



Gebonden en ongebonden testen

Workshop, 16 maart 2015

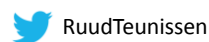
Ruud Teunissen en Rik Marselis

Even voorstellen: Ruud



TEST

- er, -engineer, -coördinator,
- teamleider, -manager,
- toolspecialist, -docent,
- auteur, -sales, -consultant,
- adviseur, -architect, -...



Even voorstellen: Rik




Management Consultant Quality & Testing bij  **SOGETI**

Bijna 35 jaar IT ervaring, zo'n 17 jaar kwaliteit & testen

Adviseur, procesverbeteraar & coach bij vele organisaties

Prince2 Practitioner, CMMI en CISA

Docent voor diverse trainingen, bijv. Agile testen

TMap, TPI en ISTQB geaccrediteerd 

Research → Auteur div. boeken en artikelen

Fellow van SogetiLabs, Spreker op div. conferenties

En daarnaast:

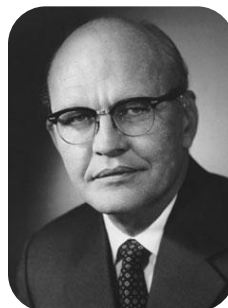
Voorzitter  (vereniging voor testers, 1700 leden)



@rikmarselis



De oorsprong



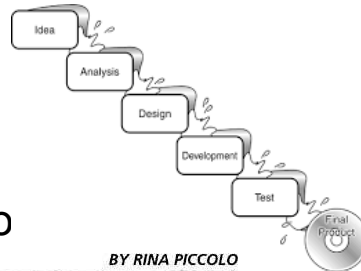
9-9-1947



IT?



- Waterval processen
- Stand-alone
- Centrale verwerking
- Beperkte business risico



SIX CHIX



Testen?



- Onbekend, geen proces, kennis & budget
- Geïntegreerd in het ontwikkelproces
“trial-and-error”
- Alleen testpioniers



Testen?



- Onbekend, **Agile** is & budget
- Geïntegreerd **Agile** proces
- Alleen testpioniers



Testen



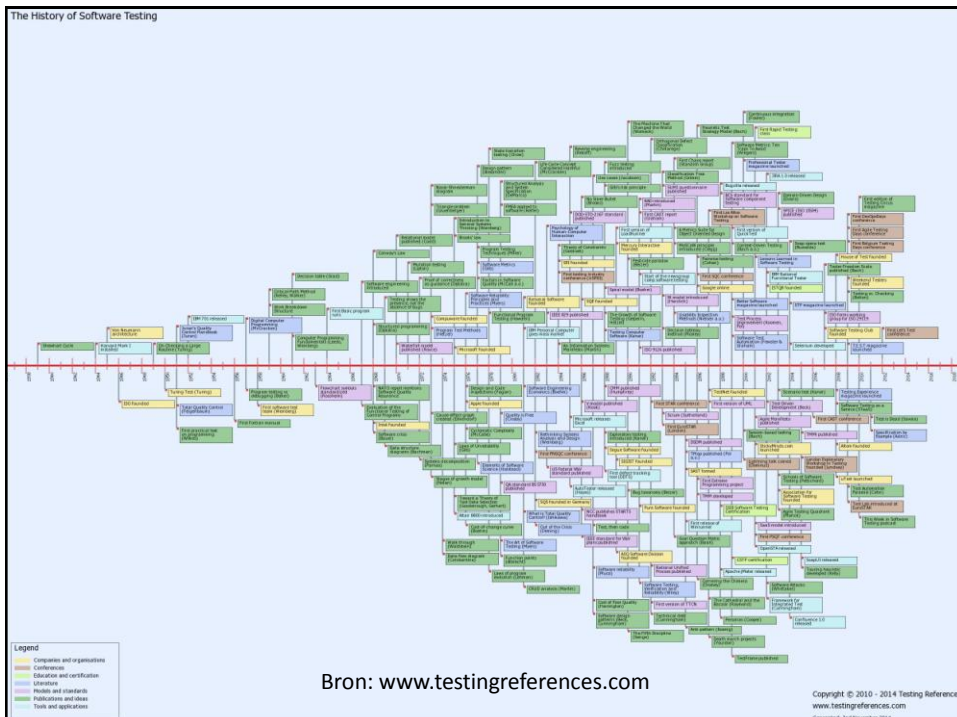
- Testers
 - gesjeesde ontwikkelaar/gebruiker, zwangere vrouwen, boventallig personeel, ...
 - pioniers [gelukkig wel]

