



Test Strategies Analytical Testing and More

*Surviving the Testing Challenges
of Today and Tomorrow*



Erik van Veenendaal
www.erikvanveenendaal.nl

Erik van Veenendaal

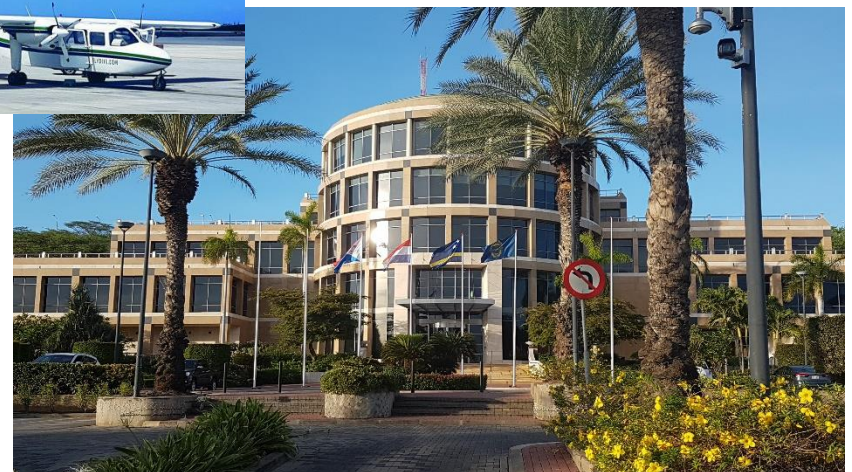
www.erikvanveenendaal.nl



- In IT & testing since 1985 working for many different clients and in many different roles
- Author “TMap”, “ISTQB Foundation” and “TMMi” and many other books and papers
- Former Vice-President International Software Testing Qualifications Board (ISTQB)
- Supporting member IREB board
- CEO TMMi Foundation
- Winner European & ISTQB International Testing Excellence Award



The other side of me ...





No More Testers!

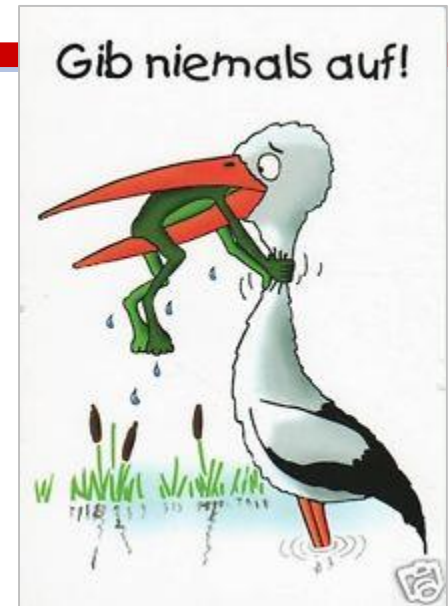
No More Testers!

No More Testers!

No More Testers!

Common Testing Challenges

- Increasing business importance
- Increasing **code size** and **complexity**
- Technology advancement
- Systems-of-systems
- **Variety of devices and OS's**
- Security vulnerabilities
- Outsourcing
- Time-to-market critical
- Number of defects hardly decreases





Duizenden gevangenen te vroeg vrij door softwarefout

Laatste update: 23 december 2015 16:53

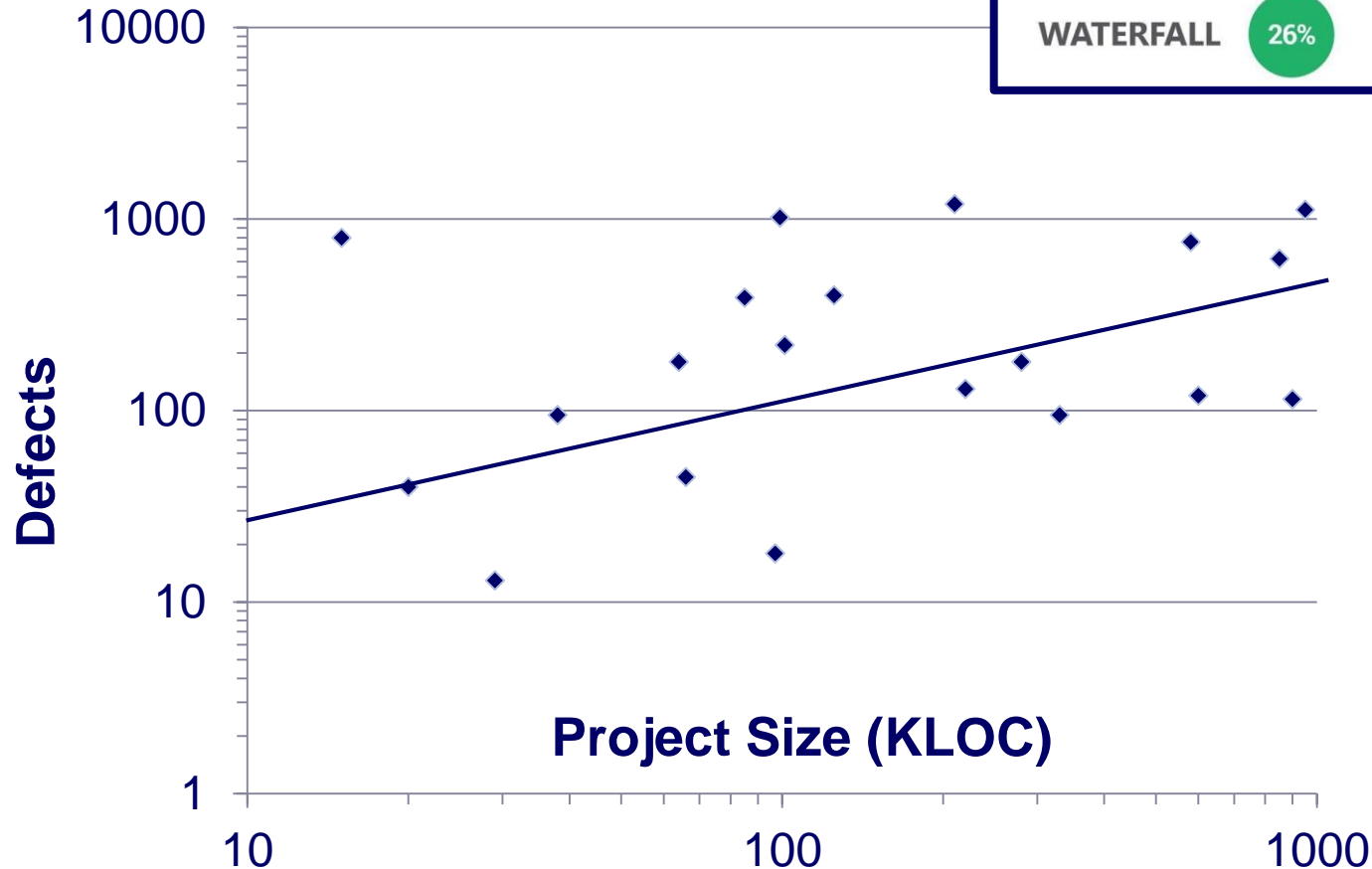
De software waarmee klanten hun hypotheek leningen afbetalen, bevatte echter een fout. Het persbureau [Reuters](#) hierdoor zijn duizenden gevangenen niet in staat om hun hypotheek af te betalen.



