When is agile better?

How the use of agile and autonomous teams affect success differently in different contexts (and other results)

We analysed the connections between software project outcome and the following factors:

- Development method
- Contract type
- Sourcing strategy
- Requirement volatility
- Project size
- Benefits management
- And a little bit about the use of autonomous teams

Philosophy: Success and failure patterns, not factors

Studies

- Four surveys with participants on IT management seminars
  - Asked to give information about their last, completed (or cancelled) project
  - 60-150 participants in each
  - From both client and provider side and many roles
- An interview-based study of 32 governmental software development projects
- Project data from an offshoring marketplace
  - More than 400.000 projects/tasks
  - Most of them very small

In spite of all the challenges, our empirical results may have value.

Weak evidence, as long as it is not misleading, is often better than no evidence.

SINCE THIS WORKSHOP IS ABOUT AUTONOMOUS TEAMS, LETS START WITH THAT …

”THE TEAM HAS SUBSTANTIAL FREEDOM IN SELECTING, SCHEDULING, PROCESSING AND/OR COMPLETING TASKS”

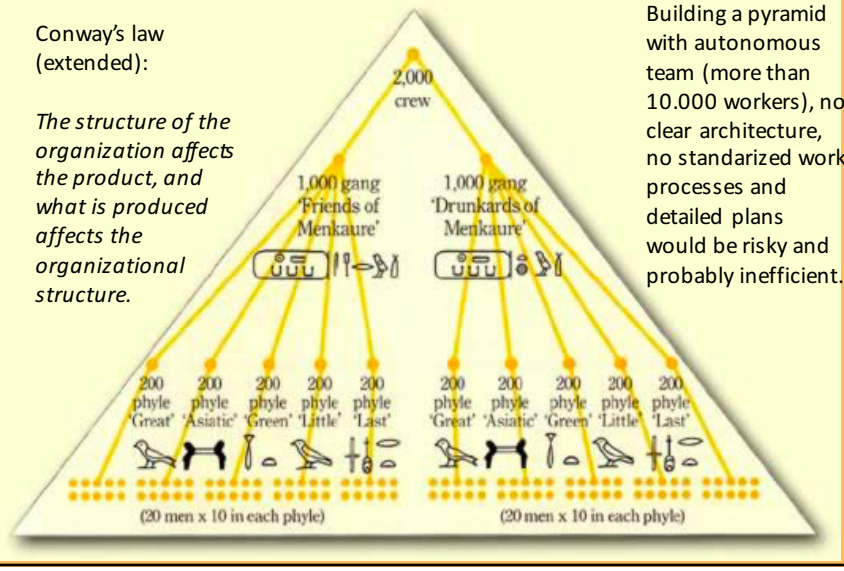
Autonomous teams are useful for many types of tasks, and is not a new way of collaborative effort

It may not fit all types of tasks.

Here: The organization of pyramid construction (Giza)

Conway's law (extended):

The structure of the organization affects the product, and what is produced affects the organizational structure.



Building a pyramid with autonomous team (more than 10.000 workers), no clear architecture, no standarized work processes and detailed plans would be risky and probably inefficient.

IS SOFTWARE DEVELOPMENT MORE LIKE GAME HUNTING IN TEAMS OR PYRAMID CONSTRUCTION?

(DOES SOFTWARE DEVELOPMENT USUALLY BENEFIT FROM THE USE OF AUTONOMOUS TEAMS?)

I GUESS YOU ALREADY THINK YOU KNOW THE ANSWER ON THIS, BUT LET'S GET EMPIRICAL. NEVER TRUST CLAIMS WITHOUT EMPIRICAL DATA.

