



# Long-term Perspective of Agile Methods

**Mira Kajko-Mattsson**

Stockholm University/KTH

- Hermann Kaindl, Vienna University of Technology
- Ken Boness, University of Reading
- Rob Pooley, Heriot Watt University
- Andreas Tael, Mejsla AB
- Ademar Aguiar, INESC Porto & FEUP/DEI



# Short-term Benefits of Agile Methods

- More accurate visibility into and control of projects,
- Better management of constantly changing requirements,
- Early detection of all kinds of problems,
- Better adherence to customer requirements,
- More efficient and cost-effective acceptance testing,
- Substantial reduction of the overall risks associated with software development.



# “Agile” Methods

- Hermann Kaindl  
Vienna Univ. of Technology, ICT,  
Austria



# Question

- If so-called agile methods are the answer, what has been the question?
- 'Heavy-weight' methods
- ISO 9000, CMM or CMMI



# Less Documentation?

- “Lose weight” by reducing documentation
- User stories instead of a requirements specification (including models)?



# Future

- It is hard to predict, especially the future!
- Will all software be developed in the future according to agile methods?
- I don't think so.
- Better to apply such a method than none at all
- Iterative and incremental development has been and will be applied before and after the rise and fall of agile methods.
- There will be new hypes!



**Thank you for your attention!**

???



