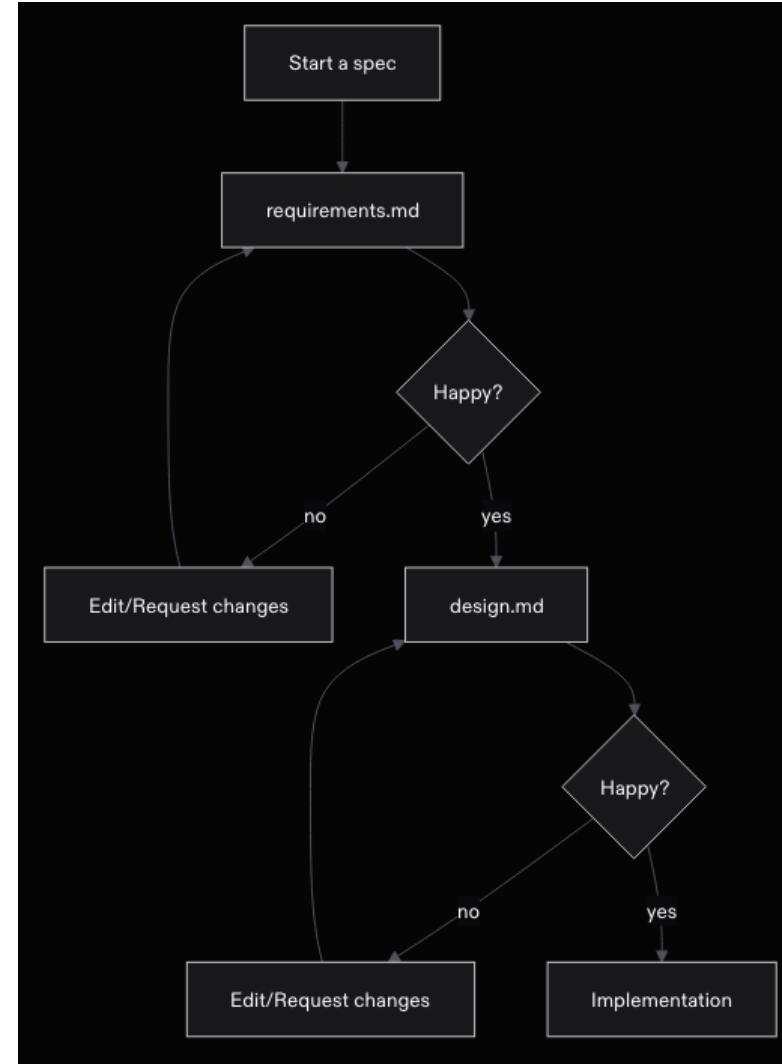
Bring structure to AI coding with spec-driven development

Kiro turns your prompt into clear requirements, structured designs, implementation tasks validated by robust tests, and code generated by advanced agents. It's the flow of AI coding, leveled up with [mature engineering practices](#).



Specs bridge the gap between conceptual product requirements and technical implementation details, ensuring alignment and reducing development iterations. Kiro generates three key files that form the foundation of each specification:

- **requirements.md** - Captures user stories and acceptance criteria in structured EARS notation
- **design.md** - Documents technical architecture, sequence diagrams, and implementation considerations
- **tasks.md** - Provides a detailed implementation plan with discrete, trackable tasks



Source: kiro.dev, 2025



Example: Spec Driven Development / 2

Requirements

The `requirements.md` file is written in the form of user stories with acceptance criteria in EARS notation. The way you wish your PM would give you requirements!

