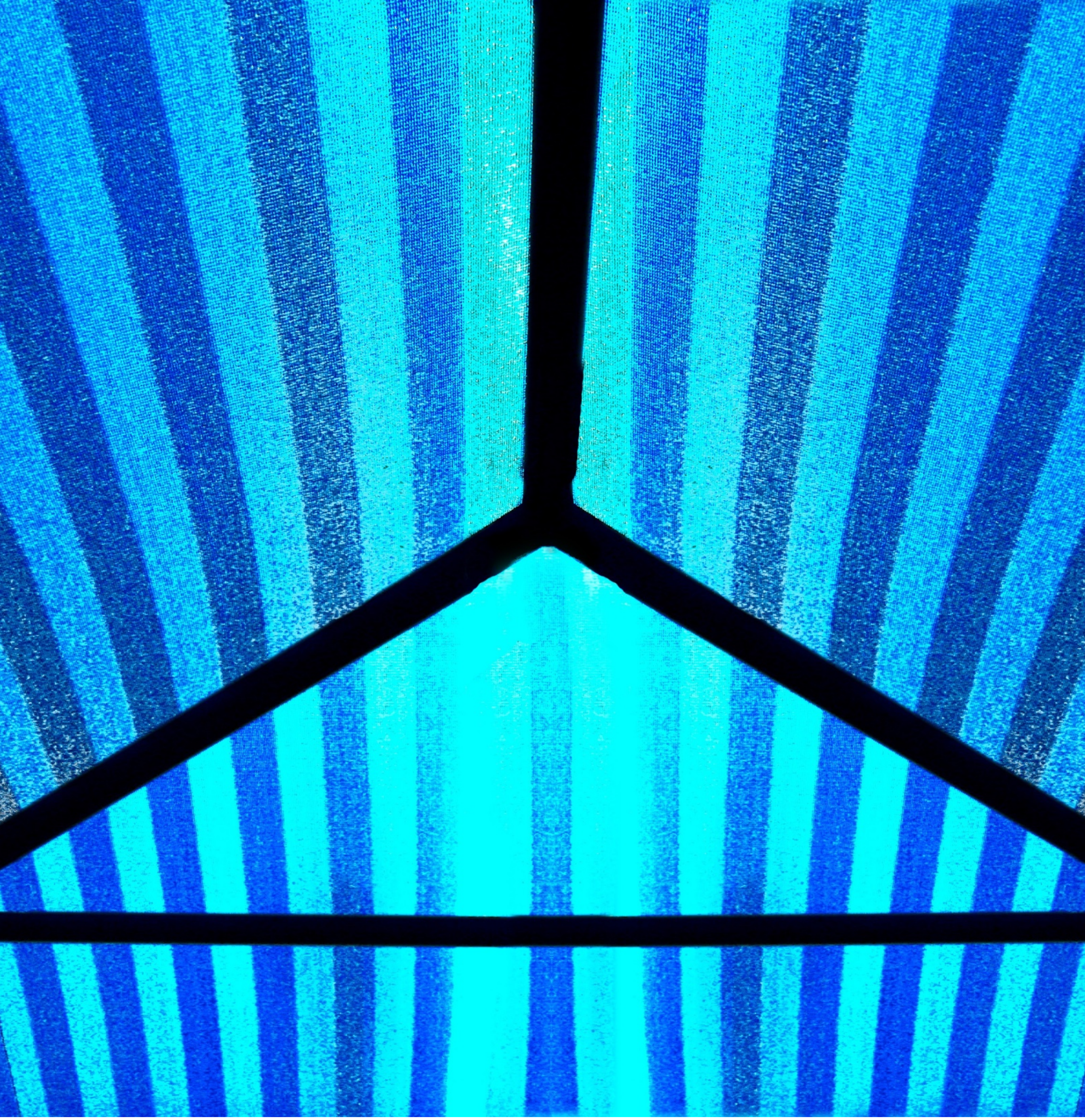

TESTING TRAPEZE

OCTOBER 2015

ERIK PETERSEN • SAMANTHA CONNELLY • OLIVER ERLEWEIN
JENNIFER HAYWOOD • HUIB SCHOOTS

HOW I BECAME A RAPID SOFTWARE TESTING TRAINER

HUIB SCHOOTS
DEN BOSCH, NETHERLANDS

> *"Because testing (and any engineering activity) is a solution to a very difficult problem, it must be tailored to the context of the project, and therefore testing is a human activity that requires a great deal of skill to do well. That's why we must study it seriously. We must practice our craft. Context-driven testers strive to become the Jedi knights of testing"*
>
> - The Dual Nature of Context-Driven Testing, James Bach

IN JUNE THIS YEAR I TAUGHT MY FIRST RAPID Software Testing (RST) class. Unlike the way somebody can become a teacher in the factory school of software testing, becoming a Rapid Testing Instructor takes more. Much more. This is my experience report of my journey towards one of the most challenging goals I have had in my life: becoming a Rapid Software Test trainer.

My early testing career

My first steps on the testing path were in a happy-go-lucky way. I didn't know much about testing and common sense helped me to do my work. I learned about structured testing methodologies. As a tester and later as a test manager, but also as a teacher, I observed many people test and learn to test. I struggled with the testing I did and saw others do. I started doubting the testing I was taught and teaching myself: it just wasn't right… There is much more to testing that wasn't told in the 'common' testing classes. And why weren't we testing actual software in these classes?

Fake testing

Testing has a bad name in many organisations and testing is often undervalued. Testing is often seen as just checking requirements and pushing buttons. I heard many people claim that everybody can test and saw companies where testing was used to train newbies in IT. In the years after the internet bubble, people used testing to get (back) into IT. Looking at our industry, it still can be done easily. Many interviewers and managers do not know how to recognize the difference between fake and professional testers. That is why so many fake testers are still running around without being detected. Do you remember your last job interview? Did you have to do actual testing? Or were they only asking questions? In the movie business even actors like Brad Pitt have to audition for a role. The director wants to see them play the character they have to play in the movie…

