The Balance of Alignment & Autonomy

24th November 2023 LeSS Meetup New York City

Mark Bregenzer

My Background

Self-employed

Organizational Design Coach, Agile Software Development Scaling Expert

Agile Experience:

- Certified Organization Designer 2023
- Certified LeSS Trainer 2015
- Scaling Scrum Fundamentals, Scrum Alliance 2015
- Certified LeSS Practitioner 2015
- Certified Scaled Agile Program Consultant 2014
- Agile Coach since 2009, Consultant since 2011
- Certified Scrum-Master since 2007

Software development Experience:

Since 1997 as developer, lead developer, subproject leader, technical Coach…

Business Areas:

Telecommunication, automotive, insurance, retail and e-commerce

www.bregenzer.eu

Agenda

/ Alignment

/ Autonomy

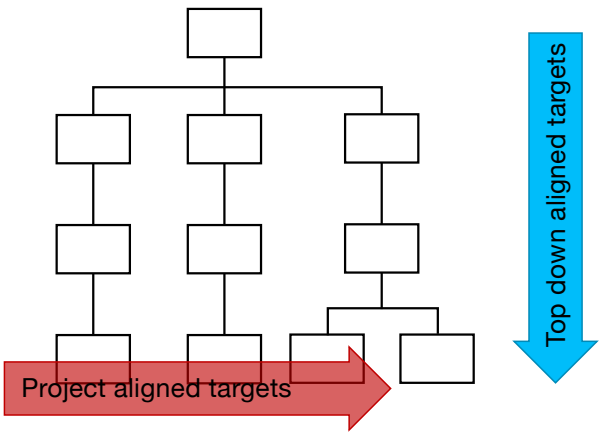
/ Agile Scaling Frameworks

/ How to balance Alignment and Autonomy

/ Good and bad Practices

Alignment

Traditional organization



Alignment is established by...

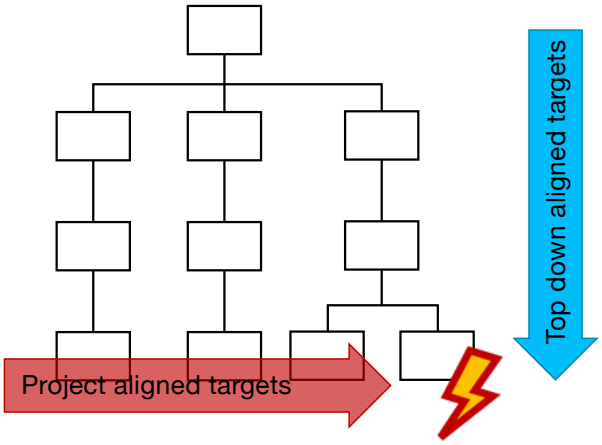
- Organizational design with separated responsibilities and accountabilities
- Fixed targets broken down on each level
- Managed by command & control with target-actual comparison

Characteristics

- Stable
- Controllable
- Focused
- Slow in decision making

Alignment

Traditional organization



Alignment is established by...

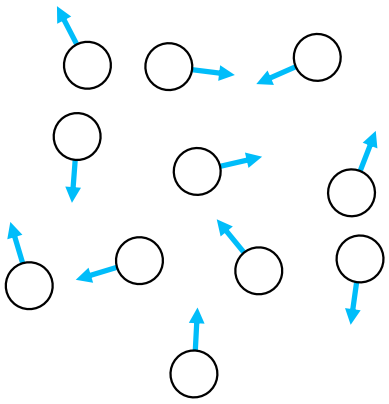
- Organizational design with separated responsibilities and accountabilities
- Fixed targets broken down on each level
- Managed by command & control with target-actual comparison

Characteristics

- Stable?
- Controllable?
- Focused?
- Slow in decision making

Autonomy

Agile organization



Alignment is not required...