Agile ≠ Silver Bullit

PROJECT SUCCESS RATES AGILE VS WATERFALL

METHOD	SUCCESSFUL	CHALLENGED	FAILED
AGILE	42%	50%	8%
WATERFALL	26%	53%	21%



“despite continuous growth in adoption of agile, many organizations are still facing the same challenges as in previous years”, World Quality Report

Top 5 Agile Benefits

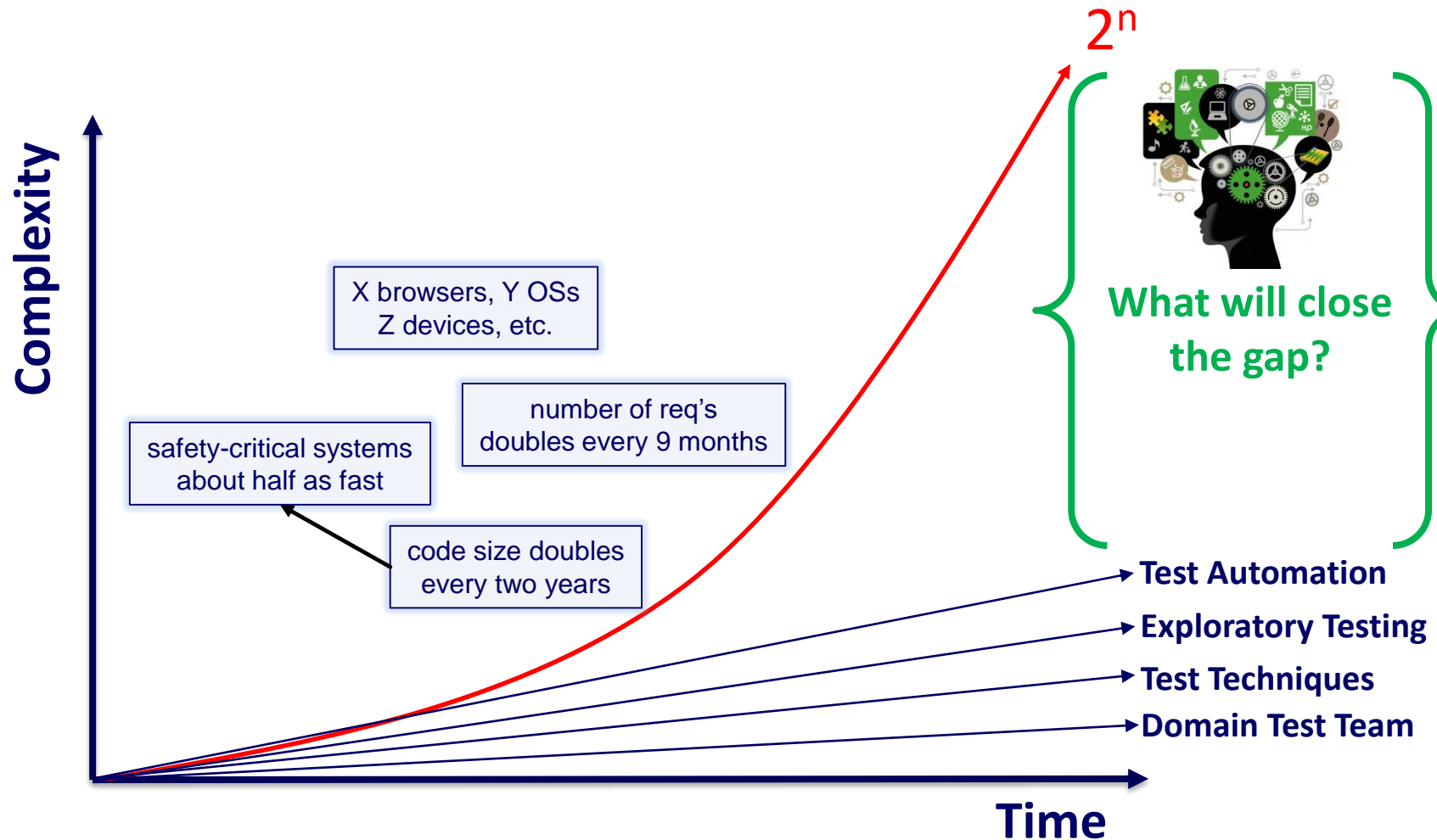
1. Ability to manage changing priorities
2. Improved project visibility
3. Increased team motivation
4. Delivery speed/time to market
5. Better delivery predictability



