



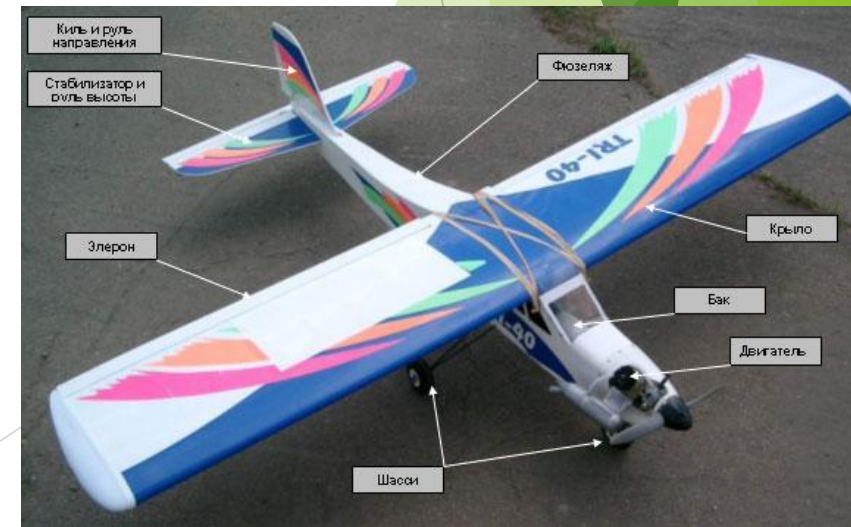
# Quest for an ultimate autotest

...Coverage?

A. Pushkarev 2018

# Spoiler

- I hope that **nothing** in this presentation is new for the audience
- The real projected value is to put things we **already** know together



# Who am I?



Alex Pushkarev

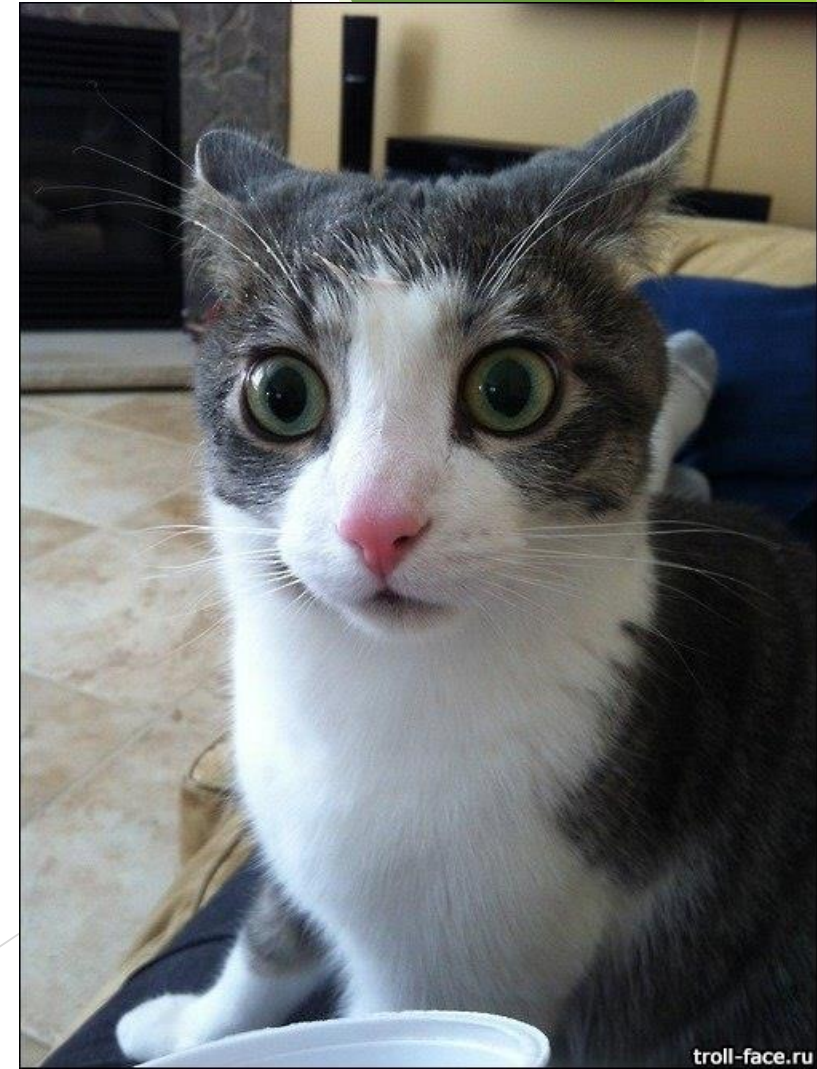
- ~ 11 years in IT
- Test focused developer/Development focused tester
- Agile fan, XP practitioner



