

“Mākslīgā intelektā (MI) balstīts automatizācijas risinājums pilnam programmatūras izstrādes ciklam”

Projekta kopsavilkums

SIA *Autentica* plāno īstenot visaptverošu iekšējo procesu automatizācijas projektu, izstrādājot un ieviešot **specializētus mākslīgā intelektā (MI) bāzētus rīkus** programmatūras izstrādes uzdevumu atbalstam un optimizācijai, kas aptvers tādas jomas kā programmatūras rakstīšana, testēšana, dokumentēšana, projektu vadība, datu analīze, zināšanu pārvaldība, kvalitātes kontrole un kiberdrošība.

Projekta realizācija notiks šādos secīgos soļos:

1. centrālas tehnoloģiju un rīku integrācijas platformas (framework) izveide,
2. lokālas lietojumprogrammas MI asistentu darbināšanai darbastacijās izstrāde,
- 3.-12. specifisku MI asistentu kopas izstrādi, ieviešanu un apmācīšanu.

MI asistentu kopu veidos šādas komponentes:

3. Koda Bāzes Izpētes Asistentu,
4. Koda Rakstīšanas Asistentu,
5. Automatizēts Koda Pārskatīšanas Asistentu,
6. Projektu Vadības administratīvu Asistentu,
7. Tīmekļa Lietotni sarunbotam,
8. Dinamiskās Datu Izpētes Aģentu,
9. Zināšanu Bāzes Papildināšanas Asistentu,
10. Kiberdrošības un Ievainojamību Aģentu,
11. Darbinieku profila uzturēšanas asistentu,
12. Kvalitātes kontroles asistentu ienākošās informācijas pārbaudei.

Katra no šīm komponentēm ir vērsta uz konkrētu izaicinājumu risināšanu un procesu optimizāciju, lai nodrošinātu būtisku programmatūras izstrādes pakalpojumu sniegšanas uzlabojumu un attīstību.

Projekta rezultātā tiks izveidota privātā infrastruktūrā darbināma MI rīku integrāciju vide, kas nodrošinās sensitīvas informācijas aizsardzību un kalpos kā ietvars arī citu/jaunu MI bāzētu rīku darbināšanai.

Projekta īstenošanas grafiks

Plānots, ka projekts tiks īstenots **8 mēnešu laikā**, pielietojot spējās izstrādes (Agile) metodi un paredzot, ka strādājoša programmatūra ir primārais darbu izpildes progressa mērījums.

Esam sagatavojuši mūsu sākotnējo priekšlikumu projekta realizācijas kalendārajam plānam, kuru paredzēts saskaņot pirms projekta darbu uzsākšanas:

Izstrādes komandas gan jaunajos, gan esošajos projektos var integrēt MI funkcionalitātes (asistentus), lai palīdzētu veikt specifiskus programmatūras izstrādes un uzturēšanas uzdevumus, piemēram, koda bāzes izpēti un koda pārskatīšanu. Līdzīgi, integrāciju ar MI var veikt kā daļu no projektu biznesa funkcionalitātes, ja un kad šāda vajadzība rodas.

Platforma veidos pamatu visām pārējām MI iespējām un ietvers arī tehnoloģisko nodrošinājumu:

- aparatūru (serverus un grafikas kartes) to konfigurāciju,
- programmatūru, kas ļauj darbināt lielos valodu modeļus (LLM) izmantojot atvērtā pirmkoda tehnoloģijas, API, datubāzi, piekļuves kontroli, lietošanas uzraudzību un paplašinātas funkcijas, kas nav pieejamas standarta risinājumos.

Tehnoloģiju modelis

Specializētā aparatūra: Nvidia L4 grafisko procesoru kartes ar vismaz 24GB, vai citas līdzvērtīgas veiktspējas kartes (izmantoto karšu modeļi un konfigurācijas tieši ietekmē risinājuma veiktspēju).

Serveru aparatūra: GNU/Linux (RHEL saderīgas distribūcijas), ar Docker atbalstu, servera resursi ar vismaz 4 CPU kodoliem, 16 GB RAM, >200 GB disku vietu, un izvietoti atbilstošā datu centra infrastruktūrā. Serverim jāatbalsta Nvidia L4 grafisko procesoru karšu darbināšana.

Piezīme! Šeit norādītas ieteicamās aparatūras prasības. Projektā esam paredzējuši no jauna iegādāties 5 (piecas) Nvidia L4 24GB grafiskās procesoru kartes, kas nodrošinās MI asistentu/aģentu darbināšanu, un kuras tiks izvietotas mūsu rīcībā esošajā datu centra infrastruktūrā.

Programmatūra:

- Modeļu palaišana ar Ollama, LLM un citu atvērtā pirmkoda tehnoloģiju.
- Servera puses risinājums (API un DB), kas nodrošina:
 - piekļuves kontroli un kontu reģistrāciju,
 - statistikas vākšanu par modeļu izmantošanu,
 - papildfunkcionalitāti, ko esošie risinājumi neatbalsta.

