Course "Softwareprozesse"

Self-organization

Lutz Prechelt
Freie Universität Berlin, Institut für Informatik

- Definition of self-organization
- Common self-organization roles
  - Mentor, Coordinator, Translator, Champion, Promoter, Terminator
- Actual roles are unique
- Agreement on roles is critical
  - Role clarity
- Empowering teams can be highly beneficial
  - Case study: Team-driven deployment

Learning objectives

- What is self-organization, really?
- Standard roles needed in (many) agile teams
- Uniqueness of roles
  - Importance of agreement on roles
- Possible results of empowerment

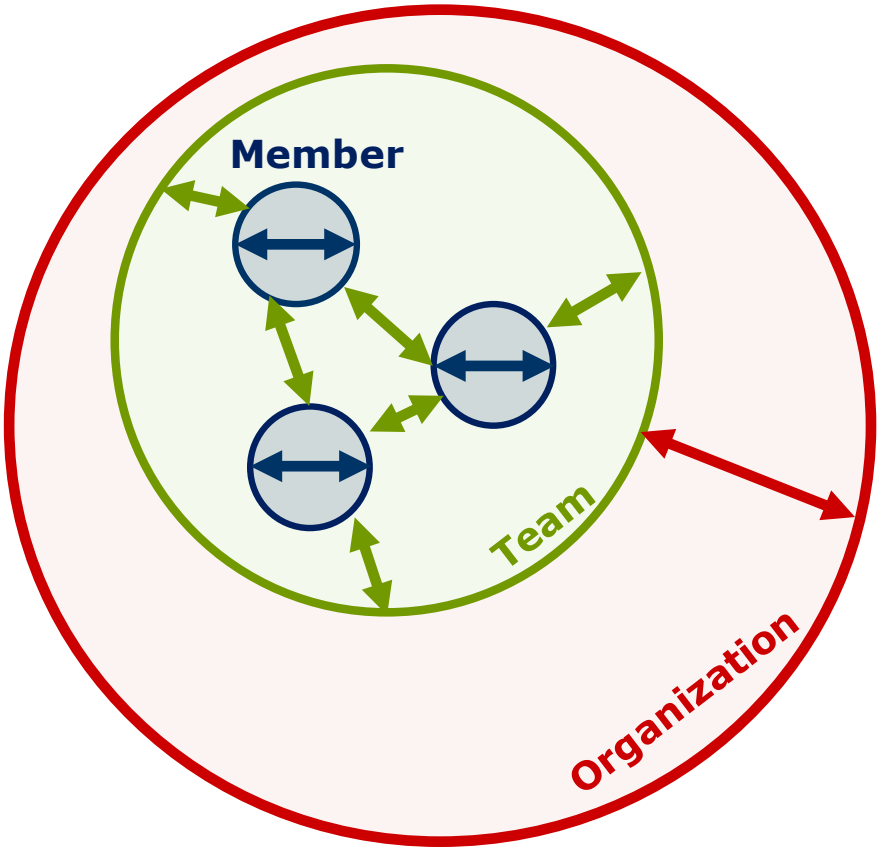
I. What is self-organization?

- Scrum Guide (p.5):
  - "*Scrum Teams are cross-functional, meaning the members have all the skills necessary to create value each Sprint.*
  - *They are also self-managing, meaning they internally decide who does what, when, and how.*"

- Elements of self-organization:
  - Having the required capabilities
  - Making decisions and behaving suitably
    - Actually behaving! Not only be able to.

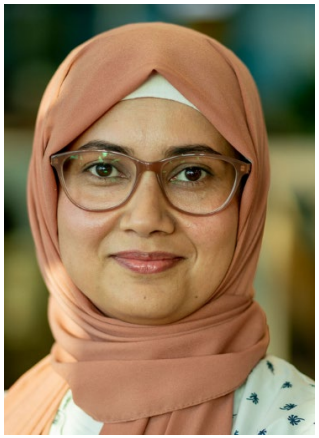
- Hackman 1986: "The psychology of self-management in organizations"
  1. Self-managing: Managing the work process
  2. Self-designing: Designing the performing unit [the team]
  3. Self-governing: Setting overall direction

