

## TESTSTRATEGIE MET BEHULP VAN HEURISTIEKEN: EEN PRAKTISCHE AANPAK VOOR IEDEREEN!

Door Huib Schoots • [hsc@improveqs.nl](mailto:hsc@improveqs.nl) [@huibschoots](https://twitter.com/huibschoots)



*Een gedegen teststrategie is de basis voor ieder testtraject. Het geeft antwoord op de vragen: waarom testen we en hoe testen we? Een teststrategie helpt de tester de juiste dingen te doen, geeft testen structuur en geeft inzicht in de testdekking.*

### Problemen in testen

In mijn dagelijkse praktijk heb ik veel projecten gedaan, aangestuurd, geadviseerd en van dichtbij mogen bekijken. Uiteraard heeft vooral het testgedeelte mijn voornaamste aandacht. Wat me opvalt, is dat veel testtrajecten niet of nauwelijks met een teststrategie werken, ondanks dat er een hoofdstuk teststrategie is opgenomen in het testplan. Ik zie een aantal problemen in de praktijk met testen die gerelateerd zijn aan teststrategie:

- Testen wordt gezien als een aantal opeenvolgende fasen of activiteiten.
- Testers zien de teststrategie als een document of een fase in plaats van voortschrijdend inzicht.
- De vertaling van een PRA naar een concrete teststrategie is lastig.
- De meeste teststrategieën zijn vaag en gebruiken relatieve inspanning. Ze geven geen inzicht in wat testers moeten doen.
- Testers hebben geen idee wat ze nu wel en niet geraakt hebben. Ik heb het dan over het 'echte product' en niet over de specificatie of requirements. Hoe maak je dat inzichtelijk?
- Het toepassen van een testtechniek is een recept geworden. Altijd toepassen van classification tree, ook al laat de situatie dat niet toe: bijvoorbeeld gedrag van een product in een classification tree stoppen.
- Er is geen (duidelijk) verband tussen de informatiebehoefte, de testmissie, de teststrategie en de testen die uitgevoerd worden.

Teststrategieën verdwijnen helaas nog veel te vaak in een la. Dit artikel geeft handvatten om een heldere en bruikbare teststrategie te maken, die wel gebruikt zal worden! Maar laten we eens een paar stappen terug doen. Wat is testen eigenlijk? En waarom testen we?

Sound strategy starts with  
having the right goal  
(Michael Porter)

### Wat is testen? En wat is de missie van testen?

Testen verzamelt informatie en geeft inzicht in de status van een product om beslissingen te informeren (Jerry Weinberg, zie ref 1). Een algemene doel van testen is om informatie te verzamelen over dingen die belangrijk zijn. Door met onze stakeholders te praten en hen vragen te stellen, komen we er achter wat zij willen weten en wat zij belangrijk vinden. Testen kan daardoor veel verschillende missies [ref 2] hebben:

- belangrijke bugs vinden;
- de kwaliteit van het product beoordelen;
- managers inzicht geven in de voortgang van het project;
- managers informatie geven voor vrijgavebeslissingen;
- informatie geven voor het voorspellen en beheersen van de kosten van productondersteuning;
- controle van de implementatie van requirements (conformiteit specificaties); ➔

- helpen bij het verbeteren van de productkwaliteit;
- ... enzovoorts.

Testen is dus informatie verzamelen voor verschillende missies. De meeste testers zijn niet gewend om op dit niveau over testmissie na te denken en hanteren gewoon steeds dezelfde impliciete testmissie. Daarom een voorbeeld van een wat minder gebruikelijke testmissie: 'het beoordelen van een product van een bedrijf dat misschien door jouw bedrijf overgenomen wordt.' Hierbij is het eenvoudig voor te stellen dat er anders getest moet worden.

## Teststrategie

Daarna is het belangrijk om de juiste methode te bepalen om de gevraagde informatie te vinden. De testmissies hebben invloed op de teststrategie omdat verschillende testmissies een andere aanpak vragen (lees: een andere teststrategie). Het doel van de teststrategie is om het testen te sturen om de missies te bereiken. Het bestaat uit richtlijnen die beschrijven wat te testen en hoe te testen. De strategie informeert in eerste instantie de stakeholders en het testteam. Later gebruiken we de strategie, of beter gezegd bouwen we de strategie uit zodat we de testen kunnen uitvoeren.

