

Scaling Agile: Beyond The Hype

Agile Business Day 2016, Venezia, Italy



Giovanni Asproni
email: gasproni@asprotunity.com
twitter: [@gasproni](https://twitter.com/gasproni)
linkedin: <http://www.linkedin.com/in/gasproni>



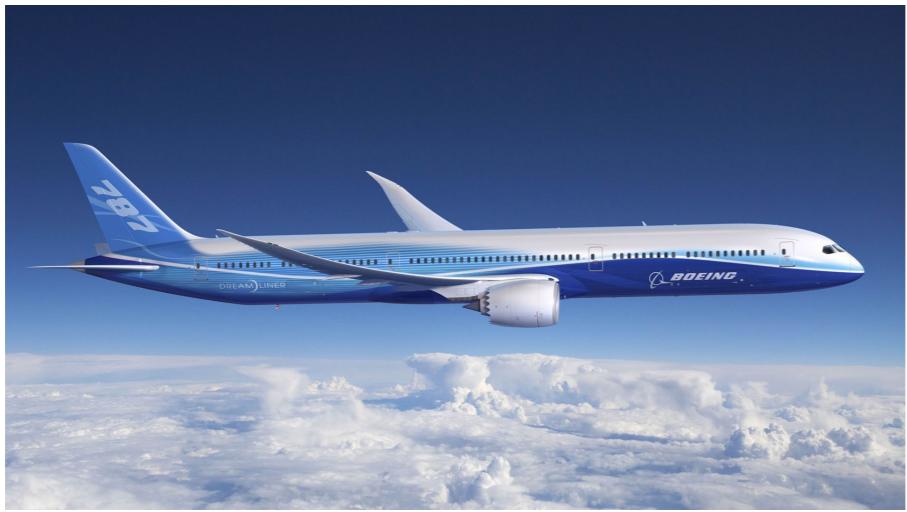




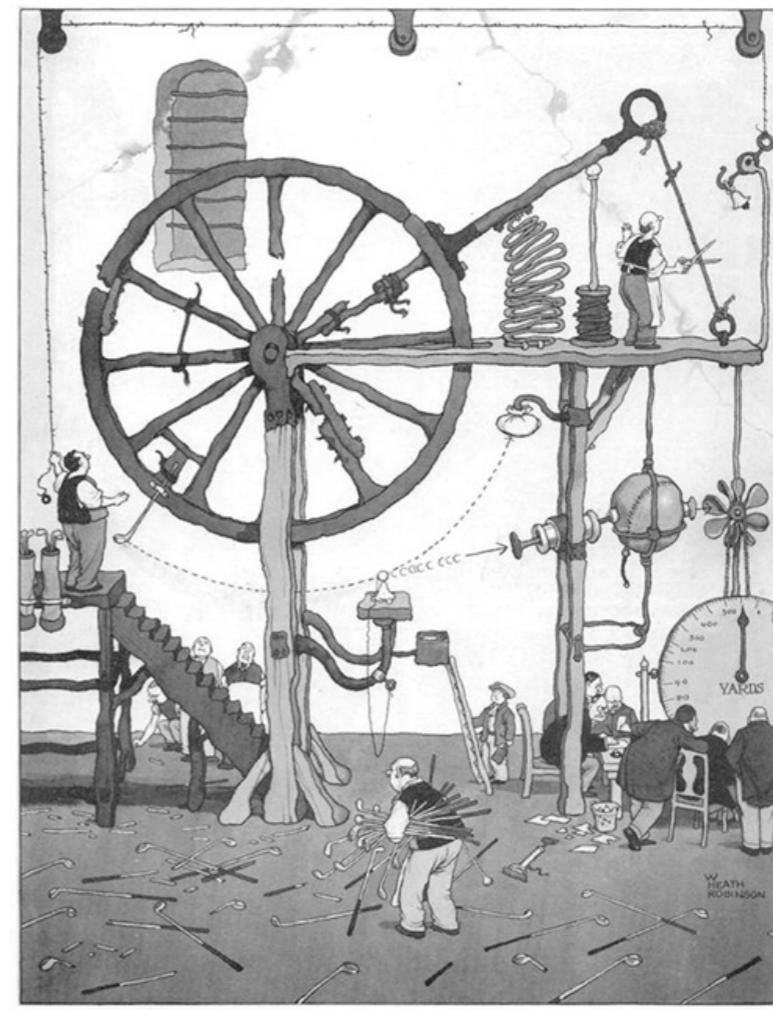
La Mia Esperienza

- Lavorato soprattutto in progetti comprendenti 4-5 teams
- Esperienza recente:
 - Alcuni progetti composti da 4-5 teams
 - Uno comprendente ~80 teams con ~700 programmatori e testers
 - Uno comprendente ~120 teams con ~1300 programmatori e testers
 - Uno comprendente 8 teams con ~80 programmatori e testers
 - Spesso chiamato per “riparare i teams e il progetto” dopo lo scaling-up





As a ... I want...so that...



Legge Fondamentale Della Scalabilità

Scaling up amplifies the bad and makes the good more difficult

Scalare amplifica i problemi e rende più difficile l'uso di pratiche adeguate.

