Antifragility

add method to the madness of software development

jovan@mqsoft.rs

Fragility

agile on waterfall budget:

- flexible scope

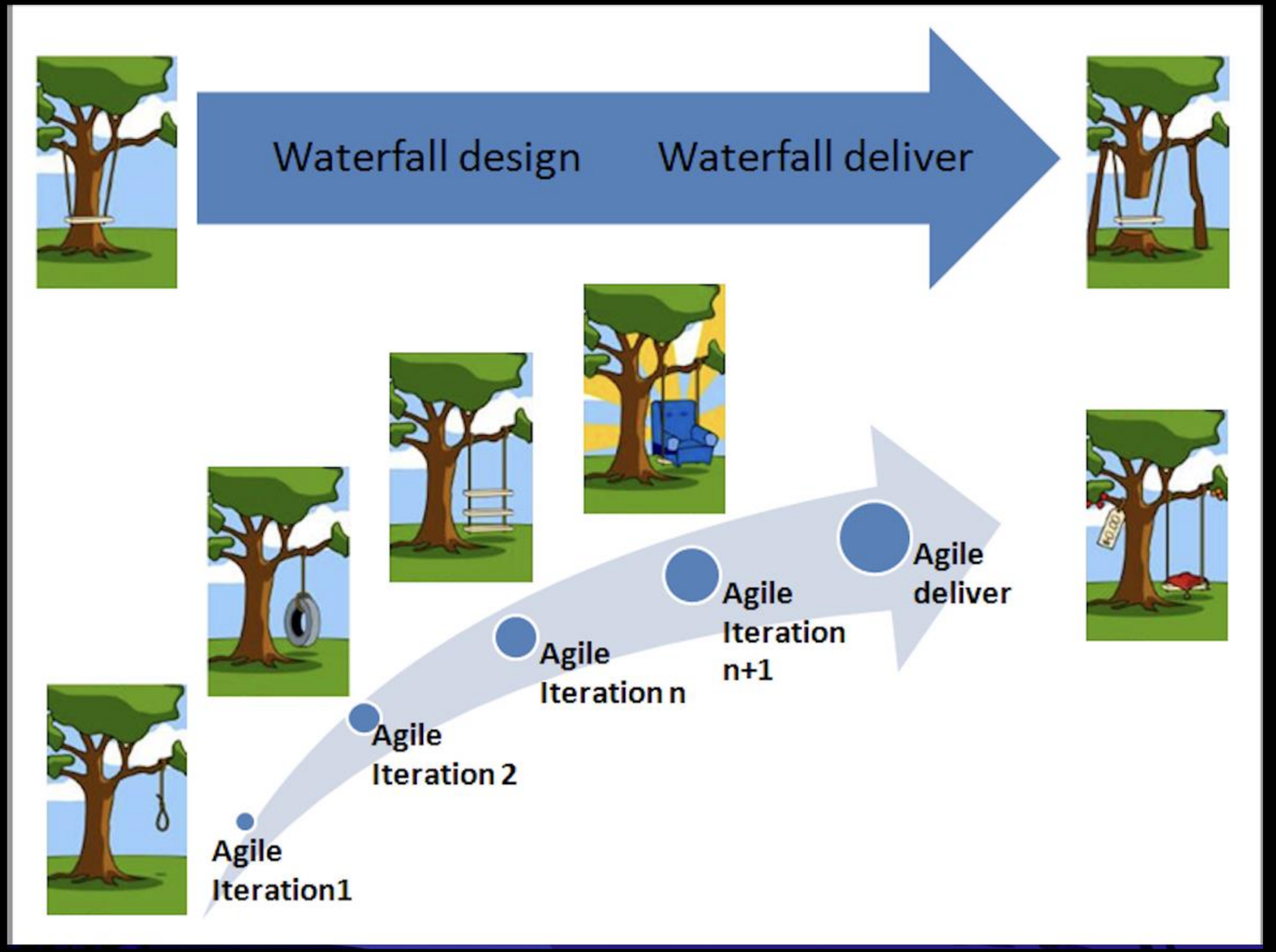
- fixed price

- fixed time

Why Agile?