> “Do you remember your last job interview? Did you have to do actual testing? In the movie business even actors like Brad Pitt have to audition for a role”

Context-driven testing

On foreign blogs I started reading about agile and context-driven testing. I found there is much more to testing than I could guess. I learned new, exciting and interesting aspects, which I hadn't considered before; like critical thinking, models and heuristics. In 2006 I read the book *Lessons Learned in Software Testing: A Context-Driven Approach*. I remember reading one of the one star reviews when I ordered the book from Amazon: “This book is VERY dangerous to a serious testing organization because it focuses on minimal documentation…” I thought: “This is a book I have to read!” I learned many new things, but I was also confused about several things. Looking back now, I don't think I fully understood what the book was teaching me. Epistemology, systems thinking, conjecture and refutation and biases. All concepts that I read about, that didn't ring bells in my brain yet… Heuristics intrigued me although I did not grasp the importance of them at first. My view on testing, proving, assuring quality started to change gradually.

Lessons from the book

Some of the lessons I read were mind blowing. The authors connected some dots in my brain. Some examples I remember, e.g. “Lesson 34: 'It works' really means it appears to meet some requirement to some degree”, made me realize that there is no such thing as proving software works. The lessons on testability, e.g. “Lesson 136: testability is often a better investment than automation”, “Lesson 137: testability is visibility and control", "Lesson 156: programmers like to help with testability" and "Lesson 169: Ask for testability features", gave me a totally different notion of the importance of making testing easier and faster. Those lessons were very influential on my testing after reading them. My two favorite lessons from the book are: "Lesson 283: apply diverse half-measures" which taught me that we need to use a diversified strategy, and "Lesson 272: if you can get a black belt in only two weeks, avoid fights", which taught me that many skills need practice to gain enough skill not to be of danger to ourselves.

