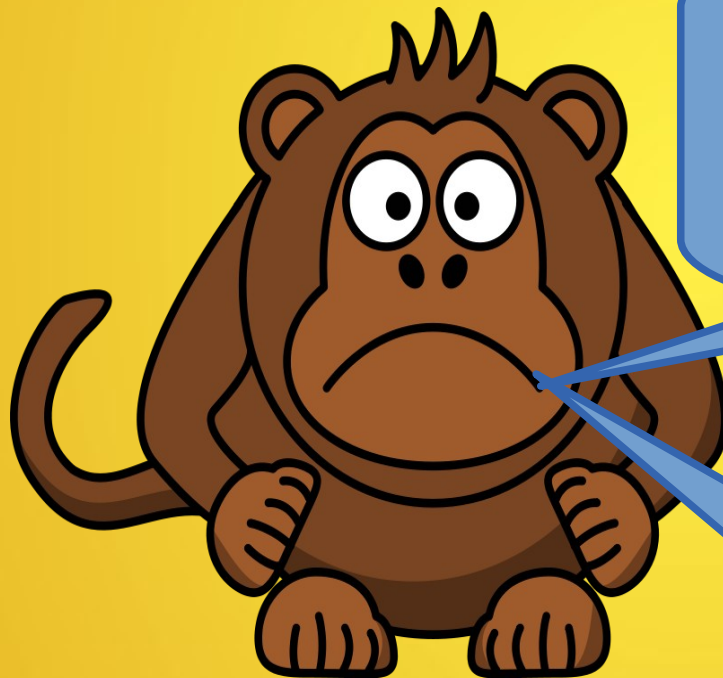


The Bane of Being A Test Automation Engineer



No one understands me!

So my psychiatrist suggested
I do this presentation.

Too hard to do properly?



Management regard test automation as an after thought. Don't plan for it.

Management are too focussed on development and developers so the testing processes often originate from developers who write a script or program – think its easy. Anyone can write a script/program but very few can build a complete functional test system that “just works”.

Management don't understand the required level of integration

- Test on a production like system.
- Be able to determine exactly what change (code, deployment, database etc...) occurred between a test passing and later failing.
- Test exactly what you ship. Need a pipeline.
- Fully automated means you spend most of your time analysing results or adding/improving tests eg. Bad use of CI tools – convoluted display of test output. Displayed success does not mean tests passed??WTF?!!\$!!

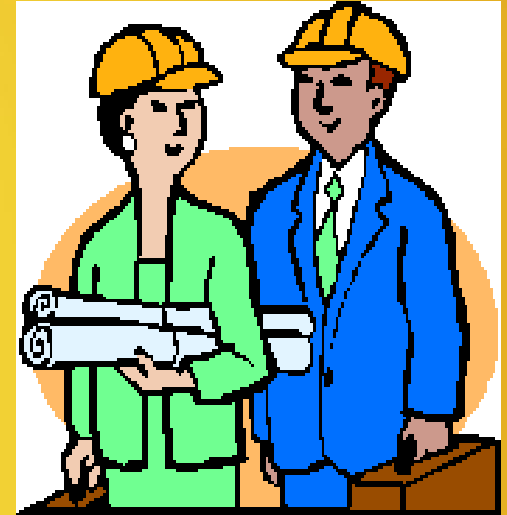
Test Engineer Vs Test Analyst

Skills gap too great. Supporting low tech skilled analysts is a burden.

Everyone should have a minimal level of technical skill especially system administration and development skills. Silo effect.

As the systems we build get more complicated the engineering effort for deployment and test automation goes up. But this is not often taken into account.

Test engineers are wedged between developers and ops people.



Some Questions to Ponder

How do you scale your software engineering effort?

