Faculty of Electrical Engineering and Informatics
Technical University of Košice

Pokročilé softvérové inžinierstvo

Softvér a jeho vlastnosti, softvérové jazyky

1. prednáška

Jaroslav Porubän

Informácie o predmete

- **Nový predmet** v študijnom programe informatika
- Predmet **zabezpečujú**
  - prof. Ing. Jaroslav Porubän, PhD.
  - doc. Ing. Štefan Korečko, PhD., Ing. Sergej Chodarev, PhD.
  - Ing. Marek Horváth, Ing. Tomáš Kormaník
- **Témy**
  - Softvér a jeho vlastnosti
  - Vývoj softvéru – metódy a nástroje
  - Softvérové architektúry
  - Formálna špecifikácia, matematika vo vývoji softvéru
  - Generatívna AI vo vývoji softvéru
- **Cvičenia**
  - priebežná práca na individuálnom zadaní, max. 40 bodov
- Moodle kľúč: **PSI2025**

Zadanie

- **Projekt - Vývoj softvérového systému**
  - **Akademický informačný systém pre TUKE** s webovým používateľským rozhraním a štruktúrovaným úložiskom údajov
  - Vytvorený len použitím **generatívnej AI**
- Individuálne zadanie
- Na katedrovom GITe vám vytvoríme skupinu s 2 projektmi
  1. Projekt *GenAI*: súbory za každé cvičenie
     - LLM komunikácia
     - zhrnutie skúseností - **negenerované cez LLM, uveďte aj použité nástroje!**
  2. Projekt *Artefakty*: Generované artefakty s použitím generatívnej AI
     - požiadavky
     - zdrojový kód
     - dokumentácia

Projekt – 1. iterácia

- Výstupy **generované cez LLM**
  - Štruktúrovaný dokument s **požiadavkami**
    - Pojmy a ich vzťahy
    - Procesy
  - Odporúčaný **proces vývoja**
  - Zloženie **tímu**, odhad ceny, časový rámec
- Výstupy **negenerované cez LLM**
  - LLM dopyty
  - stručné vyhodnotenie
- **Vytvoríme vám GIT projektovú skupinu s projektmi**
  - Sledujte si GIT
- Individuálna prezentácia **na 2. cvičení**

Štúdia použiteľnosti GenAI v SoftDev

- Účastníci: MY☺
  - študenti
  - učitelia
- Priebežne zbierať skúsenosti s použitím GenAI pri implementácii rozsiahlejšieho informačného systému študentmi - AIS
  - rôzne činnosti vo vývoji
  - rôzne jazyky, platformy, knižnice
  - rôzne modely
- Publikovať získané skúsenosti
- Informovaný súhlas so zberom údajov
- Ďakujeme za spoluprácu!

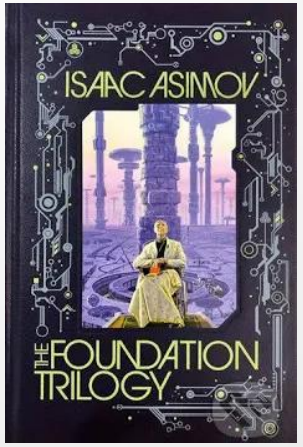
Tools for Software Development

- What is the **most important tool** for software development?
- How to develop software **more efficiently**?

