# Stories from the Flexible Company

**Bas Vodde**
**Aarhus, Denmark**
**September 2007**

## DISCLAIMER!

This presentation is based on my views and my opinion. This is not the view of the company and also others have experienced the same time in a different way :)

## Stories

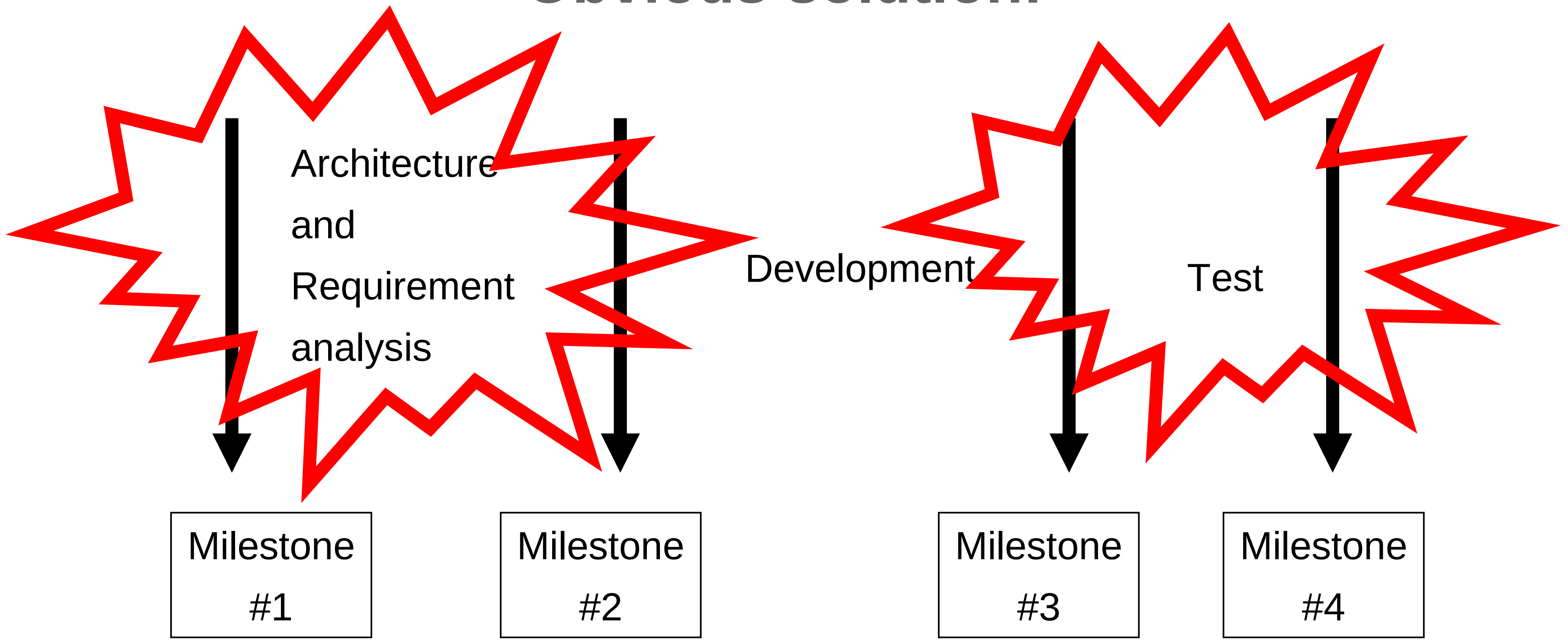
- Creation story
- First months
- Flexible team
- Scrum
- Clearcase and other ill weeds
- Becoming Flexible Company
- CMMi
- Failures
- Teaching TDD
- Metrics
- The dead Scrum Master

## Creation story

## Problems in product development

- Many faults found in testing… and by customer.
- Most important features are not implemented in time.

## Obvious solution!



## Maybe not?

**What if we would develop more efficiently and with a better quality?**

## Result:

Flexible R&D

Make R&D more flexible.

By introducing,

Agile and iterative development

To Nokia Networks

## First months

## The search

Where in Networks did people experimented with Agile?

# The Nokia Test (called so by Jeff)

**You know when you are not doing iterative development when:**

- Iterations are longer than 2-4 weeks.
- Team tries to complete specification before programming.
- An iteration does not include testing.
- Iteration does not produce workable code.
- Detailed (task level) plan are accurate estimates are expected at the beginning of a project.
- The sprint plan doesn't reflect what the team is doing.

**You know when you are not doing agile development when:**

- There is little co-operation within the team.
- Design and code is produced in individual effort.
- Progress is measures by hours spend or documents created instead of working code.
- Builds are done once every three weeks.

