# Final project – Day 1

## Uzdevums

Kādus testa līmeņus un tipus tu atceries?

Testa līmeņi ir vairāki:

1. komponenšu testēšana, kurā pārbauda, vai programmas kods un moduļi strādā pareizi. Šos testus veic izstrādātāji vai CI/CD sistēma un tie ir 100% automatizēti. Izpilda izstrādes un testa vidēs.
2. Integrācijas testos pārbauda dažādu moduļu un sistēmu savstarpēju mijiedarbību un atbilstību specifikācijai. Iekļauj arī regresijas testus.
3. Sistēmas testos pārbauda visas sistēmas vai produkta darbību, lai noteiktu atbilstību specifikācijām un vai produkts darbojas pareizi. Iekļauj sevi funkcionālos un nefunkcionālos testus. Šos testus veic testētājs testa vidē, kas ir gandrīz kopija produktam produkcijas vidē.
4. Acceptance testos pārbauda visas sistēmas pabeigtību un atbilstību izvirzītajiem mērķiem. Iekļauj alfa/beta testus . Tiek pārbaudīts, vai produkts ir gatavs nodošanai klientam saskaņā ar acceptance criteria.

Testa tipi ir:

1. Funkcionālie testi, tos veic katrā testēšanas līmenī, izmantojot melnās kastes tehniku testa case veidošanai.
2. Nefunkcionālie testi pārbauda, cik labi sistēma strādā (iekļauj performance testing, load testing, etc.), testi parasti ir automatizēti un veidoti pēc melnās kastes tehnikas; principā šie visi ir “….ility” testi.
3. Struktūras testi jeb baltās kastes testēšana, veic programmētāji
4. Testēšāna saistīta ar izmaiņām iekļauj regresija testus (vai kāda defekta izlabošana nav ietekmējusi iepriekšēju sistēmas darbību) un confirmation testus (caur tiem pārbauda, vai kāds defekts ir bijis izlabots)

## Uzdevums

Īsumā apraksti, kas ir testa plāns – ko tajā parasti iekļauj un ar kādu mērķi?

1)Testa plāns ir dokuments, kas apraksta nepieciešamos resursus (laiks, cilvēki, vides), metodes ar kādām tiks testēts, veidu un kārtību, kā tiks veikti testi un grafiks, izpildāmās aktivitātes.

2)Dokuments iekļauj arī produkta un projekta risku uzskaitījumu un iespējamos veidus, kā tos mīkstināt

3) Mērķis: lai visas projektā iesaistītās puses būtu informētas par visām plānotajām aktivitātēm, nepieciešamo resursu apjomu un kāda ir mūsu pieeja

## Uzdevums

Kādēļ ir nepieciešama testēšanas atskaite (test report)? Ko tajā iekļauj un kā tā atšķiras no testa plāna?

Testa reports informē par testu statusu (atrastās kļūdas) un veikto testu daudzumu – kas ir notestēts un strādā atbilsoši prasībām, kādas kļūdas atrastas produktā. Jāizvirza prioritātes, kuras kļūdas jālāba pirmās.

* Izpildītie testi un laiks
* Kļūdu uzskaitījums
* Iesaistītas personas
* Vides
* Gala slēdziens

Atšķiras no testa plāna ar to, ka plānā mēs plānojam, bet atskaitē mēs sekojam līdz plānam, nosakam , kādas klūdas ir izlabotas un kas vēl jālabo (tiek atspoguļota reālā situācija).

## Uzdevums

Īsumā apraksti testu veidus (test design techniques)