Domains of self-organization



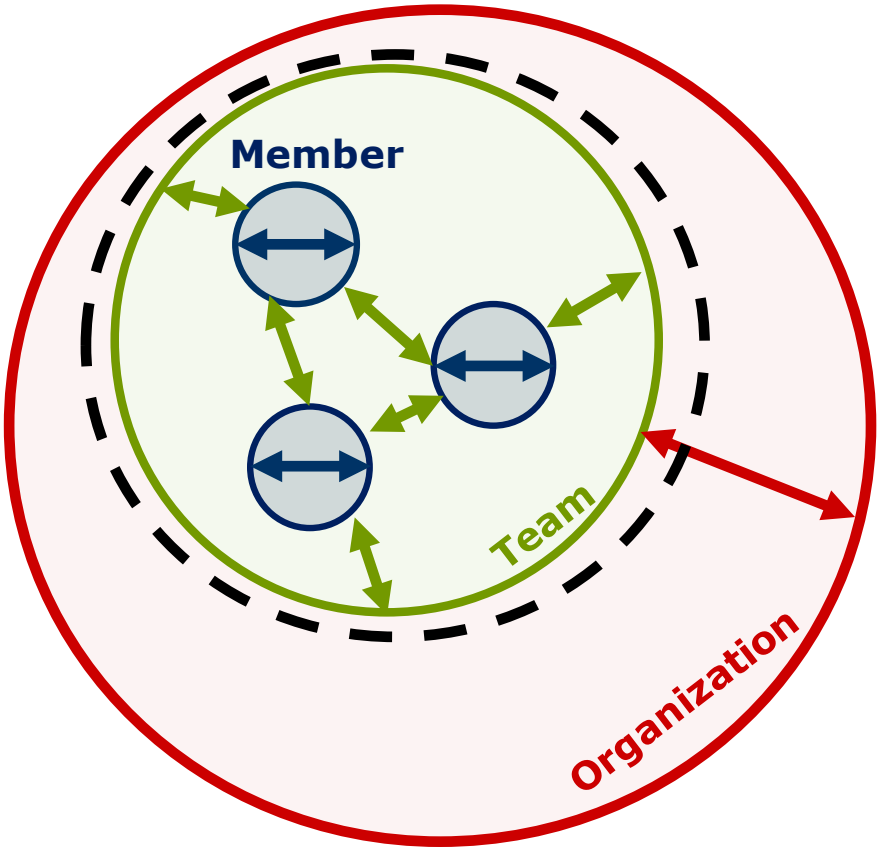
Self-organization relationships

II. Implicit, standard roles for self-organization

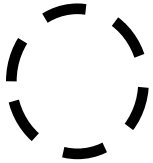


- Rashina Hoda, James Noble, Stuart Marshall: "Self-Organizing Roles on Agile SW Development Teams", IEEE Trans. on SW Engineering 2013
  - based on interviews with 58 members of 23 very diverse organizations in New Zealand, India, North America
  - finds "*six informal, implicit, transient, and spontaneous roles on Agile SW development teams that enable self-organization*"
  - These roles are implicit, not named functions or job titles

Topic:
Completeness of team behaviors



Self-organization relationships



Topic area of study

Hoda's 6 roles: Mentor, Coordinator

Freie Universität Berlin

We discuss: What's missing without it? Who will fill it? How likely to have it?

1. Mentor: encourages and supports agile practices and self-organization
   - e.g. encouraging task self-assignment: "*It took [the new Agile team] a bit of time to stop coming and asking us what they should be working on and the answer was always "pick one!" And after while it became natural [...] people were picking stuff [...] and that worked really well.*"—P25, Developer, NZ

2. Coordinator: manages customer expectations, coordinates customer collaboration
   - e.g. "*it makes sense to have a [Co-ordinator] in the middle [...] if you have some sort of problem, you don't have five people asking the same question at the other end; which normally business people don't like.*"—P1, Developer, NZ

Hoda's 6 roles: Translator, Champion

Freie Universität Berlin

We discuss: What's missing without it? Who will fill it? How likely to have it?

