

THE PROBLEMS WITH TESTING

Introduction

The world of software testing is in chaos. Sure, there are no riots but it's in chaos. Ok, so there's no looting either but chaos is ensuing. Alright, maybe chaos is the wrong word. Maybe uncontrolled state of flux is better? Yep, that's better but it's not as emotionally charged as the word chaos, so chaos it is. Software testing is in chaos. No doubt.

No one really knows which way to turn. No one really knows who is in charge and no one fully understands why testers seem to enjoy goading each other and starting fights.

The software development industry is still fairly young in the grand scheme of things and so we, as testers, are still finding our way, just like other disciplines like Development and Project Management. Unlike other disciplines though, testers don't seem to have settled down in to the flow, hence we are still experimenting,



theorizing and plotting. And in finding our way we are experiencing a whole wealth of differing opinions as to what is the right way to go about things.

The light at the end of the tunnel for the problems facing the testing community, and I do believe it is a community, is that there is no "one" coherent view despite the fact many in the testing community would have you believe otherwise. This I believe will be healthy for our community and could stop us tearing the community apart from within.

In some sense, the testing community is still recovering from its attempt to certify its way out of trouble. At one level certification has proven successful for those trying to become more employable, for those who adopt a simplified checklist approach to recruitment and those solely in the game of making lots of money, on the other hand though, it's become a pointless piece of paper, it's triggered a "genie out of the bottle" dilemma, it's been devalued by the fact almost everyone has one and it's still not made us all better testers.

Certification inflation is becoming common place where testers are needing even more, higher status certifications just to remain at the top of the pile and there are many organizations willing to help out. As more mid level and subtle



variations on certification schemes come out we are often facing option paralysis when deciding which certification scheme to sit next. This is a “genie out of the bottle” mindset where we might as well continue shooting ourselves in the foot, because...well because we’ve already started it. “But which one is best?” I hear the testers shout.

When I mentioned we were recovering from our attempt to certify our way out of trouble it’s important to point out that we are still certifying ourselves silly. But I strongly believe there are now enough people in the industry who understand that certification is not working as anticipated and are stepping up to do something about it. People are beginning to realise that certification is not the complete answer. People are beginning to ask for more.

Certification is far from being the only reason testing is in chaos though. Here in this book you will find a whole host of other reasons. I urge you to read this book with caution and care, for it holds many truths about software development and testing. Truths which may lead the testing community to understand its wrongs. Truths which may bring about a revolution (or evolution?) in software testing. Truths that threaten to bring the whole testing world to its knees. Truths that may scare you silly, shock you to the core and make your face hurt.



The biggest mistake

As there is no "one" way of looking at the world of testing, that ultimately means there is no "one" group or person in charge despite what some might think. And as no single person or group of people is in charge it leaves the door open for a whole raft of thinkers, businessmen and organisations to come in, set up stall and take charge. Some of these have good intentions, some don't.

It's ripe for those who see themselves as political leaders or those after some extra cash to step right in, fleece us silly and then leave us wondering what went wrong and who ate all the biscuits.

And so we find ourselves at an interesting point in the history of software testing. We find ourselves turning in on ourselves, bickering, fighting and potentially destroying ourselves. We run the risk of making the testing community's biggest mistake: self destruction.

There are people pushing certifications, ridiculous training courses, more metrics than we can shake a stick at, getting too hung up on terminology and shouting 'Best Practices Rule' from any parapet they can find. There are others pushing experience over certification, automation over manual testing and agile over



waterfall. And this worries me. It worries me because the more we bicker and fight with each other, the more we become irrelevant to the software development world in general.

There are some new movements to remove testing all together. People are saying testers will not exist on projects in 2 years time. Or at the very least their role will change beyond recognition.

Testers are known as troublesome, difficult and frustrating. Fairly soon we could become known as "pointless" or "irrelevant" or "a waste". I know of some people who think that way about us already.

I'm not professing to know the answer to the dilemmas of the software testing world. I'm merely a social commentator asking questions, bringing together views and leaving the decision to you. Let's not make the testing world's biggest mistake.

Let's self reflect and make a positive change, not just for our own community but for the wider software development world too.



A change that will bring about reform in testing. A change that will iron out the creases. A change that will become our crowning glory.



Let Them Fight For It

At some point there will be a moment of truth and all those who don't care about testing will disappear, leaving the purists, the test heads and the lifers. And this is no bad thing.

