1. **ivardinkite 2 esminius skirtumus tarp nuoseklaus (waterfall) ir Vmodelio gyvavimo ciklo modeliu (GCM)**

Nuoseklaus (waterfall) ir Vmodelio gyvavimo ciklo modeliai (GCM) yra du skirtingi programų kūrimo metodai, kurie naudojami sistemų kūrimo procesuose. Štai 2 esminiai skirtumai tarp jų:

1. **Proceso struktūra ir iteratyvumas:**
   * **Nuoseklaus modelis:** Nuoseklaus modelis yra linijinis procesas, kuriame kiekviena fazė vyksta po kito. Tai reiškia, kad kiekviena fazė (reikalavimų nustatymas, dizainas, kūrimas, testavimas, palaikymas) yra atliekama po to, kai prieš tai buvo baigta ankstesnė. Šis modelis yra labai struktūruotas ir griežtas.
   * **Vmodelio gyvavimo ciklo modelis:** GCM yra iteratyvus modelis, kuris leidžia atlikti tam tikras veiklas lygiagrečiai, o ne sekant linijiniu būdu. Vmodelis susideda iš linijinės plėtros, kurią atstato testavimo proceso iteracijos. Tai leidžia ankstyvąjį testavimą ir greitą grįžtamąjį ryšį.
2. **Testavimo laikas ir procesas:**
   * **Nuoseklaus modelis:** Testavimas dažniausiai atliekamas tik po to, kai visos kitos fazės yra baigtos. Tai reiškia, kad defektai gali būti aptikti vėlai proceso metu, o jų taisymas gali būti brangus ir laiko reikalaujantis.
   * **Vmodelio gyvavimo ciklo modelis:** GCM leidžia ankstyvąjį testavimą, nes testavimas vyksta nuosekliai kartu su kiekviena kūrimo fazė. Tai leidžia greičiau aptikti ir taisyti klaidas, mažinant vėlesnių pataisymų išlaidas.

Šie skirtumai parodo, kad nors abu modeliai turi savo privalumų ir trūkumų, jų naudojimas priklauso nuo projekto specifikos ir poreikių. Nuoseklaus modelis labiau tinka aiškiems ir stabiliam reikalavimų rinkiniui, tuo tarpu Vmodelio gyvavimo ciklo modelis gali būti naudingas sudėtingesnėms ir dinamiškesnėms sistemoms.

1. **iteratyvaus GCM trukumai**

Iteratyvūs gyvavimo ciklo modeliai (GCM), tokie kaip Vmodelis arba spiralės modelis, turi savo privalumų, tačiau taip pat turi ir trūkumų. Štai keletas trūkumų, susijusių su iteratyviais GCM:

1. **Sudėtingumas ir valdymas:** Iteratyvūs modeliai gali būti sudėtingesni nei linijiniai modeliai, pavyzdžiui, nuoseklusis modelis. Valdymas gali būti sunkesnis dėl daugybės iteracijų ir bendro proceso sudėtingumo.
2. **Didesnės sąnaudos:** Dėl daugybės iteracijų, kurių kiekviena atliekama su visais kūrimo etapais (reikalavimų rinkimas, dizainas, kūrimas, testavimas), gali padidėti kūrimo išlaidos.
3. **Laiko ir resursų trūkumas:** Iteratyvūs modeliai gali pareikalauti daugiau laiko ir išteklių, nes kiekvienas ciklas prasideda nuo pradžios ir apima visus etapus iki testavimo ir palaikymo.
4. **Rizika dėl reikalavimų kintamumo:** Dėl projektų iteratyvumo ir ilgesnio laiko, kai reikalavimai gali kisti, gali padidėti rizika, kad išsivysčiusi sistema gali neatitikti pradinių reikalavimų.
5. **Geras komandos bendradarbiavimas:** Siekiant sėkmingai įgyvendinti iteratyvius modelius, reikia gerai suderintos ir bendradarbiaujančios komandos. Jei komanda netinkamai bendradarbiauja arba nesugeba veikti sinchroniškai, tai gali sukelti vėlavimus ir problemas projekte.
6. **Testavimo ir kokybės užtikrinimas:** Dėl kiekvienos iteracijos testavimo reikalingas didesnis dėmesys kokybei ir testavimo strategijoms. Nepakankamas testavimas gali lemti klaidų persistavimą ateities iteracijose arba netgi produkto galutinėje versijoje.

Šie trūkumai ne visada yra taikomi visiems projektams, tačiau svarbu juos atsižvelgti planuojant ir įgyvendinant iteratyvius gyvavimo ciklus. Teisingas modelio pasirinkimas priklauso nuo projekto specifikos, kompleksiškumo ir kitų kontekstinių veiksnių.

**nuoseklaus (waterfall)** **GCM privalumai**

Nuoseklus (waterfall) modelis ir iteratyvūs gyvavimo ciklo modeliai (GCM), tokiu kaip Vmodelis, turi savo privalumų ir trūkumų. Štai keletas privalumų, susijusių su nuosekliuoju modeliu:

1. **Aiškumas ir paprastumas:** Nuoseklusis modelis yra labai aiškus ir paprastas suprasti, nes jis seka tiesioginį linijinį procesą nuo pradžios iki pabaigos. Tai gali padėti užtikrinti aiškumą ir susitarimą tarp kliento ir komandos.
2. **Griežtas kontroliavimas:** Dėl savo linijinio pobūdžio, nuoseklusis modelis leidžia griežtai kontroliuoti kiekvieną etapą ir užtikrinti, kad kiekvienas etapas būtų baigtas prieš einant prie kito. Tai gali padėti valdyti išteklius ir laiką.
3. **Aiški atsakomybė:** Kiekvienas etapas turi aiškiai apibrėžtus tikslus ir rezultatus, todėl yra aiški atsakomybė, kas atsakingas už kiekvieną etapą. Tai padeda išvengti neaiškumų ir konfliktų dėl atsakomybės.
4. **Tinkamas veiksmų sekimas:** Nuoseklusis modelis yra tinkamas projektams, kuriuose reikalavimai yra stabilūs ir gerai suprantami. Tai yra tinkamas pasirinkimas projektams su aukštu aiškumu ir maža rizika, kurių reikalavimai nesikeičia.
5. **Gerai tinka mažiems projektams:** Nuoseklusis modelis gali būti labai veiksmingas ir tinkamas mažiems ir trumpiems projektams, kai reikalavimai yra aiškūs ir nekintantys.