Source: *13th Annual State of Agile™ Report*



Testing Challenge



No More Testers, or

There has never been a time when software testers were needed more!

Test Strategies “covered”

- **Methodical**
 - Test Techniques, ISTQB, TMap, Checklists, Taxonomies
- **Directive**
 - Domain experts, Technology experts
- **Regression-averse**
 - Standard test suite, Test automation
- **Reactive**
 - Exploratory Testing (≠ Error Guessing)



Methodical Testing

Black box test techniques

5 7 8 9 10
Valid

Equivalence Partitioning

State Transition Testing

Decision Table Testing

Pairwise Testing



Defect Taxonomies

Data logic

Computation

Error Handling

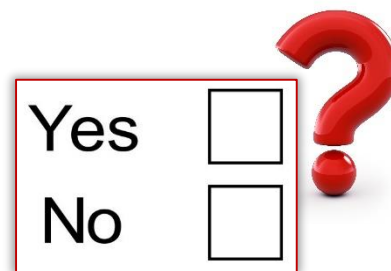
Data definition

Interfacing



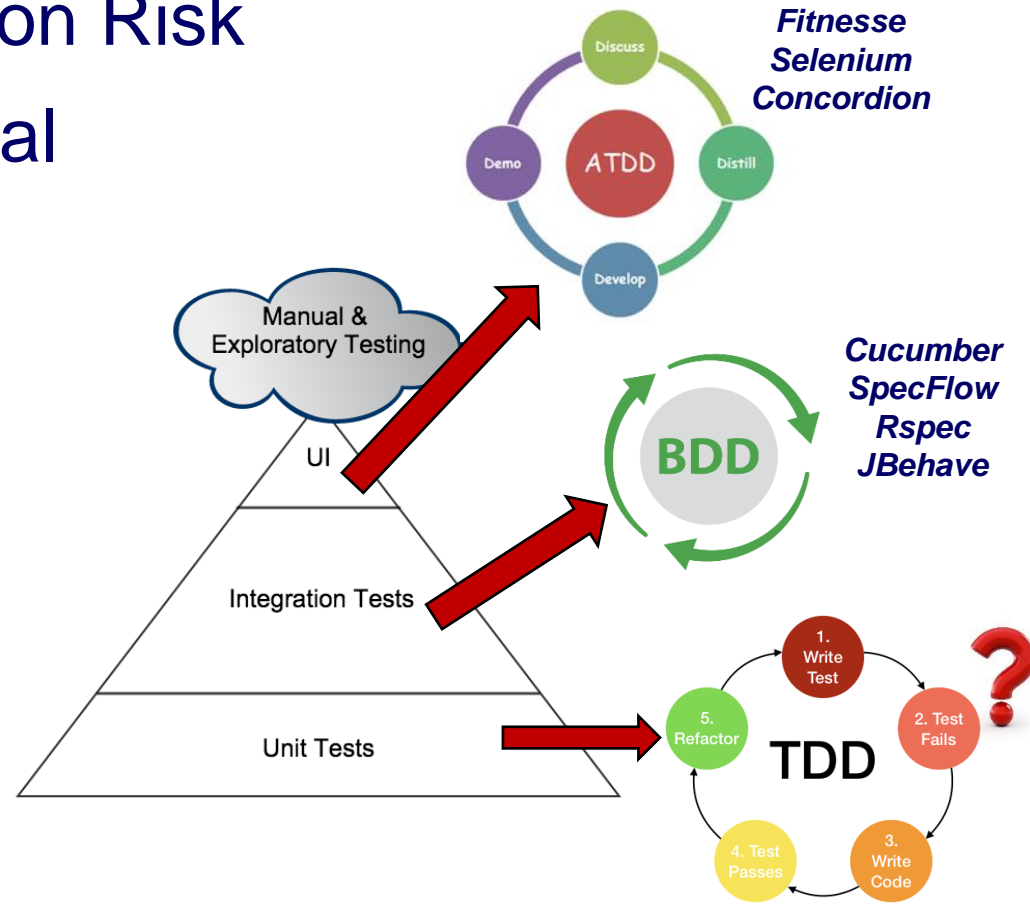
FMEA

Compound	Tox Score	Cpu Score	Risk Analysis	Risk Evaluation
Drug 6	1	10	Very Low Hazard Very Poor Cleaning	Very High Risk
Drug 1	2	8	Very Low Hazard Very Poor Cleaning	Very High Risk
Drug 9	6	8	Moderate Hazard Poor Cleaning	High Risk
Drug 10	5	7	Low Hazard Fair Cleaning	Moderate Risk
Drug 7	4	6	Low Hazard Good Cleaning	Low Risk
Drug 4	5	5	Moderate Hazard Good Cleaning	Low Risk
Drug 5	10	4	Very High Hazard Excellent Cleaning	Low Risk
Drug 3	7	3	High Hazard Excellent Cleaning	Low Risk
Drug 2	8	2	Very High Hazard Excellent Cleaning	Low Risk
Drug 8	8	6	High Hazard Exceptional Cleaning	Very Low Risk



