

# Managing with Kanban

Come rivoluzionare l'ufficio  
e vivere contenti

# What?

Overview of DFP platform (and my TEAM)

Condensed scrum

Condensed Kanban

Question and answer

# Me



*Dfp Core team Technical  
Manager*

*Lean manager in  
becoming*

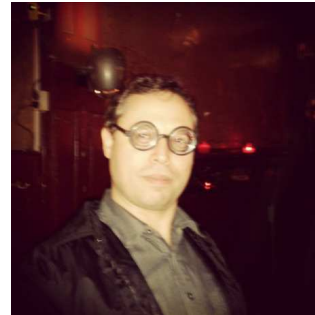
*certified Scrum Master*

*Kanban (upper case K)  
enthusiast*

*Future crossfitter*

*Hermitie husband (2013)*

# My TEAM (Dfp Core)



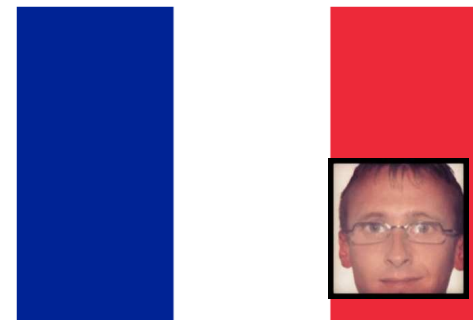
# Where?



Netherlands



Peru (-6h)



