

# TESTA GADĪJUMU VADĪBAS SISTĒMAS IZVĒLE UN IEVIEŠANA

Jurijs Grigorjevs  
“TietoEnator” SIA  
Tel.: 7286660

[Jurijs.Grigorjevs@tietoenator.com](mailto:Jurijs.Grigorjevs@tietoenator.com)

## Prezentācijas mērķis

Referāts ir veltīts testa gadījumu vadības sistēmas (Test Case Management System) ieviešanai. Referāta mērķis ir iepazīstināt auditoriju ar realizētiem ieviešanas posmiem, iekļaujot problēmas identificēšanu, prasību definēšanu, līdzīgu rīku analīzi un izvērtēšanu un beidzot ar rīka pielāgošanu un ieviešanu. Šī procesa rezultātā tika ieviests un ikdienā tiek ekspluatēts QaTraq testēšanas rīks ar paplašinātu Pro licenci.

## Ieskats izstrādes vidē

Viens no svarīgākiem faktoriem ir testējamās sistēmas specifika. Vairāk kā 15 gadu laikā tiek izstrādāta un attīstīta maksājuma karšu sistēma CardSuite Unix (agrāk Transmaster). Pa šo laiku klientu instalāciju skaits ir izaudzis un ir mērāms desmitos. Šis aspekts veicināja programmas koda sazarošanu un ar to saistītām problēmām, kas savukārt apgrūtina regulāro versiju pārtestēšanu un palielina testēšanas apjomus. Sistēmai attīstoties, parādās jaunās funkcionalitātes un izmaiņas esošajā, kas prasa regulāro dokumentācijas atjaunošanu un tās aktuālā stāvokļa uzturēšanu.

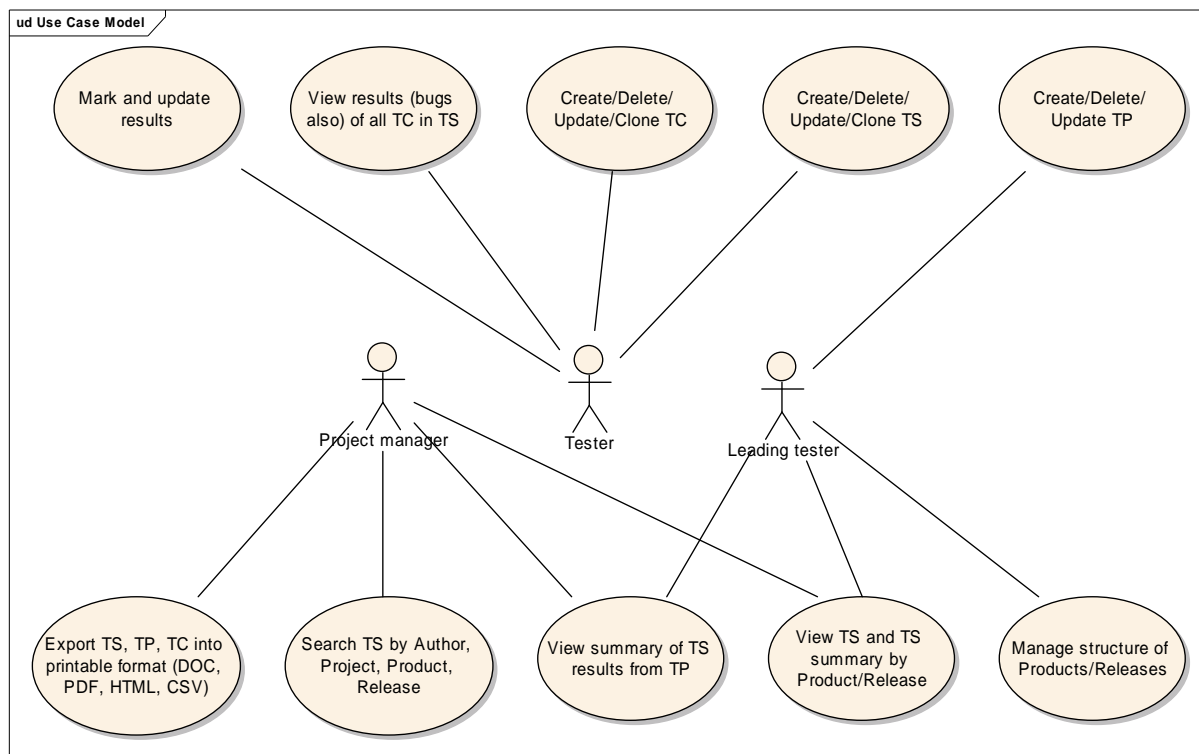
Vēl viens svarīgs faktors ir jauno cilvēku plūsma. jaunie darbinieki ir jāapmāca biznesa un tehniskām zināšanām, kā arī ir jāiemāca izstrādājamā produkta pamata principiem un produkta funkcionalitātei. Šis, savukārt, prasa aktuālas informācijas uzturēšanu ne tikai gala lietotājam, bet arī sistēmas izstrādātājam. Tieši tāpēc labi strukturēti un aprakstīti testu scenāriji var kalpot par sistēmas dokumentāciju un par svarīgo programmatūras sastāvdaļu. Sākotnēji uzņēmumā testu scenāriji tika vadīti HTML un DOC formātos, bet biežu izmaiņu, papildinājumu un regulāro versiju dēļ, bija nepieciešams šos dokumentus kopēt un vadīt versijas atsevišķi. Šādā testu scenāriju vadība radīja problēmas dokumentu atsekošanā, vajadzīgo dokumentu meklēšanā un izstrādāto dokumentu atkārtotā lietošanā.