Nors nuoseklusis modelis turi savo privalumų, svarbu suprasti, kad jis gali būti mažiau tinkamas sudėtingesniems arba kintantiems reikalavimams turintiems projektams. Taip pat gali kilti sunkumų, jei reikia atlikti pakeitimus arba tobulinti sistemą po to, kai ji jau yra pilnai sukūriama. Taigi, modelio pasirinkimas priklauso nuo projekto specifikos ir poreikių.

3.**aprasykite sistemos architekturos apibrezties (architectural definition) proceso dedamasias pagal pasirinkta proceso specifikavimo sablona**

**Sistemos architektūros apibrėžties (Architecture Definition) procesas yra** **esminis kūrimo etapas, skirtas kurti ir pasirinkti architektūros alternatyvas, kurios atitinka suinteresuotųjų asmenų lūkesčius ir sistemos reikalavimus. Pagal procesų specifikavimo šabloną, proceso dedamosios būtų tokios:**

**1.** **Architektūros alternatyvų generavimas**

**Tikslas: Kurti įvairias architektūros alternatyvas, kurios gali patenkinti sistemos reikalavimus ir suinteresuotųjų asmenų poreikius.**

**Veiklos:**

* **Konteksto analizė: Nustatyti, kokie yra pagrindiniai reikalavimai, aplinka ir apribojimai, kurie turės įtakos architektūrai.**
* **Alternatyvų kūrimas: Generuoti įvairias architektūros koncepcijas, kurios gali patenkinti nustatytus reikalavimus. Tai gali apimti įvairias struktūras, komponentų išdėstymą ir sąsajas.**

**Rezultatas: Alternatyvių architektūrų sąrašas, kurį galima analizuoti ir palyginti.**

**2.** **Alternatyvų pasirinkimas**

**Tikslas: Pasirinkti vieną ar kelias architektūros alternatyvas, kurios geriausiai atitinka suinteresuotųjų asmenų rūpesčius ir sistemos reikalavimus.**

**Veiklos:**

* **Alternatyvų vertinimas: Analizuoti kiekvienos alternatyvos privalumus, trūkumus, rizikas ir sąnaudas.**
* **Kriterijų taikymas: Naudoti vertinimo kriterijus, kad būtų galima pasirinkti tinkamiausią alternatyvą. Kriterijai gali apimti našumą, saugumą, išlaidas, sudėtingumą ir kitus svarbius faktorius.**
* **Sprendimų priėmimas: Pasirinkti vieną ar kelias alternatyvas, remiantis vertinimu ir interesų konfliktų sprendimu.**

**Rezultatas: Pasirinktos architektūros alternatyvos, kurios bus toliau išplėtotos.**

**3.** **Nuoseklių vaizdų išraiška**

**Tikslas: Pateikti pasirinktos architektūros aprašymą įvairiais nuosekliais vaizdais, kurie padės aiškiai suprasti ir komunikuoti architektūrą.**

**Veiklos:**

* **Vaizdų kūrimas: Sukurti architektūros modelius ir diagramas, tokias kaip komponentų diagramos, sąveikos diagramos, ir sistemų kontekstai.**
* **Dokumentacija: Parengti išsamią architektūros dokumentaciją, kuri apima visus pasirinktos architektūros aspektus ir detales.**
* **Komunikacija: Užtikrinti, kad architektūros vaizdai ir dokumentai būtų aiškūs ir prieinami visiems suinteresuotiesiems asmenims.**

**Rezultatas: Kompletus architektūros vaizdų ir dokumentacijos rinkinys, kuris aiškiai apibūdina pasirinktą architektūrą ir jos komponentus.**

**1. Proceso tikslas (Purpose)**

* **Aprašymas: Sistemos architektūros apibrėžties proceso tikslas yra:**
  1. **Sukurti alternatyvias sistemos architektūras: Generuoti įvairias architektūros alternatyvas, kurios atitinka sistemos reikalavimus.**
  2. **Pasirinkti tinkamiausią alternatyvą(-as): Atrinkti vieną arba kelias alternatyvas, kurios geriausiai atitinka suinteresuotų šalių poreikius ir sistemas reikalavimus.**
  3. **Išreikšti architektūrą per nuoseklias perspektyvas: Dokumentuoti pasirinktą architektūrą per įvairias suderintas perspektyvas (pvz., funkcines, struktūrines, informacines ir kt.).**

**2. Įėjimai (Inputs)**

* **Reikalavimų specifikacija: Išsamus sistemos reikalavimų dokumentas, apimantis tiek funkcinius, tiek nefunkcinius reikalavimus.**
* **Suinteresuotų šalių analizė: Suinteresuotų šalių poreikiai ir lūkesčiai, kurie turi būti įgyvendinti per architektūrą.**
* **Esamos sistemos analizė: Informacija apie esamą sistemą arba sistemų komponentus, kurie gali būti naudojami kaip pagrindas naujai architektūrai kurti.**
* **Technologinės apribos: Technologiniai apribojimai ir reikalavimai, kurie gali daryti įtaką architektūros sprendimams.**

**3. Veiklos (Activities)**

* **Architektūros alternatyvų kūrimas:**
  + **Generuojamos įvairios sistemos architektūros alternatyvos, kurios atitinka nustatytus reikalavimus ir apribojimus.**
* **Architektūros alternatyvų analizė ir vertinimas:**
  + **Įvertinamos sukurtos alternatyvos, atsižvelgiant į suinteresuotų šalių poreikius, techninius reikalavimus, rizikas ir kitus svarbius kriterijus.**
* **Architektūros alternatyvų pasirinkimas:**
  + **Pasirenkama viena arba kelios alternatyvos, kurios geriausiai atitinka visus nustatytus kriterijus.**
* **Architektūros apibrėžimas ir dokumentavimas:**
  + **Parengiamas architektūros aprašas, kuris apima visus pasirinktos architektūros aspektus ir pateikia juos per įvairias perspektyvas (pvz., komponentų schema, duomenų srautai, sąsajos ir pan.).**