Lietotājstāsts

"Es kā programmētājs jaunā projektā vēlos pievienot funkcionalitāti, kas izmanto MI tehnoloģijas datu apstrādei, nodrošinot strukturētu izvadi, klasifikāciju un integrāciju ar citām IT sistēmām."

Galvenās iezīmes

- Centralizēts MI integrācijas rāmis ar vienotu API pieeju.
- Elastīgs aparatūras atbalsts (Nvidia, Intel).
- Atbalsts atvērtā pirmkoda LLM risinājumiem (Ollama u.c.).
- Droša un lokāla datu apstrāde, kas atbilst stingrām drošības prasībām.
- Ātra biznesa loģikas ieviešana bez nepieciešamības izstrādāt sarežģītas sistēmas no nulles.
- Integrēta uzraudzība un kontrole (statistika, piekļuves pārvaldība, auditi).

Ieguvumi

- Paātrināta MI funkcionalitātes ieviešana jaunos projektos.
- Standartizēta un kontrolēta MI modeļu un rīku izmantošana.
- Samazinātas izstrādes un uzturēšanas izmaksas.
- Ilgtspējīgs pamats nākotnes MI tehnoloģiju ieviešanai.

2. Lokāla lietojumprogramma MI asistentu darbināšanai darbastacijās (Workstation App)

Apraksts

Darbstacijas lietojumprogramma, kas ļauj darbiniekiem **droši un privāti izmantot MI funkcionalitāti uz sava datora, neaugšupielādējot sensitīvu kodu vai datus mākonī.**

Risinājums pieprasījumus novada uz MI integrācijas rāmi, kas var darboties:

- uzņēmuma serveru telpā uz uzņēmuma aparatūras, vai
- datu centros ar paaugstinātu drošību.

Rāmī nodrošināti API, kas ir:

- saderīgi ar jebkuru Ollama API vai OpenAI API atbalstošu rīku, vai
- integrējami ar Ollama API vai OpenAI API formāta rīkiem.

Darbības modelis

Programmētājs strādā pie klienta projekta ar stingrām drošības prasībām, kas aizliedz koda augšupielādi ārējos servisos.

Izmantojot lokālo MI rīku, viņš var:

- saņemt palīdzību koda refaktorēšanā,
 - ģenerēt dokumentāciju,
 - izmantot citus MI palīgriekus,
- jo viss process notiek uz darbinieka datora kopā ar serveriem uzņēmumā, izmantojot iepriekš minēto platformu, neiznesot sensitīvus datus ārpus uzņēmuma robežām.

Tehnoloģiju modelis

Darba stacijas: Windows, macOS, Linux.

Funkcionalitāte: saziņa ar MI integrācijas rāmi un tā servera puses komponentiem.

Lietošana: tieši caur API (programmatiski), vai integrējot ar populāriem MI rīkiem.

Atbalstītie rīki (piemēri): Continue.dev, GitHub Copilot, JetBrains AI, RooCode, Cline, Aider, Qwen Code, Goose u.c.

Lietotārstāsts

"Es kā programmētājs, kas strādā ar sensitīviem klientu datiem, vēlos izmantot MI palīgu, bet man ir aizliegts sūtīt kodu uz ārējiem serveriem. Man ir nepieciešams rīks, kas nodrošina, ka visi dati paliek uzņēmuma vidē."

Galvenās iezīmes:

- Dati glabājas lokāli.
- Modeļus var izvēlēties pēc vajadzības.
- Elastīga integrācija ar dažādām izstrādes vidēm.

Ieguvumi

- Nodrošināta datu drošība un konfidencialitāte.
- Iespēja izmantot MI rīkus projektos ar stingrām drošības prasībām.
- Samazinātas izmaksas par mākoņpakalpojumu API izmantošanu.
- Neatkarība no interneta savienojuma kvalitātes.

MI asistentu kopas izstrādi, ieviešanu un apmācīšanu

3. Koda bāzes izpētes asistents (Codebase R&D Assistant)

Apraksts

Virtuāls asistents, kas palīdz programmētājiem **izprast un strādāt ar lielām, sarežģītām vai nepazīstamām koda bāzēm.**

Asistents nodrošina:

- kontekstuālus ieteikumus,
- koda plūsmas skaidrojumu,
- atbalstu jaunas funkcionalitātes realizācijā, balstoties uz esošajiem paraugiem un labākajām praksēm.

Praksē tas palīdz pamanīt niansas un detaļas, kas programmētājam citādi varētu paslīdēt garām.

Darbības modelis

Sprinta plānošanas laikā, Programmētājs izmanto asistentu, lai novērtētu uzdevuma apjomu nepazīstamā koda daļā un identificētu potenciālos riskus.