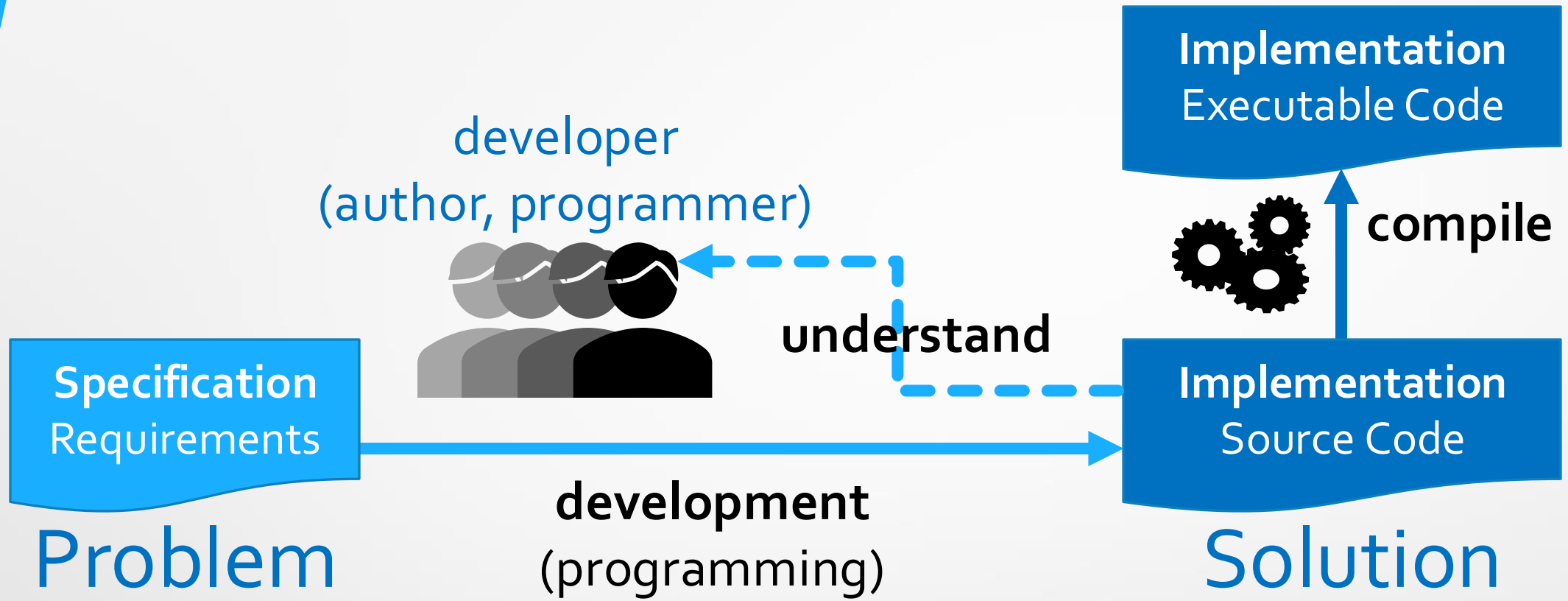
Understanding Software Properties

- What makes the software special?
  - **Costs** mainly for **development**
    - copies and distribution cost almost nothing
  - Source code with **more than a million lines and ONE LETTER MATTERS**
  - **Multi-member** development teams over many years
  - Do not degrade with use
- Future of software
  - Software anywhere – Ambient applications
  - Increasing influence on product quality

1 mil. LOC
29 x 864 p.



Understanding Software Development



How do we create a solution?

How do we write the solution?

How do we understand the solution?

How Far is from the Specification to an Implementation?

- Can we **generate a solution** from the problem definition **automatically**?
  - **Executable specification** – model vs. code discussion
- It is not just the lack of formal specification
- Let's define a problem in a formal language
  - (time, space, object, location)
  - transport for messages (email)
  - transport for objects (teleport)

Details make the difference!

Understanding Software Language

- **Source code is written in a software language**
  - Artificial language for human-computer interaction
- **Language is crucial** tool for software development
- The choice of language **affects the creation and understanding of source code**

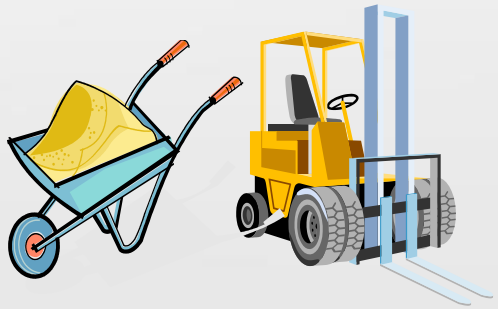
- Language **paradigm** is "style of thinking" in a language
  - Procedural, logic, functional, object-oriented, aspect-oriented,

Are There (Better) Languages or Paradigms?

- There are tenths or hundreds of languages
  - C, C++, C#, Python, Java, JavaScript, TypeScript, Python, …
- Languages
  - **General-purpose** languages
    - multi-paradigm
  - Specialized languages or **domain-specific** languages
    - HTML, CSS, SQL, configuration languages
    - new languages are born every day

General-purpose tools

vs.



