



Tien kinken in de kabel bij continue  
(test)verbetering.

Kees Blokland

## Kink 1

Je in de luren laten leggen  
door de context



*Wordt deel van de  
oplossing!*

## Kink 2

# De cultuur in de organisatie

# Dikke testplannen

risicomijdend

gewoonte

formeel

wollig – niet concreet

hierarchisch



## Kink 3

Er mist een probleem

Agile (test)assessment

AGILE IN DE  
ORGANISATIE

Gevolg?

TESTEN

ONTSLAG...

# Agile (test)assessment in 4 agile teams

		Forming				Norming				Performing	
1	Opdrachtgeve									3	2
2	Begroting en									3	3
3	Mensen									1	1
4	Interactie									0	3
5	Teamwork									2	0
6	Test proces									0	1
7	Test management	4	1	0	1	0	0	0	3	4	
8	Tester profie									1	1
9	Test automa									2	3
10	Regressie &									0	1
11	Bevindingen									1	2
12	Testomgevi									4	2

Vervolg?

Niets!





*erMoetWelPijnZijn()*

## Kink 4

# Maturity-improvement paradox

# Tussen assessment 1 en 2 zit ruim een jaar

## Afdeling A

Aandachtsgebieden	Initieel	assessment 1				assessment 2			
		Beheerst				Beheerst			
1 Opdrachtgeverschap		A	B	B	C	A	B	B	C
2 Mate van betrokkenheid		A	B	C	E	A	B	C	E
3 Teststrategie		A	A	B	E	A	A	B	E
4 Testorganisatie		A	D	D	E	A	D	D	E
5 Communicatie		B	C	C	D	B	C	C	D
6 Rapportage		A	C	C		A	C	C	
7 Testprocesbeheer		A	A	B	B	A	A	B	B
8 Begroting en planning		B	B	C	C	B	B	C	C
9 Metrieken		C	C	D		C	C	D	
10 Bevindingenbeheer		A	A	B	D	A	A	B	D
11 Testwarebeheer		B	B	D	E	B	B	D	E
12 Toepassing van de methodiek		C	D	E		C	D	E	
13 Testerprofessionaliteit		D	D	E	E	D	D	E	E
14 Testgevalontwerp		A	A	E		A	A	E	
15 Testhulpmiddelen		E	E	E		E	E	E	
16 Testomgeving		C	D	D	E	C	D	D	E

# Tussen assessment 1 en 2 zit ruim een jaar

## Afdeling B

Aandachtsgebieden		Initieel	assessment 1				assessment 2			
			Beheerst				Beheerst			
1	Opdrachtgeverschap		A	B	B	C	A	B	B	C
2	Mate van betrokkenheid		A	B	C	E	A	B	C	E
3	Teststrategie		A	A	B	E	A	A	B	E
4	Testorganisatie		A	D	D	E	A	D	D	E
5	Communicatie		B	C	C	D	B	C	C	D
6	Rapportage		A	C	C		A	C	C	
7	Testprocesbeheer		A	A	B	B	A	A	B	B
8	Begroting en planning		B	B	C	C	B	B	C	C
9	Metrieken		C	C	D		C	C	D	
10	Bevindingenbeheer		A	A	B	D	A	A	B	D
11	Testwarebeheer		B	B	D	E	B	B	D	E
12	Toepassing van de methodiek		C	D	E		C	D	E	
13	Testerprofessionaliteit		D	D	E	E	D	D	E	E
14	Testgevalontwerp		A	A	E		A	A	E	
15	Testhulpmiddelen		E	E	E		E	E	E	
16	Testomgeving		C	D	D	E	C	D	D	E