3. Translator: translates from and to business language
   - e.g. "*I might explain something in a very cryptic, technological way and [the customers] won't understand a word!*"—P2, Developer, NZ
   - e.g. "*understand the concerns of the business and translate them into priority elements that the development group can actually focus on to achieve*"—P20, Senior Agile Coach, NZ

4. Champion: organizes senior-management support for agile
   - e.g. "*executives are not the enemy; they're your best allies. […] they're not the ones who prevent you from doing stuff, they just don't know any better! [So you should] treat them as such. Educate them, gently.*"—P20, Senior Agile Coach, NZ

Hoda's 6 roles: Promoter, Terminator

Freie Universität Berlin

We discuss: What's missing without it? Who will fill it? How likely to have it?

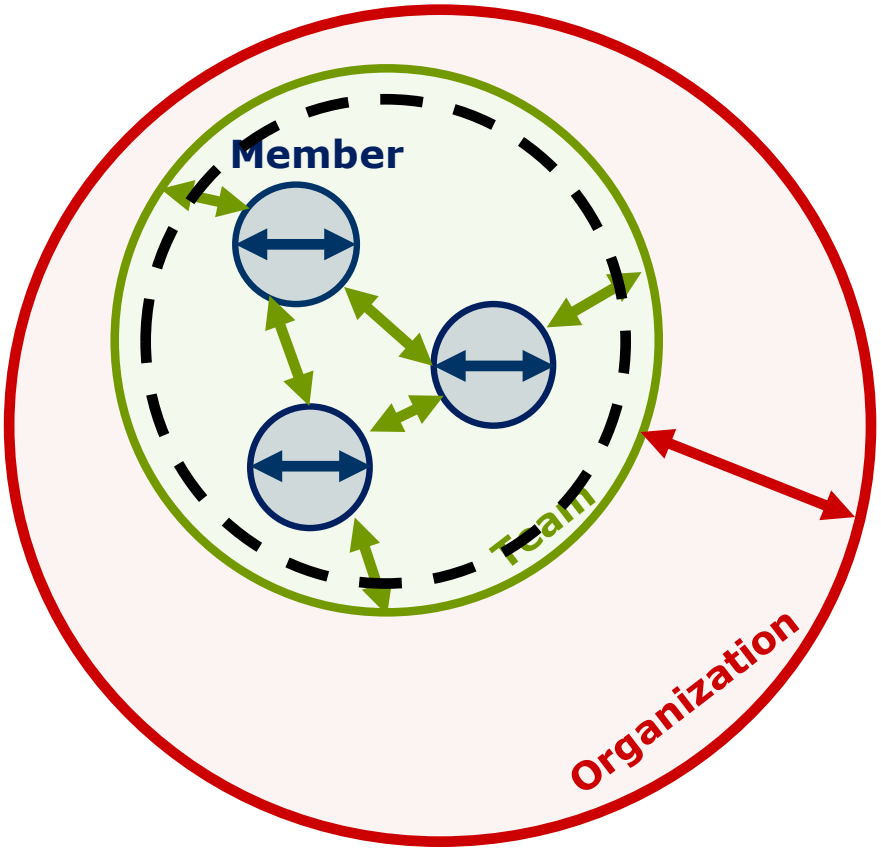
5. Promoter: promotes customers' collaboration
   - e.g.: "*I knew nothing about Agile so it was like* 'What the hell is Agile?' *and I got a brief overview and I thought* 'That seems remarkably sensible […]'. *[Next] I was like* 'What is Scrum and why do you need a coach?'"—P9, Customer Representative, NZ

6. Terminator: identifies team members who threaten the team's functioning and works to get them removed
   - e.g. "someone who isn't willing to learn and […] communicate […] they can wreck the project very very quickly. […]
     Any one of them that can't adjust to the Agile mechanism really needs to be removed pretty quickly […]
     It's not that the person is bad, they may be very very good at their job, it's just that they can't adjust to the different mechanism [of working]."—P10, Agile Coach, NZ

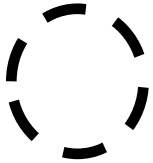
III. Implicit, unique roles for self-organization

- Helena Barke, Lutz Prechelt: "Role clarity deficiencies can wreck agile teams", PeerJ 2019
  - based on field observations + interviews from 5 very diverse agile teams
  - Finds that roles are unique: Tailored to one individual, team, and situation.
  - "Complying with a Role requires to
    - Accept the responsibility (R1-Accept),
    - Possess expertise (R2-Expertise), and
    - Act accordingly (R3-Acting)"
  - Creating the roles is a key part of self-organization
  - "The responsibility" can be anything
    - eventually something that is needed and possible
    - but initially perhaps not!

