



KONFERENCA
evolucija.ai
2026



PROGRAMI

Partner razvojne smeri



REPUBLIKA SLOVENIJA
MINISTRSTVO ZA NOTRANJE ZADEVE
IN JAVNO UPRAVO



REPUBLIKA SLOVENIJA
MINISTRSTVO ZA IZOBRAŽEVANJE, ZNANOST
IN MLADINO



REPUBLIKA SLOVENIJA
MINISTRSTVO ZA GOSPODARSTVO
DELO IN ŠPORT



REPUBLIKA SLOVENIJA
MINISTRSTVO ZA OBRAMBO





Moderator

Andreja Lampe

Kompetenčni center za umetno inteligenco

Združenje za informatiko in telekomunikacije pri GZS

Navdušujejo me prelomne spremembe, ki jih prinašajo umetna inteligenca in digitalne tehnologije. Najbolj me zanima, kako povezovati ljudi, znanje, tehnologije in organizacije ter iz teh povezav ustvarjati nove priložnosti, partnerstva in inovacije.



Kodo pišejo agenti. Kaj pa razvijalci?

prof. dr. Branko Matjaž Jurič

Fakulteta za računalništvo in informatiko

Univerza v Ljubljani

Navdušujejo me tehnologije, ki spreminjajo način, kako gradimo digitalne rešitve – od oblaka in povezovanja sistemov do umetne inteligence in AI agentov. Raziskovalno znanje rad povezujem s konkretnimi rešitvami in prenosom najnovejših tehnologij v prakso.

**„Od marca nismo napisali
niti ene vrstice kode na roko.“**

direktor razvojnega podjetja

Realnost ali fikcija?



UNIVERZA
V LJUBLJANI

FRI

Fakulteta za računalništvo
in informatiko

AI piše kodo hitro.

Človek jo mora prebrati, kar je zamudno.

Ozko grlo razvoja se je preselilo s pisanja kode na specifikacije in na preverjanje.



Napovedi zaostajajo za resničnostjo

75 %

je Gartner aprila 2024 napovedal za leto 2028 — delež inženirjev v podjetjih, ki bo pri kodiranju uporabljal umetno inteligenco

~90 %

je bila dejanska raba že v letu 2025 (DORA, anketa med ≈5.000 strokovnjaki)

Ne gre le za razširjenost — meri se tudi zmogljivost:

Dolžina naloge, ki jo agent pripelje do konca, se podvaja vsakih nekaj mesecev.



Agentov ne uporabljamo na en način, ampak na šest

S0

Dopolnjevanje kode

človek preveri vsak znak

S3

Več agentov vzporedno

človek preveri samo rezultate

S1

Pogovor ob kodi

človek preveri vsako spremembo

S4

Dolgotrajni in asinhroni agenti

človek preveri samo končni predlog

S2

Agent v seji, človek spremlja

človek preveri rezultat in del poti

S5

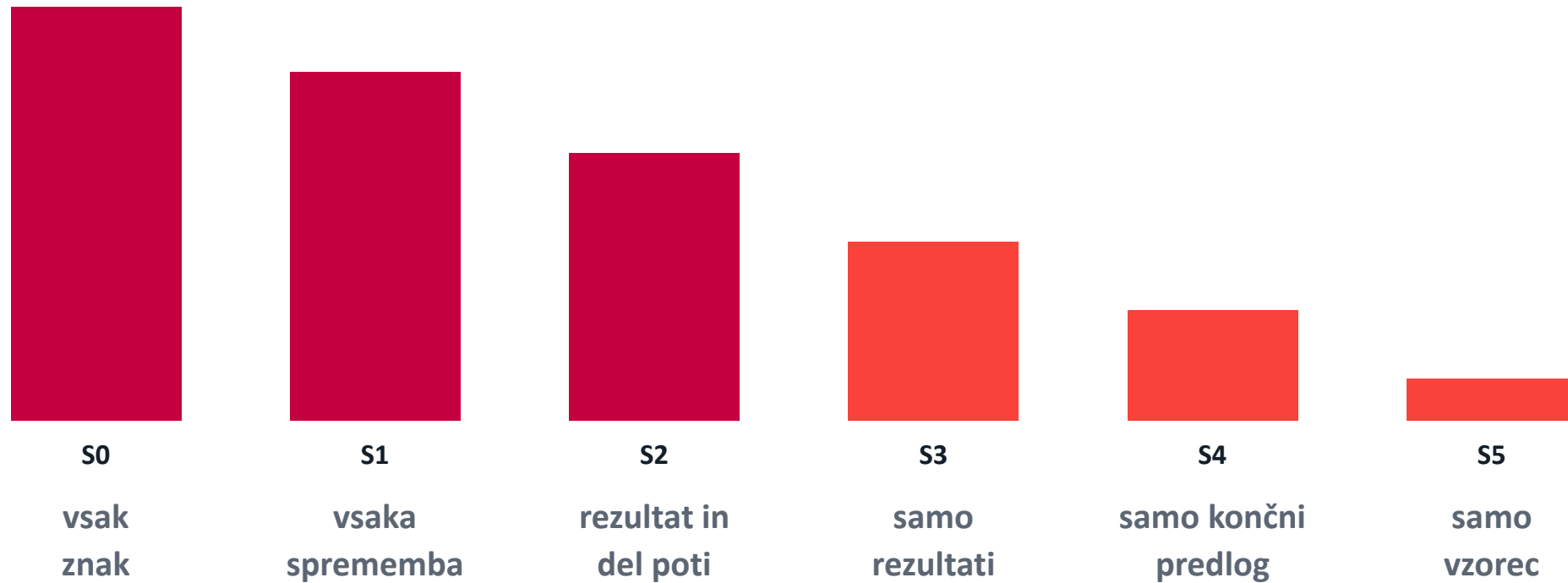
Agenti, vgrajeni v procese

človek preveri vzorec, ne primera



Koda se ni spremenila. Spremenilo se je, koliko kode pregleda človek.

Človeška pozornost na vrstico kode



Za vsako stopnjo avtonomije agenta moramo vzpostaviti ukrepe na treh nivojih

OPIŠEMO • **PREVERIMO** • **RAZVELJAVIMO**

Agentu oddamo le nalogo, ki jo znamo opisati, preveriti in razveljaviti.



UNIVERZA
V LJUBLJANI

FRI

Fakulteta za računalništvo
in informatiko

Štiri plasti specifikacije, ki jih agent potrebuje

Funkcionalna specifikacija

Kaj naj sistem dela — z robnimi primeri in izrecno s tem, česa naj ne dela.

Tehnična specifikacija in omejitve

Katera plast, kateri vzorci, katere knjižnice, kje so meje storitev in pogodbe med njimi.

Nefunkcionalne zahteve

Odzivni čas, prepustnost, razpoložljivost, varnost, sledljivost. Doslej v glavah senior developerjev.

Smernice projekta in okolja

Kako se piše dnevnik, kako teče migracija, katere različice. Danes: pravila v repozitoriju.

Kar je bilo v glavah, mora biti v repozitoriju

Specifikacija ni več dokument za arhiv.

Pravila projekta v repozitoriju (AGENTS.md)

Zapisane arhitekturne odločitve — in razlog zanje

Urejen dostop do internih virov in dokumentacije

Verziona in pregleduje se kot koda

**„Zgolj razdeliti razvijalcem
orodja ne premakne ničesar.“**

Pogoj so urejeni temelji in disciplinirana praksa
specifikacije.

McKinsey, 2026

Praksa je dobila ime: razvoj, voden s specifikacijo.

Kar čutimo in kar merimo

ČUTIMO

80 %+

strokovnjakov poroča o večji produktivnosti

Skoraj 90 % jih umetno inteligenco uporablja pri delu.

MERIMO

–19 %

so bili izkušeni razvijalci počasnejši — in ocenili, da so bili 20 % hitrejši

66 % navaja rešitve, ki so „skoraj pravilne, a ne čisto“.

45 % pravi, da razhroščevanje strojne kode vzame več časa.



Testiranje enot še zdaleč ne zadošča

Testiranje enot

nujni, a povedo najmanj o tem, ali sistem dela

Integracijsko in pogodbeno testiranje

lokalno pravilno, globalno napačno — to ujame le pogodba

Avtomatizirano funkcionalno in sprejemno testiranje

scenariji neposredno iz specifikacije

Testiranje lastnosti in fuzzing

najde tisto, česar se ni domislil ne agent ne avtor testa

Zmogljivostni in varnostni pregledi v cevovodu

nefunkcionalna zahteva ni preverjena, dokler je ne meri stroj

Mutacijsko testiranje - pove, ali bi testi napako sploh opazili



Testi, ki nastanejo iz kode, niso preverjanje.

Nastati morajo iz specifikacije.

In agent, ki jih piše, kode ne sme videti.

Verifikacija (ali smo stvar naredili prav) je avtomatizirana.

Validacija (ali smo naredili pravo stvar) ostaja človeku.



Izvajalno okolje mora biti odporno na napake

PREJ — PRED IZDAJO

Človek prebere spremembo

Pregled kode kot glavna kontrola

Testi kot zadnji korak

Odobritev pred objavo

ZDAJ — PO IZDAJI

Postopno uvajanje (Canary release)

Avtomatski povratek ob poslabšanju metrik

Stikala za funkcionalnosti

Circuit breaker, ponovni poskus, izolacija napake

Opazljivost: napaka vidna v minutah

Če hočete, da agent piše ponoči, mora vaša produkcija preživeti jutro.

Nov projekt in obstoječa aplikacija

NOV PROJEKT

35–40 %

Agent je najmočnejši: ni zgodovine, ni nenapisanih pravil.

Past: agent vedno nekaj izbere. Če arhitekture ni določil razvijalec, jo bo določil agent/model.

OBSTOJEČA APLIKACIJA

~10 %

Manjši učinek: kontekst je огromen in nenapisan.

Tu je agent najbolj podcenjen — za branje, ne za pisanje: razloži modul, poišče klicne poti, napiše manjkajoče teste.



Pri obstoječem sistemu naj agent najprej bere, ne piše



Specifikacije v obstoječem sistemu lahko “izkopljemo” s pomočjo agentov in dopolnimo.

Tehnični dolg znamo izmeriti. Dolga razumevanja ne vidimo.

TEHNIČNI DOLG

Koda je slaba. Vidimo jo, merimo jo, načrtujemo posodobitve/refaktoriranje.

DOLG RAZUMEVANJA

Koda je lahko čista, testirana in delujoča — le nikoli ni nastalo človeško razumevanje, zakaj je taka.

Ne vidimo ga, dokler ne pride prva zahteva po spremembi.

Signal iz podatkov (623 mio. sprememb, 2023–2026):

- podvajanje kode +81 %,
- delež preurejanja obstoječe kode z 21 % na 3,8 %.

Kaj mora biti urejeno danes, ne čez pet let

Utemeljitev, ne le koda

Ob vsaki netrivialni spremembi ostane zapisano zakaj. Arhitekturni zapis je zelo pomemben artefakt.

Sledljivost izvora

Zahteva → specifikacija → sprememba → izdaja. Kdo ali kaj jo je napisalo in kdo jo je odobril.

Specifikacija kot primarni artefakt

Koda je vse bolj rezultat (generirana iz specifikacij). Specifikacija in testi morajo preživeti kodo — zato jih je potrebno vzdrževati.

Enotnost sloga in izbira, kaj ostane v rokah ljudi

Jedro, izračuni in zunanje integracije ostanejo na nižjih stopnjah.



Razvijalec se seli navzgor po abstrakciji

AVTOR TREH DOKUMENTOV

Specifikacija

Testi

Pravila okolja

Koda je vse bolj izpeljanka.

ŠTIRI VLOGE, KI ŠE NISO POIMENOVANE

Pisec specifikacij in omejitev

kar je bilo znanje, mora postati zapis

Inženir preverjanja

določi, kaj šteje kot dokaz delovanja

Kurator konteksta

zastarelo pravilo je danes vir napak

Dirigent agentov

Nadzira delovanje, kriterij dokončnosti



PARADOKS JUNIORJA

-19 %

zaposlenost 22–25-letnikov v najbolj
izpostavljenih poklicih (Stanford, 2026)

Agenti so najboljši ravno pri nalogah,
na katerih se je doslej učil junior.

**Kje bomo čez pet let dobili seniorja, ki
bo napisal specifikacijo in prebral
teste?**



UNIVERZA
V LJUBLJANI

FRI

Fakulteta za računalništvo
in informatiko

Meja ni v deležu strojno napisane kode

1. Ali obstaja zapisana namera — specifikacija, ki ni le prompt v zgodovini pogovora?
2. Ali obstaja neodvisen dokaz — testi, ki ne izvirajo iz istega opisa kot koda?
3. Ali obstaja imenovan odgovorni in pot nazaj?

Trikrat da

profesionalni razvoj — tudi če je 95 % kode napisal agent

Enkrat ne

vibe coding — tudi če je agent napisal samo petino

Novi Excel?

KAJ SE BO ZGODILO

Vsak oddelek si bo znal narediti rešitev. Marketing, kadrovska, logistika. Hitro, poceni, mimo IT.

Delovala bo — zato jo bo nekdo dal v uporabo.

Čez tri leta jo bo uporabljalo štirideset ljudi in nihče ne bo vedel, kako deluje.

KAR ŽE POZNAME

To smo naredili že enkrat. V vsakem večjem podjetju danes teče nekaj kritičnega v Excelovi tabeli.

Brez lastnika, brez testov, brez zgodovine sprememb, brez poti nazaj.

Razlika bo le v tem, da bo novi Excel videti kot prava aplikacija.

Hitro narejeno ni isto kot pripravljeno za produkcijo.

Pet korakov za ponedeljek

- 1 Ugotovite, na kateri stopnji ste dejansko — in ne silite višje, dokler vsi pogoji niso izpolnjeni.
- 2 Zapišite, kar imate v glavah: pravila projekta in omejitve okolja v repozitorij.
- 3 Pri obstoječih sistemih naj agent najprej razlaga, ne spreminja.
- 4 Preverite, kaj vaši testi dokazujejo. Integracijski in funkcionalni testi pred agenti.
- 5 Poskrbite za povratek. Če izdaje na produkciji ne znate razveljaviti v petih minutah, agentov še ne potrebujete.



To menjavo smo že enkrat naredili

ZBIRNIK → VIŠJI JEZIKI

Dali smo učinkovitost.

Programi so postali večji in počasnejši, poraba procesorja in pomnilnika je zrasla.

Obdržali smo razumevanje — višji jezik je bilo lažje brati, ne težje.

VIŠJI JEZIKI → AGENTI

Dajemo razumevanje.

Koda je lahko hitrejša ali počasnejša — to ni več glavno vprašanje.

In to je edina stvar, ki je v tej stroki še nikoli nismo dali.

Kdo bo leta 2036 razumel to aplikacijo? Tisti, ki bo imel specifikacijo in teste.