Ikdienas darbā (sprinta izpildes laikā) asistents palīdz ātrāk orientēties kodā, samazinot laiku, kas nepieciešams izpētei un analīzei. To var izmantot gan rakstot jaunu kodu, gan uzturēšanas darbus projektos ar daudzu gadu vēsturi un vairāku izstrādātāju iesaisti

Tehnoloģiju modelis

Izmanto iepriekš definēto lokālo MI rīku un integrācijas rami un nodrošina divas pieejas:

- Failu kopas izpēte – analizē un apkopo informāciju no ievadītajiem failiem.
- Aģentiskie rīki – paši pieprasa failus, kurus vajadzētu apskatīt, un ar programmētāja atļauju pievieno tos kontekstam.

Tas nozīmē, ka kļūst iespējams analizēt desmitiem vai pat simtiem failu saturu ātrāk un kvalitatīvāk nekā cilvēks viens pats, nodrošināt plašāku un precīzāku kontekstu izstrādes uzdevumiem.

Lietotājstāsts

"Es kā programmētājs, kuram jāveic izmaiņas 5 gadus vecā projektā, vēlos ātri saprast, kā darbojas konkrēts modulis un kādas ir labākās prakses jaunas funkcionalitātes pievienošanai, lai varētu pabeigt uzdevumu ātrāk un ar mazāku kļūdu risku."

Galvenās iezīmes

- Koda plūsmas vizualizācija un skaidrojumi.
- Kontekstuāli ieteikumi, kas balstīti uz esošo kodu un paraugpraksēm.
- Atbalsts lielu un mantotu koda bāzu izpētei.
- Automātiska failu atlase un pievienošana kontekstam.
- Integrācija ar izstrādes procesu (plānošana, izstrāde, uzturēšana).

Ieguvumi

- Samazināts laiks jaunu darbinieku apmācībai un ievadīšanai projektā.
- Paātrināta izstrāde mantotās sistēmās (legacy code).

- Uzlabota koda kvalitāte, ievērojot esošos stilus un paternitātes.
- Samazināts "bus factor" risks, jo zināšanas par sistēmu nav koncentrētas viena cilvēka galvā.

4. Koda rakstīšanas asistents (Coding Assistant)

Apraksts

IDE integrēts rīks, kas **reāllaikā palīdz koda rakstīšanā**. Tas darbojas kā uzlabots *autocomplete*, nodrošinot ne tikai atsevišķu rindiņu, bet **veselu koda bloku pabeigšanu**, automātisku komentāru un dokumentācijas ģenerēšanu. Tāpat pieejami labās izstrādes prakses ieteikumi, kas īpaši noder programmētājiem ar mazāku pieredzi konkrētajās tehnoloģijās.

Darbības modelis

Ikdienas izstrādes procesā programmētājs izmanto asistentu, lai paātrinātu koda rakstīšanu un nodrošinātu, ka kods tiek uzreiz dokumentēts. Tas palīdz ievērot "Definition of Done" (piemēram, prasība par komentāriem un dokumentāciju), kā arī uzturēt adekvātu izstrādes tempu jeb "Velocity" (SCRUM koncepti).

Tehnoloģiju modelis

Tieša integrācija ar IDE, izmantojot iepriekš minētās sistēmas komponentes un IDE spraudņus.

Atbalstītās izstrādes vides:

- Visual Studio Code,
- JetBrains produkti,
- Visual Studio,
- citas IDE ar integrācijām caur Ollama vai OpenAI API.

Daļa no šīm izstrādes vidēm ir gan bezmaksas, gan arī pilnībā atvērtā pirmkoda, as nodrošina plašu un universālu atbalstu izstrādātājiem.

Lietotājstāsts

"Es kā programmētājs vēlos rakstīt kodu ātrāk un efektīvāk, saņemot kontekstuāli atbilstošus ieteikumus un automātiski ģenerētu dokumentāciju, lai varu koncentrēties uz problēmas risināšanu, nevis standarta koda rakstīšanu, un spētu ievērot labās prakses pat bez īpašas pieredzes konkrētajā projekta tehnoloģiju kopumā."

Galvenās iezīmes

- Reāllaika koda pabeigšana un ieteikumi (rindas, bloki, paraugi).
- Automātiski ģenerēti komentāri un dokumentācija.
- Kontekstuāli ieteikumi, balstīti uz labajām praksēm.
- Plašs tehnoloģiju atbalsts (backend, frontend, datu bāzes, skripti, CI/CD).
- Integrācija ar populārākajiem IDE rīkiem un izstrādes platformām.

Ieguvumi

- Būtiski paātrināts koda rakstīšanas process.
- Uzlabota koda lasāmība un uzturamība, pateicoties automātiski ģenerētiem komentāriem.
- Mazāk kļūdu, jo asistents piedāvā pārbaudītus koda paraugus.
- Vienkāršota jaunu tehnoloģiju apguve, jo asistents var ģenerēt paraugus nepazīstamā valodā vai ietvarā.