France

# Our Mission:

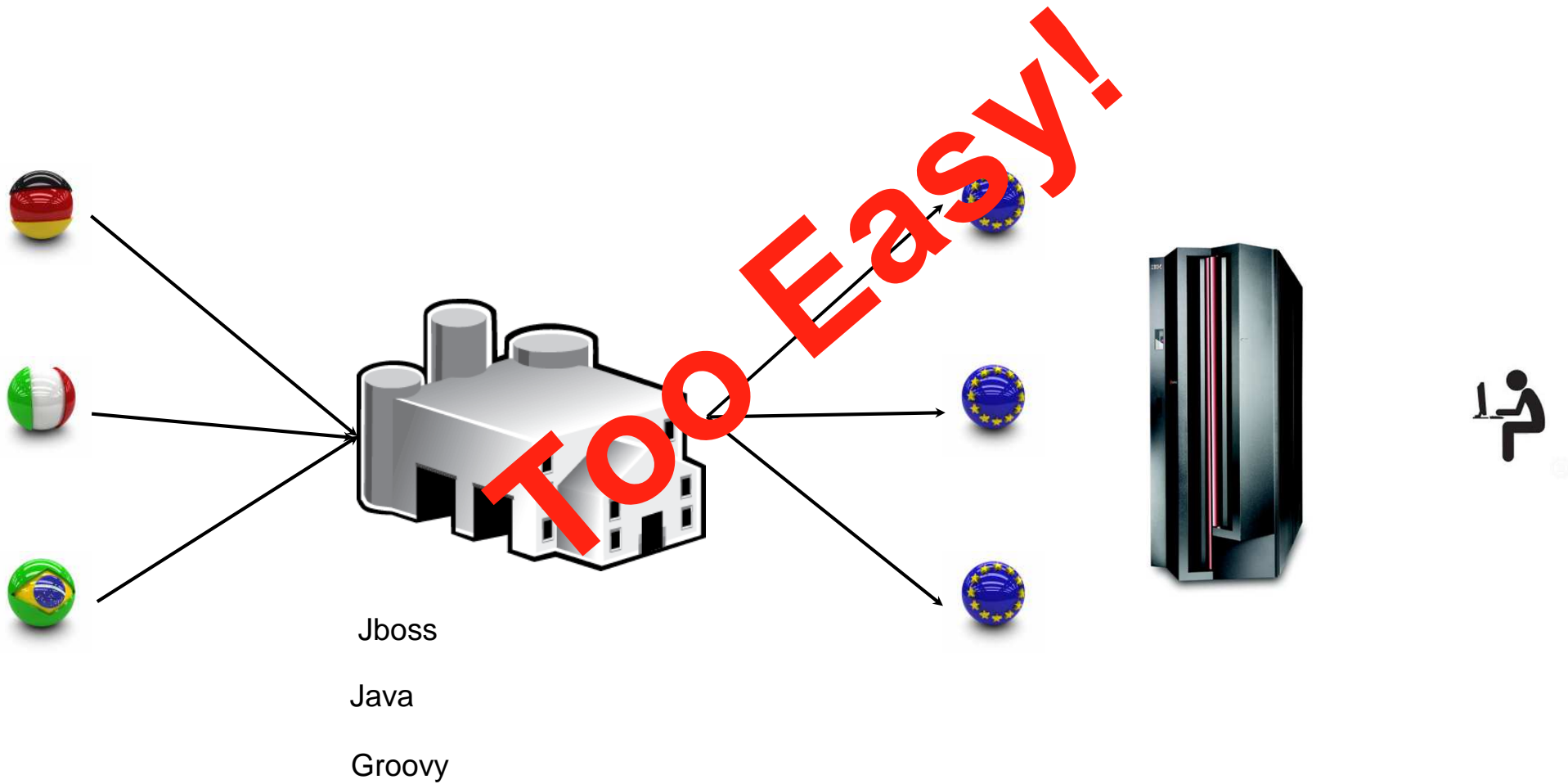
**Maintain and evolve the:**

**D**ata

**F**low

**P**latform

# Enter the **DFP**



# REAL LIFE!

## 4 FLOWS EACH COUNTRY:

- biblio
- abstract
- full text
- citation
- public
- secret

## Production support

- bug fixing
- monitoring

## NPL FLOWS:

- images
- pdf

## CPC Classification:

- reorganization

4 different projects

- dfp-admin
- dfp-console
- dfp-core
- processes

4 different external teams

Priorities changing almost daily

Special assignments

Prs, legal status

6 different product owners

Different level of experience inside the team



# Scrum approach



product owner



Scrum master

Team



# Sprint



Produce

Sprint backlog

Planning poker

0	1/2	1	2	3	5
8	13	20	40	100	?