Does it for example end up with (autonomous) teams fighting each others (as in a rugby scrum)

Survey design … (unpublished)

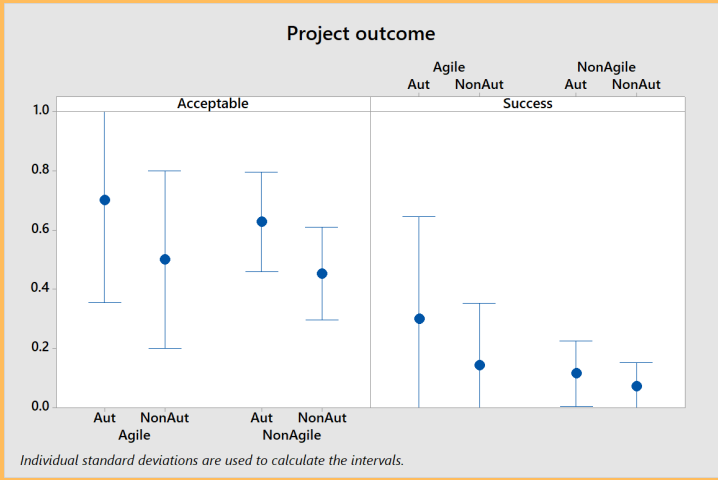
- Survey of 101 software projects (their last project, both provider and client respondents)
- ”Do you consider the development team(s) of the project to have been ”self-organized”?
  - Yes, no, don’t know (don’t know answers removed from analysis)
- 45% reported that the team(s) were self-organized
  - The question forces a dichotomy and is a subjective assessment.
  - Assumes that ”self-organized” is close to what people will think of as autonomous.
- The providers reported much higher proportion of self-organized teams than the clients (73 vs 23%).
  - Indicates a differences in use of terminology, lack of knowledge or something else …

Here is what we found ...

- Self-organized teams (average values)
  - Were more frequently used for **smaller projects** (2.3 vs. 3.0, using a scale from 1 to 4, where 2 = Small (0.1-1 mill Euro) and 3 = medium (1-10 mill Euro)
  - Were assessed to be slightly **more agile** (2.5 vs 2.8, using a scale from 1=very agile to 5=not at all agile) and used more agile practises (3 vs. 1)
    - More use of product backlog (71 vs 61%), velocity (40 vs 11%), stand up meetings (69 vs 29%), but same degree of frequent deliveries (2.2 vs 2.2, on a scale from 1=frequent deliveries to client and 4=only end-deliveries)
  - Had a slightly **less involved client** (1.9 vs 1.8, using a scale from 1="very involved" to 4 ("not much involved").
  - Were **less** likely to have a **detailed, upfront project plan** (40% vs 60%).
  - Had about the **same requirement volatility** (1.9 vs 2.0, where 1=very much and 4=very few/none) and **similar use of contracts** (only slightly less use of fixed price contracts).

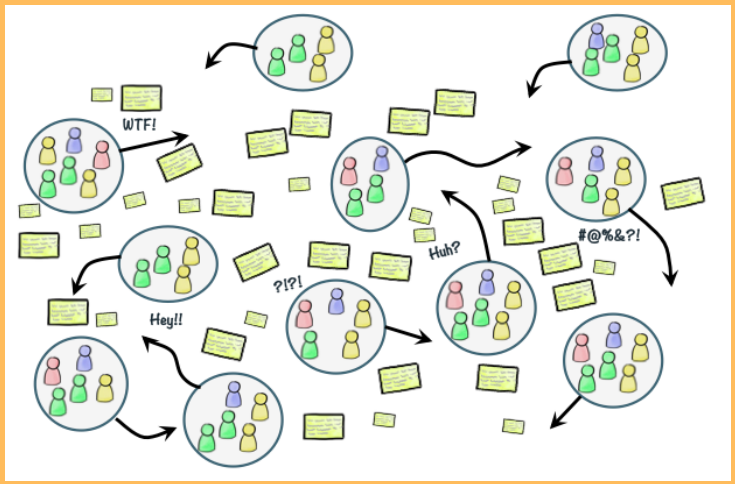
More importantly, did self-organized (autonomous) teams deliver better results?

