

# De Grootste Misverstanden

## na 15 jaar TestNet

**Onderwerp: Misverstanden**

**Datum: Mei 2012**

**Aanwezigen: <aanwezigen>**

**Classificatie: v1.0**



me-tho-disch (bn.)

1. volgens een bepaalde methode

⇒ *systematisch*





*Verzamelde ervaring*

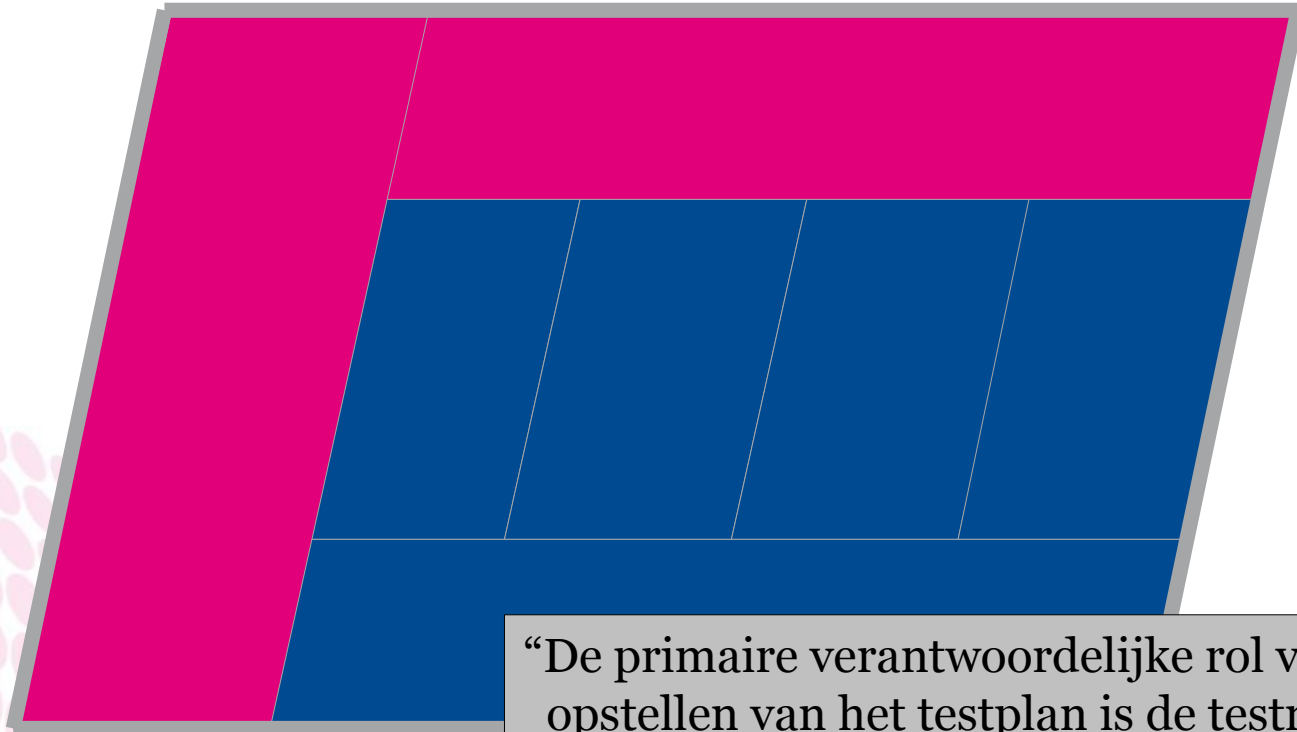




*Vaste structuur*



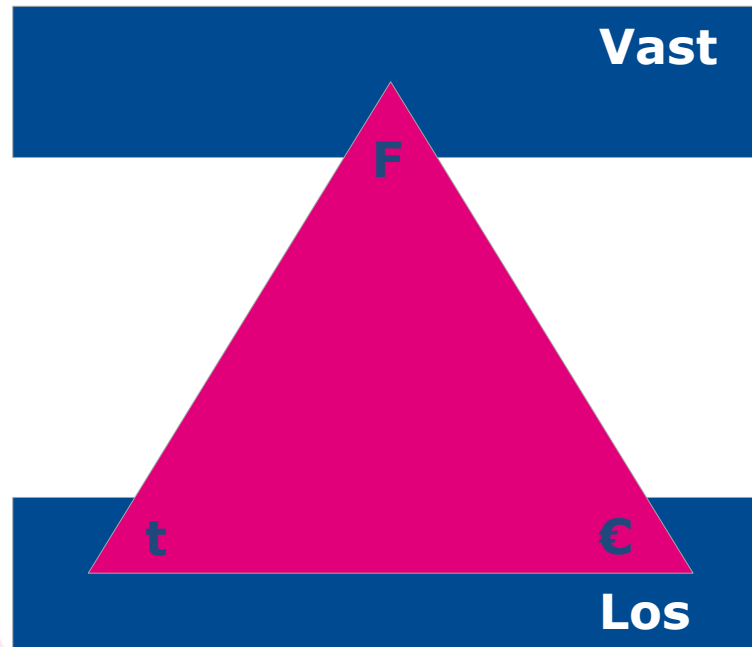
# 1. Testmanager is noodzakelijk



“De primaire verantwoordelijke rol voor het opstellen van het testplan is de testmanager, soms ook testcoördinator genaamd.”