Maintains

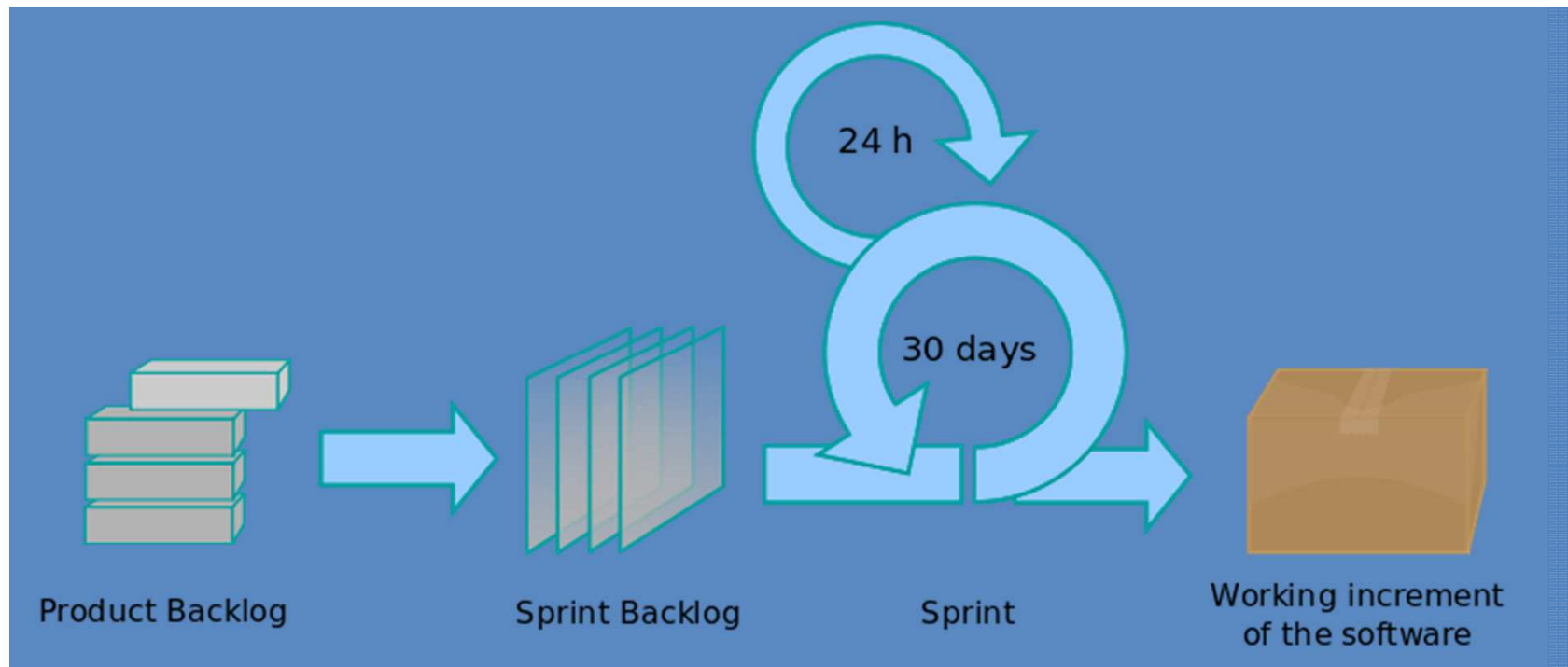
Monitors

Item #	Description	Est.	Est.
1	Build database web browser	10	10
2	Get all of records if there's any in database	8	8
3	Add browser	1	1
4	Connect user training	16	16
5	Connect user training	16	16
6	File browser and support are not of date	100	100
7	Database browser	200	200
8	Database browser	200	200
9	Database browser	200	200
10	Database browser	200	200
11	Database browser	200	200
12	Database browser	200	200
13	Database browser	200	200
14	Database browser	200	200
15	Database browser	200	200
16	Database browser	200	200
17	Database browser	200	200
18	Database browser	200	200
19	Database browser	200	200
20	Database browser	200	200
21	Database browser	200	200
22	Database browser	200	200
23	Database browser	200	200
24	Database browser	200	200
25	Database browser	200	200
26	Database browser	200	200
27	Database browser	200	200
28	Database browser	200	200
29	Database browser	200	200
30	Database browser	200	200
31	Database browser	200	200
32	Database browser	200	200
33	Database browser	200	200
34	Database browser	200	200
35	Database browser	200	200
36	Database browser	200	200
37	Database browser	200	200
38	Database browser	200	200
39	Database browser	200	200
40	Database browser	200	200
41	Database browser	200	200
42	Database browser	200	200
43	Database browser	200	200
44	Database browser	200	200
45	Database browser	200	200
46	Database browser	200	200
47	Database browser	200	200
48	Database browser	200	200
49	Database browser	200	200
50	Database browser	200	200

