



## 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.augusts.

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.2.1 "Testu grupas divu rezultātu salīdzināšanas algoritms" progresa pārskats**

Pārskats Nr.11 par periodu no 2014.gada 1.septembra līdz 2015.gada 28.februārim.

## SATURS

1.	Kopsavilkums .....	3
2.	Ievads .....	4
3.	Testu grupas divu rezultātu salīdzināšanas algoritms.....	5
3.1.	Situācijas formulējums.....	5
3.2.	Testu rezultātu kopa.....	5
3.3.	Testu rezultātu svari.....	5
3.4.	Testu rezultātu salīdzināšana.....	6
3.5.	Testu rezultātu datu realizācija Krabis.....	7
4.	Rezultāti .....	9

## 1. Kopsavilkums

Pārskata periodā (2014-09-01 – 2015-02-28) projekta „Programmatūras izstrādes rezultātu apstrādes un interpretācijas tehnoloģija” (Krabis) aktivitātes "Testu grupas divu rezultātu salīdzināšanas algoritms" ietvaros veikti šādi darbi:

1. Iespējamo testa rezultātu analīze.
2. Algoritma izstrāde.
3. Darbs apspriests regulārās projekta sanāksmēs.

## **2. Ievads**

Šajā dokumentā aprakstīts un loģiski definēts algoritms, kas Krabis tehnoloģijā tiek izmantots testu rezultātu salīdzināšanai.

### 3. Testu grupas divu rezultātu salīdzināšanas algoritms

#### 3.1. Situācijas formulējums

Testu rezultātu salīdzināšanai nepieciešami rezultāti no vairākiem testu laidieniem (darbiem) kādai testu grupai. Testu rezultātus var salīdzināt vienas grupas ietvaros. Tas ir vienu šīs grupas laidiena rezultātu var salīdzināt ar citu šīs grupas laidiena rezultātu. Katrs laidiens atšķiras no cita laidiena vai nu ar laiku, kad tas ir noticis, vai arī ar vides parametriem, kurā šis laidiens ir izpildīts. No Krabis viedokļa reti iespējami divi pilnīgi identiski testu laidieni.

#### 3.2. Testu rezultātu kopa

Testēšanas laikā katram testam tiek iegūts testa izpildes rezultāts. Vienkāršā gadījumā tas varētu būt tikai pareizs vai nepareizs, bet sistēmā *Krabis* tiek piedāvāta iespēja definēt patvaļīgu skaitu testu rezultātu.

Piemēram, testu rezultātu kopa kādā projektā varētu būt šāda:

- Tests veikts perfekti;
- Tests izdevies, ar mazāk par 3 atkātojumiem;
- Tests izdevies ar 3 līdz 10 atkātojumiem;
- Tests nav izpildīts veiksmīgi;
- Tests nav izpildīts veiksmīgi pie tam ar nepieļaujamu kļūdu;
- Tests nav izpildīts veiksmīgi ar sekām testēšanas vidē.

#### 3.3. Testu rezultātu svari

Daudz testu rezultātu ļauj precīzāk aprakstīt testa izpildes laikā notikušo un/vai iegūto rezultātu. Lai būtu iespējams masveidā apstrādāt, savstarpēji salīdzināt un automātiski interpretēt patvaļīgu daudzumu testu rezultātu, katrs iespējamais testa rezultāts tiek raksturots ar kvantitatīvu lielumu – skaitli. Šo lielumu sauc par *testa rezultāta svaru*.

Uzskatīsim, ka jebkura testa rezultāta svars ir vesels skaitlis. Pozitīvi skaitļi tiek lietoti vēlamiem, sekmīgiem, pareiziem, sagaidāmiem, citādi „pozitīviem” testa izpildes rezultātu svaram, bet negatīvi - nevēlamiem, nesekmīgiem, nepareiziem, citādi „negatīviem” testa izpildes rezultātiem. Nulles vērtība parasti tiek rezervēta „neitrāliem” testu izpildes rezultātiem. Piemēram, testa rezultāta svara vērtība  $s_i > 0$  norāda, ka tests ir beidzies veiksmīgi un izpildīts pareizi, bet  $s_i < 0$  norāda, ka tests ir beidzies neveiksmīgi vai izpildīts nepareizi. Jo lielāka ir testa rezultāta svara absolūtā vērtība  $|s_i|$ , jo nozīmīgāks

## Programmatūras izstrādes rezultātu apstrādes un interpretācijas tehnoloģija

ir šī testa izpildes rezultāts (grupas, projekta, utt. ietvaros) un tam būs lielāka ietekme turpmākajos aprēķinos, kas tiks veikti, lai iegūtu konsolidētos rezultātus.

