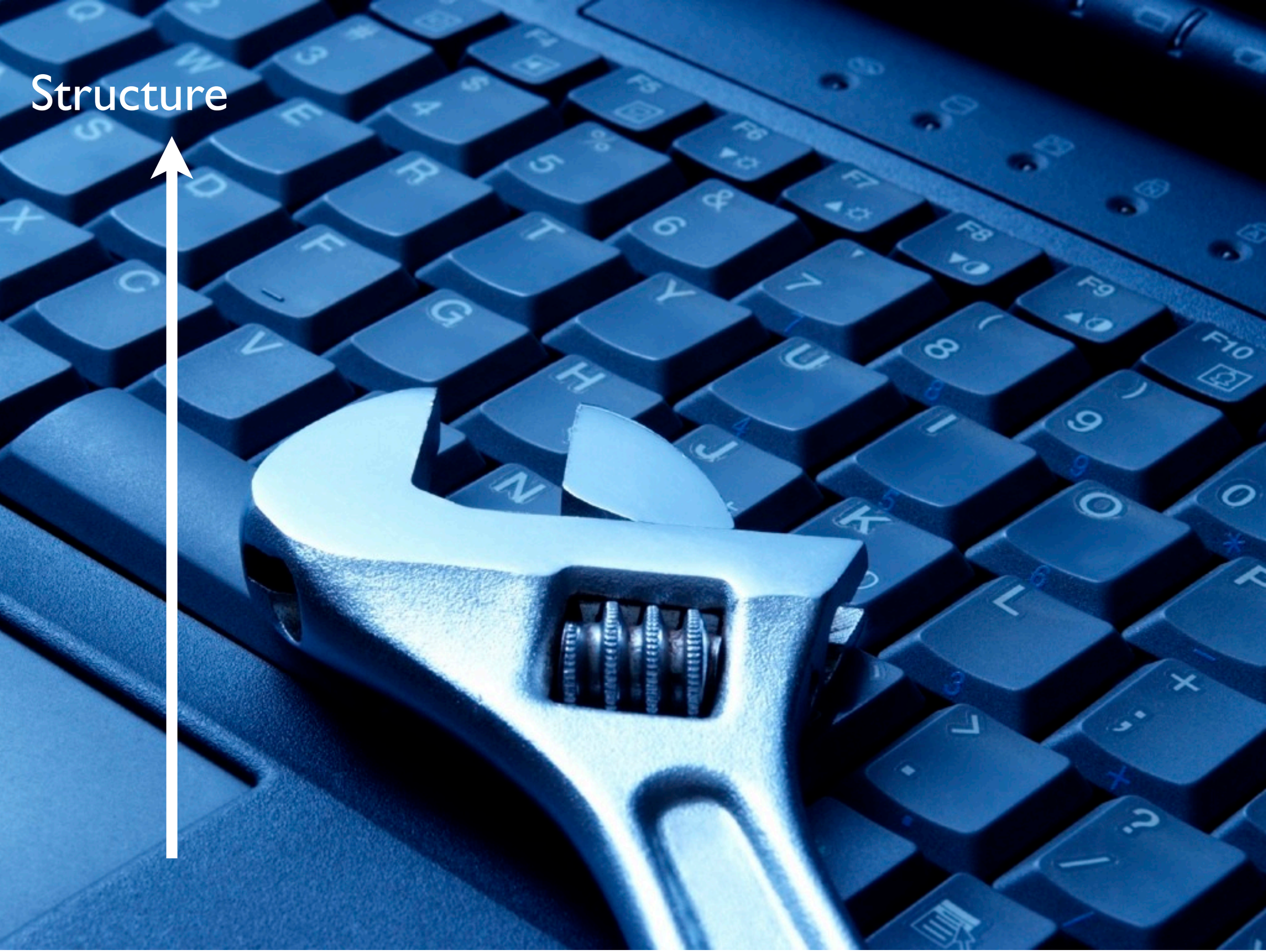
People, Refactoring

Lecture 3, EDA397/DIT191, Agile Dev Processes
Robert Feldt, 2011-04-11

måndag den 11 april 2011

Structure

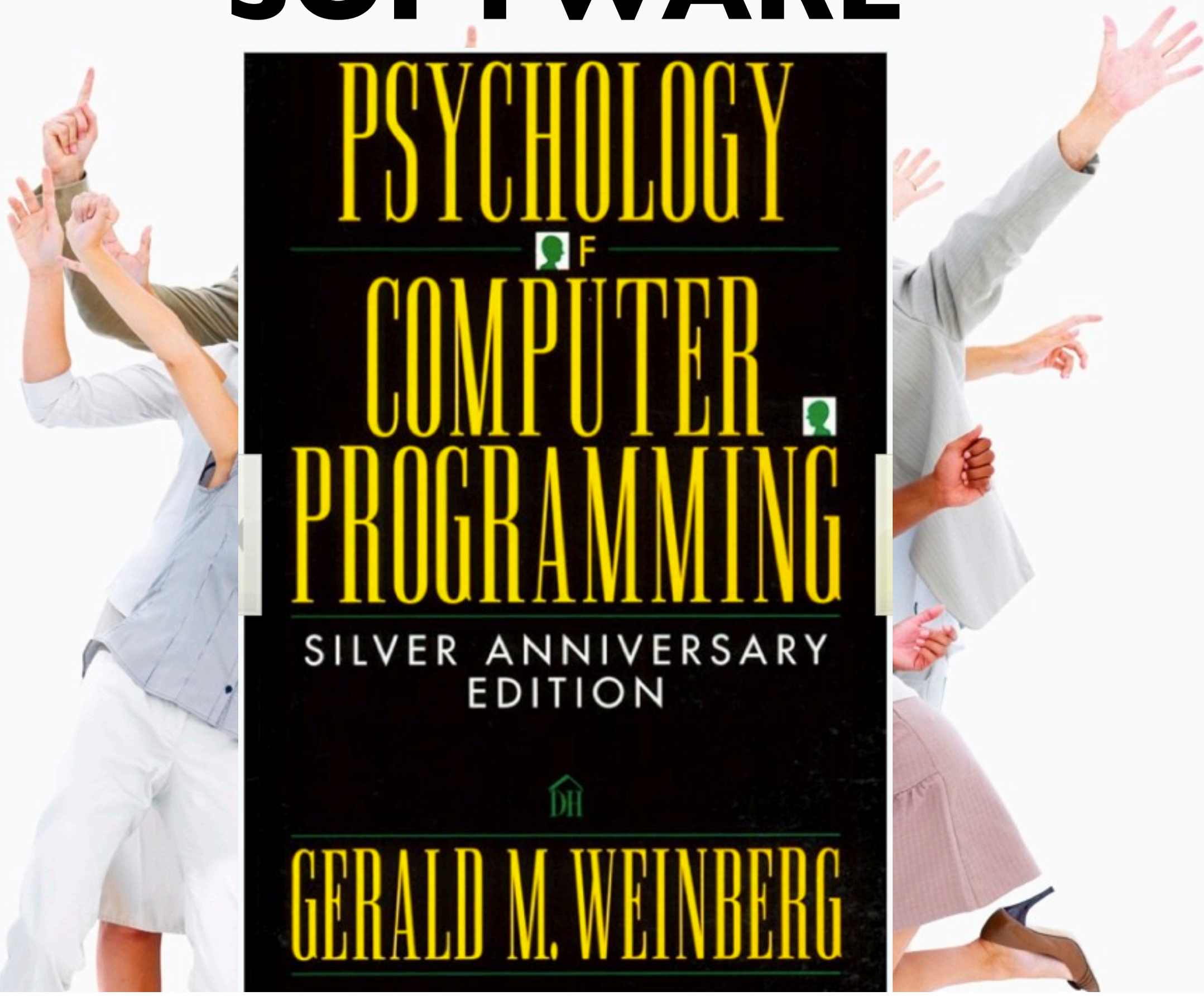


SOFTWARE

måndag den 11 april 2011

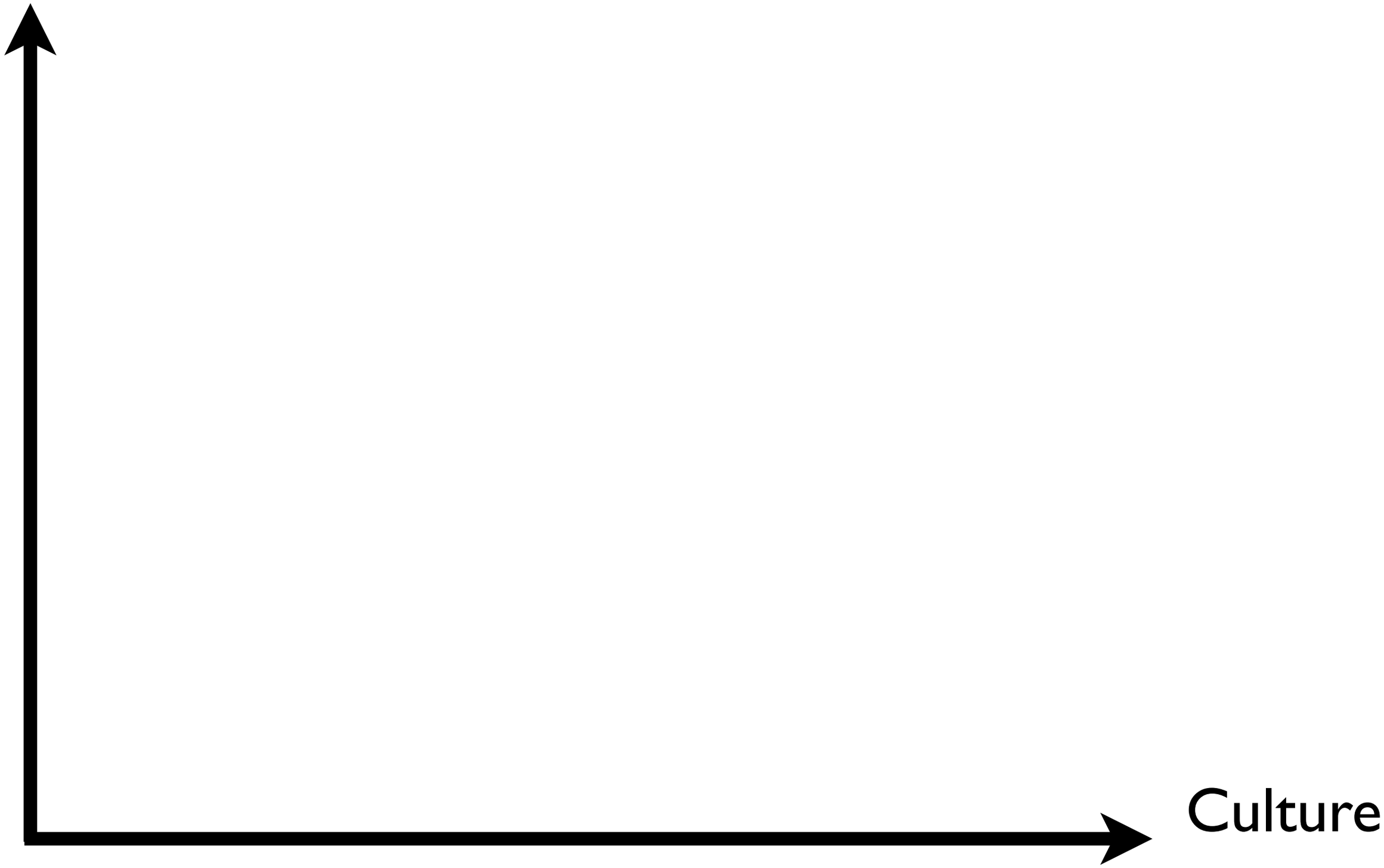
SOFTWARE

SOFTWARE

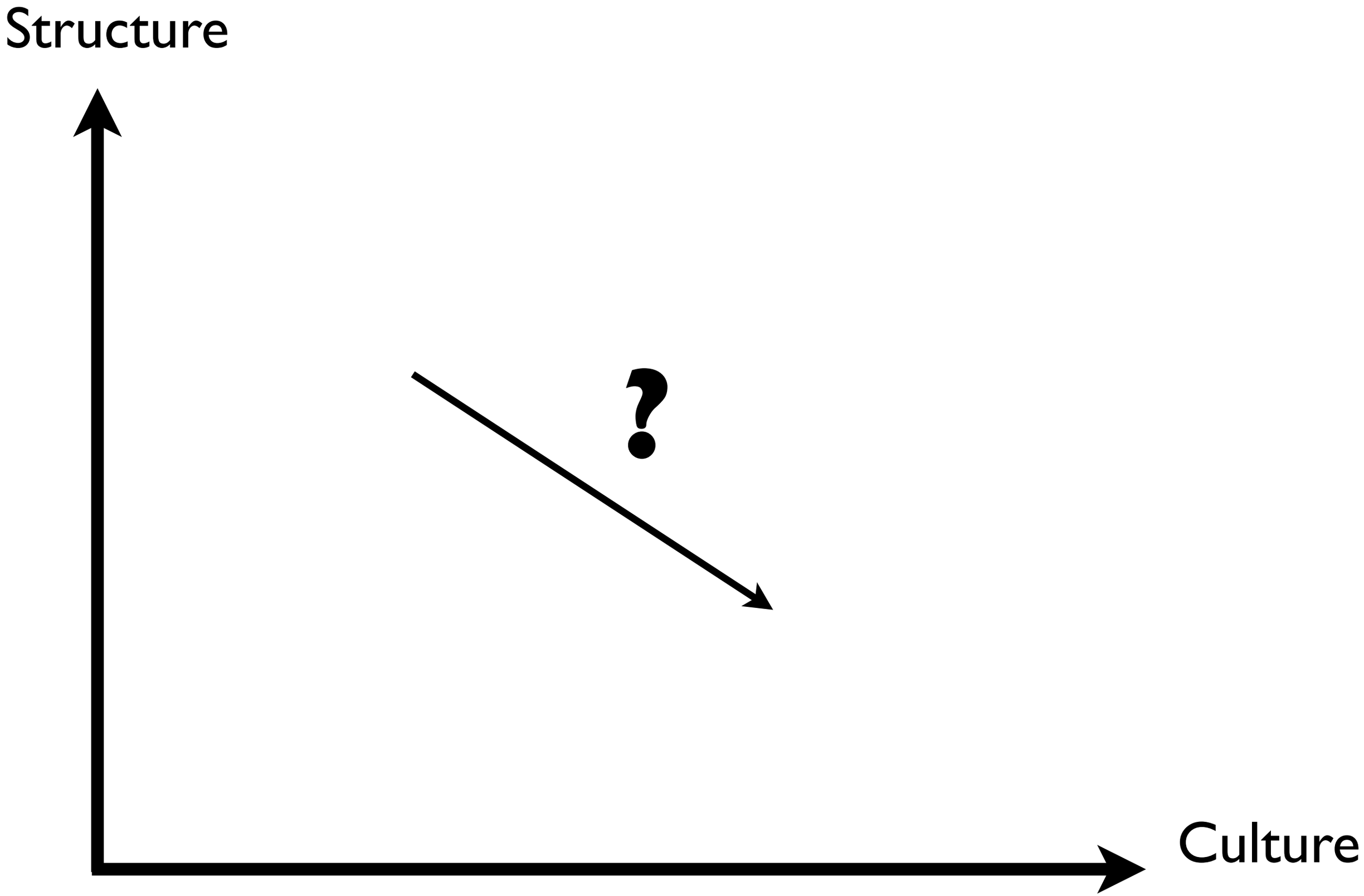


måndag den 11 april 2011

Structure



Culture


Structure
?
Culture

Structure

?

People Issues Trump
Process or Technology!?

Culture

Humans: funky & vary a lot

- Spontaneous
- Change, sometimes for no apparent reason
- Contradictory
- Full of personality
- Vary by hour, day, age, culture, temp...
- Solve problems differently
- Personal style affect others

Humans: funky & vary a lot

Diversity of methods work depending on people and their experience!

- Spontaneous
- Change, sometimes for no apparent reason
- Contradictory
- Full of personality
- Vary by hour, day, age, culture, temp...
- Solve problems differently
- Personal style affect others

Humans: funky & vary a lot

Diversity of methods work depending on people and their experience!

- Spontaneous
- Change, sometimes for no apparent reason
- Contradictory
- Full of personality

Variation => FEW general rules & always exceptions

- Vary by hour, day, age, culture, temp...
- Solve problems differently