Backlog



# How it works



# Monitors 1/2

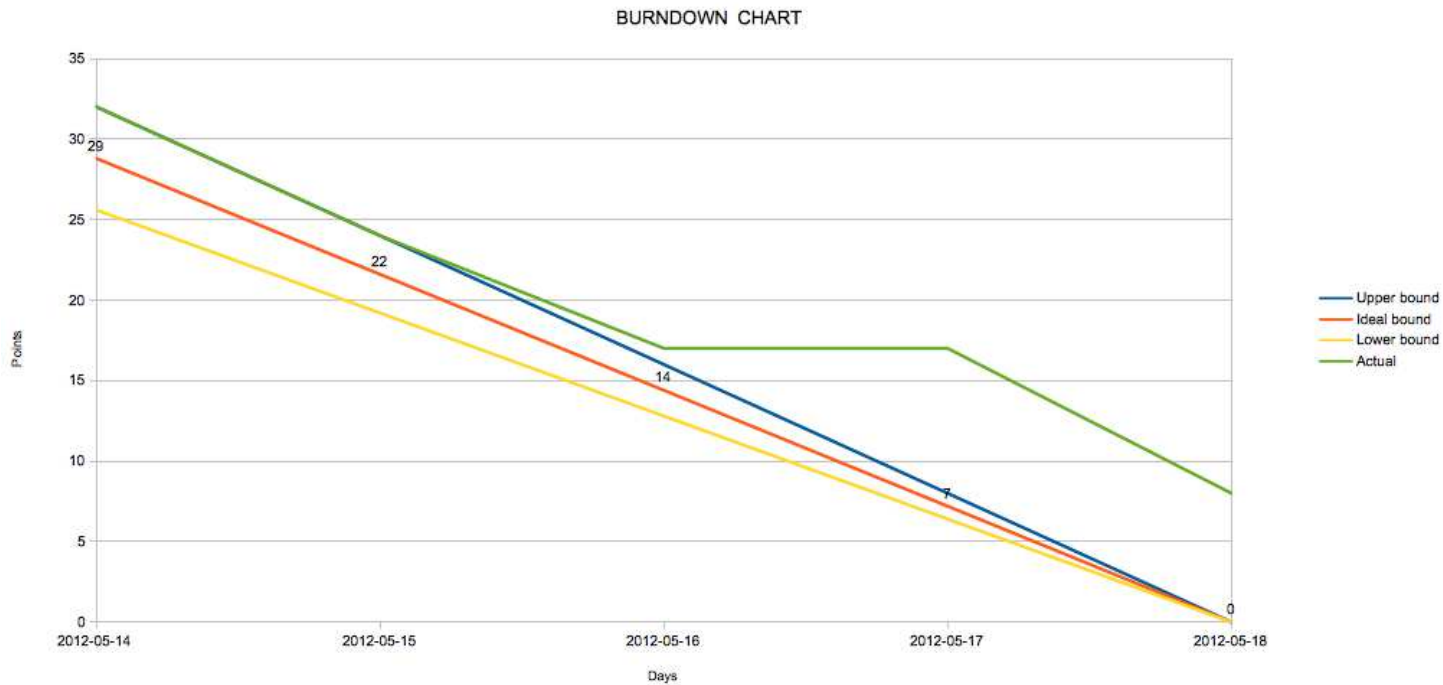


Stand up

Board

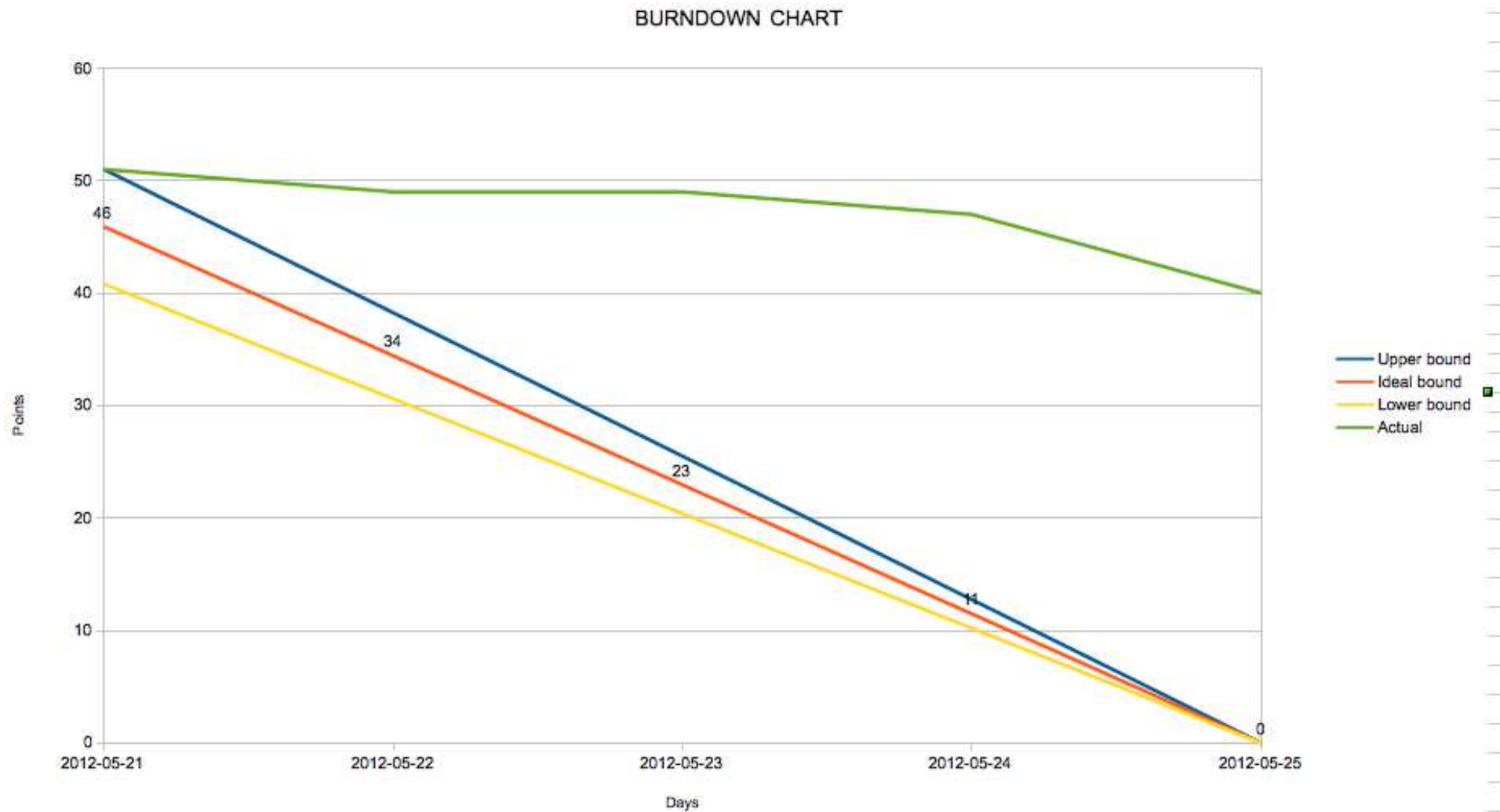


# Monitors 2/2



<u>Previous Velocity</u>	<u>Velocity</u>	<u>Work capacity</u>	<u>Total Commitment</u>	<u>Focus Factor</u>	<u>Adopted Work</u>	<u>Found Work</u>	<u>Target Value Contribution</u>	<u>Accuracy of estimation</u>	<u>Accuracy of Commitment</u>
25	25	45	32	55.56%	40.63%	53.13%	100.00%	28.13%	43.84%

# Real life strikes again!



? 1/2

Acceptance did not (always) depend on us

Demo was not taken in account while planning

External request always accepted

Board does not reflect reality

Managers tends to keep people busy, not to have work done

TEAM does not scale

? 2/2

Different types of activities:

- development
- bug fixings
- production monitoring
- support for other teams

Different type of constraint:

- production never waits
- some deadlines does not wait
- some deadlines can wait



# Consequences

```
if (end_of_sprint && !task.done?)
```

```
    task.drop
```

```
end
```

the TEAM was demotivated and ALWAYS late

Customers do not understand points

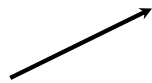
Customers always angry

# We need to do something!



# Kanban and coffee

Upper case



Done



kanban



Lower case

# More in depth

- kanban
- workflow (visualized)
- limited work in progress
- process are explicit
- measurement and management of the flow
- (?) using of models to recognize improvement opportunities (TOC, muda,...)

