Mind over Matter

- problem first

Agile Estonia 2010

2010-05-17

Mattias Skarin

crisp

Who am I?

Coach

- Help IT companies stay competitive

Entrepreneur
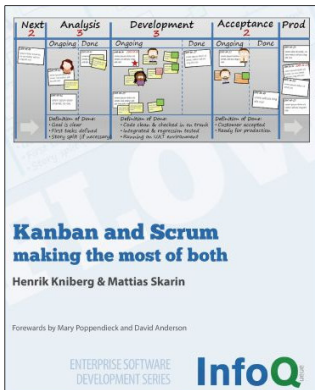
- Extreme Solutions
- Bizplant

Engineer

- 9 years software dev
- Acting CTO

Book: "Kanban and Scrum – making the most of both"

mattias.skarin@crisp.se
+46 733 66 88 08

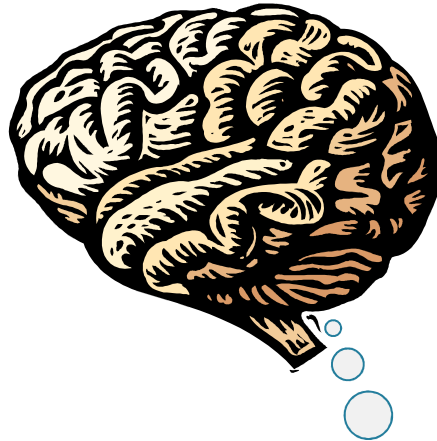crisp

Tool "tribes"

Subversion rules!

Dude - it no way near as wicked as GIT

OMG! That guy does not know how to use GIT! So lame..

crisp

The magical multi tool

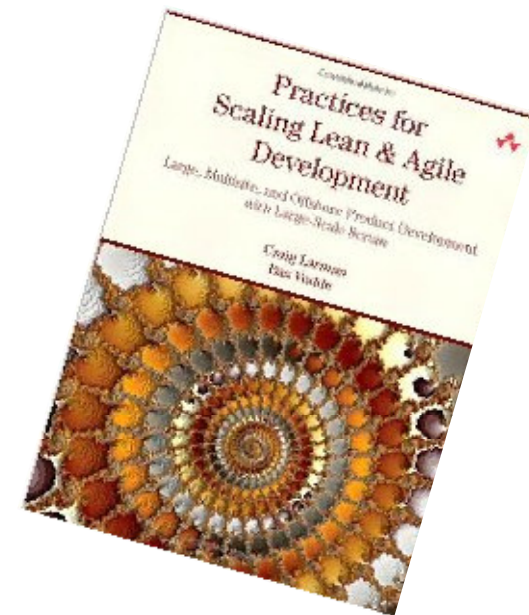"It will make your problems go away!
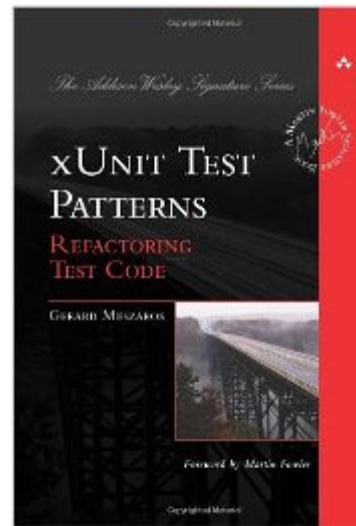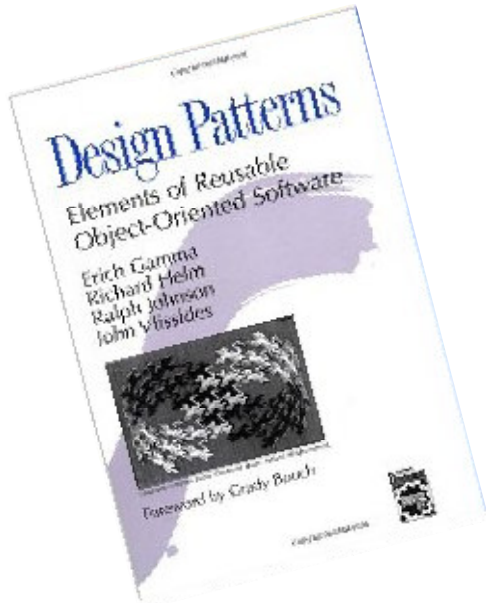
!?

