



Improve
QUALITY SERVICES

Workshop Testtechnieken & Heuristieken

September 2016, TestNet
Manon Penning & Huib Schoots



consult.



train.



practice.

Uw showmasters:

Manon Penning



**Trainer voor o.a.
ISTQB TA & TTA
Exploratory Testing**

Huib Schoots



**Trainer voor o.a.
Rapid Software Testing
Exploratory Testing**

Workshop Testtechnieken & Heuristieken

- Wat zijn testontwerptechnieken?
- Waarom belangrijk?
- Wat zijn beperkingen?
- Heuristieken
- Samenvatting

Oefening:

Hoeveel testcases?

Hoeveel testcases?

Case 1: Pairwise

Voor het berekenen van de jaarlijkse kosten voor een bankpas, gelden de volgende factoren:

1. Bank (ING, Rabobank)
2. Rekening (betaal, spaar)
3. Pas (normaal, met afbeelding)

Case 2: Grenswaarde

De voorwaarden voor het verstrekken van een lening zijn leeftijd gebonden:

1. Jonger dan 18: geen lening
2. 18 t/m 55: standaard lening
3. 55+: duurdere lening

Pairwise testen

	Bank	Rekening	Pas
1	ING	Betaal	Normaal
2	ING	Betaal	Afbeelding
3	ING	Spaar	Normaal
4	ING	Spaar	Afbeelding
5	Rabo	Betaal	Normaal
6	Rabo	Betaal	Afbeelding
7	Rabo	Spaar	Normaal
8	Rabo	Spaar	Afbeelding