Topic:
Self/team relationship



Self-organization relationships

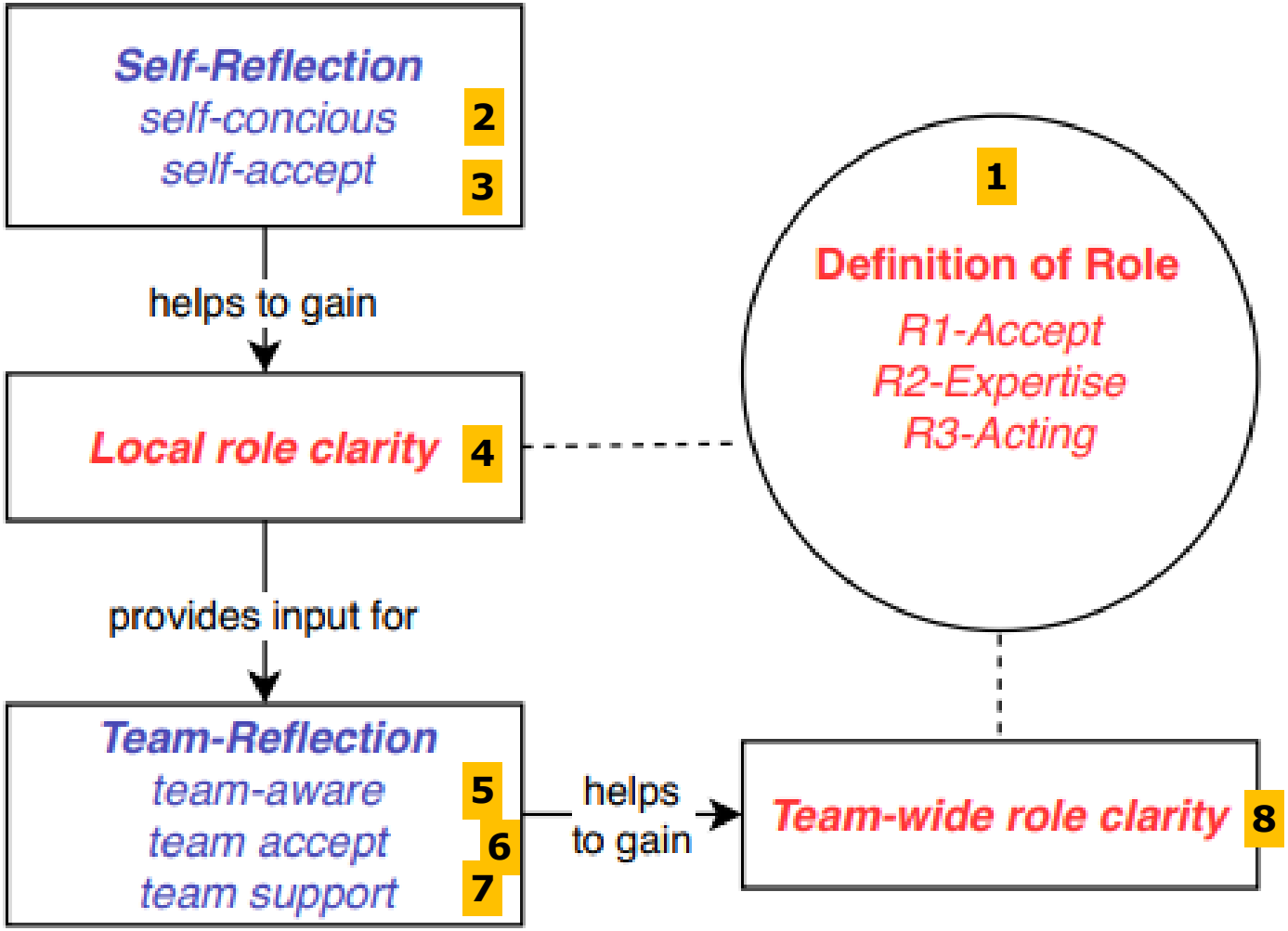


Topic area of study

The role-clarity theory

5, 6, 7 can happen before 2, 3

Deficits with 3 or 6 create strong conflict



Example 1: no role at all

- *"I feel like a babysitter. [...] we had just understood our roles, then he (*t2test3*) came into the team.*
  We do not know his role, he does not know his role.
  I encourage him to talk, but he has a different personality:
  I try to find my place by myself; he wants somebody to do it for him." (t2test1)

- General lack of t2test3's self-reflection about his role
  - Result: team frustration

Example 2: lack of coordination

- t2dev2 sometimes implements functionality on his own decision; the team gets angry:
  - "*He does things without telling anybody. […] Excuse me? How about involving us in that decision?? You force this upon us -- that's not so cool.*" (t2dev1)
  - Mediation from t2po helped: *"his attitude changed towards how one should work -- as the team expects of him."*

- Either a lack of self-reflection or a lack of self-acceptance.
  - Result: Lack of team-wide role clarity, conflict.

Example 3: distorted self-perception

- "*For him, self-image and how others perceive him are a bit different. When I give him feedback that something was not good, he is surprised:* "I thought I am the super-high performer."
  I told him to ask others as well and adjust his self-image or his performance. And indeed he tries to improve -- but it does not help: The next time around he is surprised again."
  (t5sm1)

- The available level of R2-expertise lacks self-reflection.
  - Result: Dissatisfaction

Example 4: would-be team lead

- t4dev3 wants to become team lead and assign tasks to others
  - Management denies him such a role.
- He starts to consciously disrupt meetings
  - Result: "*Everybody […] felt bad with the situation; they almost did no longer want to go to work in the mornings.*"
