

SWET3: Swedish Workshop on Exploratory Testing 19-20 november 2011

Skåne Peer conference on Teaching Testing

Olika discipliner, såsom konst, musik, vetenskap m.fl., har olika kulturer av lärande. Vissa har formen av självlärande, vissa bygger på ett livslångt förhållande mellan lärare och elev, vissa uppmuntrar kritiskt tänkande och ifrågasättande och vissa kräver att eleven har totalt förtroende för läraren utan att ifrågasätta något.

Att ta på sig rollen som lärare kan säkert ha en mängd olika drivkrafter. Kanske vill man hjälpa någon, kanske ser man det som en moralisk skyldighet att dela med sig och vidareförmedla det man fått av andra, och kanske är det viktigt för ens egen utveckling och förståelse.

Vad gör en lärare till en bra lärare? Finns det några pedagogiska utmaningar som är speciella för testning? Vad innebär det att vara mentor/coach? Hur skapar man en "lärande" miljö på arbetsplatsen? Hur förmedlar man intresse, motivation och inspiration?

När man sitter på den mottagande sidan av lärandet. Vilka förväntningar har man på sin lärare. Vilka former av stöd kan man behöva?

Organizers:

Henrik Andersson
Robert Bergqvist

Delegates:

Sigge Birgisson
Anders Claesson
Oscar Cosmo
Rikard Edgren
Johan Jonasson
Ola Hyltén
Maria Kedemo
Petter Mattsson
Simon Morley

Introducing software testing to university students

Sigge Birgisson, sigge@jayway.com

When speaking about something, it is really important to adapt and listen to your audience. I have held lectures several times for students about testing in general with a great weight on my own experiences and projects. The mission of these lectures have been to inspire and introduce young people to the area of testing and describe what it means in reality to test software.

Even though they are all students, it is important to acknowledge the differences in background and understanding of software development on an individual level. Some of the students have been studying 2nd year of digital communication and marketing while others are on their 3rd or 4th year in computer science. Some are having multiple hobby projects during their spare time, some have professional experience from before in software projects while others have just finished high school with no programming background. Some just got an ISTQB like course as their first and so far only introduction to testing while others did not know testing was a profession of its own.

My lectures change somewhat every time I am presenting with new examples and to adapt to that particular set of students with background from the teacher about their present context, but the core is the same with a high degree of agile and context driven examples of testing in the projects I have been part of.

When reflecting on my presentations, there are certain aspects of them that are stronger than others. One example is the hard part of bridging the knowledge gaps of talking about software testing if there are no or very little knowledge about programming, software development processes in general or not knowing about the difficulties in communication.

It could also be that I know that my audience just finished the ISTQB certification, and I need to find a balanced approach to rant about it from my perspectives without losing all credibility for ranting about their whole education in software testing.

I will in this presentation at SWET3 discuss the material of my student lectures and some more reflections about them. While it is not as much of teaching as speaking about software testing, I see the importance of making the introduction speaking approaches a way to enter the teaching context further on. As a part of the discussions, I would like to extend these types of software testing introductions lecture reflections to other groups that do not have hands on software development experience but are still working with software, like web designers and social media people.

Att utbilda i Lean/Agile utveckling och test

Anders Claesson 2011-10-30

Erfarenheter och utmaningar av att utbilda två affärsenheter på ett större programvaruföretag i en övergång från traditionell utveckling (V-modell) till att använda de senaste metoderna inom Lean/Agile utveckling och test.

Under hösten har jag haft förmånen att få vägleda två större affärsenheter i hur de kan tillämpa de senaste metoderna inom Agil/Lean utveckling och test. Då omvärlden förändras i en allt snabbare takt, måste företagets sätt att arbeta även gå i samma takt som sina konkurrenter, inte bara när de gäller den tekniska utvecklingen utan också hur de arbetar för att få fram nya produkter (för att bli mer effektiva).

