

# Reliable Test Effort Estimation

It's a kind of magic

Ruud Teunissen

Polteq Test Services BV – The Netherlands

# Objectives for this keynote

- Make your estimate more reliable
- *Discover the "principle" of estimation*
- *Share tips and tricks to better estimate your effort*

Enjoy Test Estimation

# How do you estimate your effort,...

...if you don't know what to do

...if you don't know how to do it

...if you don't know who will do it

...if you don't know where to do it

...if you don't know with what to do it

...if you don't know when to do it

...if you don't know what to do first, last, in between





# Test Effort Estimation - Approaches

- Experiences in similar/previous projects

# Test Effort Estimation - Approaches

- Experiences in similar/previous projects
- Historical data related to:
  - Time spent in design and realisation phases
  - Size of test basis
    - #user requirements, #pages, function points
  - Data model
    - #entities, fields
  - Size of test object
    - KLOC, # screens or fields

***Prerequisite:*** some kind of data (metrics) available to (cor)relate to the test effort

# Estimation based on ratios – historical data

- FD : Development : Functional test = 2 : 5 : 3

Formally complete FD, waterfall development method, 3GL programming language, and a structured test approach.

- (FD+TD) : (P + UT + UIT) : Functional test = 1 : 3 : 3

FD not detailed/complete, experienced developers, “starting” test approach.

- FD : Development : Functional test = 1 : 2 : 1.2

Formally complete FD, waterfall development method, experienced developers, driven by risk, and a maximized budget, and a structured test approach.

# Test Effort Estimation - Approaches

- Experiences in similar/previous projects
- Historical data
- Predefined budget
- Intuition of the experienced tester
- Extrapolation
- Work Breakdown Structure
- "Industry standard"



# Testing practice – “Industry Standard”

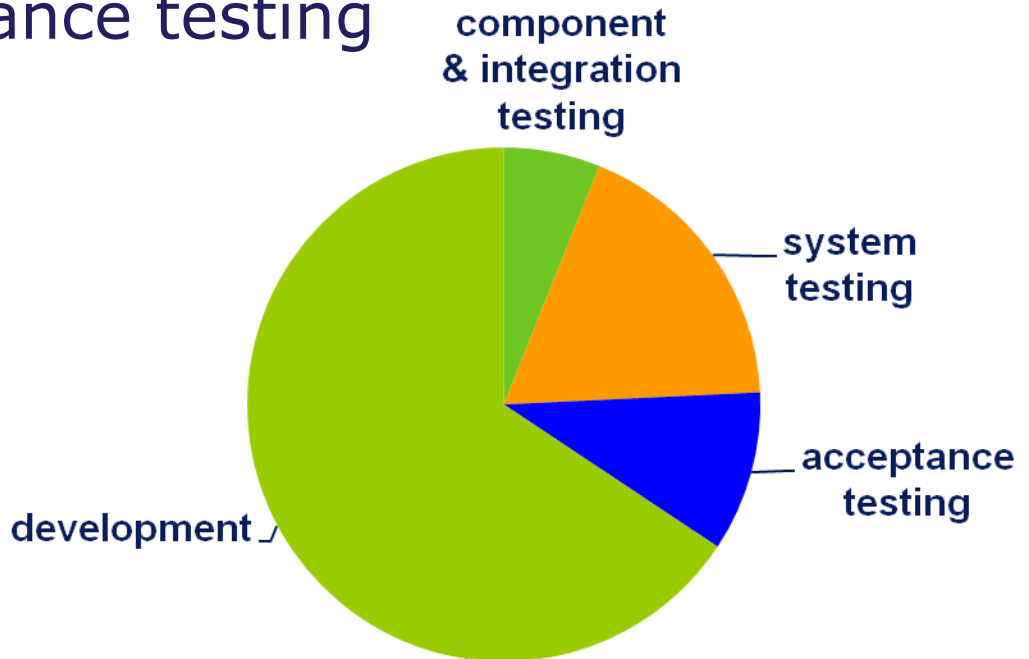
On average 35% of the hours spent on the total development project is reserved for testing

Increasing to 40%

5-7% for component and integration testing

18-20% for system testing

10% for acceptance testing



Development =	
Functional Design	+
Technical Design	+
Coding	

# Test Effort Estimation - Approaches

- Experiences in similar/previous projects
- Historical data
- Predefined budget
- Intuition of the experienced tester
- Extrapolation
- Work Breakdown Structure
- “Industry standard”
- ...

