



---

Peter Wanders  
De Black Box Dialog methode  
Voorjaarsevent Testnet: 22 juni 2009

### Samenvatting

Nog nooit heb ik heb een klant horen zeggen: “Enorm vervelend dat het IT project zo is uitgelopen, maar wel fijn dat het testproces zo goed gepland was”.

Ofwel: Al is er getracht het testproces zo goed mogelijk te plannen, voor het succes is men afhankelijk van de planning van het totale IT project. Gedurende de EuroStar 2008 conferentie heb ik de nodige sprekers aangehoord over de noodzaak om testers zo vroeg mogelijk in het ontwikkelproces te betrekken. Spreker Erik Boelen liet daar in zijn toekomstvisie de term Test Driven Analysis vallen als iets waar we naar toe moeten groeien. Hiermee doelde hij erop dat testers reeds bij het opstellen van de requirements betrokken dienen te zijn. Zo wordt ervoor gezorgd dat de formulering van de Wat-vraag vroegtijdig voldoende helder wordt gemaakt. De Black Box Dialog (BBD) Methode is een concrete invulling van deze wens maar gaat een stap verder. De BBD Methode is een eenvoudige methode om tot een zeer concrete formulering van de Wat-vraag te komen, zonder de noodzaak tot het maken van de (altijd lastige) requirements die nu gebruikt worden.

Groot voordeel is dat je met BBDs echt helder voor ogen hebt wat er concreet er gebouwd moet worden. En dat stelt je in staat zowel een project- als testplanning af te geven met voldoende nauwkeurigheid.

### Biografie

Peter Wanders is fysicus en is 10 jaar geleden in dienst van de KLM gekomen. Sinds twee jaar is hij werkzaam als Test Consultant. De eerste 8 jaar was hij programmeur/analist, waarvan de meeste tijd op de Visual Load Control (VLC) applicatie. Dit is KLM's zelfgebouwde Weight & Balance applicatie. Met deze bedrijfskritische applicatie worden beladingsplannen voor vliegtuigen gemaakt, om deze zo goed mogelijk in balans te laten opstijgen, vliegen en landen. Kwaliteit en testen van deze applicatie staan uiteraard op een hoog niveau.

Ten behoeve van het reproduceren van defects is in het VLC ontwikkelteam een tool gemaakt om alle livedata van en naar achterliggende KLM mainframe systemen te kunnen opnemen. Het naspelen en oplossen van het

defect kon zo op een later tijdstip, onafhankelijk van de achterliggende KLM mainframe systemen worden gedaan. Gefascineerd door de kracht van het record and playback principe voor testen, wilde hij dit principe ook toe kunnen passen op de data-uitwisseling tussen User Interface en Business Laag van applicaties. Van 2003 tot kort geleden heeft hij dit idee uitgewerkt en dit leidde uiteindelijk tot de Black Box Dialog Methode.



## De Black Box Dialog Methode

---

TestNet Presentatie

Peter Wanders

22 juni 2009



## Software bouwen is echt moeilijk

---

- bouwen wat de klant nodig heeft
- binnen afgesproken tijd en budget



## Kans van slagen is hoger naarmate het ontwikkelteam:

---

- het businessproces beter kent
- beter met klant communiceert en hem betrekt
- meer echt goede vaklui bevat
- er gedisciplineerder wordt gewerkt
- ruimer in tijd en budget zit



## Hoe krijgen we de kans van slagen zo hoog mogelijk als niet alles optimaal is?

---

met ondersteuning van een methode

Maar, zijn die er al niet?



## Laten we kijken naar de evolutie in ontwikkelmethoden

---



## Een blik in het verleden ...

---

- MainFrame tijd:
  - Waterval methode
  - uitgebreide specificaties
  - degelijke werkprocedures
  - duur en langzaam

## Een blik in het verleden ...

- PC tijdperk / OO programmeren:
  - goedkopere applicaties
  - minder uitgebreide specs
  - minder strakke werkprocedures
  - een ander soort ontwikkelteams
  - van waterval => agile

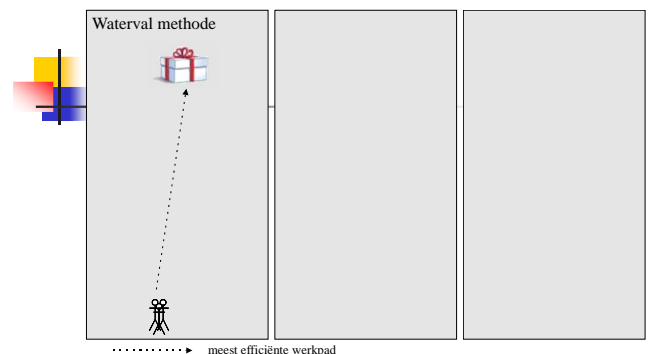
## Waarom werd Agile groter ten koste van Waterval?