But why wait until that point?

Let's just drop every tester on a deserted jungle island and let them fight for themselves. Those left standing at the end can begin to lead the new testing world.



A Tester's Battle Royale

I'd bet a year's salary the context driven testers would come out on top. They know how to adapt. They appreciate the world is diverse and complex. They are ready for this impending battle. They've researched and read around their craft as well as many associated areas of study. They aren't bound by global testing norms, best practices and scripted paths of conformism. They are free to adapt. They are born survivalists. To be honest I reckon some of the context driven purists would be handy in a fight with a stick, some old vine and a couple of coconuts.

The problem is, though, that the certification movement has got the masses. Maybe I should join them? Safety in numbers. The trouble is I'd need to do a multiple choice test to join and I don't think there would be time to revise. I can't afford to just keep taking the exam. In fact, I'm not sure how to pay for the exam anyway; would we have a currency on the Island?...a universal one?

I'm also not convinced it would truly prepare me for life in the jungle either. It's more a generic introduction to jungle terminology rather than a way of providing me safety out there in the wild. According to the blurb, to be a hero grade jungle



tester I'd have to sit the practitioner course and I really don't have the patience for that.

Anyhow, no amount of certification will give me the skills and wit I need to fight off giant man eating parrots whilst trying to work out where my next meal would come from.

Or maybe I should side with the "formal education degree is everything" group. I have a degree after all. Ok, so it's a science, media and communication degree but times are desperate, we are fighting for survival aren't we? A degree is a degree right? I can't recall them even having a testing degree when I was at university. I'm fairly sure I wouldn't have sat it if they had; I always wanted to be a film director.

Surely they can let me in? Testing is diverse after all. Science, sociology, communication, psychology, languages...they all play a part in testing. And to be honest, I don't understand this "formal education and degrees" only group. Testing's not about formal education, it's about self learning, critical thinking, creativity, communication and a whole lot more isn't it? It's about you and your



brain, not what degree you have. Testing isn't about getting a Masters and becoming a member of the elite.

I guess after writing this paragraph I've blown any chance I had of getting in to this Elite group anyway. I also wouldn't want to be in the Elite few controlling the masses; my heart lies in community projects and I can't see much of that in a future led by a select few demanding formal higher education for all.

Or maybe I should just go it alone. Branch out. Make my own decisions. Create my own movement. I'm a social tester after all. I could surely attract some followers. I could certainly tweet about it, write a few blog posts, maybe LinkedIn with some other stragglers. Digg it, reddit and delicious about it. Create some buzz and then MyFace it. Then again, there's unlikely to be any electricity so I'd have to use smoke signals and that's hardly "web 2.0".

What about the Acceptance Test Driven Development peeps? I love these guys. Test first. Good advice for life in the Jungle. Although that does require you knowing what you should be doing in advance and with all the other desperate test tribes on the loose I'm not sure I could predict my path too well. What if I accidentally ate an hallucinogenic fruit thinking it was a mango? I'd be writing



tests all day based on verifying the presence of a 24ft Bertie Basset and flying dog named Brutus.

But I believe some of these divides might be a good thing. It could be that we fight it out to lose the dead wood and work out what's important to the testing community. Maybe it's a good sign that things are advancing and the testing community is evolving.

Or maybe we need to simply continue as we are for the next 10 years until we suddenly realise we aren't relevant anymore and generation X/Y or Z have snuck in, stolen all the glory and set up camp.

Anyway, you can make your own decisions about what the future may hold for testing after reading all about what's wrong with it. So without further a-do I present the super special book "The Problems With Testing"



Automation

Automation is to blame for all of our problems in testing.

Automating software testing is no different from automating tasks outside of work. We are ultimately trying to save labour time, do tasks we would not normally be able to do and automate those tasks that are tedious. And on paper that sounds awesome. Problem is, in reality it's often not quite as a sweet as that.

We started automating mundane lifestyle tasks in earnest in the 1950s. We called them Labour Saving Devices. Apparently. I wouldn't know though, I wasn't born then.

We, as humans, like to think we can make our lives easier by automating tasks. And in some instances it has worked a treat. The washing machine, the dishwasher, the vacuum cleaner and the car are all ways we have automated a task (or replaced an obsolete task or advanced a process) and made our lives easier, cheaper and potentially more efficient.