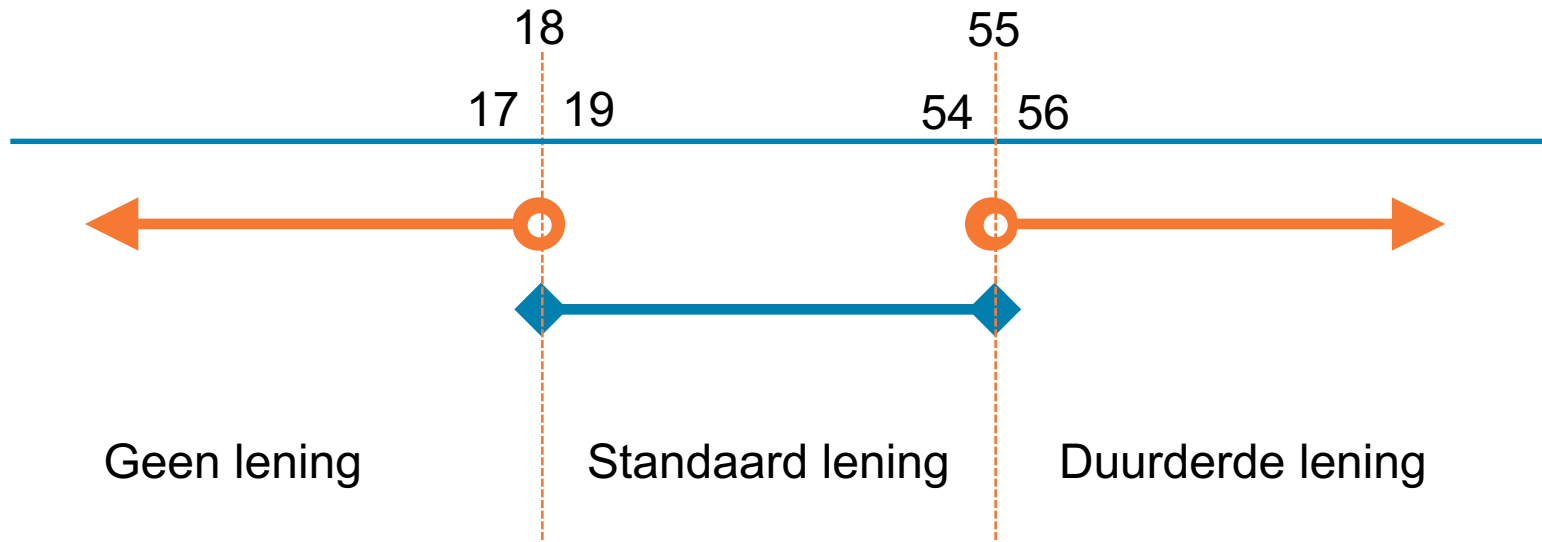
Alle mogelijke combinaties



	Bank	Rekening	Pas
1	ING	Betaal	Normaal
2	ING	Spaar	Afbeelding
3	Rabo	Betaal	Afbeelding
4	Rabo	Spaar	Normaal

Combinaties met pairwise

Grenswaardeanalyse



Normale grenswaardeanalyse → 6 testcases: 17, 18, 19 en 54, 55, 56

Lichte grenswaardeanalyse → 4 testcases: 17, 18 en 55, 56

Debrief: pairwise

Pairwise (combinaties) testen houdt geen rekening met:

- Welke variabelen je samen moeten testen en waarom
- Welke waarden van deze variabelen die je moet testen en waarom (risico, belang)
- Hoe moet je omgaan met relaties tussen de variabelen
- Welke risico's je moet letten bij het testen van deze variabelen bij elkaar
- Hoe je bepaalt of het programma echt werkt

Debrief: grenswaarde

Grenswaarde testen houdt geen rekening met:

- Welke andere grenzen zijn er?
- Welke andere risico's zijn er naast de grenzen?
- Hoe je bepaalt of het programma echt werkt?

Welk model gebruik je? (Risico's, mogelijke bugs)

Testers zijn erg gehecht aan grenzen, maar er is meer!

Een paar vragen...

- Wat zijn testontwerptechnieken?
- Testontwerptechnieken: wie gebruikt ze? Welke?
- Waarom? Waarom niet?

Testontwerptechniek

- Een testontwerptechniek is een gestandaardiseerde manier om vanuit uitgangsinformatie testgevallen af te leiden die een bepaalde dekking bereiken

(TMap Next)

- A test design technique is a procedure used to derive and/or select test cases

(ISTQB Glossary)

Belang van testontwerptechnieken?

- Geeft onderbouwde invulling van de teststrategie, geeft onderbouwing voor dekking op de juiste plaats
- Objectiviteit: garandeert een bepaalde dekking en maakt testproces onafhankelijk van de persoon die de testgevallen maakt en uitvoert
- Verhoogt de kans op het vinden van bugs
- Reproduceerbaarheid
- Lange termijn onderhoudbaarheid en overdraagbaarheid
- Testproces is beter planbaar en beheersbaar

Bronnen: TMap Next & The Testing Practitioner

Maar is dat écht belangrijk?

Belang van testontwerptechnieken?

- Geeft onderbouwde invulling van de teststrategie, geeft onderbouwing voor dekking op de juiste plaats → Onderbouwd?
- Objectiviteit: garandeert een bepaalde dekking en maakt testproces onafhankelijk van de persoon die de testgevallen maakt en uitvoert → Is dat handig?
- Verhoogt de kans op het vinden van bugs → Van een bepaald type
- Reproduceerbaarheid → Is het dat altijd waard?
- Lange termijn onderhoudbaarheid en overdraagbaarheid → Niet altijd
- Testproces is beter planbaar en beheersbaar → Hoe dan?

Bronnen: TMap Next & The Testing Practitioner

Testontwerptechnieken?

- Risico analyse?
- Modelleren?
- Opstellen teststrategie?
- Testpunt analyse?
- Kritisch denken?
- ...

Oefening:

Beperkingen van testontwerptechnieken

Beperkingen testontwerptechnieken

Bespreek in groepen:

1. Worden testontwerptechnieken breed toegepast?
Zo ja: waarom? Zo nee: waarom niet?
2. Wat zijn de beperkingen van testontwerptechnieken?

Kun je drie dingen noemen waarom het toepassen van een testontwerptechniek fout kan gaan?

