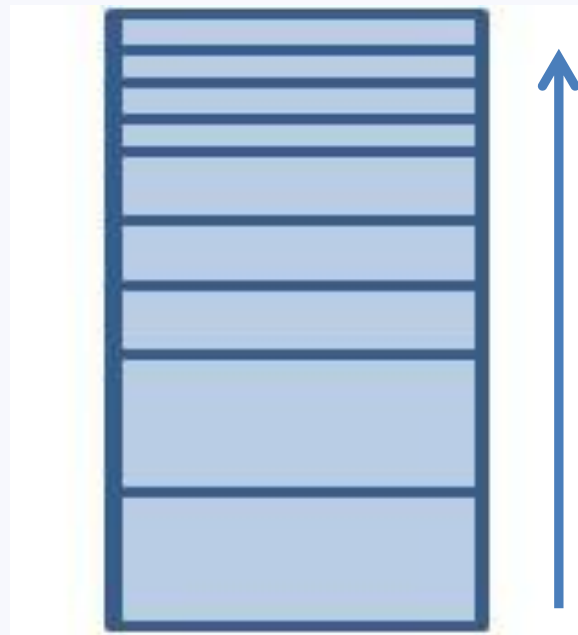




Het prioriteren
van de



best moeilijk!

Product Backlog

Prioriteren van de Product Backlog

Best moeilijk!

- Klanten willen vaak niet prioriteren
 - alles is belangrijk!
- Product managers willen vaak niet prioriteren
 - alles krijgen = maximale value!
- Waarde van een item bepalen is moeilijk

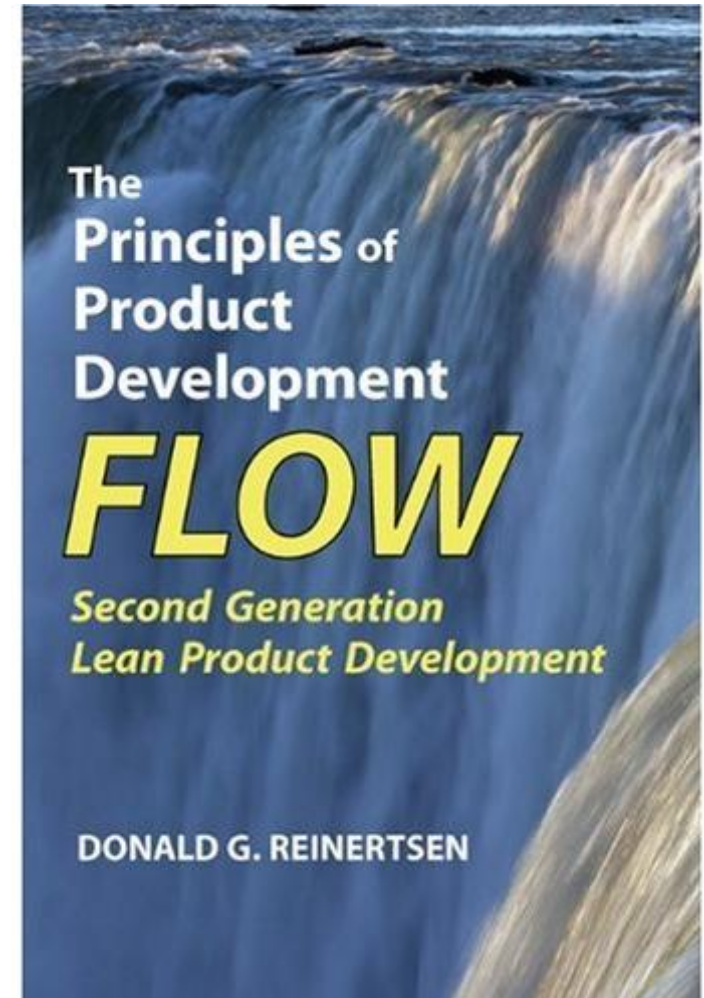
Prioriteren van de Product Backlog

Lastige waarde bepaling:

- Sommige items zijn “must haves”
 - om concurrerend te blijven
 - om marktaandeel te behouden
 - hoe bepaal je de waarde van die items?
- Sommige items zijn lastig te vergelijken
 - een nieuwe feature die vele maanden kost
 - een kleine aanpassing die twee dagen kost

Gelukkig zijn er bruikbare principes

Donald Reinertsen



Gelukkig zijn er bruikbare principes

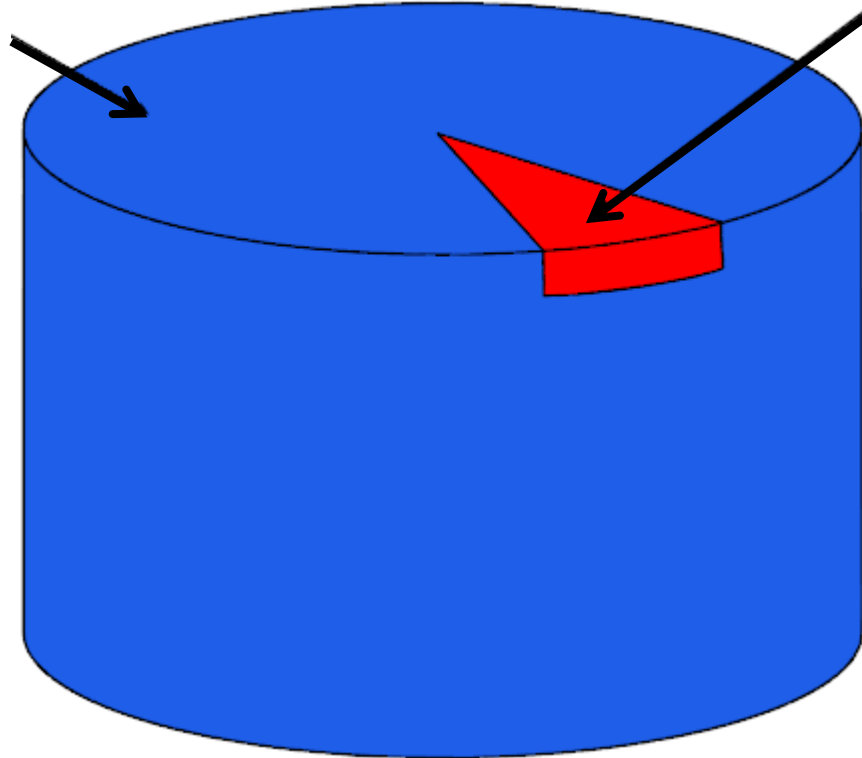
Donald Reinertsen

- Economic Principles (21)
- Queuing Principles (16)
- Variability Principles (16)
- Batch Size Principles (22)
- WIP Constraint Principles (23)
- Flow Control Principles (30)
- Fast Feedback Principles (24)
- Decentralization Principles (23)