- When t4sm2 confronts the behavior in a retrospective, t4dev3 explains it.
  - Result: *"talking helped. It did not resolve the situation or change* t4dev3*'s behavior, but the others no longer took it personally and could cope with it with more confidently."*

- Team-wide clarity about acceptable roles
  - Result: The team is robust against the dysfunctional behavior.
  - t4dev3 left the company shortly thereafter.

Example 5: emotions

Freie Universität Berlin

- t5dev1 is very unhappy with how t5dev2 acts [not described here], but does not like to talk about feelings.
  - Eventually, he took the courage to unannouncedly visit t5dev2 at home, invite him to a beer and "*confess*" (t5dev1) his feelings.
  - t5dev2 was aware of his own behavior, but surprised by t5dev1's anger about it.
  - They agree on a future behavior.
- t5dev1 does not follow up the resolution; falls silent again.
- Eventually he has an emotional outburst. Leaves the team and even the company.
  - t5dev1 and t5dev2 are both key technical people.
  - Over the course of a few months, the whole team dissolves. All-but-one members also leave the company.

- Team-accept never achieved
  - Result: Danger for the whole team!

Example 6: clarity achieved

- t4sm2 about a former team of his:
  - "*They had found each other as a team. They could rely on each other. It was clear who could do what.*
  - *One writes plenty of code, another thinks ahead, another is good at diplomacy and negotiating with the product owner, and yet another is more like "I'll run and get us person so-and-so.".*
  - *Without speaking, they knew who had which skills and who would pick up which things to be done. No need for much coordination, it arose dynamically just so.*
  - *Five strong personalities. Initially, they had had a very strong storming phase, real quarrels about how they would work. But then: fantastic productivity and fun at work.*"

- Unique roles + team-wide role clarity
  - Result: High satisfaction, productive work