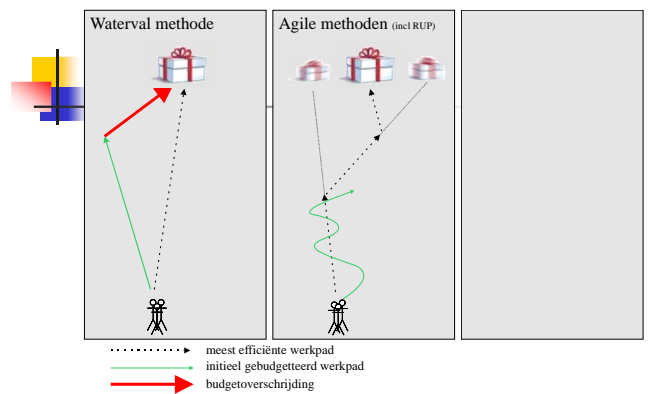
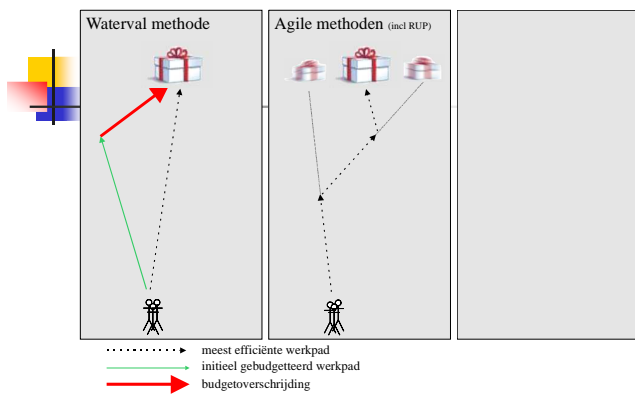
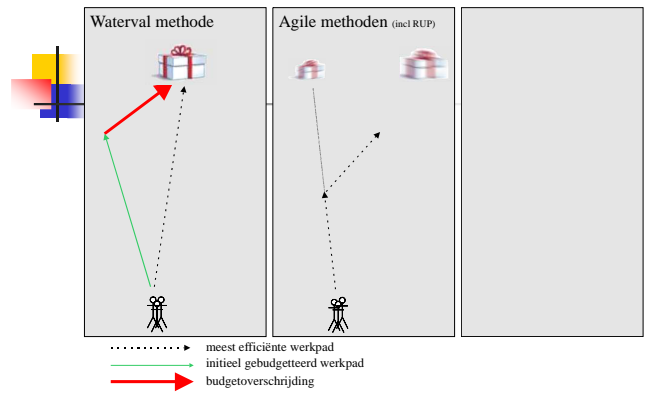
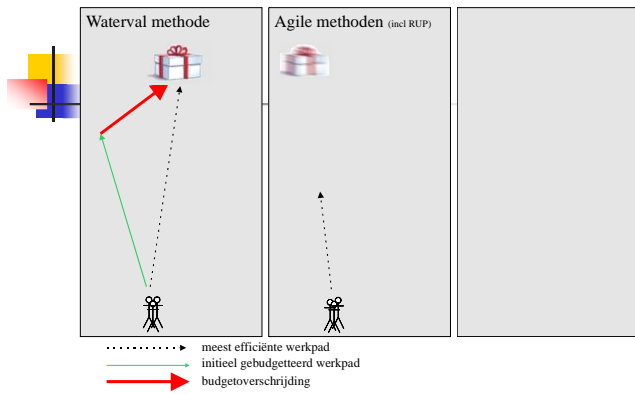
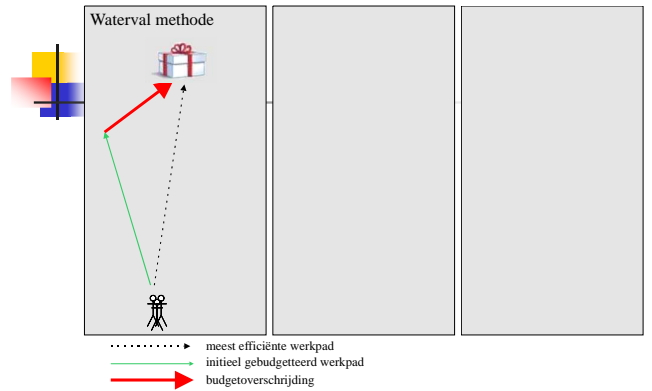
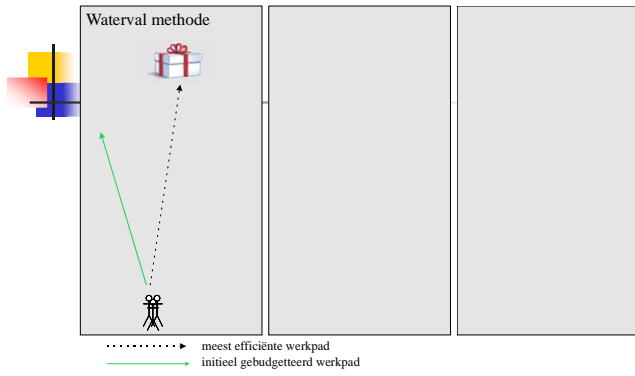
- Waterval
  - + helder beeld (bij ontwikkelteam) van het eindproduct
  - + dus project redelijk goed te plannen
  - opgeleverd eindproduct niet altijd wat klant wilde

