STAR EAST
A TECHWELL EVENT

T7
Leadership
2019-05-02 11:15

The Reality Distortion Field of Testing

Presented by:

Lloyd Roden

Lloyd Roden Consultancy
‘

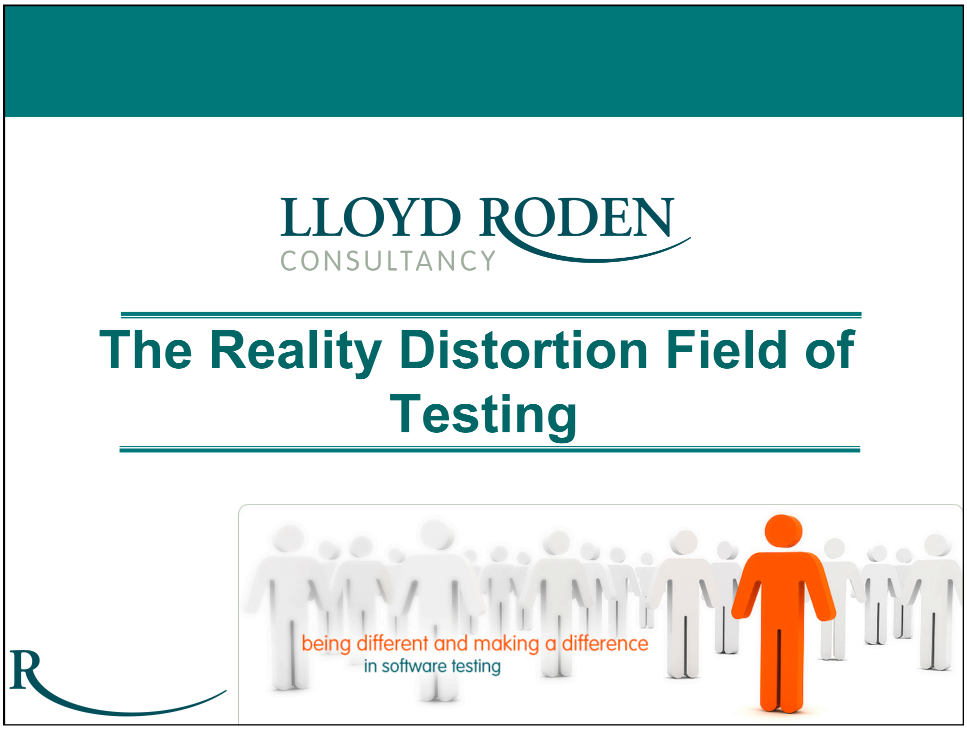
Brought to you by:

TECHWELL™

888-268-8770 ·· 904-278-0524 - info@techwell.com - http://www.stareast.techwell.com/

Lloyd Roden

With more than twenty-eight years in the software industry, Lloyd Roden has worked as a developer, test analyst, and test manager for many different organizations. Lloyd was a consultant/partner with Grove Consultants for twelve years. In 2011 he created Lloyd Roden Consultancy, an independent UK-based training and consultancy company specializing in software testing. Lloyd's passion is to enthuse, excite, and inspire people in the area of software testing. He has spoken at conferences worldwide including STAREAST, STARWEST, Better Software, EuroSTAR, AsiaSTAR, and Special Interest Groups in software testing in several countries. In 2004, he won the European Testing Excellence award.



LLOYD RODEN CONSULTANCY

The Reality Distortion Field of Testing

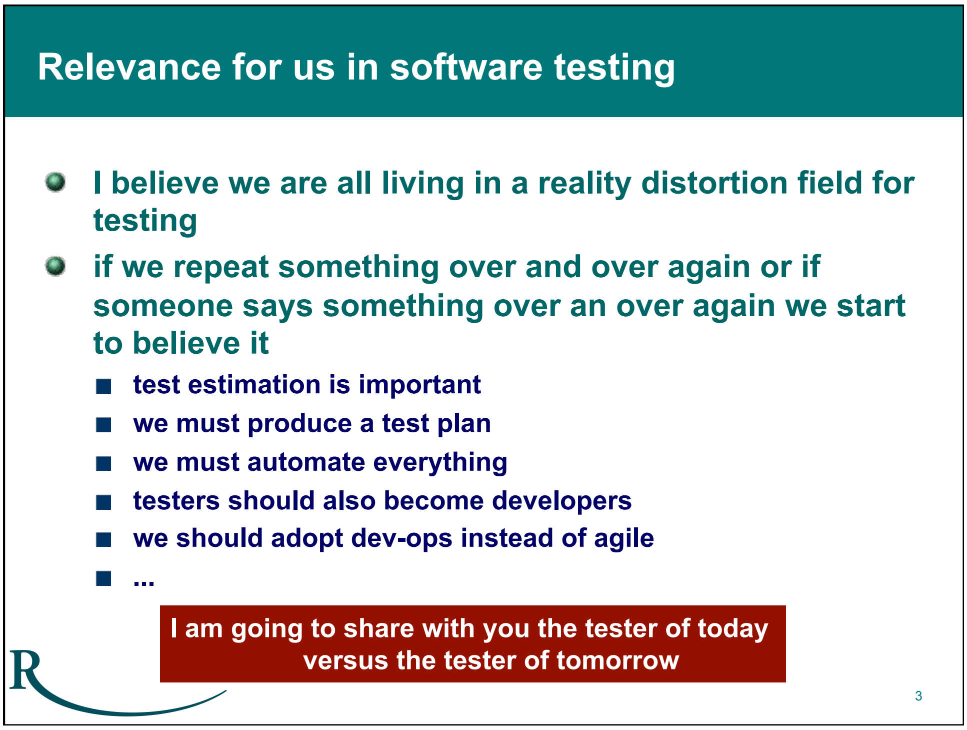
being different and making a difference
in software testing

Contents

Introduction

Today's Super Tester

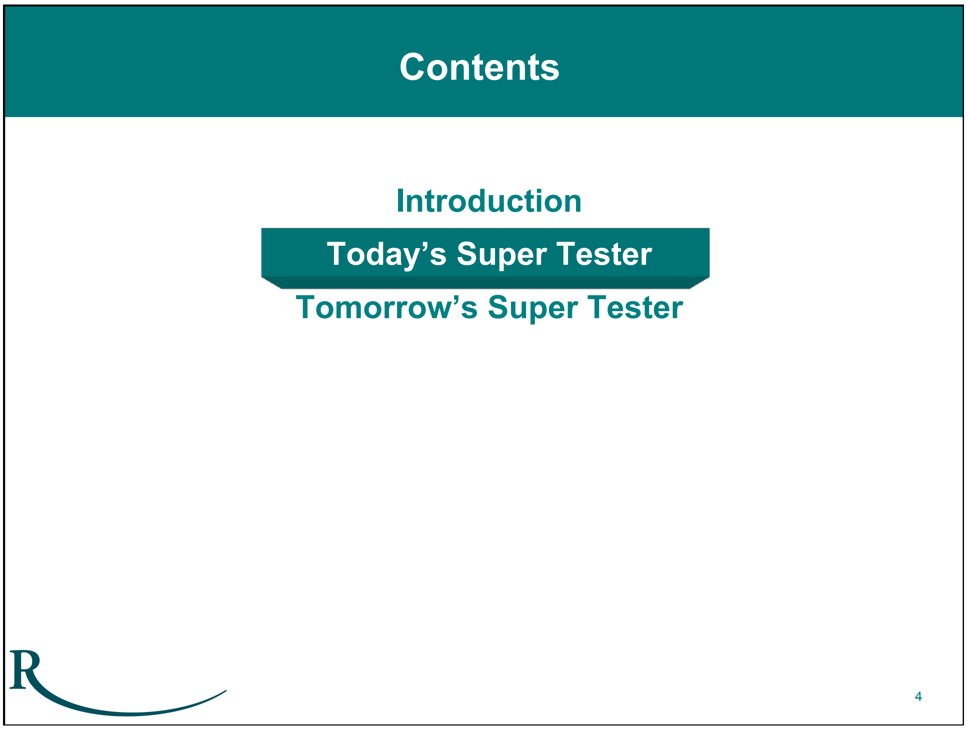
Tomorrow's Super Tester



Relevance for us in software testing

- I believe we are all living in a reality distortion field for testing
- if we repeat something over and over again or if someone says something over an over again we start to believe it
  - test estimation is important
  - we must produce a test plan
  - we must automate everything
  - testers should also become developers
  - we should adopt dev-ops instead of agile
  - ...

