



## IEGULDĪJUMS TAVĀ NĀKOTNĒ

Eiropas Reģionālās attīstības fonds

Prioritāte: 2.1. Zinātne un inovācijas

Pasākums: 2.1.1. Zinātne, pētniecība un attīstība

Aktivitāte: 2.1.1.1. Atbalsts zinātnei un pētniecībai

### **Projekts: "Programmatūras izstrādes rezultātu apstrādes un interpretācijas tehnoloģija"**

Projekta sākuma datums: 2014.gada 1.septembris.

Projekta beigu datums: 2015.gada 31.augustam.

Līguma Nr. 2014/0013/2DP/2.1.1.1.0/14/APIA/VIAA/034

ESF finansējuma saņēmējs: SIA, SWH SETS

Sadarbības partneris: Latvijas Universitātes aģentūra "Latvijas Universitātes

Matemātikas un informātikas institūts" (LU MII)

### **Projekta aktivitātes Nr.1.5 "Spraudņu izstrādes metodika" progresa pārskats**

Pārskats Nr.3 par periodu no 2014.gada 1.septembra līdz 2015.gada 28.februārim.

## SATURS

1.	Kopsavilkums .....	3
2.	Ievads .....	4
2.1.	Kas ir spraudnis.....	4
2.2.	Kāpēc nepieciešams spraudnis.....	4
3.	Projekta uzsākšana ar Krabis rīku.....	5
3.1.	Kādi testēšanas rīki tiek izmantoti projektā? .....	5
3.2.	Kādas sistēmas tiek izmantotas projektā? .....	5
3.3.	Kādas atskaites ir pieejamas elektroniskā veidā no iepriekš minētajiem rīkiem un sistēmām?.....	5
3.4.	Cik bieži iespējams iegūt informāciju no rīkiem? .....	6
3.5.	Cik bieži plānots mainīt testu aprakstu informāciju?.....	6
4.	Spraudņa realizācija ar programmas palīdzību. ....	7
4.1.	Testa laidiena datu aizpilde.....	7
4.2.	Testa pamat datu aizpilde.....	9
4.3.	Datu kvalitāte. ....	10
5.	Spraudņa realizācija ar datu “mappingu”. ....	12
6.	Datu importa realizācija ar sagataves failu. ....	13
7.	Datu ielādes risinājuma optimizācija. ....	14
8.	Rezultāti .....	15
9.	Literatūras saraksts.....	16

## 1. Kopsavilkums

Pārskata periodā (2014-09-01 – 2015-02-28.) projekta „Programmatūras izstrādes rezultātu apstrādes un interpretācijas tehnoloģija” aktivitātes "Spraudņu izstrādes metodika" ietvaros veikti šādi darbi:

1. Automātisko testa rīku un to rezultātu failu apzināšana.
2. Projekta izstrādātāja pieejamo automātisko testa rezultātu un rīku analīze.
3. Aizpildāmo datu analīze.
4. Metodikas izstrāde.
5. Darbs apspriests regulārās projekta sanāksmēs.

## 2. Ievads

Dokuments apraksta spraudņa izstrādes metodiku „Programmatūras izstrādes rezultātu apstrādes un interpretācijas tehnoloģija, (Krabis) tehnoloģijā.

### 2.1. Kas ir spraudnis.

Krabis sistēmas izpratnē spraudnis ir programma vai programmu kopums, ar kura palīdzību Krabis datu bāzē tiek importēti dati no jebkuras citas sistēmas, programmas vai rīka. Atkarībā no situācijas, spraudnis var tik izpildīts regulāros intervālos vai arī pēc vajadzības- kad pieejami jauni dati vai nepieciešama datu atjaunināšana Krabis sistēmā.

### 2.2. Kāpēc nepieciešams spraudnis.

Datus iespējams ievadīt arī manuāli, bet tas vairumā gadījumu ir pret produktivitātes principiem, tāpēc ieteicams izveidot spraudni datu automātiska importa nodrošināšanai. Protams, tas atkarīgs no testēšanas intensitātes, biežuma, testu skaita un attiecīgās situācijas. Tiek pieļauts arī gadījums, kad testu skaits nav liels, tie tiek pildīti manuāli, un netiek izmantota neviena cita testēšanas programmatūra, izņemot Krabis. Šādos gadījumos, noteikti, var iztikt bez spraudņa izstrādes. Pārējos gadījumos rekomendējam izstrādāt spraudni datu importam, lai varētu pilnvērtīgi izmantot Krabis piedāvātās iespējas, un vienmēr strādāt ar aktuālajiem datiem.