Kako bo videti oglas za službo leta 2030+

PROSTO DELOVNO MESTO · SEPTEMBER 2030

Inženir specifikacije in preverjanja (m/ž)

razvojna ekipa štirih · polni delovni čas · sistem, ki ga uporablja 400.000 ljudi

NALOGE

Opisujete, kaj mora sistem početi — funkcionalno, tehnično in nefunkcionalno.

Določate, kaj šteje kot dokaz, da to res počne.

Presojate predloge sprememb in odločate, kaj gre v produkcijo.

Skrbite, da je znanje o sistemu zapisano in točno.

PRIČAKUJEMO

Odlično branje tuje kode in hitro presojo. Pisali je ne boste veliko.

Razumevanje domene — *(npr. bančništvo)*.

Izkušnje s pogodbenimi testi, testiranjem lastnosti in verifikacijo.

Sposobnost reči »ne« in to utemeljiti.

To delovno mesto danes ne obstaja. Kompetence zanj pa se gradijo več let — in začeti je treba letos.

Čez pet let vprašanje
ne bo „si to napisal ti?“

Vprašanje bo:
„Kako veš, da to dela?“

prof. dr. Matjaž B. Jurič

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko



Umetna inteligenca kot partner pri razvoju programske opreme

izr. prof. dr. Luka Pavlič

Fakulteta za elektrotehniko, računalništvo in informatiko

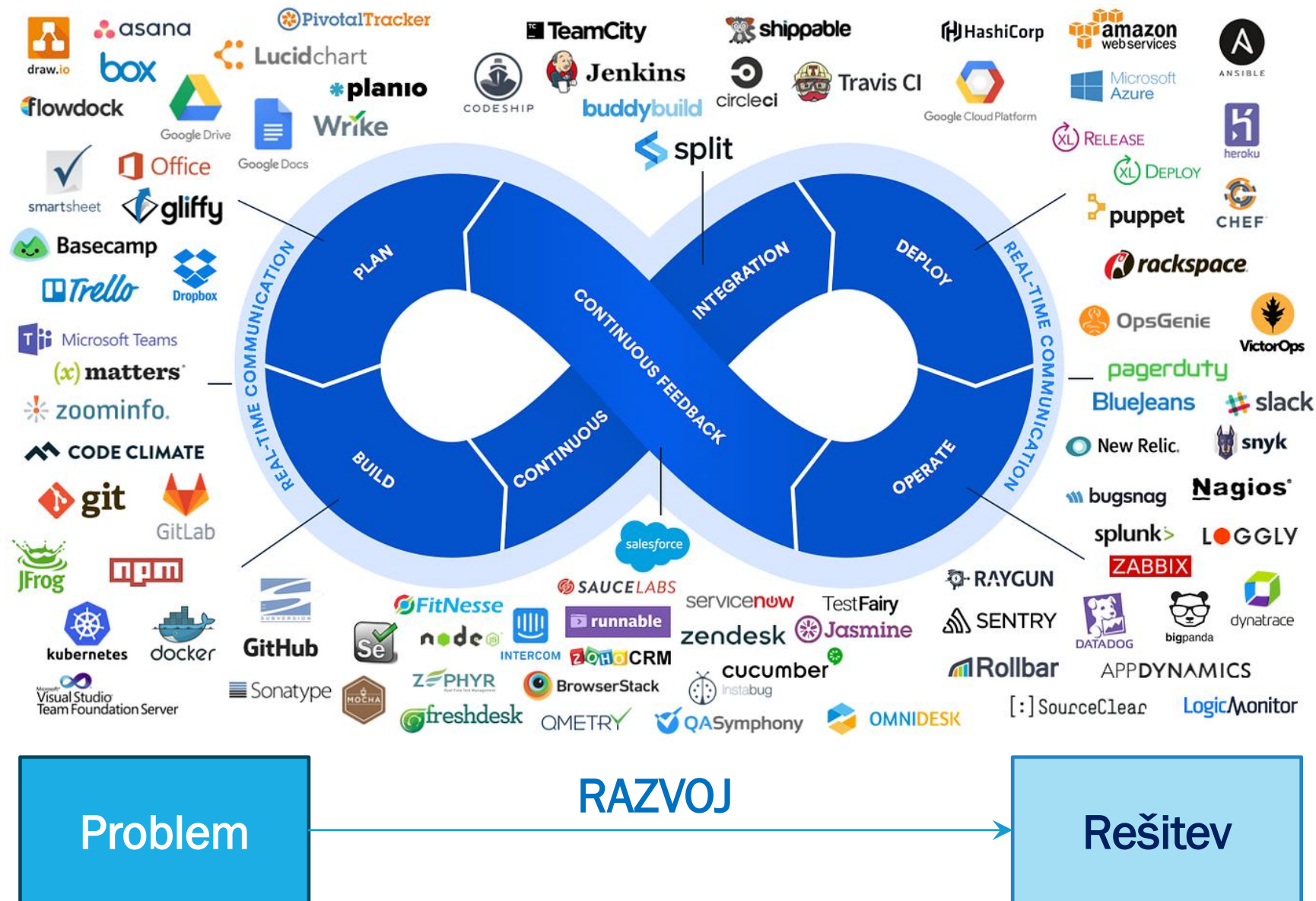
Univerza v Mariboru

Raziskujem, kako sodobne razvojne prakse, arhitekture in umetna inteligenca spreminjajo razvoj programske opreme. Posebej me zanima, kako z boljšimi procesi, orodji in obvladovanjem tehničnega dolga graditi kakovostnejše in dolgoročno vzdržne rešitve.

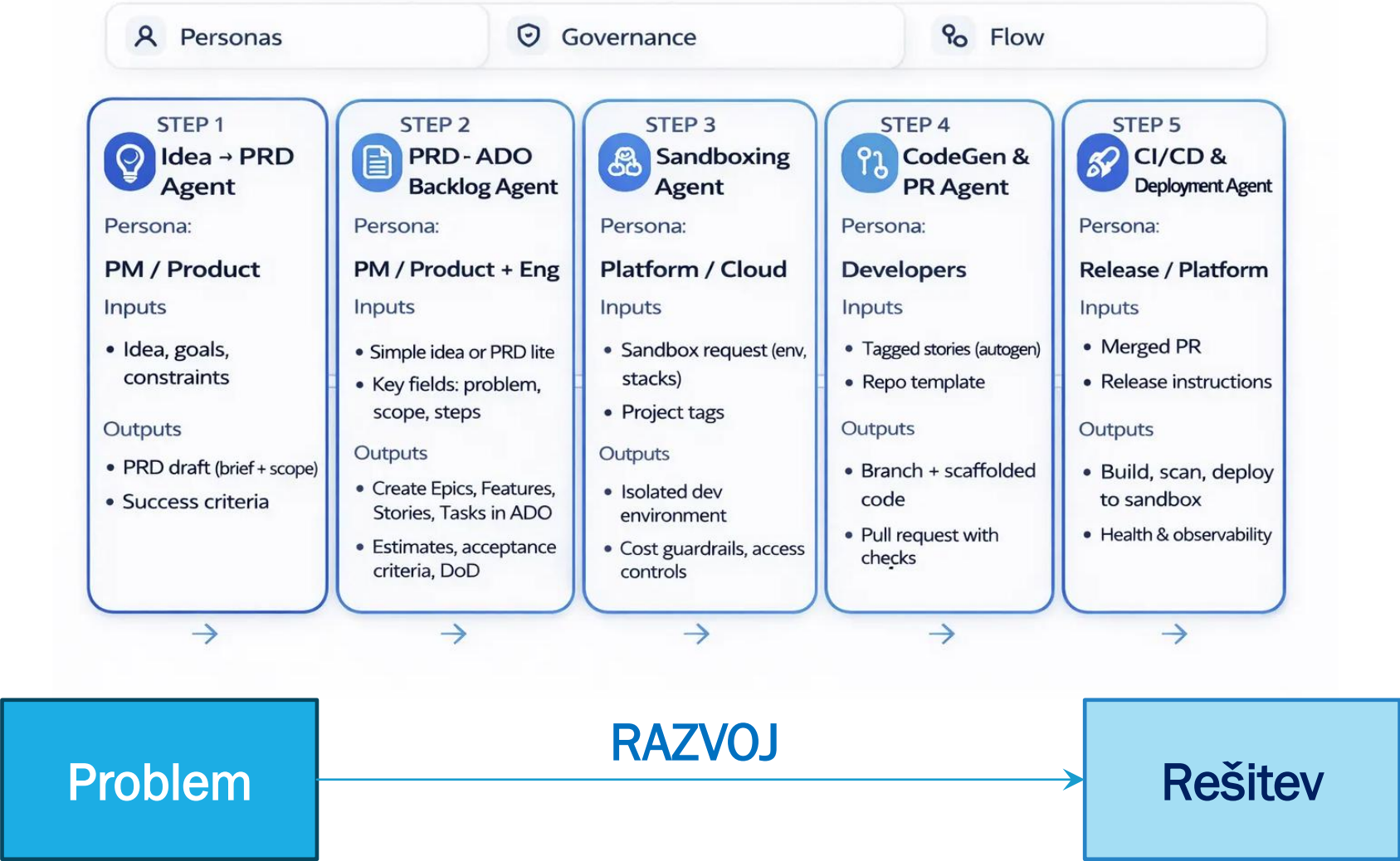


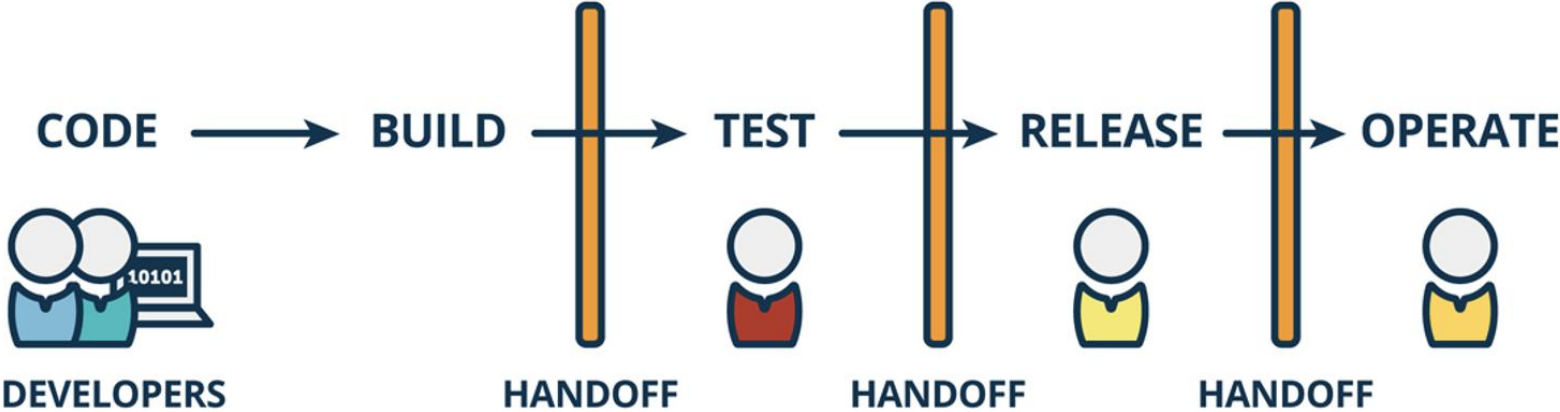
A-O-P ?



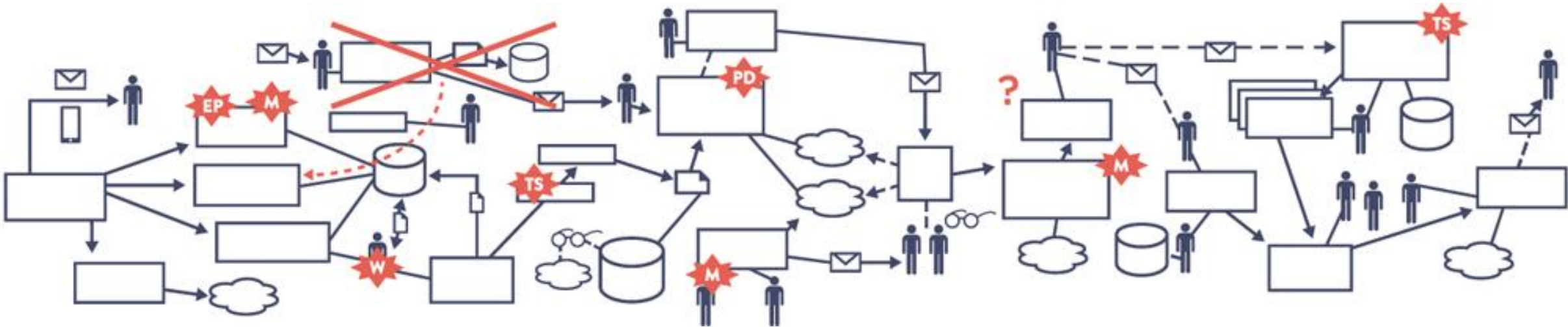


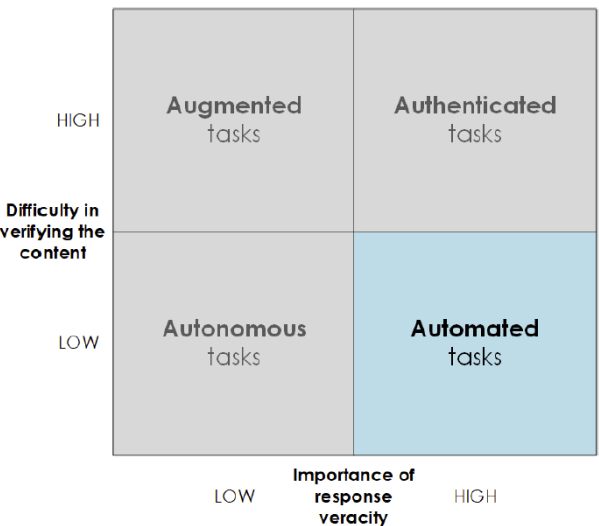
Autonomous Agentic Development Solution Workflow