- Plašs ietvaru atbalsts: React, Angular, Vue, Spring Boot, Laravel, Express.js, Django, ASP.NET, Ruby on Rails un citiem,
- Atbalsts datu bāzēm un glabātuvēm: PostgreSQL, MySQL, MariaDB, Oracle, SQLite, MongoDB, Redis, RabbitMQ, Apache Kafka,
- Atbalsts čaulas skriptiem un DevOps rīkiem: Bash, PowerShell, GitLab CI, Ansible u.c.

5. Koda pārskatīšanas asistents (Code Review)

Apraksts

MI aģents, kas automātiski veic koda pārskatīšanu (code review). Tas analizē iesniegto kodu (piemēram, GitLab Merge Request), pārbauda atbilstību stila vadlīnijām, meklē potenciālas kļūdas, drošības ievainojamības un piedāvā ieteikumus uzlabojumus.

Atruna: Industrijā parādās līdzīgi rīki, piemēram, *CodeRabbit*, taču tie pārsvarā balstās mākoņpakalpojumos, kuru izmantošana daudzos projektos nav atļauta.

Savukārt tādiem rīkiem kā *SonarQube* pietrūkst konteksta un dabīgās valodas izvades, lai sniegtu skaidrus ieteikumus.

Ir nepieciešams risinājums, kas:

- darbojas uzņēmuma infrastruktūrā,
- patur visus datus pie sevis,
- integrējas ar pirmkoda pārvaldības platformām (piem., GitLab).

Darbības modelis

Rīks tiek integrēts CI/CD procesā. Kad programmētājs izveido Merge Request, asistents:

- automātiski veic pirmo pārskatīšanas kārtu,
- publicē savus komentārus.

Šādi tiek atslogoti komandas biedri no rutīnas pārbaudēm, ļaujot viņiem koncentrēties uz sarežģītākiem loģikas un arhitektūras aspektiem:

- iespējams uzdot jautājumus asistentam,
- iesaistīt to diskusijās kopā ar citiem kolēģiem.

Tehnoloģiju modelis

Integrācija ar GitLab, izmantojot GitLab API un Webhooks.

Alternatīvi iespējams izmantot integrāciju ar Git diff lokāli, vēl pirms koda nodošanas citam pārskatīšanai, tādējādi atklājot uzlabojumus vēl ātrāk.

Lietotārstāsts

"Es kā komandas vadītājs vēlos nodrošināt, ka viss kods, kas nonāk galvenajā zarā, ir augstas kvalitātes un atbilst mūsu standartiem, bet nevēlos tērēt komandas laiku, manuāli pārskatot visu. Vēlos arī samazināt risku, ka rutīnas pārbaudēs kādas kļūdas tiek nepamanītas, tāpēc man ir nepieciešams automatizēts risinājums."

Galvenās iezīmes

- Automātiska Merge Request pārskatīšana.

- Stila vadlīniju un drošības ievainojamību analīze.
- Integrācija ar GitLab un CI/CD procesiem.
- Lokāla darbība, nodrošinot datu drošību.
- Diskusiju atbalsts, ļaujot veidot sarunu gan ar kolēģiem, gan MI asistentu.

leguvumi

- Ietaupīts komandas laiks, automatizējot rutīnas pārbaudes.
- Konsekventa koda kvalitāte visā projektā.
- Ātrāka atgriezeniskā saite programmētājiem.
- Agrīna drošības risku un kļūdu atklāšana.

6. Projektu vadības administratīvs asistents (**Project Management Assistant**)

Apraksts

Virtuāls asistents Realizēts kā **standUp risinājums** (ikdienas jautājumi: *“ko darīji”, “ko darīsi”, “vai ir bloķējošas lietas”*) un darbu organizācijas platforma, kas palīdz projektu vadītājam organizēt komandas darbu. Tas spēj analizēt darbu statusu Jira, ģenerēt atskaites, prognozēt termiņu nobīdes, identificēt "pudeles kaklus" un sagatavot dienas kārtību sapulcēm.

Darbības modelis

Gatavojoties ikdienas "stand-up" sapulcei, projektu vadītājs palūdz asistentam sagatavot kopsavilkumu par katra komandas biedra progresu, kā arī izcelt uzdevumus, kas ir bloķēti vai kavējas. Tas ļauj vadīt sapulci mērķtiecīgi un koncentrēties uz būtiskāko.

Tehnoloģiju modelis

Realizētas integrācijas ar Jira, Confluence, MS Teams un citām platformām. Katru dienu, kolēģi tur var atjaunot saviem darbiem statusu un progresu, ko redz komandas vadītājs. Papildus realizējama integrācija ar iekšējo padarītā uzskaites sistēmu. Integrācija risinājumam ar MI rīkiem.

