

“Sick of Scrum”

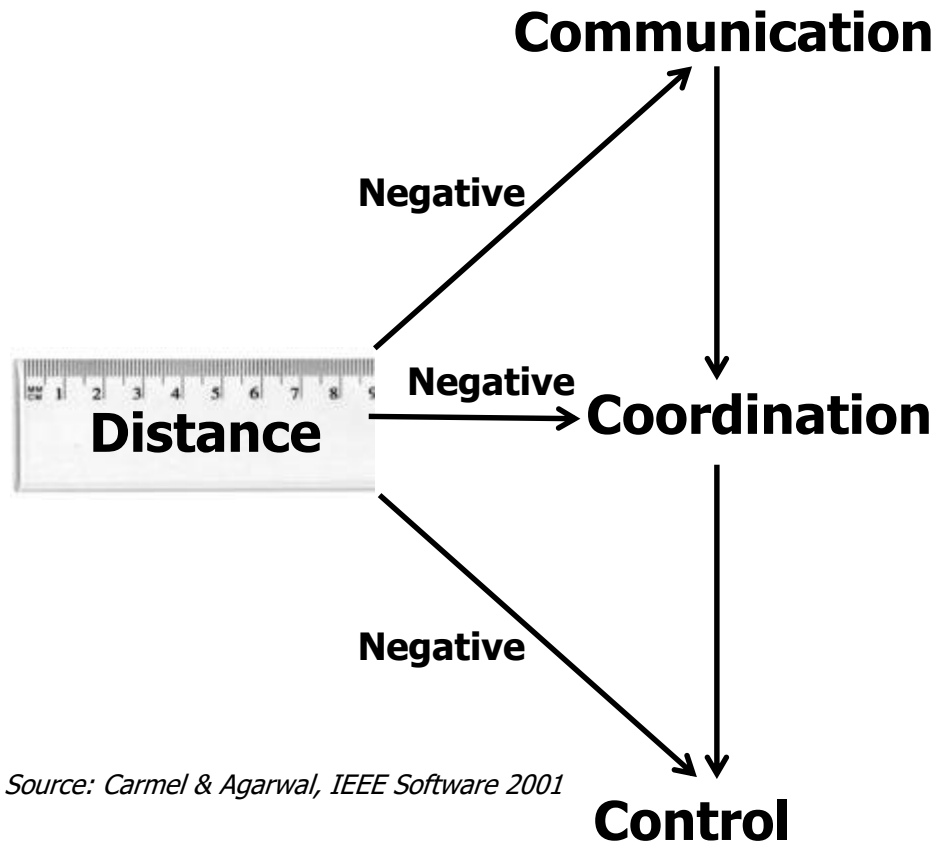
On the promise of Scrum to elevate distributed development and why it is so hard in practice