- Personal style affect others

Technology's role

- Can **AUTOMATE**
  - Tedious tasks (Avoid -Motivation, Speed up)
  - Error-prone activities (Avoid +Faults/Biases)
- Can **SUPPORT DECISIONS**
  - Transform problem/solution space
  - Collect (new) data
  - Facilitate communication & collaboration

Technology's role

But Basic Problem: SW is People-Driven and People Vary

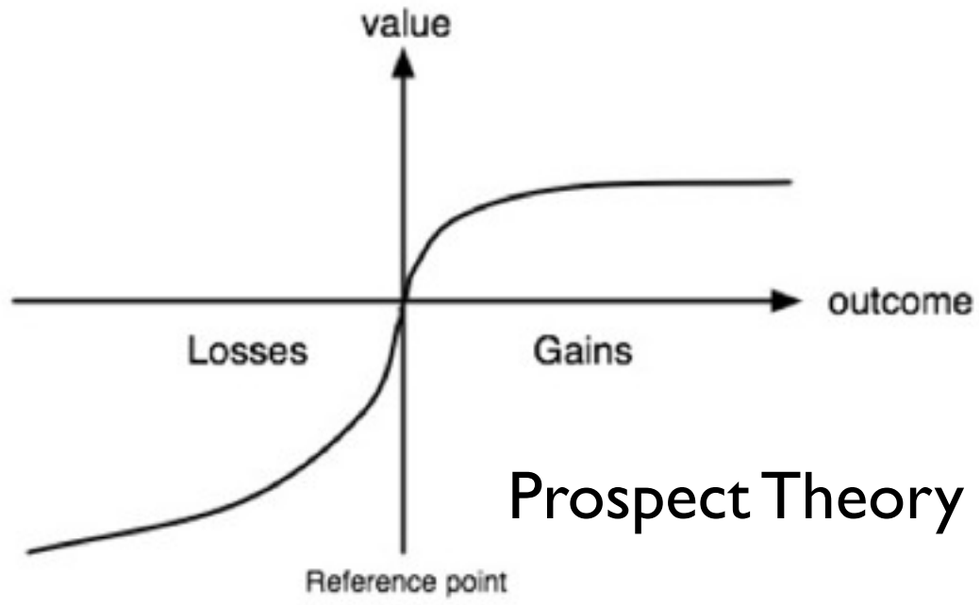
- Can **AUTOMATE**
  - Tedious tasks (Avoid -Motivation, Speed up)
  - Error-prone activities (Avoid +Faults/Biases)
- Can **SUPPORT DECISIONS**
  - Transform problem/solution space
  - Collect (new) data
  - Facilitate communication & collaboration

Some Failure Modes

- People:
  - Make Mistakes
  - Prefer to Fail Conservatively
  - Invent rather than Investigate
  - Are creatures of Habit
  - Are Inconsistent
  - Have many Cognitive Biases

Some Failure Modes

- People:
  - Make Mistakes
  - Prefer to Fail Conservatively
  - Invent rather than Investigate
  - Are creatures of Habit
  - Are Inconsistent
  - Have many Cognitive Biases



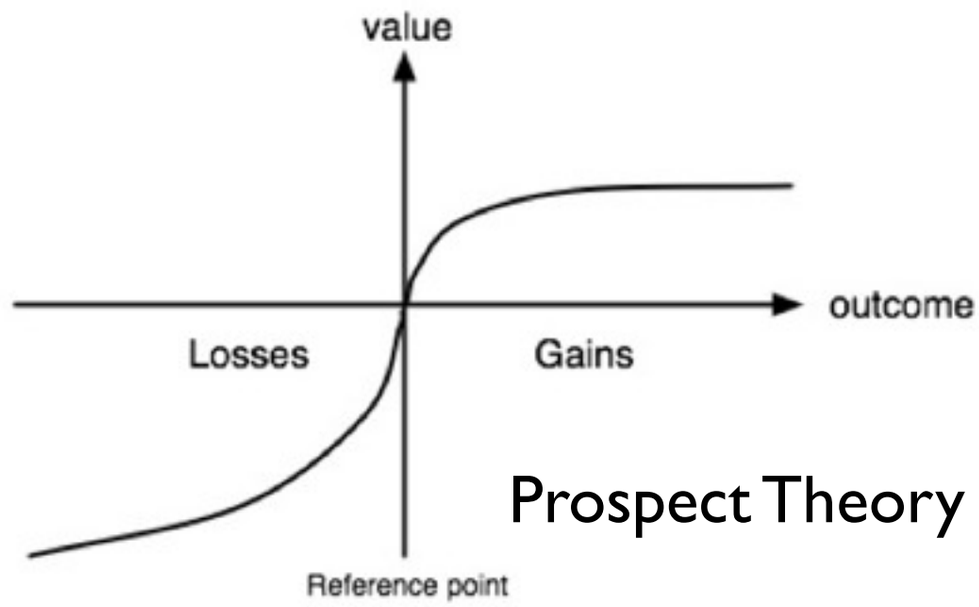
Prospect Theory

Risk-averse to gains

- Fail similar >= Fail differently
- We might be blamed if we failed and did it differently
- Serious effects in SE:
  - Less willing to test new processes...