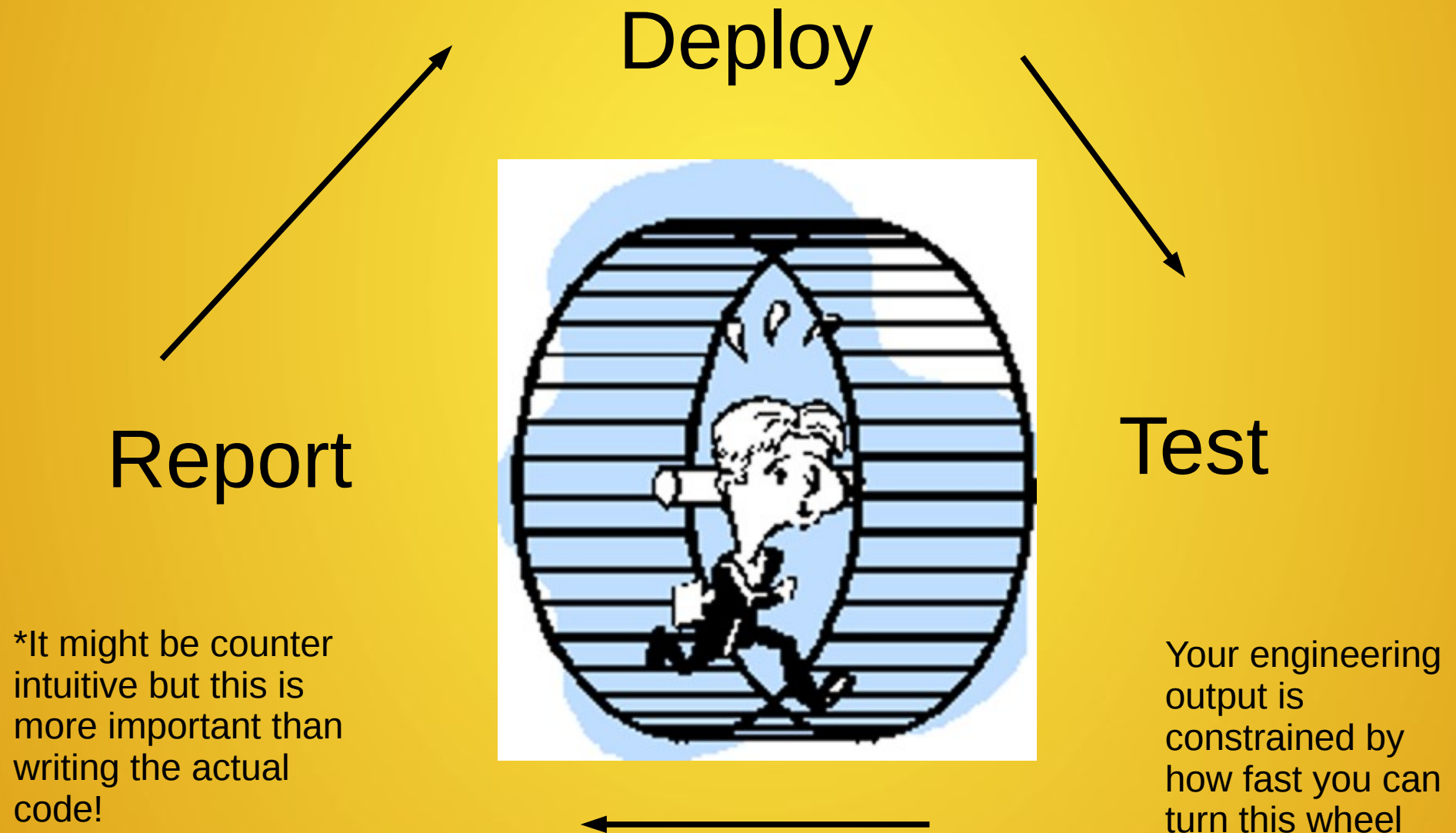
How to evaluate how good you are at delivering software and where you need to improve?

What must non-technical managers and interested parties understand about the software development process?

How to do Agile really well?

[Hint: Answer is on the next slide... really!!]