# 1. Testmanager is noodzakelijk

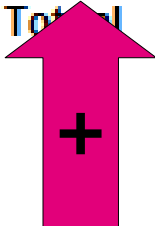
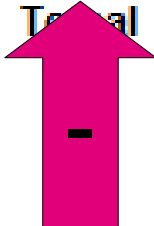


*Keuzes maken is niet moeilijk*



# 1. Testmanager is noodzakelijk

	Juistheid Resultaten		Juistheid Risico's	
	Kans %	Impact €	Kans %	Impact €
Component 1	50	500	50	50000
Component 2	50	500	50	50000
Component 3	50	5000	50	5000
Component 4	50	5000	50	5000
Component 5	50	50000	50	500
Component 6	50	50000	50	500
<b>Total</b>		<b>55500</b>	<b>Total</b>	<b>55500</b>

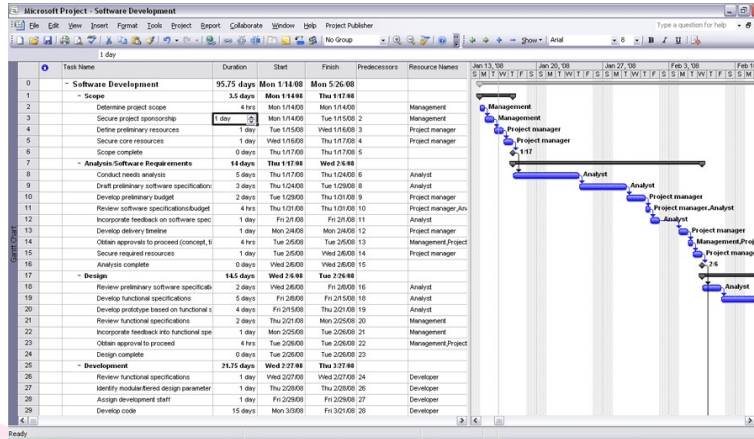



*Het vervolg is een handigheidje*

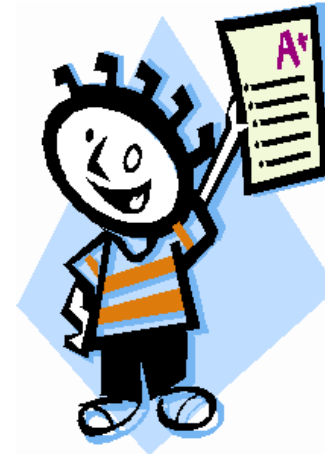




# 1. Testmanager is noodzakelijk



Task Name	Duration	Start	Finish	Predecessors	Resource Names
- Software Development	95.75 days	Mon 1/14/08	Mon 5/26/08		
- Scope	3.5 days	Mon 1/14/08	Thu 1/17/08		
1 Determine project scope	4 hrs	Mon 1/14/08	Mon 1/14/08		Management
2 Secure project sponsorship	1 day	Mon 1/14/08	Tue 1/15/08	2	Management
3 Define preliminary resources	1 day	Tue 1/15/08	Wed 1/16/08	3	Project manager
4 Secure core resources	1 day	Wed 1/16/08	Thu 1/17/08	4	Project manager
5 Scope complete	5 days	Thu 1/17/08	Thu 1/17/08	5	
- Analysis Software Requirements	14 days	Thu 1/17/08	Wed 2/4/08		
6 Contact needs analysis	5 days	Thu 1/17/08	Thu 1/24/08	6	Analyst
7 Draft preliminary software specifications	2 days	Thu 1/24/08	Tue 1/29/08	7	Analyst
8 Develop preliminary budget	2 days	Tue 1/29/08	Thu 1/31/08	8	Project manager
9 Review software specifications/budget	4 hrs	Thu 1/31/08	Thu 1/31/08	9	Project manager,Analyst
10 Incorporate feedback on software spec	1 day	Fri 2/1/08	Fri 2/1/08	10	Analyst
11 Develop delivery timeline	1 day	Mon 2/4/08	Mon 2/4/08	11	Project manager
12 Obtain approvals to proceed (concept, RFP)	4 hrs	Tue 2/5/08	Tue 2/5/08	12	Management,Project manager
13 Secure required resources	1 day	Wed 2/6/08	Wed 2/6/08	13	Project manager
14 Analysis complete	5 days	Wed 2/6/08	Wed 2/6/08	14	
- Design	14.5 days	Wed 2/6/08	Tue 2/24/08		
15 Review preliminary software specifications	2 days	Wed 2/6/08	Fri 2/8/08	15	Analyst
16 Develop functional specifications	5 days	Fri 2/8/08	Fri 2/15/08	16	Analyst
17 Develop prototype based on functional specifications	4 days	Fri 2/15/08	Thu 2/21/08	17	Analyst
18 Review functional specifications	2 days	Thu 2/21/08	Mon 2/25/08	18	Management
19 Incorporate feedback into functional specifications	1 day	Mon 2/25/08	Tue 2/26/08	19	Management
20 Obtain approval to proceed	4 hrs	Tue 2/26/08	Tue 2/26/08	20	Management,Project manager
21 Design complete	0 days	Tue 2/26/08	Tue 2/26/08	21	
- Development	24.75 days	Wed 2/27/08	Thu 3/27/08		
22 Review functional specifications	1 day	Wed 2/27/08	Wed 2/27/08	22	Developer
23 Identify modular based design parameter	1 day	Thu 2/28/08	Thu 2/28/08	23	Developer
24 Assign development staff	1 day	Fri 2/29/08	Fri 2/29/08	24	Developer
25 Develop code	15 days	Mon 3/3/08	Fri 3/21/08	25	Developer