Disclaimer

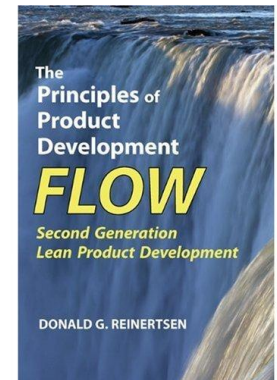
174 principes

Diepgang
van de
theorie

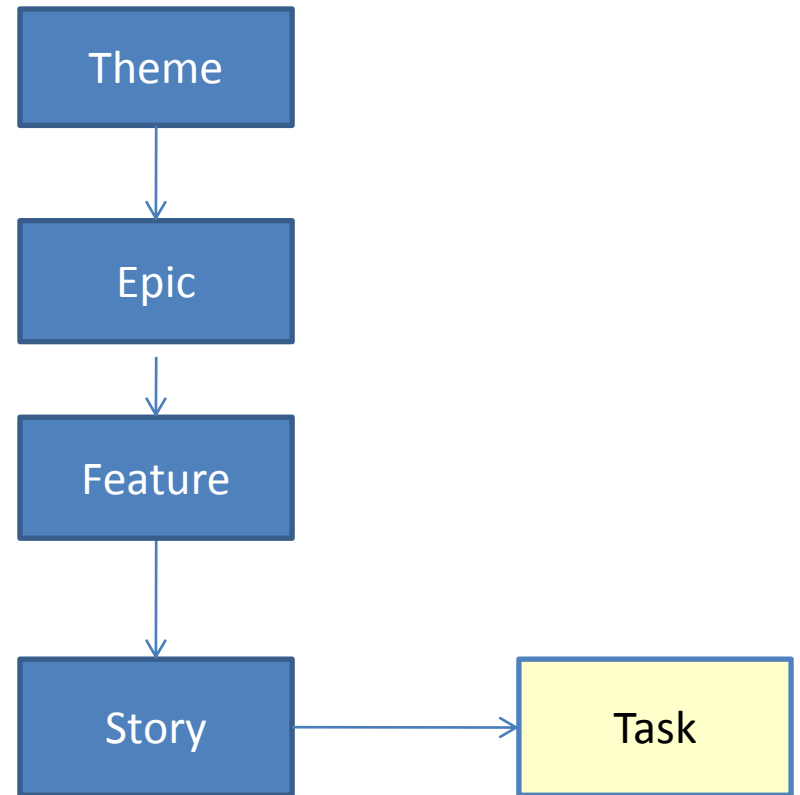
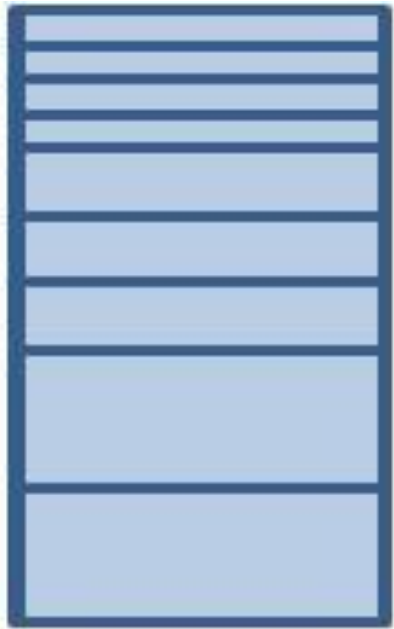


Deze
presentatie

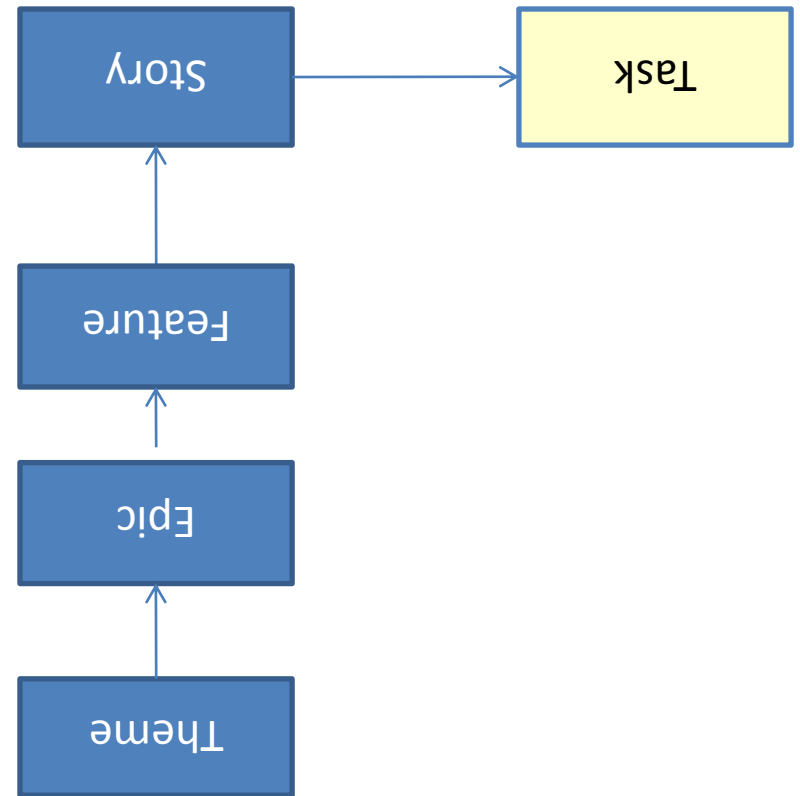
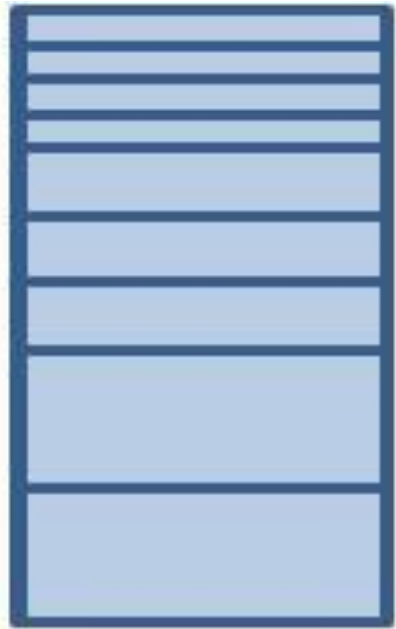
enkele
principes



