Matti Vuori, 2014-01-21

Around 80 thoughts about test automation

What thoughts are these? Their purpose is to present the diversity and richness of the world of test automation. Test automation often presented as a big heterogeneous black box that has an upper-case letter A in the lid, but the automation has many kinds of essences. Here we look into some of them (but in no means thoroughly of exactly).

Dream of the automation – dream of machine

1. Automation in the first place is the dream of the industrial society. Manual work is gotten rid of, the software is automatically created and automatically tested.
2. This dream is always present in the testing (particularly in the thinking of the factory school of testing, see Pettichord's old presentation of the schools in www.testingeducation.org/conference/wtst_pettichord_FSofST2.ppt)
3. In the visions, all testing is automatic, nothing else is needed. The automatic tests cover all essential issues and find all the defects.
4. Automation represents progress and competence. It is easy to fall in love with it. It is great for the managers of a company to tell the guests about it.
5. **A machine is an ideal quality controller, "objective" information machine. It is neutral and objective. Exact. Unambiguous. It does not change its opinion. It never lies. It remembers everything. It repeats everything in the same way every time. A machine it is easy to believe …**
6. The machine that knows and decides things brings security to humans' lives! Everyone needs that sometimes.
7. A human being is only needed to dust of the machine sometimes.
8. Automation is the world view of mechanistic, structurally complex, but logical machine. Nowadays it has been begun to notice that the world is not quite of such in its nature.
9. When the human element is removed from the equation, also the best properties of the human being are removed, and not just the bad ones, such as unsuitability to very fast or repetitive work and the need to pay salary.
10. Standardization and perfect homogeneity are connected to the dream. There is no variation in the way machines work. Also the errors of the automation are systematic and we can get rid of them one at a time, soon reaching perfectness.
11. Automation is technical and to technical things there is always associated the idea of best practices of the period, which are brought into use on a shared path of making the world perfect everywhere.
12. Because of it technical nature, automated systems are easy to certify.
13. One characteristic of the dreaming of automation that behind the utopia there is naiveté and ignorance – or a purpose to earn money by selling automation to others. It is natural that the utopia is maintained by the sellers of tools. With very expensive testing tools money is made, and if marketing is based on meanings, the dream of a fully automatic world is a very strong one.
14. It is not rare that there are many defects in the tools in the products created by people who advertise 100 percent test automation… advertise hundred-per-cent automation either.
15. **Yet, from the existence of the dreams and utopian ideas we must not deduce that there would not also be much sense in automation!**

There are many kinds of test automation

16. Test automation is used as a general term but for example automated unit testing and load testing of the information system are very different things. It is important to analyse the different species of automation so that one can understand what the question is about and to react to each one in a right way.
17. The thought of the core of automatic tests can vary: Repetition of planned test cases … trying all variations of an interface … trusting to determinism and planning of details … trusting to chance and to oncoming possibilities.
18. The abstraction level naturally varies: Code, functions … abstract actions, keywords … different models of the program among others state machines … the user's actions … use cases and user stories … business processes … statistical use profiles … etc...
19. Many kinds of things can be automated. The test automation associates with the execution of tests. This holds true for the unit testing and many types of scripted testing. The second basic area is reporting – as “reports” or “radiators". The model based testing advertises the automation of the test design but the test models are not automatically (usually) created at all. It is as important to automate for instance the creation and configuration of test environments or the creation of the test database or other data – in those tasks very much time can be consumed.

How will the automation find bugs?

