## Atkļūdošanas un akcepttestēšanas pārskats

*SR: Veic programmatūras vienību izstrādi un vienībtestēšanu, izstrādājamās programmatūras vienību apvienošanu, integrācijas un akcepttestēšanu atbilstoši izstrādātajai programmatūras prasību specifikācijai un projektējuma aprakstam. (T.A. 2.4.6.)*

## Vienībtestēšana

Ir sastopams testēšanas veids, kurā kodu sadala vienībās un testē katru vienību atsevišķi, lai pārbaudītu un apstiprinātu, ka vienība strādā, kā paredzēts. Šo testēšanu parasti veic paši izstrādātāji, veidojot kodu. Vienības var būt: metode, funkcija, objekts.

Vienībtestēšana ir baltās kastes testēšanas paveids. Baltās kastes programmēšanā testētājs zina iekšējo programmas kodu un struktūru, pārbauda lielākoties datu plūsmu no ievaddatiem līdz izvaddatiem, vai pareizi norit nelielo vienību darbība, pārbaudot iespējamās ievades un izejas datus.

Vienībtesti ir automātiskie un pašrocīgie, kuros pārbauda programmas darbības sazarojumus, ciklus, funkcijas, kopējo datu pārskatu.

Labā prakse vienībtestēšanā:

* testēšanas gadījumiem jābūt neatkarīgiem;
* testēšanas gadījumi nav pārāk sarežģīti un lieli, citādi jāpatērē pārāk daudz resursu pārbaudei;
* veido skaidrus un testu izskaidrojošus nosaukumus testam;
* pirms veidot koda nākamo daļu izlabo testā notvertās kļūdas;
* testus veido, jau kodējot.

Testēšanas (piemērs) telefona numuram :

|  |  |  |
| --- | --- | --- |
| Testa numurs | Ieejas dati | Izejas dati |
| 1 | 0234567 | Tas nav Latvijā derīgs parastais lietotāja tālruņa numurs, jo mazāk par 8 cipariem! |
| 2 | -013 | Tālruņa numuram jābūt pozitīvam! |
| 3 | 012345678 | Tas nav Latvijā derīgs parastais lietotāja tālruņa numurs, jo vairāk par 8 cipariem! |
| 4 | “” | Izvada kļūdas paziņojumu. Nekas nav ievadīts, ievadi tālruņa numuru! |
| 5 | 21234567 | Ievadīts derīgs tālruņa numurs! |

Testēšanas rezultāti:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Testa | Ieejas dati | Izejas dati | Iegūtais | Piezīmes |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| numurs |  |  | rezultāts |  |
| 1 | 2234567 | Tas nav Latvijā derīgs parastais lietotāja tālruņa numurs, jo mazāk par 8 cipariem! | atbilst | Kļūdas paziņojums. Programma turpina darbu, līdz ievada 8 ciparu tālruņa numuru. |
| 2 | -123 | Tālruņa numuram jābūt pozitīvam! | atbilst | Kļūdas paziņojums. Programma turpina darbu, līdz ievada pozitīvu skaitli, kas atbilst tālruņa numuram. |
| 3 | 123456789 | Tas nav Latvijā derīgs parastais lietotāja tālruņa numurs, jo vairāk par 8 cipariem! | atbilst | Kļūdas paziņojums. Programma turpina darbu, līdz ievada 8 ciparu tālruņa numuru. |
| 4 | “” | Izvada kļūdas paziņojumu. Nekas nav ievadīts, ievadi tālruņa numuru! | neatbilst | Programma neizvada kļūdas paziņojumu |
| 5 | 21234567 | Ievadīts derīgs tālruņa numurs! | atbilst | Programma izvada atbildi un beidz darbu. |

## Akcepttestēšana

Akcepttestēšana notiek programmas izstrādes beigās, kad pārbauda visu sistēmu, vai tā atbilst specifikācijai, vai tā ir pietiekama, vai darbojas korekti un ir gatava lietošanai. Tas ir melnās kastes testēšanas paveids, kad nevar izpētīt visu koda iekšpusē notiekošo, bet ievada datus un redz izvades rezultātu. Melnās kastes programmēšanā testētājs nezina detaļas par to, kas notiek vienības iekšpusē, vai neprot programmēt, bet zina ievaddatus un izvaddatus.