# Question

**What effort is required to test our new website?**



**I need your answer  
within 2 minutes**



**BEST IN TEST!**

**Laatste nieuws**

- Masterclass Software Testen  
In januari start Polteq weer met een nieuwe Masterclass Software Testen voor Junior Testers.
- Koninklijke Onderscheiding voor Martin Pol  
Op zaterdag 11 september is Martin Pol benoemd tot Ridder in de Orde van Oranje-Nassau.
- Meer nieuws

Zoeken

**Agenda**

- TMap® Next Advanced (English)  
04 okt,05 okt,06 okt,18 okt | Amersfoort
- Agile Testen  
04 okt | Amersfoort
- TMap® Next Advanced (English)  
04 okt,05 okt,06 okt,18 okt | Amersfoort

- ISTQB Foundation
  - » Bekijk opleidingsagenda
  - » Bekijk evenementenagenda

Onze nieuwe brochure **Testopleidingen** is uit! **Klik hier en vraag hem gratis aan!**

**Polteq e-Learning**  
NEW 



Home

**Polteq: Leaders in software testing**

Polteq is met haar 100 medewerkers het grootste 100% in software testen gespecialiseerde bedrijf in Nederland.

Naast detachering van testengineers en testmanagers en het uitvoeren van testprojecten bij opdrachtgevers, levert Polteq een breed scala aan testdiensten. Bijvoorbeeld op het gebied van verbetering van testprocessen, het inrichten van testorganisaties, het adviseren over en implementeren van testautomatisering, het managen van testoutsourcing en het geven van testopleidingen en cursussen (inclusief certificering - ISTQB en TMap®).

— Testdiensten — Opleidingen — Vacatures



Polteq; kwaliteit in testen. Dat is de beste omschrijving voor de diensten en trainingen die Polteq u te bieden heeft. De professionals die bij Polteq werken kunnen in de functies van testengineer tot testmanager worden ingezet in een project. Daarnaast levert Polteq een volledig pakket van testdiensten.

[Lees verder](#)

Polteq biedt verschillende trainingen aan die gericht zijn op het behalen van een internationaal of nationaal erkend certificaat. Naast de certificeringstrainingen zijn er praktijktrainingen, maatwerktrainingen en trainingen voor iedereen die algemene testkennis wil opdoen.

[Lees verder](#)

In januari 2011 start Polteq weer met een nieuwe Masterclass Software Testen. Bekijk hier het actuele vacatureoverzicht van Polteq.

[Lees verder](#)

- Some additional info:
  - ← Lay out for all pages
  - Latest news, Search, Calendar, Promotions
  - ← some with clickable subitems
  - some lead to own “page”
  - Register for training
  - Request for info
  - Available in English & Dutch

# Testing our new website

1

2

3

4

5

6

7

8

9

10

# Question

**What would make  
your test effort estimate  
more reliable?**

# Make your effort more reliable

1

2

3

4

5

6

7

8

9

10

# The “average” or “similar” project

- Is a myth...
- There are always differences, no matter how small, that have an influence on the required effort



# Insight into:

*...my list...*

- Software Development
  - Life Cycle, Maturity, Experience, ...
- Quality Assurance
  - Reviews & Inspections, Quality Control, ...
- Expected quality per test level
  - Defects, Test Cycles, ...
- Test Process
  - Life Cycle, Maturity, Experience, ...
- Strategy
  - What? How thorough?
- Available Knowledge
  - Business, Domain, System, Application, ...
- Technical Environment
  - Infrastructure, Tools, ...

# You need to know what to test, ...

- the software, including documentation
- the infrastructure, including procedures
- the organization, user and operations
- the required quality level

## ... how it is designed and developed, ...

- traditional waterfall
- V-model
- iterative, incremental, ...
- agile, xp, ...

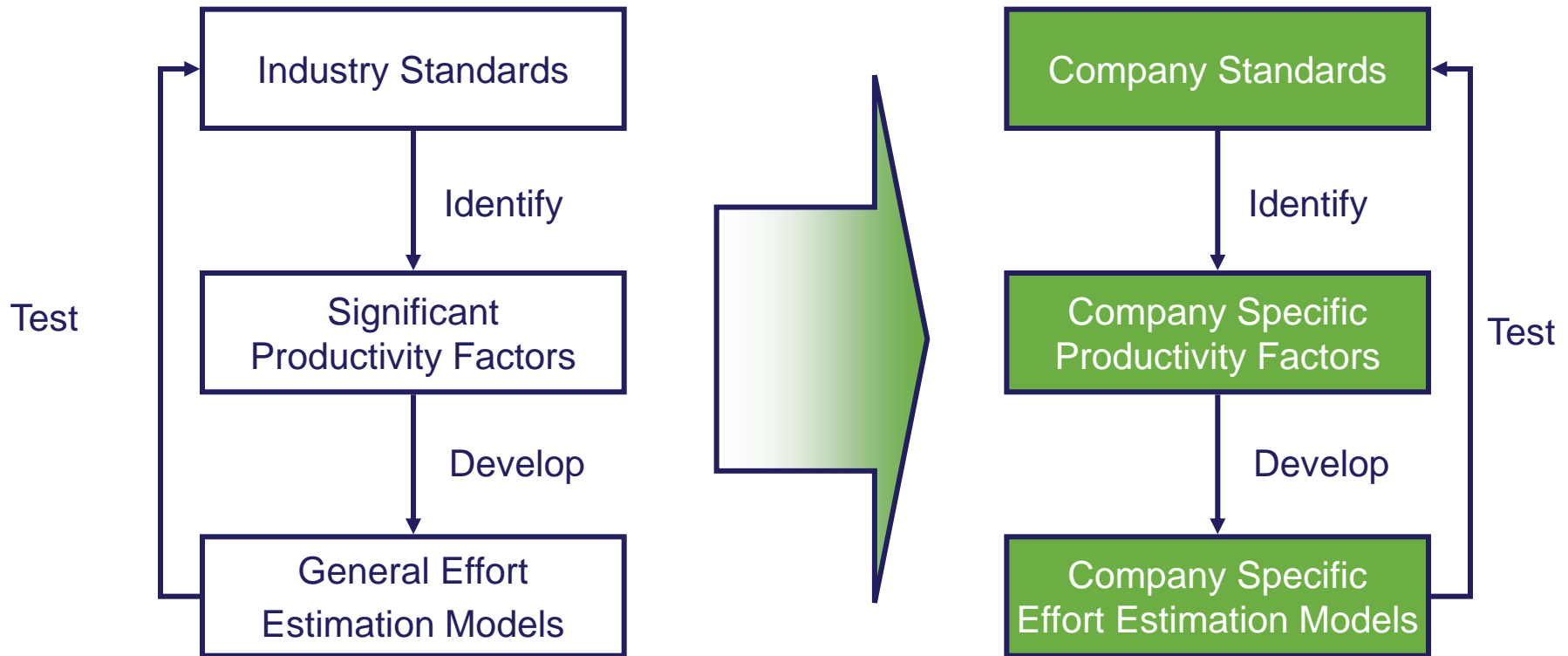
# ...and what stage of the project you're at!