Risk-averse to gains

- Fail similar >= Fail differently
- We might be blamed if we failed and did it differently
- Serious effects in SE:
  - Less willing to test new processes...



Prospect Theory

Premium on Originality

- Many cultures put premium on originality
- We learn to invent rather than search/find

Some Cognitive Biases: Decision-making or Behavioral

- Anchoring = rely too much on one piece of info
- Bandwagon = do or believe as majority
- Bias blind spot = "I'm less biased than you..."
- Confirmation = seek info that supports, not counters
- Framing = draw different conclusions from same info depending on its presentation
- Zero-risk = prefer small-risk-to-zero over greater-reduction-in-large-risk

Some Cognitive Biases: Social

- Actor-observer = overemphasize personality (vs. situation) in explaining behavior of others
- False consensus = overestimate degree to which others agree with me
- Halo = traits "spill over" from one area to another
- Assymetric insight = "I understand you more than you understand me"
- Self-serving = claim more responsibility for success than failure

IEEE TRANSACTIONS ON
SOFTWARE
ENGINEERING

Cognitive Heuristics in Software Engineering: Applying and Extending Anchoring and Adjustment to Artifact Reuse

December 2004 (vol. 30 no. 12)

pp. 873-888

Jeffrey Parsons, IEEE Computer Society
Chad Saunders

DOI Bookmark: http://doi.ieeecomputersociety.org/10.1109/TSE.2004.94

ABSTRACT

The extensive literature on reuse in software engineering has focused on technical and organizational factors, largely ignoring cognitive characteristics of individual developers. Despite anecdotal evidence that cognitive heuristics play a role in successful artifact reuse, few empirical studies have explored this relationship. This paper proposes how a cognitive heuristic, called anchoring, and the resulting adjustment bias can be adapted and extended to predict issues that might arise when developers reuse code and/or designs. The research proposes that anchoring and adjustment can be manifested in three ways: propagation of errors in reuse artifacts, failure to include requested functionality absent from reuse artifacts, and inclusion of unrequested functionality present in reuse artifacts. Results from two empirical studies are presented. The first study examines reuse of object classes in a programming task, using a combination of practicing programmers and students. The second study uses a database design task with student participants. Results from both studies indicate that anchoring occurs. Specifically, there is strong evidence that developers tend to use the extraneous functionality in the artifacts they are reusing and some evidence of anchoring to errors and omissions in reused artifacts. Implications of these findings for both practice and future research are explored.

Discipline & Tolerance

- Creatures of habit => resist new
- Inconsistent in what we do/think
- Choice of paradigm:
  - Discipline = enforce specific behavior
  - Tolerance = tolerate variation and differences

Discipline & Tolerance

- Creatures of habit => resist new
- Inconsistent in what we do/think
- Choice of paradigm:
  - Discipline = enforce specific behavior
  - Tolerance = tolerate variation and differences

Discipline & Tolerance

- Creatures of habit => resist new
- Inconsistent in what we do/think
- Choice of paradigm:
  - Discipline = enforce specific behavior
  - Tolerance = tolerate variation and differences

Discipline & Tolerance

- Creatures of habit => resist new
- Inconsistent in what we do/think
- Choice of paradigm:

Harder to attain, May be more effective

- Discipline = enforce specific behavior
- Tolerance = tolerate variation and differences

Discipline & Tolerance

- Creatures of habit => resist new
- Inconsistent in what we do/think
- Choice of paradigm:

Harder to attain, May be more effective

- Discipline = enforce specific behavior

- Tolerance = tolerate variation and differences

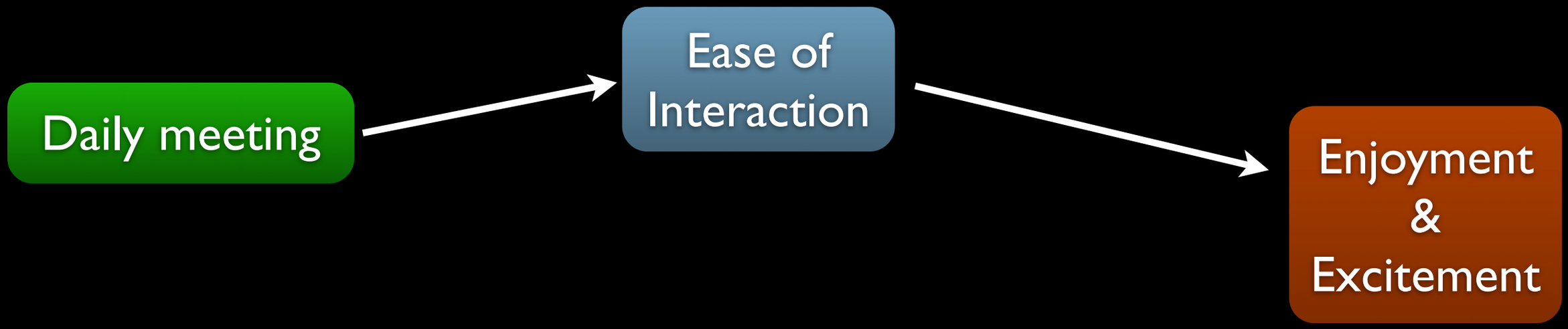
Easier to adopt, May be less productive

Some Success Modes

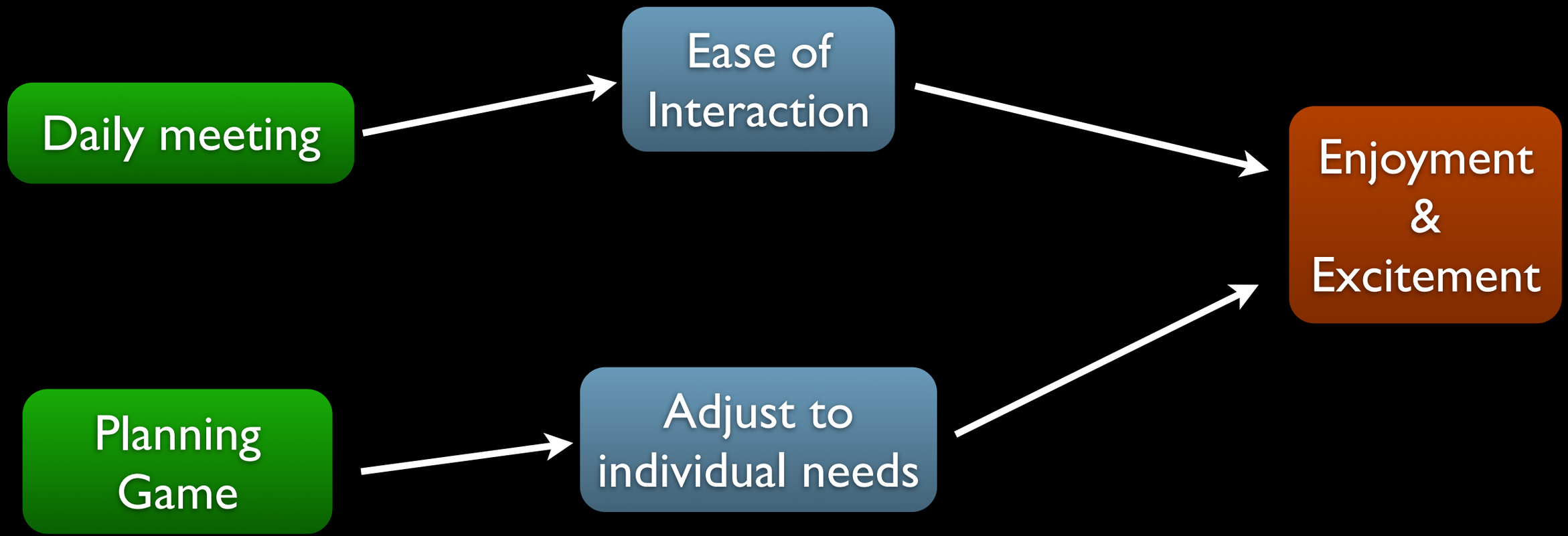
- People generally work better:
  - from Examples (concrete and tangible)
  - by Altering rather than from scratch
  - by Watching
  - by Getting Feedback

Motivation in 22 agile devs [Whitworth2007]

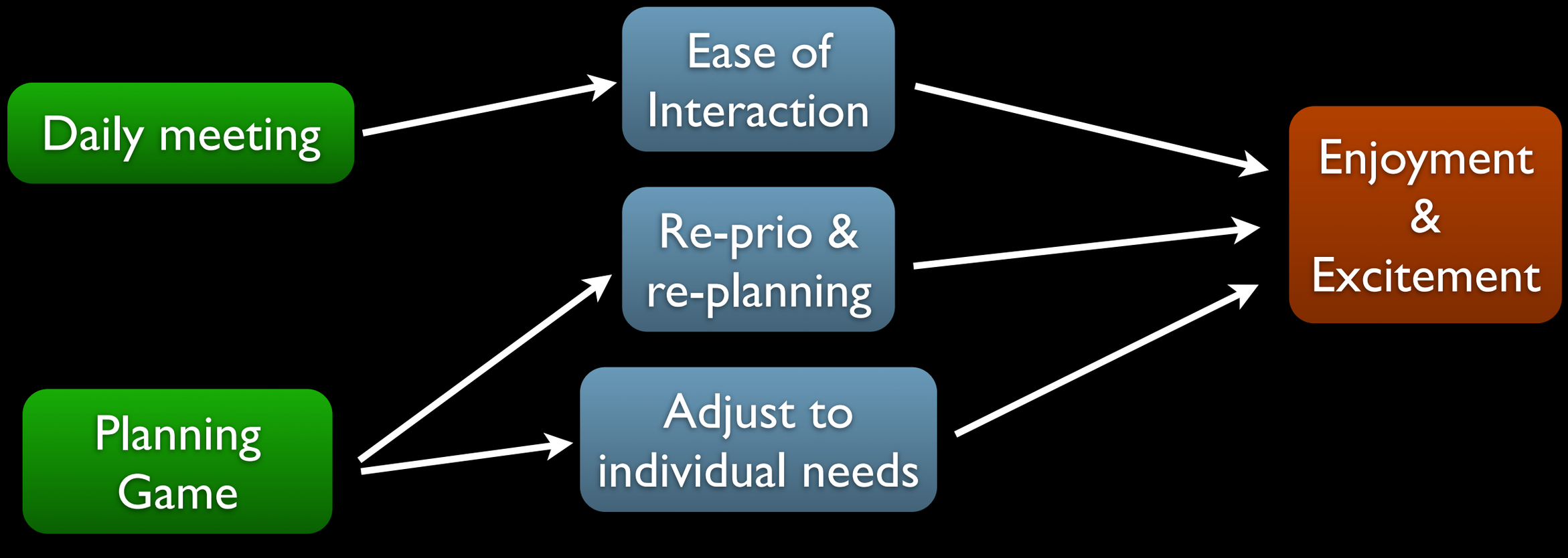
Motivation in 22 agile devs [Whitworth2007]