Pirms testēšanas jāizveido tabula ar visām programmas darbībām, kuras programmatūrai jāpiedāvā atbilstoši specifikācijai. Lietotāji pārbauda produktu un novērtē, vai tas strādā, kā aprakstīts.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Programmatūras iespējamās darbības, funkcijas** | **Jā** | **Daļēji** | **Nē** | **Piezīmes** |
| Ievada klientu apmeklēšanas datumu un laiku. |  | x |  |  |
| Mainīt klienta apmeklēšanas datumu |  |  | x |  |
| Klientu datus uzglabāt datu bāzē. | x |  |  |  |
| Maina klienta datus datu bāzē | x |  |  |  |

## Lietotāja ceļvedis

*SR: Izstrādā un prezentē izveidotās programmatūras vienkāršotu izvēršanas (t. sk. ieviešanas) plānu, lietotāja ceļvedi un uzturēšanas plānu, ievērojot tās lietotāju*

*mērķauditorijas specifiku. (T.A.2.4.7.)*

Jāapraksta:

1. Ievads. Kam paredzēta programmatūra?
2. Programmatūras apraksts:
   1. Ko sistēma atbalsta, kādus pārlūkus?
   2. Kā var pieslēgties sistēmai?
   3. Kas notiks, ja pieslēdzas un kļūdās paroles vai lietotājvārda ievadē?
   4. Ko redz lietotājs savā lietotāja klasē?
3. Ieviešanas plānu, uzturēšanas plānu, lietotāju mērķauditorijas specifiku Piemērs:

Šīs sistēmas lietotāja ceļvedis ir paredzēts visām lietotāju klasēm – bet apskatītajā piemērā ir tikai viena klase, kosmētikas kabineta īpašniece.

Sistēma atbalsta Mozilla Firefox, Opera.

Lai pieslēgtos sistēmai, ir jāatver tīmekļa pārlūks un jāievada vietnes adrese. Tad jāievada lietotājvārds, parole un jānospiež uz pogu *pieslēgties*.

Ja pieslēdzas un kļūdās, lietotājvārda vai paroles ievadē parāda kļūdas ziņojumu “Nepareiza parole/lietotājvārds!”.

Kosmētikas kabineta īpašnieces funkcijas:

var aplūkot, ievadīt un mainīt klienta datus,

var aplūkot, ievadīt un mainīt pakalpojuma datus, var aplūkot, ievadīt un mainīt izejvielu datus,

var aplūkot, ievadīt un mainīt darījuma datus.

Klienta dati:

Pakalpojuma dati:

Izejvielu dati:

Darījuma dati:

Ieviešanas plāns – plāno, kā ieviest programmatūru dzīvē, nosakot konkrētus termiņus un mērķus, kā arī aprakstot konkrētās lietošanas grupas.

## Piemērotās licences pamatojums

*SR: Salīdzina atvērto kodu licences un to atšķirības, izmanto un piemēro atbilstošāko no licencēm savam programmatūras projektam. (T.A.3.1.4.)*

## Atvērtā koda licences:

* Licences bez autortiesību ierobežojumiem (Copyleft) licences - jāizplata tikpat brīvi
  + Aparatūrai
    - CERN OHL ( open hardware licence) līdzīgas atļaujas kā atvērtā pirmkoda programmatūrai)
    - TAPR OHL līdzīgas atļaujas kā atvērtā pirmkoda programmatūrai, bet atvērtā koda aparatūrai
  + Vispārējas
    - GNU GPL (General Public License) – nedrīkst piemērot papildu nosacījumus, kas slēptu kodu
    - CC Share Alike - izveidoto darba atvasinājumus nedrīkst licencēt stingrāk
* Atļauju dodošas (permissive) licences
  + MIT licence jānorāda atsauce uz autoru (*attribution*)
  + Free BSD - Berkley Software Distribution – modificētās licences
  + Creative Commons (CC) Attribution - jānorāda atsauce uz autoru (*attribution*)

Atvērtā koda bibliotēka parasti nozīmē, ka kodu var mainīt, kopēt, publicēt, bet dažreiz autors ierobežo:

* Neaizsargātie darbi (public domain):
  + lietot var visi bez ierobežojumiem,
  + nav jāmaksā,
  + nav autortiesību.
* Brīvprogrammatūras(freeware):
  + lietot var visi, bet var neļaut kopēt, izplatīt un nepieejams kods,
  + nav jāmaksā,
  + ir autortiesības.
* Brīvā programmatūra (free software):
  + lietot var visi, pat kopēt un izplatīt kodu,
  + var prasīt ziedojumus, licence var aizliegt ierobežot atvasinātos darbus,
  + ir autortiesības.
* Atvērtā pirmkoda programatūra (open-source software):
  + lietot var visi, pat kopēt, mainīt tiešsaistēs grupās, bet var būt licence,
  + kodu nav jāmaksā, bet par papildpakalpojumiem lietotājiem var prasīt apmaksu,
  + ir autortiesības.

|  |  |
| --- | --- |
| Saīsinājumi | Skaidrojumi |
|  | CC - Creative Commons, atvērtā licence. Tā aizsargā tikai daļu no  autortiesībām. Autors var izvēlēties dažādus atvērto licenču paveidus, lai precīzāk nodefinētu, ko vēlas un ko nevēlas atļaut darīt ar savu darbu. |
|  | BY - attribution, var lietot, kopēt, pavairot, veidot atvasinātus darbus, miksēt, bet jāuzrāda autora vārds uzvārds, kas veidojis darbu, jābūt atsaucei. |
|  | SA - share alike, tad, kad cits autors izveido darba atvasinājumus, tos nedrīkst licenzēt ar stingrāku licenci nekā sākotnējā licence. Savukārt, ja nav SA, tad, izmantojot kāda darbu un remiksējot, veidojot atvasinājumus rezultātu, drīkst  licencēt stingrāk. |
|  | NC - non-commercial, nelietot komerciāliem nolūkiem, nedrīkst gūt peļņu no  darba izmantošanas. |
|  | ND - no derivatives, nedrīkst veidot darba atvasinājumus, izmainīt vai  pārveidot darbu, piemēram, lietot daļu no attēla kopā ar citiem attēliem. |
|  | Public domain - visbrīvākā licence, darbu var lietot, veidojot kopijas, darbu var mainīt, var izmantot komerciālos nolūkos, var mainīt licences veidu |

## Darbā noteikti jābūt:

1. **Datu bāzei ar vairākām tabulām:**

SR: 2.3.2. Plāno datubāzi, t. sk. izveido ER modeli konkrētā uzdevuma datu apstrādes risinājumam.

SR: 2.4.17. Izveido vienkāršu datu apstrādes programmatūru (sistēmu), datu uzglabāšanai izmantojot paša veidotu datubāzi ar vairākām tabulām.

## Vismaz viena bibliotēka un/vai API:

SR: 2.4.10. Izmanto programmēšanas valodas un tās bibliotēku dokumentāciju un palīdzības sistēmu, lai patstāvīgi apgūtu citas to piedāvātās iespējas, kas nepieciešamas konkrētās programmatūras izstrādei.

SR: 2.4.11. Meklē un pievieno atvērtā koda bibliotēkas un lieto API (programmsaskarni) specializētu funkciju veikšanai sava programmēšanas projekta īstenošanai.

SR: 2.4.13. Izvēlas un lieto atbilstošas programmēšanas valodas konstrukcijas, datu tipus un dažādas bibliotēkas, veidojot programmas doto uzdevumu un problēmu risinājumam.

## Vismaz divām dažādām datu struktūrām:

SR: 2.4.14. Izmanto dažādas datu struktūras (t. sk. masīvi, kopas, ieraksti, steks, rinda, saraksts, koks, grafs, datne) un ar tiem saistītos pamatalgoritmus.

SR: 2.4.19. Veido dotā uzdevuma (problēmas) risinājumu, izmantojot gatavus algoritmus un/vai pielāgojot vai kombinējot tos, un/vai izstrādājot jaunus algoritmus. Izprot un skaidro dažādu algoritmu darbību, pielāgo tos dažādām nestandarta situācijām, ja nepieciešams, veidojot jaunas datu struktūras.

## Lietotāja piekļuvei ar datu aizsardzību.

SR: 3.1.2. Izmanto kriptogrāfijas metodes konkrētā uzdevuma risinājumā