- Project initiation
  - global estimate ( $\pm 30\%!!!!$ )
- High level tests
  - insight into "Functional" solution ( $\pm 10\%!!!!$ )
- Low level tests
  - insight into "Technical" solution ( $\pm 10\%!!!!$ )

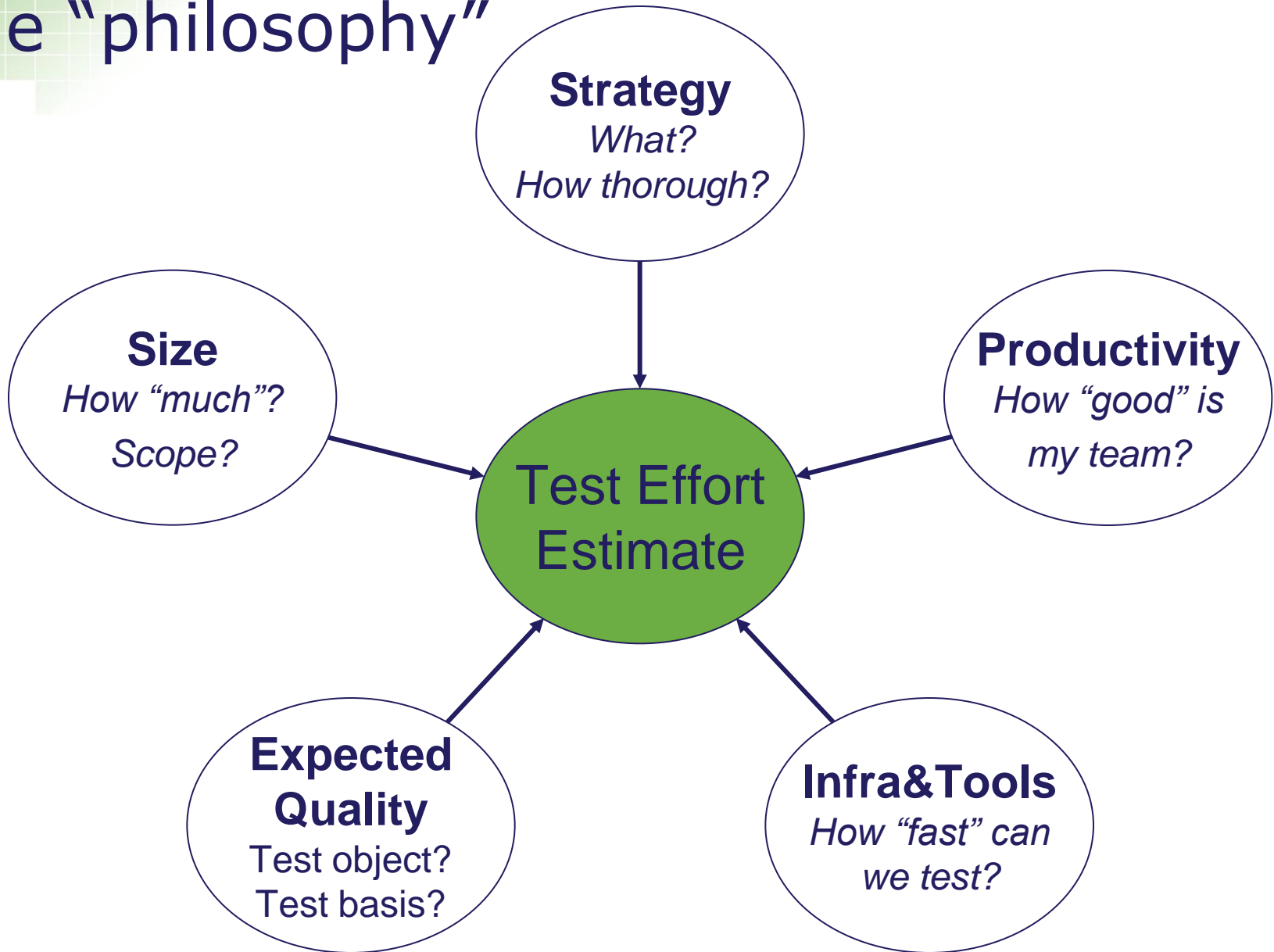
Management Expectation in a PRINCE2 environment