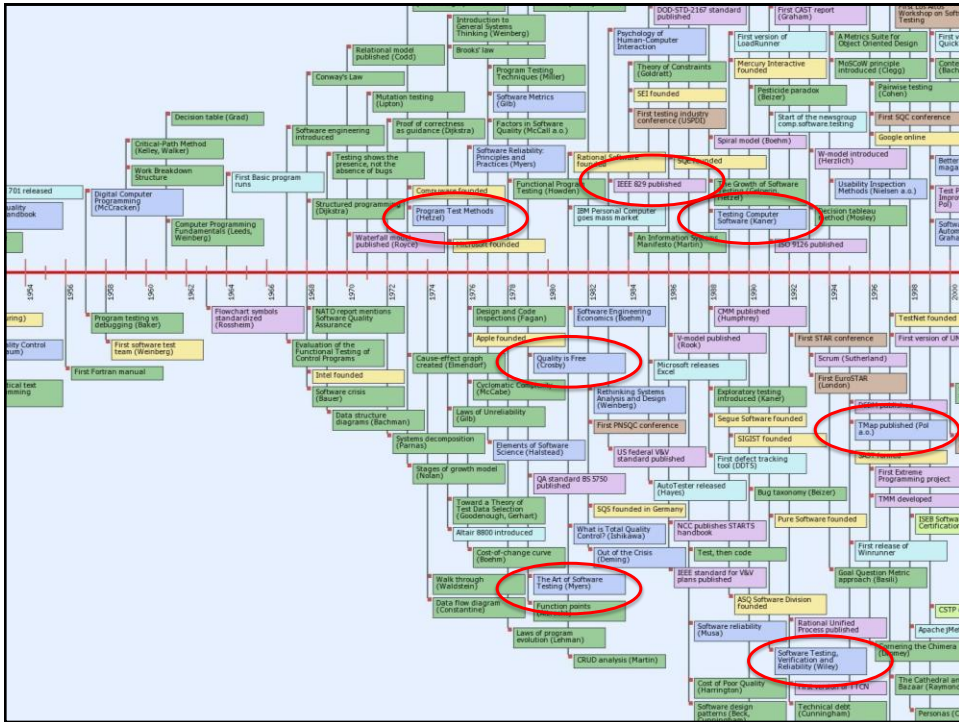


Testen



- Testers
 - gesjeesde ontwikkelaar/gebruiker, zwangere vrouwen, boventalig personeel, ...
 - pioniers [gelukkig wel]
- Testen [enkel uitvoering]
 - op basis van kennis van de systemen
 - niet op basis van testkennis & expertise
- Technieken [eerste ideeën]
 - waar gaat het fout in productie?
 - grenswaarden, beslissingen, batchverwerking





Statisch testen

Ik wil een groene auto ...





OF




Dynamisch testen



Gebonden testen



- Testontwerptechnieken
- Dekkingsvormen
- ...

Welke testontwerptechnieken ken je?



15

Welke testontwerptechnieken ken je?

TMap NEXT

- Equivalence classes
- Boundary value analysis
- Decision points (CC, DC, CDC, MCDC, MCC)
- Orthogonal arrays
- Pairwise testing
- Data cycle test (CRUD)
- Statistical usage (op.prof.)
- Right paths / Fault paths
- Checklist
- Decision table
- Data Combination Test
- Elementary comparison test
- Process Cycle test
- Real-life test
- Semantic test
- Syntactic test
- Use case test
- Error Guessing
- Exploratory testing
- Reviews
- Walkthroughs
- Inspections

ISTQB

- Equivalence partitioning
- Boundary value analysis
- Decision table testing
- Cause Effect graphing
- State transition testing
- Use case testing
- User story testing
- Structure based (SC, CC, DC, CDC, MCDC, MCC)
- Error Guessing
- Exploratory testing
- Orthogonal arrays
- Pairwise testing
- Domain analysis
- Defect based techniques
- Checklist
- Combinatorial / Class. tree
- Informal review
- Technical review
- Walkthrough
- Inspection
- Static analysis

Torbjorn Ryber

- Exploratory testing (this is an approach, not a technique)
- Inspection
- Walkthrough
- Technical review
- Informal review
- Modelling
- Equivalence partitions
- Boundary values
- Domain tests
- Business process testing
- Use case testing
- State based testing
- Decision tables & trees
- Elementary comparison
- Combinatorial testing
- Data cycle testing
- Syntax testing
- Time cycle testing
- Program Logic (SC, BC, PC)
- Data flow testing

Boris Beizer

- Control-flow testing
- Loop testing
- Data-flow testing
- Transaction-flow testing
- Domain testing
- Syntax testing
- Finite-state testing

And more...