## TCMS izvēle

Prasību definēšanai TCMS sistēmai bija izveidota ekspertu grupa no 6 cilvēkiem, kas sastāvēja no testēšanas ekspertiem, projektu vadītāja un citiem izstrādes ekspertiem. Tas bija nepieciešams, lai definētu prasības no dažādām lomām. Kā pamata prasības testa scenāriju vadības sistēmai tika izvirzītas sekojošas:

- testa gadījumu un testa scenāriju atkārtota lietošana no vienotas vietas;
- testa gadījumu un testa scenāriju versiju vadība;
- plašas meklēšanas iespējas;
- rīka integrēšana ar esošām iekšējām sistēmām, cena un pielāgošanas iespējas.

Tas bija galvenās prasības, uz kurām bija fokusēts, izvēloties jaunu sistēmu. Definējot detalizētākas prasības, tika apskatīta sistēmu funkcionēšana no vairāku lomu lietošanas viedokļa. Tika analizētas 3 lomas: Testētājs (Tester), Vadošais testētājs (Leading tester) un Projektu vadītājs (Project manager). Kā prasības TCMS sistēmai tika nedefinētie lietošanas gadījumi, kas ir nepieciešami testēšanas procesa realizēšanai un tie ir parādīti nākamajā attēlā.



1. attēls. Lietošanas gadījumi no 3 lomām

Tester loma ir domāta testēšanas gadījumu izpildītājam, kuram tiek dots testēšanas uzdevums. Leading tester pēc būtības nodarbojas ar testēšanas darbu plānošanu, testēšanas veidu definēšanu un testēšanas procesa atsekošanu. Tam ir būtiski sekot līdz izpildītiem un paliktiem testēšanas gadījumiem. Project manager loma nedaudz pārklājas ar iepriekšējo lomu pielietojot vēl meklēšanas un eksporta iespējas.

