

# Tester Types

## An introduction

by *Rob Lambert.*

After many years working as a software tester I have met some incredibly interesting people. Along the way these people have in some way shaped the person that I am today, and for that I am eternally grateful. What these people also did was give me the insight and personalities that make up the basis of the Tester Types. These peers, colleagues and managers have all given me the inspiration I needed to make the 19 Tester Types you see in this book.

I may have exploded their personalities, over exaggerated their behaviour and dramatised their traits but none-the-less these people I met exist in some way over the course of this book. I do hope that all of you will be able to relate to these people in some way also.

Ultimately though, The Tester Types are a bit of fun. They are caricatures of people we may have worked with brought to life through superb

artwork courtesy of The Software Testing Club (Thanks Rosie).

But ultimately this ebook is yours to keep, share and pass on. The ebook is for you to refer to whenever you need inspiration, solice, a good laugh, somewhere to vent your anger or just a bit of downtime.

All we ask is that you abide by the [Attribution No Derivatives](#) copyright, reference the creators and have fun in what you do.

Thanks

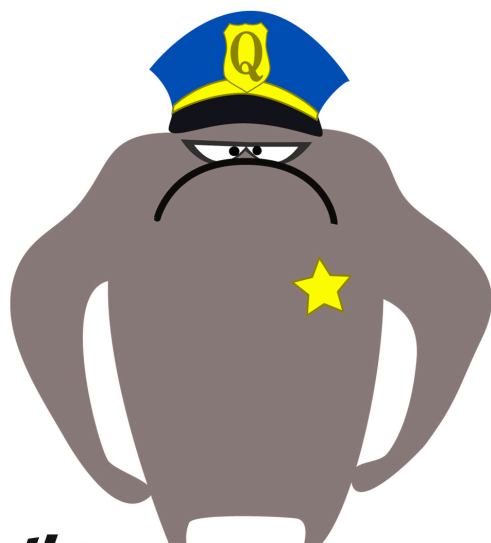
Rob (The Social Tester)

---

IN COLLABORATION WITH THE SOFTWARE TESTING CLUB



THE ENFORCER



*the enforcer*

**Observed Behaviour :** Measuring quality, talking quality, instilling quality, punishing those who lack quality

**Favourite Phrase :** Quality, quality, quality

**Nemesis :** Project Management, Management, Programmers, sometimes other testers

The Enforcers are the awesome force in testing that will stand for nothing else but quality. They are often called the quality police and are at times known to be too quality focussed. They will enforce quality all over, from the build and release to the documentation. Even if no-one asked for their input.

They will rip any spec to pieces, enforce quality in every meeting they attend and settle for nothing but the most ardent quality in every aspect of the software development life cycle. The Enforcer lives, breathes and sleeps quality.

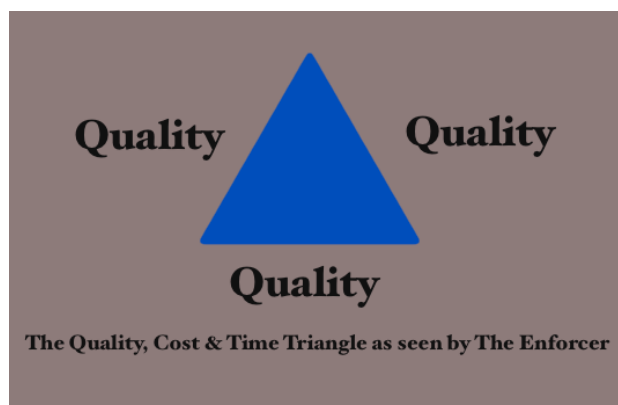
Show The Enforcer the quality, time and cost triangle and all they see are three sides saying 'quality'.

The Enforcer is respected and required on every project but The Enforcer often over-looks commercial aspects meaning they can be known as troublesome, unbending and inflexible. The Enforcer often has a reputation of sticking their nose in or bullying. It's quality or nothing for some enforcers. The Enforcer also believes they are running the project.

It's quite common for The Enforcer to not understand that quality is value to some person who matters, and that as a tester, The Enforcer is not the person who matters most.

Unlike Clint Eastwood though, the testing Enforcer doesn't carry around a .44 Magnum. Well, not all of them anyway...

QUALITY TRIANGLE



**Teststuff**

Test management  
www.teststuff.com

Any type you are - Teststuff is for you!