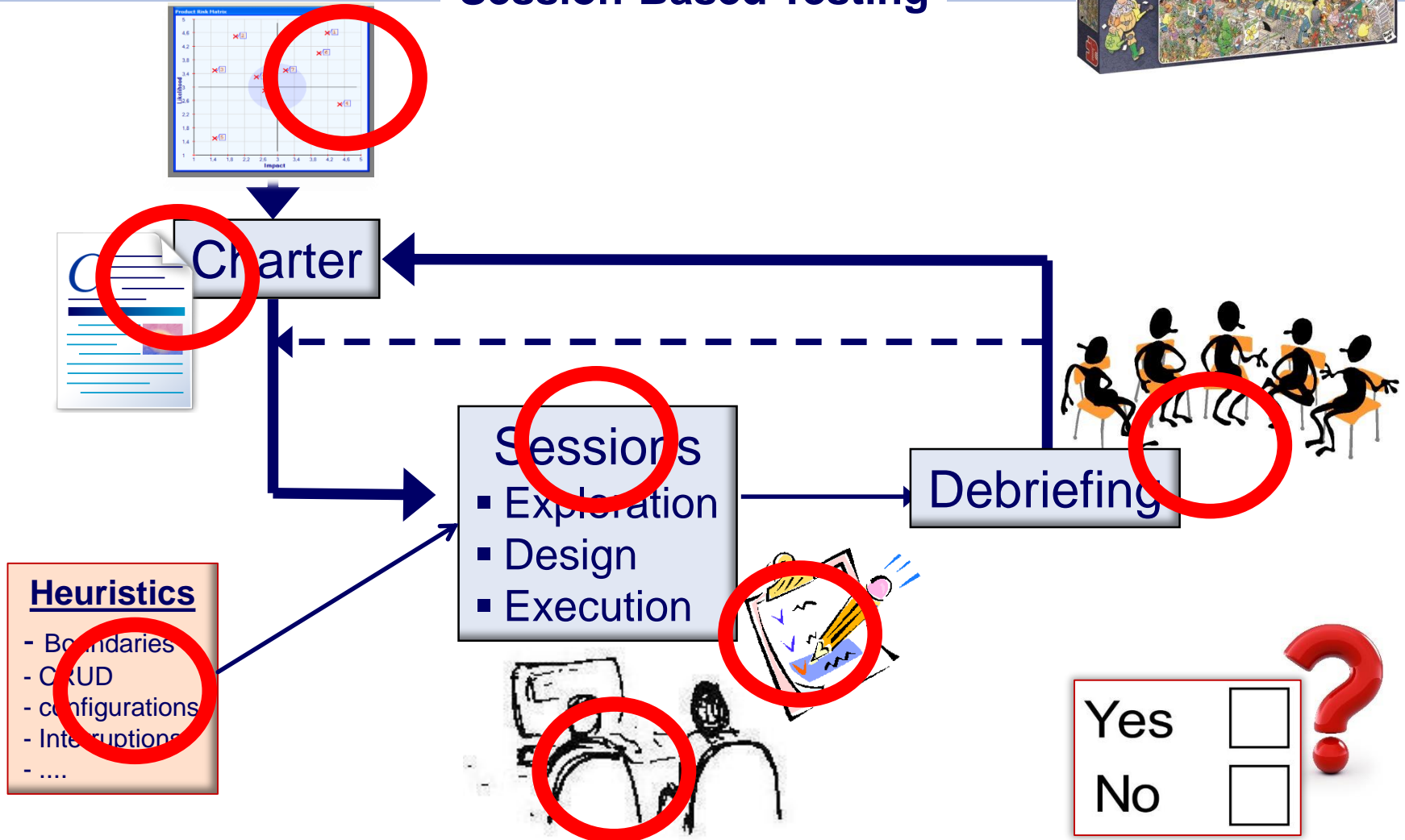
Regression-Adverse

- Manage the Regression Risk
- Structural or Functional
- Coverage
- Maintenance
- Knowledge and Skills
- Tools do not Test
- ROI



Reactive Testing

Session-Based Testing



Test Strategies “uncovered”