[Developer in trouble]

Big ball of mud

Paper clip

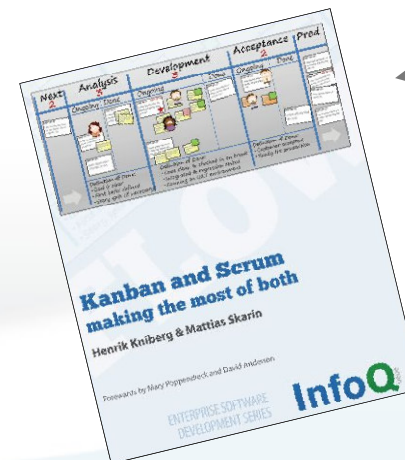# A solution for everything

..exept for solving problems

YEAH - MY BOOK TOO :)
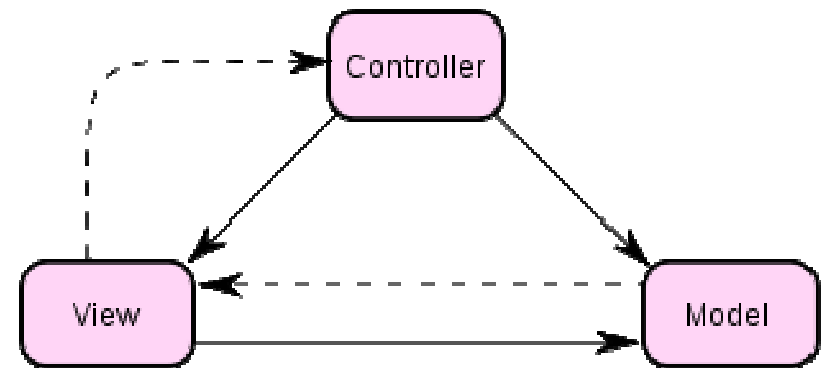
Agile toolbox

JIT   Rup

A3   Team

NoSql   User stories

Continuous integration

XP

TDD   BDD

MMF   Maven

Scrum

AppEngine

Ant

Kanban
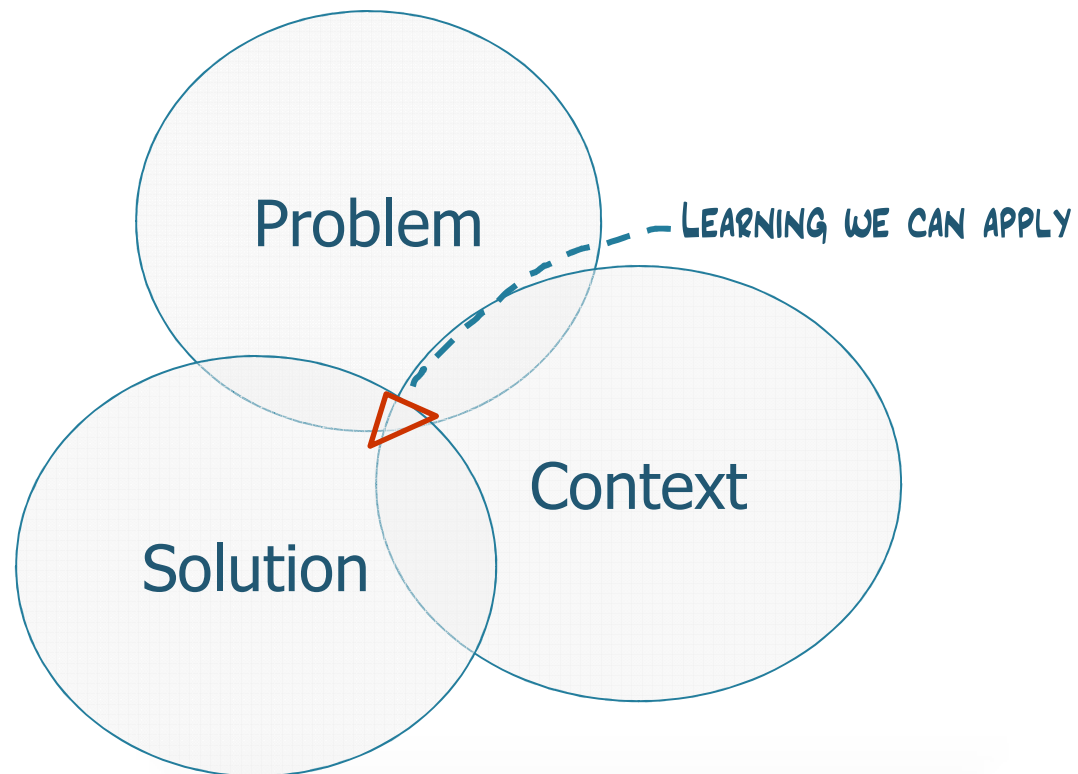
GIT

Six sigma

WIP

