

PLANNING 4

THE DOT GAME

KANBAN
PRINCIPLES

KANBAN IN
PRACTICE

WRAP UP &
DISCUSSION

DOING 2

INTRODUCTIONS



DISCUSSION &
QUESTIONS

DONE!

BOOK HOTEL
& DO
TRAVEL ARR.

A Practical Introduction
To Kanban



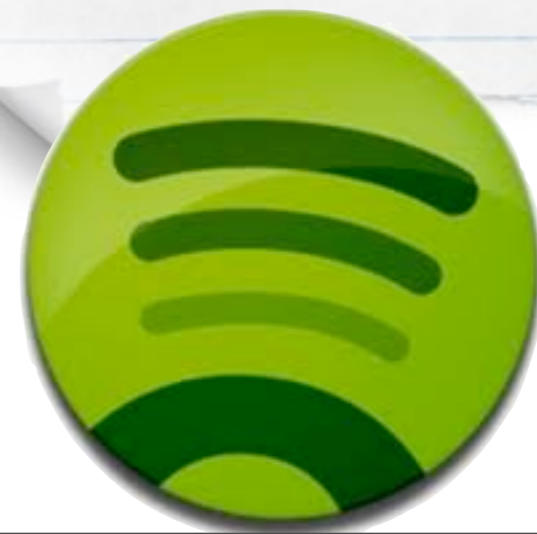
Marcus

*Consultant, BDD, .NET, Kanban, Salvationist
Three kids, Practical agilist, Scrum*



Joakim

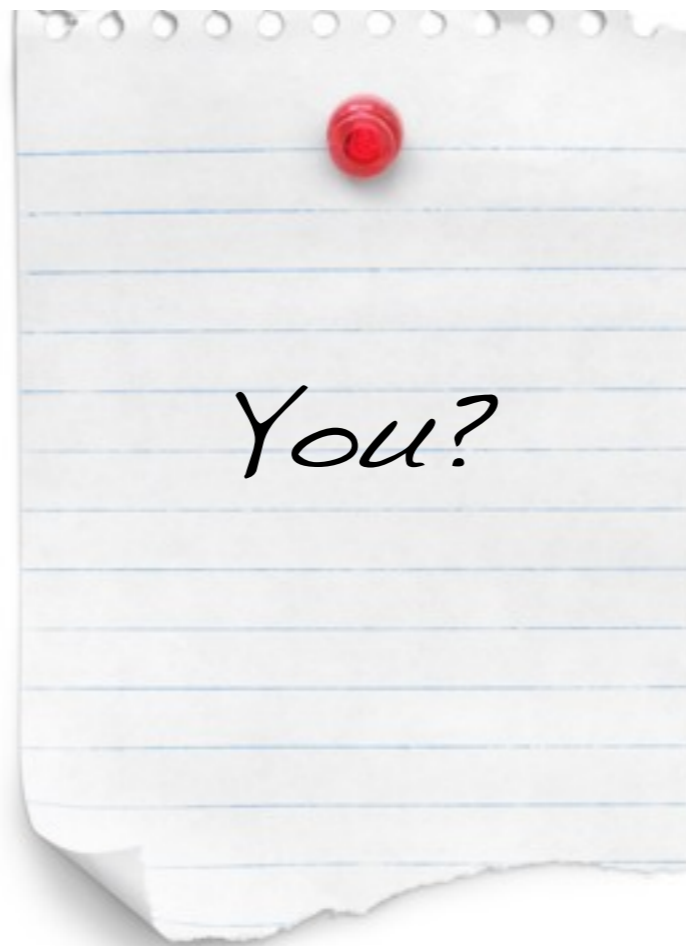
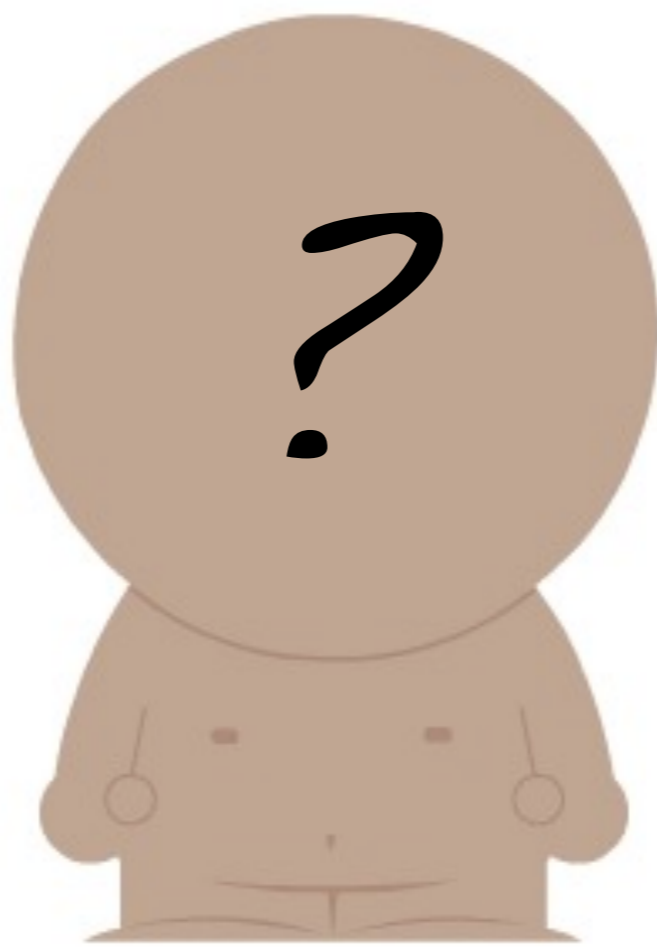
Agile & Kanban Coach, Norrbottning, AIK, Father of three, .NET developer, Cineast, Runner



aveja group

måndag 13 februari 12

This is us...



måndag 13 februari 12

Hands up

Developers?

Testers?

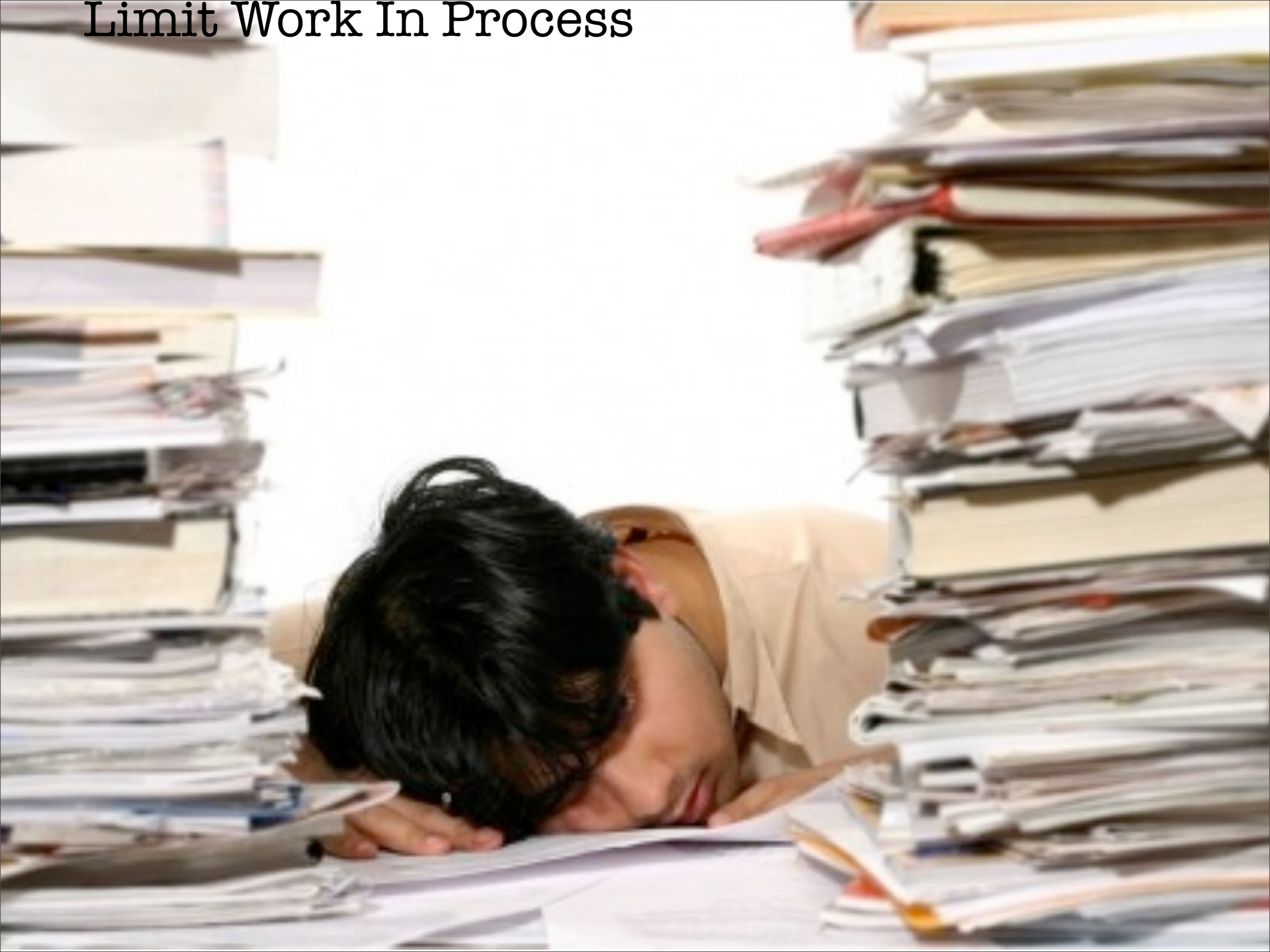
Fixed costs? Sorry, managers?

Scrum-users?

Kanban-users?

This will take about 3 hours if we are the only ones talking. It will take about 4 hours AND be more interesting if you interrupt us with questions.

Limit Work In Process



måndag 13 februari 12

Kanban is based on a very simple principle – agree on a limit to work-in-process...



BÄTTRE
BILDER?

Pull new work only when
something is finished

måndag 13 februari 12

...and pull new work from the queue only when something is finished.






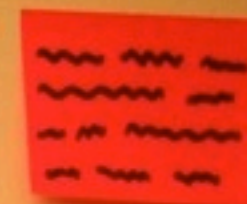
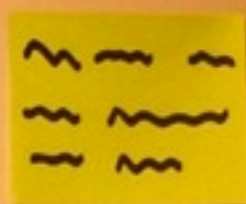
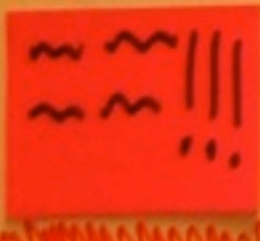
TODO

ANALYZE
2

DEV
34

TEST
2

DONE

	DOING	DONE	DOING	DONE		
<p>BUG 12.5%</p> <p>MAINTEN. 25%</p> <p>TECHNICAL 12.5%</p> <p>FEATURE 50%</p>			 		 	 
<p>URGENT!</p> 						

Visualize Workflow

måndag 13 februari 12

And use visualization to help you do that

Simple huh? But what does that mean in practice?

PLANNING 4

THE DOT GAME

KANBAN
PRINCIPLES

KANBAN IN
PRACTICE

WRAP UP &
DISCUSSION

DOING 2

DISCUSSION &
QUESTIONS

DONE!

BOOK HOTEL
& DO
TRAVEL ARR.

INTRODUCTIONS

PLANNING 4

KANBAN
PRINCIPLES

KANBAN IN
PRACTICE

WRAP UP &
DISCUSSION

DOING 2

THE DOT GAME



DISCUSSION &
QUESTIONS

DONE!

BOOK HOTEL
& DO
TRAVEL ARR.

INTRODUCTIONS

Let's practice
counting, shall we?

måndag 13 februari 12

Let's run a small game to see some of the benefits that Kanban can give you.

3 projects - part 1

Roman numbers

(I - X)

I

II

III

...

Letters

(A - J)

A

B

C

...

Numbers

(1-10)

1

2

3

...

måndag 13 februari 12

We have these 3 projects

- Project 1 - Roman numbers
 - This is superimportant! And hard too...
 - Write in black
- Project 2 - Letters
 - Also superimportant. Highest priority
 - Write in red
- Project 3 - Numbers
 - This is top prio with in our organization.
 - Really - the big guy is checking on us for this, people
 - Write with blue

Since they all are important - please do as much as possible on each. Like this
[CLICK]

Write time for each finished project and the total time

3 projects - part 2

Roman numbers

(I - X)

I

II

III

...

Letters

(A - J)

A

B

C

...

Numbers

(1-10)

1

2

3

...

måndag 13 februari 12

Eeh - that was a bit slow wasn't it?

Really - project I (roman numbers) is important. Out of business important.
Do that first.

Then - when you're done do #2

Finally #3 - it's just for fun anyway. ... But important

As before:

- Note the time for each finished project
- And the total time

What happened there?



måndag 13 februari 12

What happened? [AUDIENCE]

How did it feel?

What about quality?

What about time for each project?

Total time?

Why?

Can you see any similarities to work?

What could the different pens represent?

How about the priorities?

PLANNING 4

KANBAN
PRINCIPLES

KANBAN IN
PRACTICE

WRAP UP &
DISCUSSION

DOING 2

Jocke!

DISCUSSION &
QUESTIONS

DONE!

BOOK HOTEL
& DO
TRAVEL ARR.

INTRODUCTIONS

THE DOT GAME

PLANNING 4

DOING 2

DONE!

KANBAN
PRINCIPLES



BOOK HOTEL
& DO
TRAVEL ARR.

KANBAN IN
PRACTICE

DISCUSSION &
QUESTIONS

INTRODUCTIONS

THE DOT GAME

WRAP UP &
DISCUSSION



måndag 13 februari 12

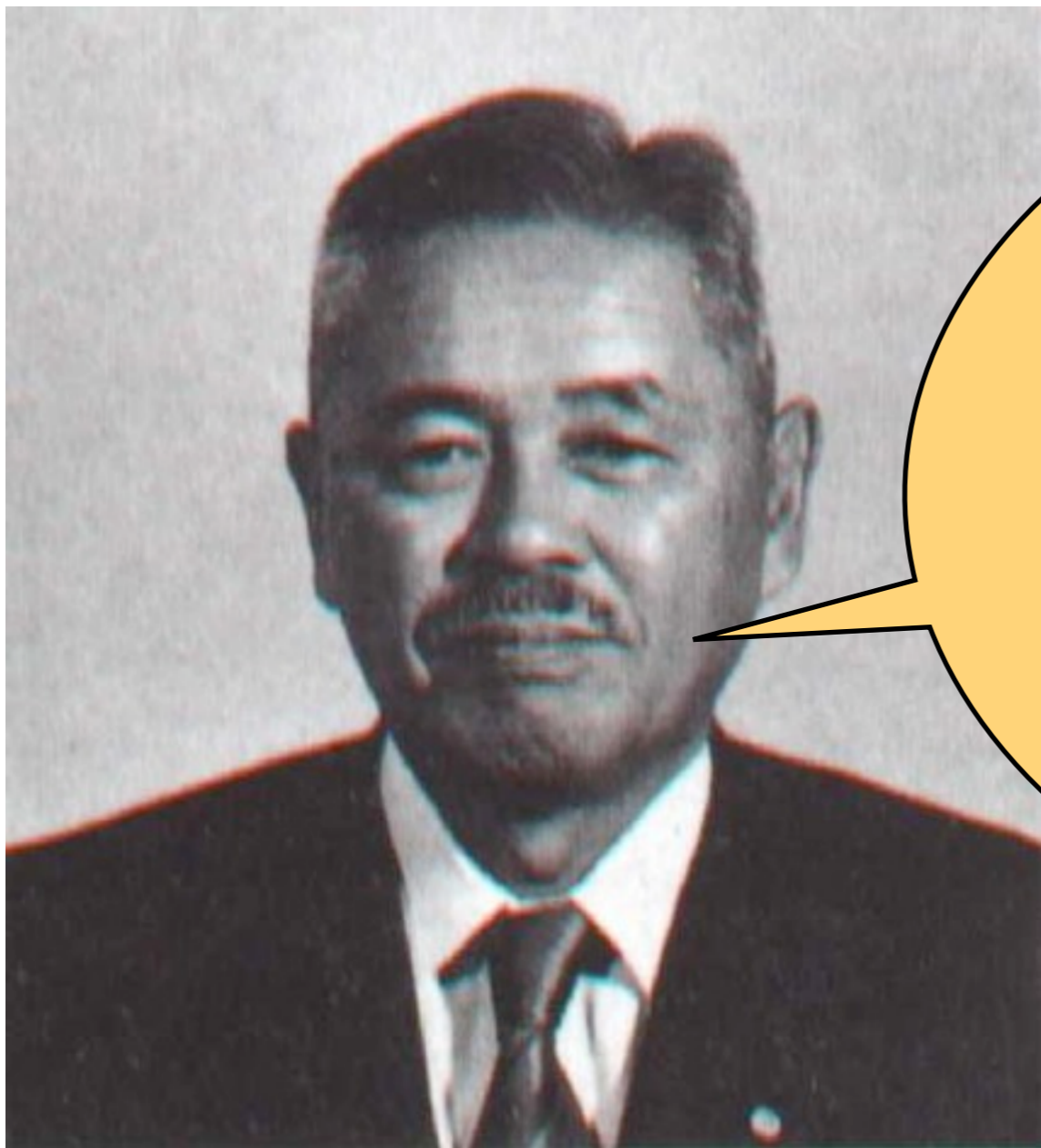
This is a Kanban. Kanban is actually Japanese for visual card or sign.

But it's also the name of a signaling system in manufacturing for pulling material between processes when needed instead of pushing it out according to a schedule or a plan. It was invented by...Toyota



måndag 13 februari 12

...as a part of Toyota Production System, TPS. What we in the West call lean manufacturing.



The two pillars of the Toyota production system are **just-in-time** and automation with a human touch, or **autonomation**. The tool used to operate the system is **kanban**.

Taichi Ohno

måndag 13 februari 12

Taichi Ohno, the father of TPS, said that:

“The two pillars of the Toyota production system are just-in-time and automation with a human touch, or autonomation. The tool used to operate the system is kanban.”

Just-in-Time means that you do things when there is an actual need for them instead of keeping inventories Just-in-Case.

Autonomation means that if an abnormal situation arises in production, the machine stops, the worker is signaled and will stop the production line. This prevents defects from being reproduced and focuses attention on understanding and removing the problem when it occurs, ensuring that it never recurs.

Just like Continuous Integration can be used in software development.

The idea with JIT is to achieve a flow...



FORMULERA OM?

FÖR MKT JOYCE

Poppendiecks?

So that we...

- Don't build features that nobody needs right now
- Don't write more specs than we can code
- Don't write more code than we can test
- Don't test more code than we can deploy

måndag 13 februari 12

...where value is added in a continuous flow from idea or concept to value in the hands of the customers.

Instead of moving big batches of work between different phases in a workflow through hand-offs, we focus on moving small units of work through the entire workflow quickly and without interruptions.

We do this so that we... <CLICK * 5>

We simply try to decrease the risk of wasteful over-production. But, equally important, this makes it easier to discover problems and waste that is always present in a process, such as

unnecessary hand-offs
defects and rework
bad specifications
unavailable resources and so on

Because when there is no inventory of other work to use or to be working on when something happens, every problem will result in a slowdown of the flow. Or make the flow uneven or even...



måndag 13 februari 12

...stop it entirely.

And this is a Good Thing because it gives us a chance to understand how our work works, to remove impediments and to eliminate waste.

In summary you could say that lean is all about engaging everyone in your organisation in identifying and solving problems. And that Kanban is a tool to help you do that.



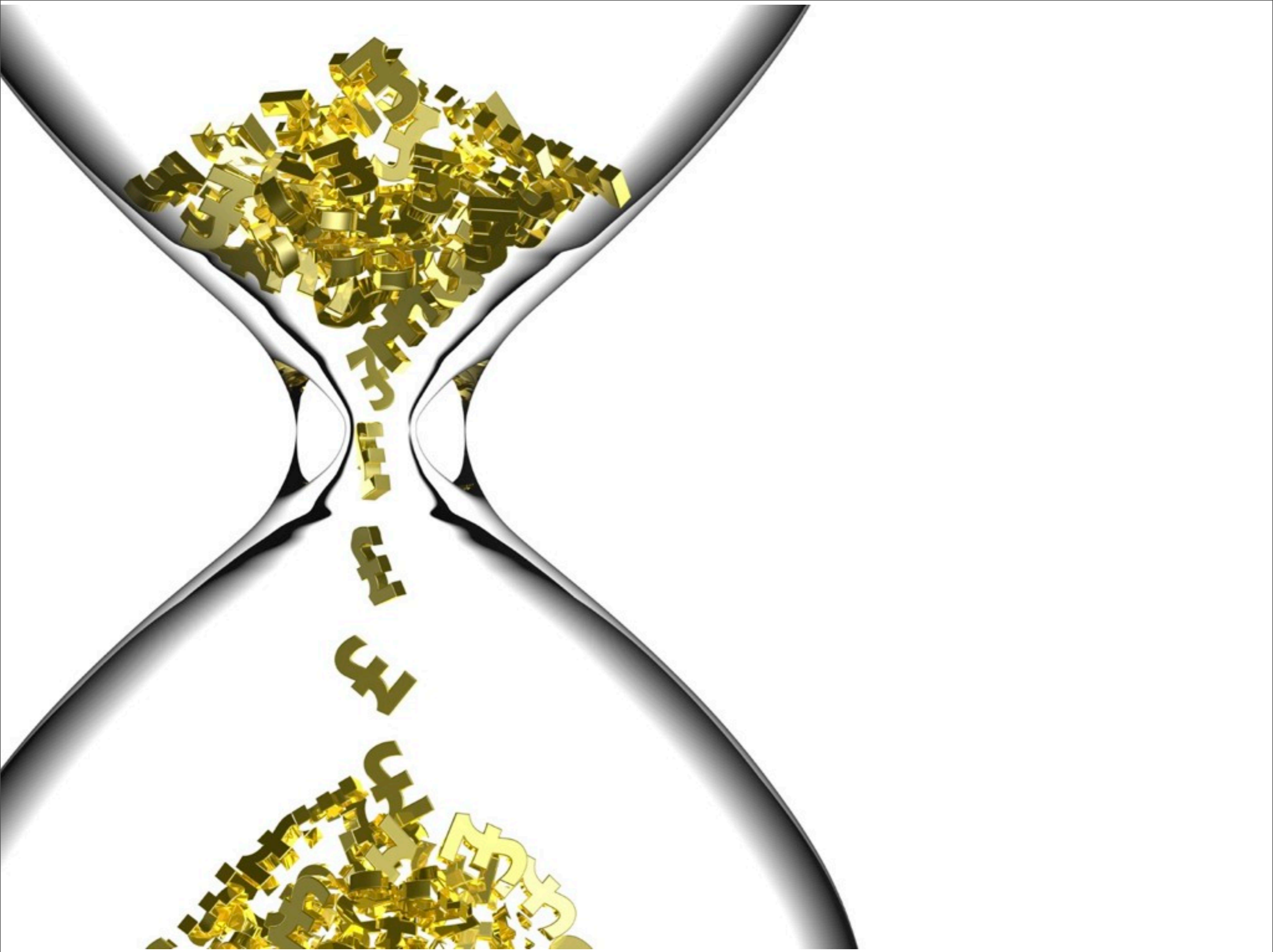
BÄTTRE BILDER? ✕

måndag 13 februari 12

To get faster flow, we focus on cycle time and try to drive it down.

Cycle time.

Lead time.



måndag 13 februari 12

Time is money. The faster we can get value delivered, the earlier our customer can buy and/or benefit from the value and the more money we will make.



måndag 13 februari 12

How do we get shorter lead times? What can we do? We can try to increase the rate by which we complete stuffs, to go faster.

Difficult, expensive – otherwise we would have done it already.

Fortunately there is an easier way.

Little's Law

$$\text{Cycle Time} = \frac{\text{Work-in-Process}}{\text{Avg Completion Rate}}$$

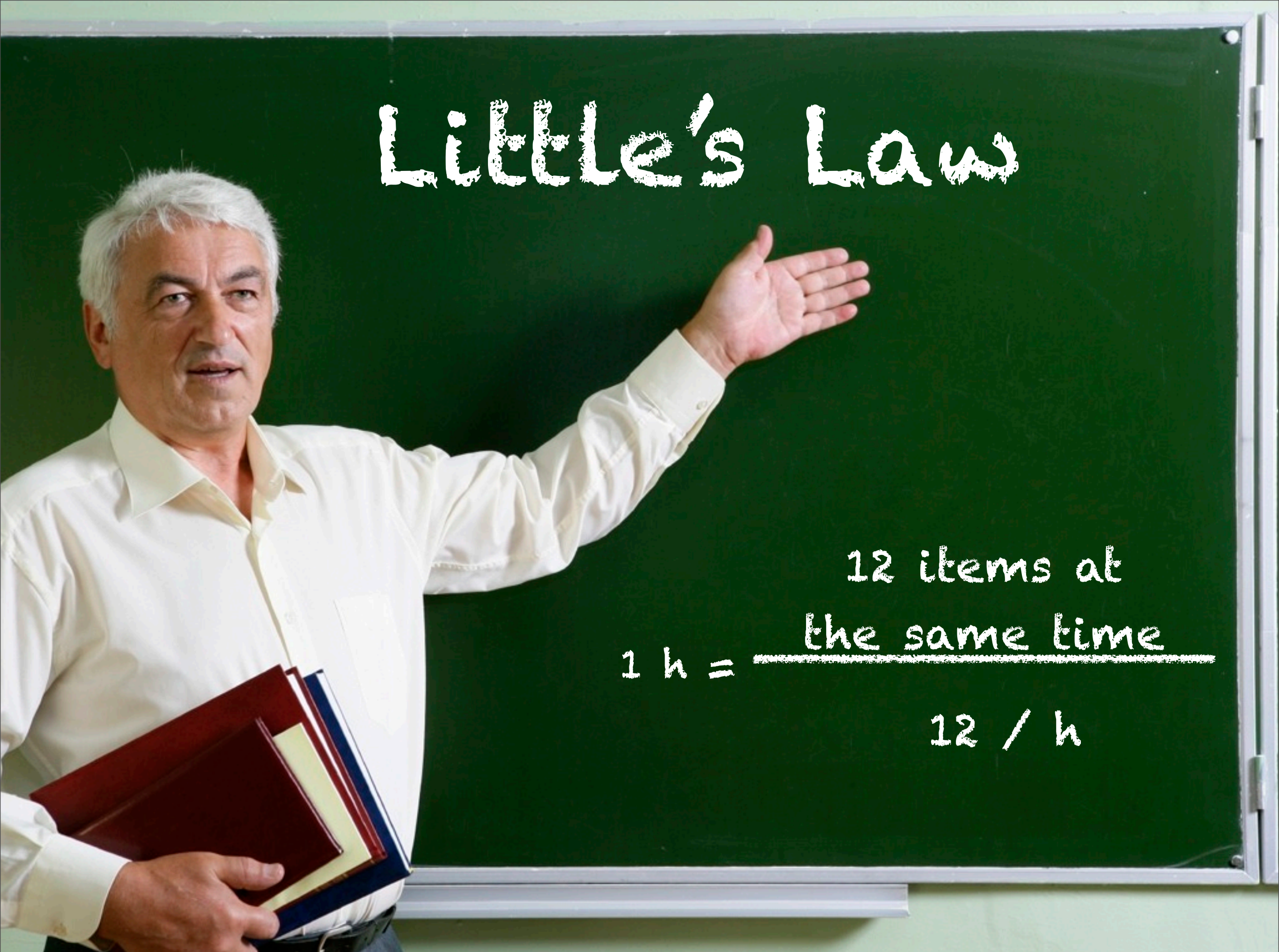
måndag 13 februari 12

Little's Law from Queueing theory tells us with mathematical certainty that the more things in process, the longer the cycle time.

If we have an average completion rate of, let's say one feature per week, and we are working on three features at the same time – we will get a cycle time of three weeks. If we instead worked on one feature at a time, we would get a cycle time of one week.

So the more we limit our Work-in-Process, often referred to as WIP, the shorter our cycle time.

Little's Law

A man with white hair, wearing a white button-down shirt, stands in front of a green chalkboard. He is holding several folders in his left hand and pointing with his right hand towards the chalkboard. The chalkboard has the title 'Little's Law' at the top and a mathematical equation written in white chalk. The equation is: 1 h = (12 items at the same time) / (12 / h).
$$1 \text{ h} = \frac{12 \text{ items at the same time}}{12 / \text{h}}$$

måndag 13 februari 12

So... that means – if we do 12 things at the same time with a average speed of 12 per hour.

That gives...

Yes – 1 hour per item.

Little's Law

$$0,5 \text{ h} = \frac{6 \text{ things at the same time}}{12 / \text{h}}$$

måndag 13 februari 12

But by just lowering the number of items to six
same speed for each

We get a time of just 0,5 h per item

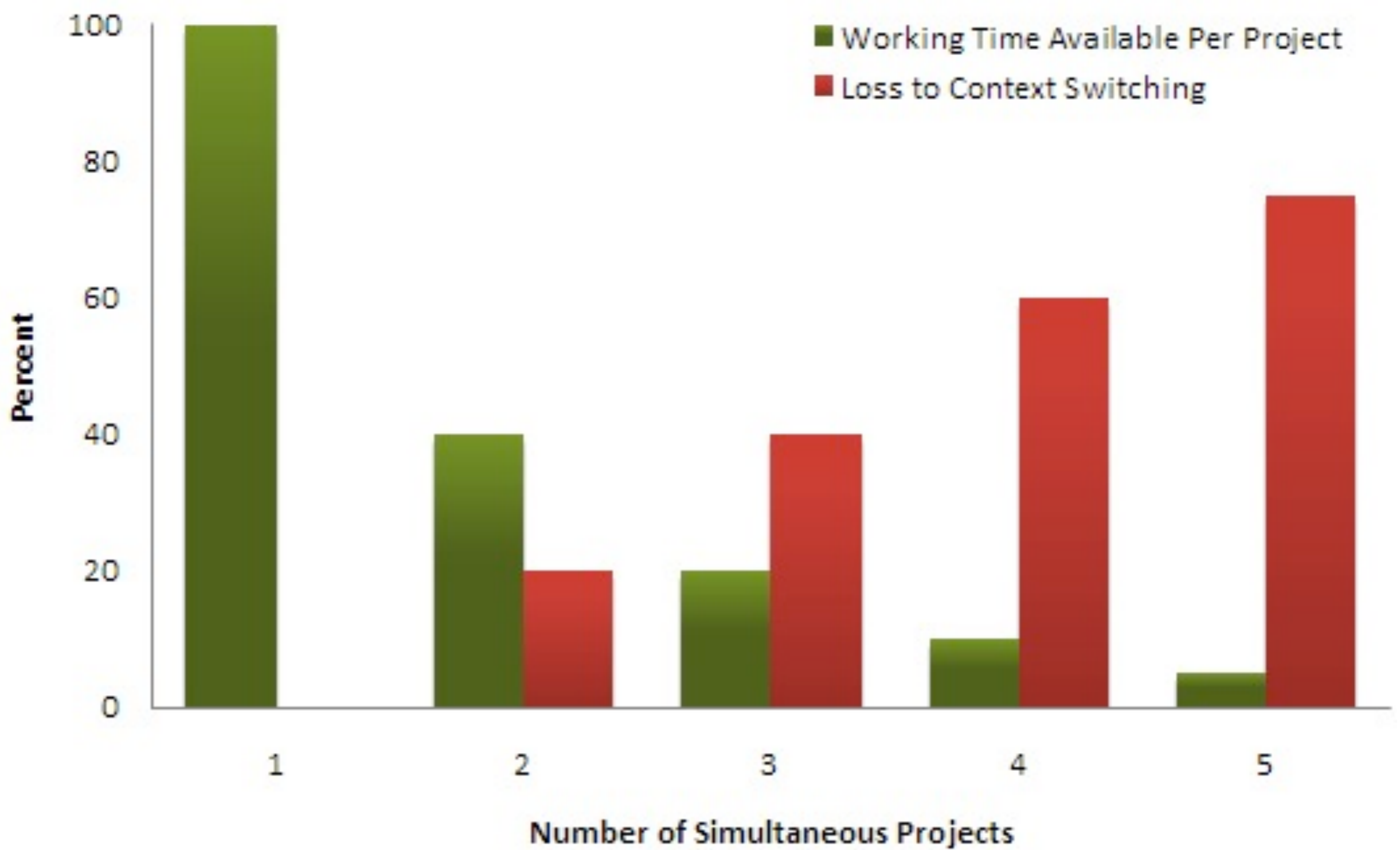
Little's Law

$$2 \text{ h} = \frac{24 \text{ items at the same time}}{12 / \text{h}}$$

måndag 13 februari 12

And reverse – doubling the number of items in progress

and you get a doubled cycle time for each item



Source: Quality Software Management: Systems Thinking, Gerald Weinberg

måndag 13 februari 12

In knowledge work it's even worse because of context switching. 20% time is lost to context switching per task, so fewer tasks means less time lost.

One study on context switching showed that context switching also represented a 10-point fall in IQ – more than twice that found in studies of the impact of smoking marijuana. So if you have to choose...



Marcus!

Work In Progress

In Software Development

måndag 13 februari 12

OK, we now know that we shall reduce the number of items we work with at the same time – called work in process WIP.

And that's quite easy to see and note when you have 3 pens in your hand – but in software development?

What is work in process there?

Specifications Waiting To Be Implemented



måndag 13 februari 12

It could be specifications that haven't been implemented yet.

They are written but not started. They are Work in process since they rot in a way. A specification has a best-before-date.

Do you agree? If you write a specification today – what are the odds that you can take it and use it straight off, a year from now?

That's work in process – loading our process with work.

Non-Integrated Code



måndag 13 februari 12

The same with code that you have written but not integrated against other code

The mug says; "They call me mr It works on my computer guy".
Any of you here?

Such code is not done – it's work in process



Byt bild
Trasiga /
konstiga
saker

© 2009 Andrew Tyler

© 2009 Andrew Tyler

Untested Code

måndag 13 februari 12

And code that isn't tested. Work in process

To not have a quick way to find out if the code you write works or not is a very good way to build a stock of not finished work.

Code Not Yet In Production



måndag 13 februari 12

And finally – code sitting on the shelf

All code not in production, sitting on the shelf, waiting to be deployed is work in process.

All this talk about “done”
disturbs me.

Code is in production or
not!

Michael Feathers

måndag 13 februari 12

I love this quote/tweet from Michael Feathers.

And even in production we're not completely done, right? Things can come back?



måndag 13 februari 12

And work in process is bad.

Not only that it loads our process with work but also that it prolongs our feedback loop.



måndag 13 februari 12

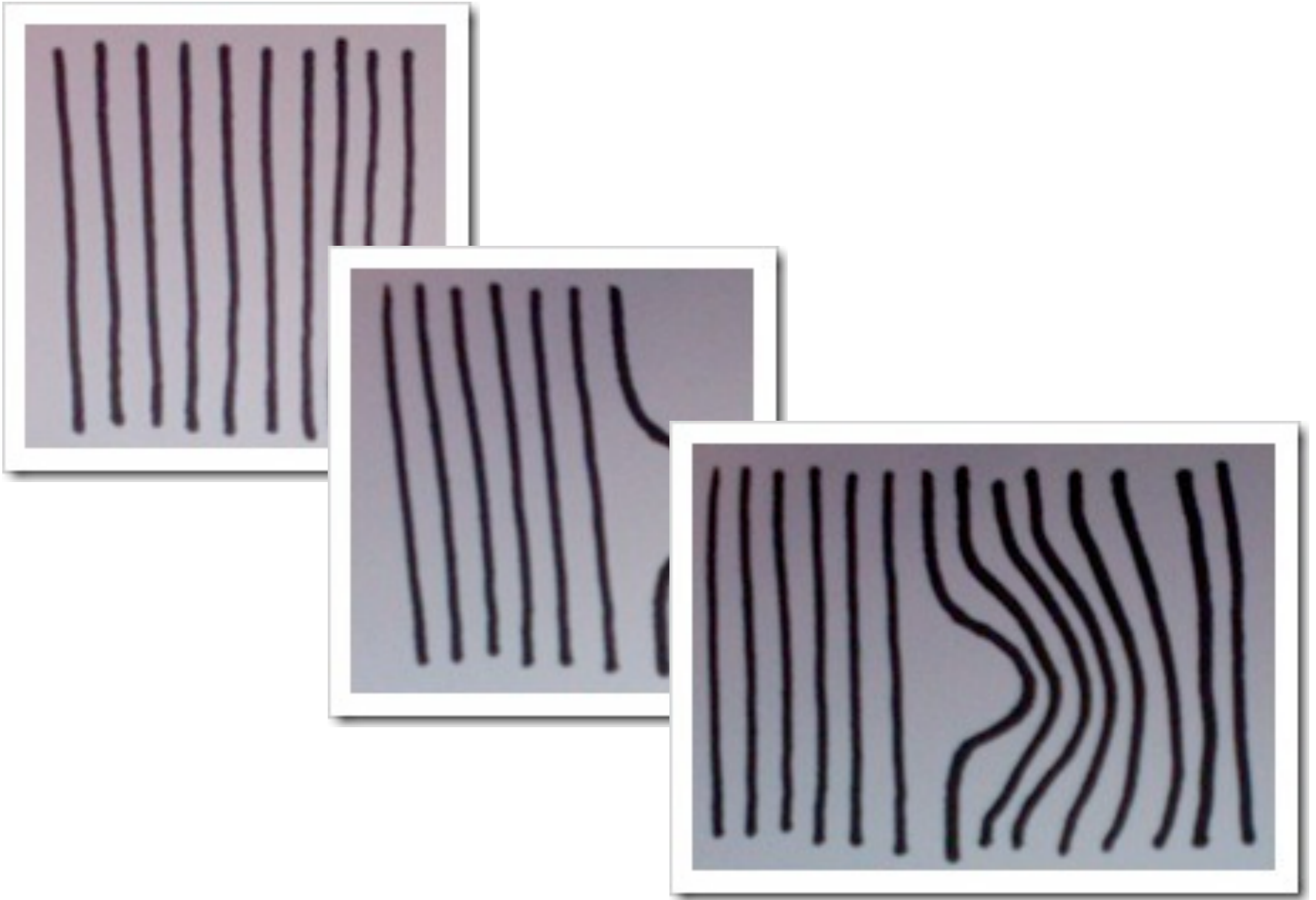
And it gets worse. Long cycle times, or delays, actually creates more work.

