

Social Human Architecture for Beginners

Sebastian Schürmann
@sschuermann



Whois @sschuermann

- Past: Developer for over 10 years (Javascript, PHP, Mysql, Web and some Mobile)
- Past: Teamlead for about 5 Years
- Now: Just a Scrum-Master for 5 Teams
- Still hacking on code ... but please don't tell anyone



Thesis:

The Job as a Senior Developer or Team-Lead is about excelling in Technology, Architecture and Programming!

It's not!

It's about **people**, how to make them working together as **groups** and how to **motivate** them to do the right things the right way.

People

Groups

Motivation

People

“I am not young enough to know everything.”
O.Wilde

Introversion / Extraversion

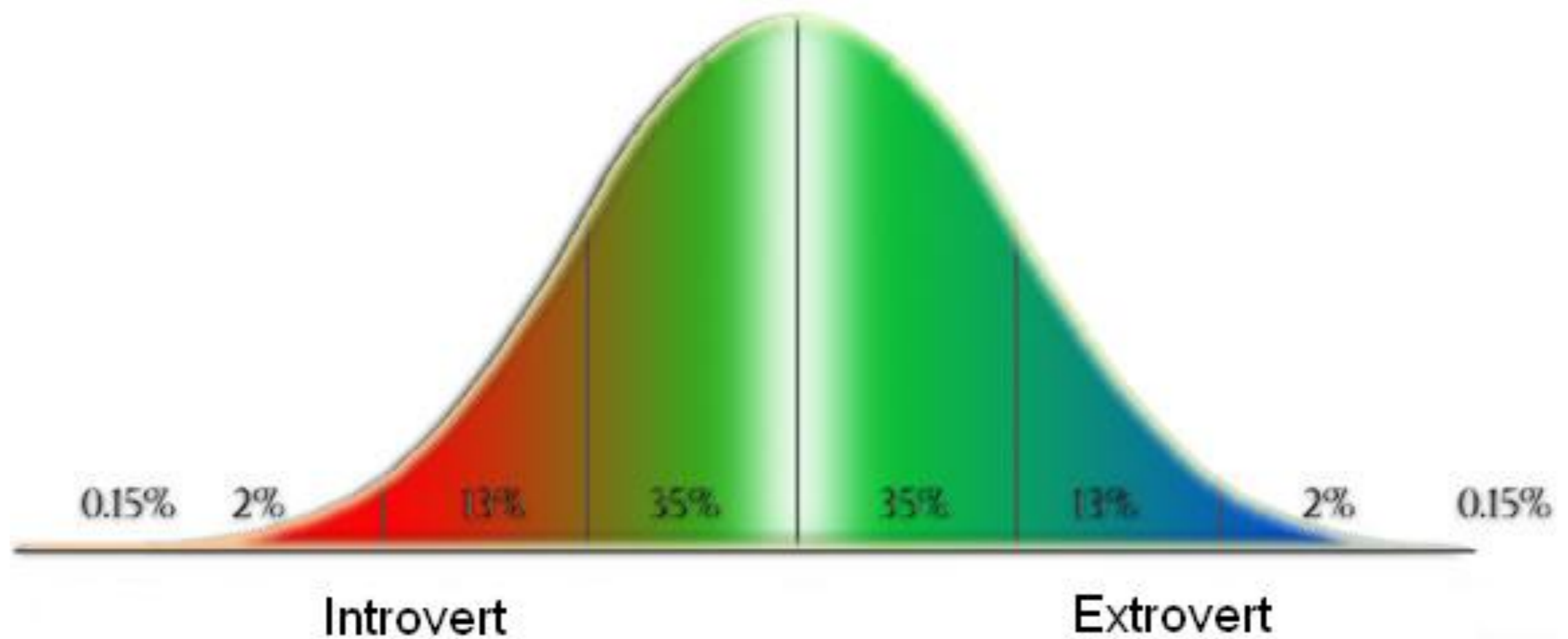
Extraversion tends to be manifested in outgoing, talkative, energetic behavior, whereas introversion is manifested in more reserved and solitary

Introversion/Extraversion

- The thinking is just internal or externalized
- Both engage in different activities when feeling pleasant
- Arousal and pleasantness are positively correlated for extraverts. (Other way around for introverts)



Same continuum?



How much of an
Introvert are you?

How much of an
extrovert are you?

Does this account for
thinking AND feeling?

Practical application

- Talk with your team about these traits
- Let everyone express on which end of the scale you feel you are
- Ask introverts actively, they might be still thinking
- Work silently in meetings to some extend

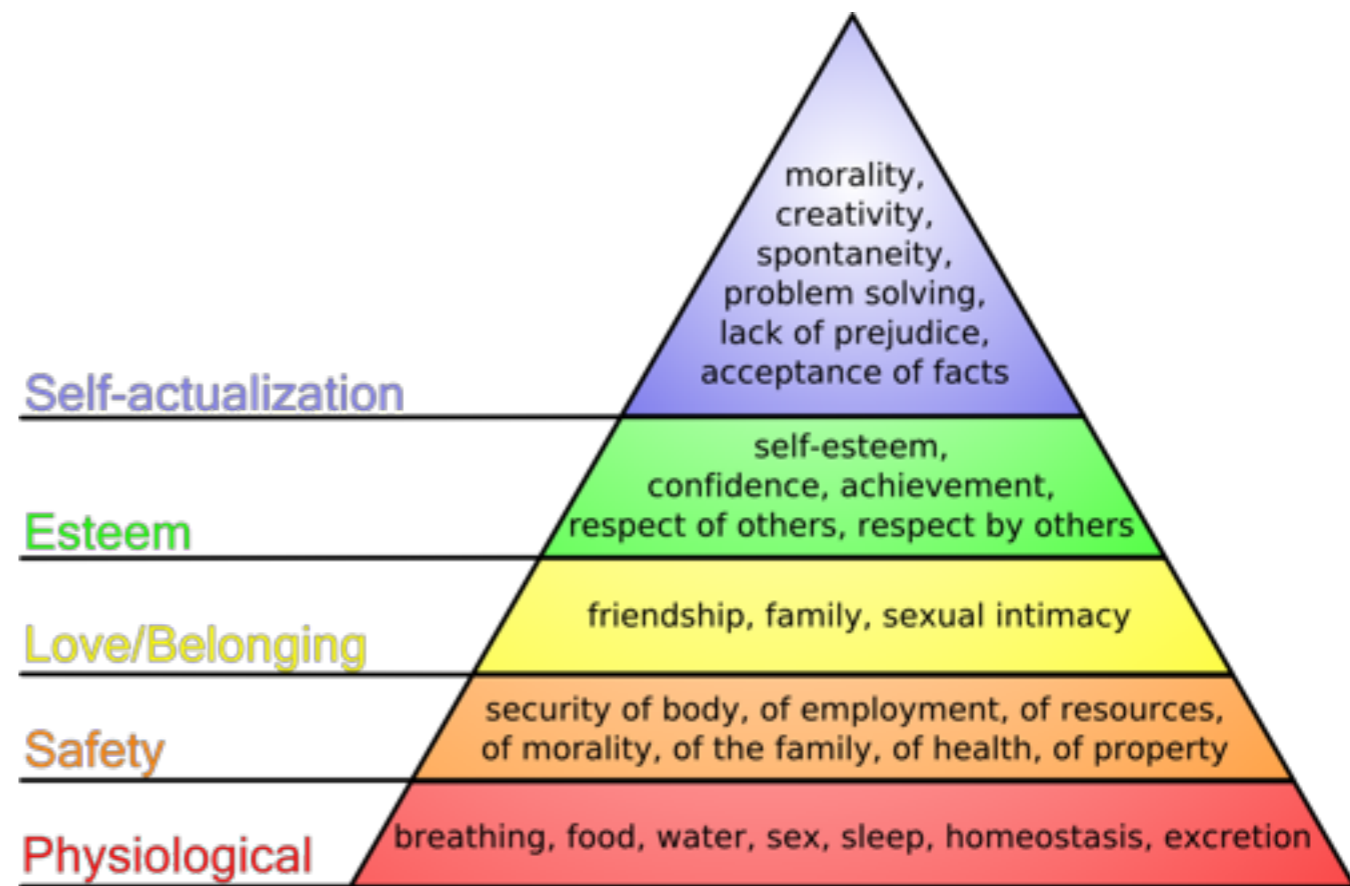


Maslow's hierarchy of needs

"the study of crippled, stunted, immature, and unhealthy specimens can yield only a cripple psychology and a cripple philosophy."