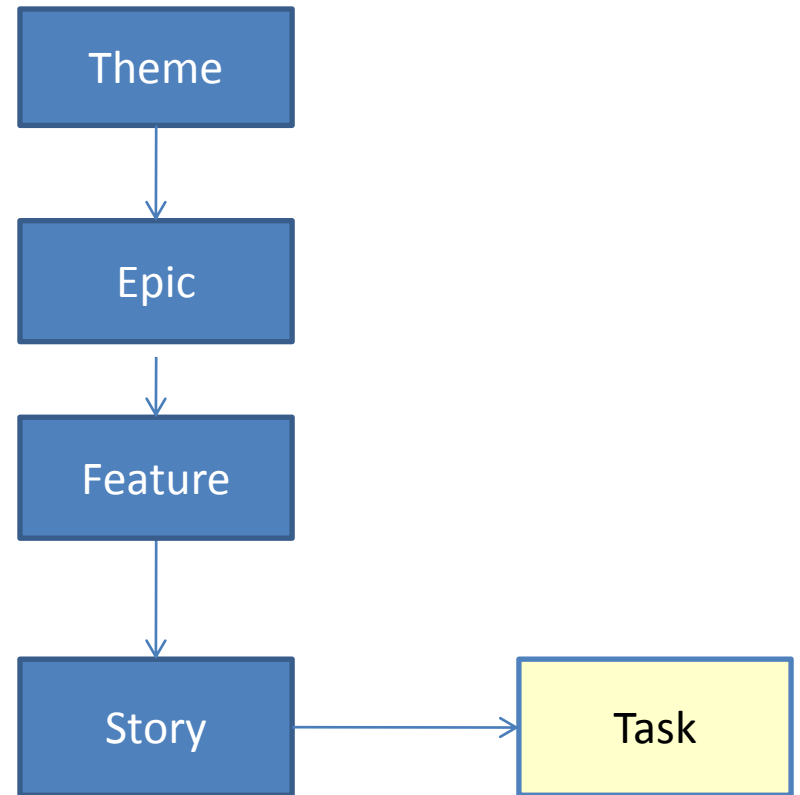
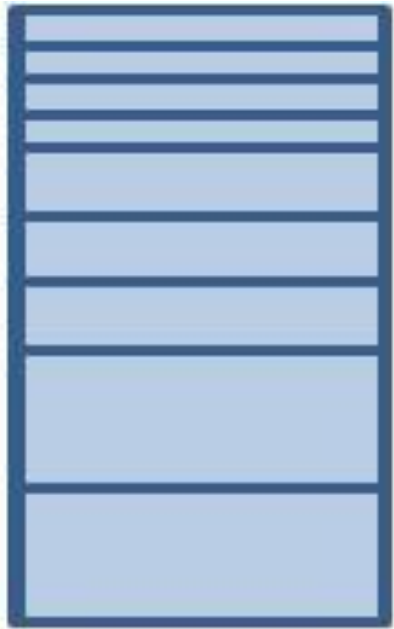
Prioriteren van de Product Backlog



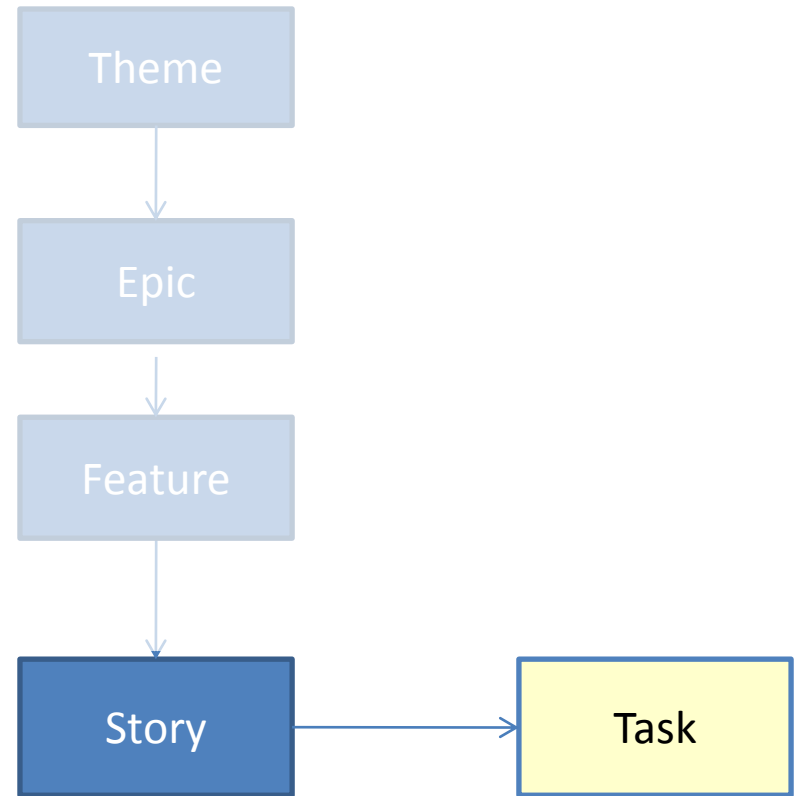
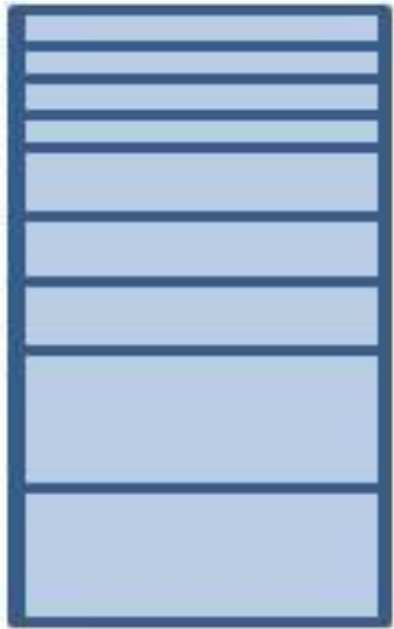
Prioriteren van de Product Backlog