*Even geduld a.u.b.*

## Kink 5

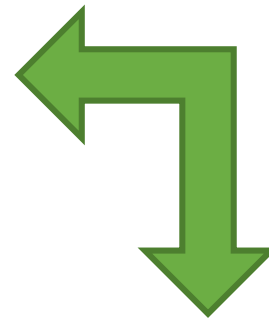


# Trekken, niet duwen

KANBAN board with improvement tasks

Story	To Do	In Process	To Verify	Done
As a user, I... 8 points	Code the... 9 Code the... 2 Test the... 8	Test the... 6 Code the... DC 4 Test the... SC 8	Test the... SC 5	Code the... SC 4 Test the... SC 4 Test the... SC 5 Test the... SC 6
As a user, I... 6 points	Code the... 8 Code the... 4	Test the... 8 Code the... DC 8		Test the... SC 4 Test the... SC 5 Test the... SC 6

*Just in time*



★ Activiteiten waar verbeteringen op gepland zijn

## Kink 6



Te grote plannen

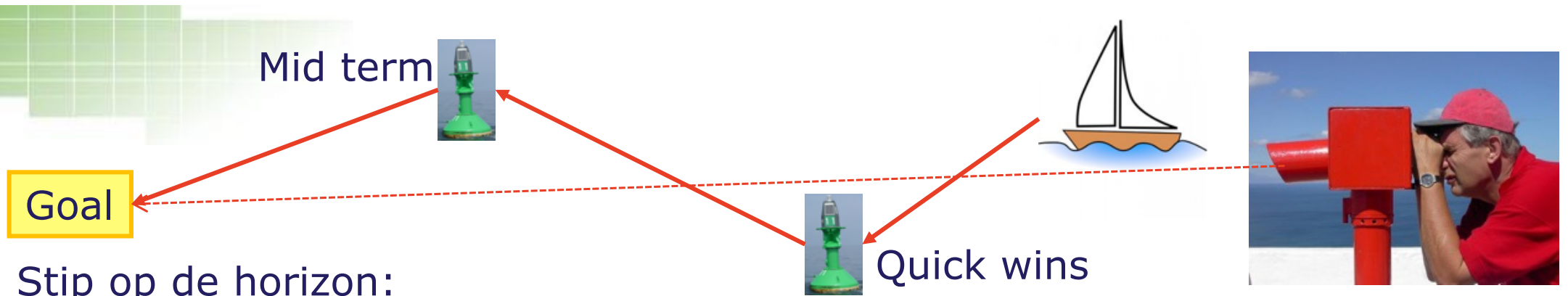


# 1 Table of contents

General IT services	
<b>1</b>	<b>Table of contents</b>
1	Table of contents
2	Executive summary
3	Introduction
4	Background
5	Objectives
6	Methodology
7	General IT services
8	IT services
9	IT services
10	IT services
11	IT services
12	IT services
13	IT services
14	IT services
15	IT services
16	IT services
17	IT services
18	IT services
19	IT services
20	IT services
21	IT services
22	IT services
23	IT services
24	IT services
25	IT services
26	IT services
27	IT services
28	IT services
29	IT services
30	IT services
31	IT services
32	IT services
33	IT services
34	IT services
35	IT services
36	IT services
37	IT services
38	IT services
39	IT services
40	IT services
41	IT services
42	IT services
43	IT services
44	IT services
45	IT services
46	IT services
47	IT services
48	IT services
49	IT services
50	IT services
51	IT services
52	IT services
53	IT services
54	IT services
55	IT services
56	IT services
57	IT services
58	IT services
59	IT services
60	IT services
61	IT services
62	IT services
63	IT services
64	IT services
65	IT services
66	IT services
67	IT services
68	IT services
69	IT services
70	IT services
71	IT services
72	IT services
73	IT services
74	IT services
75	IT services
76	IT services
77	IT services
78	IT services
79	IT services
80	IT services
81	IT services
82	IT services
83	IT services
84	IT services
85	IT services
86	IT services
87	IT services
88	IT services
89	IT services
90	IT services
91	IT services
92	IT services
93	IT services
94	IT services
95	IT services
96	IT services
97	IT services
98	IT services
99	IT services
100	IT services