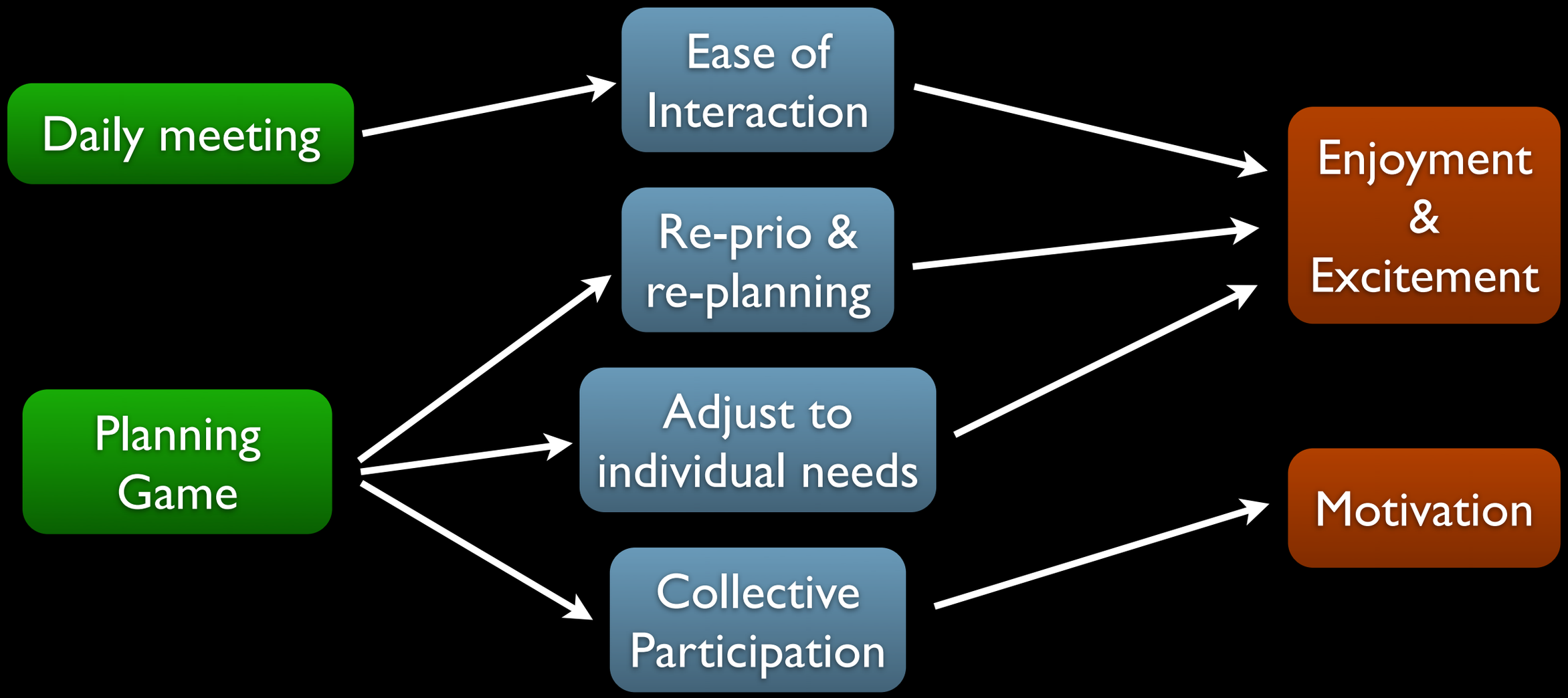
Motivation in 22 agile devs [Whitworth2007]



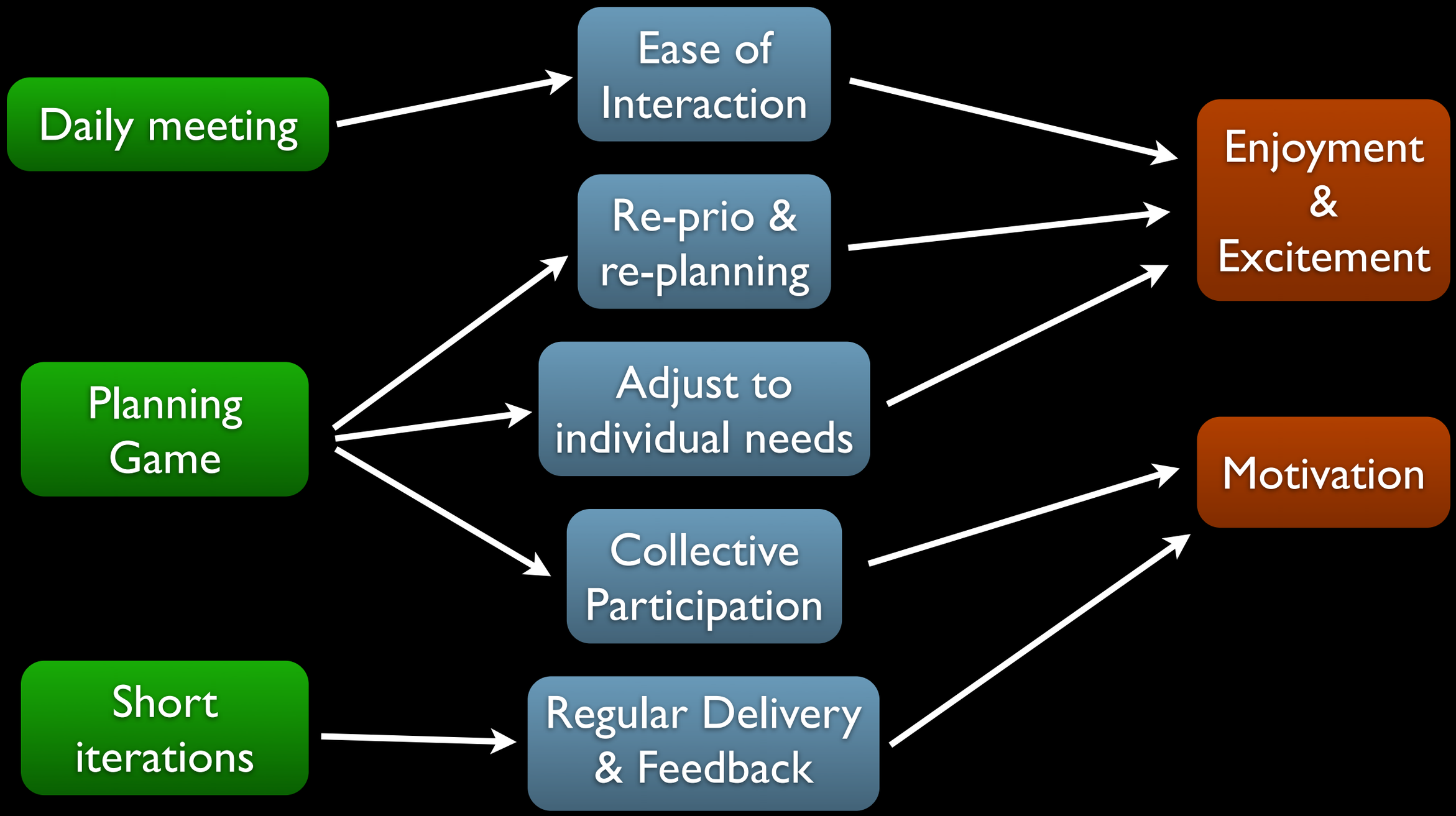
Motivation in 22 agile devs [Whitworth2007]



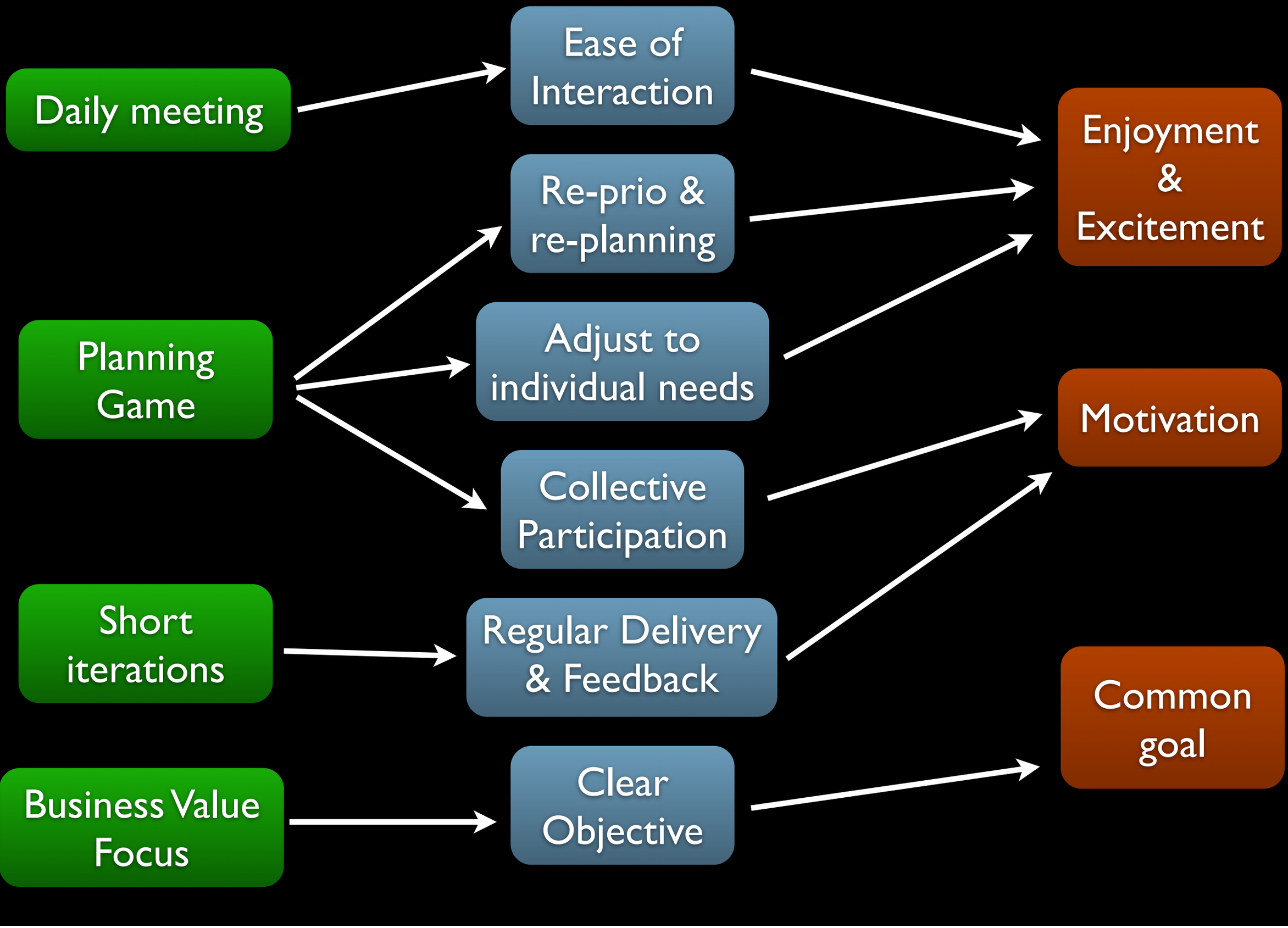
Motivation in 22 agile devs [Whitworth2007]