Utmaningen är dels att sälja in ett nytt arbets sätt, samt att göra det på ett så bra och tydligt sätt att eleverna kan gå tillbaka till sin dagliga arbetsmiljö och börja tillämpa vad de lärt sig. I regel behövs mer än en kort kurs för att byta arbets sätt. Någon form av fördjupande (längre) kurs, coaching och workshops är möjliga vägar att gå vidare med för att etablera nya principer i hur man kan arbeta enligt en ny modell.

Insikter som jag fått efter att ha hållit ett stort antal kurser är bl.a.

- Att visa på hur ett nytt arbets sätt även kan tillämpas i de metoder som själva kursen utvecklats och framförts på. Att exempelvis ha feedback och retrospectives efter varje kurstillfälle för att gradvis förbättra kursen till att bättre svara mot kundens behov, förväntningar, krav och önskemål, samt ge önskad effekt hos de som deltar på kursen.
- Hur väcka intresse så att alla kan känna sig delaktiga och förstå budskapet.
- Hur man kan strukturera upplägg, innehåll och framförande i en kurs så att man skapar en röd tråd som är lätt att följa.

En pilot-kurs kördes för alla chefer på affärsenheten. Syftet var att ge dem tillfälle att ställa frågor samt ge feedback både på framförande, innehåll samt budskap. Cheferna måste känna att de kan ställa sig bakom samt förstå vad kursens innehåll syftar till att förändra på ett klart och tydligt sätt. När man tar fram en kurs är det svårt att gissa vilka frågeställningar som dyker upp. I det ständiga förbättringsarbetet med kursen fördes svaren på flera av de större frågorna successivt in i materialet. I slutet blev den riktigt bra och alla kunde känna att de fått en god förståelse av budskapet som kursen förmedlar.

Efter varje kurstillfälle fick även alla elever ge sin feedback på framförande och innehåll. Totalt hölls 19 kurstillfällen (2 timmar långa, plus frågestund).

Outtalad utbildare på min arbetsplats

Oscar Cosmo

Min erfarenhetsbeskrivning handlar om min roll som outtalad utbildare på min arbetsplats. Webbranschen, en branch med rötterna i reklam och PR snarare än klassiska teknikriktade områden, har historiskt en begränsad erfarenhet av test. När jag började på Creuna ansågs det vara en belastning och i vissa led även som ett tecken på att bolaget går åt fel håll och vill bli IBM.

Som ensam testare och testkunnig över huvud taget har jag haft en roll att utbilda anställda och införa test som en naturlig del av företagskulturen.

Arbetet började i tysthet med att jag deltog i ett projekt. Jag använde all min kunskap för att visa projektmedlemmarna vad test gick ut på, inte genom att snacka så mycket utan bara långsamt föra in stödrutiner för test i form av buggrapportering, testmiljöer osv.

Intresset för mitt arbete ökade i takt med att projektet spottade ur sig bättre och bättre leveranser och att teamet verkade ha roligt och koll.

Nästa steg var att stoppa in fingrarna i nästa projekt och i samma veva i allt större utsträckning tala om test på projektledarmöten och i andra forum där diskussionen passar in.

Idag lever jag ett liv där jag i alla fall en gång i veckan har besök av en nyfiken projektledare eller utvecklare som vill veta mer om något specifikt ämne.

Jag tror att den här metoden för att upplysa om test, genom att sporra eget intresse, är ett vinstrecept. Däremot krävs det tid och därmed en tro från "de som betalar" för att få tid till att genomföra sin gameplan.

Teaching Testing Basics with Open Source Applications

Rikard Edgren

Together with Henrik Emilsson I am currently teaching at a higher vocational study ("Yrkeshögskola") for software testers, spanning over 2 years.

It is twenty students without testing experience, all on-site, they have five hours of teaching three days a week.

We do a lot of exercises, at least 2 per day, since we believe learning is amplified by experiencing.

Based on the first six weeks of teaching testing basics, I'd like to share how we have built exercises by using open source software from sourceforge.net.