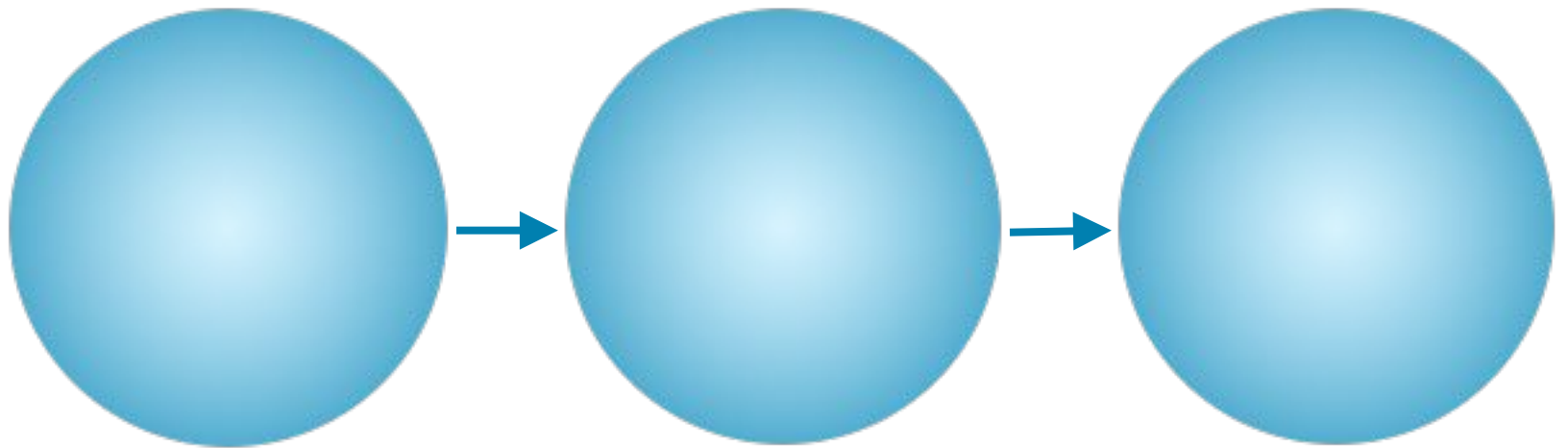
Debrief:

Beperkingen van testontwerptechnieken

Valkuilen

- Het model zien als waarheid
 - Denk nog even terug aan de grenswaarde oefening
- Beperkingen van de techniek niet kennen
 - Denk nog even terug aan de pairwise oefening
- Testontwerptechnieken beperken “onbewust” het testen
- Het zoeken naar één soort bug
- Leercurves worden uit elkaar getrokken