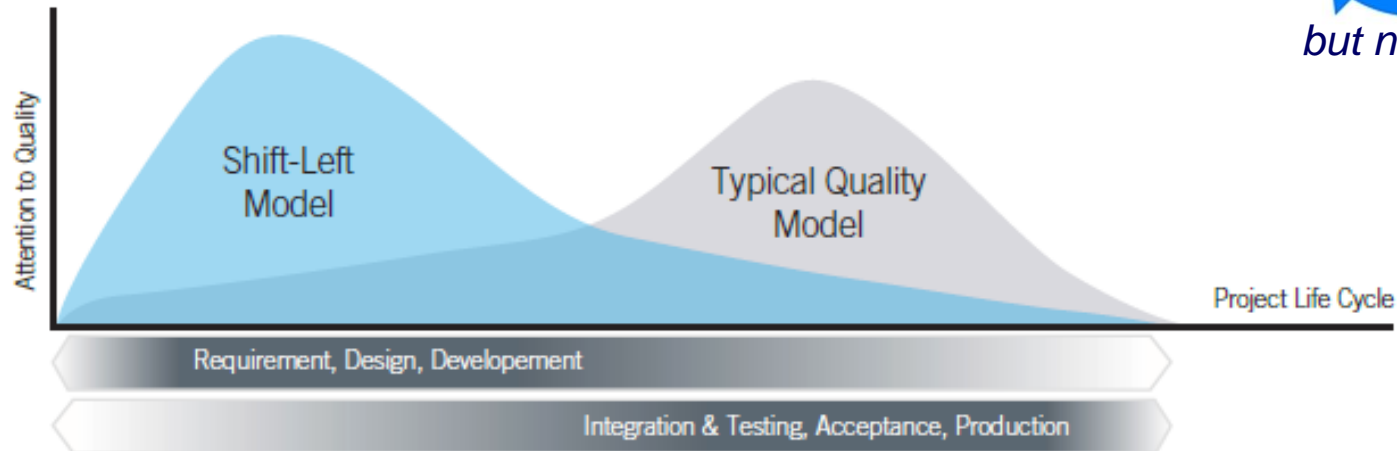
- Shift-Left
 - Test early and often
- Process-Compliant
 - ISO 29119, TMMi, TPI-Next
- Model-based
 - Proces model, state model, operational profiles
- Analytical
 - Risk-based testing
- Artificial Intelligence in Testing



Shift-Left



but not only!



- Much more efficient
- Some defects are hard to find at late stages
- Prevents defect multiplication

Shift-Left Practices

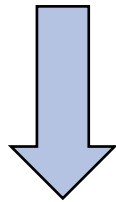
- Quality is a team effort
- Focus on unit & integration testing
- Code coverage
- Static and dynamic analysis
- Effective reviews
- Early test design
- Early product risk-analysis
- Requirements engineering



Yes	<input type="checkbox"/>
No	<input type="checkbox"/>



Shift-Left Management



Invest skills/tools
Upfront effort
Understand
Lead & Support



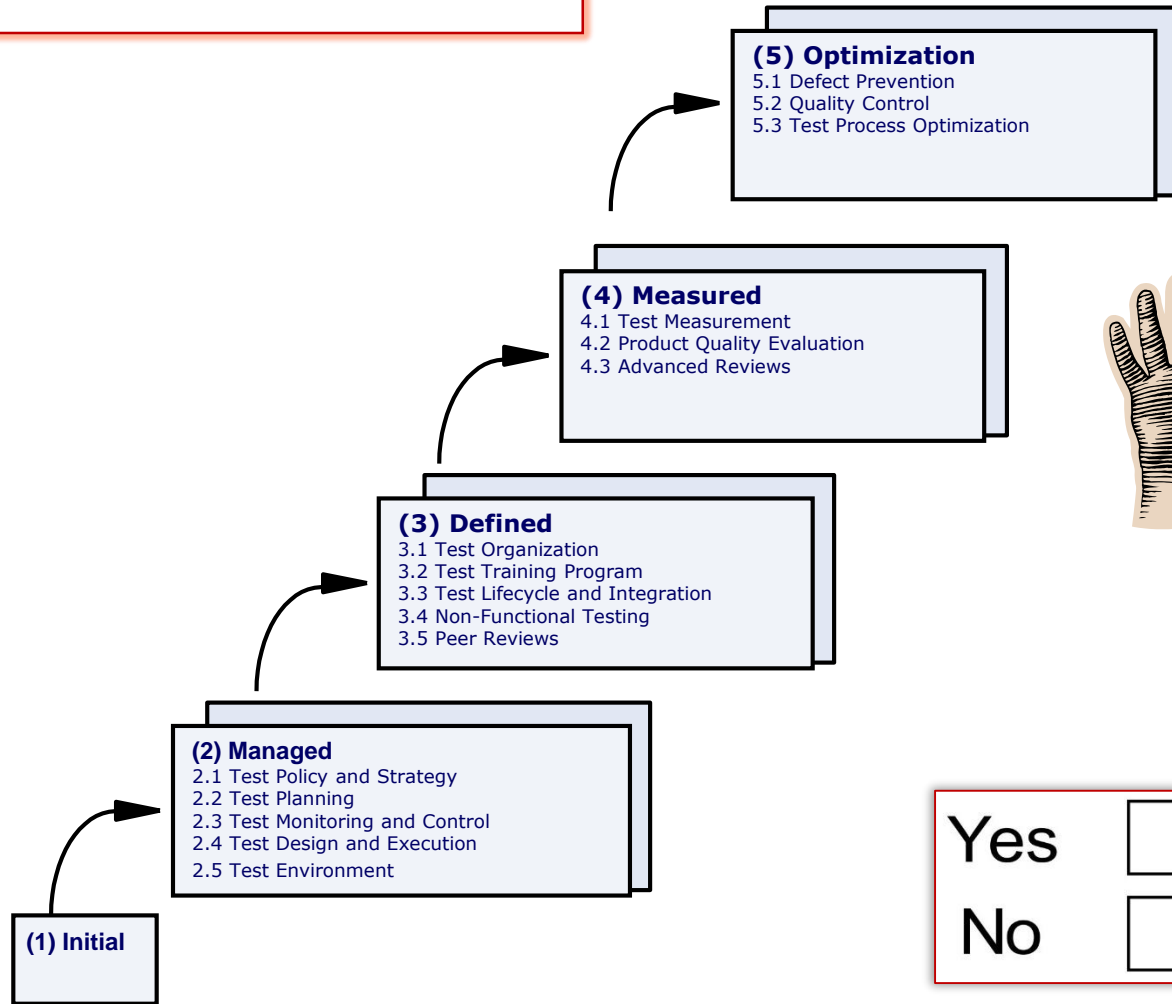
Train developers on testing
Train testers on technical testing