### 3. Projekta uzsākšana ar Krabis rīku.

Uzsākot testēšanas projektu ar Krabis rīku ieteicams veikt vairākas aktivitātes saistībā ar projekta vides analīzi un ietekmējošo faktoru noskaidrošanu. Krabis sistēmā tiek apskatīti un analizēti testu rezultāti, kas ir nozīmīgs posms projekta attīstībā. Bieži vien no testu rezultātu informācijas tiek pieņemti biznesa lēmumi, kam, neizbēgami, ir sekas, tāpēc rekomendējam rūpīgi apzināt situāciju, un sagatavot Krabis rīku darbam.

Tālākajās nodaļās uzskaitīti nozīmīgākie jautājumi, kas jāatbild darba uzsākšanas procesā.

#### 3.1. Kādi testēšanas rīki tiek izmantoti projektā?

Ļoti nozīmīgs jautājums, kas nosaka veicamos darbus. Ja iespējams, tad ieteicams izmantot rīkus, no kuriem viegli iespējams eksportēt datus vispārpieņemtos formātos (.csv vai .xml u.c.). Ja šis lēmums jau ir pieņemts, tad atliek strādāt ar to, kas ir. Jāapzinās visi izmantotie rīki, kas ģenerēt testu rezultātus projekta ietvaros.

#### 3.2. Kādas sistēmas tiek izmantotas projektā?

Vai ir problēmziņojumu reģistrācijas sistēma?

Vai ir darba uzdevumu uzskaites sistēma?

Vai ir kāda cita veida sistēma, kas ietekmē projekta gaitu un var ietekmēt testu rezultātus? Piemēram, automatiska projekta būvēšanas sistēma, kas uz noteiktu laiku aptur testēšanas serveri.

No šīm sistēmām iespējams importēt datus Krabis sistēmā, ja ir tāda nepieciešamība.

#### 3.3. Kādas atskaites ir pieejamas elektroniskā veidā no iepriekš minētajiem rīkiem un sistēmām?

Lai varētu nodrošināt regulāru datu plūsmu, svarīgi zināt kādus datus iespējams iegūt no identificētajām sistēmām. Vairums sistēmu un rīku raksta vismaz kaut kāda veida žurnāla failus.

Svarīgi arī noskaidrot vai ir iespējams atskaišu datus iegūti programmatiski (caur API)? Šis ir nozīmīgs posms projekta organizācijā. Ja datus iespējams iegūt programmatiski, tad varam rēķināties ar vairāk automatizētu sistēmu, salīdzinot ar situāciju, kad datus iespējams iegūt tikai ar cilvēka manuālu līdzdarbību.

### **3.4. Cik bieži iespējams iegūt informāciju no rīkiem?**

Labākajā gadījumā varam iegūt datus uzreiz pēc testu veikšanas. Mēdz būt projekti, kur datus nav iespējams iegūt uzreiz. Svarīgi apzināt datu plūsmas un intervālus, kuros dati tiek atjaunināti.

Kad tas noskaidrots, var izdarīt secinājumus par organizatoriskiem jautājumiem projektā.

### **3.5. Cik bieži plānots mainīt testu aprakstu informāciju?**

Atkarībā no projekta specifikas, un fāzes kurā projekts atrodas, mēdz būt dažādas situācijas saistībā ar testu pamat datiem- testu aprakstiem, grupām un savstarpējo atkarību. Tālāk uzskaitītas vairākas situācijas, kas var palīdzēt saprast šo jautājumu.

- Projekti, kur testu pamat dati ir definēti un nemainās, vai mainās ļoti reti. Situācija biežāk sastopama projektiem uzturēšanas fāzē. Projekts tiek uzturēts, bet tiek minimāli papildināts. Līdz ar to arī testu pamat dati mainās ļoti maz.
- Projekti, kur testu pamat dati mainās bieži, tiek regulāri papildināti. Piemēram, tipiski izstrādes projekti agrā stadijā, kur tiek regulāri izstrādāta jauna funkcionalitāte un jauni testu piemēri.
- Projekti, kur testu piemēri ir nemainīgi, bet to sadalījums grupās regulāri mainās. Tas nozīmē, ka tiek mainīta tieši kārtībā, kādā testi tiek izpildīti.