*Al het andere is projectmanagement*





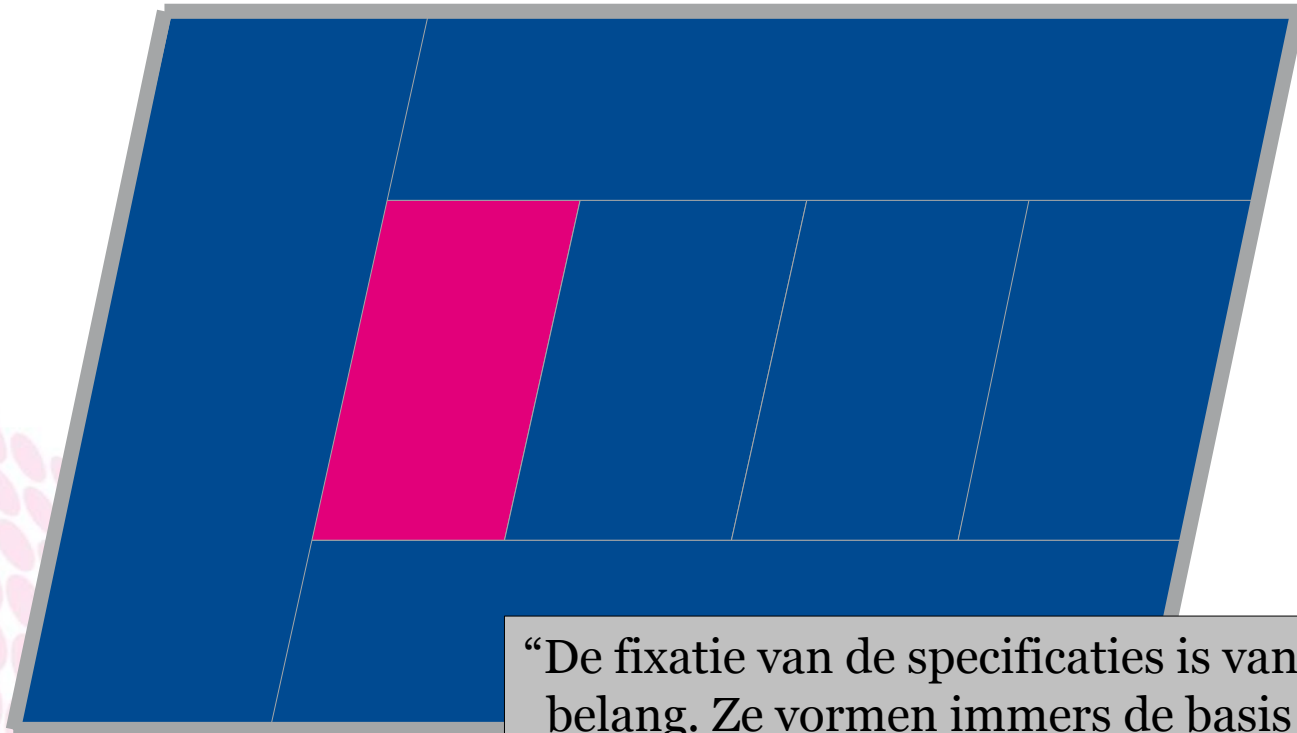
# 1. Testmanager is noodzakelijk



*Kortom; Testmanager is het 'pak van de keizer'*



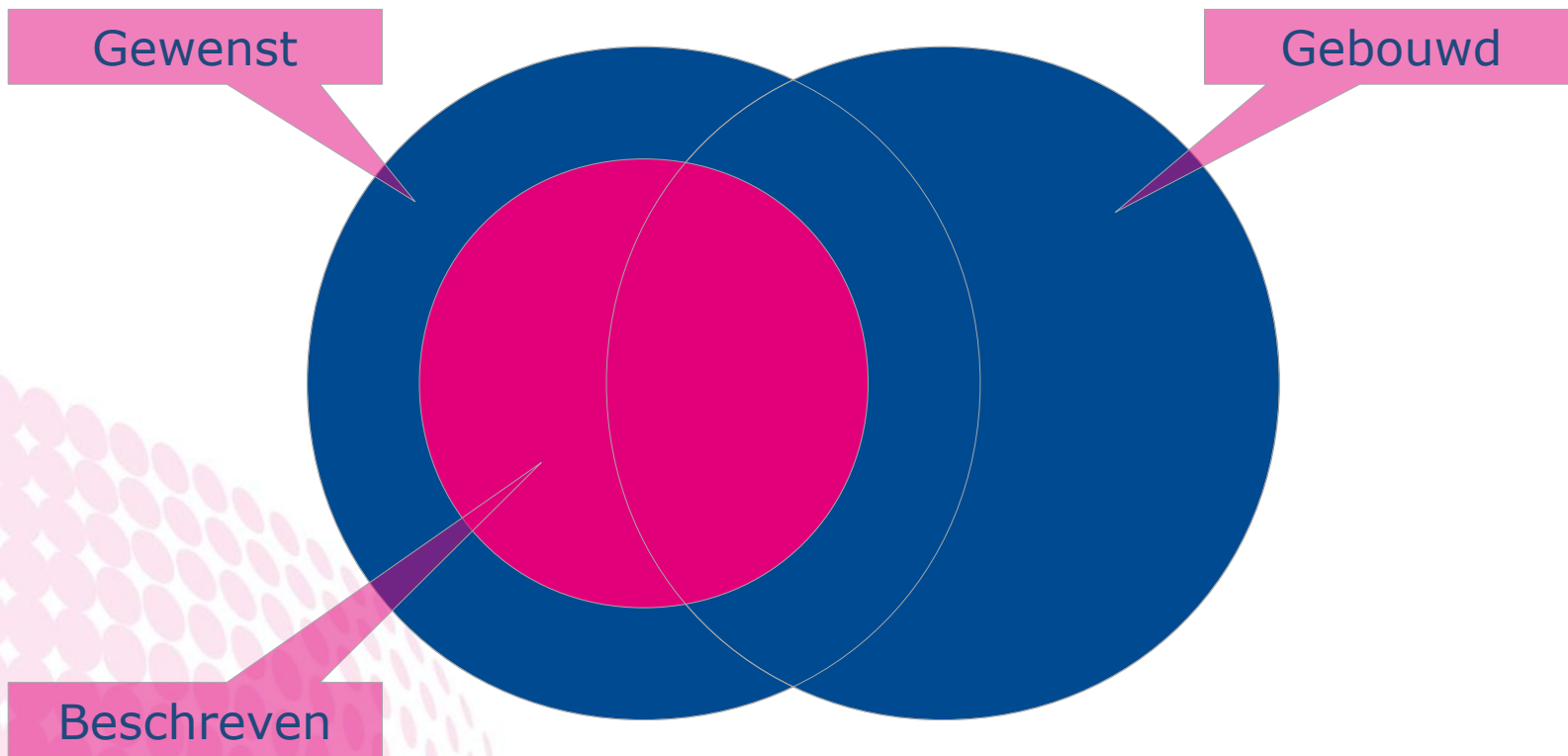
## 2. Technieken zijn afhankelijk van documentatie



“De fixatie van de specificaties is van groot belang. Ze vormen immers de basis voor zowel de testers als de ontwikkelaars...”