16

Workshop 1: intro



- Bekijk de testuitdaging
- Gebruik één testontwerptechniek om testgevallen te specificeren
- Als je klaar bent hang je één kopie van je testgevallen aan het kanban-bord en krijg je de instructie om de test uit te kunnen voeren
- Log de werkelijke resultaten en gebruik indien nodig een bevinding-formulier

Workshop 1: T.O.T.

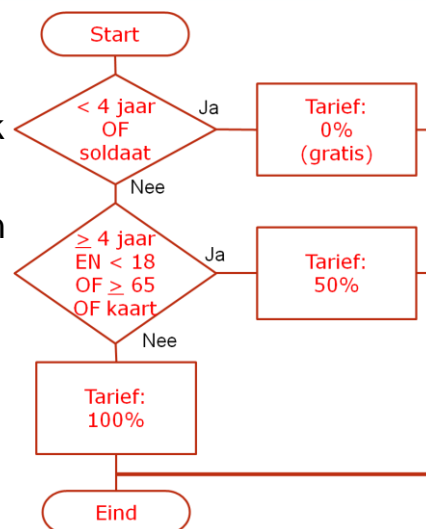


Specificeer fysieke testgevallen.

Kies een testontwerptechniek van het Kanban-bord

Houd de werkelijke resultaten bij en eventuele bevindingen.

Klaar?
Probeer nog een techniek



Terugkoppeling workshop1

- Hoeveel testgevallen per techniek?
- Welke dekking per techniek?
- Fout gevonden?
- Oorzaak van de fout gevonden?



Coverage based test design techniques

Uitwerking voor BerekenOVpercentage voorbeeld

Proces Cyclus Test
Beslissingstabel
Elementaire Vergelijkingen Test
DataCombinatie Test



Coverage group example 1

Process

Process Cycle Test

21

Elaboration PCT

Price of travel in public transport

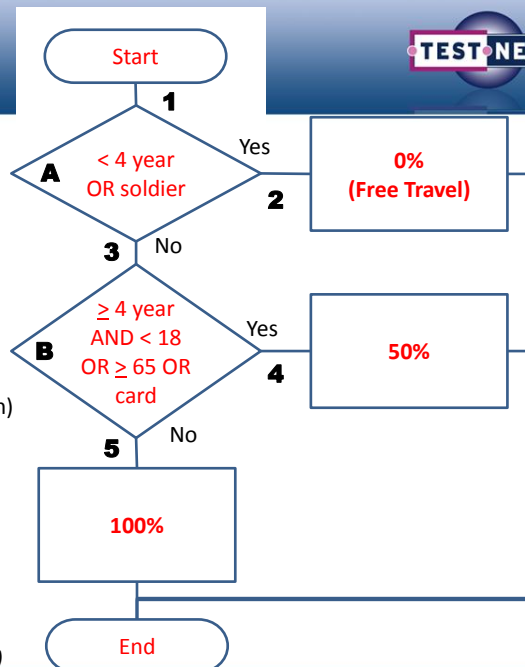
TM – 1 (every path):
 Testcase a: 1,2
 Testcase b: 1,3,4
 Testcase c: 1,3,5

TM – 2 (every pathcombination)