**4. Išėjimai (Outputs)**

* **Architektūros aprašas: Dokumentas, kuriame išsamiai aprašoma pasirinkta sistemos architektūra per suderintas ir nuoseklias perspektyvas.**
* **Architektūros sprendimų protokolai: Dokumentai, kurie fiksuoja atliktus sprendimus, įskaitant alternatyvų analizės ir pasirinkimo pagrindimą.**
* **Suderinamumo analizė: Įvertinimas, ar pasirinkta architektūra atitinka visus sistemos reikalavimus ir suinteresuotų šalių poreikius.**

**5. Įrankiai ir metodai (Tools and Methods)**

* **Modeliavimo įrankiai: Programinė įranga, skirta architektūros modeliams kurti (pvz., UML, SysML).**
* **Analizės metodai: Alternatyvų vertinimo metodai, pvz., kokybės atributų scenarijų metodai arba daugialypės kriterijų analizė (MCDM).**
* **Dokumentavimo įrankiai: Priemonės, leidžiančios sistemingai dokumentuoti architektūrą ir su ja susijusius sprendimus.**

**6. Vaidmenys ir atsakomybės (Roles and Responsibilities)**

* **Architektas: Atsakingas už architektūros alternatyvų kūrimą, jų vertinimą ir galutinės architektūros apibrėžimą.**
* **Reikalavimų analitikas: Renkasi reikalavimus ir bendradarbiauja su architektu siekiant užtikrinti, kad architektūra atitiktų visus reikalavimus.**
* **Suinteresuotų šalių atstovai: Dalyvauja architektūros alternatyvų vertinimo procese, siekiant užtikrinti, kad jų poreikiai būtų atspindėti galutiniame sprendime.**

**7. Kontrolė ir kokybės užtikrinimas (Control and Quality Assurance)**

* **Architektūros peržiūros: Reguliarios peržiūros, skirtos patikrinti, ar architektūra atitinka reikalavimus ir suinteresuotų šalių poreikius.**
* **Kokybės tikrinimas: Testavimo ir vertinimo veiklos, skirtos užtikrinti, kad architektūra būtų patikima, saugi ir atitiktų nustatytus standartus.**

**8. Sėkmės kriterijai (Success Criteria)**

* **Atitikimas reikalavimams: Pasirinkta architektūra pilnai atitinka visus funkcinius ir nefunkcinius reikalavimus.**
* **Suinteresuotų šalių patvirtinimas: Architektūros sprendimas patvirtintas visų svarbiausių suinteresuotų šalių.**
* **Architektūros dokumentacijos kokybė: Parengta dokumentacija yra aiški, nuosekli ir išsami, leidžianti lengvai įgyvendinti architektūrą vėlesniuose sistemos kūrimo etapuose.**

**9. Rizikos ir rizikos valdymas (Risks and Risk Management)**

* **Technologinė rizika: Rizika, kad pasirinkta architektūra gali būti nesuderinama su būsima technologine plėtra.**
* **Reikalavimų pokyčiai: Rizika, kad pasikeitus reikalavimams, architektūra gali tapti nebetinkama.**
* **Integracijos rizika: Rizika, susijusi su pasirinktos architektūros integracija į esamas sistemas ar infrastruktūrą.**

**pateikite pavyzdziu (min 3) kas pagal CMMI-DEV 2.0 rekomenduoja transformuoti suinteresuotu asmenu (stakeholders) reikalavimus i vartotojo (customer) ir i produkto reikalavimus**

**A diagram of a software requirements

Description automatically generated**

**Sugrupuoti reikalavimus pagal tam tikra tematika. Isrinkti pasikartojancius bandyti perziureti reikalavimus ir suras tar nera loginiu klaidu ar jie vienam kitam nepriestarauja. Neaiskius reikalavimus patikslinti su suinteresuotais asmenim reikavimam parasyti user stories ir padengti priemimo kriterijais**

**** **Reikalavimų rinkimas ir analizė (Requirements Development and Analysis):**

* **Šioje veikloje renkamos visų suinteresuotų šalių reikalavimų specifikacijos. Pavyzdžiui, jei klientas reikalauja didelio saugumo lygio (pvz., „Sistema turi užtikrinti duomenų konfidencialumą“), šis reikalavimas būtų detalizuotas kaip konkrečios saugumo funkcijos ar mechanizmai produkto reikalavimuose.**

**** **Reikalavimų suskaidymas ir detalizavimas (Requirements Elaboration and Decomposition):**

* **Transformuojant suinteresuotų šalių reikalavimus į specifinius produkto reikalavimus, kiekvienas aukšto lygio reikalavimas yra suskaidomas į mažesnius, labiau valdomus komponentus. Pvz., „Sistema turi būti vartotojui patogi“ gali būti suskaidytas į reikalavimus dėl vartotojo sąsajos dizaino, reakcijos laiko ir prieinamumo.**

**** **Reikalavimų suderinamumo patikrinimas (Requirements Validation and Verification):**

* **Užtikrinama, kad suinteresuotų šalių reikalavimai būtų teisingai transformuoti į vartotojo ir produkto reikalavimus. Tai gali būti atlikta naudojant prototipus ar kitus modeliavimo metodus, siekiant patikrinti, ar produkto reikalavimai atitinka pradinį suinteresuotų šalių poreikį. Pvz., „Sistema turi būti suderinama su esamomis klientų sistemomis“ yra verčiama į specifikacijas, susijusias su suderinamumu su konkrečiomis sistemomis, naudojant testavimo scenarijus.**

**** **Suinteresuotųjų šalių poreikių surinkimas:**

* **Šio proceso metu identifikuojami ir dokumentuojami suinteresuotųjų šalių poreikiai ir lūkesčiai. Tai apima visų suinteresuotųjų šalių reikalavimų rinkimą, įskaitant galutinius vartotojus, klientus, reguliavimo institucijas ir vidinius organizacijos padalinius.**