Try it now for FREE →



## THE DRAMA QUEEN

**Behaviour** : Mass panic when any defect is found, flailing arms, cries of panic, tears of horror

**Favourite Phrase** : Look, Look, it's broken

**Nemesis Team Member** : The Programmer

Drama Queen over-reacts with each bug found. Whether it is a spelling mistake or a catastrophic failure The Drama Queen always reacts the same. The Drama Queen can be seen hurtling towards the programming team, returning with at least 5 of them whilst wildly pointing at the screen and shouting 'Look, Look, it's broken'. In reality it's often 'just a bug' that is fixed immediately with little or no fuss.

The Drama Queen really gets programmers backs up as they are never sure whether their code is really that bad or whether they've just overlooked something. Cry Wolf?



## THE COMEDIAN



**Observed Behaviour** : Laughing, joking, playing practical jokes, having fun

**Favourite Phrase** : You know what would be funny?

**Nemesis** : Test Leads

The Comedian often brings a light hearted approach to testing. They are at work to do a great job, but they are also there to have a bit of fun. The comedian is a funny kind of person who always has a joke and a giggle, even when times are tough. Even after pulling a 14 hour shift the comedian is still on top form. Well, joke wise anyway.

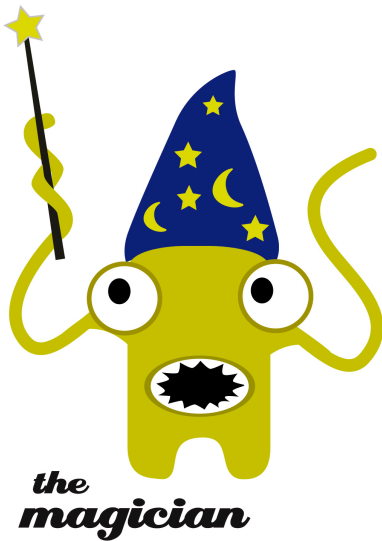
The Comedian often creates hilarious test data and artifacts.

The Comedian's defect reports, although accurate, have been

known to make developers cry with laughter.

At times though, the comedian can appear to lack the professionalism and maturity to move through the testing ranks and be respected by other departments. The comedians main goal is to have fun at work and make the working environment a fun place. And despite the fact they can sometimes be annoying, especially after the 100th practical joke involving rubber insects and a digital camera, you can't deny their approach to work is refreshing.

## THE MAGICIAN



**Observed Behaviour :** Instant bug finding, being in the right place at the right time, magic baby

**Favourite Phrase :** I don't go looking for defects. Defects find me.

**Nemesis :** Project Management, Management, Programmers, Testers

Often called Forrest Gump as they always appear to be in the right place at the right time. They sit down to test an application that's already been tested

heavily and instantly find a show-stopper. Every piece of software they touch, they find a defect in - instantly.

Every tester at some point has a short but magical spell as the Magician but some testers have the skills permanently.

The problem arises when the Magician abuses their power through the following ways:

1. They become arrogant and hold show off sessions. This can lead to embarrassment and resentment.
2. They use their magician skills to hide defects and make disappear all evidence of poor performance. The truth will come out.
3. They turn their project team in to farmyard animals. Although funny, it's often seen as abusing power. Health and safety policy is well and truly breached too.
4. Wearing increasingly outrageous magician outfits and

pointy hats. No one likes fancy dress.

It's at times like this when the Magician is stripped of their power and demoted to that of human tester.

A Magician is never late either. They arrive precisely when they mean to. Maybe this explains the magical bugs.

And although many of us believe we have little control over whether we become a magician or not there are some steps all testers could be doing to increase our chances. These are:

1. Read all you can about the subject
2. Have a strong desire to learn
3. Practice, practice, practice

And if you go about doing this, then one day maybe you too could become a magician.

 **testing**times  
INDEPENDENT SOFTWARE TESTING

## THE CHUCK NORRIS

**Observed Behaviour** : Desk smacking, shouting, swearing, punching, ranting, stomping, kicking and general aggression

**Favourite Phrase** : I'm gonna put my fist through it

**Nemesis** : Anyone with a weak constitution

The Chuck Norris has a bad attitude. The Chuck Norris gets annoyed when they find a bug, even more annoyed when they have to raise it in the defect tracking system and almost apocalyptic if the fix fails. The Chuck Norris can't do any operation on the computer without it getting on their 'goat'. The Chuck Norris hates programmers, detests project managers and positively despises the end user.

The Chuck Norris is known to slam their fist on the desk because the tab order is out, punch programmers because they uttered the words 'Is it really an issue?' and roundhouse their monitor be-

cause the help text was wrong. The Chuck Norris is an animal.

(Note to the real Chuck Norris. Please don't hurt me. <http://www.chucknorrisfacts.com/>)



## THE EXPLORER



**Observed Behaviour** : Stepping from the beaten path, headphones on, lots of note taking, less scripted testing, contemplation poses.