Yes! Especially when working agile with frequent deliveries to client

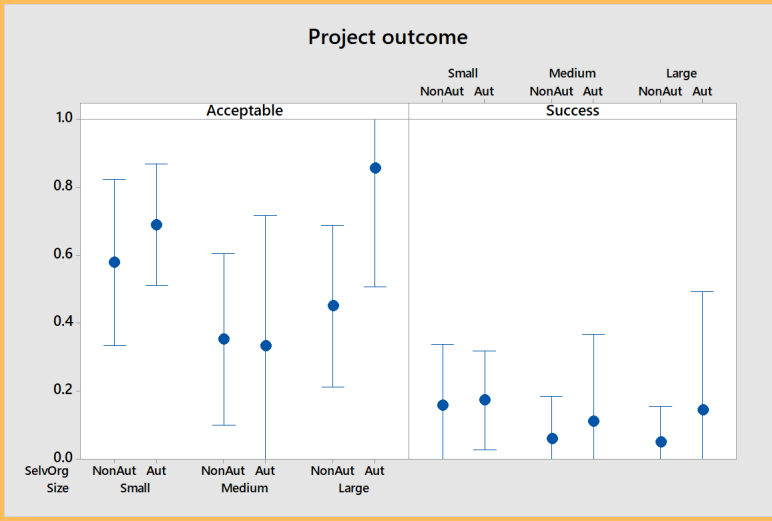


Agile = Perceived as "very agile"/"agile" and with freq. deliveries during the project execution.
Acceptable = Perceived as acceptable or better wrt client benefits, time control and cost control
Successful = Perceived as successful wrt client benefits, time control and cost control

What about scaling? Does autonomous teams on large projects lead to chaos?



It seems to scale well ...



Small =< 1 mill Euro, Medium = 1-10 mill Euro, Large = > 10 mill Euro

Selected results (related to agile) from our surveys

A survey of 63 Norwegian software development projects

ELSEVIER

Information and Software Technology

Volume 78, October 2016, Pages 83-94

A survey on the characteristics of projects with success in delivering client benefits

Magne Jørgensen

Show more

https://doi.org/10.1016/j.infsof.2016.05.008

Get rights and content

When looking at agile projects we found that "agile is not agile"

The numbers show the increase (in percent points) in proportion of successful projects

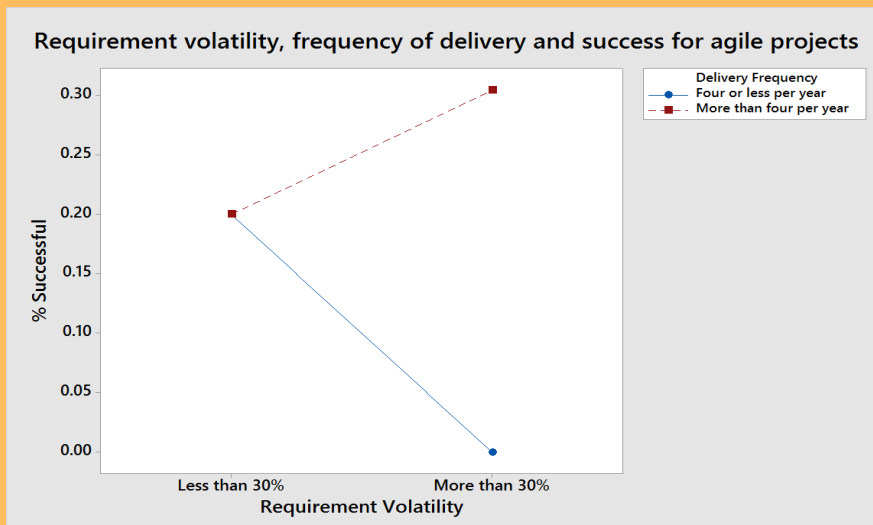
	Agile	Frequent delivery to production	Flexible scope
Client benefits	16%	22%	29%
Technical quality	21%	6%	32%
Budget control	2%	22%	29%
Time control	8%	11%	24%
Efficiency	11%	5%	24%

Agile was only connected with more client benefits when including frequent delivery to production and flexible scope.

Agile projects not including these practices were LESS successful than non-agile projects!