There is nothing so useless as doing efficiently that which should not be done at all  
(Peter Drucker)

Wat maakt een teststrategie tot een goede teststrategie? Een goede teststrategie is onder andere: [ref 3]

- Context-gerelateerd. Iedere situatie is anders. Ieder project is anders. Een teststrategie moet op maat gemaakt worden zodat het perfect past in de context.
- Specifiek. Niemand heeft iets aan vage statements. Hoe specifieker de teststrategie, hoe beter de stakeholders het begrijpen en hoe makkelijker het is om de testen te ontwerpen en ze uit te voeren.
- Risico-gebaseerd. Het belangrijkste risico dient als eerste getest te worden en verdient in veel gevallen de meeste testinspanning.
- Afwisselend & divers. Iedere methode of techniek is gericht op het vinden van andere informatie en vind ook andere bugs.
- Praktisch. Testen moeten uit te voeren zijn en bij voorkeur zo makkelijk mogelijk.
- Gedragen. Het is belangrijk dat de stakeholder erachter staan.
- Gerechtvaardigd. Iedere test die je uitvoert, moet bijdragen aan het doel en waarde toevoegen. Ook het kostenaspect speelt hiermee: is het de moeite waard?
- Flexibel. Een teststrategie verandert voortdurend door nieuwe inzichten, gevonden informatie en veranderende prioriteiten. Bovendien weten we aan het begin van het project nog lang niet alles en zal de nieuwe informatie ook invloed hebben op onze strategie.

## Teststrategie met heuristics

Oké, maar hoe maak je dan een goede teststrategie? Een teststrategie zou een leidraad moeten zijn voor de tester om testen uit te voeren die voldoen aan de missies. Maar het wordt ook gebruikt om de testinspanning te bepalen en de testplanning te maken. Om een goede teststrategie te maken moeten we aan ontzettend veel dingen denken. Ik gebruik het Heuristic Test Strategy Model van James Bach en Michael Bolton om me te helpen. Het doel van dit model is om me te helpen aan alle belangrijke aspecten te denken als ik test. →

The essence of strategy is choosing what not to do  
(Michael Porter)

### Wat zijn heuristics?

Heuristieken zijn feilbare methoden voor het oplossen van een probleem of het nemen van een beslissing. Het zijn snelkoppelingen om complexe problemen op te lossen of kortweg vuistregels, ezelsbruggetjes of geheugensteuntjes. Ze worden gebruikt om binnen een redelijke tijd voor moeilijke problemen haalbare oplossingen te bepalen die goed genoeg zijn. Op de Engelstalige Wikipedia vond ik deze definitie [ref 4]: 'Heuristiek zijn ervaring-gebaseerde technieken voor het oplossen van problemen, om een oplossing te leren en ontdekken die niet gegarandeerd optimaal is. Waar een uitputtende zoektocht onpraktisch is, worden heuristische methoden gebruikt om het proces te versnellen bij het vinden van een bevredigende oplossing via mentale snelkoppelingen die de cognitieve belasting bij het nemen van een beslissing vergemakkelijken.'

Wil je meer weten over heuristics? Lees dan deze blog posts van Michael Bolton [ref 5] en James Bach [ref 6]:

- 5 - <http://www.developsense.com/blog/2012/04/heuristics-for-understanding-heuristics/>
- 6 - <http://www.satisfice.com/blog/archives/462>

### Heuristic Test Strategy Model

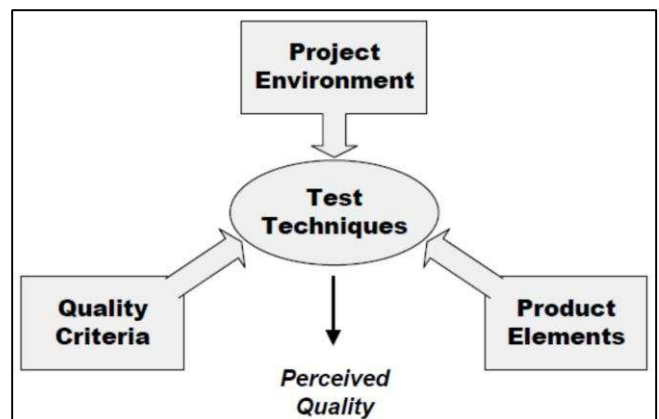
Het heuristic test strategy model (HTSM, [ref 7]) is ontwikkeld door James Bach en bestaat uit vier lijsten met heuristieken die helpen bij het opstellen van een gedegen teststrategie.