Maslow's hierarchy of needs

- Result of studying extraordinary and not mentally unhealthy people
- lower levels have to be covered to make the upper levels work
- The order and importance of these levels might be influenced by the type of environment



Practical Application

- Talk to your developers about the hierarchy
- Come up with your own hierarchy for your team
- See what unfulfilled needs your team has



Non-Violent Communication

... focuses on three aspects of communication: self-empathy, empathy and honest self-expression

Non-Violent Communication

- Communication and Conflict resolution process
- Based on the idea that everyone has a capacity for compassion
- Culture teaches us very violent ways of communicating
- Identify your needs and the needs of others
- Communication process, spiritual practice and a set of values



4 Components of NVC

- Observation (Just the facts)
- Feelings (Just feeling good or bad wont cut it)
- Needs (we all have them)
- Honest Request



Assumptions of NVC

Common sense hitting home



1. All human beings share the same basic needs



2. Our world offers sufficient resources for meeting everyone's basic needs



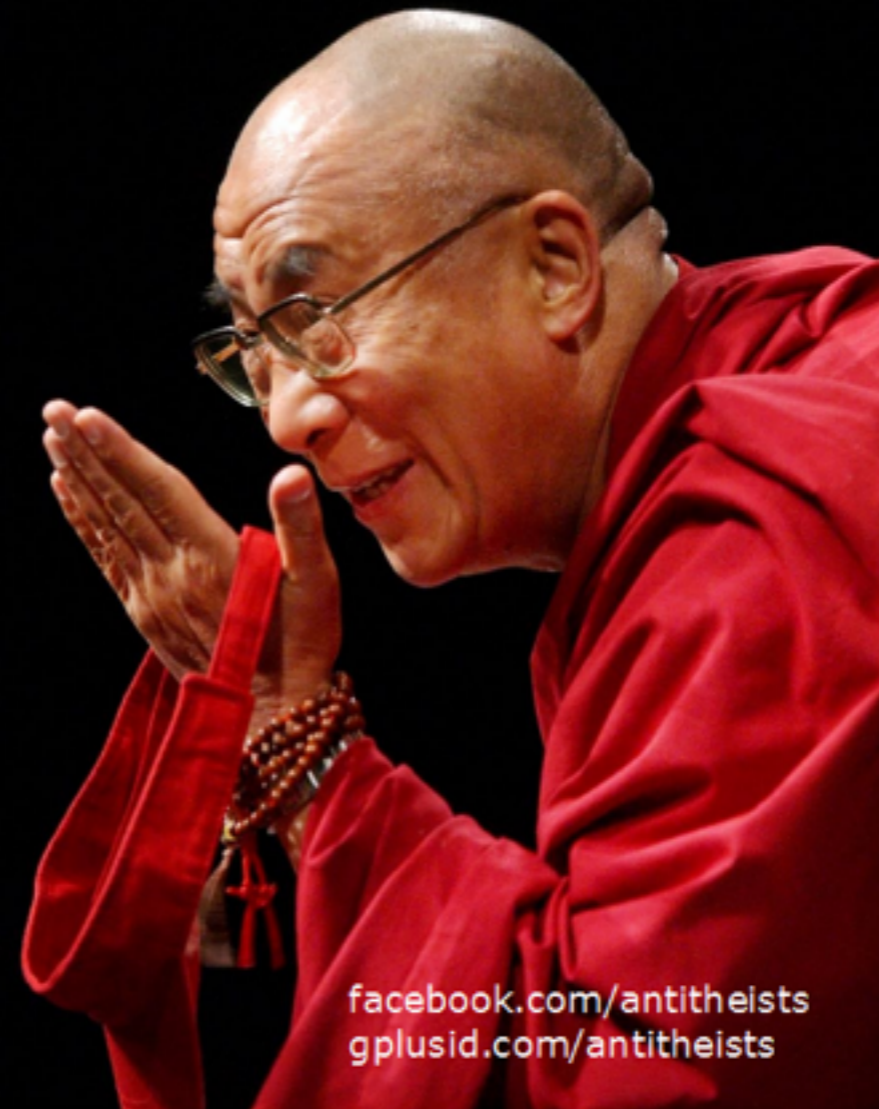
3. All actions are attempts to meet needs



4. Feelings point to needs unmet

“Whether one believes in a religion or not, and whether one believes in rebirth or not, there isn't anyone who doesn't appreciate kindness and compassion.”

Dalai Lama



facebook.com/antitheists
gplusid.com/antitheists

5. All human beings have the capacity for compassion



6. Human beings enjoy giving



PAIR PROGRAMMING

It's a fight waiting to happen.
Wait, you wanna do what, where!?!?!?!???

seanthelexguy.com/blog

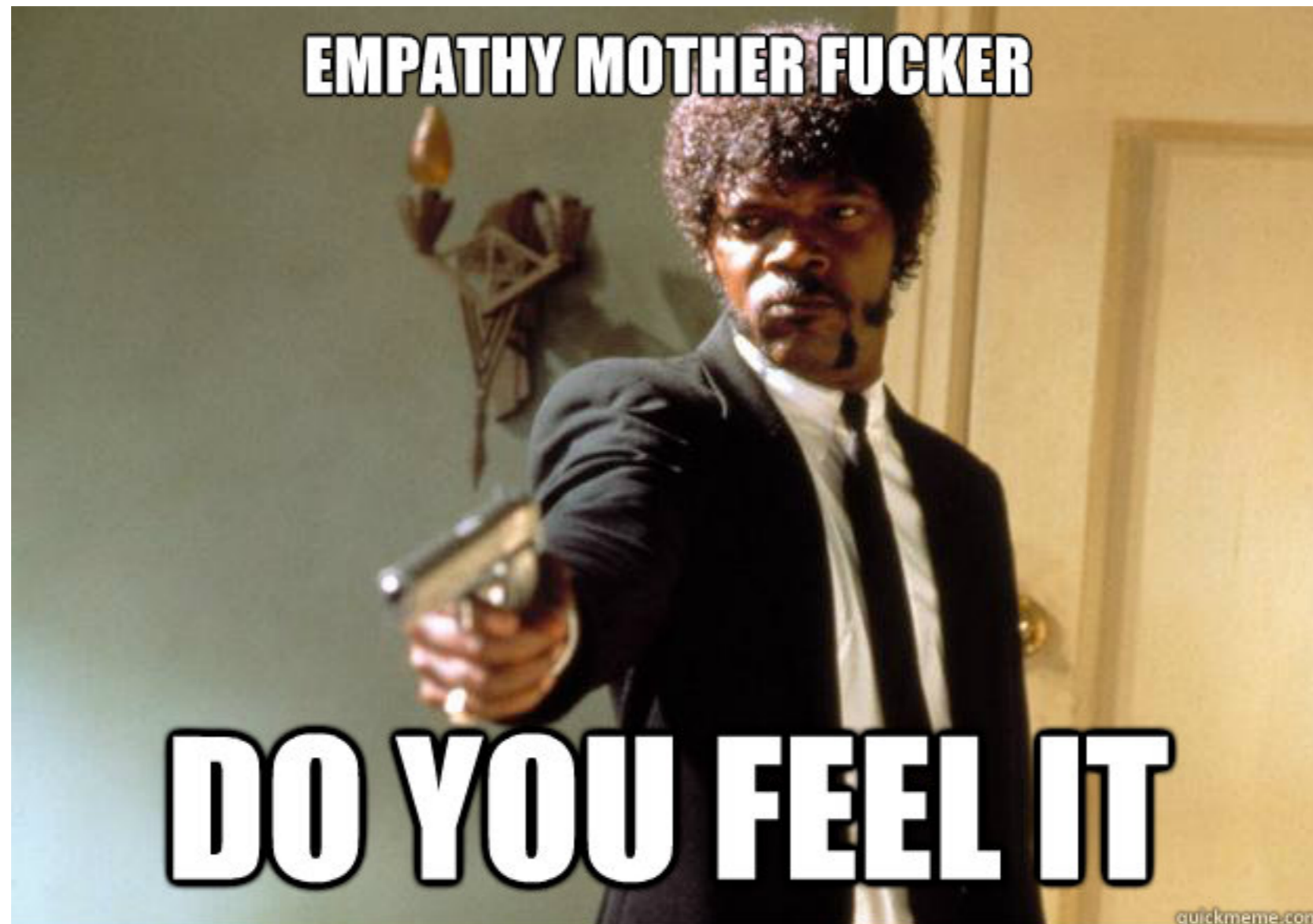
7. Human beings meet needs through interdependent relationships



8. Human beings change



9. Choice is internal



10. The most direct path to peace is through self-connection

Let's assume one of your
team-mates fucks up.

You have 4 choices

4 Choices of NVC

- It's his fault!
- It's my fault!
- What were his feelings and needs in that moment?
- What are my (unmet) feelings and needs?

Practical Application

- RTFM! the Book from Rosenberg
- Come to terms with your own feelings and needs when working with groups
- Start dividing out Observations, Feelings and Needs
- When enraged, think of the 4 choices

Groups

I like people who are working on practical things and who are working in teams. It's not so important to get the glory. It's much more important to get something that works. It's a better way to live.