Motivation in 22 agile devs [Whitworth2007]



Motivation in 22 agile devs [Whitworth2007]



Up to 2006

Introduction & Adoption

Human & Social Factors

Perceptions

Comparisons

Introduction & Adoption

Human & Social Factors

Perceptions

Comparisons

Agile practices **easy to introduce** and work well

Difficult to intro in **large/complex organizations**

Benefits:
Customer collaboration
Defect handling processes
Learning among developers
Estimation of time/cost easier

Some studies saw **pair programming as inefficient**

XP works best with **experienced teams**

Introduction & Adoption

Human & Social Factors

Perceptions

Comparisons

XP well accepted in **different organizations** (hierarchical structure to little or no control)

Good **interpersonal skills** and **trust** important for successful XP teams

Individual **autonomy must be balanced** with team autonomy

Making progress tracking **visible and audible** important

Important **standardization of collaborative work**

Introduction & Adoption

Human & Social Factors

Perceptions

Comparisons

Customers liked **more (give/get) feedback**

On-site customer stressful/unsustainable

Developers more satisfied with work and product

Pair programming considered tiring since it required focused concentration

Pair programming **hard when skills differ much**

Test-driven development was difficult

Introduction & Adoption

Human & Social Factors

Perceptions

Comparisons

Agile can more **easily incorporate changes** and **show business value**

Can be **combined with traditional stage-gate project management**

Subjects **believe agile increases productivity**

Linking Personality to Views & Attitudes

Linking Personality to Views & Attitudes

47 Industrial SW Engineers

Linking Personality to Views & Attitudes

Personality Test

47 Industrial SW Engineers

Linking Personality to Views & Attitudes

Personality Test

SE Views & Attitudes

47 Industrial SW Engineers

Linking Personality to Views & Attitudes

Personality Test

Patterns?

SE Views & Attitudes

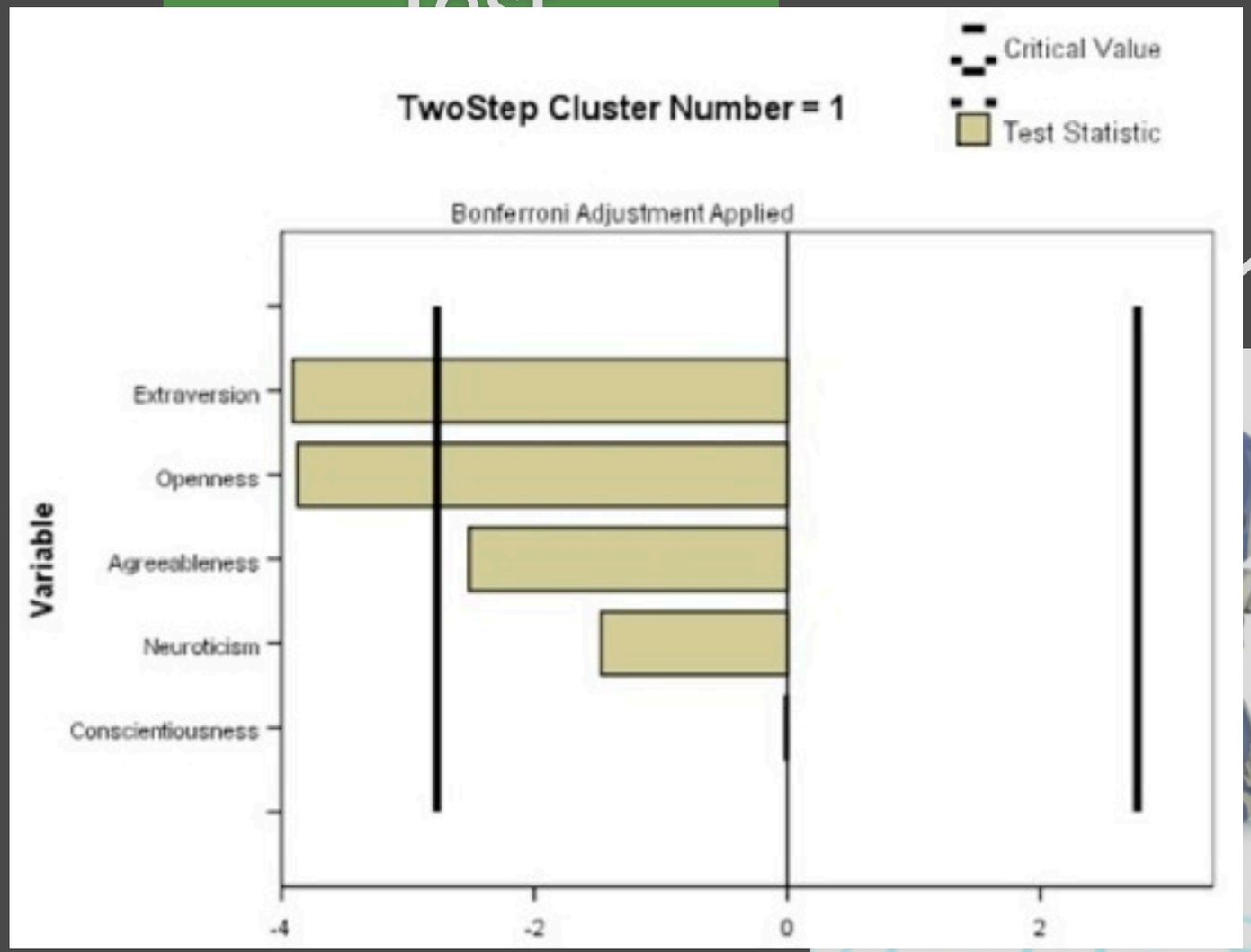
47 Industrial SW Engineers

Linking Personality to Views & Attitudes

Personality Test

Patterns?

SE Views & Attitudes



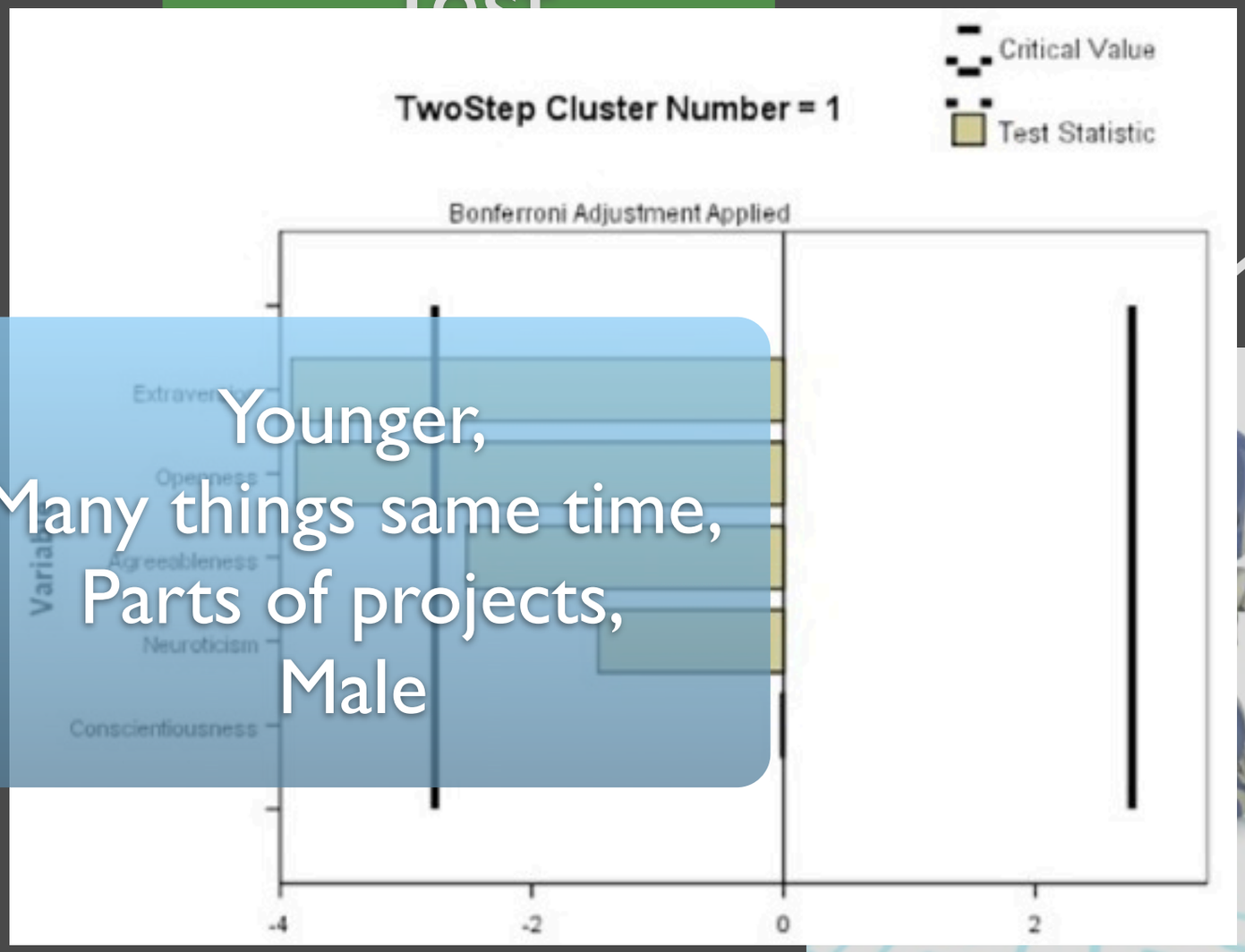
47 Industrial SW Engineers

Linking Personality to Views & Attitudes

Personality Test

Patterns?

