Introduction
Comparing American testing conferences with the European equivalences I noted some differences. Europe seems to be rather focused on methodology and structured working processes, while in the US there is a large emphasis on executing the right tests and finding bugs. A survey that Collis held among 230 visitors of the STARwest conference in October 2008 showed that popular test standards such as T-map and ISTQB were used, but other methods were more popular. Among these other methods were SCRUM, RUP, XP and Agile. All of these are very much related to the software development, whereas the testing methods T-map and ISTQB focus more on testing as a separate profession. These results might partly be explained by the fact that Europe has a lot of companies that provide services and have more administrative operations, while the industry in the US is more products based.

On the American test front, there is a lot of attention for Agile testing, the front runners of this movement, sometimes say they feel that they have to evangelize the message. Many of the presentations that I hear on Agile and exploratory testing focus on the benefits of these approaches. They focus less on how these can be integrated in a structured approach. To my opinion this is precisely why these front runners feel they need to evangelize. A lot of the testers and their managers have difficulties with making the transition from traditional testing towards agile/exploratory testing. The previous mentioned survey indicates that more than 50% of the testers work with predefined scripts. The same survey shows 51% of the responders uses a structured approach, but feels it sometimes as a hindrance. I think we can help this group by showing how they can obtain some agility, without having to say a farewell to the structured approach they are used to. This can be done by integrating exploratory testing into their existing test approach. By providing a solution that better aligns with their experience, we lower the threshold and make it easier for this group to join the group of ‘agile’ testers.

What is Exploratory testing?
Exploratory testing is originated from the idea that the traditional waterfall test approach with predefined scripted tests does not match the dynamics of modern software development, Nor is it aligned with one of the principal requirements for a knowledge economy: learning. Exploratory testing aims to increase the flexibility of the test process and continuously include new insights and ideas in the approach.
James Bach, one of the front runners of exploratory testing defines it as follows:

“An approach to software testing that emphasizes the personal freedom and responsibility of each tester to continually optimize the value of his work by treating learning, test design and test execution as mutually supportive activities that run in parallel throughout the project.”

James Bach [Bach/Bolton 08]

The definition emphasis that within exploratory testing the tester takes his responsibility in order to deliver added value all the way through the project. The tester has freedom and is challenged to adapt his testing and the approach he uses in order to do so. In short:

“Exploratory testing is a mindset on how to use a set of skills” Jon Bach

Thus, testing has a lot to do with the right mind-set, or attitude if you like and being a good tester that has skills and knows to effectively use them to add value to the organization. But what does a tester have to do, to become an exploratory tester?

Michael Bolton says: “everyone does exploratory testing; they just don’t admit it. If they investigate bugs, they explore. If they do anything other than exactly what the script says, they explore. If they vary any aspect of their approach, they explore. And so on” [Bolton 08].

To my personal opinion this description is true, but I have experienced that this wide definition of exploratory testing makes it a concept that is often too hard to grasp. Therefore there is often misunderstanding about when to call testing exploratory. Besides, intentionally or not, exploratory testing is often used as a cover-up for doing scripted testing in a poor way. Both misunderstanding and misuse blur the success-stories of exploratory testing and make people reluctant to start applying it. Therefore I plead to use a more strict definition.

Session based exploratory testing gives a lot of structure and is quite easy to explain. To testers, stakeholders and other people involved. In short: Session based exploratory testing uses test charters that define the required tests on a high level. During a series of subsequent test-sessions the testers, often working in pairs, process the charter assigned to them. They get the freedom to define the tests they want to execute during the assigned time. During this session the testers need to be critical, creative and are challenged to find as many bugs as possible. After each session a debriefing is held, in which experiences are shared and new areas of attention can be addressed. These might lead to new, previous unplanned test charters.

Without contradicting the any above definitions we can also define Exploratory testing as an approach that is a combination of Test techniques, Process control and Attitude.
Figure 1: Exploratory testing consist out of Attitude, Process and Test Techniques

Let me try to explain how we can integrate exploratory testing into a structured approach by using these three elements.

**Test techniques**

Test design techniques like Boundary Value Analysis, State Transition testing, or any other can be regarded as tools that we testers use to derive the test cases needed to find bugs, or to make a statement about the quality of the system. In a scripted approach, the test design techniques describe the steps the tester needs to take in order translate the test basis into test cases.

The purpose of exploratory testing is to drive test design; it has therefore test design technique like aspects. Session based exploratory testing describes the steps that are taken during the sessions, these are define charters, assign charters to the testers, execute the charter, have a debriefing and if necessary define new charters, assign the next charters to the testers, etc.
A big difference is that doing exploratory testing test design and execution go hand in hand, or as James Bach says “learning, test design and test execution as mutually supportive activities that run in parallel throughout the project”, while within a scripted approach the design and execution are separated in time.