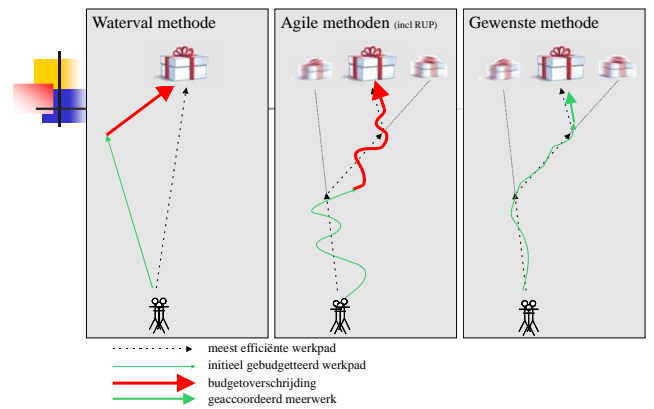
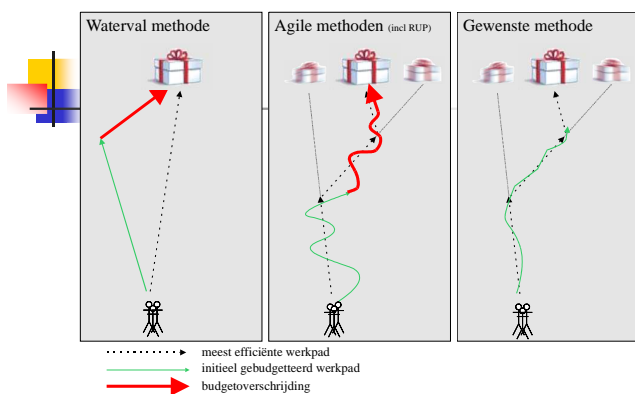
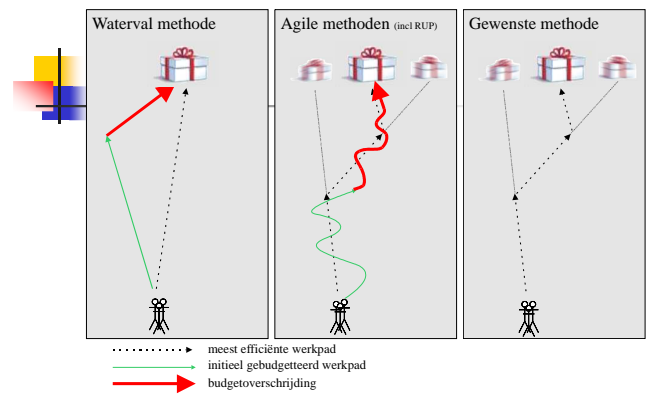
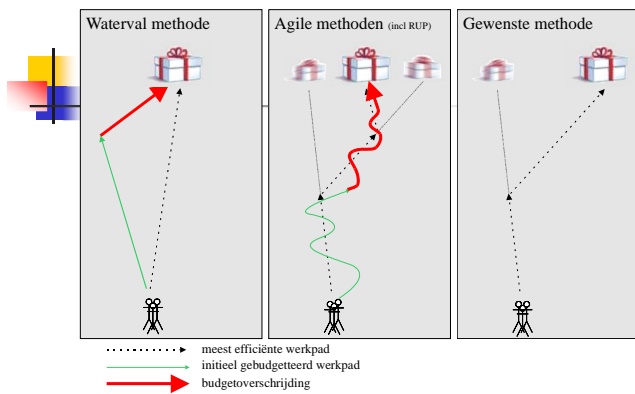
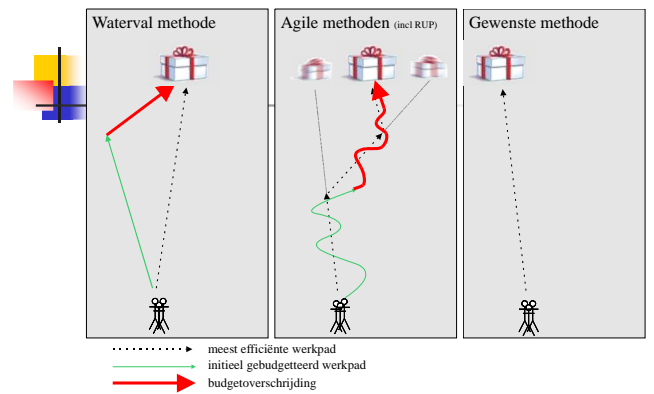
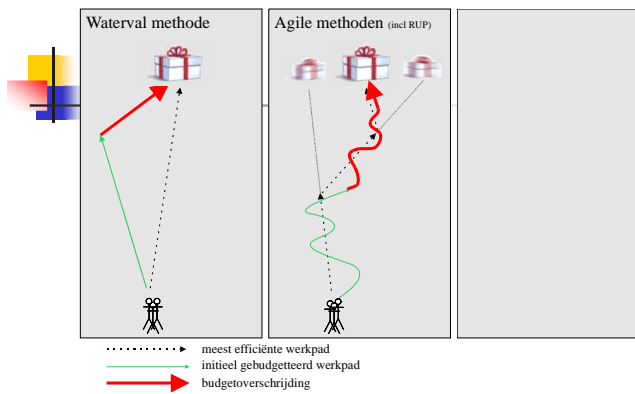
## Waarom werd Agile groter ten koste van Waterval?