**Rini van Solingen – professor in global software engineering
D.M.vanSolingen@tudelft.nl**

Contents

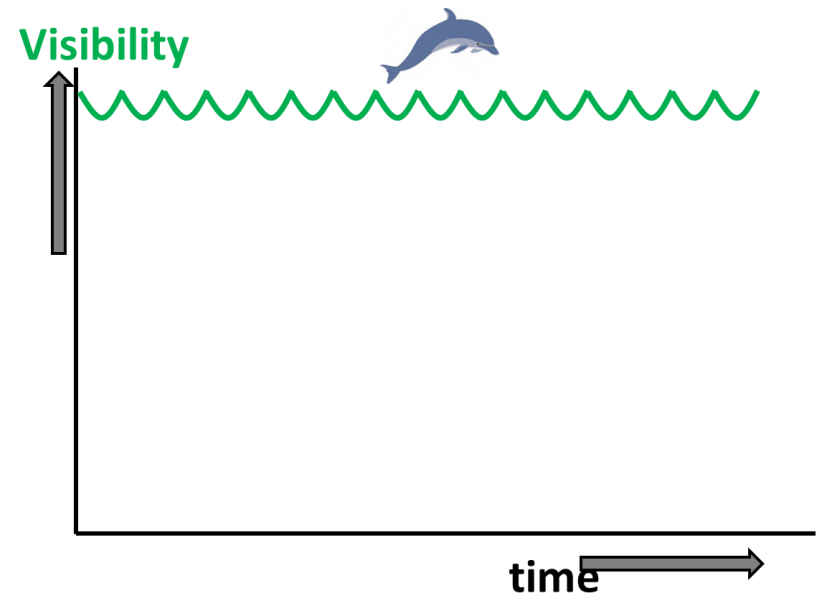
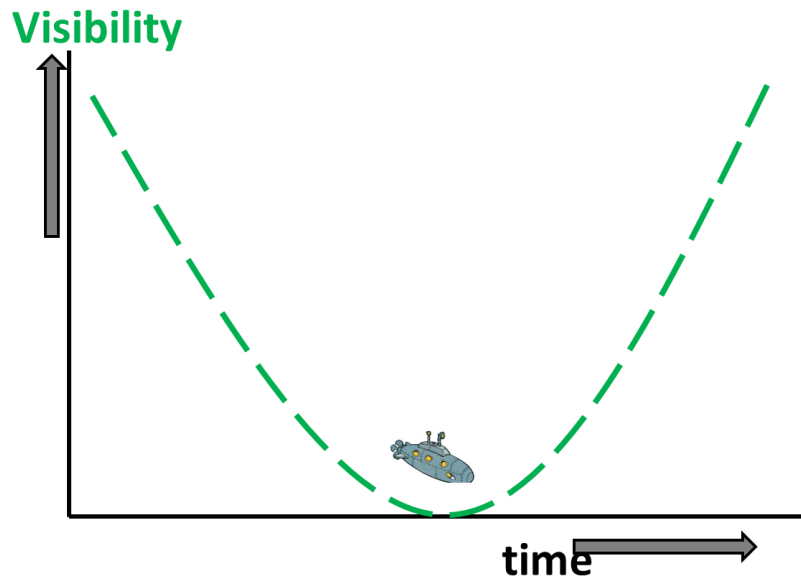
- Who am I, today?
- Adding distance to SW-Engineering
- Scrum and distribution – Sense or NonSense?
- 7 (anti) Patterns to get sick of Scrum
- Adding Scrum to distributed SW-Engineering
- Conclusion