Kad šis jautājums ir noskaidrots, ir labs pamats, lai pieņemtu informētu lēmumu par to kā organizēt spraudņa darbību. Ja vajadzīgs, tad spraudni var paredzēt tādu, ka tas maina testu un testu grupu pamat datus sistēmā.

## 4. Spraudņa realizācija ar programmas palīdzību.

Viens no spraudņa realizācijas veidiem ir programma. Tas nozīmē no Krabis neatkarīga programma, kas iegūst informāciju no ārējiem avotiem (atskaitēm, API u.c.), tad pieslēdzas Krabis datu bāzei un ieraksta datus tabulās.

Šī spraudņa programma var būt realizēta jebkurā programmēšanas valodā, ar nosacījumu, ka šī valoda atbalsta iespēju pieslēgties Oracle datu bāzei.

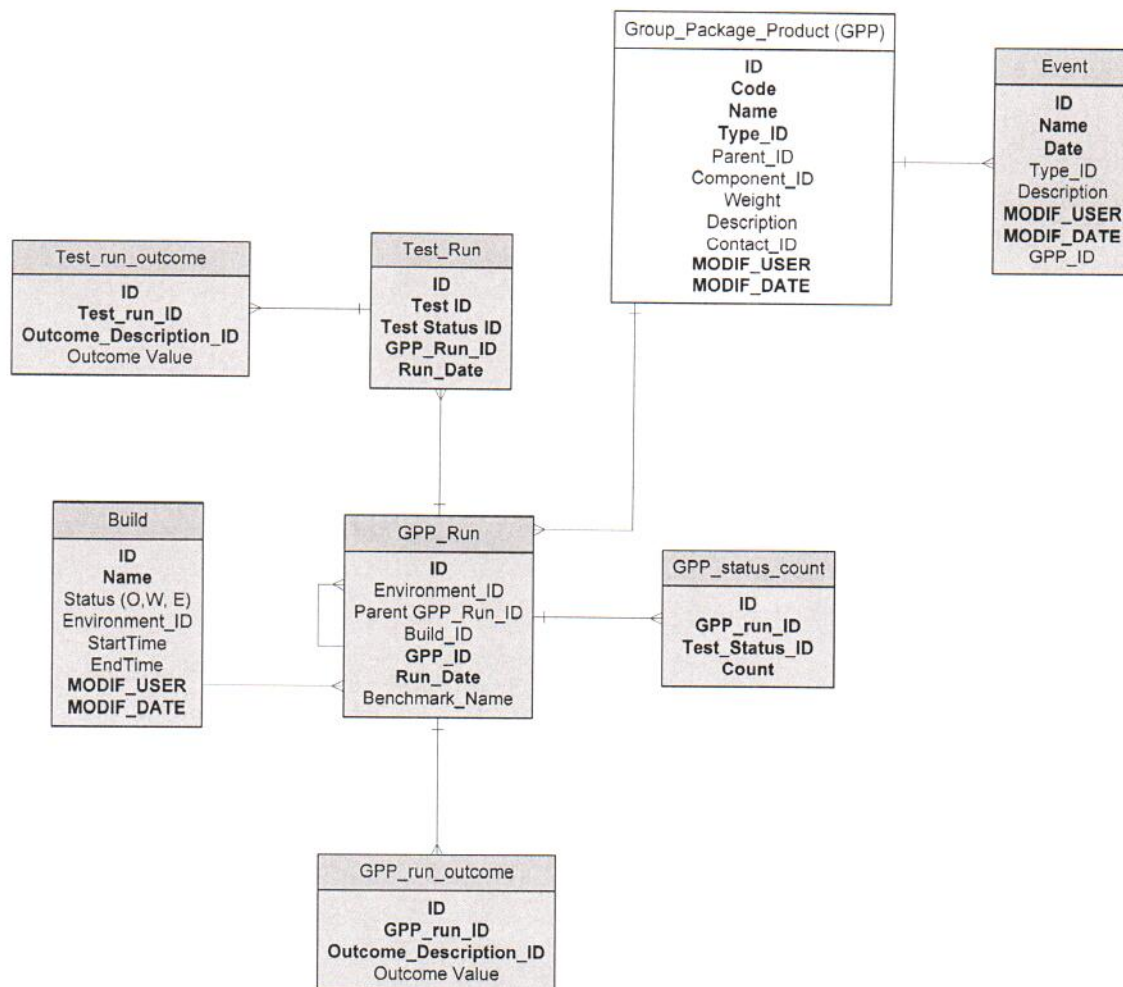
### 4.1. Testa laidiena datu aizpilde

Tipiskā gadījumā spraudnis aizpilda datus tikai testa laidienā tabulās, bet netiek mainīti vai papildināti dati testu pamat datu tabulās. Ja importējamajos datos novērota neatbilstība ar testu pamatdatiem (piemēram jauns testa piemērs), tad programma brīdina lietotāju par šo situāciju, un lietotājam šī situācija ir jāizlabo, lai varētu pabeigt datu importu.

Attēlā numur 1 norādītas tabulas, kas tiek aizpildītas tipiskā gadījumā.



## Programmatūras izstrādes rezultātu apstrādes un interpretācijas tehnoloģija



Attēls 1: Testu laidieņu datu tabulas attēlotas pelēkā krāsā

Visi dati pakārtoti tabulas Group\_Package\_Product (GPP) ierakstam, kas ir struktūras “saknes” elements.

Ar GPP ierakstu tiek saistīti pakārtotie GPP ieraksti un GPP\_Run ieraksti.