Process Compliant - TMMi

70% is at TMMi level 1



Source: IEEE Software,
Jan/Feb2018

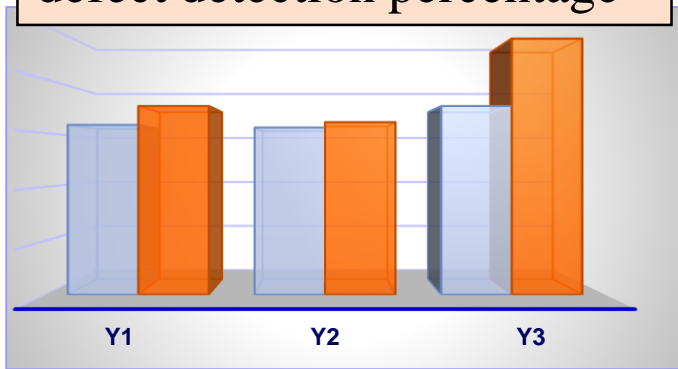


Yes	<input type="checkbox"/>
No	<input type="checkbox"/>

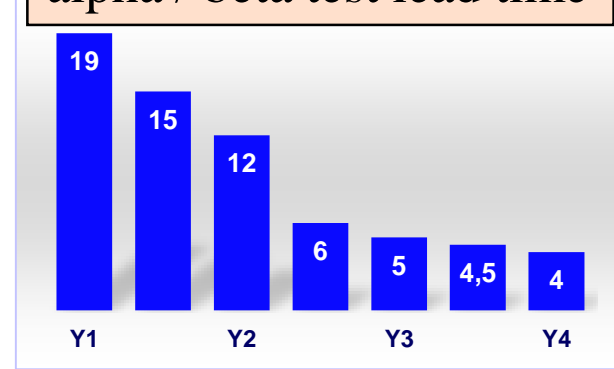
A large, 3D red question mark icon is positioned to the right of the form.

Results TMMi Level 2 & 3

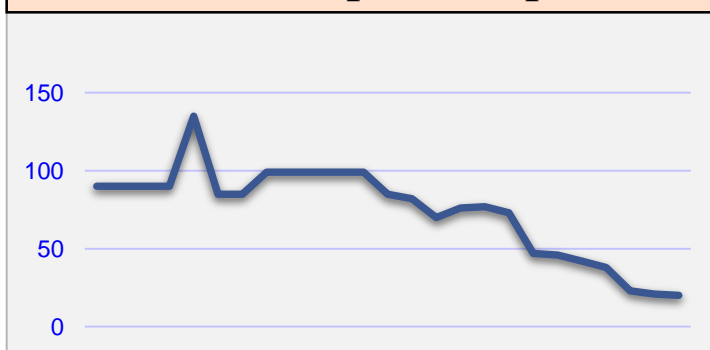
defect detection percentage



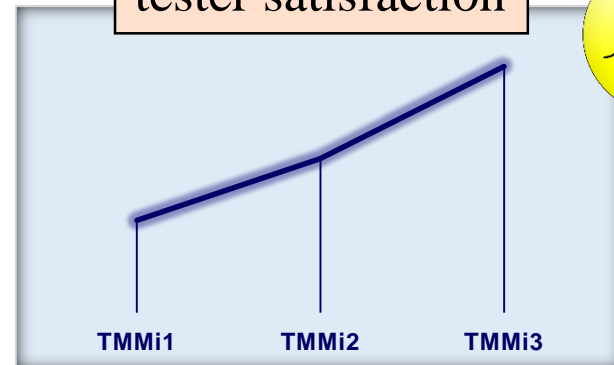
alpha / beta test lead time



% deviation - spent vs. planned



tester satisfaction



Analytical Testing

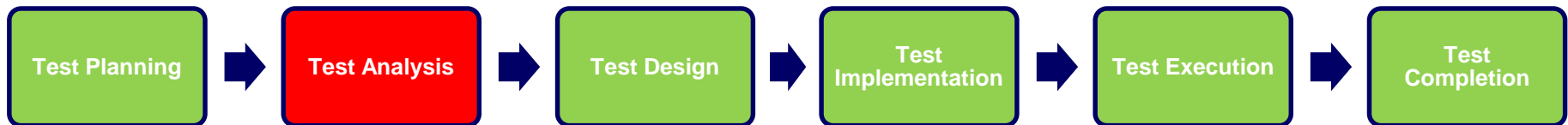
Analytics is the discovery, interpretation, and communication of **meaningful patterns** in data

Analytics, information resulting from the **systematic analysis** of data or statistics, these analytics can help you better **decisions**

Analytics often involves studying past **historical data** to research potential **trends**, to improve the business by gaining **knowledge**

Analytical Testing

- ~~Test team analyzes the test basis to identify the test conditions to cover (ISTQB, 2018)~~
- Analytical Testing \neq Test Analysis



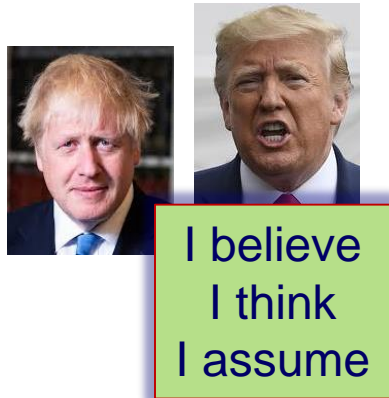
Analytical Testing

Testing based on a systematic analysis of data to discover useful information, supporting test improvements.