For example – take the example of introducing a bug. The delay from writing it until you find it actually create more work:

- Consider the situation when someone writes a bug and is **immediately** told of the bug.
- Now consider the situation when someone writes a bug but finds out about the bug **2 months later**.

Even assuming nothing in the code base has changed they will take considerably longer to fix the problem because they won't remember what they did.

This is the **additional** work that is created by the delay.



måndag 13 februari 12

And these delays causes us to work around things.

If we don't get quick feedback we will build on non-verified requirements, functions and tests etc.

<<CLICK>>

For example – here is our well-aligned and designed system when we start.

<<CLICK>>

But then we did a wrong assumption and happen to introduce a bug

<<CLICK>>

and all the work after that point is affected by that assumption.

We build in a defect that takes time to fix, or might be near on impossible to fix for cost reason for example.

Seems familiar? Anyone experienced this?



måndag 13 februari 12

With fewer things happening at the same time, low WIP, you will face lower risk.

We will become less sensitive for:

- changes, in requirements and requests from the customer
- quick changes in our competition
- changes in technology (new frameworks, patches etc).



måndag 13 februari 12

You can even make money straight off, by reducing the amount of work in process.

At the company level for example, administrative costs
– if we don't run as many projects at the same time the demand of lots of reporting will diminish.

One common thing is that when we get later more reporting is required from us

FÖRSTÅR
INTE
RIKTIGT
BILDERNA...

KOPPLAT TILL
TRANSAKTIONS-
OCH
KOORDINATIONS-
KOSTNADER.
SENARE?



<http://www.practicalgrace.org/daily/wp-content/uploads/2009/01/exhausted-runner.jpg>

måndag 13 februari 12

And consider deployment.

It's often easier to manage smaller thing than big ones, right?

What would you rather do – small release, with small changes often, or big release with big changes seldom.

The answer seem obvious, but many companies is doing the complete opposite today.

“It hurts so I don't want to do it” should rather be
“It hurts – let's do it often, until the pain goes away”.

<CLICK>

For example – running. That hurts when you start – but if you don't continue to run often, it will continue to hurt.



http://www.cecscareers.com/images/youth_excited.jpg

måndag 13 februari 12

Shorter lead times – less work in process – will enhance our motivation.

If you know that people are waiting for your delivery we feel important and push ourselves to not miss that delivery.

But if the next step in the process has a 4-week queue we don't feel any reason to hurry, don't we.

<CLICK>

It's easy to establish a culture where it's ok to leave work un-done. "They will get back to us... we will know"

In this situation the motivation drops



måndag 13 februari 12

With quicker and more regular deliveries we will build trust.

Which in turn makes it easier to involve users, customer and stakeholders

Dan North ÖreDev 2009:

Hitchhikers guide to the Galaxy – Arthur Dent asks for a cup of tea.

That's how IT is perceived by business – a cost rather than value-adding

Liftarens guide till galaxen och när Arthur Dent ber rymdskeppet om en kopp te. Allt blir svart och tyst och efter en lång lång stund kommer rymdskeppet tillbaka online med något som smakar som diskvatten...

Limit work in process



måndag 13 februari 12

But how do we do it?

How can we limit the work in process?

Besides from the obvious: reducing the number of things.

Smaller Units Of Work



måndag 13 februari 12

Instead of working on big use cases that takes weeks or months to finish, maybe business value can be sliced into smaller parts and delivered incrementally.

As people often do with user stories in Scrum and XP.

User story slicing / Sashimi
Tracing bullet / Walking skeleton

Limit Work To Capacity



måndag 13 februari 12

Have you ever experienced that?

Not the donkey, but having someone pushing work onto you or your team and setting a deadline?

This builds queues and increases the amount of WIP.

You should instead limit the work to the capacity and let the team pull work when they've finished the work in process.

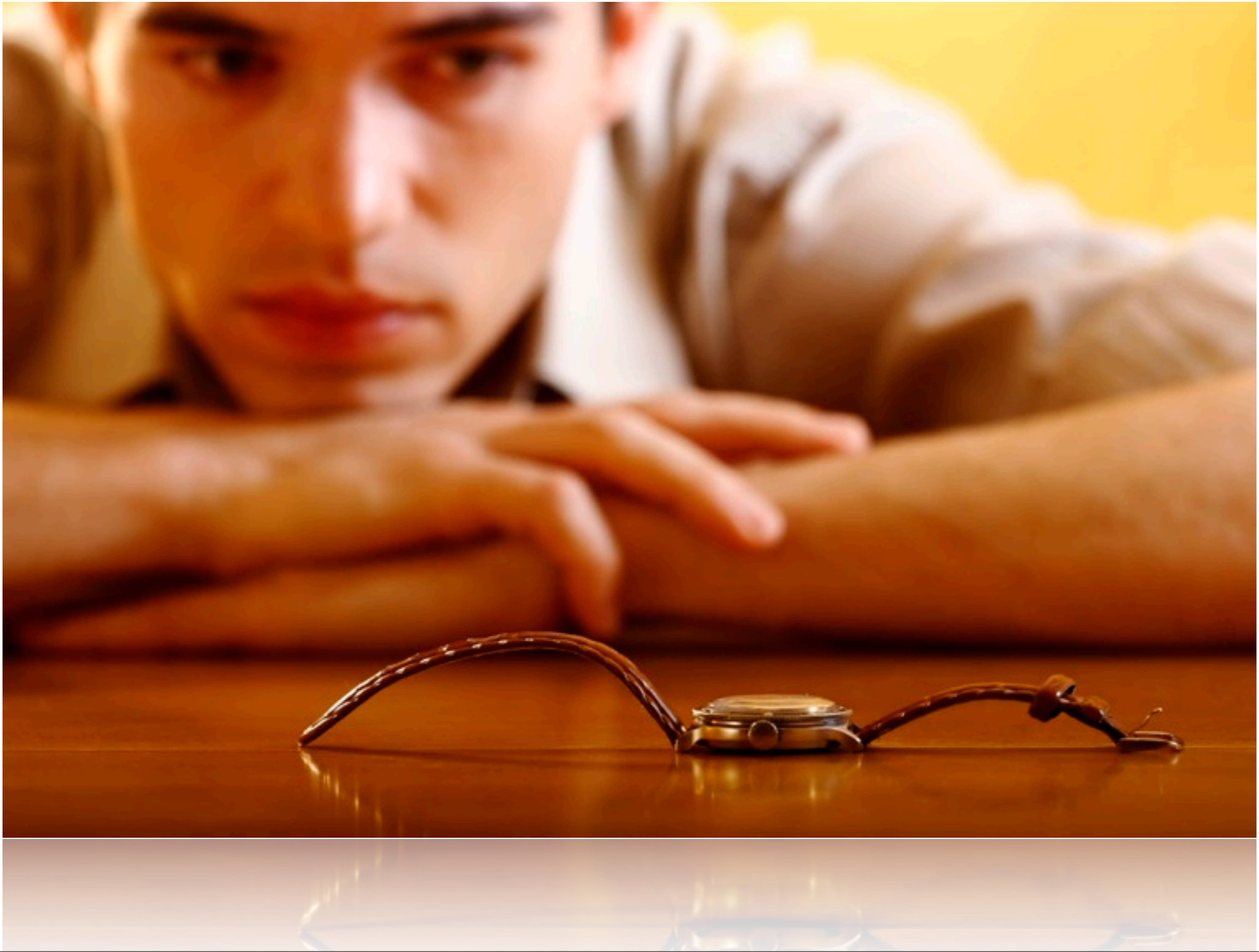
Less Rework



måndag 13 februari 12

Build Quality In from the start.

Less Waiting



måndag 13 februari 12

Hand-offs, gold-plating, bad specifications etc.

Because what do you typically do if you have to wait?

You pull new work and thereby increase the WIP and the lead time of all other work in process.

Remove Impediments



måndag 13 februari 12

Same thing with impediments.

Exploit and Elevate Bottlenecks



måndag 13 februari 12

And bottlenecks. I don't know if you're familiar with this terminology from Theory of Constraints? More on that later.

Cross-Functional Teams



måndag 13 februari 12

Another great way of achieving some of the things just mentioned is to work in cross-functional teams that have all the necessary skills needed to bring a customer request all the way to production and value for the customer. Then you reduce the risk of hand-offs, unavailable resources and so on.

Colonel John "Hannibal" Smith: Leader – strategic
Lieutenant Templeton "Face" Peck: Charmer and actor.
Sergeant Bosco "B.A." Baracus: The muscles!
and
Captain "Howlin' Mad" Murdock: Mad helicopter driver



måndag 13 februari 12

Mr. Hitoshi Yamada, 61 years old. He is a successor of Taiichi Ohno,

He is called "The Factory Reconstructor".

He receives requests and goes into the factory to undertake reformation of production lines and education of managers.

He comes to the production line.

As soon as he approaches the line, he shouts.

“Those are all work in process.”

He points out all work in process everywhere. one place after another.

"These are all waste, mountains of waste!"

“These are all work in process, and you know that it is a bad thing!”

“It is after a long time I saw this much waste.”

What would happen if Yamada-san came to your workplace in order to help you reduce work in process? Would he find any?

måndag 13 februari 12

In order to easier discover work in process, queues and impediments, we need to visualise our work. In order to visualise the work, we must understand how the work works.

For some organisations I've worked with this is a huge and immediate improvement because it means they have to explicitly define a process and policies that until now have been implicit. And in the process of doing so inconsistencies are dealt with, conflicting policies can be resolved and knowledge and good practices shared. It is suddenly possible to hold rational discussions of how to improve the policies.

TODO

ANALYZE

DEV

TEST

DONE

ORDER LIST
As an order clerk, I want to see a list of sortable orders, so that I can get a quick overview of the current situation

~~~~~  
~~~~~  
~~~~~

CHANGE ORDER NUMBER  
As a user, I want the order number to show the month in which the order was placed

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

måndag 13 februari 12

For a small development team, maybe it's as easy as this: analyze, develop, test/verify, and then you're done!



# Porta BUF SPRINT 1 2

Velocity 30  
Velocity 70

Klar

SPRINT 2.

UTV

Utest

Flödes test  
i utv

Paketering  
Klar

Promote  
klar

Systemtest  
i utv

A&D

Pågående

Klar

Pågående

Klar

Pågående

Klar

DB

INFA

Pågående

Klar

Stack of sticky notes on the left side of the board, mostly yellow and orange.

Sticky notes in the 'Pågående' column under UTV.

Vertical stack of sticky notes in the 'INFA' column, including orange ones with 'ST' and yellow ones with 'DB'.

Handwritten notes on the right side of the board:  
x Batch id!  
x Logging. w/  
→ JMS-connection.p



måndag 13 februari 12

Because every team is unique the visualisations will vary a great deal from one team to another.



Att göra

Pågående

BI-BAM

Test

KANBAN

Klart för Prod

Klart!

Att göra

Pågående

Test

Klart för Prod

Klart!

klintar!

urgent!





1.5 Orange

|             | 6 APPROVED     | 5 S. PENDING   | 6 IMPLEMENTED  | 5 T. PENDING   | RESOLVED       |
|-------------|----------------|----------------|----------------|----------------|----------------|
| TASKS       | [Sticky notes] | [Sticky notes] | [Sticky notes] | [Sticky notes] | [Sticky notes] |
| MAINTENANCE | [Sticky notes] | [Sticky notes] | [Sticky notes] | [Sticky notes] | [Sticky notes] |
| BAM         | [Sticky notes] | [Sticky notes] | [Sticky notes] | [Sticky notes] | [Sticky notes] |
| MAD         | [Sticky notes] | [Sticky notes] | [Sticky notes] | [Sticky notes] | [Sticky notes] |
| URGENT      | [Sticky notes] | [Sticky notes] | [Sticky notes] | [Sticky notes] | [Sticky notes] |

Timeline (bottom):

- URGENT CHANGE
- change ready
- PROD 1.4
- PROD 1.5
- PROD 1.6
- PROD 1.7
- PROD 1.8
- PROD 1.9
- PROD 1.10
- PROD 1.11
- PROD 1.12
- PROD 1.13
- PROD 1.14
- PROD 1.15

Orange Kiwi

| ENV    | LEV  | SAS |
|--------|------|-----|
| PROD   | Low1 | 1.3 |
| VER    | 23/1 | 1.4 |
| TEST 1 |      | 1.4 |
| TEST 2 | Low3 | 1.5 |

Legend:

- BUGS
- NORMAL
- URGENT

Graphs:

måndag 13 februari 12



|                                                |                                               |  |  |                                           |
|------------------------------------------------|-----------------------------------------------|--|--|-------------------------------------------|
| 11 / 17                                        | TEAM BERÄKNING FAST                           |  |  |                                           |
| (5) 97/97<br>(ET) 1/1<br>(TWS) ?<br>(KBT) ?    | TEAM BATCH & EXT. TJÄNSTER                    |  |  | Produkt<br>batch 276<br>batch<br>artiklar |
| 5/8                                            | TEAM GUI & LOKALA UTSKRIFTER (DR)             |  |  | LOKAL<br>DATA-ASSIST<br>XL                |
| KLINO: 122/140<br>ETAB: 154/155<br>SART: 69/70 | TEAM GUI-TEST<br>SERVA<br>GUI-TEST<br>DR-ETAB |  |  | SERVA<br>AUT-SERVE<br>DR-<br>KAS          |
|                                                | CM & DBA                                      |  |  | My System<br>DR-TEST                      |
|                                                | FÖRVALTNING                                   |  |  |                                           |

måndag 13 februari 12

This is a board for several teams in a larger project.

The board is used as a progress indicator ... like a Scrum-of-Scrums

Note the traffic lights

And the difference in backlogs/approaches for each team



# Kanban

Make  
Work  
Visible

Limit  
Work in  
Process

Help  
work to  
flow

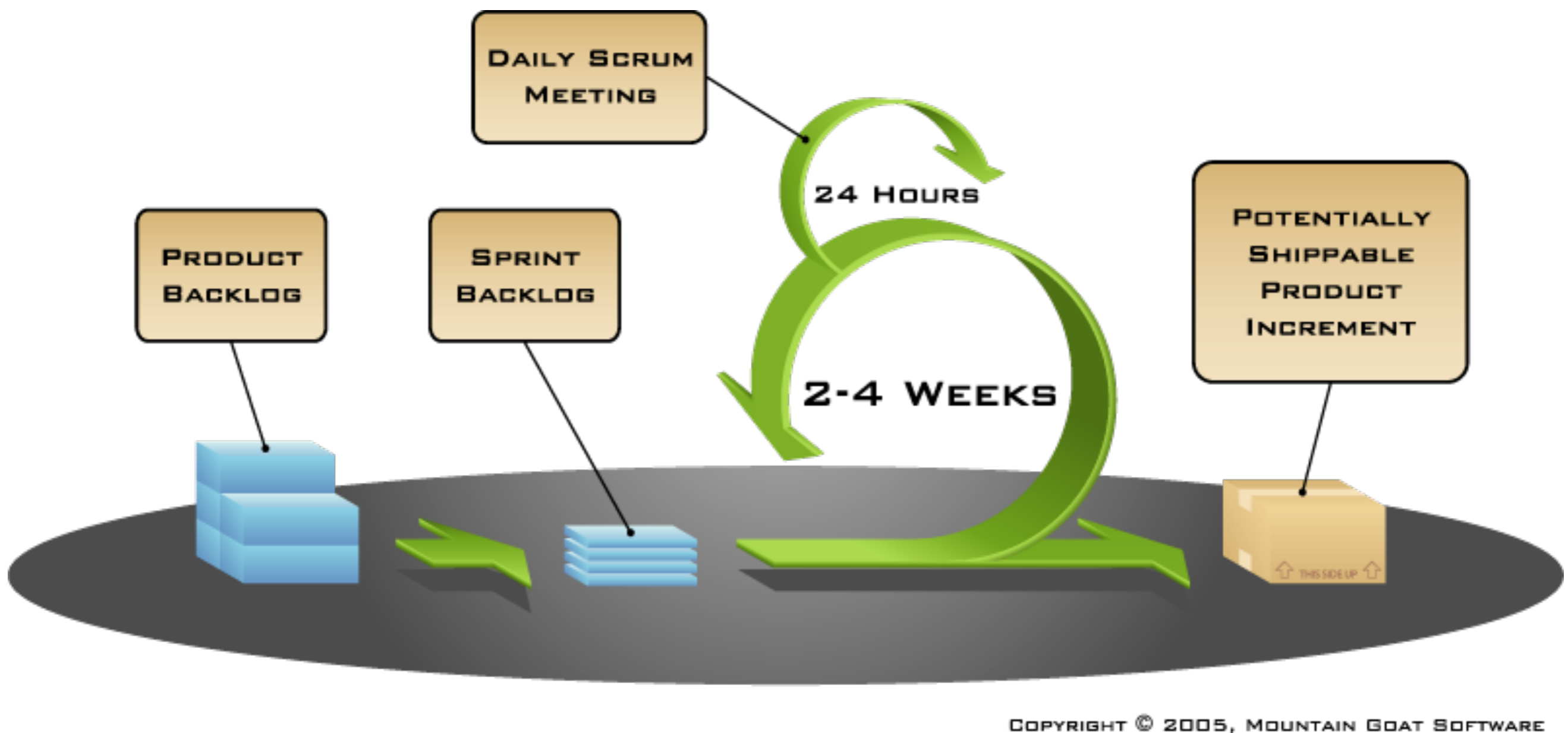
Source: <http://www.kanban101.com/>

måndag 13 februari 12

So, in summary – and this particular summary is from Janice Linden Reed’s excellent Kanban101 site – you can say that Kanban is all about making the work visible, limit work in progress and help the work to flow and to flow fast, i.e., focus on reducing lead times.

These principles can probably be applied to whatever method you’re currently working with, whether it is ad hoc, Scrum or waterfall. Chances are you’re already applying them to a certain degree.





måndag 13 februari 12

Do you know Scrum?

Scrum can in a way be seen as an implementation of the principles I've talked about.

Do you see it?

Sprints = limit WIP.

Definition of done, potentially shippable.

Breakdown of work in small, vertical slices of shippable value.

Cross-functional teams to minimise waiting and help flow.

PO present. Less hand-offs and waiting, specify requirements Just-in-time etc.

**Sprint planning = sort of limiting work to capacity**

Visualisation = Scrum board, burndown

To take something from requirement to potentially shippable inside of a timebox can be seen as a way of focusing on flow to bring problems and waste to the surface. That's a not uncommon way of describing Scrum.

**TDD** = limit the amount of unverified code through short test-and-implement-cycles.



# Why Should I Use Kanban?



måndag 13 februari 12

If many agile methods already incorporates Kanban principles, why should I use Kanban?

**KANBAN ÄR INTE EN METOD!**





måndag 13 februari 12

Where other agile methods, e.g. Scrum, isn't a natural fit, e.g., operations and maintenance. These teams often have to be very responsive and it's difficult for them to plan work for a 4 week timebox and then shut the door so to speak. They don't have one clear product owner and there is no point in batching the work for a review or demo with stakeholders – it has to be reviewed and deployed immediately.

Some more normal development teams don't see a need to use time intervals as a WIP limit. How often does a customer order "4 weeks of software please".

Maybe you don't see a reason to use the same intervals, the same cadence, for planning work as you do for reviewing or releasing it or reflecting how to improve, i.e., retrospectives. But if you're doing Scrum and start using different cadences, you can't call it Scrum anymore because it will upset a bunch of guys with beards.

That's probably one of the reasons why Kanban is becoming so popular these days. The minute you start planning the work just-in-time, you have to stop calling it Scrum.





måndag 13 februari 12

Reached plateau with improvement with your current method. Need inspiration. New tools/ techniques. Metrics. Focus on reducing WIP and shortening lead times.

An understanding of the effects of WIP and flow have helped many Scrum teams to work more as a team, see bottlenecks, and so on.





**¡VIVA LA EVOLUCIÓN!**

måndag 13 februari 12

Evolutionary introduction of agile and lean ideas in hostile or conservative environments. Or in any environment for that matter.

Kanban can be seen as an agile approach to introducing agile ideas. You simply start where you are and take it from there. Introducing Scrum or XP is often more of a big design up front approach, where people are expected to take a leap of faith and hope for the best.

In many cases not only people working with the software development are expected to change but also the business, just because someone in IT wants to try out a new method.



PLANNING  
4

DOING  
2

DONE!

KANBAN IN  
PRACTICE

WRAP UP &  
DISCUSSION

DISCUSSION &  
QUESTIONS

BOOK HOTEL  
& DO  
TRAVEL ARR.

INTRODUCTIONS  
THE DOT

KANBAN  
PRINCIPLES



# PLANNING 4

# DOING 2

# DONE!

KANBAN IN  
PRACTICE



BOOK HOTEL  
& DO  
TRAVEL ARR.

DISCUSSION &  
QUESTIONS

INTRODUCTIONS

THE DOT

KANBAN  
PRINCIPLES

WRAP UP &  
DISCUSSION



# Kanban in practice

Community, jmf user stories, scrum board i Scrum

Marcus



måndag 13 februari 12

So, let's look at how we do this in practice. Because there is actually more to Kanban than these basics.

The things we now will talk about are based on common practices in the Kanban community and should be regarded as examples rather than a blue print. This is one, albeit common, way to be working with Kanban.



TODO

DOING

DONE

ORDER LIST

As an order clerk, I want to see a list of sortable orders, so that I can get a quick overview of the current situation.

~~~~~  
~~~~~  
~~~~~

CHANGE ORDER NUMBER

As a user, I want the order number to show the month in which the order was placed.

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

måndag 13 februari 12

One of the good things of Kanban is that it's an evolution, not a revolution. You can adapt Kanban step by step.

Start with visualizing your workflow. In our example here we already have a Scrum board. So we start from there and just change it to better reflect our actual workflow.



TODO

DOING

DONE

ORDER LIST

As an order clerk, I want to see a list of sortable orders, so that I can get a quick overview of the current situation.

~~~~~  
~~~~~  
~~~~~

CHANGE ORDER NUMBER

As a user, I want the order number to show the month in which the order was placed.

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

måndag 13 februari 12

So a good starting point could be to visualize your workflow. If you are working with Scrum you are doing that already.

But maybe your workflow could be a little more defined than Todo, Doing and Done. What steps are you actually doing?

<<Talk about more complex orgs where visualisation is the main benefit.>>



# TODO

# DOING

# DONE

ORDER LIST  
As an order clerk, I want to see a list of sortable orders, so that I can get a quick overview of the current situation.

~~~~~  
~~~~~  
~~~~~

CHANGE ORDER NUMBER
As a user, I want the order number to show the month in which the order was placed.

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

måndag 13 februari 12

Make room for some more columns



# TODO

# DOING

# DONE

ORDER LIST  
As an order clerk, I want to see a list of sortable orders, so that I can get a quick overview of the current situation

~~~~~  
~~~~~  
~~~~~

CHANGE ORDER NUMBER
As a user, I want the order number to show the month in which the order was placed

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~


TODO

DOING

DONE

ORDER LIST
As an order clerk, I want to see a list of sortable orders, so that I can get a quick overview of the current situation.

~~~~~  
~~~~~  
~~~~~

CHANGE ORDER NUMBER  
As a user, I want the order number to show the month in which the order was placed.

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

måndag 13 februari 12
and move the "Done" items to that column

TODO

ANALYZE

DONE

ORDER LIST
As an order clerk, I want to see a list of sortable orders, so that I can get a quick overview of the current situation.

~~~~~  
~~~~~  
~~~~~

CHANGE ORDER NUMBER  
As a user, I want the order number to show the month in which the order was placed.

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

måndag 13 februari 12

Maybe you start by analyzing a story, so lets a put a column to represent that work.

- what's included
- how should it be solved
- break down into task
- etc.

TODO

ANALYZE

DEV

DONE

ORDER LIST
As an order clerk, I want to see a list of sortable orders, so that I can get a quick overview of the current situation

~~~~~  
~~~~~  
~~~~~

CHANGE ORDER NUMBER  
As a user, I want the order number to show the month in which the order was placed

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

måndag 13 februari 12

And then we'll add a column for the development of the story



TODO

ANALYZE

DEV

TEST

DONE

ORDER LIST

As an order clerk, I want to see a list of sortable orders, so that I can get a quick overview of the current situation

Handwritten scribbles representing a list of items.

CHANGE ORDER NUMBER

As a user, I want the order number to show the month in which the order was placed

Handwritten scribbles representing a list of items.

Handwritten scribbles representing a list of items.

Handwritten scribbles representing a list of items.

Handwritten scribbles representing a list of items.

Handwritten scribbles representing a list of items.

Handwritten scribbles representing a list of items.

Handwritten scribbles representing a list of items.

Handwritten scribbles representing a list of items.

Handwritten scribbles representing a list of items.

måndag 13 februari 12

And finally we need to verify and test the feature so that we are sure that the quality is good enough for our team to release.



TODO

ANALYZE

DEV

TEST

DONE



måndag 13 februari 12

One way of handling handoffs between specialists is to introduce queues so that work is buffered. In this way the next step can see when things are ready to be pulled from the Done-queue of the previous step.

It doesn't have to be hard specialization – even if it's roughly the same people doing the work this can be useful to visualize the flow. So that we can see where impediments, unevenness and other problems occur more explicit on the board



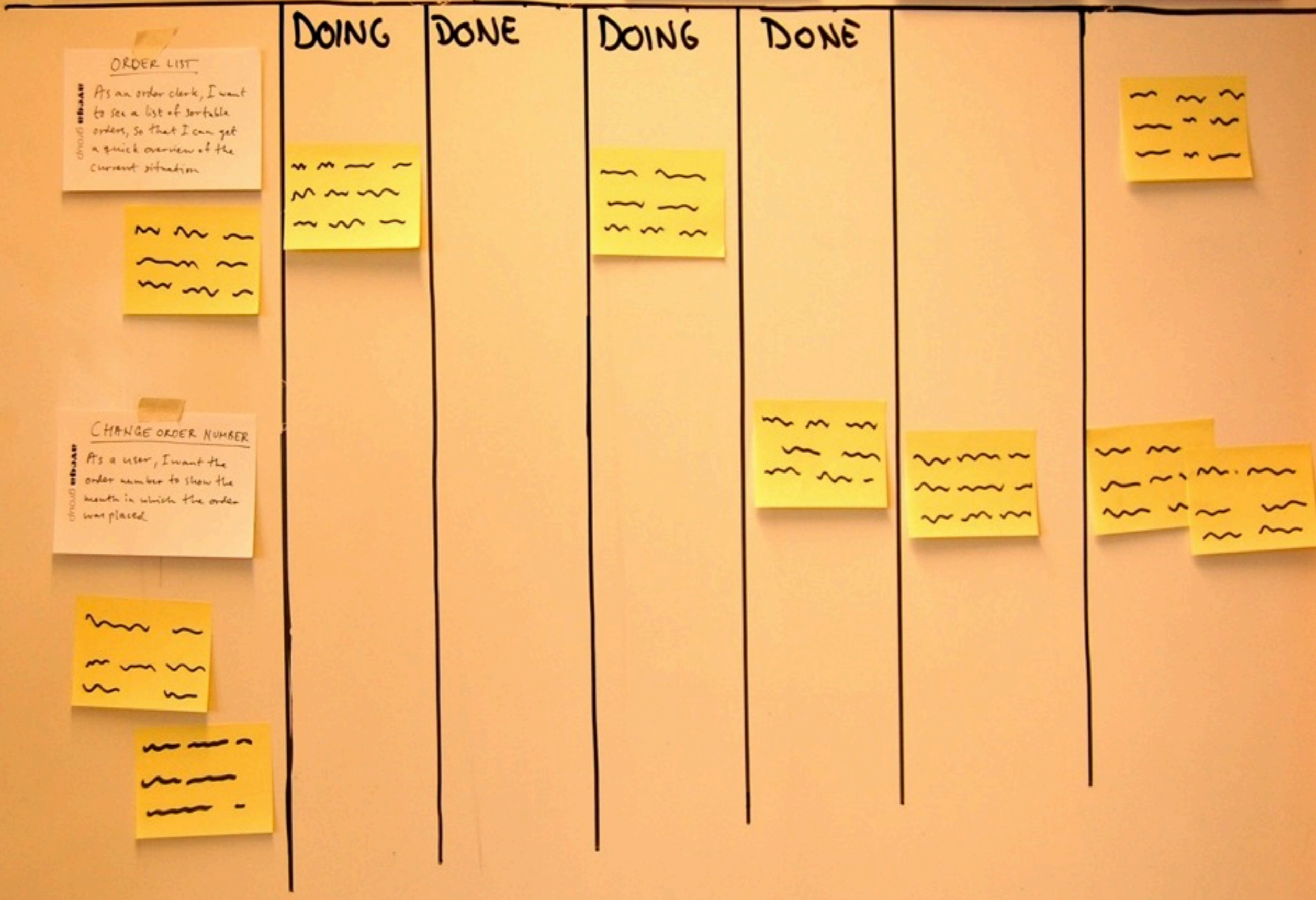
TODO

ANALYZE

DEV

TEST

DONE



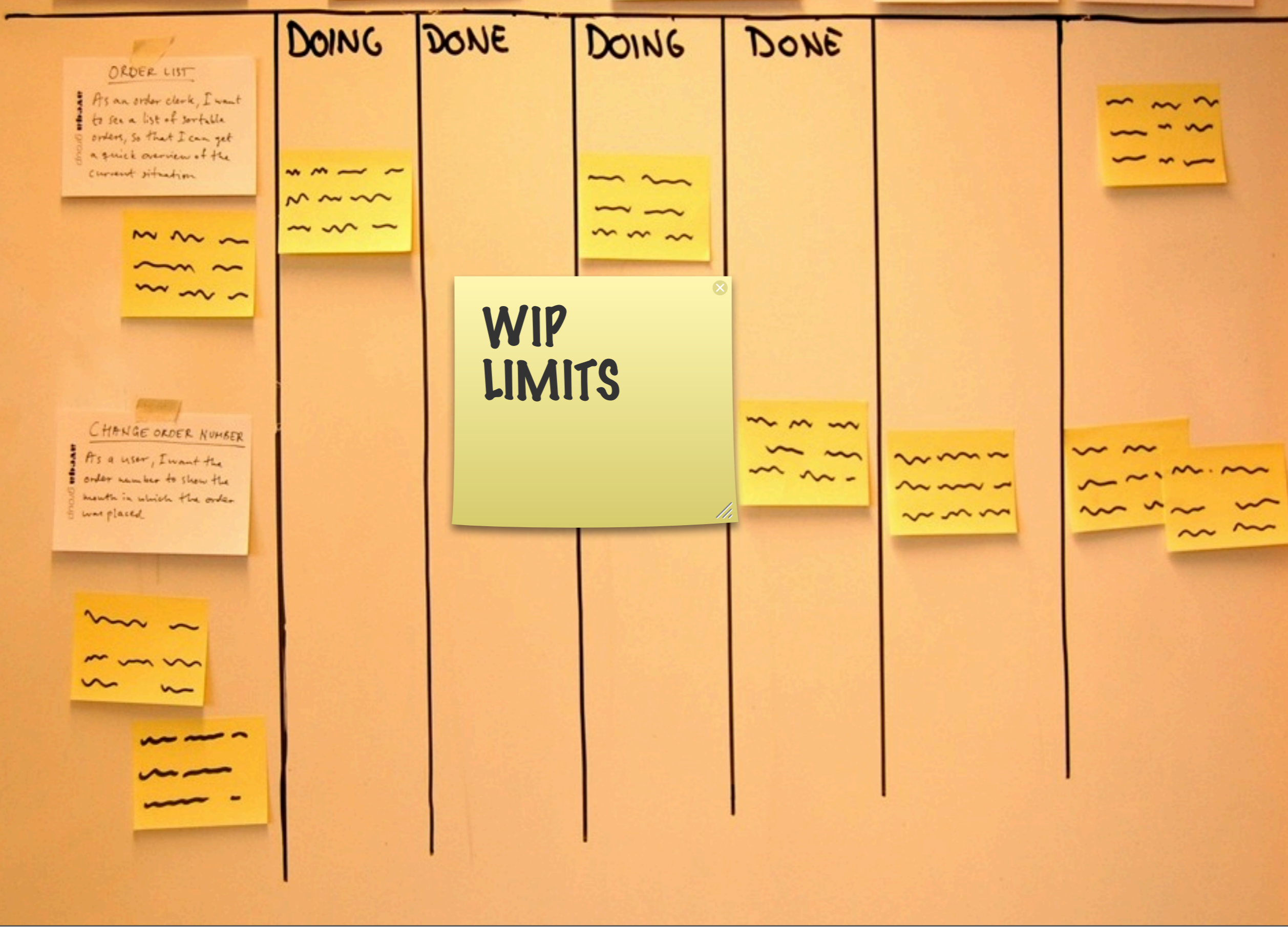
måndag 13 februari 12

Let's add a done-queue for development, to give a signal to Test when new work is ready to pull.

In this case we added the Done-queue in the Development-step. You could also argue that it should be in the Test-step and be called "Ready for test" or something.



TODO ANALYZE DEV TEST DONE



måndag 13 februari 12

The next step is maybe to just agree on a policy to prefer completing work and help others complete work instead of just starting new work.

**Stop starting and start finishing.**

Maybe you all try to work on only one item at a time.

Soon you want to make the work in progress limits more explicit by setting a maximum number of items in progress.

- There are many different approaches to come up with that number.
- It could be 2 per person in the team, so that you don't have to be idle if one item is blocked. Or a bit less, perhaps it is unlikely that everyone is blocked at the same time.
  - For an experienced Scrum team that practices pair-programming maybe it is the team size divided by two.

A lower number is generally better than a higher since we want to limit the WIP as much as possible to get better lead times, faster feedback and forcing yourself to remove impediments. But if your organization is not capable of dealing with relentless continuous improvement, having a low limit will cause the flow to stall and the limits might not be respected anyway.

Be sure not to fall into the trap of not setting any WIP-limit at all! Kanban is just 3 rules - don't remove 1 of them this soon :)

- Any number! We will change it as we see problems and opportunities.





måndag 13 februari 12

In this example we have a WIP limit of 2 for Analyze, 3 for Development and 2 for Test.

(No limit on ToDo in this example.)

Let's look at the board in action and see how the visualization and WIP limits help the team.



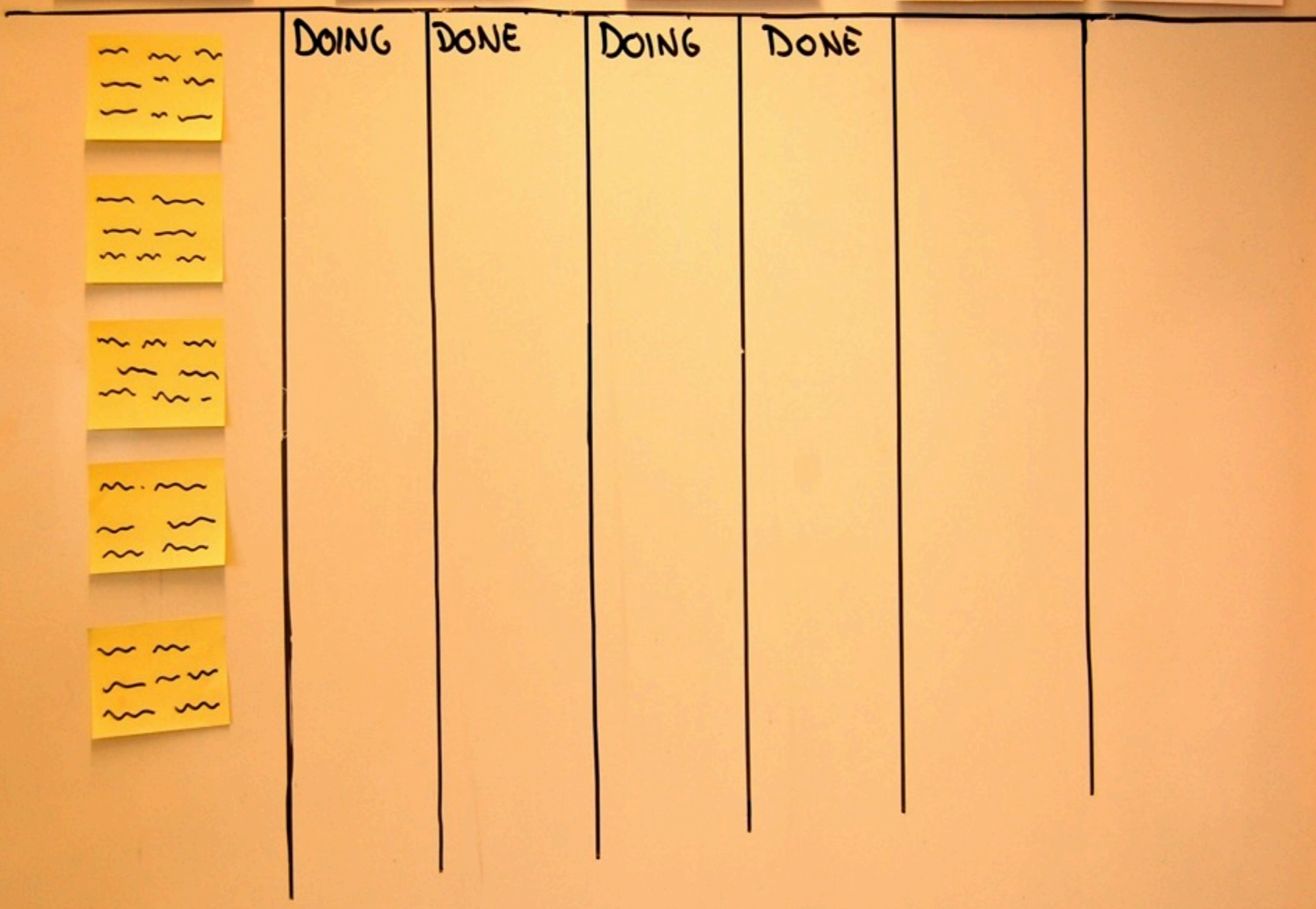
TODO

ANALYZE  
2

DEV  
3

TEST  
2

DONE



måndag 13 februari 12

So a normal flow to the board is not too hard to imagine right?  
Here's the initial stage...



