



*Management:  
Doing the non-obvious!*

*Diego Pacheco*

# About me...



- ☐ *Cat's Father*
- ☐ *Head of Software Architecture*
- ☐ *Agile Coach*
- ☐ *SOA/Microservices Expert*
- ☐ *DevOps Practitioner*
- ☐ *Speaker*
- ☐ *Author*



*diegopacheco*



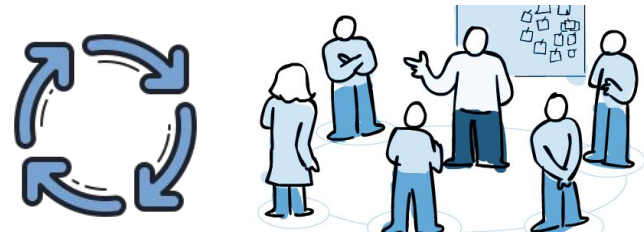
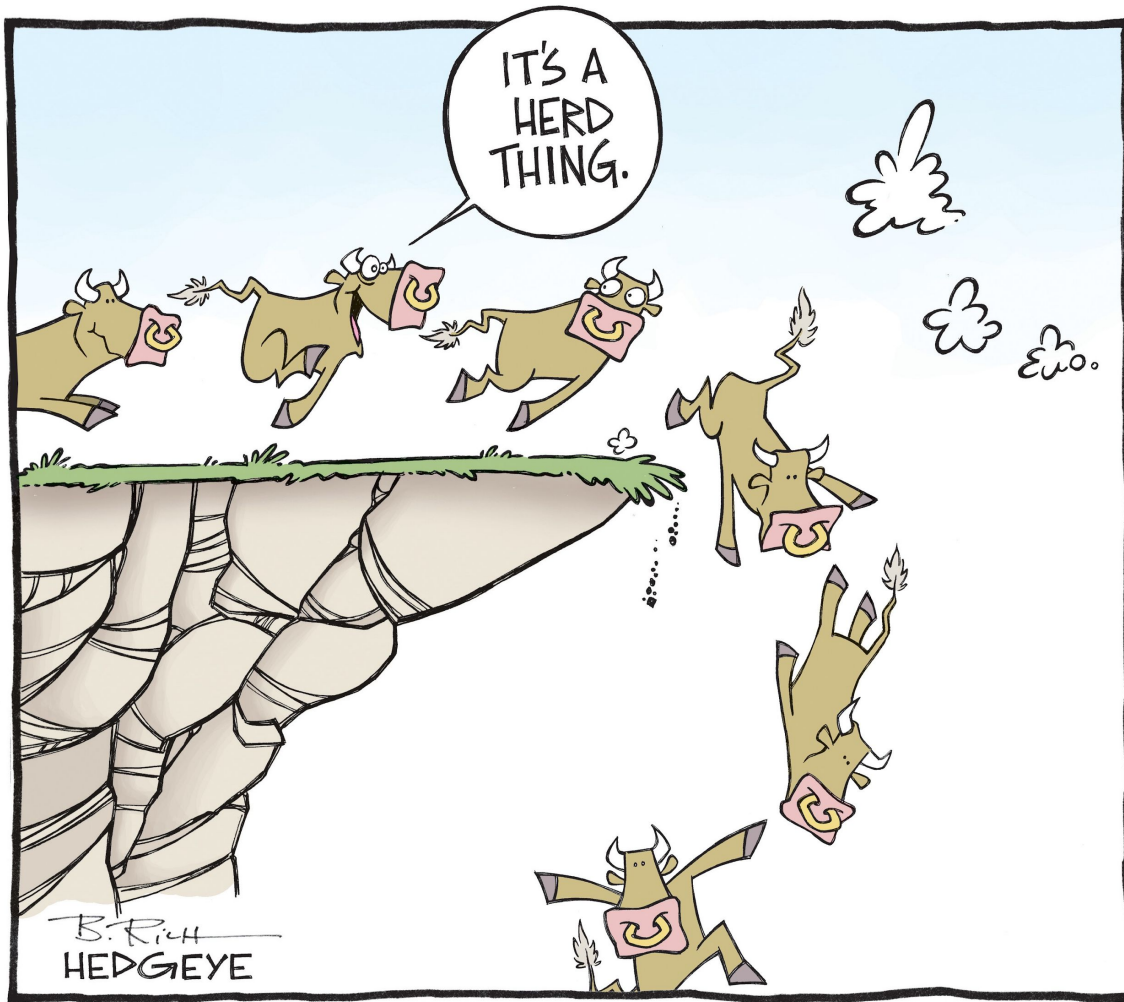
*@diego\_pacheco*