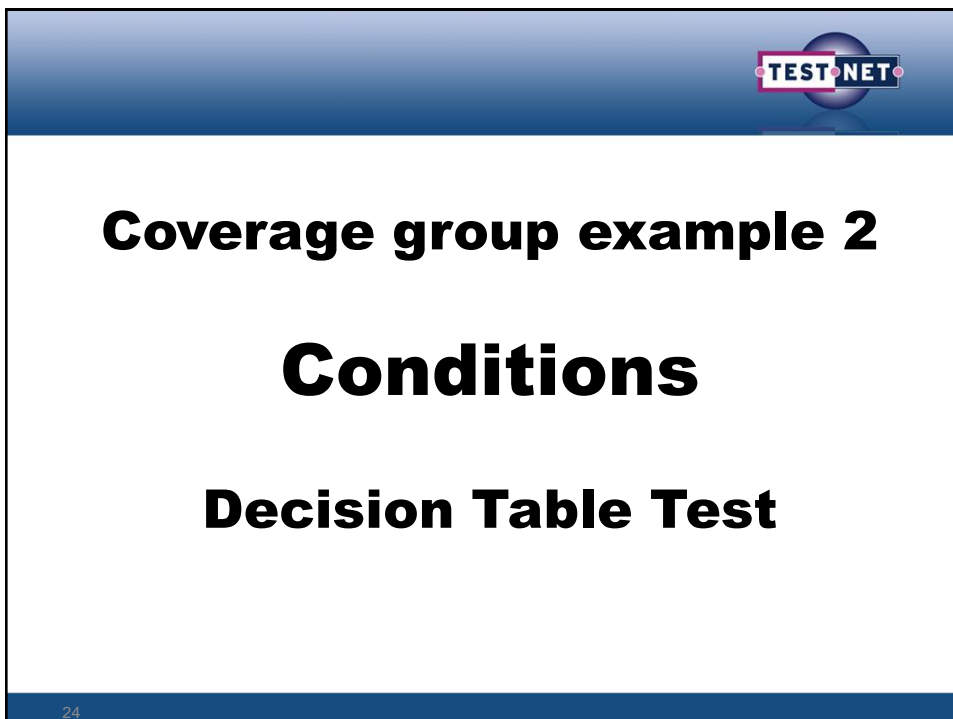
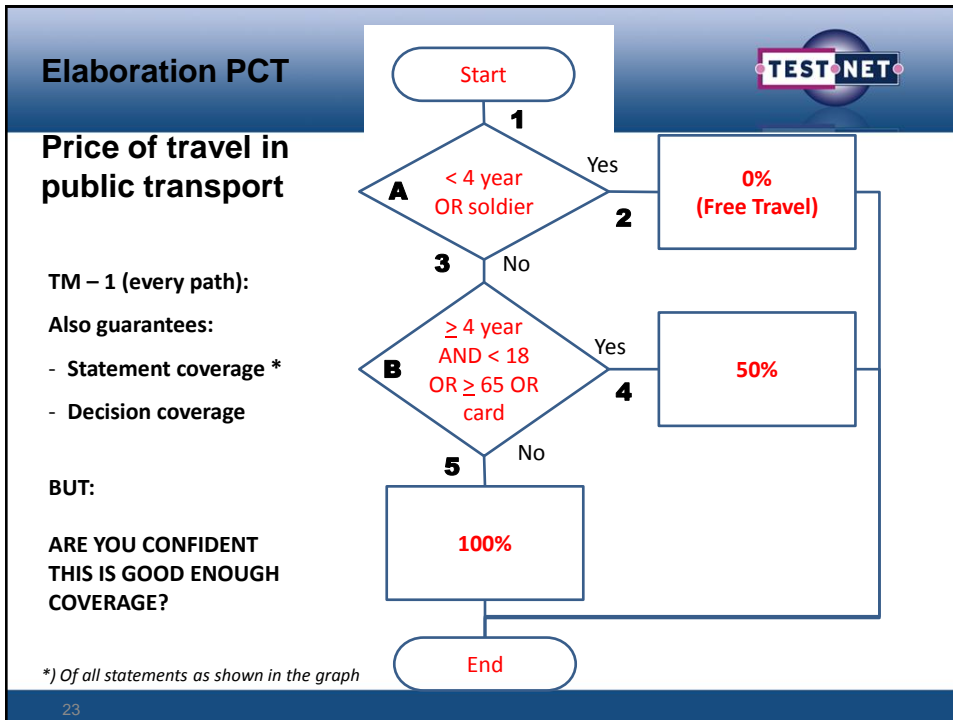
A: 1-2, 1-3
 B: 3-4, 3-5

Testcase a: 1-2
 Testcase b: 1-3 & 3-4
 Testcase c: 1-3 & 3-5

*(so, in this simple example,
 no difference, both 3 testcases)*



22



Decision table “collapsed for irrelevant card”



Condition																				
< 4 year	N	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y						
≥ 4 & < 18	N	N	N	N	N	N	Y	Y	Y	N	N	N	Y	Y						
> 18 & < 65	N	N	Y	Y	N	N	Y	N	Y	N	Y	N	Y	N	Y					
Soldier	N	Y	N	N	Y	N	Y	-	N	Y	-	-	-	-						
Card	-	-	N	Y	-	-	-	-	-	-	-	-	-	-						

Condition																				
0%		X			X		X		X	X										
50%		X			X		X													
100%			X																	
not possible								X		X	X	X								

13 test cases

Beware:
Only collapse if this doesn't increase your risk to an unacceptable level

Decision table “collapsed for irrelevant soldier”

Condition																				
< 4 year	N	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y						
≥ 4 & < 18	N	N	N	N	N	N	Y	Y	Y	N	N	Y	Y	Y						
> 18 & < 65	N	N	Y	Y	N	N	Y	N	Y	N	Y	N	Y	N	Y					
Soldier	N	Y	N	N	Y	N	Y	-	-	-	-	-	-	-						
Card	-	-	N	Y	-	-	-	-	-	-	-	-	-	-						

Condition																				
0%		X			X		X		X											
50%		X			X		X													
100%			X																	
not possible								X		X	X	X								

12 test cases

Remaining:
8 test cases with “normal” outcome
(compare these with results of other techniques)