E. van Veenendaal (2018)

historical data, statistical analysis, modelling meaningful patterns, research potential trends

Where are we?



Testing



Politics

Opinion

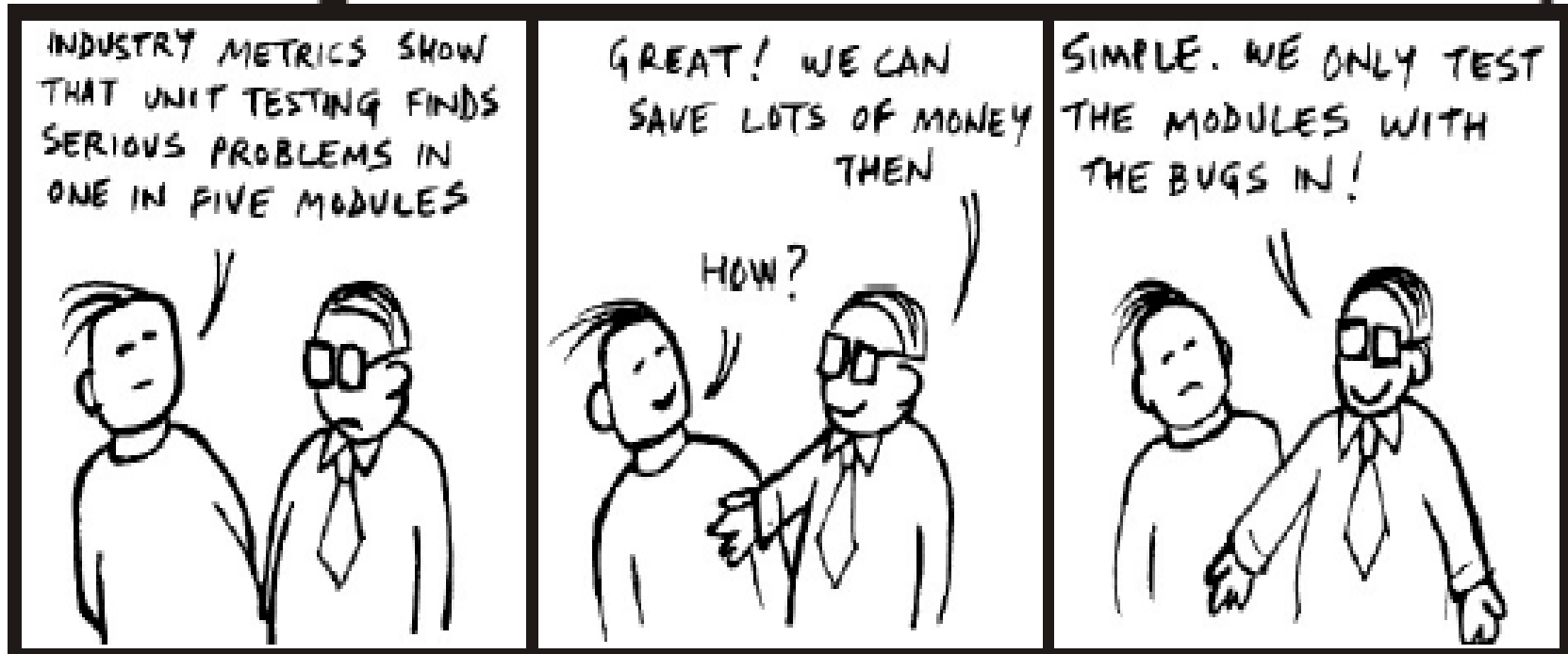
Engineering

Evidence

- Exhaustive testing is not possible
- An intelligent way to close the gap

Testing Ted

Gilchrist & Downing



Things to Analyze

- Defect based

- complexity
- size
- re-use
- technology
- lifecycle

.....



- Impact

- usage patterns
- configurations
- ..

- Tune!

- test techniques
- test automation

Using Analytics & Mastering Analytical Testing will be a Revolution in Software Testing



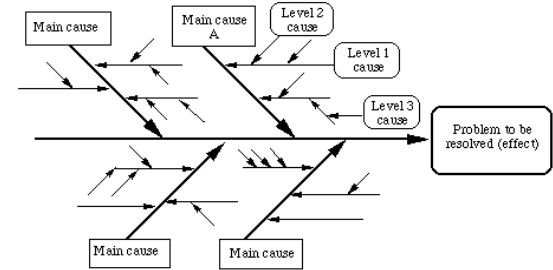
Tomorrow

as a tester you never assume!