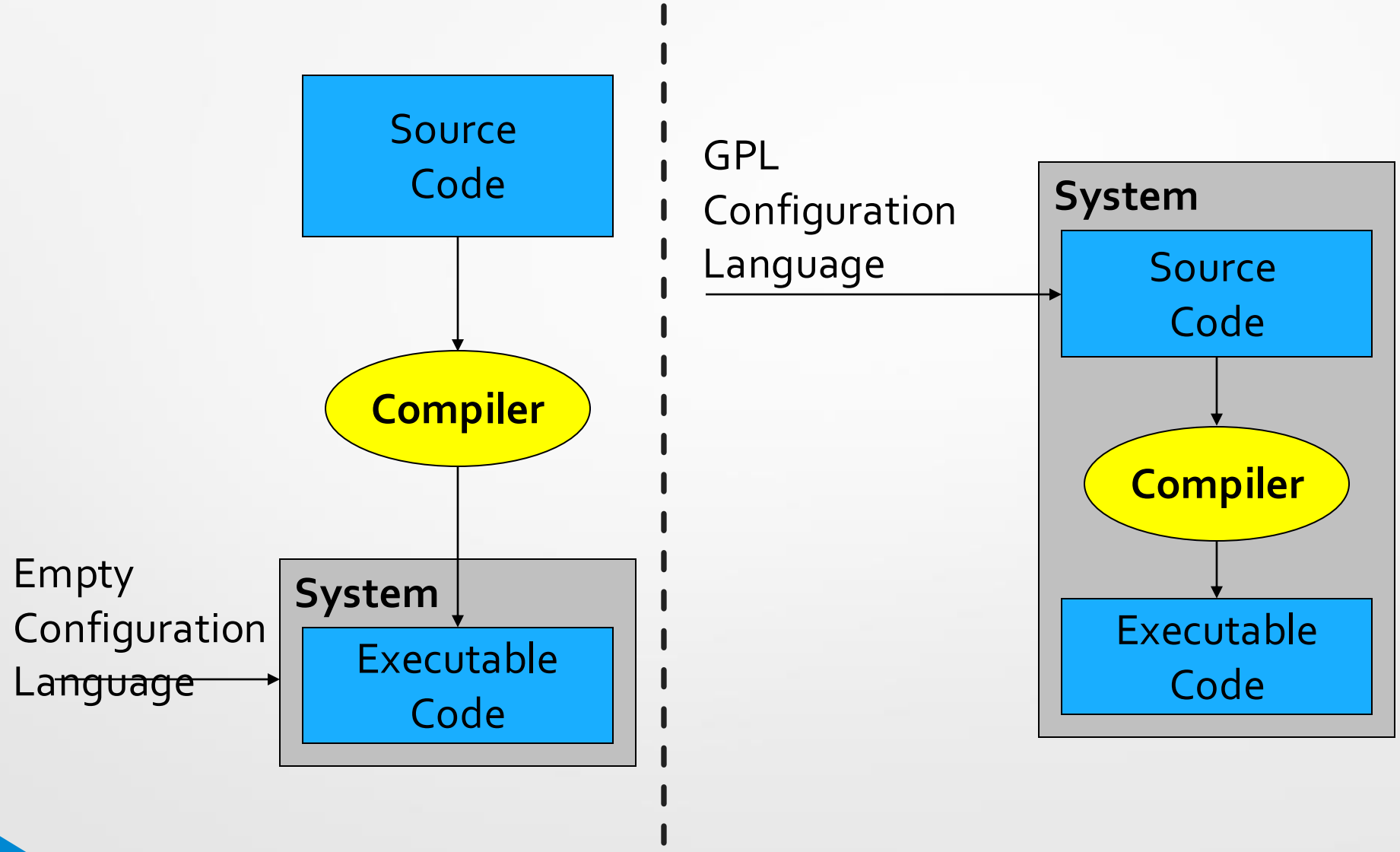
Specialized tools

What is a Software Language?