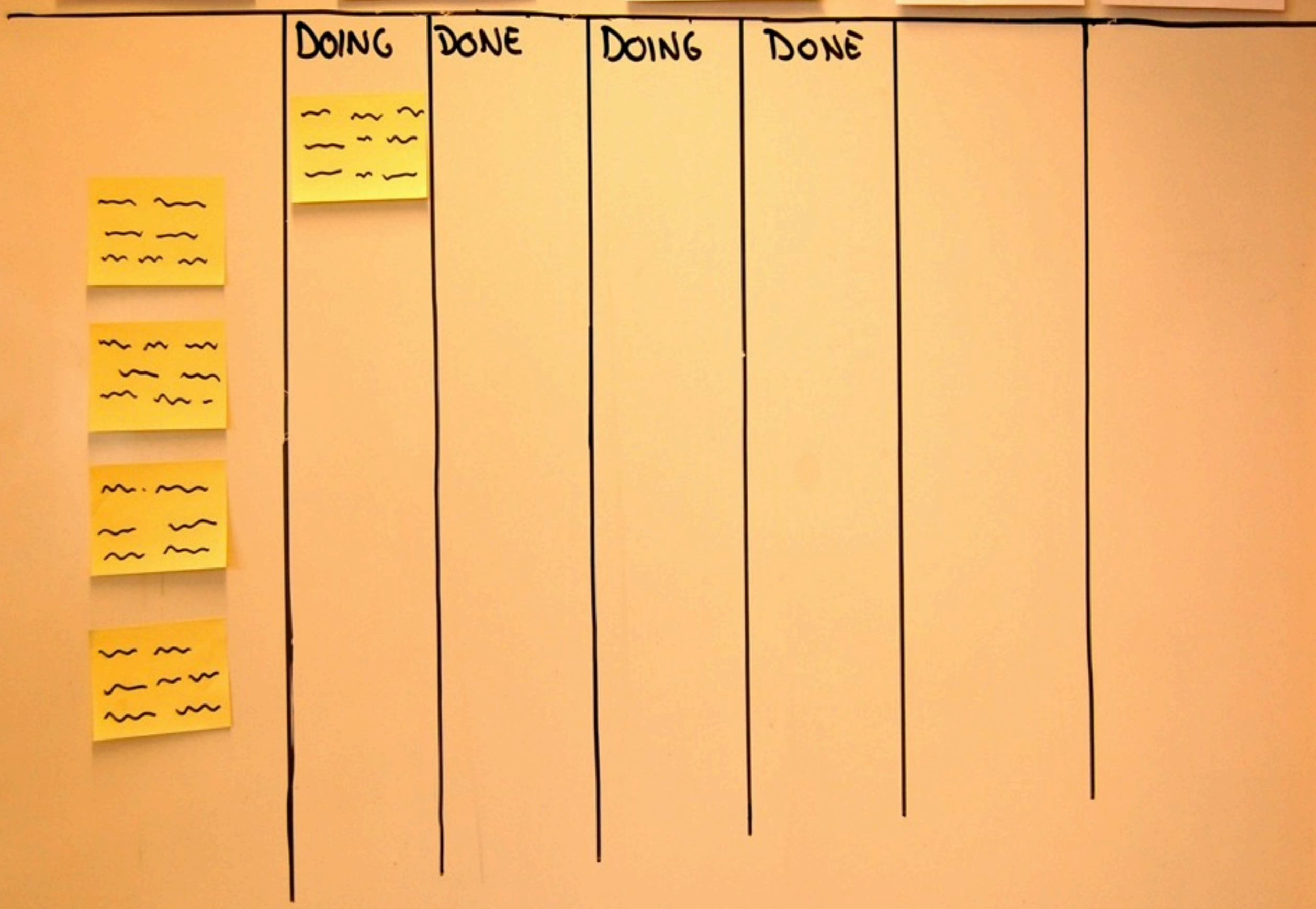
TODO

ANALYZE  
2

DEV  
3

TEST  
2

DONE



måndag 13 februari 12

Work is **pulled** from the backlog into the Analyze column as the WIP-limit allow.



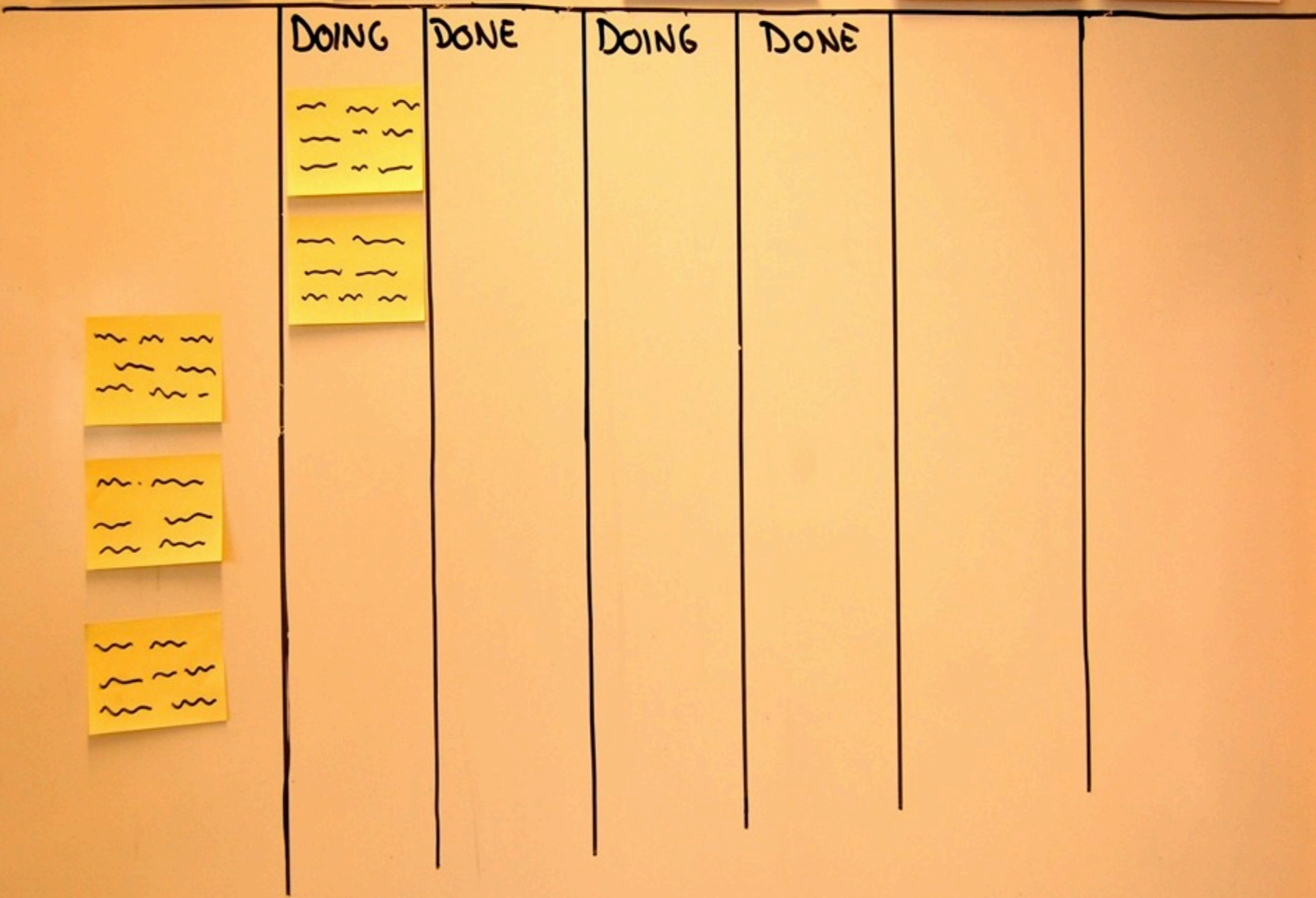
TODO

ANALYZE  
2

DEV  
3

TEST  
2

DONE



måndag 13 februari 12

One could wonder what Dev and Test is doing in these early stages of the flow... They should of course be helping out, but in practice this does not always work.

Well, the start will get a bit bumpy since no work is ready to be pulled. But this is for the very first time only, since a Kanban board never will be reset. In Scrum, for example, this situation is likely to occur in each sprint.



TODO

ANALYZE  
2

DEV  
3

TEST  
2

DONE

DOING

DONE

DOING

DONE

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

måndag 13 februari 12

As work progress you monitor the flow of items.

TODO

ANALYZE
2

DEV
3

TEST
2

DONE

DOING

DONE

DOING

DONE

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

måndag 13 februari 12

Follow the WIP limits and only pull new work to your capacity



TODO

ANALYZE  
2

DEV  
3

TEST  
2

DONE

DOING

DONE

DOING

DONE

~ ~ ~  
~ ~ ~  
~ ~ ~

~ ~ ~  
~ ~ ~  
~ ~ ~

~ ~ ~  
~ ~ ~  
~ ~ ~

~ ~ ~  
~ ~ ~  
~ ~ ~

~ ~ ~  
~ ~ ~  
~ ~ ~



TODO

ANALYZE  
2

DEV  
3

TEST  
2

DONE

DOING

DONE

DOING

DONE

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~


TODO

ANALYZE
2

DEV
3

TEST
2

DONE

DOING

DONE

DOING

DONE

~ ~ ~
~ ~ ~
~ ~ ~

~ ~ ~
~ ~ ~
~ ~ ~

~ ~ ~
~ ~ ~
~ ~ ~

~ ~ ~
~ ~ ~
~ ~ ~

~ ~ ~
~ ~ ~
~ ~ ~

måndag 13 februari 12
As we can see...

TODO

ANALYZE
2

DEV
3

TEST
2

DONE

DOING

DONE

DOING

DONE

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

måndag 13 februari 12

... work is flowing along nicely since...



TODO

ANALYZE  
2

DEV  
3

TEST  
2

DONE

DOING

DONE

DOING

DONE

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

måndag 13 februari 12

... each step is only pulling work when it's done ...

TODO

ANALYZE
2

DEV
3

TEST
2

DONE

DOING

DONE

DOING

DONE

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

måndag 13 februari 12

... and not over their capacity.



TODO

ANALYZE  
2

DEV  
3

TEST  
2

DONE

DOING

DONE

DOING

DONE

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~


TODO

ANALYZE
2

DEV
3

TEST
2

DONE

DOING

DONE

DOING

DONE

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

måndag 13 februari 12

Tam-tam-tam - working with Kanban is a breeze!



TODO

ANALYZE  
2

DEV  
3

TEST  
2

DONE

DOING

DONE

DOING

DONE

~ ~  
~ ~  
~ ~

~ ~  
~ ~  
~ ~

~ ~ ~  
~ ~ ~  
~ ~ ~

~ ~ ~  
~ ~ ~  
~ ~ ~

~ ~  
~ ~  
~ ~



TODO

ANALYZE  
2

DEV  
3

TEST  
2

DONE

DOING

DONE

DOING

DONE

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~


TODO

ANALYZE
2

DEV
3

TEST
2

DONE

DOING

DONE

DOING

DONE

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~



TODO

ANALYZE  
2

DEV  
3

TEST  
2

DONE

DOING

DONE

DOING

DONE

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

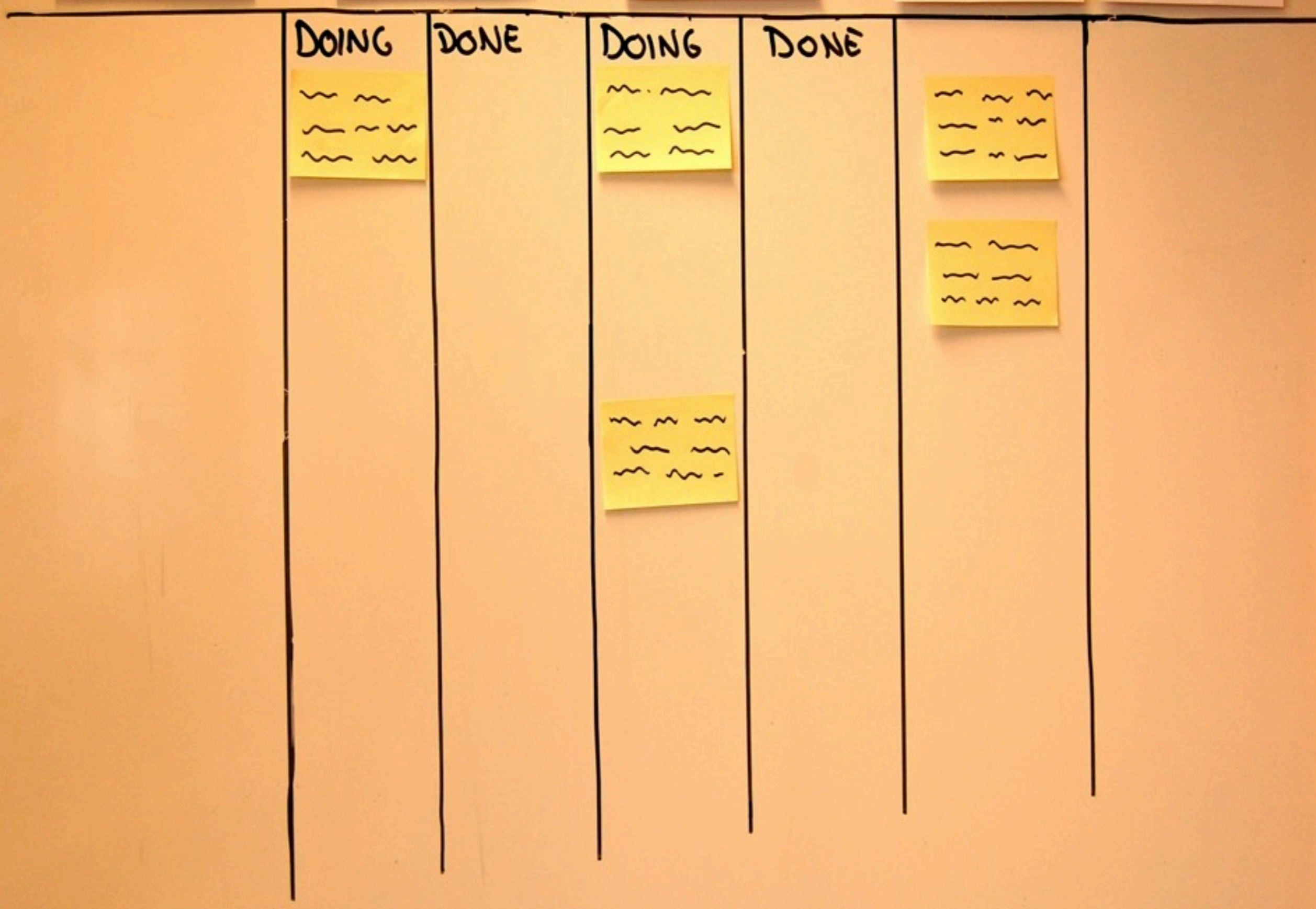

TODO

ANALYZE
2

DEV
3

TEST
2

DONE



måndag 13 februari 12

We've started work on the last feature in our backlog.

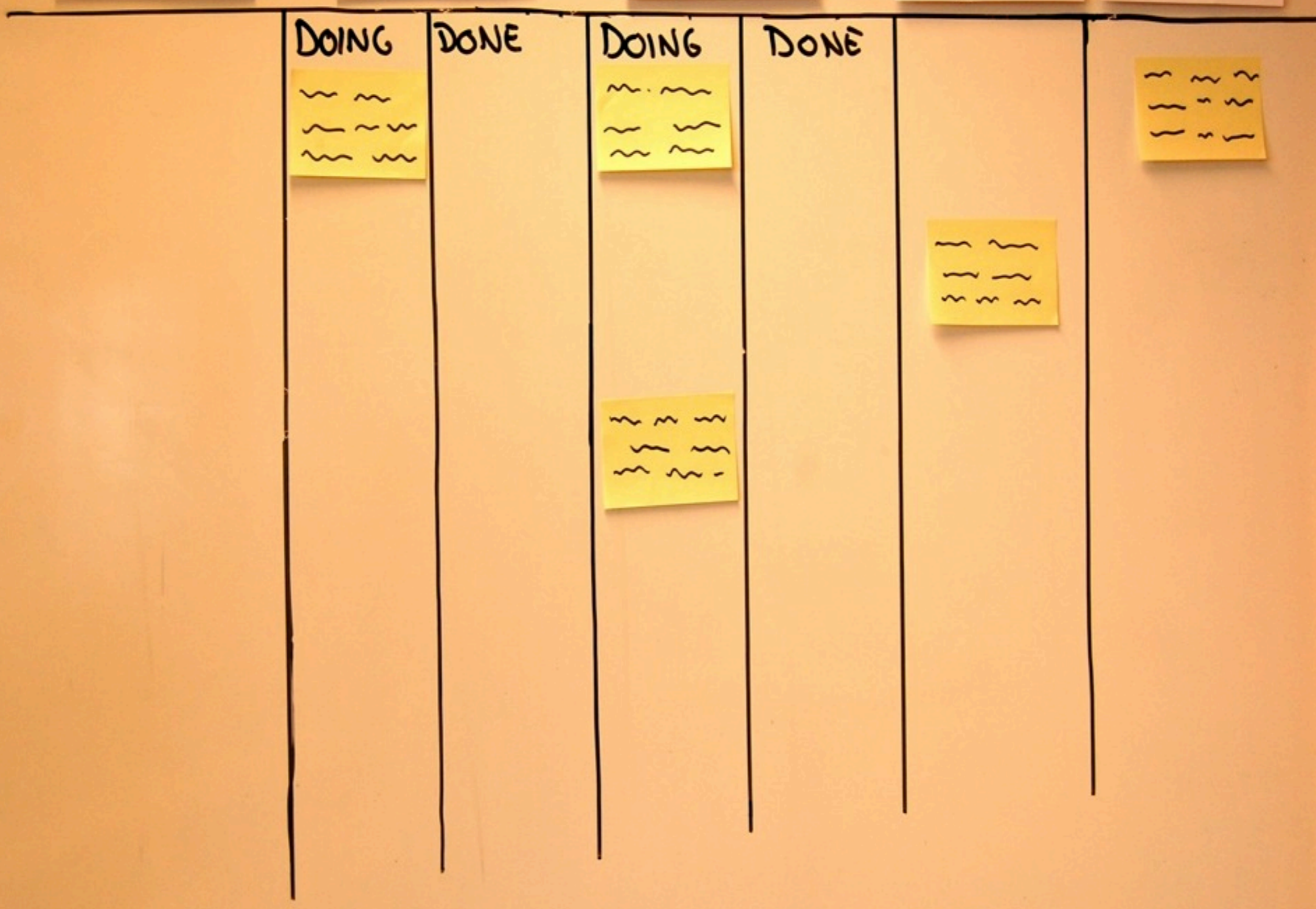
TODO

ANALYZE
2

DEV
3

TEST
2

DONE



måndag 13 februari 12

About the same time as the first feature is ready to be launched!

Yeah! Champagne to all!
Or cake at least...

TODO

ANALYZE
2

DEV
3

TEST
2

DONE

DOING

DONE

DOING

DONE

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~



TODO

ANALYZE  
2

DEV  
3

TEST  
2

DONE

DOING

DONE

DOING

DONE

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~


TODO

ANALYZE
2

DEV
3

TEST
2

DONE

DOING

DONE

DOING

DONE

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~



TODO

ANALYZE  
2

DEV  
3

TEST  
2

DONE

DOING

DONE

DOING

DONE

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

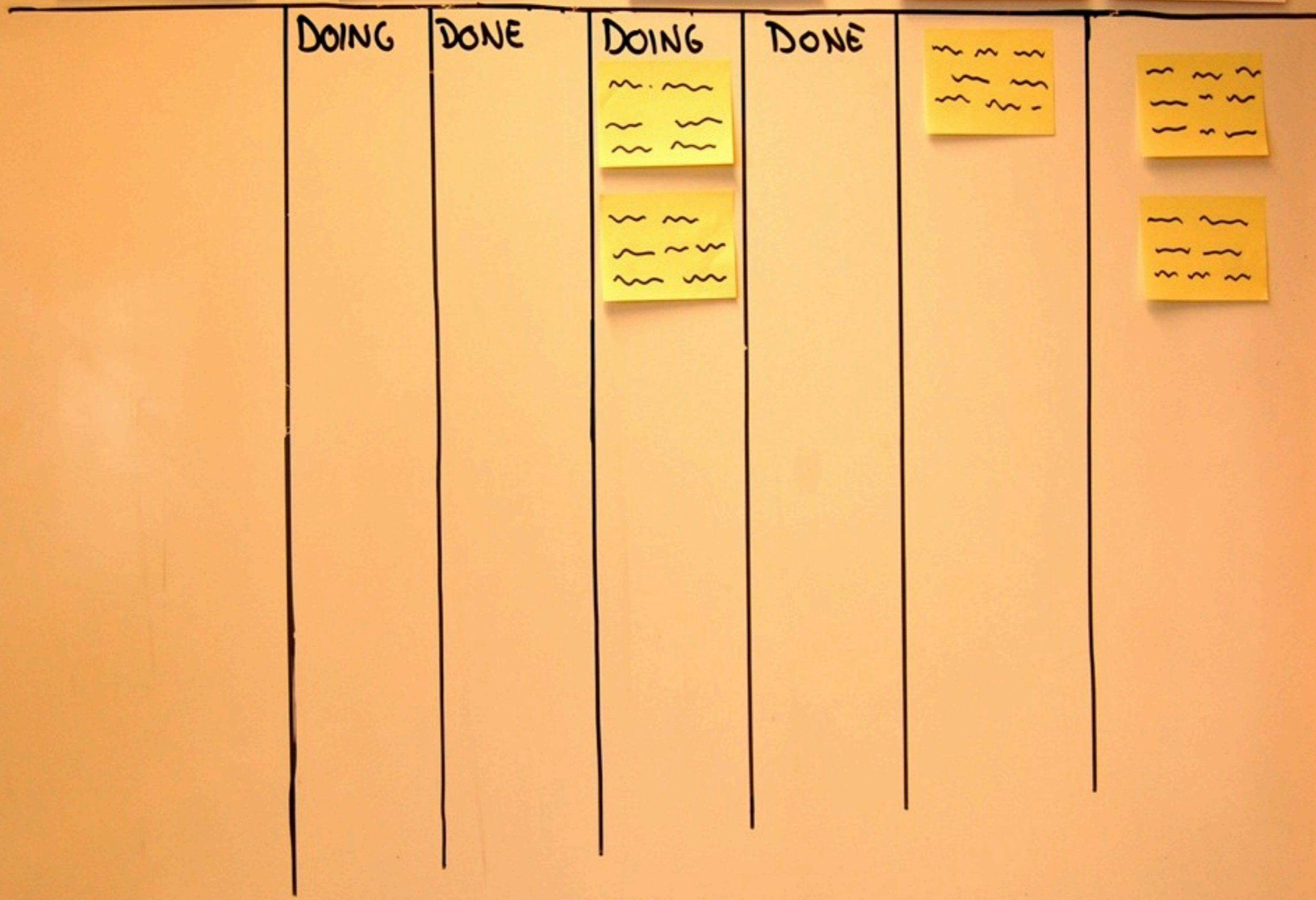

TODO

ANALYZE
2

DEV
3

TEST
2

DONE



måndag 13 februari 12

And we've pulled the last feature into development.

TODO

ANALYZE
2

DEV
3

TEST
2

DONE

DOING

DONE

DOING

DONE

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~



TODO

ANALYZE  
2

DEV  
3

TEST  
2

DONE

DOING

DONE

DOING

DONE

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~


TODO

ANALYZE
2

DEV
3

TEST
2

DONE

DOING

DONE

DOING

DONE

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~



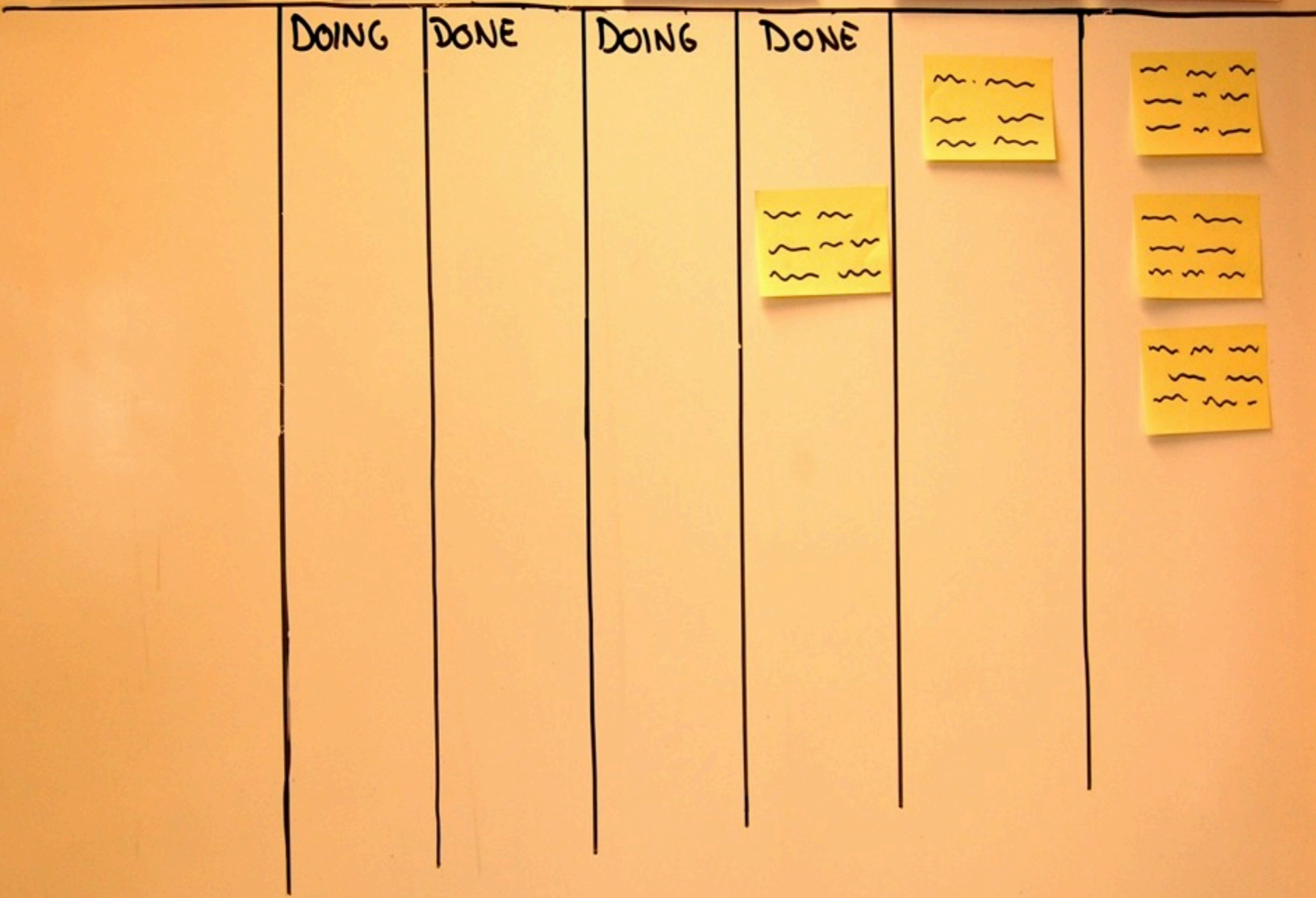
TODO

ANALYZE  
2

DEV  
3

TEST  
2

DONE



måndag 13 februari 12

Now the development of the last feature is done and we're ready to do the last bit of testing



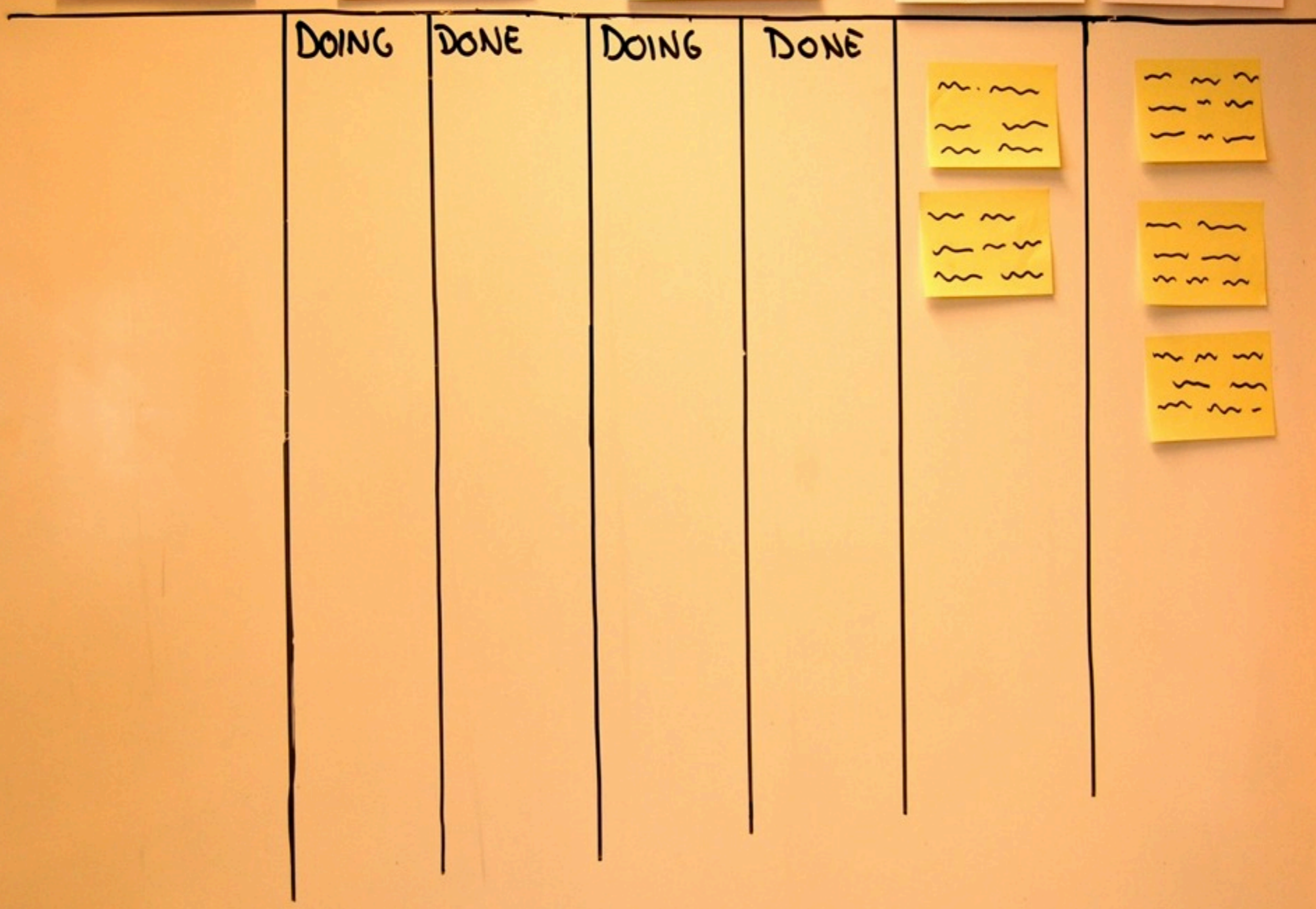
TODO

ANALYZE  
2

DEV  
3

TEST  
2

DONE



måndag 13 februari 12

The testers are working along to bring our goodness to the customers.



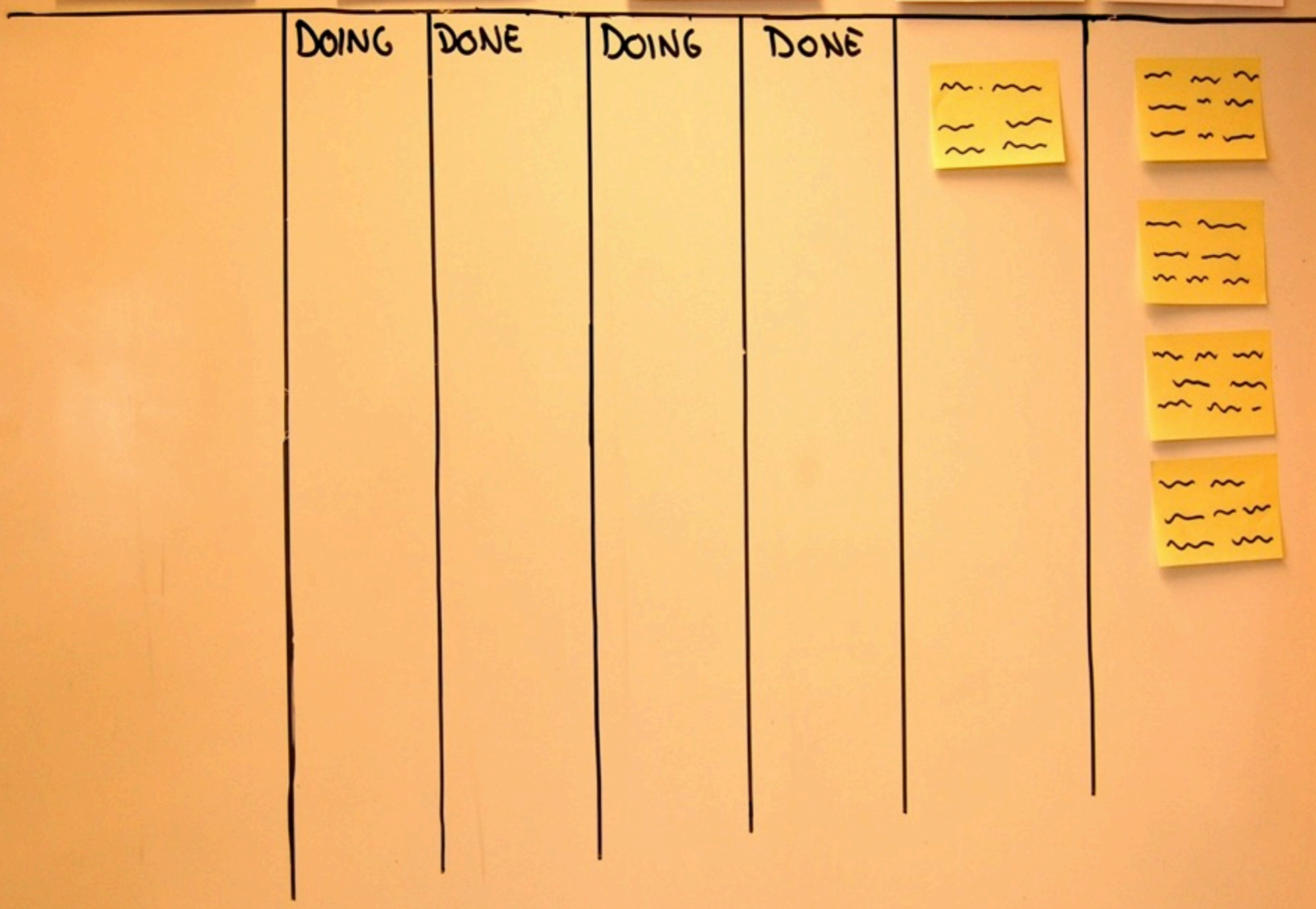
TODO

ANALYZE  
2

DEV  
3

TEST  
2

DONE



måndag 13 februari 12

Closing in on the last feature.