Project Management Approach: Projects in Controlled Environments

# Tune the industry standard to your situation



# The “philosophy”



“It’s better to be honest, than to have to admit you did not tell the whole truth at the start”

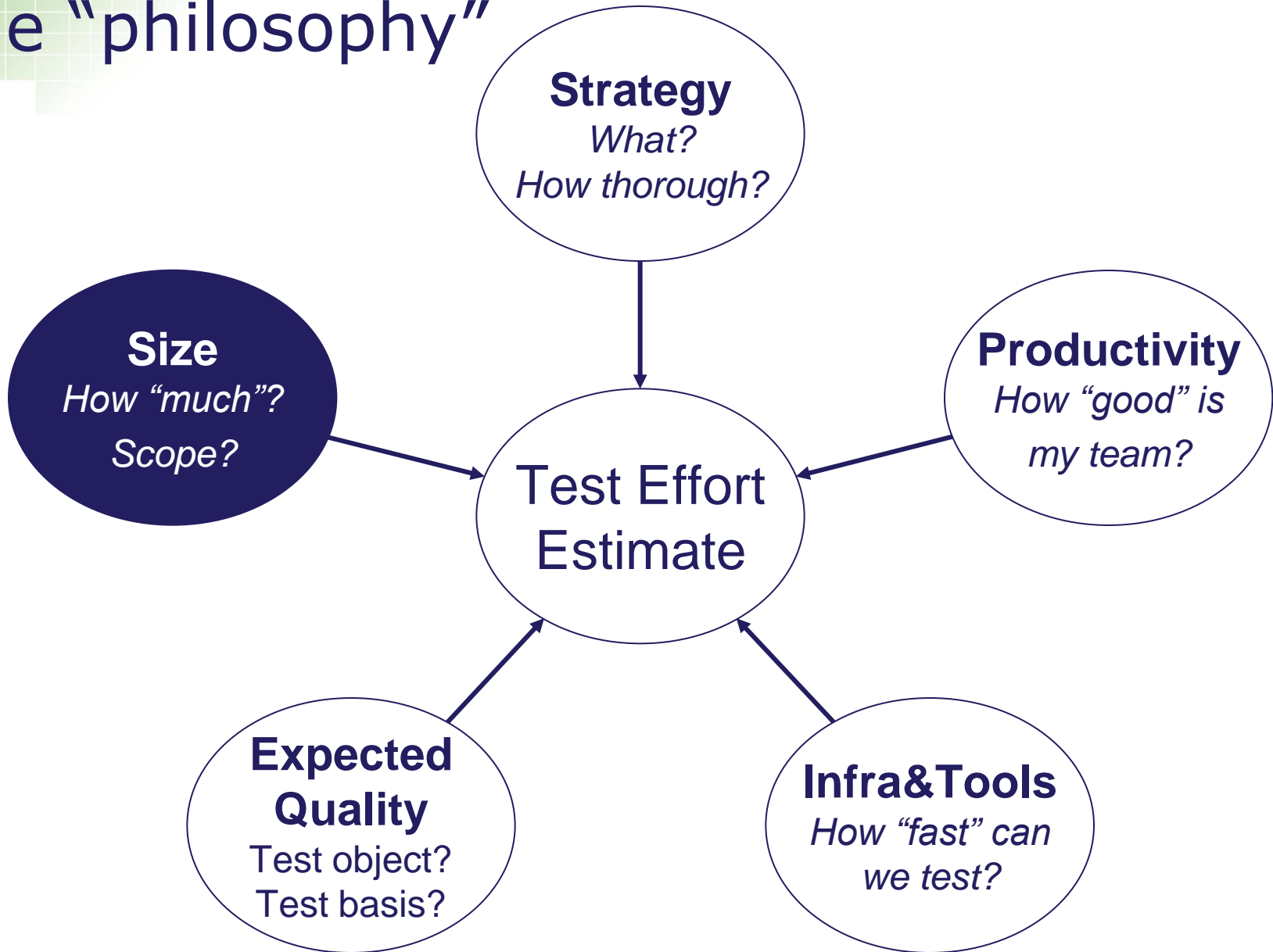
or designs

- Developers
  - say there are no “weak areas” in the software and ...  
*... act “surprised” by the number of bugs found and start disputing bugs when reported...*
- Testers
  - create beautiful test plans and strategies, but...  
*... are unable to execute them efficiently by lack of time, resources, knowledge and/or experience...*