Adding distance to sw-engineering

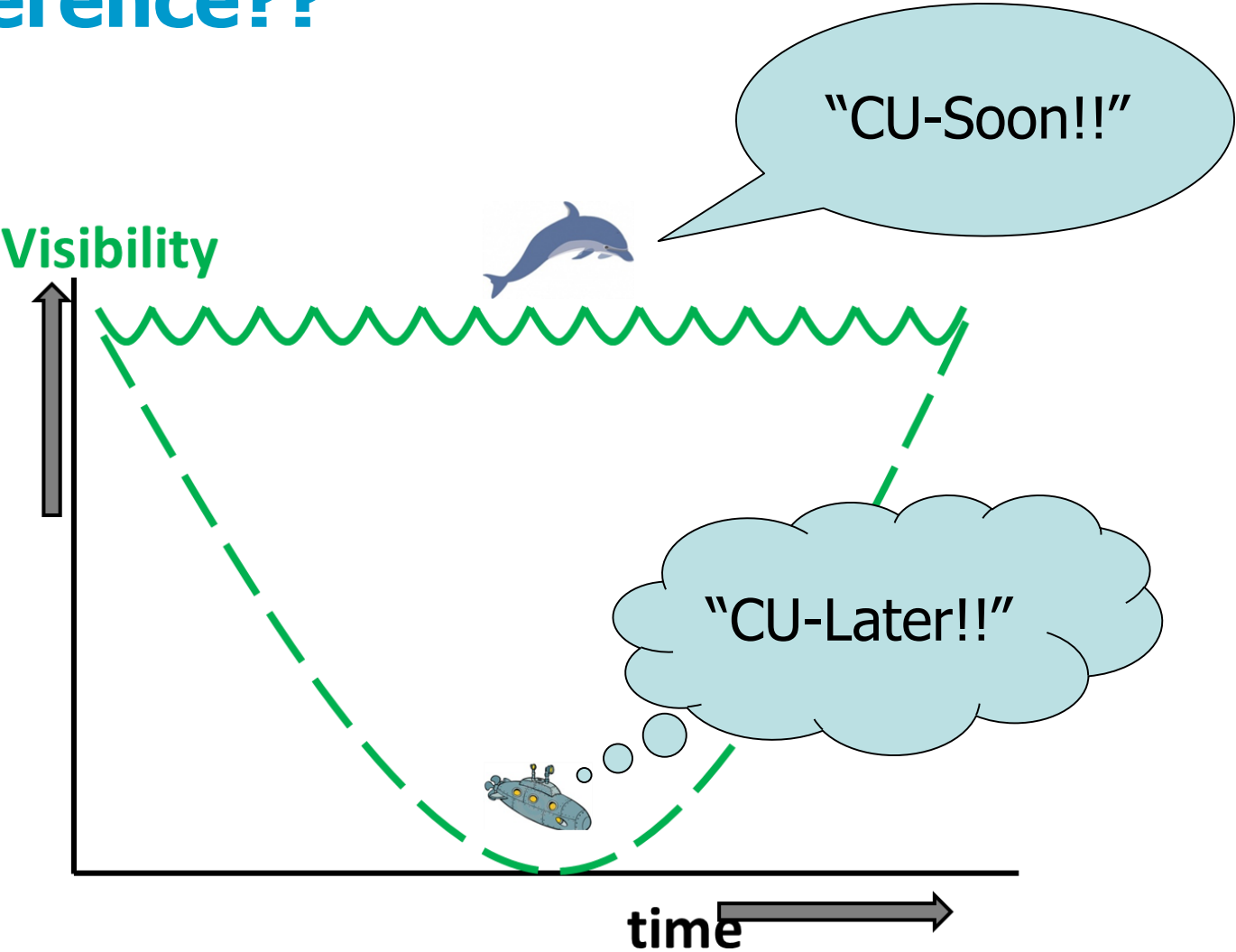


Source: Carmel & Agarwal, IEEE Software 2001

U-boat or Dolphin model – Preference?



Preference??



Amblyopia



March 30, 2017

6

7 (anti) Patterns – Sick of Scrum

- 1. Eiffel Tower pattern** (Software engineering is simple)
- 2. TV Evangelist pattern** (Religious fanatic Agilists)
- 3. Alice in Wonderland pattern** (Where do we want to go? And why?)
- 4. Hammer-Thumb pattern** (The problem is never in the approach)
- 5. Firing Squad pattern** (Governance is not Agile: purchase leads)
- 6. Superman pattern** (Product-ownership is never perfect)
- 7. Mother in Law pattern** (Fight or Flee – Hide or solve problems)



1. Eiffel Tower pattern

- Looking from the top of the Eiffel tower, everybody on the ground has the same size
- Software engineering is complex
- Customers don't know what they want and developers don't know how to build it
- Everything changes over time, so do requirements

Solution:

- Work in iterations and let the need for change emerge
- Anticipate request for changes in contract

2. Television Evangelist pattern



- “That’s not Agile!” - “Just do as I say, don’t do as I do!”
- Religious fanatic Agilist – annoying
- Often bureaucrats that have forgotten the Why
- Discussions on process don’t build products!

Solution:

- Ask them to cite the Agile manifest - completely!!
Ensure that you know it by heart -> cause they don’t
- Point at the ‘right sided behaviour’ that kills the left

3. Alice in Wonderland pattern



- “Would you tell me, please, which way I ought to go from here?”
“That depends a good deal on where you want to get to.”
“I don't much care where –”
“Then it doesn't matter which way you go.”
- Agile or Scrum as a goal on it's own
- Purpose and Why are not clear by themselves

Solution:

- Make the WHY really explicit
- Repeat the WHY as a mantra
- Measure the extend in which the WHY is realized

4. Hammer-Thumb pattern

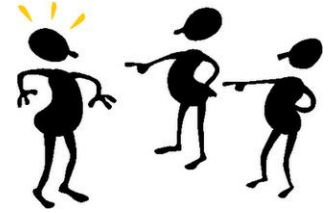


- Intrinsic abuse for over-processing
- Forgetting the product and focus on events & processes
- Blaming the method
- Looking for external factors or responsables for failure

Solution:

- ask for DONE product in every possible opportunity
- Discuss end-user/customer value continuously
- Invite key stakeholders to Sprint Reviews

5. Firing Squad pattern



- Purchase in the lead
- Agile collaboration with Non-Agile contract
- Contract are not revised, new agreements not captured
- End-customer happy; Agile supplier pays penalties