I am going to share with you the tester of today versus the tester of tomorrow

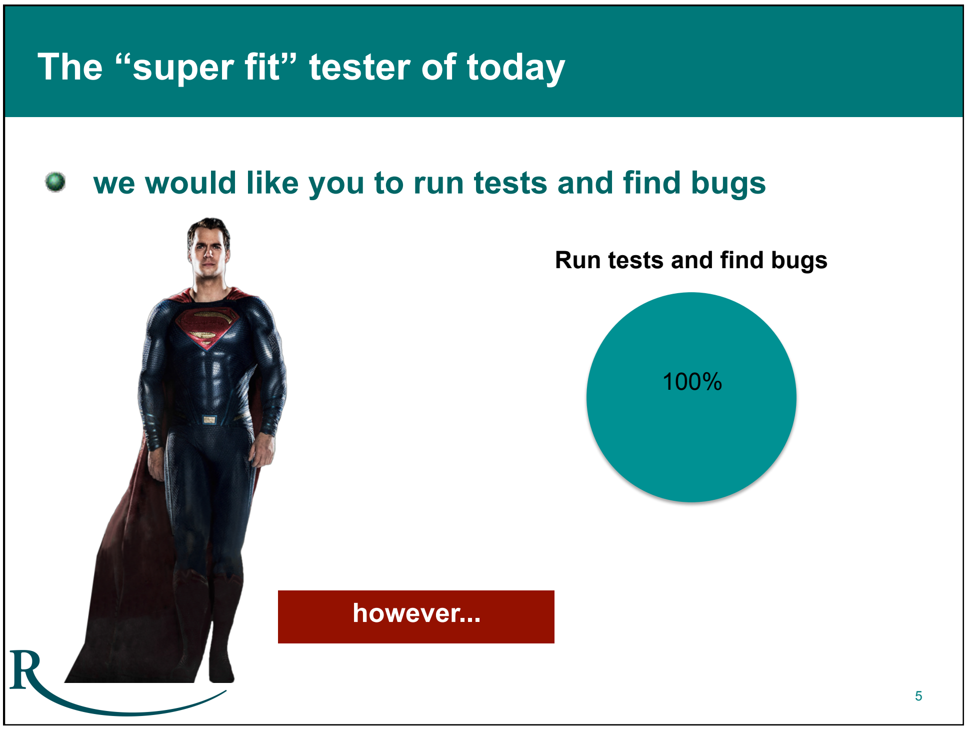


Contents

Introduction

Today's Super Tester

Tomorrow's Super Tester

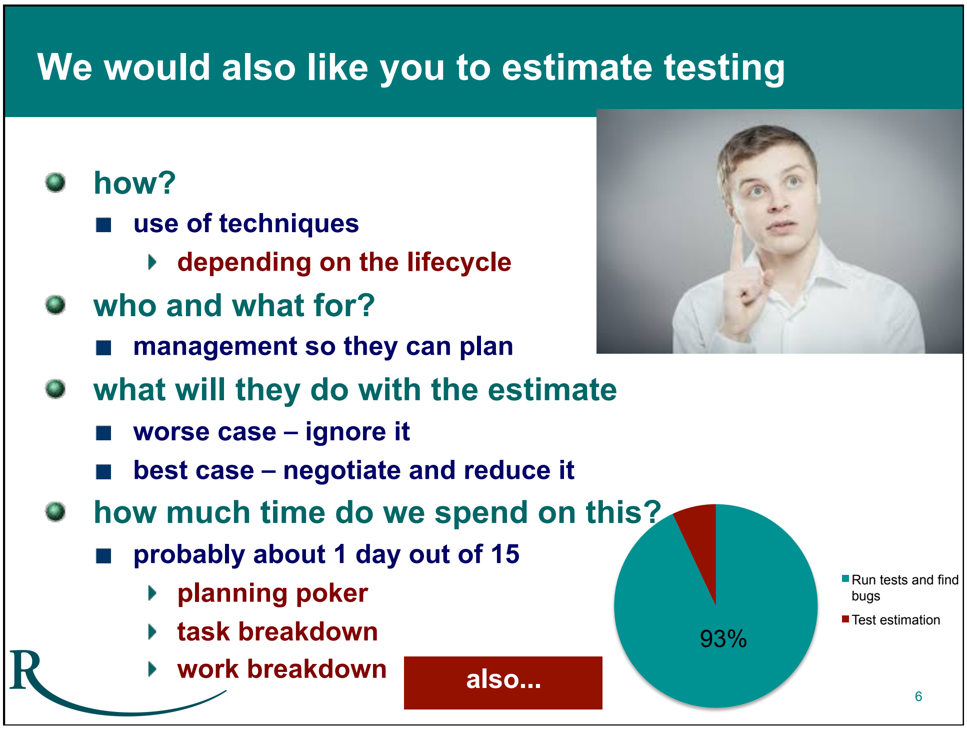


The “super fit” tester of today

- we would like you to run tests and find bugs

Run tests and find bugs
100%

however...

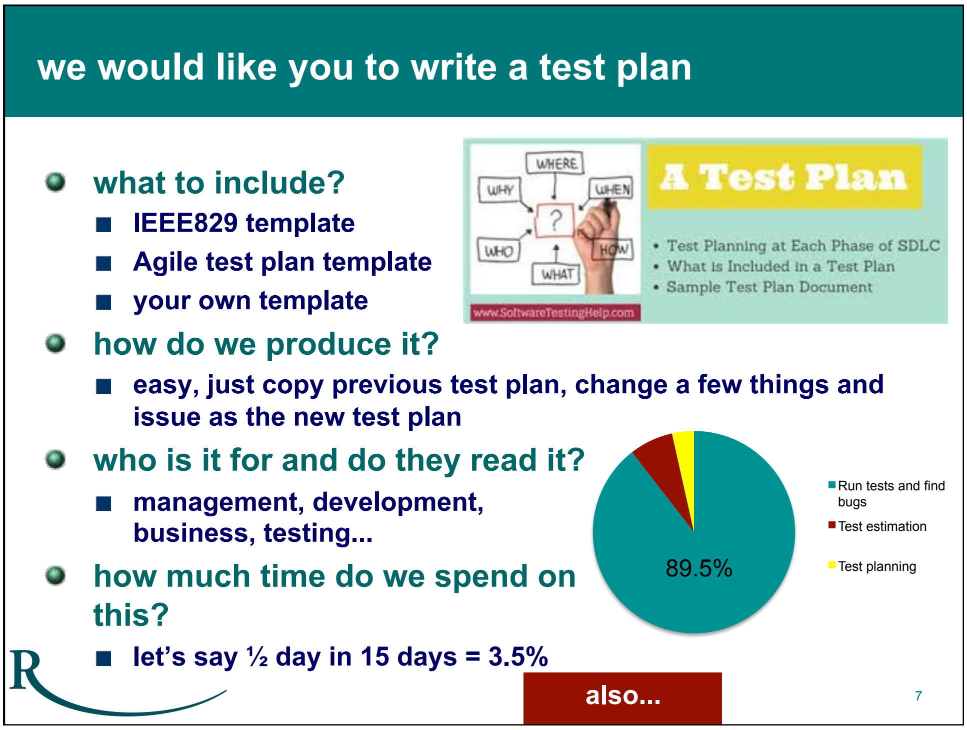


We would also like you to estimate testing

- how?
  - use of techniques
    - depending on the lifecycle
- who and what for?
  - management so they can plan
- what will they do with the estimate
  - worse case – ignore it
  - best case – negotiate and reduce it
- how much time do we spend on this?
  - probably about 1 day out of 15
    - planning poker
    - task breakdown
    - work breakdown

also...