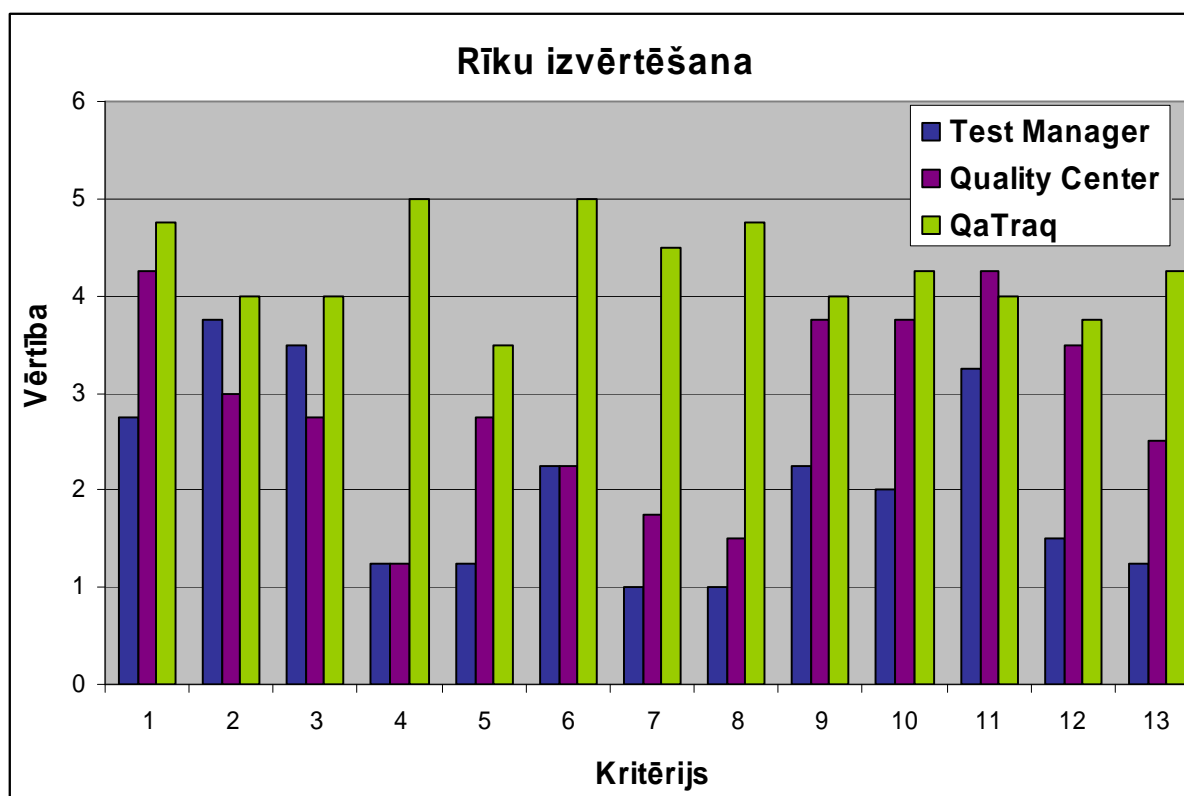
Lai aprakstītu testēšanas gadījumu (TC) un testēšanas scenāriju (TS) saturu, tika definēti sekojošie aprakstošie aspekti:

- TC: Title, Product/Component, Set of preconditions, Content, Test steps, Expected results vs Depending Parameters, Author and Change history.
- TS: Title, Short description, Product/Component, Release, Set of tested components, Test environment, Status (draft, reviewed), Tester, Set with TC.

Izejot no definētām lomām un aprakstošām īpašībām, tika sastādīta kritēriju kopa, pēc kuras arī notika izvēlēto rīku izvērtēšana. Tika nedefinēti sekojošie kritēriji:

1. Rīka ieviešana uzņēmumā. Cik ērts ir rīks no lietojamības, pieejamības (web, client-server), pieeju tiesības kontroles viedokļiem?
2. Cik labi rīks ir spējīgs nodrošināt TC aprakstītus aspektus?
3. Cik labi rīks ir spējīgs nodrošināt TS aprakstītus aspektus?
4. Vai ir iespējams realizēt/iemigrēt/lietot no cita resursa TE esošo moduļu/produktu/komponenšu struktūru un versijas?
5. Vai meklēšanas iespējas nodrošina prasītus kritērijus?
6. Vai rīks nodrošina atkārtoti lietot TC/TS?
7. Vai rīks ir integrējams ar JIRA uzdevumu vadīšanas sistēmu?
8. Vai rīkā ir realizēta TC/TS/TP versiju kontrole ar pieeju pie dažādām versijām?
9. Vai ir iespēja pielikt rezultātiem papildus failus?
10. Cik rīks ir lietojams no TESTER lomas viedokļa?
11. Cik rīks ir lietojams no TEST LEADER lomas viedokļa?
12. Cik rīks ir lietojams no PROJECT MANAGER lomas viedokļa?
13. Cik lielā mērā rīks nodrošina definētās nefunkcionālas prasības?

Rīku izvērtēšanas rezultāti ir atspoguļoti 2. attēlā.



*2. attēls. Rīku izvērtēšanas rezultāts*

Rezultāts parādīja, ka konkrēti šim uzstādītam mērķim QaTraq ir piemērotāks variants. Abi pārējie rīki ir visai spēcīgi un sevī ietver ne tikai TCMS, bet arī defektu atsekošanas sistēmu un vairākas testēšanas gadījumu automatizēšanas iespējas. Ņemot vērā, ka uzņēmumā veiksmīgi darbojas pašu rakstīts testēšanas automatizēšanas karkass (TAF) un integrētā defektu vadīšanas sistēma, tad arī nebija nepieciešamības pēc tik jaudīgiem rīkiem kā Test manager un Quality center.

QaTraq ir bezmaksas atvērta koda produkts ar iespēju pašiem modificēt un pielāgot to savām vajadzībām. Savukārt, lai nodrošinātu eksporta, atskaišu iespējas, papildus tika iegādāta QaTraq Pro pievienojumprogramma, kas arī nodrošina šīs iespējas.