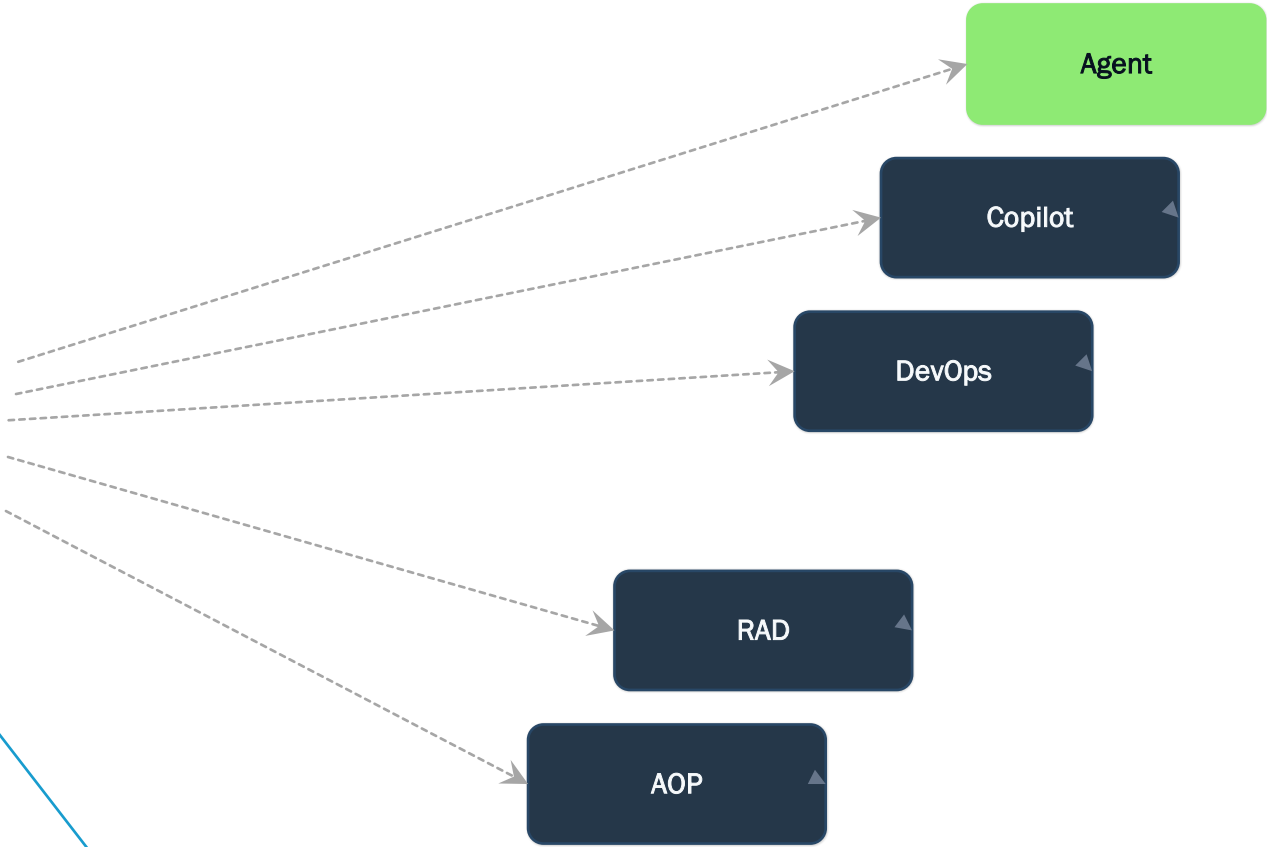
Pressman: 40 (a/d) – 20 (i) – 40 (t/d)

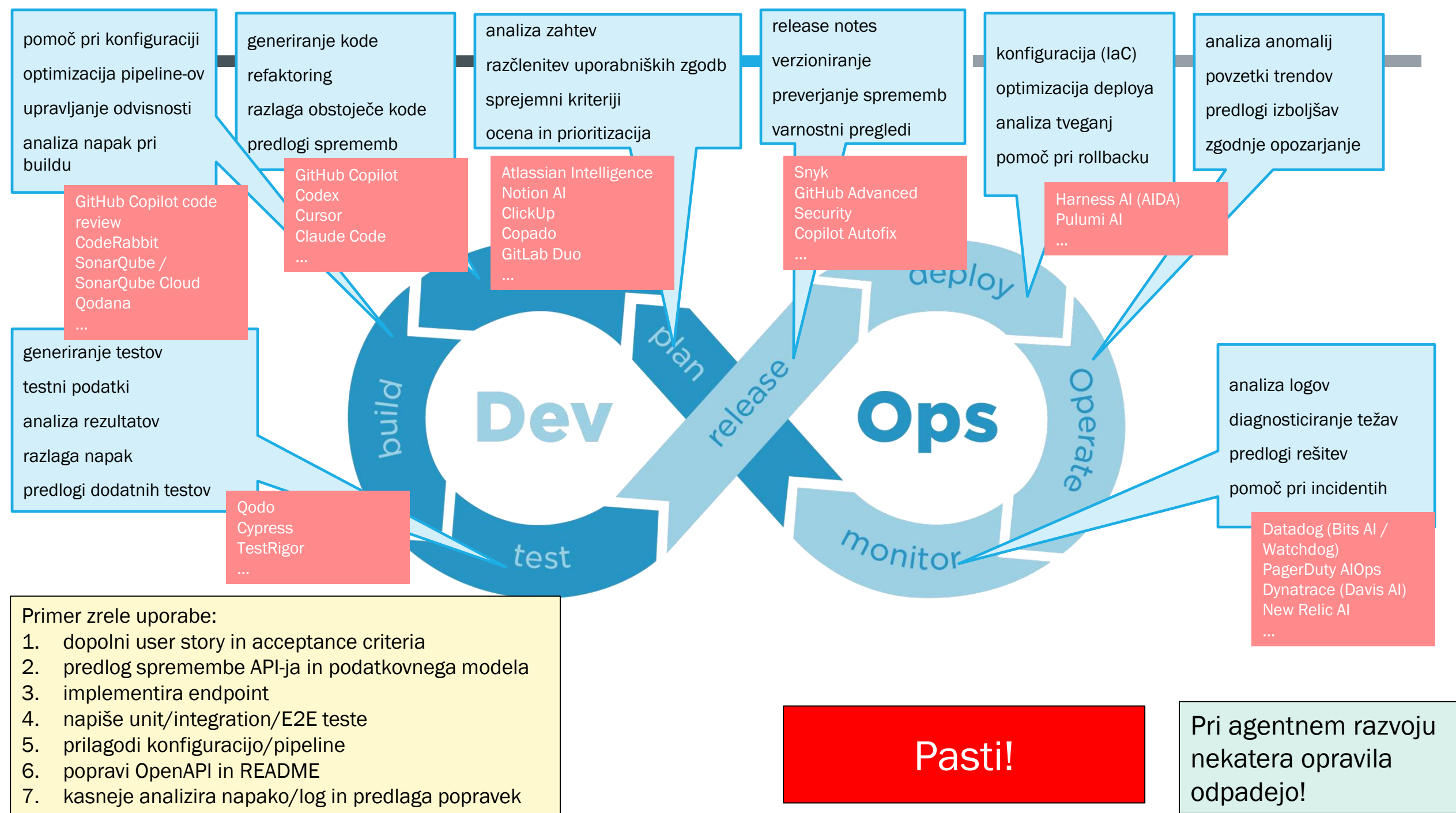


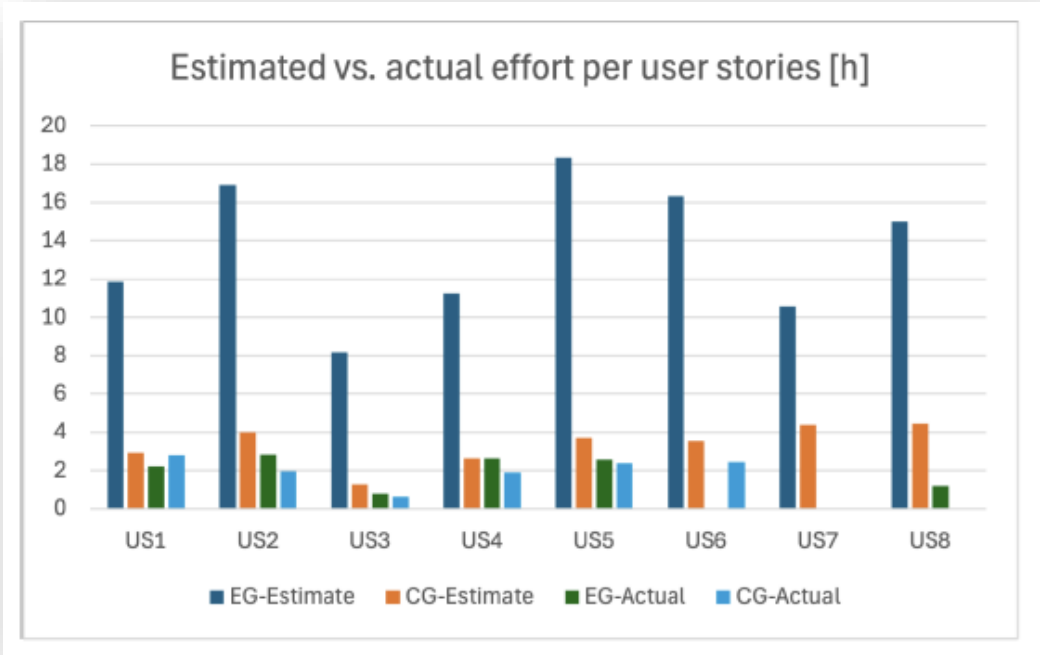
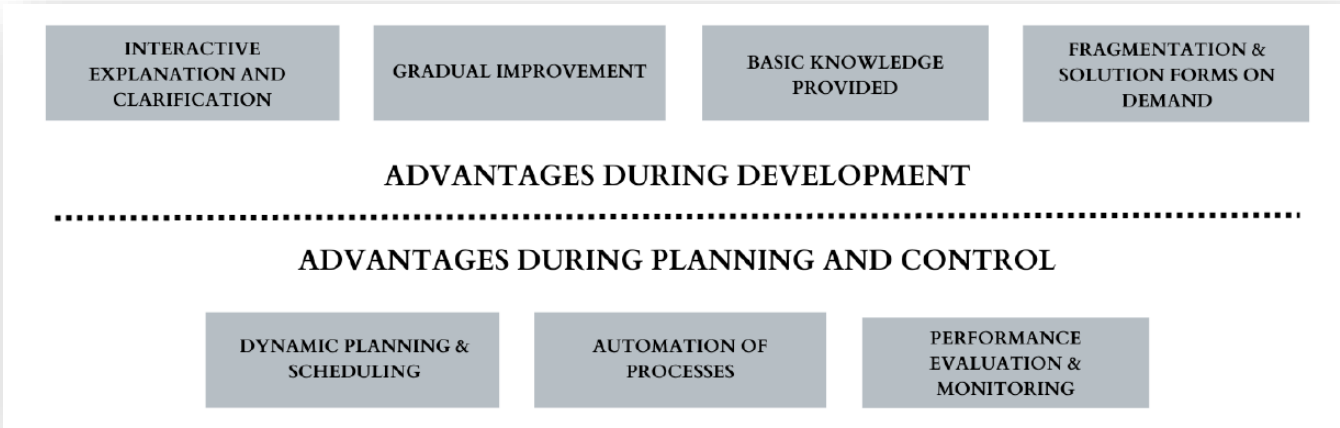
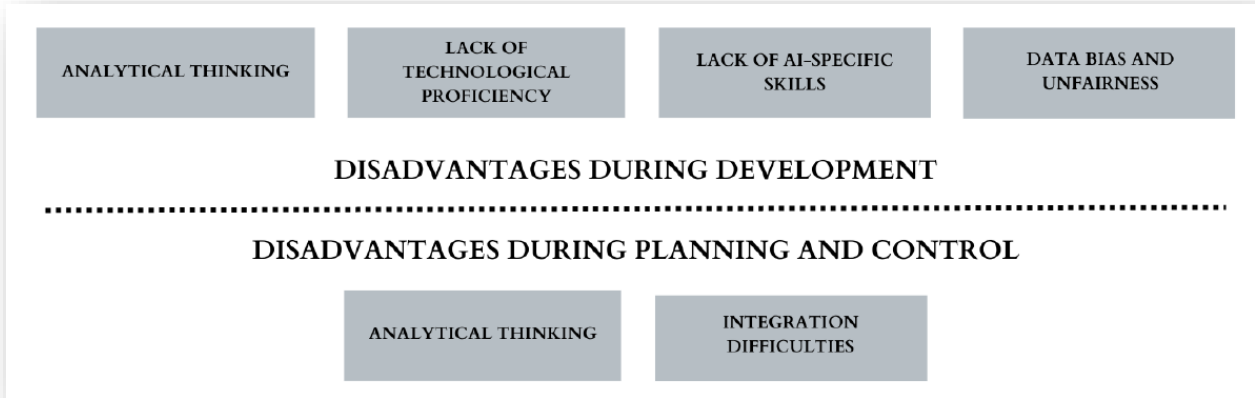


Ne žrtev **AI**!
AI je nova **priložnost**!

Isti **cilj**
Različni **pristopi**







NOVO OZKO GRLO: KAJ SPLOH ŽELIMO?

USERSTORY / AC

Garbage in



Garbage out

USERSTORY
AC, DoR
AGENTS.md / CLAUDE.md
ADR
OpenAPI
CodingConventions
TestStrategy
Skills
...

Garbage in



Garbage out
at scale

AI zelo učinkovito pospeši izvedbo napačno razumljene zahteve.

NAŠE IZKUŠNJE (IN PROPOROČILA)

- Teža se seli na področja kot so
 - Zbiranje zahtev
 - Zagotavljanje kakovosti – žal vidimo tendenco po opuščanju; testiranje (le) na nivoju kode?
 - Kibernetska varnost
- Zakaj bi AI zaupali (...bolj kot ljudem)?
 - cca 30% popolnoma zaupa AI
 - cca 45% ne zaupa AI
 - cca 65% - “almost” right / “good enough”
- Čas dostave rešitve se poveča
 - Odvisno od izkušenj!
 - “Juniorji” – ne
 - “Seniorji” – da
- Kakovost – tipično ni sprememb
 - “Juniorji” - slabše

Morda to pojasni
stanje v industriji?

AI kot **ojačevalec**.

Provokacija: **kdo
bodo Seniorji leta
2031?**

JUNIOR 2026

razume problem → uporablja AI → bere kodo → testira →
debugira → razloži odločitev



Univerza v Mariboru

Fakulteta za elektrotehniko,
računalništvo in informatiko

Hvala za pozornost.

Luka Pavlič

UM FERi

Inštitut za informatiko



KONFERENCA
evolucija.ai
2026



REPUBLIKA SLOVENIJA
MINISTRSTVO ZA NOTRANJE ZADEVE
IN JAVNO UPRAVO



REPUBLIKA SLOVENIJA
MINISTRSTVO ZA IZOBRAŽEVANJE, ZNANOST
IN MLADINO



REPUBLIKA SLOVENIJA
MINISTRSTVO ZA GOSPODARSTVO
DELO IN ŠPORT



REPUBLIKA SLOVENIJA
MINISTRSTVO ZA OBRAMBO





Agentno vodena infrastruktura za skalabilno uporabo modelov umetne inteligence

doc.dr. Mladen Borovič

Fakulteta za elektrotehniko, računalništvo in informatiko

Univerza v Mariboru

Ukvarjam se s povezovanjem umetne inteligence in visokozmogljivega računalništva ter raziskujem, kako zmogljiva računska okolja uporabiti za učinkovitejši razvoj in uporabo sodobnih AI rešitev.