## Suddenly

## Share information, be open!

Flexible R&D blog

AgileNet Blog

Flexible R&D programs list

T-Shirt

Newsgroups

Scrum Master mailing list

Reports & presentations

Newsletter

Product Owner Mailing list

Agile Wiki

AgileNokia wiki

Open Space Gathering

Continuous Integration Wiki

## What do we do?

Give and arrange training

Facilitation

Organize Gatherings

Create Communities

Coaching

Project Support

## Support, NOT control

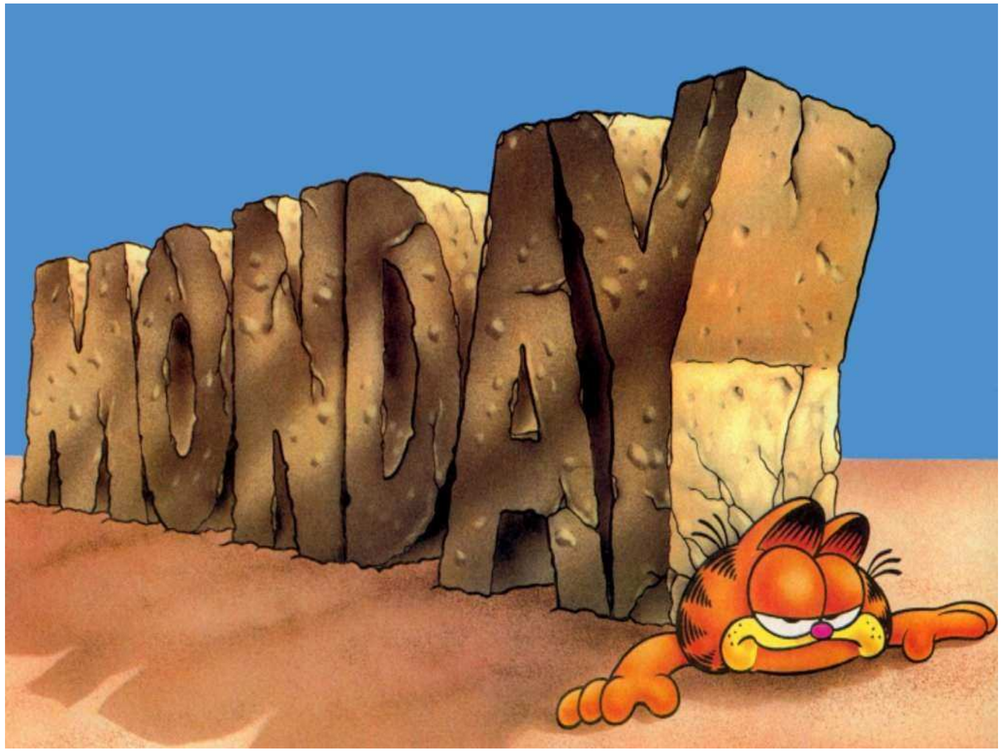
## Flexible Team

## No evangelists, cross functional team!

- Cmm assessor, process architect.
- Test specialist, TPI assessor
- Developer
- Requirement process expert
- Project advisor
- Quality & process expert

And… 2 agile evangelists.

