If you are ambitious, you have to be lazy - a strategy for good developing - the sequel

By Curt Carlsson

NENCA

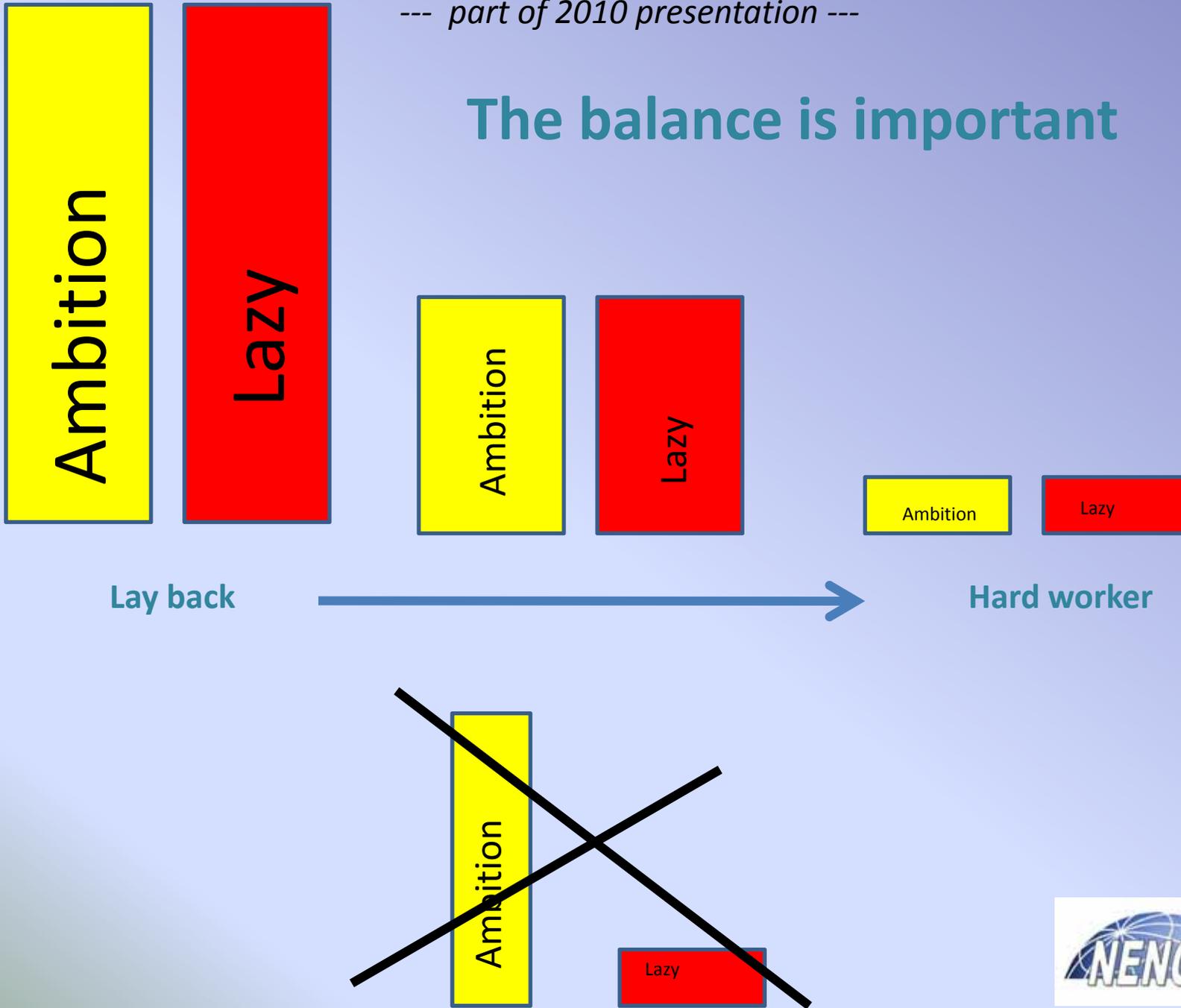**On the last Scandinavian FME User Conference 2010 in Malmö
I introduced this thesis:**