It's not all good though. Anyone remember the electric pizza cutter, the desktop Almond slicer or the mini candy floss machine? Potentially good ideas. Badly marketed? Badly designed? Pointless?

Software testing automation is very much like lifestyle automation; in some instances it is invaluable, in others not so. If done correctly rewards are bountiful, regression is taken care of, impossible manual things become possible and testing can be done early and often. It can also remove the tedium in our daily testing lives. Done badly and we are left to wallow in a heap of expensive kit, heavy weight instruction manuals and defunct skill-sets.

The problem with automation is the impossible belief that it will replace humans. That we can somehow do all of our testing without the human mind. Machines will replace us. But this impossible belief is just that... impossible.

Take the trusty bread-maker. Some people love it, some people hate it.

It still requires a human to load it carefully and accurately. It still takes a human to design the recipes, calibrate the machine and test the process many times before



releasing it to the general public. It still takes a human to set the machine correctly and start it going.

The machine does indeed mix it and then bake it. (Or mix it ready for you to bake it)

A human has to remove it when it's done. A human has to analyze it by observing, tasting, touching and smelling and then deciding whether the bread is good. A human troubleshoots any problems. A human is still essential.

It also still needs a human to have designed and built the machine, written the instructions, designed the marketing and packaged the product for sale.

So why are we convinced that automation is the silver bullet to replacing human testers? Unless of course your testing is mindless clicking and ticking and checking...

Good testing is still a very human activity, which works best when combined with machines to do the tasks **they** are good at, leaving us to do the things **we** are



good at. A nice balance between mind and machine. And when automation is used in this way it's incredibly powerful.

However, we buy these expensive labour saving devices in the belief that it will save time, maybe save some money and definitely add more quality but in reality it sits on the side until the next big argument with senior management over why we used it once and then left it alone. And so we try the whole sorry process again, just to make sure we didn't have something set wrong in the first place.

So with ardent determination we use the said tool, in this example, the bread maker, and spew out something that sort of resembles a loaf of bread.

Despite the fact the loaf is half the normal size of shop bought bread, has a whiff of plastic about it and is a distinct grey colour we are still pleased with our automation. We bask in glory at our automation. We have baked a loaf. We are experts. We have succeeded in automation.

We test it.

We eat some.

We gag.



We then get angry, we then shout about how expensive this machine is turning out to be and we then throw this awful bready result to the birds.

The birds, in a combination of shock and disgust, stage a rare dirty protest by scooping up this bready delight and dropping it from a great height on to your car, cracking your windscreen and causing £ 800 worth of damage.

In a fit of rage and with a distinct feeling of being scammed, you throw your labour saving device on the floor before jumping up and down on it shouting "you were supposed to save me time, money and effort" or something like that.

You then confine what remains of the labour saving device to the back of the shelf. And after a period of time and a couple of arguments with the management (in this instance, my wife) you try it again (after fixing it) and wonder why you ever left it alone in the first place. And you repeat this process until you go postal and walk out with your t-shirt over your head grumbling about bad management and dodgy best practices.



Don't get me wrong. There are many people out there using their labour saving devices (automation) with exceptional and consistent results. Especially those who use their automation early in their product development process.

But these people have planned for months (sometimes years), bought wisely (or used an Open Source tool), learned how to use it, have become experts, experimented with ingredients, set expectations, accepted the bad with the good and have taken time to understand what they want their device to do. They are craftspeople using tools when tools need to be used. And the tools they are using are fit for purpose.

And that about sums up test automation. Great in concept, occasionally executed with precision and value but more often than not left on the shelf, rarely used correctly and despised by all who touch it.



Best Practices

Best Practices, or lack of them, is the reason testing is in a mess.

One of the major fallouts of having no imposed direction or clear leadership is that many testers find themselves flocking to depressing testing forums.

On these forums we have testers posting questions inviting other testers to comment on why they are indeed an idiot. Or to put it another way we attract lots of people claiming their way is the only way and berating anyone who disagrees.

Best Practices for more than one moment in time are a myth. They simply do not exist. There is no best practice but only relative judgments. Your Best Practice might not work for me. So why insist on it?

Don't get too precious about your own personal Best Practice because sooner or later you'll find it is no longer the best. Sooner or later something will have changed or some more information will become apparent. Sooner or later it could be the Worst Practice.



Best Practices are moments in time when something went well, for someone, on some project and in some context. There's no guarantee that it will work again in the future as your context is forever changing. If it does work again, you could have just got lucky.