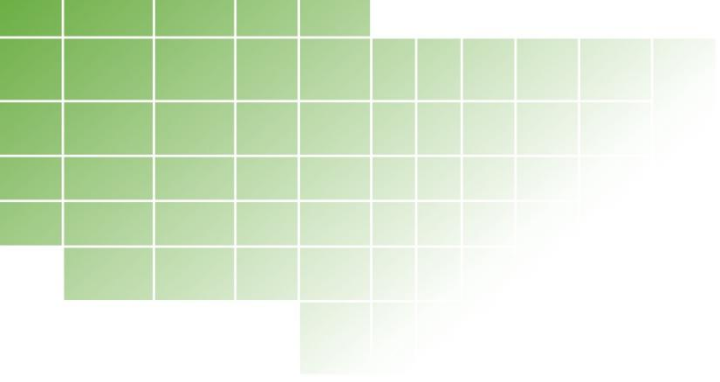
The “hardest” part

# The “philosophy”



# “Size” – you have to start somewhere...

- % of project budget
- Size of test basis (designs, requirements, use cases,...)
  - #user requirements, #pages, function points
- Data model
  - #entities, fields
- Size of test object
  - KLOC, # screens or fields
- Expected number of test designs, test procedures, test scripts, test cases, ...
- Poker points
  
- Intuition (is it magic after all?)



# Calculation Model Explained

# What will be your car insurance premium?

- Let's assume that \$35 per \$1000 (value of the car) is the standard premium
- Questions to be answered are:
  - Experience of the driver:
    - Young, eager, "rookie"? Experienced? Expert?
  - Address:
    - Different areas have different "risk profiles"
  - Type of car:
    - Economy or compact? Intermediate or family car? SUV?
  - Age of car:
    - Brand new? Recent? Old?
  - Damages and claims during recent years:
    - And what about your track record?

# Define the possible answers...

- High → an increase of the premium
- Medium → no impact
- Low → a decrease of the premium

Answer	Description
High	You've just obtained your driver's licence. <i>So you still have to learn how to drive...</i>
Medium	You've had your driver's licence and car for some time now, but you have not driven too many miles yet. <i>It's not experimental anymore and you have some experience, but you're not an expert yet.</i>
Low	You've had your driver's licence for a long time now, and you have driven lots of miles in different cars and circumstances. <i>You're familiar with driving under any circumstance.</i>

# Define the impact

- Start with two categories – relative to each other
  - Major impact
  - Minor impact

Question	Category
1. Experience of the driver	Major
2. Address	Minor
3. Type of car	Major
4. Age of car	Minor
5. Damages and claims	Major

# Calculation model

- “proven technology”
  - just look at the way insurance companies define the premium you have to pay for your car insurance  
*(at least how they say they do it 😊)*

- **Major**

“high” : “medium” : “low” = 8 : 4 : 2

- **Minor**

“high” : “medium” : “low” = 4 : 2 : 1

Question	Impact	Answer	Factor
1. Experience of the driver	Major	High	8
		Medium	4
		Low	2
2. Address	Minor	High	4
		Medium	2
		Low	1
3. Type of car	Major	High	8
		Medium	4
		Low	2
4. Age of car	Minor	High	4
		Medium	2
		Low	1
5. Damages and claims	Major	High	8
		Medium	4
		Low	2

Question	Impact	Answer	Factor
1. Experience of the driver	Major	High	8
		Medium	4
		Low	2
2. Address	Minor	High	4
		Medium	2
		Low	1
<div style="background-color: #1a237e; color: white; padding: 5px; display: inline-block;"> <math>\Sigma</math> "medium" = 16 </div>	Major	High	8
		Medium	4
		Low	2
4. Age of car	Minor	High	4
		Medium	2
		Low	1
5. Damages and claims	Major	High	8
		Medium	4
		Low	2

# Calculation model continued

Question	Answer	Factor
1. Experience of the driver	High	8
2. Address	Medium	2
3. Type of car	High	8
4. Age of car	Low	1
5. Damages and claims	Medium	4
$\Sigma$ Factors	23	
<i>Risk Indicator (<math>\Sigma</math> Factors / 16)</i>	1,44	

# Premium Calculation

- Risk Indicator = multiplying factor  
Minimum premium = \$17.5  
Maximum premium = \$70  
Our premium =  $\$35 * 1.44 = \$50$
- Risk indicator → ranges