# Why do we need “ultimate” coverage recipe?

**-> We don't!**

**-> We just want our tests not to be useless...**



# Why do we need [automated] tests?

-> We don't!

-> We need a stability  
and certainty...





Mission: gather data and information



# “Best practices” will fail you

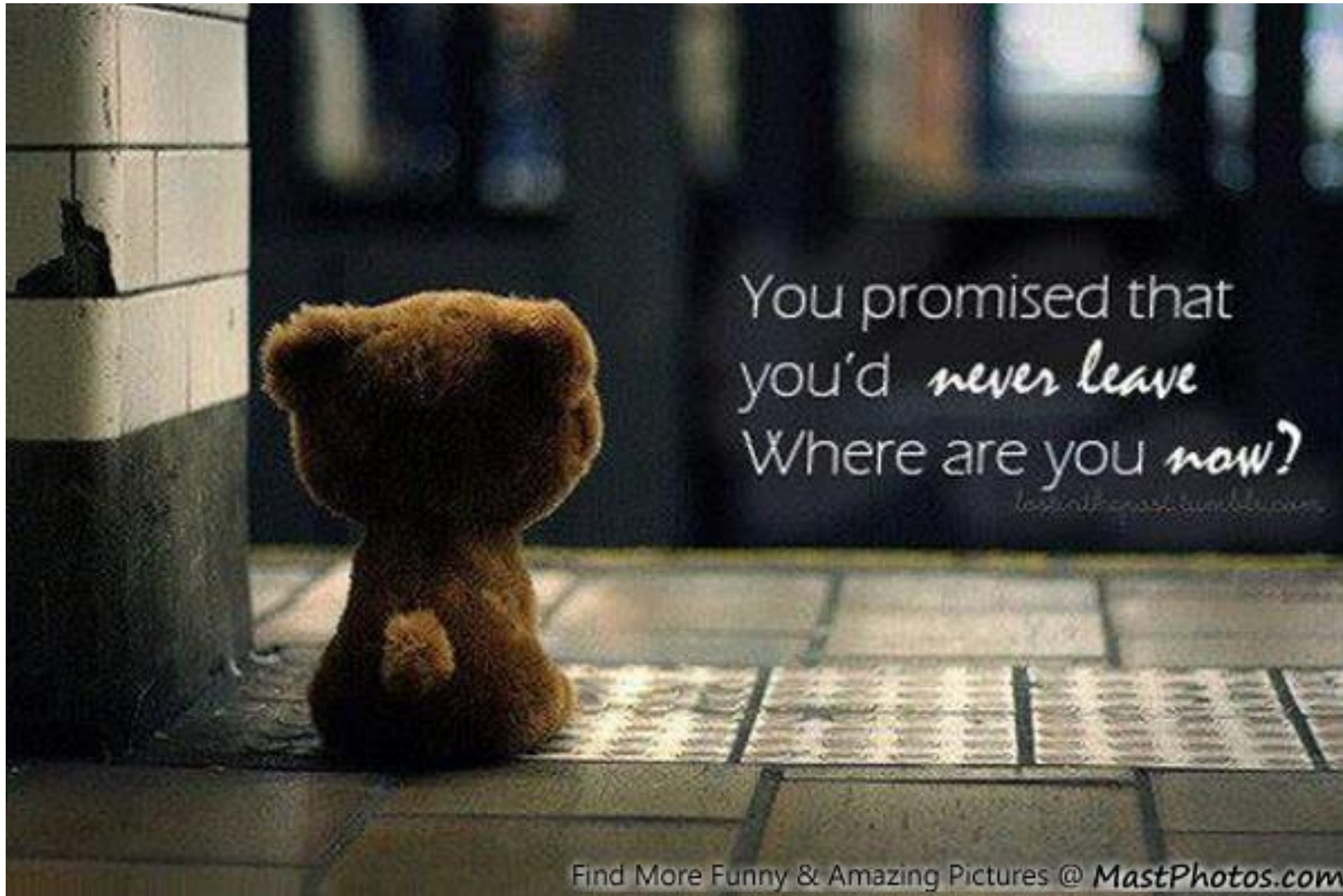
-> There's always a context where best practice will fit