--- part of 2010 presentation ---

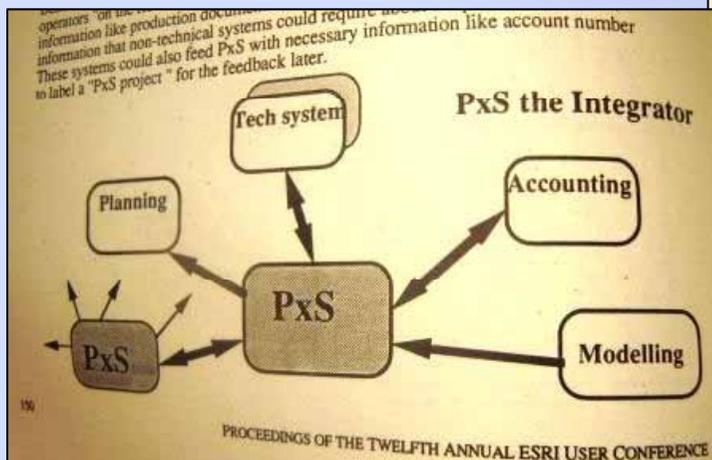**If you are ambitious, you have to be lazy - a strategy for good developing**
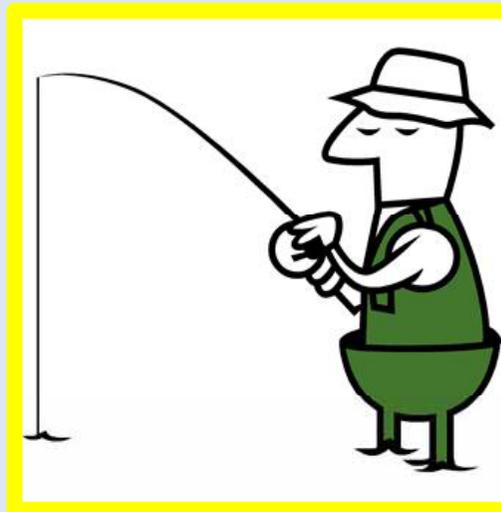
The balance is important

Ambition

Lazy

Ambition

Lazy

Ambition

Lazy

Lay back ⟶ **Hard worker**
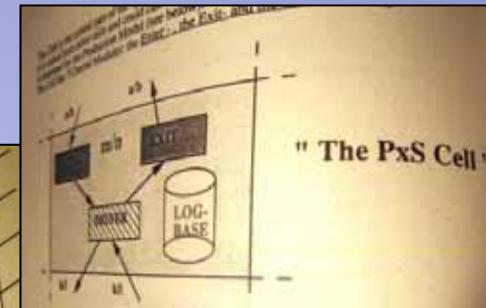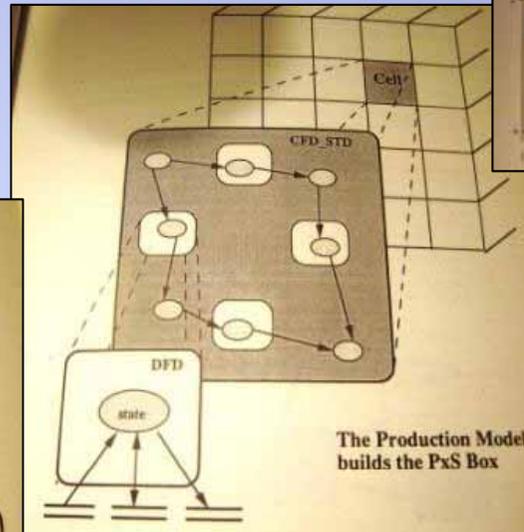
Ambition

Lazy

NENCA

Digital Chart of the World – a good example

In 1990 **ESRI –California** got a BIG order to create the first digital global map database. Scale 1:1 000 000

"The PxS Cell"

PxS the Integrator

The Production Model builds the PxS Box

PROCEEDINGS OF THE TWELFTH ANNUAL ESRI USER CONFERENCE

I was sign up to create a production management system for that project.

Picked up some design ideas from the 1970´s and added that to methods of the 1990's Worked….but…
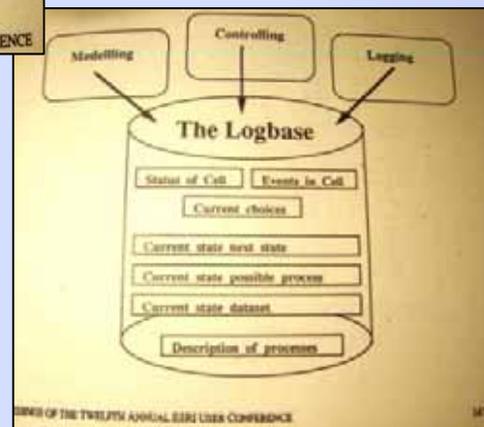
It was just <u>too much of work</u> to implement (puh) so…
I redesigned it and got a 5 times more generic code but 5 times less lines to key in!
But still too much…
redesigned…got more generic…

The Logbase

--- *part of 2010 presentation* ---

NENCA

A lazy person with ambitions get a lot easier issue then it comes to:
-Maintenance
-Extend the functionality
-Rewriting
-Documentation
-Use the system/Support the user

But have this strategy any relevance in the FME case?