Yet this does not deter some people.

Some people are ardent believers that their way of working is the only way and so flock to forums and discussion groups to berate anyone who disagrees.

Unfortunately, these people have taken up other social media channels and so nowhere is sacred. Not only are some testing forums a no go (not The Software Testing Club by the way – we moderate heavily) but Twitter, LinkedIn and Facebook are heading the same way. I'm not safe to ask a simple question anywhere.

They now write their own blogs extolling the virtues of "MY WAY". They now comment on other people's blogs with the same old drone of best practices. It's



got to the point now where some clever and talented testers are scared to write anymore blog posts because of these testing cyberspace "Best Practice" bullies.

But let's get one thing clear. Best Practices are not the true, honest and real stories of testing success or failure. We all want to hear about real stories because they are what gives the testing world the human face and helps us learn. A balanced "this is how I approached it" is great. It gives you reference points. It gives you advice.

A Best Practice is when the person talking cannot (and will not) comprehend that it might not work for everyone.

To make matters worse there are organisations who grab an idea and start to call these Best Practices "Industry Best Practices" and sell consultancy and products off the back of a myth. I shudder with horror when I read those words "Industry Best Practice". And seconds later I go elsewhere for my testing information.

Context is real. It does exist. Ideas, suggestions, recommendations and concepts are what the testing community needs, not more Best Practices.



Certifications

Well. Here goes. Time to put the cat amongst the pigeons.

Certification is the reason testing is in the state it is. For sure. No doubt about this at all. Certifications are ruining this industry.

Certification is one of the most hotly debated topics in the testing world. Hotly contested and passionately discussed.

Certifications may well have come out of the good intentions from people wanting to raise the bar in testing but they've morphed into a way of making testers more employable and recruiters lives all that easier. They have resulted in below average testers and hero-grade testers being compared like for like because they both may or may not have a certificate.

Sadly I know great testers who can't get interviews because they don't have a certification.



Certification is a good thing for some recruiters, the certifiers and many testers keen to show their aptitude.

Let's be honest about it all. The certification is a piece of papers that says you sat a course and passed an exam. It's not certifying **you**, it's certifying the course you sat. And that can be a very good thing.

"I certify that Rob has sat a three day course on software testing".

Instead our industry is certifying the **person**. We are saying "**Rob** is of such an excellent testing skill level that he warrants this certification. Well done **Rob**."

It wouldn't be quite so bad but we are now being taught testing certification courses by people who aren't testers. They are "**certified**" trainers teaching **certification** courses to people who want to become "**certified** testers". Ask these people a genuine testing question and they won't be able to answer it....I know...I asked my trainer several questions and he couldn't answer a single one. (Yes I know....I have a foundation cert....I was young, naïve and tricked in to it)



They also make a select few people a lot of money whilst they dilute any positive effects the certifications may well have brought. And with new ones popping up all the time it's only a matter of time before certification inflation becomes a widespread disaster. At which point the job that needed a foundation now needs an intermediate and so on. It's at this point that even more companies jump on the certifications bandwagon to offer even more possible certifications. We end up spending over £ 1500 a year to keep ourselves on a level with our peers.

One thing that is clear is that certification simply isn't working at raising the bar. Visit some of the testing forums and some conferences to see some of the ridiculous questions being asked by people fully certified up.

Certifications bring each and every tester down to the same base level. They treat each and every tester as a "unit". A "unit" that can be tested, marked, understood, categorized, analysed and then awarded a certification.

But could it be that the content is good but the award mechanism and training process needs overhauling? What if the content was delivered through training from an experienced tester? Would that make them better? Would that be the introduction new testers need? What if we all agreed that the certification



certified the course and not the person? What if we simply changed the amount of emphasis we placed on them? What if we treated them as what they are, just a small part of a testers education?

Maybe that's too optimistic.

The certification industry is literally worth millions. Could it be that certifications are about money? Where does all the money go? Back in to the industry? Does it provide more free and open access training? Or is it pumped back to the community? I have no idea.

So certifications are certainly a way to make money. Which is fine, it's got a good business model. No-one can begrudge people for making money. But is the certification helping the testing community or is it ruining it? Is it making testers "better" at what they do or simply giving them some generic terminology and a piece of paper? Is that a bad thing? Is it spoiling testers with false education?