**** **Reikalavimų analizė:**

* **Surinkti reikalavimai yra analizuojami siekiant suprasti jų reikšmę, svarbą ir poveikį produktui. Šiuo etapu reikalavimai yra peržiūrimi ir, jei reikia, patikslinami.**

**** **Reikalavimų transformavimas:**

* **Suinteresuotųjų šalių reikalavimai transformuojami į išsamius vartotojo ir produkto reikalavimus. Tai apima reikalavimų vertimą į specifikacijas, kurias gali suprasti ir įgyvendinti kūrimo komanda.**
* **Vartotojo reikalavimai apima bendruosius tikslus ir funkcionalumą, kurio tikisi klientas, o produkto reikalavimai yra techninės specifikacijos, kurias turi atitikti galutinis produktas.**

**** **Reikalavimų validacija:**

* **Užtikrinama, kad reikalavimai atitinka suinteresuotųjų šalių poreikius ir lūkesčius. Tai atliekama per reikalavimų peržiūras, prototipus ar kitus validacijos metodus.**

CMMI-DEV 2.0 (Capability Maturity Model Integration for Development) rekomenduoja transformuoti suinteresuotų asmenų (stakeholders) reikalavimus į vartotojo (customer) ir produkto reikalavimus siekiant užtikrinti, kad sistema ar produktas atitiktų visų suinteresuotų šalių poreikius ir tikslus. Štai trys pavyzdžiai, kaip tai galėtų atrodyti:

1. **Suinteresuotų asmenų reikalavimai į vartotojo reikalavimus:**
   * **Suinteresuotas asmuo:** Finansų departamento vadovas
   * **Reikalavimas:** Sistema turi turėti galimybę generuoti finansinius ataskaitas ir analizės duomenis.
   * **Vartotojo reikalavimas:** Vartotojas nori turėti funkcionalumą, leidžiantį generuoti finansinius ataskaitas pagal nustatytus kriterijus, pavyzdžiui, pelno ir nuostolių analizę, balanso ataskaitas ir kt.
2. **Suinteresuotų asmenų reikalavimai į produkto reikalavimus:**
   * **Suinteresuotas asmuo:** Marketingo vadovė
   * **Reikalavimas:** Sistema turi turėti įgaliotus vartotojus pritaikyti ir redaguoti marketingo kampanijas.
   * **Produkto reikalavimas:** Produktas turi turėti vartotojo sąsają, leidžiančią marketingo personalui lengvai sukurti, tvarkyti ir redaguoti reklaminius turinius ir kampanijas, įskaitant reklaminių pranešimų, skelbimų ir socialinių tinklų pranešimų kūrimą ir platinimą.
3. **Suinteresuotų asmenų reikalavimai į vartotojo ir produkto reikalavimus:**
   * **Suinteresuotas asmuo:** Techninis palaikymo specialistas
   * **Reikalavimas:** Sistema turi turėti lengvą diegimą ir konfigūraciją bei būti palaikoma minimaliu techniniu personalu.
   * **Vartotojo ir produkto reikalavimas:** Produktas turi turėti aiškų ir lengvai naudojamą diegimo procesą, leidžiantį techniniam personalui lengvai diegti ir konfigūruoti sistemą be didelių papildomų išteklių. Be to, sistema turi turėti aiškų dokumentacijos ir palaikymo vadovą, leidžiantį vartotojams greitai rasti reikiamą informaciją ir sprendimų techniniams klausimams.
4. **kodel specifikuojant reikalavimus reikia aprasyti priemimo/tikimo kriterijus. Nurodykite bent 3 priezastis**

**Paaiškinti tam, kad:**

* **Užtikrinti, jog visi vienodai supranta problemą.**
* **Padėti komandos nariams suprasti, kada istorija (užduotis) yra užbaigta.**
* **Padėti patikrinti istoriją naudojant automatizuotus testus**

Specifikuojant reikalavimus yra svarbu aprašyti priėmimo/tikimo kriterijus, nes tai padeda užtikrinti aiškų supratimą tarp suinteresuotų šalių ir suteikia pagrindą kokybės kontrolės procesui. Štai trys priežastys, kodėl tai yra svarbu:

1. **Aiškumas ir suderinamumas:** Priėmimo/tikimo kriterijai nurodo konkretų kokybės lygį arba veikimo standartą, kurio turi būti laikomasi, kad reikalavimai būtų laikomi patenkinamais. Tai užtikrina aiškų supratimą tarp suinteresuotų šalių ir leidžia joms tiksliai nustatyti, kada sistema ar produktas yra pasiruošęs priimti ir įgyvendinti.
2. **Kokybės kontrolė ir vertinimas:** Priėmimo/tikimo kriterijai suteikia pagrindą kokybės kontrolės procesui, nes nustato, kaip bus įvertinama sistema ar produktas prieš patenkinant jį. Tai leidžia suinteresuotoms šalims objektyviai įvertinti, ar sistema atitinka jų lūkesčius ir reikalavimus, ir reikia ar ne tolesnės veiklos.
3. **Rizikos mažinimas ir problemų sprendimas:** Priėmimo/tikimo kriterijai padeda sumažinti riziką ir išvengti problemų ateityje, nes nustato, kokio kokybės lygio ar veikimo standarto turėtų laikytis sistema ar produktas. Tai suteikia galimybę anksti nustatyti bet kokias neatitikimus arba problemas ir imtis veiksmų jas išspręsti, prieš paleidžiant sistemą ar produktą į naudojimą.
4.  **Projekto sėkmės užtikrinimas**:
5. Aiškiai aprašyti priėmimo kriterijai padeda užtikrinti, kad projektas būtų sėkmingas, nes tiksliai apibrėžia, ką reiškia „sėkmingai įvykdytas reikalavimas“. Tai leidžia efektyviai valdyti projekto eigą ir sumažinti riziką, kad bus nesusipratimų ar klaidų įgyvendinant reikalavimus.
6.  **Sutartiniai įsipareigojimai**:
7. Jei projektas vykdomas pagal sutartį, priėmimo kriterijai dažnai tampa pagrindu įvertinti, ar darbai atlikti pagal sutartį. Tai gali turėti įtakos mokėjimams, projekto galutiniam priėmimui ir ginčų sprendimui.
8.  **Tinkamumas naudojimui**:
9. Priėmimo kriterijai taip pat padeda įvertinti, ar galutinis produktas yra tinkamas naudoti praktikoje. Jie užtikrina, kad produktas atitiktų ne tik techninius reikalavimus, bet ir vartotojų lūkesčius bei poreikius.