LLM analizē šos datus, meklē anomālijas un tendences, un ģenerē kopsavilkumus un atskaites dabīgā valodā. Var izmantot vizualizācijas bibliotēkas, lai attēlotu datus grafiski. Risinājums realizēts kā tīkla lietotne (web app), kas integrēta ar MI rāmi.

Lietotājstāsts

"Es kā projektu vadītājs, kurš pārvalda trīs projektus vienlaicīgi, vēlos katru rītu saņemt kodolīgu kopsavilkumu par katra projekta statusu, riskiem un komandas noslodzi, lai es varētu efektīvi plānot savu dienu."

Galvenās iezīmes

- Automātiski ģenerēti statusa pārskati par projektiem un komandām.
- Integrācija ar projektu vadības platformām (Jira, Confluence, MS Teams).
- Reāllaika anomāliju un risku detektēšana.
- Grafiski vizualizēti dati ar tendencēm un prognozēm.
- Web aplikācija, kas strādā drošā uzņēmuma vidē un ir integrēta ar MI rāmi.

leguvumi

- Ietaupīts projektu vadītāja laiks, automatizējot atskaišu veidošanu.

- Proaktīva risku identificēšana un pārvaldība.
- Datus balstīta lēmumu pieņemšana.
- Uzlabota projektu prognozējamība, plānošana un kontrole.

7. Tīmekļa lietotne sarunbotam (Chatbot)

Apraksts

Viegli lietojama **tīmekļa lietotne, kas darbiniekiem nodrošina piekļuvi valodu modeļiem caur čata saskarni**. Tā ļauj:

- veidot sarunas par dažādām tēmām (ne tikai tehniskām),
- augšupielādēt dokumentus analīzei un izgūt no tiem saturu,
- pārslēgties starp vairākiem MI modeļiem atbilstoši vajadzībām.

Šis rīks ir paredzēts plašam lietotāju lokam un neprasa iepriekšēju tehnisko pieredzi.

Īpaši nozīmīgi ir tas, ka iespējams izmantot valodu modeļus, kas optimizēti darbam latviešu un citās ES valodās, piemēram, EuroLLM projektā izstrādātos modeļus (<https://euollm.io/>).

Darbības modelis

Mārketinga nodaļas darbiniekam ir nepieciešams sagatavot preses relīzi latviešu un angļu valodā.

Izmantojot čata lietotni, viņš:

- augšupielādē produkta tehnisko specifikāciju,
- lūdz MI asistentam sagatavot relīzes melnrakstu, uzsverot galvenās priekšrocības,
- pārslēdzoties uz daudzvalodu modeli (piem., EuroLLM), lūdz adaptēt tekstu angļu auditorijai.

Rezultātā abas versijas ir gramatiski pareizas, stilistiski atbilstošas un pielāgotas mērķauditorijai.

Tehnoloģiju modelis

Risinājums ir bāzēts uz atvērtā pirmkoda platformas OpenWebUI, kas ir integrēta ar uzņēmuma izstrādāto MI rāmi. Tas nodrošina lietotāju pārvaldību, sarunu vēsturi un iespēju pieslēgt dažādus lielos valodu modeļus (LLM), tostarp specializētus modeļus, kas optimizēti konkrētām valodām vai uzdevumiem, piemēram, EuroLLM, kas nodrošina kvalitatīvu teksta ģenerāciju 24 oficiālajās ES valodās, ieskaitot latviešu valodu.

Risinājums bāzēts atvērtā pirmkoda platformas OpenWebUI un integrēts ar uzņēmuma izstrādāto MI integrācijas rami, un nodrošina:

- lietotāju pārvaldību,
- sarunu vēstures glabāšanu,
- pieslēgumu dažādiem lielajiem valodu modeļiem (LLM),
- iespēju izmantot specializētus modeļus (piem., EuroLLM, kas nodrošina kvalitatīvu teksta ģenerāciju visās 24 ES valodās).

Lietotājstāsts

"Es kā mārketinga speciālists vēlos efektīvi sagatavot preses relīzes un citus mārketinga materiālus vairākās valodās. Man ir nepieciešams rīks, kas spēj analizēt tehnisko dokumentāciju un automātiski ģenerēt saistošu saturu, kā arī adaptēt to dažādām mērķauditorijām un valodām, lai varētu ātrāk reaģēt uz tirgus pieprasījumu un ietaupīt laiku, kas parasti tiek veltīts tulkošanai un tekstu rediģēšanai."

Galvenās iezīmes

- Vienkārša un intuitīva čata saskarne visiem darbiniekiem.
- Dokumentu augšupielāde un analīze.
- Daudzvalodu atbalsts, ieskaitot latviešu valodu.
- Integrācija ar uzņēmuma MI rāmi, nodrošinot drošu un privātu lietošanu.
- Elastīga modeļu izvēle (specializēti LLM konkrētiem uzdevumiem vai valodām).