**Favourite Phrase** : It looked interesting

**Nemesis** : The Checklister

The Explorer is what's known as a responsive thinker. They set themselves a charter, often defined from a test case and they explore the app looking for interesting scenarios and paths reacting to information being given back by the app and in turn altering their test ideas.

The defects they find are often show stoppers, are truly inventive and are often so difficult to fix that programmers have exploded in rage. The Explorer can test anything, in any state and at any time. They aren't bound by

the constraints of a checklister. They are free. They explore.

The downside to The Explorer though is that they can often be a maverick. They often have their own ideas which they pursue to the detriment of the rest of the team. Some Explorers refuse to believe that predefined test cases are valuable, some refuse to believe checklisting is even considered testing. However, it's increasingly becoming common knowledge that Explorers and exploratory testing are an invaluable addition to the testing process.

## THE CHECKLISTER

**Observed Behaviour:** Checking, bean counting

**Favourite Phrase :** But the test case didn't say to check that

**Nemesis :** The Explorer

There are two extremes of checklister. One negative. One positive. And those inbetween.

At one extreme we have the stereotypical image of a software tester. An untrained, robotic person who simply runs pre-defined tests checking boxes and adding to the bean count. They don't question the test or the software or the spec and they certainly don't explore off the path. Do these testers \*really\* exist? Apparently so.

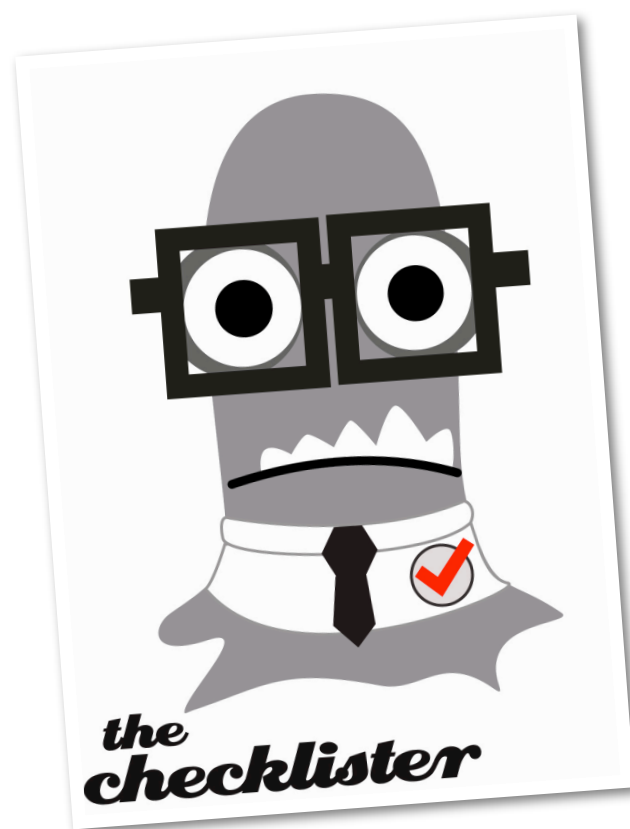
This portrayal of software testing is the one that leads people to believe that testing can and should be automated. That it's unskilled, unthinking, tedious, boring, repetitive....

At the other end of the extreme are those who use scripted tests as a reminder, a prompt, a true checklist or a guide of what to test. It's a source of information that helps guide the human tester to find defects, not a document that needs updating constantly. Checklists and scripts ensure testers don't miss stuff. That they share knowledge. They add structure. That they can report on metrics when needed but aren't constrained by them.

These testers mix exploration with scripted at a balance which suits the project, their skill level and

the needs of the project. These testers write tests junior testers can run whilst the tester learns and explores at the same time. And this image is a positive one. In fact, it's a great portrayal. Unfortunately, it's not the most common view in the industry.

The checklister is such a powerful image of testing that it's proving tricky to break. If only we could convince people (testers, managers and programmers) that the robotic, bored, undervalued bean counting checklister isn't all that testing is about.....



BECAUSE SOMETIMES A CHECKLIST IS NOT THE BEST APPROACH

Practitest - A QA Management System  
for YOU!

No Installations. No Risk.

[www.practitest.com](http://www.practitest.com)

PractiTest

THE SAGE



*the sage*

expense of others. They never shout and never interrupt. They listen, consider and then respond with wisdom and kindness.

The rest of the office never know the Sage is at work unless they need to speak to them such is their quietness and serenity.

The Sage always has a beard (if male) . They know everyone, everything and have hilarious, yet meaningful stories to tell. But they never force their information on others.

The Sage has always written at least two books.