Certifications are certainly dividing the community intensely. A sensible debate cannot take place about certification and no matter how degrading many testers find them, there are countless others who love them.



The increase in the number of global certifications, and suppliers all keen to have a piece of the certification pie reflects a deeper problem within the testing community: utter confusion amongst many managers, testers and testing organisations over what actually constitutes good testing. Many people simply cannot state what is “good testing” and so turn to certification as a form of comfort blanket, as a way of proving to the software development industry that testers are valuable and professional. And could this actually be a good thing? I’m not so sure.

The global certification trolley dash represents a cultural shift towards quick wins, quick fixes and instant gratification at the expense of long term learning, studying and work experience. As societies and culture move in a vastly new direction, maybe being able to “buy” a qualification or certification is now the future. Maybe certification are part of this long term learning? Maybe Dinosaurs like me who appreciate experience and learning have no place in this instant results testing future? Time will tell.



Mega Complex Measurements of Doom

I attended a conference a while ago where someone talked for a whole hour about a very well known framework for working out how mature your testing is within your organisation.

During the first ten minutes of the talk the crowd started getting fidgety. After about thirty minutes, three people walked out, two people were asleep, one of the sponsors had passed out and one person went crazy and charged out of the Fire Exit.

It's not that the theory isn't sound, although I have my doubts, it's just that there is no fun and sexy way of delivering content like this. There's only so many fun ways you can talk about measuring maturity (or whatever other complex model you want to talk about) without sounding a little bit dull.

In business value terms I'm still not convinced that recording all this data is of much use. I've always gone down the road of just getting things done. I'd prefer to crack on and try creating something than spend months prepping, measuring and then filling in forms and spreadsheets to work out whether we are mature or



not. In some ways I know very well that I am immature. It's in my nature. It's why I write books like these.

I once spent 6 months preparing a cost benefit analysis spreadsheet of 1,000,000 worksheets and over 7 trillion calculations regarding 4 requirements, but unfortunately I only left 1 day for testing so the project got canned. And that's why Mega Complex Measurements of Doom are to blame for the state of testing. There's too much emphasis on numbers and not enough emphasis on satisfying the customer and just getting things done.



Testing Events

Testing Events are to blame for the state of the testing world. Testing Events attract the following types of people:

- People who don't like testing and don't want to be there, but they were forced by management or they got a free ticket.
- Consultants looking for the next role.
- Recruiters looking for their next clients.
- Vendors looking for someone to sell their wares to.
- Testers wanting to break free from the monotony of daily life and make a start in the world of public speaking.
- Testers looking for a day out of the office.

Every delegate (attendee) leaves experiencing mixed emotions. It's too hot, too cold, too boring, too cutting edge, too bland, too edgy, sound was too quiet, sound was too loud, bad slides, bad presenters, rubbish vendor stands, too many sales people, too many consultants, not enough consultants or maybe the venue was too smelly.



Whatever the problem it's fair to say that many people derive great value from Testing Events too (myself included in this) whereas some people just find them pointless. It's fair to say though that almost every conference on testing presents the testing world with even more problems, as new ideas and concepts are shared, aired, discussed or seeded.

If you agree with what's been talked about it's a waste of time you being there. If you disagree you come away feeling that the speaker is a buffoon and the organiser should get someone more relevant.

At the lunch break you snaffle all the free food, collect all the free mints from the vendors and then drink copious amounts of coffee. In the evening you drink all the free alcohol, dance like a nutter and then barf on one of the keynote speakers.

You think back to the event with fond memories, then you remember that the sound was too loud, the visuals were so fast you felt queasy and the crowd was hating every minute of it. You vow never to attend another testing conference, as you book yourself on the next one. They are ruining our industry... or are they becoming the best way to spread the word of knowledge?



Job Hunting

Day 1 - Decision made to apply for a new job. Promotions are few and far between and opportunities are scarce in my current company. Now is the time to make the change.

Day 2 - Scouring the internet job boards I begin to wonder whether I am still qualified to work as a software tester. I don't have enough certifications.

Day 3 - I receive 9 billion email feeds for jobs, most of which are located 4 hours drive away and involve a pair of rubber gloves, some spanners and a car ramp.

Day 4 - I fine tune my CV which ultimately results in me rewriting large chunks of it, worrying about whether I used the right tense and depressing thoughts of all of the years I've wasted in jobs I hate.