*<http://diego-pacheco.blogspot.com.br/>*



<https://diegopacheco.github.io/>



**“Discipline is doing  
what you don’t want  
to do when you  
don’t want to do it.”**

Posted in the wrestling room at  
Graham High School, St. Paris, OH



# Leadership > Management



# Forget Simple

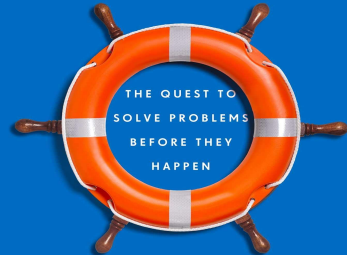




# Tunneling Effect (Top Performance can be Gamed)



Upstream



DAN HEATH

New York Times bestselling coauthor of  
Made to Stick and Switch

# Ego is the Enemy

- ❑ Ego: is unhealthy to believe in our own importance.
- ❑ Ego Distocerce the reality around us.
- ❑ Ego is the enemy of Building, maintaining, and recovery (Innovation)
- ❑ Removing ego widens our perspective.
- ❑ Having a clear head is like sobriety (no ego).
- ❑ I.e: Kirk Hammett(Metallica/Satriani), Ken Shamrock(+,-,=)
- ❑ Solution: Suppress/Channel Ego early with Humility





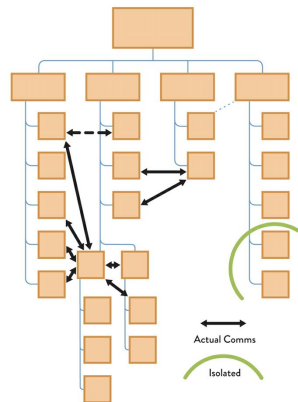
*Require everybody communicate with everybody is  
a recipe for mess.*



# Org Chart != Innovation



**Figure 1.2: Obstacles to Fast Flow**



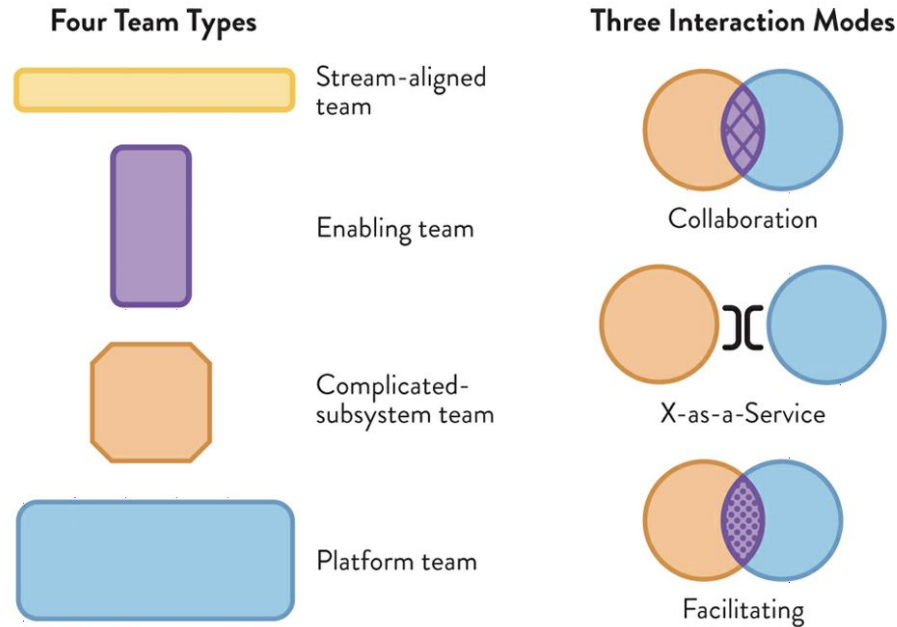
**Figure 1.1: Org Chart with Actual Lines of Communication**