**Observed Behaviour :** Pipe Smoking, contemplating, meditation

**Favourite Phrase :** Hmm. Let me tell you a story

**Nemesis :** Experts and Intellectuals

The Sage is a software tester whose been around the block a few times and survived to tell the tale. They are most likely in the older years of life and probably fought in at least one war. Methodology war that is. The Sage is quiet, reserved and contemplative. They are an expert but are humble about their status. They are an intellectual but not at the



THE SOCIALISER



*the socialiser*

**Observed Behaviour :** Laughing, chatting, enjoying their work and having fun

**Favourite Phrase :** Let's get a drink and chat about it

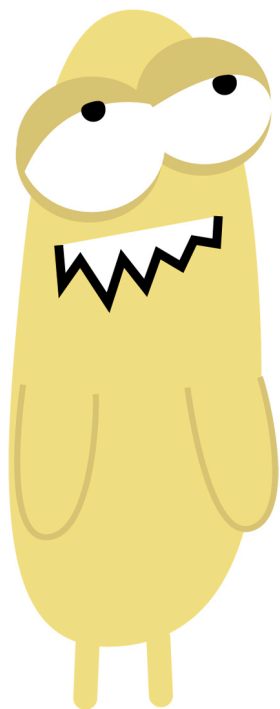
**Nemesis :** Very few people

Every team should have a Socialiser. The Socialiser is the tester who enjoys being at work and has a laugh. The Socialiser doesn't just go to work, sit there, do the job and go home. They want to have a laugh and they want to make friends. They often organise social events and are very effective at bridging the gap between teams or team members.

The Socialiser is often referred to as The Moral Officer due to their key role as team night out organiser and coffee break manager.

The Socialiser sometimes needs a kick in the right direction when their social life is starting to take precedence over work.

## THE DAY DREAMER



## *the day dreamer*

**Observed Behaviour** : Snoozing, staring, posturing, thinking and concentrating on anything but the job in hand

**Favourite Phrase** : When I find my real calling in life

**Nemesis** : Every tester who enjoys their job

The daydreamer is generally a tester who doesn't want to be a tester. They all have at least one of the following stories:

1. They thought testing would be easy and quickly realise it's not
2. They couldn't hack it in another department so moved to testing
3. They landed in testing by mistake and then got stuck
4. They once enjoyed testing but got ground down by bad management, boring test approaches and lack of respect

They spend more time thinking about their lunch, their car, their holiday, their aspirations and their evening activities than they do about the job in hand. They are a lost soul looking for meaning and they're finding no meaning at work. I should know, I was once a Day Dreamer myself.

Some of these Day Dreamers get the testing bug and are hooked for life. Others re-ignite their passion for testing whilst others fizzle out in boring "QA" jobs. Others move on to other careers in other industries whilst all the time searching for the real meaning of life.

The Day Dreamer is an interesting character to have on the team. They often get lumbered with the boring and uninspiring jobs which adds to their ever increasing disdain for testing. They are neither uber effective nor completely irrelevant. For most Day Dreamers a good manager or mentor is all that is required to bring them back on focus. For others, sadly, a move to another job is the best thing they can do.

We all have moments of day dreaming, often with positive effect. After all, a break to gather thoughts, reflect and switch focus is healthy. The problem arises when you start to day dream all day...

### THE MAGAZINE BY THE SOFTWARE TESTING CLUB

Check out the all new magazine by The Software Testing Club.

Visit our website to see how you can contribute:

<http://magazine.softwaretestingclub.com>



## THE BOSS



**Observed Behaviour** : Chilled, relaxed, in control, great communicator and has the respect of the team

**Favourite Phrase** : Not a problem

**Nemesis** : Senior Management

The Boss is essentially the opposite of the Micro-manager. The Boss gets R.E.S.P.E.C.T. The Boss can say 'No' to management and often does, which

makes them unpopular above, but truly respected by those below. But The Boss doesn't see anyone as being below or above. They just see a team and a set of goals and respect all for their opinions and skills. And if The boss has goals that are vague or impossible, The Boss will say so.

The Boss picks a team of people who will work well together, not just individual talent. The Boss picks a well balanced team of experienced and junior team members. The Boss considers the social make up of the team, rather than the individual team members. The Boss understands that junior team members need to learn and often have the most creative minds. The Boss understands that teams can make or break a project.

The Boss buys cakes and beer for their team. The Boss trusts every member of their team. The Boss is fair, yet critical at the same time.

In other words. The Boss is a leader...an inspirer. They have integrity, honesty and trust. They stand up for what they believe in, but are not arrogant or misguided. And if you find a boss like this, you are very lucky indeed.