Jerry Weinberg

- He is the „oldschool“
- Book: Psychology of Programming (1971)
- PSL - Problemsolving / Leadership Training



Egoless Programming

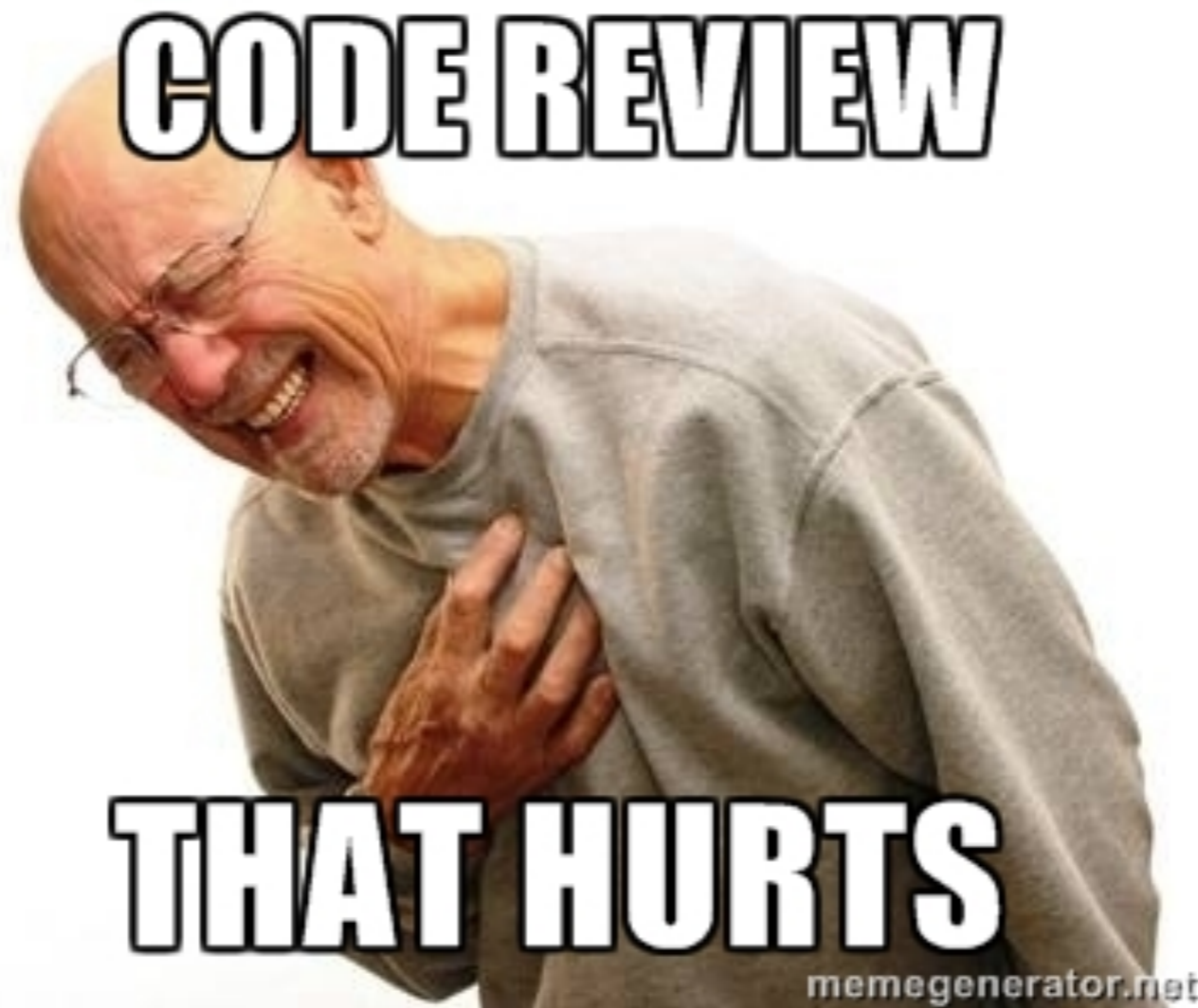
The ego is that ugly little troll that lives underneath the bridge between your mind and your heart.

Gael Greene



"MISTAKE"

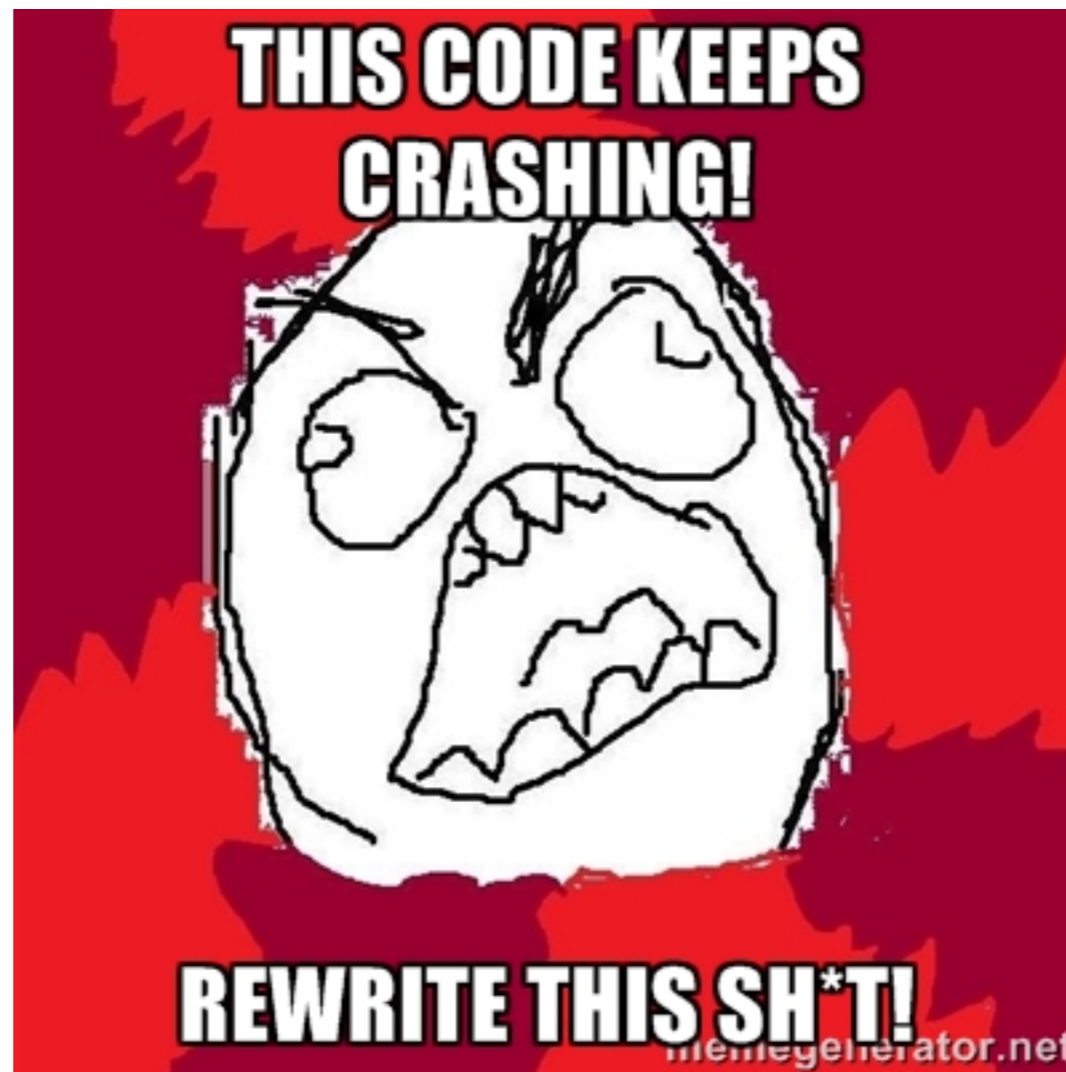
1. Understand and accept that you make mistakes