In practice, people communicate laterally or “horizontally” with people from other reporting lines in order to get work done. This creativity and problem solving needs to be nurtured for the benefit of the organization, not restricted to optimize for top-down/bottom-up communication and reporting.

Org Chart Thinking Is the Problem



# Teams as ways to model communication and Architectures



**Figure 0.1: The Four Team Types and  
Three Interaction Modes**

*Pre-meetings before meetings == waste of time*



TRUST





# Why Feedback is hard



- ❑ Feedback is hard - most of people and companies just give up.
- ❑ People don't want to give feedback because often, the result is a catastrophe.
- ❑ Learning about ourselves can be painful.
- ❑ Trigger reactions - we need to understand our reactions (you cannot deal with fire by shutdown fire alarm)
- ❑ Sometimes 90% of feedback is wrong but that 10% could be enough for you to grow.
- ❑ Decouple the Feedback from the relationship (we never do that).

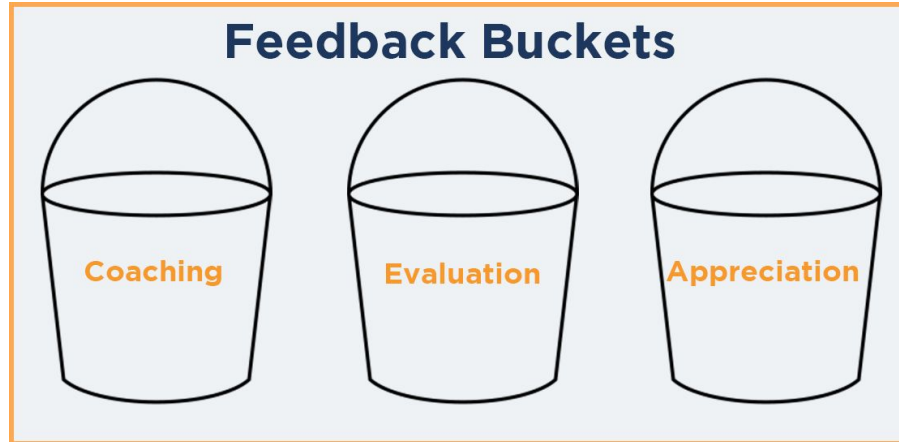
# Why Feedback is hard



- ❑ *When giving feedback we feel right. When we receive feedback feels wrong.*
- ❑ *The better you understand them better you make it useful for you or to understand why you are being misunderstood.*
- ❑ *Emotion trigger derail feedbacks (disqualify by HOW/WHEN/WHO did it).*
- ❑ *IF anything the person says would trigger something on you - maybe you should say your thing first.*
- ❑ *Emotions distort the feedback.*

# Types of Feedback

- Feedback could mean: Appreciation, Coaching, and Evaluation.
- When asking feedback you can ask what you need.