> "Lesson 272: if you can get a black belt in only two weeks, avoid fights" ... taught me that many skills need practice to gain enough skill not to be of danger to ourselves

Two Futures of Software Testing

In 2008 I saw Michael Bolton's talk 'Two Futures of Software Testing' at EuroStar in The Hague. A couple of weeks after the conference I asked my manager if I could go to a workshop for the Rapid Software Testing class by James Bach or Michael Bolton. It was kindly refused because the content of the class was controversial and not the things our customers nor my colleagues and I needed. So I kept on reading the stuff online.

In the years 2007 – 2010 I was a trainer for ISTQB and TMap courses. While teaching the same class several times and telling the same story over and over again, I got the chance to really think about it. I remember one day when I drove home thinking about what we did in class. I had been teaching a three-day class where we never actually tested anything. We only talked about testing and did exercises on paper. Over-simplified exercises like applying a test specification technique on an over-simplified example written out on paper.

Observing people testing

In 2007-2009 I was manager of a team of enthusiastic, smart young people. Most of them joined our company directly from school and were trained in beginners' classes for a couple of weeks to become testers. Ironically these classes were called "master classes". These were intensive training weeks with lots of exercises and practice. After these master classes the testers went to work for clients in projects. As a coach and manager I observed what they (and others in their projects) were doing. It occurred to me that they were filling templates, trying to apply standards and often a happy-go-lucky approach was used. I started wondering why they were not using the test design techniques from the training. Were they too difficult to use? At the start

> “In my head the steps to take were often clear, but the things I came up with didn’t fit the standard processes and ways of working provided by the testing methodologies I knew”

of a project they filled in a test plan template and started creating testing cases out of requirement and design documentation. The strategy in their work was vague. They claimed they were using their “common sense” to guide their testing, but were not able to explain how they worked nor how the techniques they used were working. As a coach I wasn’t able to really help them. I was struggling myself and didn’t know a different approach. In my head the steps to take were often clear, but the things I came up with didn’t fit the standard processes and ways of working provided by the testing methodologies I knew. I started using terms such as “pragmatic testing” and more and more I was stepping away from templates and formal testing documentation like Master Test Plans and physical test cases.

Taking action: DEWT

In 2010 I switched jobs and joined a bank as team manager. Testing was not highly appreciated and my team was struggling with the same issues as many other testers. At a Dutch testing conference in October 2010 I met several people who where discussing their issues with testing and wanting to do something. We all wanted to talk with likeminded people about a different way of testing. Maybe raise our voices to influence others? Challenged by James Bach, we started to discuss a get together over Twitter. Several months later the Dutch Exploratory Workshop on Testing (DEWT) was born. DEWT was a major driver towards Context-Driven Testing for me. Discussing testing with people who took RST made it clear that this was the stuff I wanted to learn more about.

In the testing expertise group at the bank I suggested we try something different. We decided to ask James Bach and Michael Bolton to teach RST for us. I got in contact with Michael and several months later almost 50 testers took RST in two groups. I was blown away by the content and exercises (see my blog post about it here and here). I learned so much and many dots in my mind were connected. After a week of RST at the bank, Michael joined the first DEWT peer conference during the weekend. I was lucky to spend 7 days with Michael. We had great conversations. That week I asked Michael if RST could be taught by people in a different language. We talked about that idea and a seed was planted in my brain. After a week full of RST and DEWT I was so inspired. My mind was overflowing with ideas. A year later we invited Michael back at the bank to do exploratory testing workshops. Again I was able to spend a lot of time with him and got the opportunity to observe his way of working. I especially enjoyed the experiential approach in his workshops and his provoking questions to trigger people's thinking.

The classes and conversations with Michael inspired me to start blogging and speaking at conferences to share my ideas. Blogging and writing articles helps to get your thoughts together. I started many blogs without finishing them. I realized I needed to learn more about the topic to make my writing worthwhile.