Scalare crea nuovi problemi

Prerequisiti Per Poter Scalare

- Obiettivi condivisi
- Architettura adeguata
- No colli di bottiglia
- Automazione
- Comunicazione efficace ed efficiente
- Capacità di management e tecniche adeguate
- Metriche per misurare produttività e qualità
- User stories di buona qualità
- Prioritizzazione e pianificazione

Primo Paradossal

La gran parte dei progetti vengono scalati perché non ne hanno i prerequisiti

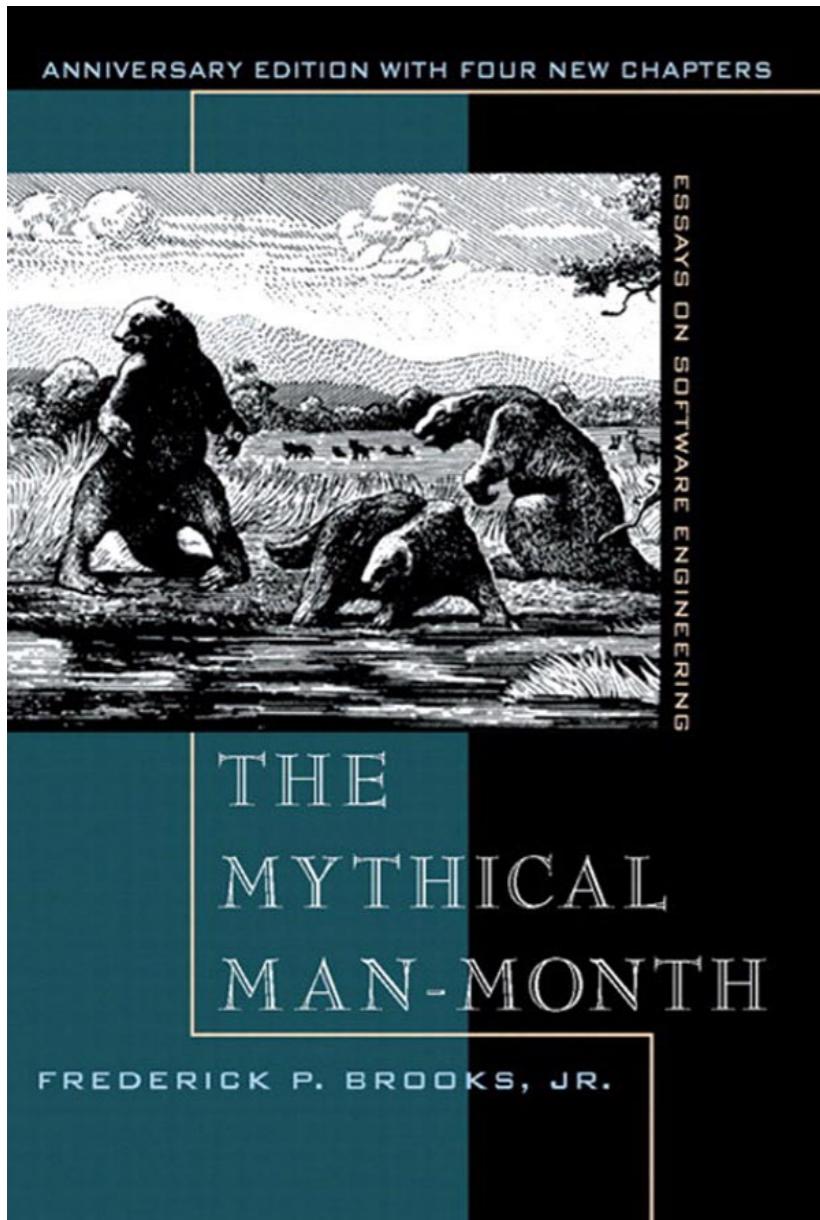
Secondo Paradosso

I progetti che hanno i prerequisiti per poter essere scalati hanno una necessità minore di scalare