One piece flow

- are we toolheads?

crisp

How much better are we?

Every (bad) tool started off with good intentions

Problem
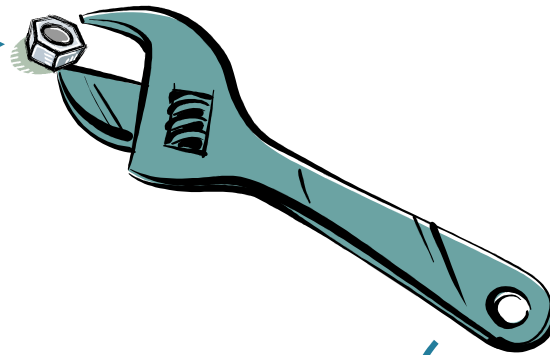
Solution
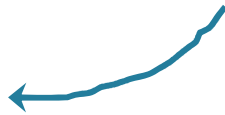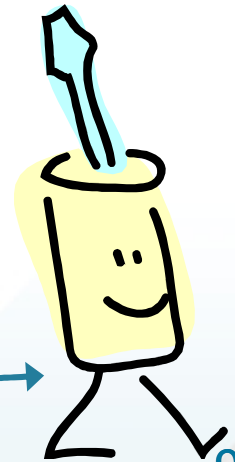
Context

— Learning we can apply

Every tool needs...

A PURPOSE

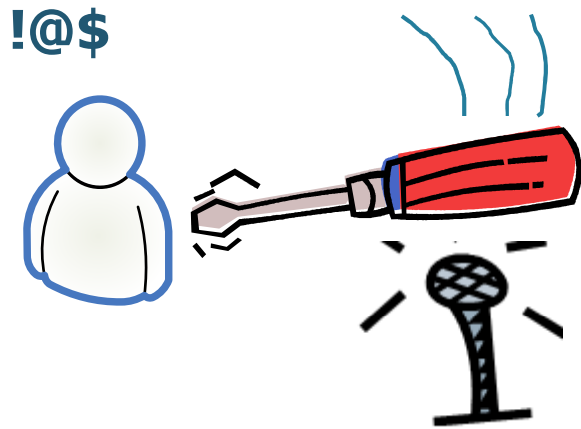DIRECTION

Are you using the tool or is it using you?