Strategie – uitvoering – rapportage

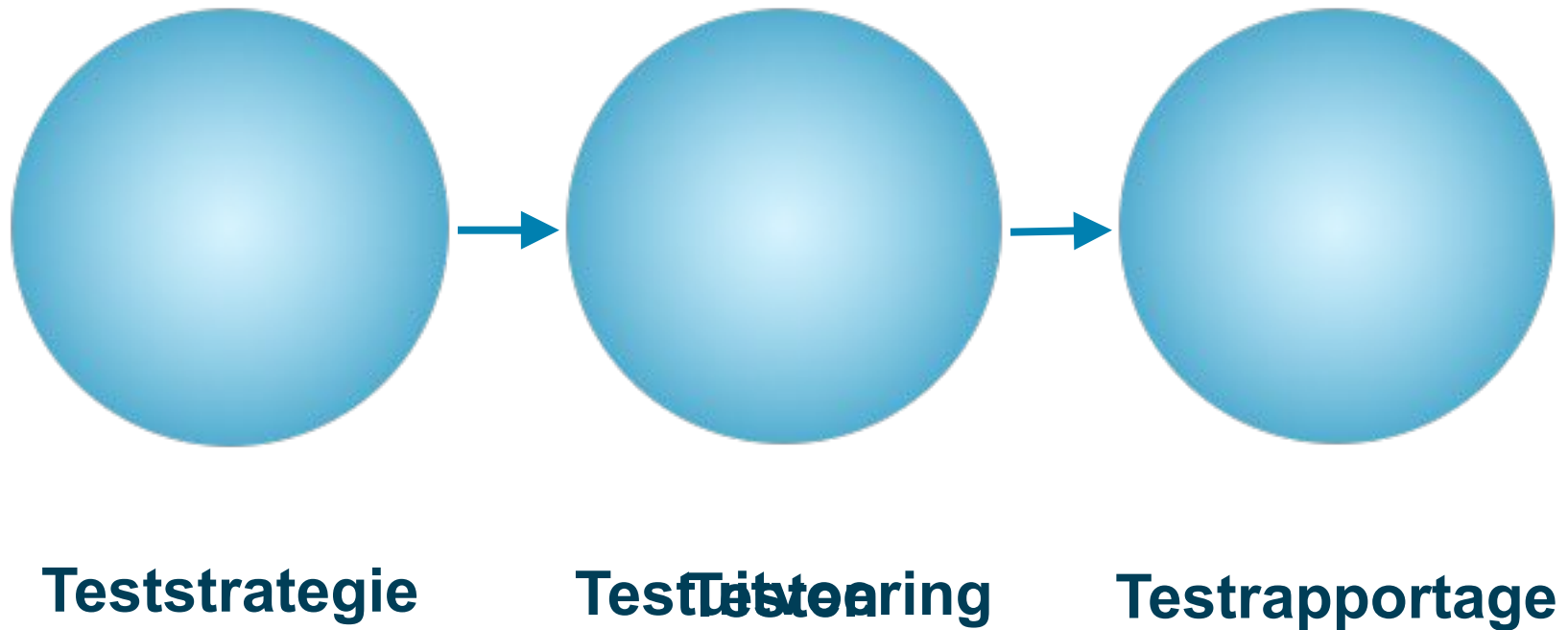


Teststrategie

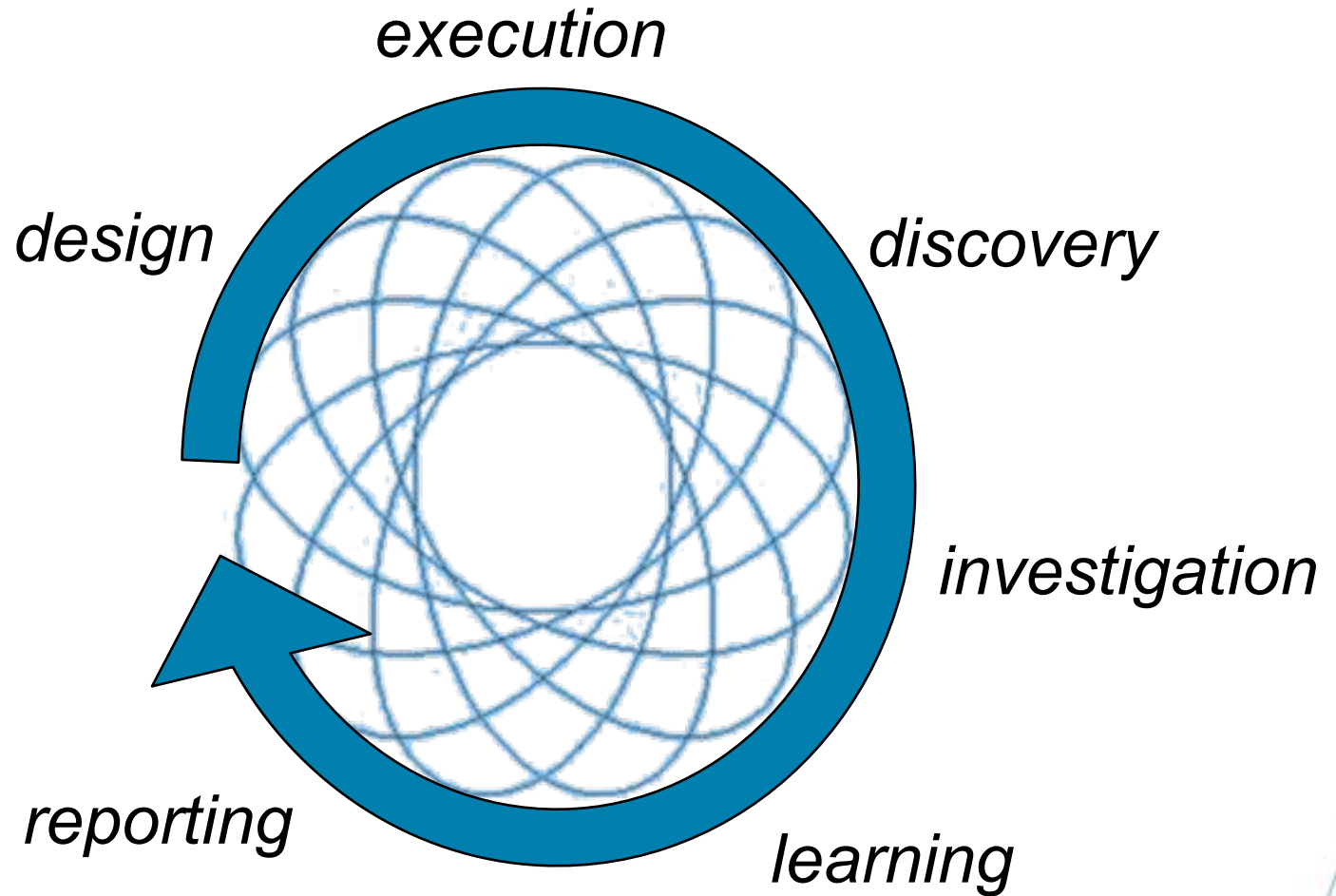
Testuitvoering

Testrapportage

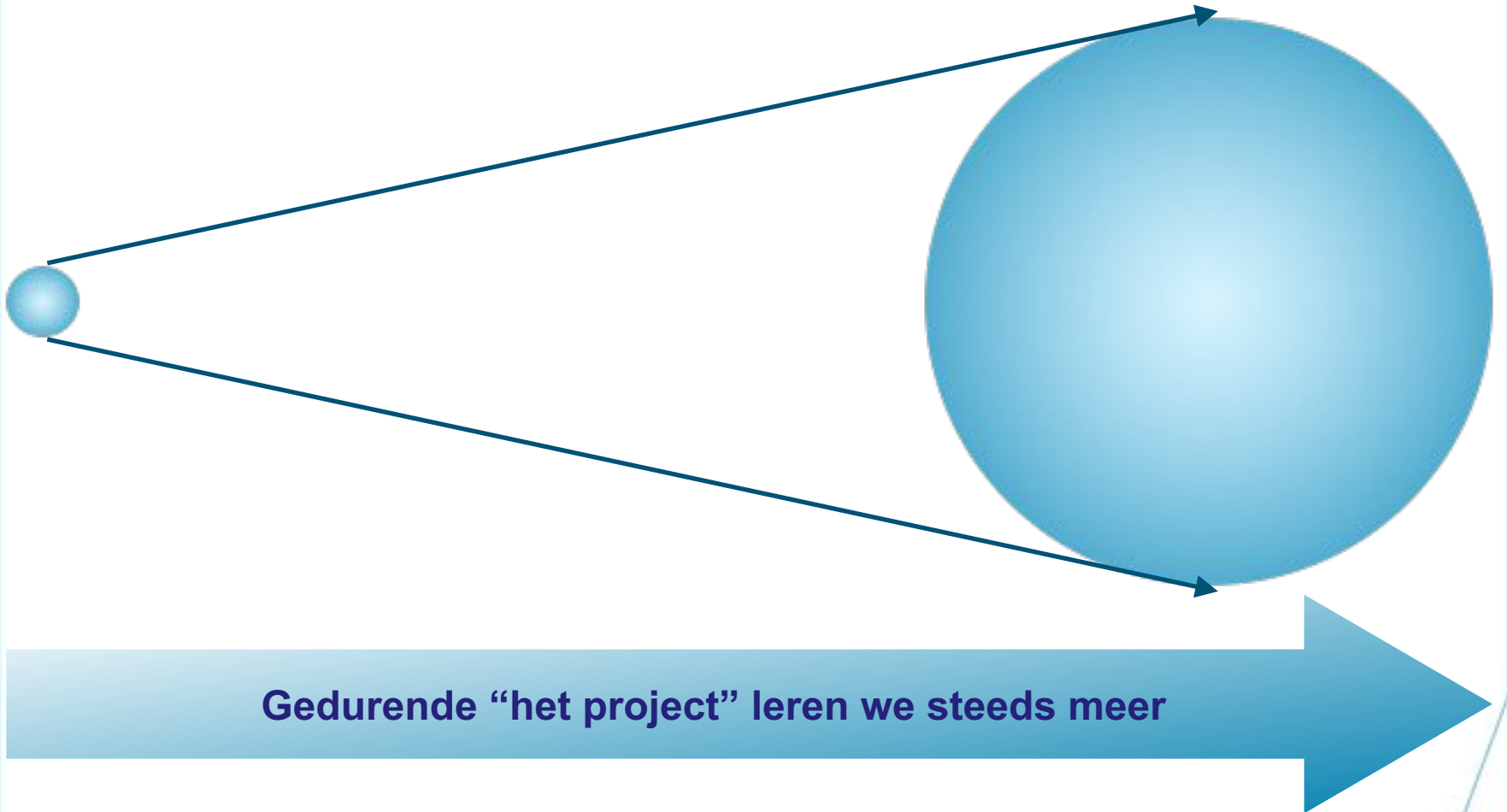
Strategie – uitvoering – rapportage



Zoeken naar waarde en risico



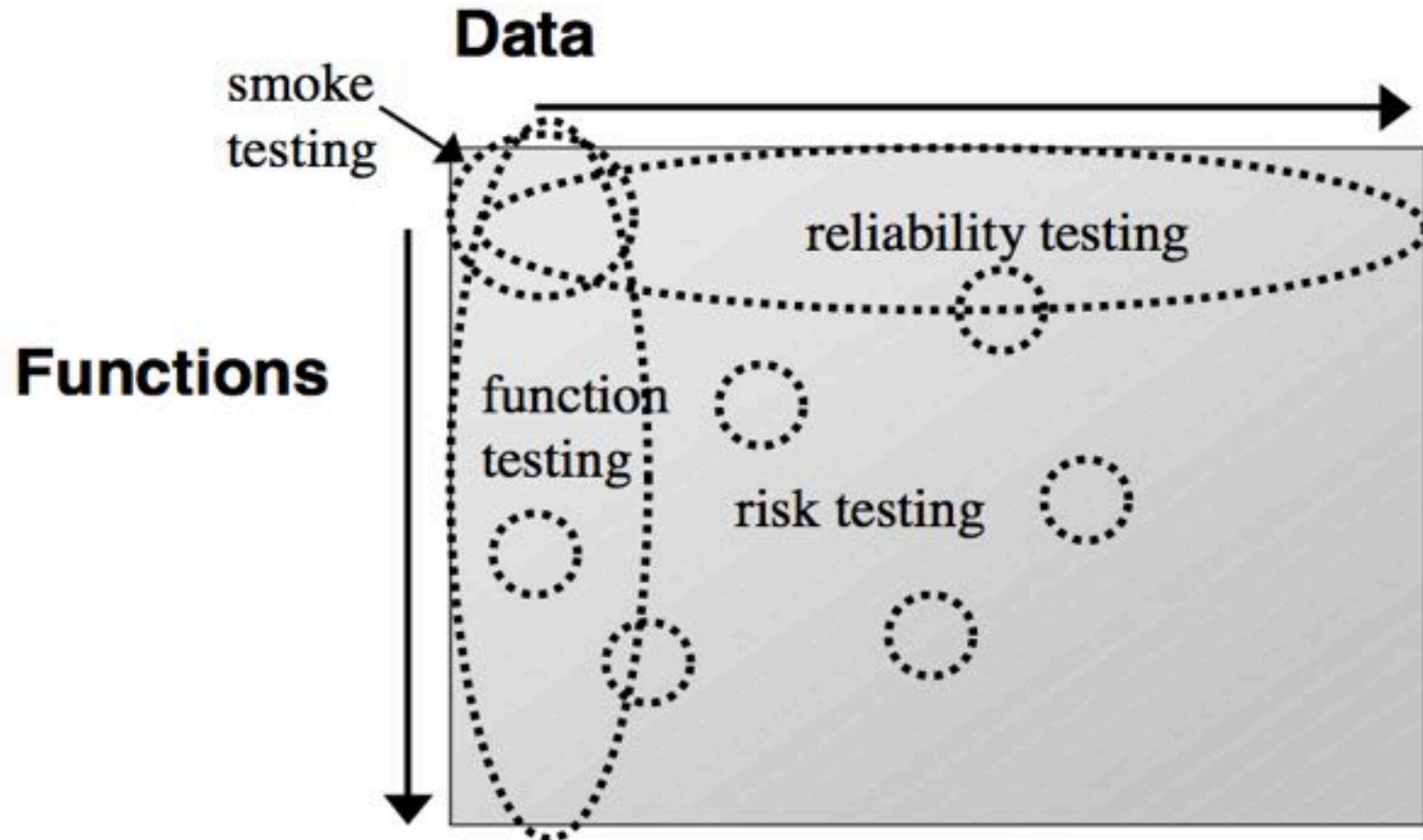
Een groeiende test strategie



Oplossingen

- Weet wat je doet, ken het trucje achter de techniek
- Diversifieer, diverse gedeeltelijke maatregelen

Diversifieer!



<http://www.satisfice.com/presentations/etta.pdf>

Oplossingen