93%
🟩 Run tests and find bugs
🟥 Test estimation



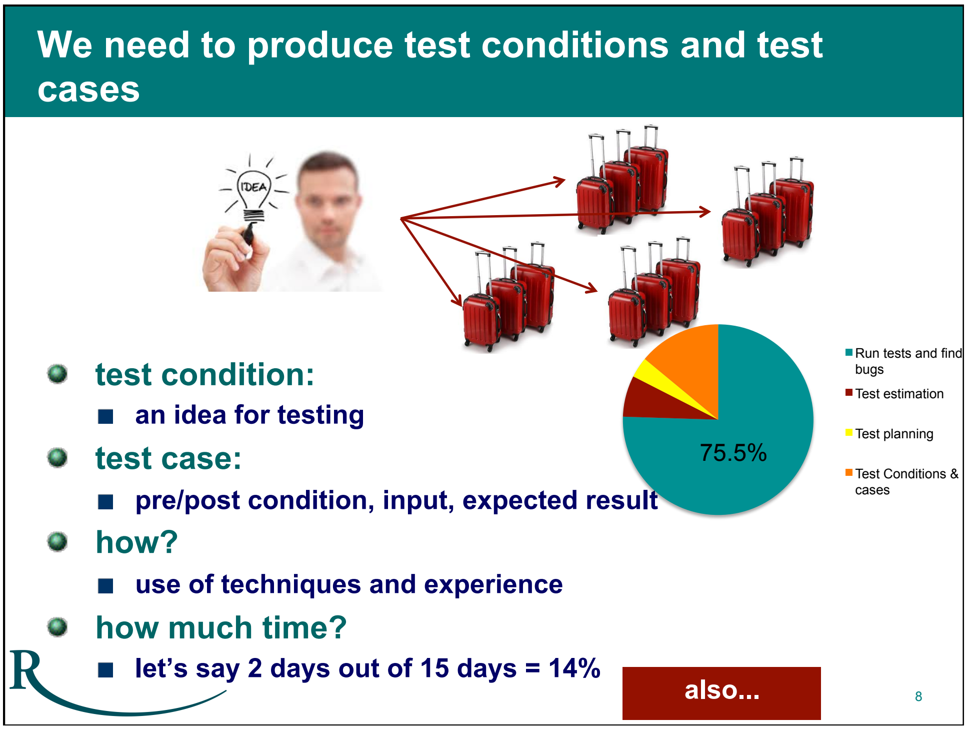
we would like you to write a test plan

- what to include?
  - IEEE829 template
  - Agile test plan template
  - your own template
- how do we produce it?
  - easy, just copy previous test plan, change a few things and issue as the new test plan
- who is it for and do they read it?
  - management, development, business, testing...
- how much time do we spend on this?
  - let's say ½ day in 15 days = 3.5%

WHERE
WHY
WHEN
?
WHO
HOW
WHAT
www.SoftwareTestingHelp.com
A Test Plan
• Test Planning at Each Phase of SDLC
• What is Included in a Test Plan
• Sample Test Plan Document
89.5%
Run tests and find bugs
Test estimation
Test planning

also...

7



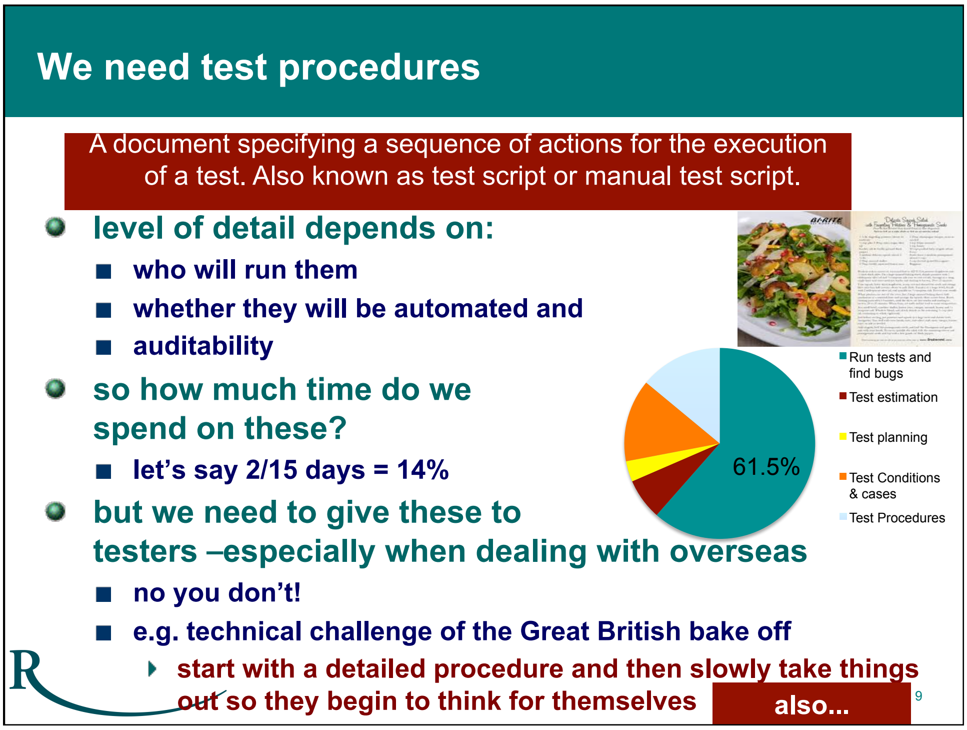
We need to produce test conditions and test cases

- test condition:
  - an idea for testing
- test case:
  - pre/post condition, input, expected result
- how?
  - use of techniques and experience
- how much time?
  - let's say 2 days out of 15 days = 14%

IDEA
75.5%
Run tests and find bugs
Test estimation
Test planning
Test Conditions & cases

also...

8

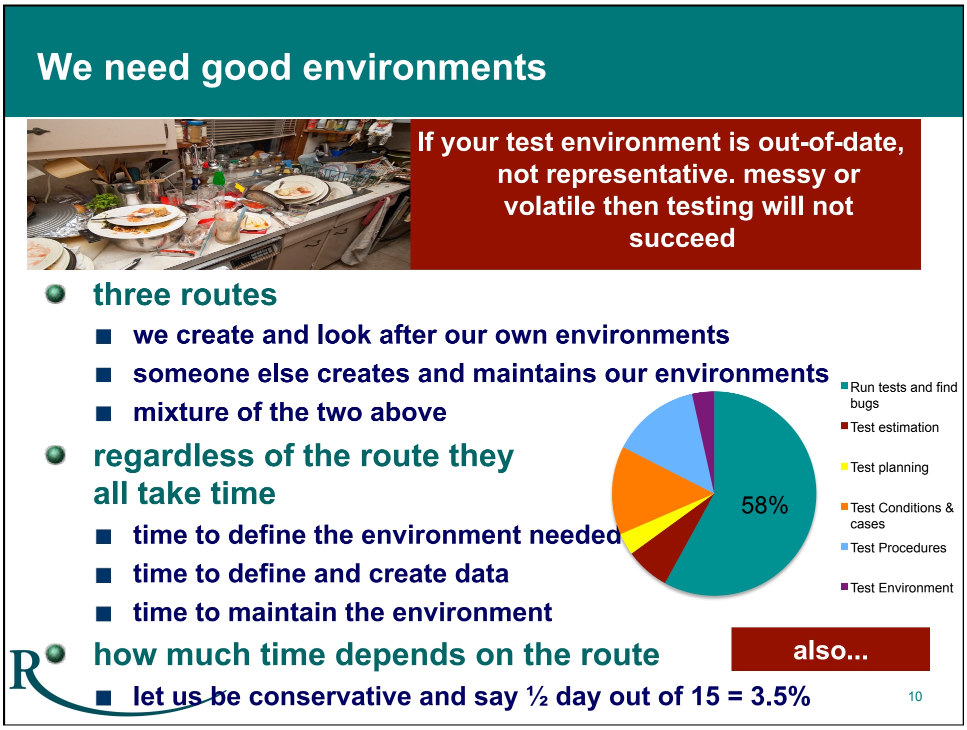


We need test procedures

A document specifying a sequence of actions for the execution of a test. Also known as test script or manual test script.