* Static testing- iekļauj pārskatus (review), inspekcijas (formāls review) un statisko analīzi (ar dažādu toolu palīdzību, piem., Sonarqube) testējamajam produktam bez koda palaišanas(code-analysis, code-review, peer-review). Uzmanība tiek vērsta vairāk uz dokumentāciju. Iedalās kategorijās: formal, informal, walkthrough. Var tikt veikts gan manuali, gan automatizēti.
* Black-box technique- izmanto pārsvarā testētāji gan funkcionalajā, gan nefukcionalajā testēšanā, salīdzinot ieejas datus ar izejas datiem un novērtējot, vai rezultāts ir vēlamais un atbilstošs prasībām. Balstās uz dokumentāciju nevis uz koda saturu. Testi tiek veikti neidziļinoties to kodā,struktūrā, bet vairāk testējot to kā vienu veselumu, end to end. Iekļauj equivalence partioning, robežvērtību analīzi, atbilstību lēmuma tabulai , un vai izpildās stāvokļu pārejas (state transition testing). Iekļauj arī testus no lietotāja viedokļa, vai programmas darbība atbilst gaidītajam rezultātam saskaņā ar esošu dokumentāciju.
* White-box technique- pārsvarā nepieciešami programmētājiem, lai sagatavotu atbilstošus testus programmas koda un koda integrācijas līmenī (rakstot automatizētos testus) – tiek veikta paša koda analīze. Testi tiek veikti iedziļinoties koda struktūrā, moduļu uzbūvē. Šie testi ir papildinoši black box testiem un tos nepieciešams zināt arī testētājiem.Testē, izpildot pašu kodu un neizmantojot lietotāja saskarni.
* Tehnika, kas balstīta pēc defektiem un pieredzes (dynamic testing)- šos testus veic, kad ir ierobežots laiks testu veikšanai, veic bez nekāda testa plāna bet tikai uz iepriekš iekrāto pieredzi. Testi tiek veikti palaižot kodu, pārbaudot atsevišķu programmas moduļu izpildīšanos ,vai savstarpējo integrāciju. Checklist, exploratory, error guesing.
* Citas tehnikas- TDD jeb agile tehnika= raksta testus un tad izstrādā programmatūru, kas atbilst šiem testiem ar mērķi, lai testi ir OK; BDD jeb tehnika balstoties uz biznesa prasībām (no biznesa uzvedības puses) vai jebkura cita Agile tehnika; KDD – veido testus ar visbiežāk izmantotajām operācijām

## Uzdevums

Kā atšķiras produkta un projekta riski? Dod pāris piemērus pie katra veida.

Projekta riski ir vairāk organizatoriski (nav pieejams testa vides un atbisltošie tooli), bet produkta ir saistīti tieši ar kvalitātīva produkta izstrādi- piem., sarežģīta arhitektūra, noslogoti grafiki progammētājiem- testētājiem

Produktu riski – ietekmē izstrādājamo produktu (tā atbisltību lietotāja vajadzībām, veiksmi tirgū)

* Produkts ir strādājošs, bet ne tādos apstākļos kā lietotājs tos lietos
* Nav izstrādāta, ir palaista garām, būtiska funkcionalitāte, bez kuras produkts nav lietojums
* Nav droši lietojams (piem., veselības jomā)

Projekta riski – ar projektu pārvaldību saistīti riski (riski, kas saistīti ar produkta piegādājamību laikā un prasītā apjomā)

* Bieži tiek mainītas prasības
* Galvenais izstrādātājs saslimst
* Komandu konflikti

## Uzdevums

Kas obligāti ir jāiekļauj kļūdas pieteikumā? (defect)

Preconditions, vide, defekta Nr. utml. Detalizēti, tas izpaužas šādi:

* Enviroinment – kurā vidē, kāda produkta versijā tiak atrasta kļūda
* Steps to reproduce –kādi soļi ir jāveic, lai atkārtotu/pieredzētu kļūdu
* Expected result – kam bija jānotiek šo soļu rezultātā
* Actual result – kas notika
* Visual Proof (screenshots, video) – uzskates materiāls, vieglākai kļūdu uztveršanai
* Severity/ priority – ko tas ietekmē, nozīmē un cik steidzami jālabo

## Uzdevums

Uzraksti pats savu testēšanas plānu – produktam “OnMenu”, kas ļauj aplūkot dažādu ēstuvju ēdienkartes un arī pasūtīt ēdienu.

*Introduction to testing* daļā jums bija piemērs *test report* un *test plan*. Tajā prezentācija arī ir template ko tu droši vari izmantot – tikai aizpildi pats.