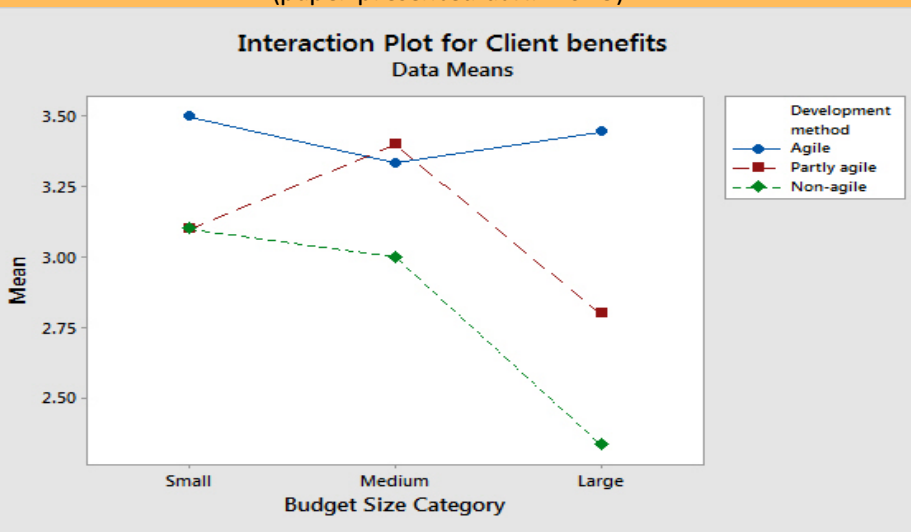
Similar results in our later follow-up surveys and studies

Similarly, in a more recent study (unpublished), we found that the presence of frequent deliveries in agile projects was mainly important when connected with high requirement volatility



Agile software projects seem to be less affected by large project size

(paper presented at XP 2018)



Analysis of data about more than 400.000 small projects (offshoring marketplace) and an in-depth survey of 35 large governmental projects

ELSEVIER

International Journal of Project Management

Volume 35, Issue 8, November 2017, Pages 1573-1586

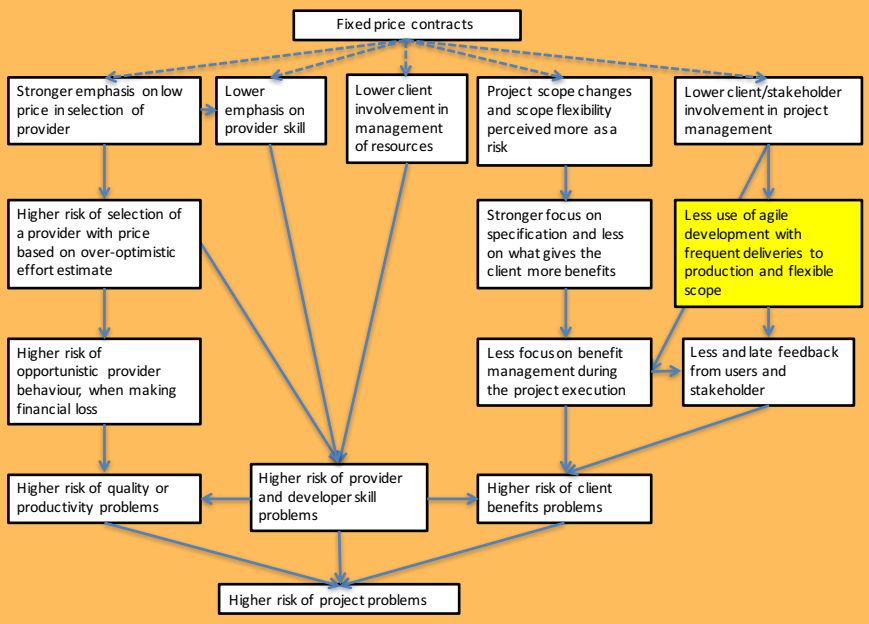
Direct and indirect connections between type of contract and software project outcome