Umetna inteligenca in HPC

- Tesna povezanost visokozmogljivega računalništva (HPC) in umetne inteligence
- Projekti kot so EuroCC, EuroCC2, DIGI-SI, Fortissimo Plus, EU.FFICIENT in SLAIF



- Izkušnje na podlagi sodelovanja s podjetji v obliki svetovanja za HPC
- **Ugotovitev:** infrastruktura HPC je na voljo, vendar se zalomi pri prehodu v produkcijo. Uspeh temelji na komunikaciji med HPC strokovnjaki in podjetji.
- **Ideja:** kako z avtomatizacijo izboljšati proces prehoda v produkcijsko HPC okolje?

Ročna alokacija računskih virov



- V sklopu projekta SLAIF smo izvedli prototipno preoblikovanje gruče HPC Maister
 - Prej: **raziskovalna infrastruktura HPC** (vrsta poslov SLURM)
 - Zdaj: **hibridna infrastruktura HPC** (vrsta poslov SLURM + oblačna infrastruktura OpenStack)
- Uporabniki lahko na oblačni infrastrukturi pridobijo računske vire
 - Vnaprej pripravljene slike z OS, konfiguracije računskih in podatkovnih virov, omrežne nastavitve
 - Konfiguracije po meri
- Sodelujoči z uspešnim prehodom v produkcijsko okolje HPC
 - XLAB, Valira AI, Kalmia, Inova IT, Igea, Genis, UNI Maribor Grand Prix Engineering
- **Ročna alokacija virov je robustna in deluje, vendar ima tudi svoje pomanjkljivosti**



Ozko grlo pri alokaciji računskih virov

- Vprašanja iz naših izkušenj
 - "Vemo kaj bi radi dosegli, ne vemo pa točno kakšno infrastrukturo potrebujemo."
 - "Kakšno infrastrukturo naj izberemo, da bo skaliranje v prihodnosti čim bolj enostavno?"
 - "Vsebniki ali virtualni stroji?"
 - "Kateri veliki jezikovni model naj uporabimo?"
 - "Ali bo grafična procesna enota imela dovolj VRAM-a za poganjanje naše storitve?"
 - "Kombinacije kvantizacije, KV-cache in velikih jezikovnih modelov – kaj je optimalno za nas?"
- Alokacija premalo ali preveč računskih virov, dodatno svetovanje je lahko zamudno
- Proces prehoda v produkcijo se lahko zavleče tudi za nekaj mesecev
- Kako lahko izboljšamo proces svetovanja tako za uporabnike, kot za strokovnjake?

Pogovorni sistem kot vmesnik

- **Ideja:** pogovorni sistem, ki uporabnika vodi skozi proces in mu priporoča različne konfiguracije infrastrukture HPC
- V ozadju sistem RAG
 - Shranjene konfiguracije
 - Domensko znanje HPC
 - Informacije o dostopih do HPC
 - Dokumentacija SLING
 - Primeri iz delavnic
- Agenti v ozadju izvajajo proces svetovanja

(KONCEPT)

Potrebujem okolje za zagon odprtokodnega LLM modela za povzemanje internih dokumentov v slovenščini. Pričakujem okoli 10 sočasnih zahtevkov. Kaj predlagaš?

Glede na tvoje zahteve in trenutno razpoložljivost na gruči HPC Maister, predlagam naslednjo optimalno postavitev:

Priporočen model: GaMS-3-12B-Instruct

Potrebna infrastruktura:

- 1x Navidezni stroj (VM)
- 16 vCPU jeder
- 64 GB RAM
- 1x NVIDIA V100 (32GB) GPU

Želiš, da pripravim to konfiguracijo in te preusmerim na zaslon za končno potrditev?

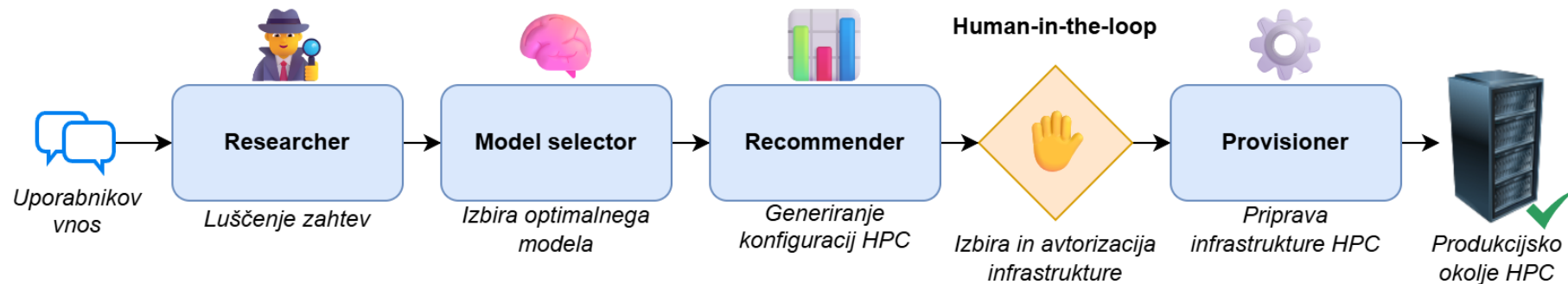


Univerza v Mariboru

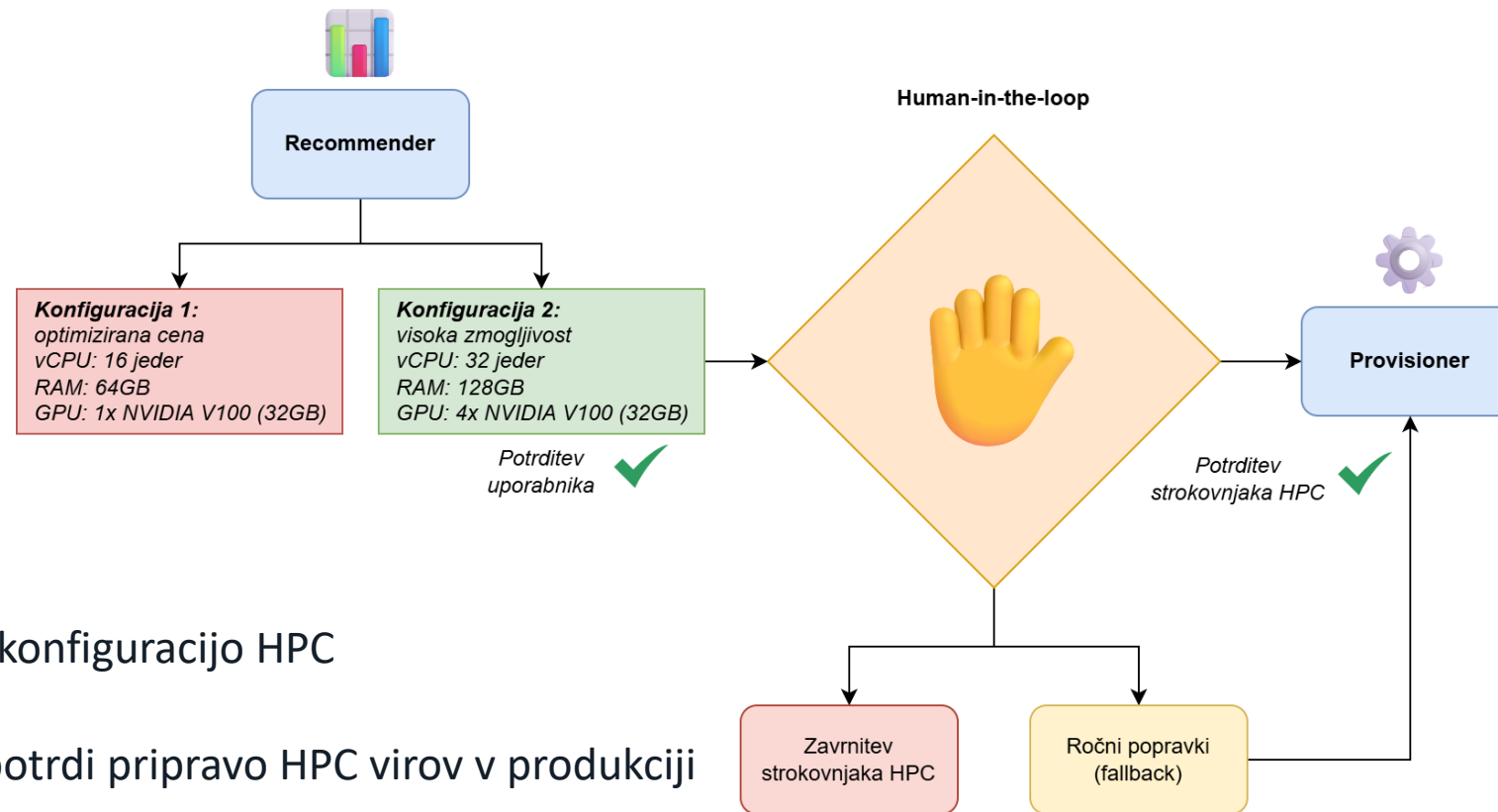
Fakulteta za elektrotehniko,
računalništvo in informatiko

Uporaba agentov v ozadju

- V zalednem delu se izvajajo agenti, ki so povezani v delovni tok
 - Razumevanje uporabnikovih zahtev (Researcher)
 - Izbira optimalnega modela (Model selector)
 - Generiranje konfiguracij HPC (Recommender)
 - Priprava infrastrukture (Provisioner)



Human-in-the-loop



- Uporabnik potrdi konfiguracijo HPC
- HPC strokovnjak potrdi pripravo HPC virov v produkciji
- V primeru težav (tudi halucinacij) je na voljo ročna alokacija računskih virov

Uporabniška izkušnja

Ročna alokacija računskih virov

- Več korakov
- Ročno brskanje po dokumentacijah
- Potreba po svetovanju strokovnjaka HPC
- Razvijalci se ukvarjajo s preходом na produkcijsko infrastrukturo
- Dolgotrajno klikanje po nadzornih ploščah v oblaku in pisanje konfiguracijskih skript

Agentno vodena infrastruktura

- Dva koraka
- Pogovor s pogovornim sistemom
- Pregled priporočenih konfiguracij in potrjevanje
- API in delujoč model v nekaj minutah

- Porabljen čas: dnevi, tedni, meseci

- Porabljen čas: minute, ure

- Prehod iz infrastrukture, ki jo je potrebno obvladati, na infrastrukturo, ki "posluša"
- Koncept agentno vodene infrastrukture HPC bomo začeli testirati predvidoma v prvi polovici 2027



Univerza v Mariboru

Fakulteta za elektrotehniko,
računalništvo in informatiko

Hvala za pozornost.

doc.dr. Mladen Borovič

UM FERi



KONFERENCA
evolucija.ai
2026



REPUBLIKA SLOVENIJA
MINISTRSTVO ZA NOTRANJE ZADEVE
IN JAVNO UPRAVO



REPUBLIKA SLOVENIJA
MINISTRSTVO ZA IZOBRAŽEVANJE, ZNANOST
IN MLADINO



REPUBLIKA SLOVENIJA
MINISTRSTVO ZA GOSPODARSTVO
DELO IN ŠPORT



REPUBLIKA SLOVENIJA
MINISTRSTVO ZA OBRAMBO





Uporaba AI agentov kot pomočnikov pri razvoju programske opreme

Luka Podgoršek

Data Platform Partner Technical Specialist

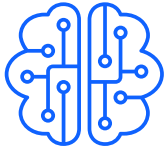
IBM

Povezujem podatkovne platforme, umetno inteligenco in sodobne arhitekture v rešitve za konkretne poslovne izzive. Posebej me zanima, kako AI in novi razvojni pristopi spreminjajo način, kako načrtujemo in gradimo programske rešitve.