EARS (Easy Approach to Requirements Syntax) notation provides a structured format for writing clear, testable requirements. In a spec's `requirements.md` file, each requirement follows this pattern:

```
WHEN [condition/event]
THE SYSTEM SHALL [expected behavior]
```

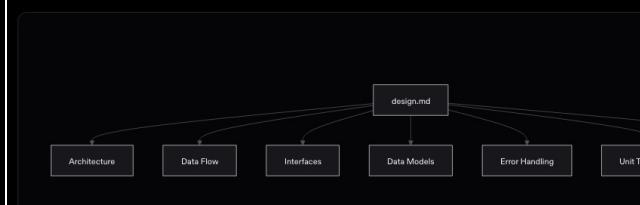
For example:

```
WHEN a user submits a form with invalid data
THE SYSTEM SHALL display validation errors next to the relevant fields
```

Design

The `design.md` file is where you document technical architecture, sequence of implementation considerations. It's a great place to capture the big picture of how the system will work, including the components and their interactions.

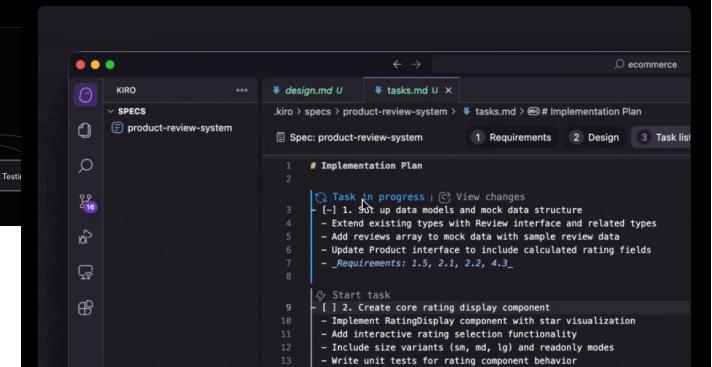
Kiro's specs offer a structured approach to design documentation, making it easier to understand and collaborate on complex systems. The `design.md` file is a great way to capture the big picture of how the system will work, including the components and their interactions.



Implementation plan

The `tasks.md` file is where you provide a detailed implementation plan with discrete, trackable tasks and sub-tasks. Each task is clearly defined, with a clear description, expected outcome, and any necessary resources or dependencies. Kiro's specs offer a structured approach to implementation plans, making it easier to understand and collaborate on complex systems.

Kiro provides a task execution interface for `tasks.md` files that displays real-time status updates. Tasks are updated as in-progress or completed, allowing you to efficiently track implementation progress and maintain an up-to-date view of your development status.



Course Rules



This is a **blended learning** course. It blends self study and class.

If it has 2 ECTS, it is supposed to take you 50 hours (60 minutes each).
10 units (45 min each) are in class. That's 15%.

If it has 3 ECTS, it is supposed to take you 75 hours (60 minutes each).
14 units (45 min each) are in class. That's 14%.

You have to do things at home and you have to do things in class.

At home: You

- study the pdf(s)
- do the home exercises (searchable via the string „home exercise”).
- **don't** upload/mail any results unless told otherwise.

In class: You **participate actively** in

- the group exercises and
- the individual exercises



if asked to: you present your home exercise results.

„25 Arbeitsstunden entsprechen 1 ECTS-Credit, gemäß § 3 Abs. 2 Z 4 FHG idgF. Der Workload umfasst: die Teilnahme an Lehrveranstaltungen, eigenständige Vorbereitung und Nachbereitung, [...]“

„25 working hours correspond to 1 ECTS credit, according to [see above]. The workload includes: attendance at courses, independent preparation and follow-up, [...]“

<https://www.technikum-wien.at/studieninformationen/ects/>

„Ein ECTS-Anrechnungspunkt steht für 25 Echtstunden á 60 Minuten an tatsächlichen Arbeitsaufwand für die Studierende/den Studierenden.“

„One ECTS credit represents 25 hours of actual workload of 60 minutes each for the student.“

https://www.oesterreich.gv.at/de/themen/bildung_und_ausbildung/hochschulen/universitaet/Seite.160120



Each 2-ECTS course has three units (2+4+4 hours, 45 min each).

Each 3-ECTS course has four units (2+4+4+4 hours, 45 min each).

Each unit (= evening) has a **dedicated topic**. Moodle is structured by unit, plus a general part for cross-cutting topics, e.g. cross cutting case studies.