■ Perform Root Cause Analyses

- Select defects for analyses
- Identify root cause
- Determine common cause
- *Build statistics and start understanding*

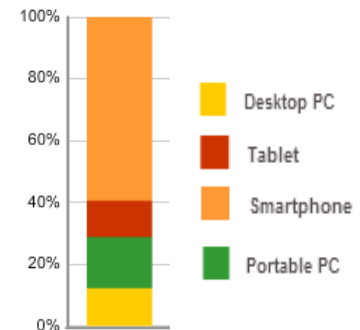
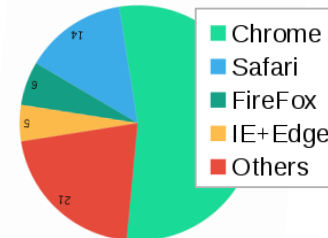
Likelihood



■ Investigate Real-Life Usage

- *Build statistics and start understanding*

Impact



Risk-Based Testing

- “using the results to improve testing”
- Risk = Likelihood x Impact 🙄
- Much better Risk = f (Likelihood, Impact)

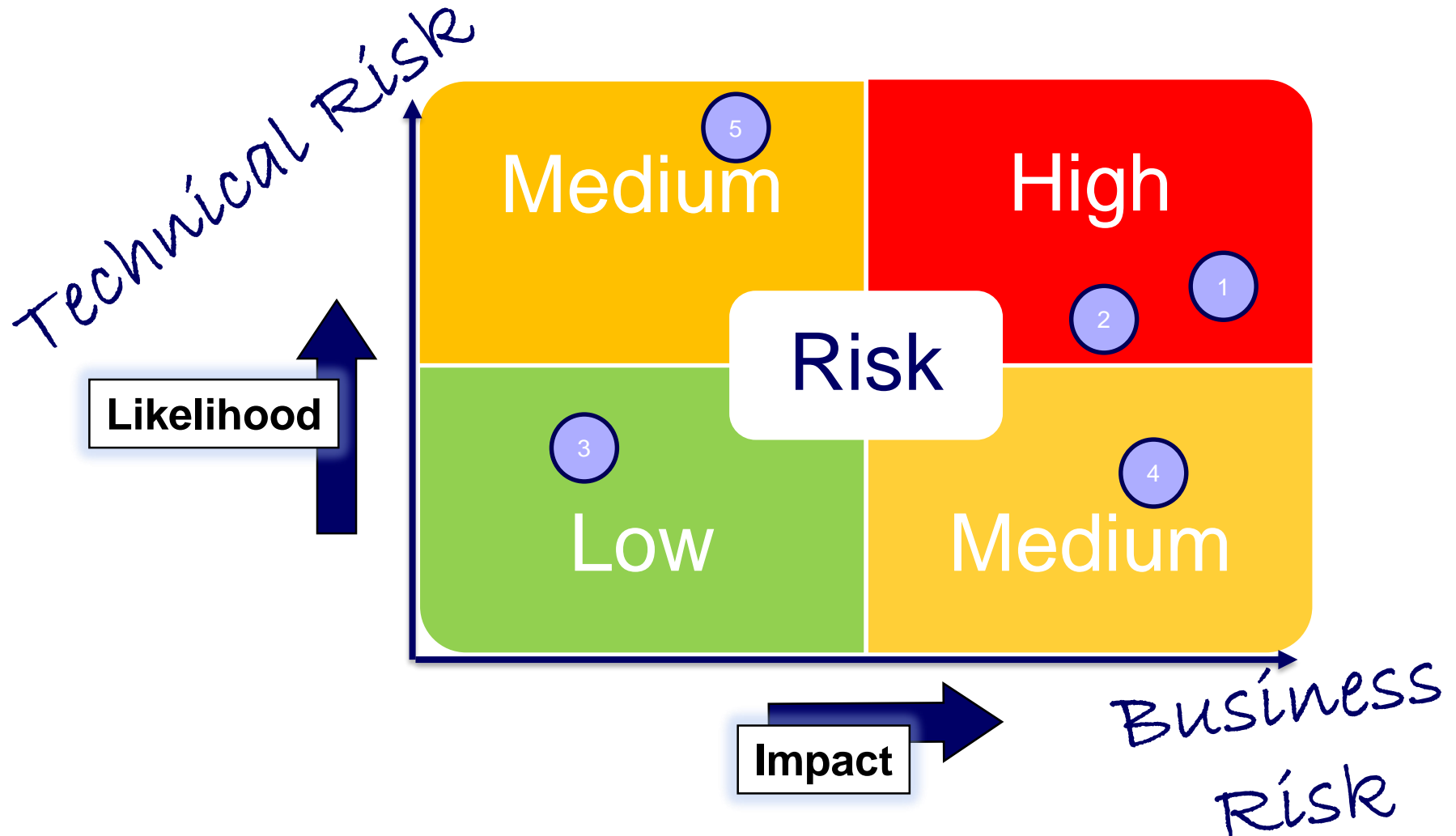
- **Analyse** every test item

- use likelihood **data / knowledge**
- use impact **data / knowledge**

AND - validate / discuss with experts, stakeholders



Product Risk Matrix



For the Testing Community

- Software systems are growing
 - Recalls due to defects are growing
 - Our answers are driven by opinion not evidence
-
- We need a theory of software testing
 - We need to start doing research
 - this shall cover Analytics in Testing
 - We need to become an engineering discipline



For the Tester



- It depends
- Consider the available test strategies
 - ❖ The “easy” ones - It’s AND rather than OR
- Consider more advanced test strategies
 - ❖ Tomorrow starts Today - It’s not optional!

analyze, study, investigate, know
(don’t just follow social media)

THE END

**THANK
YOU!**





www.erikvanveenendaal.nl



erik@erikvanveenendaal.nl



@ErikvVeenendaal