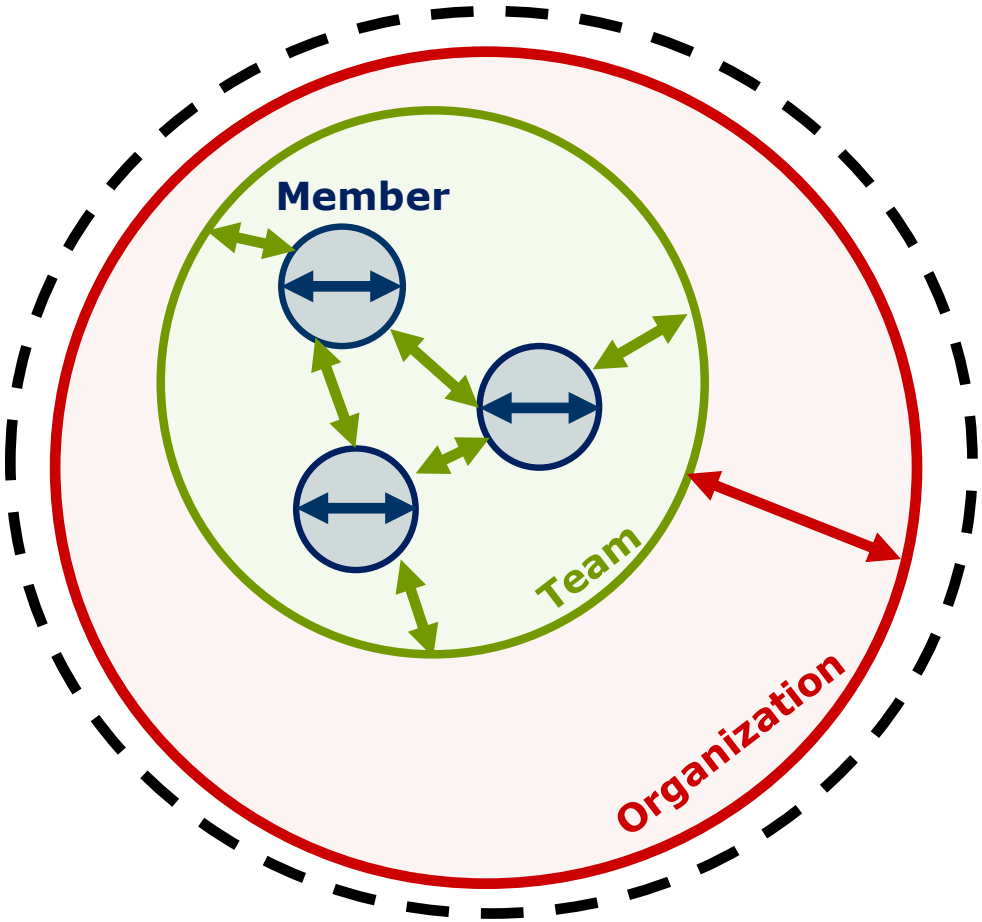
Yeah!

IV. Empowerment: Team-in-Organization self-organization

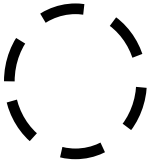
Freie Universität Berlin

- Lutz Prechelt, Holger Schmeisky, Franz Zieris: "Quality Experience: A Grounded Theory of Successful Agile Projects Without Dedicated Testers", Int'l. Conf. on SW Engineering 2016
  - based on field observations and interviews in 3 agile teams, 1 with testers, 2&3 without (1&2 from the same organization)
  - Finds that quality assurance based on frequent deployments (rather than extensive manual testing) has many benefits
    - but requires a highly modular architecture and trust from the organization

Topic: Organizational outcomes



Self-organization relationships



Topic area of study

"Quality Experience"