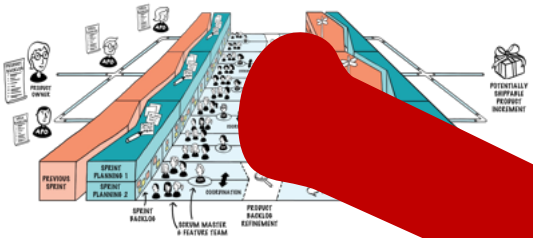
- Organizational design with independent small teams
- Interdisciplinary teams
- Self-organized, self-managed

Characteristics

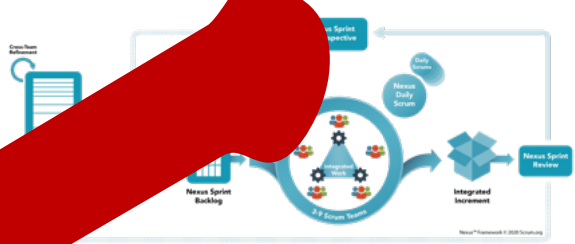
- Flexible
- Hard to control
- Single/self focused
- Fast in decision making

Agile Scaling Framework Try To Solve That Problem

LeSS/LeSS Huge

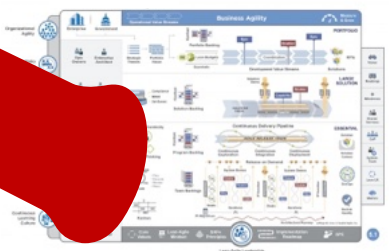


NEXUS



S@S

SAFe



Let’s Talk About The Essentials

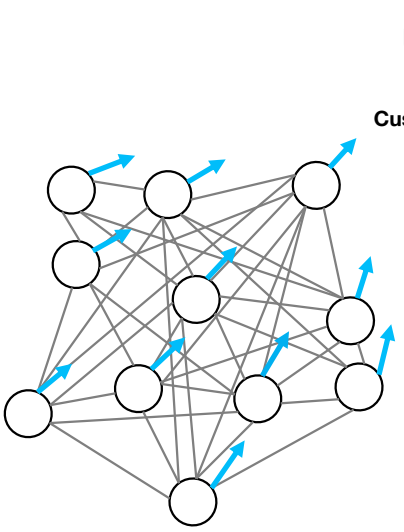
In Scaled Agile Product Development we want to achieve a good **balance** of

Autonomy for fast decisions & adaptiveness and

Alignment for consistent products and services

Networks Are Best In Dealing With Complexity

A network of agile teams



What give this organization orientation (Alignment)?

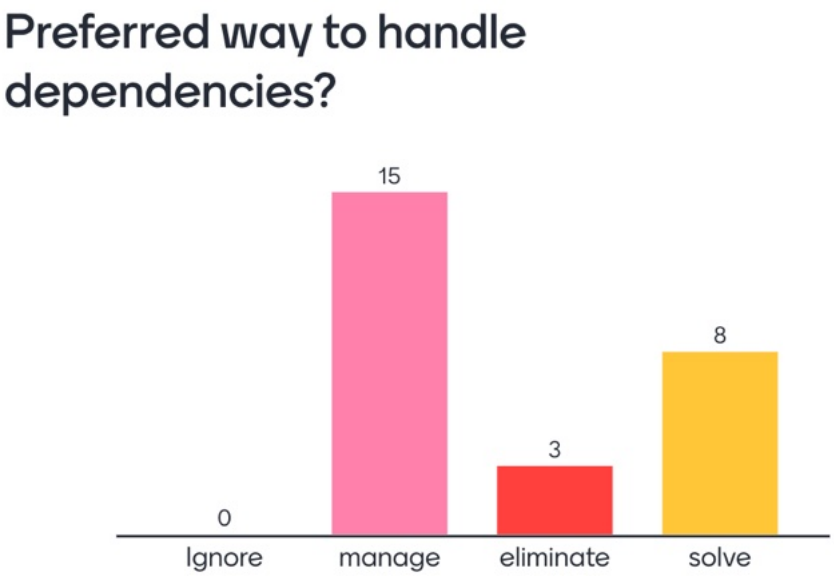
Product Vision, Common Values & Principles

But this creates many dependencies!

How should we handle dependencies?

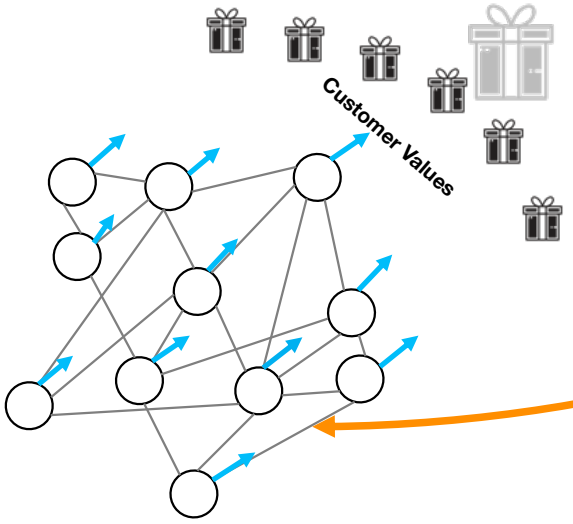
How Should We Handle Dependencies?