- level of detail depends on:
  - who will run them
  - whether they will be automated and
  - auditability
- so how much time do we spend on these?
  - let's say 2/15 days = 14%
- but we need to give these to testers –especially when dealing with overseas
  - no you don't!
  - e.g. technical challenge of the Great British bake off
    - start with a detailed procedure and then slowly take things out so they begin to think for themselves

also...

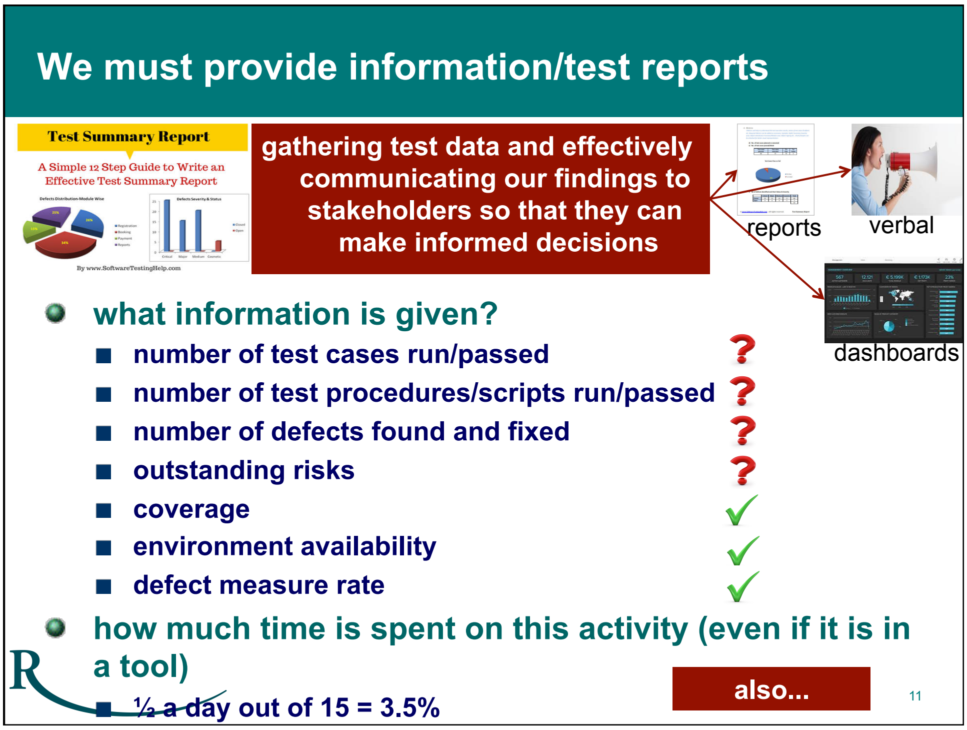


We need good environments

If your test environment is out-of-date, not representative. messy or volatile then testing will not succeed

- three routes
  - we create and look after our own environments
  - someone else creates and maintains our environments
  - mixture of the two above
- regardless of the route they all take time
  - time to define the environment needed
  - time to define and create data
  - time to maintain the environment
- how much time depends on the route
  - let us be conservative and say ½ day out of 15 = 3.5%

also...



We must provide information/test reports

Test Summary Report
A Simple 12 Step Guide to Write an Effective Test Summary Report
By www.SoftwareTestingHelp.com

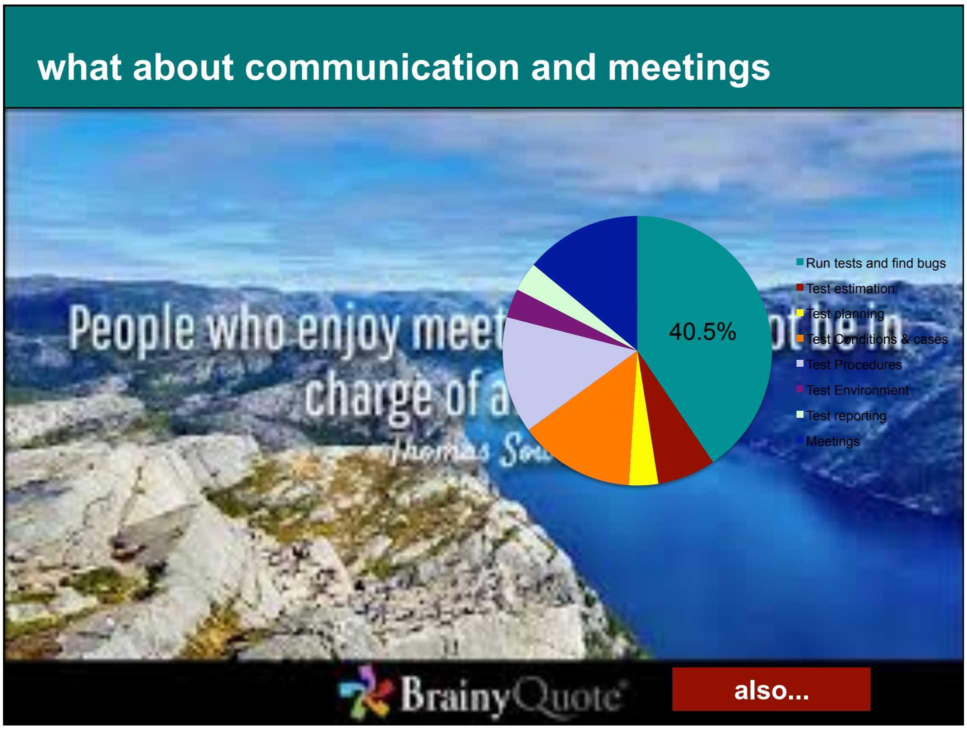
gathering test data and effectively communicating our findings to stakeholders so that they can make informed decisions

reports
verbal
dashboards

- what information is given?
  - number of test cases run/passed ?
  - number of test procedures/scripts run/passed ?
  - number of defects found and fixed ?
  - outstanding risks ?
  - coverage ✓
  - environment availability ✓
  - defect measure rate ✓
- how much time is spent on this activity (even if it is in a tool)
  - ½ a day out of 15 = 3.5%

