Course Program ETSN20 Software Testing

2019 HT2, Per Runeson, last updated 2019-10-29 http://cs.lth.se/etsn20

Aim. The objective of the course is to give basic and advanced knowledge & skills within testing for large-scale development of systems completely or partly based on software. The course gives practical skills in methods and techniques for software testing.

Course elements

- Seven *lectures* give a theoretical overview and help for private studies.
- One *guest lecture* gives a connection to industry practice
- Four *lab sessions* give practical training in applying different test techniques and relate theory to practice through discussions of problems and solutions.
- One *project* gives practical skills and training in different areas of software testing by literature search, analysis and presentation.
- *Written exam* assesses the individual skills.

Assessment

- The project is graded fail / G (pass) / VG (pass with distinction), based on project deliverables. Revised projects are graded pass.
- Approved lab sessions and project presentation are mandatory for passing the course.
- The final grade is fail / 3 / 4 / 5, which is based on the written exam and the project grade, where VG gives 5 extra points, for those who passed the 30 pass limit on the exam.

Literature

Books

K Naik and P Tripathy, *Software Testing and Quality Assurance: Theory and Practice*, Wiley, ISBN: 978-0-471-78911-6, 2008 (available via www.lub.lu.se)

N Matloff and P J Salzman, *The Art of Debugging with GDB, DDD, and Eclipse*. No Starch Press, ISBN: 978-1-59327-174-9, 2008, http://it-ebooks.directory/book-1593271743.html

Articles

V Garousi, M V Mäntylä, A systematic literature review of literature reviews in software testing, Information and Software Technology, 80: 195-216, 2016, doi 10.1016/j.infsof.2016.09.002

J D Hagar, T L.Wissink, D R Kuhn, R Kacker:, Introducing Combinatorial Testing in a Large Organization. IEEE Computer 48(4): 64-72, 2015, doi 10.1109/MC.2015.114

M Beller, A Bacchelli, A Zaidman, and E Juergens, Modern Code Reviews in Open-source Projects: Which Problems Do They Fix?, Proceedings of the 11th Working Conference on Mining Software Repositories (MSR 2014), pp. 202–211, 2014. doi 10.1145/2597073.2597082

C. Sadowski, E. Söderberg, L. Church, M. Sipko, and A. Bacchelli. Modern code review: a case study at Google. IEEE/ACM 40th International Conference on Software Engineering: Software Engineering in Practice Track (ICSE-SEIP), pp. 181–190. ACM, 2018. doi 10.1145/3183519.3183525

H Petersson, T Thelin, P Runeson, C Wohlin, Capture–recapture in software inspections after 10 years research–theory, evaluation and application, Journal of Systems and Software, 72(2):249-264, 2004, doi 10.1016/S0164-1212(03)00090-6.

L Chen, Continuous Delivery: Huge Benefits, but Challenges Too, IEEE Software, 32(2): 50-54, 2015. doi: 10.1109/MS.2015.27

A. Memon et al., Taming Google-scale continuous testing, IEEE/ACM 39th International Conference on Software Engineering: Software Engineering in Practice Track (ICSE-SEIP), pp. 233-242, 2017. doi: 10.1109/ICSE-SEIP.2017.16

M. V. Mäntylä, B. Adams, F. Khomh, E. Engström, and K. Petersen. On rapid releases and software testing: a case study and a semi-systematic literature review. Empirical Software Engineering, 20(5):1384–1425, 2015. doi 10.1007/s10664-014-9338-4

J. A. Whittaker, The 10-Minute Test Plan, IEEE Software, 29(6): 70-77, 2012. doi 10.1109/MS.2012.25

L. Jonsson, M. Borg, D. Broman, K. Sandahl, S. Eldh, and P. Runeson. Automated bug assignment:

Ensemble-based machine learning in large scale industrial contexts. Empirical Software Engineering, 21(4):1579–1585, 2016, doi 10.1007/s10664-015-9401-9

E I Laukkanen and M V Mäntylä, Survey Reproduction of Defect Reporting in Industrial Software Development, International Symposium on Empirical Software Engineering and Measurement, 2011, pp. 197-206, 2011 doi 10.1109/ESEM.2011.28

Articles for the project to be found in the LubSearch database (www.lub.lu.se).

Content

Week	Lecture	Lecture area	Literature	Project	Lab	Lab area
1	L1	Introduction, Unit test	Naik 1, 3, Garousi	Form groups + decide subject by Fri		
	L2	White-box test techniques	Naik 4, 5			
2	L3	Black-box test techniques	Naik 6, 9.2-9.6, Hagar		Lab1	White-box testing
3	L4	Debugging, Reviews, Reliability	Matloff 1, Naik 10.1-10.4, 15, Beller, Sadowski, Petersson	Tue: Deliver outline	Lab2	Black-box testing Report lab 1+2
4	L5	Lifecycle, Continuous testing, Documentation	Naik 7.1-7.4, 12.1-12.9, Chen, Memon, Mäntylä, Whittaker	Meeting w supervisor	Lab3	Debugging No lab report
5	L6	Organization, Tools, Automation	Naik 12.10-12.16, 16.1-16.4, Jonsson		Lab4	Inspection and estimation Lab report
6	L7	Quality, metrics	Naik 13, 17, Laukkanen	Fri: Final report		
7	Guest	Testing in practice		Presentation		

Schedule

	Week	Mo	We	Th	Fr
Nov 4-8	1	L1 10-12 M:E (PR)		L2 8-10 E:C (PR)	
Nov 11-15	2	L3 10-12 E:C (PR)		Lab 1 10-12, 13-15, E:Ravel(SR)	Lab 1 8-10, E:Ravel(SR)
Nov 18-22	3	L4 10-12 E:C (PR)		Lab 2 10-12, 13-15, E:Ravel(RR)	Lab 2 8-10, E:Ravel(RR)
Nov 25-29	4	L5 10-12 M:E (PR)		Lab 3 10-12, 13-15, E:Ravel(RR)	Lab 3 8-10, E:Ravel(RR)
Dec 2-6	5	L6 10-12 M:B (PR)		Lab 4 10-12, 13-15, E:Ravel(SR)	Lab 4 8-10, E:Ravel(SR)
Dec 9-13	6	L7 10-12 E:C (PR)			
Dec 16-20	7	Guest lecture 10-12 E:C			8-12 Project presentations (EE, SR, PR) E:3336
Jan 13-17	Exam		Exam 14-19: MA 10G-H		

Personnel

Per Runeson, PR, per.runeson@cs.lth.se, 046-222 9325, E:2420, Lectures, Projects, Exam
Sergio Rico, SR, sergio.rico@cs.lth.se, Labs, E:2416, Projects
Rasmus Ros, RR, rasmus.ros@cs.lth.se, E:2408, Labs
Emelie Engström, EE, emelie.engstrom@cs.lth.se, E:2418, Projects
Ulrika Templing, ulrika.templing@cs.lth.se, 046-222 80 40, E:2179, Course administration