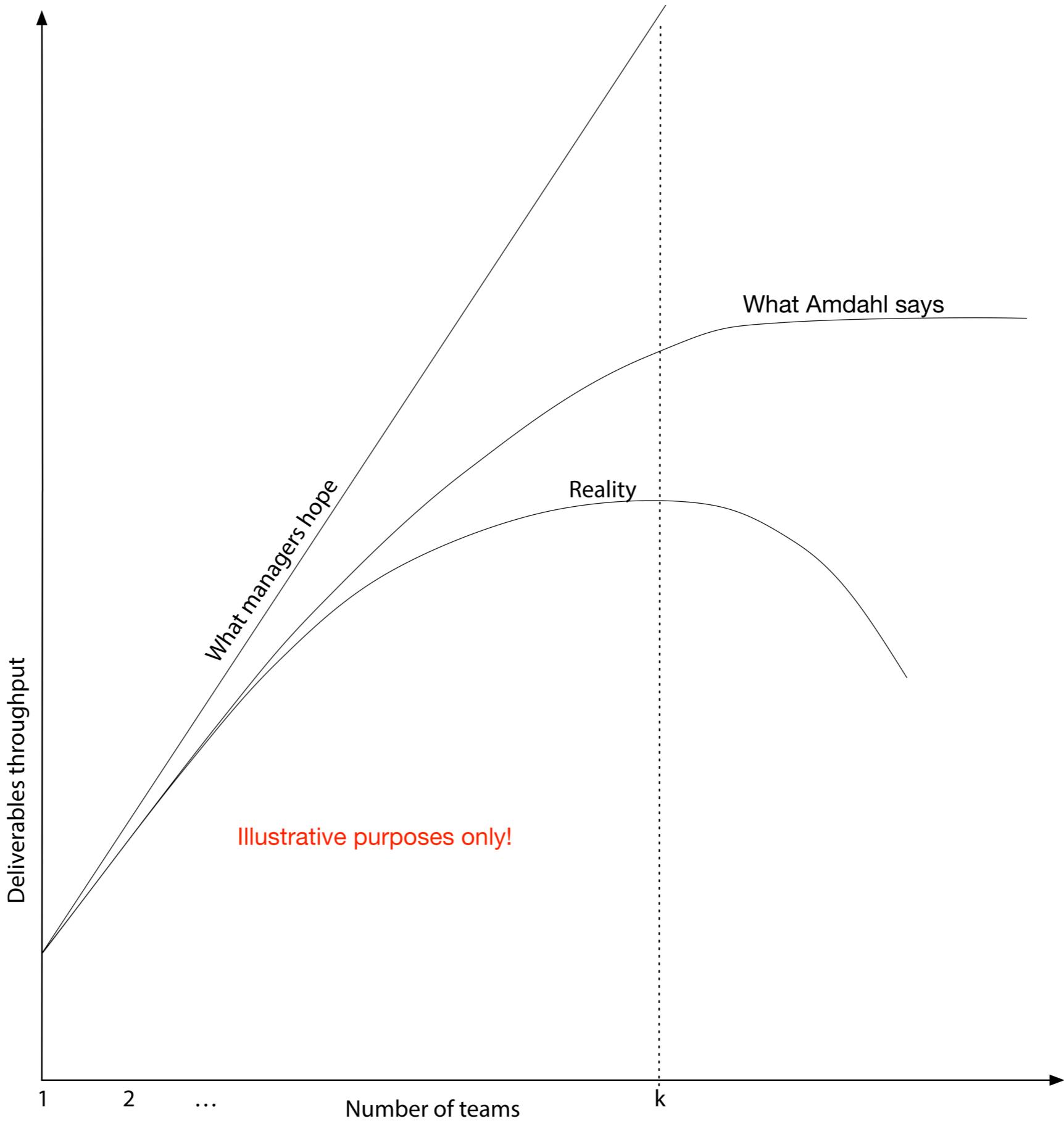
Come Procedere?

- Di quanto si può scalare?
- Che processi e strutture si devono adottare?
- Feature o component teams?
- Come procedere per aggiungere nuovi teams?

