Skaidrė 1 — Pradžia

[S1] Laba diena, gerbiama vertinimo komisija. Šiandien jums pristatysiu savo informatikos brandos darbą tema **„Saugumo ir kodo kokybės balansavimas programavimo praktikose"**.

Skaidrė 2 — Tikslas / Problema

[S2] Šiuolaikinio programuotojų pasaulio dilema – kaip išlaikyti tinkamą balansą tarp programinio kodo saugumo ir jo kokybės.
[p1] Tikslas: nustatyti ir įgyvendinti programavimo praktiką (gairių rinkinį), užtikrinant ne tik patvaraus, bet ir patikimo bei ilgaamžio programinio kodo kūrimą.
[p2] Problema aktuali: saugumo priemonės apsunkina kodo struktūrą, pablogina įskaitomumą ir priežiūrą. Be to, dažniau DI, pavyzdžiui vadinamas **„Vibe Coding"** kelia papildomų nežinomųjų sistemos vientisumui.

Skaidrė 3 — Esminiai principai

[S3] Išskirti esminiai principai problemai suvaldyti:
[p1] Simbiozė: būtina derinti kalibruotą automatizuotą analizę su atidžia ir koncentruota žmogaus peržiūra.
[p2] Matavimas: vertinant kelis aspektus sudėtingoje sistemoje, vieno rodiklio niekada nepakanka.
[p3] Sisteminimas: darbus privalu atlikti sistemiškai, greitai, pasitelkiant aiškias ir pakartojamas gaires.
[p4] Struktūra: norint sumažinti kognityvinį nuovargį skaitant kodą, privaloma skatinti išskaidymą į modulius.

Skaidrė 4 — DevSecOps filosofija

[S4] Principai remiasi moderniu požiūriu – **DevSecOps filosofija**. Saugumas (Sec) nėra paliekamas tik sistemos testavimo ar išleidimo etapui. Jis tampa integrali dalis – pradedant planavimu, baigiant kodo rašymu, testavimu bei sistemos stebėjimu.

Skaidrė 5 — Atviro kodo įrankis CQaS

[S5] Sukurtas įrankis **Code Quality and Security („CQaS")**.
[p1] Naudota Python kalba, apimant **virš 7000 kodo eilučių**.
[p2] Funkcija: padėti prižiūrėti kodo sudėtingumą, įskaitomumą, PEP8 standarto atitiktį, apskaičiuoti apytikslę **techninę skolą** (struktūrinis sudėtingumas ir būsimų modifikacijų „sunkumas") bei dubliavimąsi.
[p3] LT/EN palaikymas. **[p4]** v2.0 įdiegtas kodo peržiūros režimas, kuris interaktyviai paprašo programuotojo atkreipti dėmesį į įtartinas kodo vietas.

Skaidrė 6 — Eksperimento eiga

[S6] Iteracinė daugiavariatinė analizė. Sukurtos **13 versijų, apimančios beveik 13000 eilučių**.
[1: Bazinis] v1.0 – nesaugus ir nekokybiškas žmogaus (mano) kodas.
[2: Tobulinimas] kodo išskaidymas, lygintas žmogaus darbas su DI.
[3: Sąrašai] kontroliniai sąrašai (pilni/susiaurinti) - analizuojami skirtumai DI/žmogaus.
[4: Automatinė analizė] statinio ir dinaminio kodo analizės įrankių lyginimas.
[5: Kodo peržiūra] išsami peržiūra ir taisymas (Žmogus vs DI).
[6: Dokumentacija] DI komentarų ir dokumentacijos naudos vertinimas.

Skaidrė 7 — DI (ChatGPT-5.2-Thinking) įvertinimas

[S7] Rezultatai buvo gana kategoriški – poveikis dažniausiai neigiamas nepaisant **samprotavimo gabėjimų**.
[p1] DI nesugeba įvertinti pilno ir gilaus projekto konteksto.
[p2] Pavedus DI atlikti išskaidymą, sistema tapo **fragmentuota**, atsirado nesusipratimų tarp modulių.
[p3] Integracija į bendrą sistemą reikalavo itin kruopščios ir daug laiko atimančios žmogaus peržiūros.
[p4] Išvada: DI įrankiai kol kas toli gražu negali pakeisti sistemingo žmogaus atliekamo darbo.