What about software?

# Kanban


Jira ID

- Run PT in INTG 1905

- Talk with Fabier to check the files

INPOTEL: 5/10/2012

OK



Size












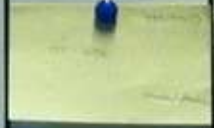








⊙ Small

Other info

Class of Service

22

Detailed description: The image shows a whiteboard with handwritten notes in black and blue ink. At the top right, the number '1905' is written, with an arrow pointing to it from the label 'Jira ID'. Below this, two main tasks are listed: '- Run PT in INTG' and '- Talk with Fabier to check the files'. In the lower-left area, 'INPOTEL: 5/10/2012' is written in blue ink, with an arrow pointing to it from the label 'Other info'. To the right of this, the letters 'OK' are written in black. In the center of the whiteboard is a small square photograph of a man with glasses and a dark shirt, smiling. At the bottom right, the text '⊙ Small' is written, with an arrow pointing to it from the label 'Class of Service'. To the right of the photo, the word 'Size' is written, with an arrow pointing to it from the label 'Size'. The number '22' is located in the bottom right corner of the whiteboard area.

READY 12	ANALYSIS 4	DEV 4		READY 8 FOR ACCEPTANCE	ACCEPTANCE 4		BUILD 8	
		IN PROGR	DONE		IN PROGR	DONE	IN PROGR	DONE
 Experts      		  		    				
External								

INTG €

UAT €

OSA €

IO €

DONES

[Yellow sticky note]

[Yellow sticky note]

[Yellow sticky note]

[Yellow sticky note]



ON VACATION

ON HOLD

[Yellow sticky note]

[Yellow sticky note]

[Yellow sticky note]

LEGEND

- ! EXPEDITE CLASS
- FIXED DELIVERY DATE CLASS
- STANDARD CLASS
- ▲ INTANGIBLE CLASS
- X EXTERNAL CLASS





WIP

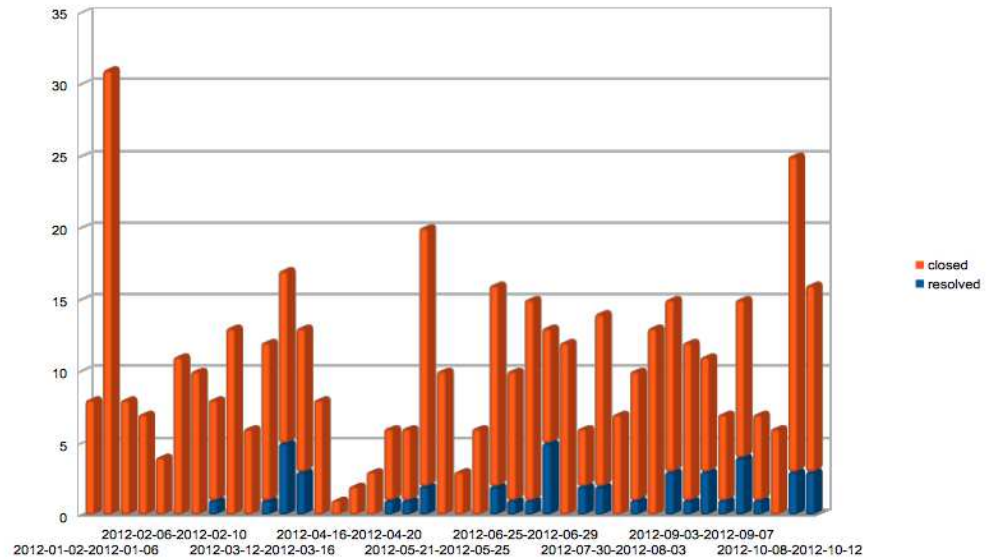
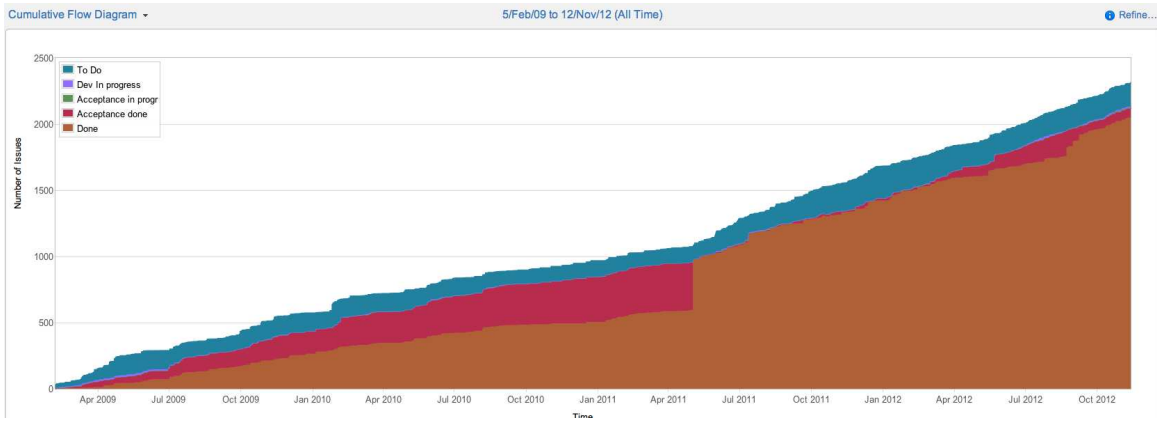
80