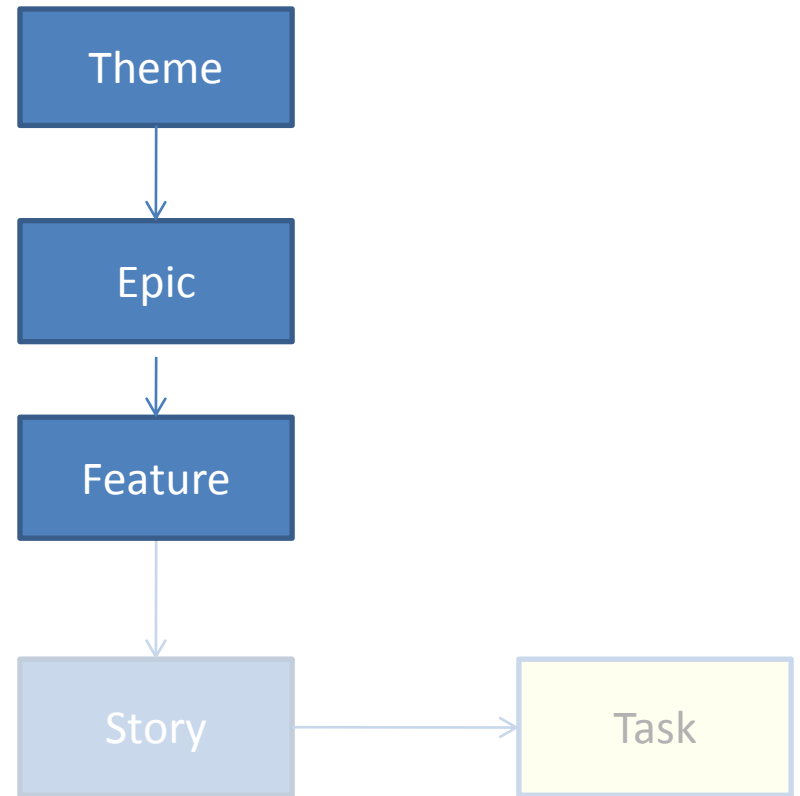
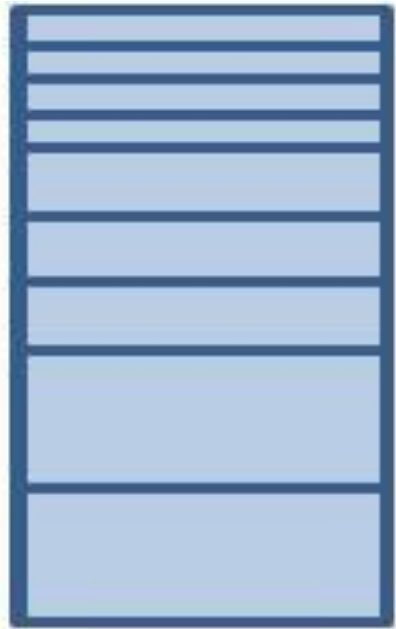
Prioriteren: *twee momenten*



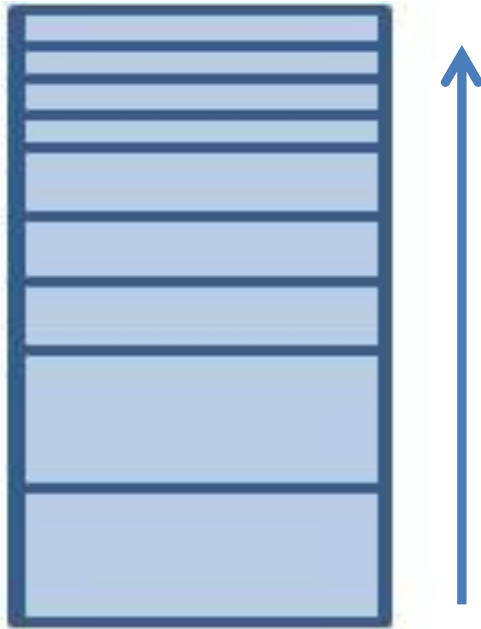
Prioriteren: *items komende sprint*



Prioriteren: *items latere sprints*

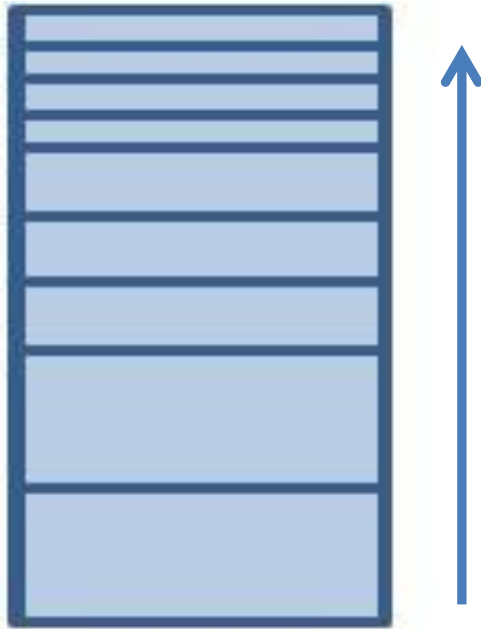


Prioriteren: *items komende sprint*



Dit is het
gemakkelijkste deel
;-)

Prioriteren: *items komende sprint*



- Doel volgende sprint bekend
- Resultaat vorige sprint bekend
- Beperkt aantal items
- Items ongeveer even groot
- Prioriteren voor korte termijn
- Impact keuzes is beperkt

Prioriteren: *items komende sprint*

Item	Business waarde	Rank
Story 1		
Story 2		
Story 3		
Story 4		
Story 5		

Prioriteren: *op Business waarde*

Item	Business waarde	Rank
Story 1	8	8
Story 2	8	8
Story 3	6	6
Story 4	4	4
Story 5	3	3

Rank = Business waarde

Prioriteren conform Scrum

Prioriteren: *op Business waarde*

Item	Business waarde	Rank
Story 1	8	8
Story 2	8	8
Story 3	6	6
Story 4	4	4
Story 5	3	3

Toch kan het beter:

- Voorrang aan items die aansluiten bij (sprint) doelstelling
- Voorrang aan items die risico's reduceren

Prioriteren: *op Business waarde ++*

Item	Business waarde	Doel stelling	Risico reductie	Rank
Story 1	8	4	2	14
Story 2	8	2	3	13
Story 3	6	2	1	9
Story 4	4	4	7	15
Story 5	3	5	8	16