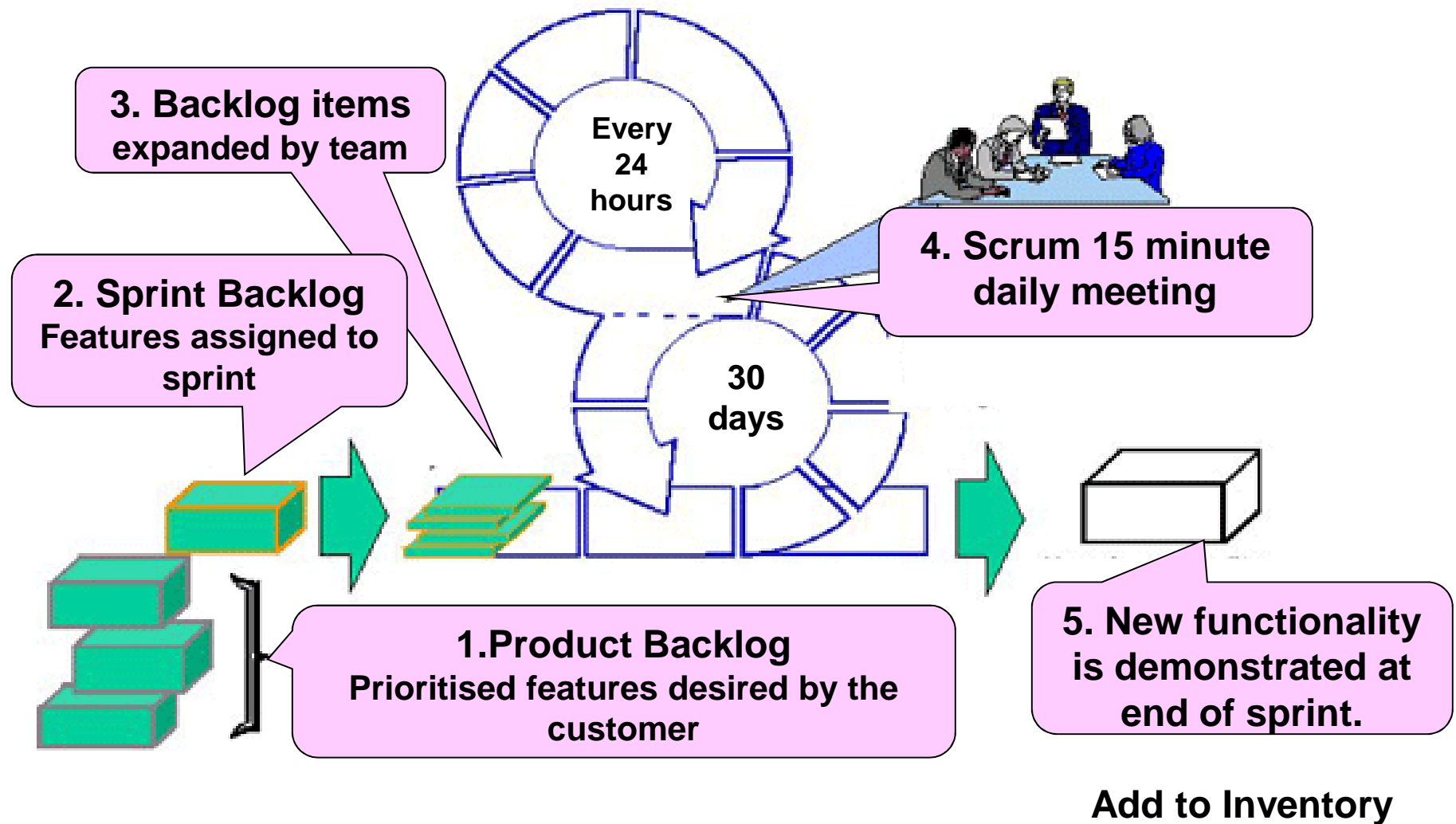
# Agile ICSEA 2009

Ken Boness



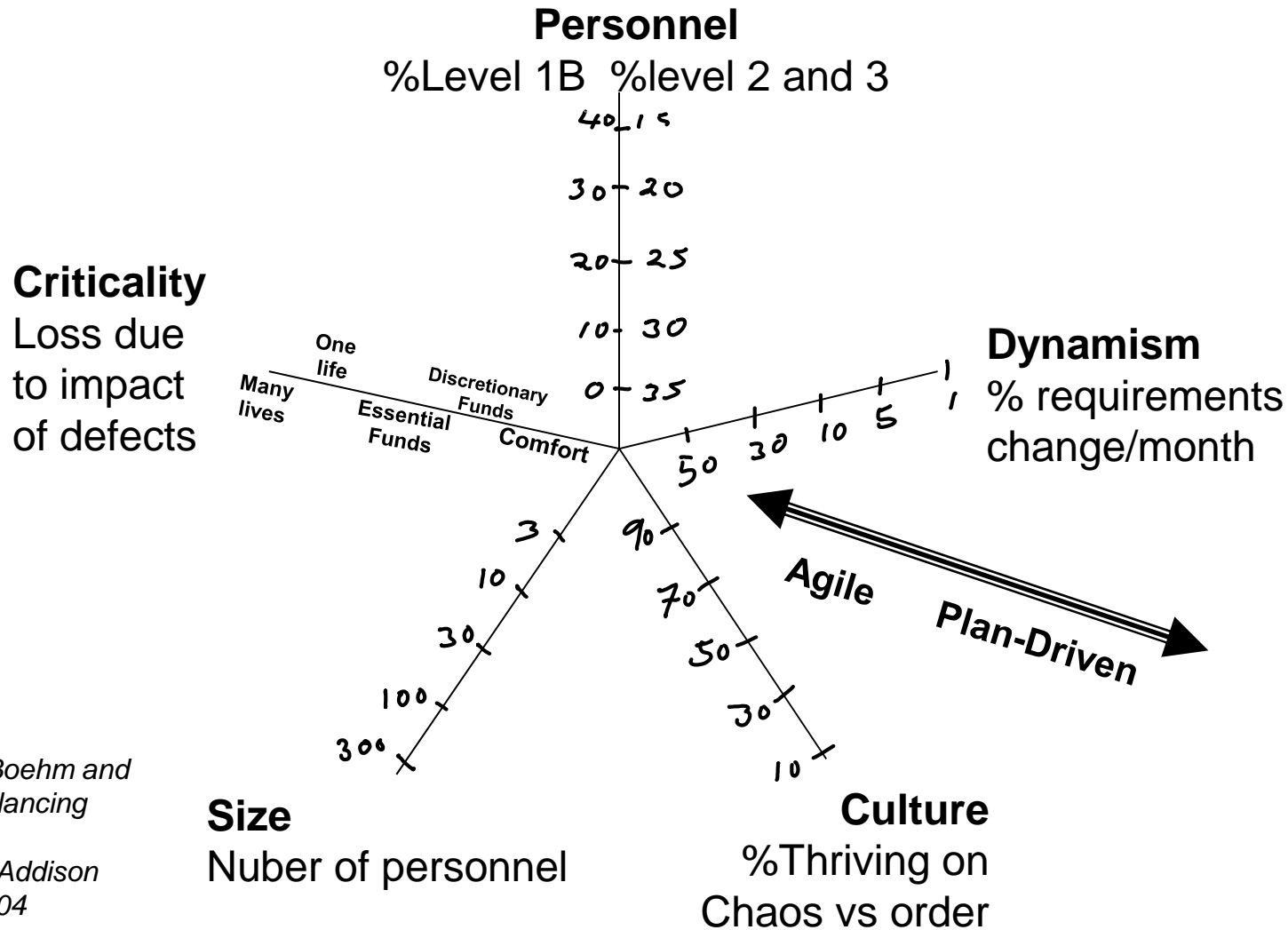


# An Agile Approach - Scrum





# Home Ground



Based on Boehm and Turner, "Balancing Agility and Discipline", Addison Wesley, 2004



# Perspectives

## ■ Vincenti

- Radical and Normative Engineering
  - ◆ Good enough to improve
  - ◆ Follow best practice recipe

## ■ Action Research

- Learning cycle
  - ◆ Try – learn - improve



# Psychological Issues in Agile

## ■ The deal

- ♦ Stability during each sprint/timebox to **complete** work.
- ♦ Fidelity to change specifications at each sprint boundary.

## ■ Happy syndrome

- ♦ Testers can provide evidence on whether the work is complete and correct; and
- ♦ Realistic (feasible) estimates of what can be completed.

## ● Inducing

- ♦ Product Management get predictability in return for reasonable patience; positively encouraged to play the game.
- ♦ Confidence and success reinforcing the deal.

Ctd...



# Psychological Issues (Ctd.)

## ■ Unhappy syndrome

- ◆ Testers cannot provide evidence whether the work is complete and correct; or
- ◆ Overoptimistic (infeasible) estimates of what can be completed.

## ● Inducing

- ◆ Developers do not complete work; this degenerates to iteration.
- ◆ Bad surprises when commitment milestones arrive.
- ◆ Loss of confidence and corruption of the deal; e.g. change requests abound.



# Confidence

- We must have **confidence** in the value chain.
- Confidence is predicated on **evidence**.