Coverage group example 3

Conditions

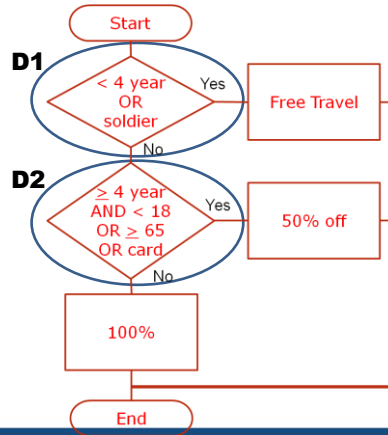
Elementary Comparison Test

ECT & MCDC



Steps in Elementary Comparison Test:

1. Identify the decision points
- 2a. Determine test situations per decision point, using MCDC
(MCDC = Modified Decision Coverage)
- 2b. Draw a graph to combine test situations
3. Create Logical Test Cases
4. Create Physical Test Cases
5. Create Test Script



29

MCDC with 2 conditions



D1 = < 4 year OR soldier	TRUE (1) = 0%		FALSE (0) = go to D2	
0 ← Neutral value	A	B	A	B
(A) < 4 year	D1.1	1 .0	D1.3	0 .0
(B) soldier	D1.2	0 1	— 0 — 0 —	

STEP 2:

D1.1, D1.2 and D1.3 are our Test Situations

30

MCDC with 4 conditions



D2 = ≥ 4 AND < 18 OR ≥ 65 OR CARD

1 0 0

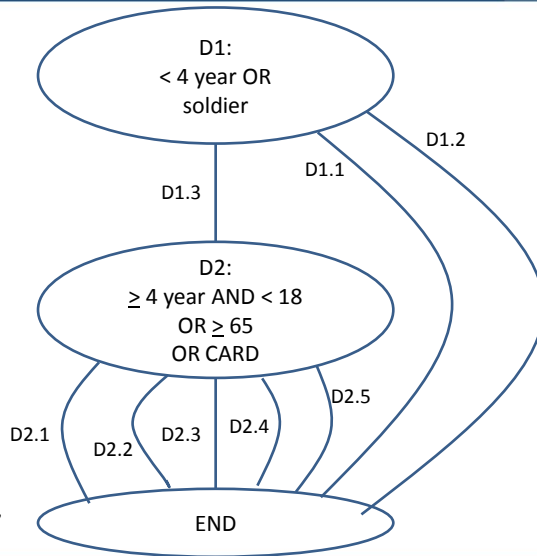
← Neutral values (AND = 1, OR = 0)

TRUE (1)
= 50 %

FALSE (0)
= 100 %


	TRUE (1) = 50 %				FALSE (0) = 100 %					
	A	B	C	D	A	B	C	D		
(A) ≥ 4	D2.1	1	.1	.0	.0	D2.4	0	.1	.0	.0
(B) < 18		1	1	0	0	D2.5	1	0	0	0
(C) ≥ 65	D2.2	1	0	1	0	1	0	0	0	0
(D) CARD	D2.3	1	0	0	1	1	0	0	0	0