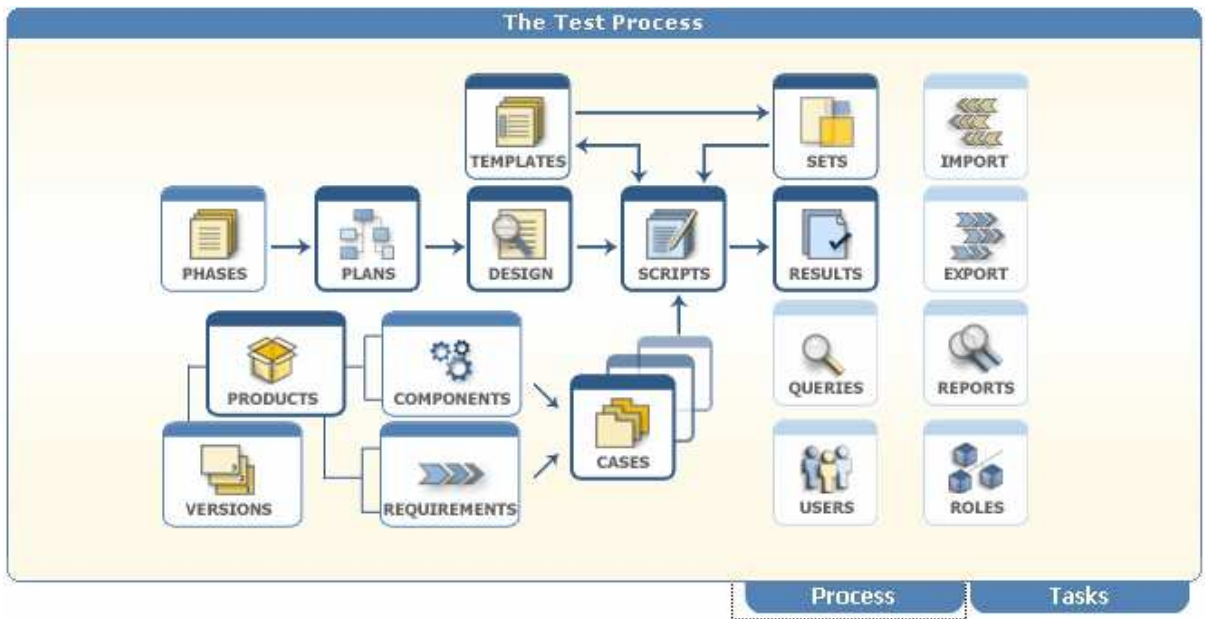
## TCMS ieviešana

QaTraq ieviešanas process notika 2 fāzēs: izmēģinājuma un pilnas ekspluatācijas posmi. Izmēģinājumā fāzē QaTraq apzināti tika iedots lietošanai 2 jauniem testētājiem, kuri nav strādājuši ar veciem scenārijiem un kas varēja sniegt objektīvus komentārus par sistēmas lietošanas ērtumu un iespējām. Rezultātā svarīgākie ieteikumi bija ņemti vērā un realizēti pirms reālas ekspluatēšanas.

QaTraq Pro pievienojumprogramma prasa licenzēšanu uz paralēliem vienlaicīgiem pieslēgumiem (informāciju par izmaksām var dabūt [www.testmanagement.com/](http://www.testmanagement.com/) mājas lapā).

Pieejas tiesību kontrole ir integrējama ar LDAP servisu. QaTraq pēc noklusējuma piedāvā sekojošās pieejas lomas: Administrator, Test Analyst, Test Manager, Test Team Lead un Tester. Atkarībā no izvēlētās lomas (vienam lietotājam var būt vairākas lomas), lietotājiem var ierobežot pieeju pie QaTraq objektiem: izveidot jaunus, atjaunot, dzēst, kopēt un citas darbības.