Ieguvumi

- Demokratizēta piekļuve MI tehnoloģijām visiem uzņēmuma darbiniekiem.
- Paātrināta satura (e-pastu, dokumentu, prezentāciju) radīšana.
- Uzlabota daudzvalodu komunikācija un dokumentu tulkošanas kvalitāte.
- Vienkārša un intuitīva lietotāja pieredze, kas neprasa apmācību.

8. Dinamiskās datu izpētes aģents (**Data Exploration Assistant**)

Apraksts

Virtuāls aģents, kas spēj pieslēgties dažādiem **datu avotiem** (datubāzēm, API, žurnālfailiem) un veikt **dinamisku datu izpēti un analīzi dabīgā valodā**.

Tas nodrošina:

- atbildes uz lietotāju jautājumiem par datiem,
- proaktīvu sistēmas uzraudzību,
- automatizētu apziņošanu par kritiskiem notikumiem.

Darbības modelis

Pēc jaunas programmatūras versijas piegādes (deployment) produkta īpašnieks vēlas pārliecināties, ka nav radušās problēmas.

Aģents, kas ir pieslēgts lietotnes monitoringa rīkiem (APM), proaktīvi pamana jaunas kļūdas un brīdinājumus.

Kad produkta īpašnieks uzdod jautājumu, aģents sagatavo strukturētu pārskatu, piemēram:

“Pēc pēdējās piegādes ir fiksēti 2 jauni brīdinājumu veidi, kas saistīti ar datu validāciju. Papildus tam reģistrēta viena zemas prioritātes kļūda, kas ietekmē profila datu mainīšanu. Iesakām izveidot uzdevumu nākamajam sprintam, lai novērstu brīdinājumus, pirms tie kļūst par nopietnākām problēmām.”

Tehnoloģiju modelis

RAG (Retrieval-Augmented Generation) arhitektūra, izmantojot Qdrant risinājumu un zināšanu bāzes.

Integrācija ar datu avotiem un monitoringa rīkiem, piemēram:

- GlitchTip, Sentry, Apache SkyWalking,
- Matomo Analytics, Uptime Kuma, Zabbix,
- audita žurnāli un citi datu avoti.

Aģents apkopo datus no šiem avotiem, apvieno tos ar LLM analīzi un sniedz saprotamus kopsavilkumus ar ieteikumiem.

Lietotājstāsts

*"Es kā produkta īpašnieks vēlos būt pārliecināts par katras piegādes kvalitāti. Nākamajā dienā pēc jaunas versijas izlaišanas, tā vietā, lai traucētu izstrādātājus ar lūgumu pārbaudīt žurnālfailus, es vienkārši pajautāju aģentam: 'Vai pēc vakardienas piegādes ir kādas problēmas?'
Es saņemu viegli saprotamu kopsavilkumu par sistēmas stāvokli un ieteikumus tālākai rīcībai. Tas man ļauj ātri novērtēt situāciju un pieņemt lēmumu, vai ir nepieciešama steidzama komandas iesaiste."*

Galvenās iezīmes

- Daudzkanālu datu integrācija no monitoringa un analītikas rīkiem.
- Reāllaika datu apstrāde un brīdinājumu ģenerēšana.
- Dabīgās valodas interfeiss datu vaicājumiem.
- Proaktīva problēmu identificēšana un ieteikumi to risināšanai.
- Automatizēta sadarbība ar komandām (piem., Jira uzdevumu ģenerēšana).

Ieguvumi

- Ātrāka atgriezeniskā saite par jaunās versijas kvalitāti.
- Proaktīva problēmu identificēšana, pirms tās ietekmē lielu lietotāju skaitu.
- Uzlabota datu balstīta komunikācija starp produktu un tehniskajām komandām.
- Samazināta manuālas žurnālfailu analīzes nepieciešamība.

9. Zināšanu bāzes papildināšanas aģents (**Dynamic Knowledge Base**)

Apraksts

Virtuālais asistents, kas integrējas ar **projektu pārvaldes platformām** (Jira, Trello), **e-pastu un saziņas kanāliem** (MS Teams, Slack), lai **automātiski identificētu un saglabātu svarīgu informāciju**, kas citādi varētu pazust sarunu plūsmā.

Šis risinājums nodrošina, ka būtiski lēmumi, uzdevumi un diskusiju rezultāti tiek strukturēti un ilgtermiņā saglabāti zināšanu bāzē.

Darbības modelis

Sprinta **apspriešanas** (Sprint Review) laikā komanda diskutē par paveikto un nākamajiem soļiem.

Asistents, kas ir pieslēgts sanāksmes atšifrējumam, automātiski identificē pieņemtos lēmumus un izveido ierakstu zināšanu bāzē.

Vēlāk šos datus iespējams izgūt, analizēt un izmantot projektu pārvaldībā vai atskaišu sagatavošanā.

Tehnoloģiju modelis

MI aģenti, kas monitorē definētus saziņas kanālus.