# Ask for advice - might get feedback.



- ❑ *Laughing about yourself makes things much easier (destroy anxiety and takes much less energy).*
- ❑ *Learning from feedback is also how we hold our identity to make us more resilient from feedback adverse.*
- ❑ *Move from fixed mindset to growth mindset - give up simple labels and cultivate complexity.*
- ❑ *Simple Labels == All or Nothing (Smart or Stupid)*
- ❑ *Growth vs fixed mindset - is not who you are but what you do.*



# Debugging Teams



- ☐ You will need to change scope!
- ☐ There is a way to get more with less?
- ☐ Sometimes compromises need to be done?
  - ☐ Schedule
  - ☐ Technical solution
- ☐ It's a coaching issues(Missing guidance?)
- ☐ It's a relationship issue?
- ☐ It's lack of skills or personal issue going on?

# Pre-Modertens to find issues

- ❑ Scenario modeling
- ❑ We can imagine issues before they happen
- ❑ This can be useful to detect problems before they happen
- ❑ Some ideas:
  - ❑ Would the software scale?
  - ❑ Would the dependencies honor the contract?
  - ❑ Would we be able to rollback?
  - ❑ Would work without parameter  $x,y,z$ ?
  - ❑ What is dependency  $X$  is not done?

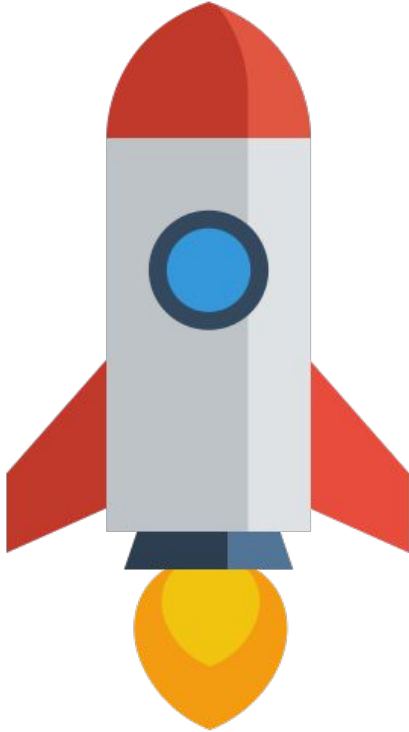


# Role Playing to Train Leaders

- ❑ *Coders do dojos and managers do what(... nothing)?*
- ❑ *How we train leaders? Only through experience?*
- ❑ *No. We can do role playing exercises.*
- ❑ *Easily see the struggles in training before they happen in real life.*
- ❑ *Easy start: Use cases + Questions*
- ❑ *Full role playing games(effective and fun)*



*Release First, fix the bug later*



**VS**





# Policies Vs Judgement (Netflix way)

- ❑ Netflix risk is lack of innovation. However, Netflix does not have innovation Fridays instead they give employees space
- ❑ Don't seek to please your boss - do what is better for the company
- ❑ Treat people as grown ups
- ❑ Expect people to have amazing judgement
- ❑ No Need to have policies for everything
- ❑ Metaphor is a high sports team, not a family.
- ❑ Keeper Test: Would you try to save person X if they quit tomorrow? If not you might replace the person now.



# Policies Vs Judgement (Netflix way)

- ❑ 4A Feedback Guidelines: Aim to Assist, Actionable, Appreciate, Accept or Discard.

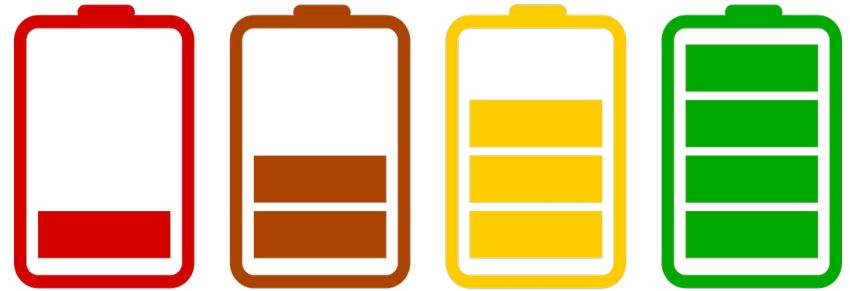
CORPORATE  
REBELS

# FEEDBACK

1. AIM TO ASSIST 2. ACTIONABLE 3. APPRECIATE 4. ACCEPT OR DISCARD



# Emotional Bank Account



# Win-Win or NO DEAL!

- ❑ What I want is not more important than my relationship with you. Short term win could be a long term loss.
- ❑ In the long term if it is not a win-win for both will be a win-lose or lose-lose.
- ❑ Focus on interest, not on positions.