**Projectfactoren:** context-factoren die van cruciaal belang zijn om te kunnen beslissen welke specifieke testen uitgevoerd moeten worden.

- Missie: Je doel in dit project, zoals afgesproken met je klanten.
- Informatie: informatie over het product of project die nodig is voor het testen.
- (Relatie met) Developers: hoe werk je samen met de programmeurs en de andere teamleden?
- Testteam: wie zal de testen uitvoeren of ondersteunen?
- Apparatuur en tooling: hardware, software of documenten die nodig zijn om het testen uit te voeren.
- Planning: de volgorde, duur en synchronisatie van project gebeurtenissen.
- Testitems: het te testen product.
- Deliverables: de waarneembare producten van het testproject.

**Productelementen:** een product bestaat uit meerdere elementen of producteigenschappen. De verschillende gezichtspunten moeten helpen de juiste testvragen te formuleren.

- Structuur: verschillende (onder)delen van het fysieke product.
- Functionaliteit: de functies, alles wat het product doet.
- Data: alles wat wordt verwerkt.
- Interfaces: elk kanaal waarmee het product interacteert.
- Platform: alles waar het product van afhankelijk is.
- Operaties: hoe het product gebruikt wordt.
- Tijd: elke relatie van het product met tijd.



**Kwaliteitsattributen:** bepaalde eisen die aangeven wat of hoe het product zou moeten zijn. Elk van de items op deze lijst kan worden gezien als een potentieel risicogebied. →

- Geschiktheid: kan het de vereiste functies uitvoeren?
- Betrouwbaarheid: werkt het goed en is het bestand tegen fouten?
- Bruikbaarheid: hoe gemakkelijk is het te gebruiken?
- Charisma: hoe aantrekkelijk is het product?
- Veiligheid: hoe goed is het product beveiligd?
- Schaalbaarheid: hoe goed kan het product uitgebreid of ingekrompen worden?
- Compatibiliteit: hoe goed werkt het met externe componenten en configuraties?
- Prestaties: hoe snel is het?
- Installeerbaarheid: hoe gemakkelijk kan worden geïnstalleerd?
- Ontwikkeling: hoe goed kunnen we het product maken, aanpassen en testen?

Software Quality Characteristics [ref 8] is een uitgebreidere lijst. Deze is ook in het Nederlands vertaald [ref 9] en is online te vinden. De ISO25010 (voorheen ISO9126) lijst met kwaliteitskenmerken kan ook als alternatief gebruikt worden.

**Algemene testtechnieken:** een 'algemene techniek' is een techniek die eenvoudig en universeel genoeg is om te gebruiken in zeer verschillende omstandigheden.

- Functietesten: test wat het kan doen.
- Domeintesten: deel de gegevens op in logische verzamelingen.
- Stresstesten: overweldig het product.
- Flowtesten: doe de ene handeling na de andere.
- Scenariotesten: test een meeslepend scenario.
- Claimtesten: controleer elke bewering over het product.
- Gebruikerstesten: betrek de gebruikers.
- Risicotesten: stel je een probleem voor en ga ernaar op zoek.
- Automatische controle: check een miljoen verschillende feiten.

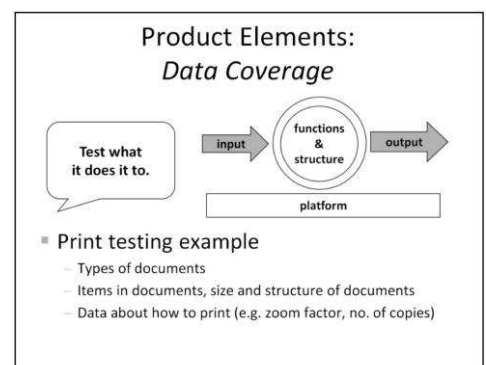
Vele specifieke technieken (zoals de technieken in TMap Next) zijn gebaseerd op één of meer van deze negen. Zo is de Data Combinatie Test een vorm van domeintesten, de Elementaire Vergelijkingen Test een vorm van functietesten en Error Guessing een risicotest.

Bekijk deze korte video waarin Michael Bolton het gebruik van het HTSM uitlegt.

