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 definiiton) proceso dedamasias pagal pasirinkta proceso specifikavimo sablona**

Sistemos architektūros apibrėžties procesas yra svarbus etapas, kuriame nustatoma sistemos struktūra, komponentai, sąsajos ir jų veikimo principai. Tai leidžia sukurti bendrą supratimą apie sistemą ir užtikrinti, kad visi suinteresuoti asmenys supranta, kaip sistema turėtų veikti ir kokias funkcijas atlikti.

Čia pateiksiu pagrindinius žingsnius, kaip galima būtų struktūruoti sistemos architektūros apibrėžties procesą, remiantis pasirinktu proceso specifikavimo šablonu:

1. **Reikalavimų analizė:** Pirma, reikia atlikti išsamią reikalavimų analizę, kad būtų nustatyti sistemos funkcionalūs ir nefunkciniai reikalavimai. Tai apima supratimą apie vartotojų poreikius, verslo tikslus ir technines galimybes.
2. **Struktūros apibrėžimas:** Remiantis surinkta reikalavimų analize, būtina apibrėžti sistemos struktūrą. Tai apima identifikavimą, kokios komponentės sudaro sistemą, kaip jos sąveikauja tarpusavyje ir kaip jos organizuojamos hierarchiškai.
3. **Komponentų identifikavimas:** Šiame etape identifikuojami pagrindiniai sistemos komponentai. Tai gali būti programinės įrangos moduliai, duomenų bazių schemos, išorinės sąsajos arba bet kokie kiti elementai, kurie sudaro sistemą.
4. **Sąsajų apibrėžimas:** Po komponentų identifikavimo svarbu nustatyti, kaip šie komponentai sąveikauja tarpusavyje. Tai apima nustatymą, kokios funkcijos kiekvienas komponentas teikia, kaip jie perduoda duomenis ir kaip vyksta komunikacija tarp jų.
5. **Veikimo principų nustatymas:** Galiausiai svarbu nustatyti sistemos veikimo principus ir apribojimus. Tai gali apimti veikimo tvarką, duomenų saugumo reikalavimus, naudojamus technologijų standartus ir kitus svarbius aspektus, kurie turi būti įtraukti į sistemos architektūros apibrėžtį.

Šie žingsniai padeda struktūruoti ir apibrėžti sistemos architektūrą, užtikrinant, kad visi suinteresuoti asmenys turėtų bendrą supratimą apie tai, kaip sistema turėtų veikti ir kokias funkcijas atlikti. Tai svarbus procesas, kuris padeda užtikrinti, kad sistema būtų kokybiškai suprojektuota ir atitiktų kliento poreikius bei verslo tikslus.

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

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**

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.

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**

CMMI-DEV (Capability Maturity Model Integration for Development) techninio sprendimo (technical solution) procese yra kelios pagrindinės veiklos, kurios padeda plėtoti ir įgyvendinti techninį sprendimą sistemos ar produkto kūrimo procese. Štai pagrindinės veiklos, kurios gali būti vykdomos šiame procese:

1. **Reikalavimų analizė ir specifikavimas:** Šioje veikloje analizuojami ir aiškinami reikalavimai, kurie bus pagrindas techniniam sprendimui. Tai apima reikalavimų rinkimą, jų struktūrizavimą, prioritetizavimą ir specifikavimą taip, kad būtų galima suprasti, ką sistema turi daryti ir kaip ji turi veikti.
2. **Architektūros kūrimas ir projektavimas:** Šioje veikloje yra kuriama sistemos ar produkto architektūra, nustatomi pagrindiniai komponentai, jų sąsajos ir sąveikos, bei bendras sistemos struktūros modelis. Tai leidžia suplanuoti, kaip sistema bus organizuota ir kaip jos komponentai veiks kartu.
3. **Techninių sprendimų kūrimas:** Šioje veikloje yra kuriama techninės problemos sprendimo strategija ir realizuojami konkrečūs techniniai sprendimai, atitinkantys sukurtos architektūros ir reikalavimus. Tai gali apimti programinės įrangos kūrimą, sistemos integraciją, duomenų bazių kūrimą ir kt.
4. **Techninio sprendimo vertinimas:** Šioje veikloje yra vertinami ir įvertinami sukurti techniniai sprendimai, siekiant užtikrinti jų tinkamumą ir veiksmingumą. Tai apima techninės dokumentacijos peržiūrą, sprendimų analizę ir galimų alternatyvų įvertinimą.
5. **Palaikymo ir tobulinimo planavimas:** Šioje veikloje yra planuojamas techninio sprendimo palaikymas ir tobulinimas ateityje. Tai apima priemonių, skirtų techninio sprendimo palaikymui, įdiegimui ir tobulinimui, nustatymą ir vykdymą.

Šios pagrindinės veiklos yra būtinos techninio sprendimo procese, siekiant užtikrinti, kad sistema ar produktas būtų sukurtas atitinkančiu reikalavimus ir būtų veiksmingas bei patikimas.

**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.

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 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.

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

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.