Software-Entwicklung mit KI

Daniel Zimmermann, Raphael Juchli

cnlab Herbsttagung 2024: KI und Sicherheit
Gleisarena, Zürich, 4. September 2024

cnlab

Agenda_

1. Motivation
2. Eigene Erfahrungen
3. Tools
4. Gegenwärtiger Einsatzzweck
5. Datenschutz und Urheberrecht
6. Fazit
7. Zukunft

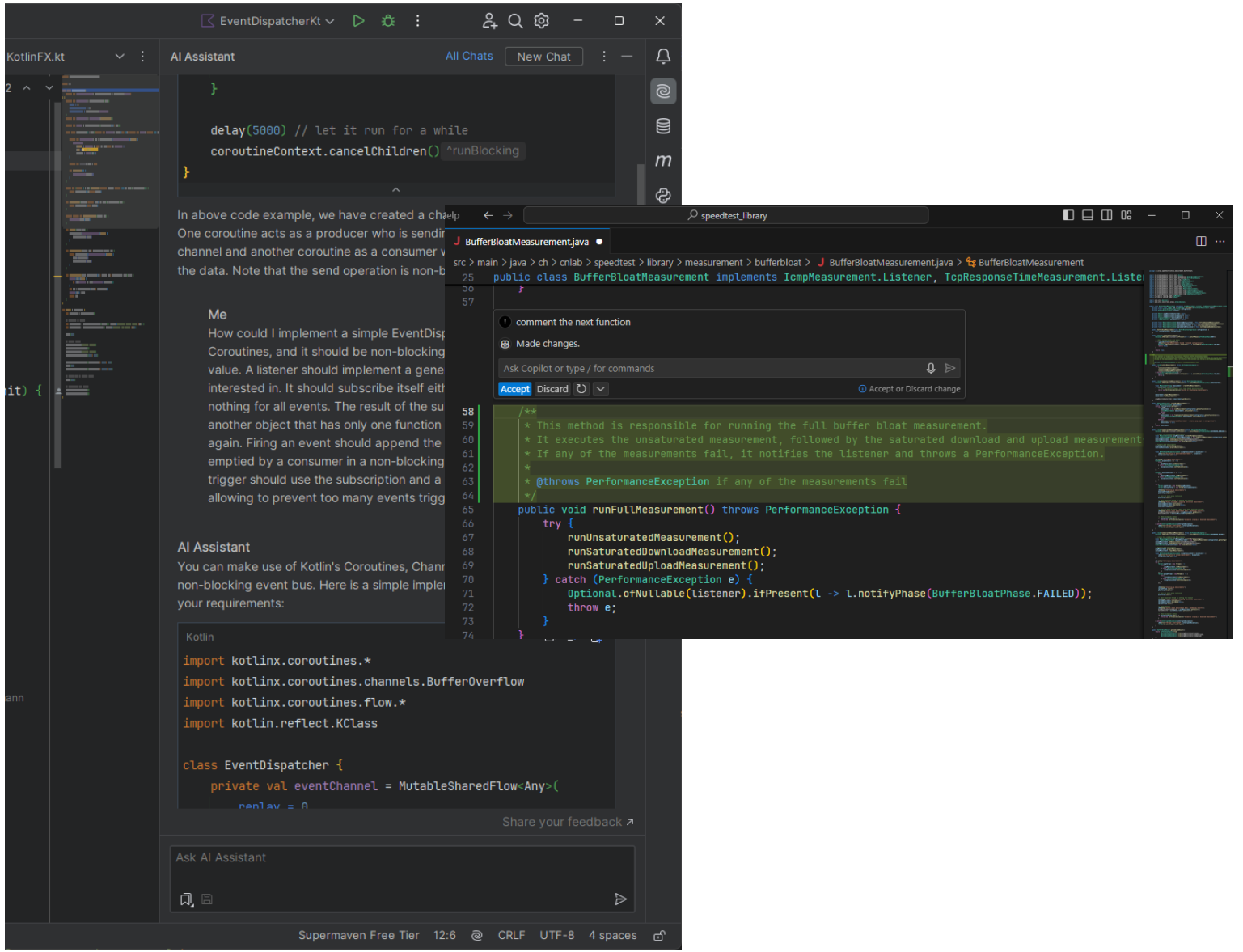
Motivation

Ziel

- Arbeitserleichterung
- Verbesserung der Code-Qualität

Einsatzmöglichkeiten

- Chat (Prompting)
- Code-Vervollständigung



Eigene Erfahrungen

Eigene Erfahrungen

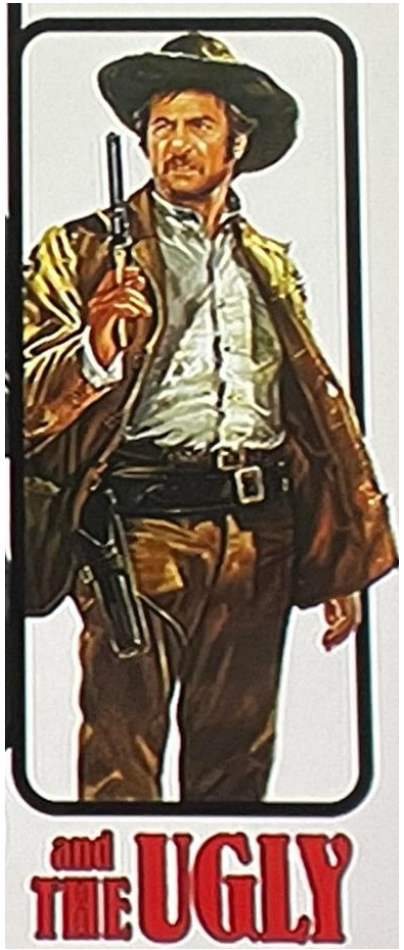
- Chat: Erstellen eines Event-Dispatchers, welche die Arbeitszeit von zwei Stunden auf 15 Minuten reduziert hat.
- Editor: Erstellen einer statischen Datenstruktur, die einer existierenden Struktur stark ähnelt und mit zusätzlichen Datenfelder ergänzt werden muss. Statt Fleissarbeit von einer Stunde, konnte die Aufgabe in fünf Minuten erledigt werden.
- Chat: Ersetzten einer MQTT-Bibliothek in einer bestehenden Anwendung. Es war ein Aufwand von eine Stunde ohne eine aufwendige Studie der Dokumentation.
- Chat: Die Konfiguration von nicht häufig verwendeten Tools ist deutlich einfacher als die Nutzung bestehender Suchmaschinen und dem Studium der dazugehörigen Dokumentation. Beispiel: Konfiguration eines Caddy-Webserver (Kommandozeilen-Tool) für einen Test.

Eigene Erfahrungen

- Chat: Implementieren einer DNS-Messmethode führte dazu, dass der gute Vorschlag nicht genutzt werden konnte, da die benötige Bibliothek veraltet ist. Mit einigen weiteren Anpassungen wurde auf eine neue Bibliothek gewechselt. (Zeitersparnis von etwa ein bis zwei Stunden.).

- Chat: Für das Implementieren einer neuen Funktion wurde eine API verwendete, die es überhaupt nicht gibt. Der Chat behauptete das diese API zur Standartbibliothek gehört.

Eigene Erfahrungen