- <https://www.youtube.com/watch?v=SNHWCD2qg3U>

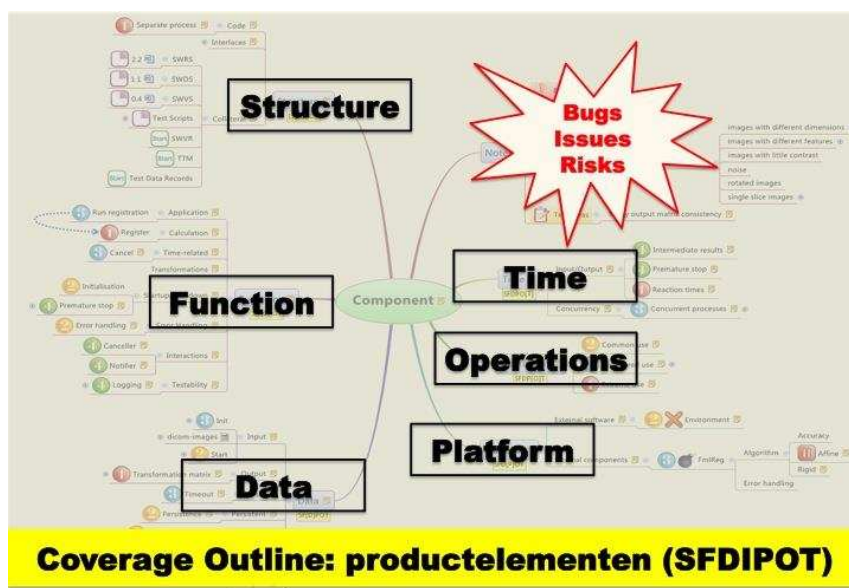
## Hoe gebruik je het HTSM?

De projectfactoren helpen je bij een analyse van de context. Het Heuristic Test Strategy Model bevat vervolgens zeven verschillende perspectieven op een product zoals structuur, functionaliteit, data (figuur rechts), interfaces, platform, operaties en tijd. Het gaat hier om een coverage model dat uitgebreid is beschreven in Rapid Software Testing [ref 10]. ➔





Er zijn vele manieren waarop je de informatie over productelementen kan modelleren. Onderstaand is een voorbeeld van een mind map van Ruud Cox. Andere vormen zijn ook mogelijk, het gaat er uiteindelijk om dat het onderliggende product goed gemodelleerd wordt.



Naast productelementen maken de kwaliteitsattributen inzichtelijk welke aspecten van het product getest moeten worden.

## Testtechnieken

Met dit gevulde model in de hand kan een tester dan bepalen welke testtechnieken het meest geschikt zijn. Om de dekking en diepgang inzichtelijk te maken kan hiermee ook aangegeven worden welke delen van het product en welke kwaliteitsattributen (of aspecten van het product) daarmee geraakt worden en welke niet. De keuze voor een bepaalde testtechniek wordt ingegeven door de projectfactoren te beschouwen en vervolgens per productelement, per kwaliteitsattribuut te bedenken welke testtechniek toegepast kan worden.

## Diversifieer

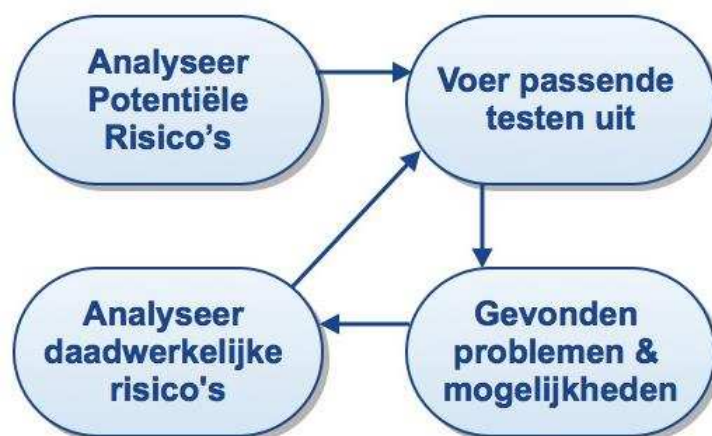
Er is niet één techniek die alle bugs vindt. Verschillende technieken vinden verschillende bugs. Bovendien is het zo dat geen enkele techniek perfect toegepast kan worden. Gebruik daarom veel verschillende invalshoeken, benaderingen en technieken. Zelfs al wordt geen enkele volledig uitgevoerd. Het is beter om tien technieken half toe te passen, dan één helemaal perfect [ref 11]. Datzelfde geldt natuurlijk ook voor testideeën: het is in de meeste gevallen beter om veel testideeën te hebben die veel verschillende testen kort en snel doen, dan één zeer uitgebreid en bijna perfect uitgewerkt idee.