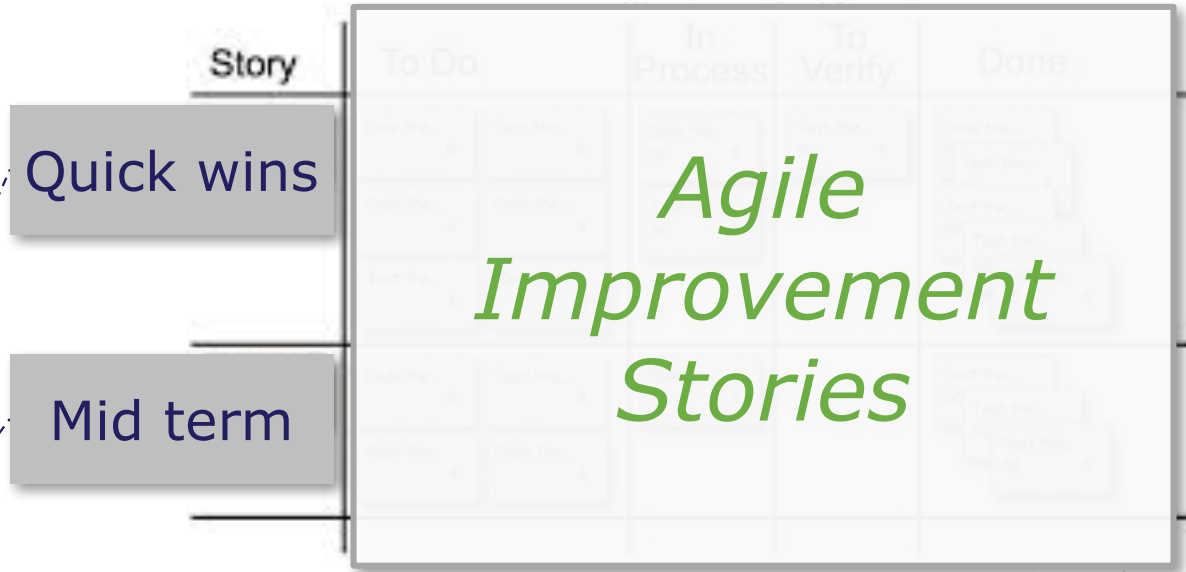
## Outline implementation





Stip op de horizon:

- wat willen we bereiken?
- waar was het om begonnen?
  
- wat kunnen we nu oppakken
- kost weinig inspanning en geld
- heeft snel resultaat
  
- wat levert sowieso iets op
- maar kost wel wat meer tijd
- tegen acceptabele kosten



## Kink 7

Geen direct nut - *gezien*

A background image showing several people crossing a river by stepping on large, dark rocks. The water is shallow and clear. The people are wearing casual outdoor clothing. The scene is outdoors with trees and greenery in the background.