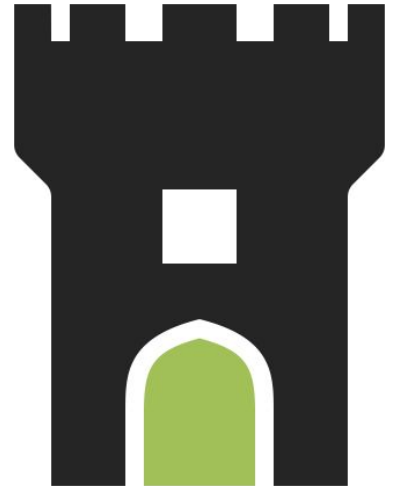
# Win-Win or NO DEAL!

- ❑ Listen twice as much as you speak.
- ❑ Principled Negotiation (Roger Williams)
- ❑ Look for Options for mutual gain:
  - ❑ 1 - See the problem from another point of view
  - ❑ 2 - Identify the key issues and concerns (not positions)
  - ❑ 3 - Determine what results would be acceptable
  - ❑ 4 - Identify new options for results.



# *The Tower - Fight shallowness*

- ❑ *1922 Carl Young retreat (The tower)*
- ❑ *2h non-interrupted writing (no electricity) + long walks*
- ❑ *Deep work necessary to improvement*
- ❑ *Only way to learn or do complex work*
- ❑ *Depth Destroy behavior:*
  - ❑ *Immediate email response*
  - ❑ *Social media presence*