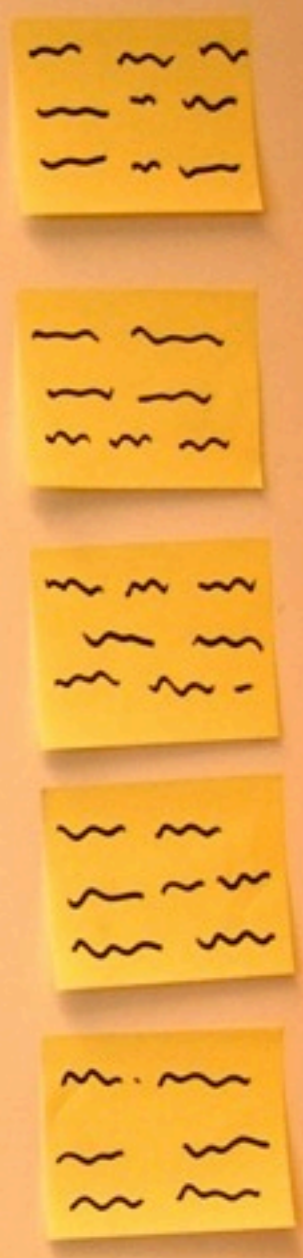
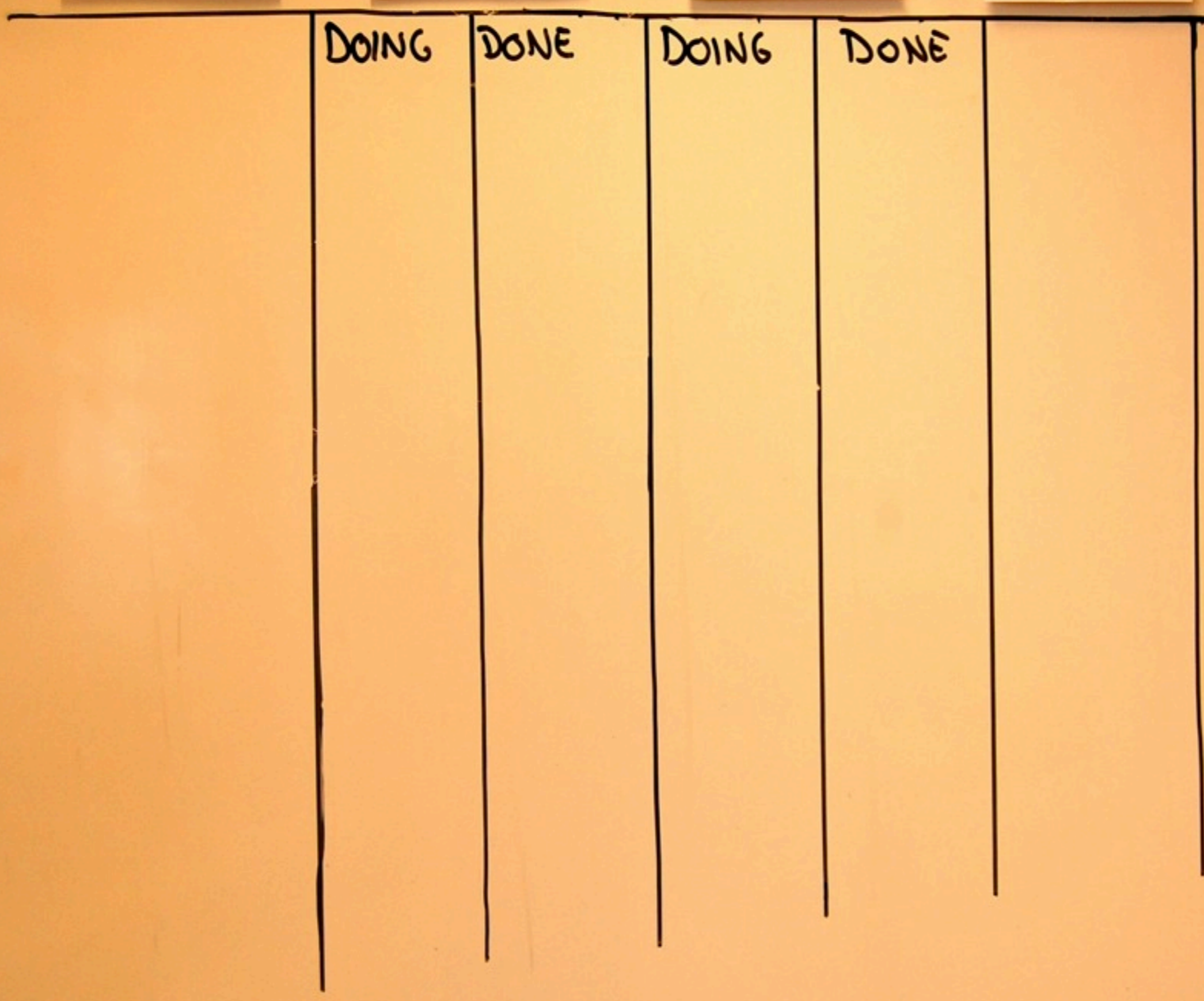
TODO

ANALYZE  
2

DEV  
3

TEST  
2

DONE



måndag 13 februari 12

And we're done!

Ok - that wasn't so exciting right?

Let's instead take a look on how our Kanban board can and should evolve over time. It's not done - strive to find way to improve your process.



TODO

ANALYZE  
2

DEV  
3

TEST  
2

DONE



måndag 13 februari 12

Here is a situation where it's not possible to share work and all Developers are busy on one feature each.



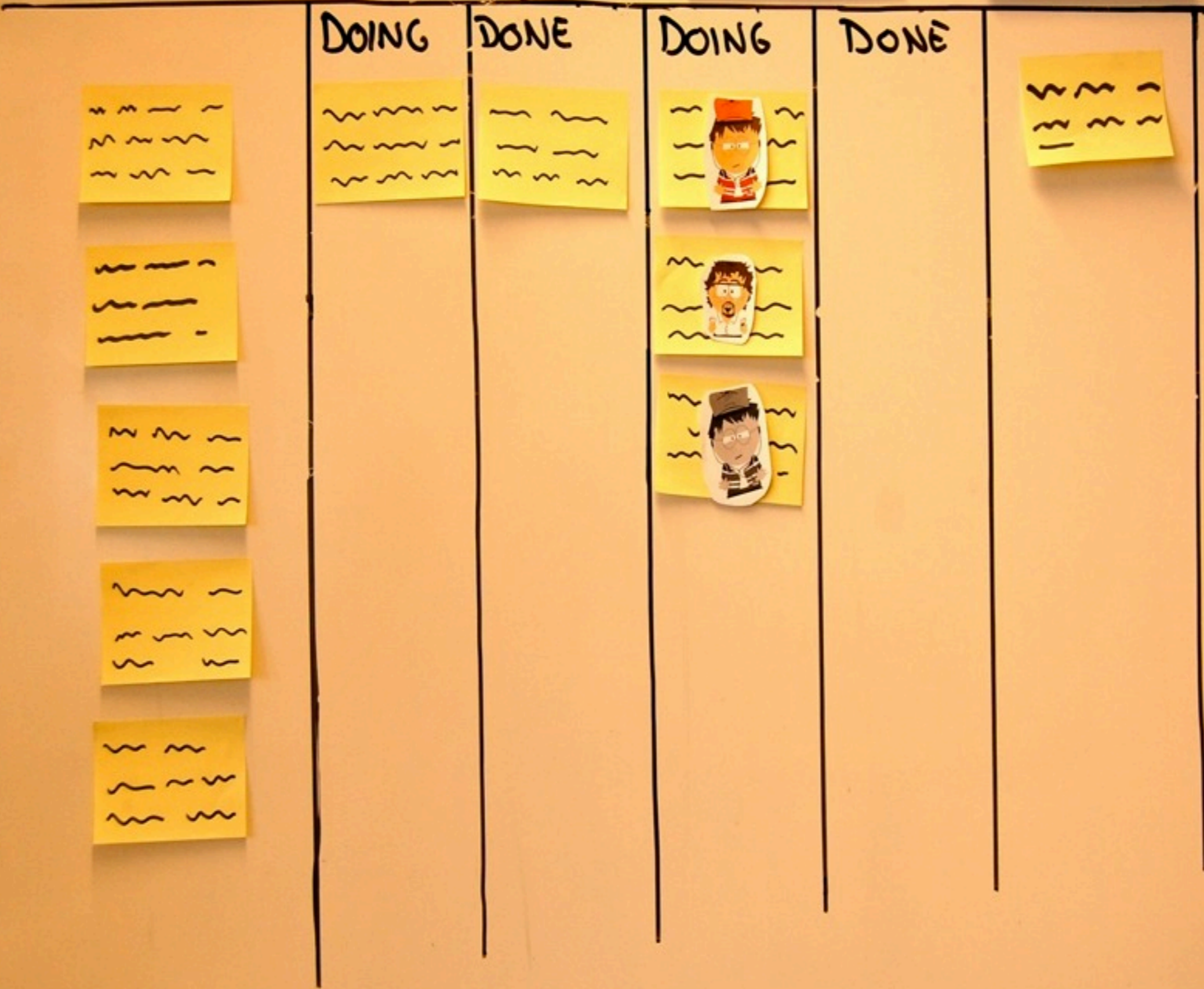
TODO

ANALYZE  
2

DEV  
3

TEST  
2

DONE



måndag 13 februari 12

And suddenly a new Developer is added to the team. Mikaela is newly employed.

What should she do?



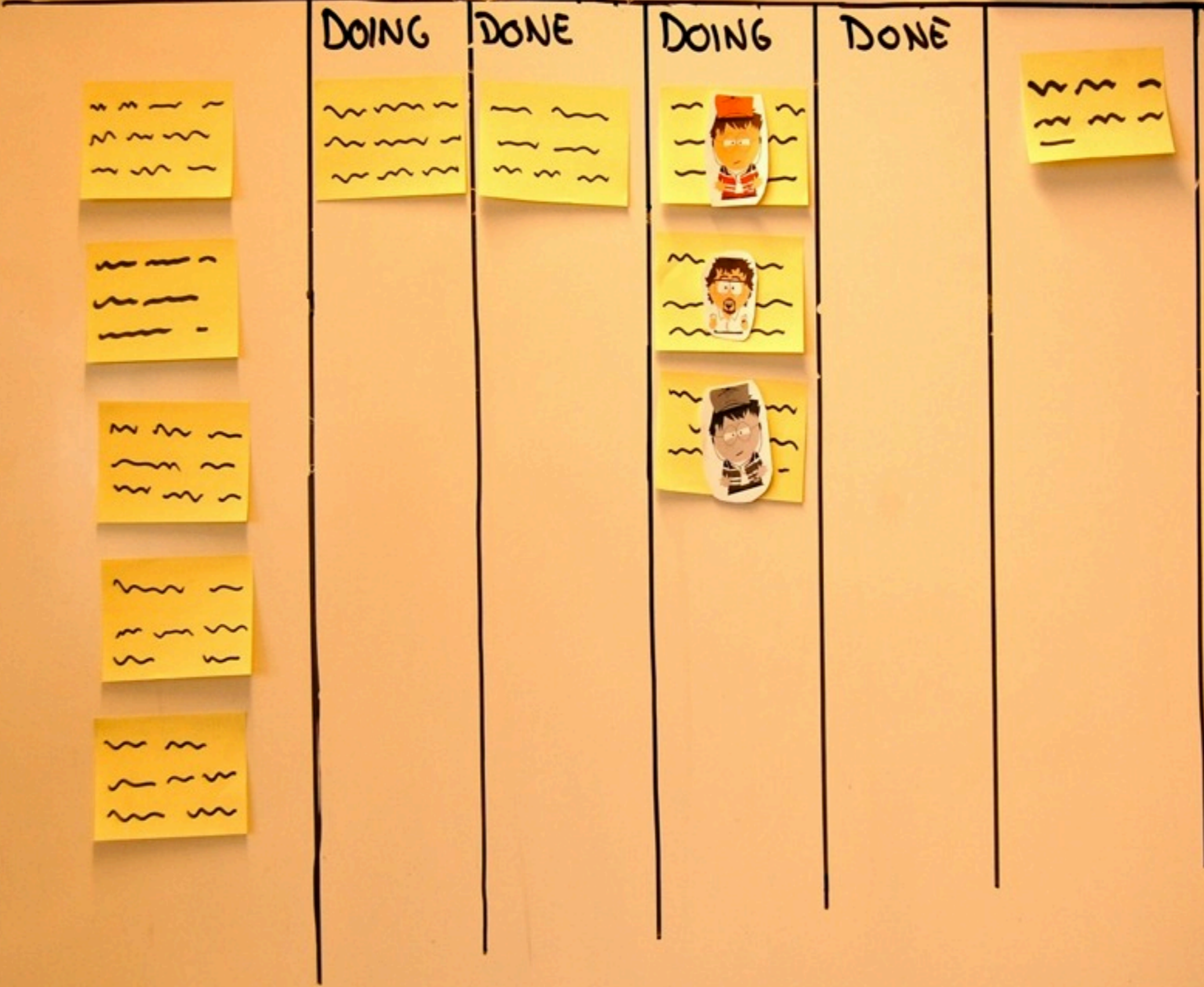
TODO

ANALYZE  
2

DEV  
34

TEST  
2

DONE



måndag 13 februari 12

This could be a time to increase the Work In Process limit.



TODO

ANALYZE  
2

DEV  
34

TEST  
2

DONE



måndag 13 februari 12

And then she can pull new work.

Of course the WIP limit could also be increased or decreased simply because you find that it was too low/high to start with.

- Too high WIP limit -> work idle
- Too low WIP limit -> people idle



TODO

ANALYZE  
2

DEV  
3

TEST  
2

DONE

DOING

DONE

DOING

DONE

**Jocke!  
QUEUE AS  
LEADING  
INDICATOR**

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

måndag 13 februari 12

That wasn't to educating since everything went well. It was good - but not educating...

How do we handle problems? Let's reset the board to a problem situation and find out.

Here all columns are fully loaded. Test is up to their limit and Development is finishing off the last story.

TODO

ANALYZE
2

DEV
3

TEST
2

DONE

DOING

DONE

DOING

DONE

~ ~ ~
~ ~ ~
~ ~ ~

~ ~ ~
~ ~ ~
~ ~ ~

~ ~ ~
~ ~ ~
~ ~ ~

~ ~ ~
~ ~ ~
~ ~ ~

~ ~ ~
~ ~ ~
~ ~ ~

~ ~ ~
~ ~ ~
~ ~ ~

~ ~ ~
~ ~ ~
~ ~ ~

~ ~ ~
~ ~ ~
~ ~ ~

~ ~ ~
~ ~ ~
~ ~ ~

måndag 13 februari 12

Actually, the Development team want to pull some new work. They are sitting idle...



måndag 13 februari 12

But we cannot allow that – that would break their Work in progress limit off 3, now wouldn't it?

So here we have a bottleneck in the Test-step. We want to move things along, but since Test are up to their WIP-limit we cannot.

What's good is that queues like this doesn't happen instantly. You can see it build up and you can react to prevent it. The queue is a "leading indicator". Compare that with burndown in Scrum which is an example of an trailing indicator that gives you the figures on how things went.

What to do? Theory of Constraints gives some options:

- Exploit the bottleneck so that it's used to the maximum, for example remove other work from the testers so that they are only doing the bottleneck activity
- Elevate the bottleneck by increasing the capacity, for example employ more testers or automate tests

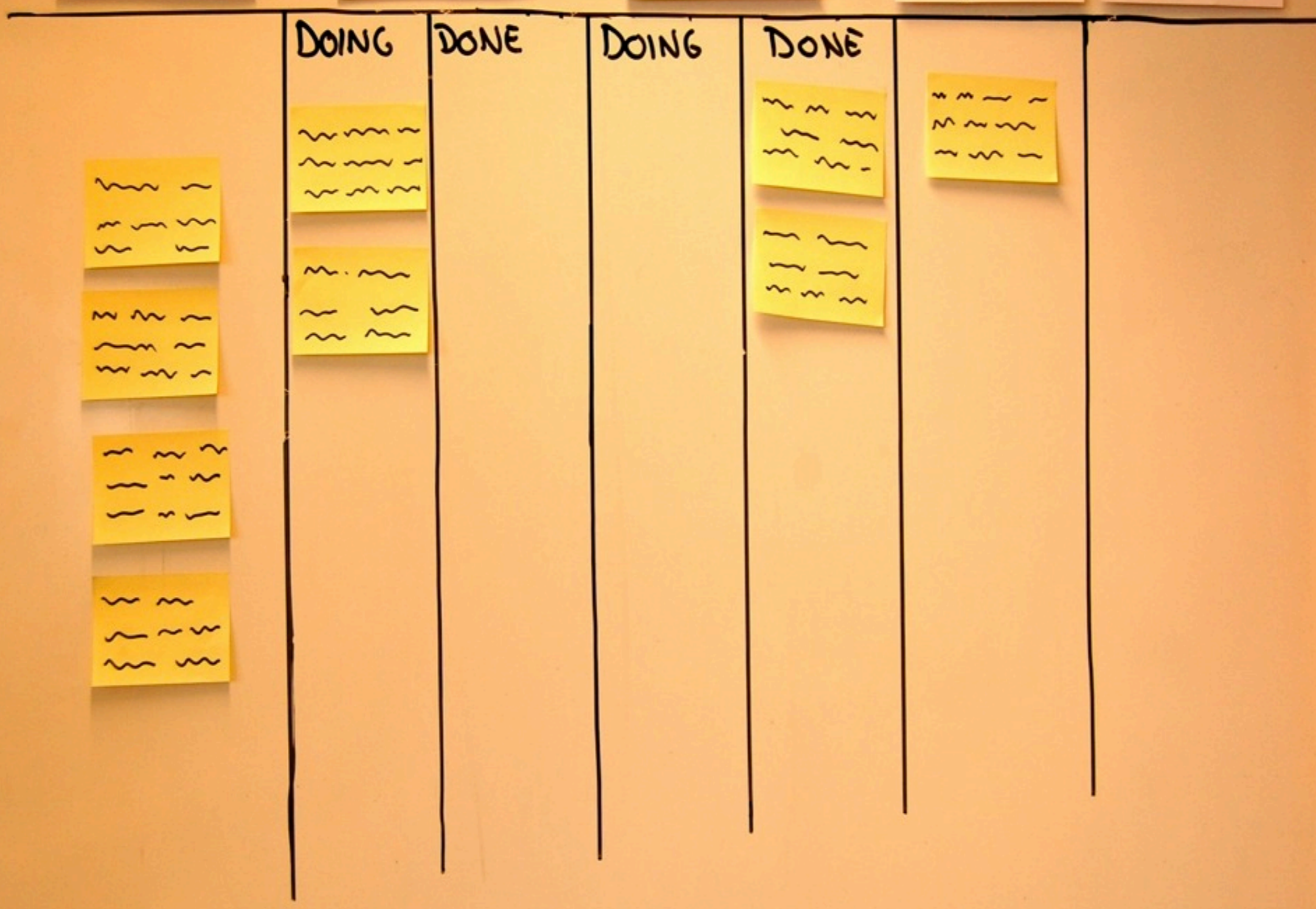
TODO

ANALYZE
2

DEV
3

TEST
2

DONE



måndag 13 februari 12

Another thing to look out for is starvation.

This situation gives us empty columns or starvation.

TODO

ANALYZE
2

DEV
3

TEST
2

DONE



måndag 13 februari 12

The analyze don't have anything done and Development is idle with no more work to pull.

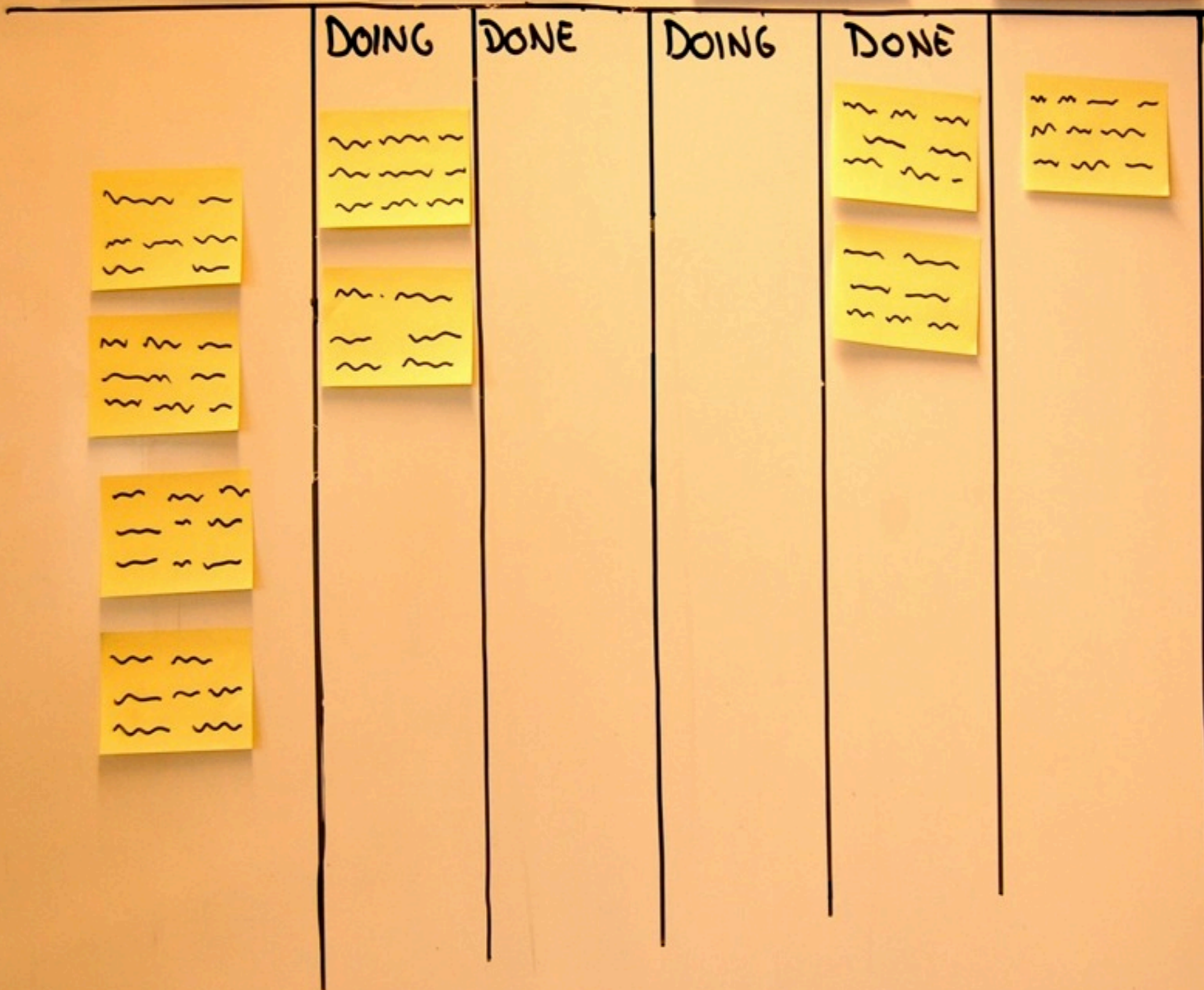
TODO

ANALYZE
2

DEV
3

TEST
2

DONE



måndag 13 februari 12

This of course not good either but is another leading indicator. You can see when it's about to happen and maybe react early, rather than face the figures afterwards.

TODO

ANALYZE
2

DEV
3

TEST
2

DONE

DOING

DONE

DOING

DONE

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

måndag 13 februari 12

We can see the bottleneck building up earlier. This is the state of the board right before the bottleneck just discussed.



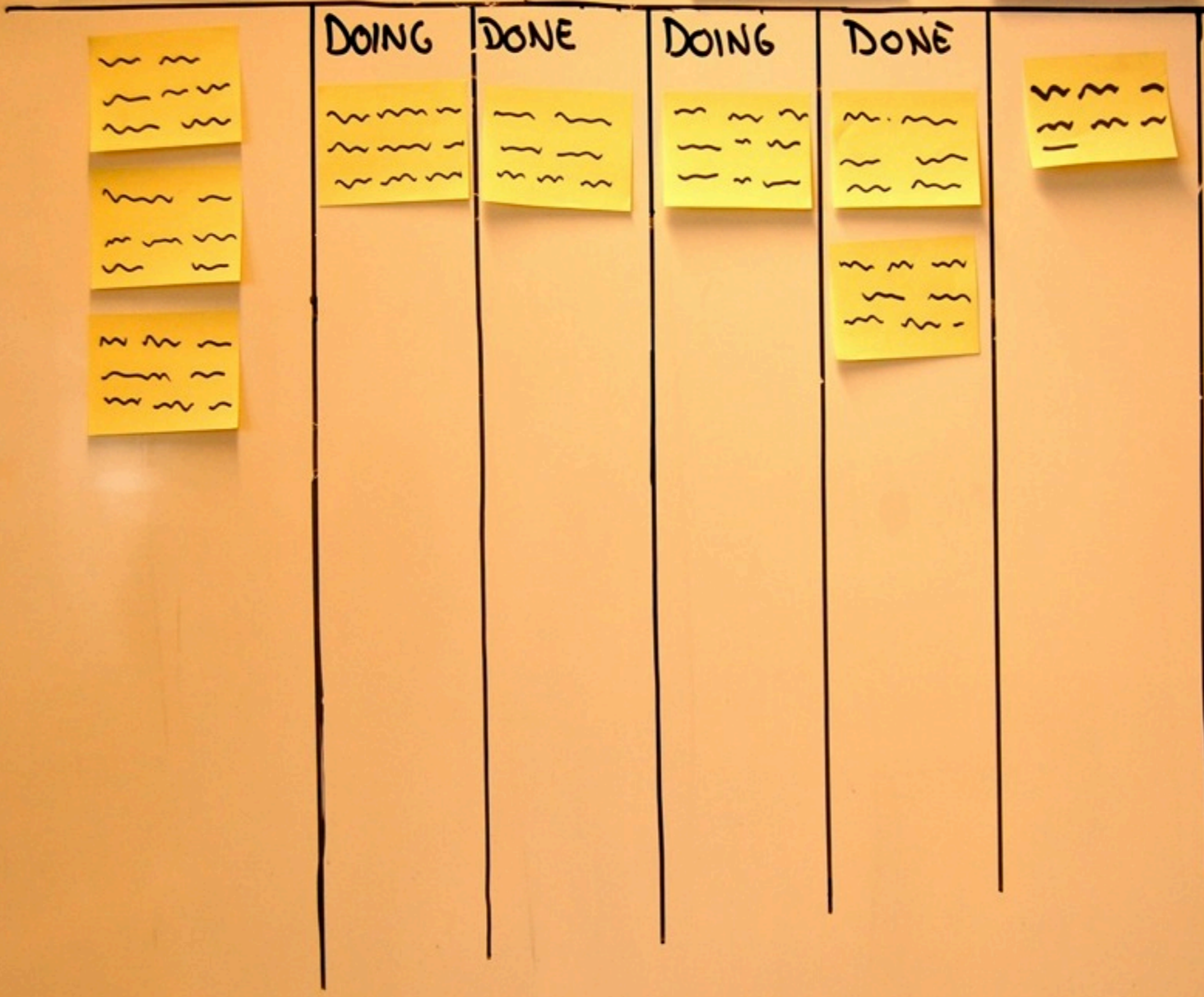
TODO

ANALYZE  
2

DEV  
3

TEST  
2

ACCEPT



måndag 13 februari 12

Another kind of bottleneck is what we call a Non-instant availability resource.

Say for example that you want your product owner to accept all items before shipping them.

We add a new column for that - Accept.



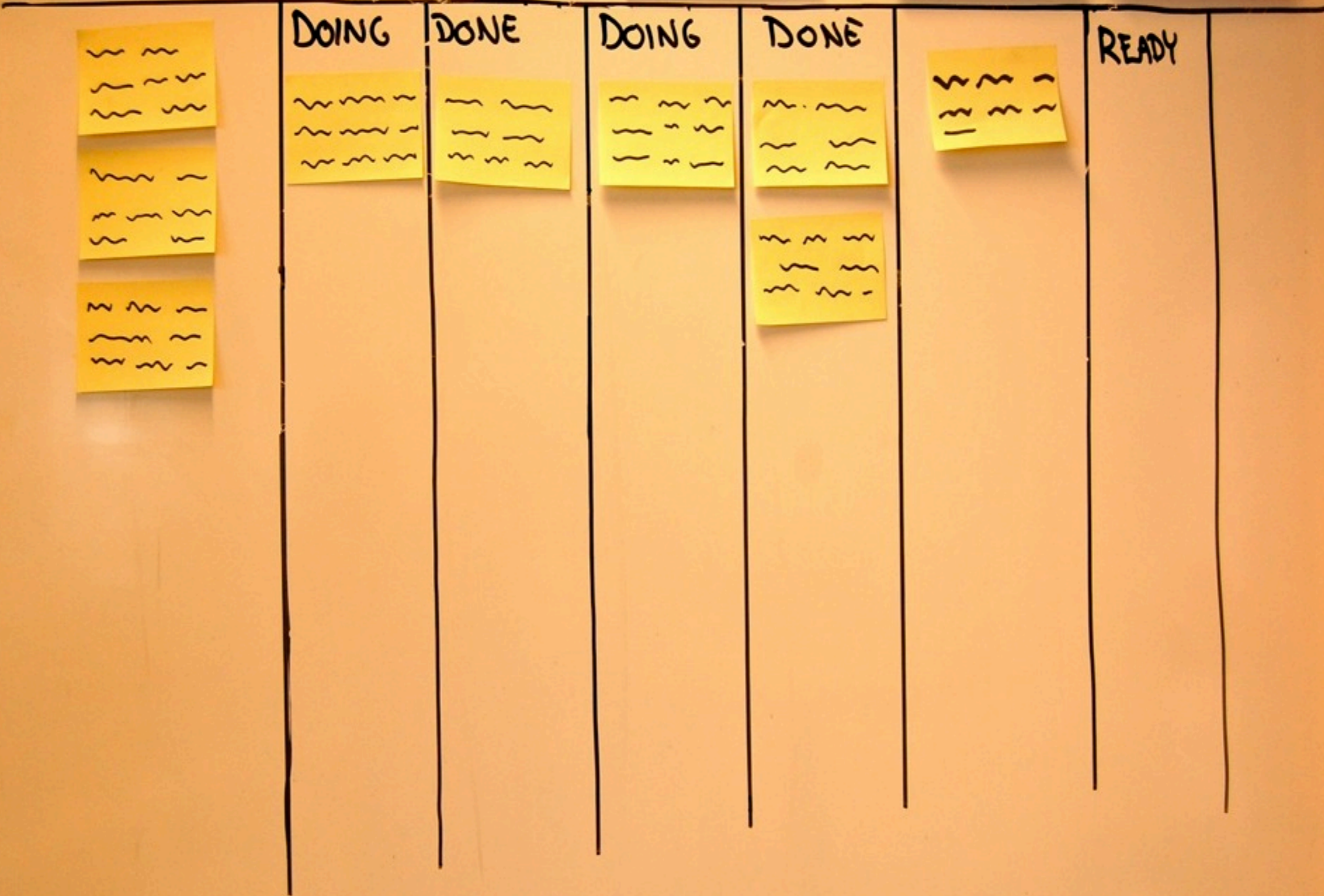
TODO

ANALYZE  
2

DEV  
3

TEST  
2

ACCEPT



måndag 13 februari 12

And since the product owner is what is called a Non instant availability constraint we add a queue for things that is ready to accept.



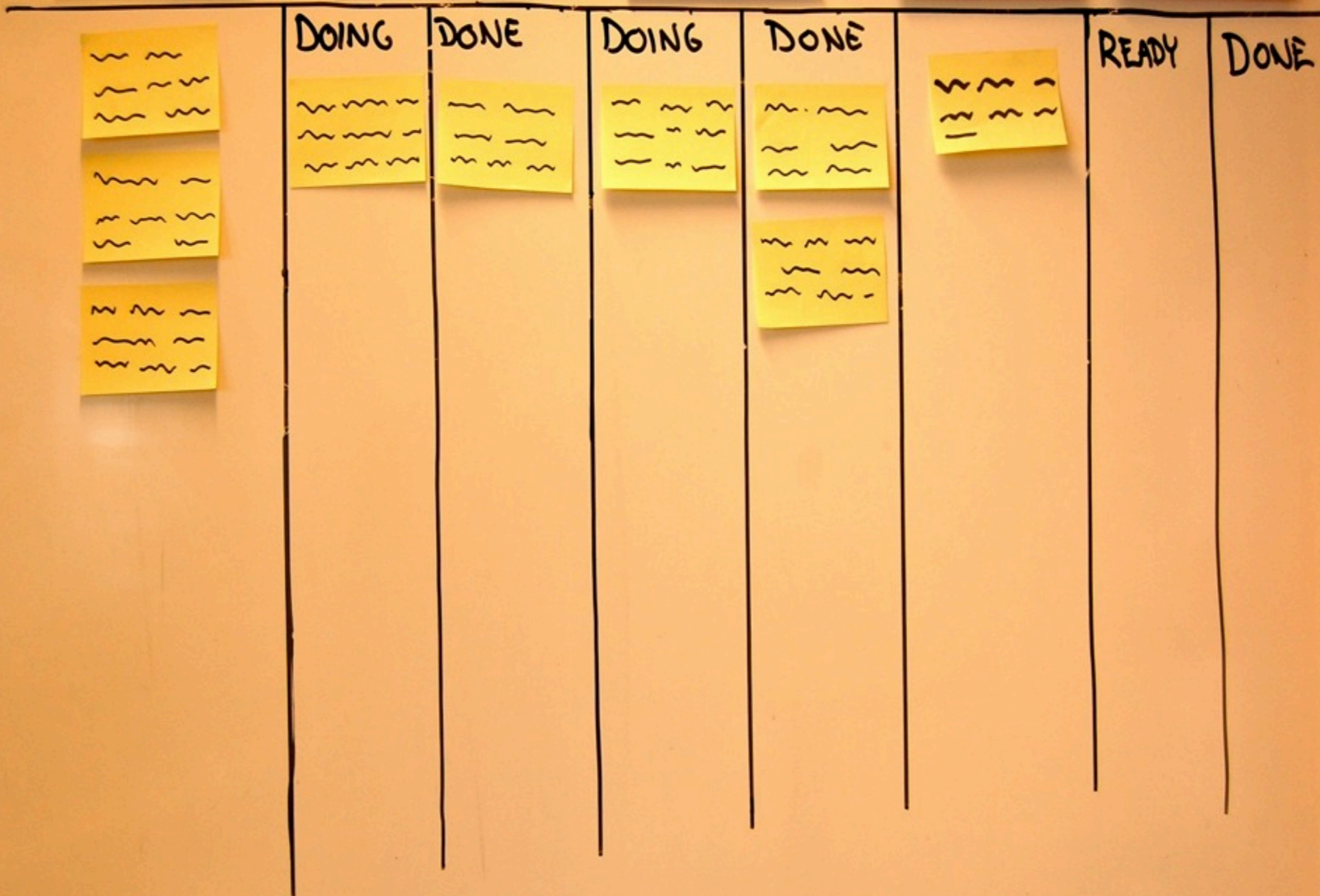
TODO

ANALYZE  
2

DEV  
3

TEST  
2

ACCEPT



måndag 13 februari 12

And a column for accepted items.



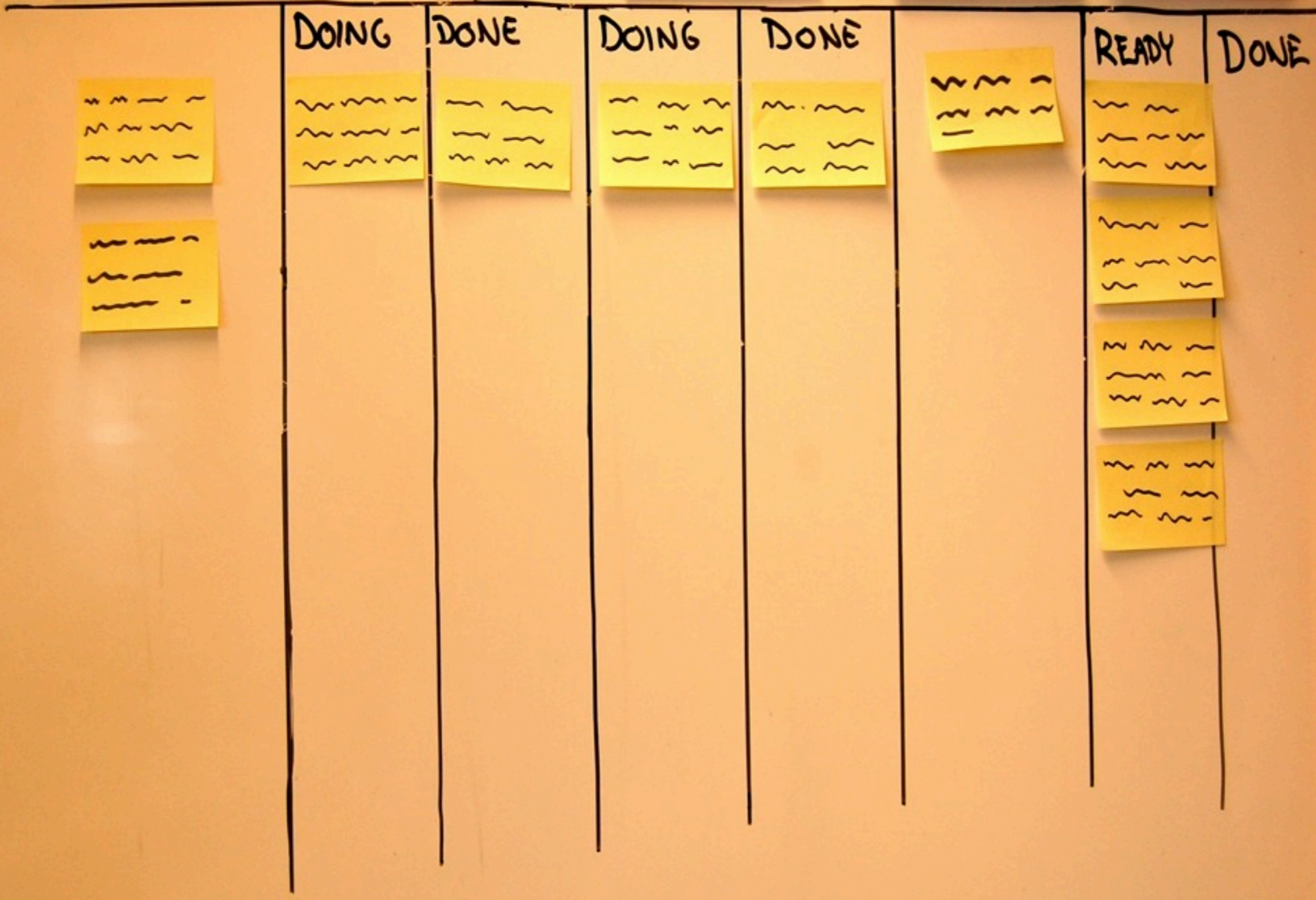
TODO

ANALYZE  
2

DEV  
3

TEST  
2

ACCEPT



måndag 13 februari 12

The product owner cannot be with us all the time, but when she's here she can accept many items at once. It doesn't take too long.