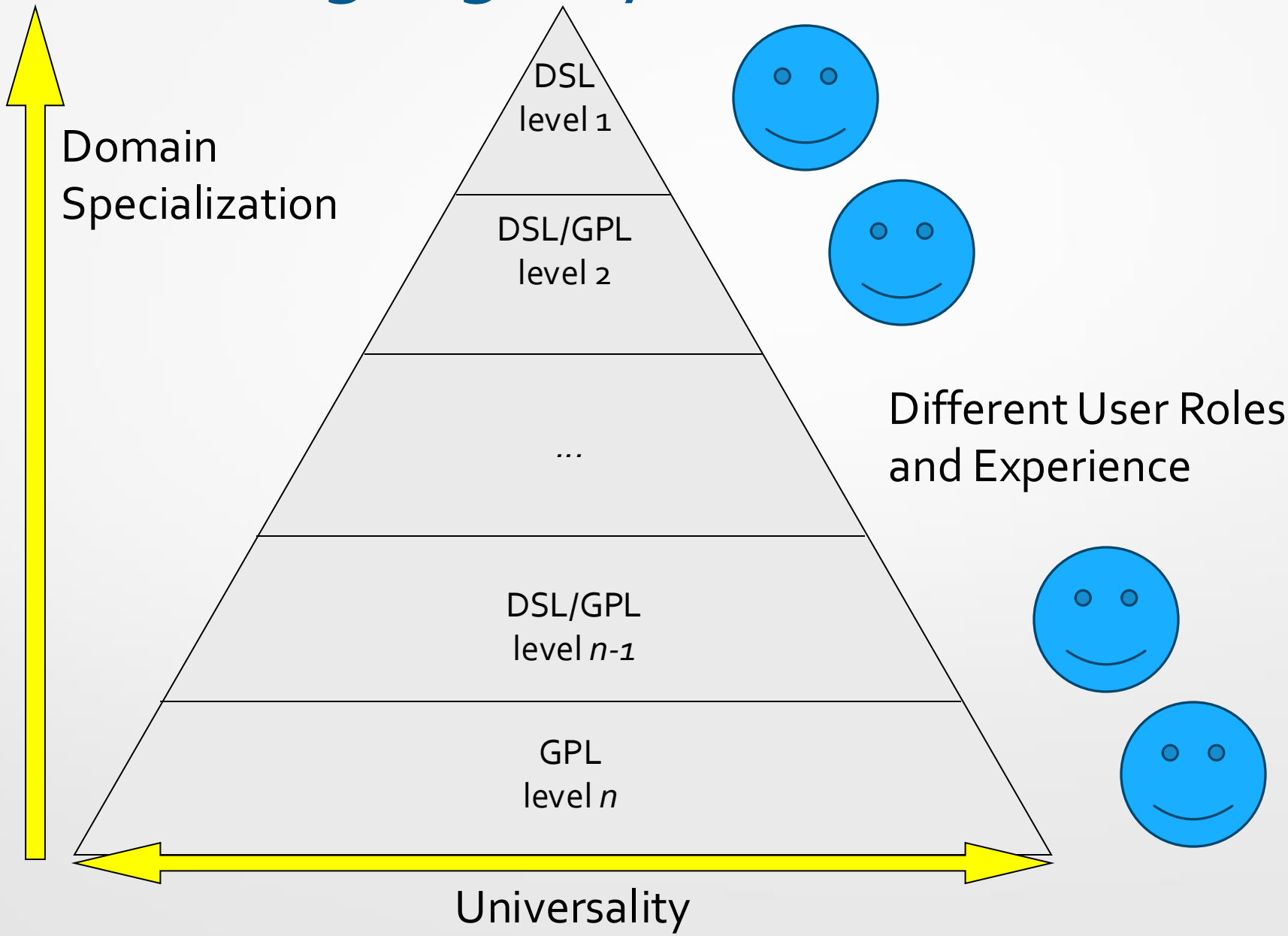
- **Domain-specific language** can have various notation
  - textual
  - graphical
  - form-based

- DSL
  - every UI of an application can be seen as a software language
  - every API can be recognized as embedded language
- Search for a new paradigms
  - Visual programming, Low-code, No-code, Vibe coding

Simple or Complex Language?



Language Pyramid



Výskumný projekt, rok 2019

- **Prirodzený jazyk** je univerzálnym spôsobom komunikácie medzi ľuďmi.
- V kontraste s tým, každý program musí byť reprezentovaný **špecifickou formálnou notáciou**, najčastejšie textovým programovacím jazykom. Mať presnú, jednoznačnú špecifikáciu správania programu je užitočné a nevyhnutné.
- Použitie notácie je jednou z hlavných príčin **kognitívnej náročnosti** programovania.
- Navrhnúť kontextovo závislý prístup k dialógu medzi človekom a počítačom v softvérovom inžinierstve. Ten umožní programátorom rozšíriť ich vývojové prostredia o kontextovo závislého **virtuálneho asistenta** (chatbota), ktorý im pomôže s kognitívne náročnými úlohami **pomocou dialógu v prirodzenom jazyku**.
- **Model dialógu programátora s počítačom**
- **Model softvérového riešenia**
- **Model vedomostí programátora**