Yes, indeed!

With all the 250+ formats, the standard transformers, your homemade transformers, the levels, dynamic …. You name it. <u>You are in for a careful selection of the possibilities to solve a problem.</u> - *BE LAZY!*

According to my thesis before I am sure you can find out what kind of label I will set on these guys.
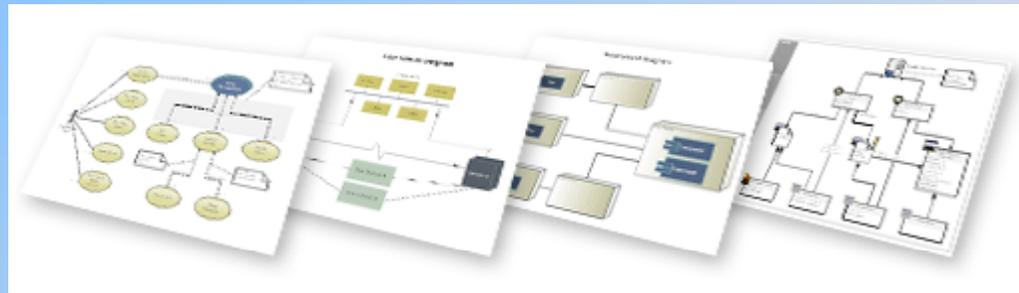
They must be the most lazy guys in the world!

--- end of 2010 presentation ---

Something more to think of for Ambitious Lazy people?

1

- Bye your software

- Ask somebody if he/she has something you can use

-- Have a method (Eg. UML) with supporting software in your pocket, just in case you are in for a bigger system design and implementation

NENCA

BATCH

This sounds old fashion but is very easy, safe and efficient.

Batch 2012??

In a world of clouds, desktop style, iPADs, Internet, on-line-society, etc ??

YES! Think Batch!

NENCA

Mainframes like Control Datas computers had an excellent logg outprinted, IF any error accured.
On which record (punch card) and position within the record the program stopped, and the whole calling chain back up to start....

Routine A

Subroutine C

Subroutine F

```
c       This program finds the divisors of an integer input by the user.
c       The divisors are printed to a file.
        read *, n
        write (1,*) "Here are the divisors of ", n, " :"
        k = 0
        do i = 1, n
```

```
c       This program finds the divisors of an integer input by the user.
        The divisors are printed to a file.
        integer n, k, d(10)
        open (unit = 1, file = "divisors")
        print *, "Enter a positive integer :"
        read *, n     Line 6 in subroutine F
        write (1,*) "Here are the divisors of ", n, " :"
        k = 0
        do i = 1, n
            if (mod(n,i) .eq. 0) then
                k = k + 1
                d(k) = i
            end if
            if (k .eq. 10) then
                write (1,5) (d(j), j = 1, 10)
                k = 0
            end if
        end do
        write (1,5) (d(j), j = 1, k)
5       format (10i7)
        close (1)
```
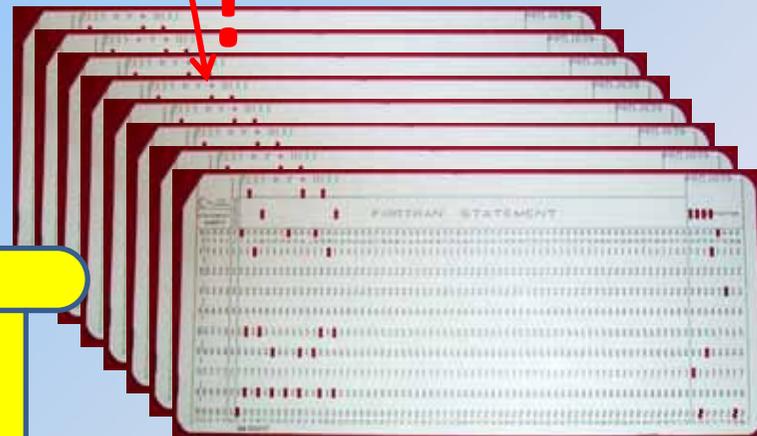
?

!

5th record, 20th column

Logging
Line 6 in subroutine F stopped on 5th record , 20th column, called from subroutine C which was called from Routine A

Once upon time...

NENCA

Sweden's 3:rd National Precision Leveling 1979 - 2004
25 years of work by cars - 100 000 km

Think Batch!

They had a field computer and *were not allowed
to move before they had a measuring within the
error tolerance for each stop.* No way around!

NENCA

FME is a pure batch processing program,

but with a dynamic possibility to introduce parameters during the execution.