https://www.menti.com/2te6ufonez



We Can Reduce Dependencies & Complexity…

…by organizing around Customer Value



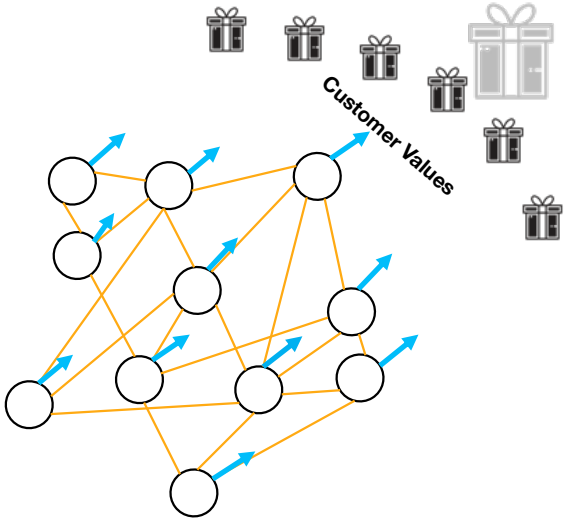
This leads us to a Product Organization and Feature Teams.

We decouple Feature Teams dependencies By splitting up customer values. This gives a team autonomy.

But what are these remaining dependencies?

Dependencies…

…in a Customer Centric product organization come from the real product (technical).



Feature Teams work in parallel to create the product. By integrating their work results into the product they might affect other team's work.

Continuous Integration and fast feedback loop make these affects transparent.

Continuous Integration ensures Alignment while Keeping the teams self-organized/self-managed.

Barriers for integration are barriers for coordination and therefore, for Alignment. So…

What Is Continous Integration?

https://www.menti.com/2te6ufonez

What is Continuous Integration? Use key words only.
76 responses

Continuous Integration Is An Employee Behavior

It means that working results on all levels will be integrated in the target product as fast as possible.

Continuous Integration Has Far Reaching Implications

Some Good Practices To Keep The Balance

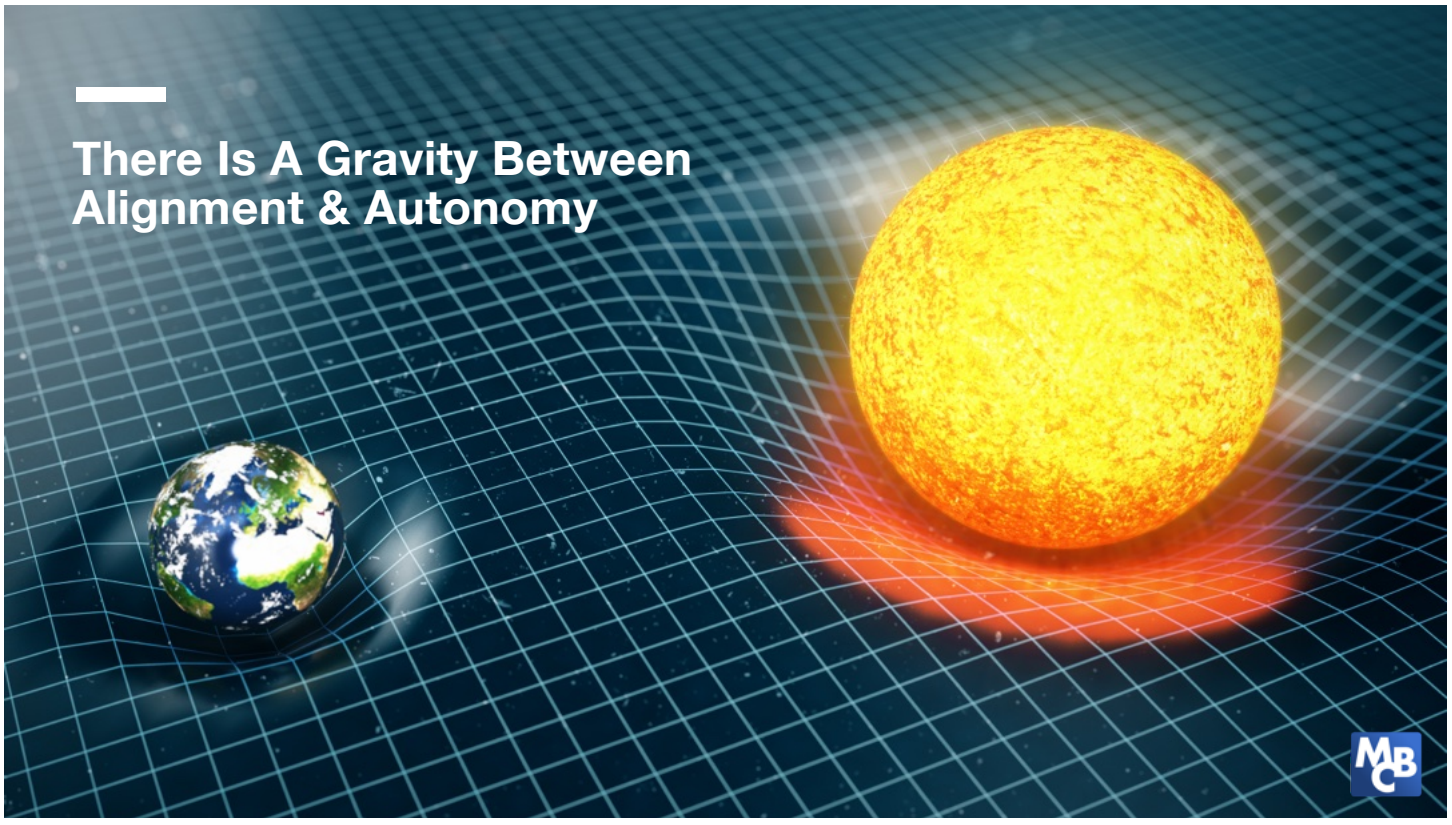
Organizational Design
- Build Product Groups/ decouple Org. from architecture
  Organize by Cust. Value/ establish Feature Teams
- Enable business/functional career paths parallel
  to hierarchical career paths