Risk Indicator	Premium
< 0,7	25
0,7 - 0,9	30
0,9 - 1,1	35
1,1 - 1,3	40
> 1,3	45

# Project Risk Indicator

## Business – Development – Testing

A Real Life Case by Ruud Teunissen

## Q1 Business Risks

- |   |                                                                  |
|---|------------------------------------------------------------------|
| 2 | The priority of the project relative to other projects is low    |
| 4 | The priority of the project relative to other projects is normal |
| 8 | The priority of the project relative to other projects is high   |

## Q2 Technology in operation

- |    |                                                                        |
|----|------------------------------------------------------------------------|
| 4  | The technology has been used several times within the organization     |
| 8  | The technology is new, but similar to others within the organization   |
| 16 | The project uses a technology which is totally new to the organization |

### Q3 Experience Design and Development Team

2	An experienced design and development team
4	A mixture of experienced and non-experienced design and development team
8	A non-experienced design and development team

### Q4 Complexity of the application

2	The degree of complexity of the processing (simple, standalone) of the application relative to other applications is low
4	The degree of complexity of the processing (medium complexity, medium interfaces) of the application relative to other applications is normal
8	The project is focussing on (a) complex application(s) with many interfaces to other applications.

## Q5 Development Method

2

The project development method for the application has been commonly used within the organization for several times in the past

4

The project is to be developed using a new development method, similar to others within the organization

8

The project is to be developed using a new development method, which is considerate experimental within the organization

## Q6 New build / Maintenance

4

The project is a maintenance project, updates on the current applications

8

The project is a combination of a changes project and new build

16

The application is primarily new build

## Q7 Expertise test team

2	An experienced test team
---	--------------------------

4	A mixture of experienced and non-experienced test team
---	--------------------------------------------------------

8	A non-experienced test team
---	-----------------------------

## Q8 Level of re-usability testware

4	A usable, general initial data set and specified test cases are available
---	---------------------------------------------------------------------------

8	A usable general initial data set (tables, etc.) is available
---	---------------------------------------------------------------

16	No re-usable testware is available
----	------------------------------------

# Project Risk Indicator

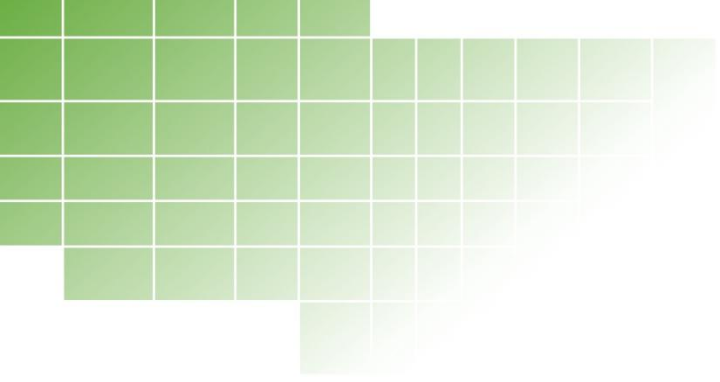
$$\text{PRI} = (A1 + A2 + A3 + A4 + A5 + A6 + A7 + A8) / 44$$

Project Risk Indicator	% Project time spent on testing
< 0.70	29%
0.70 – 0.90	32%
0.90 – 1.10	36%
1.10 – 1.30	41%
> 1.30	45%

# The "philosophy"

## Checklist





# Detailed Effort Estimate

# Question

**What quality aspects do you normally cover with your test?**



**Let's vote**



## Quality aspects to be covered

1	Suitability	
2	Accuracy	
3	Interoperability	
4	Portability	
5	Security – Functional & Technical	
6	Reliability	
7	Efficiency – Performance, Load, Stress	
8	Usability	

# Context – Historical Data Available

- System Test
  - Functional and Non-Functional Tests
- Experience from previous projects
  - Functional Design 1
  - Realization 2
    - = Technical Design + Code + UT + UIT
  - System Test 2
    - = 85% Test + 15% Test management
  - *Quality Aspects Covered* *Average*
    - Functionality 75%
    - Performance 10%
    - Security 10%
    - Continuity 5%

# Context – Historical “Way of Estimating”

<i>Test Effort Estimate</i>		
Functional Design		200 hrs
Realization		400 hrs
System Test		400 hrs
<i>Functionality</i>	<i>75%</i>	<i>300 hrs</i>
<i>Performance</i>	<i>10%</i>	<i>40 hrs</i>
<i>Security</i>	<i>10%</i>	<i>40 hrs</i>
<i>Continuity</i>	<i>5%</i>	<i>20 hrs</i>

# The "philosophy"



# Strategy

avg \* 3/2

1. Functionality	High	1,13
	Average	0,75
	Low	0,50
	Out of scope	0,00
2. Performance	High	0,15
	Average	0,10
	Low	0,07
	Out of scope	0,00
3. Security	High	0,15
	Average	0,10
	Low	0,07
	Out of scope	0,00
4. Continuity	High	0,08
	Average	0,05
	Low	0,03
	Out of scope	0,00
Strategy = 1. + 2. + 3. + 4.		