- Agile
  - + de betrokkenheid van alle stakeholders maakt tijdig bijsturen richting bruikbaar eindproduct mogelijk
  - Onderweg bepalen wat je gaat maken, maakt plannen vooraf heel moeilijk

Grafisch weergegeven wordt dit:









## Waar moet een methode aan voldoen om dit te kunnen?

- de kennis van alle betrokkenen is even belangrijk
- de methode moet een taal leveren die iedereen kan spreken – van eindgebruiker tot ontwikkelaar
- de methode moet zich aanpassen aan de *intuïtieve* denkwijze van alle betrokkenen



## Wat maakt het software ontwikkelen nu dan zo moeilijk?

Dit leg ik uit aan de hand van het Requirements Review Dilemma



## Statisch-beschrijvende functionele requirements (I)

### Kenmerken:

- ze zijn beschrijvend opgesteld, in de derde persoon enkelvoud
- ze zijn tijdloos, bevatten geen procesflow informatie

format: "Het systeem moet *zus of zo* kunnen"



## Statisch-beschrijvende functionele requirements (II)

- Ze zijn vaak multi-interpretabel
- Fundament waarop iedereen in het ontwikkelproces zich baseert



## Van Requirements tot Model

- Uit Requirements document volgen de specificatiedocumenten (F.O. + Use Case beschrijvingen + UML Activity diagrams + andere UML producten)
- Abstract en academisch
- Kloof met denkwereld van gebruikers (concreet denken + processflow kennis)



## Wat is het Requirements Review dilemma?

- Dilemma voor opdrachtgever:  
Moet hij de opdracht tekenen?
  - Requirements-document niet concreet genoeg voor eindgebruikers, dus niet reviewbaar
  - Om ook maar iets concreets en reviewbaars te krijgen moet hij eerst tekenen

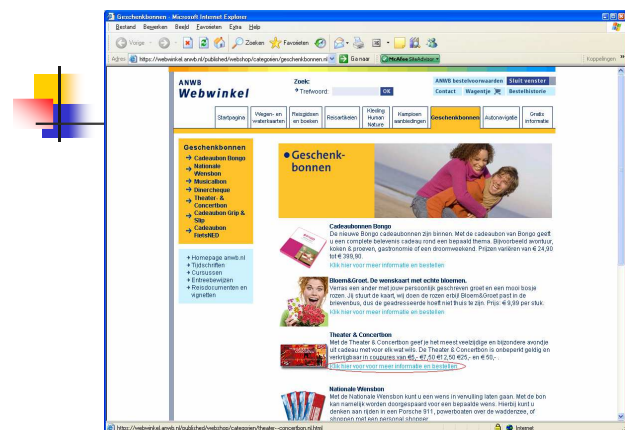
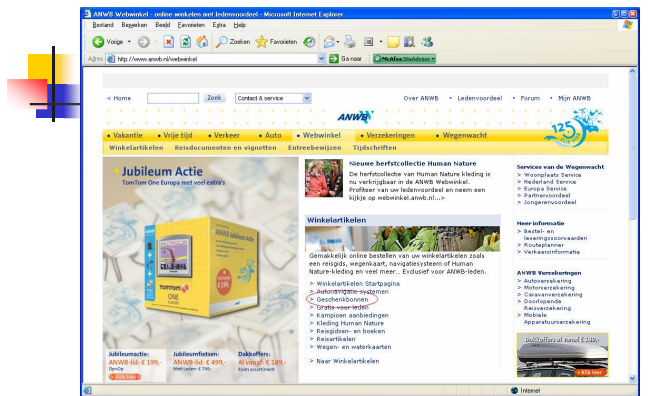
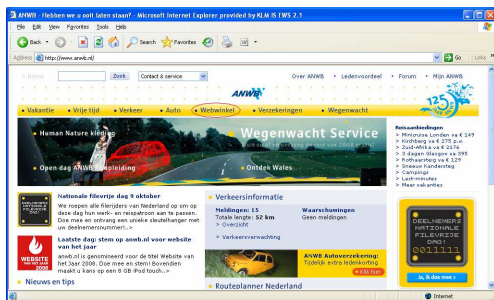
# De Black Box Dialog Methode

kan het Requirements Review Dilemma oplossen.

