

hyväksymispäivä

arvosana

arvostelija

## Jatkuva integraatio ja regressiotestaaminen

Piia Hartikka

Helsinki 20.9.2017

Aine

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

# Sisältö

<b>1</b>	<b>Johdanto</b>	<b>1</b>
<b>2</b>	<b>Regressiotestaamisen merkitys jatkuvassa integraatiossa</b>	<b>3</b>
2.1	Luottamus ohjelmiston toimivuuteen . . . . .	3
2.2	Testikokonaisuuden valinta . . . . .	5
2.3	Regressiotestit jatkuvassa muutoksessa . . . . .	7
	<b>Lähteet</b>	<b>8</b>

# 1 Johdanto

Jatkuva integraatio on työskentelytapa, jossa ohjelmistokehittäjä integroi tuottamansa koodia jatkuvasti järjestelmään. Näin ohjelmiston uusien versio on aina tarjolla ja se toimii. Ohjelmakoodi sisältää automatisoidut testit, jotka ajetaan säännöllisesti testaukseen varatulla palvelimella. Se lisää projektin läpinäkyvyyttä ja tietoa ohjelmiston ajankohtaisesta kehitysvaiheesta. Jatkuvan integraation tarkoitus on korvata perinteisen ohjelmistokehityksen päättävä integraatiovaihe.

Regressiotestaaminen on nimitys toiminnoille, joilla testataan ohjelmiston oikeellisuutta muutosten jälkeen. Sen voi toteuttaa esimerkiksi regressiotesteillä. Regressiotestit ovat testikokonaisuus, joka voi sisältää yksikkö-, integraatio- ja järjestelmätestejä.

Tutkielmassa tarkastellaan jatkuvan integraation ja regressiotestaamisen suhdetta. Miten ne vaikuttavat toisiinsa ja miten niitä voisi kehittää.

Luvussa kaksi käsitellään regressiotestaamisen merkitystä jatkuvassa integraatiossa. Luvussa kolme käydään läpi regressiotestaamisen kehittämistä ja sen seurauksia jatkuvan integraation toteuttamiselle. Luvussa neljä esitetään johtopäätökset.



## 2 Regressiotestaamisen merkitys

### 2.1 Luottamus ohjelmiston toimivuuteen

Ohjelmistotuotannossa pyritään tuottamaan ohjelmisto, joka täyttää sille asetetut vaatimukset. Jatkuva integraatio auttaa ohjelmistokehittäjiä saavuttamaan vaatimukset ja tuottamaan enemmän bisnesarvoa asiakkaille. Ohjelmistokehittäjien on helpompi toteuttaa jatkuvaa integraatiota, kun he voivat luottaa ohjelmiston uusimman version toimivuuteen. Jatkevassa integraatiossa ohjelmistokehittäjä kommitoi työnsä, jonka jälkeen erillisellä palvelulla buildataan ohjelmisto ja testataan, että se toimii.



## 2.2 Testikokonaisuuden valinta

Regressiotestien tulee testata ohjelmiston oikeellisuutta ja paljastaa virheet. Siten testikokonaisuuden valintaan tulee kiinnittää erityistä huomiota. Eräs tapa on ajaa kaikki ohjelmiston testit ja testata kaikki testitapaukset. Se vie kuitenkin isossa projektissa liikaa aikaa, kun buildaamisen kesto on rajoitettu jatkuvassa integraatiossa kymmeneen minuuttiin.

Testitapauksia voidaan yhdistää tai niitä voi priorisoida.

Ohjelmistokehitystiimi joutuu päättämään, millä testikokonaisuudella ohjelmiston oikeellisuus voidaan varmistaa.





## 2.3 Regressiotestit jatkuvassa muutoksessa

Ohjelmiston kehittämiseen kuuluu samalla uusien testien kehittäminen. Ohjelmiston kasvaessa regressiotestien joukkoa joudutaan kasvattamaan tai ainakin muuttamaan.

## Lähteet

- dSCJdPB<sup>+</sup>17 de S. Campos Junior, H., de Paiva, C. A., Braga, R., Araújo, M. A. P., David, J. M. N. ja Campos, F., Regression tests provenance data in the continuous software engineering context. New York, NY, USA, 2017, ACM, sivut 10:1–10:6, URL <http://doi.acm.org/10.1145/3128473.3128483>.
- ERP14 Elbaum, S., Rothermel, G. ja Penix, J., Techniques for improving regression testing in continuous integration development environments. New York, NY, USA, 2014, ACM, sivut 235–245, URL <http://doi.acm.org/10.1145/2635868.2635910>.
- Hil16 Hilton, M., Understanding and improving continuous integration. New York, NY, USA, 2016, ACM, sivut 1066–1067, URL <http://doi.acm.org/10.1145/2950290.2983952>.
- LGL<sup>+</sup>16 Li, N., Guo, J., Lei, J., Li, Y., Rao, C. ja Cao, Y., Towards agile testing for railway safetycritical software. New York, NY, USA, 2016, ACM, sivut 18:1–18:4, URL <http://doi.acm.org/10.1145/2962695.2962713>.
- LIH17 Labuschagne, A., Inozemtseva, L. ja Holmes, R., Measuring the cost of regression testing in practice: A study of java projects using continuous integration. New York, NY, USA, 2017, ACM, sivut 821–830, URL <http://doi.acm.org/10.1145/3106237.3106288>.
- MGN<sup>+</sup>17 Memon, A., Gao, Z., Nguyen, B., Dhanda, S., Nickell, E., Siemborski, R. ja Micco, J., Taming googlescale continuous testing. Piscataway, NJ, USA, 2017,

IEEE Press, sivut 233–242, URL [https://doi.org/10.1109/ICSE\-\\$SEIP](https://doi.org/10.1109/ICSE\-$SEIP).  
2017.16.

VW16 Vöst, S. ja Wagner, S., Tracebased test selection to support continuous integration in the automotive industry. New York, NY, USA, 2016, ACM, sivut 34–40, URL <http://doi.acm.org/10.1145/2896941.2896951>.