Exploratory testing can be used as tool in the very same way as test design techniques, with the same purpose. Doing so enables exploratory testing to be integrated in the test approach that is used in most scripted test projects. Emphasizing the technique like aspects of exploratory testing and explaining the steps that are taken while applying it, will make the approach more easy to grasp. It clearly explains what exploratory testers do. This will increase its acceptance.

Figure 2: Steps taken in exploratory testing

- Define Charters
- Assign Charters
- Test Charters
- Debriefing
- Preparation
- Test execution
- Session evaluation
**Process**

The process aspects of exploratory testing deal with risk analysis, test strategy, registration and reporting. These are processes that are widely used within our traditional test approaches. Merging and aligning these processes has two advantages. 1) Traditional testers recognize more of their testing in the exploratory approach and are more likely to accept it 2) The agility of the exploratory approach will cross over to the traditional processes making them more flexible. Let us look at these processes in more detail:

**Test Risk analysis**

During the debriefing sessions charters are assigned to test-pairs based on their priority. The risk analysis needed to establish this priority needs not to be different from the risk analysis as done in a traditional test project. In fact, it rather is the same.

The survey indicated risk was often used in order to differentiate the test depth. 21% of the responders did not do really have a process to evaluate the risk and indiced to use gut feeling. 62% used risk analysis, suggesting some kind of process. However, of this group only 42% involved stakeholders in the assessment, the other 58% did not and testers determined the risks themselves. One could state that this is not necessary a bad thing. A lot of risk might be quite technical and therefore beyond the stakeholders perception. Although true, I think it is very important to have stakeholders involved. For one, we are doing the testing for them and we will need to show the added value of our activities. Secondly involving the stakeholders in the process, will gain the trust they have in the test process and the advice we testers will give them regarding the release of the system.

In order to get stakeholders involved, we should avoid the testing (either exploratory or scripted) to be something for testers only. We testers should make the risk analysis understandable and more attractive for the stakeholders. This can be done by avoiding tedious excel sheets and an overdoses of facts and figures to be filled in. This can be done by explaining why we ask them for information, how we will use their input and how we will use the risk in the testing process and reporting. In TestGoal I introduce a one dimensional risk assessment that is more lightweight, but encourages stakeholder involvement [TestGoal].

**Test strategy**

If we do an overall risk assessment, we can select for each area of attention what the best risk mitigation will be. Testing can be done scripted, or in an exploratory way. The test strategy aims to appoint what the most effective tools are for testing specific areas of the system. Scripted testing can be a requisite for areas where governance is required on coverage and test results. This might require the usage of test design techniques. This can also be the case for areas of the system that contain for difficult calculations or business rules. For parts of the system that lack a solid design or functions that are quite sensitive to change, a more flexible approach like exploratory testing can be more effective.
Regarding Exploratory testing as tool in the very same way as test design techniques enables exploratory testing to be integrated in the test approach. This approach enables good overall decision making that delivers an consistent story to the management. This will increase the trust management has on the test team.

Additionally; Test design techniques can be used in a formal way. The techniques to be used are pre-described in the test plan, and are used in order to create a logical test design. Coverage is controlled based upon this logical test design, which in turn gets translated to physical test cases. Many test organizations do not use test design techniques in a formal way. In this case testers should still use the test design techniques implicitly during review, test design or during the execution of the exploratory tests. Having knowledge of test design techniques enables testers to look at the system through different glasses, and helps to find errors in different domains. If it is believed that the used test techniques should be controlled, these can always be added to the test charter as a suggestion.

Recording test results
In a scripted test project, each executed test is checked passed or failed. When this is done properly it is known, what tests have passed and failed and what tests have been executed. This latter being an indication for the progress of the testing. Having an indication of progress is important, because with progress the trust in the systems increases and the greater the certainty of the release advise.

How are test results and progress recorded when doing exploratory testing? We do have the bugs that are registered in the bug tracking systems, we have personal notes and sometimes video is used to capture the activities of the testers. But how can these be integrated in the overall test reporting?

An exploratory testing session can contain dozens, hundreds of ideas, questions asked (and answered), and test actions taken. However one should not forget that the charter is defined for a certain reason, it should align with some identified risk and should eventually help answering a question that stakeholders have.

When Exploratory testing is done by means of test charters. Each charter will be closed with a conclusion. If desired the test charter can also be extended with additional instructions for the tester in the charter or an instruction to a pre-described checklist.
If we integrate these items into the system we use for our scripted tests we avoid that the exploratory tests are controlled using separate lists and include all results in one report.