# Manage and measure flow

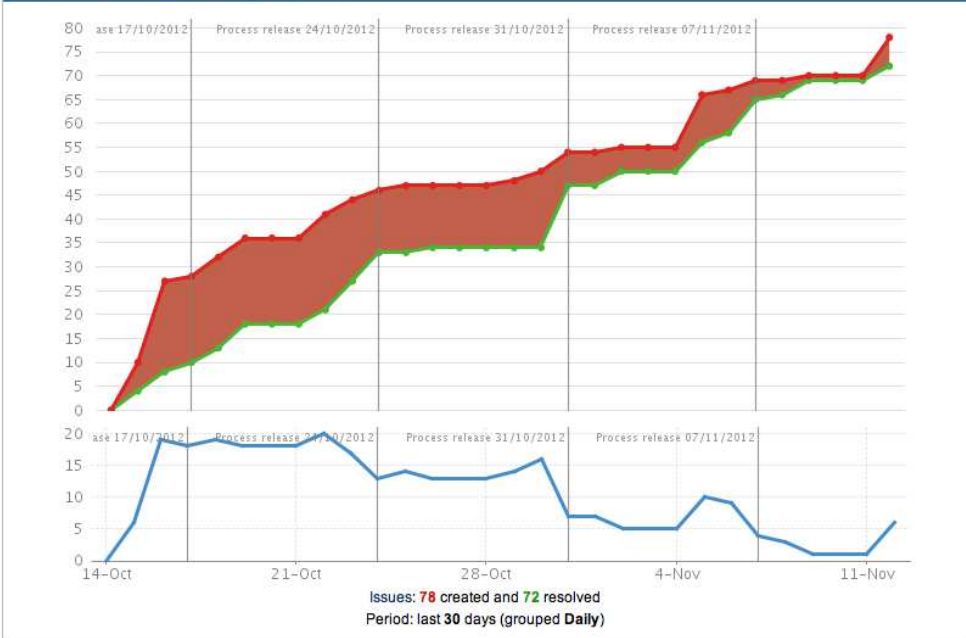
Tickets per week

Cumulative flow chart

Cycle/Lead time



**Created vs. Resolved Chart: Data Flow Platform**



# Explicit process and Rules

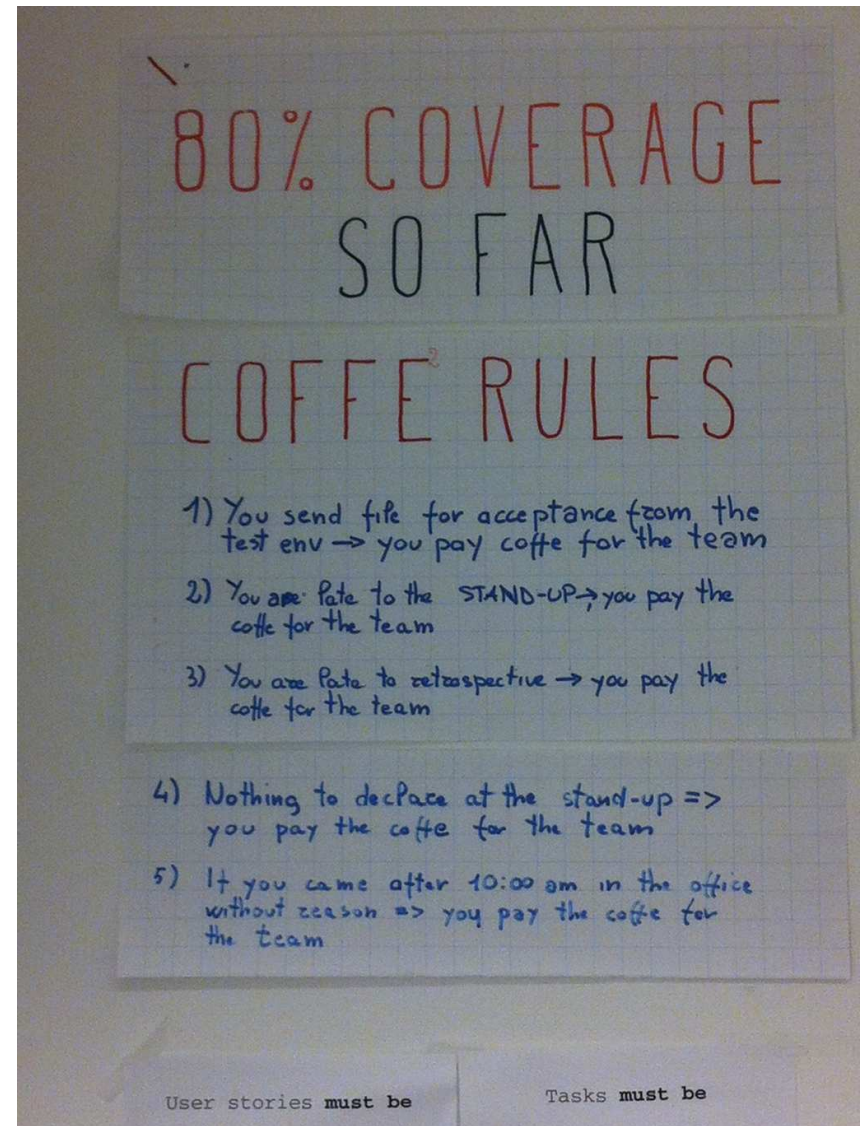