Simply put what we do all boils down to this*

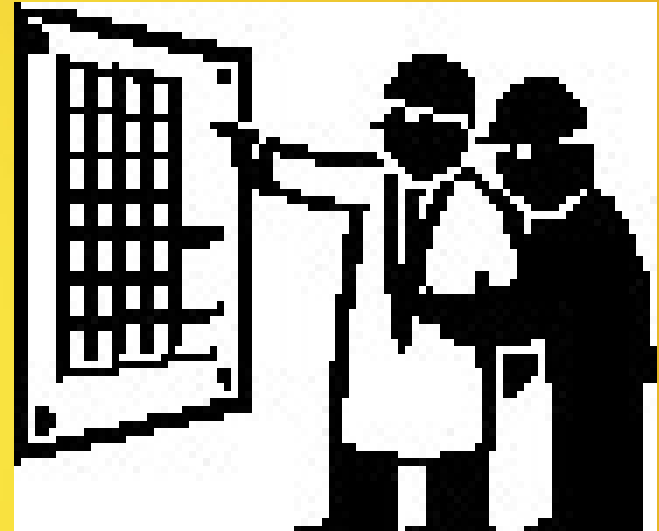


How to do my job well

We have moved from a shrink wrapped to a web hosted model of software distribution. More rent, less buy.

Support parallel stream of development – what scaling actually means.

A test engineers job also includes deployment automation as its fundamental in being able to perform tests efficiently.



I know this is self serving but ... We have Principle Software Engineers, so why not Principle Test Engineers as well!

Test Ops – Test and Deployment Automation (plus reporting) using production like systems.

Dev Ops concept is broken and obsolete!!!

A Single Coherent Vision

Test Ops		
Development	Testing	Ops
Fully automated deployment Multiple deployments Test tool integration	Testing infrastructure CI systems Reporting systems Testing systems and tools	Production pipeline Versioning

Dev Ops concept is obsolete. Need to have a single team that covers all of this!

Good Reporting

The changes shown in the 'Guilty' column should reflect every change to your production like test system and not just code changes.

The tree is currently CLOSED

Build Time	Guilty	Mac Debug Cibr	Mac Opt Cibr	Mac Opt Dep	Win32 VC6.0Cibr	Win32 VC6.0Dep	shrike Linux Cibr	shrike Linux Dep	speedracer SunOS/sparc 5.6 Dep
Click time to see changes since time	Click name to see what they did								
03/07 09:59									L
09:54				L		L		L	
09:51					L			L	L
09:50									L
09:46						L	L		
09:36			L					L	
09:35		L						Lk:43K Bl:21M	
09:31		L		L					L
09:28									
09:27								L	
09:25					L			Lk:43K Bl:21M	
09:22					L				L
09:21									
09:19						★ L		L	
09:15				L				Lk:43K Bl:21M	
09:07				L					
09:06								L	L
09:03								Lk:43K Bl:21M	
03/07 08:54							L	Lk:43K Bl:21M	★ L
08:49			L		★ L		L	Lk:43K Bl:21M	
08:46			Warn:456				Warn:265		
08:44	rods							Lk:43K Bl:21M	★ L
08:41		L						L	★ L
08:38		Warn:156		L				Lk:43K Bl:21M	★ L
08:32								L	
08:27								Lk:43K Bl:21M	
08:26								L	
08:21					L			Lk:43K Bl:21M	L
08:11									L
08:10								L	
08:09				L				Lk:43K Bl:21M	
08:04									
08:02									

You want to see the link between build times, status and changes on ONE page

You should be reporting test status accurately so the colours actually mean what they imply

Guru Meditations*

Student: Our product would have shipped on time if only we had spotted that bug earlier.

Guru: To make that bug visible earlier you need to reduce the time and labour it takes to deploy a “production like” system and run your test suites. The more iterations of this you can do in a given amount of time the higher the chances of a show stopping bug being identified in time to fix it.

Student: I don't think our Sales and Marketing team know enough about our product. What can I do?

Guru: A “production like” system is no “holy cow”. In the same way that testing and ops need to deploy such a system with minimal effort so do other teams. The best way to learn is by doing, so make it easy for them to do the doing.

Student: Thank you. I will think about this more.