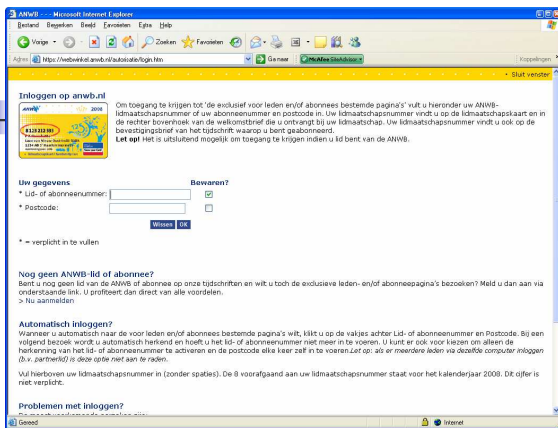
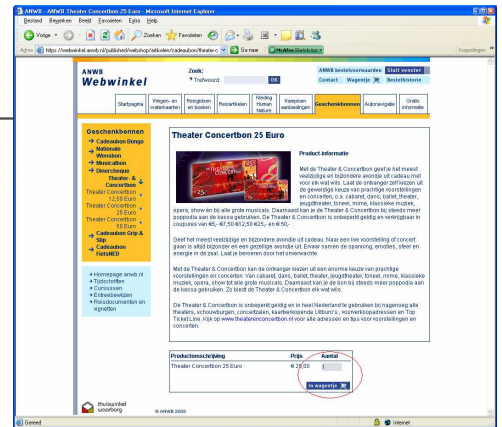
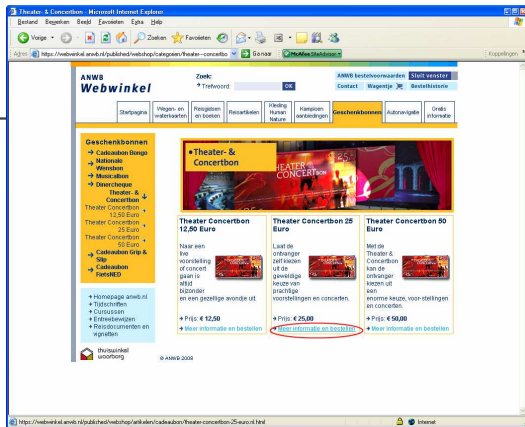
# Wat zijn Black Box Dialogs?

En hoe voldoen ze aan de hiervoor gestelde methode-eisen?

# Een stukje praktijk...







## Anders kijken naar Functionele requirements

- Hoe kan je functionaliteit als het kopen van een theaterbon anders beschrijven dan in statisch-beschrijvende functionele requirements?
- Ofwel:
  - Concreet
  - In taal van eindgebruikers
  - Ook handzaam voor IT-ers

?

## Door te kijken naar de dialoog

tussen de gebruiker en het systeem (als Black Box).

## Door te kijken naar de dialoog

tussen de gebruiker en het systeem (als Black Box).  
(Maar hoe leg je het vast...)

## Fundamenteel kijken naar dialogen (I)

- In een kantoor:



Wendy



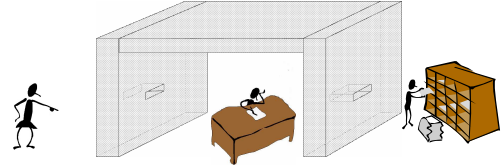
Bob



Alice

## Fundamenteel kijken naar dialogen (II)

- Geen mondelinge communicatie meer



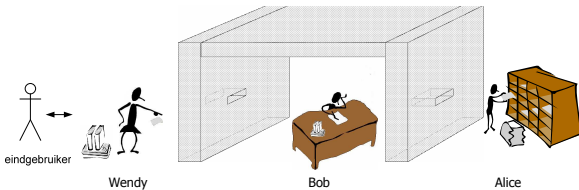
Wendy

Bob

Alice

## Fundamenteel kijken naar dialogen (III)

- Instructies en parameter sheets => Maximale restricties



Wendy

Bob

Alice

## BBD kern: Parameters en Instructies scheiden

- Communicatie tussen systeemdelen op basis van parameter sheets
- Intelligentie beleggen per systeemdeel
- TVB per systeemdeel helder
- Minimale afhankelijkheden tussen systeemdelen

## Maar hoe werkt het in de praktijk?



## Voorbeeld

- Geldautomaat dialogen als requirements
  - Black Box Dialogs (Wat-vraag)
  - System dialogs (IT-Wat-vraag)

## Stap 1

Bepaal alle Taken / Use Cases die je met de software wilt uitvoeren

+  
Begin met verzamelen van business rules per Use Case

Doe dit met:

- Business analyst
- Business Management



A	B	C	D
1	Use Case overzicht met Business rules		
2			
3			
4			
5	1	UC1BR1	Alleen TopBank klanten mogen aanloggen
6		UC1BR2	Na 3 keer verkeerde PinCode wordt pas door apparaat ingeslikt
7		UC1BR3	
8	2	UC2BR1	Mag onbeperkt aantal malen per dag gedaan worden
9		UC2BR2	
10	3	UC3BR1	Klant kan maximaal 1000 euro per dag opnemen
11		UC3BR2	Klant mag maximaal 3 keer per dag een bedrag opnemen
12		UC3BR3	
13	4	UC4BR1	Kan alleen naar binnenlands rekeningen
14		UC4BR2	Naar TopBank rekening is bedrag direct overgemaakt
15		UC4BR3	

## Stap 2

Inventariseer mogelijke dialogen  
(in oplopende complexiteit)  
per Use Case

Doe dit in een workshop samen met:

- Eindgebruikers
- Business analyst
- Test analyst



### Mogelijke dialogen voor Use Case "inloggen":

A	B	C	D
1	Dialogenoverzicht		
2	1	Succesvolle login	
3	2	Mislukte login door gebruik van niet-bankpas	
4	3	Succesvolle login met 1 keer foute Pincode	
5	4	Succesvolle login met 2 keer foute Pincode	
6	5	Afgebroken login na 1 (of 2) keer foute pincode	
7	6	Mislukte login met 3 keer foute pincode	
8			

## Terugkoppelmoment

overleg met opdrachtgever over  
wat er verder uitgewerkt moet  
worden



## Stap 3

Werk per Use Case de dialogen uit  
en maak het parameter sheet.