Tipi umetne inteligence

AI, ki napoveduje



Machine learning

1

AI, ki ustvarja



Generative AI

2

AI, ki klepeta



AI assistants

3

AI, ki dela



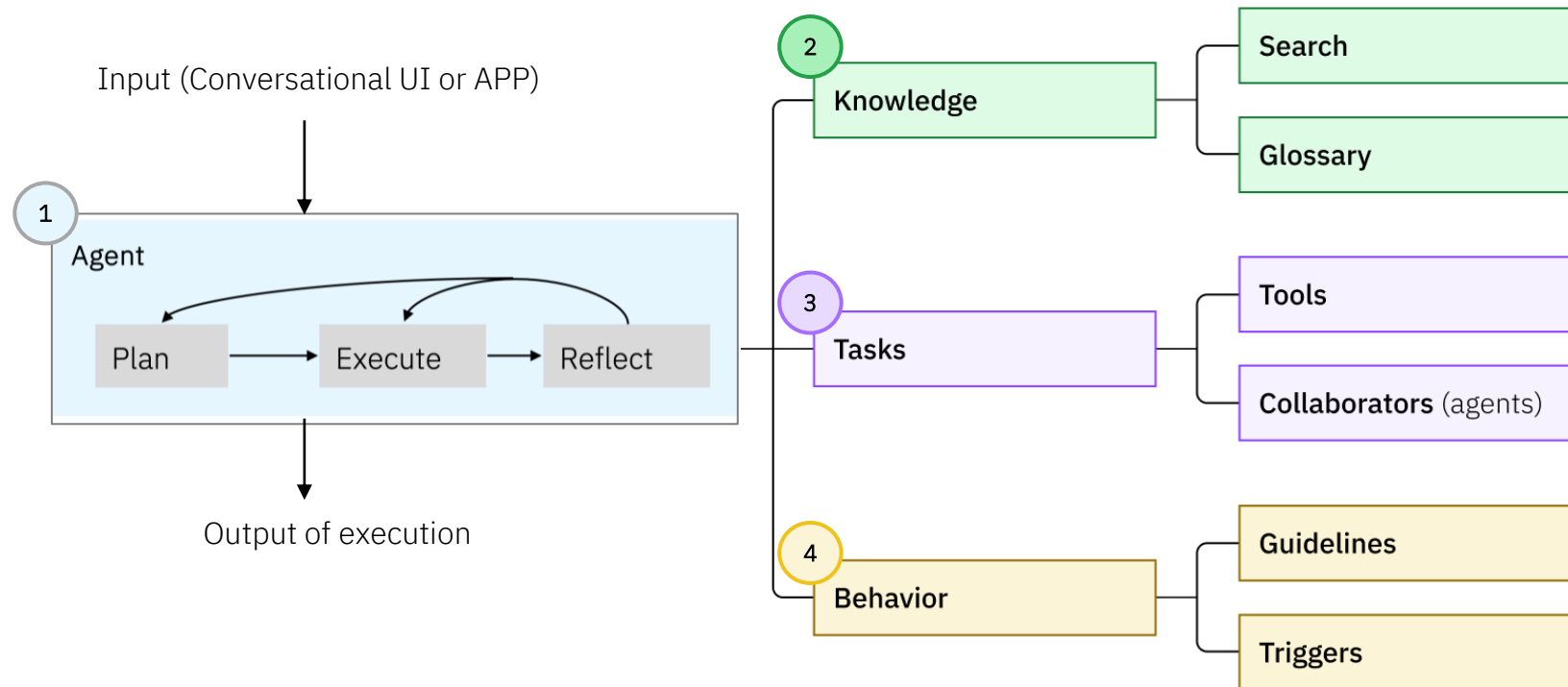
AI agents

4

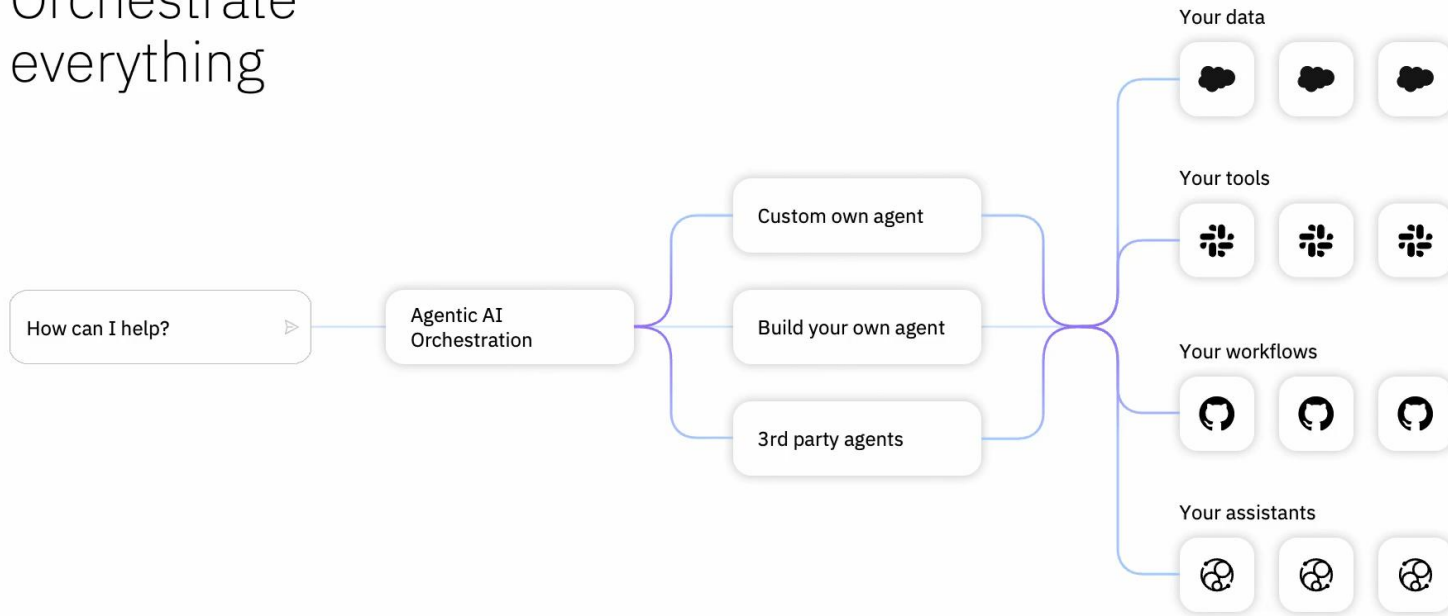
AI agenti

AI agent je aplikacija ki **deluje avtonomno** in je sposobna **razumeti**, **načrtovati**, in **izvesti naloge** (s strani človeka ali drugega agenta).

AI agenti uporabljajo LLM-je za sklepanje in se lahko povezujejo z orodji, drugimi modeli in drugimi IT sistemi za potrebe **izpolnjevanja zadanih nalog**.



Orchestrate everything

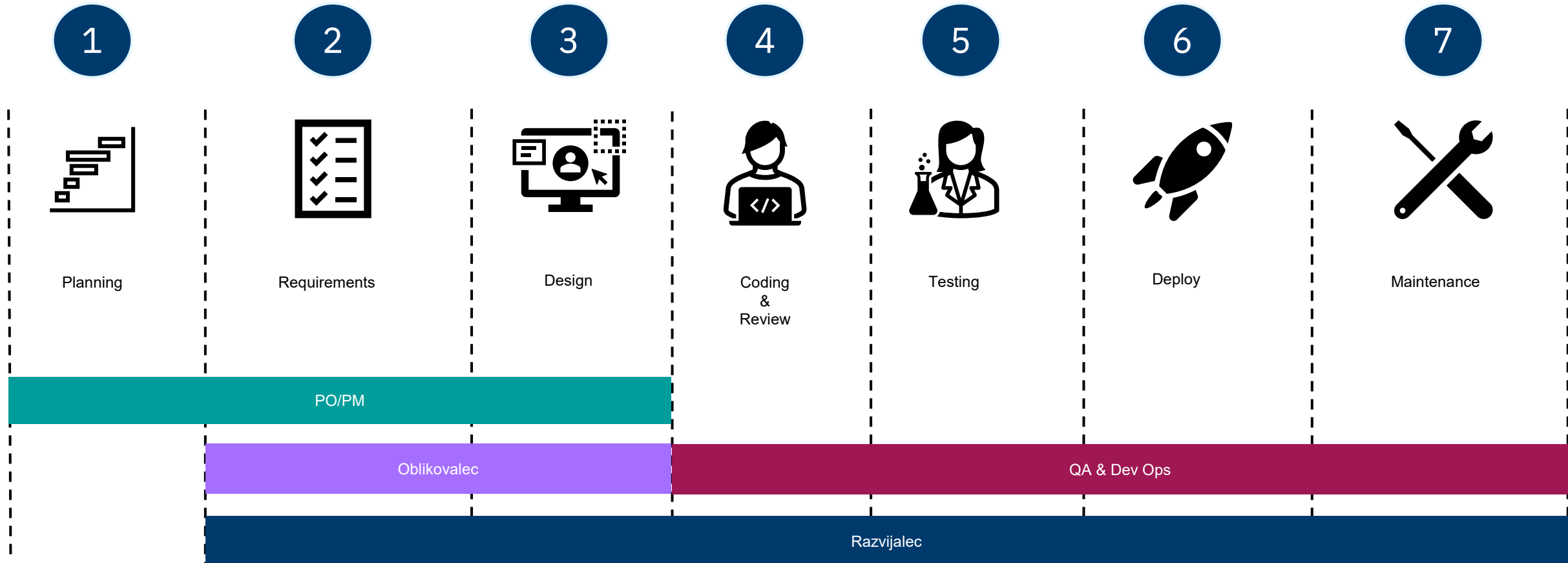


Kako lahko uporabimo AI agente pri razvoju programske opreme?



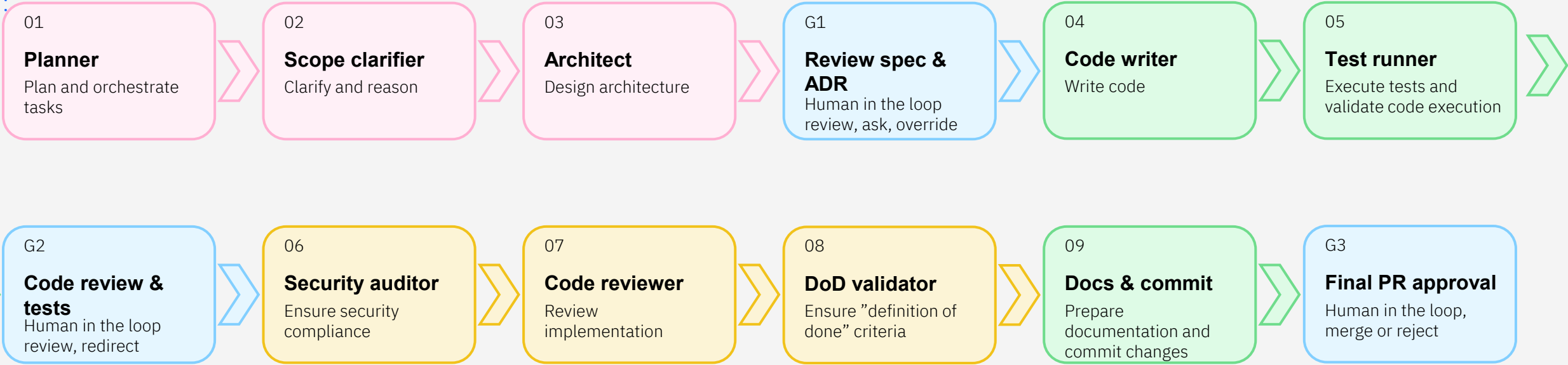
Življenjski cikel razvoja programske opreme (SDLC¹)

AI agenti vas lahko podpirajo v vseh fazah SDLC



Primer: Potek izvajanja AI agentskega sistema

Agents



- Design/Plan agents
- Human quality gate
- Execution agents
- Validator agents

Primer: Dodaj “Izvozi podatke v CSV” funkcionalnost na admin portal

Agent

S:

Design & Planning

Execution

Validation

Finalization

Planner → Breaks feature into tasks

Scope Clarifier → Defines filters, fields, limits

Architect → Designs API¹ + ADR² (streaming, async)

Code Writer → Implements API + UI

Test Runner → Runs unit, integration, perf tests

Security Auditor → Access control, vulnerabilities

Code Reviewer → Quality, maintainability

DoD Validator → Ensures “done” criteria met

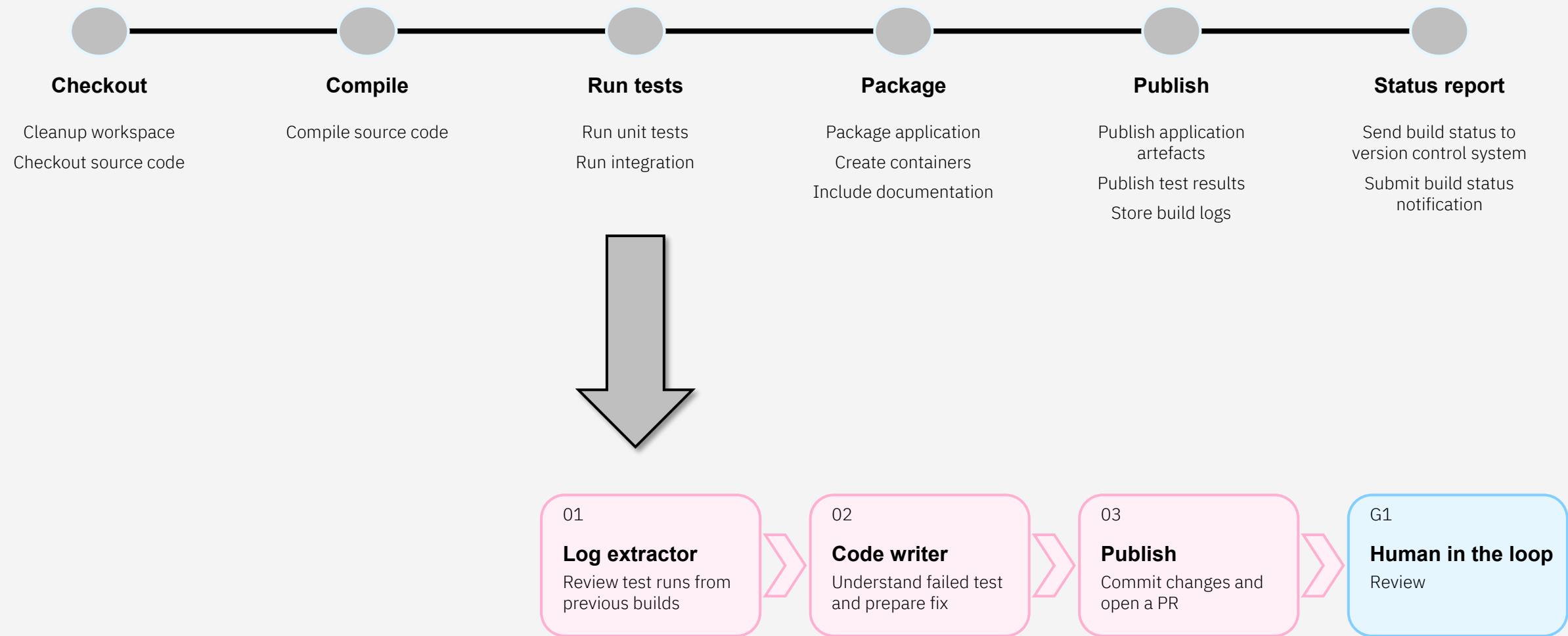
Docs & Commit → Docs, release notes, PR³

G1 → Review spec & architecture

G2 → Code review & test validation

G3 → Final approval & merge

Primer: Korak v CI/CD cevovodu ne uspe



AI agent za prvi pregled kode

Open

Demo PR #2

LukaPodgorsek wants to merge 6 commits into main from demo

github-actions bot commented on Feb 10

Hi @LukaPodgorsek, I've received your request, and I'm working on it now! You can track my progress [in the logs](#) for more details.

github-actions bot reviewed on Feb 10

View reviewed changes

github-actions bot left a comment

Review Summary

This pull request adds a `.gitignore` file to exclude PDF files from the knowledge-bases documents directory. The change is straightforward and serves a clear purpose of preventing binary PDF files from being committed to the repository.

General Feedback

- The `.gitignore` pattern `*.pdf` is appropriate for excluding PDF documents from version control
- This is a minimal change with low risk
- Consider documenting in the README or project documentation why PDFs are excluded, if this is part of a larger knowledge base management strategy

adk-project/knowledge-bases/documents/.gitignore

Outdated

@@ -0,0 +1 @@

1 + *.pdf

No newline at end of file

github-actions bot on Feb 10

The file is missing a newline at the end. POSIX standards require text files to end with a newline character. This can cause issues with some tools and version control systems.

Notifications

Customize

Unsubscribe

You're receiving notifications because you were mentioned.