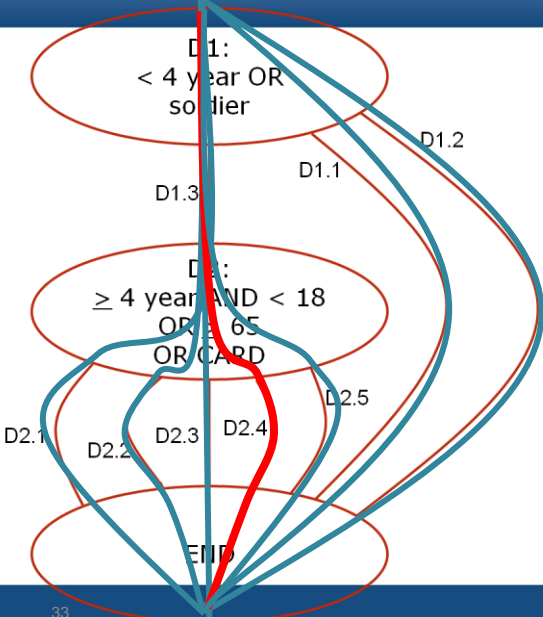
ECT & MCDC → Graph



STEP 2b:
Create Logical Test Cases,
Sub-step 1: create graph

ECT & MCDC → Testcases





D1.3 : ≥ 4 YEAR, NOT soldier
 D2.4 : < 4 YEAR, < 18 YEAR,
 < 65 YEAR, NO card

This is a →


GOOD Logical Testcase
 IMPOSSIBLE Physical Testcase

So we have 7 Logical Testcases
 And 6 Physical Testcases

STEP 3:
 Create Logical Test Cases,
 Sub-step 2: combine test situations

STEP 4 and 5:
 Create Physical Test Cases
 and Test Scripts

33



Coverage group example 4

Data

Data Combination Test

34

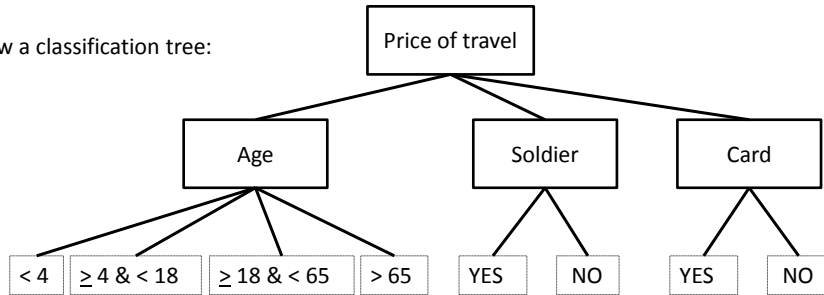
Data Combination Test



3 data elements:

- Age → 4 values (< 4, ≥ 4 & < 18, ≥ 18 & < 65, ≥ 65)
- Soldier → 2 values (YES & NO)
- Card → 2 values (YES & NO)

Draw a classification tree:

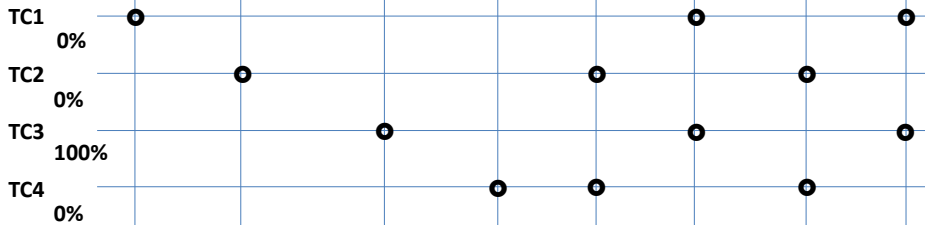
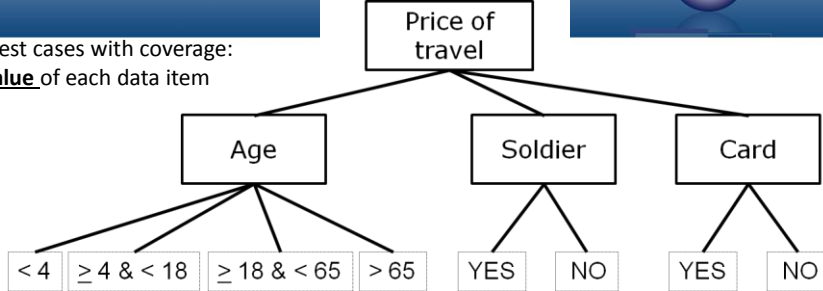