Šios priežastys pabrėžia, kodėl specifikuojant reikalavimus yra svarbu įtraukti aiškų ir konkretų priėmimo/tikimo kriterijų. Tai užtikrina, kad visos suinteresuotos šalys turi bendrą supratimą apie tai, kada sistema ar produktas yra pasiruošęs būti priimtas ir naudojamas.

**6** **ivardinkite pagrindines veiklas CMII-DEV techninio sprendimo(technical solution) procese**

**is to select, design, and implement solutions to requirements.**

**** **Produktų komponentų sprendimų pasirinkimas (TS.SG 1 Select Product Component Solutions):**

* **Pasirenkami produktų arba jų komponentų sprendimai iš galimų alternatyvų. Šioje veikloje yra atliekama įvairių sprendimų analizė, įvertinamos alternatyvos, atsižvelgiant į reikalavimus, rizikas ir kitus veiksnius, kad būtų pasirinktas tinkamiausias sprendimas.**

** Dizaino kūrimas (TS.SG 2 Develop the Design):**

* **Kuriami produktų arba jų komponentų dizainai. Šioje veikloje yra detalizuojami pasirinkti sprendimai, kuriant išsamų dizainą, kuris atitinka visus nustatytus reikalavimus. Dizainas taip pat gali apimti techninių specifikacijų, architektūros ir kitų su produktu susijusių aspektų apibrėžimą.**

** Produkto dizaino įgyvendinimas (TS.SG 3 Implement the Product Design):**

* **Įgyvendinami produktų komponentai ir susijusi palaikymo dokumentacija pagal sukurtą dizainą. Šioje veikloje yra įgyvendinamas fizinis arba programinis produktas, kuris atitinka sukurto dizaino specifikacijas, įskaitant ir visą reikalingą dokumentaciją, kad produktas galėtų būti naudojamas, palaikomas ir toliau vystomas.**

**7-** **Pateikite bent 5 kriterijas architekturos alternatyvu svarstymui**

Architektūros alternatyvų svarstymas yra svarbus proceso etapas, siekiant pasirinkti tinkamiausią architektūros sprendimą organizacijos poreikiams. Čia pateikiu penkis kriterijus, kuriuos galima naudoti architektūros alternatyvų svarstymui:

1. **Skalė:** Vertinkite, kaip kiekviena architektūros alternatyva gali prisitaikyti prie organizacijos augimo ir plėtros poreikių. Svarstykite, ar architektūros sprendimas gali būti lengvai plečiamas ir prisitaikytas, kad atitiktų besikeičiančius poreikius ir mastus.
2. **Efektyvumas:** Įvertinkite, kokiais būdais kiekviena alternatyva gali užtikrinti sistemos ar produkto efektyvumą. Svarstykite, ar architektūros sprendimas leis efektyviai naudoti resursus, optimizuoti veiklą ir sumažinti išlaidas.
3. **Lankstumas:** Įvertinkite, kokiais būdais kiekviena alternatyva gali suteikti lankstumo organizacijos veiklai. Svarstykite, ar architektūros sprendimas leis lengvai prisitaikyti prie besikeičiančių reikalavimų, technologijų ir verslo sąlygų.
4. **Saugumas:** Įvertinkite, kaip kiekviena alternatyva užtikrina sistemos ar produkto saugumą ir duomenų apsaugą. Svarstykite, ar architektūros sprendimas atitinka saugumo standartus ir reikalavimus bei leidžia efektyviai valdyti rizikas.
5. **Priežiūra ir palaikymas:** Vertinkite, kokiais būdais kiekviena alternatyva gali būti priežiūros ir palaikymo požiūriu. Svarstykite, ar architektūros sprendimas leis lengvai palaikyti ir atnaujinti sistemą ar produktą, taip užtikrinant ilgalaikį naudą ir veikimo stabilumą.
6.  **Funkcinis atitikimas reikalavimams**:
7. Ar architektūra atitinka visus suinteresuotų šalių ir vartotojo reikalavimus? Šis kriterijus apima reikalavimų įgyvendinimo galimybes, funkcionalumą, kurį architektūra gali užtikrinti.
8.  **Našumas ir patikimumas**:
9. Kaip siūloma architektūra užtikrina našumo, patikimumo, prieinamumo ir mastelio didinimo reikalavimus? Tai apima ir sistemos atsako laiką, klaidų toleranciją, sistemos stabilumą ir gebėjimą veikti skirtingomis apkrovomis.
10.  **Lankstumas ir plėtojimo galimybės**:
11. Kiek architektūra yra lanksti ir lengvai pritaikoma ateities poreikiams, įskaitant naujų funkcijų integravimą, technologijų keitimą ir sistemos plėtrą?
12.  **Saugumas**:
13. Ar architektūra užtikrina reikiamą saugumo lygį, apsaugant duomenis ir sistemą nuo neteisėtos prieigos, kibernetinių grėsmių ir kitų rizikų?
14.  **Suderinamumas ir integravimas**:
15. Kaip siūloma architektūra derės su esamomis sistemomis ir technologijomis? Tai apima integravimo su kitomis sistemomis galimybes, taip pat suderinamumą su naudojamomis technologijomis ir standartais.
16.  **Kaina ir išlaidos**:
17. Kokios yra siūlomos architektūros kūrimo, įgyvendinimo, priežiūros ir eksploatacijos išlaidos? Šis kriterijus padeda įvertinti ekonominę naudą, palyginti su kitomis alternatyvomis.
18.  **Sudėtingumas ir rizikos**:
19. Kiek sudėtinga yra įgyvendinti ir palaikyti siūlomą architektūrą? Kokios yra galimos rizikos ir jų valdymo galimybės? Tai gali apimti techninę sudėtingumą, projekto vykdymo rizikas ir galimus trūkumus.
20.  **Technologinis tinkamumas**:
21. Ar architektūroje naudojamos technologijos yra patikrintos ir tinkamos? Šis kriterijus apima technologijų brandumą, palaikymą ir tinkamumą numatytai sistemai.
22.  **Gyvavimo ciklo aspektai**:
23. Kaip architektūra įtakoja viso sistemos gyvavimo ciklo procesus, nuo kūrimo iki utilizavimo? Tai apima sistemos ilgaamžiškumą, atnaujinimo galimybes ir palaikymą per visą gyvavimo ciklą.