SE Views & Attitudes



Younger,
Many things same time,
Parts of projects,
Male

47 Industrial SW Engineers

Linking Personality to Views & Attitudes

Personality

Patterns?

SE Views & ...udes

The GLM estimated from these variables is:

$$E = c + a_{93} + a_{92} + a_{91} + a_{77}$$

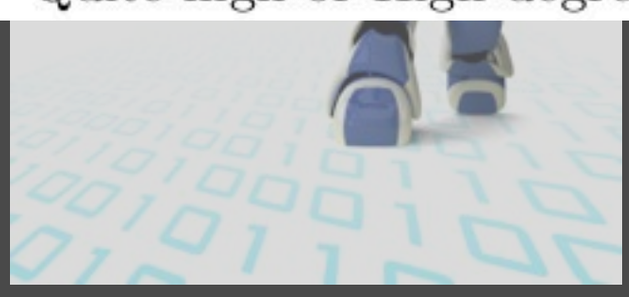
where $c = 33.265$ is the intercept,

$$a_{93} = \begin{cases} -3.640 & \text{for answer 'By yourself'} \\ 0 & \text{for answer 'In a team'} \end{cases},$$

$$a_{92} = \begin{cases} -1.118 & \text{for answer 'One thing at a time'} \\ 0 & \text{for answer 'Several things at once'} \end{cases},$$

$$a_{91} = \begin{cases} 4.672 & \text{for answer 'After a given schedule, project plan'} \\ 0 & \text{for answer 'As the day develops'} \end{cases}, \text{ and}$$

$$a_{77} = \begin{cases} -4.365 & \text{for answer 'Low or Quite low degree'} \\ 0 & \text{for answer 'Quite high or High degree'} \end{cases}.$$



47 Industrial SW Engineers

Linking Personality to Views & Attitudes

Personality

Patterns?

SE Views & ...udes

The GLM estimated from these variables is:

$$E = c + a_{93} + a_{92} + a_{91} + a_{77}$$

where $c = 33.265$ is the intercept,

$$a_{93} = \begin{cases} -3.640 & \text{for answer 'By yourself'} \\ 0 & \text{for answer 'In a team'} \end{cases},$$

$$a_{92} = \begin{cases} -1.118 & \text{for answer 'One thing at a time'} \\ 0 & \text{for answer 'Several things at once'} \end{cases},$$

$$a_{91} = \begin{cases} 4.672 & \text{for answer 'After a given schedule, project plan'} \\ 0 & \text{for answer 'As the day develops'} \end{cases}, \text{ and}$$

$$a_{77} = \begin{cases} -4.365 & \text{for answer 'Low or Quite low degree'} \\ 0 & \text{for answer 'Quite high or High degree'} \end{cases}.$$

Prefer working (with)?

47 Industrial SW Engineers

Linking Personality to Views & Attitudes

Personality

Patterns?

SE Views & ...udes

The GLM estimated from these variables is:

$$E = c + a_{93} + a_{92} + a_{91} + a_{77}$$

where $c = 33.265$ is the intercept,

$$a_{93} = \begin{cases} -3.640 & \text{for answer 'By yourself'} \\ 0 & \text{for answer 'In a team'} \end{cases},$$

$$a_{92} = \begin{cases} -1.118 & \text{for answer 'One thing at a time'} \\ 0 & \text{for answer 'Several things at once'} \end{cases},$$

$$a_{91} = \begin{cases} 4.672 & \text{for answer 'After a given schedule, project plan'} \\ 0 & \text{for answer 'As the day develops'} \end{cases}, \text{ and}$$

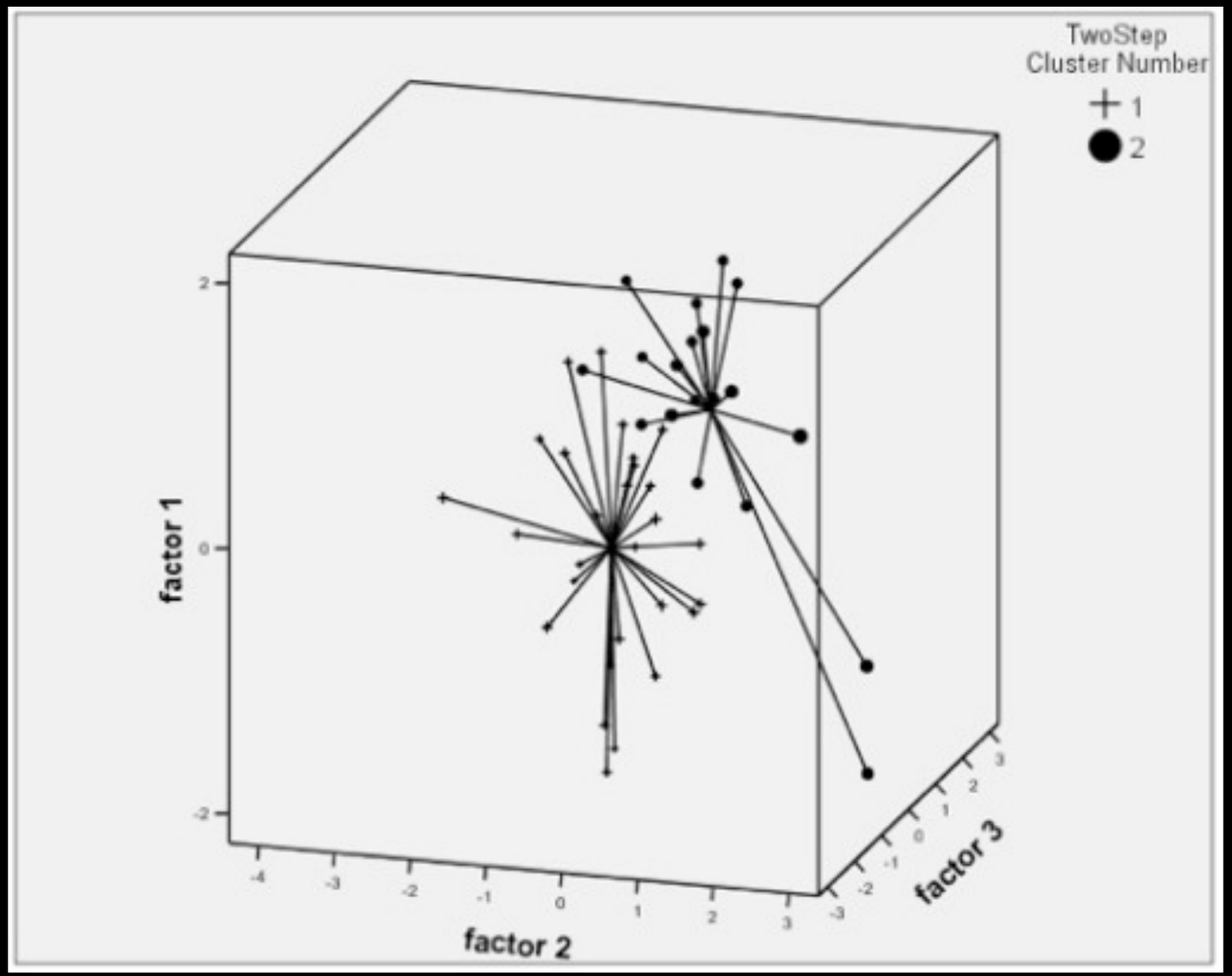
$$a_{77} = \begin{cases} -4.365 & \text{for answer 'Low or Quite low degree'} \\ 0 & \text{for answer 'Quite high or High degree'} \end{cases}$$

Prefer working (with)?

Take decisions affecting quality?

47 Industrial SW Engineers

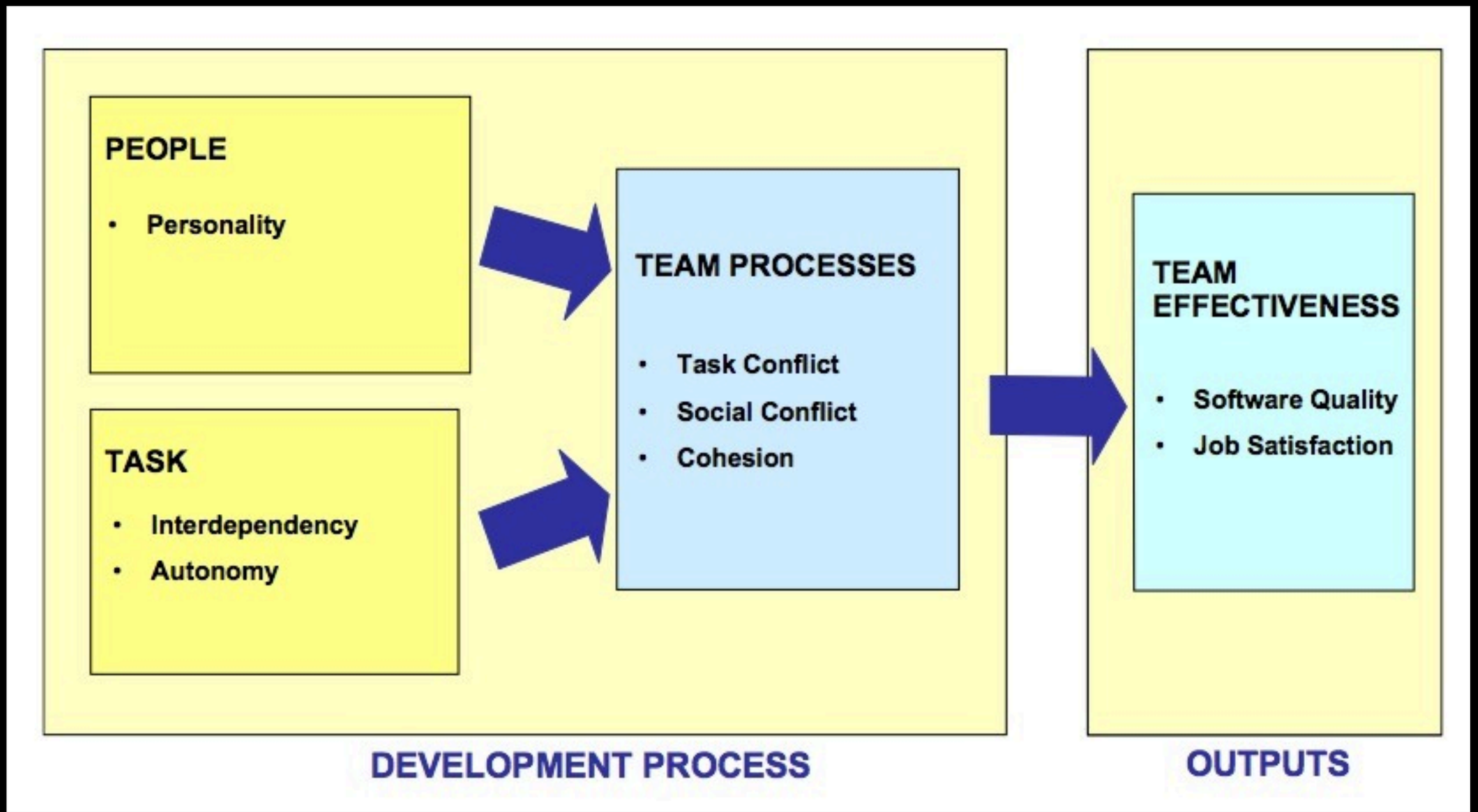
Personality and Software Engineering



- Intense personality <->
  - multiple projects
  - parts of projects
- Age & Gender differences
- Higher Extraversion <->
  - prefer team work
  - prefer plan & schedule
- Higher Openness <->
  - whole project responsibility