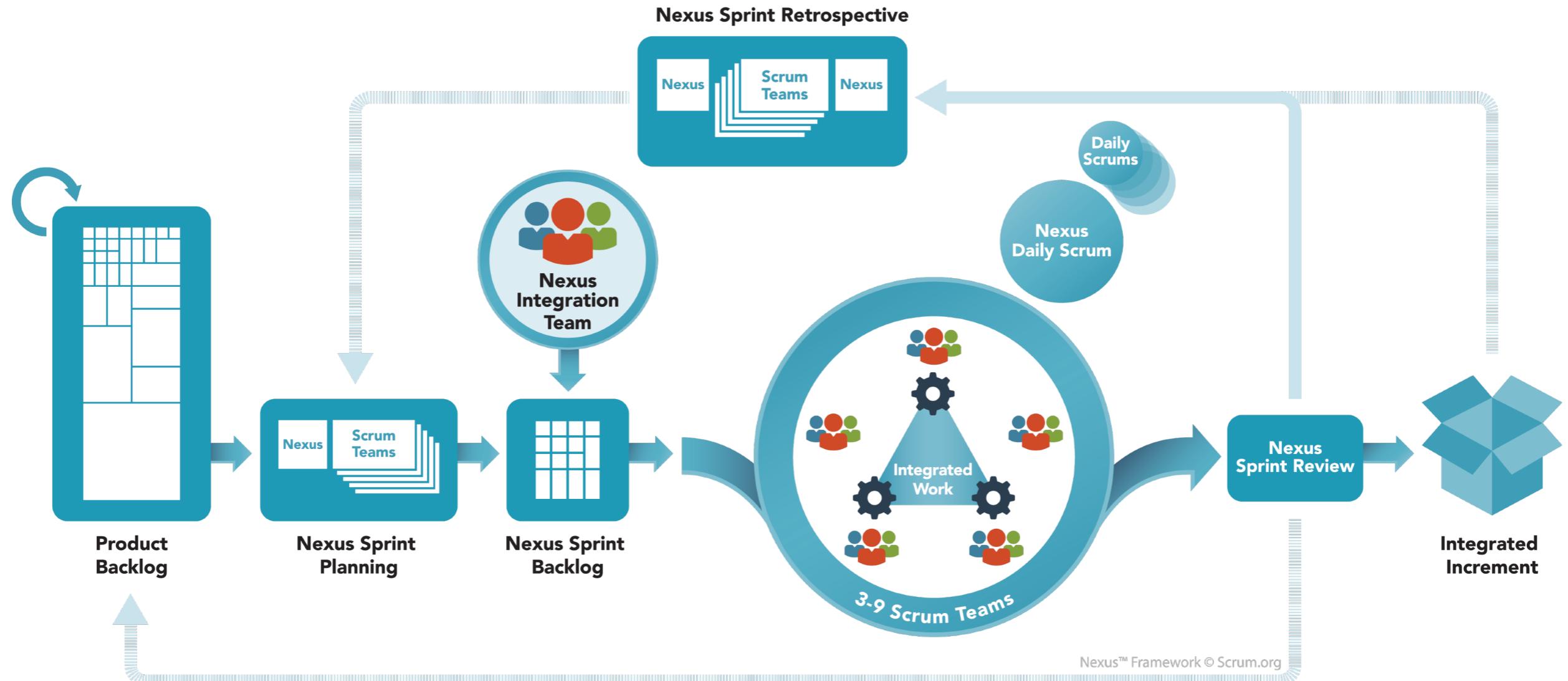
Amdhal's Law

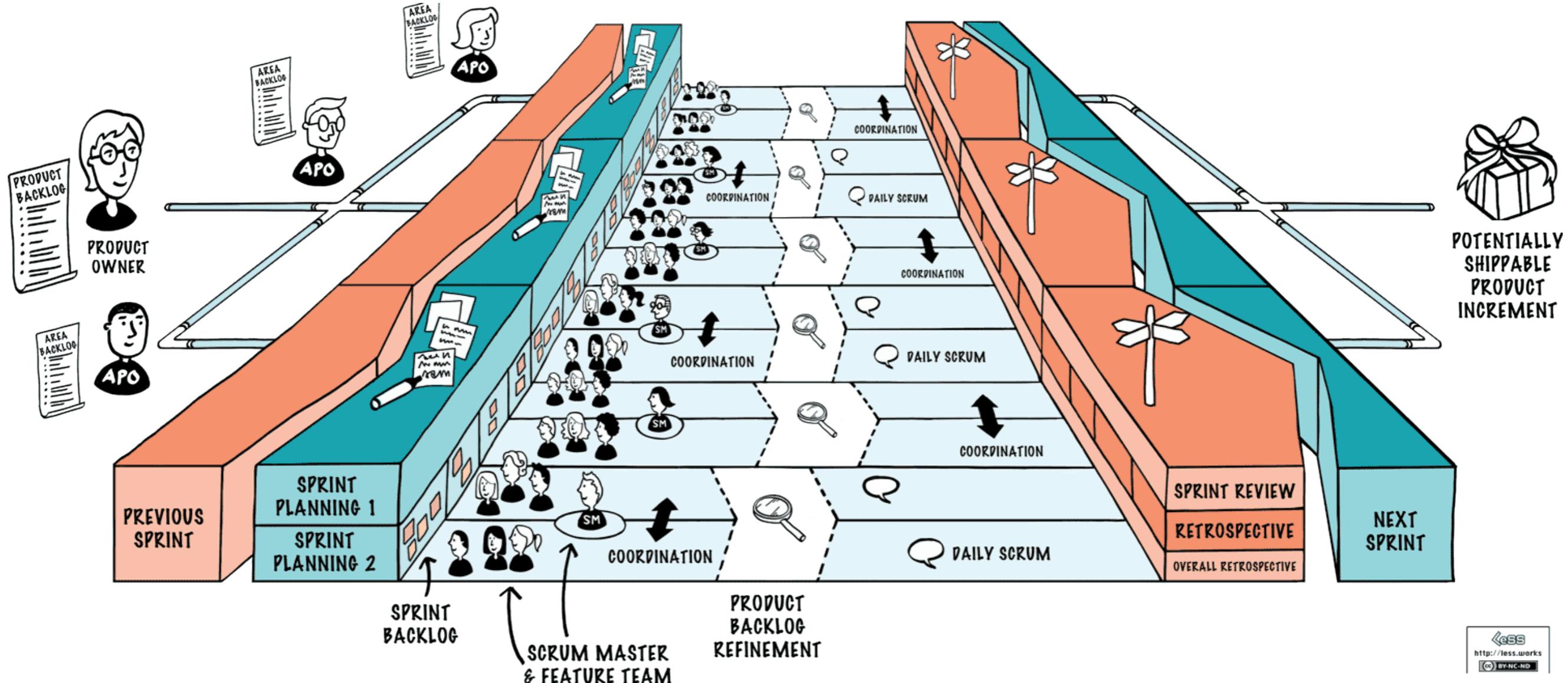


“The number of months of a project depends upon its sequential constraints. The maximum number of men depends upon the number of independent subtasks. From these two quantities one can derive schedules using fewer men and more months. (The only risk is product obsolescence.) One cannot, however, get workable schedules using more men and fewer months.”