Ar dažādām testa rezultāta svaru vērtībām var raksturot arī sarežģītākus gadījumus. Piemēram, var šķirot gadījumus, ka testa izpilde nav vai ir beigusies korekti (ar vai bez izpildes laika kļūdas) un testa rezultāta atbilstība sagaidāmajam (atbilst/neatbilst). Piemēram,  $s_i = -1$  gadījumos, ka programma ir darbu beigusies korekti, bet rezultāts ir nepareizs (neatbilst sagaidāmajam), bet  $s_i = -5$  gadījumos, ka programma ir beigusies darbu ar izpildes laika kļūdu un pārtraukusi testa izpildi, neiegūstot pārbaudāmu rezultātu. Abos gadījumos testa rezultāta svars ir negatīvs, bet to nozīmīgums ir atšķirīgs - otrais rezultāts ir ievērojami sliktāks nekā pirmais. Atkarībā no konkrētā projekta specifikas, rezultātu savstarpējā „labuma” vai „sliktuma” kritēriji (un, tāpat testu rezultātu svaru vērtības) var būtiski atšķirties.

Iepriekš minētajai testu rezultātu kopai šajā piemērā saistīta ar svariem:

- Tests veikts perfekti.  $s = 10$ ;
- Tests izdevies, ar mazāk par 3 atkārtojumiem.  $s = 5$ ;
- Tests izdevies ar 3 līdz 10 atkārtojumiem.  $s = 1$ ;
- Tests nav izpildīts veiksmīgi.  $s = -5$ ;
- Tests nav izpildīts veiksmīgi pie tam ar nepieļaujamu kļūdu.  $s = -10$ ;
- Tests nav izpildīts veiksmīgi ar sekām testēšanas vidē.  $s = -10$ ;

### 3.4. Testu rezultātu salīdzināšana

Izpildot testu  $t_i$  vienā izpildes vidē vairākkārt, iespējams salīdzināt tā svaru katrā no izpildes reizēm (testa laidieniem). Ja testa  $t_i$  rezultāta svars divos testa laidienos (hronoloģiski augošā secībā) ir attiecīgi  $s_{i1}$  un  $s_{i2}$ , tad teiksim, ka testa rezultāts ir *uzlabojies*, ja  $s_{i2} > s_{i1}$ , vai *pasliktinājies*, ja  $s_{i2} < s_{i1}$ . Ja  $s_{i1} = s_{i2}$ , tad testa rezultāti *nav atšķirami*. Tas nenozīmē, ka saturīgie testa rezultāti ir vienādi (piemēram, izpildes laikā var būt bijusi atšķirīga kļūda, bet testa rezultāta svars abos gadījumos var būt vienāds). Tikai ārkārtīgi detalizēti izstrādātās testu rezultātu svaru sistēmās (kur katrai situācijai ir garantēti atšķirīga testa rezultāta svara vērtība) var apgalvot, ka no testu rezultātu svaru atšķirības seko arī testu rezultātu savstarpējā atšķirība.

Sarežģītākos gadījumos testa rezultāta svars var ietvert papildus informāciju. Piemēram, ar noteiktām vērtībām vai vērtību diapazoniem var kodēt izņēmumgadījumus. Šādos gadījumos šīs “īpašās” skaitliskās vērtības nedrīkst izmantot vienkāršās salīdzināšanās un divu testa rezultāta svaru salīdzināšanai jāizmanto sarežģītāka funkcija  $\sigma(s_{i1}, s_{i2})$ , kur  $\sigma(s_{i1}, s_{i2}) > 0$  nozīmē, ka testa rezultāts ir uzlabojies,  $\sigma(s_{i1}, s_{i2}) < 0$  – ka pasliktinājies, bet  $\sigma(s_{i1}, s_{i2}) = 0$ , ka rezultāti nav atšķirami.