## Uitwerking dialog 1

A	B	D	E	F	
15	Dialog 1	Successvolle login			
16	Draaiboek stap				
17	Wendy	Bob	Beschrijving	resultaat actie	Parametersheet wijziging
18	10		(Wendy start use Case door Pin Pas in geldautomaat te steken)		
19			Wendy maakt nieuw Pasheet en vult pasgegevens in		pasgegevens
20			Bob controleert of het een TopBank bankpas is	OK	
21			Bob geeft Pasheet aan Wendy voor Pin Code		
22	30		Wendy vult pincode in op Pasheet		Pincode
23			Bob controleert of het de pincode + pas nummer OK is bij bank	OK	
24			Bob geeft aan "Login OK" en Wendy mag nu de "geld opnemen" en "Saldo bekijken" use Cases starten		
25	50				
26					

## Het parameter sheet

A	B	C
Paramsheet	Login	
	Pasgegevens	
	Bank	
	rekening nummer	
	Pasnummer	
	Geldig Tot	
	PinCode	
	UserMessages	
	ErrorMessage	
	Message	

## Uitwerking dialog 2

A	B	D	E	F	
26	Dialog 2	Mislukte login (niet bankpas)			
27	Draaiboek stap				
28	Wendy	Bob	Beschrijving	resultaat actie	Parametersheet wijziging
29	10		(Wendy start use Case door Pin Pas in geldautomaat te steken)		
30			Wendy maakt nieuw Pasheet en vult pasgegevens in		pasgegevens
31			Bob controleert of het een TopBank bankpas is	Niet OK	
32			Bob vult in op Pasheet de ErrorMessage: "Dit is geen bankpas"		Message: "Dit is geen bankpas"
33			Bob geeft Pasheet aan Wendy		
34			(geldautomaat geeft pas terug)		
35	21		Wendy leest melding (en neemt pas uit geldautomaat)		
36			Welkomsscherm wordt getoond		
37					
38					
39					

## Stap 4

Vul parameterwaarden in voor de dialogen

## Dialogen met parameter waarden ingevuld:

27					
28	Dialog 2	Mislukte login door gebruik van niet-bankpas			
29	Draaiboek stap				
30	Wendy	Bob	Beschrijving	resultaat	Parametersheet wijziging
31	10		(Wendy start use Case door Pin Pas in geldautomaat te steken)		
32			Wendy maakt nieuw Pasheet en vult pasgegevens in		pasgegevens
33			Bob controleert of het een TopBank bankpas is	Niet OK	
34			Bob vult in op Pasheet de ErrorMessage: "Dit is geen bankpas"		Message: "Dit is geen bankpas"
35			Bob geeft Pasheet aan Wendy		
36			(geldautomaat geeft pas terug)		
37	21		Wendy leest melding (en neemt pas uit geldautomaat)		
38			Welkomsscherm wordt getoond		
39					
40					
41	<b>testdata voor dialogcollectie tm dialog 2</b>				
42	Bank	rekening	Geldig tot	PinCode	verwacht resultaat melding
43	TopBank	1234567	1-Jan-11	1234	Ingelegd
44	RaboBank	0730436	1-Jan-11	7369	ErrorMelding: Dit is geen TopBank pas
45	TopBank	1234567	<Vandaag - 1	1234	Ingelegd
46	TopBank	1234567	<Vandaag	1234	Ingelegd
47	TopBank	1234567	<Vandaag + 1	1234	ErrorMelding: Deze pas is niet meer geldig - insame pas
48	TopBank	1234567	1-Jan-11	1111	ErrorMelding: Foute Pincode
49					

Hiermee hebben we een ondubbelzinnig gespecificeerde WAT vraag

Net zoals je zou willen wanneer je je eigen huis laat bouwen

## Met de BBD's de IT in (I)

Laat IT bedrijven offerte maken

## Met de BBD's de IT in (II)

Voor de offerte maakt het IT bedrijf eerst *System dialogs* uit de BBD's

## Stap 1

Breid het parametersheet uit  
(maak het systeem breed)

## Parametersheet – Systeem breed

Parametersheet	Login
	PasGegevens
	Bank
	rekening nummer
	Pasnummer
	Geldig Tot
	PinCode
	UserMessages
	ErrorMessage
	Message
	Connectegegevens Bank
	IP Address
	login naam
	password
	connectieTime-out in secs

