From Waterfall to Agile

Norwegian Labour and Welfare Administration (NAV)
Torstein Skarra
Test Centre of Excellence at NAV

NAV 3 years ago

19 000 ++ users internally
Millions of external users

1/3 of national budget

500 + employees in IT
2-300 consultants

3-400 systems
X no of platforms

PLAN-BUILD-RUN
Waterfall

Development based on projects and suppliers

NAVs mission is to execute governmental laws and directives

- Not an IT company
- Flexible capacity
- Play the market

Every system had a contract. Build and bugfix at fixed prices

“Clever” contracts destroy agile

Complex rules – stupid behaviour
Simple rules – intelligent behaviour

What this meant for Test:

- Our job was at the end. A massive AT to control and verify
- Tester role was to detect errors

Learning and success factors

- Unique systems ➔ build them yourself
- Hire and build competence
- Strengthen community
- Avoid fixed price contracts

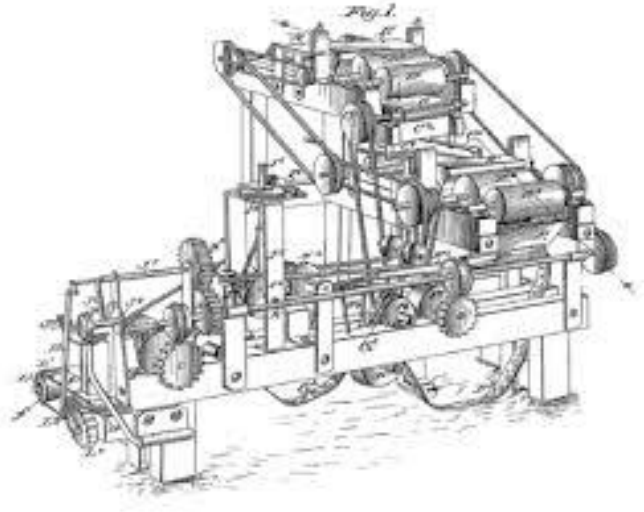
Truths to be killed:

~~*NAV is not an IT company*~~

~~*Testing is all about detecting bugs*~~

We buy code and work in projects

- “Code is like a machine. You buy it and you run it forever...”
- Planning, coordination and testing handles complexity



What this meant for Test:

- Test managers was more like a PM
- Expertise in planning & coordination changes our focus

How projects destroy agile

- Projects optimize on the *project*
- Projects are short sighted entities
- Projects focus on activities
- Projects isolate

What this meant for Test:

- Projects focus on roles, responsibilities and phases
  - Thus the roles in test became highly defined and rigid
  - It took the “thinking out of testing”
- Projects kill our integrity

Learning and success factors

- The goal is not to handle complexity, but to reduce it
  - “Two Pizza team”
  - Long term ownership
  - Broader skill set for testers

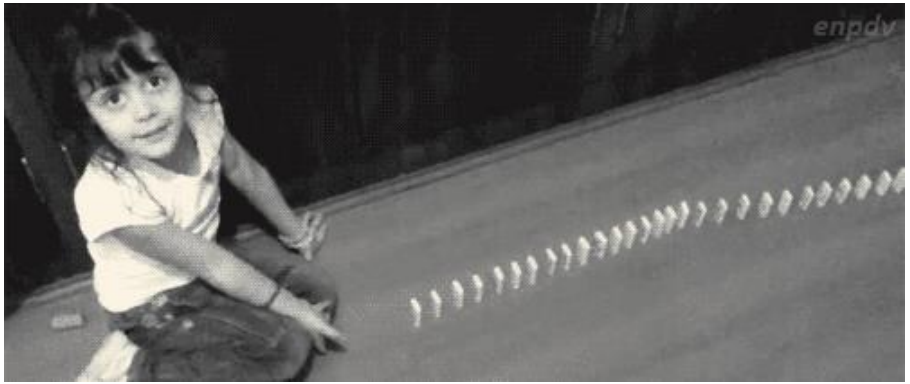
Truths to be killed:

~~*Projects works for SW delivery*~~

~~*Remedy for complexity is planning, coordination and more testing*~~

AT and RT is vital for quality and risk reduction

- Extensive AT and RT at end
- High cost accepted
- Economy of scale says



What this meant for Test and Test managers:

- ”Guardians of quality” and “all mighty toll gate”
- Test-methodology with roles, responsibilities and process
- Central unified function

R = C * P

- Risk = Consequence * Probability
- Took us 2 years to sell this logic
- Differentiate and distribute

What this meant for Test:

- Quality becomes everybody's concern
- We lost our role as *guardians from errors and risk*
- Trust testing performed by others (even automated tests)

Learning and success factors

- Think R = C * P
- Shared responsibility for quality
- Differentiate and distribute quality work

Truths to be killed:

~~*The target must be zero errors*~~

~~*Someone must be responsible for Q*~~

Handovers ensures quality

- Release to P is risky and costly
- Formal handovers are good
- Separation of roles

What this meant for Test:

- Test represented a third party, a neutral quality worker

Creating customer value every day

- Connect with the users
- From building it right to building the right thing
- Continues for fast learning
- Formal handover are “waste”

What this meant for Test:

- Integrate with the team
- Shift left
- Shift right
- Continuous testing

Learning and success factors

- “I’m a test person” → “I’m a quality person”
- Avoid internal handovers
  - also within the team
- Shift left and right

Truths to be killed:

~~*Testing in itself ensures quality*~~

~~*Handover is a quality measure*~~

I need my own stable test environment!