20. The traditional challenge of test automation is finding new bugs.
21. The ideal of automation is the smooth going of things and therefore also the automatic machine have a “tendency” to be trivial, and positive. In the world there is a huge number of test automation that runs tests that do not have much value.
22. There are natural reasons to this. The makers of the test automation have pressure to make things “work”, just like the writers of the software to be tested. Creating any kinds of tests that run is a significant step, after which there is pressure to start automating something else.
23. The low level integration testing the world of the magic: often fluent integration automation is created and it is imagined that the good testing will come magically along with it automatically [sic.]. But integration is a different thing that integration testing. There is a big difference between good and bad integration testing.
24. The traditional regression testing is performed over a functioning system and with it we can catch the breaking of things when the system is worked on.
25. With unit testing even new bugs are caught, but mostly (in test driven development) there are bugs as default and catch the moment that they are repaired, as the testing should uncover only the removal of bugs when the implementation proceeds.
26. Model based testing can find new bugs, as can fuzzing.
27. The central advantage of model based testing is the going through a huge space of possibilities in a way which is not possible in any other style of testing.
28. Usually, test automation is not sufficient as such but the same things need to be tested in other ways. Often that way is exploratory examining testing.
29. It is simplifying naiveté to think that one point of view would be enough for the testing of complex systems.
30. Often the bugs are caught in designing of the tests, when the system is analysed.
31. General question: Why automate tests if at the first execution of a test is enough for finding the defects? Repetition is of course the reason for it. There can be many kinds of the repetition needed: regression testing, testing with another set of data, testing in another environment.
32. Often it is thought that the automatic tests are always linear – that they have a certain

order, decided in advance, from which there is no deviation. This is often true but model based testing works otherwise.

33. One central advantage of the automation is the fact that it does not become tired. Humans should not be used as workers even in monotonous repetitive tasks, not to mention tests taking several days.

The great pretender

34. A lot is talked about the crowd testing. With load testing automates a couple of computers can mimic the activity of a large number of users.
35. The automation can also be used to shrink time. Things can be done much faster than it would happen in a real life.
36. Automation is a time machine among other things!

When does test automation pay off?

37. A thought of it being free is often connected to automation. When the devices have been acquired and have been put to use, nothing costs hardly anything after that.
38. In reality, automation is quite expensive.
39. There is a traditional rule that testing is worth automating if it will be repeated at least four times – if the infrastructure exists.
40. The work of the maintaining the tests is surprisingly expensive. A one-off, contracted test development is not enough.
41. The more dynamic the software development is, the more challenging is the automating of tests.
42. The lower the abstraction level of tests, the more often the tests will break.

The changing culture of tools, in other words the beauty of small and open

43. A couple of years ago, the basic style of test automation was to acquire an expensive test automation application from a big tool manufacturer, the licenses of which were very carefully controlled, with big bureaucracy and long delays.
44. **The culture has changed totally. The central tools are free, open source tools that can be given to everybody. The culture can develop at project level in an agile manner and not based on corporation level decisions and being tied to their compromises.**
45. The tools are the results of the culture. For example, agile Linux-like culture produces very different tools in a bottom-up manner that for example the developments in the abstract world of modelling or the strict world of test management.
46. The tools are often ordinary, light and agile. Where before for example TTCN 3 was thought as a basic tool and language, it is now often Python or similar – simple and suited to many every-day tasks from serious software development to ad-hoc scripting.
47. That conclusion should not be drawn from this that TTCN-3 would not be important any more. It is it but the selection of possible tools has gotten richer than before. There were very not many kinds of cars at in the early days either, but now we know that it is good for a taxi, shopping bag, van and formula car to be different.
48. It is essential to take cultural matters into consideration when choosing tools. It will be good if the tool is “at home” in the environment it is used, with the people who use it.

Testability

49. When programs are tested automatically debugged, they must be possible to control programmatically.
50. In that case all kinds of testability are important. This has already been known for a long time.
51. **However, testability is forgotten in the development of every new platform. It is not in the biggest priorities even at the application level.**
52. A stricter culture is needed regarding testability. A testability review is an important part of development processes.
53. However, the test adapters are developing and the help of the open source tools is

available either directly or by integrating tools together. For example, a testing tool can use Robot Framework between itself and the test target.