## Dekking en diepgang

Hoe hoger het risico, hoe belangrijker de uit te voeren test is. Dat hoeft niet per se te betekenen dat de testen ook meer diepgang hoeven te hebben of dat er een formele(re) testtechniek wordt ingezet. Onderstaande tabel is een voorbeeld hoe een bepaalde testspecificatietechniek direct gekoppeld wordt aan de hoogte van het risico. →

Laag risico	Gemiddeld risico	Hoog risico
Test use cases	Equivalentieklassen	Beslissingstabellen
Error guessing	Test use cases	Equivalentieklassen
Statement coverage 60%	Statement coverage 70%	Decision coverage 70%

Ik denk dat dit in veel gevallen een onjuiste benadering is! Ik kan me vele situaties voorstellen waarin een hoger risico met minder testen en een informelere testtechniek toch meer zekerheid geeft. Denk bijvoorbeeld aan een login van een intranet applicatie. Die is enorm belangrijk omdat, als dat niet werkt, de gehele applicatie niet bereikbaar is. De login is vaak relatief eenvoudig te testen. Het testen van de lay-out en syntax van alle schermen erachter is veel meer werk.



De juiste test is van veel factoren afhankelijk en kan op vele manieren worden gediversifieerd. Een manier om hiermee te variëren is de gekozen testtechniek. Andere manieren zijn: het aantal testtechnieken die gebruikt worden op hetzelfde deel, de hoeveelheid combinaties die getest worden, de diepgang van de testen, een vastgestelde timebox, et cetera.

## Van klein naar groot

Een teststrategie groeit gedurende het project: we komen steeds meer te weten over het product en we krijgen een steeds beter beeld over de risico's. Hierdoor zullen prioriteiten veranderen, te testen onderdelen aangevuld worden, et cetera. Tijdens het project leren we en daardoor groeit de strategie, van globaal naar steeds meer detail.

Een testtraject begint met een risicoanalyse. Voor de gedefinieerde risico's worden testen uitgevoerd waardoor we weer leren en meer te weten komen over de daadwerkelijke situatie. We leren niet alleen over problemen, maar ook over mogelijkheden die weer tot nieuwe inzichten kan leiden.

## Teststrategie versus testuitvoering versus rapportage

De teststrategie is een belangrijk onderdeel van testen en kan niet los gezien worden van de testuitvoering en de rapportage. Een goede teststrategie geeft de testuitvoering richting en op alle onderdelen van de teststrategie zal later gerapporteerd worden. ➔

Veel testers zien de strategie als een los onderdeel zoals de afbeelding rechts. Ik zie testen als een integraal geheel en kan de test strategie dan ook niet los zien van de testuitvoering en de rapportage (afbeelding links).



concrete en specifieke uitwerking

van wat ik ga testen. Tijdens de testuitvoering voeg ik daar meer details aan toe en wordt de strategie uitgebreid. Uiteindelijk zal mijn testrapportage ook rechtstreeks gebaseerd zijn op de onderdelen uit de teststrategie.

## Testideeën

Het is vaak lastig om goede testen te verzinnen. Daarom gebruik ik tijdens het opstellen van de teststrategie, het ontwerpen van testen en de testuitvoering vele checklists die me helpen om snel ideeën te genereren. Deze checklists zijn eigenlijk

ook heuristieken.

Enkele voorbeelden zijn:

- The Little Black Book on Test Design van Rikard Edgren met daarin 37 sources for Test Ideas maar het hele (gratis) ebook bevat vele bronnen ter inspiratie voor testers. - <http://www.thetesteye.com/papers/TheLittleBlackBookOnTestDesign.pdf>
- Test Heuristics Cheat Sheet van Elisabeth Hendrickson - <http://testobsessed.com/wp-content/uploads/2011/04/testheuristicscheatsheetv1.pdf>
- You Are Not Done Yet checklist van Michael Hunter - <http://www.thebraidytester.com/downloads/YouAreNotDoneYet.pdf>
- 36 days of web testing van Rob Lambert als bron voor testideeën voor websites - <http://thesocialtester.co.uk/wp-content/uploads/2012/10/36DaysOfWebTesting.pdf>
- Op mijn blog zijn meer bronnen te vinden die als inspiratie kunnen dienen. - [http://www.huibschoots.nl/wordpress/?page\\_id=441#ideas](http://www.huibschoots.nl/wordpress/?page_id=441#ideas)

## Ten slotte

Een goede teststrategie neemt ook unit testen (of hoe je de testen ook wil noemen die developers uitvoeren) mee. Testen is een teamsport en unit testen zijn de basis van het testen: zorg dus dat je weet wat de ontwikkelaars doen! En help ze daarbij.