2. You are not your code



3. No matter how much Karate you know, someone else will know more



4. Don't rewrite code without consultation



5. Treat people who know less than you with respect, deference and patience



6. The only constant in the world is
change



7. The only true authority stems from knowledge, not position.

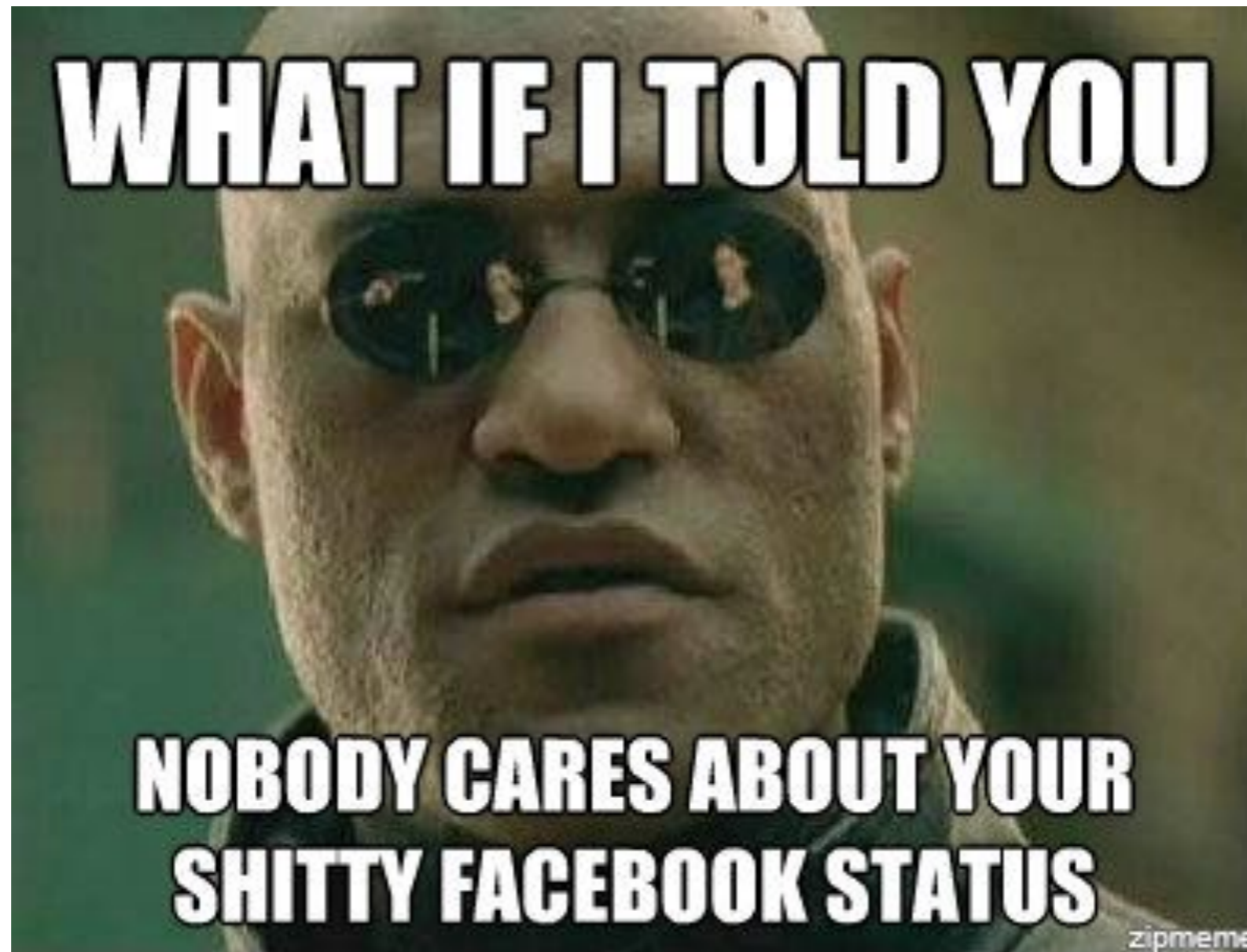


THE ONLY ONE

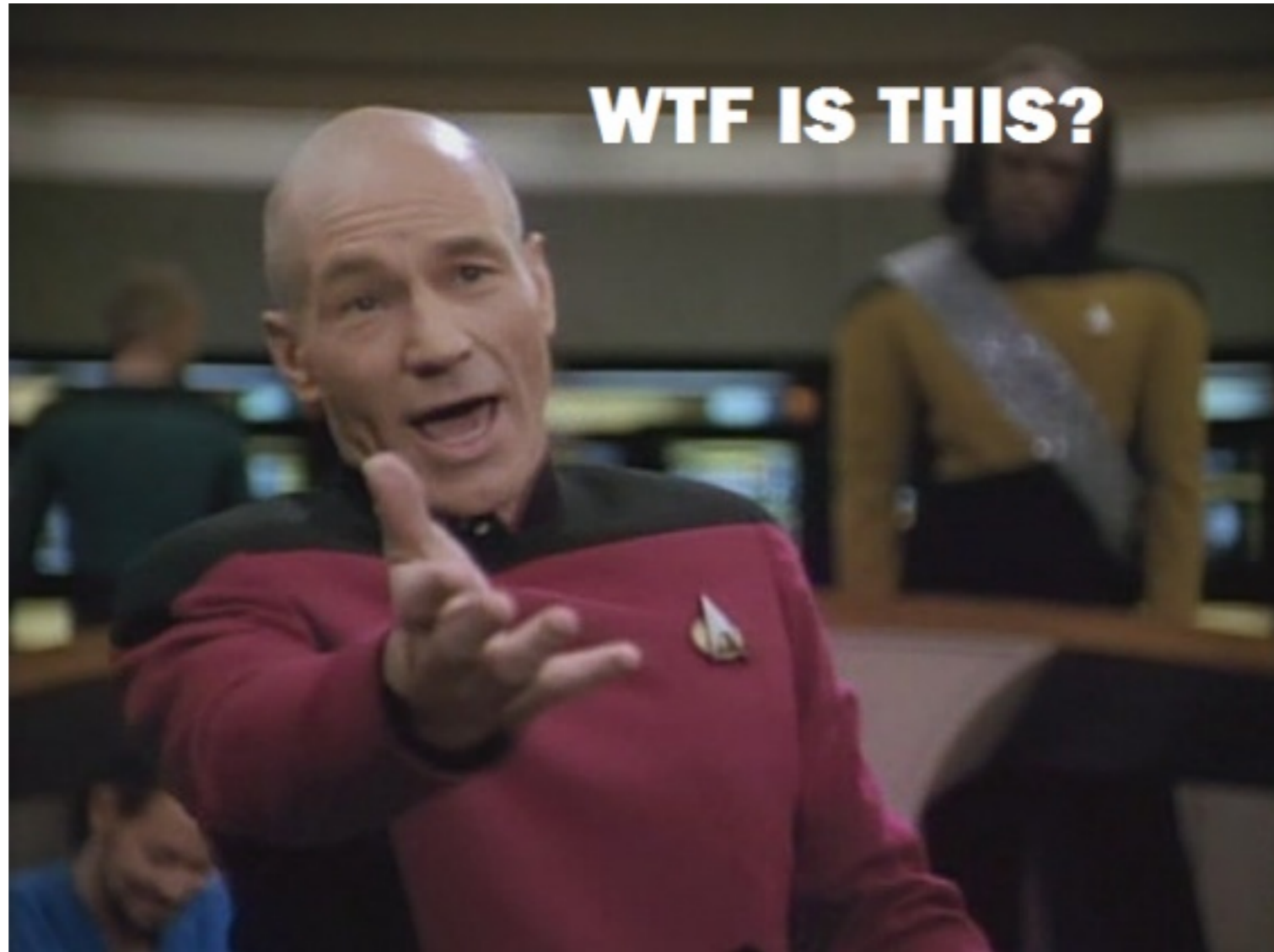
He beat Chuck Norris and you know it.