Funkcionalitāte ietver:

- sarunu analīzi,
- lēmumu, uzdevumu un problēmu identificēšanu,
- automatisku zināšanu bāzes papildināšanu.

Papildu iespējas:

- automatiska uzdevumu ģenerēšana Jira vai Trello,
- Confluence lapu papildināšana ar informāciju,

- ziņu publicēšana Teams vai Slack kanālos,
- sapulču transkriptu apstrāde (piemēram, Teams integrācija ar runas atpazīšanu) un to satura pievienošana zināšanu bāzei.

Lietotājstāsts

"Es kā projektu vadītājs vēlos būt drošs, ka visas svarīgās detaļas, kas tiek pieminētas Teams čatos vai citos datu avotos (e-pastos), tiek fiksētas un apkopotas zināšanu bāzē, lai varētu saņemt statusa atskaites, pārskatus un realizēt cita formāta datu izgūvi."

Galvenās iezīmes

- Integrācija ar saziņas kanāliem un projektu pārvaldības rīkiem.
- Automātiska lēmumu un uzdevumu identificēšana un saglabāšana.
- Zināšanu bāzes dinamiska papildināšana reāllaikā.
- Atbalsts transkripcijai no tiešsaistes sapulcēm.
- Elastīgas darbplūsmas integrācijas ar citām IT sistēmām.

Ieguvumi

- Samazināts informācijas zudums ikdienas komunikācijā.
- Vienmēr aktuāla un papildināta zināšanu bāze.
- Automatizēta administratīvo uzdevumu veikšana.
- Uzlabota komandas komunikācija un caurspīdīgums.

10. Kiberdrošības un ievainojamību aģents (Cybersecurity Agent)

Apraksts

Virtuāls aģents, kas proaktīvi **monitorē uzņēmuma IT infrastruktūru, pirmkodu un konfigurācijas**, lai identificētu potenciālās ievainojamības, sliktās prakses un drošības riskus.

Tas darbojas kā **pastāvīgs drošības slānis**, kas palīdz uzņēmumam savlaicīgi reaģēt uz draudiem un samazināt incidentu risku.

Darbības modelis

Šis aģents darbojas nepārtraukti kā daļa no CI/CD procesa. Katru reizi, kad tiek pievienots jauns kods vai bibliotēka, aģents to skenē, meklējot zināmas ievainojamības (CVEs). Ja tiek atrasts risks, tas automātiski izveido darba uzdevumu (piemēram Jira) un paziņo komandai.

Līdzīgi, spēj arī dot ieteikumus par kodu, īpaši daļām kas ir saistītas ar finansiāliem datiem, datu validācijām, autentifikāciju, autorizāciju, kā arī citiem svarīgiem programmatūras aspektiem.

Šis aģents darbojas nepārtraukti kā daļa no CI/CD procesa. Katru reizi, kad tiek pievienots jauns kods vai bibliotēka tas veic ievainojamību skenēšanu (CVEs), un ja risks tiek atrasts, automātiski tiek izveidots uzdevums (piemēram, Jira) un paziņots komandai.

Aģents sniedz arī ieteikumus par koda drošību, īpaši daļām, kas skar:

- finanšu datus,
- datu validāciju,
- autentifikāciju un autorizāciju,
- citus kritiski svarīgus aspektus.

Tehnoloģiju modelis

Integrācija ar pirmkoda pārvaldības sistēmām (piemēram GitLab) un nepārtrauktās integrācijas risinājumiem, piemēram GitLab CI. Nodrošina piekļuvi būvēto artefaktu skenēšanai (piemēram, skenēt Docker konteinerus ar Trivy rīka palīdzību), ievainojamību pārskatus un riska novērtējumus, ieteikumus remediācijai.

Papildus tam, iespējama arī tālāka integrācija ar ievainojamību datu bāzēm, lai konkrēti identificētu kuri sistēmas komponenti un bibliotēkas rada vislielākos riskus, lai varētu efektīvāk pieņemt lēmumus par tehnoloģiju izvēli un komponentu atjaunošanas grafikiem un nozīmi.

Lietotājstāsts

"Es kā IT drošības speciālists vēlos nodrošināt, ka mūsu sistēmas ir pasargātas no jaunākajiem apdraudējumiem, un saņemt brīdinājumus, ja tiek identificēta kāda kritiska ievainojamība."

Galvenās iezīmes

- CI/CD integrācija ar automatizētu drošības skenēšanu.
- Ievainojamību monitorings (CVEs, konteineri, bibliotēkas).
- Automātiska uzdevumu ģenerēšana (piem., Jira).
- Drošības prakses ieteikumi kodam un konfigurācijām.
- Integrācija ar drošības datu bāzēm efektīvākai risku prioritizācijai.