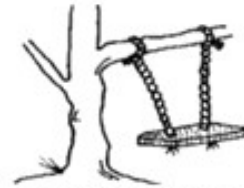
## 2. Technieken zijn afhankelijk van documentatie



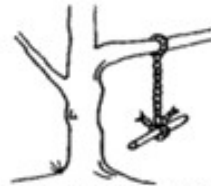
*Documentatie is een beperking*



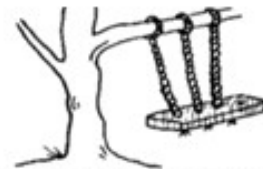
## 2. Technieken zijn afhankelijk van documentatie



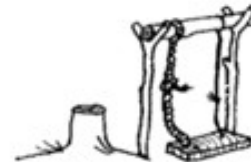
What the user really wanted



What the user asked for



How the system was designed



How it actually works



As the programmer wrote it

*Het gaat om interpretatie, niet om werking*



## 2. Technieken zijn afhankelijk van documentatie

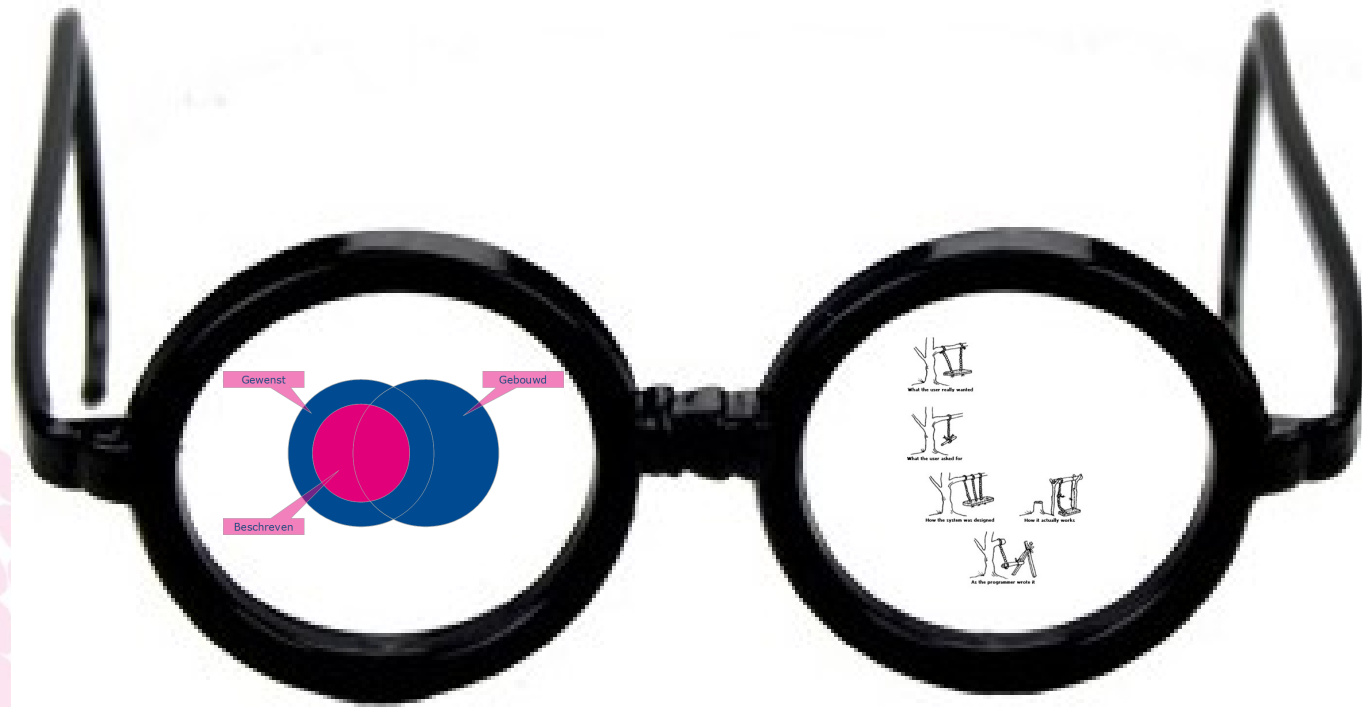


- Vraag 1; Welke condities zijn er (van belang)?
- Vraag 2; Welke acties zijn er (van belang)?
- Vraag 3; Wat is de interactie tussen de condities?
- Vraag 4; Welke acties moeten volgen uit welke condities?