## Stap 2

Combineer alle dialogen in een  
*Black Box Dialogs collection*

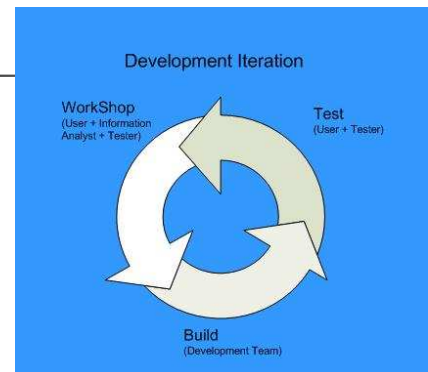
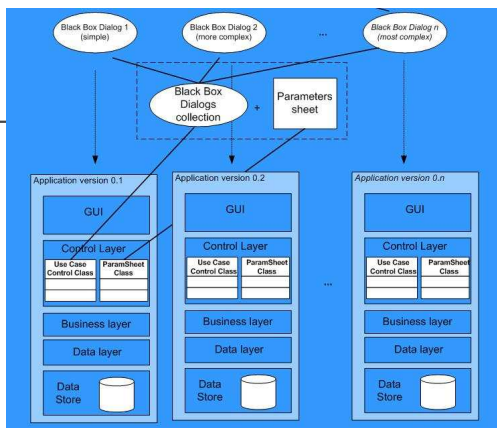
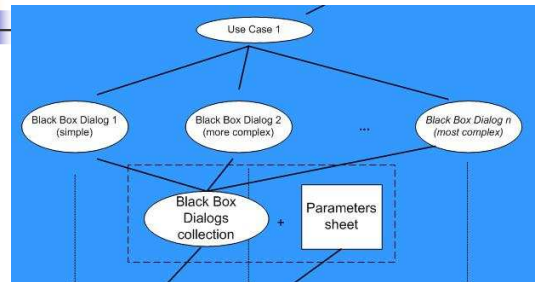
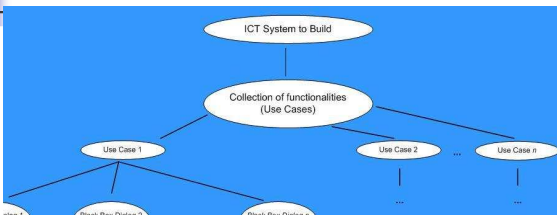
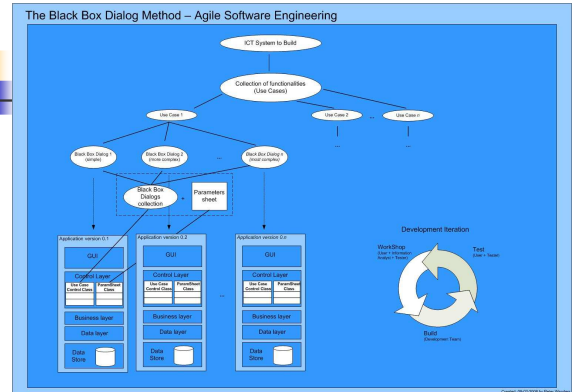
## Collectie van alle mogelijke Black Box Dialogs

	A	B	C	D	E
1	<b>Draaiboek voor Log-In dialoog</b>				
2					
3	<b>Draaiboek stap Nummer</b>	<b>Actie</b>	<b>Beschrijving</b>		<b>Exception flow</b>
4	Wendy	Bob	Bank	Wendy start use Case door Pin Pas in geldautomaat te steken	
5				Wendy maakt nieuw PinSheet en vult pasgegevens in	
6				Bob controleert of het een TopBank bankpas is	
7		10			Geen TopbankPas
8					Bob vult in op PinSheet de ErrorMessage: "Dit is geen bankpas"
9					Wendy leest melding en neemt pas uit ATM - <b>einde dialoog</b> - Welkomsscherm wordt getoond
10		11		Wendy vult pincode in op PinSheet of drukt op "Stoppen"	
11		20		Bob verhoogt PIN pogingstelsel	
12		30			3 keer fout Pin Code
13					Bob geeft opdracht pas in te nemen - <b>einde dialoog</b> - Welkomsscherm wordt getoond
14				Bob controleert of de pincode + pas nummer OK is bij bank	
15			X		Foutte Pincode
16					Bob vult in op PinSheet de ErrorMessage: "De pin code is fout. Je hebt nog <3 poging(en) pogingen"
17					Ga verder met stap 30
18				Bob slaat sessie informatie op in de Engine	
19				Bob geeft aan "Login OK" en Wendy mag nu de "geld opnemen" en "Saldo bekijken" Use Cases starten	
20		40			
21					

### Stap 3

Laat de bouwers de boel concreet maken!

- Iteratief werken – per iteratie een afgesproken aantal dialogen
- User interface wordt bepaald
- Stubs en drivers voor maximale specificatie snelheid

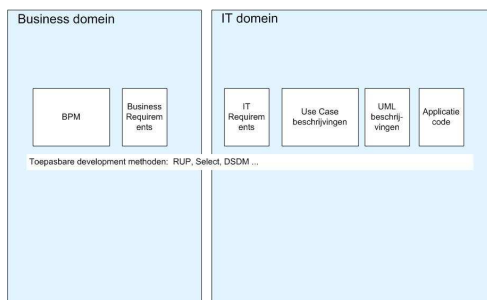


## Is dit nieuw?

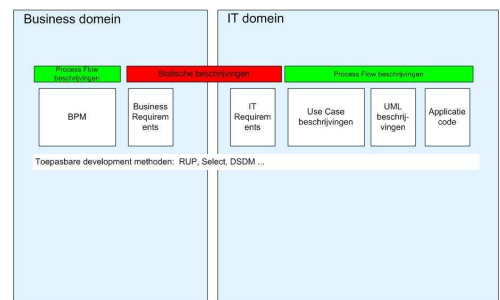
- Use case uitwerkingen bevatten tegenwoordig:
  - Algemene beschrijving,
  - Activity Diagram,
  - Beschrijvingen van
    - Basic Flows,
    - Alternative Flows,
    - Error Flows.