Šie kriterijai padės organizacijai išsiaiškinti, kokia architektūros alternatyva geriausiai atitinka jų poreikius ir tikslus, padės priimti informuotus sprendimus ir užtikrinti sėkmingą architektūros sprendimo įgyvendinimą.

**8-** **Pateikite nefunkcinio reikalavimo ir jo itakos architekturai pavyzdi pakomentuokite kas architekturoje bus itakojama tarsi noretumete savo komandai tai isaiskinti**

**Nefunkcinis reikalavimas:** **Sistemos atsako laikas turi būti ne ilgesnis nei 200 milisekundžių bet kuriuo paros metu, nepriklausomai nuo vartotojų skaičiaus.**

**Komentaras apie įtaką architektūrai:**

**Kai tokį reikalavimą priskiriame sistemai, jis daro didelę įtaką tam tikriems architektūros sprendimams. Štai kaip šis reikalavimas gali paveikti architektūrą:**

1. **Technologijų pasirinkimas:**
   * **Duomenų bazės: Reikės pasirinkti greitas, aukštos prieinamumo duomenų bazes, kurios palaiko greitą skaitymą ir rašymą. Tai gali reikšti NoSQL duomenų bazės pasirinkimą, pvz., „Cassandra“ arba „Redis“, kurios gali efektyviai valdyti didelius duomenų srautus ir užtikrinti trumpą atsako laiką.**
   * **Caching mechanizmai: Reikės integruoti duomenų talpyklos (caching) sprendimus, pvz., „Memcached“ arba „Redis“, kad būtų galima greitai pasiekti dažnai naudojamus duomenis ir sumažinti apkrovą duomenų bazėms.**
2. **Tinklų architektūra:**
   * **Turinio tiekimo tinklai (CDN): Norint sumažinti duomenų perdavimo vėlavimus, gali tekti integruoti CDN, kuris padės paskirstyti turinį arčiau vartotojų geografiniu požiūriu.**
   * **Load Balancer: Sistemos atsako laiko užtikrinimui gali tekti naudoti apkrovos balansavimo sprendimus (load balancer), kurie paskirstytų vartotojų užklausas tarp skirtingų serverių, taip užtikrinant greitą atsaką net ir esant dideliam vartotojų skaičiui.**
3. **Sistemos architektūros modelis:**
   * **Microservices: Galima pasirinkti mikropaslaugų (microservices) architektūros modelį, kuris leis skirtingoms sistemos dalims veikti savarankiškai ir efektyviai skirstyti užduotis. Tokiu būdu galima užtikrinti, kad net ir viena paslauga užtrukusi nepaveiks visos sistemos atsako laiko.**
   * **Horizontalus skalavimas: Reikės paruošti architektūrą horizontaliam skalavimui, t. y., pridėti daugiau serverių, kai auga vartotojų skaičius, užtikrinant, kad atsako laikas išliks nepakitęs.**
4. **Monitoringas ir optimizacija:**
   * **Performance monitoring tools: Norint užtikrinti, kad sistema visada atitiktų atsako laiko reikalavimus, reikės įdiegti monitoringui skirtus įrankius, kurie realiu laiku stebės sistemos veikimą ir leis greitai aptikti bei pašalinti našumo problemas.**

Nefunkcinis reikalavimas gali būti, pavyzdžiui, "Sistema turi turėti aukštą našumą, kad būtų galima apdoroti didelį kiekį duomenų per sekundę." Šis reikalavimas turi itin svarbią įtaką sistemos architektūrai, nes jis tiesiogiai apibrėžia, kaip sistema turėtų būti sukuriama ir kaip jos komponentai turi būti suderinti, kad būtų užtikrintas norimas našumo lygis.

Pavyzdžiui, norint užtikrinti aukštą našumą, galima numatyti šias architektūrines sprendimo kryptis:

1. **Skaliuojamumas:** Svarbu, kad sistema būtų skaliuojama horizontaliai, t.y., kad ji būtų lengvai išplečiama pagal didėjantį apkrovos dydį. Tai gali reikšti, kad sistema turi būti išdėstyta taip, jog būtų galima pridėti papildomų serverių arba išplesti duomenų bazės talpą.
2. **Optimizavimas:** Būtina optimizuoti sistemos veikimą, kad būtų pasiektas reikiamas našumo lygis. Tai gali apimti efektyvų duomenų indeksavimą, duomenų kachingą, operacijų paralelizavimą ir kt.
3. **Tinkamos technologijos pasirinkimas:** Svarbu pasirinkti tinkamas technologijas, kurios gali efektyviai tvarkyti didelį duomenų srautą ir užtikrinti greitą sistemos atsakymo laiką. Pvz., gali būti naudojama sparčių duomenų bazių sistema arba in-memory technologijos.
4. **Atsparumas klaidoms:** Būtina numatyti mechanizmus, kurie užtikrintų sistemos atsparumą klaidoms ir sutrikimams. Tai gali būti automatizuotos atsarginės kopijos, klaidų atkūrimo mechanizmai arba lankstūs failovero planai.
5. **Testavimas:** Nepaprastai svarbu įgyvendinti intensyvų testavimo procesą, siekiant užtikrinti, kad sistema veiktų efektyviai ir patikimai, net ir esant dideliam našumui. Tai gali apimti streso testavimą, apkrovos testavimą ir kitus testavimo metodus.