Visualiseer je teststrategie! Gebruik bijvoorbeeld mind maps om een teststrategie vast te leggen. Maar er zijn veel grafische manieren om je testen te ondersteunen. Maak daar gebruik van, want testen is vaak complex en door te tekenen, diagrammen te maken of op andere manieren je (denk)werk te visualiseren, zal het meestal makkelijker gaan.

Lees eens wat meer over 'test framing [ref 12]'. Test framing is een belangrijke vaardigheid die ons helpt om op een logische, samenhangende en toch snelle manier onze testen te rechtvaardigen: bij elke test kunnen aangeven wat het verband is met de testmissie(s). Het doel van de test framing is om te kunnen om heldere en geloofwaardige antwoorden te geven op vragen als 'Waarom voer je deze test uit?' of 'Hoe voer je deze test uit?'.

Houd rekening in je testen met testbaarheid [ref 13] en vraag hierom vanaf het eerste moment! Testbaarheid maakt het testen sneller en praktischer. Bijvoorbeeld door de juiste ontwerptechniek of diagrammen te vragen aan →

ontwerpers, kunnen testontwerpen efficiënter gemaakt worden. Door bij programmeurs om logging te vragen, kunnen testresultaten sneller geanalyseerd worden. Ook ondersteunen tools zoals 'perlclip [ref 14]' je testen. Neem testbaarheid mee in je teststrategie en praat erover met je team!

## Probleemoplossing

Er is geen standaard recept voor het maken van een strategie of het uitvoeren van testen. Heuristieken helpen bij het oplossen van problemen. Zie de teststrategie als een oplossing voor een complex probleem: 'Hoe voldoen we aan de informatiebehoefte van de stakeholders op een zo efficiënt mogelijke manier?'. Dit probleem is in iedere situatie anders en vraagt daarom iedere keer om een andere oplossing. Ik hoop dat testers na het lezen van dit artikel inzien dat 'probleemoplossing [ref 15]' beter bij testen past dan 'recept denken'.

*Met dank aan Ruud Cox voor de vele discussies over dit onderwerp en zijn review, Joep Schuurkes voor een uitgebreide review, Rikard Edgren voor inspiratie en uiteraard James Bach en Michael Bolton voor het delen van hun kennis over HTSM.*

## Referenties

- 1 - <http://www.dorsethouse.com/books/perf.html>
- 2 - <http://www.testingeducation.org/BBST/foundations/BBSTFoundationsNov2010.pdf> (slide 69 e.v.)
- 3 - <http://thetesteye.com/blog/2013/09/what-is-a-good-test-strategy>
- 4 - <http://en.wikipedia.org/wiki/Heuristic>
- 7 - <http://www.satisfice.com/tools/htsm.pdf>
- 8 - [http://thetesteye.com/posters/TheTestEye\\_SoftwareQualityCharacteristics.pdf](http://thetesteye.com/posters/TheTestEye_SoftwareQualityCharacteristics.pdf)
- 9 - [http://dewt.files.wordpress.com/2013/03/thetesteye\\_softwarekwaliteitkenmerken1.pdf](http://dewt.files.wordpress.com/2013/03/thetesteye_softwarekwaliteitkenmerken1.pdf)
- 10 - [http://www.satisfice.com/info\\_rst.html](http://www.satisfice.com/info_rst.html)
- 11 - [http://www.kaner.com/pdfs/test\\_docs\\_pnsgc.pdf](http://www.kaner.com/pdfs/test_docs_pnsgc.pdf)
- 12 - <http://www.developsense.com/resources/TestFraming.pdf>
- 13 - <http://www.satisfice.com/tools/testable.pdf> en <http://www.developsense.com/blog/2009/07/testability/> en
- 14 - <http://www.satisfice.com/tools.shtml>
- 15 - <http://nl.wikipedia.org/wiki/Probleemoplossing> ←