[Feldt2010]

Input - Process - Output Model



[Acuna2009]

Personality and Teams

Table 1
Summary of the findings of social psychology and software engineering research on teams

	Cohesion	Conflict	Performance	Satisfaction
Conscientiousness		−[3]	+[3] +[40] +[50] +[20]	
Extraversion	+[3] +[50]	−[3]	+[4] +[3]	
Agreeableness	+[3] +[40]	−[3] −[40]	+[3] +[40] +[50]	
Neuroticism	−[3] −[50]	+[3]	−[3]	
Openness to experience				+[37] (Task autonomy as moderator)
Cohesion		−[3]	+[54]	

[Acuna2009]

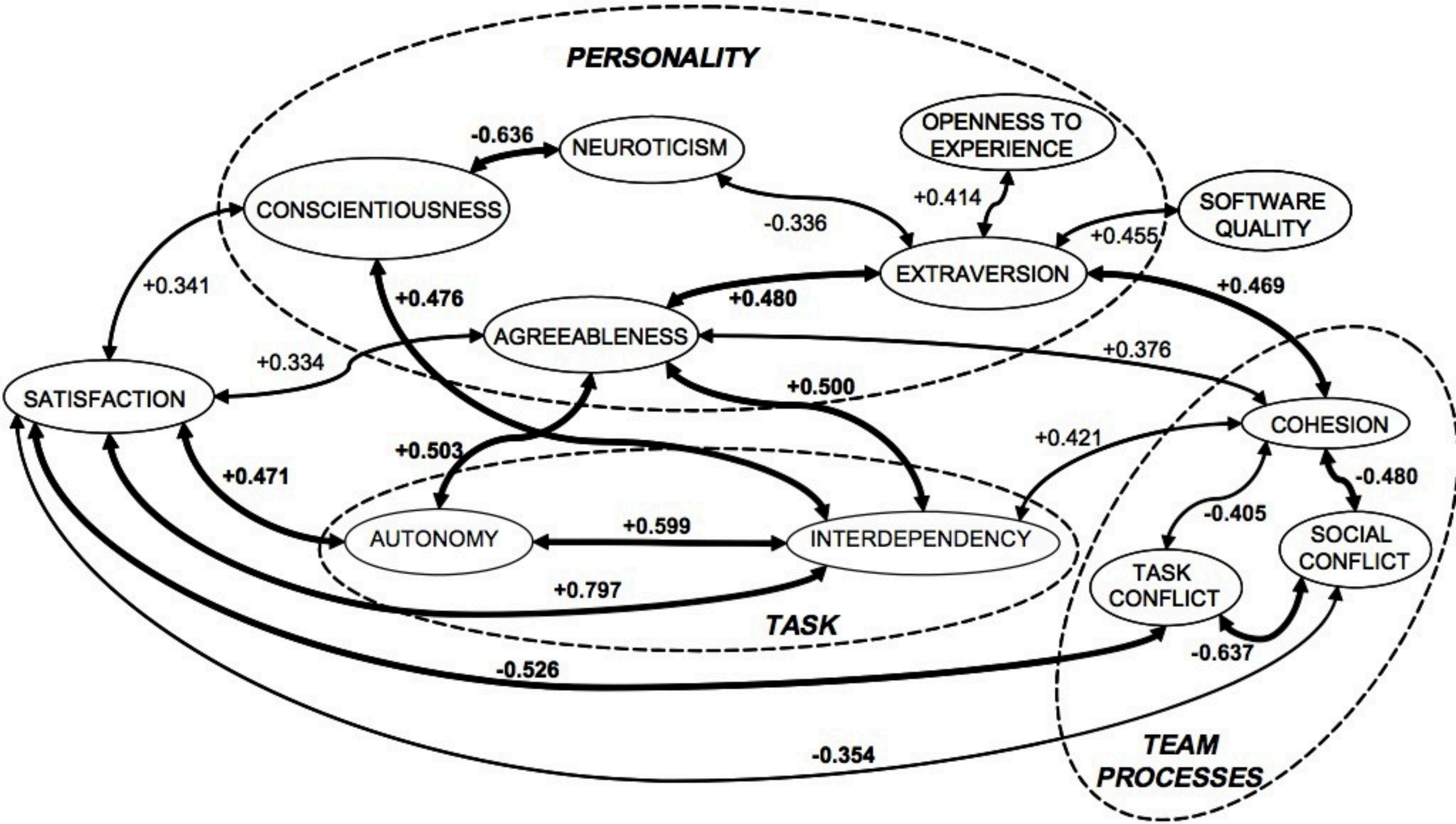


Fig. 2. Correlations between personality, team processes, task characteristics and quality or satisfaction.

[Acuna2009]

Personality in XP teams

- Repertory grid technique of 9 XP teams

[Young2005]

Pe ms

- R ms

good with people	4	4	3	2	2	2	2	good problem solver
structured/organised	4	2	2	2	4	2	2	multi-tasking
understanding (supportive)	4	2	2	2	2	4	2	working on their own
patient	5	4	4	2	2	4	2	achiever/juggling
team workers	4	4	4	2	1	2	2	individualistic
helpful	4	4	2	2	2	2	2	patient
purposeful	4	2	2	4	4	4	2	meaingful
team-minded	5	2	2	2	1	2	2	people minded
specific	5	4	2	2	2	2	2	big picture
detailed	5	4	2	2	2	2	2	look ahead
realistic in problem solving	5	4	2	2	2	2	2	does not panic (no way of passing on problems)
team player	4	4	2	2	2	2	2	single-minded
team/joint decisions	5	4	4	4	2	2	2	decision making
good people manager	5	4	4	4	2	2	2	instructing
leader	5	5	4	4	2	2	2	indirect leading
organiser	1	4	4	4	2	2	2	leader
sharing	5	4	4	2	2	2	2	doer
problem solver	5	4	4	4	2	2	2	guidance
disciplined	5	4	2	2	1	1	1	conductor

Bad team member

Good team member

XP team member

Team leader

Technical lead

Technical Architect

Opps manager

Table 5: Good (XP) Team Member constructs, MDS coordinates and ratings

Good Team Member (–2.08, 1.17)					
Dimension 1	Dimension 2	Construct	Constructs Left (rating value is 1)	Ratings Given	Constructs Right (rating value is 5)
–0.47	1.47	1	Flexible	1	More individualistic (make own decisions)
–1.09	1.04	3	Interesting	1	Formal
–1.20	1.30	5	Good communicator (knowledge transfer)	1	Making assumptions
–1.06	0.52	7	Ability to explain to people with different abilities and backgrounds	2	Ability to interact with customer (good at presenting technical information in an accessible way)
–0.85	1.26	8	Good analytical skills	2	Needs to know the right questions to ask the technical people
–0.61	1.43	47	Analytical	2	Requires energy
–0.65	1.40	48	Starting things	3	Closure
–0.56	0.85	51	Passion for learning	2	Passion for good systems
–1.35	0.87	58	Desire to make things work	1	Scheduling

"analytical personality, good interpersonal skills, passion for extending knowledge base"

Table 6: Bad Team Member constructs, MDS coordinates and ratings

Bad SD Team Member (1.57, –1.32)					
Dimension 1	Dimension 2	Construct	Constructs Left (rating value is 1	Rating given	Constructs Right (rating value is 5)
1.42	–1.11	2	Good communicators	-	Patient
0.98	–1.77	4	Rigid	4	Flexible
1.89	–0.56	6	Stable	3	More agile
1.05	–0.93	11	Team leading (drive forward)	2	Put knowledge and skills into practice (creativity)
1.92	–0.35	14	Discipline	-	Ability to focus on single task and see it to completion
1.02	–1.06	15	Thoroughness	-	Imagination
1.25	–1.24	16	Completion/ finishing	-	Flexibility
1.08	–1.57	17	Willingness to be dictatorial	1	Mentoring/supportive/ Patience

Flexible, more inclined to team leadership, willingness to be dictatorial, inclined towards domineering management, little sharing of knowledge and support to others.