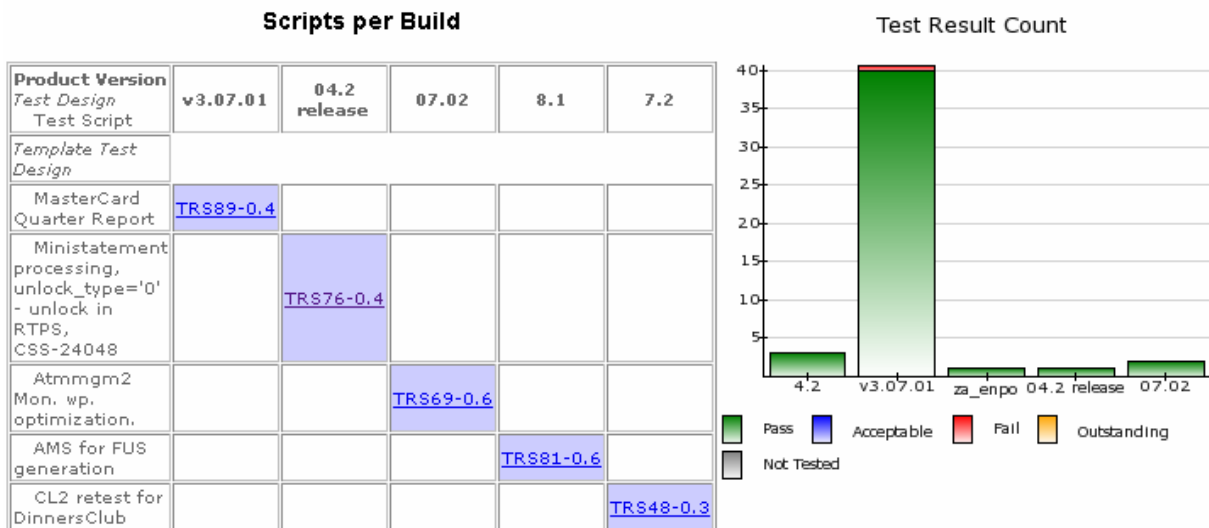
QaTraq struktūra un galvenie darbības principi ir parādīti 3. attēlā.



3. attēls. QaTraq struktūra

QaTraq galvenā priekšrocība salīdzinājumā ar pārējām apskatītām sistēmām ir testēšanas gadījumu ērtās atkārtotas lietošanas iespējas. Testēšanas gadījumi bez izpildes rezultātiem sastāda sava veida krātuvi, no kuras tie var tikt iekļauti testēšanas scenārijos. Tas ļauj izvairīties no nevajadzīgas kopēšanas un dabūt informāciju par katra testēšanas gadījuma iekļaušanu testēšanas scenārijos – vēsture ar rezultātiem, kuros scenārijos konkrēts testēšanas gadījumus bija iekļauts.

QaTraq nodrošina arī grafiskās atskaites, turklāt vaicājumus atskaitēm ir iespējams definēt ar PL/Sql kodu.



4. attēls. Grafiskās atskaites

Pār pārējo QaTraq un QaTraq Pro funkcionalitāti var noskaidrot no tīmekļa resursiem [1], [2], [3] un citiem.

## Secinājumi

Šāda pusformālā pieeja ļāva izanalizēt un gala rezultātā izvēlēties piemērotāku TCMS sistēmu, kas tika ieviesta u veiksmīgi tiek lietota ikdienā. Līdz ar jebkādas jaunās sistēmas ieviešanu nāk priekšrocības un trūkumi. Ar sistēmas ieviešanu galvenās iegūtas priekšrocības ir: testēšanas gadījumu atkārtotā lietošana (it īpaši šis faktors ir efektīvs, ja ir nemainīgā funkcionalitāte pie vairākiem klientiem un dažādām konfigurācijām), testēšanas procesa atsekošana (izpildīto testēšanas gadījumu apjoms), meklēšanas iespējas, pieeja caur tīmekli. Savukārt var minēt arī šādus trūkumus: sistēmas darbības ir atkarīga no testēšanas gadījumu nosaukšanas, grupēšanas pa komponentēm un to saturu, jo būtiskākais ir atrast vajadzīgo testēšanas gadījumu starp simtiem citu. Šis faktors ir ļoti aktuāls CardSuite Unix sistēmai un tas tika atrisināts ar atbildīgu personu izdalīšanu katram produktam. Šāda nozīmēšana ir nepieciešama, lai administrētu produktu struktūru un atsekot testēšanas gadījumu izveidošanu un piederīgu definētām komponentēm.

Rakstā aprakstītājs rīku salīdzinājums ir balstīts uz šauru kritēriju kopu, kas tika izvirzīta konkrētas problēmas risināšanai, un nevar tikt uzskatīts par vispārīgo rīku salīdzinājumu.

## Atsauces

1. <http://sourceforge.net/projects/qatraq/> : QaTraq.
2. <http://demo.traqhost.com/> : demo of QaTraq.
3. [www.testmanagement.com/](http://www.testmanagement.com/) : Software test case management. QaTraq Pro.