Rank = Business waarde

**+ bijdrage aan (sprint)doelstelling
+ bijdrage aan risico reductie**

Sluit aan op best practice van RUP

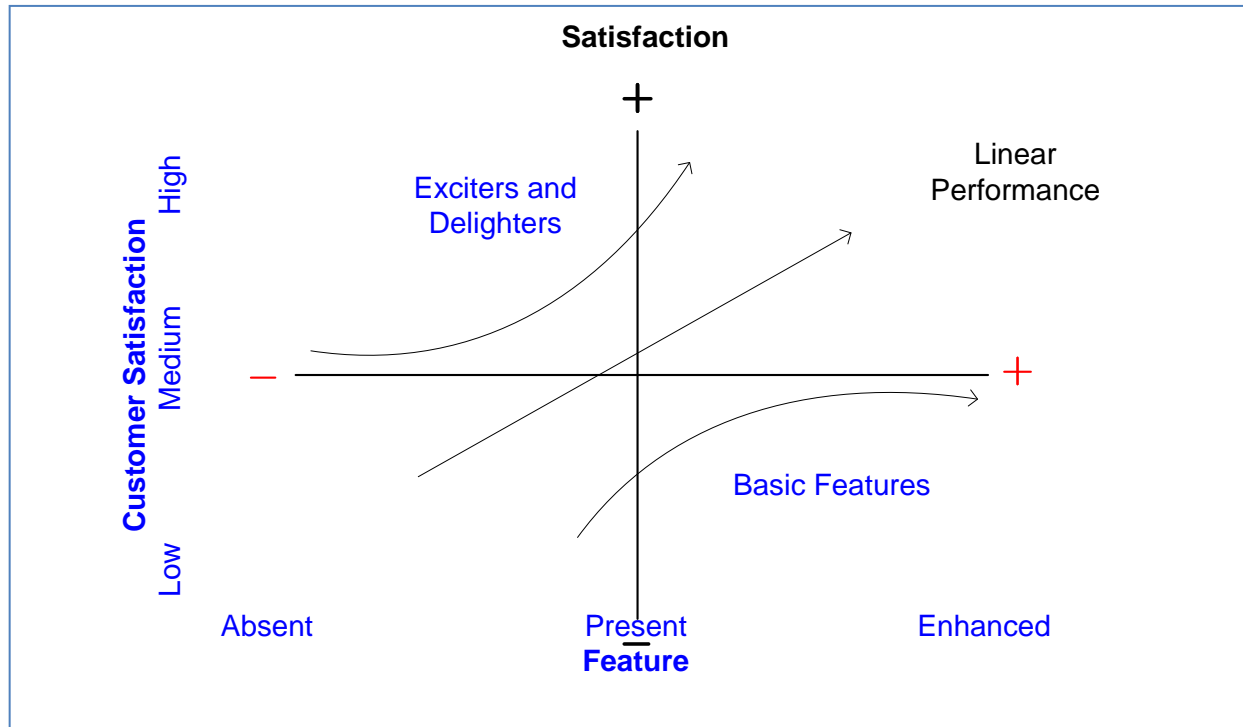
- Hier simpele optelling
- Kan ook gewogen

Prioriteren: *op Business waarde ++*

Item	Business waarde	Doel stelling	Risico reductie	Rank
Story 1				
Story 2				
Story 3				
Story 4				
Story 5				

- Hoe bepaal je eigenlijk de Business waarde?


Business waarde: *Kano model*



Noriaki
Kano

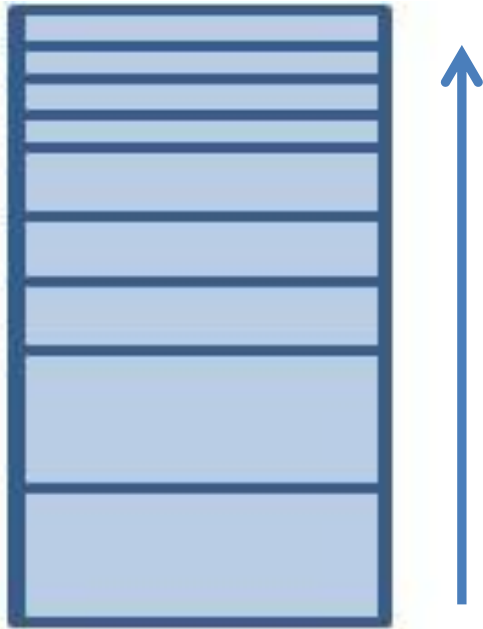
1. Eerst focus op **Basic Features**
2. Daarna focus op **Exciters & Delighters**

Wanneer rank bepalen?

Item	Business waarde	Doel stelling	Risico reductie	Rank
Story 1	8			
Story 2	8			
Story 3	6			
Story 4	4			
Story 5	3			

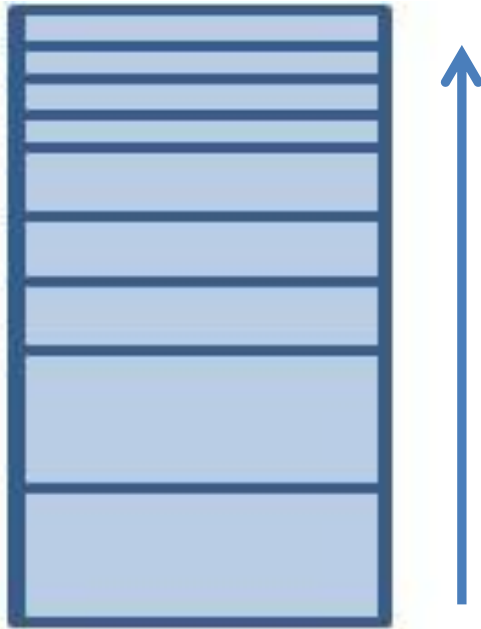
- Business waarde continu bepalen
- Doelstelling waarde bepalen als doelstelling bekend (aan het begin van een sprint)
- Risico reductie bepalen als bekend wat al is gebouwd

Prioriteren: *items latere sprints*



Dit is het
moeilijkste deel
;-(

Prioriteren: *items latere sprints*



- Veel items
- Verschillende groottes
- Niet allemaal even ver uitgewerkt
- Prioriteren voor lange termijn
- Impact keuzes kan groot zijn

Prioriteren: *items latere sprints*

Lastig business waarde te vergelijken:

- Hoe vergelijk je

- een nieuwe feature die vele maanden kost
- een kleine aanpassing die twee dagen kost

Prioriteren: *items latere sprints*

Lastig business waarde te vergelijken:


Item	Business waarde	
Feature 2	300	
Feature 3	30	

Groot item is meer waard,
maar kost ook meer Effort!