Real time applications.. *Think Nano Batch!*

When the "interactive dialog" with the
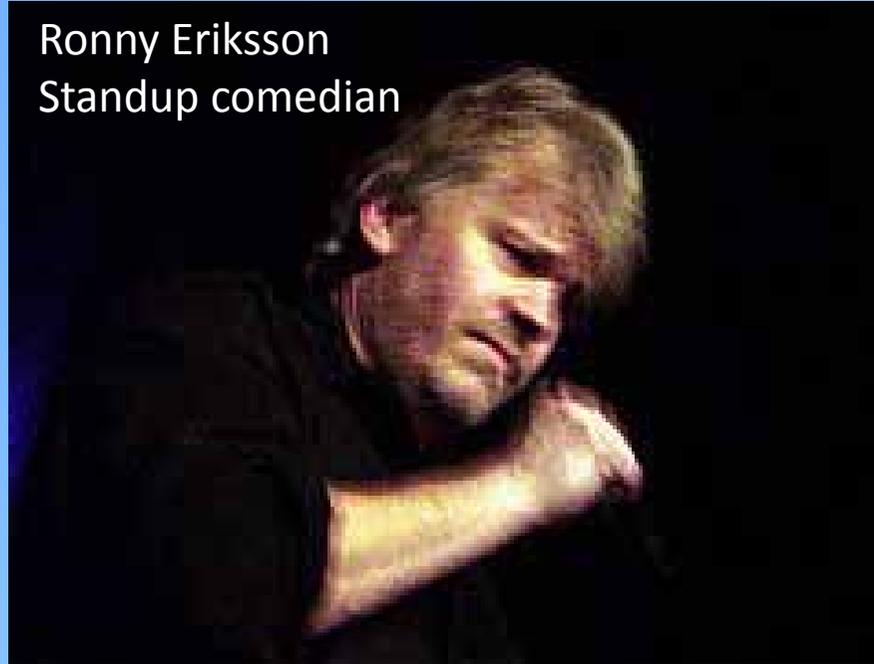computer started in the 1970's the
whole logging looked like that:

/ Run MyProgram
/Error!
/ *** Data mishmash ***
/

And most of the desktop error
"analyses" are of the same quality,
still…

3

Ronny Eriksson
Standup comedian

"It is never too late to give up"

The best for some system is to never be released!

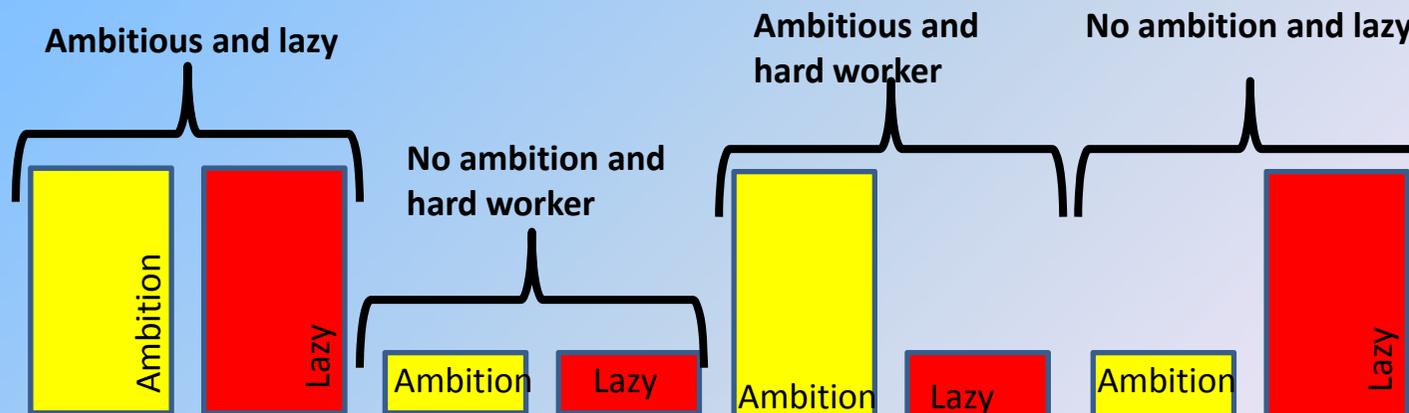# IF you are ambitious, you have to be lazy

....

ELSE ?

We are different. Some people can't just be lazy and some want to be given the design to implement.

A "fundamental management law" is that is should be a composite of different personalities in a team for an optimum work together.

If a team is too homogeneous it is NOT good.

All type of personalities are needed for a good job

Ambitious and lazy

No ambition and hard worker

Ambitious and hard worker

No ambition and lazy

Ambition

Lazy

Ambition

Lazy

Ambition

Lazy

Ambition

Lazy

NENCA

Thanks for your attention!