We allow up to 4 items in the Ready for Accept queue.



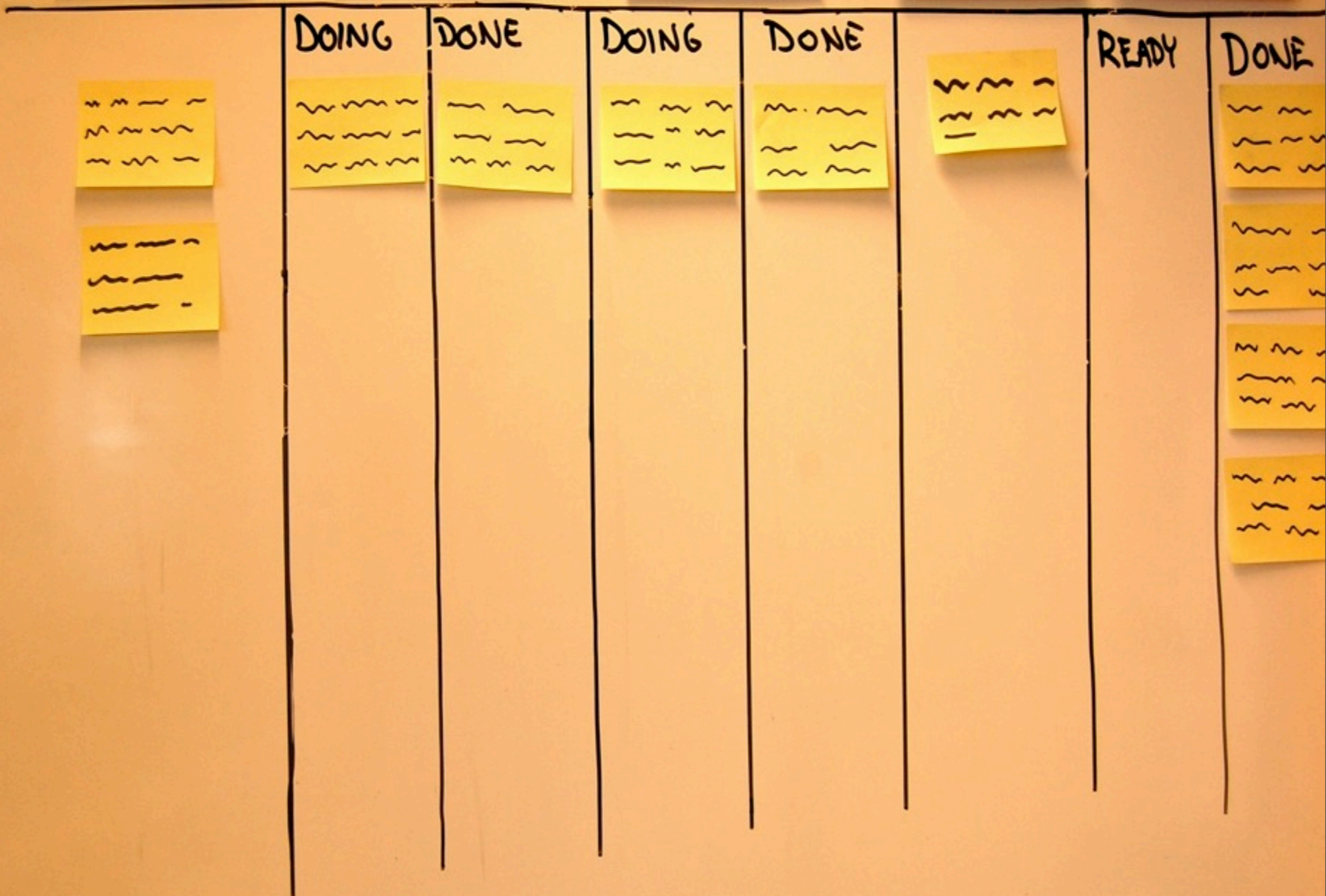
TODO

ANALYZE  
2

DEV  
3

TEST  
2

ACCEPT



måndag 13 februari 12

and as you see all items are quickly accepted when she had the time to sit down with and do it.



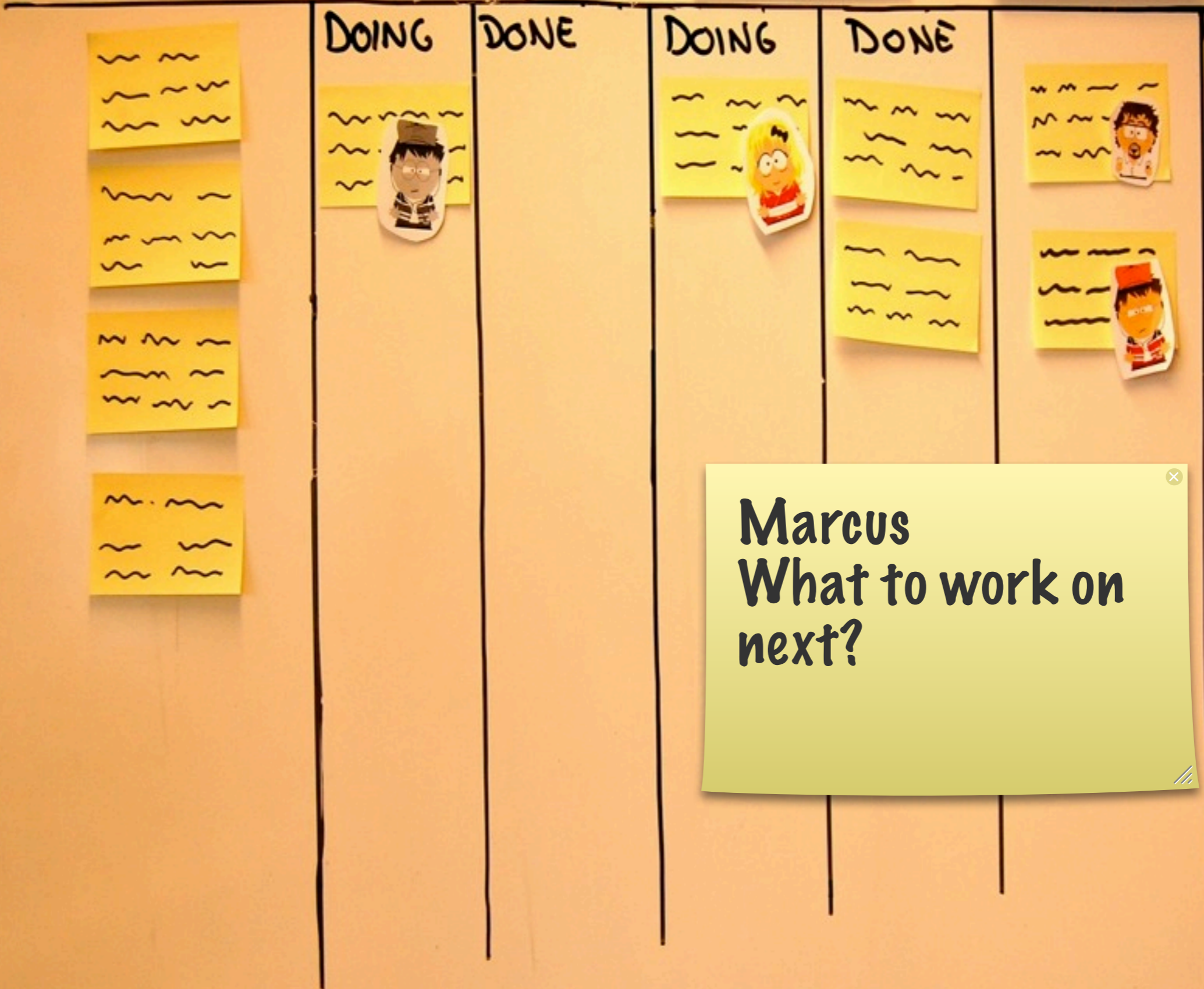
TODO

ANALYZE  
2

DEV  
3

TEST  
2

DONE



måndag 13 februari 12

That was one example of the flow of work on a Kanban board.

But what should I be working on next, when I've finished something? It's always easier to start new work than it is to finish something. What are the strategies and guidance in Kanban?

Here is Joakim – the tester – ready to complete the feature he is testing right now.



TODO

ANALYZE  
2

DEV  
3

TEST  
2

DONE

~ ~ ~  
~ ~ ~  
~ ~ ~  
~ ~ ~  
~ ~ ~  
~ ~ ~  
~ ~ ~  
~ ~ ~

DOING

DONE

DOING

DONE



måndag 13 februari 12

Yes! It's done!



TODO

ANALYZE  
2

DEV  
3

TEST  
2

DONE

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

DOING

DONE

DOING

DONE

~~~~~  
~~~~~  
~~~~~



~~~~~  
~~~~~  
~~~~~




?

~~~~~  
~~~~~  
~~~~~



~~~~~  
~~~~~  
~~~~~

måndag 13 februari 12

What should he do now?



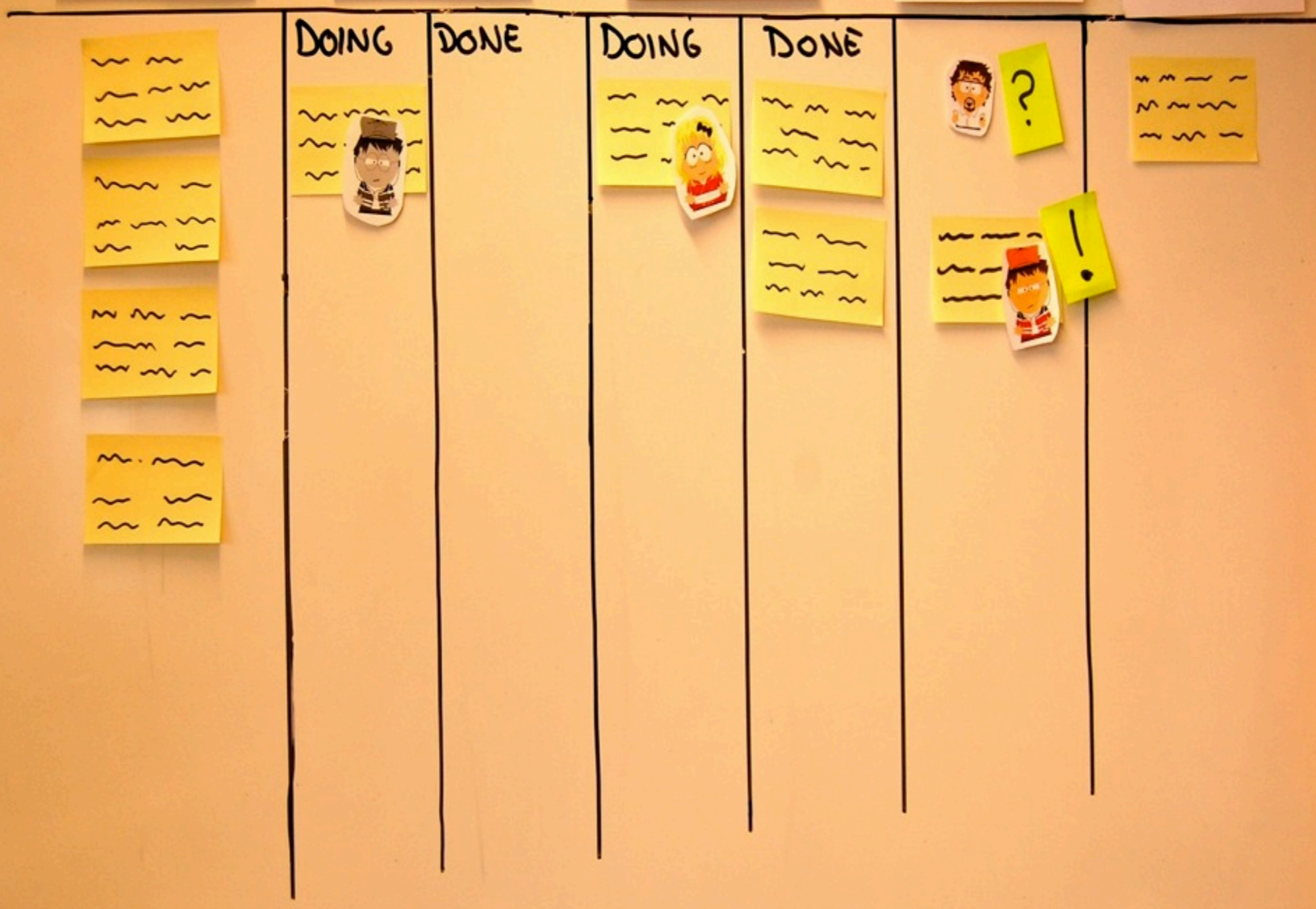
TODO

ANALYZE  
2

DEV  
3

TEST  
2

DONE



måndag 13 februari 12

Since there is work in process in his column – the testing Marcus is doing – he should first and foremost see if he can help finish the work in process.

That's our main goal – to finish work in process.



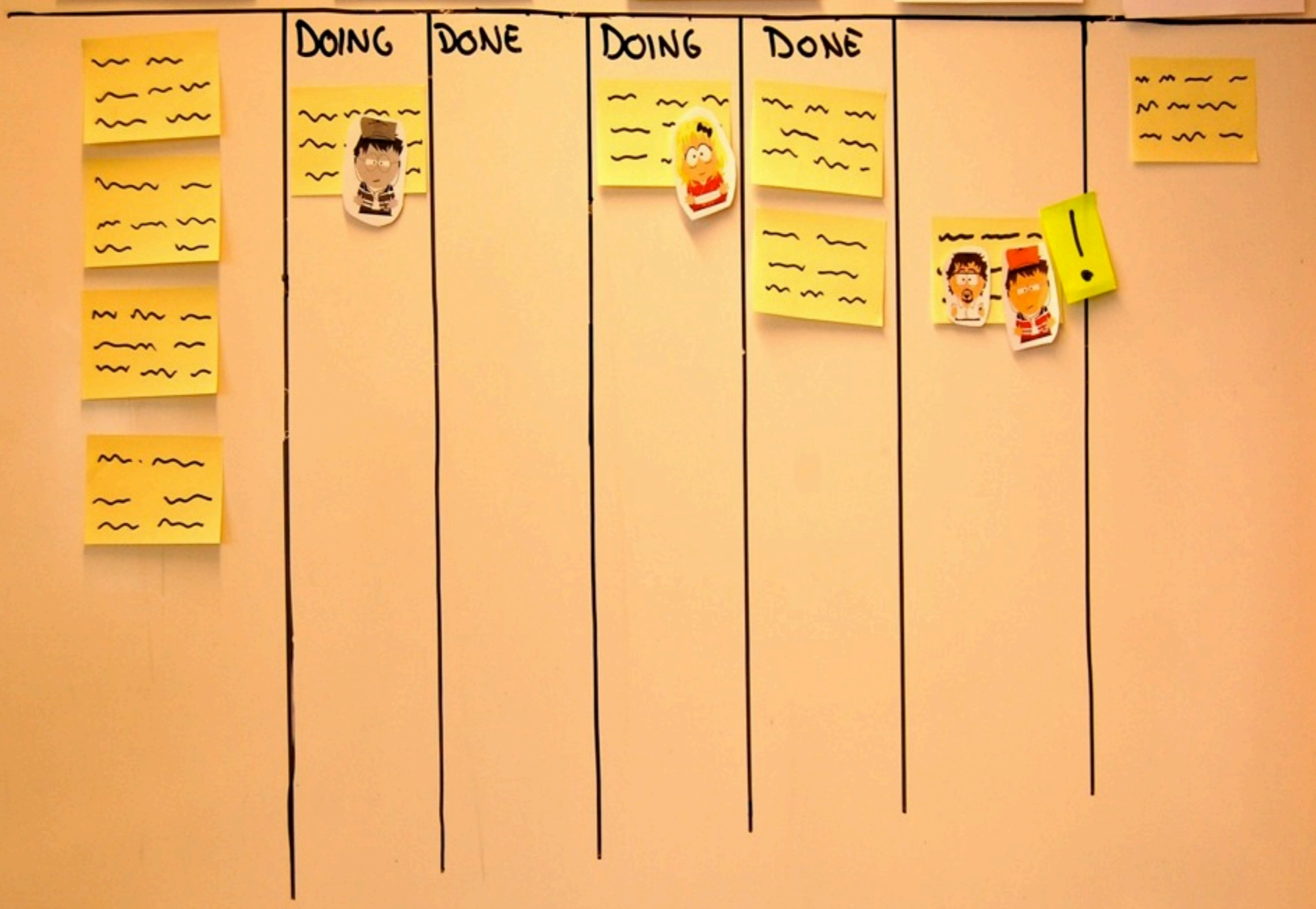
TODO

ANALYZE  
2

DEV  
3

TEST  
2

DONE



måndag 13 februari 12

So Joakim and Marcus will test that feature together



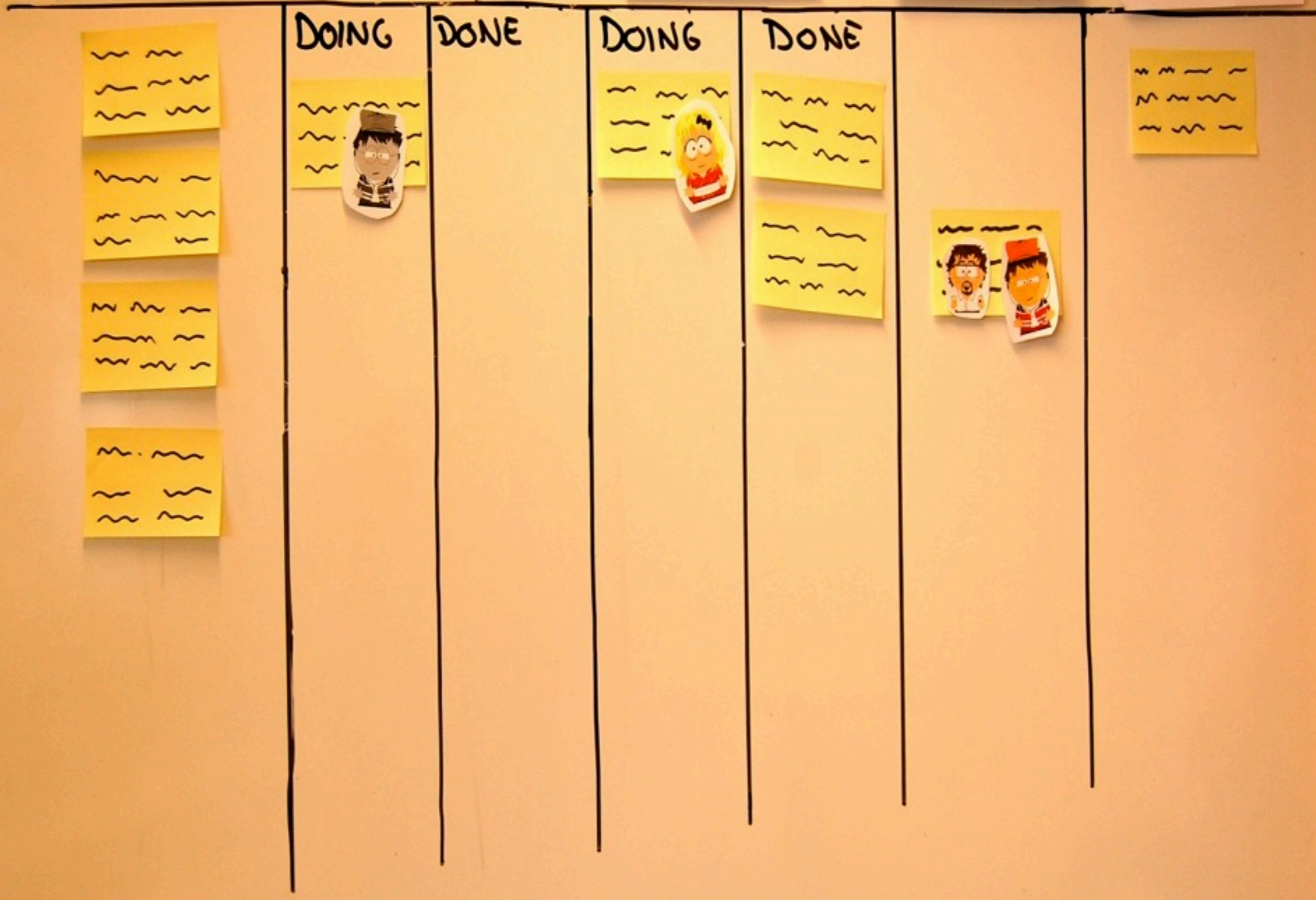
TODO

ANALYZE  
2

DEV  
3

TEST  
2

DONE



måndag 13 februari 12

They start to test it...



TODO

ANALYZE  
2

DEV  
3

TEST  
2

DONE

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

DOING

DONE

DOING

DONE

~~~~~  
~~~~~  
~~~~~



~~~~~  
~~~~~  
~~~~~



~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~



~~~~~  
~~~~~  
~~~~~



~~~~~  
~~~~~  
~~~~~

måndag 13 februari 12  
...and before long...



TODO

ANALYZE  
2

DEV  
3

TEST  
2

DONE

~~~~~  
~~~~~  
~~~~~  
~~~~~  
~~~~~  
~~~~~  
~~~~~  
~~~~~  
~~~~~  
~~~~~

DOING

DONE

DOING

DONE



~~~~~  
~~~~~  
~~~~~



måndag 13 februari 12

... that feature is also Done!

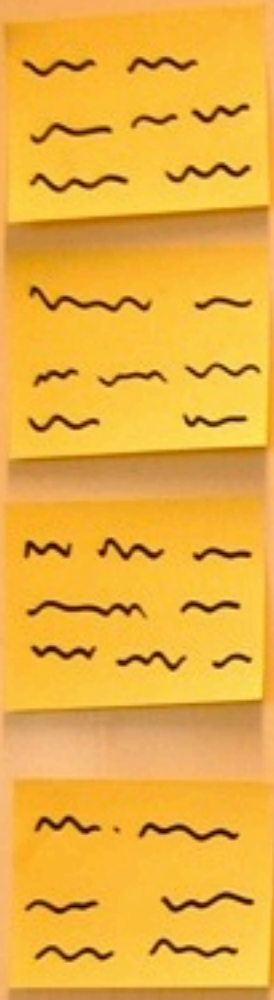




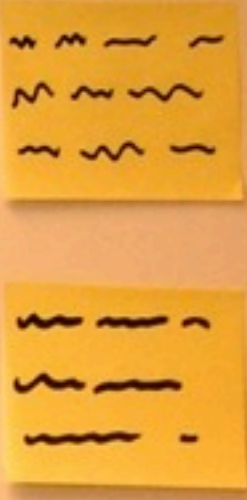
TODO

ANALYZE
2

DEV
3

TEST
2

DONE

	DOING	DONE	DOING	DONE		DONE
						

måndag 13 februari 12

What now? Both Joakim and Marcus is ready for some more work. Nothing to finish in their column.

TODO

ANALYZE
2

DEV
3

TEST
2

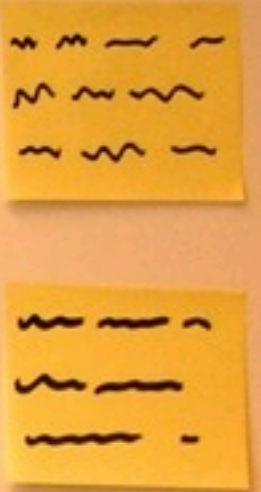
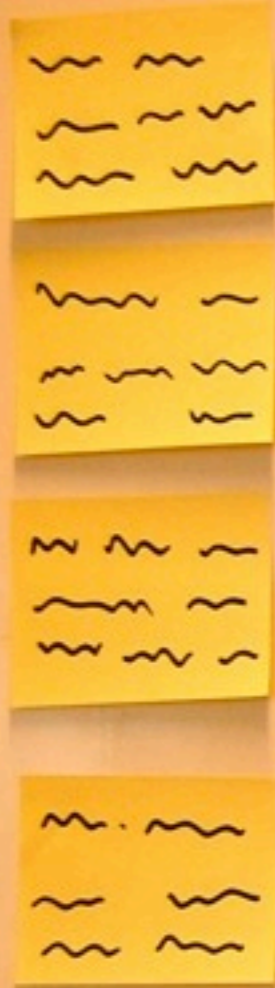
DONE

DOING

DONE

DOING

DONE



måndag 13 februari 12

But there is new work to be pulled from the Done-queue of the Development step.

TODO

ANALYZE
2

DEV
3

TEST
2

DONE