# *The Tower - Fight shallowness*

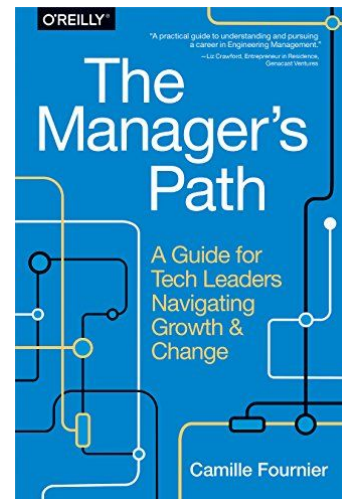
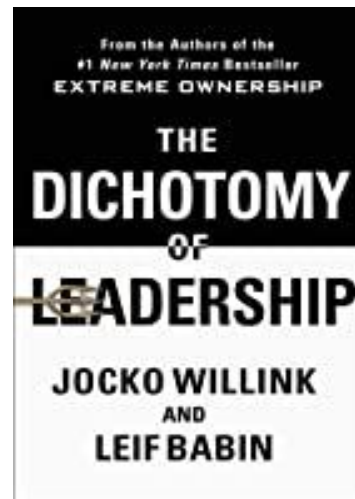
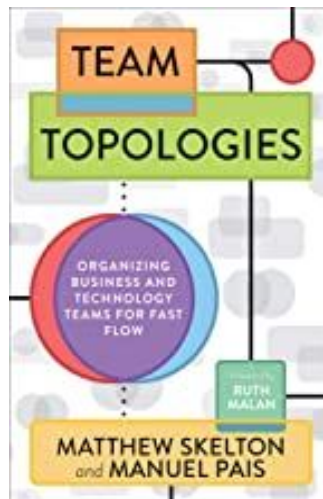
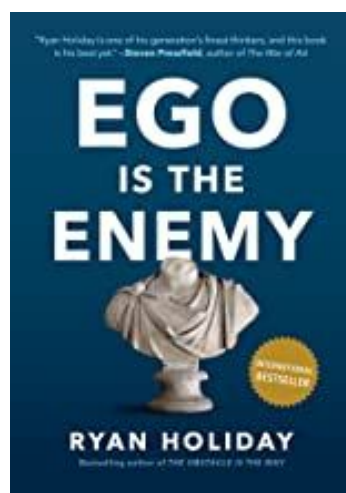
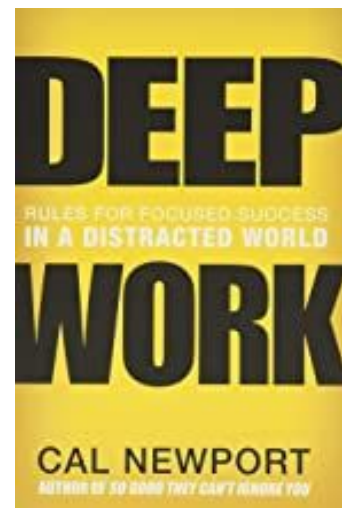
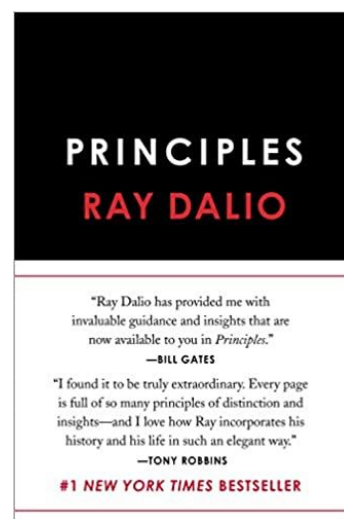
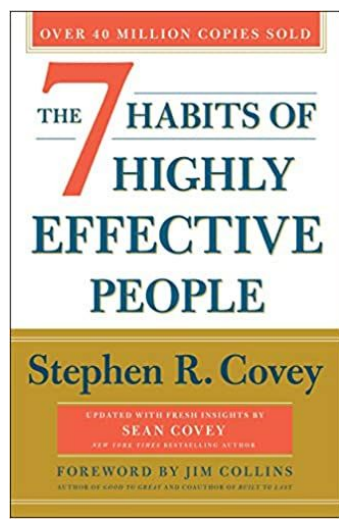
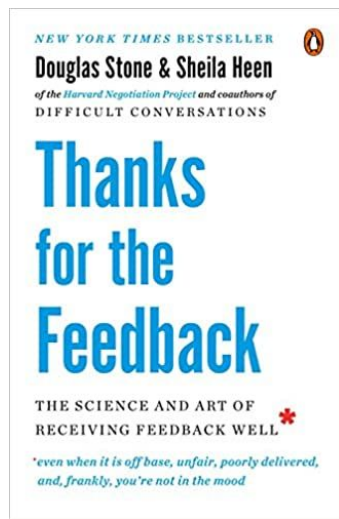
- ❑ *Free time is unstructured and requires more work to be enjoyed*
- ❑ *Dive Deeper rather than keep answer emails faster like your exhausted peers. (i.e Amazon Deep Dive Principles)*
- ❑ *Internet sabbath*
- ❑ *Schedule Internet usage at Home and at Work.  
Even after work.*