  - Key evidence comes from well conducted and reported verification and validation tests.
- If we are not confident about the completed stock of code:-
  - We have lost control of our navigation.
  - We cannot show increased company value.
  - We cannot deliver product without large cost risks.



# Summary

- There is a “home ground”
- Relates to lessons learned by Vincenti
- It resembles action research
- High degree of discipline
  - Rules can be simple
    - ◆ But must be obeyed!
- Psychology matters
- Puts great demands on the testing team



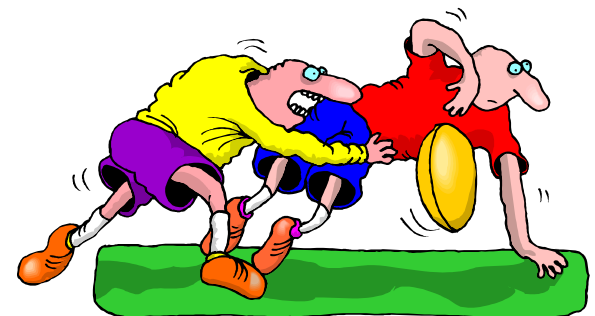
# Agile ICSEA 2009

Robert Pooley





- Agile methods are good in providing tight communications within a team working on a software project.
- Experience from the games industry
  - supports the popularity of agile methods in that sector, but
  - highlights some problems.





- The games industry runs projects where
  - all contributors need to stay “on message”
  - at all stages.
- preference is for Scrum, which emphasizes
  - cross technology team working
  - self defining planning, based on “sprints”
- This allows some of the weaknesses of more software focused agile methods to be reduced.





- These weaknesses include
  - a need to involve users continuously,
  - maintain a clear distinction between users and developers.
- In a development environment like games,
  - it is impossible to separate these roles
  - agility becomes rigidity if you are not careful.