The self study material is **structured** in (typically) one pdf file per unit, which is structured chapters, sub-chapters etc. shown in a table of content. The file has the same name as the Moodle-section (plus version information).

Exercises are **integrated** into the study material. In other words: They are next to the study material they are about. **Advantage for students:** No need to pair study material and relevant exercises manually.



As This is No Lecture, You Need to Be Present

Particularly important parts of the self study material **may** be presented in class by the lecturer (this is why the self study pdf is based on ppt).

Advantage for students: This gives students a good opportunity to ask questions and eliminate knowledge gaps.

This **does not make class a lecture**. The focus in class is on active participation in the exercises, which is why **presence is mandatory**.

However, presence is **not sufficient**. You need to **participate actively** in the exercises. If you fail to do so, you are notified the same evening by the lecturer. **Advantage for students:** No worries on the way home.

If you miss a unit or more, you can ask per email for **compensation** exercises. (Deadline: Last unit of the course with the missed unit)

Advantage for students: Unavoidable events don't lead to bad grades.



Role of Symbols

The **symbols** near the edges of the slides and the cloud symbols are for course management of the lecturer. **The learner can safely ignore them all!**

There are slides with solutions or solution checklists to some of the exercises. They are handed out to the students **after** the exercises. Their filename contains the word „extended“. Note: The dashboard-slide at the beginning of some slidedecks contains buttons for the generation of various versions of the slidedeck (e.g. the extended one). It can be ignored by the student.

Some presented slides (e.g course management slides for the lecturer) are **not** in the self study material. This is intentional. **The learning material is complete** nonetheless. If you believe there is a lack of structure, a gap or if something is unclear, **please contact the lecturer immediately!**



There is **no final exam**. Instead, the parts of an exam referring to each topic are brought forward as **individual exercises** using **pen & paper**.

Individual exercises may be open book (you may use your laptop and its resources) and closed book (you may not). Unless told otherwise, an individual exercise is closed book, as individual exercises often check **whether the self study material has been learned**.

Advantages of individual exercises for the students:

- The student still has the information fresh in his/her memory.
- Also, some questions may be put as open-book questions (by contrast, a final exam would be fully closed-book).

You **don't get intermediate results** of the individual exercises. You wouldn't either if there were a final exam.



To successfully pass the course, you must achieve in total at least **60%** of the achievable points across all individual exercises.

If that is the case, your grade depends on your active participation in class (measured in hours, not in evenings.):
90/80/70/60/<60% of the time or more → 1/2/3/4/5

Rationale: In this course, just like in the industry, you need to have basic grasp of the theory (that is the 60%). But beyond that, it is practice that counts the most (the exercises).



Using Artificial Intelligence

You **are encouraged to use AI** for solving the home exercises. For individual practice, no AI is allowed. For the class group practice, you will be instructed depending on the exercise.

In any case, the end-to-end responsibility for the final result rests entirely with you.

If you use AI, document the prompt and the model.

WEGEN KI

Die Zahl arbeitsloser Programmierer hat sich beinahe verdoppelt

Arbeitslosigkeit steigt laut Daten des AMS in Österreich deutlich an,
hauptverantwortlich seien dafür ChatGPT & Co. KI-Software programmierte



Rohit Mittal



@rohitdotmittal



...

OpenAI has been paying \$500k+ base salaries to Member of Technical Staff since 2023, highest being \$650k.

With stocks, these people would be making like \$1.5M per year.

The fight for AI talent is intense.

EMPLOYER	JOB TITLE	BASE SALARY	LOCATION	SUBMIT DATE	START DATE
OPENAI OPCO LLC	MEMBER OF TECHNICAL STAFF	650,000	SAN FRANCISCO, CA	09/01/2023	10/03/2023
OPENAI OPCO LLC	MEMBER OF TECHNICAL STAFF	530,000	SAN FRANCISCO, CA	04/18/2024	09/01/2024
OPENAI OPCO LLC	MEMBER OF TECHNICAL STAFF	477,000	SAN FRANCISCO, CA	04/11/2024	10/01/2024