- Editor: Explizite Ausdrücke können dazu führen, dass die KI einfach den Dienst quittiert oder diese löscht. In unserem Beispiel gab es eine Variable **ASS.** Diese sowie die nachfolgenden Variablen wurden ohne Warnung gelöscht. Später fehlten im Code diese Variablen für diverse Berechnungen. Eine einfache, 10-minütige Aufgabe hat dadurch vier Stunden Fehlersuche verursacht.

- Editor: Generierung von sinnlosen Texten oder Vorschlägen, die nichts mit der Anfrage zu tun haben.

Eigene Erfahrungen

Erlebte Hürden

- Halluzinationen/Erfinden von APIs die es nicht gibt
- Vorschläge von Code mit falscher oder alten API-Versionen
- Integration des Codes kann mitunter zusätzlichen Aufwand nach sich ziehen

- Alle Erfahrungen sind sehr subjektiv
- Am Ende bleiben viele positive Erlebnisse nicht lange im Gedächtnis, weil sie ja «nur» die Arbeit erleichtert haben, negative Erlebnisse prägen sich eher ein
- Der Autor ist von der Inline-KI begeistert, aber Arbeitskollegen sehen ihn eher als störend/invasiv und verwenden lieber den Chat

Tools – Cloud

Integriert in die Entwicklungsumgebung

- Github Copilot
  (Visual Studio Code)
- JetBrains AI Assistant
  (JetBrains Tools)
- Google Gemini
  (Android Studio)
- Supermaven
  (diverse Tools)

Code-Assistenten diverser Cloud-Anbieter

- AWS Code Whisperer
- Google Gemini
- Oracle Code Assist
- IBM Watson
- Anthropic - Claude 3.5 Sonnet

Chat-basierten KI-Lösungen für Entwicklungsarbeit

- ChatGPT
- Bing
- Perplexity

Tools – On Premise / Self Hosted

- Llama 3 Code Instruct
- Mistral – Codestral
- phi3
- gemma

- Vorteile
  - Eigene Daten und Code sind nicht in der «Cloud»
  - Trainieren auf Basis der eigenen Daten und eigenen Codes

- Nachteile
  - Unterhalt der Software, Hardware und Modelle
  - Der Aufwand des Trainierens

Gegenwärtiger Einsatzzweck

Was können Sie?