~~~~~  
~~~~~  
~~~~~  
~~~~~  
~~~~~  
~~~~~  
~~~~~  
~~~~~  
~~~~~  
~~~~~

DOING

DONE

DOING

DONE

~~~~~  
~~~~~  
~~~~~  
~~~~~  
~~~~~



~~~~~  
~~~~~  
~~~~~  
~~~~~  
~~~~~



~~~~~  
~~~~~  
~~~~~  
~~~~~  
~~~~~



~~~~~  
~~~~~  
~~~~~  
~~~~~  
~~~~~



~~~~~  
~~~~~  
~~~~~  
~~~~~  
~~~~~  
~~~~~  
~~~~~  
~~~~~

måndag 13 februari 12

Great! They pull new work...

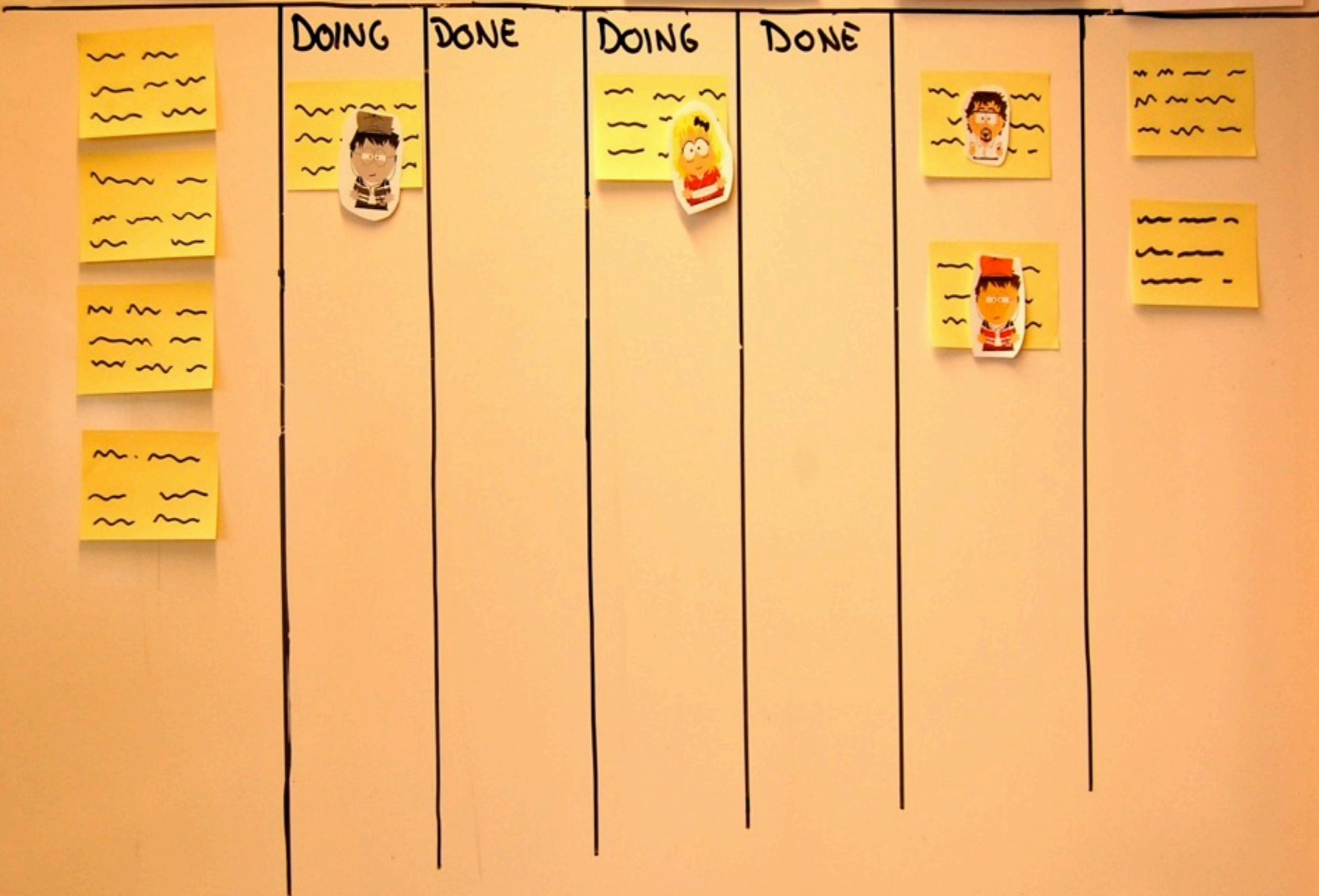
TODO

ANALYZE
2

DEV
3

TEST
2

DONE



måndag 13 februari 12

Each starting a feature – to get work to done as quickly as possible.

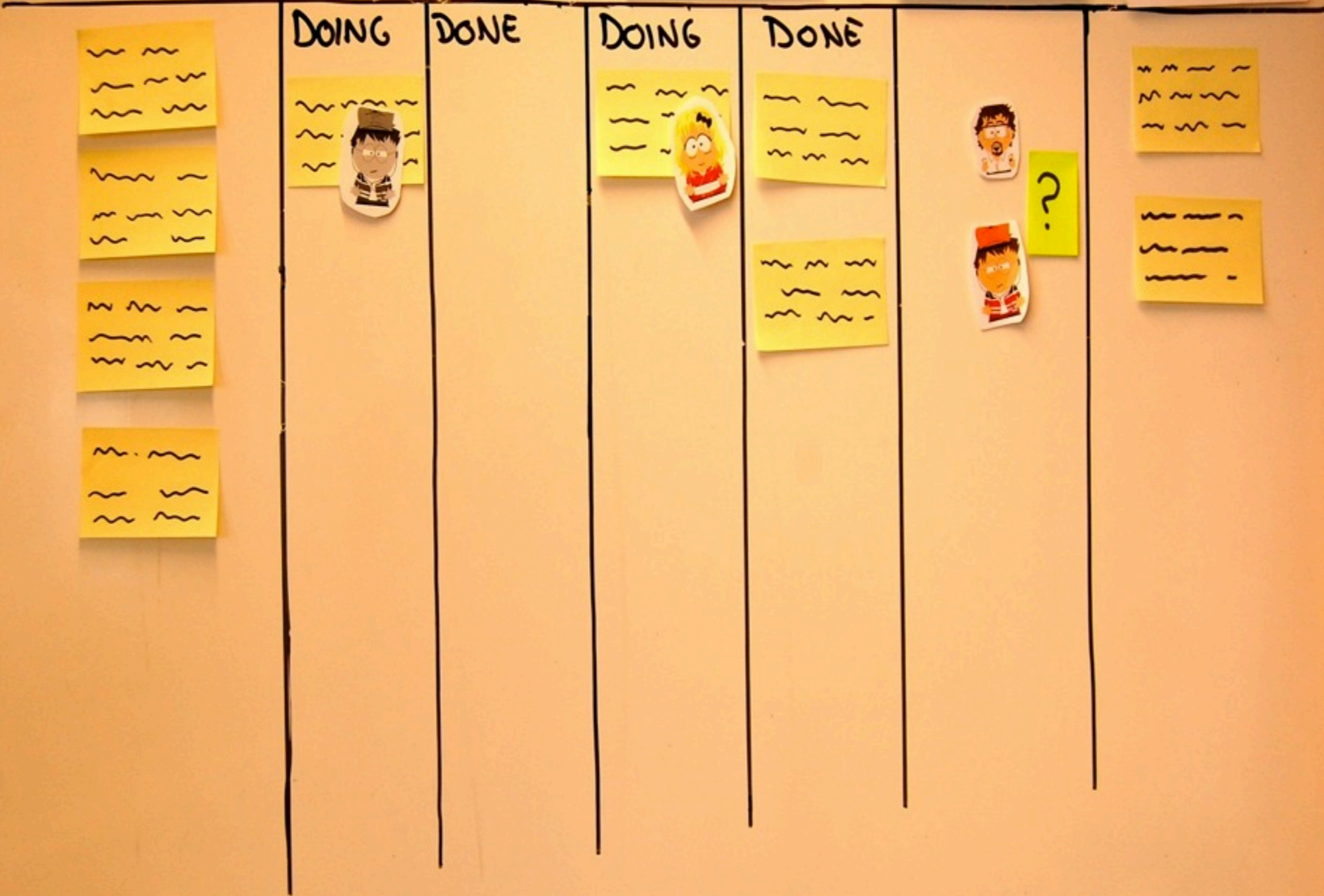
TODO

ANALYZE
2

DEV
3

TEST
2

DONE



måndag 13 februari 12

Another way to handling the situation...

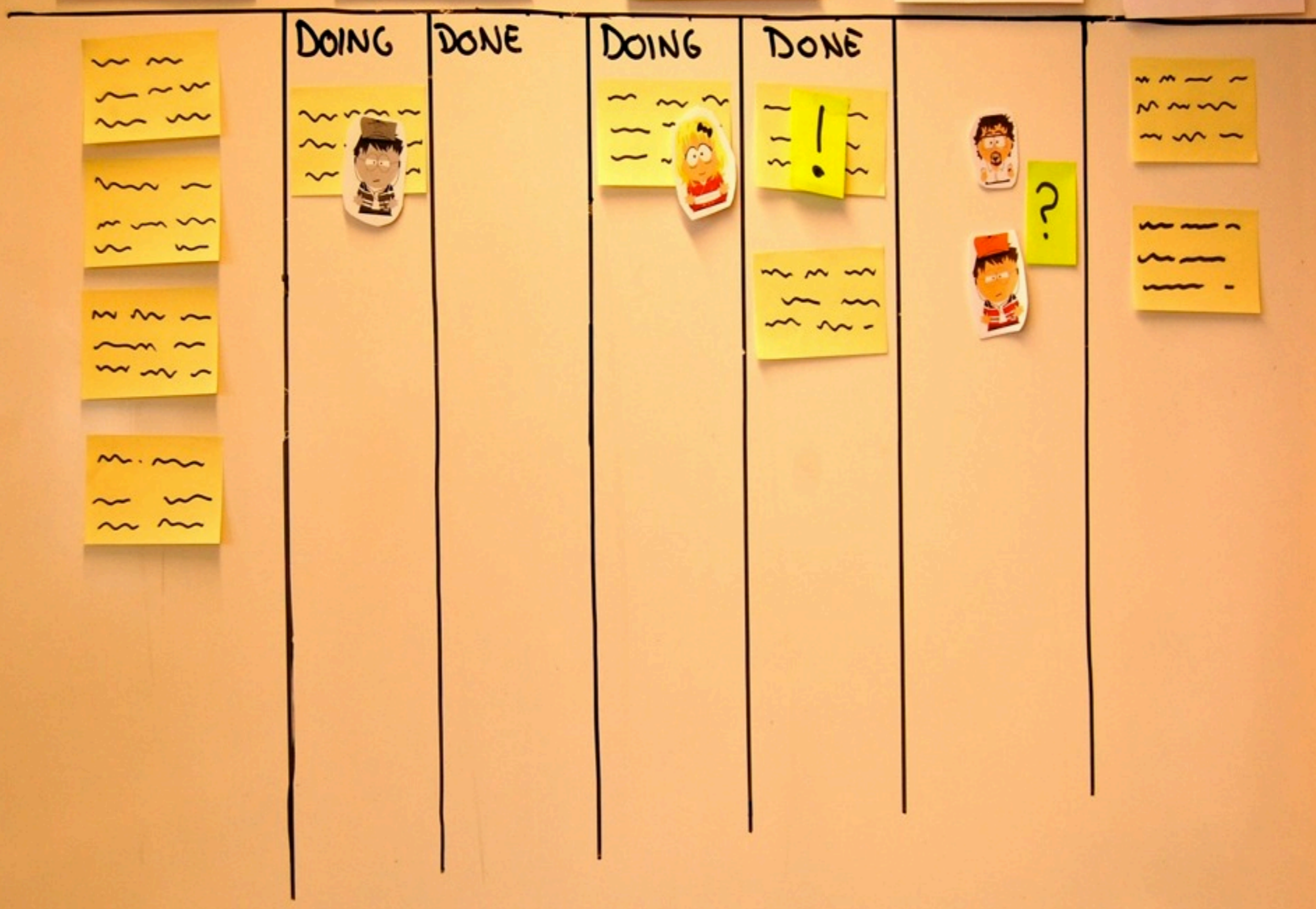
TODO

ANALYZE
2

DEV
3

TEST
2

DONE



måndag 13 februari 12

would be to work together...

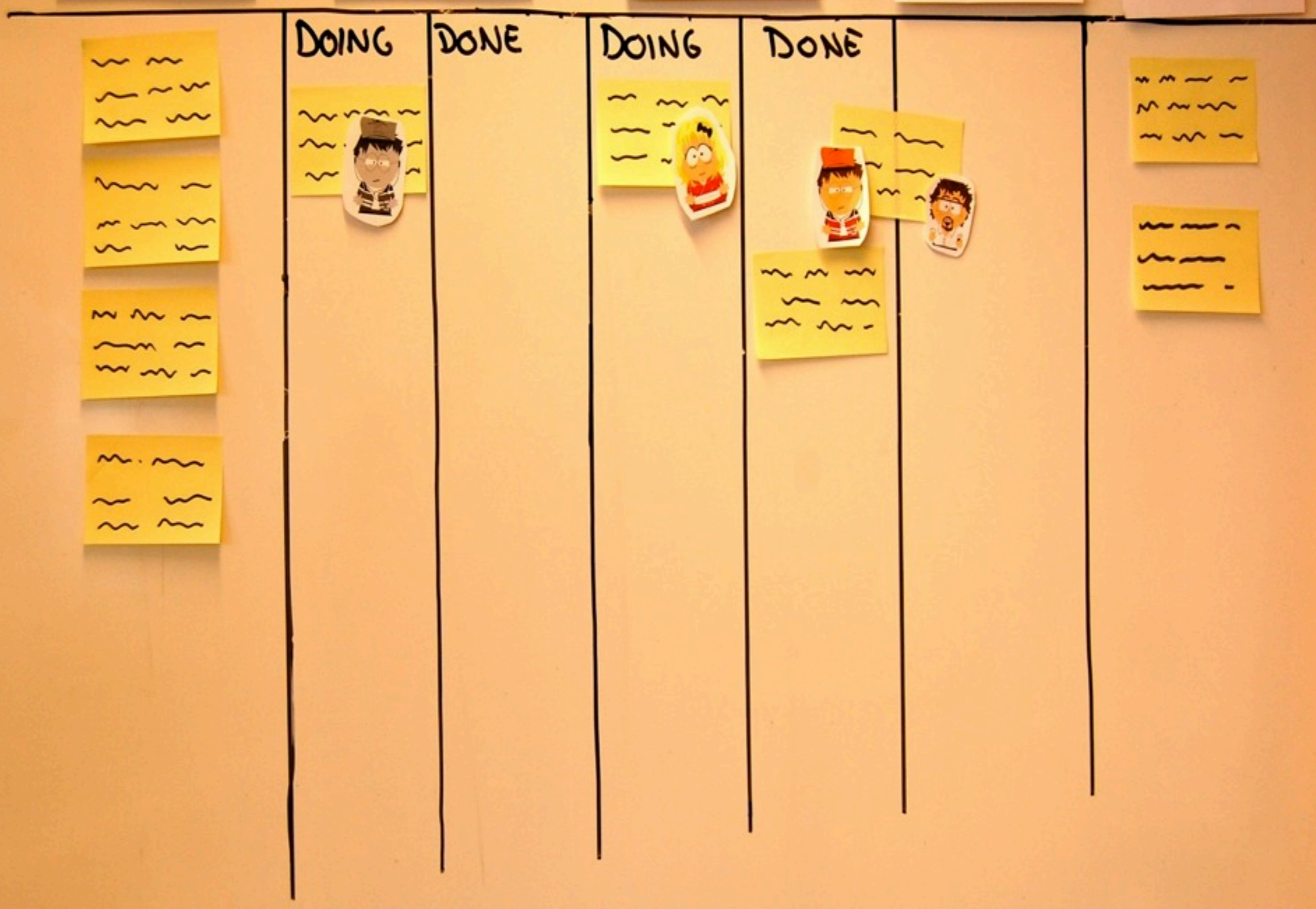
TODO

ANALYZE
2

DEV
3

TEST
2

DONE



måndag 13 februari 12

... on the highest prioritized item...

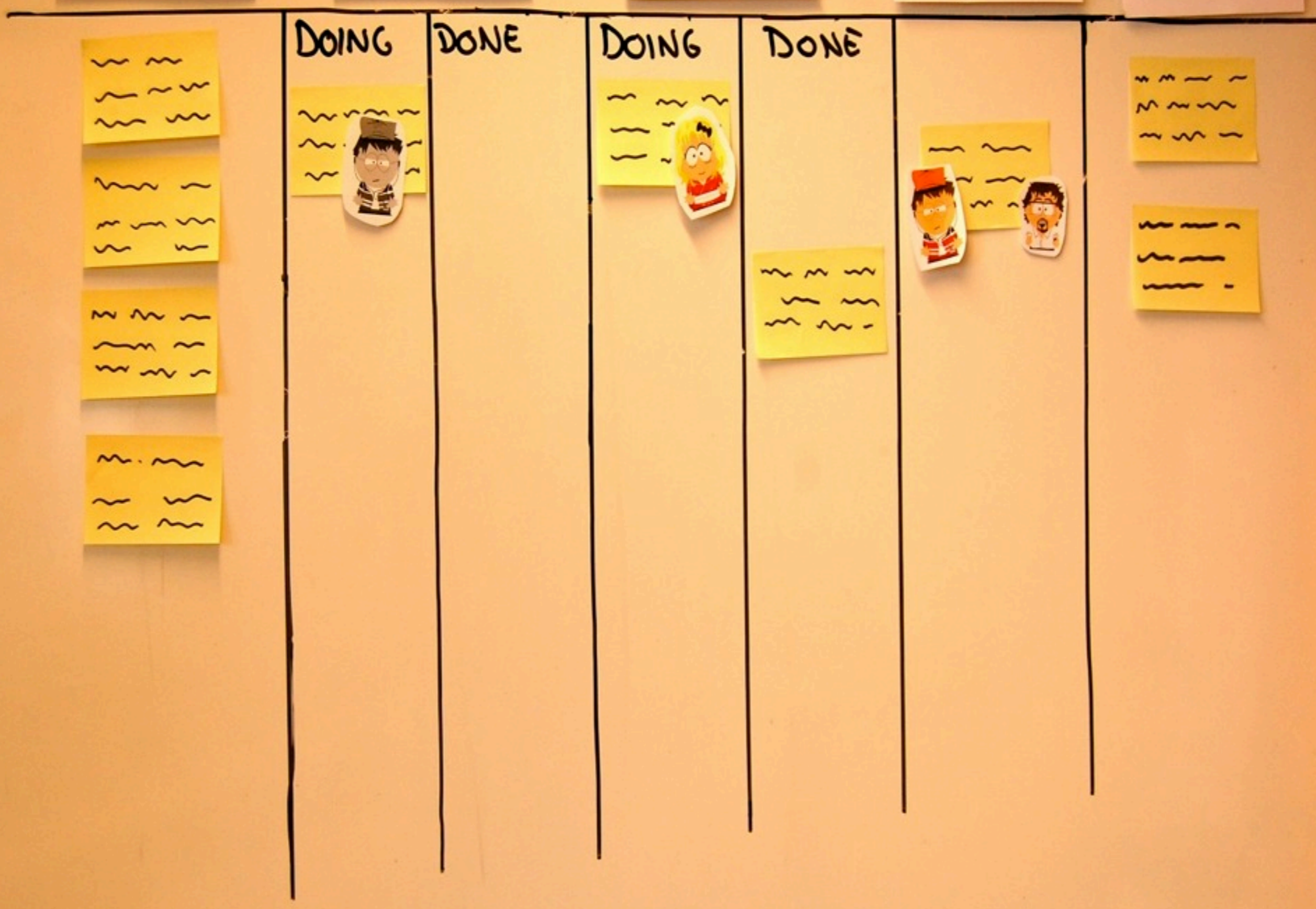
TODO

ANALYZE
2

DEV
3

TEST
2

DONE



måndag 13 februari 12

... to quickly get that item done.

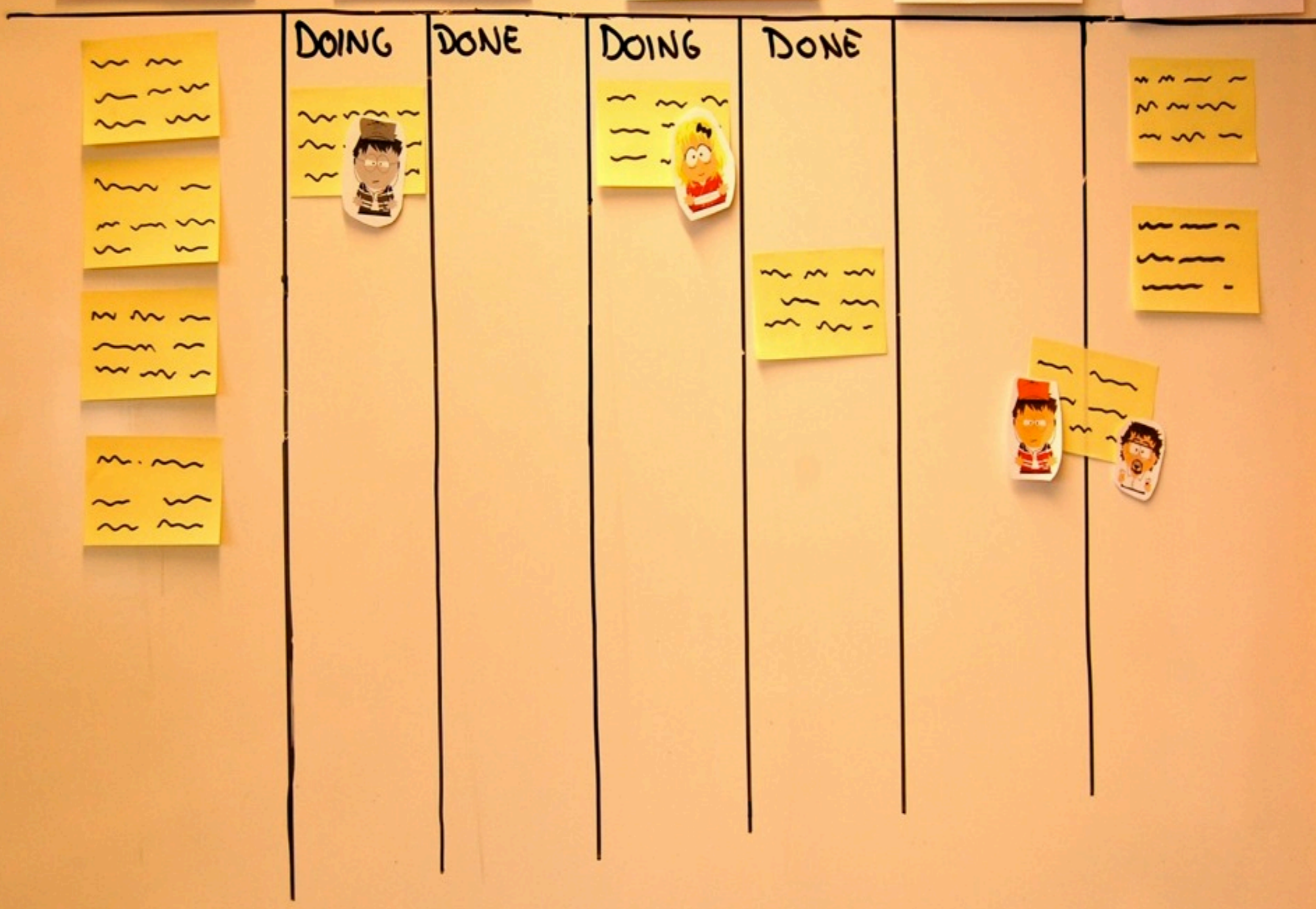
TODO

ANALYZE
2

DEV
3

TEST
2

DONE



måndag 13 februari 12

Look - they are on their way to being done

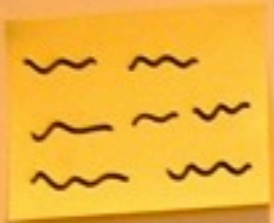
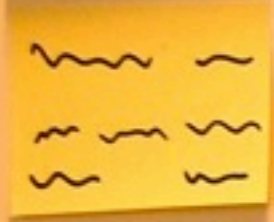
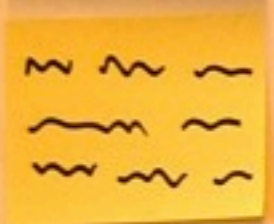
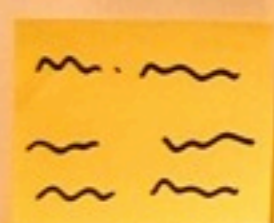



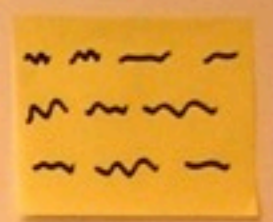
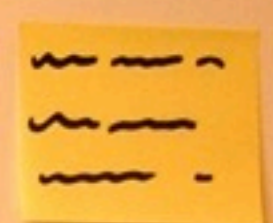

TODO

ANALYZE
2

DEV
3

TEST
2

DONE

	DOING	DONE	DOING	DONE		
   						  

måndag 13 februari 12

Well - that was quick! Great work guys!

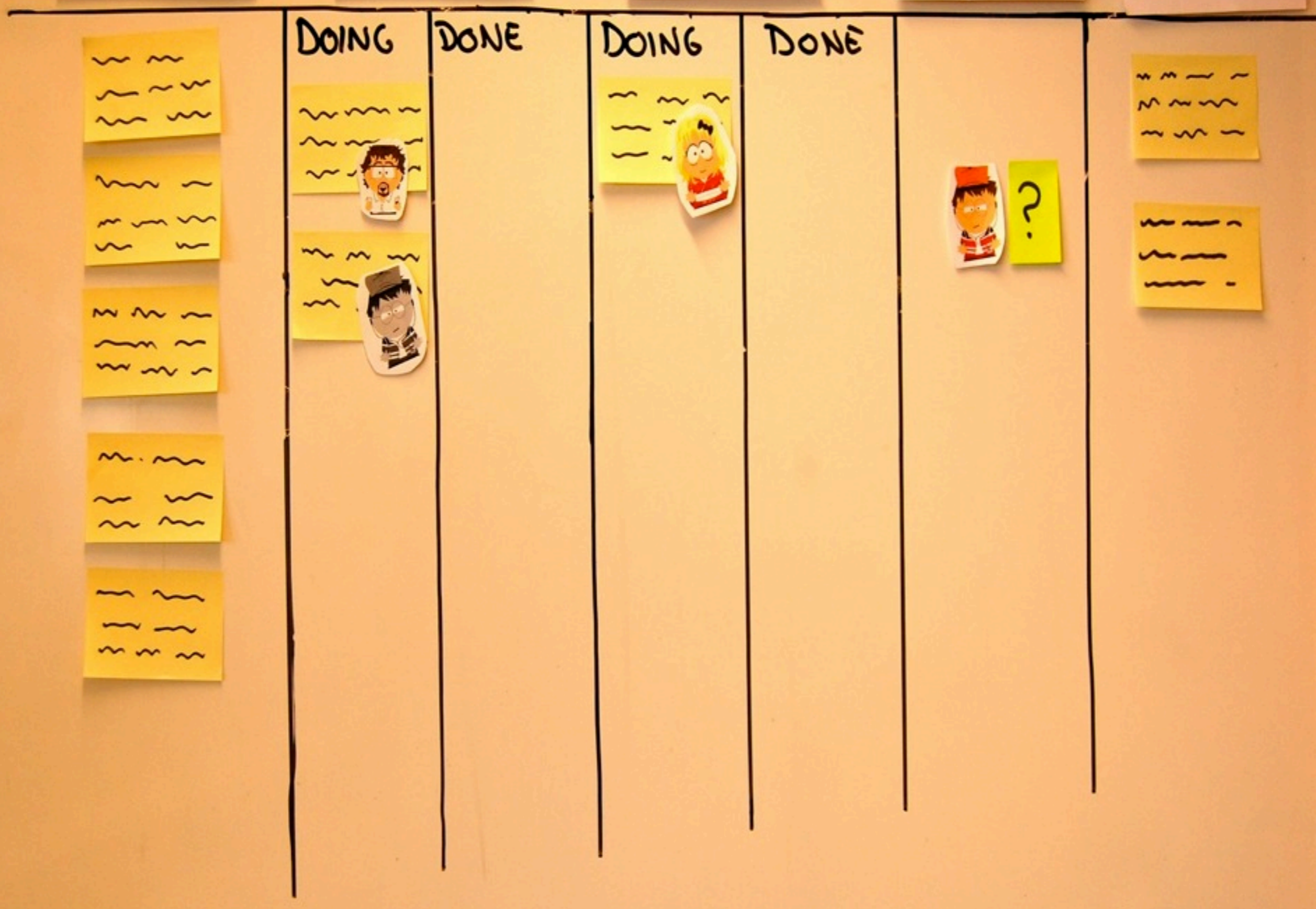
TODO

ANALYZE
2

DEV
3

TEST
2

DONE



måndag 13 februari 12

Here is a case where Marcus, the tester is left with no work in his column AND no new work to pull.

What should he do now?

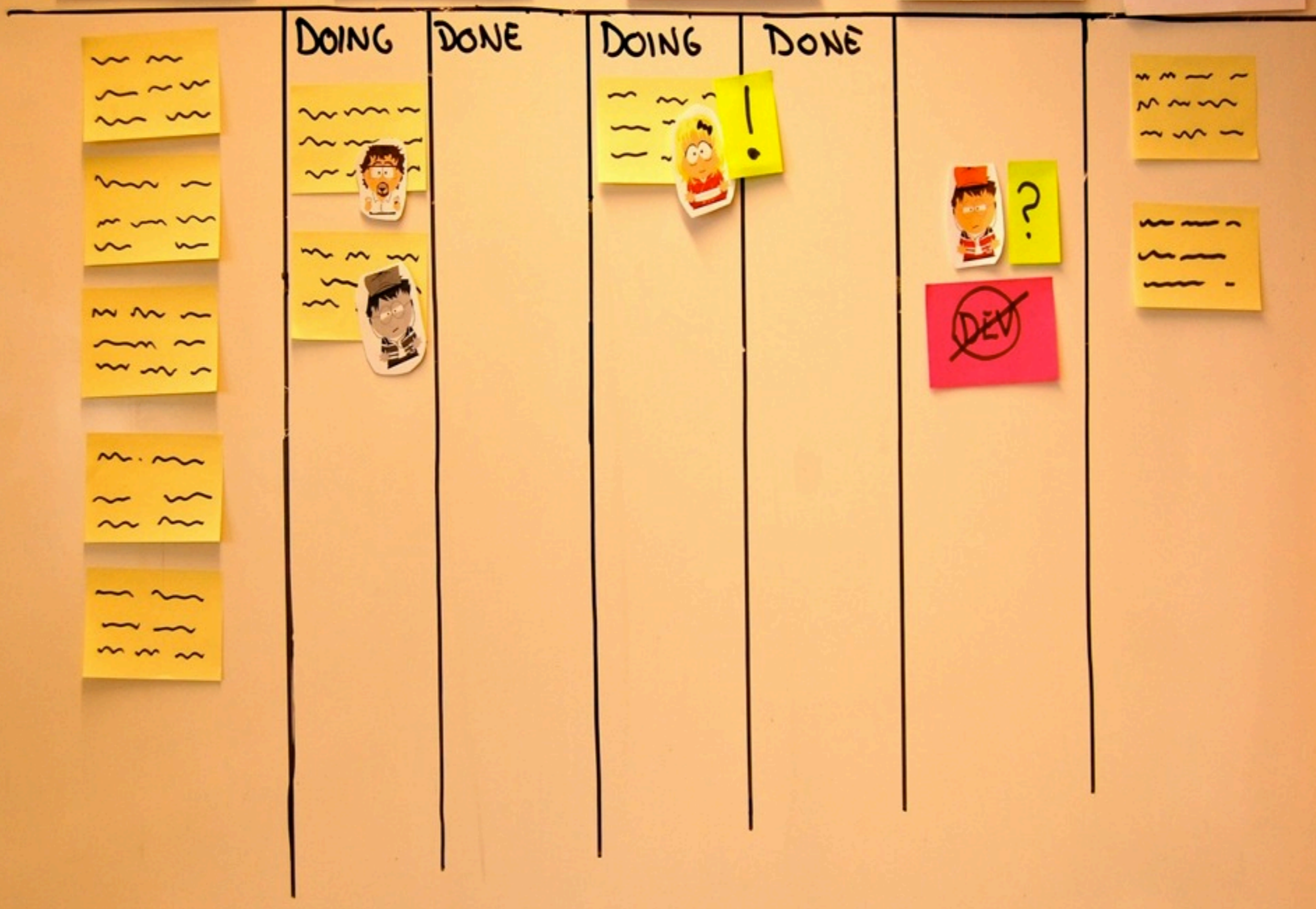
TODO

ANALYZE
2

DEV
3

TEST
2

DONE



måndag 13 februari 12

But that is sadly not possible since Marcus has no programming skills.

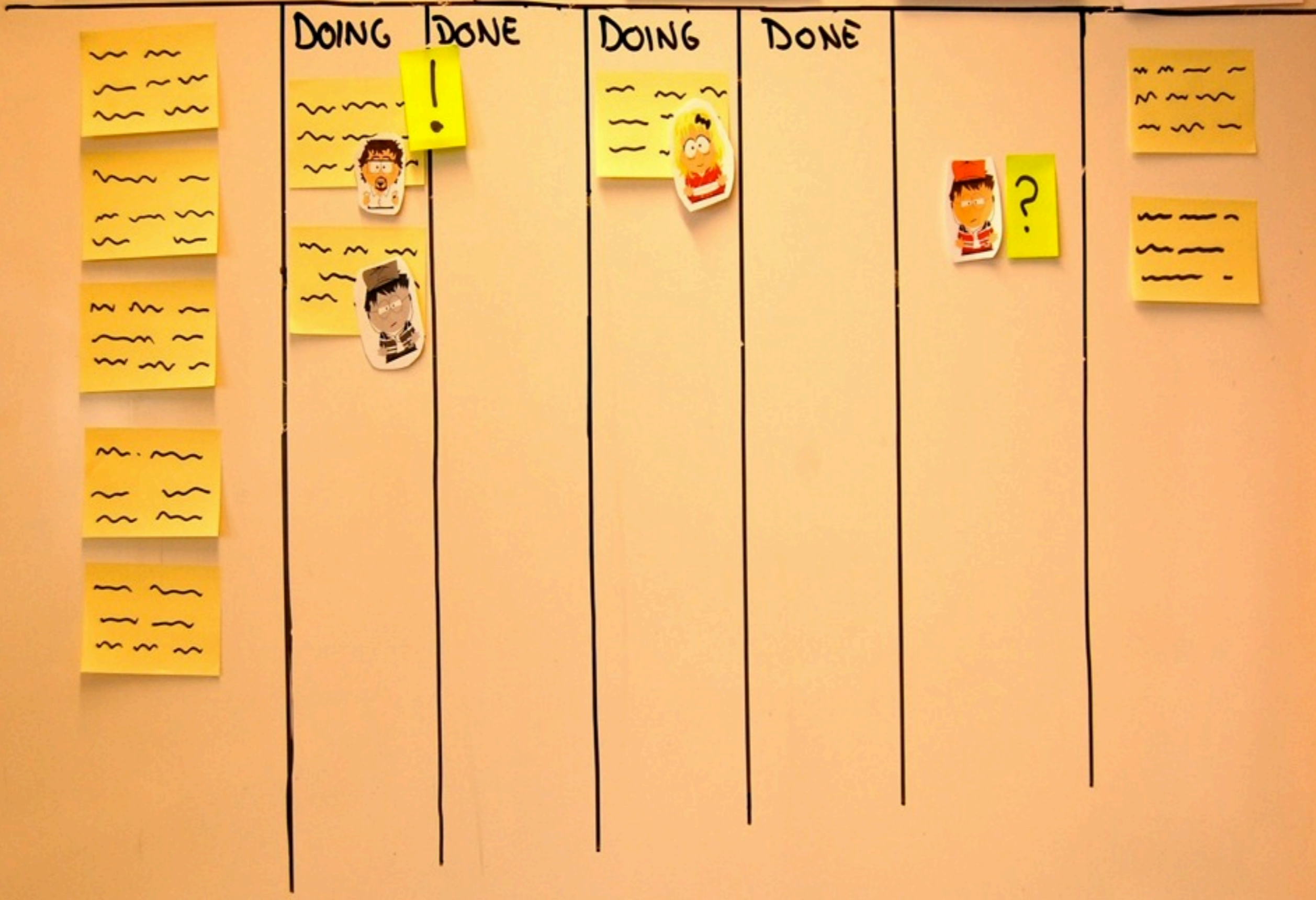
TODO

ANALYZE
2

DEV
3

TEST
2

DONE



måndag 13 februari 12

But... we have a bottleneck in Analyze. They are fully loaded and no new work is flowing from them.

Both Joakim and Black-And-White-Marcus is working. Could Full-Color-Marcus possibly help over there?

TODO

ANALYZE
2

DEV
3

TEST
2

DONE

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

DOING

DONE

DOING

DONE



~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

måndag 13 februari 12

Yes, he can help Joakim to finish his work.



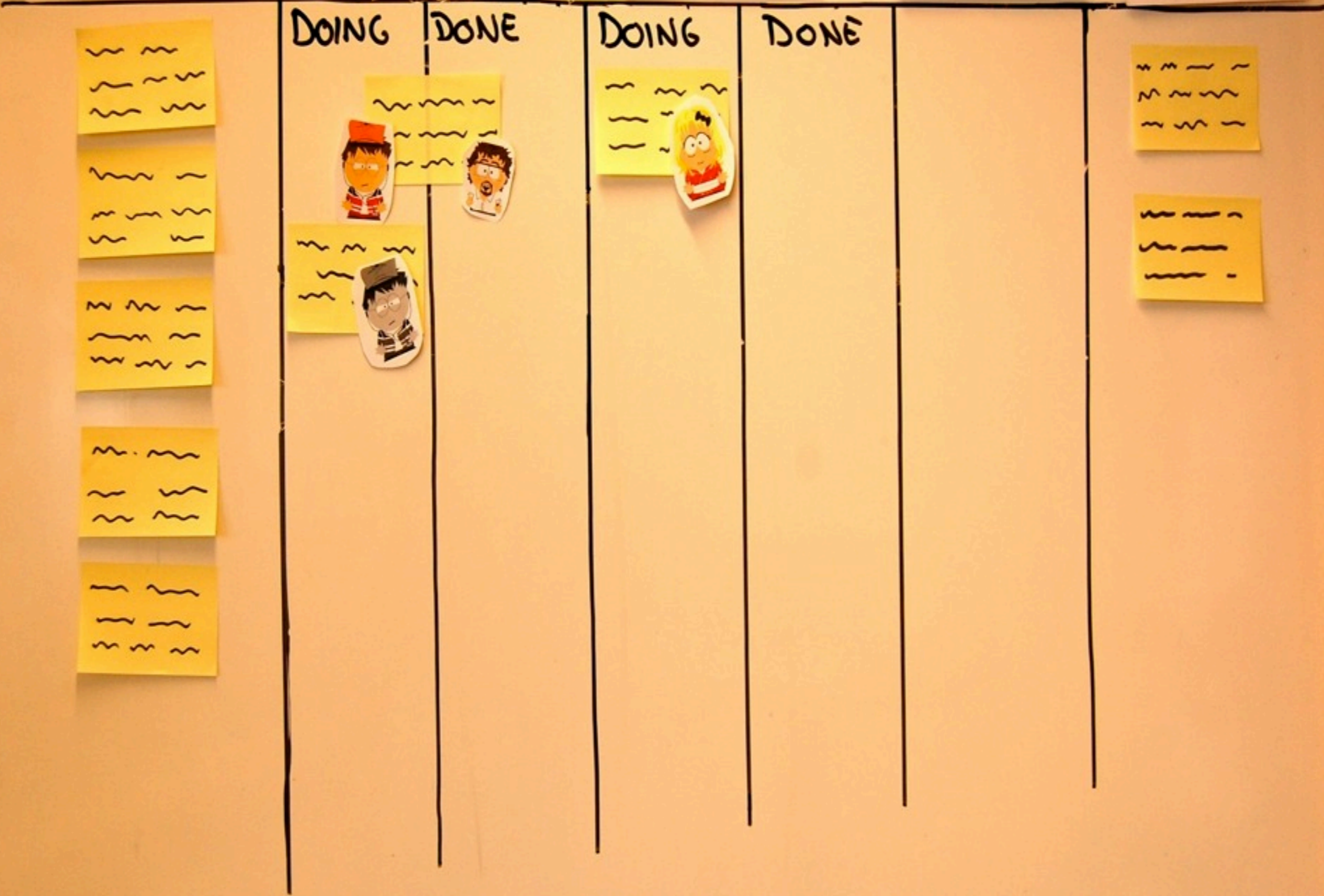
TODO

ANALYZE  
2

DEV  
3

TEST  
2

DONE



måndag 13 februari 12

In this way they can resolve the bottleneck and move it to ready for development.



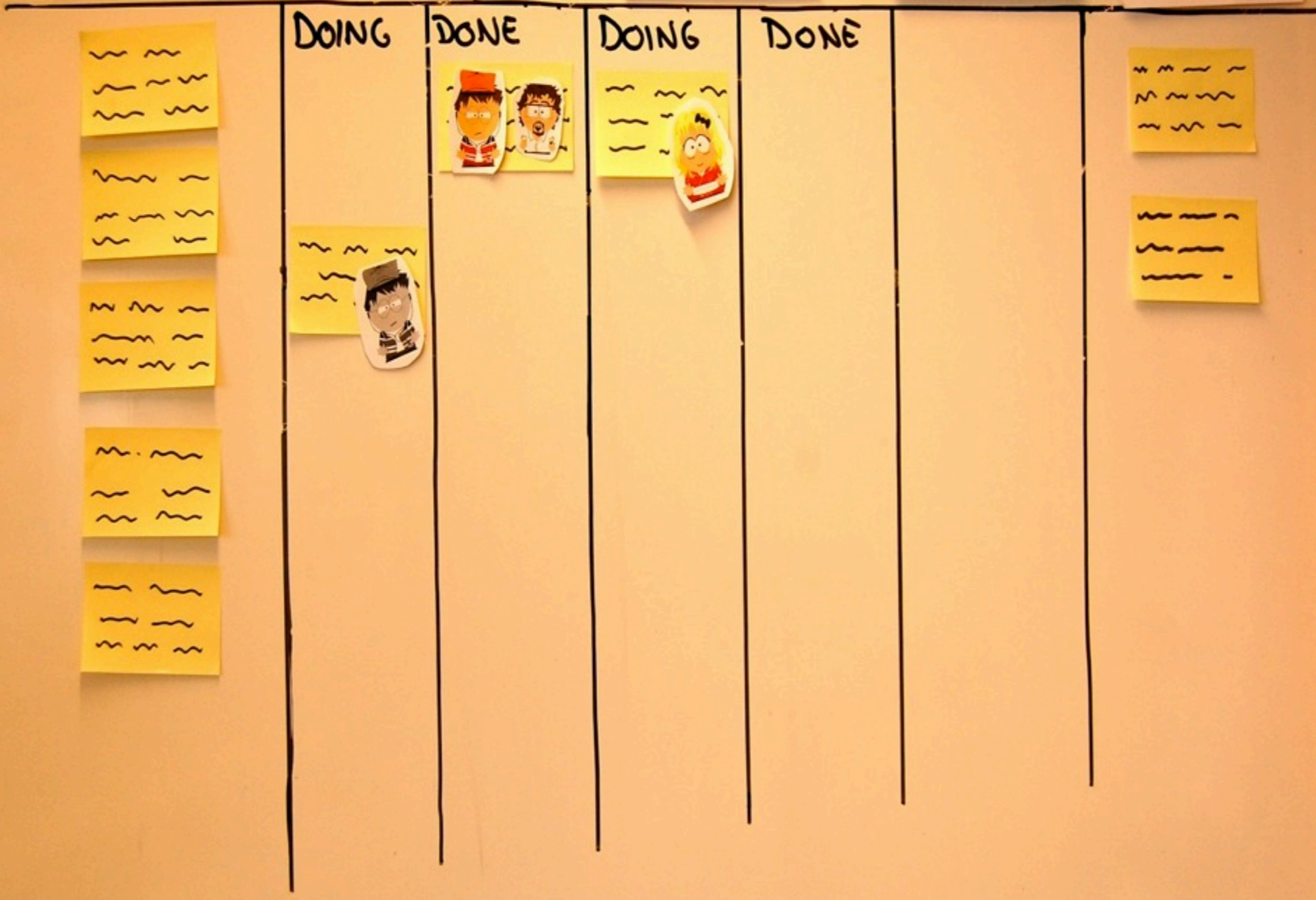
TODO

ANALYZE  
2

DEV  
3

TEST  
2

DONE



måndag 13 februari 12

Great - another bottleneck resolved!

But, if he couldn't have done that either? What should Marcus do then?

Well - find other interesting work maybe. Prepare that automation of tests or get that pesky build server to run faster. Something to help the team.

Strive to not pull new work. You could, and break your WIP limits - but remember that then all work to come will suffer and take longer time to complete.



# What to work on next?

1. Can you help finish an existing item? Do that.
2. Don't have the right skills? Find a bottleneck and work to release it.
3. Don't have the right skills for that? Pull work from the queue.
4. Can't do that either? Find other interesting work.

Source: David Joyce, "Pulling Value Lean and Kanban"

måndag 13 februari 12

So in summary, what should you work on next?

1. Can you help finish an existing item? Do that.
2. Don't have the right skills? Find a bottleneck and work to release it.
3. Don't have the right skills for that? Pull work from the queue.
4. Can't do that either? Find other interesting work.



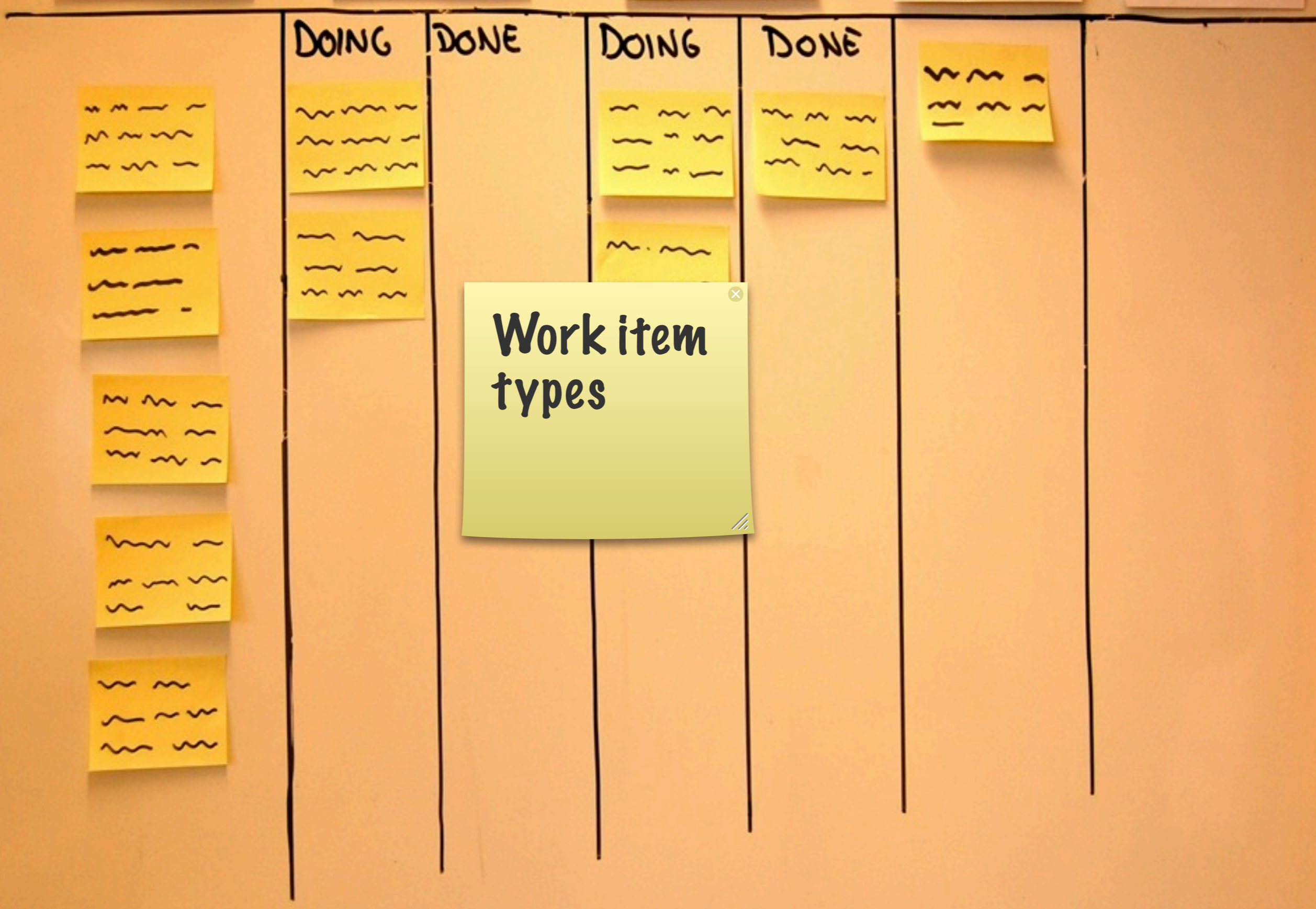
TODO

ANALYZE  
2

DEV  
34

TEST  
2

DONE



Work item types

måndag 13 februari 12

With different work item types you can visualize how different kind of work should be treated in different ways. To enable the team to easier be self managed.

Up to now all the items on our board looks and are treated the same.



TODO

ANALYZE  
2

DEV  
34

TEST  
2

DONE



måndag 13 februari 12

But they are not, e.g., this is a bug.



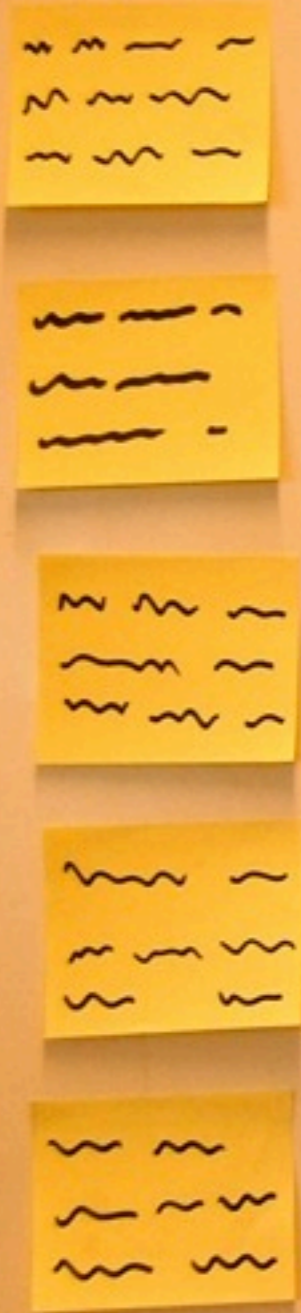
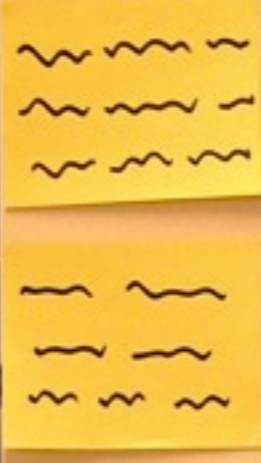
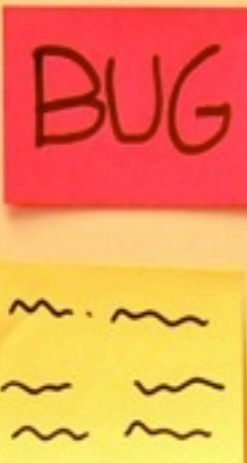
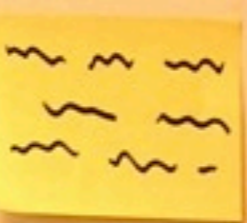

TODO

ANALYZE  
2

DEV  
34

TEST  
2

DONE

|                                                                                    | DOING                                                                             | DONE | DOING                                                                               | DONE                                                                                |                                                                                     |
|------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
|  |  |      |  |  |  |

måndag 13 februari 12

And here is a maintenance issue that we are testing.



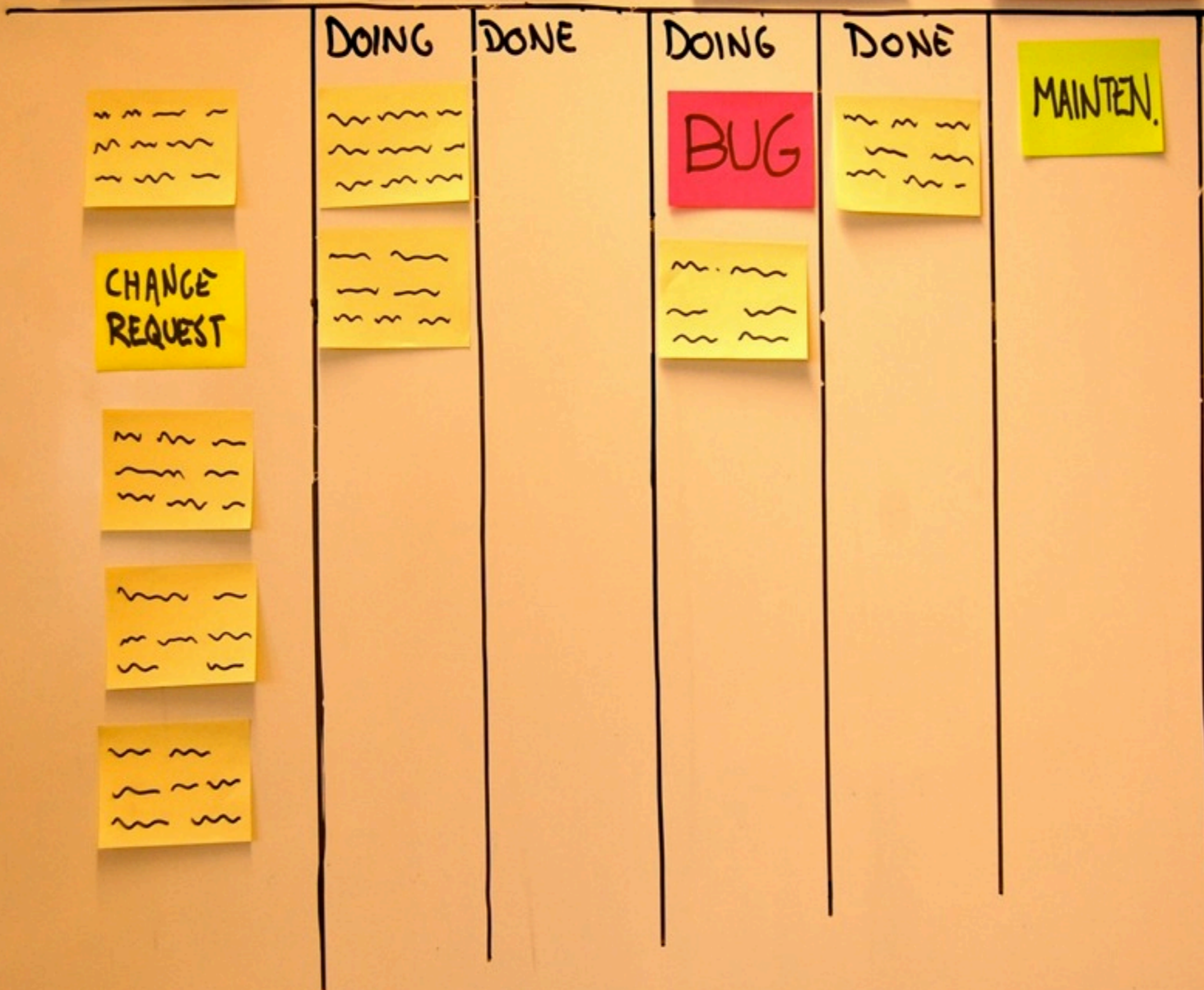
TODO

ANALYZE  
2

DEV  
34

TEST  
2

DONE



måndag 13 februari 12

Over here is a change request that is waiting to be pulled.



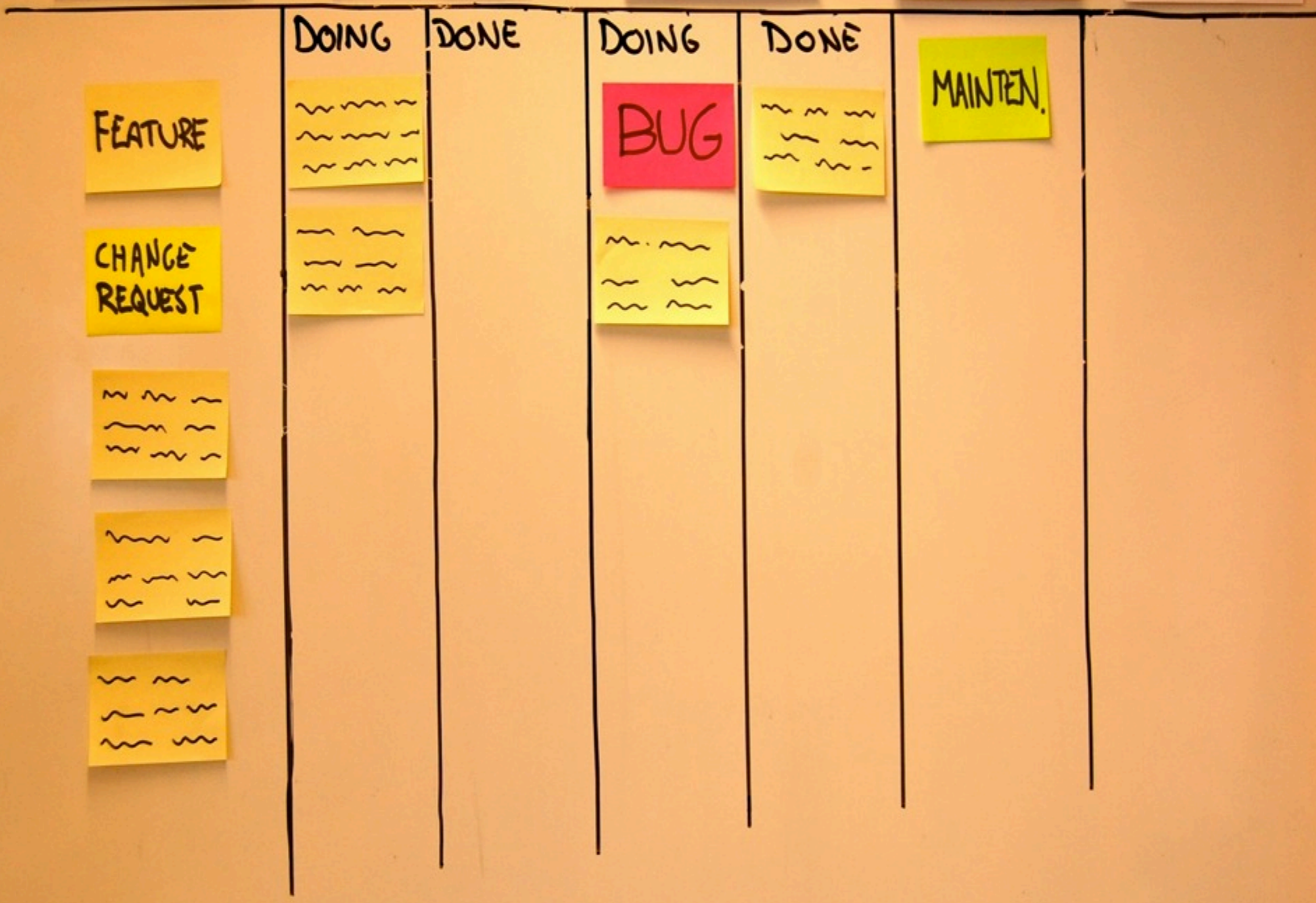
TODO

ANALYZE  
2

DEV  
34

TEST  
2

DONE



måndag 13 februari 12

And these normal yellow ones are ordinary features



TODO

ANALYZE  
2

DEV  
34

TEST  
2

DONE

DOING

DONE

DOING

DONE

BUG

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

MAINTEN.

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

CHANGE
REQUEST