-> And won't fit





# Figure out the context





# Figure out the context

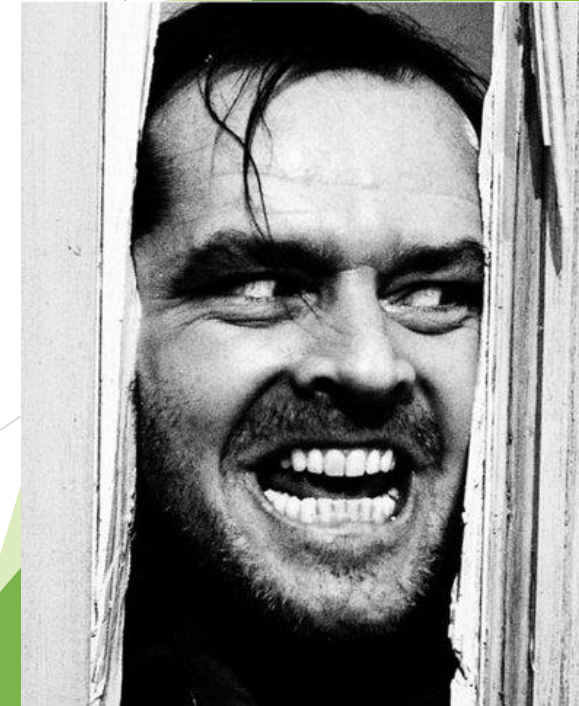


- Agile never meant we should not have a strategy





















# Adequate test automation coverage

- Can be measured via lines of code (*i.e. Code coverage*)
- Can be measured via feature\stories mapping (*i.e. traceability matrix, feature coverage, etc*)



# Code coverage tools

- JaCoCo

JaCoCo											
JaCoCo											
Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed
 <a href="#">org.jacoco.agent.rt</a>		81%		86%	32	106	57	262	24	70	7
 <a href="#">jacoco-maven-plugin</a>		90%		83%	18	114	23	276	2	66	0
 <a href="#">org.jacoco.core</a>		99%		99%	25	864	30	2,022	17	506	0
 <a href="#">org.jacoco.report</a>		99%		98%	7	446	8	1,107	2	306	0
 <a href="#">org.jacoco.ant</a>		99%		99%	4	137	8	385	3	96	0
 <a href="#">org.jacoco.agent</a>		85%		75%	3	11	5	30	1	7	0
Total	514 of 16,926	97%	42 of 1,151	96%	89	1,678	131	4,082	49	1,051	7

Code Coverage Report for JaCoCo 0.6.3-SNAPSHOT

Created with JaCoCo 0.6.3.20130301-1402

- DotCove

- CodeCov

# Feature coverage

Requirement ID	Requirements tested	Use case 1.1	Use case 1.2	Use case 1.3	Use case 1.4	...
Test cases	34	2	4	7	5	
TC1.1.1	2	X		X		
TC1.1.2	1				X	
TC1.2.1	3	X		X	X	
TC1.2.2	1		X			
TC1.2.3	2			X	X	
...						