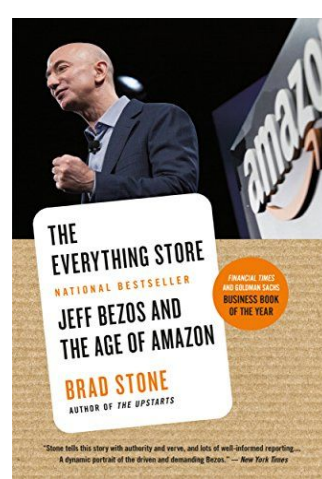
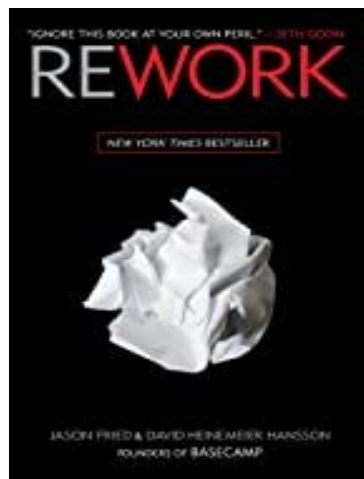
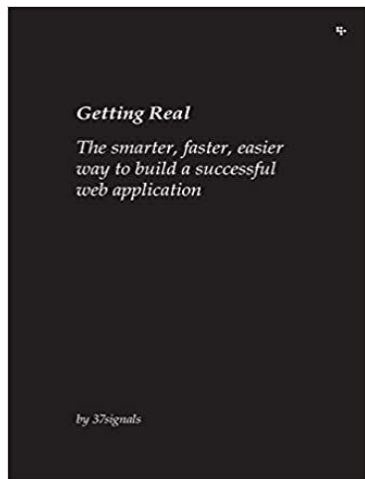
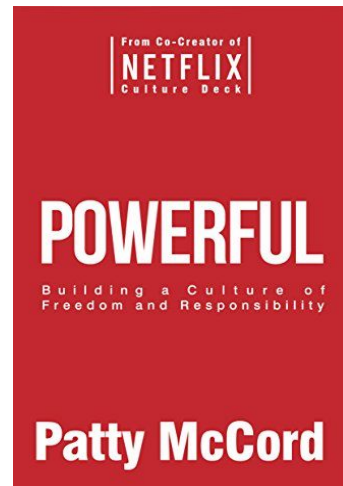
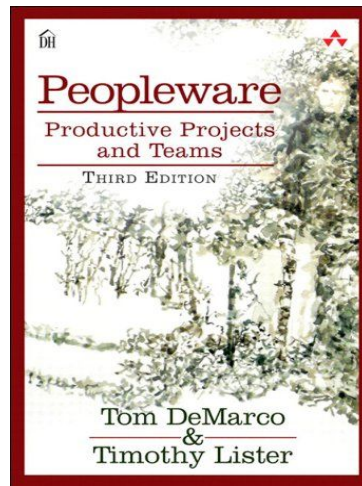
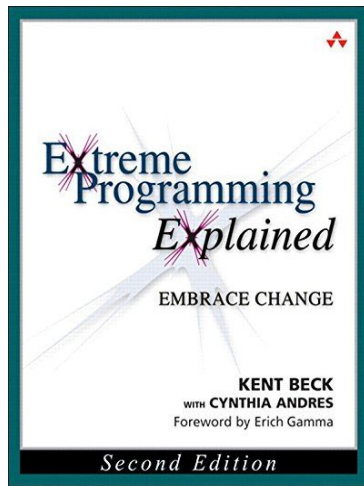
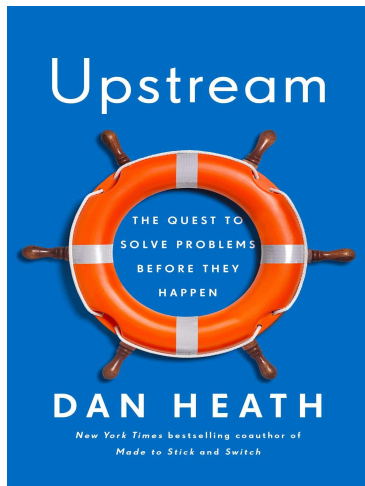
# The Tower - Fight shallowness

- ❑ How long can you do Deep work in a day? Beginner = 1h a day, used to it = 4h but not more.
- ❑ Schedule every minute of your day.
- ❑ Make yourself hard to access.
- ❑ Offices are Interruption Factories
- ❑ Favor Async(37 signals)
- ❑ Urgency != Anxiety













*Management:  
Doing the non-obvious!*

*Diego Pacheco*