Without it, a tool has limits

!@$

..well..almost.. ☺

"Tool first" improvement approach

The better option

Know the problem, then you run the tools

Let's pause for a minute

Let's talk product development

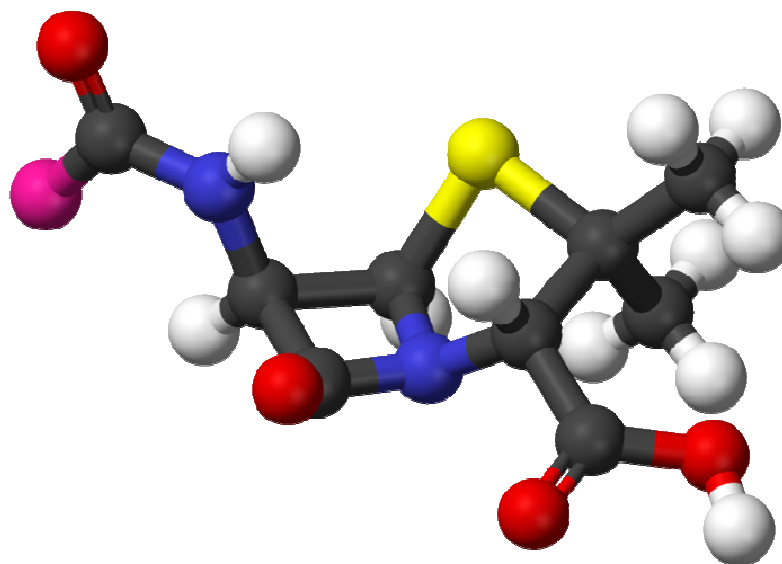| What to do |
|---|

How to do it

Continuous discovery

Customer
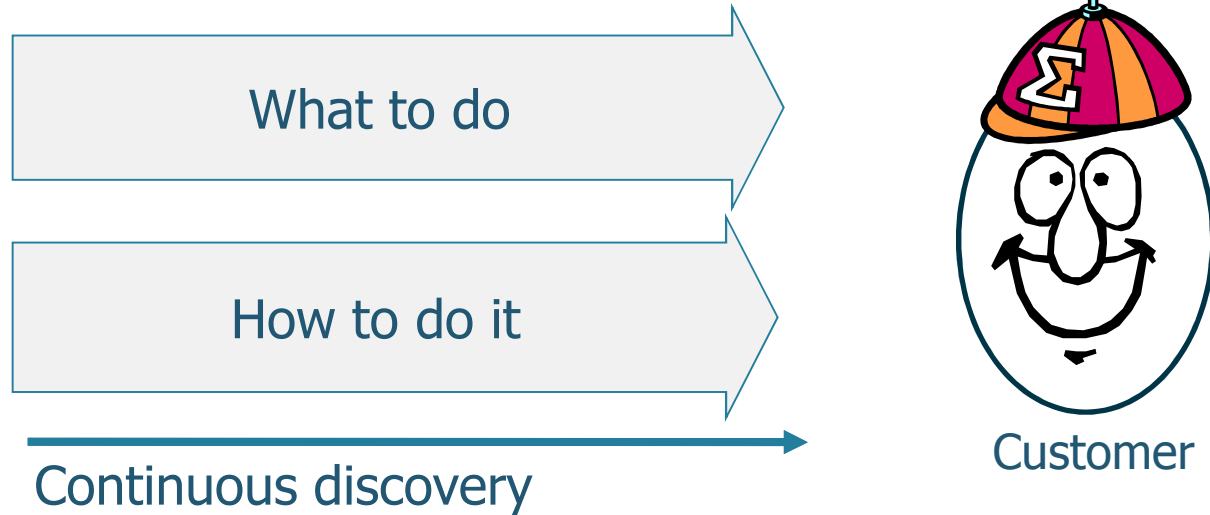
"We can't drive out all variation without also driving out all value added"
 - Don Reinertsen

New knowledge arrives through experimentation

How do we move from

Experimental chaos

..to evolutionary improvement?
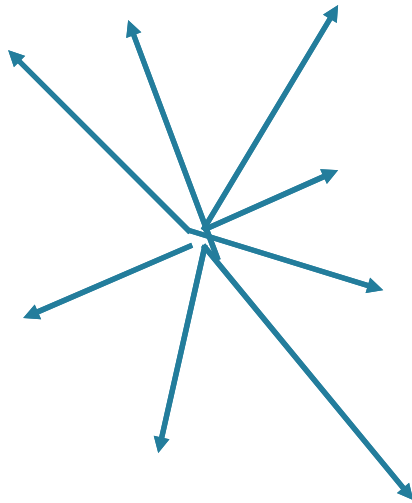
"My developers are goldplating"

[Business guy]

For product development

THINKING TOOLS

crisp

The Kanban Formula

$$Y = [D\ (Tw+Tp)\ (1+\textcolor{red}{a})]\ /\ A$$

[Open ended solutions]

Understand "why"

Improvement

[Purpose and direction]

Know "why" = baseline

"I don't have time"

"What is next step that would make this better?"