*Stel de juiste vragen*



## 2. Technieken zijn afhankelijk van documentatie

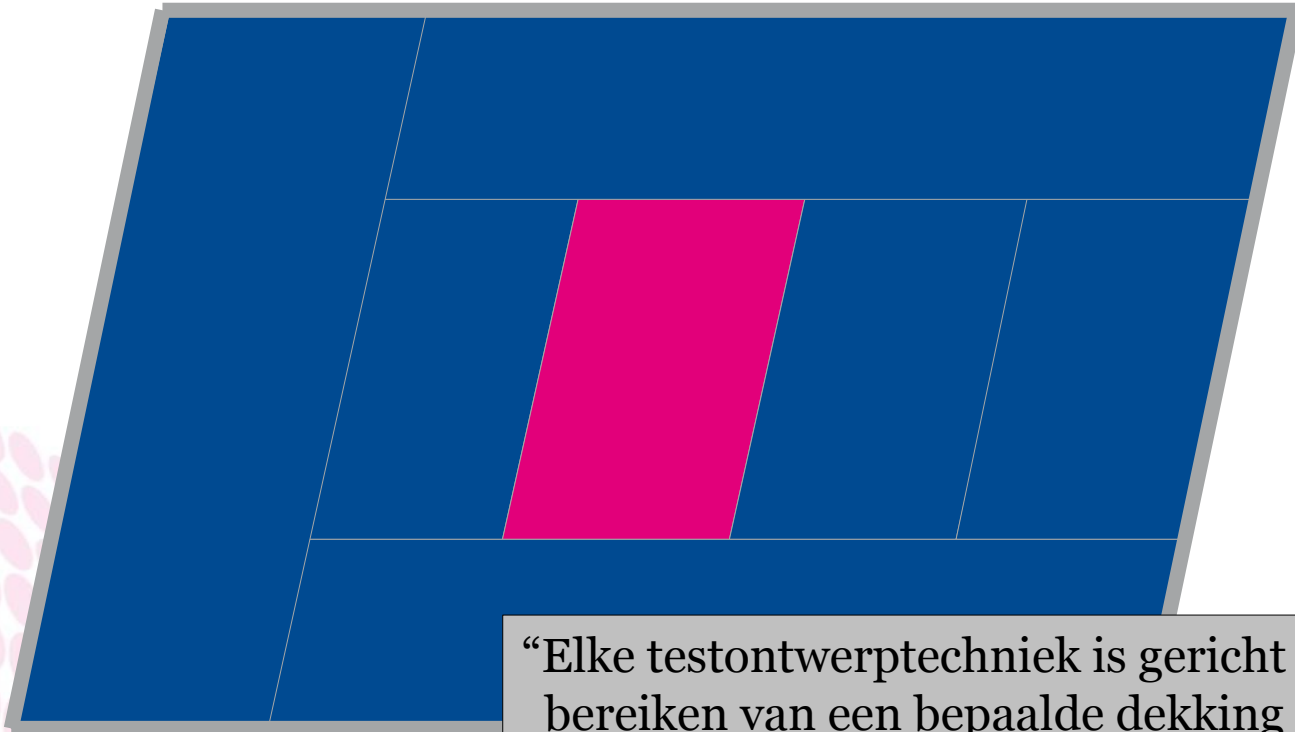


*Kortom; Focus op documentatie maakt blind*





### 3. Testen moet bevindingen opleveren

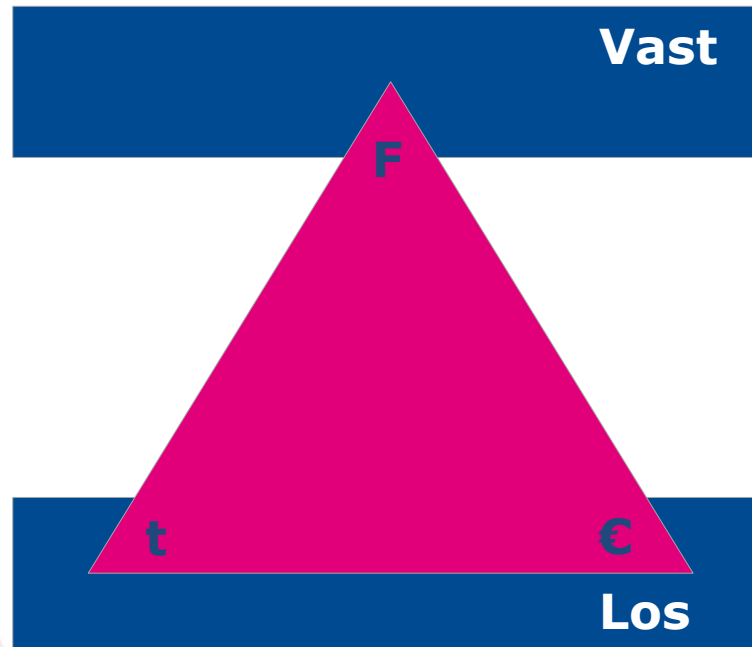


“Elke testontwerptechniek is gericht op het bereiken van een bepaalde dekking om bepaalde soorten fouten te vinden.”