Funny-Pictures-Blog.com

8. Fight for what you believe, but gracefully accept defeat



9. Don't be the guy in the room



10. Critique the code instead of people -
be kind to the coder, not the code.

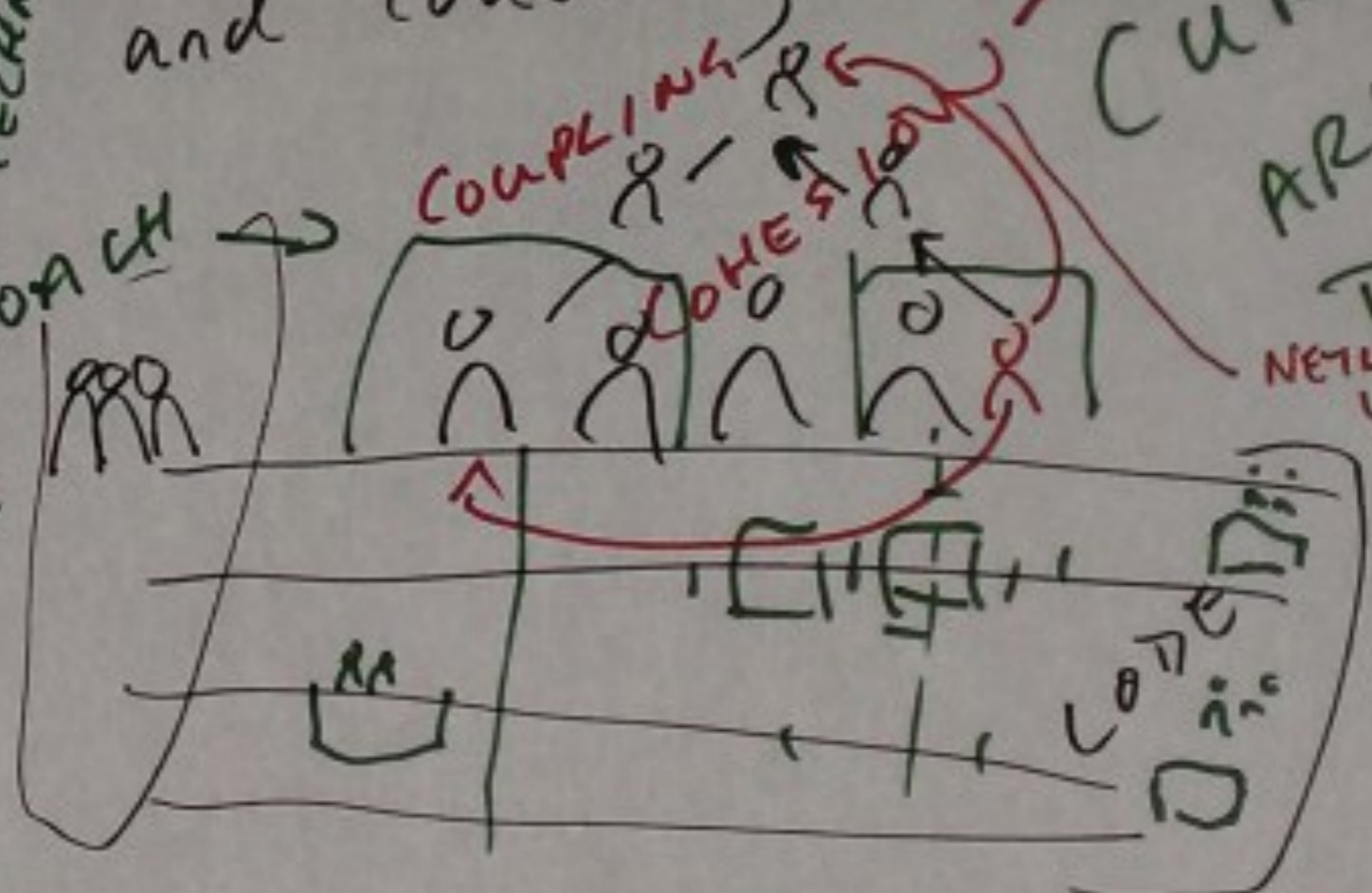
Practical Applications

- Discuss and install this as rules with your teams
- Buyin by everyone
- Sign it and display it where everyone can see it

Conway's Law

organizations which design systems ...
are constrained to produce designs
which are copies of the communication
structures of these organizations

will produce
of the organization's communication structure
ATTIVE IN CHARGE LUNCH TOGETHER - HEAVY METAL / TECHNO
Implications for team structure? and coaching?
- SOCIAL -> REFERENCE METAPHOR
CURIOUSITY ARCHITECTURE
NETWORK CYCLE
GROWTH - ORG
PRODUCT - COACH
COUPLING & COHESION
HIERARCHY ORGANIC
CULTURE ORGANIC
BEHAVIOURS ORGANIC
VALUE-STREAM MESS
AGENT MESS
AD-HOC PROPERTY
ROTATION TEAM SIZE?
SUB TEAMS INTERFACES
VOLUNTARY



Conway IRL - Some questions and remarks

- Offshore Teams are more likely to write modular software
- Can you see all the source code in your company?
- Are technical decisions made up the hierarchy?
- Did you merge with another company recently?
How is that influencing your codebase?

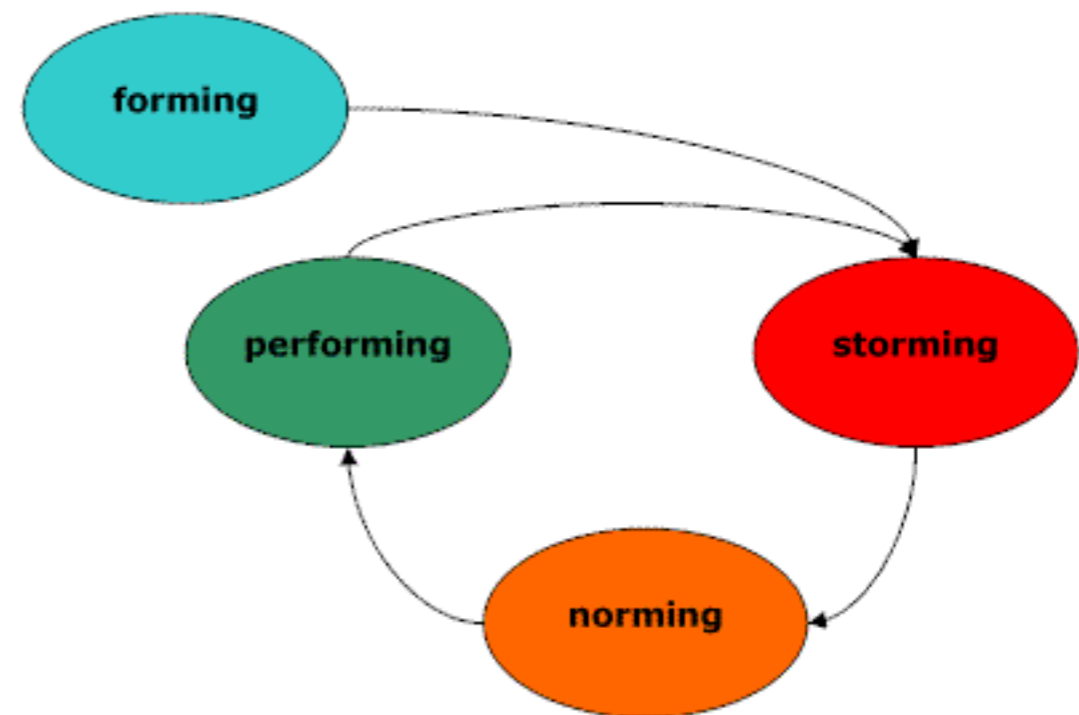
Practical Application

- Review your Organizations structure and dive into your code
- Find similarities and ask yourself: Is this good or bad
- You might have to change the structure of your org to change your code!

Tuckman's stages of group development

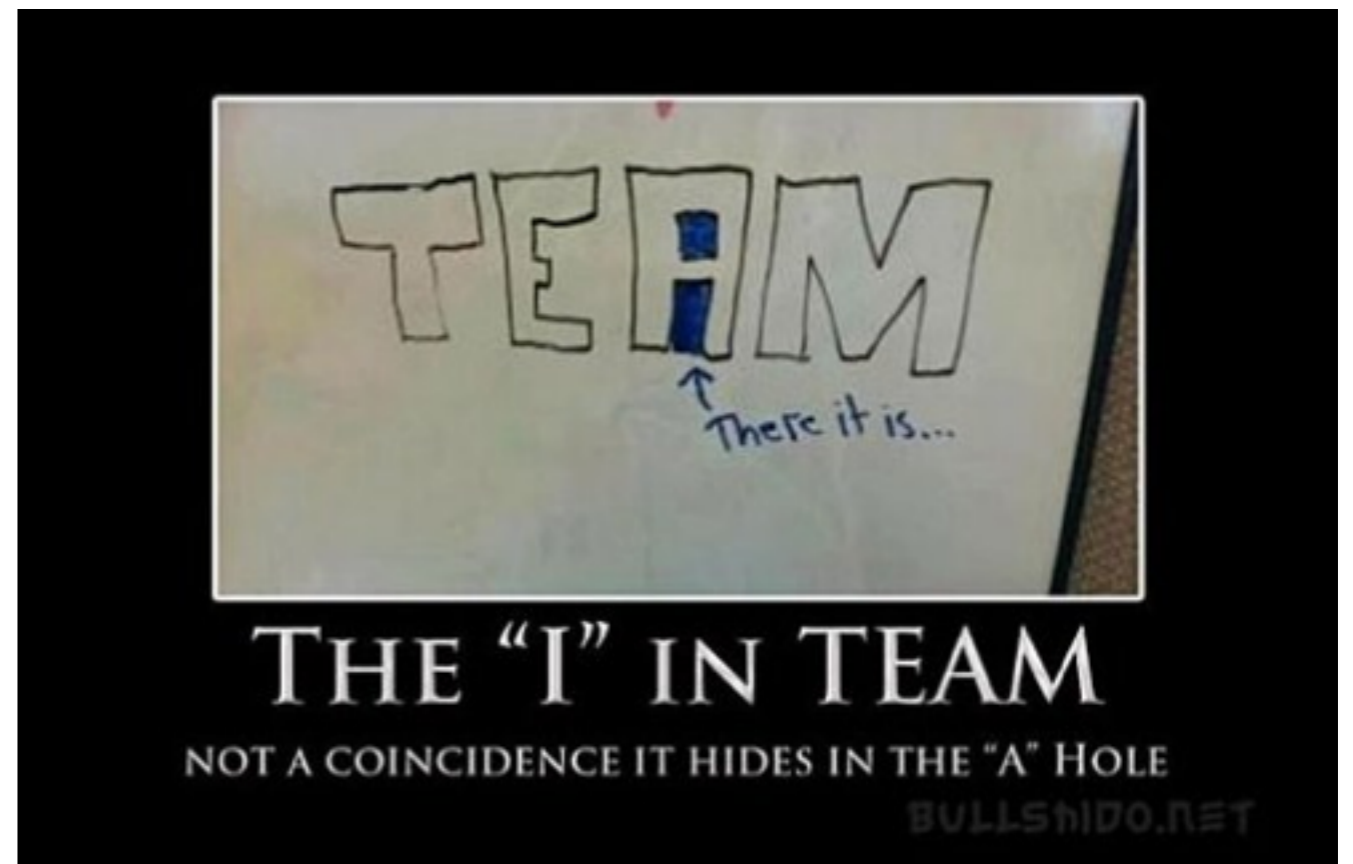
Stages of Group Development

- Groups behaviors change over time
- People need time to adjust
- You never stop storming/
norming
- What's performing after all?