ruimte voor persoonlijke groei

tester wordt belangrijker

grote projecten opdelen in kleine, haalbare stukken

ruimte voor inhoudelijk leiderschap

minder multi tasking

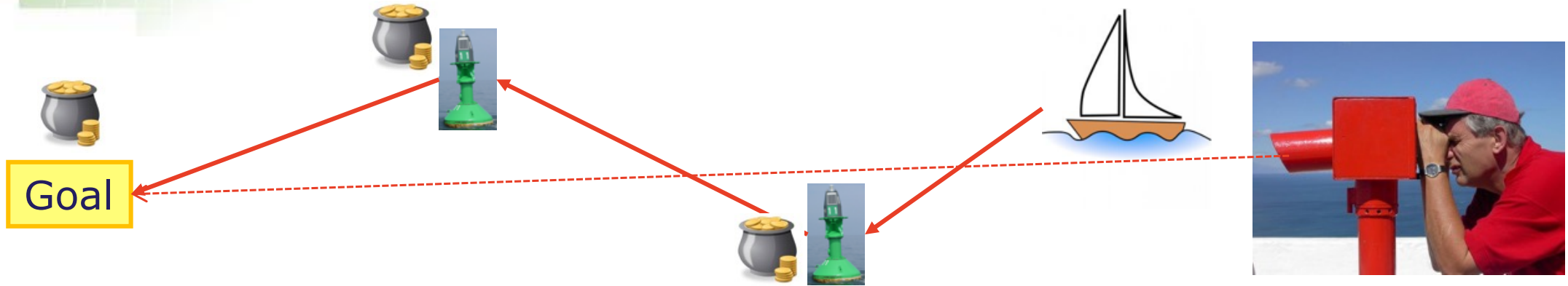
de business komt nu op het kritieke pad

minder testsoorten

testautomatisering, maar nu echt!

hé, het werkt!

# Stepping stones



## Kink 8



Op het verkeerde been

# De Fase Van De Grote Verwarring





*Explore more!*





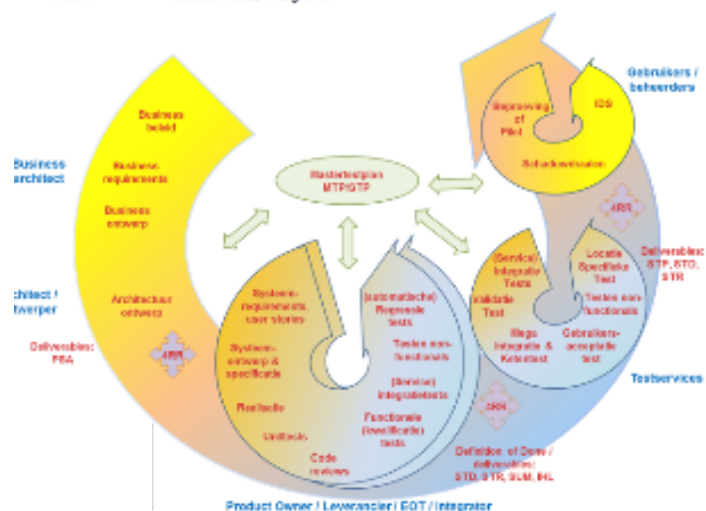
## Kink 9



Het testproces en systeemontwikkelproces zijn nauw met elkaar verweven. De één levert de producten, die door de ander getoetst en getest worden. Binnen ProRail wordt de relatie tussen deze processen sinds 2013 gevisualiseerd in het zogenaamde WAU-testmodel, dat staat voor Waterval Agile United. In dit hoofdstuk wordt het WAU-model toegelicht. In het hoofdstuk erna worden de testvormen behandeld waarbij aangegeven wordt waaruit de testbasis voor een specifieke testvorm bestaat.

Onderstaande schema toont het model. De betekenis van de kleuren is globaal als volgt, maar is bewust niet scherp afgekaderd, omdat de grenzen in de praktijk ook niet zo scherp zijn:

- Goel - ProRail architecten, ontwerpers, beheerders of andere ProRail onderdelen, in ieder geval buiten het werkgebied van TS.
- Roze - Leverancier / Eigen Ontwikkel Team
- Blaauw - Testservices, Integrator



#### Toelichting:

- Het WAU-testmodel begint linksboven met de fasen waarin het systeem wordt bedacht en ontworpen, van wens, wat, beleid, kans en/of probleem naar (globale) eisen, architectuur en ontwerp. Momenteel gebeurt dit hoofdzakelijk op sequentiële (waterval) wijze.
- Daarna wordt het systeem gerealiseerd (gebouwd), normaal gesproken door een leverancier (uitbesteding) en bij uitzondering door een Eigen Ontwikkel Team (EOT). Dit traject is weergegeven als een itererend proces waar vaak met de methode van scrum wordt gewerkt. In het model is een aantal van de belangrijkste activiteiten weergegeven

(opstellen van gedetailleerde requirements of user stories, softwareontwerp, realisatie) met nadruk op de mogelijke **testvormen** die hierin toegepast kunnen worden.