- Weet wat je doet, ken het trucje achter de techniek
- Diversifieer, diverse gedeeltelijke maatregelen
- Exploratory Testen
 - Dingen buiten business logica (kleuren, knippen)
 - Verkleint het risico dat je het risico niet kent
- Test “alles”, automatiseer of snel en goedkoop

Snel en goedkoop testen

Gebruik naast gedegen, zorgvuldige testen ook snelle, goedkope testen

Een snelle test is een goedkope test
die enige waarde heeft,
maar weinig voorbereiding, kennis of tijd vergt
om uit te voeren.

Snel en goedkoop testen

Gebruik naast gedegen, zorgvuldige testen ook snelle, goedkope testen

- “Happy” paden
- Tour het product
 - Voorbeeld data
 - Variabelen
 - Bestanden
 - Complexiteit
 - Menu's & Windows
 - Toetsenbord en muis
- Input attack
- “Click Frenzy”
- “Schoen test”
- Blink testen
- Error Message Exploit
- Onderbrekingen
- Ondermijning
- Aanpassingen
- Dog Piling / cache stampede
- Continu gebruik
- Functie interactie
- Klik op Help
- Resources uithongeren
- Meerdere Instanties
- Rare configuraties
- Goedkope tools

Oplossingen

- Weet wat je doet, ken het trucje achter de techniek
- Diversifieer, diverse gedeeltelijke maatregelen
- Exploratory Testen
 - Dingen buiten business logica (kleuren, knippen)
 - Verkleint het risico dat je het risico niet kent
- Test “alles”, automatiseer of snel en goedkoop
- Heuristieken (testontwerptechnieken zijn heuristieken...)

Heuristieken



Definitie heuristiek

Een feilbare methode voor het oplossen van een probleem

- Geven een bruikbare structuur om problemen op te lossen
- Géén regel: kan werken, maar kan ook mislukken

De engineering methode is het gebruik van heuristieken om de beste verandering te veroorzaken in een slecht begrepen situatie binnen de beschikbare middelen.