1. Search for applications.

a) We search on sourceforge.net, using vague search terms like "application", "software"; narrow down to alpha or beta products; look at them ordered by Latest Release Date.

b) We want the application to be pretty new or active, to give the chance of giving feedback for real.

c) We want the application area to be somewhat familiar to most of the students, e.g. a journal is more appropriate than a Fourier expression generator.

d) We want the application to be relatively stable, in order to not be disrupted by too much problems.

e) There should be few system requirements, so setup is fast and without problems.

f) We have seen that any application has a wealth of interesting things to look at, most have memory handling issues, installation problems and usability concerns. This is good, as you can always broaden the way you look at the application, and the way you look at software testing. You can do a lot of testing on very small applications.

2. Get to know the application yourself.

a) To familiarize and be able to explain it, and know about a couple of problems.

b) The most important step, to be able to tie the application to something we want to teach.

Examples: We found an application with a dubious installation process (old Firefox installed in background) that was almost too problematic when trying installation testing.

We found a very simple application that violated a lot of Usability categories.

I will show you one of many shutdown applications we have used, one was perfect as the first test application, to learn about Benevolent Start heuristic, and generic test ideas.

3. Teach with exercise

If time permits, our experience show that you get the best results with the following steps:

a) Show an example of what you are teaching.

b) Explain what you want them to learn, and the theory behind it.

c) Instruct how to do this, with an example.

d) Introduce the application, invite students to ask questions about the purpose of the software, and targets for the testing effort.

e) Let them test (in groups), walk around and ask them how they are doing, suggest further test ideas for those that are stuck.

4. Summarize findings

- a) Ask them what they experienced.
- b) Talk about major findings, and tie them to what you want them to learn.

5. Invite to extra work

- a) Tell them that they can summarize findings and send to the developer of the open source application (they haven't got the courage yet...)
- b) Tell them to try the same exercise objective on another application.

Lessons:

It takes time to build up a good battery of applications to test.

We found it easy to look at an application, and find a suitable exercise. But more difficult to have a good exercise idea, and then search for an appropriate application. This is easier after some time, when you have a larger collection of applications of different types.

For people new to testing, it is good to use many different applications, to gain familiarity from a tester perspective.

It is also good to re-visit applications you are familiar with, to show that new techniques and heuristics can be used to make a more thorough testing.

If you are an experienced tester (yes, it takes 10 years) it is remarkably easy to help students come up with more ideas, and pinpoint problems.

Not being too familiar with the software yourself has been an advantage to us, since it wakes up our curiosity and investigative mind.

What we don't do (yet):

Build up elaborated scenarios, with details, traps, roles etc (we often have a target user group though.)

Setup assignments to test a whole application, and report all findings to the open source project. (We will do this later, with the risk of not being able to use the application in future classes...)

Summary:

It is not difficult, quite fun, and good learning to use open source applications to teach testing basics. There is no need to inject faults when the real world is full of them.

Teaching Testing Online

Johan Jonasson

The Association for Software testing offers a series of 4 week online courses in software testing to its members and the testing community. One of the reasons for this is because part of the AST's mission is to improve the practice of software testing by furthering education of testers. The AST believes that too much of what's being taught today in commercial testing courses emphasize only superficial knowledge and basic ideas, and that this knowledge is not deep enough to be applicable in the students' day-to-day work. Through their online courses, the AST hopes to foster a deeper level of learning, and at SWET3 I hope to share some of my experiences from this course program with you.

The program is simply called "Black Box Software Testing" and so far I've taken the first two courses, called "Foundations" and "Bug Advocacy". By the time SWET3 kicks off, I'll be about half-way through the third BBST course, called "Test Design". There are more courses in the works, but so far these are the three main courses available (combined with an Instructor's course).

Earlier this year, I started out as an assistant instructor for the AST and I've taken part in one Foundations class and two Bug Advocacy classes since then. I have gathered impressions and experience from the material and the teaching methods and I will do my best to share the pros and cons as I see them. My basic premise is that the way the BBST courses are made available and taught, combined with the contents of the courses, is by far the most valuable piece of training you can get as a professional tester today. I base the "valuable" argument on the fact that the Foundations class is free for all AST members, and at the moment, the following courses are priced at \$200 each. That is for 4 weeks of interactive, instructor led training, complete with video lectures, quizzes, homework, and a final exam. And you don't even have to drag a worthless certification back home with you when you're done.