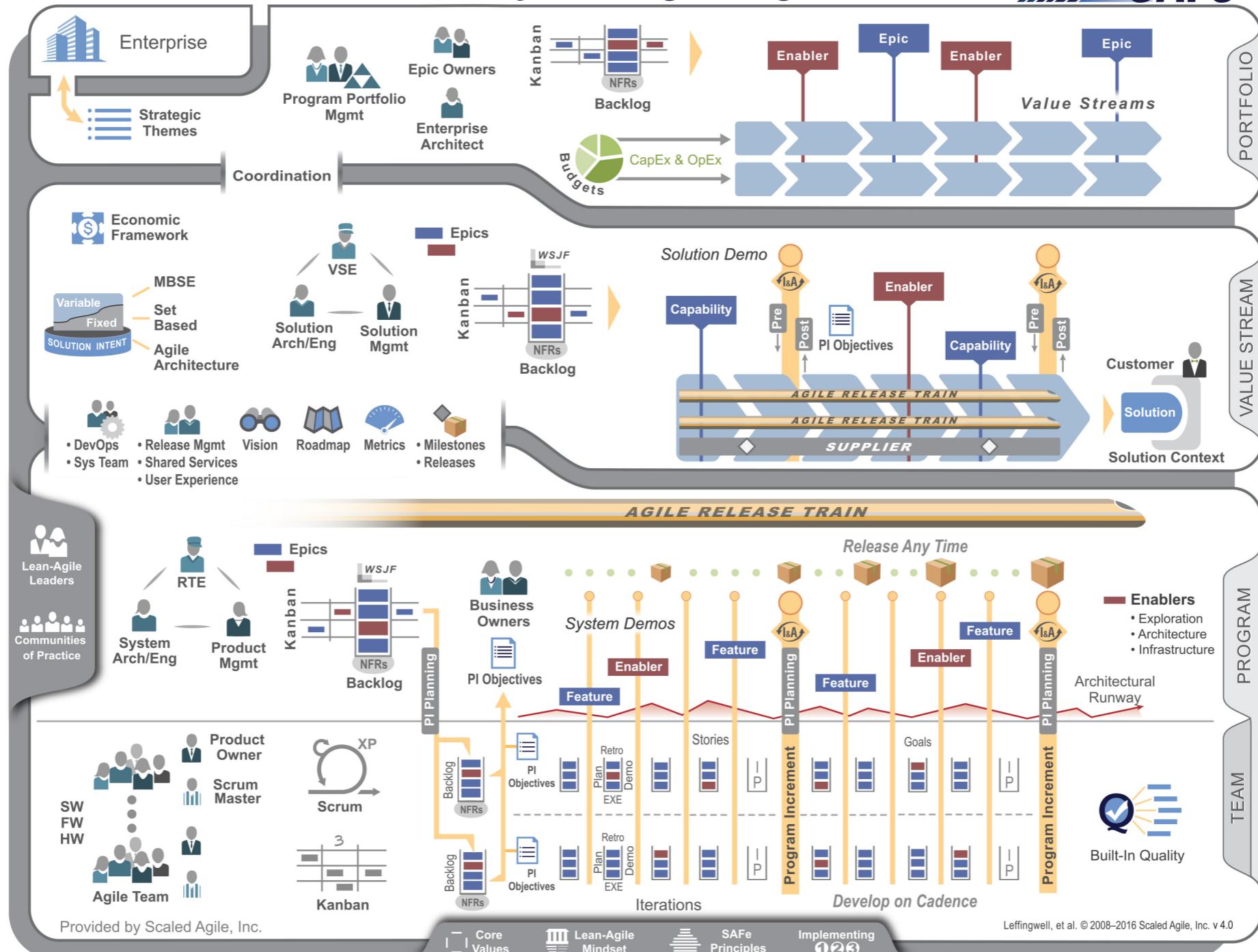


NEXUS™ FRAMEWORK



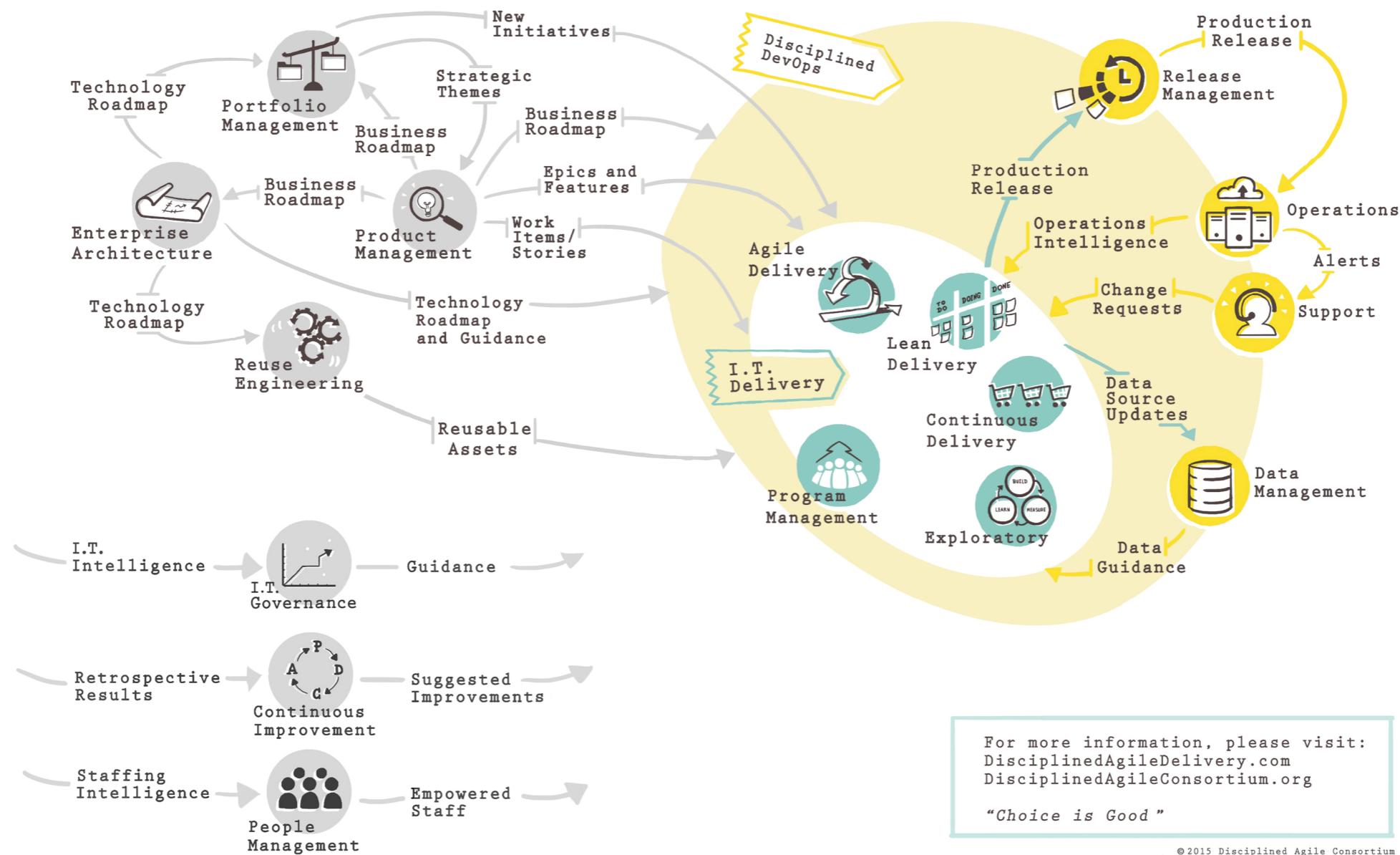