# How to misuse code coverage

- > Low coverage may show there's a problem
- > High coverage does not prove there's no problem

<http://www.exampler.com/testing-com/writings/coverage.pdf>



# What is adequate?

Minimizing risk	Limited resources	Minimizing time
<ul style="list-style-type: none"><li>• Duplication is alright</li><li>• Both code and feature coverage are used</li><li>• It is OK to have separate QA team as a gatekeeper</li></ul>	Common sense?..	<ul style="list-style-type: none"><li>• Duplication is not acceptable</li><li>• Separate teams are waste</li><li>• It is ok to use only code coverage</li></ul>



# What is adequate

## Minimizing risk

- Duplication is a
- Both code and coverage are u
- It is OK to have separate QA te gatekeeper



"Gentle technical discussion on IRC channel"

Francisco Goya

Oil on canvas

time

on is not  
le

teams are

o use only  
erage

# Risk-driven test automation coverage

- > We don't need coverage
- > We need a certainty...
- > That a particular bad thing does not happen





# Two different views on Test Automation



**Alexander Pushkarev** @AQAguy

31 Aug

Replying to @AQAguy

6. That is why I think developers (not some automation guys) should be responsible for writing test automation for application.



**Ayush Trivedi**  
@ayushtrivedi

That's like asking a pharma company to say their drugs are safe without any independent audit.  
[#softwaretesting](#) [#CIO](#)