Atsižvelgiant į šį nefunkcinį reikalavimą, komanda turėtų atidžiai apsvarstyti šias architektūrines kryptis ir sukurti sistemos architektūrą, kuri leistų užtikrinti aukštą našumą, patikimumą ir skalėjimąsi ateityje. Taip pat svarbu nuolat stebėti ir vertinti sistemos veikimą, kad būtų galima greitai reaguoti į bet kokias problemas ar poreikius, atsirandančius sistemos naudojimo procese.

8- parasykite uz ka atsakingas psi techininis procesas is standarto ISO/IEC TS 33061:2021

**Apibrėžti reikalavimus programinės įrangos sistemai,**

**Paversti reikalavimus efektyviu produktu,**

**Leisti nuosekliai atkurti produktą, jei reikia,**

**Atsikratyti produktu, kai jis yra išimtas iš naudojimo**

ISO/IEC TS 33061:2021 yra tarptautinis standartas, kuris aprašo procesų vertinimo modelį, skirtą programinės įrangos, sistemų ir paslaugų kūrimo procesų gebėjimams vertinti. Šis standartas yra dalis platesnio ISO/IEC 330xx serijos standartų, kurie apima programinės įrangos ir sistemų inžinerijos procesų vertinimą.

**PSI (Product System Implementation) techninis procesas** šiame kontekste yra atsakingas už šiuos pagrindinius aspektus:

1. **Sistemos ir jos komponentų įgyvendinimas (Implementation of the System and Its Components):**
   * PSI techninis procesas apima veiklas, susijusias su sistemos komponentų kūrimu, jų integravimu ir sistemos kaip visumos įgyvendinimu. Tai apima ne tik programinės įrangos kūrimą, bet ir techninės įrangos integravimą, konfigūravimą bei sistemų testavimą.
2. **Reikalavimų įgyvendinimas (Implementation of Requirements):**
   * Šis procesas užtikrina, kad visi nustatyti sistemos reikalavimai būtų tinkamai įgyvendinti. Tai reiškia, kad sukurta sistema turi atitikti visus techninius, funkcinius ir nefunkcinius reikalavimus, kurie buvo apibrėžti ankstesniuose projektavimo etapuose.
3. **Techninių sprendimų integravimas (Integration of Technical Solutions):**
   * PSI procesas yra atsakingas už techninių sprendimų integravimą į galutinę sistemą. Tai apima įvairių komponentų ir posistemių integravimą, siekiant užtikrinti, kad sistema veiktų sklandžiai ir efektyviai.
4. **Sistemos patikra ir testavimas (System Verification and Testing):**
   * Procesas apima sistemos ir jos komponentų testavimą, siekiant patvirtinti, kad ji atitinka visus reikalavimus ir veiklos tikslus. Patikra apima funkcinius testus, našumo vertinimus, saugumo patikrinimus ir kitas testavimo veiklas.
5. **Dokumentavimas ir diegimas (Documentation and Deployment):**
   * PSI procesas taip pat atsakingas už sistemos dokumentacijos paruošimą ir galutinio produkto diegimą vartotojo aplinkoje. Tai apima ir naudotojų apmokymus, diegimo instrukcijas bei pagalbą paleidimo metu.

Apibendrinant, PSI techninis procesas iš ISO/IEC TS 33061:2021 standarto yra atsakingas už visų su sistemos kūrimu, integravimu, testavimu ir diegimu susijusių techninių veiklų vykdymą, užtikrinant, kad sukurta sistema atitiktų visus nustatytus reikalavimus ir būtų paruošta naudoti.

**9- parasykite uz ka atsakingas psi techininis procesas is standarto ISO/IEC 12207:2017**

**Apibrėžti reikalavimus programinės įrangos sistemai,**

**Paversti reikalavimus efektyviu produktu,**

**Leisti nuosekliai atkurti produktą, jei reikia,**

**Atsikratyti produktu, kai jis yra išimtas iš naudojimo**

**ISO/IEC/IEEE 12207:2017 divides software life cycle**

**processes into four main process groups:**

**1. Agreement**

**• address the cooperation and agreement with other organisations**

**2. Oganizational project-enabling processes**

**• processes are performed on the level of the development**

**organisation and provide the environment needed to perform**

**projects.**

**3. Technical management**

**• Project processes**

**4. Technical processes**

**• describe the various processes or phases within a software product**

**life cycle, from stakeholder requirements definition to software**

**disposal.**

** Produktų komponentų integravimas:**

* **Užtikrina, kad visi sistemos komponentai (programinė įranga, aparatinė įranga, ir kt.) būtų tinkamai integruoti į vieną veikiančią sistemą. Tai apima atskirų komponentų derinimą, jų suderinamumo patikrinimą ir integruotos sistemos veikimo patvirtinimą.**

** Sistemos integracijos planavimas:**

* **Planavimas ir strategijos kūrimas, kaip bus vykdoma komponentų integracija. Tai apima veiksmų seką, integracijos eiliškumą, išteklių paskirstymą ir integracijos laiko grafikus.**

** Integracijos testavimas ir validacija:**

* **Atlieka testavimo ir validavimo veiklas, siekiant užtikrinti, kad integruota sistema atitiktų nustatytus funkcinius ir nefunkcinius reikalavimus. Šio proceso metu identifikuojami ir sprendžiami bet kokie integracijos metu atsiradę nesuderinamumai ar klaidos.**

** Galutinės sistemos paruošimas naudojimui:**

* **Užtikrina, kad integruota sistema būtų paruošta eksploatacijai. Tai apima galutinį testavimą, konfigūravimą, dokumentacijos rengimą ir vartotojo mokymus, jei tai yra būtina.**

Pagal ISO/IEC 12207:2017 standartą, Paslaugų Techninis Procesas (Service Technical Process) yra atsakingas už programinės įrangos kūrimą, palaikymą ir tobulinimą. Šis procesas apima įvairias veiklas, skirtas kurti, palaikyti ir tobulinti programinę įrangą, kad ji atitiktų klientų poreikius ir organizacijos tikslus.