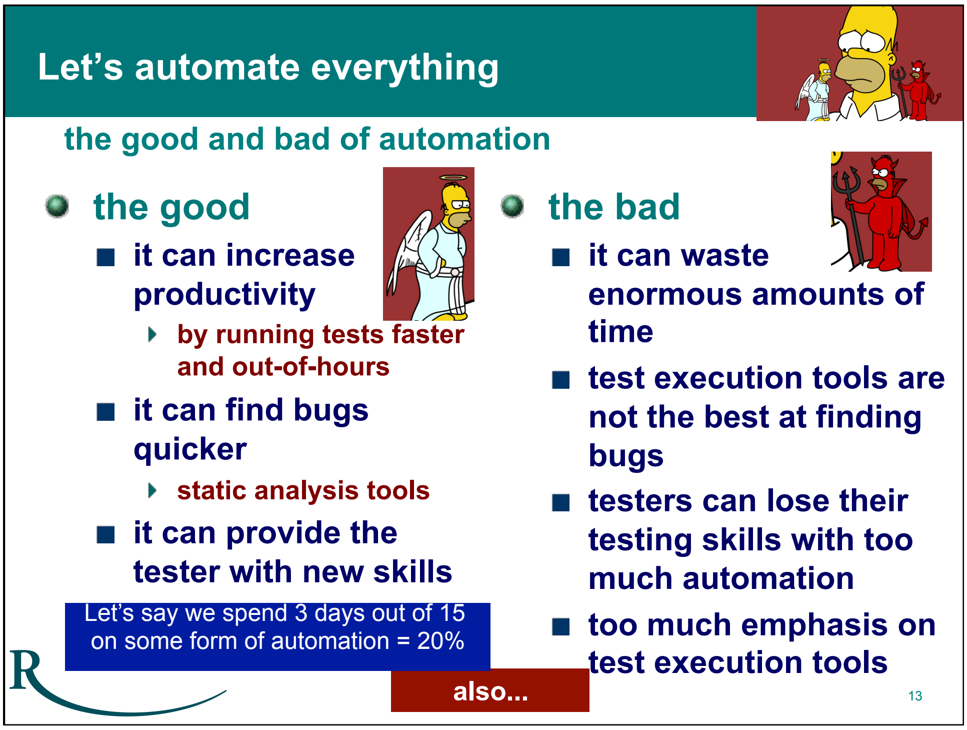
also...

11



what about communication and meetings

also...



Let's automate everything

the good and bad of automation

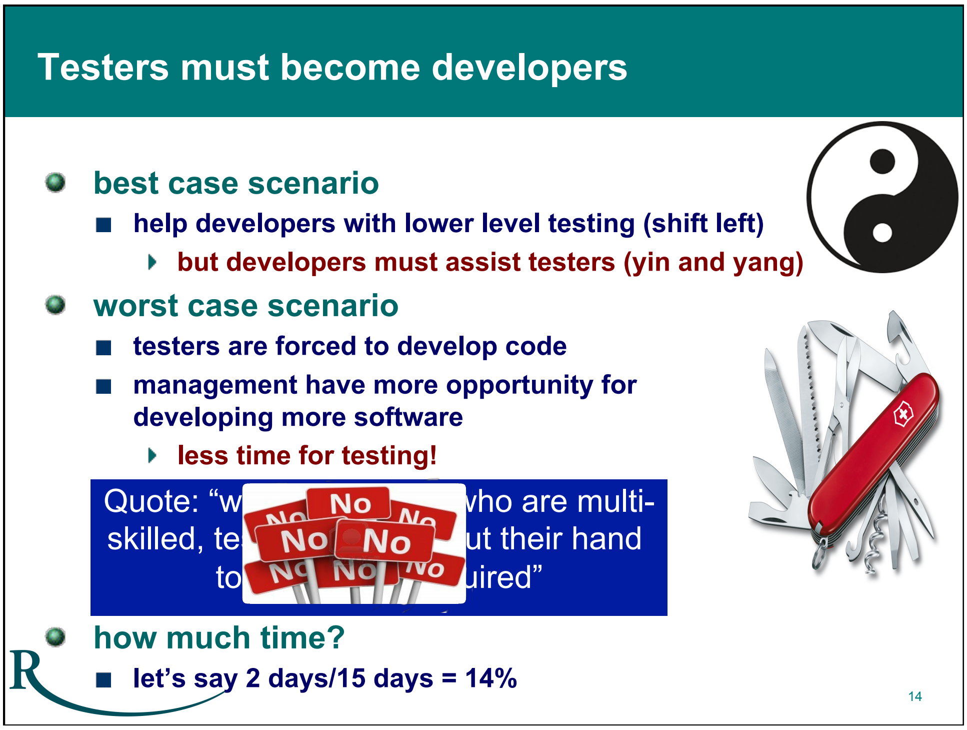
- the good
  - it can increase productivity
    - by running tests faster and out-of-hours
  - it can find bugs quicker
    - static analysis tools
  - it can provide the tester with new skills

Let's say we spend 3 days out of 15 on some form of automation = 20%

- the bad
  - it can waste enormous amounts of time
  - test execution tools are not the best at finding bugs
  - testers can lose their testing skills with too much automation
  - too much emphasis on test execution tools

also...

13



Testers must become developers

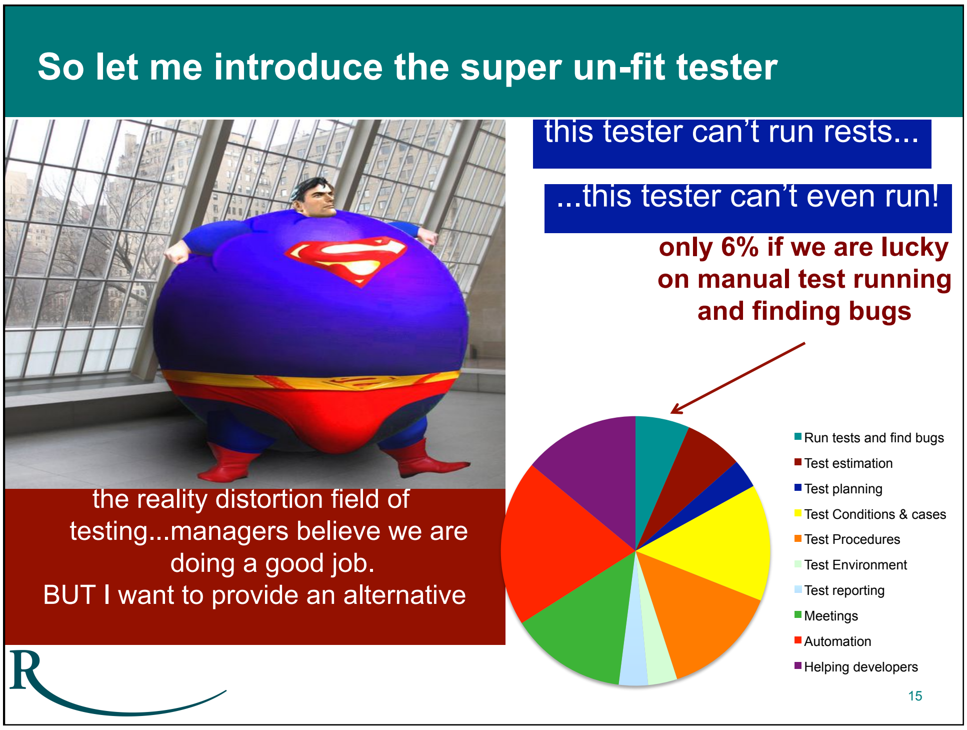
- best case scenario
  - help developers with lower level testing (shift left)
    - but developers must assist testers (yin and yang)
- worst case scenario
  - testers are forced to develop code
  - management have more opportunity for developing more software
    - less time for testing!