Expedite Class

Fixed date delivery  
class

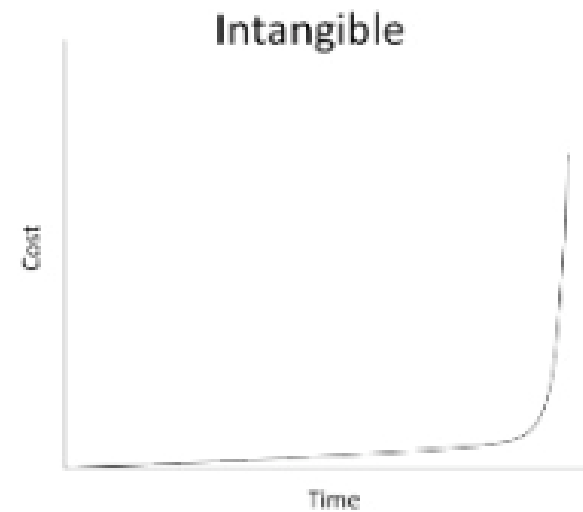
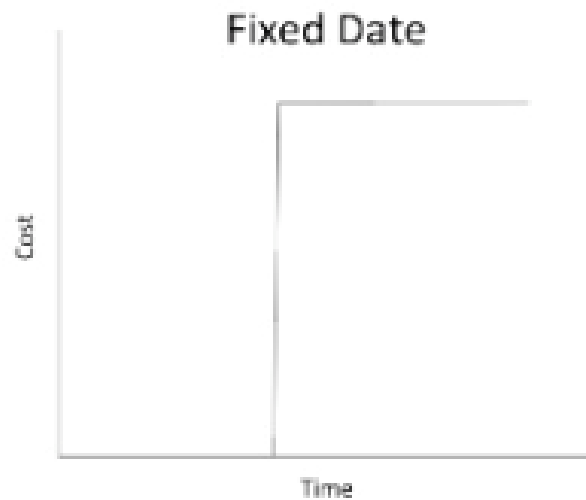
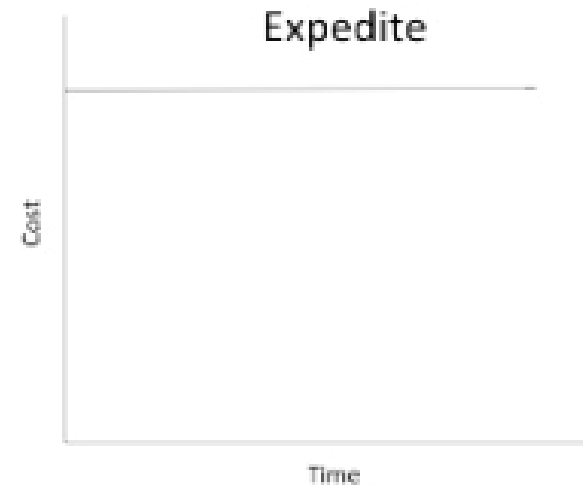
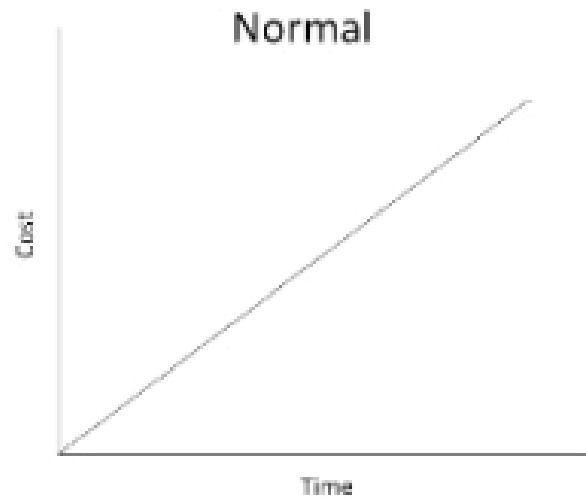
Standard class