# PRETTYGOODTESTING®

**PrettyGoodTesting is dedicated to test and QA**

We build our business on more than one third of a century within test and QA – and we intend to stay focused on providing our clients with sustainable assistance solely within this profession.

[www.PrettyGoodTesting.com](http://www.PrettyGoodTesting.com) **DEDICATED AND INDEPENDENT**

## THE AUTOMATOR

**Observed Behaviour** : Automation, automation, automation

**Favourite Phrase** : But I can't do that in this testing tool

**Nemesis** : Manual Testers

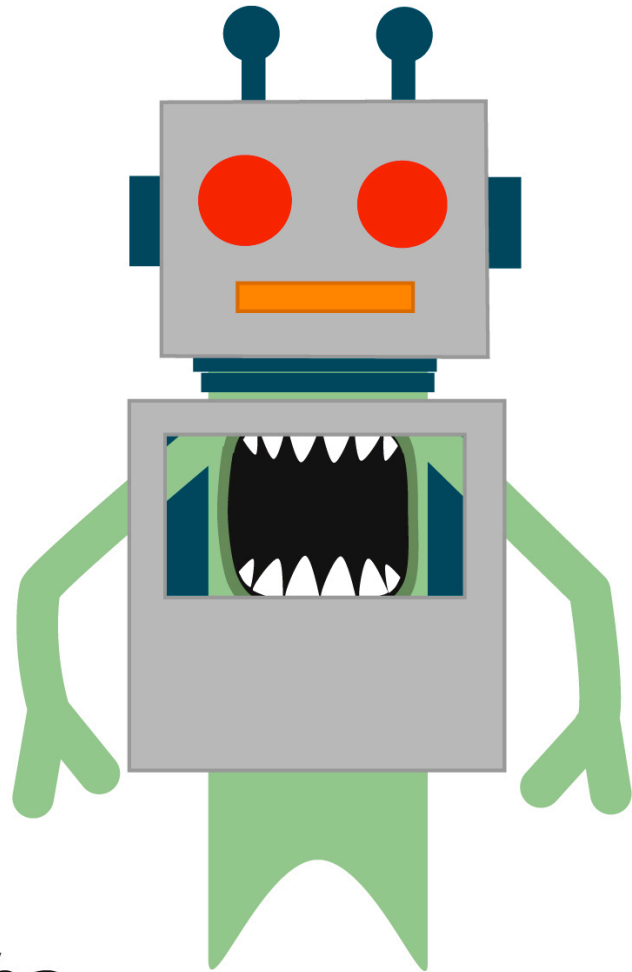
The Automator is a really interesting character. They live to automate. They automate everything. They talk like a programmer and act like a tester.

They are great when they arrive on earth in the form of a toolsmith who knows just enough about everything to be helpful. A toolsmith knows how to code and has an understanding of testing. Often a deep understanding and experience of testing. And they are good coders. They solve testing problems with tools. And not just automate tests. If it needs doing...we'll find the right tool to do it.

It's when they arrive on earth in the form of a one tool tester that they start to lose their intrinsic appeal. The one tool tester only knows QTP or Winrunner or any other of the big vendor tools. It's not that they couldn't understand, learn and use oth-

ers, it's just that it would go against their code of automation ethics to use anything other than their beloved tool. They often have brand loyalty, not test problem solving loyalty. If it can't be done in this tool...it can't be done.

I love the Automators. They are unique amongst the development community. Part tester, part programmer. Part human, part machine.



## *the automator*

AUTOMATION DOESN'T NEED TO BE DIFFICULT

For innovative, automated testing solutions that also empower

Manual Testers visit <http://www.origsoft.com/tester-types> today!



Original Software

## THE WANDERER



**Observed Behaviour :** Wandering

**Favourite Phrase :** Just nipping over there

**Nemesis :** Test Leads and Managers

The Wanderer is an amazement to most testers. They appear to do nothing except wander around the office or the car park. They always have a notepad in hand and always appear to be off to a meeting. Yet all they do is wander. It's amazing how they manage to remain employed. Either they have less work than most or they are simply uber efficient.

The Wanderer always looks busy though. The notepad and purposeful walk give off an air of importance, when in reality they are probably off to the carpark for a crafty cigarette or off to the toilets to play on their mobile phone. The Wanderer rarely talks

to anyone and always avoids eye contact with seniors.

At times all of us fulfill the role of The Wanderer. In small doses wandering is good for the mind and body. It allows time to clear your mind, ease the strain from the screen and get some fresh air. But when 90% of your time is wandering and 10% is working it's time to question your real motivations.