The simplest way of doing is to add one test for each charter, item on the check list or instructions for the tester. We do not instruct the tester how to execute the test, but we do register the aims of the charter and ask the tester to give his expert opinion on each of the items. By doing so, all exploratory results can be processed along with the other test results, thus leading to a test reporting that gives a one combined picture of all test activities. One has to be aware of the difference in weight between the results of elementary scripted tests and the results of test charters. But this problem also exists with scripted testing since the effort needed to check a test as passed may differ here also.
Test reporting

The registered test results yield great information on the progress of the test execution and the quality of the system. Ideally the test results can mapped to the items addressed in the risk analysis or function of the system, and thus enable benefit based reporting. This means that the test results are translated into items that live among the stakeholders and will increase their involvement as much as the added value testing has to them. The risk analysis should contain items that have a meaning for the stakeholders (and are therefore addressed by them), the test report should address to which degree this risk is mitigated.

By integrating the exploratory tests in the overall reporting, we obtain two advantages. The results of all charters are translated to items that have meaning for the stakeholders and they get the complete picture. The exploratory testers are encouraged to understand what the relation between risk and test charter is. This will yield in a better understanding in the purpose of the test charter and help them with executing the tests with added value for the stakeholder.

By involving some key stakeholders in the debriefing sessions, he will gain more insight in the quality of the system and the problems the test team is dealing with. The test report will come to live and the key stakeholder will better know how to read the test report. He will take notice not only of the results from the ET sessions, also of the scripted test results.

Attitude

The third element to address is Attitude. Actually exploratory testing is all about attitude. Describing the steps that we take during the session, embedding it in the testing approach that is also used for scripted testing, will help in lowering the thresholds. It will make it easier for people to start using it. Lacking the right attitude, they will fail creating the success story.

The attitude required for good exploratory testing is no different than that of good scripted testing. In both cases the tester should be critical, creative and be able to recognize it, when the system shows strange our unwanted behavior. In both cases testers should enjoy their work, as this will enable them to put the required persistence and energy into their testing.
Another aspect of attitude is the business alignment. We need to have a good focus on the anticipated business result and a good understanding on how as a tester we can add value to the organization. Both the traditional tester and that of the exploratory testers should align their work on business goals, eliminate work that does not add value to the business and communicate with their stakeholders in clear language.

Getting experienced with applying exploratory testing will help gaining a result-driven attitude. Since exploratory testing values individuals and interactions over processes and tools, it will make the tester more aware of his attitude. He will learn that there are more options than most test methods describe, and try to explore those options in order to increase his added value.

By presenting an version of exploratory testing that is aligns with the experience of the traditional tester, we can get more testers to start using it. We are offering a extra tool for his testers-tool-box. While using exploratory testing I trust the emphasis will gradually shift from the step-wise technique like approach to the required skills and attitude. This ‘agile’ mind-set will challenge the tester in reviewing the tools he is familiar with, and find more effective ways to use them.

*Closure*

Embedding exploratory testing in a structured test approach will make it more attractive for traditional testers to use, since it better aligns with their experience. Using exploratory testing as a technique will enlarge the toolbox of the tester and make it easy to embed in a more methodological structure. Gaining experience with exploratory testing will help looking at the existing tools in a new way. This will help in gaining a result-driven attitude.

With the above remarks, I hope I have lowered the threshold for applying exploratory testing. If you’ve never done exploratory testing, why not start with implementing it in your traditional test project. People, who have done it in my project, became enthusiastic;

“The ET session gave us clear understanding of the quality of the system. This was achieved in a very short period. The fun about ET is that its fundamentals are easily understood”

‘Exploratory testing is testing on the edge. ET means taking the most out of people; this implies you’re dealing with people issues. It is exciting to find the edge of ‘we have tested all the essential’.
Acknowledgements

I would like to thank the following colleagues for giving me advise and reviewing this whitepaper:

Michal Bolton (Developsense)
David Gorena (Microsoft)
Martin Storm (Collis)
James Bach (Satisfice)
Paul Jorgenson (Grand Value State University)
Anko Tijman (Ordina)
Erik Boelen (QAconsult Services)

References

[TestGoal] TestGoal, Result-driven testing, Derk-Jan de Grood (Springer 08)
[Jon Bach 08] Keynote STARwest 2008 “Most project managers insist that all testing is documented. Exploratory testers don’t know how to balance documentation and testing. (Jon Bach STARwest 2008)"
[Bach/Bolton 07] Rapid software testing v2.1.5 2007, Satisfice, inc
[Bolton 08] Personal email exchange

Profile

Derk-Jan de Grood has broad, hands on experience as test engineer, test manager and advisor in a large range of industries. As manager of a test department he learned how to implement test methods the practical way. He gives lectures at various Dutch universities and is author of the first educational book specially written for teaching software testing at Dutch universities. Derk-Jan is also the author of TestGoal, the result-driven test philosophy. Besides that, he is a passionate, inspiring speaker at major testing conferences all over the world such as the STAR conferences in the USA and Europe. Derk-Jan is Consultant and Product manager for Valori.