Item	Business waarde	Effort	Return on Investment
Feature 2	300	100	3
Feature 3	30	6	5

Prioriteren: *items latere sprints*

Item	Business waarde	Effort	Return on Investment	Rank
Feature 2	300	100	3	3
Feature 3	30	6	5	5


 Hoogste rank

Waarde en Effort hoeven niet precies te kloppen.
Als ze ten opzichte van elkaar maar kloppen

Relatieve rank = Relatieve ROI = Relatieve $\frac{\text{Waarde}}{\text{Effort}}$

Prioriteren: *items latere sprints*

Item	Business waarde	Effort	Return on Investment	Rank
Feature 2	300	100	3	3
Feature 3	30	6	5	5

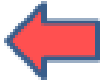
 Hoogste rank

Deze manier van prioriteren wordt veel gebruikt.

Is ook geen speld tussen te krijgen ... of wel?

Prioriteren: *items latere sprints*

Item	Business waarde	Effort	Return on Investment	Rank
Feature 2	300	100	3	3
Feature 3	30	6	5	5

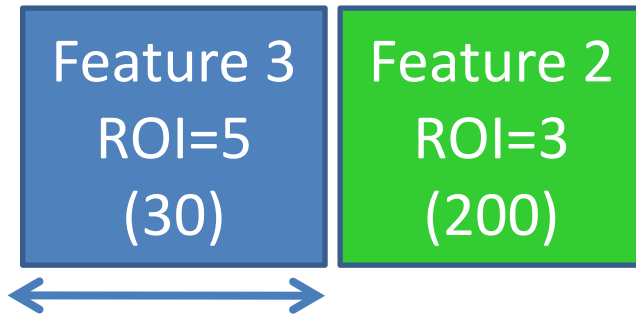
 Hoogste rank

Wat als waarde van Feature 2 afneemt in de tijd?

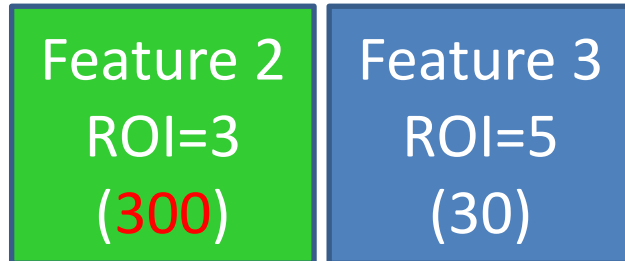
- nu uitvoeren: waarde 300
- ieder maand later uitvoeren: 50 van waarde af
- > 2 maanden later kost 100 punten waarde!



Prioriteren: *items latere sprints*



Totale waarde: 230



Totale waarde: 330

De Cost of Delay ($2 \times 50 = 100$) is **belangrijk!**

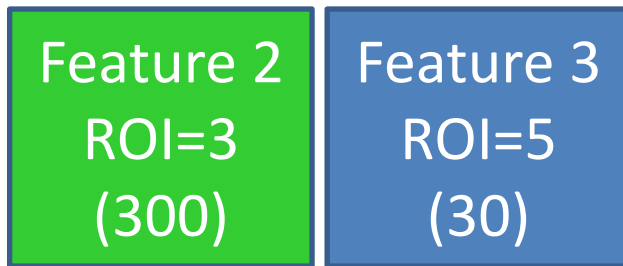
Gelukkig zijn er bruikbare principes



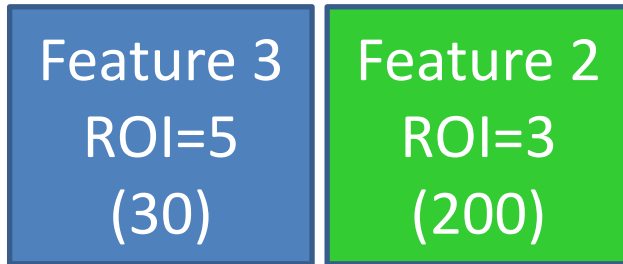
Donald Reinertsen

De Cost of Delay ($2 \times 50 = 100$) is allerbelangrijkst!

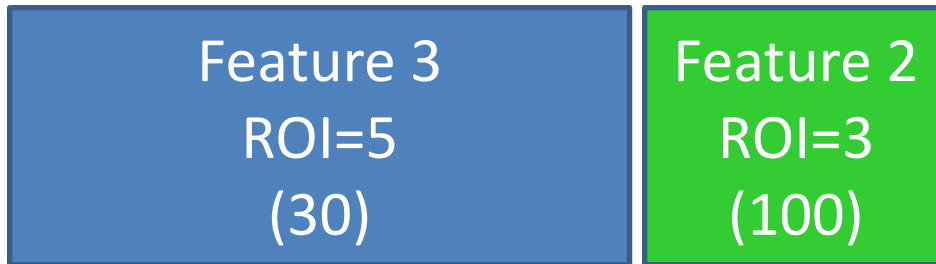
Prioriteren: *items latere sprints*



Cost of Delay: 0



Cost of Delay: 100
(2x50)



Cost of Delay: 200
(4x50)

Gelukkig zijn er bruikbare principes

Drie methoden



Donald Reinertsen

- Shortest Job First
- High Cost of Delay First
- Weighted Shortest Job First

Nieuwe voorbeelden: CoD per week

Gelukkig zijn er bruikbare principes



Donald Reinertsen

Twee variabelen

1. Effort (in weken)
2. Cost of Delay (per week)

Item	Effort	Cost of Delay
Feature A	1	3
Feature B	3	3
Feature C	10	3

Shortest Job First

Als Cost of Delay voor twee items gelijk is
Kies dan Shortest Job First
(= minste Effort)