avg \* 2/3

# Expected Quality

5. Complexity	H	12
	M	6
	L	3
6. Quality Functional Design	H	3
	M	6
	L	12
7. Quality Previous Tests	H	2
	M	4
	L	8
8. Experience Development Team	H	2
	M	4
	L	8
Expected Quality = ( 5. + 6. + 7. + 8. ) / 20		

# Infra & Tools + Productivity

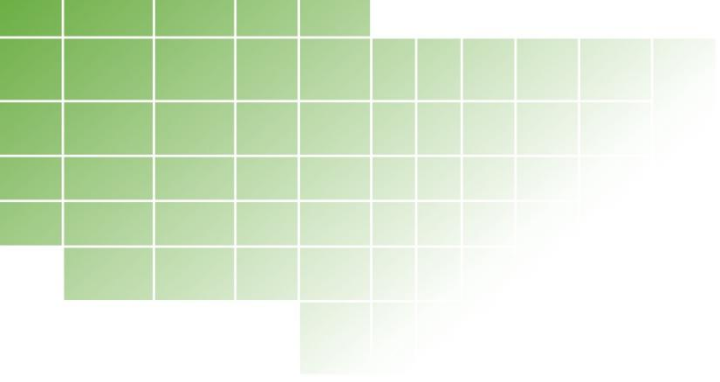
9. Test Environment	“Test Friendly”	3
	“Usable”	6
	“Test Unfriendly”	12
10. Tool Support	Defect and Time Management	1
	Defect or Time Management	2
	“Sorry”	4
11. Test data	Available and Described	2
	Available	4
	“Sorry”	8
Infra & Tools = ( 9. + 10. + 11. ) / 12		
12. Productivity	Experienced (Test and System)	0,7
	Experienced (Test or System)	1,0
	Inexperienced	1,5

# New Way of Estimating

## *Test Effort Estimate - Calculated*

<b>Strategy</b> Functionality High, Performance Out of Scope, Security Average, Continuity Out of Scope	1,23
<b>Expected Quality</b> Complexity Average, Quality Previous FD and Test High, Experience High	0,65
<b>Infra &amp; Tools</b> Test Environment “Usable”, Test Tools “Defect Management”, Test Data “Sorry”	1,33
<b>Productivity</b> Experienced (Test and Matter)	0,70
<b>System Test =</b> Strategy * Expected Quality * Infra & Tools * Productivity	298 hrs

# Proportianate Estimate



# Agile Estimate

# Estimate Size – Derive Duration

Size

Calculation

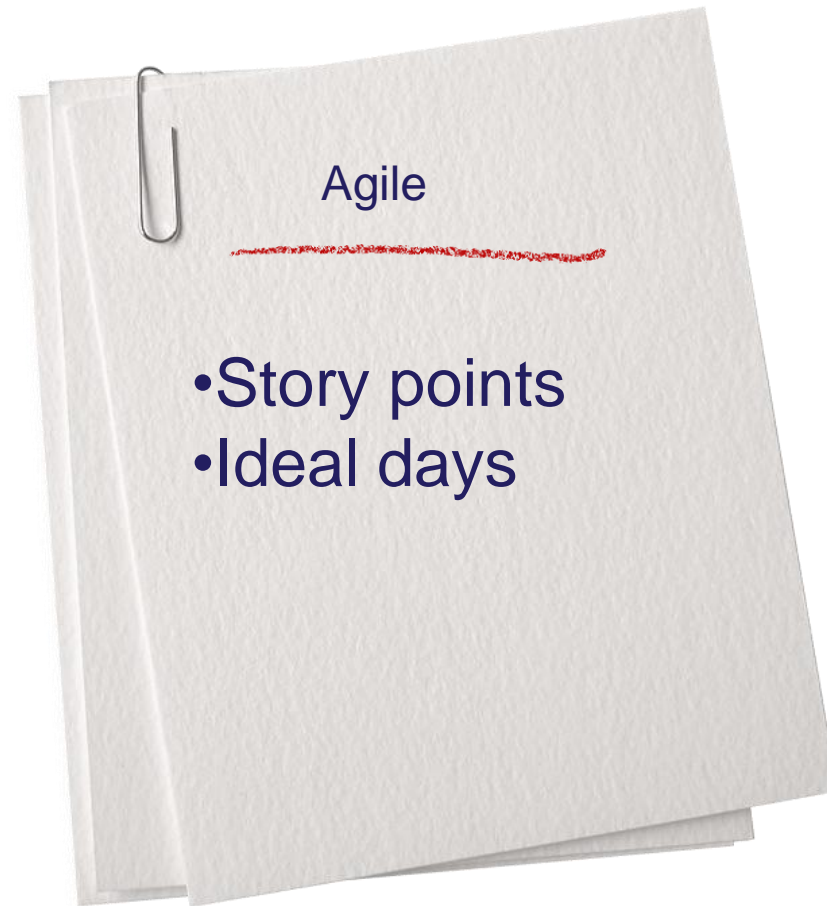
Duration

100 points

Velocity =  
20

$100/20 =$   
5 sprints

# Measures of size



# Story points

- The “bigness” of a task  
Influenced by
  - How hard it is
  - How much of it there is
- Relative values are what is important:
  - A login screen is a 2
  - A search feature is an 8
- Points are unit-less