### 3. Testen moet bevindingen opleveren



*Testen heeft vele doelen*



### 3. Testen moet bevindingen opleveren

42

42

42

42

42

42

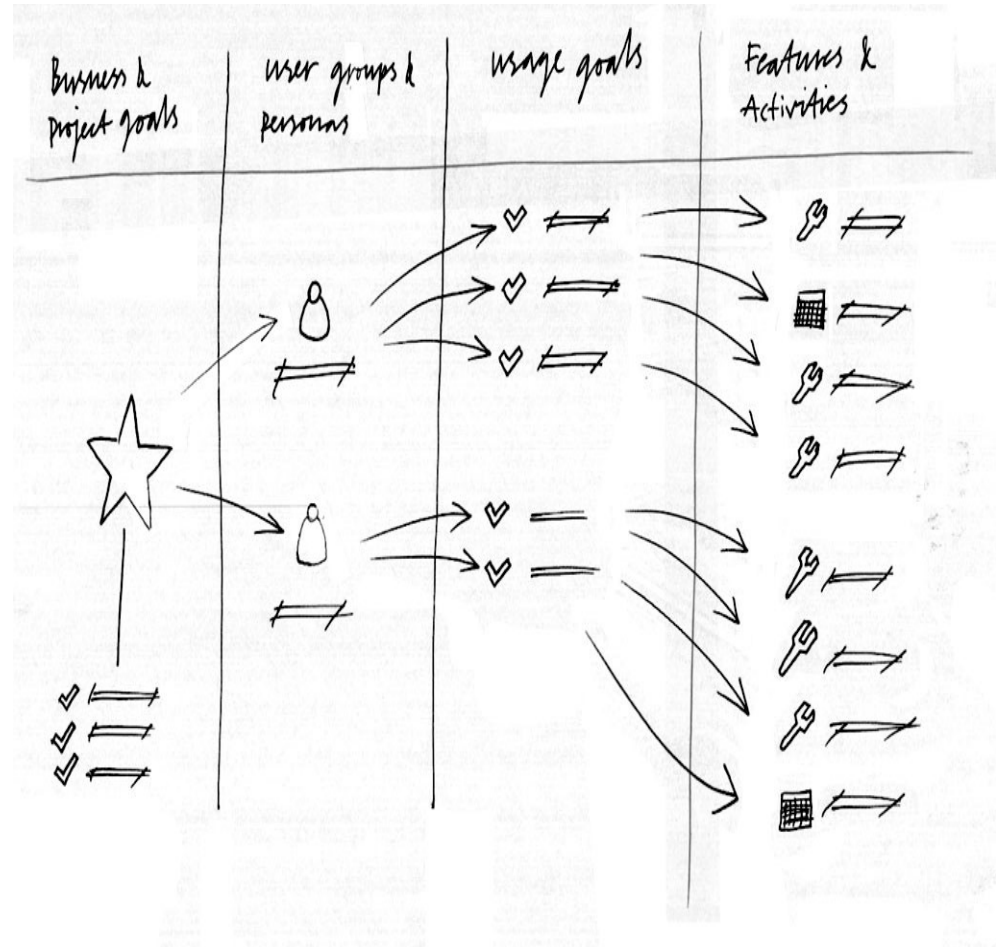
“These are not  
the answers  
you are looking for”



*Bevindingen zeggen niets*



# 3. Testen moet bevindingen opleveren



*Het effect is bepalend*





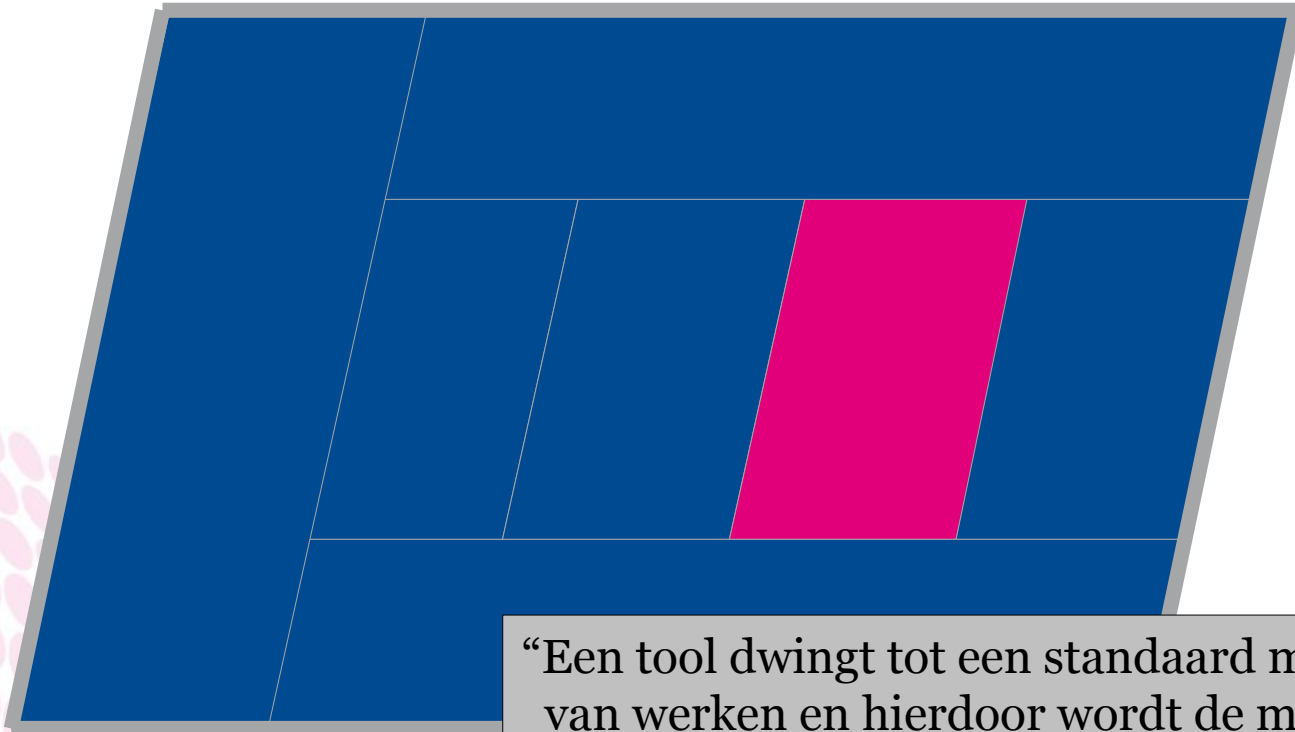
### 3. Testen moet bevindingen opleveren



*Kortom; Testen moet inzicht geven*



## 4. Testen kun je automatiseren



“Een tool dwingt tot een standaard manier van werken en hierdoor wordt de menselijke factor uitgeschakeld.”