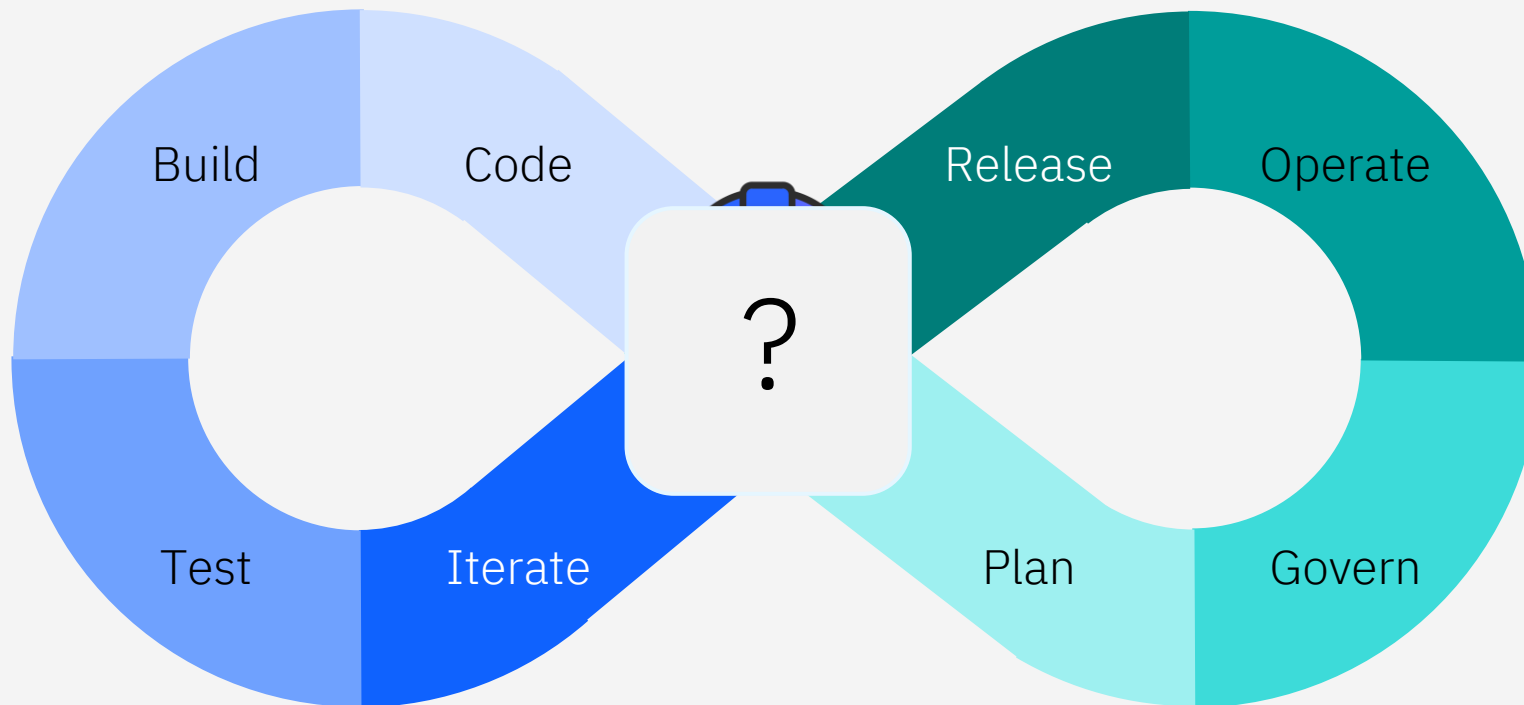
1 participant

Lock conversation

IBM Data Platform | © 2025 IBM Corporation

58

SDLC ni linearen proces



Hvala



Luka Podgoršek

Copy the legal disclaimer text shown here:

© Copyright IBM Corporation 2026

IBM and the IBM logo are trademarks of IBM Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on ibm.com/legal/copytrade.

This document is current as of the initial date of publication and may be changed by IBM at any time.

IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion. Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision. The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

THE INFORMATION IN THIS DOCUMENT IS PROVIDED "AS IS" WITHOUT ANY WARRANTY, EXPRESS OR IMPLIED, INCLUDING WITHOUT ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY OR CONDITION OF NON-INFRINGEMENT. IBM products are warranted according to the terms and conditions of the agreements under which they are provided.

Examples presented as illustrative only. Actual results will vary based on client configurations and conditions and, therefore, generally expected results cannot be provided.

Not all offerings are available in every country in which IBM operates.

It is the user's responsibility to verify the operation of any non-IBM products or programs with IBM products and programs. IBM is not responsible for non-IBM products and programs.

The client is responsible for ensuring compliance with all applicable laws and regulations. IBM does not provide legal advice nor represent or warrant that its services or products will ensure that the client is compliant with any law or regulation.



Hvala za pozornost.

Luka Podgoršek

IBM



KONFERENCA
evolucija.ai
2026



REPUBLIKA SLOVENIJA
MINISTRSTVO ZA NOTRANJE ZADEVE
IN JAVNO UPRAVO



REPUBLIKA SLOVENIJA
MINISTRSTVO ZA IZOBRAŽEVANJE, ZNANOST
IN MLADINO



REPUBLIKA SLOVENIJA
MINISTRSTVO ZA GOSPODARSTVO
DELO IN ŠPORT



REPUBLIKA SLOVENIJA
MINISTRSTVO ZA OBRAMBO





Zakaj pregled kode ne dohaja hitrost razvoja?

Lovro Hudrap

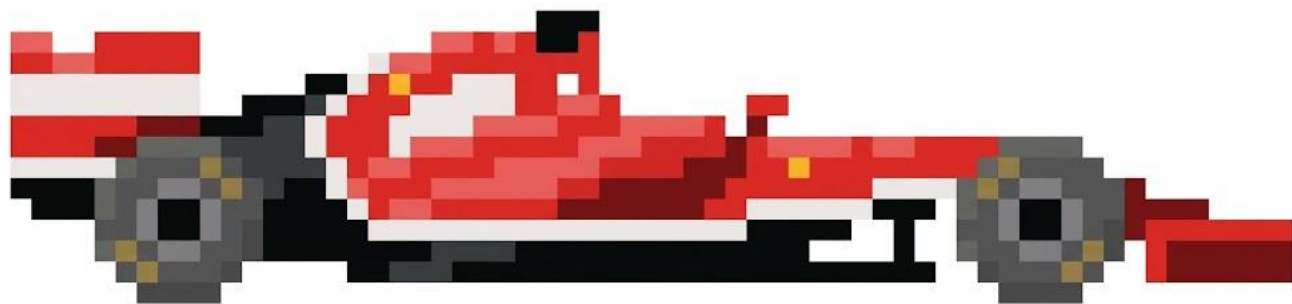
Tehnični direktor (CTO)

Cloud9devs

Tehnologijo povezujem s podjetniško vizijo in konkretnimi poslovnimi rezultati. Zanima me, kako z dobrim inženirstvom, pravimi ekipami ter uporabo umetne inteligence graditi in skalirati rešitve, ki ustvarjajo dejansko vrednost.

Dirke se dobivajo v

OVINKIH



Vsaka razvojna ekipa je dobila močnejši motor

+40%

autocomplete

+140%

interaktivni agenti

+180%

avtonomni agenti

Učinek na število commitov pri več kot 100.000 razvijalcih

Večina pospeška se izgubi pred produkcijo

+180%

več commitov

pregled
združevanje
usklajevanje

+30%

več produkcijskih izdaj

Hitrost kodiranja $\neq$ hitrosti dostave.

Koda je videti bolje. Incidenti pravijo drugače

94%

kodo ob pregledu oceni kot kakovostnejšo od tiste, ki jo spiše izključno človek

78%

vodij po uvedbi AI orodij poroča o več incidentih v produkciji

Raziskava med 200 tehnološkimi odločevalci v ZDA

Zakaj pregled odpove

01

Količina

AI lahko v nekaj minutah ustvari tisoče vrstic kode.

02

Videz

Urejena in samozavestna koda lahko skriva skrbno prikrite napake.

03

Kopičenje

Nepreverjena koda se začne kopičiti hitreje, kot jo lahko ekipe obvladujejo.

Več kode pomeni več dela za isti par človeških oči.

Trije nivoji pregleda kode

1

Avtomatska preverjanja

Jasen, preverljiv odgovor

2

Neodvisen AI-pregled

Projektni kontekst in robni primeri

3

Presoja razvijalca

Izkušnje, namen in odgovornost

Vsak nivo naslavlja drugačno vrsto tveganja.

Prvi nivo: kar ima jasen preverljiv odgovor

Se koda prevede?

So testi uspešni?

So tipi pravilni?

Je zaznana varnostna ranljivost?

Če preverjanje pade, sprememba ne sme napredovati na naslednji nivo.

Drugi nivo: neodvisen AI-pregled

- 01 Agent, ki je napisal kodo, je ne pregleduje.
- 02 AI Pregledovalec navede ugotovitve, kode ne popravlja.
- 03 Pravila pregleda živijo v repozitoriju in se izboljšujejo.

>90%

od približno 65.000 sprememb na teden

>75%

pripomb razvijalci ocenijo kot koristne

Tretji nivo: človek presoja tisto, kar šteje

PRESOJA

Je arhitekturna odločitev ustrezna?

Ali sprememba sledi poslovnemu namenu?

Kakšno tveganje prinaša?

Pozornost sledi tveganju.

Ljudje pregledajo manj, a tisto kar šteje

KAKO

Prečiščen transkript delovne seje
Odločitve med razvojem
Zavrnjena opozorila

KAJ

Povzetek dejanskih sprememb
Izvedeni testi in rezultati
Odprta tveganja

Odgovornost ostane na človeku.



DIRKA SE BO ODLOČALA V OVINKIH

Vsi imamo dostop do podobnih modelov.

Na ravni bo zato prednost, ki jo prinaša sama tehnologija, vse manjša.

SKENIRAJ QR KODO IN POŠLJEMO VAM AI PRIROČNIK





Hvala za pozornost.

Lovro Hudrap

Cloud9devs



KONFERENCA
evolucija.ai
2026



REPUBLIKA SLOVENIJA
MINISTRSTVO ZA NOTRANJE ZADEVE
IN JAVNO UPRAVO



REPUBLIKA SLOVENIJA
MINISTRSTVO ZA IZOBRAŽEVANJE, ZNANOST
IN MLADINO



REPUBLIKA SLOVENIJA
MINISTRSTVO ZA GOSPODARSTVO
DELO IN ŠPORT



REPUBLIKA SLOVENIJA
MINISTRSTVO ZA OBRAMBO





AI spreminja razvoj programske opreme – in ne le zato, ker hitreje pišemo kodo



Petra Kakovič Bizjak, David Orešič

Senior Agile Project Manager (Petra), Client Partner (David)

Endava

AI v razvoju programske opreme ne vidiva le kot orodje za hitrejše pisanje kode, temveč kot spremembo celotnega načina, kako nastajajo digitalne rešitve. Zanima naju, kako te spremembe pretvoriti v boljše odločitve, učinkovitejše sodelovanje in konkretno poslovno vrednost.

Escalators and AI?



FASTER CODE

MVP



6

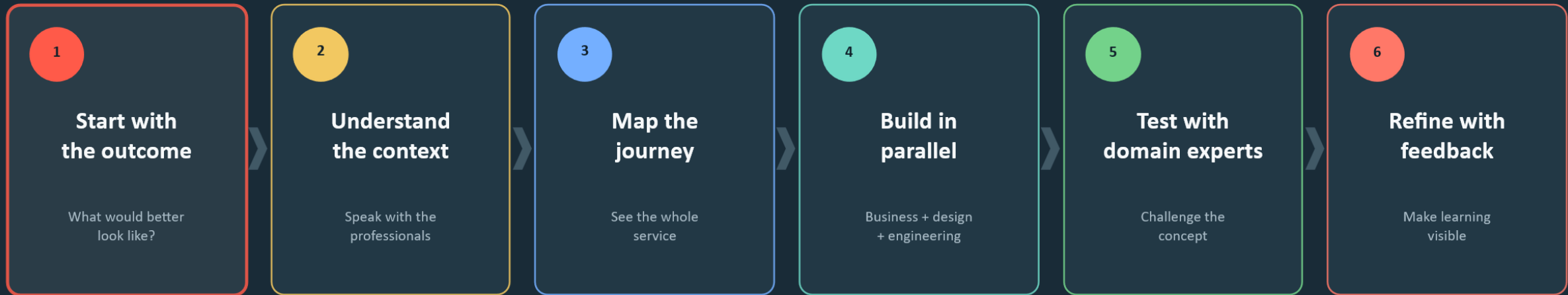
Proof of Concept

A public-sector discovery: outcome first, AI second

3-WEEK DISCOVERY

A complex, document-intensive professional decision process

No institution or client is identified



The technology was not the starting point.

The starting point was reducing time to a well-supported conclusion while keeping professional judgement decisive.

Outcome → context → journey → prototype → expert feedback → refinement

AI Planning Evaluation Assistant

Use this AI-powered assistant to evaluate planning applications, review policy compliance, and make informed planning decisions with confidence.

This service helps planning officers assess applications against local and national planning policies, identify potential issues, and streamline the decision-making process.

Select Your Local Planning Authority Region

Choose the local authority area you are working with. This will filter planning applications and policy documents to your selected region.

Active Planning Authority

Barnet



Manage Planning Applications

Review planning applications, collaborate on assessments, and keep decision workflows moving with confidence.

Manage Policy Library

Manage policy references, keep guidance up to date, and connect supporting documents to planning decisions.



Active Planning Applications in Barnet

[Return to home](#)

View and manage your planning applications for Barnet. Track the status of document processing and AI assessments, review completed evaluations, and start new application reviews.

Upload a new application

Upload planning application documents to begin an AI-powered evaluation. The system will process your documents, extract key information, and provide an assessment against local and national planning policies.

[Start new application review](#)

Planning Application	Documents	Created	AI Status	Action
25-3834-HSE	 6 docs	 Apr 15, 08:15 AM	Completed 	Delete
26-2930-HSE	 9 docs	 Apr 15, 08:13 AM	Completed 	Delete

25-3834-HSE

📍 6 Winterburn Close, New Southgate, Barnet, London, N11 3DJ

AI Recommendation

📄 RECOMMENDATION

REJECT

🔗 CONFIDENCE

High



REJECT (High confidence): the first-floor rear extension is a fixed-geometry overdevelopment on a prominent end-of-terrace corner plot, causing unacceptable character and neighbour-amenity harm that cannot realistically be cured by conditions.

Application Summary

Householder planning permission is sought for demolition of the existing rear conservatory, construction of a two-storey rear extension, and a single-storey front infill extension.

Officer's Notes

Here you can add additional context on the application to help both human and AI reviewers.

[Add notes](#)

No officer's notes have been added yet. Click "Add notes" to provide additional context for the AI evaluation.

Analysis Summary

📄 Application Documents

- The scheme is a householder proposal for a two-storey rear extension after demolition of a conservatory, plus a single-storey front infill [App Doc 4](#).
- The host dwelling is a small end-of-terrace house on a prominent corner plot with a flank to a public footpath [App Doc 16](#).
- The first-floor rear extension is 3m deep, 3.8m wide, up to 5.8m high, and only 1m from the shared boundary with No. 7 [App Doc 17](#).
- Barnet's own delegated report concludes the scheme should be refused for both character and neighbour-amenity reasons [App Doc 24](#).

Results:

- 90-98% decision accuracy
- 98-99% justification accuracy

 PlanVerify

David Orešič

Application OverviewApplication AnalysisDocuments & RevisionsDecision Letter

[View all applications](#)

Application Analysis

 Chat

Provide feedback & Update answer

Show details

 Download PDF

 Delete run & re-evaluate

AI Recommendation

 RECOMMENDATION
REJECT

 CONFIDENCE
High

REJECT (High confidence): the first-floor rear extension is a fixed-geometry overdevelopment on a prominent end-of-terrace corner plot, causing unacceptable character and neighbour-amenity harm that cannot realistically be cured by conditions.