Forming

- People mostly observe
- Conflict is generally avoided
- Analyzing the task and situation at hand



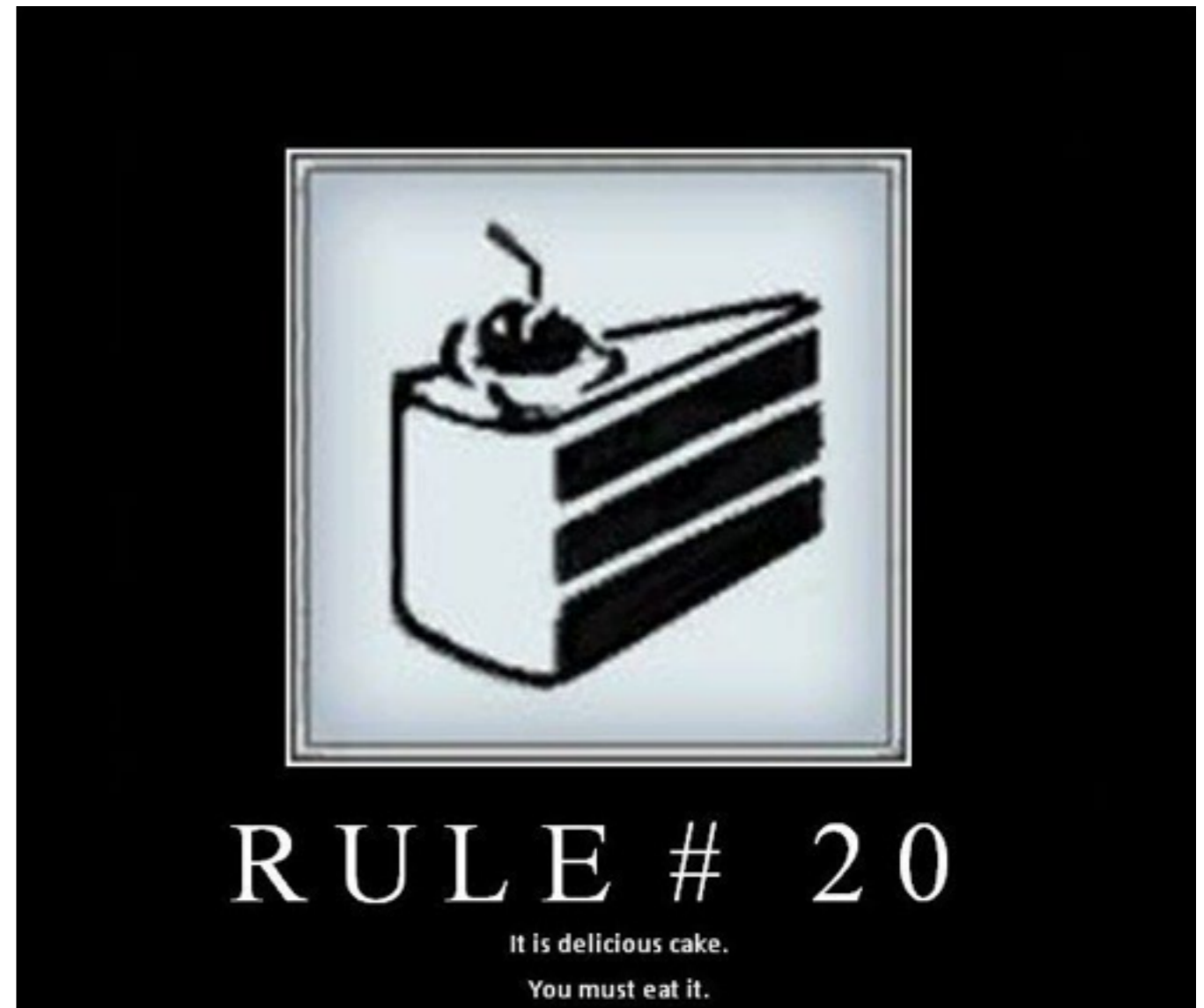
Storming

- Ideas compete for consideration
- Some teams never leave this stage
- Emphasize tolerance there
- Moderate this phase - Don't be directive



Norming

- The common goal is known
- Some original ideas are give up, others are improved on
- General norms (e.g. code-style, conflict resolution) are implemented
- Write this stuff down!
- Inform newbies



Performing

- Not every team reaches this stage
- No (or little) need for external supervision
- Dissent is welcome



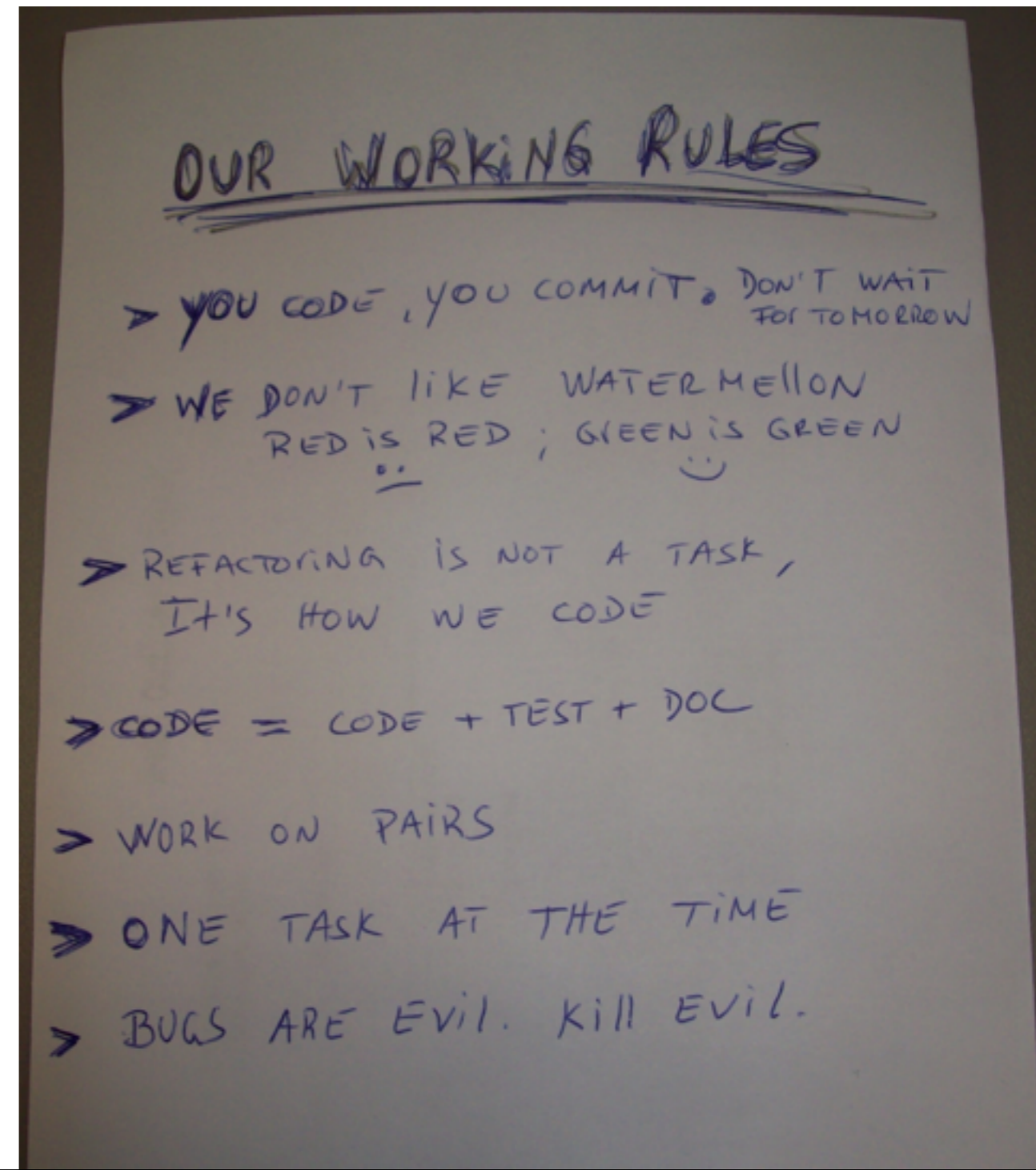
Back to Storming?