## 4. Testen kun je automatiseren



*Softwareontwikkeling draait om prototypes*



## 4. Testen kun je automatiseren



*Testen is onvoorspelbaar*





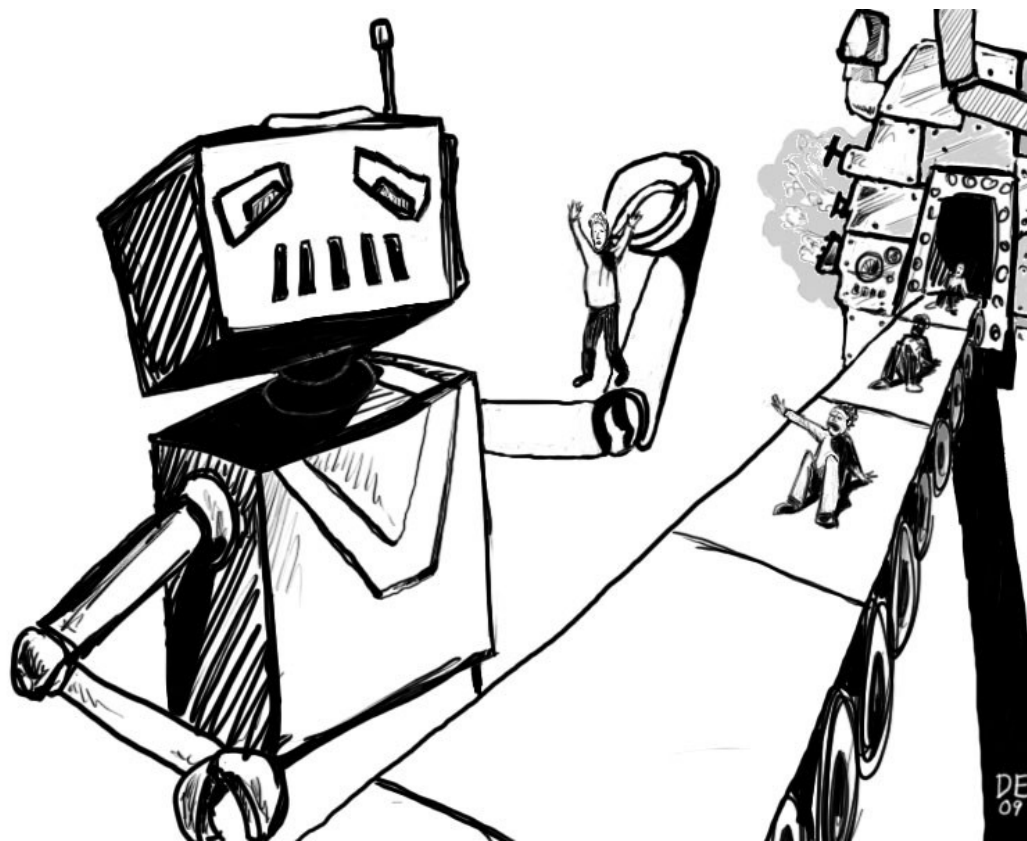
## 4. Testen kun je automatiseren



*Automatisering is herhaling*



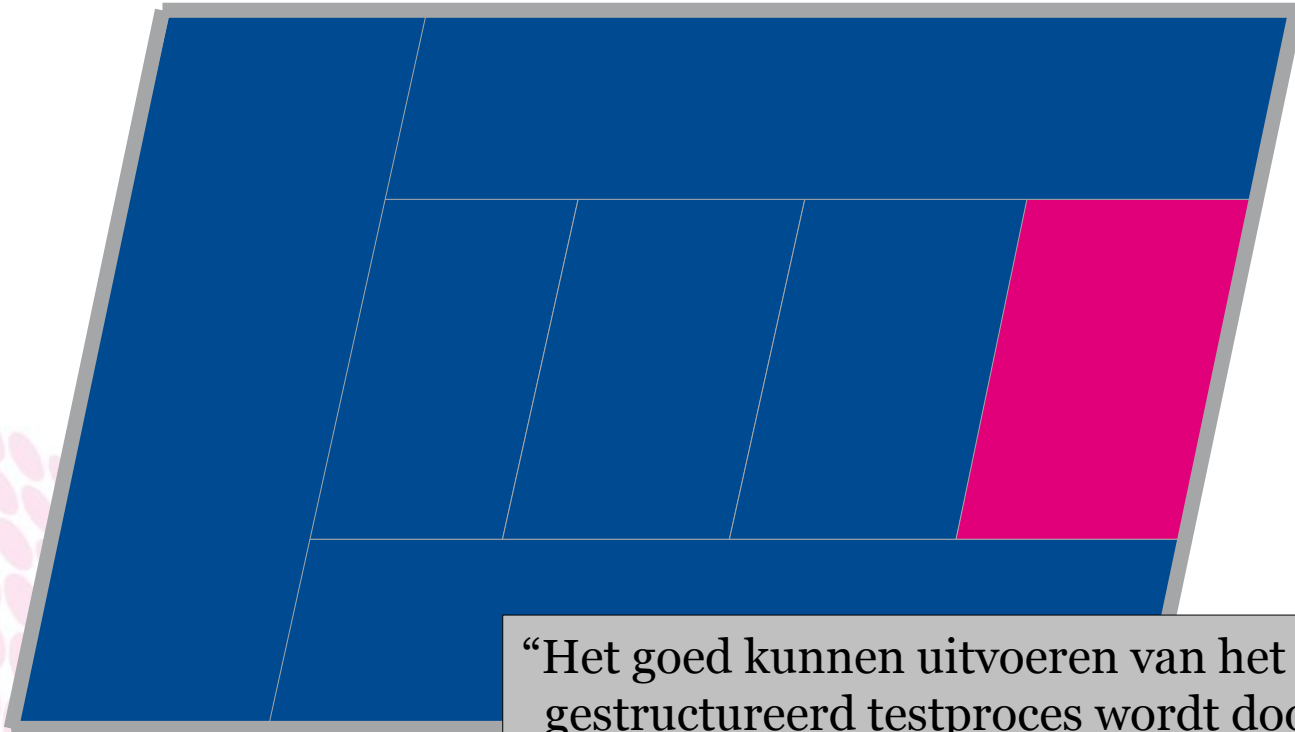
## 4. Testen kun je automatiseren



*Kortom; Controleren is geen testen*



## 5. Testers worden goed door van elkaar te leren



“Het goed kunnen uitvoeren van het gestructureerd testproces wordt door TMap ondersteund door een complete gereedschapskist.”