Business (Analysis)
- Business analysis in the teams
- From up-front to incremental analysis

Budgeting
- Teams as lowest level of resource planning
- Budgeting product group not features or items
- More information:
  - Beyond Budgeting https://bbrt.org/
  - Silicon Valley Budgeting

(Development) practices
- Coordinating through integration
- Continuous Refactoring, TDD, ATDD
  High level of (test) automation
- Fast Feedback with the build system
- Build system reflects the business view
- Use Communities

Infrastructure
- Common development environment
  and shared Build Pipeline
- Multiple test environments (virtually)
- Team rooms & rooms for multi-team events
  with plenty of wall spaces
- Remote work:
  - Video conferencing tool
  - Electronic whiteboards



Some Bad Practices To Avoid

Organizational Design
- Specialized groups for e.g. UX, architecture, analysis, integration, testing…
- Component teams only
- Item (Epic, Feature) owner
- IT PO
- Quality Manager (Roles)

Business (Analysis)
- Analyze the whole Product Backlog or features at once
- Separate business analysis in front of dev teams

Budgeting
- Measure progress on the used budget
- Budgeting of single tiny features
- Budgeting of individuals or single teams

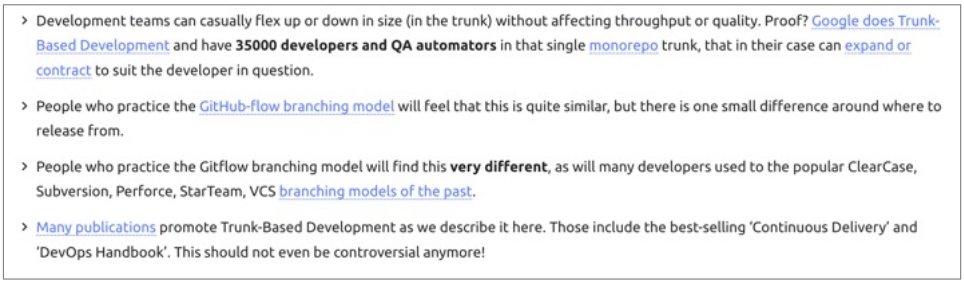
(Development) practices
- Coordinating for integration
- (Feature) branches > 2h
- Pull requests before integrating
- Hardening sprints
- Asynchronous integration points (Sprints)
- Playing the contract game

Infrastructure
- No shared build system pipe
- Each team or team member has their own special development environment

No Feature Branches?