The testers' competence

54. Accomplishing the automation is largely manual work! The automatic tests are rarely created automatically even though it is a part of the utopia. The programmers build as with manual work unit tests that later automatically run. The building of the model in model based testing is usually demanding manual work. Behind the development of automated user interface tests there is often interaction with various professionals. So it is even teamwork.
55. The creation of the test automation is serious software engineering work and the maintenance of test scripts (and other assets) is serious and demanding maintenance work.
56. So competence that exceeds the traditional tester's competence is needed.
57. Most testers need the understanding about the basic ideas of automation, targets of its application and restrictions of test automation and the ability to execute tests.
58. Specialized experts are still needed for demanding automating tasks.
59. It is important to create tools which hide difficult details and allow the testers to use the abstraction level characteristic of them, and the mental models and concepts.

Process of taking automation into use

60. The traditional reaction to the taking automation to use is that it is a big change and affects people in many ways. The change is implemented using a heavy transformation process, seeking for the solution that is the best for the whole organization.
61. Such a process includes the analysis of needs, looking for alternative tools, their comparison and evaluation; assessing some tools in detail, tailoring some tool for use supported by a pilot project, trainings, going to larger scale usage and follow-up of how the automation is used. In other words, a similar process as in taking a serious information system into use.
62. That kind of a serious process is reasonable because there are a lot of tools available to choose and the wrong choice is always harmful.
63. However, many tools are small-scale and elegant and do not need a very big supporting infrastructure. They can be brought into use in an agile manner as needed.
64. The challenge is then the transfer of the good things of the automation from the first team to other and from the first unit to other units.
65. The challenge is especially in thin and agile organizations in which there is no time for “general matters" and one wants to get directly to the point in the projects without any planning stages in which there would be time to think about the tools thoroughly.

Sometimes Siberia teaches

66. The amount of work on the maintenance of the tests has taught a lesson to many organizations. When software changes, the tests will easily break. And the software will change when it is being developed and repaired and changed when the business changes.
67. When the automation once gets in bad shape, the situation will not be salvaged, because the motivation will be first thing to get lost.
68. However, one must not think that automation would not work. One just must invest in its ability to manage in the changing world with good planning and sound choices (techniques, targets of application).

About teaching of the testing automation in education

69. The core of the teaching in universities is to teach the central principles of things and to avoid concentrating on the tools.
70. The big challenge is time. There is no time have many courses about testing and one can only include so many things in one course.
71. Test automation has traditionally been the repeating of the systematic manual testing with an automat. In that situation, the tool used is clearly in a side role – the main thing is how to write good test cases for the automat.
72. In unit testing, the significance of the tool is big: unit testing is done in the programmer's context and with programmer's tools. Learning of any other style of working or any separate way or working would not make sense. That is why xUnit testing tools are often seen in the teaching.
73. The integration testing which is done in continuous integration has often been hoped to be included in the testing course, but in that the central thing is the integration process that is part of the software construction process. It is mainly a logistic process. Testing, however, uses principles that are learned from other contexts.
74. In the system testing, the core contains the principles that are learned through manual testing. And automation doesn't make much sense if the tests are run only once.
75. However, an opportunity can be given for the students that wish to learn automation to try it.
76. In some educational institutions, model based test automation has come to the side of traditional automation. By trying it, new things are learned.
77. So the automation is not central to everybody but for some of the students, its learning provides new opportunities, stimuli and skills needed by the business world.
78. In the educational institutions, we can hopefully promote a diverse, sensible testing culture where the paradigms and competences supplement each other.

A national treasure

79. **It is great that there are many top level experts of different types of test automation in our country!** Let's be on top of that are too, but let's also remember the other elements of the rich world of testing.

About the future

80. Not many years have passed since a testing expert stated that the only party that has made money with test automation are the testing tool vendors… But now, with the combined effect of many things, we come to the situation where the culture of merchants has turned more into a living, ordinary culture in companies.
81. Still the biggest threats of test automation are the religious attitudes towards it, mechanistic thinking in testing and tool-centeredness.
82. As a small conclusion one could state that test automation is not just automation technology, but management of the wholes, making right choices, mind control, sane and sensible leading and balanced use of all kinds of testing competencies.
83. **Automation is a good hired man but a bad master!**