- De acceptatietest wordt aangestuurd en uitgevoerd door Testservices. Na oplevering vanuit de leverancier aan de acceptatietest zijn ook hier een aantal **mogelijke** testvormen weergegeven. Ook de acceptatietest heeft een itererend karakter. Zo kan bijvoorbeeld worden afgesproken dat elke leveranciersoplevering (sprint) voor acceptatietesten wordt opgeleverd, of om de zoveel sprints, of alleen de laatste 2 of 3 sprints, of zelfs één keer helemaal tegen het einde.
- Dit onderscheid in (test)verantwoordelijkheden wordt in de praktijk vertaald in het groeperen van testactiviteiten in testsoorten, waarbij een testsoort staat voor een groep van testactiviteiten die gezamenlijk wordt uitgevoerd en aangestuurd. Deliverables van de verschillende testsoorten zijn de STP, STD en STR. Gezien bovenstaande zijn dus twee testsoorten te onderscheiden: het leveranciers testen en het acceptatietesten. Tests zoals unittesten, performancetesten, functionele (kwaliteits)testen, regressietesten, validatietesten zijn daarmee testvormen (=gericht op een of meer kwaliteitsaspecten) binnen één of beide testsoorten.
- Nu de keuze voor een agile ontwikkelingsaanpak steeds vanzelfsprekender wordt, moet de vraag worden gesteld of een separate acceptatietest nog wel nodig is. Immers, vanuit de agile/scrum gedachte worden alle ontwerp-, bouw- en testactiviteiten met zeer grote voorkeur uitgevoerd binnen de agile/scrum-teams zelf. Binnen ProRail is momenteel een separate acceptatietest nog wel de standaardregel, al mag hier beargumenteerd van af worden geweken. Belangrijkste redenen voor deze keuze zijn:
  - De acceptatietest wordt uitgevoerd in een representatieve testomgeving die bij ProRail staat en die niet aanwezig is bij de verschillende leveranciers. Deze omgeving staat onder beheer van ProRail (Testservices) en kan niet zomaar door projecten gebruikt worden om "dingen uit te proberen"
  - Testen van non-functionals (met name hoog-beschikbaarheid), testen van de schaal van systemen en operationeel beheersaspecten zijn normaal gesproken erg belangrijk en vereisen een representatieve omgeving en specifieke kennis, die beide bij Testservices aanwezig zijn
  - De capaciteit van Testservices is onvoldoende om haar testers voor alle lopende projecten gedurende het hele agile/scrum-traject in te zetten bij deze teams.
  - De testfasen aan de leveranciers zijn (momenteel) niet heel hoog en betreffen meestal enkel het opleveren van deliverables zoals testplan, -design en -rapport.
  - Bij de leveranciers is minder systeem- en bedrijfskennis beschikbaar waardoor er een grotere kans is dat de kwaliteit in bepaalde aspecten onvoldoende is. Wegens onvoldoende capaciteit bij Testservices is dit niet op te heffen door inzet van haar testers op de teams.
  - Met veel leveranciers, veel projecten en veel scrum teams is de (test- en software)kwaliteit niet altijd even voorspelbaar. Een onderdeel vanuit ProRail Testservices blijft daarom belangrijk.
  - Configuratiebeheer van de software en de omgevingen is niet altijd zoals het hoort, zodat het formeel overdragen naar de A-omgeving een test op zichzelf blijft.
- Na acceptatietesten vindt implementatie plaats. Ook dit heeft steeds vaker een itererend karakter, in de vorm van pilots, schaduwdraaien of daadwerkelijke (al of niet stapsgewijze) in-productie-namen.
- Het model als geheel heeft ook een itererend karakter, om duidelijk te maken dat het traject doorlopen wordt om een release in productie te brengen, maar daarna vaak gelijk alweer start met een volgende release.
- Zowel bij de leveranciers als bij de acceptatietests wordt nadrukkelijk gesteld dat de genoemde testvormen mogelijkheden zijn, maar niet per se voorgeschreven. De reden hierachter is om flexibiliteit in de totale testaanpak te verkrijgen. Als stelregel geldt dat des te vroeger iets getest kan worden, des te efficiënter dit is. Echter, dit is niet altijd mogelijk, bijvoorbeeld omdat de testomgeving niet representatief genoeg is, omdat het systeem nog niet volwassen genoeg is, of omdat getest moet worden met andere systemen, die niet