SAFe® 4.0 for Lean Software and Systems Engineering



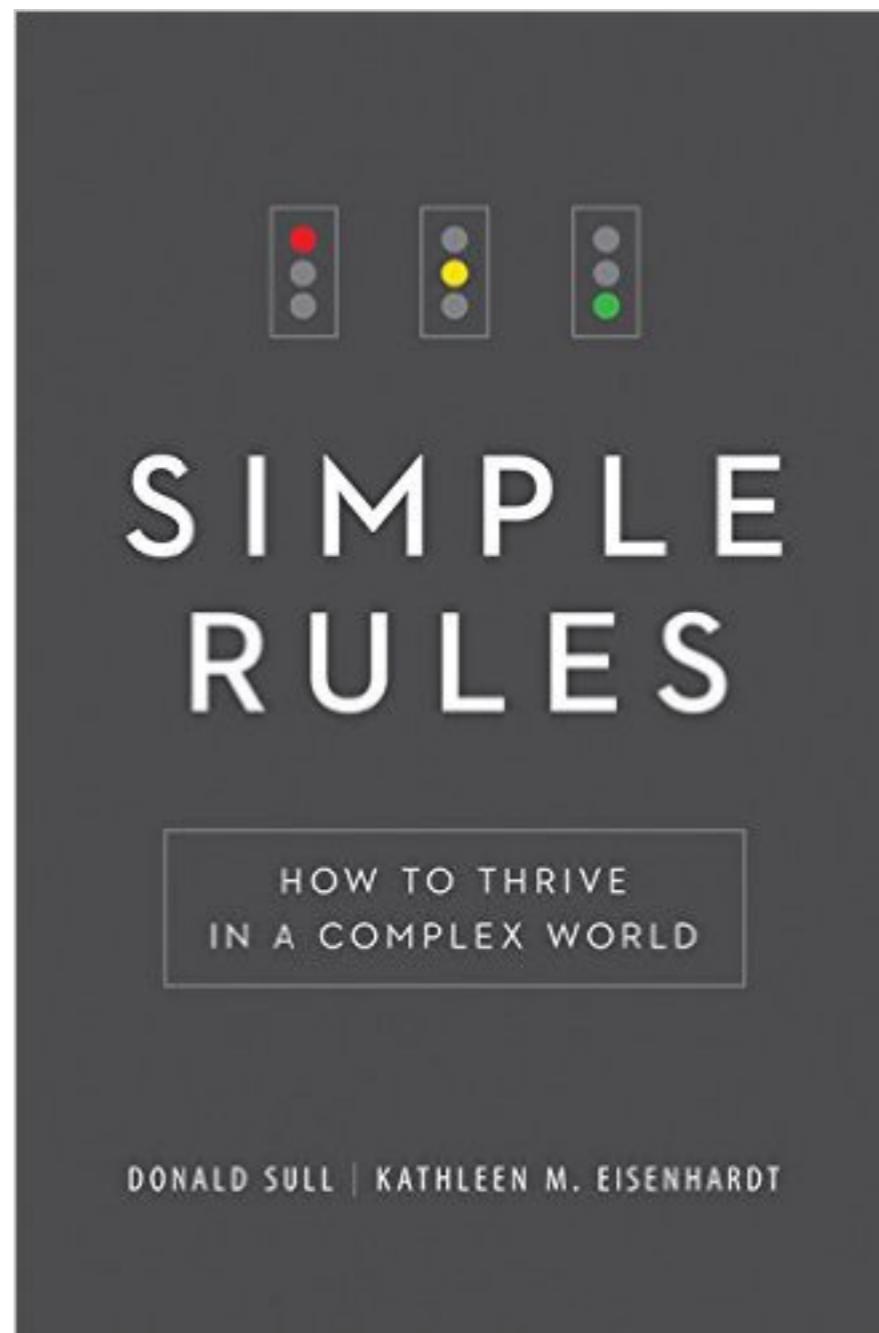
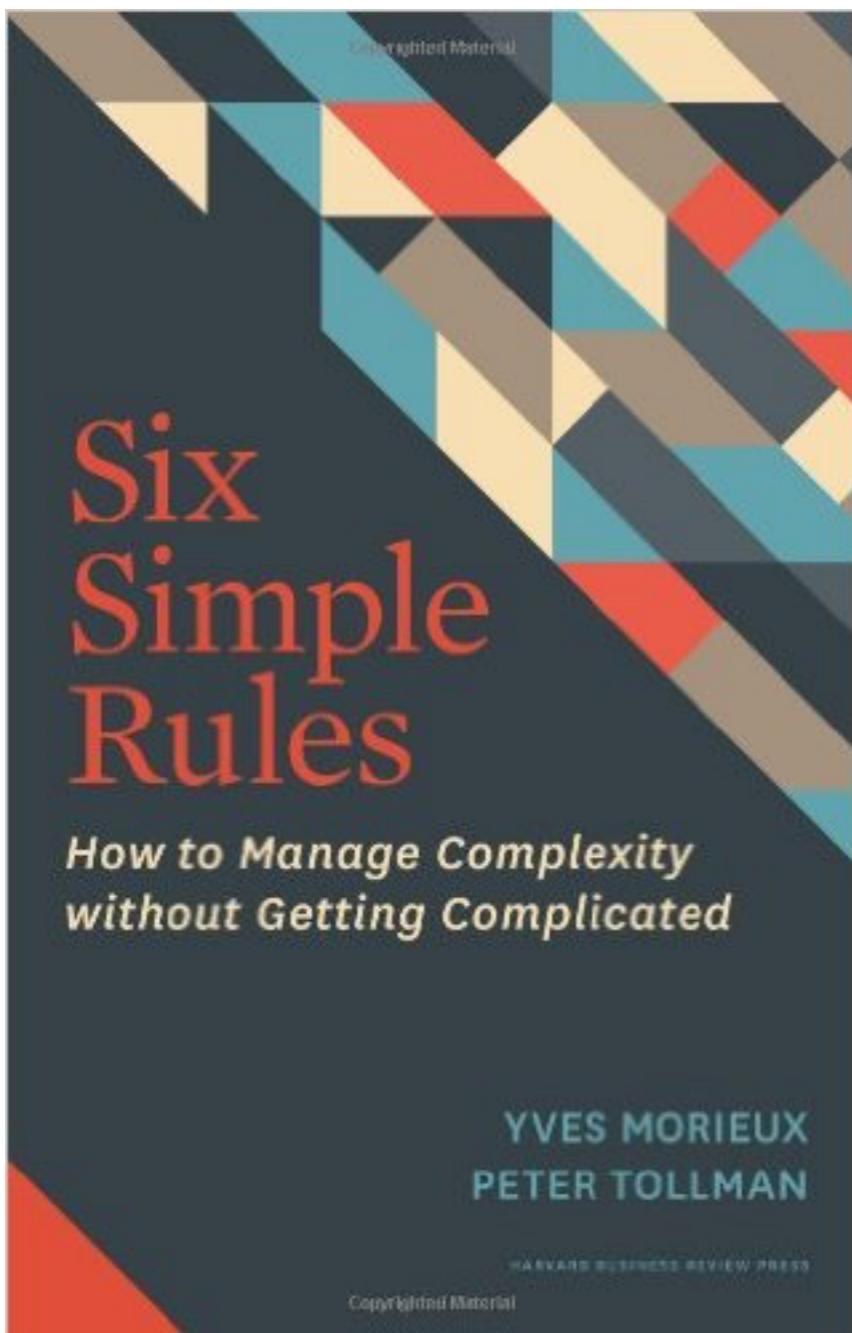
Disciplined Agile 2.0