Day 5 - I sift through the mass of emails and spot a gem of a role. Just down the road, good salary and working for a well known company. Result. I eagerly prepare a cover letter, fine tune the CV further and run a spell check. I realise I



have no testing certification but am confident my 10 years of experience will counter this. After applying for the role I sit back and await a call.

Day 6 – No call

Day 7 – No call

Day 8 - I receive an automated response from the consultant thanking me for my application and informing me that Leonard will be in touch very soon.

Day 9 - No call

Day 10 - Still no call

Day 11 - I phone Leonard who appears to be on holiday but Sarah is now taking over and she promises she will get back to me very soon.

Day 12 - No call

Day 13 – No call



Day 14 – No call

Day 15 - I venture through the email job feeds yet again.

Day 16 - I receive an email from Ted who has taken over from Sarah who is off ill. Ted informs me that I have no certifications and hence am not eligible for the role. I explode with rage and phone Ted. Ted is out at lunch so I speak to Darren, who promises that either Ted, Sarah or Leonard will get back to me.

Day 17 - No call

Day 18 - I apply for a second role. Again, this one requires certification but I'm yet again sure that experience will count for more.

Day 19 - Dave calls me about the first role; apparently Ted has quit. Dave informs me that certification is the single biggest advancement in the testing world as it allows recruiters to sift through the applications for suitable candidates. As Dave can now immediately discard applications, he is confident this adds better value for their clients.



I ask Dave how many applications he received for the role. He informs me that there were 6. I ask why they needed to discard based on certifications when there are only 6. Dave gets riled. I asked Dave why he phoned me back.

If their policy of binning “no certification candidates” means I wouldn’t be put forward for the job anyway, then why call? Dave takes offence at my incessant questions and hangs up. I phone back but it turns out Dave has gone to lunch and only Bill can help me, but his specialty is project management and unless I hold a scrum master certification he really cannot help me.

Day 20 – No call

Day 21 – No call

Day 22 - I receive a phone call from Ted which initially baffles me but it turns out he has moved companies and is now dealing with my second application. He informs me that certifications are the biggest advancement in the testing world and without it; I’m not going to get a job.



Day 23 - I book myself on the certification training course. I pay my fee and sit the course.

Day 24 – I sit the exam.

Day 25 - I sleep all day. I feel a strange sense of violation.

Day 26 - I sit in and drink heavily hoping to blot out the horror of applying for jobs. My wife berates me, my children mock me and I wonder why I bothered changing jobs in the first place.

Day 27 – I sleep all day.

Day 28 - I find out I have passed the certification exam. I feel numb.

Day 29 - I phone Sarah back and explain I now have the certification. She is not there but Sheila informs me that the interview is taking place tomorrow. I panic.

Day 30 - I phone in sick to work and head down the road to my interview. I turn up and am promptly told that it is an assessment day and I will be judged against



the other candidates for all aspects of testing such as number of buzz words I know and how many test scripts I reckon I can run through in a day. It feels more like a Testing Top Trumps than a job interview.

In a mad panic I scan the room and notice that all of my competitors are younger than me and exude confidence in their knowledge of buzzwords. One of them even has the cheek to wear a "Will Twitter for food" t-shirt. What no suit?

The assessment day starts badly.

I have to stand and introduce myself to the 5 other candidates. I try a light joke about certification being like taxes but it does not go down well. In a mixture of excitement and nerves I spill my water down my trousers and now it looks like I have wet myself.

It only takes a minute before my trousers start to itch. I don't feel comfortable asking for more water as the other testers are now staring and holding back giggles. I feel like an easy target and am racking my brain to think of cool buzzwords to drop in to the icy atmosphere.



Next up is the triangle test. I fail as I ask too many questions.

My trousers are now really starting to itch as the assessment day enters the buzzword and Testing Bingo round. "Facebook". "Agile". "TDD". "BDD". "Thinking outside the box". "Lateral diversion". "Twitter". "Lean". "Kanban". "Blue Sky Thinking" I sink in to a depressive state as I contemplate the future of testing.

The other candidates seem happy to randomly shout out buzzwords, tick boxes, provide metrics and certify themselves to the hilt. I don't. I sit quietly festering with rage at the conformity I see before me. In a moment of red mist I stand up and move swiftly for the exit

Day 31 - I realise I am very happy where I am currently working.



Sunset Reviews

The Sunset Review. Oh the joys. That ridiculous meeting that happens at the end of the tedious project with the same tedious people.