- **Syntaxfragen ersetzen**
  - Anfragen an Suchmaschinen
  - Anfragen an Stack Overflow
- **Erstellen von Code-Fragmenten**
  - bekannte Algorithmen (Suche, Sortierung)
  - Einfache/Entkoppelte Funktionen

Was können sie nicht?

- **Komplexitätsgrenzen** und Kontext-Grösse
  - Code-Analyse ist ab einer bestimmten Grösse nicht mehr möglich
  - Begrenztes Generieren von komplexen oder grossen Code-Fragmenten
- Garantieren keine **Qualität** der Ausgaben
  - Korrektheit
  - Vollständigkeit
  - Sicherheitsrelevante Probleme

Gegenwärtiger Einsatzzweck

Was können Sie?

- Erstellen von Tests
- Code Review, Code erklären
- Refactoring
- DB-Schemas aus Prompts generieren
- Format-Konversionen
- «Transpiling» - Übersetzen von einer Programmiersprache in eine andere (z.B. Java auf JavaScript)

Was können sie nicht?

- Grössere oder mehrere voneinander abhängige Funktionen erstellen

Datenschutz & Urheberrecht

- Abhängig vom eingesetzten Produkt
  - Genau die Terms-of-Service und Privacy-Policies studieren!
  - Man muss dem Anbieter der KI vertrauen!

- Kann von
  «all meine Daten gehören dann der KI»
  bis zu
  «wir versprechen, deine Daten nicht zu verwenden»
  reichen.

- Mögliche Problemstellungen
  - Die Modelle sind oft auf Basis von OpenSource-Projekten trainiert, die eventuell inkompatible Lizenzen für proprietären Code enthalten.
  - Dein Firmen-interner Code wird exponiert und zum Trainieren der Modelle verwendet.

Datenschutz & Urheberrecht, Teil 2 (Stand September 2024)

- Github Copilot
  - Training mit Eingaben ist einstellbar in den Settings
  - No. For GitHub Copilot for Business users, GitHub does not store Contextual Prompts or Suggestions from use of Copilot Chat beta at rest and hence such data is not used to train models. For GitHub Copilot for Individual users, GitHub only retains this data from use of Copilot Chat beta if the user has opted to enable such collection in their current Copilot settings
  - No. We follow responsible practices in accordance with our Privacy Statement to ensure that neither your Prompts or Suggestions will be shared or used as suggested code for other users of GitHub Copilot

- JetBrains AI Assistant / Grazie
  - Nutzung der Eingaben für Training in EAP Build. In Release Build werden diese Daten nicht erhoben und genutzt
  - Trotzdem sollte man die Einstellungen in der ID überprüfen und die detaillierte Datenkollektion abschalten

- Gemini
  - In der kostenbelasteten Variante werden die Eingaben und Antworten nicht weiter verwendet

Fazit – KI in der Software-Entwicklung bei cnlab

Software-Architektur (k.A.*)
Unterstützt bei der Evaluation und Struktur für neue Anwendungen.

UI- und UX-Design (k.A.*)
Automatische Generierung von komplette Web- und App-Layouts sowie Graphiken für Prototypen.

CI/CD (25-50%*)
Generieren von Konfigurationen und Unterstützung bei der Bereitstellung von Anwendungen.

Software-Entwicklung (~50%*)
Generieren von konkreten Lösungen für verschiedene Probleme.

Software-Entwicklung (50-75%*)
Generieren von Code-Dokumentation.

Qualitätssicherung (~50%*)
Definieren von Unit- und Funktionalen-Tests.

Datenschutz und Urheberrecht
Studiere die Terms-of-Service und Privacy-Policy.

* Potenzial zur Zeitersparnis

Zukunft

- Stufe 1 – KI als Werkzeug für Entwickler
  - An dieser Stelle befinden wir uns gerade

- Stufe 2 – KI als einem Entwickler gleichgestellter Agent
  - Der Code muss in dem Fall wie bei jedem Entwickler kontrolliert werden
  - Vorstellbar, aber der Zeitpunkt ist unklar

- Stufe 3 – KIs entwickeln komplette Systeme und müssen nur noch mit dem Produktmanager interagieren
  - Es ist noch umstritten, ob es je so weit kommt

Vielen Dank für Ihre Aufmerksamkeit_

Für Fragen und eine Demonstration bestehender Tools stehen wir Ihnen beim Apéro gerne zur Verfügung.

cnlab software AG
Obere Bahnhofstrasse 32b
CH-8640 Rapperswil-Jona
Schweiz