- over-

"I wish for the big refactor in the sky"
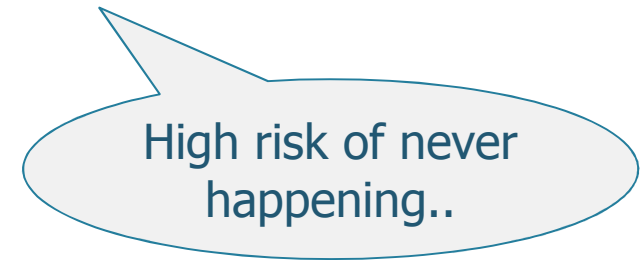
High risk of never happening..

Don't copy - improve

crisp

There are always more than one solution to every problem

Which ones have you explored?

Undefined problem
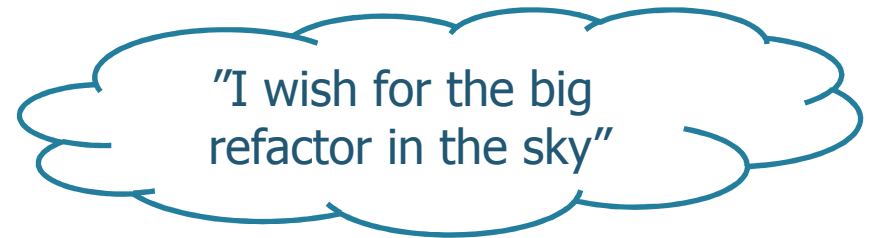

.. **big wrench**
.. **big cost**
.. **big threshold**

Defined problem
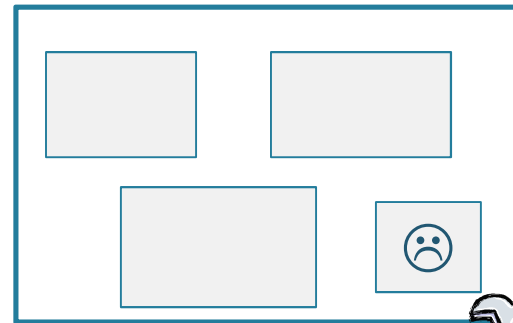
.. **small wrench**
.. **small cost**
.. **small threshold**

"Understanding the problem pays off
with smaller, light weight solutions"

crisp

Problem solving algo

Write it down

1. **Surface problem**   "this looks odd"
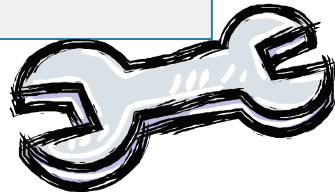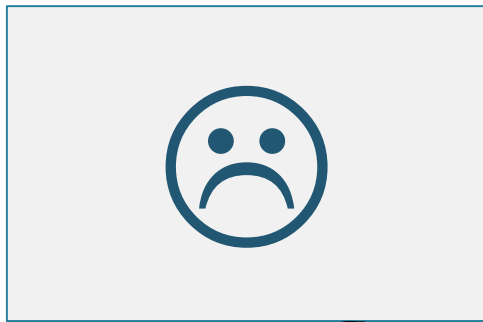2. **Understand problem**   "what, where, when, how"
3. **Idea generation**   "always two solutions"
4. **Try it out**   "there is much we don't know"
5. **Regenerate**   "no idea is perfect first time"

"As in everything else, however, regardless of good intentions, an idea does not always evolve in the direction hoped for by its creator." (Ohno 1988)

Let's try it out :)