- Change in
 - Leadership
 - Tools
 - Team Members
 - Company Ownership
- Or mistakes made ;)



Practical Application

- Find out in what stage your team is
- Explicitly start norming stuff (by writing down the rules - everyone participates)
- Make a list of common conflicts and try to find norms with developers.



Motivation

“Control leads to compliance; autonomy leads to engagement.”

D.Pink

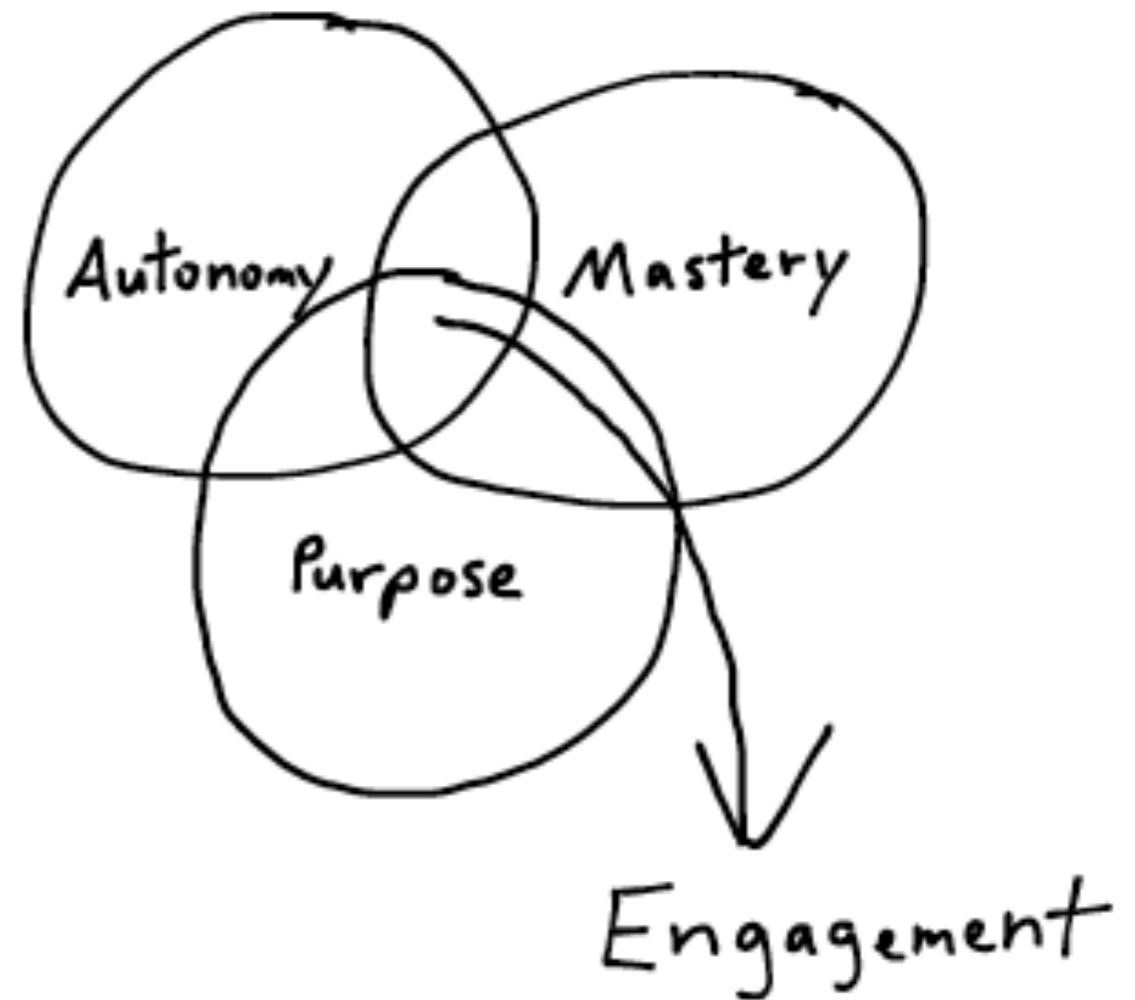
A-M-P

Autonomy-Mastery- Purpose

When it comes to motivation, there's a gap between what science knows and what business does.

Autonomy - Mastery - Purpose

- Carrots and Sticks are so last century
- Now that most basic needs are off the table it boils down to autonomy, mastery and a given purpose
- If all 3 are given, this leads to engagement without the need for constant pushing



Autonomy

- The desire to direct our own lives
- Choose the technology we like
- Change our workplaces
- Choose the right processes



Mastery

- The urge to get better at what we do
- A natural thing for every craftsman
- This is why we are here in the first place, right?



Purpose

- The goal that is greater than oneself
- It's not just making money for the company you work for
- This is what helps us create norms to adjust our very different expectations
- Explains: Why the f*** are we doing this?



Practical Application

- Make sure teams know the Purpose (Company Vision and Mission)
- Play „Delegation Poker“ to make sure team works as autonomous as possible
- Make sure you create an environment where people can deliberately learn (and fail)
- Remove any Carrots and any Sticks

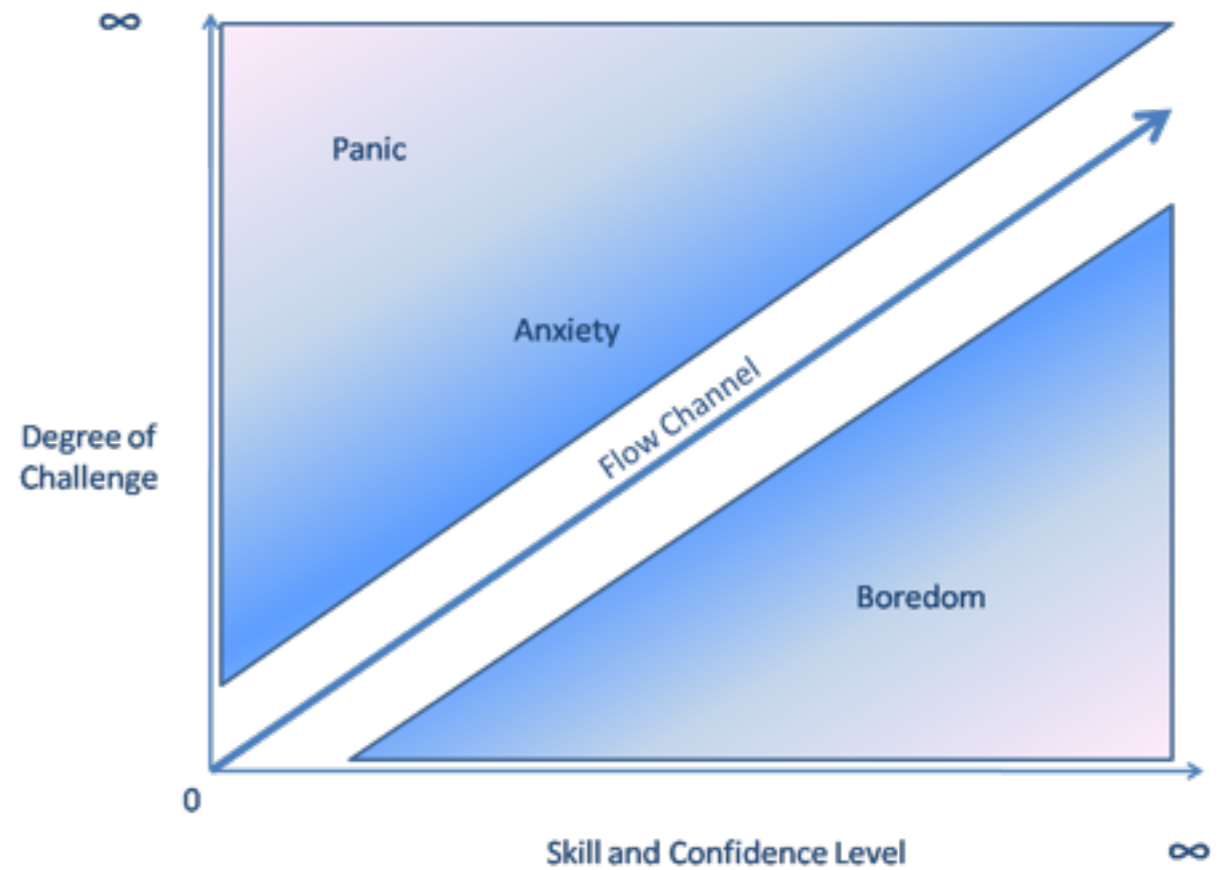


Flow

“...It is when we act freely, for the sake of the action itself rather than for ulterior motives, that we learn to become more than what we were.”