Tabulā GPP\_Run tiek saglabāti dati par vienu GPP laidieņu. Ar tabulu GPP\_Run tiek saistīti dati no tabulām Test\_Run (Testa laidieņš) un GPP\_Run\_outcome (GPP laidieņa rezultāts).

Tabula GPP\_Run satur norādi uz tabulu Build (Būvējumi).

Tabula Test\_run\_outcome saistīta ar tabulu Test\_run.



## Programmatūras izstrādes rezultātu apstrādes un interpretācijas tehnoloģija

Tabula Event paredzēta informācijai par dažādiem notikumiem un laiku, kurā tie notikuši, piemēram kad atjaunotas aplikāciju serveru versijas. Šie notikumi var palīdzēt testēšanas rezultātu analīzes procesā.

Sīkāku informāciju par datu bāzes tabulu saturu var atrast [1].

### **4.2. Testa pamat datu aizpilde.**

Praksē mēdz būt projekti, kur testu pamat datu situācija ir mainīga, piemēram, bieži izstrādāti jauni test piemēri, vai bieži mainās testēšanas vides. Ar spraudņa palīdzību iespējams šo informāciju ierakstīt Krabis datu bāzē. Tas nozīmē, ka Krabis lietotājam nebūs šī pamat informācija jāievada, “rokas” režīmā, un tā būs pieejama Krabis sistēmā pēc spraudņa izpildes.

Piezīme. Šādos spraudņa realizācijas gadījumos vēlams informēt sistēmas lietotājus, par pamatdatu ierakstīšanas faktu, piemēram, ierakstot ierakstu Krabis žurnāla (Log) tabulās.

Piemēram, ja importējot testu rezultātu datus tiek konstatēts, ka Krabis tabulās nav atrodams testa piemērs “X”, tad šis testa piemērs tiek ierakstīts Krabis datu bāzēt attiecīgajā tabulā. Informatīvs ieraksts tiek ierakstīts Log tabulā.

Attēlā nr. 2 redzamas Krabis pamat datu tabulas kopā ar laidiena datu tabulām (pelēkā krāsā).