As a user, I want to be able to have some but not all items in my selection highlighted on my screen

# Ideal days

- How long something would take if
  - it's all you worked on
  - you had no interruptions
  - everything you need is available
- The opposite of an ideal day
  - The duration of a football game is 90 minutes
    - So we expect 90 minutes high level football!
  - The effective playing time is much shorter
    - Euro 2000: 24 group games, on average 53 minutes

# Comparing the approaches

- Story points help drive cross-functional behavior
  - Story point estimates do not decay
  - Story points are a pure measure of size
  - Estimating in story points is typically faster
- 
- My ideal days cannot be added to your ideal days
  - Ideal days are easier to explain outside the team
  - Ideal days are easier to estimate at first
  - Ideal days can force companies to confront time wasting activities

# Estimate by analogy

- Comparing a user story to others
  - “This story is like that story, so its estimate is what that story’s estimate was.”
- Don’t use a single gold standard
  - Triangulate instead
- Compare the story being estimated to multiple other stories

# Use the right units

- Can you distinguish a 1-point story from a 2?
  - How about a 17 from an 18?
- Use a set of numbers that make sense  
I like: 0,  $\frac{1}{2}$ , 1, 2, 3, 5, 8, 13, 20, 40, 100, ?
- Nature agrees:
  - Musical tones and volume are distinguishable on a logarithmic scale

# Planning poker



- An iterative approach to estimating
- Steps
  - Each estimator is given a deck of cards, each card has a valid estimate written on it
  - Customer/Product owner reads a story and it's discussed briefly
  - Each estimator selects a card that's his or her estimate
  - Cards are turned over so all can see them
  - Discuss differences (especially outliers)
  - Re-estimate until estimates converge

# Planning poker—an example



Estimator	Round 1	Round 2
Erik	3	5
Thijs	8	5
Hans	2	5
Ruud	5	8

# How much can I get by <date>?

- Determine how many sprints you have
- Estimate velocity as a range, e.g.:
  - Our low speed is 14 points a sprint
  - Our average speed is 18 points a sprint
  - Our high speed is 22 points a sprint
- Multiply low velocity × number of sprints
  - Count off that many points
  - These are “Will Have” items
- Multiply high velocity × number of sprints
  - Count off that many more points
  - These are “might haves”

# The "philosophy"

## Checklist



# Question

**Ever been in a position where  
the budget for your test was  
fixed?**



# Proportionate estimation

Based on:

- Total available budget for testing
- Risk class (test strategy) translated to a “weight factor”
- Size

Example: 200 hrs available for 4 components to test

# Proportionate estimation

200 hrs available for 4 components to test

Solution I:

Equal division of hours

Component	Effort
1	50
2	50
3	50
4	50
<b>Total</b>	<b>200</b>

# Proportionate estimation

200 hrs available for 4 components to test

Solution II:

Estimation based on size:

Comp.	Size	Perc.	Effort
1	10	10%	20
2	20	20%	40
3	30	30%	60
4	40	40%	80
<b>Total</b>	<b>100</b>	<b>200</b>	<b>200</b>

# Proportionate estimation

200 hrs available for 4 components to test

Solution III:

Proportionate:

include risk class (strategy)  
translation into "weight factor"

Risk Class	Factor
A	1,50
B	1,00
C	0,75

Comp.	Size	Risk Class	Factor	S x F	Scaling	Effort
1	10	A	1,50	15,0	2,11	31,58
2	20	B	1,00	20,0	2,11	42,11
3	30	B	1,00	30,0	2,11	63,16
4	40	C	0,75	30,0	2,11	63,16
<b>Total</b>				95	$200 / 95 = 2,11$	200

Questions?

Ruud Teunissen

Polteq Test Services BV – The Netherlands

[ruud.teunissen@polteq.com](mailto:ruud.teunissen@polteq.com) - <http://www.polteq.com>

Thank you very much for your  
attention & participation & input &  
questions & ...

**Ruud Teunissen**

Polteq Test Services BV – The Netherlands

[ruud.teunissen@polteq.com](mailto:ruud.teunissen@polteq.com) - <http://www.polteq.com>

# About the speaker

## **Ruud Teunissen**

*Polteq Test Services, The Netherlands*

In the testing world since 1989, Ruud Teunissen has held numerous test functions in different organizations and projects: tester, test specialist, test consultant, test manager, etcetera. Ruud is co-author of Software Testing - A Guide to the TMap® Approach and is a frequent speaker at (inter)national conferences and workshops. He was a member of the program committee for Quality Week Europe and EuroSTAR. Ruud is currently International Test Consultant at Polteq Test Services BV.



TPA® and TMap® are registered trademarks of Sogeti