Kai kurios svarbios veiklos, kurias atlieka Paslaugų Techninis Procesas, gali apimti:

1. **Reikalavimų analizė ir specifikavimas:** Proceso metu inžinieriai analizuoja klientų poreikius ir reikalavimus bei nustato, kaip jie turėtų būti pateikti programinės įrangos funkcionalumui ir techninėms charakteristikoms.
2. **Projektavimas:** Proceso metu inžinieriai projektuoja sistemos architektūrą ir komponentų struktūrą, kad būtų užtikrinta efektyvi ir patikima programinės įrangos veikla.
3. **Kodavimas ir testavimas:** Proceso metu programuotojai kuria programinę įrangą, o kokybės inžinieriai atlieka testavimą, kad būtų užtikrinta, jog programa veikia tinkamai ir atitinka specifikacijas.
4. **Implementacija ir integracija:** Proceso metu programinė įranga yra įdiegiama ir integruojama su kitomis sistemomis bei komponentais, siekiant užtikrinti bendrą veikimą ir sąveiką.
5. **Palaikymas ir tobulinimas:** Proceso metu vykdomos palaikymo veiklos, įskaitant klaidų taisymą, atnaujinimus ir įdiegimą, taip pat nuolatinis programinės įrangos tobulinimas atsižvelgiant į klientų grįžtamąjį ryšį ir besikeičiančius poreikius.

Paslaugų Techninis Procesas yra esminis komponentas, kuris leidžia organizacijoms kurti, palaikyti ir tobulinti aukštos kokybės programinę įrangą, užtikrinant, kad ji atitiktų klientų lūkesčius ir organizacijos tikslus.

**10-** **Pateikite reikalavimo detalia specifikacija iki DoR busenos sistema sugeneruoja saskaita uz patvirtinta uzsakyma**

**Pavadinimas:** Sąskaitos generavimo sistema už patvirtintą užsakymą

**Trumpas aprašymas:** Sistema turi automatiškai generuoti sąskaitą, kai užsakymas yra patvirtintas.

**Detalus aprašymas:**

1. **Užsakymo patvirtinimas:** Sistema gali nustatyti, kai kliento pateiktas užsakymas yra patvirtintas.
2. **Duomenų surinkimas:** Sistema surenka reikiamus duomenis generuoti sąskaitą, įskaitant kliento informaciją (vardą, adresą, kontaktinius duomenis), užsakymo informaciją (prekių pavadinimus, kiekius, kainas), mokėjimo informaciją ir kitus atitinkamus duomenis.
3. **Sąskaitos sudarymas:** Pagal surinktus duomenis, sistema automatiškai sugeneruoja sąskaitą, kurioje nurodomos užsakymo detales ir suma.
4. **Sąskaitos pristatymas:** Sugeneruota sąskaita yra pristatoma klientui elektroniniu formatu per el. paštą arba pateikta klientui per klientų sąsają (pvz., internetinės parduotuvės paskyros puslapį).
5. **Sąskaitos saugojimas:** Sistema saugo sąskaitų kopijas elektroniniu formatu tam, kad jos būtų prieinamos vėlesniam peržiūrėjimui arba dokumentų palaikymui.

**Reikalavimai:**

* Sistema turi automatiškai identifikuoti patvirtintus užsakymus.
* Duomenys, reikalingi sąskaitos generavimui, turi būti patikimi ir tikslūs.
* Sugeneruotos sąskaitos turi atitikti įprastus sąskaitos formatavimo ir informacijos reikalavimus.
* Sistema turi būti pakankamai patikima ir stabiliai veikti, kad užtikrintų sąskaitos generavimo proceso veiksmingumą.

**Kriterijai, kurie patikrina ar užduotis atitinka DoR:**

* Užsakymas yra sėkmingai patvirtintas sistemoje.
* Sąskaita yra sugeneruota automatiškai ir yra teisingai pateikta, atitinkanti reikalavimus.
* Sugeneruota sąskaita yra sėkmingai pristatyta klientui.

Ši detalioji specifikacija užtikrina, kad komanda ir suinteresuotos šalys supranta, ką sistema turi padaryti ir kaip tai bus vertinama. Tai taip pat suteikia aiškų pagrindą tolimesniam sistemos vystymui.

**Definition of Ready (DoR):**

* User Story is clear
* User Story is testable
* User Story is feasible
* User Story defined
* User Story Acceptance Criteria defined
* User Story dependencies identified
* User Story sized by Development Team
* Scrum Team accepts User Experience artefacts
* Performance criteria identified, where appropriate
* Scalability criteria identified, where appropriate
* Security criteria identified, where appropriate
* Person who will accept the User Story is identified
* Team has a good idea what it will mean to Demo the User Story

Proceso brandos lygiai (CMMI)

0. Neegzistuojantis procesas

– Nėra sistemišką veikimą pagrindžiančių įrodymų

• 1. Pradinis procesas (Initial)

– Chaotiškas – neprognozuojamos kainos, terminai ir kokybė

• 2. Atkartojamas procesas (Repeatable)

– Intuityvus – kaina ir kokybė smarkiai varijuoja, pakankamas terminų valdymas,

neformalios procedūros

• 3. Apibrėžtas procesas (Defined)

– Kokybinis – patikimai nustatomos kainos ir terminai, gerėjanti, bet

neprognozuojama kokybė

• 4. Kiekybiškai valdomas procesas (Quantitatively Managed)

– Kiekybinis – pakankama statistinė produktų kokybės kontrolė

• 5. Nuolatos gerinamas procesas (Optimizing)

– Kiekybinės metrikos naudojamos nuolatiniam proceso automatizavimo plėtimui ir

proceso gerinimui

Benefits from Using CMMI

1. Better customer satisfaction

2. Increased quality

3. More accurate schedules

4. Lower development costs

5. Substantial return on investment

6. Improved employee morale and reduced turnover

Ivardinkite esmini skirtuma tarp inkrementinio ir

Prototipais, koks tai protatipp tipas

Iso: verification