Quote: "w who are multi-skilled, te ut their hand to uired"

No
No
No
No
No
No
No
No

- how much time?
  - let's say 2 days/15 days = 14%

14



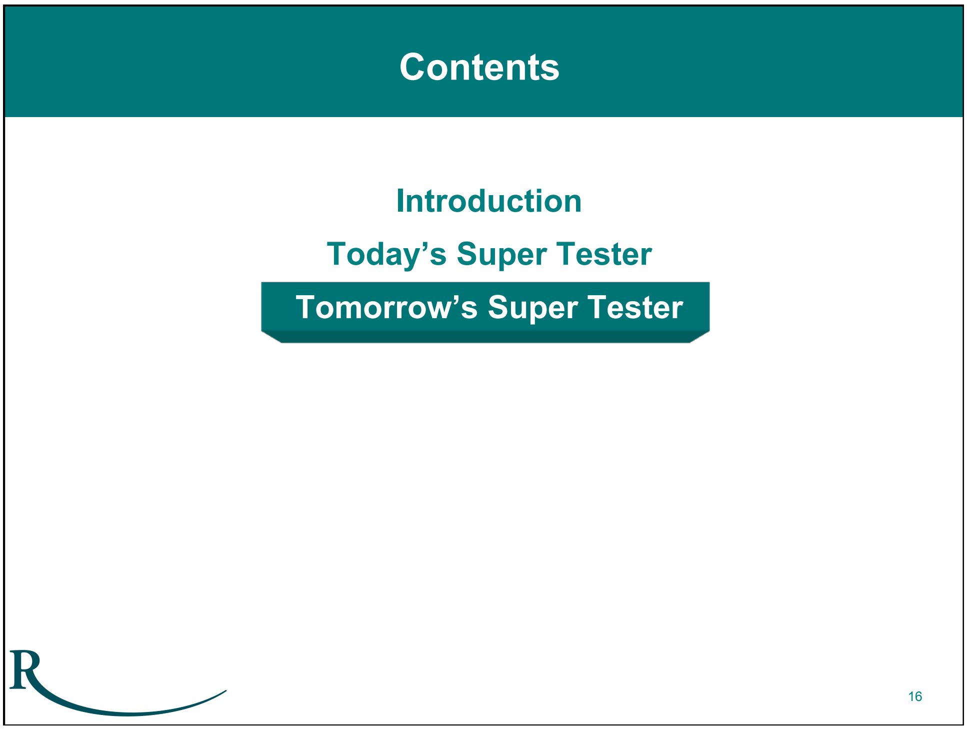
So let me introduce the super un-fit tester

this tester can't run rests...

...this tester can't even run!

only 6% if we are lucky on manual test running and finding bugs

the reality distortion field of testing...managers believe we are doing a good job.
BUT I want to provide an alternative

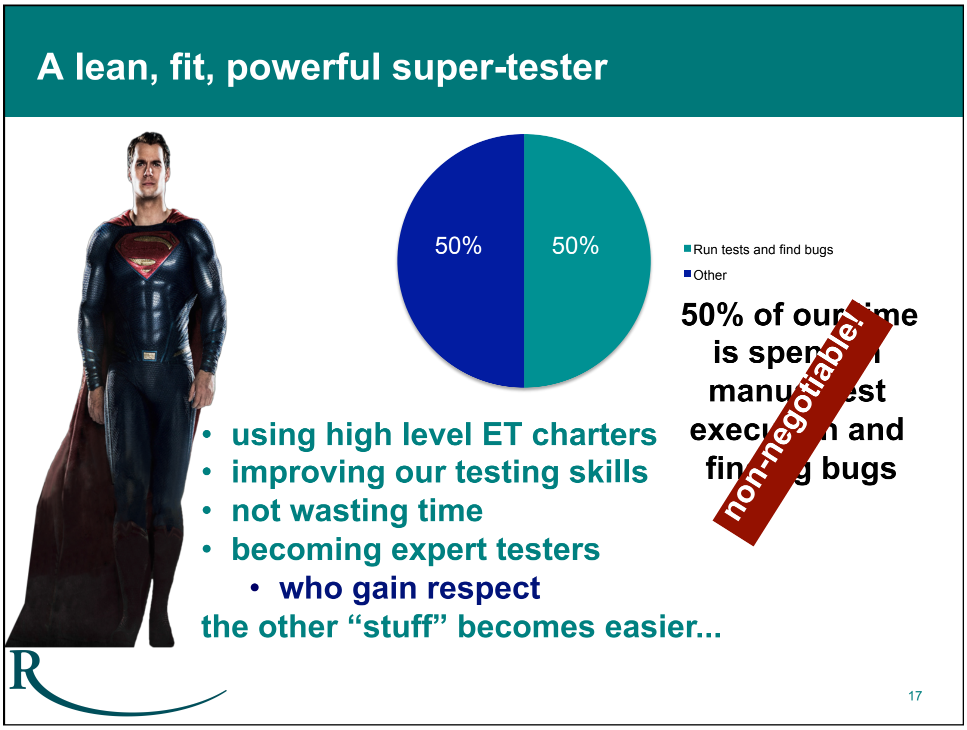


Contents

Introduction

Today's Super Tester

Tomorrow's Super Tester



A lean, fit, powerful super-tester

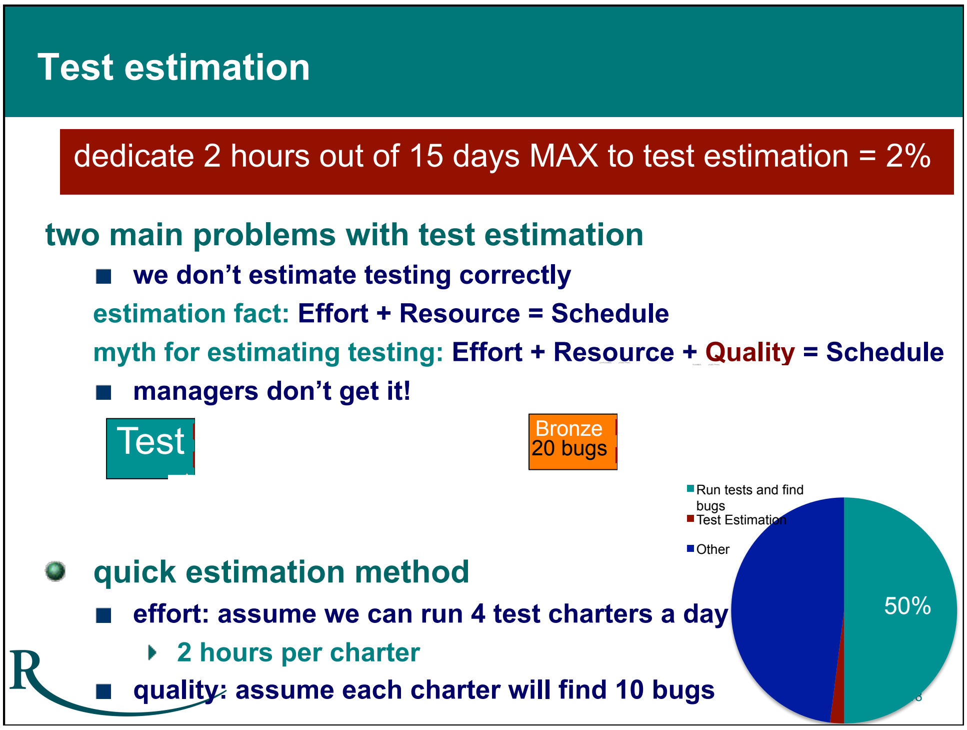
50% of our time is spent in manual test execution and finding bugs

non-negotiable!

- using high level ET charters
- improving our testing skills
- not wasting time
- becoming expert testers
  - who gain respect

the other "stuff" becomes easier...