Executive Summary

Key Facts

Policy and Constraints Assessment

Reasoning

Uncertainty, Missing Info, and Risks

Policy Conflicts and Tension

Conditions or Reasons for Refusal

Citations

Key Facts

application_type: Householder application for planning permission for works or extension to a dwelling [App Doc 1](#) [App Doc 9](#) .

site_address: 6 Winterburn Close, New Southgate, London, N11 3DJ [App Doc 1](#) [App Doc 16](#) .

proposal: Demolition of the existing rear conservatory; erection of a two-storey rear extension; single-storey front infill extension [App Doc 4](#) [App Doc 16](#) .

host_property: Two-storey end-of-terrace dwelling on a prominent corner plot; principal elevation faces the street and flank elevation adjoins a public footpath [App Doc 16](#) .

existing_baseline: There is an existing single-storey side extension forming part of the current built form, approved in 2010; previous 2010 refusal for a two-storey side extension is recorded in the delegated report [App Doc 16](#) [App Doc 17](#) .

proposed_dimensions: Ground-floor rear extension: 3.0m depth, 4.7m width, 2.4m height; first-floor rear extension: 3.0m depth, 3.8m width, 4.4m to eaves, 5.8m max height, positioned 1m from the shared boundary; front infill: 2.0m length, 2.4m width, 2.6m to eaves, 3.6m max height [App Doc 17](#) .

materials: External materials proposed to match existing yellow multi-facing brickwork, brown concrete roof tiles, white UPVC windows and doors [App Doc 6](#) .

internal_changes: Application form states 179 sqm added and one additional bedroom/bathroom [App Doc 5](#) ; proposed plans show reconfigured ground floor and three bedrooms at first floor [App Doc 30](#) .

consultation: 23 neighbour consultation letters sent; 1 objection received raising privacy, visual impact and loss of light among other matters [App Doc 18](#) .

geospatial_constraints: Site lookup indicates: not within conservation area, not within flood risk zone, not within green belt, not within a listed building boundary, and not within a heritage-at-risk area [Source 1](#) .

endava

Hvala za pozornost.

Petra Kakovič Bizjak, David Orešič

Endava



KONFERENCA
evolucija.ai
2026



REPUBLIKA SLOVENIJA
MINISTRSTVO ZA NOTRANJE ZADEVE
IN JAVNO UPRAVO



REPUBLIKA SLOVENIJA
MINISTRSTVO ZA IZOBRAŽEVANJE, ZNANOST
IN MLADINO



REPUBLIKA SLOVENIJA
MINISTRSTVO ZA GOSPODARSTVO
DELO IN ŠPORT



REPUBLIKA SLOVENIJA
MINISTRSTVO ZA OBRAMBO





Od AI razvojnih orodij do zasebnega razvojnega okolja: produktivnost in digitalna suverenost



Janez Martin Kričej, Timotej Petrič

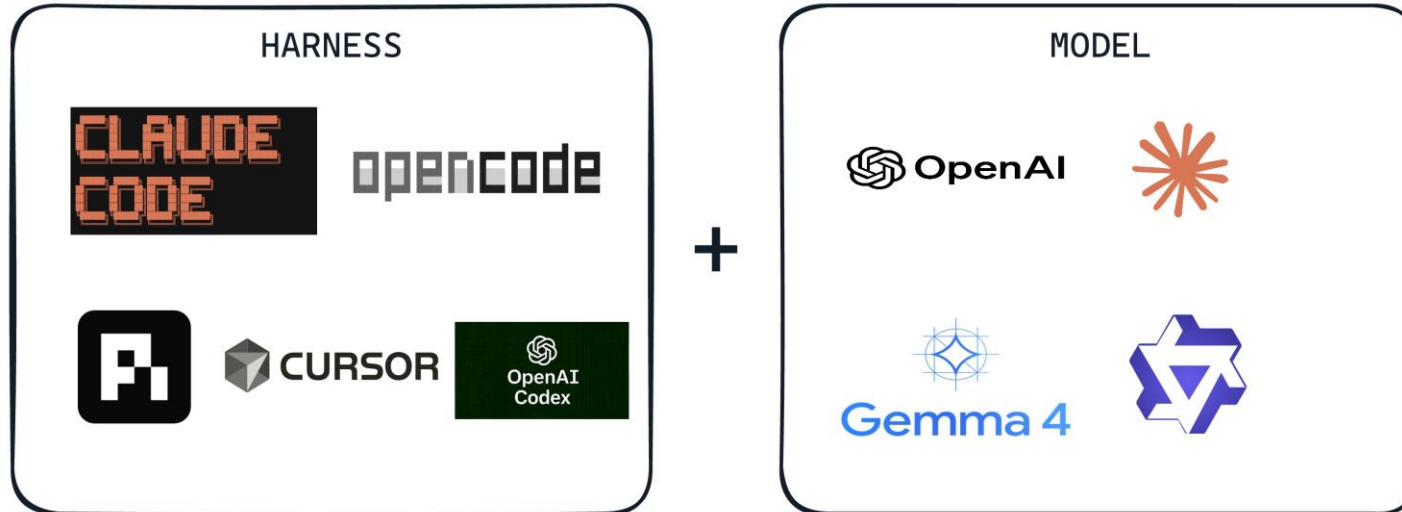
AI Platform Engineer (Janez), AI Software Engineer (Timotej)

Agenda OpenSystems

Raziskujemo, kako AI razvojna orodja in agenti spreminjajo razvoj programske opreme ter kako njihove zmogljivosti prenesti v varna, nadzorovana in zasebna razvojna okolja. Povezujeva razvoj, avtomatizacijo, AI Ops in sodobno infrastrukturo v praktične rešitve za uporabo umetne inteligence pri vsakodnevnem razvoju.

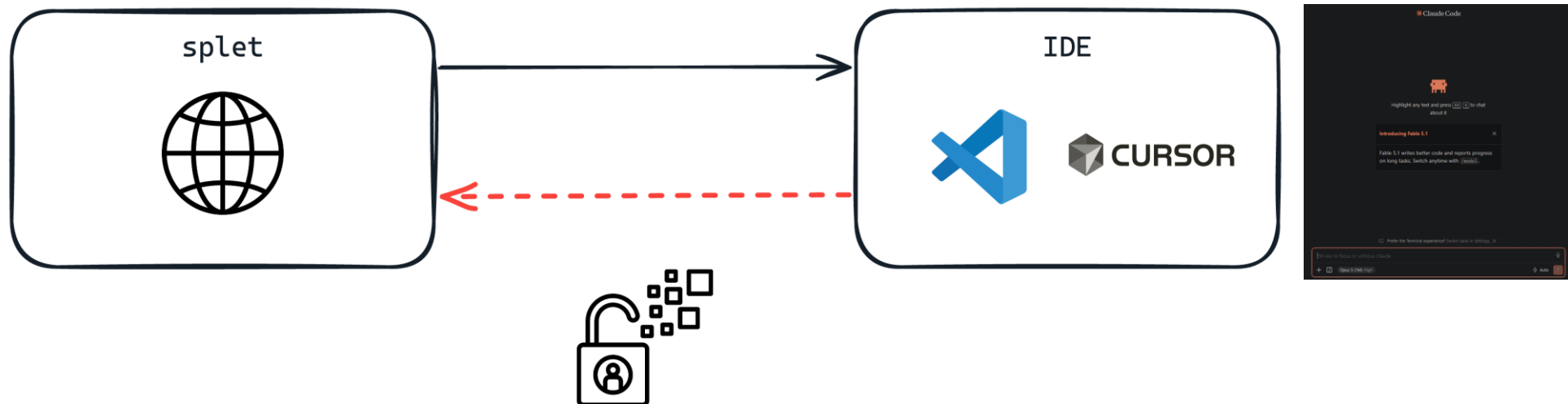
Agent = orodje + model

- Kje se produktivnost izvaja?
- Agent – dve ločeni odločitvi
- Ključen je nadzor nad obema komponentama.



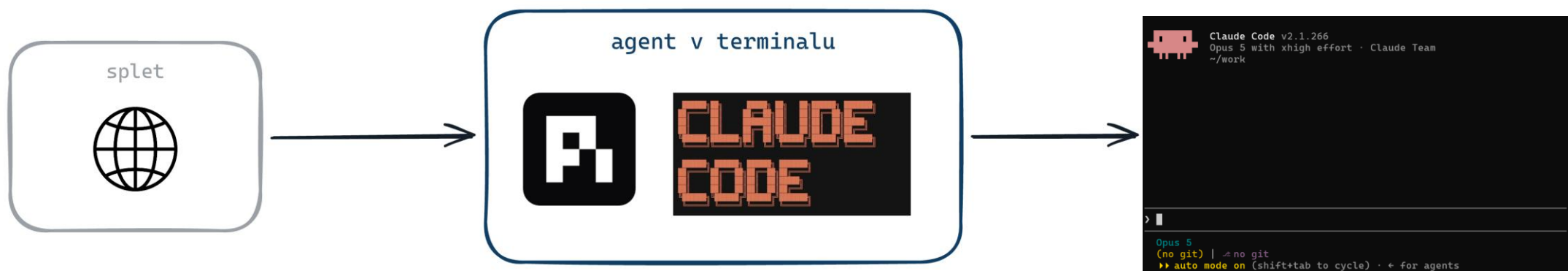
Doba kopiraj → prilepi

- Spletni vmesnik -> prompt -> kopiraj/prilepi
- Nihče ni sledil, kaj vse je pristalo v spletnem vmesniku
- Koda stranke v oblak?



Agent v terminal (CC)

- Agent v terminalu, dostop do repozitorija, naloga dejansko opravljena
- Prvič: od "predloga kode" do izvedbe
- Brez procesa, brez pravil `curl -fsSL https://claude.ai/install.sh | bash`
- OpenCode: 1:1 odprtokodna zamenjava, vir inference po izbiri
- Pridobljen čas gre v dokumentacijo, teste, logiranje in raziskavo



Uporaba agentov -> nove ranljivosti

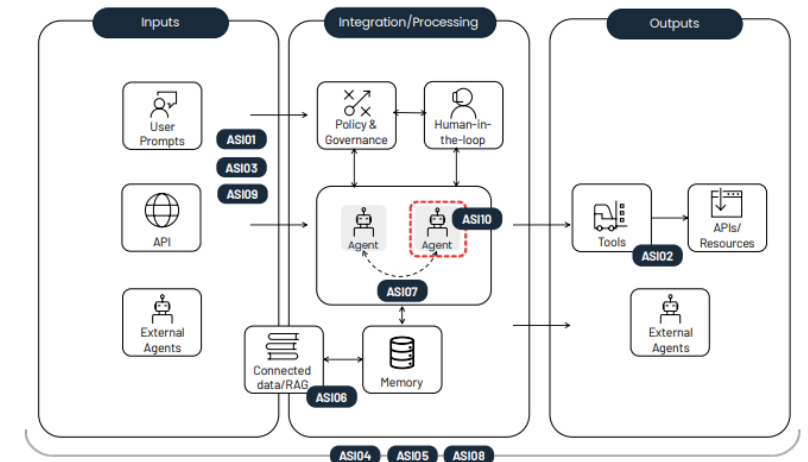
- Brez omejitev -> dostop do vseh datotek
- Več CVE-jev
- Analiza ranljivosti in postopen nadzor

Claude Code RCE Vulnerability: How a Malicious Pull Request Executes Code

AI Aug 6, 2026

Claude Code Leak 2026: What The 512,000-Line Source Exposure Actually Revealed

Agentic Top 10 At A Glance



ASI01: Agent Goal Hijack

ASI03: Identity & Privilege Abuse

ASI05: Unexpected Code Execution (RCE)

ASI07: Insecure Inter-Agent Communication

ASI09: Human-Agent Trust Exploitation

ASI02: Tool Misuse & Exploitation

ASI04: Agentic Supply Chain Vulnerabilities

ASI06: Memory & Context Poisoning

ASI08: Cascading Failures

ASI10: Rogue Agents

Sprememba pravil uporabe

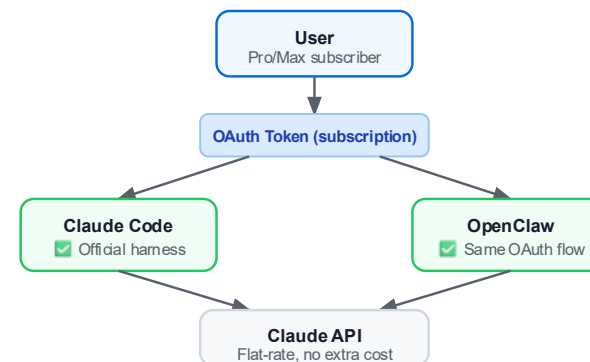
- Anthropic, apr. 2026 – omejitev naročnin
- Pay per usage (Anthropic API)
- Skrivanje “thinking” izhoda
- AGENTS.md podpora?

```
· Bootstrapping... (2s · thinking with xhigh effort)
```