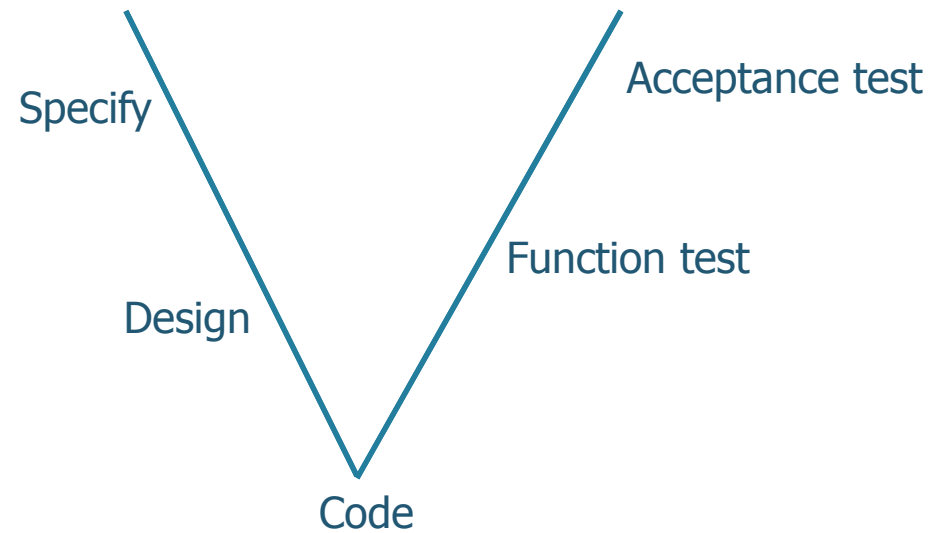|	
Surface problem	Standardized work does not apply to software! It's a different animal!
Understand problem	"We want improvements to start from a known baseline"
Idea generation	"What if we visualized work before improving? And improvements are made by the people closest to the work"
Try it out	People close to the work don't have time
Regererate	Let's move one inch forward at a time. At same time let's begin to plan in slack.

THE IMPORTANCE OF EXPERIMENTING

crisp

Visiting Toyota 2009

Specify

Design

Code

Function test

Acceptance test

The balance act

"Understand"

"Curosity"

-How does that apply..

- What if.."

Team culture
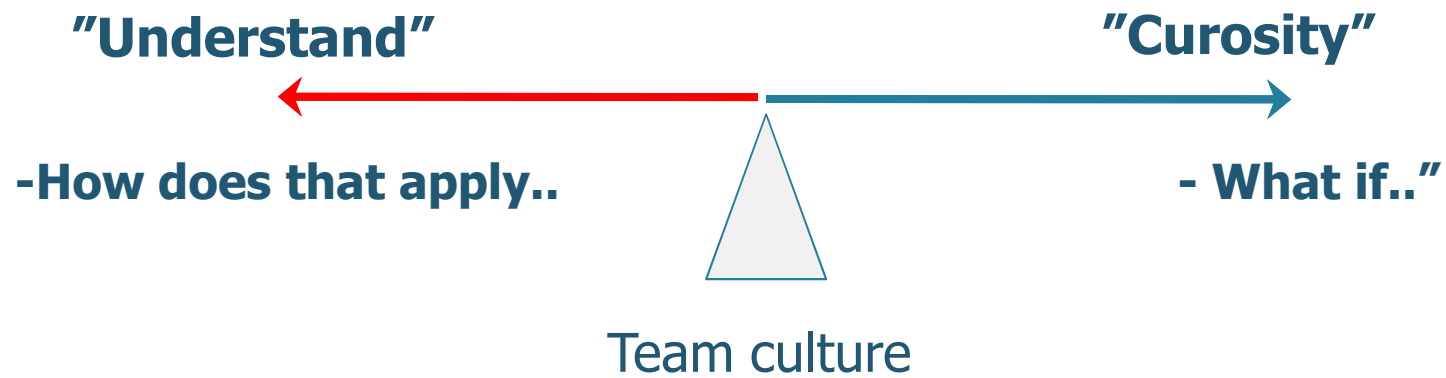
crisp

Summary

- **Through experimentation, we learn**

- **Share using problem, context and solution**

- **Start with the problem (not the tool)**

crisp

We will never arrive

Thanks for listening!

mattias.skarin@crisp.se

Blog:
http://blog.crisp.se/mattiasskarin