> "Blogging and writing articles helps to get your thoughts together. I started many blogs without finishing them. I realized I needed to learn more about the topic to make my writing worthwhile"

(Peer) Conferences and Skype

In 2011 I was lucky that one of the speakers at Agile Testing Days was ill, so I got the chance to present my 'So you think you can test' talk (see my blog post on this topic here). My first international conference talk was nerve wracking, but it gave me an enormous energy boost. I talked about context, collaboration, skills and the importance of practice and my talk, surprisingly, inspired other people! Since then I visit (peer) conferences more regularly. Over the years I met many awesome testers and I got the chance to learn from them by working with them and have long discussions at various meetups, conferences and via Skype. It gave me the change to regularly talk to many like-minded colleagues about context-driven testing.

With James I did a couple of Skype coaching sessions via instant messaging (more information on coaching via instant messaging see here). Skype is a great tool to work with testers all over the world. Coaching while typing instead of talking forces you to carefully think about what you are doing and what you are trying to say. With James as my demanding coach, I realized how important it is to be precise in what you say. What others dismiss as semantics, can be very important to deeply understand each other. A ninety minute Skype session with James resulted in my studying for some days to fully understand what was said. I also did a couple of Skype coaching sessions with Ilari Hendrik Aegerter before I started to coach people on Skype myself.

> “I needed to learn to be less impulsively helpful and be more patient so students could experience the traps set in the exercises for them to learn”

This is very rewarding as I got the chance to help testers who are serious about their craft. I also got the chance to deeply think about RST related topics while working with my students.

Instructor in the making…

In 2012 I had dinner with James where we discussed RST classes in the Netherlands. He mentioned that I could become one of the RST trainers, but that there would be a long way to go to get that far. This made me so happy and I realized that it was now up to me to convince Michael and James that I was capable of teaching RST. The same week James and I started with some of the RST exercises. Talking about RST in class is important, but doing the exercises right is the most important part since the class is experiential. I needed to learn to be less impulsively helpful and be more patient so students could experience the traps set in the exercises for them to learn. One of the biggest worries of James and Michael was that I would violate the RST namespace, so it might look that we have contradicting idea of what was taught. I needed to stay consistent with the RST principles and namespace at all times. This took quite a lot of practice and experimentation to find out what kind of way would fit me as a person! Michael and James encouraged me to develop my own style, my own exercises and my own slides. By checking back with them on a regular basis, they helped me to develop step by step and I earned their trust to a level that they let me be an instructor.

Questions and challenges

James likes to question and challenge people. Over the last three years, whenever James was in the Netherlands or Belgium, we spent a lot of time together to learn as much as possible. On those occasions he asked me to explain exercises or concepts from RST. The better my answers were, the more he pushed me by asking difficult questions. I enjoyed working this way, because it helped me prepare for difficult questions in classes. I remember one occasion when I was driving James from his hotel to an evening session at one of his clients. I was driving my car through heavy traffic in the pouring rain, while James was sitting next to me increasing the pressure with every answer I gave. After 20 minutes I asked if we could switch topics, since I really had to focus on the road. James explained that I was doing a good job. The better my answers would be, the more he would challenge them.

Practice, workshops and testing dojos

I was fortunate not only to manage or consult over the last few years. I also participated in projects while working as a tester for clients. These projects gave me many opportunities to try out and experiment with heuristics, session-based testing and coverage outlines that I took from RST. That work also gave me lots of examples to share in class: sharing experiences instead of appealing to authority is a context-driven way of working and a core principle in RST. Three years ago I started doing workshops where I used topics from RST: at clients were I was coaching people or teams, in workshops for clients and in workshops at conferences. Each time I took just a small part, like test strategy, testing story or oracles to focus on. These workshops were opportunities to practice explaining the concepts of RST and to experiment with various exercises. I always prepare for workshops by practicing and creating examples by myself. The interesting part of workshops at conferences is that on most occasions experienced context-driven testers were in the room. Their feedback and critical questions were very helpful.