## THE MICROMANAGER



**Observed Behaviour :** Bean counting, spying, backstabbing, whispering

**Favourite Phrase :** Where are my metrics?

**Nemesis :** All testers

The Micromanager is normally a test lead or manager who isn't fully in charge. They have a team and often a set of testing goals. The goals are normally vague and impossible like 'no defects by

release' or 'exhaustively tested' or 'use all best practices'. The Micromanager often misses the most important trait that sets 'The Boss' aside from them. It's called guts. The Micromanager can't say 'No' to management. They simply bow, accept and then rain terror on their team.

So they bean count. They spy on their team. They micro manage. They back stab. They always blame someone else. They whisper. They plot. They always pick the 'best' and most experienced people and stick them in their team regardless of team moral or dynamics. They have favourite team members, more often than not it's The Expert. The Micromanager says things like "Come on, let's work as a team and do it my way" or "It's not a blame culture, but it's definitely their fault"

The Micro manager doesn't trust anyone because they know, deep inside, that the team will achieve great things without them.



## THE NITPICKER

**Observed Behaviour** : Continuous cross checking of all documentation with application

**Favourite Phrase** : Yeah, but this email you sent 2 years ago states.....

**Nemesis** : Everyone

The Nit-Picker takes longer to test than most, raises an enormous amount of defects of which only 1% are genuine bugs. The Nit-Picker raises defects for everything that isn't documented, even if it works and makes sense. They use the medium of defects to talk to the rest of the project team. They often raise several instances of one bug, often with the sole aim of raising their bug count. They take a long time to test, cross reference every piece of documentation and craft beautifully detailed bug reports.

On the plus side, they are extremely thorough.

The Nit-Picker is not receptive to change though. The Nit-Picker requires everything to be docu-

mented and every change to be written down. Any conversation that takes place needs a signed paper follow up of all agreed details. Change controls terrify the Nit-Picker. Missing documentation is a Nit-Pickers worst nightmare.

The Nit-Picker often spends a lot of time producing cross reference documents and matrices that on the surface seem useful but end up costing time and money to maintain. The Nit-Picker believes a standard set of metrics should be applied to all testing projects. The Nit-Picker loves Best Practices. Often the Nit-picker focusses on the things that don't matter and argues about irrelevant points until the cows come home.



Working with a Nit-Picker is a challenge. They not only require a constant feed of documentation but often refuse to start work until the documentation is signed off in triplicate under the false belief testing cannot start until an agreed date and drop of software. They require patience, time and paperwork and this often proves too much for most testers.

## THANK YOU TO OUR SPONSORS:

Testing Times - iMeta - PractiTest - Teststuff  
Agile Testing Days - Original Software  
Pretty Good Testing

## THE EXPERT



## *the expert*

**Observed Behavior :** Interrupting and explaining how things work

**Favourite Phrase :** I know all about that

**Nemesis :** Everyone

Like most of the tester types there are two sides to each of them. Often a positive and negative side. And The Expert is no different.

The positive Expert is someone who is undoubtedly an expert in

their field. They know their stuff. They are "The Oracle". And they are always willing to help. We all no doubt strive in some way to be an expert in our chosen fields, technologies or roles. And once we reach The Expert level many of us are humble and helpful.

However, the negative Expert is someone who feels they are an expert when in reality, they may not be. And if they are an expert they ensure the whole world knows about it. The kind of per-

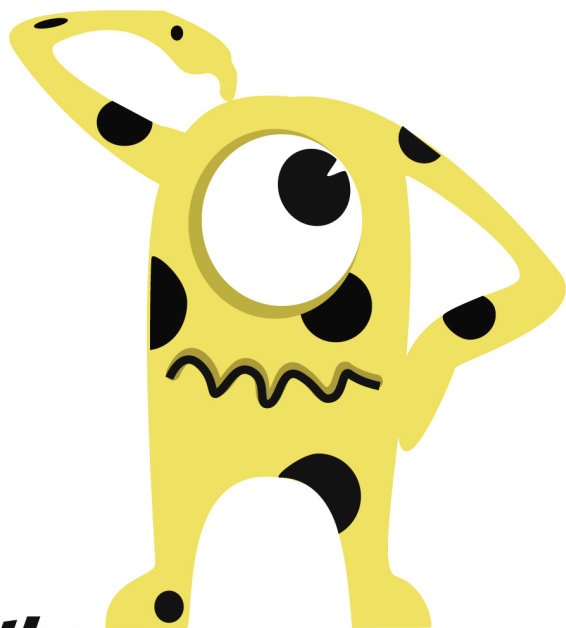
son who you only know is at work because you can hear them....being experts. But you can only hear them when they can jump in to a conversation and be The Expert. Otherwise they sit quietly, waiting for the right conversation. Waiting to pounce and show off about how much they know.