Collaboration & Personality in Pair Programming

Audio recordings of 44 pairs performing a change task

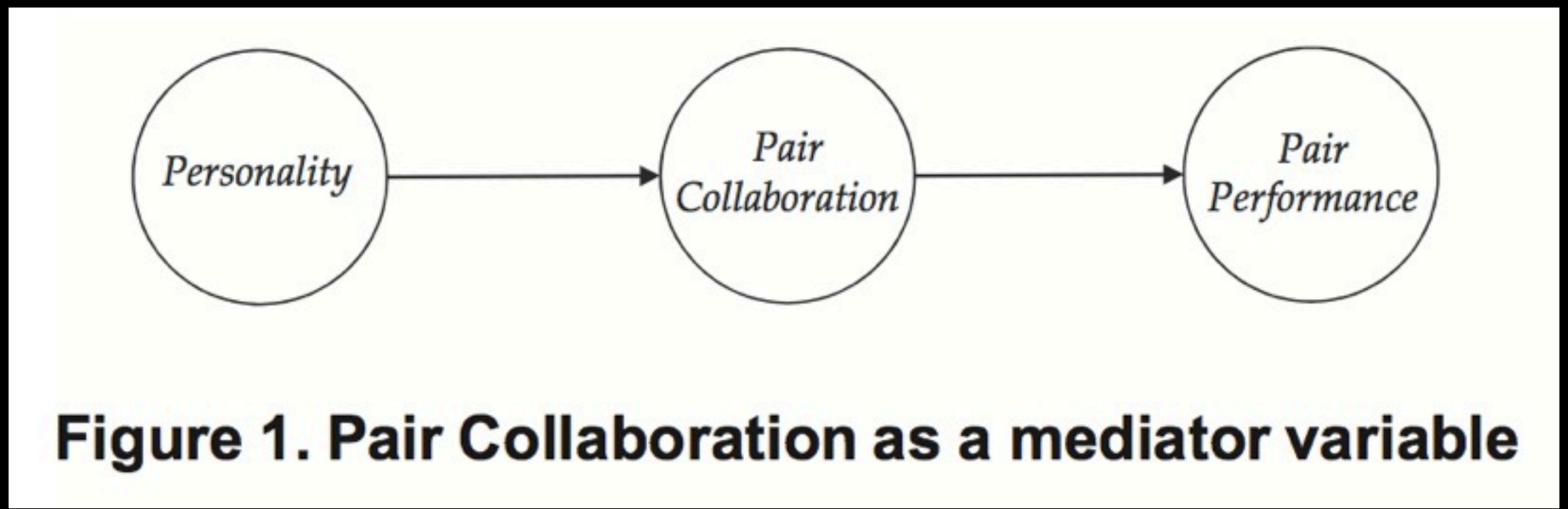


Figure 1. Pair Collaboration as a mediator variable

[Walle2009]

Collaboration & Personality in Pair Programming

Audio recordings of 44 pairs performing a change task

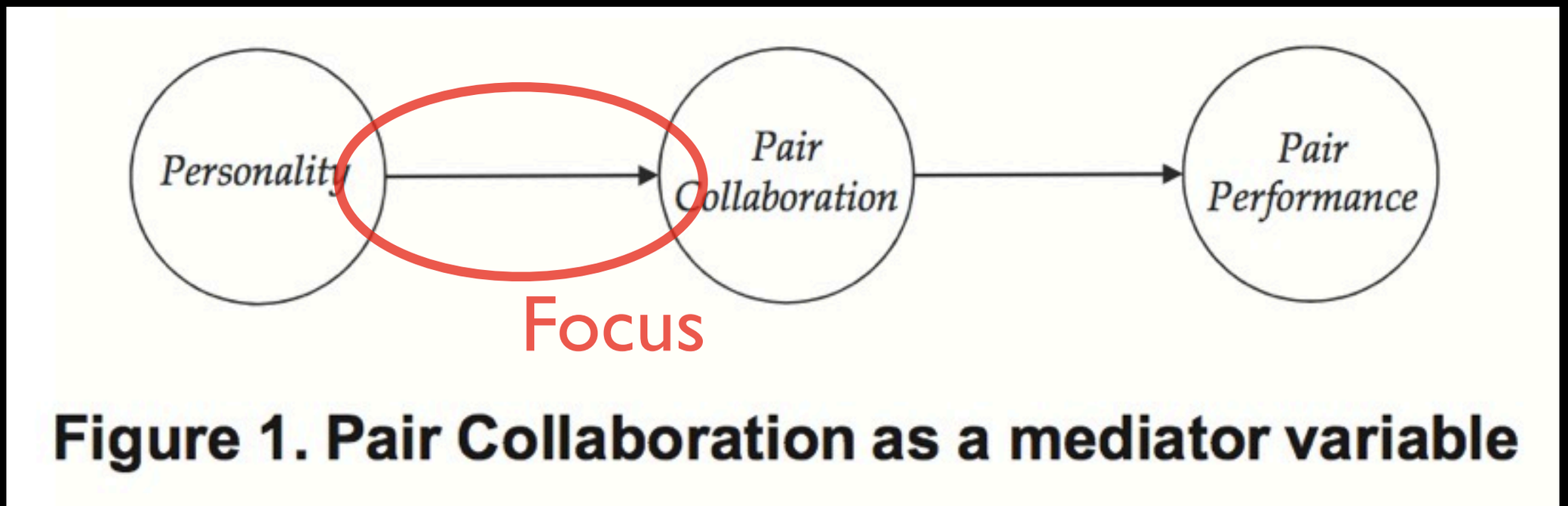


Figure 1. Pair Collaboration as a mediator variable

[Walle2009]

Collaboration

- Def: "*Situation in which all parties contribute new information to a task*"
- In contrast to cooperation: "*Splitting into sub-tasks and working on them independently*"

[Walle2009]

Results

- Personality affects collaboration
- Variability in personality increases amount of communication-intensive collaboration
- Extraversion: no connection to interruptions
- Later results have shown that task complexity and expertise have stronger effect

[Walle2009]

Refactoring

- "*a change made to the internal structure of software to make it easier to understand and cheaper to modify without changing its observable behavior*"
- Claims about refactoring:
  - Refactoring helps developers to program faster
  - Refactoring improves the design of the software
  - Refactoring makes software easier to understand
  - Refactoring helps developers to find bugs

[Fowler, Moser2008]

Refactoring

Refactorings in Alphabetical Order

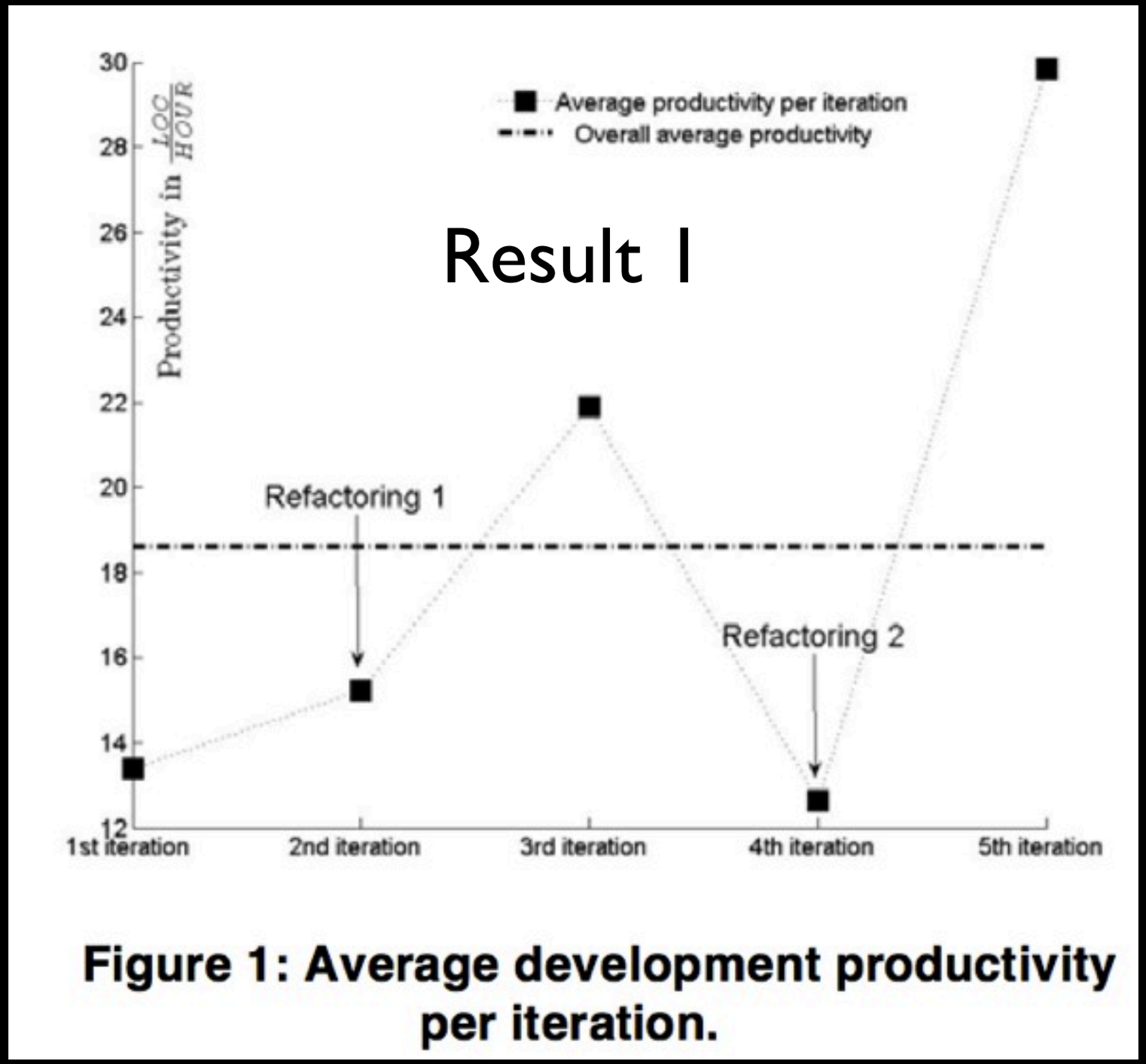
This is a simple list of refactorings both from the original book and some later sources. Sadly I haven't had
extra material to what's in the Refactoring book. Refactorings marked with NEW are in addition to those in

| Russian | German |

- Add Parameter
- Change Bidirectional Association to Unidirectional
- Change Reference to Value
- Change Unidirectional Association to Bidirectional UPDATED
- Change Value to Reference
- Collapse Hierarchy
- Consolidate Conditional Expression
- Consolidate Duplicate Conditional Fragments UPDATED
- Convert Dynamic to Static Construction *by Gerard M. Davison* NEW
- Convert Static to Dynamic Construction *by Gerard M. Davison* NEW
- Decompose Conditional
- Duplicate Observed Data
- Eliminate Inter-Entity Bean Communication *(Link Only)*
- Encapsulate Collection
- Encapsulate Downcast
- Encapsulate Field
- Extract Class

[Fowler]

Results from one case



Result 1

Figure 1: Average development productivity per iteration.

Result 2:
Refactoring can limit complexity and coupling

[Moser2008]

Sources

- Dybå & Dingsöyr, "What do we know about agile sw dev?", IEEE Software, 2009
- Feldt et al, "Links Between the Personalities, Views and Attitudes of Software Engineers", IST, 2010
- Whitworth et al, "Motivation & Cohesion in Agile Teams", XP Conf 2007
- Acuna et al, "How do Personality, Team Processes and Task Characteristics Relate to Job Satisfaction and Software Quality?", IST, 2009.
- Walle et al, "Personality and the Nature of Collaboration in Pair Programming", ESEM 2009.
- Young et al, "Personality Characteristics in an XP Team...", 2005.
- Moser et al, "A case study on the impact of refactoring...", 2008.