## Monday Morning



## Result:

Agile evangelists…

## Scrum

Not selected! Emerged

Simple

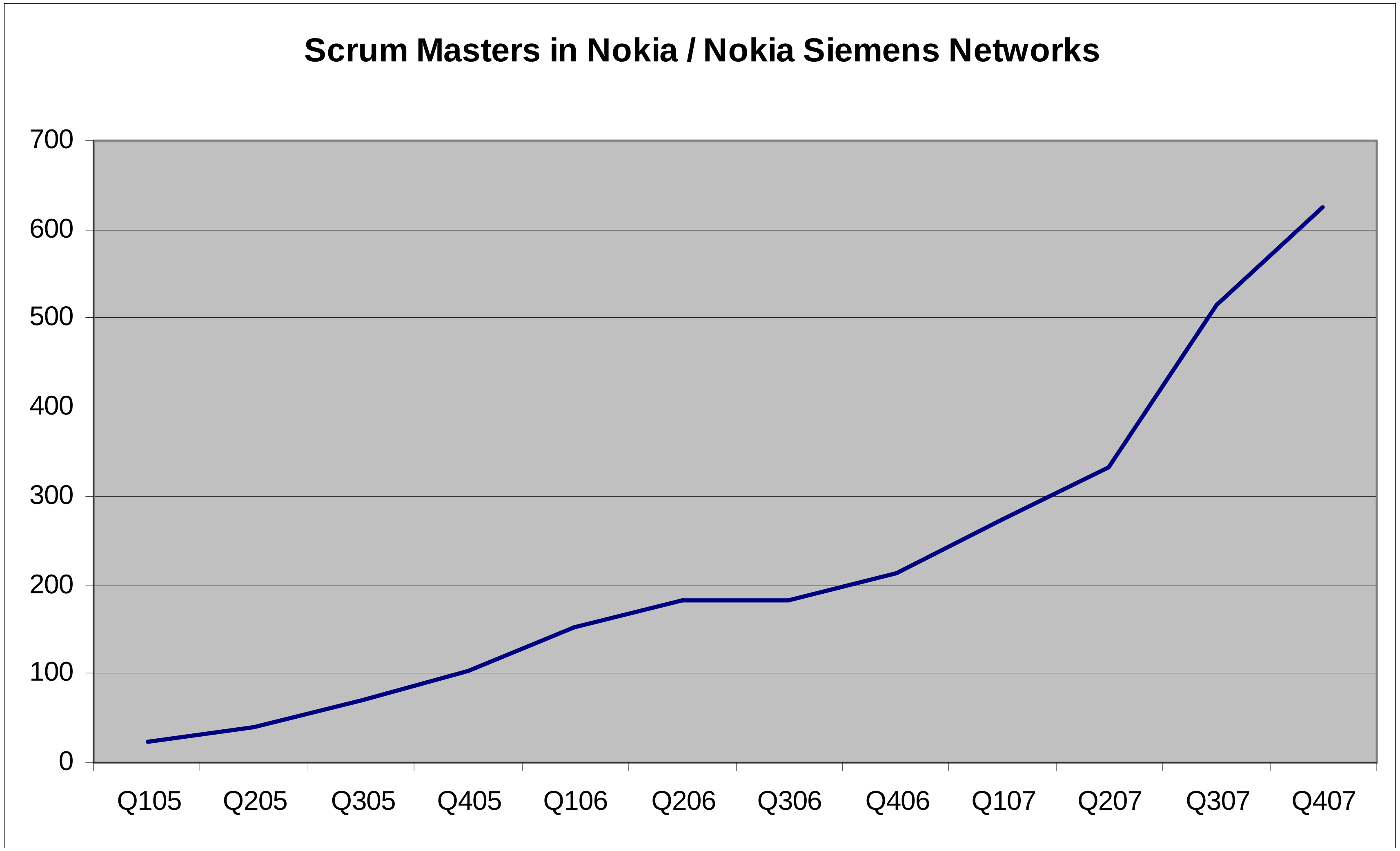
A good start

Focus on management (?)

Craig & Ken

Good training available

## Scrum Masters



## Clearcase and other ill weeds

## Individuals and interaction over process and tools

But some tools are always trouble.

## Continuous integration

“1% of my time goes in setting up continuous integration. 99% goes into getting it working with clearcase”

“Luntbuild is nice and works fine, but not with clearcase”

“Clearcase is not made for this style of development. At least don’t use dynamic views”

## Get rid of it!

- Cost-benefit analysis clearcase vs subversion.

- Users survey.

- Support in transition.

- Other examples.

But… “Ill weeks grow apace”.

## Becoming the Flexible Company

## “Making R&D more flexible”

How about that for a sub-optimization?

How is a Flexible R&D going to be useful for our customers?

What about Flexible Product Management
or Flexible delivery, or…?

## Flexible R&D -> Flexible Company

The goal of Flexible (Agile) software development must always be to work more flexible with the customers.

That’s where the ultimate benefit for the company comes. That’s how agile development can grow.

## CMMi

## Finding the roots!

Cmmi ->

Roots in Quality management, Deming, Juran

Agile ->

Roots in Lean… based on Deming, Juran, Ishikawa, Ohno

Same roots.

We should be able to agree, right?

## BUT!

What are the assumptions behind CMMi?

## Cmmi: don’t

## Failures

## Easier to ruin scrum than to make it good.

## After one release, switch to new management.

**Strong super project management.**

**Developers jailed.**

## Teaching TDD

## Lecture in TDD

50 people at 9:00

5 people at 17:00

## Change -> TDD needs to be *done*, not told.

## Only way to teach Test-Driven-Development

Get a coach. Let him sit beside you for weeks.

## Fault finding profile and other useless metrics.

## Fault finding profile

Percentage of faults found in testing phas^H^H^H^H level.

Goal: Measure development and improve early defect finding.

## What if test failed does not mean fault found?

Was that an assumption behind the metric?

## Do we report defects from the CI system?

Assumption of:

Fault Finding Profile

## How do we estimate the total tests when test-driving?

Assumption of:

Testing progress

## Conclusion:

## When using Scrum: All metrics must change.

Especially metrics tied to performance evaluation :)

## The dead scrum-master

## Push the organization

Don’t push too hard, or you’ll end up in China.