Flow State

- Also know as „in the Tunnel“
- State of concentration
- Fully immersed in the task at hand
- Less consciousness for the environment



How to reach flow state?

- Find a challenge
- Develop the skills in order to meet the challenge
- Set clear goals
- Focus completely on the task at hand
- Make sure you've set aside sufficient time
- Monitor your emotional state



How does Flow-State feel?

- You are completely involved in what you are doing
- There is a sense of ecstasy - being outside of reality
- You know what needs to be done and get immediate feedback
- You have the skills and know it's doable
- You lose sense of self and worries
- Losing track of time
- Intrinsic motivation - Producing flow becomes its own reward



Practical Application

- There is something called group flow. Hammer this in the head of your developers
- If something is too easy for you: Explain it to someone else
- If something is too hard: get some help
- If something is too boring for everyone: Automate it ;)
- Try a Fed-Ex Day!

Theory X - Theory Y

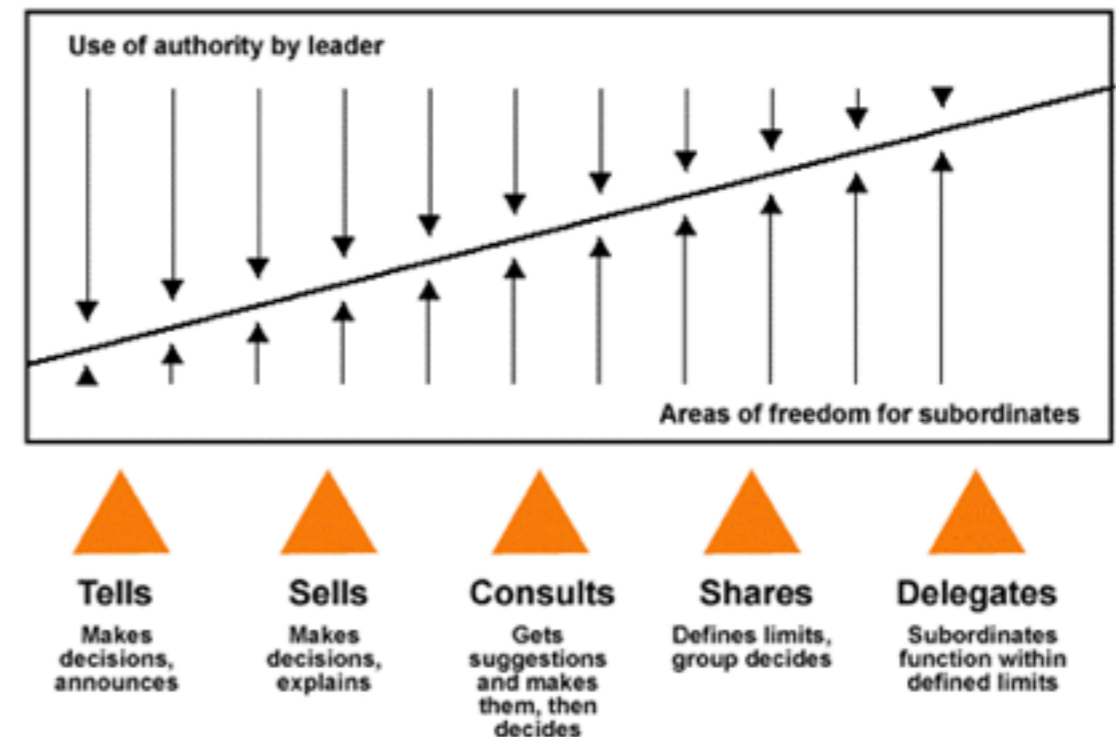
Theory X

- People are
 - lazy
 - avoiding work
 - inherently disliking work
- Therefore need to be
 - supervised
 - controlled
 - in a hierarchy
- Purely based on the idea of the „Homo Economicus“



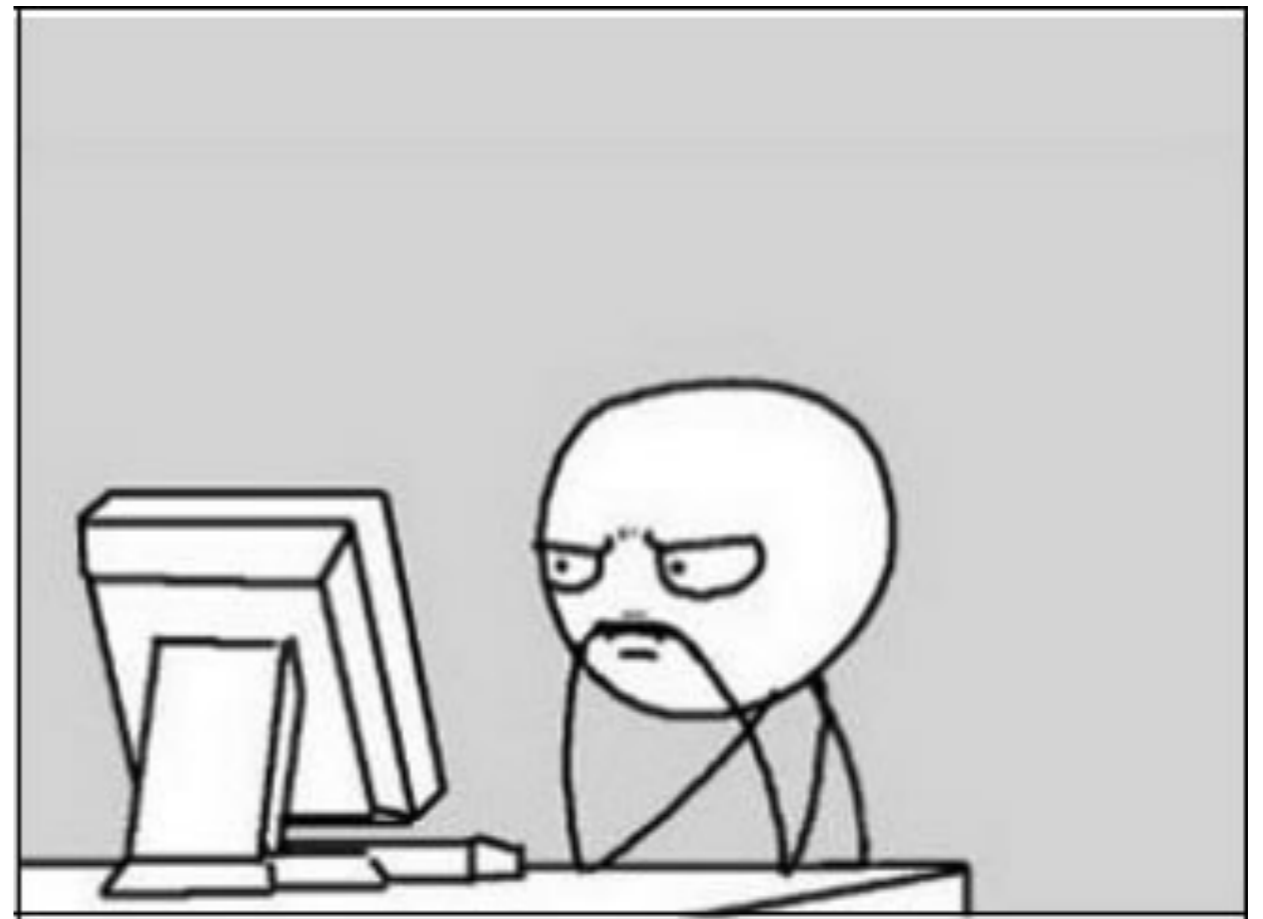
Theory Y

- People are
 - ambitious
 - self controlled
 - enjoy working duties
- Therefore they need
 - the right conditions to work
 - a goal to direct their energy to
 - a climate of trust



Practical Applications

- How much Y and how much X are you?
- How do you control after command?
- What would you need to become more Y?
- Why are you still keeping those type Y attitudes? (goto Non-Violent-Communication)



Conclusions!

Clicko ergo sum!

Now that you work with
people, you are back at
„Beginner Level“

RTFM!

No need to go „Neuroscience“
and „Brain Research“. Most of
the stuff is pretty old.

Get a Training (by a
Professional)

Get a Mentor/Coach!

Success = Look outside
the Window

Failure = Look in the mirror

Jim Collins

Thesis:

The Job as a Senior Developer or Team-Lead is about excelling in Technology, Architecture and Programming!

It's not!

It's about **people**, how to make them working together as **groups** and how to **motivate** them to do the right things the right way.

People

Groups

Motivation

People

Groups

Motivation

Questions?