Billy Vaughan Koen - Discussion of the Method

Zie “Heuristics for Understanding Heuristics”
www.developsense.com/blog/2012/04/heuristics-for-understanding-heuristics/

Voorbeeld van heuristiek: Apollo 11



Heuristieken uit RST

- Focus & defocus
- Creep & Leap (<http://www.satisfice.com/blog/archives/467>)
- Steeplechase (<http://www.satisfice.com/blog/archives/467>)
- Dead Bee (<http://www.satisfice.com/blog/archives/39>)
- Plunge in & quit

Testtechnieken in RST

“Ways to test...”?

The General Test Techniques

FDSFSCURA

- Function testing
- Domain testing
- Stress testing
- Flow testing
- Scenario testing
- Claims testing
- User testing
- Risk testing
- Automatic checking

Oefening:

NS Reisplanner

NS Reisplanner

Opdracht: hoe zou je dit testen?

Reisplanner

Van

Naar

☐ Via Station

Reisdatum dd-mm-jjjj

Tijd : ☒ Vertrek ☐ Aankomst

☐ Terugreis plannen

→

NS Reisplanner

Opdracht: hoe zou je dit testen?

- Welke technieken of aanpak?
- Waarom?
- Welke beperkingen?



The screenshot shows the NS Reisplanner interface. It has a yellow background and a blue border. The title 'Reisplanner' is at the top left. Below it, there are input fields for 'Van' and 'Naar', each with a 'Station' dropdown menu and a text input field. A radio button labeled 'Via Station' is below these. The 'Reisdatum' field shows '08 09 2016' and a date picker icon. The 'Tijd' field shows '19 00' and two radio buttons: 'Vertrek' (selected) and 'Aankomst'. A radio button labeled 'Terugreis plannen' is below. At the bottom right, there is a blue button with the text 'Geef reis en prijs' and a right arrow icon.

Debrief:

NS Reisplanner

Samengevat

- Weet wat je doet: wat, waarom, wanneer
- Ben je bewust van de beperkingen
- Diversifieer
- Gebruik ook eens andere heuristieken

Vragen & discussie



Manon Penning & Huib Schoots



manon.penning@improveqs.nl

huib.schoots@improveqs.nl



Improve Quality Services

Prof. Dr. Dorgelolaan 30

5613 AM Eindhoven

Tel: 040 – 2021803

<http://www.improveqs.nl/>

Referenties

- Pairwise Testing: A Best Practice That Isn't - <http://www.satisfice.com/articles/pairwise.pdf>
- BBST Blackbox testing - <http://www.testingeducation.org/BBST/>
- Rapid Software Testing - <https://www.improveqs.nl/training/rapid-software-testing/>
- Developing the Right Test Documentation - http://www.kaner.com/pdfs/test_docs_pnsqc.pdf
- Explaining Testing to Anybody - <http://www.satisfice.com/presentations/etta.pdf>
- To Repeat Tests or Not to Repeat - <http://www.satisfice.com/blog/archives/24>
- How to come up with test ideas - <http://erik.brickarp.se/2016/08/how-to-come-up-with-test-ideas.html>
- When Do We Stop a Test? - <http://www.developsense.com/blog/2009/09/when-do-we-stop-test/>
- Heuristics for Understanding Heuristics - <http://www.developsense.com/blog/2012/04/heuristics-for-understanding-heuristics/>
- Testing Mnemonics - http://www.qualityperspectives.ca/resources_mnemonics.html
- Test Heuristics Cheat Sheet - <http://testobsessed.com/wp-content/uploads/2011/04/testheuristicscheatsheetv1.pdf>

Recommended books:

- Lessons learned in SW Testing – Kaner, Bach & Pettichord
- Test Design, a BBST workbook – Kaner, Fiedler
- The little black book on Test Design – Rikard Edgren
(<http://www.thetesteye.com/papers/TheLittleBlackBookOnTestDesign.pdf>)
- How to Break Software: A Practical Guide to Testing – James Whittaker
- A practitioner's guide to software test design – Lee Copeland
- Software Testing – A craftsman's approach – Paul C. Jorgensen
- The Craft of Software Testing – Brian Marick