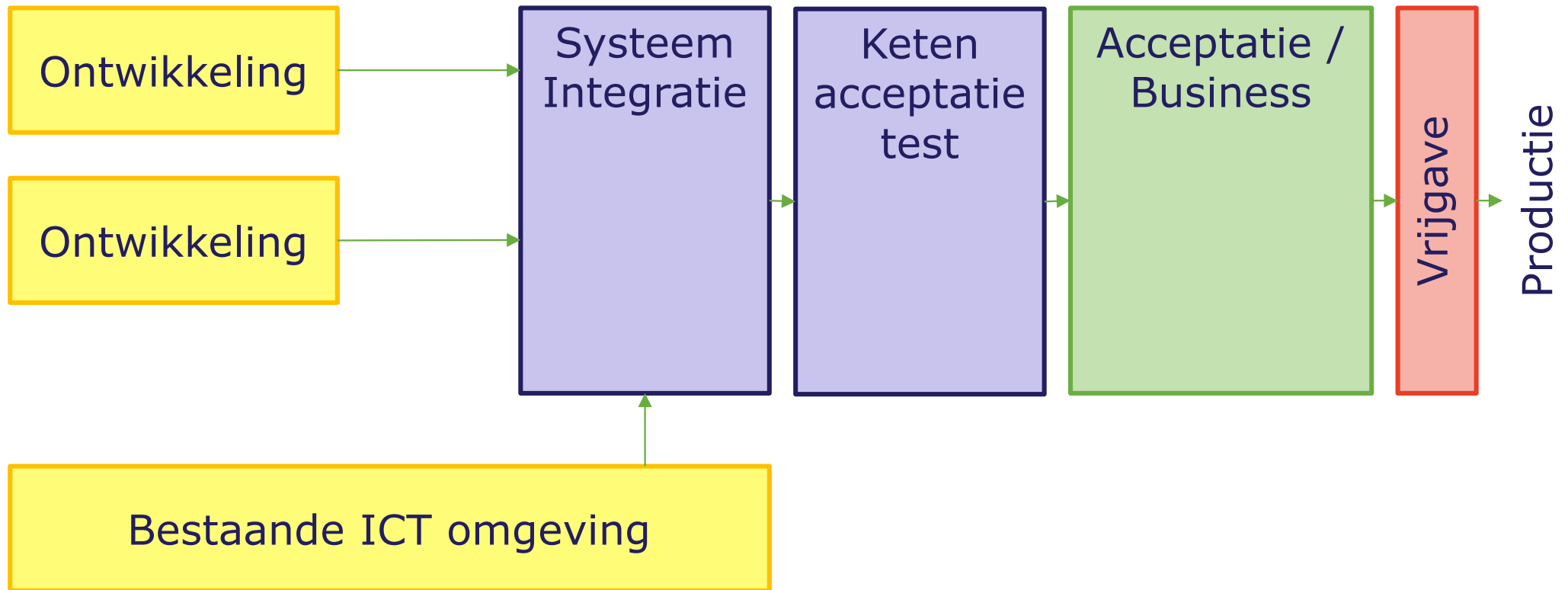
beschikbaar zijn bij de leverancier. Ook kan het gewenst zijn vanuit risico- en efficiëntieoverwegingen bepaalde testvormen dubbel uit te voeren. Bijvoorbeeld een performancetest kan bij de leverancier worden uitgevoerd in een minder representatieve testomgeving om vroegtijdige signalen te ontdekken, terwijl een representatieve performancetest in de acceptatietest wordt uitgevoerd om zoveel mogelijk zekerheid te hebben dat er straks in productie geen problemen zullen ontstaan.