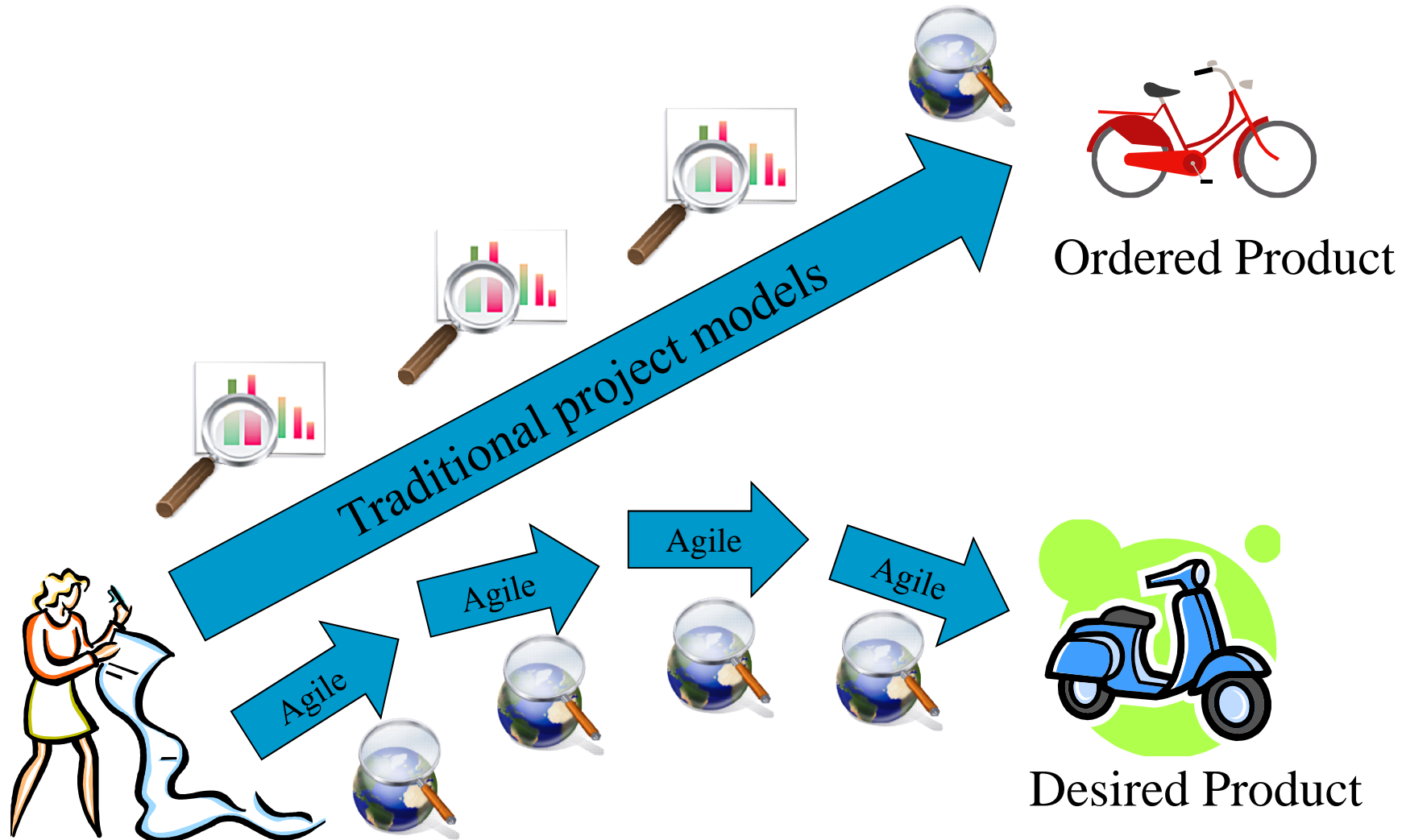
**Andreas Tael**

Mejsla AB  
Sweden



**“It’s better to be roughly right than precisely wrong”**

John Maynard Keynes





**LONG-TERM PERSPECTIVE OF AGILE METHODS**  
ICSEA 2009, SEPTEMBER 20-25, 2009, PORTO, PORTUGAL

**Ademar Aguiar**  
INESC Porto, FEUP/DEI  
Universidade do Porto