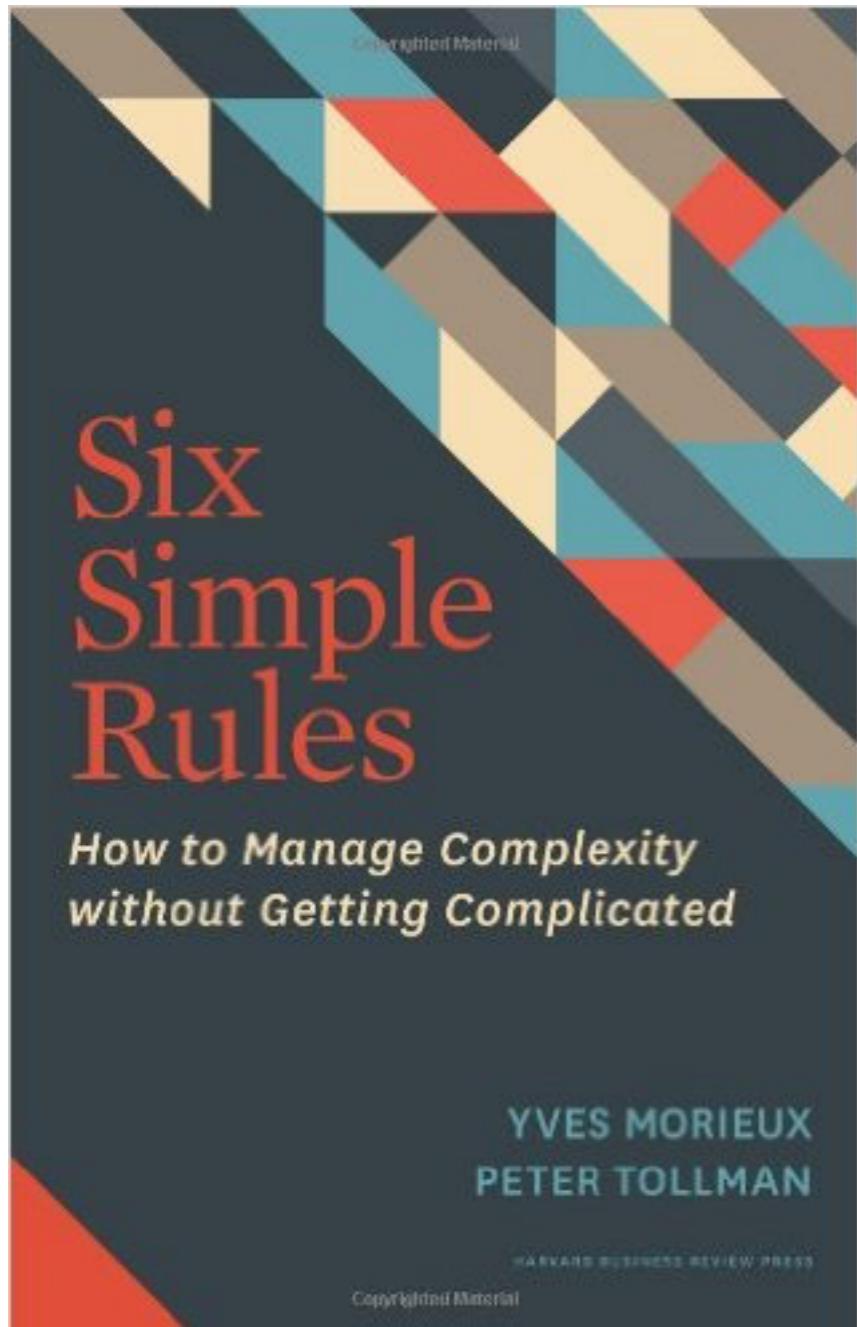
A Process Decision Framework for Enterprise I.T.





WITTFMT?





1. Understand What Your People Do
2. Reinforce Integrators
3. Increase the Total Quantity of Power
4. Increase Reciprocity
5. Extend the Shadow of the Future
6. Reward Those Who Cooperate

Organizzazione Dei Team



- Feature
- Component
- Mix

Aggiungere Persone E teams



THE SYSTEMS BIBLE

THE BEGINNER'S GUIDE
TO SYSTEMS LARGE AND SMALL

BEING

THE THIRD EDITION OF *SYSTEMANTICS*

BY

JOHN GALL

GENERAL SYSTEMANTICS PRESS WALKER MINNESOTA

“A complex system that works is invariably found to have evolved from a simple system that worked.”

Da Evitare!

Team creati dai manager senza considerare le implicazioni tecniche

Descaling



Giovanni Asproni

@gasproni

Before scaling up your methodology, consider scaling down the mess. That often is a better solution

Conclusioni

- Scalare, spesso, non è necessario
- Quando lo è, bisogna farlo con cautela
 - Processi complessi e maggior struttura non sono sempre efficaci
 - Spesso le soluzioni semplici che tengono conto delle necessità delle persone sono migliori di metodologie complesse
 - C'è sempre un limite a quanto un progetto può essere scalato
- È possibile che ridurre le dimensioni del progetto (scaling down) abbia migliori risultati