> “Observing testers working is instructive. Same counts for observing students doing exercises in class, testing software or practicing their testing skills otherwise”

Observing testers working is instructive. Same counts for observing students doing exercises in class, testing software or practicing their testing skills otherwise. Teaching a variety of workshops gave me the change to observe people working and practice giving feedback and running debriefs.

Videos

RST has many exercises and it would require a lot of time with Michael or James to train me in all of them. To get the necessary practice and feedback from James or Michael, I started recording myself while I practiced RST exercises with my colleagues. These recording were send to James and Michael for feedback. Sometimes we went through these recordings minute by minute to discuss what was happening. The detailed feedback by James and Michael helped me to deeply understand the exercises and improve my teaching skills. I invested a lot of effort to learn the experiential and Socratic approach. The videos helped me learn about the planted traps, possible scenarios and gave me insight in the learning opportunities in each exercise.

Feedback

James and Michael gave me a ton of feedback on many things I did. Getting feedback from two totally different people with different styles gave me insights in many possible ways of working and teaching. Sometimes feedback can be hard to receive. Working with both Michael and James is tough and demanding, but very rewarding. I learned much and became a better tester, trainer, coach and person. I got the chance to learn many skills and I became better in receiving feedback without feeling attacked. A great example was the last class I co-trained with James early in 2015. In this class I got the opportunity to do an exercise. While doing this exercise I made a mistake and James stopped me and started giving me feedback. I realized that I was enjoying his feedback despite of the students were watching me. When we were back on track, I explained to the class that I enjoyed what just happened. To become truly excellent at something, details matter, and honest feedback is a key for learning.

Seven classes

Since 2011 I have participated in seven RST classes with both James and Michael. In the first class I was a student myself. The next class I was involved as organizer for my employer, but I also attended and learned by actively participating. I grew into a more active role by actively participating in the class and by being a coach during exercises. In the last classes I was co-trainer and I did parts of the class, mostly running exercises.

Taking the same class more than once gave me the opportunity to deeply understand the mindset and the skill set. Each class gave me new insights. For me it was like digging through different layers of the earth, slowly toward the core. Each class revealed new layers, new things to learn, exciting opportunities to explore. First because James and Michael keep developing the RST material, but more important, because there is so much to learn: theory, testing skills and instructor skills to run the class successfully.

When I started this journey, I thought it would be much easier. It was a lot of hard work, a struggle sometimes, but it was totally worth it! A journey in which I learned a lot about myself and the testing craft. I am looking forward teaching the next class in December in the Netherlands this year! See you there?

Parts of this article were published on the DEWT blog in 2014.

Huib is a tester, consultant and people lover. He shares his passion for testing through projects, consulting, coaching, training, and giving presentations on a variety of test subjects. With almost twenty years of experience in IT and software testing, Huib is experienced in different testing roles. Curious and passionate, he is an agile and context-driven tester who attempts to read everything ever published on software testing. Huib is one of four instructors of Rapid Software Testing. A member of TestNet, AST and ISST, black-belt in the Miagi-Do School of software testing and co-author of a book about the future of software testing. Huib maintains a blog on magnifiant.com and tweets as @huibschoots. He works for Improve Quality Services, a provider of consultancy and training in the field of testing.

THE TRAPEZE ARTISTS:

EDITOR
KATRINA CLOKIE

DESIGN & LAYOUT
ADAM HOWARD

REVIEW PANEL
BRIAN OSMAN
SARAH BURGESS

Testing Trapeze accepts no liability for the content of this publication, or for the consequences of any actions taken on the basis of the information provided.

Image Credits: Cover, Contents, Editorial, Testing Circus, Event, Community, Next Time, Trapeze Artists