The negative Expert is often known to be randomly talking away when in reality they are simply still responding to a conversation that ended over 4 hours ago. If The Expert is ever proven wrong they normally respond with "It used to work that way" or "The Oxford English Dictionary is wrong" or "I was actually talking about a subtle version of that which is why what I was saying seemed wrong"

So in aiming to be an Expert it would be prudent to understand how to use those skills once at that level. Help people or shout about how good you are? You decide...

  
**softwaretesting**  
 a community for software testers

## THE DRAFTER



## *the drafter*

**Observed Behaviour** : Initially confused look as to what exactly they are doing working as a tester

**Favourite Phrase** : I'm not really a tester, I'm just helping out

**Nemesis** : Seasoned testers

The Drafter is a member of another team brought in to test.

Sometime Drafters are brought in under the misconception by management that "anyone" can test.

Sometimes Drafters are brought in simply to help out.

They can come from any other department in the company; admin, finance, programming, support, documentation, sales, etc

Some Drafters continue down the testing career even though they may lack the critical thinking and passion needed to cut it.

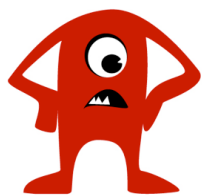
However, many Drafters get the testing bug, find they like the job, find they have that natural aptitude to root out bugs and then move on to become superb testers.

Others just fizzle out until the company they work for either

1. goes bust or closes
2. hire someone else who makes The Drafter look incompetent
3. the Drafter retires

Of the 3 testers consulted in my scientific study, a whopping 100% were originally drafters. And they rock.

The time between being called a Drafter and becoming a fully fledged tester can vary per person, but generally speaking as soon as you start defending quality and asking difficult questions then you can safely assume you've made the move.



### STUCK? GOT A QUESTION? TESTING RELATED?

The Software Testing Club Exchange is a community driven knowledge exchange for Software Testing.

Visit us: <http://exchange.softwaretestingclub.com>

## THE NETWORKER



**Observed Behaviour** : Networking, talking, wandering

**Favourite Phrase** : I know someone....

**Nemesis** : No-one (if they are sensible)

The Networker basically knows everyone, everywhere. You have a problem with Active Directory, the Networker knows the girl who wrote part of it. You have a problem with Oracle databases. The Networker's brother designed it. The Networker knows everyone inside and outside of the company. They have contacts everywhere.

Other testers should work out who The Networker is on their first day at work, they are an invaluable resource. The downside of working with The Networker is that anyone you talk about The Networker to, knows The Networker. So you need to be careful about what you say. Nobody dislikes The Networker - not publicly anyway.

Building your own network can be an invaluable process though. Information within a company often flows through the most unlikely of sources and being able to tap that information when needed can often make the difference between success and failure.

# PRETTYGOODTESTING®

**PrettyGoodTesting is independent because you need optimal solutions**

Although our consultants are updated in most models, methods and tools on the market (including Open Source), we insist on staying independent. This, we believe, is the best way to remain your trusted advisor within test and QA, and to bring you good value every time.

[www.PrettyGoodTesting.com](http://www.PrettyGoodTesting.com) **DEDICATED AND INDEPENDENT**

## THE INTELLECTUAL

**Observed Behaviour** : Spends lunch times writing poetry, reading foreign language books and studying science

**Favourite Phrase** : "The end of centrally engineered one size fits all initiatives, an intelligent approach to cutting red tape through redesign of processes and cultures, a renewed emphasis on strategic development so as to better equip our testing to meet the amorphous challenges of managing cross platform bugs, risks and opportunities"