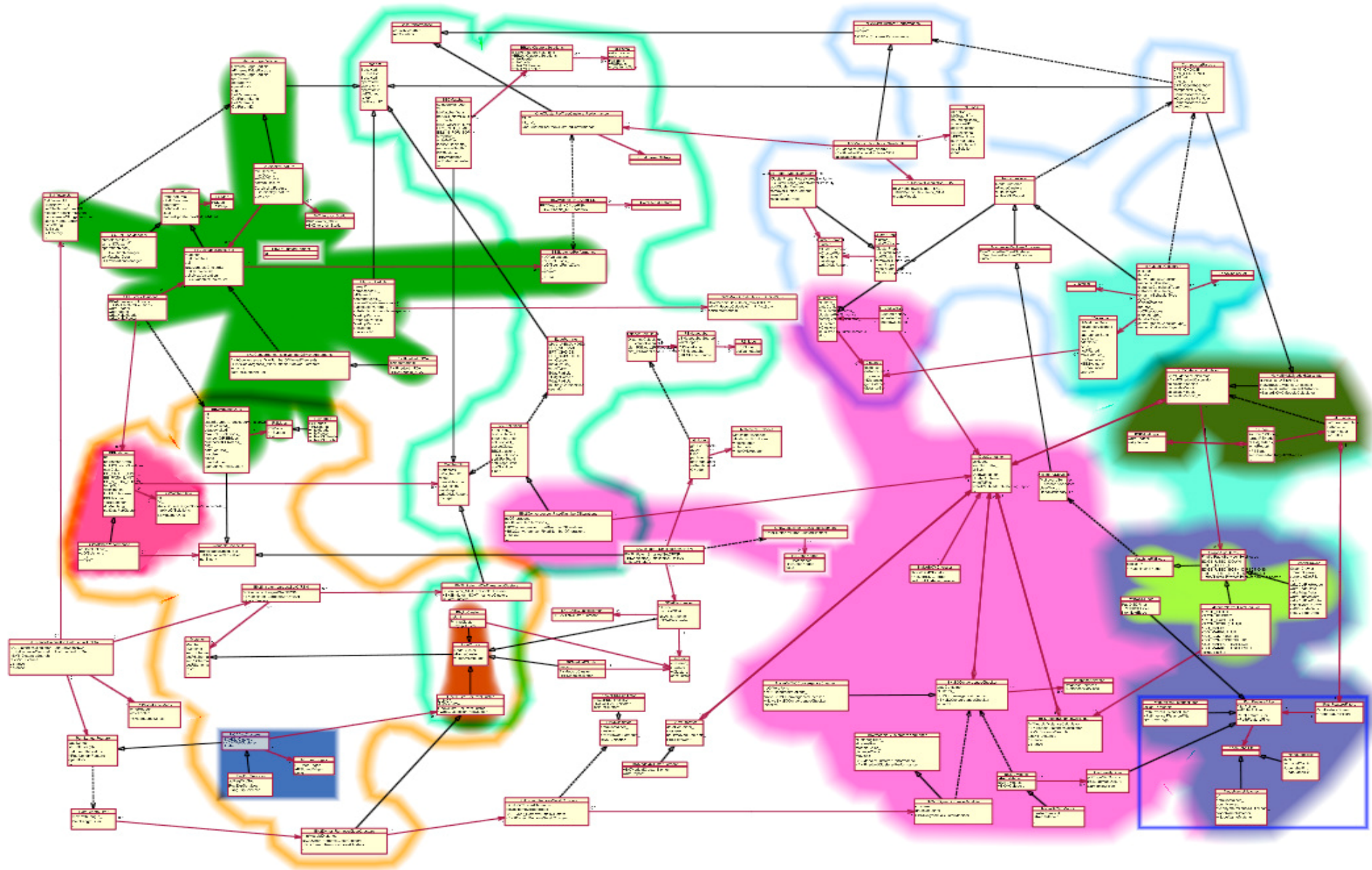


**“Things” Change Over Time...**



**“Live things” change more...**





**“Software things” change even more...**



# Types of software changes

- Requirements and Priorities
- Technology and Tools
- People and teams
- Interactions and behavior
- Software complexity and unpredictability
- ...