~~~~~  
~~~~~  
~~~~~

FEATURE

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

måndag 13 februari 12

Now it's more visual to us what we are working on. If the board goes all red with bugs we know that we're having a quality issue for example.

To remind us and others what the color mean we could add a legend or key for the different items.

Avoid "the yellow sea".



AS A .....

I WANT TO .....

**Jocke**  
**Whats on**  
**a card**

måndag 13 februari 12

First and foremost it should describe the feature. In this case in the form of a user story.



AS A .....

I WANT TO .....



måndag 13 februari 12

The use of an avatar is an easy way to show who is working on what.

It could also be used to limit work in process since you only can put your avatar on one card.

A tip is to not go to far from the reality with your avatar since that will miss to communicate who the avatar is. Dogs for example is not a good idea



2010-03-01

AS A .....

I WANT TO .....



måndag 13 februari 12

Here is a hard deadline added to the card – so everyone knows how we are doing against it.



TFS: 52879

2010-03-01

AS A .....

I WANT TO .....



måndag 13 februari 12

If an electronic system exists with more information you could add the id of the item.



TFS: 52879

2010-03-01

AS A .....

I WANT TO .....

A: 2010-01-19



måndag 13 februari 12

“Stamp” the date of each stage to get data for lead and cycle times that can be used to create cumulative flow diagrams, for metrics to improve or make predictions etc.

This card entered the Analyze step 2010-01-19



TFS: 52879

2010-03-01

AS A .....

I WANT TO .....

A: 2010-01-19

D: 2010-02-01



måndag 13 februari 12

and did not reach Development until 2010-02-01...

What was going on there?



TFS: 52879

2010-03-01

AS A .....

I WANT TO .....



A: 2010-01-19

D: 2010-02-01



måndag 13 februari 12

Use other items to signal deviations, such as delays or prioritization.

As you can see we have plenty of room on our card ...



TFS: 52879

2010-03-01

AS A .....

I WANT TO .....



A: 2010-01-19

D: 2010-02-01



måndag 13 februari 12

Yes, our hand writing is that bad.

=====



# Design goal

- Facilitate decision making
- Visualization info team members to optimize outcomes
- Type & Class of work

måndag 13 februari 12

Design goals:

Facilitate decision making

Visualization info team members to optimize outcomes

– Risk mgmt

– Customer satisf

– Economics

Type & Class of work

– Set of policies

– Needed info to facilitate following the policy

Visual cue 50% & 100% of lead time/SLA

Specialization – Needed – Done

Number of days in columns (dots on or next to card)

Dots to denote clock ticking on SLA (in Waiting for External Group), then blocker sticky when late against SLA

Cost of delay symbol

Other things:

Progress bar, dots on or next to card

Red dot/magnet





måndag 13 februari 12

If we want our team to be self organized, we need to help them to know which feature to work on next.

With classes of service we get a prioritization and policy approach that can help team members to chose their next work item.

It will also help you increase customer satisfaction by using different service levels for different types of work.

A common situation is to introduce an expedite/urgent/silver bullet for things that is need to be prioritized over other work.



TODO

ANALYZE  
2

DEV  
34

TEST  
2

DONE

|                | DOING | DONE | DOING | DONE |   |  |
|----------------|-------|------|-------|------|---|--|
| BUG            | 3     | 0    | 3     | 1    | 2 |  |
| MAINTEN.       | 3     | 0    | 3     | 0    | 1 |  |
| CHANGE REQUEST | 3     | 0    | 0     | 0    | 0 |  |
| FEATURE        | 3     | 0    | 0     | 0    | 0 |  |
| URGENT!        |       |      |       |      |   |  |

måndag 13 februari 12

Those items are often given their own "swim lane"



TODO

ANALYZE  
2

DEV  
34

TEST  
2


DONE

BUG

~~~~~  
~~~~~  
~~~~~

DOING
~~~~~
~~~~~

DONE  
~~~~~  
~~~~~  
~~~~~

DOING
~~~~~
~~~~~

DONE  
~~~~~  
~~~~~  
~~~~~

~~~~~
~~~~~

~~~~~  
~~~~~  
~~~~~

MAINTEN.

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~

CHANGE
REQUEST

~~~~~  
~~~~~  
~~~~~

FEATURE

~~~~~  
~~~~~  
~~~~~

URGENT!



måndag 13 februari 12

Everyone is happily working on their items...

TODO

ANALYZE
2

DEV
34

TEST
2

DONE

		DOING	DONE	DOING	DONE		
BUG	[wavy lines]	[wavy lines, cartoon character]	[wavy lines]	[wavy lines, cartoon character]	[wavy lines]	[wavy lines, cartoon character]	[wavy lines]
MAINTEN.	[wavy lines]			[wavy lines, cartoon character]			
CHANGE REQUEST	[wavy lines]						
FEATURE	[wavy lines]						
URGENT!	[wavy lines, red border]						

måndag 13 februari 12

... when suddenly an urgent feature is introduced.

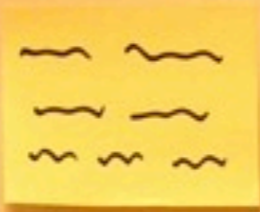
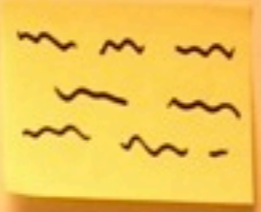

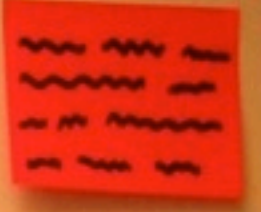




TODO

ANALYZE
2

DEV
34

TEST
2

DONE

	DOING	DONE	DOING	DONE		
BUG			 		 	
MAINTEN.						
CHANGE REQUEST						
FEATURE						
URGENT! 			 			

måndag 13 februari 12

A common policy for urgent features is that it should be prioritized over all other work. Just let go and start to work on the urgent feature to move it quickly through the whole process.

TODO

ANALYZE
2

DEV
34

TEST
2

DONE

DOING

DONE

DOING

DONE

BUG

~~~~~  
~~~~~  
~~~~~



~~~~~  
~~~~~  
~~~~~



~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

MAINTEN.

~~~~~  
~~~~~  
~~~~~



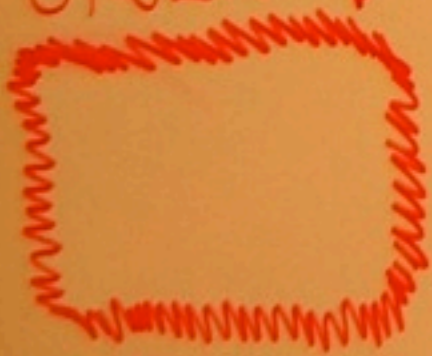
CHANGE
REQUEST

~~~~~  
~~~~~  
~~~~~

FEATURE

~~~~~  
~~~~~  
~~~~~

URGENT!



måndag 13 februari 12

Phew – the developers were quick. The item is pulled by the tester and get going on it immediately.



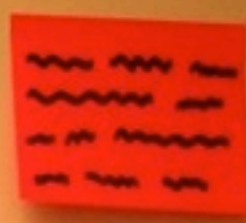


TODO

ANALYZE
2

DEV
34

TEST
2

DONE

		DOING	DONE	DOING	DONE		
BUG	~~~~~ ~~~~~ ~~~~~		~~~~~ ~~~~~ ~~~~~		~~~~~ ~~~~~ ~~~~~		
MAINTEN.	~~~~~ ~~~~~ ~~~~~						
CHANGE REQUEST	~~~~~ ~~~~~ ~~~~~						
FEATURE	~~~~~ ~~~~~ ~~~~~						
URGENT!							

måndag 13 februari 12

And in no time the item is handled.

TODO

ANALYZE
2

DEV
34

TEST
2

DONE

	DOING	DONE	DOING	DONE		
BUG						
MAINTEN.						
TECHNICAL						
FEATURE						
URGENT!						

måndag 13 februari 12

Another kind of policy is a minimum number or percentage of items of a certain kind.

TODO

ANALYZE
2

DEV
34

TEST
2

DONE

	DOING	DONE	DOING	DONE		
<p>BUG 12,5%</p> <p>MAINTEN. 25%</p> <p>TECHNICAL 12,5%</p> <p>FEATURE 50%</p>						
						
<p>URGENT!</p> 						

måndag 13 februari 12

So here we should strive to have 50% features, 12,5 % (1) technical stories, 25% maintenance and the rest bugs.

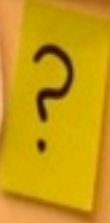
TODO

ANALYZE
2

DEV
34

TEST
2

DONE

	DOING	DONE	DOING	DONE		
<p>BUG 12.5%</p> <p>MAINTEN. 25%</p> <p>TECHNICAL 12.5%</p> <p>FEATURE 50%</p>			 			   
<p>URGENT!</p> 						

måndag 13 februari 12

And with that policy in place Marcus, the tester, can easily see what kind of work item should be the next to prioritize for the todo queue – since a maintenance issue was finished, we put a new maintenance issue in the todo queue.

TODO

ANALYZE
2

DEV
34

TEST
2

DONE

	DOING	DONE	DOING	DONE		
<p>BUG 12.5%</p> <p>MAINTEN. 25%</p> <p>TECHNICAL 12.5%</p> <p>FEATURE 50%</p>	 		 	 	 	 
<p>URGENT!</p> 						

måndag 13 februari 12

Here's another situation.

Test-Marcus is ready to pull new work from the dev done queue. There are two items ready to be pulled.

Which one should he choose?

TODO

ANALYZE
2

DEV
34

TEST
2

DONE

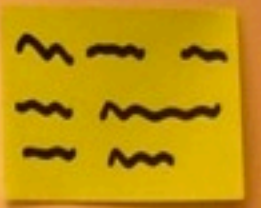
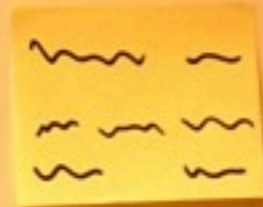
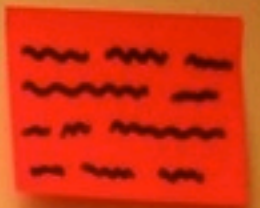
DOING

DONE

DOING

DONE

BUG 125%
 MAINTEN. 25%
 TECHNICAL 125%
 FEATURE 50%



URGENT!



måndag 13 februari 12

The maintenance features (light green) has certainly been in the queue long, but we have a policy in place that tells us to always prioritize features (yellow) above maintenance, so First-In-First-Out does not apply in this case.

TODO

ANALYZE
2

DEV
34

TEST
2

DONE

	DOING	DONE	DOING	DONE		
<p>BUG 12.5%</p> <p>MAINTEN. 25%</p> <p>TECHNICAL 12.5%</p> <p>FEATURE 50%</p>			 			 
<p>URGENT!</p> 						

måndag 13 februari 12

Test-Marcus can then easily know what to pick.

More powerful when there is more work than in this simple sample.

Also: use to guarantee a minimum of a certain type of stories, such as retrospective actions, dealing with technical debt, low prioritised maintenance or customers.



måndag 13 februari 12

Many Kanban teams use the short daily stand-up that is common in agile methods such as Scrum.

Kanban standup is that the meeting facilitator **enumerates work, not people**. Traverse the board from **right to left** (downstream to upstream) in order to emphasize pull.

The **board shows the status**, instead we focus on the exceptions. THE SMELLS.

- Any bottlenecks?
- Impediments not handled?
- Are we keeping our WIP limits?
- Is the prioritization clear?

Fokus on smells?
Do you see any here? Or is it a yellow sea?

JOAKIM FLERA LAPPAR
HINDER-LAPPEN
MIKAELA EJ MEST PRIO



måndag 13 februari 12

How are defects handled on a Kanban board? Can items travel backwards?

This is one of the most common questions we get when doing this presentation. Here are some different strategies for handling defects.

When Test-Marcus finds a defect he could...

TODO

ANALYZE
2

DEV
34

TEST
2

DONE

	DOING	DONE	DOING	DONE		
<p>BUG 12.5%</p> <p>MAINTEN. 25%</p> <p>TECHNICAL 12.5%</p> <p>FEATURE 50%</p>						
						
<p>URGENT!</p> 						

måndag 13 februari 12

1. Mark the feature as blocked

TODO

ANALYZE
2

DEV
34

TEST
2

DONE

	DOING	DONE	DOING	DOING	DONE	DONE
<p>BUG 12.5%</p> <p>MAINTEN. 25%</p> <p>TECHNICAL 12.5%</p> <p>FEATURE 50%</p>	 		 	  	 	
<p>URGENT!</p> 						

måndag 13 februari 12

2. Create a new defect work item


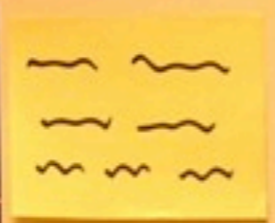

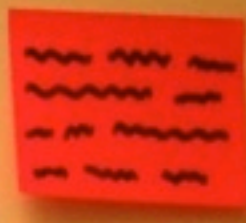


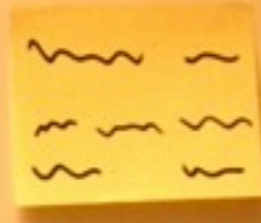
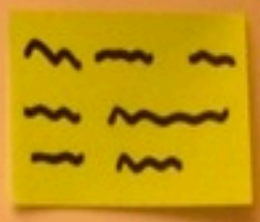
TODO

ANALYZE
2

DEV
84

TEST
2

DONE

	DOING	DONE	DOING	DONE		
<p>BUG 12.5%</p> <p>MAINTEN. 25%</p> <p>TECHNICAL 12.5%</p> <p>FEATURE 50%</p>						
						
<p>URGENT!</p> 						

måndag 13 februari 12

3. And place it in ready for development. Or maybe in Todo if it needs to be analyzed before development.


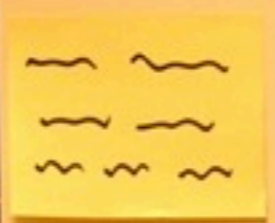

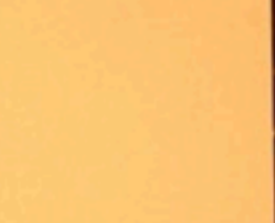

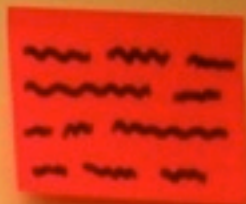
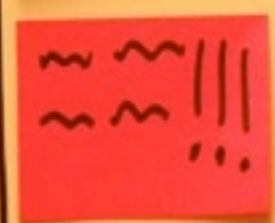



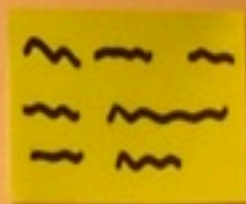

TODO

ANALYZE
2

DEV
34

TEST
2

DONE

	DOING	DONE	DOING	DONE		
<p>BUG 12.5%</p> <p>MAINTEN. 25%</p> <p>TECHNICAL 12.5%</p> <p>FEATURE 50%</p>						
						
<p>URGENT!</p> 						

måndag 13 februari 12

4. Work on something else in the meantime. Given that Test-Marcus cannot resolve the issue himself.



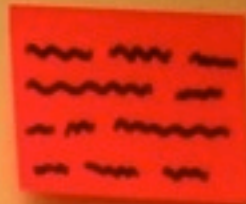

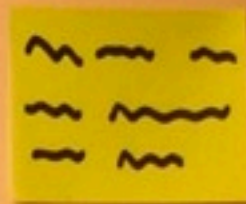

TODO

ANALYZE
2

DEV
34

TEST
2

DONE

	DOING	DONE	DOING	DONE		
<p>BUG 12.5%</p> <p>MAINTEN. 25%</p> <p>TECHNICAL 12.5%</p> <p>FEATURE 50%</p>						
						
<p>URGENT!</p> 						

måndag 13 februari 12

Another approach is that Test-Marcus

1. Mark his feature as blocked and
2. Creates a new defect work item




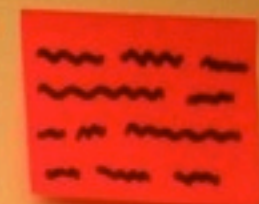


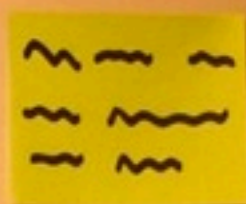
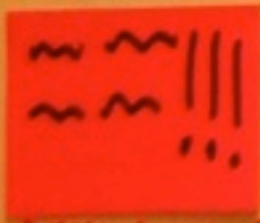
TODO

ANALYZE
2

DEV
34

TEST
2

DONE

	DOING	DONE	DOING	DONE		
<p>BUG 12.5%</p> <p>MAINTEN. 25%</p> <p>TECHNICAL 12.5%</p> <p>FEATURE 50%</p>						
						
<p>URGENT!</p> 						

måndag 13 februari 12

3. and place it in the Urgent swim lane for quick processing


TODO

ANALYZE
2

DEV
34

TEST
2

DONE

	DOING	DONE	DOING	DONE		
<p>BUG 12.5%</p> <p>MAINTEN. 25%</p> <p>TECHNICAL 12.5%</p> <p>FEATURE 50%</p>	 		 		 	 
<p>URGENT!</p> 						

måndag 13 februari 12

Test-Marcus could also.
1. Mark his feature as blocked


TODO

ANALYZE
2

DEV
34

TEST
2

DONE

	DOING	DONE	DOING	DONE		
<p>BUG 12.5%</p> <p>MAINTEN. 25%</p> <p>TECHNICAL 12.5%</p> <p>FEATURE 50%</p>						 
<p>URGENT!</p> 						

måndag 13 februari 12

2. "stop the line" and call for help. The whole team is "swarming" on the issue and work together to resolve it.

What policy you use, and wether blocked features and new defect items count against the WIP limit, depends on how much tension you want to create for your team and your organization. If you are a kick-ass-die-hard-zero-defects-team you probably want to stop the line or at least make work blocked by defects to count against the WIP limit and thus trigger a discussion and perhaps some improvement suggestions.

If not, well start with what you can do and work towards a target.





TODO

ANALYZE
2

DEV
34

TEST
2

DONE

	DOING	DONE	DOING	DONE		
<p>BUG 12.5%</p> <p>MAINTEN. 25%</p> <p>TECHNICAL 12.5%</p> <p>FEATURE 50%</p>						
						
<p>EXIT</p>	<ul style="list-style-type: none"> • ACCEPTANCE CRIT. DEFINED • REVIEWED BY P.O. • ... 		<ul style="list-style-type: none"> • 90% CODE COVERAGE • DEPLOYED TO TEST ENVIRONMENT • ... 			

måndag 13 februari 12

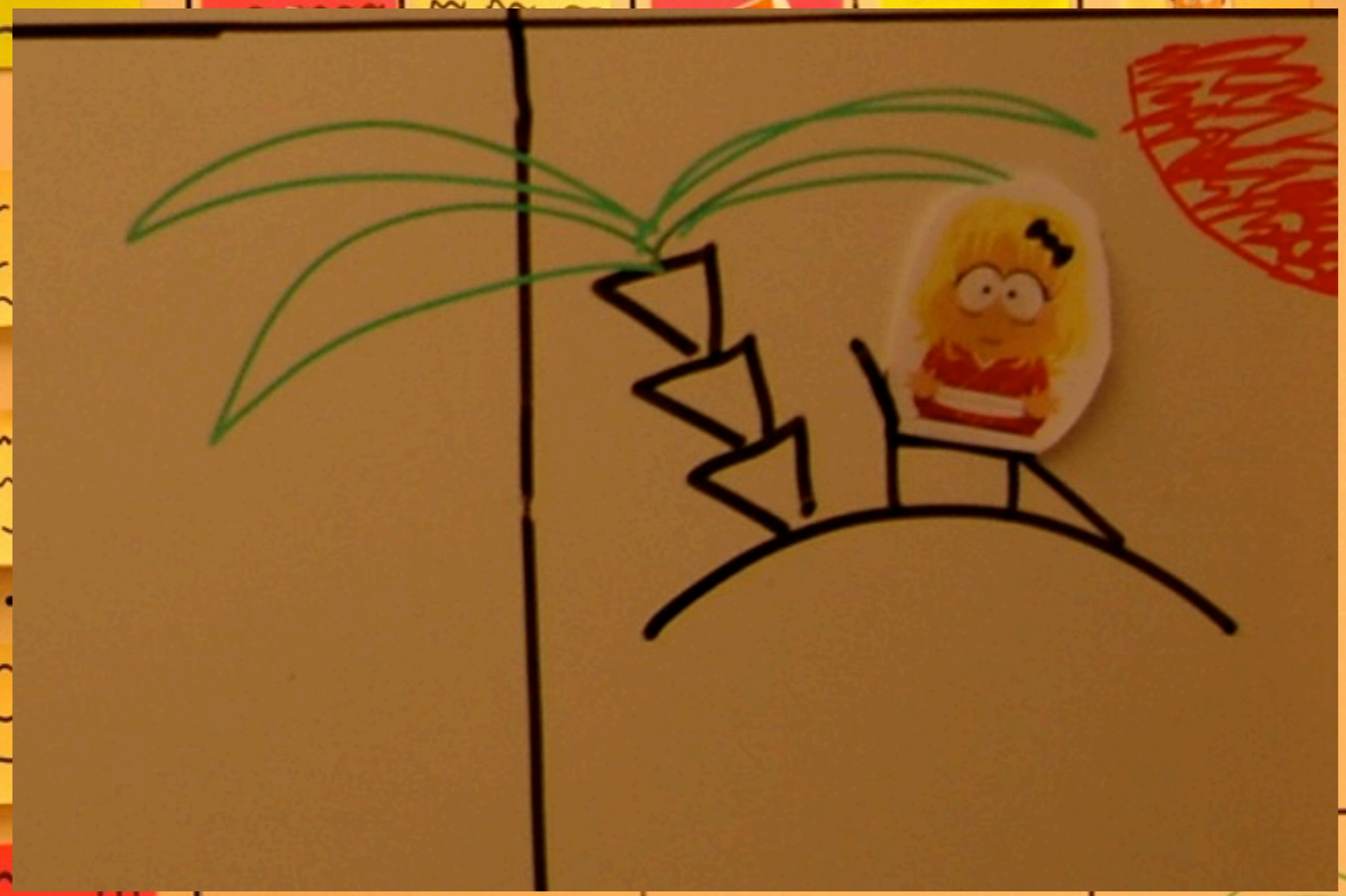
You could further help the team handle specialization with exit criteria for each column.

By defining exit criteria for what is needed to move items into the Done column, we make it explicit what has to be done before handing work off to the next step. This can be used as “standardized work” in the lean sense of using as a baseline for improvement and better communication and collaboration between different steps in the workflow.

TODO 6 ANALYZE 2 DEV 34 TEST 2 DONE

BUG
MAINTEN.
TECHNICAL
FEATURE

DOING DONE DOING DONE



måndag 13 februari 12

The board can be used to communicate other important information.

Here is a team member on vacation...

TODO
6

ANALYZE
2

DEV
34

TEST
2

DONE

BUG

MAINTEN.

TECHNICAL

FEATURE

Handwritten notes on a yellow sticky note.

Handwritten notes on a yellow sticky note.

Handwritten notes on a yellow sticky note.

Handwritten notes on a yellow sticky note.

Handwritten notes on a red sticky note.

Handwritten notes on a yellow sticky note.

DOING
Handwritten notes and a small cartoon character sticker.

DONE
Handwritten notes on a yellow sticky note.

DOING
Handwritten notes and a small cartoon character sticker.

DONE
Handwritten notes on a yellow sticky note.

Handwritten notes and a small cartoon character sticker.

Handwritten notes on a red sticky note.

Handwritten notes on a yellow sticky note.

Handwritten notes on a yellow sticky note.



måndag 13 februari 12

... and here is the same team member down with the swine flu.

Some teams really like the visualization in Kanban – embrace it!



måndag 13 februari 12

We have now reached the point where you cannot call it Scrum anymore. We will stop doing sprint planning and start doing our planning just-in-time, or at least not coupled to the same cadence as review/demo, retrospective etc.

How is the Todo-column filled with new items?

Start by filling the Todo-column or "backlog" with as many as the team usually commits to for a sprint, maybe with a small buffer.

Agree on a cadence based on coordination costs and reprioritisation needs.

TODO

ANALYZE
2

DEV
34

TEST
2

DONE

		DOING	DONE	DOING	DONE		
BUG	[wavy lines]	[wavy lines]	[wavy lines]	[wavy lines]		[wavy lines]	[wavy lines]
MAINTEN.	[wavy lines]			[wavy lines]	[wavy lines]	[wavy lines]	[wavy lines]
TECHNICAL	[wavy lines]						
FEATURE	[wavy lines]						
	[wavy lines]						
	[wavy lines]						

måndag 13 februari 12

If coordination costs are low: event-driven planning.

Decouple the planning from the release and retrospective cycle and start planning just in time.

TODO

ANALYZE
2

DEV
34

TEST
2

DONE

	DOING	DONE	DOING	DONE		
BUG						
MAINTEN.						
TECHNICAL						
FEATURE						

måndag 13 februari 12

Maybe once every week or when event driven with an order point when new items should be added.

TODO

ANALYZE
2

DEV
34

TEST
2

DONE

		DOING	DONE	DOING	DONE		
BUG		3 yellow sticky notes		1 red sticky note	1 yellow sticky note	1 yellow sticky note	1 red sticky note
MAINTEN.				1 yellow sticky note		1 yellow sticky note	1 yellow sticky note
TECHNICAL	1 red sticky note			1 yellow sticky note			1 yellow sticky note
FEATURE	1 yellow sticky note						1 yellow sticky note
	1 yellow sticky note						
	1 yellow sticky note						
	1 yellow sticky note						

måndag 13 februari 12

We are closing in on the order point

TODO

ANALYZE
2

DEV
34

TEST
2

DONE

BUG

MAINTEN.

TECHNICAL

FEATURE

.....

.....

.....

.....

.....

.....

.....

.....

.....

DOING

~~~~~  
~~~~~  
~~~~~

DONE

~~~~~  
~~~~~  
~~~~~

DOING

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

DONE

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~  
~~~~~  
~~~~~

måndag 13 februari 12

When all the items above the order point is moved the rest is moved up.

Note the instruction card that tells you to call to a meeting to get new items in the backlog from the product owner. That is referred to in the industry as a Kanban card



TODO

ANALYZE  
2

DEV  
34

TEST  
2

DONE

BUG

MAINTEN.

TECHNICAL

FEATURE

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

DOING

~~~~~  
~~~~~  
~~~~~

DONE

~~~~~  
~~~~~  
~~~~~

DOING

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

DONE

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

måndag 13 februari 12

Three new items are prioritized by the product owner (or who ever is doing the prioritization in your organization)...



TODO

ANALYZE  
2

DEV  
34

TEST  
2

DONE

BUG

MAINTEN.

TECHNICAL

FEATURE

~ ~ ~  
~ ~ ~  
~ ~ ~

~ ~ ~  
~ ~ ~  
~ ~ ~

~ ~ ~  
~ ~ ~  
~ ~ ~

~ ~ ~  
~ ~ ~  
~ ~ ~

~ ~ ~  
~ ~ ~  
|||  
|||

~ ~ ~  
~ ~ ~  
~ ~ ~

DOING

~ ~ ~  
~ ~ ~  
~ ~ ~

DONE

~ ~ ~  
~ ~ ~  
~ ~ ~

DOING

~ ~ ~  
~ ~ ~  
~ ~ ~

~ ~ ~  
~ ~ ~  
~ ~ ~

~ ~ ~  
~ ~ ~  
~ ~ ~

DONE

~ ~ ~  
~ ~ ~  
~ ~ ~

~ ~ ~  
~ ~ ~  
~ ~ ~

~ ~ ~  
~ ~ ~  
~ ~ ~

~ ~ ~  
~ ~ ~  
~ ~ ~

~ ~ ~  
~ ~ ~  
~ ~ ~

~ ~ ~  
~ ~ ~  
~ ~ ~

~ ~ ~  
~ ~ ~  
~ ~ ~

måndag 13 februari 12  
and added to the backlog



TODO  
6

ANALYZE  
2

DEV  
34

TEST  
2

DONE

BUG

MAINTEN.

TECHNICAL

FEATURE

~ ~ ~  
~ ~ ~  
~ ~ ~

~ ~ ~  
~ ~ ~  
~ ~ ~

~ ~ ~  
~ ~ ~  
~ ~ ~

~ ~ ~  
~ ~ ~  
~ ~ ~

~ ~ ~  
~ ~ ~  
|||  
|||

~ ~ ~  
~ ~ ~  
~ ~ ~

DOING

~ ~ ~  
~ ~ ~  
~ ~ ~

DONE

~ ~ ~  
~ ~ ~  
~ ~ ~

DOING

~ ~ ~  
~ ~ ~  
~ ~ ~

~ ~ ~  
~ ~ ~  
~ ~ ~

~ ~ ~  
~ ~ ~  
~ ~ ~

DONE

~ ~ ~  
~ ~ ~  
~ ~ ~

~ ~ ~  
~ ~ ~  
~ ~ ~

~ ~ ~  
~ ~ ~  
~ ~ ~

~ ~ ~  
~ ~ ~  
~ ~ ~

~ ~ ~  
~ ~ ~  
~ ~ ~

~ ~ ~  
~ ~ ~  
~ ~ ~

~ ~ ~  
~ ~ ~  
~ ~ ~

måndag 13 februari 12

A WIP-limit on the backlog indicates how much that should be planned in advance.



# Metrics

Jocke!



måndag 13 februari 12

Tracking WIP

Lead Time, spectral analysis to find outliers and items that just barely missed failed to meet the target time.

Due Date Performance

Throughput

Issues and Blocked Work Items

Escaped defects (as percentage of WIP/throughput)

Failure load/demand.



# Story Points



måndag 13 februari 12

Do you use story points? Do you understand them? :-)

I've met teams that have been doing Scrum for a year and still don't get it.

Customers and stakeholders don't see the point, especially when the points mean different things to different teams.

SM/PO/PL transforms SP to days to be able to plan.

What happens to the velocity if the team changes? Vacations, sick days and so on.

Can you be certain that more SP per sprint actually is improved velocity and not just a change in estimation? Something that was 5 SP with low knowledge of domain maybe later is estimated to 3 or 2. Velocity is suddenly lower, when work is actually being finished faster.

Too subjective and at risk of manipulation.



# Real Data



måndag 13 februari 12

Why not collect data that considers the entire value stream? Analysis, deployment, service class, delays due to impediments etc.

In contrast lean uses 'lead time' which is much harder to game as it records total time from when a customer requested the work to when the finished work was received by the customer.

Real data = real commitments



# Throughput (Capacity)

| <u>Last 4 Weeks</u> |            |        |       |       |
|---------------------|------------|--------|-------|-------|
|                     | Throughput |        |       |       |
| Start Date          | Small      | Medium | Large | TOTAL |
| 30-Mar-09           | 7          | 7      | 0     | 1     |
| 23-Mar-09           | 9          | 4      | 5     | 1     |
| 16-Mar-09           | 3          | 2      | 1     |       |
| 09-Mar-09           | 6          | 5      | 1     | 1     |



# Cycle Time

| Mean Cycle Time - All Records |      |
|-------------------------------|------|
| Size                          | Days |
| Small                         | 7.4  |
| Medium                        | 17.0 |
| Large                         | 25.2 |

| Mean Cycle Time - Engineering |      |
|-------------------------------|------|
| Size                          | Days |
| Small                         | 6.4  |
| Medium                        | 12.5 |
| Large                         | 15.0 |

| Name | Cycle | Lead  |
|------|-------|-------|
| ...  | 12.0  | 13.0  |
| ...  | 41.6  | 112.4 |
| ...  | 5.0   | 22.0  |
| ...  | 14.0  | 18.0  |
| ...  | 6.0   | 14.0  |
| ...  | 10.0  | 13.0  |
| ...  | 5.5   | 6.0   |
| ...  | 9.0   | 32.2  |
| ...  | 28.0  | 29.0  |
| ...  | 6.3   | 11.7  |
| ...  | 15.0  | 19.0  |
| ...  | 3.9   | 8.8   |
| ...  | 23.5  | 47.5  |

måndag 13 februari 12

Till höger: feature set/område. Den med högre var mycket legacy-kod, tredje part inblandad o.s.v. Bra data om man ska jobba med detta feature set senare.



TODO  
6

ANALYZE  
2

DEV  
34

TEST  
2

DONE

BUG

MAINTEN.

TECHNICAL

FEATURE

YOUR WAIT  
TIME FROM  
HERE IS  
18 DAYS  
↓

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
|||  
|||

~~~~~  
~~~~~  
~~~~~

DOING
~~~~~  
~~~~~  
~~~~~

DONE  
~~~~~  
~~~~~  
~~~~~

DOING
~~~~~  
~~~~~  
~~~~~

DONE  
~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

~~~~~  
~~~~~  
~~~~~

måndag 13 februari 12

This is referred to as Disneyland wait times, from the queues at Disneyland that tells you that you only have to wait 30 more minutes before 1,5 minutes of roller coaster joy is yours.