35

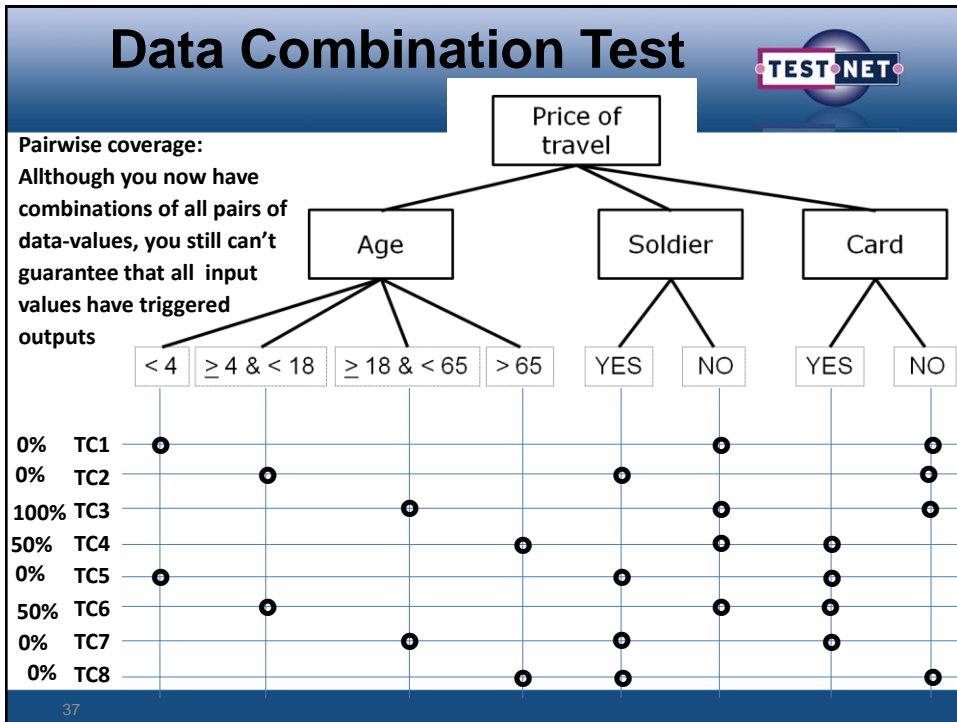
Data Combination Test




Create test cases with coverage:
Every value of each data item






36



Vervolg






En dan nu:
Op naar ongebonden testen

- Professional staff
 - Trained
 - Certified
- (Early) involvement
 - Risk based
- Independent testing
- Test competence center
- Recognized as a function

We're on the map

Process
 Theory
 Schedule
 Manual
 Independent
 Discipline
 Structure
 Standards
 Strict



Loslaten



- Testontwerptechnieken zijn prima
- Maar niet als enige aanpak
- Gebruik je hele gereedschapskist
- Combineer !!



Ongebonden testen



- Gebaseerd op ervaring
- Error guessing
- Exploratory testing

Even definiëren:



Error guessing

Testers gaan, zonder het gebruik van gedocumenteerde testgevallen, het systeem ongestructureerd testen *

It is largely an intuitive and ad-hoc process **

Exploratory testing

Elke vorm van testen waarbij de tester zijn testontwerp maakt tijdens de testuitvoering.

De informatie die wordt verkregen tijdens het testen wordt gebruikt om nieuwe en betere testgevallen te ontwerpen. *

Source:

* TMap NEXT

** The art of software testing, Glenford J. Myers

Meer definities van E.T.:



"An approach to software testing that emphasizes the personal freedom and responsibility of each tester to continually optimize the value of his work by treating learning, test design and test execution as mutually supportive activities that run in parallel throughout the project."



"Simultaneously designing and executing tests to learn about the system, using your insights from the last experiment to inform the next."



Source:
Presentation of Huib Schoots at TestNet

E.T. in TMap NEXT



Er wordt wel beweerd dat Exploratory Testing slecht beschreven is in TMap NEXT.
Op zich is het best goed beschreven, maar het staat heel verspreid door het boek...

Pagina Onderwerp

192	ET binnen teststrategie
196	Definitie
236	Veranderende testbasis
258	Ontbrekende testbasis
297	Session based
319	Pairing
567	Testen van "uiterlijk"
675	Vergelijking Error Guessing en Exploratory Testing
677	Definitie en beschrijving

E.T. = Gestructureerd !



- Charter met Scope en Timebox
- Ontwerp een testgeval en voer het uit
- Gebruik een “orakel” voor de uitvoervoorspelling
- Log de resultaten, o.a. t.b.v. hertesten en bevindingen
- Debriefing

E.T. = teamwork



Werk in tweetallen:

- **Samen het beste volgende testgeval bedenken**
- **De één voert de test uit, de ander logt**
- **Samen het resultaat beoordelen**



Her- en regressietesten kan door één tester, op basis van log

Ongebonden testen



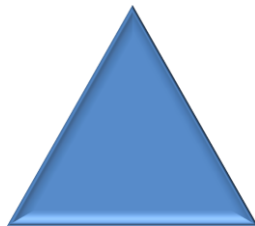
- Ontwerp een testgeval (met verwachte uitkomst !)
 - Voer het uit
 - Beoordeel het resultaat
 - Ontwerp het volgende testgeval
 - Enz.
-
- Voor het ontwerpen gebruik je waar nodig (risico / testdoel / enz.) een T.O.T.

Workshop 2: Charter



Ga in kleine groepen (2 à 4 mensen) de charter uitvoeren.

HOUD JE LOG BIJ !!!



De beroemde "Triangle-exercise" uit het boek van Glenford Myers

Charter:	
Opsteller charter	
Naam:	Rik Marselis
Functie:	Testmanager
Uitvoerders	
Materiedeskundige(n):	Jij
Tester(s):	Je TestNet-vakgenoot
Tijdsbestek	
Datum:	16 maart 2015
Aanvangstijd:	ergens op de avond
Tijdsduur:	25 minuten
Testobject	
Triangle programma van James & Jon Bach, (usb-stick of download van www.marselis.eu)	
Doelgroep: Leerlingen basisschool (die wel Engels kunnen lezen)	
Scope	
Functioneel gebied: Doet het programma wat het moet doen?	
Features to be tested:	
Functionaliteit: Invoer en uitvoer	
Usability	
Features not to be tested: Performance, Security, Portability	
! Let op ! Na afloop van de testsessie invullen:	
Conclusie:	
<input type="text"/>	
Advies:	
<input type="text"/>	