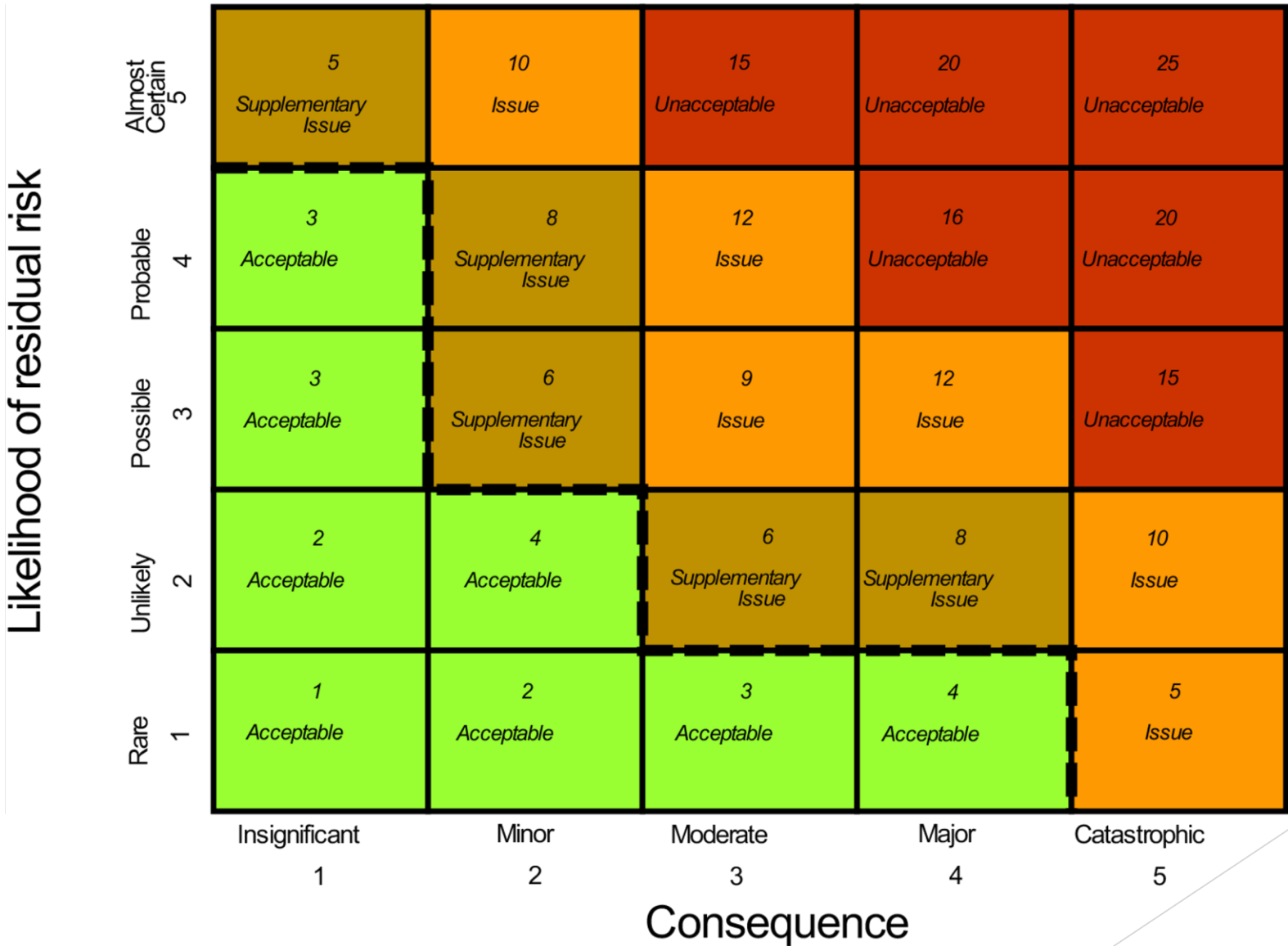
23:21 - 4 Sep 2017



See Ayush Trivedi's other Tweets



# Risk-driven test automation coverage



# Testing dashboard

Testing Dashboard				Updated: 2/21	Build: 38
Area	Effort	C.	Q.	Comments	
file/edit	high	1	--		
view	low	1+	☹	1345, 1363, 1401	
insert	low	2	--		
format	low	2+	☹	automation broken	
tools	blocked	1	☹	crashes: 1406, 1407	
slideshow	low	2	☹	animation memory leak	
online help	blocked	0		new files not delivered	
clipart	none	1	☹	need help to test...	
converters	none	1	☹	need help to test...	
install	start 3/17	0			
compatibility	start 3/17	0		lab time is scheduled	
general GUI	low	3	--		

# Application Demo - Requirements

1. As a system administrator I want to be able to add new user to a system
2. As a system administrator I want my user form to be automatically validated
3. As a system administrator I want my user form check if password is strong enough



# Show me the application



<https://github.com/senpay/feature-test-master-class>



# Wonna-be algorithm

- Understand the context  
*[and challenge context assumptions]*
- Define a goal/mission/risks  
*[does anybody have an idea what the heck you're doing here]*
- Use an appropriate way to support your coverage
  - *Code coverage*
  - *Feature coverage*
  - *Risk coverage*

Final element: the team of individuals who care about the crafts



Individuals and interactions over processes and tools

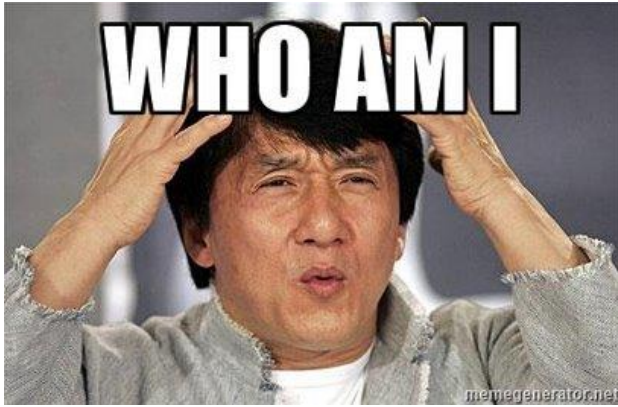
# Essential complexity and Accidental complexity

## Why software testing is hard?

- It is because software is complex.
- The more complex the thing you test - the more complex it is to test it
- Your design (for testability) is influencing your test automation success much higher than frameworks and tools



# Should you have any questions



**Alex Pushkarev**

<http://aqaguy.blogspot.com>

<https://www.linkedin.com/in/alexpushkarev/>

<https://twitter.com/aqaguy>