Item	Effort	Cost of Delay
Feature A	1	3
Feature B	3	3
Feature C	10	3



CoD: 3



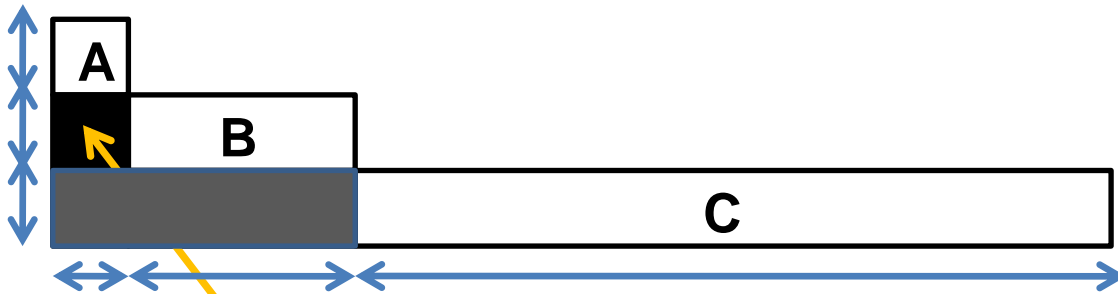
CoD
gelijk



Effort: 1 3

10

Shortest Job First



Totaal: **15**
 $(1 \times 3 + 4 \times 3)$

Totaal: **69**
 $(10 \times 3 + 13 \times 3)$

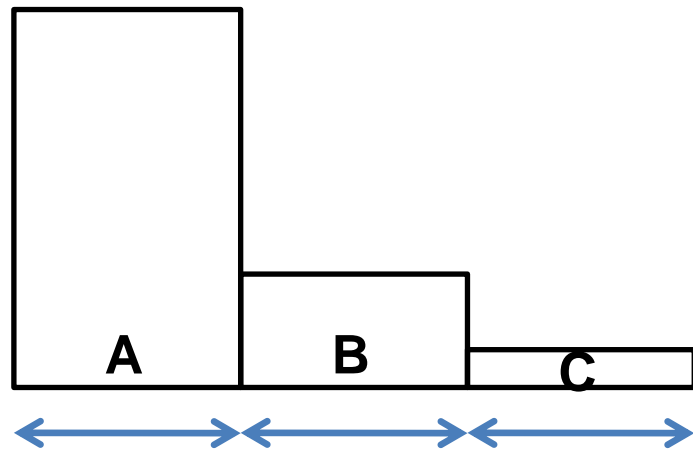


Item	Effort	Cost of Delay
Feature A	1	3
Feature B	3	3
Feature C	10	3



High Cost of Delay First

Als Effort voor twee items gelijk is
Kies dan High Delay Cost First



Effort: 3 3 3

10

3

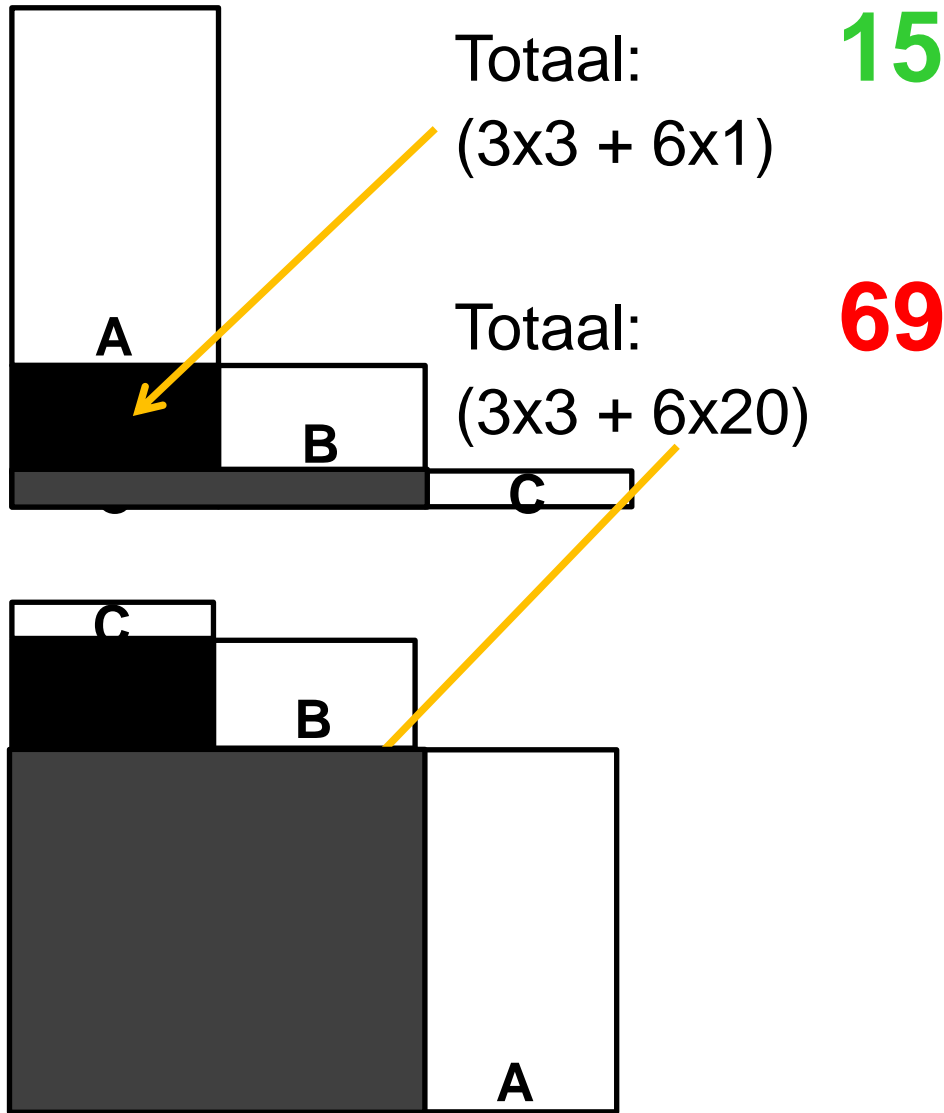
1 CoD

Item	Effort	Cost of Delay
Feature A	3	10
Feature B	3	3
Feature C	3	1



Effort
gelijk

High Cost of Delay First



Item	Effort	Cost of Delay
Feature A	3	10
Feature B	3	3
Feature C	3	1

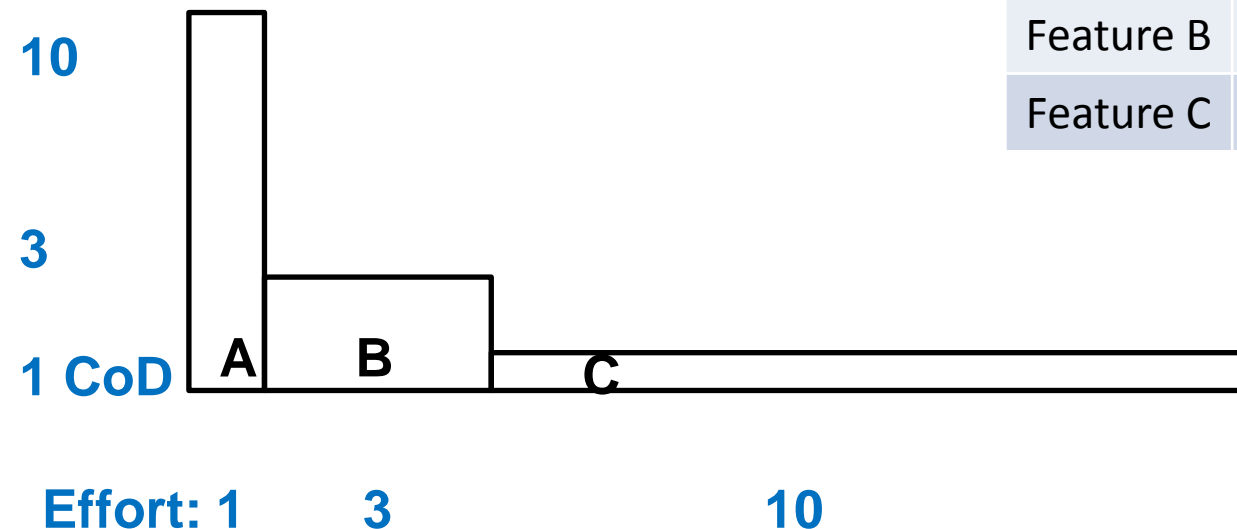


Weighted Shortest Job First

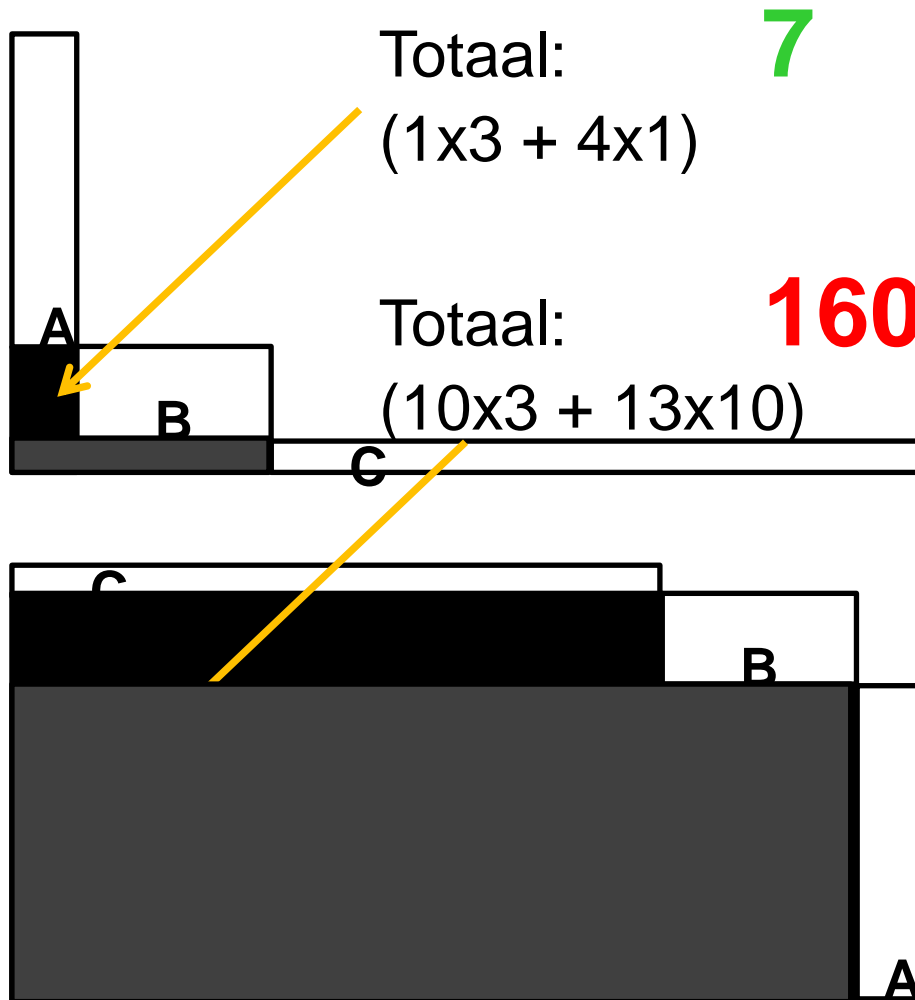
Als Effort en Cost of Delay voor twee items verschillen

Kies dan Weighted Shorted Job First

Item	Effort	Cost of Delay	Weight C/E
Feature A	1	10	10
Feature B	3	3	1
Feature C	10	1	0,1



Weighted Shortest Job First



Item	Effort	Cost of Delay	Weight C/E