Solution:

- Exchange requests (procedure or approval trace)
- Wow in the contract, with responsibilities for roles
- Even better: Wow document that is changeable is mentioned in contract

6. Superman pattern



- Product owner dependency
- Expectation that any PO knows every answer and has brilliant vision towards the future - unrealistic
- Assumption that multiple PO's and CPO's decide easy and fast

Solution:

- Visible Flow to Ready with clarity by teams
- Agile Portfolio Management with PI events when large
- Step-wise simplification and dependency reduction

7. Mother in Law pattern

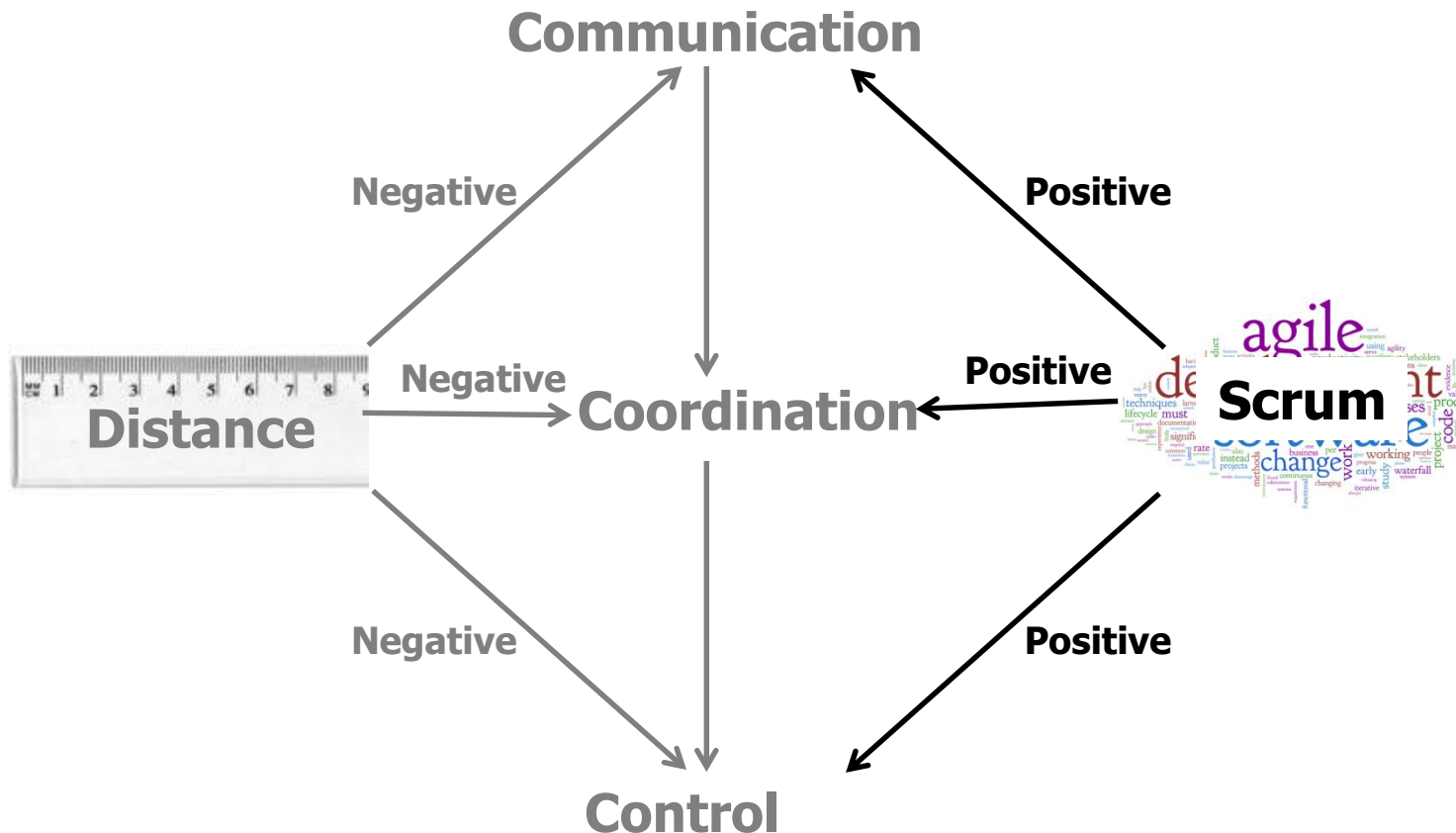


- Dyslike of visible dysfunctioning
- Fight or flee when confronted with structural issues
- Blame yourself or blame your supplier (who created the context in which the failure was created?)
- Especially present with customer-supplier relationships