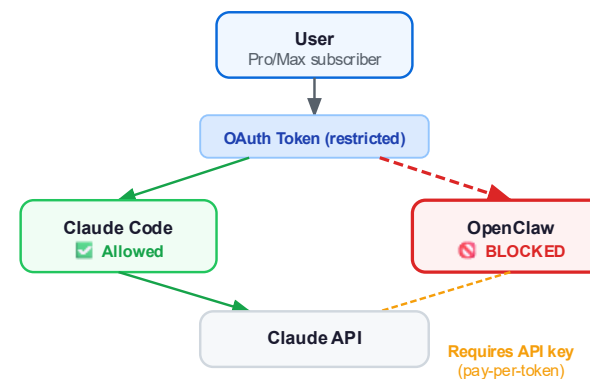
How Anthropic Blocked Third-Party Tools

OAuth authentication flow — before and after April 4, 2026

BEFORE — The OAuth Loophole



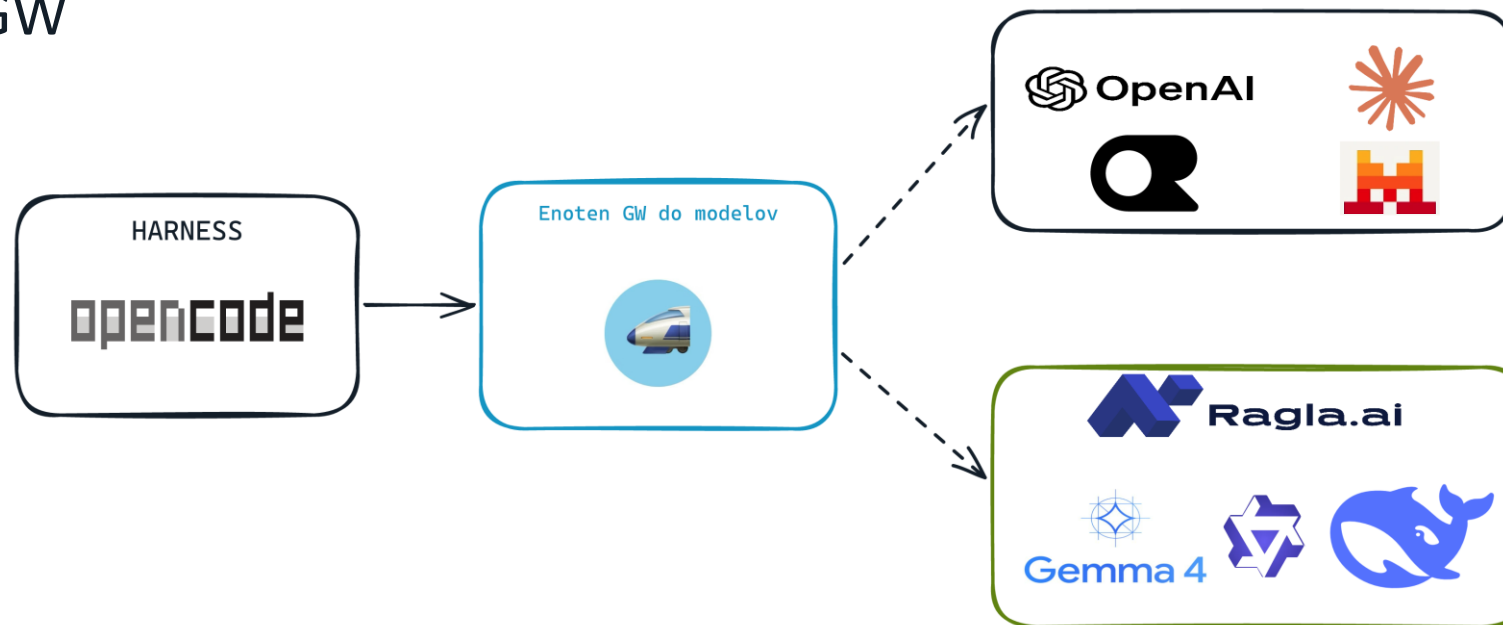
AFTER — April 4, 2026 Enforcement



© DecodeTheFuture.org · April 2026

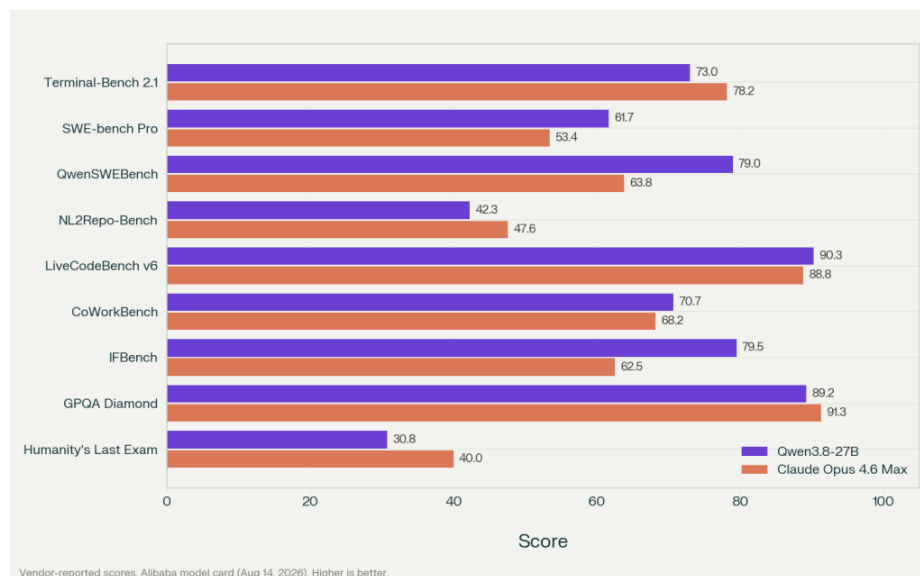
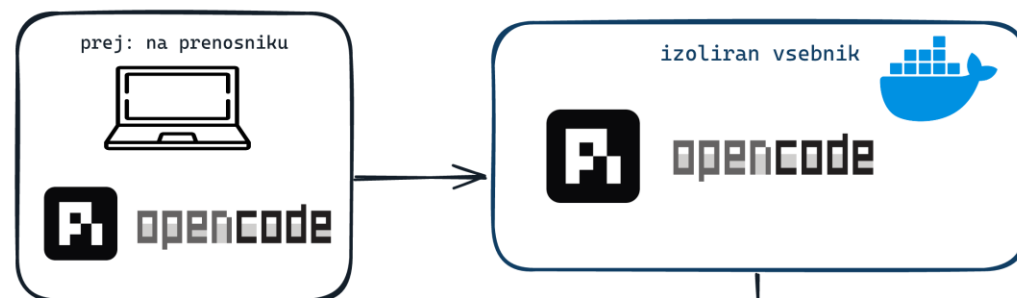
Ločitev orodja in modela

- Zamenljivost agentov
- Odprtokoden harness
- Kombinacija lokalnih in javnih modelov (zaprtokodnih)
- En API GW



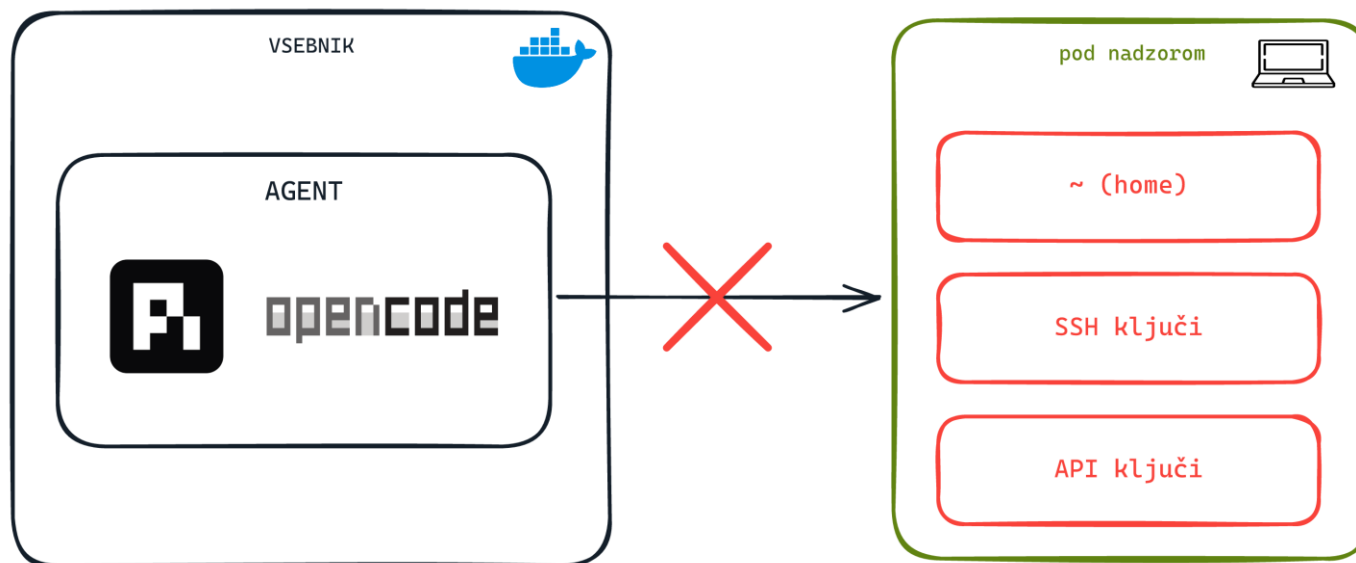
Iz prenosnika v vsebnik

- Opencode / pi.dev
- Izolacija v vsebnik
- Opredelitev pravic in dostopa
- RH Validated



Zmogljivosti izolacije

- Onemogočen dostop
- Agent ne “pobegne”
- Proxy nadzor -> kam se povezuje
- Izogib “prompt injection scraping”



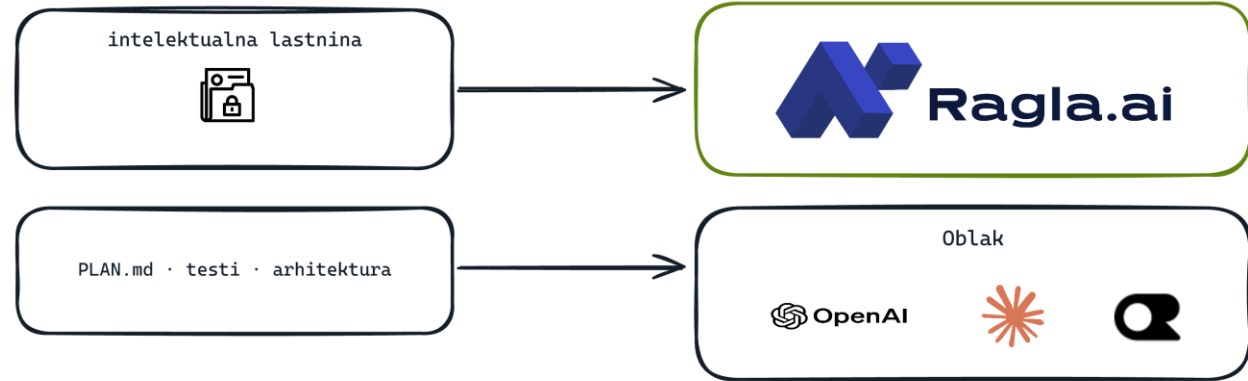
Your AI Agent Can Read Every Secret in Your .env File Right Now — Here's Proof

#ai #tutorial #webdev #security

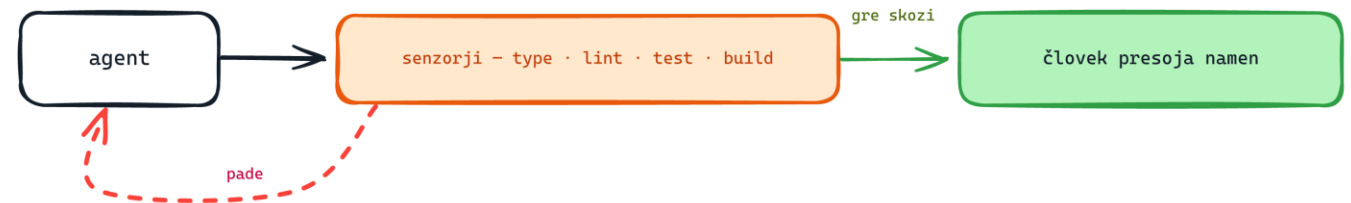
Razvoj

- Podrobna specifikacija
- Opredelitev kritičnosti/tveganj
- Agent teče v vsebniku
- MCP za zunanje baze znanj
- Iterativno implementiranje dobrih praks

usmeritev glede na tveganje



preverjanje – back pressure



Skladna produktivnost



Izolirani agenti – ORCA IDE

do handoff but save it to handoff.md where PRD is

- Skill(handoff)
 - ↳ Successfully loaded skill
- Write(docs/issues/widget-terms-and-conditions/handoff.md)
 - ↳ Wrote 60 lines to docs/issues/widget-terms-and-conditions/handoff.md
 - 1 # Handoff – Widget Terms and Conditions
 - 2 consent
 - 3 ## State: design
 - 4 complete, ready for
 - 5 Nothing has been implemented yet. This e from a raw feature ur
 - 6 research, an extensive design grill, a
 - ... +178 lines
- Saved to docs/issues/widget-terms-and-conditions/handoff.md, alongside the PRD. It points to the ADR, PRD, and all 7 issues rather than duplicating their content, and flags the decisions a fresh agent shouldn't re-litigate plus recommended skills (tdd-implement-scope, code-review, run) for the next session.

* Sautéed for 23s · done 11:41 AM

※ recap: Designed and fully scoped the widget Terms and Conditions consent feature (ADR, PRD, 7 ready-for-agent issues, handoff doc all saved). Everything's ready for implementation – next step is running issues 01 and 03 through an implementation agent. (disable recaps in /config) new task? /clear to save 312.6k...

Read 1 file

- Good, structure is valid. Now running tests.
- Ran 1 shell command
- All passing. Now typecheck, lint, build.
- Ran 1 shell command
- Now let's update the backend gate and the test tenant's config, then verify visually + keyboard-only.
- Ran 2 shell commands
- This is the second pre-existing, unrelated gap the earlier subagent found (test expects lang: "en" but the site declares sl) – a content mismatch, not something to fix as part of this work. Let's check integration tests quickly, then update the live test tenant's config.
- Searched for 1 pattern, ran 2 shell commands
- Update(tests/integration/api/test_chatbot_config.py)
 - ↳ Added 1 line
 - 607 "session_id_label":
 - 608 "Session ID",
 - 609 "copied_confirmation
 - 610 "Copied!",
 - 611 "notice_text": "",
 - 612 "consent_text": "",
 - 613 },
 - "ally": {
 - "close_label": "Clos
 - e",
- Re-running integration tests · 42s
 - ↳ \$ make test-integration 2>&1 | tail -10 (41s) (ctrl+b to run in background)
- Pollinating... (42m 2s · ↓ 138.2k tokens)
 - ↳ Tip: Use /clear to start fresh when switching topics and free up context

Want me to check orca status or demonstrate any of these?

- Build · Qwen/Qwen3.8-27B-FP8-opencode · 40.7s

what can orca do?

+ Thought: 1.4s

Orca is a multi-agent dev environment – a CLI + app for orchestrating coding agents. Core capabilities:

- Worktrees – create/track git worktrees, spawn Codex/Claude agents inside them, link Linear issues, hand off work between agents
- Terminals – create PTYs, send input, wait for command exit (terminal create/send/wait)
- Embedded browser – full automation: navigate, snapshot, click, fill forms, eval JS, screenshots, file uploads

Build · Qwen/Qwen3.8-27B-FP8-opencode AG-Smartypants-VLLM

/home/timotejp/work/ag-rag-fc config.yaml 36.3K ctrl+p commands

OpenCode

opencode

Ask anything... "Fix a TODO in the codebase"

Build · Qwen/Qwen3.8-27B-FP8-opencode AG-Smartypants-VLLM

config.yaml tab agents ctrl+p commands

- Tip Use /status or ctrl+x s to see system status info

~/work/ag-rag-fc:feat/widget-terms-and-conditions ● 2 MCP /status 1.18.29

<https://github.com/stabliyai/orca>



Hvala za pozornost.

Janez Martin Kričej
Janez.kricej@agenda.si
040 287 370 (GSM)



KONFERENCA
evolucija.ai
2026



REPUBLIKA SLOVENIJA
MINISTRSTVO ZA NOTRANJE ZADEVE
IN JAVNO UPRAVO



REPUBLIKA SLOVENIJA
MINISTRSTVO ZA IZOBRAŽEVANJE, ZNANOST
IN MLADINO



REPUBLIKA SLOVENIJA
MINISTRSTVO ZA GOSPODARSTVO
DELO IN ŠPORT



REPUBLIKA SLOVENIJA
MINISTRSTVO ZA OBRAMBO





KONFERENCA
evolucija.ai
2026

Gospodarska
zbornica
Slovenije
Chamber of Commerce
and Industry of Slovenia



PROGRAMI

Partner razvojne smeri



Hvala za pozornost.