- Ook kan het gewenst zijn andere testvormen te hanteren dan de in het model genoemde voorbeelden. Dit is zonder meer toegestaan.
- Een speciale categorie vormen de tests die als gevolg van SIL-classificatie uitgevoerd moeten worden, zoals bijvoorbeeld een Factory Acceptance Test. De SIL-classificatie stelt voornamelijk eisen aan (de onafhankelijkheid van) de partij die de test uitvoert. Daarom moet goed worden nagedacht over welke partij (leverancier of acceptant) de test uitvoert, en garandeert deze aan de SIL-eisen moet voldoen.
- Om over het gehele ontwikkel- / testtraject efficiënt te kunnen testen, is flexibiliteit in de keuze van welke testvormen wanneer worden toegepast noodzakelijk. Maar levens moet getoetst worden dat uiteindelijk (bij het in productie gaan) de kwaliteitsattributen de testende partij hebben die de stakeholders noodzakelijk achten. Daarom is aansturing van de keuze van testvormen via een masterestplan essentieel. TS voert de regie over het opstellen en bewaken van het masterestplan.
- De 4R-figuren staan voor momenten waarop 4 Richtingen Reviews uitgevoerd moeten worden. Dit zijn reviews waarbij verschillende rollen een deliverable vanuit 4 richtingen beoordelen:
  - Omhoog: Voldoet het aan normen, standaarden, referentie-architectuur?
  - Links: Is het consistent met andere documentatie (in dezelfde taal/fase)?
  - Rechts: Is het bruikbaar (voor gebruikers)?
  - Omlaag: Is het (ontwerpbaar), bouwbaar, testbaar, implementeerbaar?

Rollen zijn bijvoorbeeld architect, ontwerper, ontwikkelaar, leider, QA.