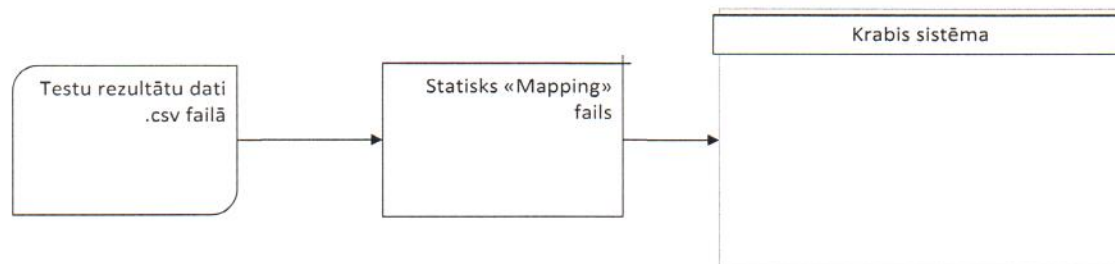


Programmatūras izstrādes rezultātu apstrādes un interpretācijas tehnoloģija

Piemēram, ja nav zināms testu izpildes rezultāts, tad var ieviest jaunu rezultātu “unknown” un saistīt rezultātus ar šo vērtību.

## 5. Spraudņa realizācija ar datu “mappingu”.

Cits variants kā ielādēt datus Krabis rīkā, ir izmantojot .csv datu failus. Shēma attēlota attēlā numur 3.



Attēls 3: Spraudņa realizācija ar datu “mappingu”.

Šāds datu importa mehānisms var krietni atvieglot datu ielādi relatīvi vienkāršās situācijās. Šis mehānisms ir praktiski ērts projektos, kur testu rezultātu datus iespējams iegūt .csv faila formātā. Krabis piedāvā iespēju importēt datus no .csv faila, izmantojot “Mapping” faila palīdzību.

Mapping fails realizēt loģisko saiti starp testu rezultāta .csv failu un Krabis tabulām.

Šī datu ielādes mehānisma priekšrocība ir tāda, ka spraudnis nemaz nav jāprogrammē, jo viss vajadzīgais jau ir realizēts Krabis rīkā.

Vajadzīgās lietas, lai varētu importēt datus ar mapping faila palīdzību ir:

- nodefinēts datu avots;
- nodefinēts datu mappings (datu atbilstība importa faila ar Krabis datu bāzi).



## 6. Datu importa realizācija ar sagataves failu.

Vienkāršākais veids kā importēt testu rezultātu datus Krabis sistēmā ir izmantojot sagataves failu. Tas ir .xls vai .csv datu fails ar definētu struktūru. Attēlā numur 4 redzams datu ielādes sagataves fails.

	A	B	C	D
1	Test_ID	Result	Start time	End time
2	Test 1	passed	2015-03-02 12:00:00	2015-03-02 12:01:00
3	Test 2	failed	2015-03-02 12:02:00	2015-03-02 12:05:00
4	Test 3	passed	2015-03-02 12:06:00	2015-03-02 12:10:00
5				
6				
7				
8				
9				

Attēls 4: Datu ielādes sagataves fails

Sagataves fails paredzēts ātrai Krabis sistēmas iedarbināšanai. Tas ir vienkāršākais veids kā ielādēt datus projekta uzsākšanas fāzē, kad testu dati jau ir pieejami, bet spraudnis vai datu mappings vēl nav izstrādāts.

Sagataves fails jāielādē no Krabis sistēmas lietotāja saskarnes, izvēloties GPP, ar kuru šie testu rezultāti saistāmi.

## **7. Datu ielādes risinājuma optimizācija.**

Plānojot datu importa risinājumu, svarīgs uzdevums ir apzināt datu apjomu, kas tiks apstrādāts katrā datu ielādes iterācijā. Importa programmas tehniskais risinājums jāpielāgo datu apjomam.

Piemēram, ja importējamo datu apjoms ir relatīvi neliels katrā importa iterācijā, un visus datus iespējams ielādēt operatīvajā atmiņā (RAM), tad risinājums ir relatīvi vienkāršs-iespējams ielasīt visus datus programmas operatīvajā atmiņā un tad tos apstrādāt.

Gadījumos, kad datu apjoms ir liels, un nav iespējams visus iterācijas datus ielādēt operatīvajā atmiņā, risinājums ir jāveido tā, lai datus varētu ielādēt pa daļām.

## 8. Rezultāti

Aktivitātes ietvaros ir izstrādāta spraudņa izstrādes metodika.

## 9. Literatūras saraksts

[1] Projekta aktivitātes Nr.1.1 "Loģiskais datu modelis" progresu pārskats