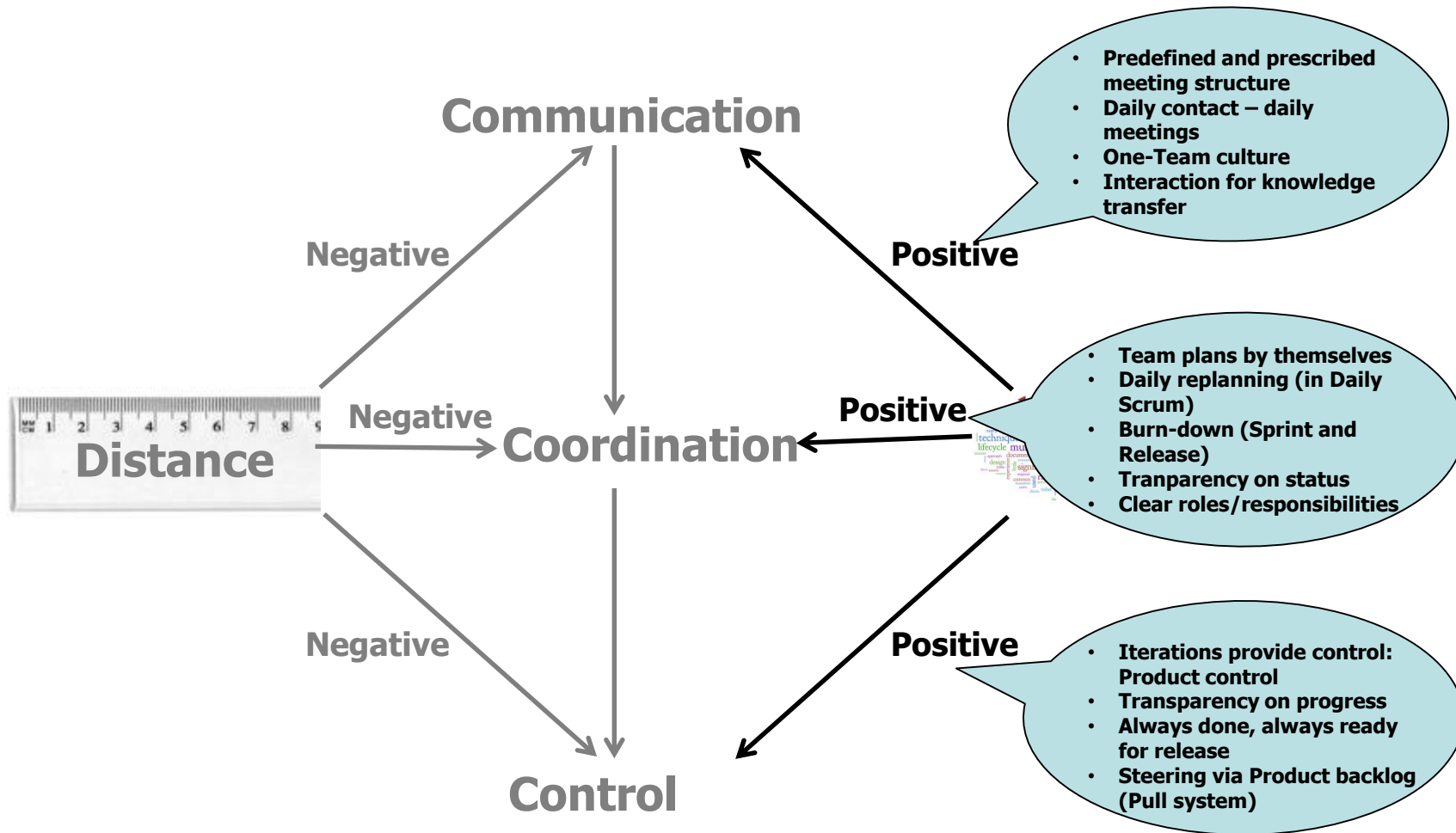
Solution:

- lead by example: embrace every chance to improve
- hold a future perspective and (help) solve the issue

Adding Scrum to distr. sw-engineering

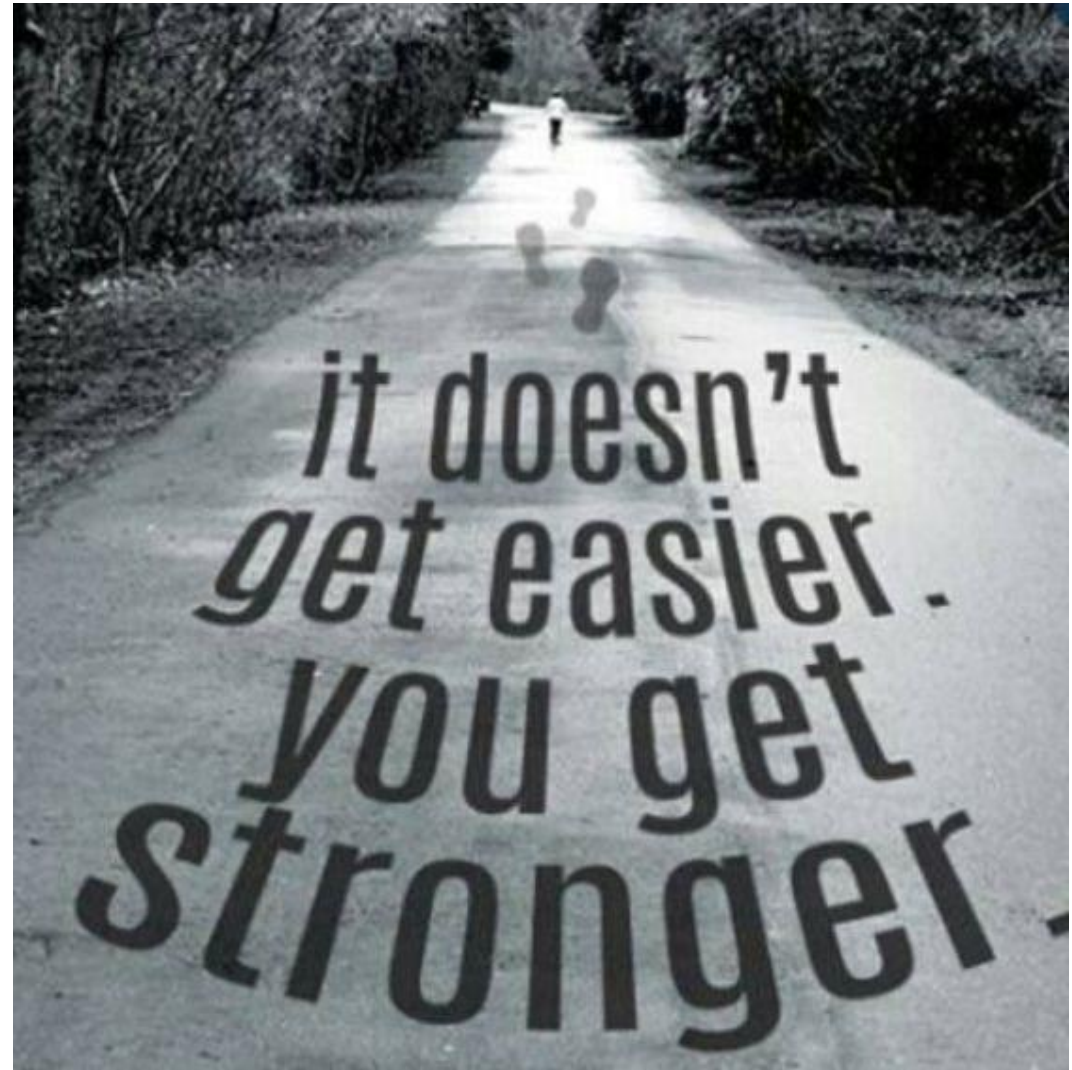


Adding Scrum to distr. sw-engineering



Conclusion

You can be sick of Scrum, but just don't forget: it is you who feels sick, not Scrum....



Thank you for your attention

Questions?

D.M.vanSolingen@tudelft.nl