Nabespreking



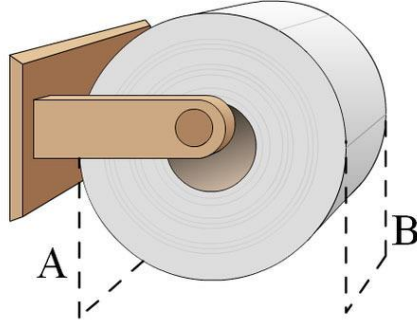
- Hoeveel testgevallen uitgevoerd?
- Hoeveel bevindingen?
- Hebben we alle 14 punten uit de lijst van Myers?
- Welke T.O.T. heb je gebruikt?



De beste aanpak?



THE GREAT

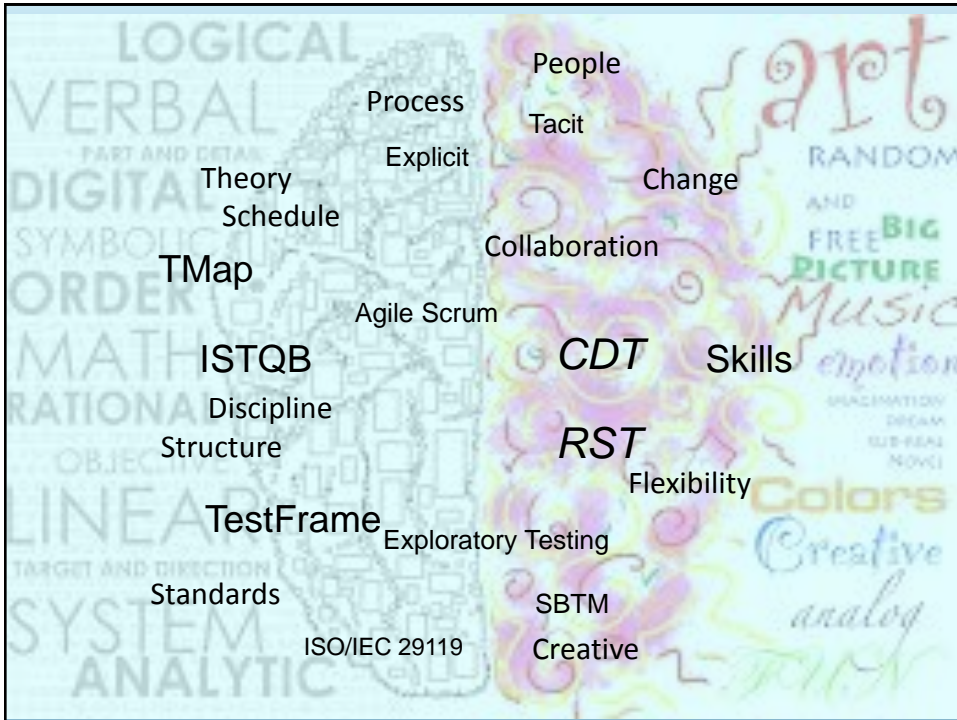


DEBATE

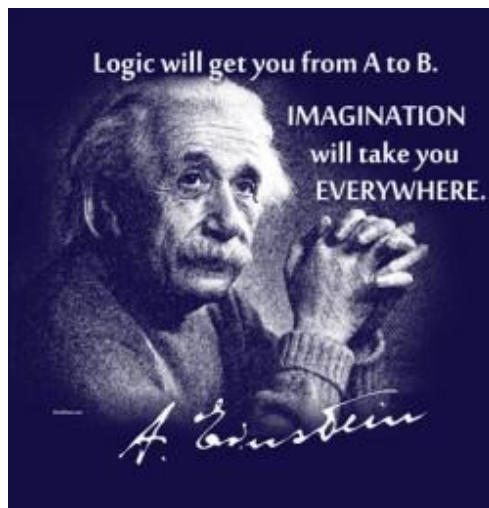
De beste aanpak?



**Een mix
van
ongebonden
en
gebonden
testen**



Afsluiting



Conclusion



- Lesson1: **choose smart**, based on the real problem you are facing, and save time by knowing what other coverage(s) you achieve as well.
- Lesson 2: be aware that the **obvious** technique **may not** be the best. (PCT is not very effective, DT-MCC is not very efficient)
- Experience based testing should be **combined** with coverage based testing:
 - When you for example come across some combined conditions, do a quick MCDC
 - When you see some boundary, do boundary value analysis
 - When you have a process, cover every path
- This way you will achieve **both effective and efficient** testing without an overload of activities



Hartelijk bedankt voor je
deelname aan deze workshop