Feature A	1	10	10
Feature B	3	3	1
Feature C	10	1	0,1



Gelukkig zijn er bruikbare principes

Drie methoden



Donald Reinertsen

- Shortest Job First
- High Cost of Delay First
- Weighted Shortest Job First

Maar hoe bepaal je de Cost of Delay?

Cost of Delay

	Cost of Delay			
Item	Business waarde	Doelstelling	Risico reductie	Totaal
Story 1	8	4	2	14
Story 2	8	2	3	13
Story 3	6	2	1	9
Story 4	4	4	7	15
Story 5	3	5	8	16

**Cost of Delay =
Business waarde +
Doelstelling waarde +
Risico reductie**

- Hier simpele optelling
- Kan ook gewogen

Weighted Shortest Job First

Item	Cost of Delay				Effort	WSJF (C/E)
	Business waarde	Doel stelling	Risico reductie	Totaal		
Story 1	8	4	2	14	7	2
Story 2	8	2	3	13	2	6,5
Story 3	6	2	1	9	3	3
Story 4	4	4	7	15	3	5
Story 5	3	5	8	16	4	4

- Schaal: 10 is hoogste, 1 is laagste
- Doelstelling = bijdrage aan sprintdoelstelling, releasedoelstelling, business doelstelling

Conclusie

- Met paar parameters is goed te prioriteren
- Waarden zijn in project goed in te schatten
- De opbrengst van goed prioriteren is fors
- Reindertsen heeft goede ideeën

Vragen