Intangible class

External class



# How to assign a COF



# Class of service AND cycle time

StandardClass 69%

IntangibleClass

12%

2009: 67

ExternalClass 2%

2010: 59

Expedite 7%

2011: 41

FixedDateDelivery

2012: 23

10%

How do I implement it?

# Visualize the flow



**Set the WIP  
limit**

Find the  
bottleneck

# Exploit the bottleneck

Increase the limit

:goto 3



**WARNING**

**NEVER EVER  
BREAK THE WIP  
LIMIT!**

# Why?

Create a pull system

Create slack time

# Pull system

Ready 12	Analysis 4	Dev 4		RFA 8	Acceptance 4		Build 8	
		In Progr	Done		In Progr	Done	In Progr	Done
█	█							
█	█	█	█					
█								
█	█	█						
█								
█	█							
█				█				
█				█				
█	█			█				
				█				
█				█				

## The M/M/1/k Queue

WIPCAP	1	2	5	10	20	Infinite
Average Cycle Time	1.0	1.5	2.8	4.6	7.2	10.0
Time in Queue	0	0.5	1.8	3.6	6.2	9.0
Utilization Percent	47%	63%	79%	85%	89%	90%
Empty Percent	53%	37%	21%	15%	11%	10%
Blocking Percent	47%	30%	13%	5%	1%	0%

Note: Assumes 90 percent utilization.

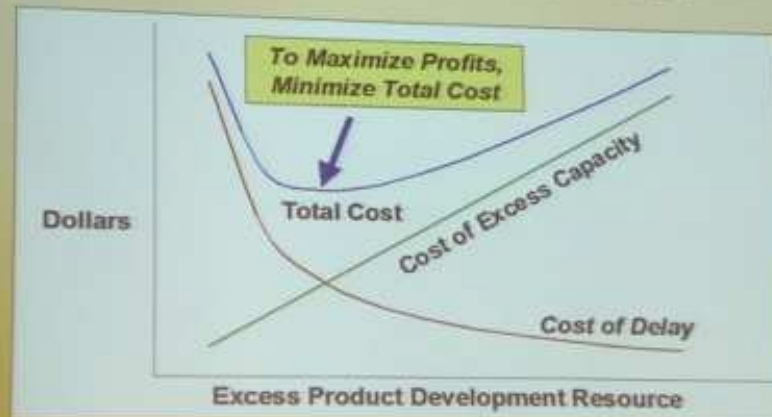
WIP constraints tradeoff reductions in cycle time against blocking and underutilization costs.

Copyright © 2012, Routledge & Taylor & Francis

23



## The Economics of Queues



How many product developers can draw the red curve?

Copyright © 2012, Deloitte & Associates

# Slack time (is not evil)

Improve the process

Improve yourself

Automate

Manage emergency

Do important but not urgent things  
(Intangible class)

Spend time on FB...

# What we have done during slack time...

Cleaned up the database

Migrated application to new version of grails and TB

Revamped rollback for NPL

Closed some memory leaks

Throw balls at each other and drink coffee together

# Consequences

```
tasks.each do |task|  
  task.work_until_done  
end
```

TEAM is not late anymore...

customers are happy (not always)

We are pulling from the customers

# Why you should use it

Easy to Implement (you need only post-it)

Works out of the box (docdb)

Scales from small to big team (from 1 to 50)

Organization of work comes for free (even with distributed team and resources)

Continuos improvement (state of mind)

Swarming in problem

Work is finished when is finished!

Deliver value to customer

Clients do understand days

● ?  
dank je wel

# Links and Resources

<http://scrumalliance.org>

<http://rapidscrum.com/MetricsVideos.php>

<http://www.agilemanagement.net>

The Definitive Guide! KANBAN, Successful  
Evolutionary Change For Your Technology  
Business

[www.lovadina.org](http://www.lovadina.org)

sayat.me/nicolalovadina

twitter.com/nicolalovadina

# Personal Kanban

For a team of 1





Who said that Kanban board must be boring?