Skaidrė 8 — Kiekybinis vertinimas ir Trimean

[S8] **[p1]** Rezultatai sunormalizuoti min-max metodu.
[p2] Pasirinktas **triskaitis vidurkis (angl. Trimean)**.

> *„Noriu pabrėžti vieną įdomų faktą apie šį terminą: ieškodama, kaip šią sąvoką taisyklingai pavadinti lietuviškai, radau tik 1 vienintelį šaltinį... konsultavausi su VLKK bei Vikipedijos bendruomene... taip gimė mano inicijuotas Vikipedijos straipsnis."*

TV idealiai subalansuoja rodiklius ir yra atsparus ekstremalioms reikšmėms, kas yra svarbu mažos-vidutinės apimties darbuose.

Skaidrė 9 — Rezultatų matrica

[S9] Eilutės – rodikliai, stulpeliai – versijos. Raudoni skaičiai – versijos reitingas. Antraštės dešinėje – matuoti rodikliai. Tamsiau = prastesnis, šviesiau = geresnis.
Efektyviausios praktikos: žmogaus vadovaujama kodo peržiūra (5.1), žmogaus atliktas kodo išskaidymas (2.1) ir statinis saugumo testavimas (4.1).
Prasčiausiai pasirodė: bazinė versija ir beveik visos versijos, kurioms vadovavo dirbtinis intelektas. Paskutinis stulpelis įrodo brandos darbo sėkmingumą.
Paskutinis stulpelis: ne eksperimento dalis, o galutinis rezultatas

Skaidrė 10 — Galutinė 7-oji iteracija

[S10] Sukurta galutinė iteracija integravus hibridinę praktiką į įrankį.
[p0] 1-oji vieta rezultatų matricoje.
[p1-2] Saugumo klaidų skaičius sumažėjo net **73%**.
[p3] Techninė kodo skola sumažėjo milžinišku skirtumu – **virš 92%**.
[p4] Pasiektas visiškas 100% duomenų tipų užtikrinimas be konfliktų (skaičius negali būti sumaišytas su masyvu).
[p5] Bendras rodiklių vidutinis patobulėjimas siekė beveik 80%.

Skaidrė 11 — Rekomendacijos (Konstatavimas)

[S11] Konstatuojama sudaryta praktika bendruomenei:
1. Išskaidymas: visuomet skaidykite kodą į nepriklausomus modulius (pamatas).
2. Sąrašai: aktyviai naudokite kontrolinius sąrašus sisteminiam mąstymui.
3. Peržiūra: žmogaus vadovaujama kodo peržiūra yra viena efektyviausių praktikų.
4. DI vengimas: siūlau vengti aklai pasikliauti DI išvestimi (architektūra/kriptografija).
5. SPST: naudokite statinės analizės priemones (CQaS, Bandit) klaidoms sulaikyti kuo anksčiau.

Skaidrė 12 — Darbo pabaiga / Išvados

[S12] Tikslas pasiektas:
[p1] Eksperimentiniu būdu įrodyta ir nustatyta veiksminga praktika, kuri realiomis sąlygomis suderina saugumą ir kokybę į vieną vienetą.
[p2] Šios praktikos ilgalaikiam ir automatizuotam palaikymui sėkmingai sukurtas atviro kodo įrankis.

Skaidrė 13 **Padėkos**

[S13] Didžiulė padėka:
[p1] Darbo vadovui (-ei) ...
[p2] Esu dėkinga konsultantui (-ei) ...

Skaidrė 14 **Klausimai**

[S14] Ačiū už dėmesį ir skirtą laiką. Dabar mielai atsakysiu į visus jūsų klausimus.