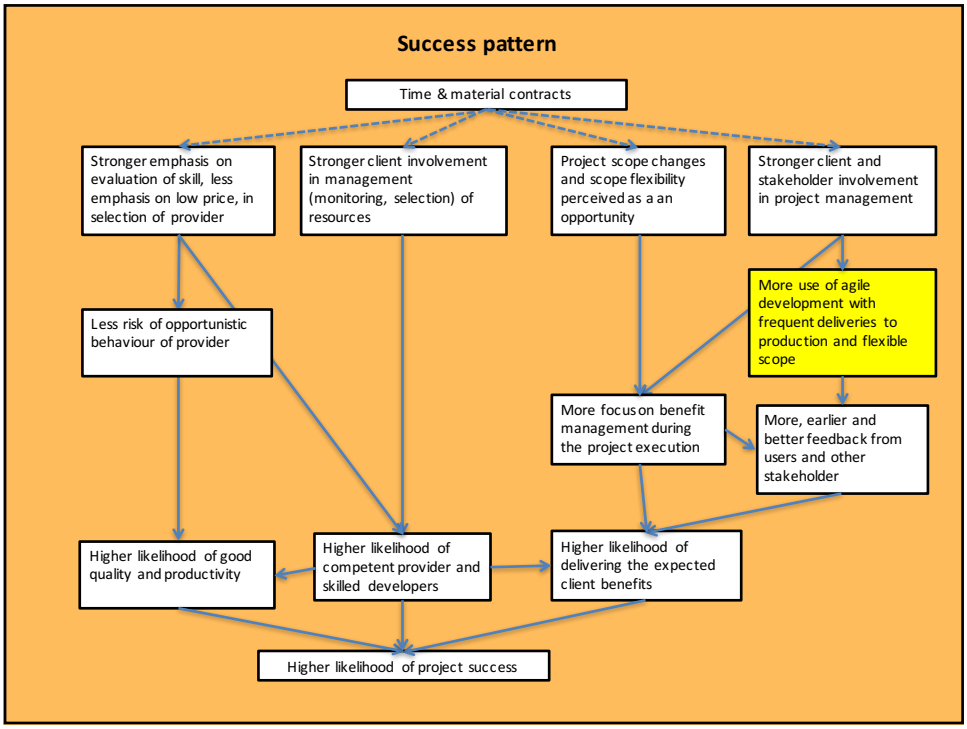
Magne Jørgensen, Parastoo Mohagheghi, Stein Grimstad

Show more

https://doi.org/10.1016/j.ijproman.2017.09.003

Failure pattern starting with the choice of contract



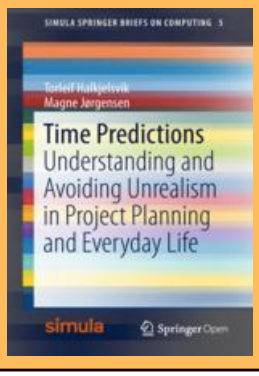


What I wanted to say ...

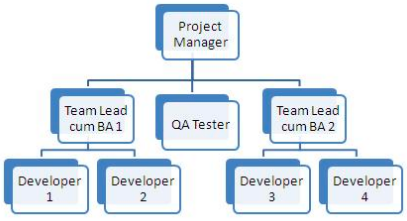
- The evidence (although not very strong) suggests that autonomous teams are more successful.
  - The causal connections may be complex
- Agile is not agile, and especially «frequent deliveries to production» (enabling feedback) and »flexible scope» is connected with more success.
  - This is especially the case when there is a high requirement volatility (which to some extent is caused by the feedback) and when projects get larger.
- It is when we analyse success and failure patterns, not factors, that we get the most useful results and the best insight.

QUESTIONS?

Download my newest book for free: tinyurl.com/timepredictions



Pyramid construction organizational thinking common in many workplaces



Analysis challenges:

- Poorly defined concepts, e.g., what is agile and what is an autonomous team?
- Forcing dichotomies on continuous scales
- Cause-effect vs correlation
- Subjectivity in measurement
- Little control of sample representativeness (convenience samples, mainly from Norway)
- Missing context information