Ieguvumi

- Uzlabota sistēmu kiberdrošība.
- Proaktīva un nepārtraukta risku uzraudzība.
- Ātrāka reakcija uz drošības incidentiem.
- Automatizēta atbilstība drošības standartiem.

11. Darbinieka profila uzturēšanas asistents (Talent Assistant)

Apraksts

MI risinājums, kas ļauj projekta vadītājam, tiešajam vadītājam gada garumā veidot darbinieka profilu ar vienkāršiem pierakstiem (piem., darba uzdevumu izpildes piemēri, iniciatīvas, komunikācijas stils, sasniegumi). Ikgadējo pārrunu laikā asistents apkopo šos pierakstus, papildina tos ar datiem no projektu vadības sistēmām un sagatavo strukturētu atskaiti par darbinieka sniegumu.

Darbības modelis

Projekta vadītājs regulāri ievada īsus komentārus tieši sistēmā vai nosūta tos caur integrētu saziņas kanālu (MS Teams, e-pasts). MI automātiski sasaista pierakstus ar konkrēto darbinieku, kategorizē tos (piem., "sasniegumi", "jomas, kur uzlaboties") un analizē tendences. Pārrunu brīdī vadītājs saņem gatavu kopsavilkumu ar faktiem, piemēriem un ieteikumiem nākotnes attīstībai.

Tehnoloģiju modelis

Integrācija ar projektu vadības rīkiem (Jira, Trello, MS Teams), saziņas platformām un dokumentu glabātuvēm. MI algoritmi veic teksta apstrādi un klasifikāciju, nodrošinot, ka visi pieraksti ir viegli pārskatāmi un izmantojami.

Lietotājstāsts

"Es kā projekta vadītājs vēlos regulāri fiksēt darbinieka sasniegumus un jomas, kur nepieciešams uzlabojums, lai ikgadējo pārrunu laikā man būtu gatavs strukturēts pārskats par viņa sniegumu un es varētu nodrošināt objektīvu un uz attīstību vērstu atgriezenisko saiti."

Galvenās iezīmes

- Vienkārši pieraksti gada garumā, kas tiek automātiski strukturēti.
- Automātiska kategorizācija pēc tēmas un nozīmes.
- Integrācija ar esošajiem rīkiem, lai minimizētu papildu darbu vadītājiem.
- Tendencēm balstīta analīze, kas izceļ izaugsmi un problē zonas.
- Gatavs kopsavilkums pārrunām ar faktiem un piemēriem.

Ieguvumi

- Vienkāršota ikgadējo pārrunu sagatavošana.
- Datu balstīts novērtējums, kas balstās uz visa gada informāciju.
- Mazāks subjektivitātes risks.

12. Kvalitātes kontroles asistents ienākošās informācijas pārbaudei (Data Quality and Compliance Assistant)

Apraksts

Virtuāls MI risinājums, kas automātiski analizē ienākošo informāciju (piem., Jira pieteikumus, klientu e-pastus, formu ievades), lai noteiktu, vai tajā nav personas datu, sensitīvas informācijas vai neatbilstoša formāta datu. Šis process palīdz nodrošināt GDPR atbilstību un samazināt risku, ka sensitīva informācija nonāk neatbilstošā vidē.

Metodoloģiskais rāmis

Jaunā Jira uzdevuma izveides brīdī sistēma analizē ievades laukus un pielikumus, meklējot personas datus (piemēram, vārds, uzvārds, personas kods, e-pasts) gan aprakstos, gan komentāros, gan pievienotajos attēlos. Ja tādi tiek atrasti, pieteikumam tiek pievienota atzīme, ka tas satur personas datus un tiek informēta atbildīgā persona šo datu nepieciešamības izvērtēšanai un anonimizēšanai.

Tehnoloģiju modelis

Integrācija ar Jira API un dokumentu analīzes modeļiem, kas izmanto Named Entity Recognition (NER) un regulāro izteiksmju noteikšanu. Var darboties lokālajā infrastruktūrā, lai nodrošinātu datu drošību.

Lietotājstāsts

"Es kā kvalitātes vadītājs vēlos, lai visi jaunie Jira pieteikumi automātiski tiktu pārbaudīti uz personas datu un sensitīvas informācijas klātbūtni, lai mēs varētu ievērot GDPR prasības un mazināt riskus, ka sensitīvi dati tiek saglabāti neatbilstošā veidā."

Galvenās iezīmes

- Automātiska personas datu noteikšana (tekstā un pielikumos).
- GDPR atbilstības nodrošināšana.
- Integrācija ar Jira un citiem rīkiem.
- Reāllaika atzīmju un paziņojumu ģenerēšana.
- Lokāla infrastruktūras atbalsts datu drošībai

Ieguvumi

- Automatizēta atbilstības pārbaude GDPR prasībām.

- Mazāks cilvēkfaktora radīto kļūdu risks.
- Ātrāka un drošāka datu apstrāde.
- Samazināts juridisko un reputācijas risku apjoms.