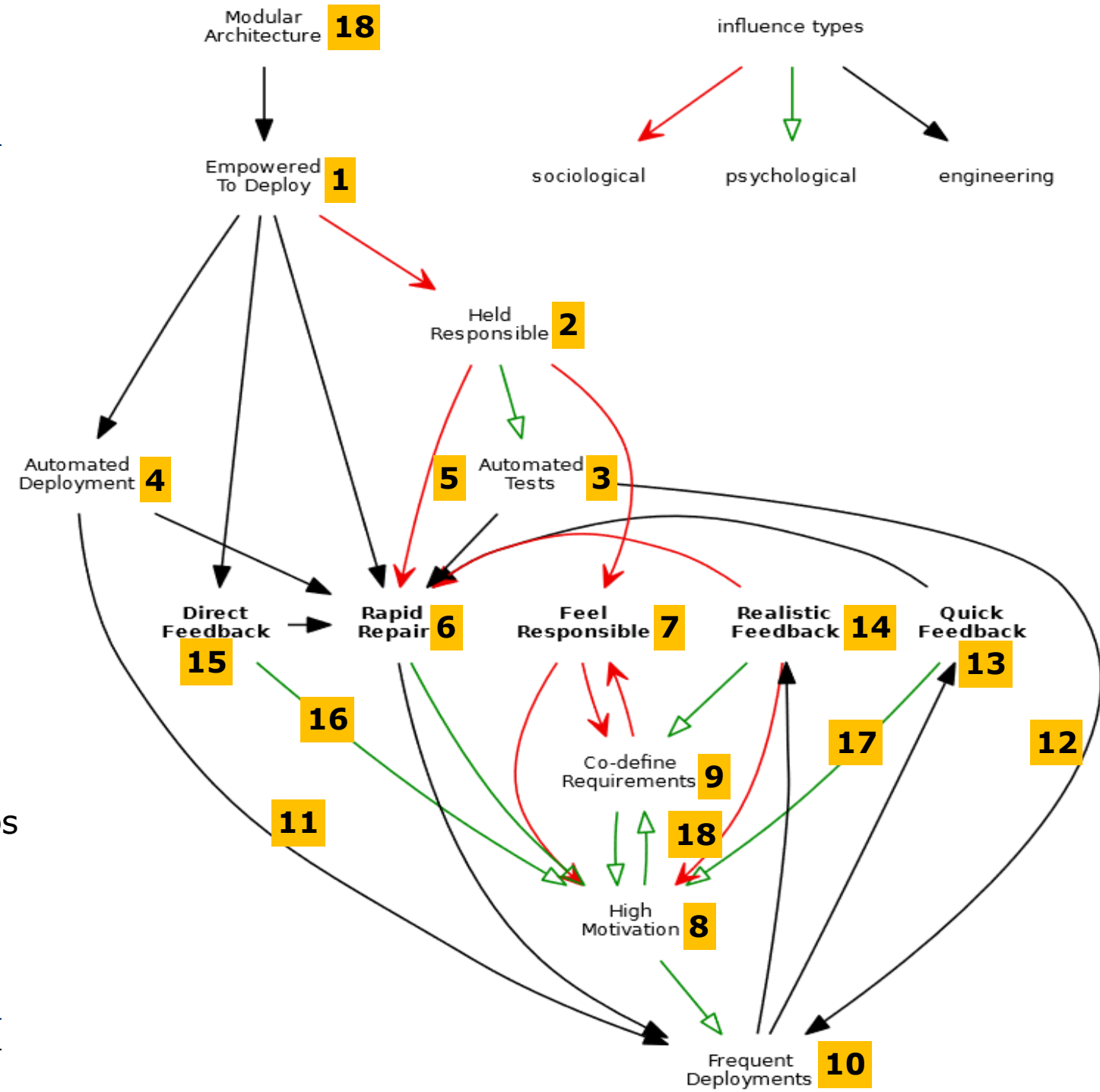
Starting point:

- Team gets empowered to deploy versions

Outcomes:

- Rapid repair of problems
- Frequent deployments, rapid progress

(some relationships are not shown)



Interpretation in terms of the Agile Manifesto

- The organization actively relates to the team in steps 1 & 2
  - 1:Empowered to deploy
  - 2:Held responsible
  - "5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done."

- Then various agile principles are at work:
  - "2. Welcome changing requirements"
  - "3. Deliver working software frequently"
  - "7. Working SW is the primary measure of progress."

- But the whole thing cannot work without a clean architecture:
  - "9. Continuous attention to technical excellence and good design enhances agility."

Summary

- I. Self-organization means to be
  - self-managing (tactical behavior) and
  - self-designing (creating a process for the team)
- II. Several roles are often needed for self-org:
  - Mentor (team), coordinator (team/customer), translator (team/customer), champion (mgmt), promoter (customer), terminator (team/mgmt)
- III. Actual roles are tailor-made and unique
  - Non-agreement on roles produces strong conflict
- IV. Empowering teams can create big improvements for the organization

Thank you!