17



Test estimation

dedicate 2 hours out of 15 days MAX to test estimation = 2%

two main problems with test estimation

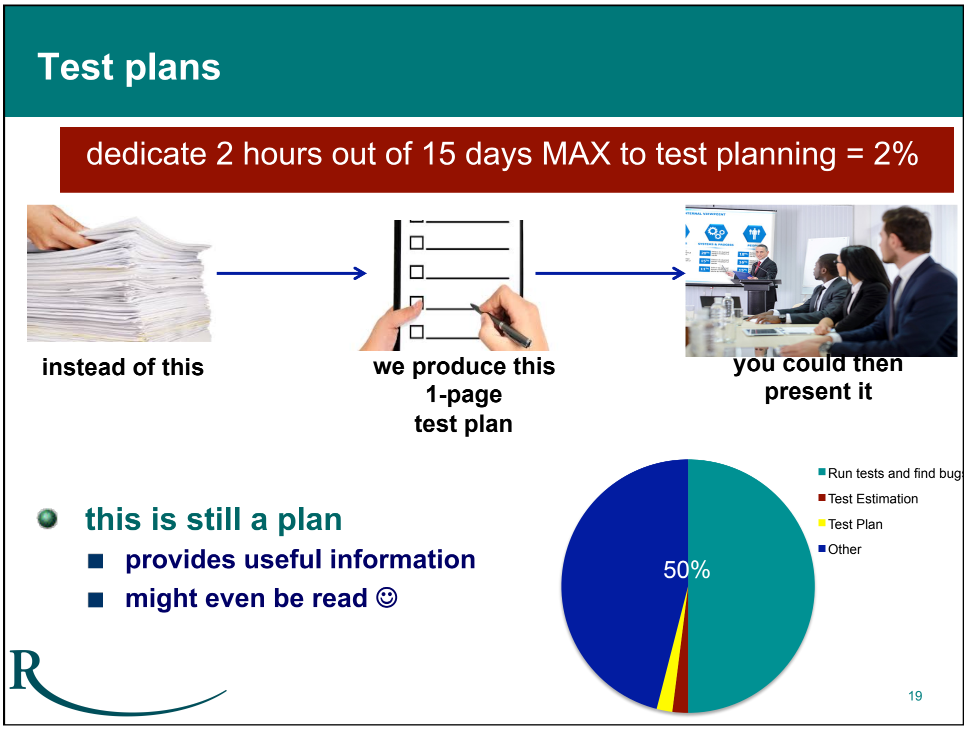
- we don't estimate testing correctly

estimation fact: Effort + Resource = Schedule

myth for estimating testing: Effort + Resource + Quality = Schedule

- managers don't get it!

- quick estimation method
  - effort: assume we can run 4 test charters a day
    - 2 hours per charter
  - quality: assume each charter will find 10 bugs



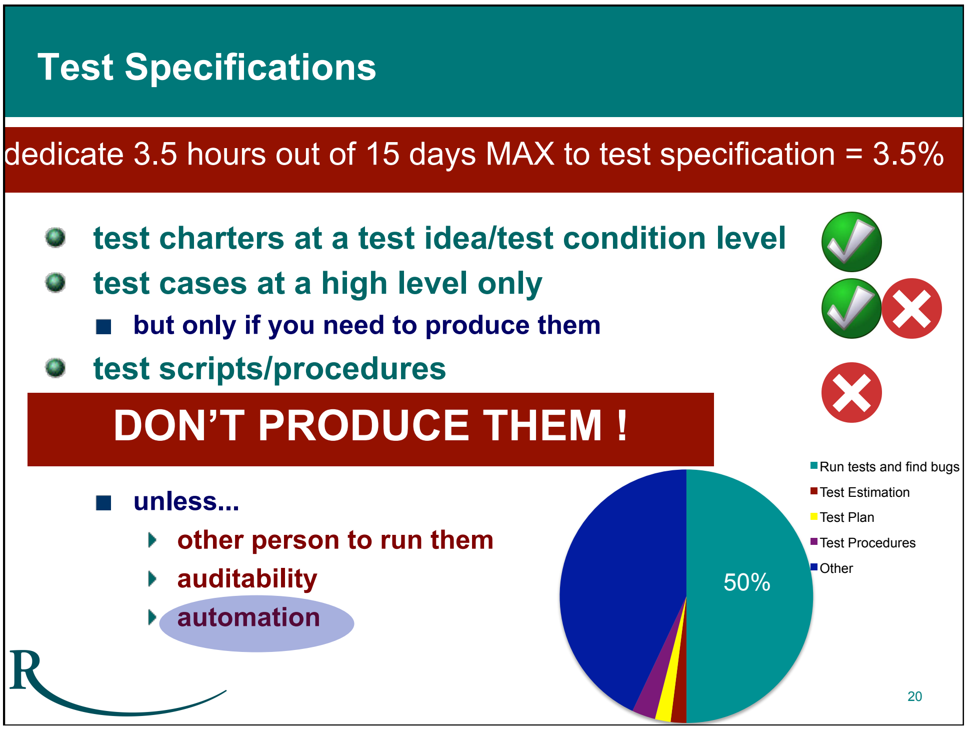
Test plans

dedicate 2 hours out of 15 days MAX to test planning = 2%

instead of this → we produce this 1-page test plan → you could then present it

- this is still a plan
  - provides useful information
  - might even be read ☺

50%
■ Run tests and find bug
■ Test Estimation
■ Test Plan
■ Other
19



Test Specifications

dedicate 3.5 hours out of 15 days MAX to test specification = 3.5%

- test charters at a test idea/test condition level ✓
- test cases at a high level only ✓ ✗
  - but only if you need to produce them
- test scripts/procedures ✗

DON'T PRODUCE THEM !

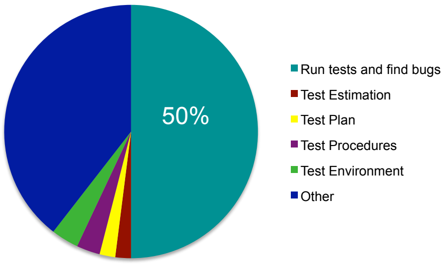
- unless...
  - other person to run them
  - auditability
  - automation

50%
■ Run tests and find bugs
■ Test Estimation
■ Test Plan
■ Test Procedures
■ Other
20

Test environment

dedicate 3.5 hours out of 15 days to test environment = 3.5%

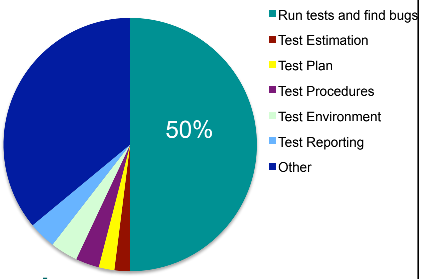
- this will be the same as before, but keep a note on how much time you are spending on setting up and maintaining your environment



Test reporting made easier

dedicate 3.5 hours out of 15 days to test reporting = 3.5%

- tasks = charters
  - estimate the number of bugs you will find per charter & report
  - 15 minute de-brief at the end of each day (verbal)
  - provide P.R.O.O.F
    - Product Assessment
    - Results
    - Observations
    - Obstacles
    - Future Plans
- move away from number of test cases and number of procedures
  - and all the irrelevant data that managers request –ask why?

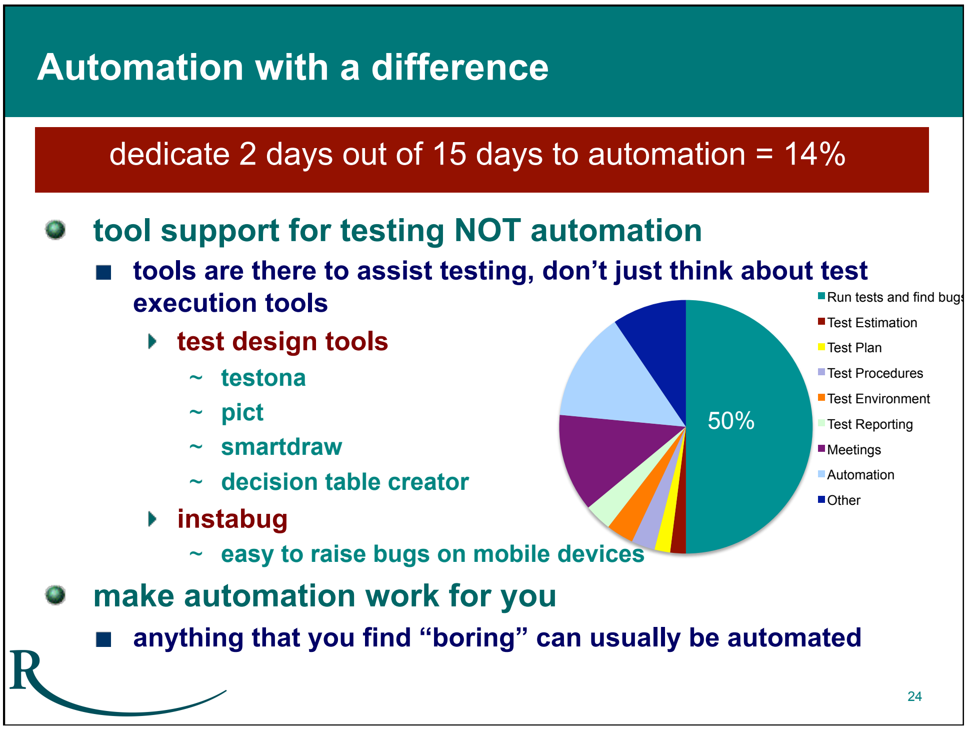


Meetings and communication

dedicate 1 hour per day = 15 hours to meetings = 14%

- important but be in control:
  - during test execution
    - dedicated time
    - no interruptions
    - no emails
    - no link calls
    - no texts
- explain to people...they will eventually understand
  - have some sign or signal
  - not responding to other people's interruptions
    - only talking about 2 hours max for each charter
- bring in meeting etiquette