Screenshots from Webpages
ATLASSIAN
&
Trunk-Based-Development

https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow

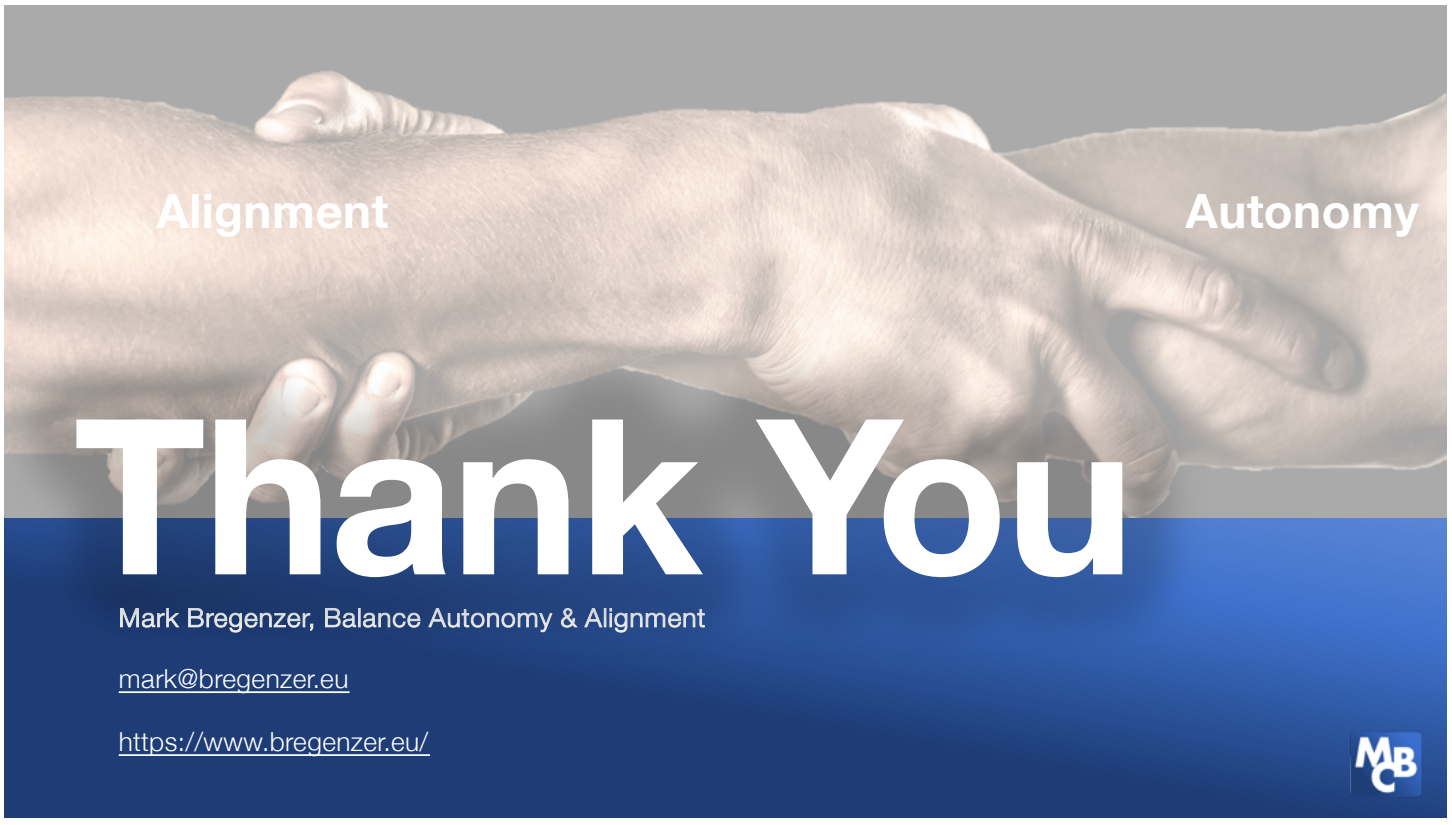


https://trunkbaseddevelopment.com/#trunk-based-development-for-smaller-teams

Keep in Mind

Every planning activity to achieve alignment is speculative.

Alignment via Continuous Integration is empirical.



Thank You

Mark Bregenzer, Balance Autonomy & Alignment

mark@bregenzer.eu

https://www.bregenzer.eu/