- NAV had ~30
- Compatibility
- 70% of errors were false falses

What this meant for Test:

- Very competent in data and environment handling
- Compensated for test-dependencies and complexity

More or fewer test environments?

- Enforce the reduction
- Pain driven development
  - Fewer code branches
  - Stable code branches
  - Smaller releases

What this meant for Test:

- 70% less waste in testing

Learning and success factors

- Reduce number of test environments
  - Team owns the problem
  - Support with skills and resources

Truths to be killed:

~~*Number of test environments is a constraint in SW development*~~

Legal contract: a substitute for trust and good faith?

- Legal contract is a game of rules
- Defined roles and routines
- Checklists and reporting
- Win or loose – us & them

What this meant for Test:

- Test became agents of distrust

Trust is essential

- Trust is essential to achieve high performance
- Identify and remove distrust-functions
- Praise mistakes

Truths to be killed:

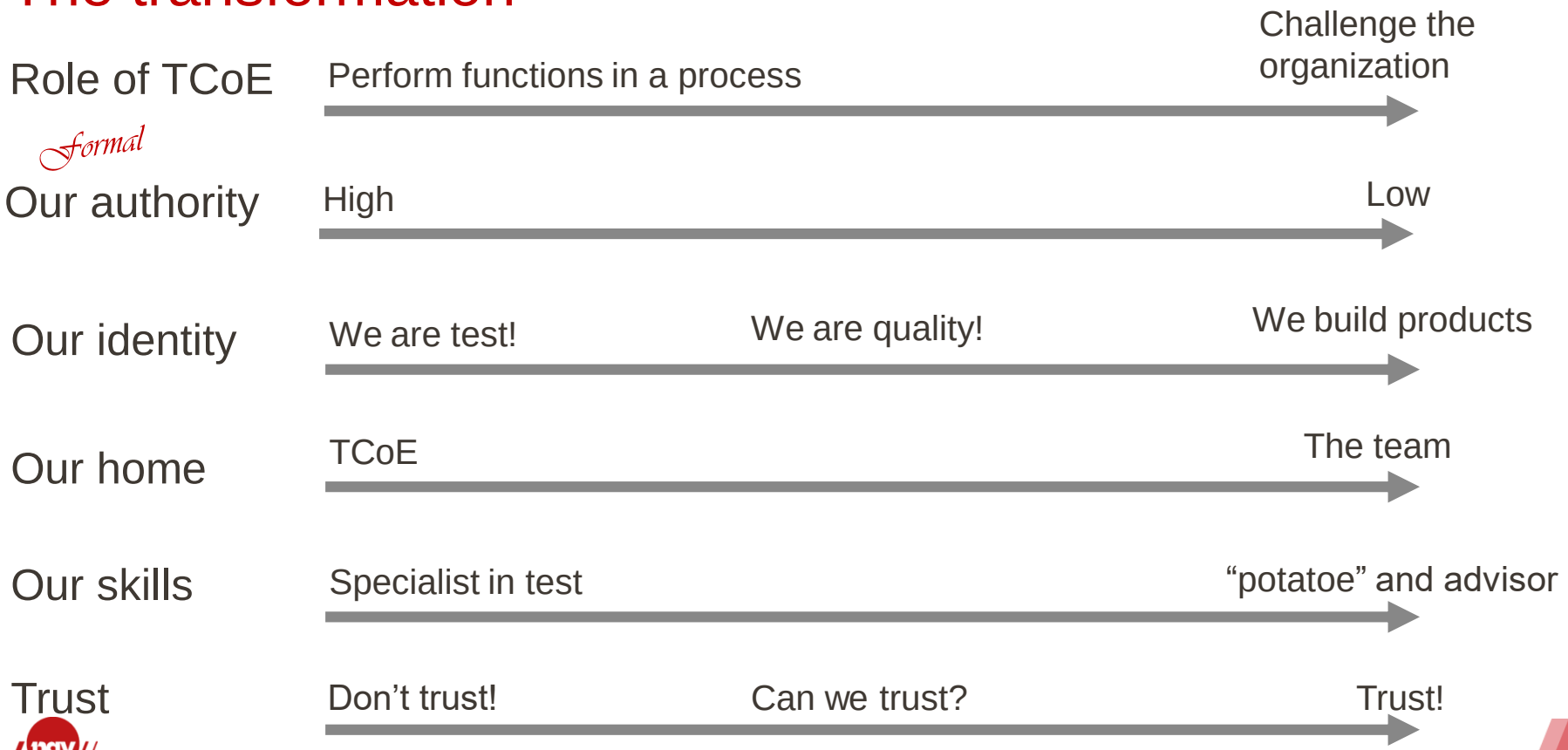
~~*You cannot trust a contractor*~~

~~*You cannot trust a programmer*~~

Lessons learned at NAV

- Unique services & systems = You're an IT company
- SW projects kill agility
- Release small
- And continuously
- Few test environments
- Trust

The transformation



sli.do
#testcon2019

Torstein Skarra
LinkedIn torsteinskarra