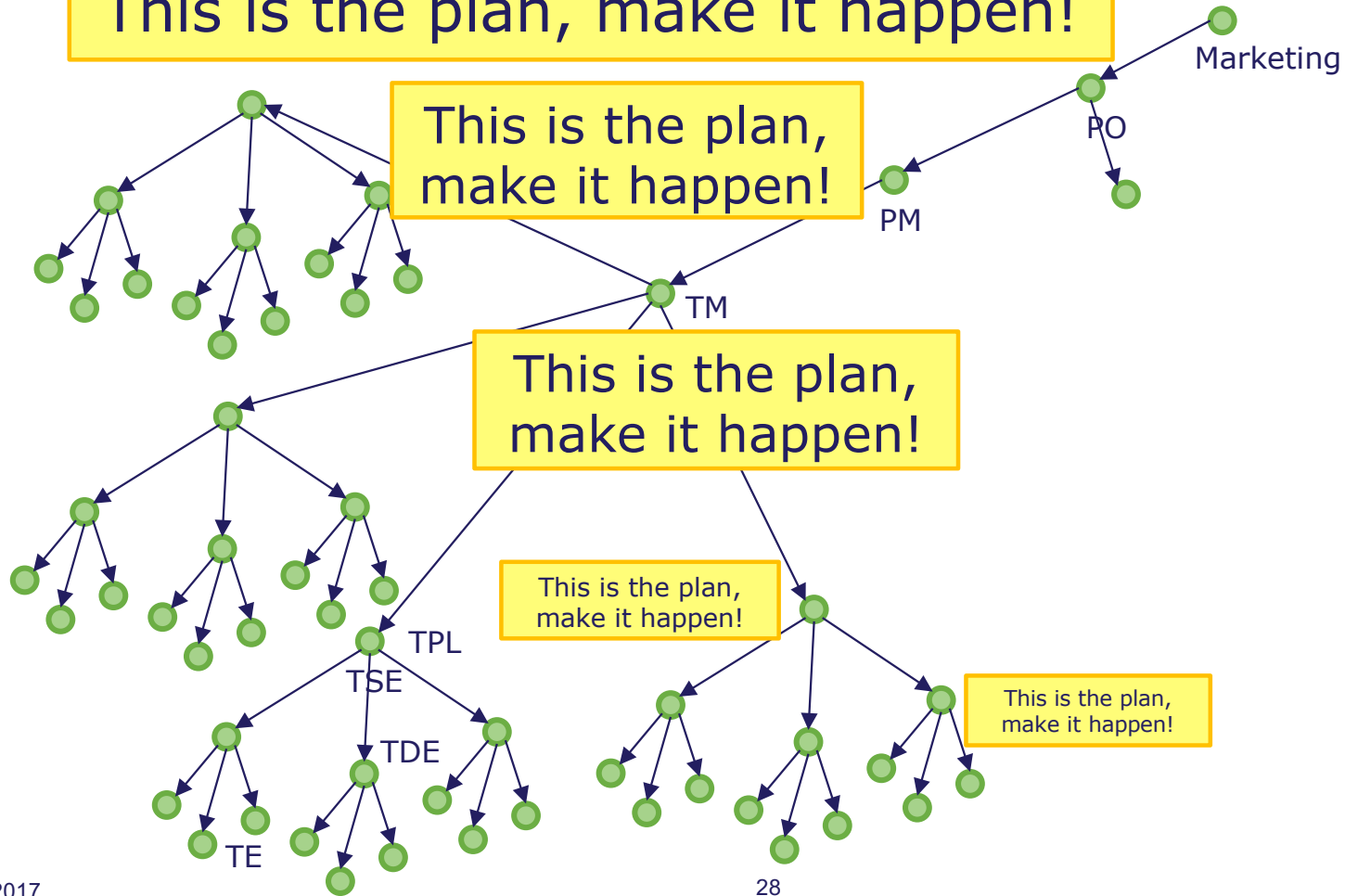
Het model geeft een aantal voorbeelden van deliverables (PSA, STP, STD, STR) maar probeert vooral niet volledig te zijn en legt nadruk op testdeliverables. Bij scrum wordt veel gebruik gemaakt van een Definition of Done (DoD), waarin staat wat allemaal opgeleverd en uitgevoerd moet zijn voordat een sprint klaar ("done") is. Dit is een goede koppeling om eisen aan leveranciers testen aan op te hangen.

- In blauw zijn testrollen die partijen getrokken die globaal verantwoordelijk zijn voor de verschillende team- en activiteiten.



This is the plan, make it happen!

This is the plan, make it happen!



## Kink 10

# De ivoren toren

# Ivoren toren





WASHINGTON — On Monday, surrounded by other white men, President [Donald Trump](#) signed [an anti-abortion executive order](#) that has far-reaching consequences for women's reproductive health access worldwide.





*Daal af!*



# Hoe houdt/breng je verbetering op gang?

- Laat je niet in de luren leggen en *wordt deel van de oplossing*
- *Maak tastbaar* hoe de cultuur een organisatie parten speelt
- Maak mensen bewust van hun *probleem*
- Heb realistische verwachtingen, *dus geduld*
- *Timing is alles*
- Zoek je weg langs *stepping stones* met direct nut
- Laat je niet op het verkeerde been zetten en *explore more*
- Voorkom grote plannen met *agile improvement stories*
- Gebruik *plaatjes*
- *Daal af* uit de ivoren toren!



# Bedankt!

[kees.blokland@polteq.com](mailto:kees.blokland@polteq.com)