Of course there are some drawbacks to this teaching method too. It would naturally be nice to have both the instructor and students in the same room. It would be nice if the communication would be more real time. Some of the teaching mechanisms, e.g. quizzes that requires a lot of precision reading to pass, might not suit everybody. Even though each course runs over 4 weeks, the pace is still sometimes frantic, demanding at least 8-10 hours per week from the students. And that's the Foundation course estimate. The follow-up courses are much tougher.

After giving you my experience report, and after having a refreshing discussion about its pros and cons, I hope I've at least given you some valuable insight into this method of teaching testing. And maybe you'll be interested in trying it out as well.

Teaching test to a group of individuals

Maria Kedemo

About 5 years ago I started working at Securitas Direct. I was very excited over my new job as a test manager with the assignment to start a test department meaning getting the right staff, methods and tools to support the development process.

Having worked as a contractor for several years in temporary teams and temporary projects I had for some time been wishing for my very own team, a team that was mine, to evolve and to grow together with. I had a vision of the A-team or the Justice League of testers with distinct skills and expertise that was there to serve and help the development department to deliver our products with high quality. I was determined to introduce exploratory testing and Session Based Test Management to be able to handle the vast number of test activities and products that the development department worked with.

I still have that vision but the effort and time of getting there has not been what I expected. The given prerequisites have also changed over time.

Today my team members do not work in a test team. They are all scattered out in different development teams (Scrum teams) either working as the only tester or together with another tester. The context has changed and the testers need to broaden their skills since we do not have the team to fall back on any longer.

The development teams have various ways of working and different views of test which gives the tester a lot of challenges.

This has also given me a challenge in teaching test to the testers as individuals and to help the development teams to solve problems that have arisen along the road. Among other things I've struggled with:

- How can I help?
- What do I need to teach? How do I strengthen the testers self confidence?
- How do I get the developers to accept the tester?
- How do I arouse interest and enthusiasm?
- How do I introduce Session based testing?
- How do we keep our knowledge alive?
- How can I teach my testers to be more professional?
- Why have it taken such a long time?
- How can I still get my A-team?

Vilka är förutsättningarna?

Petter Mattsson

Sammanfattning

Detta är en inblick i vad jag kommer att beröra i min presentation ämnad för SWET3. Jag kommer att lägga vikten på att dela med mig av min erfarenhet kring vilka förutsättningar jag anser är viktiga kring själva lärandet. Allt baseras på mina egna erfarenheter och iakttagelser.

Läraren

Vilka kvalifikationer bör en lärare ha för att kunna lära någon något?

När det gäller testning vet jag att många hade svarat att läraren själv måste ha kunskaper inom testtekniker, metoder, begrepp och terminologier. Detta kan jag hålla med om till viss del men...

Erfarenhetsmässigt så har jag kommit fram till att det är annat som genererar ett mer långsiktigt och värdefullt lärande. Nämligen att lägga vikten på att skapa en inspirerande miljö för lärandet. Den viktigaste funktionen en lärare har är att inspirera och motivera eleverna så pass att de själva hittar sina vägar till hur de lär sig saker på bästa sätt och hur de hittar tillfredställelsen i lärandet. Det är som med allt annat här i livet, om man gör någonting som man inte tycker är kul gör man hellre någonting annat.

Eleven

Så vad är elevens roll i det hela och hur ser kraven på eleven ut?

Det som ofta blir fel när man kommer till själva eleven är att man försöker lära ut saker till elever som inte är intresserade av det de blir lärda inom. Detta fenomen kan uppstå i två fall, antingen om eleven inte har information nog för att bedöma om ämnet är något för honom/henne eller om ämnet helt enkelt inte intresserar eleven trots insikt i ämnet.

En inspirerande lärare kan i många fall vara lösningen till båda av fallen ovan. En inspirerande lärare kan skapa ett intresse som kanske inte tidigare fanns hos eleven. Det är viktigt att aktuellt ämne läggs fram och inleds på rätt sätt, baserat på åhörarna och baserat på erfarenhet hos läraren.