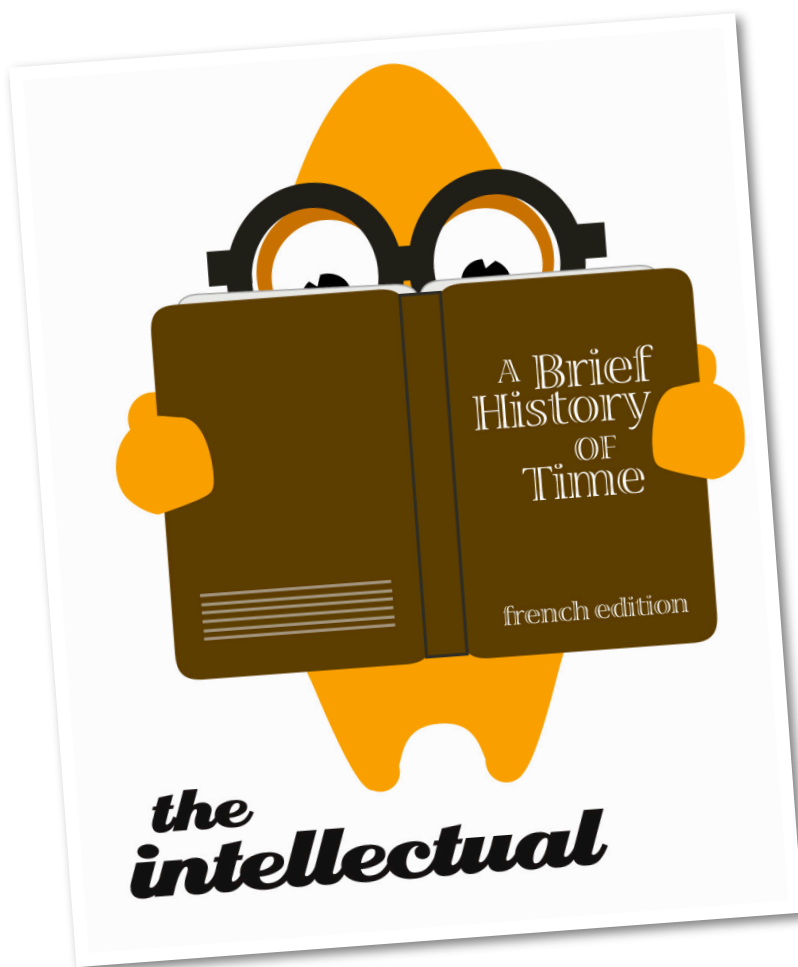
**Nemesis** : Anyone who doesn't understand them

The Intellectuals are great. They are not only good testers but they also make your average programmer feel stupid (not an easy thing in my book). A conversation between an intellectual programmer and an intellectual tester though is something you may only get to hear once in your career. It will also either send you to sleep or knock you unconscious, either of which could be more enjoyable.

The problem with the intellectual tester is that they use words, phrases and sentence structures that don't mean anything to most people. I need

three hours, a dictionary, a thesaurus and Google translate to work out what the intellectual is trying to say.

The Intellectual and The Expert go well together. The intellectuals use of long words and jargon automatically makes them an expert (as no-one can truly understand what they say - except other Intellectuals). But not all experts are intellectuals.



### TESTING DOESN'T NEED TO BE COMPLICATED TO BE EFFECTIVE

The test platform that is flexible, intuitive and methodological.

Can't believe it? Try it yourself Now Free!

[www.practitest.com](http://www.practitest.com)

PractiTest



---

**CONTRIBUTORS****ROB LAMBERT**

Rob Lambert has been in the testing field for around 10 years now during which time he has worked on a variety of software products and has helped deliver a number of high profile software solutions. Rob has managed teams both large and small, worked in many areas of software testing (functional, non-functional, integration, UAT etc) and more recently has been involved in several agile projects.

Rob is primarily interested in the communications and people side of software development. His focus is on helping and guiding teams who are moving from traditional to agile methodologies by helping to improve communication and team dynamics. Rob also spends considerable time promoting communication within software testing and the process of Kaizen (improvement).

Rob is heavily involved in the software testing community and has a keen passion for social media and collaboration. He has recently become a community manager of the popular Software Testing Club site, contributes hints and tips for @DailyTestingTip on twitter and is active in writing articles and white papers for several leading testing magazines.

Rob is the founder of Testing Revolutions, a test group dedicated to expanding the understanding and use of social media in software testing. He is also building and developing a new concept regarding communication and team interaction called PAC Testing. (Purpose, Audience and Context Testing).

Rob runs The Social Tester (<http://thesocialtester.posterous.com/>) blog where the Tester Types originally appeared.

He is a regular in the testing community, speaking, running workshops and holding workshop sessions.

But most importantly Rob is trying to inject a bit of much needed fun in to software testing.

In his spare time Rob writes screenplays, novels, short stories and TV shows (none have been successful though...). He lives with his wife and son in Winchester, UK.

---

**ROSIE SHERRY****SOFTWARE TESTING CLUB**

Rosie started out in the software testing world 10 years ago and along the way founded the Software Testing Club - a lovely and professional community of practice for software testers.

Rosie's interests lies within the complex mixture of software testing, user experience, the social web, education and community building.

You can usually find Rosie at:

- [www.softwaretestingclub.com](http://www.softwaretestingclub.com)
- [www.twitter.com/rosiesherry](http://www.twitter.com/rosiesherry)
- [www.rosiesherry.com](http://www.rosiesherry.com)
- [www.schux.com](http://www.schux.com)