**Agility is important**  
to be able to respond  
with quick easy grace





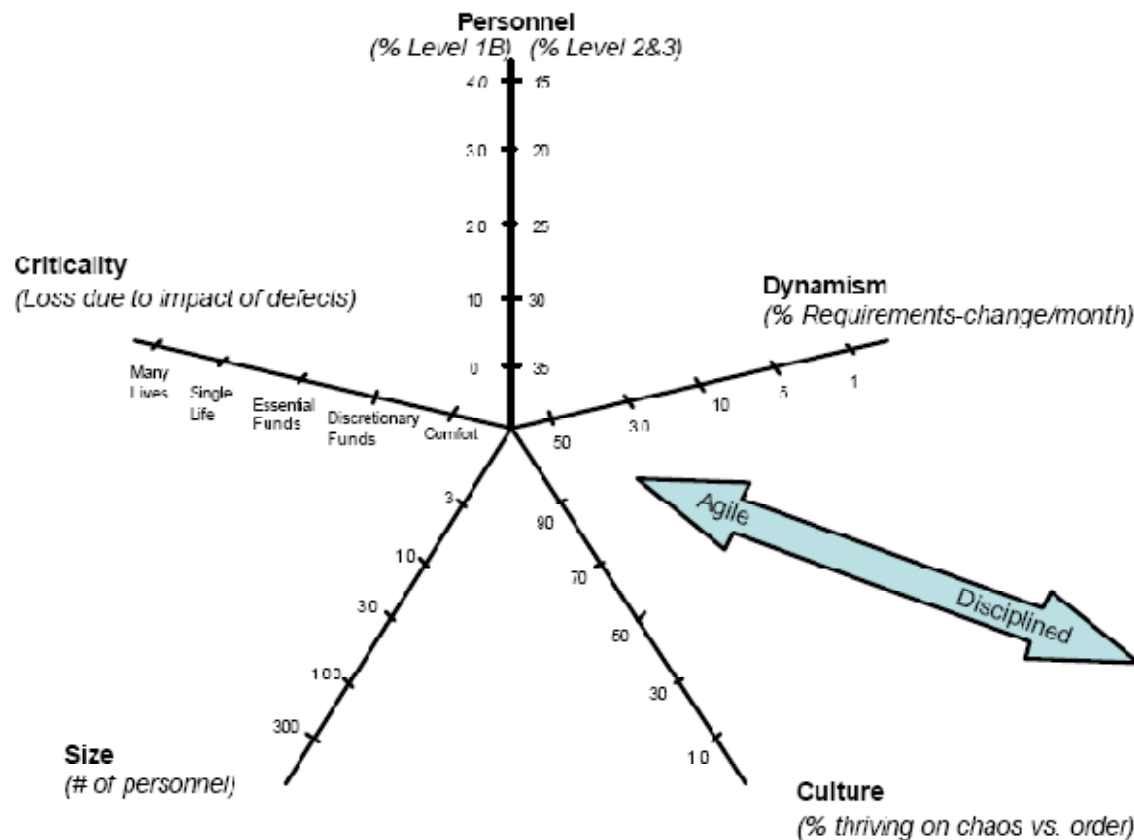
**“Software engineering** is the application of a **systematic, disciplined, quantifiable** approach to the development, operation and maintenance of software, i.e. the application of engineering to software.” [IEEE]

**Goals: high quality, high productivity, high predictability**



# Best Process

- The simplest process that ensures the optimal level of team's capabilities, discipline, communication and knowledge.





# Controversial Agile Practices?

- Pair programming
- Test Driven Development
- Refactoring
- Collective Ownership
- Continuous Integration
- The Planning Game
- Small Releases
- On-site Customer



# Short term perspective



Agile practices are apparently a waste of time and resources... but are cool! 😊



Software development is a knowledge intensive activity that requires a lot of social interaction.



# Long term perspective

As human factors are highly-valued by agile practices...

Agile practices are definitely worthy to consider! Try it!





Thank you