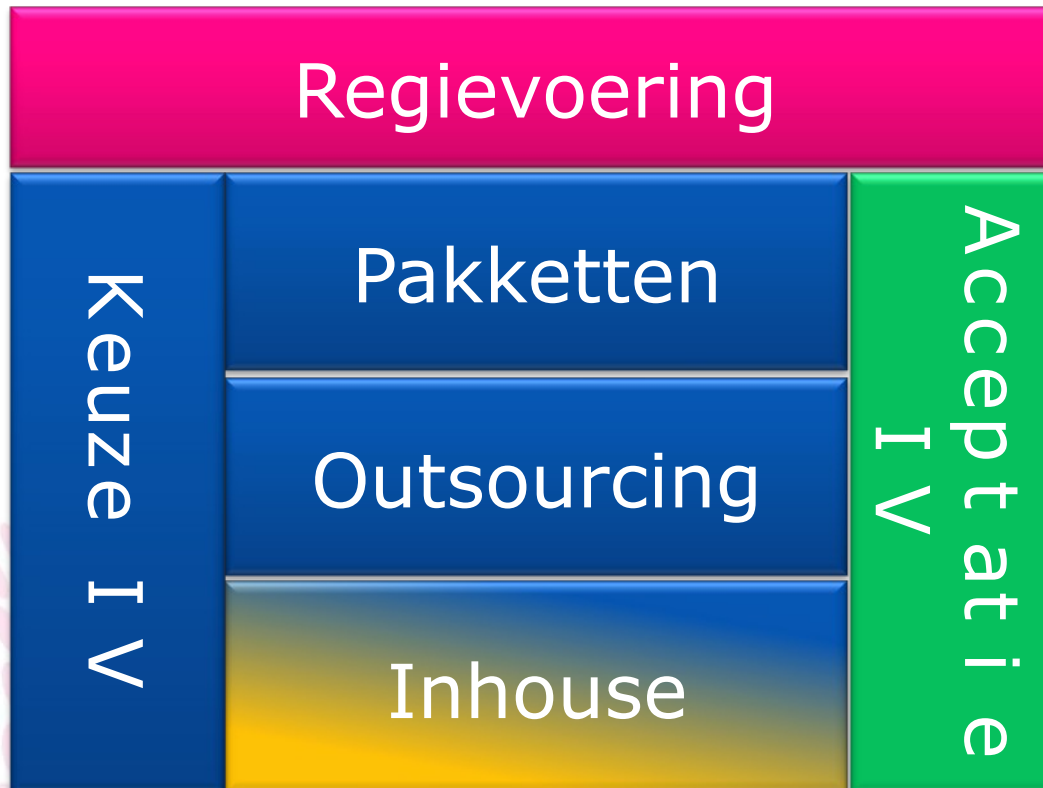
## 5. Testers worden goed door van elkaar te leren

- **Fricctie**; agile ontwikkelen vs. uitbesteden/standaardisering
- **Versnippering**; SOA, Cloud, overstijgende ketens
- **Vervlechting**; core business toenemend gelijk aan ICT

*De wereld verandert...*



## 5. Testers worden goed door van elkaar te leren

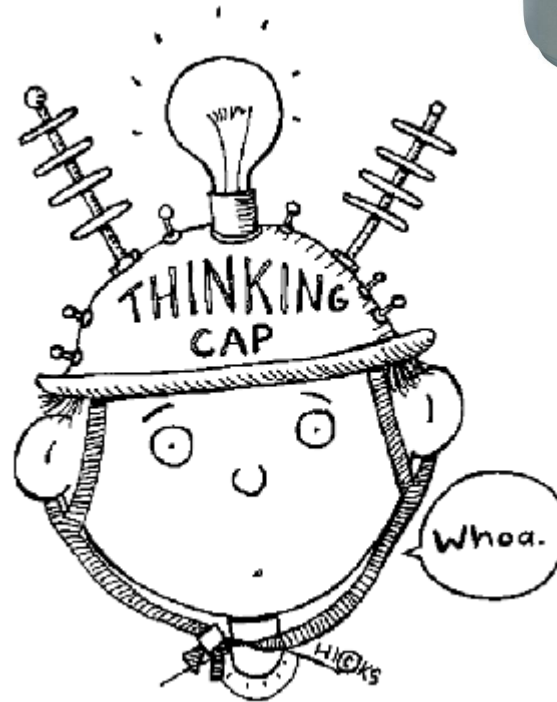
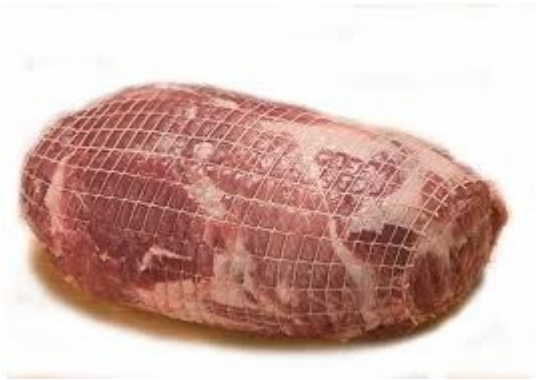


*Testen vindt plaats op snijvlakken*





## 5. Testers worden goed door van elkaar te leren



*'Best Practices' bestaan niet*



## 5. Testers worden goed door van elkaar te leren



*Kortom; Goed in je vak word je buiten je vak*