These meetings are organized with the best intentions. The problem is very few projects run smoothly and in Sunset Reviews tempers are running high. A blame culture ensues and after about 10 minutes the air is so thick you could cut it with a knife.

After a couple of lose comments and finger pointing a fight breaks out which leaves you dazed and confused. In a scene that wouldn't be out of place in Fight Club you wander around the edge of the room absorbing the scene and making notes ready for the police questioning.

Not knowing whether to join in, run away or call the police, you do the next logical thing and grab your camera to take some photos. These will make excellent additions to your blog posts. The one with the Project Managers in a headlock whilst being wedged by the Tech Lead will be priceless. You could even get a whole presentation out of this. Pure Gold.



Then you get slapped in the face by the Project Manager so you return a volley of cat slaps only to be slapped around the back of the head by the documentation team lead. Pretty soon everyone is punching the Deployment Manager and shouting abusive comments about various members of his family.

It all climaxes with an elbow drop by the Dev Manager on to the Test Lead's head.

After much huffing and puffing everyone is exhausted and decides to sit and relax whilst waiting for the riot police to turn up. Some bright spark suggests we take down some notes ready for the next major release. Overall though, this has been one of the most successful Sunset Reviews the company's ever held.



Meetings

Meetings are exactly why testing is in the state it is in today. Time being wasted in meetings is on the increase and the negativity that accompanies most meetings is enough to drag you down for the rest of the day.

10:00

You pitch up for the 10am meeting eager to discuss project details with the rest of the team only to find that the previous meeting is over running.

10:02

After standing around making over enthusiastic “watch checking” gestures you decide to wander around to the kitchen to make a coffee.

10:05

You return with your coffee. The meeting room is still busy. No other attendees have turned up for your meeting.



10:06

The Project Manager from the over running meeting holds up a finger to indicate he will be no longer than one minute. You resist waving a different finger back.

10:10

The previous meeting finally finishes. Attendees leave the room scowling at you for making them end their meeting before any decisions were made.

10:15

Still no one has turned up. On a positive note though you've bagged the best seat at the back where you can "people watch" through the window.

10:17

With no one turning up you decide to go and make another coffee.

10:18

Whilst making a coffee you suddenly become overwhelmed with a streak of mischievousness and decide to unscrew the lids on all the sugar shakers and balance them on the top. So it looks like the lid is still on.



10:20

You return to the meeting room to find all attendees in their waiting for you. You are gobsmacked but make your apologies. You now only have the choice of one chair.

The one next to the projector at the front.

10:21

The Business Analyst (BA) decides to make a phone call and leaves the room.

10:23

The BA now returns but the Project Manager (PM) decides she now wants to make a call so leaves the room.

10:24

The Project meeting finally kicks off. First agenda point is raised. How to eliminate waste in the project.

10:24:20

You bite your lip and keep your late meeting comments to yourself.



10:26

You start to have heart palpitations from all the caffeine. Your foot starts to tap sporadically. You have an urge to throw rolled up balls of paper at the tech lead.

10:27

Apparently the defect count is down.

10:28

Smiles all round

10:28:30

The low defect count is because the build is broken and no-one is testing. General feeling of disappointment.

10:36

After more pointless defect stats the urge to throw paper is becoming increasingly strong. You now need the toilet. Badly.



10:45

The PM announces, after checking corporate emails on his Blackberry, that the MD has just had a sugar shaker full of sugar dumped in his coffee and demands that the child responsible for this behavior steps forward.

10:45:20

Your face turns grey. You feel faint.

10:46

You start humming.

10:46:20

You are asked to stop humming. And to sit still.

10:48

Discussion moves to how we solve the quality problem.

10:49

Blame session erupts and heated exchanges are made.



10:50

You start to feel quite faint now after more blaming and shouting. You've also noticed how the support manager has a faint smell of Angel Delight about him. Strawberry flavour with a hint of marker pen.

10:52

Someone suggests a coffee break, someone else suggest we just abandon the meeting. The PM suggests we press on.

11:09

Meeting ends after being harangued by the next meeting attendees. The group disperse. You charge to the toilet. No decisions were made.

11:15

You turn up to your training session fifteen minutes late. Today's topic: time management.



Upfront Test Cases

Upfront Test Cases are to blame for everything that is wrong with Software Testing.