23



Automation with a difference

dedicate 2 days out of 15 days to automation = 14%

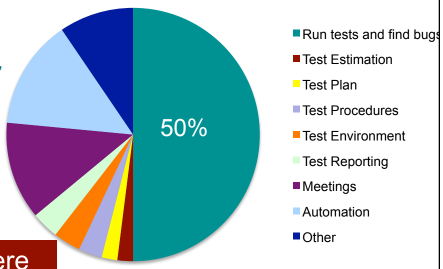
- tool support for testing NOT automation
  - tools are there to assist testing, don't just think about test execution tools
    - test design tools
      - ~ testona
      - ~ pict
      - ~ smartdraw
      - ~ decision table creator
    - instabug
      - ~ easy to raise bugs on mobile devices
- make automation work for you
  - anything that you find "boring" can usually be automated

24

Assisting developers

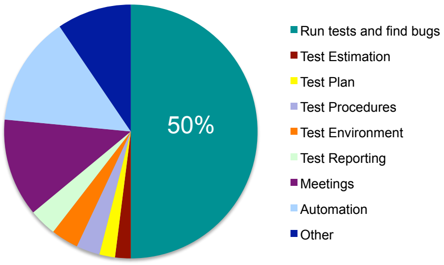
we now have 7.5% left for assisting developers (~1day) ☺

- yes but we need yin and yang
  - we need to have balance
- if developers help testers
  - then the number goes up ☺
- resist becoming a developer
  - become an experienced and expert tester



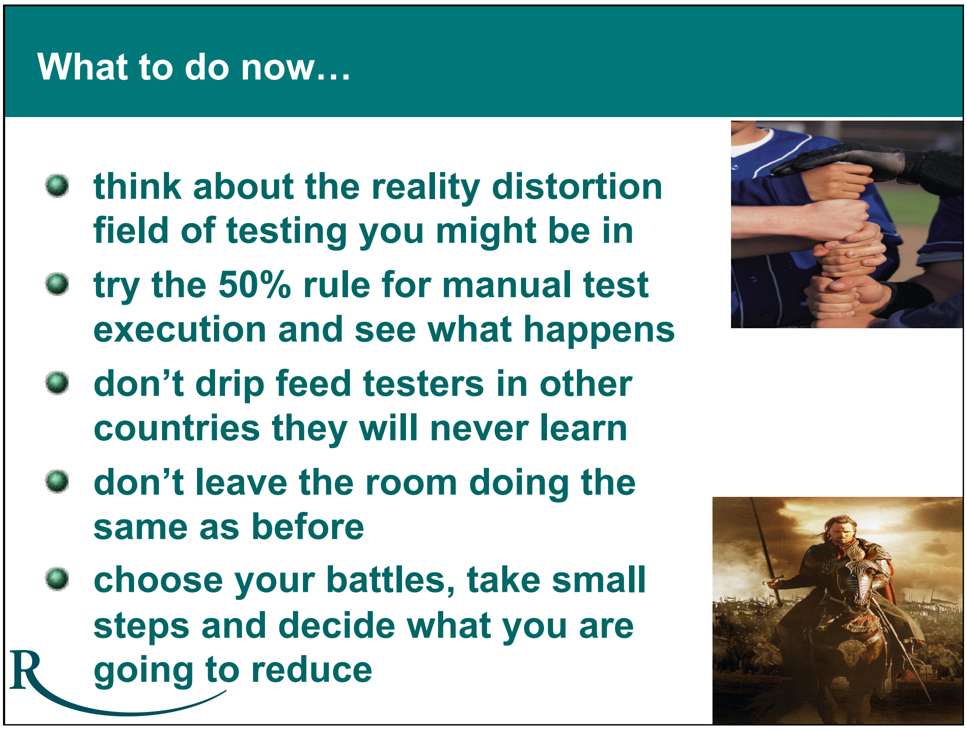
50% manual exploratory testing where we run tests and find bugs is non-negotiable

The super-fit tester of tomorrow



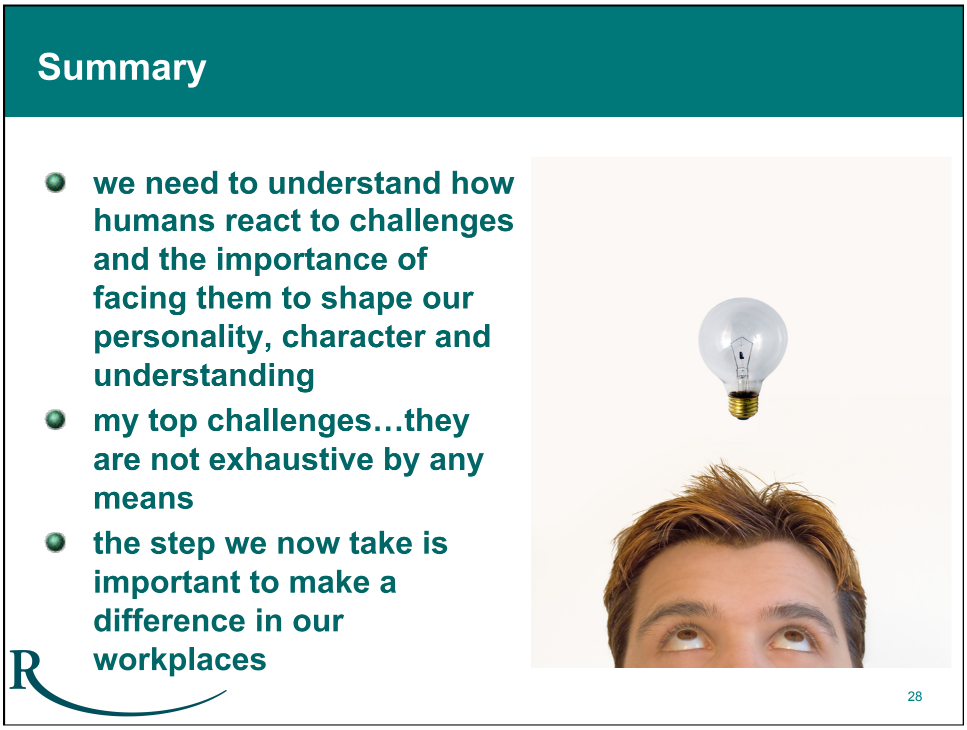
50% Manual Exploratory Testing

- finding bugs
- gaining confidence
- save money
- happier customers
- happier testers



What to do now…

- think about the reality distortion field of testing you might be in
- try the 50% rule for manual test execution and see what happens
- don't drip feed testers in other countries they will never learn
- don't leave the room doing the same as before
- choose your battles, take small steps and decide what you are going to reduce



Summary

- we need to understand how humans react to challenges and the importance of facing them to shape our personality, character and understanding
- my top challenges…they are not exhaustive by any means
- the step we now take is important to make a difference in our workplaces