

hyväksymispäivä

arvosana

arvostelija

Regressiotestaaminen osana jatkuvaa integraatiota

Piia Hartikka

Helsinki 26.9.2017

Kandidaatintutkielma

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

Sisältö

1	Johdanto	1
2	Regressiotestaaminen	3
2.1	Mistä hyöty tulee	3
2.2	Turha testaaminen	3
2.3	Testien kehittyminen	3
2.4	Testituloksien analysointi	3
3	Testikokoelman priorisointi ja rajaaminen	4
3.1	Rajalliset resurssit ja laatutekijät	4
3.2	Testidatan seuraaminen	4
3.3	Joku siisti menetelmä	4
4	Johtopäätökset	5
	Lähteet	6

1 Johdanto

Jatkuva integraatio korvaa perinteisen ohjelmistokehityksen päättävän integraatiovaiheen. Se on työskentelytapa, jossa ohjelmistokehittäjä integroi työtään jatkuvasti ohjelmistokehityksen päälinjaan. Näin ohjelmiston uusi versio on aina tarjolla ja se toimii. Ohjelmisto buildataan ja ohjelmakoodin sisältämät automatisoidut testit ajetaan jatkuvaan integraatioon varatulla palvelimella.

Regressiotestaaminen on nimitys toiminnoille, joilla testataan ohjelmiston oikeellisuutta muutoksen jälkeen. Regressiotestit ovat testikokonaisuus, joka voi sisältää yksikkö-, integraatio- ja järjestelmätestejä. On erilaisia menetelmiä regressiotestaukseen ja oikean testikokonaisuuden valitsemiseen. Regressiotestaamisen onnistuminen on edellytys jatkuvan integraation onnistumiselle.

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. Jatkuvassa integraatiossa ohjelmistokehittäjä kommitoi työnsä, jonka jälkeen erillisellä palvelulla buildataan ohjelmisto ja testataan, että se toimii.

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.

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

Aineessa tarkastellaan regressiotestaamisen merkitystä jatkuvalle integraatiolle.

2 Regressiotestaaminen

2.1 Mistä hyöty tulee

Mitkä ovat ne syyt, jotka tekevät regressiotestaamisesta hyödyllisen?

2.2 Turha testaaminen

Iso osa testeistä menee aina läpi, joten niitä ei kannata ajaa ollenkaan. Kuinka ne tunnistetaan?

2.3 Testien kehittyminen

Ohjelmiston muuttuessa ja kehittyessä myös testikokonaisuus kehittyy. Se ei ole ilmaista.

2.4 Testituloksien analysointi

Mitä ne kertovat? Mitä niiden pitäisi kertoa?

3 Testikokoelman priorisointi ja rajaaminen

3.1 Rajalliset resurssit ja laatutekijät

3.2 Testidatan seuraaminen

3.3 Joku siisti menetelmä

4 Johtopäätökset

Tekstiä.

Lähteet

- dSCJdPB⁺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⁺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⁺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.2017.16](https://doi.org/10.1109/ICSE\-$SEIP.2017.16).
- VW16 Vöst, S. ja Wagner, S., Tracebased test selection to support continuous in-

tegration in the automotive industry. New York, NY, USA, 2016, ACM, sivut 34–40, URL <http://doi.acm.org/10.1145/2896941.2896951>.