Man måste kräva en hel del av eleverna också. När man väl har tagit sig över tröskeln kring att skapa intresset gäller det att börja arbeta mot eleverna. Man vill uppleva kreativitet, proaktivitet, ansvar och ett innovativt lärande från elevernas sida. Det är här lärartekniken kring de öppna diskussionerna kommer in. En duktig lärare "lurar" in eleverna i detta beteende.

Verktygen

Att ha rätt verktyg är enormt viktigt. Tänk på att verktyg inte bara behöver vara hårda.

Något som ofta görs på fel sätt i detta är att man inleder med att pumpa in testtekniker m.m. i lärandet vilket gör att man tappar bort det viktigaste av allt, frågan om vad syftet egentligen är med det man lär sig och vem man gör det för. Man borde alltså göra tvärtom och börja med syftet istället. Det gör att tekniker och annat hanteras mer logiskt och att de kan sätta in detaljlärandet i ett logiskt perspektiv. Att hela tiden hålla en öppen diskussion är också oerhört viktigt. Eleverna måste känna att de verkligen är delaktiga. Fråga efter deras synpunkter, fråga vad de tycker, anpassa lärandet runt

eleverna, lär dig från dem lika mycket som de lär sig av dig och använd det för att utveckla lärandet dynamiskt. Tryck på vikten av att nätverka med andra, helst utanför sitt eget företag och organisation. Tryck på vikten av att skaffa sig mentorer. Tryck på vikten av att alltid vilja förändra sin vardag till att göra den lite bättre imorgon baserat på vad som hände idag. Tryck på att känslan av att man ofta uppfinner hjulet på nytt är en dålig känsla, ett varningstecken. Få dem att förstå värdet av att lita på människor framför dokumentation.

Management

Så hur kommer management in i bilden?

För att över huvud taget kunna ägna sig åt lärande i alla dess former så behövs tid för det. I alla fall om man ska kunna göra det på bästa sätt. Det är lättare att lära sig saker om man tillåts att göra det. Dom som sprudlar av proaktivitet lär sig oavsett fast inte lika effektivt som om dedikerad tid hade funnits för det.

Som vi alla vet är det någon form av manager som beviljar just tid för personlig utveckling hur nu än formatet ser ut för det. Så hur kan man påverka ett sådant beslut? Hur kan man få managern att inse vilken viktig investering det är, alltså personlig utveckling hos medarbetarna? Det finns ett antal vägar man kan gå. Jag kommer att beskriva det som jag upplever vara det mest effektiva sättet. Det är inte snyggt men det är effektivt...

Underhåll

Det klurigaste av allt. Hur behåller och utnyttjar man känslan av att man lärt sig någonting spännande och nytt? Hur bibehåller man viljan att ständigt lära sig nya saker?

Inspirerande och proaktiva ledare är ett krav. Stöttning är ett krav. Tillit är ett krav. Medarbetarnas vilja kring att ständigt förbättras är ett krav. Feedback från sin chef är ett krav. Synlig uppföljning är ett krav.

Mindset Changes: Changing the direction of the oil tanker

Simon Morley

This experience report will look at some experiences of changing understanding and perception of testing amongst non-testers (especially managers) and testers.

This is not a 'complete' experience report - it's ongoing. Introducing mindset changes - talking to testers, non-testers, stakeholders about re-focussing testing on what I (and others) believe is important. This includes:

- What's important?

Are we asking good questions? Do we need to ask good questions?
This is the re-connection with what testing is doing...

- Troublesome words

Some words get in the way of talking about testing.

- Ignoring numbers for the sake of numbers

Relation to cognitive biases - demonstrating that people connect feelings to numbers and eventually numbers trigger feelings...

- Understanding frames

Use examples from operation

- Sowing seeds for change - understanding things can be different

When we can see potential problems in the way 'we' think about testing, then we can be more open to new/different ways of thinking...

- Difficulties in talking to 'established' testers

- Difficulties in talking to managers.