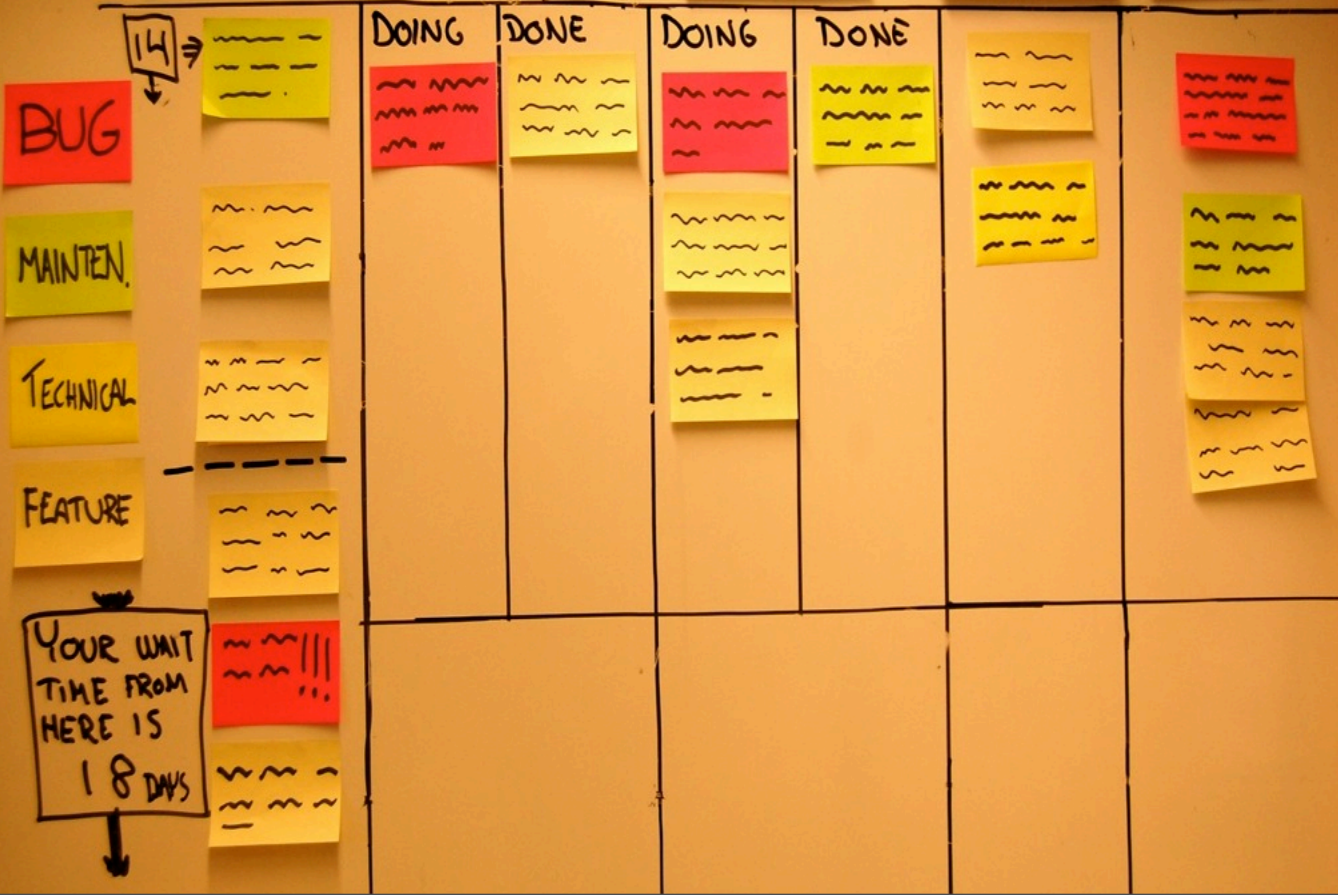
TODO
6

ANALYZE
2

DEV
34

TEST
2

DONE



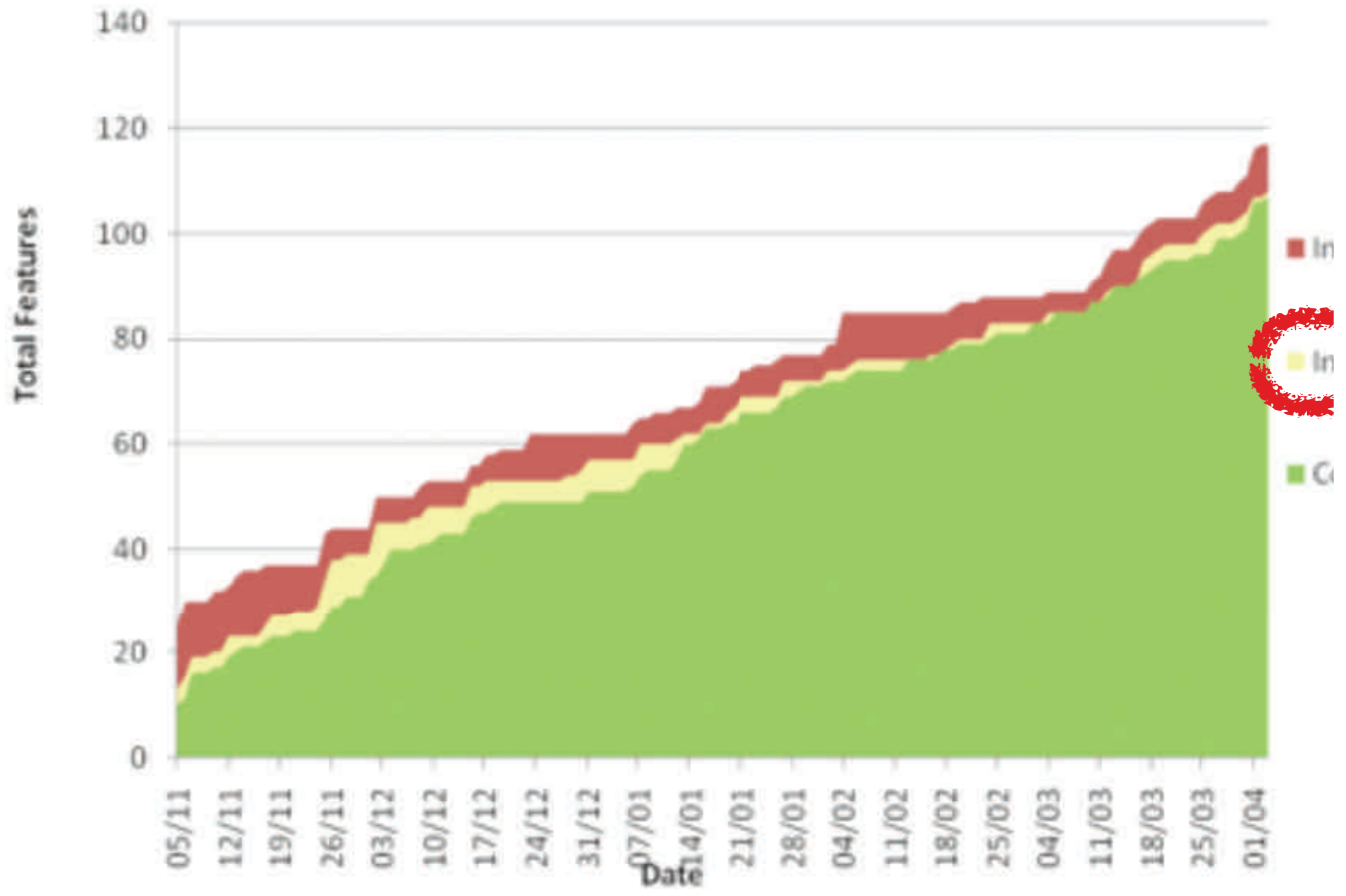
måndag 13 februari 12

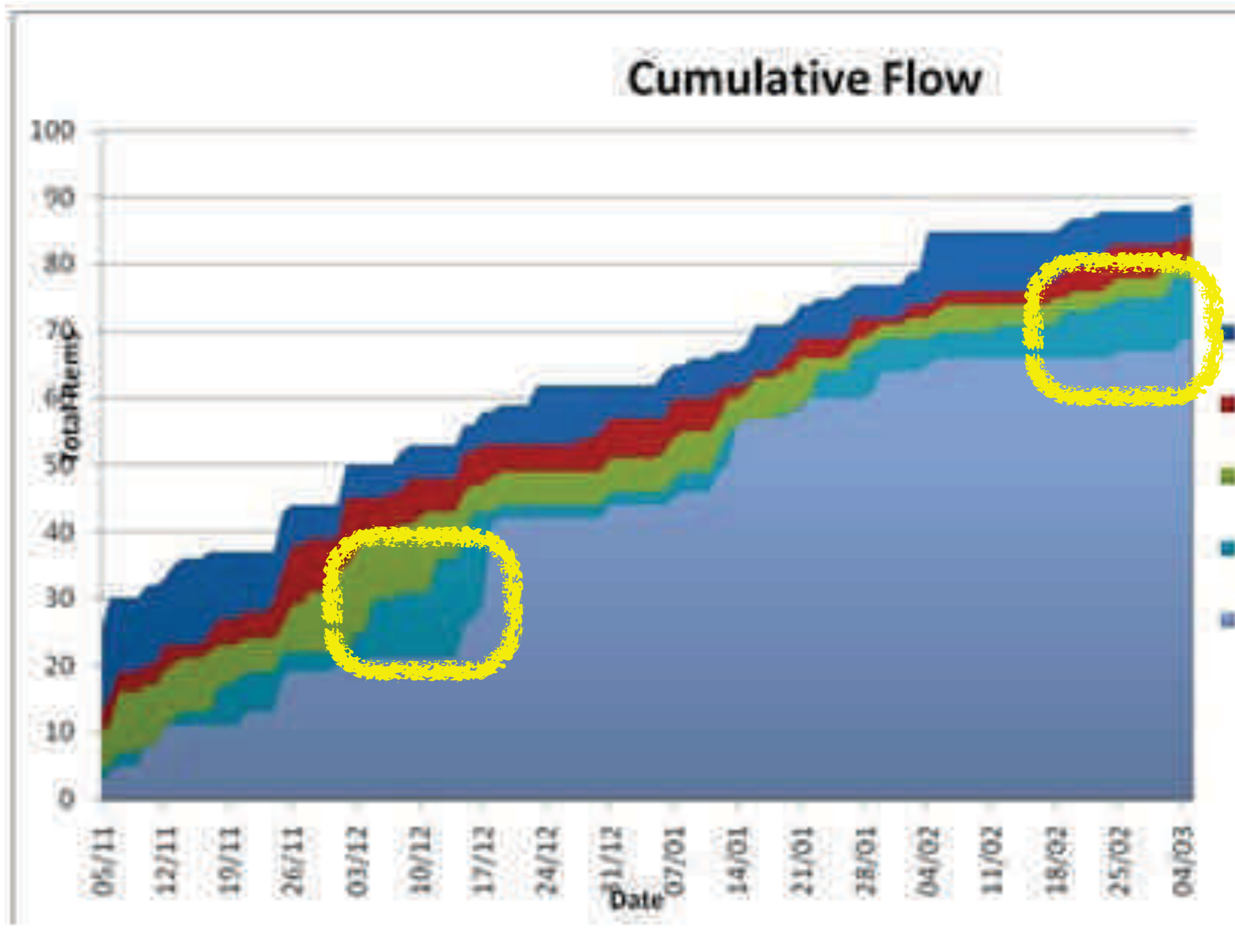
Here we can see that the item on top only have 14 days to go before it's in production.

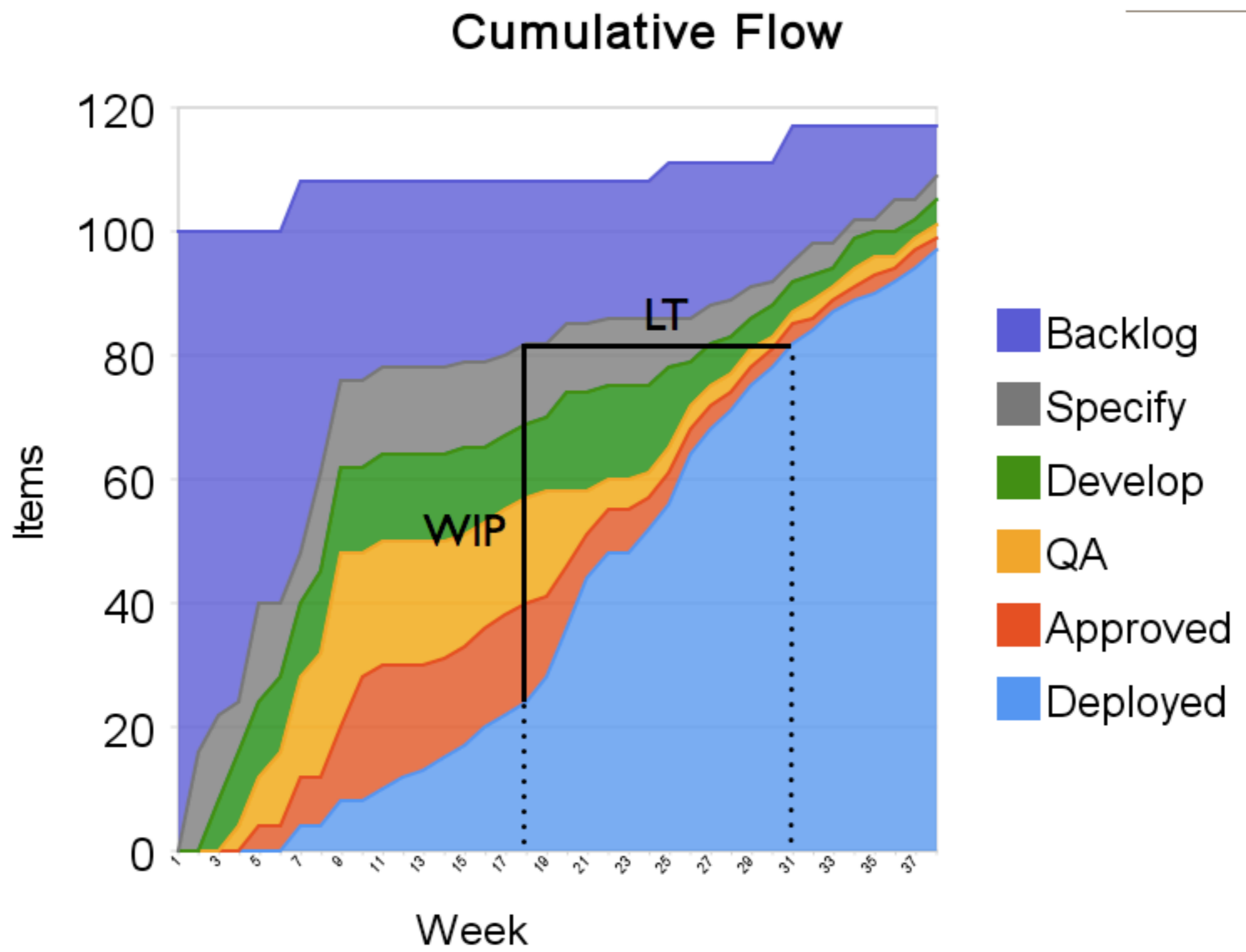
Many managers and clients actually value predictability more than speed (to a certain amount of course), but when there are no policies for how work flows through the system and no predictable delivery times lots of requests have to be expedited and the development team confuses need for predictability with need for speed and thus get disturbed by "changed prioritization" all the time.

Disneyland wait times, or actually SLA:s and due date performance, can sometimes help solve this problem.

Cumulative Flow



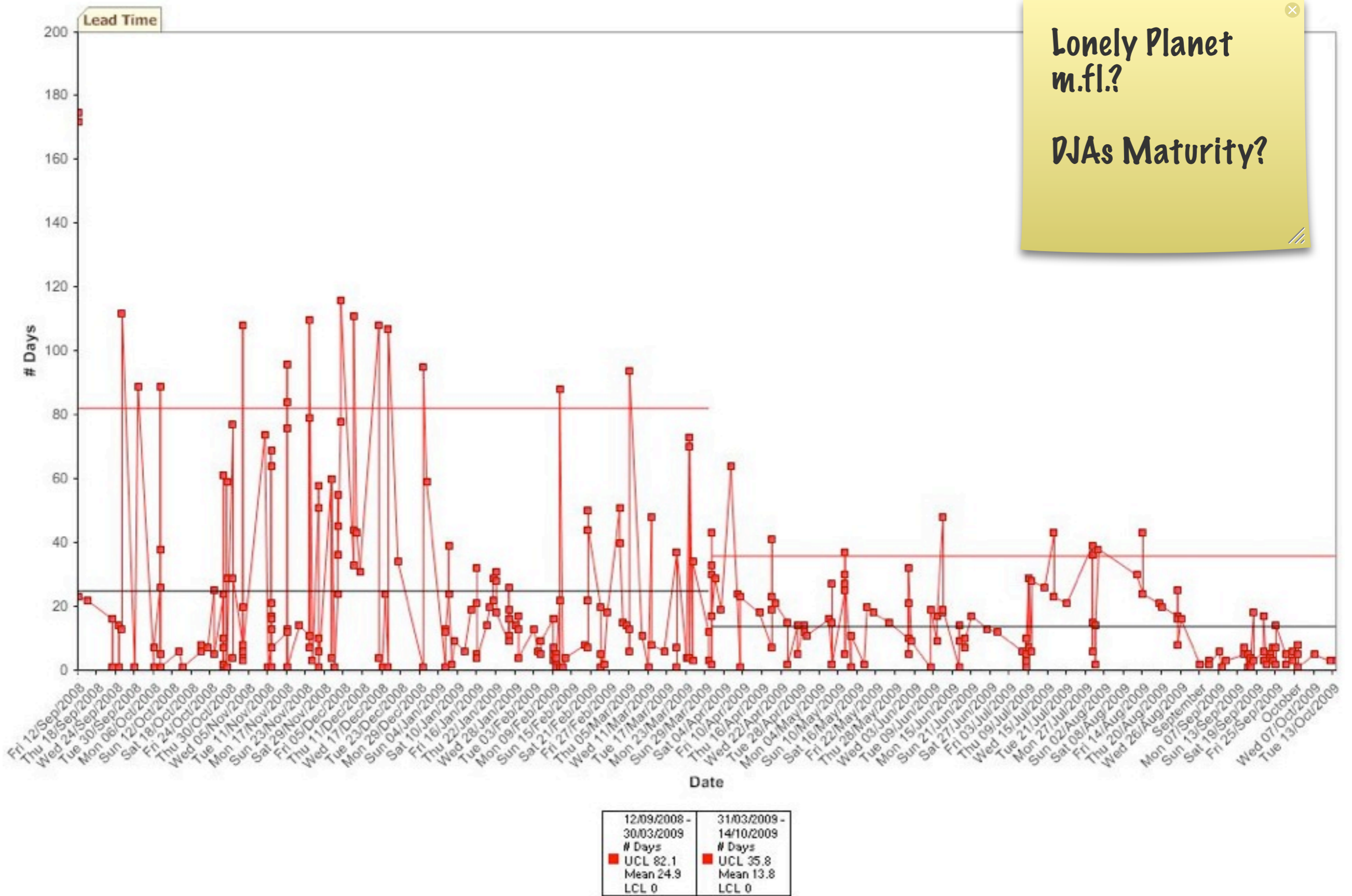




måndag 13 februari 12

En sorts burndown-chart men som kommunicerar mer information, t.ex.

- hur mycket WIP man har
- hur det påverkar ledtider
- flaskhalsar
- etc.



Lonely Planet
m.fl.?

DJAs Maturity?

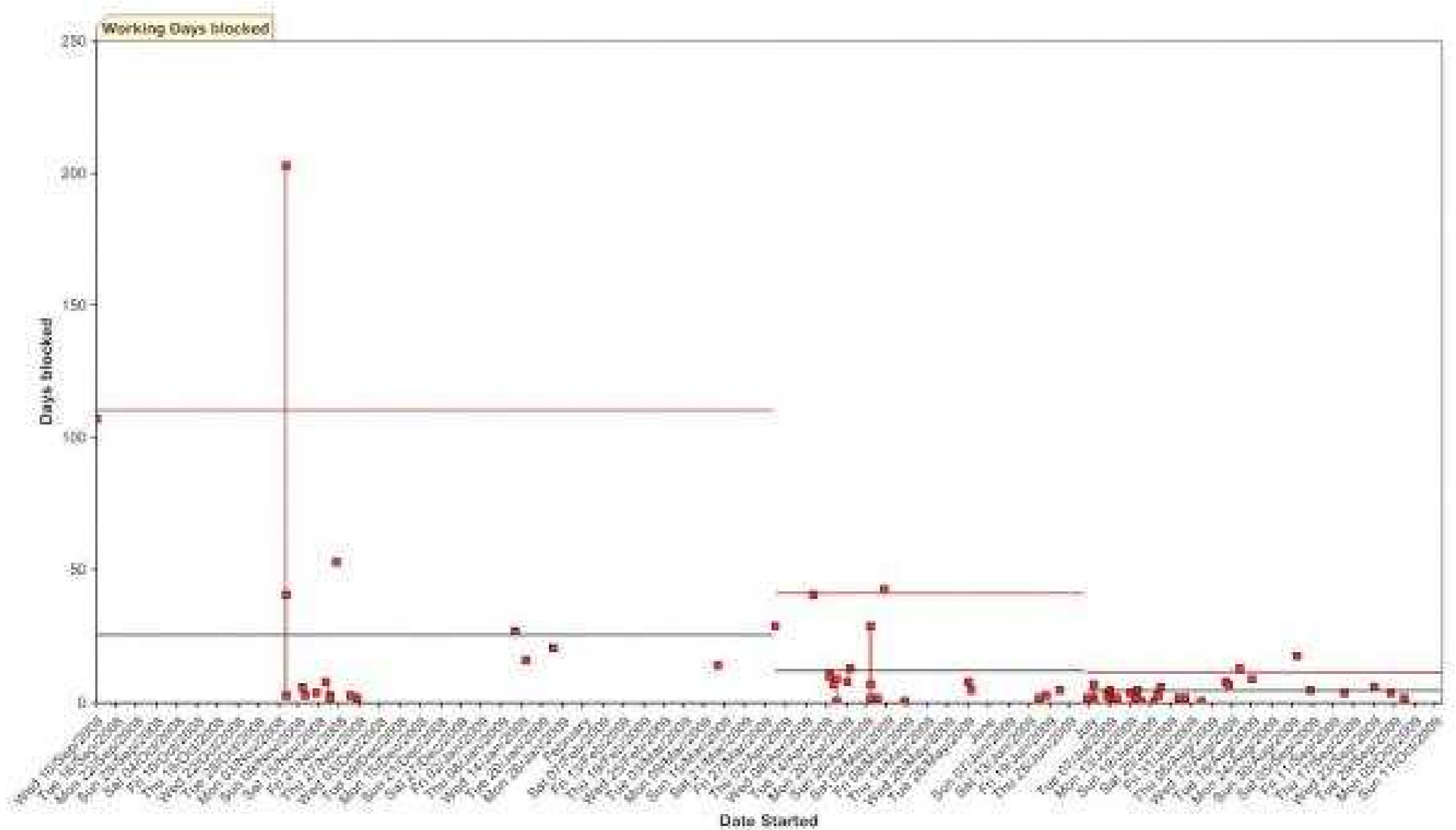
måndag 13 februari 12

Kontroll diagram, eller Statistical Process Control (SPC)

Upper control limit visar vad som tog onormalt lång tid, troligen på grund av särskilda orsaker som bör utsättas för grundorsaksanalys.

Förutom att den illustrerar outliers visar den också hur saker förändras över tid.

Ledtid i ett team som använde Kanban för att förbättra sitt Scrum-arbete. Såväl ledtid som variation minskade.



måndag 13 februari 12

Här ser vi hur man kan mäta issues identified and time to resolve.

Lean looks at 'blockers' or impediments as 1st Class items to be addressed. Lean seeks data to be used by the team for self management and routine process improvement.

The daily stand up is concerned to identify and remove anything that is preventing progress. To do this 'blockers' are actively identified, assigned, tracked, escalated and removed. This is a mechanism for making continuous improvement routine. Evidence of the effectiveness of this is shown below in a statistical process control chart (Fig. 9). The periods on the chart have been split from September 2008 - March 2009; April - June 2009, and from July - October 2009.

Over the 12 months the number of working days items were blocked was reduced by 81% from a mean of 25.8 days to 4.9 days. The outlier in 2008 was a result of waiting for a 3rd party to complete their work (a special cause). This is powerful data to use when discussing performance with 3rd parties.

Actively looking for and recording blockers increased the number of blockers raised which is beneficial. These were then being removed at a faster rate by the team as more experience was gained. This data was also used in Retrospectives and quarterly reviews. Re-occurring blockers were investigated and root-cause analysis performed. The 81% reduction in the number of working days that work was 'blocked' is evidence of effective continual improvement.



The fact that we should set numeric improvement objectives and track their delivery numerically, is powerful; but it is not the main point. The main purpose of quantification is to force us to think deeply, debate, agree, and specify, exactly what we mean; so that others, later, cannot fail to understand us.

Tom Gilb

PLANNING
4

DOING
2

DONE!

BOOK HOTEL
& DO
TRAVEL ARR.

DISCUSSION &
QUESTIONS

INTRODUCTIONS

THE DOT

KANBAN

KANBAN IN
PRACTICE

WRAP UP &
DISCUSSION

PLANNING
4

DOING
2

DONE!

WRAP UP &
DISCUSSION



DISCUSSION &
QUESTIONS

BOOK HOTEL
& DO
TRAVEL ARR.

INTRODUCTIONS

THE DOT

KANBAN

KANBAN IN
PRACTICE

Kanban



måndag 13 februari 12

Before we wrap this up we have to mention something about the problems with Kanban. Of course Kanban has its critics as well. So what are the pitfalls to watch out for?



**Here's
Kanban!**

måndag 13 februari 12

All work and no play makes Jack a dull boy...

Ken Schwaber in a blogpost

No cadence, no celebration – just work, work, work...

– Have cadence for these things if you want to, but it doesn't have to be the same for all.

Scania – no deviations is normal. Don't celebrate that!



måndag 13 februari 12

Timeboxing is good!

- Use timeboxing for each work item against its SLA or average completion time.



måndag 13 februari 12

Sometimes you need a revolution...

Mer om lean/
agile/ToC för
förtåelse?



måndag 13 februari 12

Don't let Kanban become an excuse for not doing any planning at all, not working agile, not focus on good teamwork etc.

Don't slip into not adding WIP-limit for example. That creates the tension we need

Of particular importance when introducing Kanban in a non-agile, non-mature environment.



Marcus!

måndag 13 februari 12

Ok, so now you have visualized your work, limited your WIP, and monitor the flow. So now what? Now the next level begins. Relentless continuous improvement.

Remember that the WIP-limit is something that we have taken upon ourselves in our pursue of excellence.

We're stretching our limits to become even better.

So when everything is working fine – let's lower the limit to become even better. New problems to solve become obvious – solving them leads to a even better process.

Compare this to the suggested approach for team that think that a sprint is too short – shorten it further. 4 weeks -> 2 weeks

marcus

Does It Work?



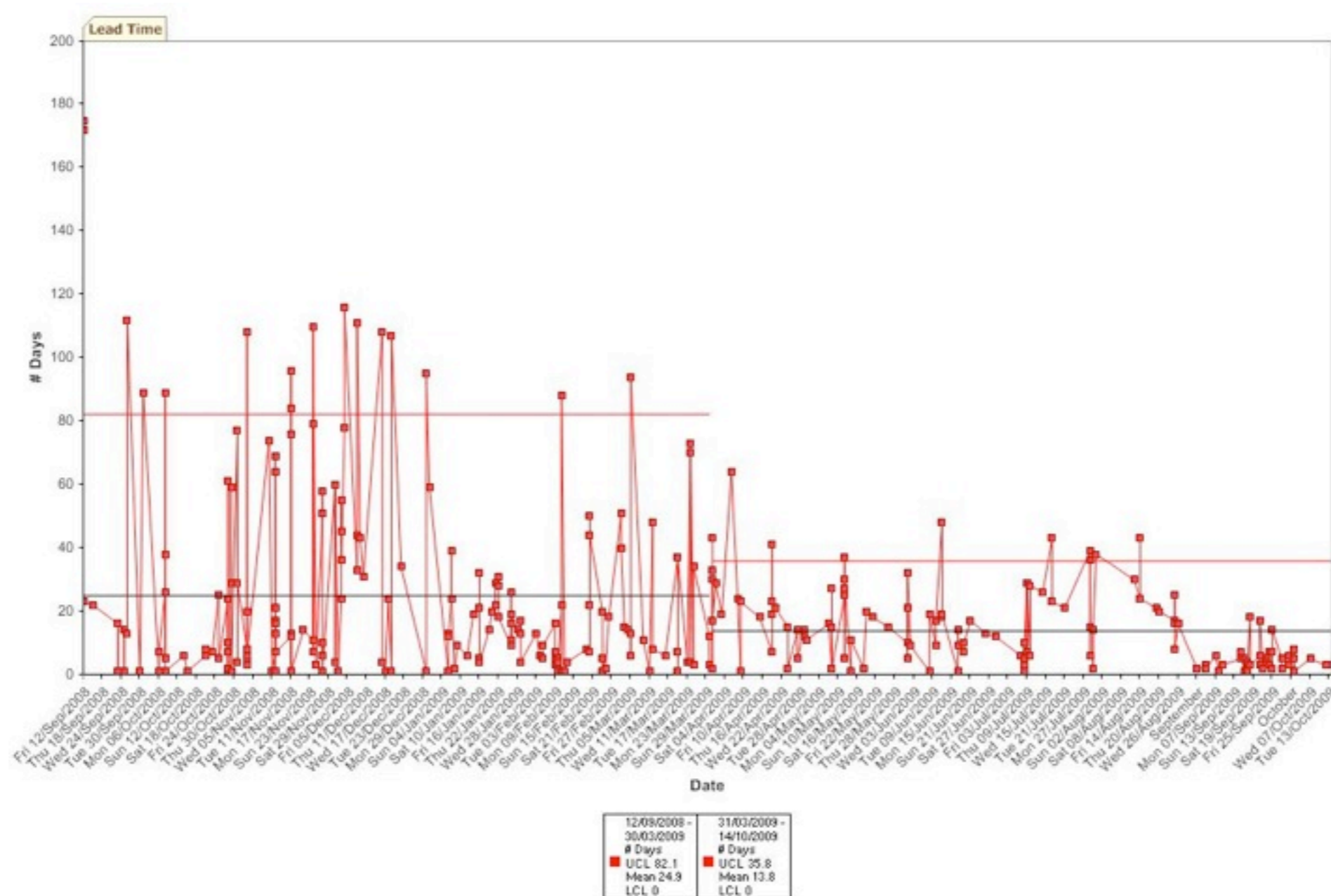
måndag 13 februari 12

Why take my word for it?

This part of a Kanban introduction gets more and more irrelevant as the adoption of Kanban spreads. There is even conferences these days where you can go and listen to lots of people sharing their experiences about how Kanban has worked for them.

I would be surprised if you don't hear about people succeeding with Kanban and flow-based systems later this week.

BBC Worldwide Case Study



David Joyce's blog: <http://leanandkanban.wordpress.com/>

måndag 13 februari 12

Lean Software Management: BBC Worldwide Case Study

<http://leanandkanban.wordpress.com/2010/02/11/bbc-worldwide-kanban-case-study/>

Abstract: This case study examines how the lean ideas behind the Toyota Production System can be applied to software project management. It is a detailed investigation of the performance of a 9 person software development team employed by BBC Worldwide based in London over a 12 month period.

The data was collected by one researcher in 2009. It involved direct observation of the development team, the kanban boards, the daily stand up meetings, semi structured interviews with a wide variety of staff, and statistical analysis.

The evidence shows that over the 12 month period, **lead time to deliver software improved by 37%**, and **consistency of delivery rose by 47%**. Output increased, while **defects reported by customers fell 24%**. The significance of this work is that it shows how to considerably improve the performance and consistency of Agile software development techniques.

The conclusion is that the adoption of lean and kanban ideas have **enabled the maturity of an Agile software process to move rapidly towards CMMI Level 4**. The faster, more responsive delivery has **reduced both technical and market risks**. The **drawbacks** are that **it may not fit well with existing corporate standards** and **managers may find their new role challenging**.

Stop starting Start finishing!

måndag 13 februari 12

So what should you do?

Start tomorrow!

Agree on a simple policy to stop starting, start finishing

Then visualize and study the workflow

Limit WIP and improve lead times

Monitor flow and use leading indicators such as queue utilization and bottlenecks

Use metrics to improve

Start where you are and improve from that



Marcus Hammarberg
Twitter: @marcussoftnet
www.marcussoft.net
marcus.hammarberg@avegagroup.se



Joakim Sundén
Twitter: @joakimsunden
www.joakimsunden.com
sunden@spotify.com

avega group



<http://bit.ly/kanbanjfokus>

måndag 13 februari 12

Here you'll find the presentation

måndag 13 februari 12

Den här är bara här för att visa att det är slut ;)