Wat is dan de impact van BBDs op het huidige software ontwikkelproces?

## Vergelijking ontwikkelmethoden (I)



## Vergelijking ontwikkelmethoden (II)



## Weg met de *Statisch-beschrijvende Functionele Reqs*

Ze worden vervangen door:

- Black Box Dialogs +
- "parameter waarden met Expected results tabellen"

Geen multi-interpretabele functionele requirements meer!

## Dialogen met parameter waarden ingevuld:

27	Dialog 2	Mislukt login door gebruik van niet-bankpas	Logisch Test geval
28			
29			
30	Onaafbeëindigd		
31	Wendy Bob	Beschrijving	resultaat Parametersheet wijziging
32	10	(Wendy start use Case door Pin Pas in geldautomaat te steken)	
33		Wendy maakt nieuw Pasheet en wilt pasgegevens in	pasgegevens
34	20	Bob controleert of het een TopBank bankpas is	Niet OK
35		Bob valt in op Pasheet de ErrorMessage "Dit is geen bankpas"	Message: "Dit is geen bankpas"
36		Bob geeft Pasheet aan Wendy (geldautomaat geeft pas terug)	
37		Wendy leest melding (en neemt pas uit geldautomaat)	
38	21	Welkomsscherm wordt getoond	
39			
40			
41	testdata voor dialogcollectie tm dialog 2		
42	Bank rekening	Geldig tot	PinCode verwacht resultaat melding Business rule
43	TopBank 1234567	1-Jan-11	1234 Ingelogg
44	FokoBank 8735436	1-Jan-11	7598 ErrorMessage: Dit is geen TopBank pas
45	TopBank 1234567	<Vandaag> - 1	1234 Ingelogg
46	TopBank 1234567	<Vandaag>	1234 Ingelogg
47	TopBank 1234567	<Vandaag> + 1	1234 ErrorMessage: Deze pas is niet meer geldig - innama pas
48	TopBank 1234567	1-Jan-11	1111 ErrorMessage: Foute Pincode
49			
40	* * * * * \UseCase\Login / Login_ParamSheet / Login_draiboek / GeldOpnemen / GeldOpnemen_draiboek / ...		

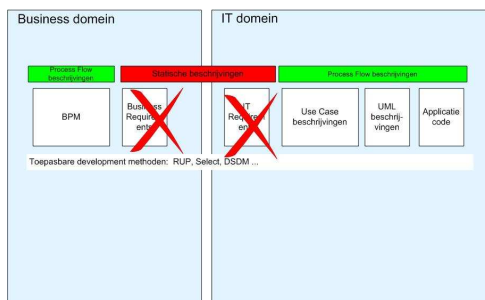
## BBD Methode = Test Driven Analysis

Want:

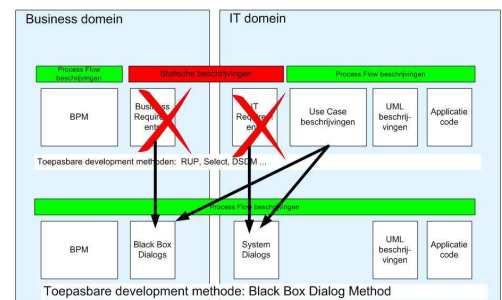
- Black Box Dialogs = logische testgevallen
- + "parameter waarden met Expected results tabellen" => fysieke testgevallen

N.B. non-functional requirements blijven zoals ze nu zijn!

## Vergelijking ontwikkelmethoden (III)



## Vergelijking ontwikkelmethoden (IV)



## Samenvatting BBD Methode

- Gebruik bij nieuwbouw van applicaties
- Requirements bepalen in workshops
- BBD's per Use case als regielaag in applicatie
- Iteraties eenvoudiger te bepalen
- Deels beproefd in praktijk

## Samenvatting – gebruik van BBD Methode leidt tot:

- Kans op ontevreden gebruikers ↓
  - Taalgebruik
  - Totale Processen benadering
- Testen
  - Kwaliteit ↑
  - Ultieme "test driven development"
- Onderhoudskosten ↓
  - Complexiteit ↓
- Portabiliteit ↑
- Iteraties makkelijker te bepalen in agile development





## Samenvatting - vervolg

Gevolg:

- Uitloop planning ↓
- Kosten ↓
- Kwaliteit ↑



## Statement:

Bij gebruik van de BBD Methode in een nieuwbouw project krijg je meer waar voor minder geld en sneller.



## Wat nu verder?

In september/october organiseert BlackBoxDialog.org workshops waarin deelnemers concreet met BBD Methode aan de slag gaan.

Geïnteresseerd?

Meld je aan door mail te sturen naar [BBDworkshops@blackboxdialog.org](mailto:BBDworkshops@blackboxdialog.org)



## Niet alles duidelijk?

Lees het verhaal rustig na op [www.blackboxdialog.org](http://www.blackboxdialog.org)

of stuur een mail naar:

[peter.wanders@blackboxdialog.org](mailto:peter.wanders@blackboxdialog.org)