### **3.5. Testu rezultātu datu realizācija Krabis**

Attēlā numur 1 attēlotas Krabis datu loģiskais modelis, kurās tiek glabāti testu rezultāti un to svāri.

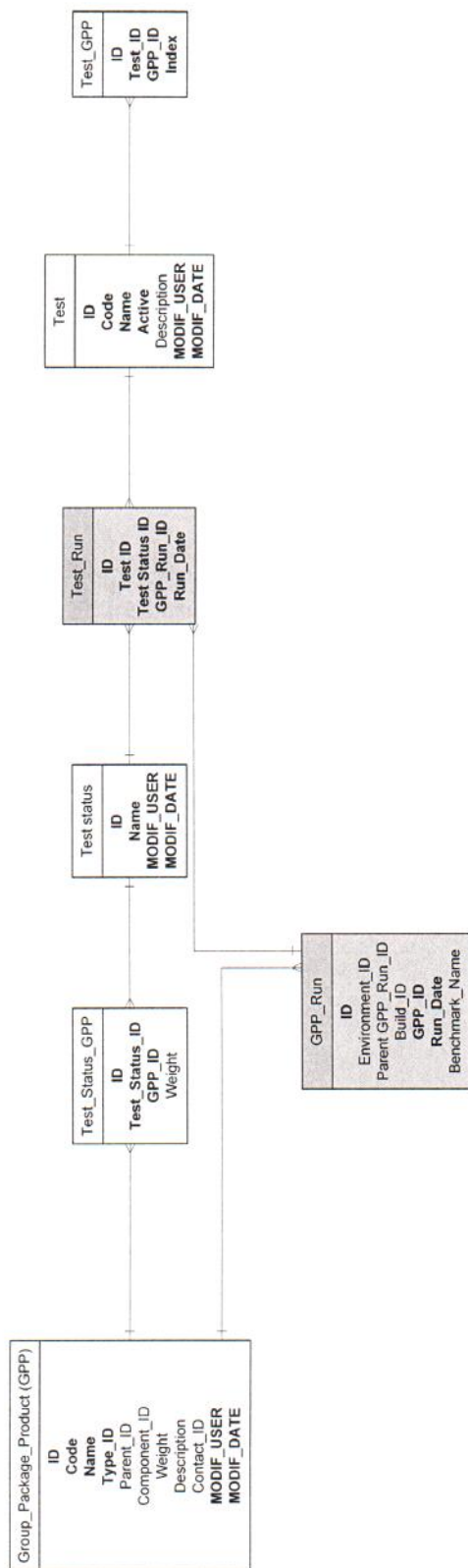
Tabulas Testa laidiens (Test\_Run) un Sadaļas laidiens (GPP\_Run) satur informāciju par laistiem testiem un to grupām.

Tabula Test\_Run satur norādi uz tabulu Test status, kas apzīmē rezultātu, kāds ir iegūts, izpildot šo testu. Tabula Test\_Status\_GPP satur attiecīgā testa rezultāta svaru šajā grupā. Šāds risinājums ļauj vienam konkrētam testu rezultātam piedalīties vairākās grupās ar dažādiem rezultātu svāriem.

Šos rezultātu datus tālāk iespējams salīdzināt kā minēts šajā dokumentā.

Detalizēta informācija par datu bāzes laukiem un to nozīmi atrodama datu bāzes dizaina dokumentā.

Programmatūras izstrādes rezultātu apstrādes un interpretācijas tehnoloģija



Attēls 1: Krabis tabulas testu rezultātu glabāšanai



## **4. Rezultāti**

Aktivitātes ietvaros ir izstrādāts algoritms, kas Krabis tehnoloģijā tiek izmantots testu rezultātu salīdzināšanai.