As the testing community splits in to two halves and throws eggs at each other about scripted versus exploratory the centre line is ripe for the taking. The centre line is not the medium or the middle ground or sitting on the fence. The centre line is the real world testing that's being done around the world by a large number of testers who are pragmatic enough to realize that there is a time and a place for both scripted and exploratory testing.

It's a sad state of affairs when we have two camps arguing it out about who is right when it's clear that both are wrong.

"Exploratory Testing is ad-hoc, chaotic and unstructured with no way to measure test completeness" I hear from one side.

"Scripted testing doesn't give the flexibility to evolve and learn about the system. It locks in your ignorance" is the retort



"Idiots. Exploratory Testing is so random, unskilled and pointless" comes the battle cry.

"Not at all, at least we don't spend months writing tests only to find that the software doesn't match the tests. Losers." comes the return volley.

As both sides are often completely one sided it leaves the middle ground ripe for the taking. The testing community appears to have forgotten about just doing good testing and is instead praising people for being "out there", "experimental" and "far out" or towing the company line and bean counting, formalising and generalising. So whilst some people spin out to the sides it leaves the middle ground being plugged by the people who just get stuff done.

These people write awesome test cases that are flexible enough to change. They do exploratory testing , generating loads of new test ideas from their sessions. They build on the results and feedback the exploration gives them but have containers and tests for reporting, re-running and communicating to other testers. They can report indicators of progress but are not driven by metrics and they push new and interesting boundaries; not because they want to make



names for themselves or make testing out to be more important than it is, but because they want to deliver good software and do a damn good job. They are experimenting, but in the real world.

So whilst some testers get on, push boundaries and get the job done others are busy writing sterile, boring, complex, weird, dry, flabby, pointless, scientific, futuristic, historical, angry, self loathing and futile test cases or simply coming up with theories that simply don't work in the real world.

Upfront design is the stereotype of testing and it's going to take more than "far out" theory to break it. The testing world needs more centre line people. People who know how to do good testing but also appreciate there's a line between theory and reality. The edges are getting full, let's fill up the middle.



Conclusion

So as you can see, there are lots of reasons why Testing is in the state it's in. People, process, maturity, immaturity and a whole host of other reasons. I can't help but think that we need to sit and ponder some of the challenges we face. We need to meditate on where we are and look for new and exciting ways to push us forward. And the more I do that, the more questions that arise in my mind. But these questions are good. They are questions that could start a discussion. They are questions that could bring about change.

Here are some questions to ponder:

- Are we in need of a revolution? An evolution of the mind?
- Have we been sold a one sided story for years and years?
- Have we standardized people out of free thinking?
- Are lots of people talking loud, but saying nothing?
- Are we a generation with unprecedented access to media, culture and social sources of learning and still not getting it right?
- Or are we seeing a shift? A tide of change? A movement for alternative educations?
- Are we still adding value in our fast moving world?
- Why am I asking so many questions?



Whatever we think about testing and the state it's in, it seems clear to me that the facts matter less than the hype or myths we love to believe. And until we break the stereotypes and mainstream beliefs about standardization and conformity it will always be an uphill battle.

I tend to organize my life around a few core values. One of which I think sums up what we need to do.

“If you want to guarantee change you need to be the one to make the change”

End Note

This book was intended to be a fun and “sniping” look at some of the problems we face as a community. It is not intended to alienate or target any community, company or organisation. Comedy can be a good way to look at complicated issues and learn more about the topic of testing. We need to have conversations about testing and its future.

Please feel free to join the big discussion with others in the community at The Software Testing Club forum. www.softwaretestingclub.com



About the author

Rob Lambert is the Creative Director of [The Software Testing Club](#), the editor of [The Testing Planet](#) and blogs as [The Social Tester](#).

Rob is also an experienced Test Manager specializing in communication, agile, creativity, collaboration and team dynamics. Rob is a regular on the speaking circuit, organizes local user groups in the UK and is a prolific writer.

Rob is a communications graduate who is currently exploring and studying Ethnographic Research, The Semantic Web, Communications and Social Studies. Rob has a passion for Social Media and Communication, is working hard to break down the stereotypes of testing and is pro-active in creating fresh and vibrant content for the testing community.

Rob can be found at the following locations:

Blog: thesocialtester.co.uk

Twitter: twitter.com/rob_lambert

LinkedIn: <http://www.linkedin.com/in/robertlambert>