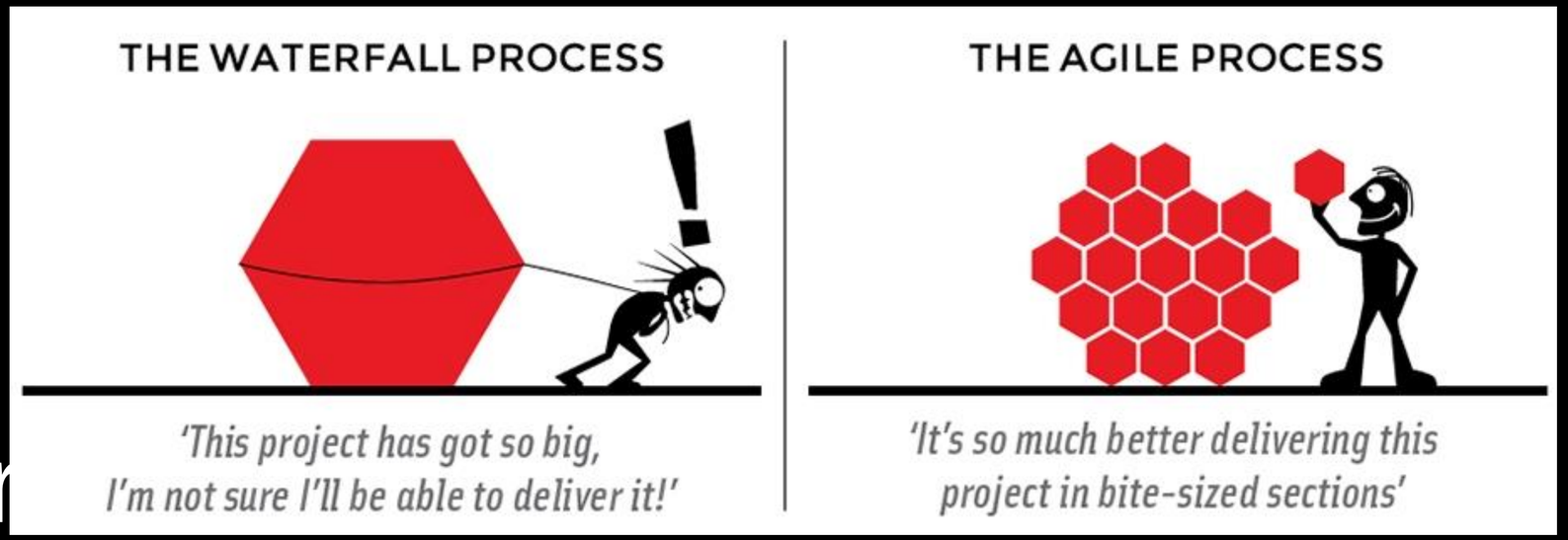
challenges:

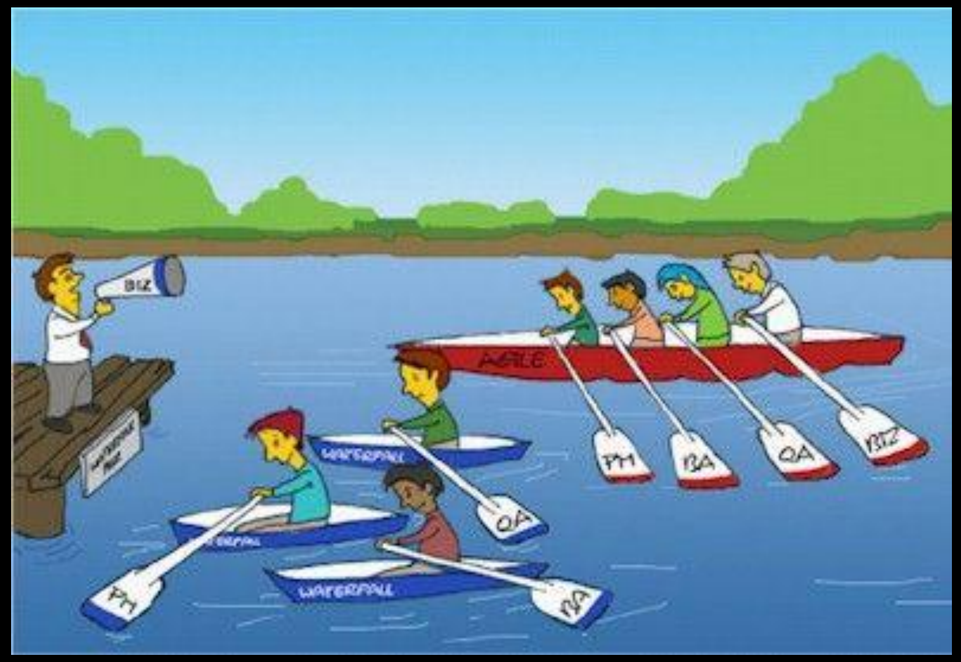
- I will know it when I see it
- FOMO
- Estimations
- Parkinson's law
- Definition of Done Done
- Backlog



What to take from SCRUM?

- divide-and-conquer or "big" vs "small"
- let's do it together or no blaming allowed
- User Stories
- small team
- short meetings

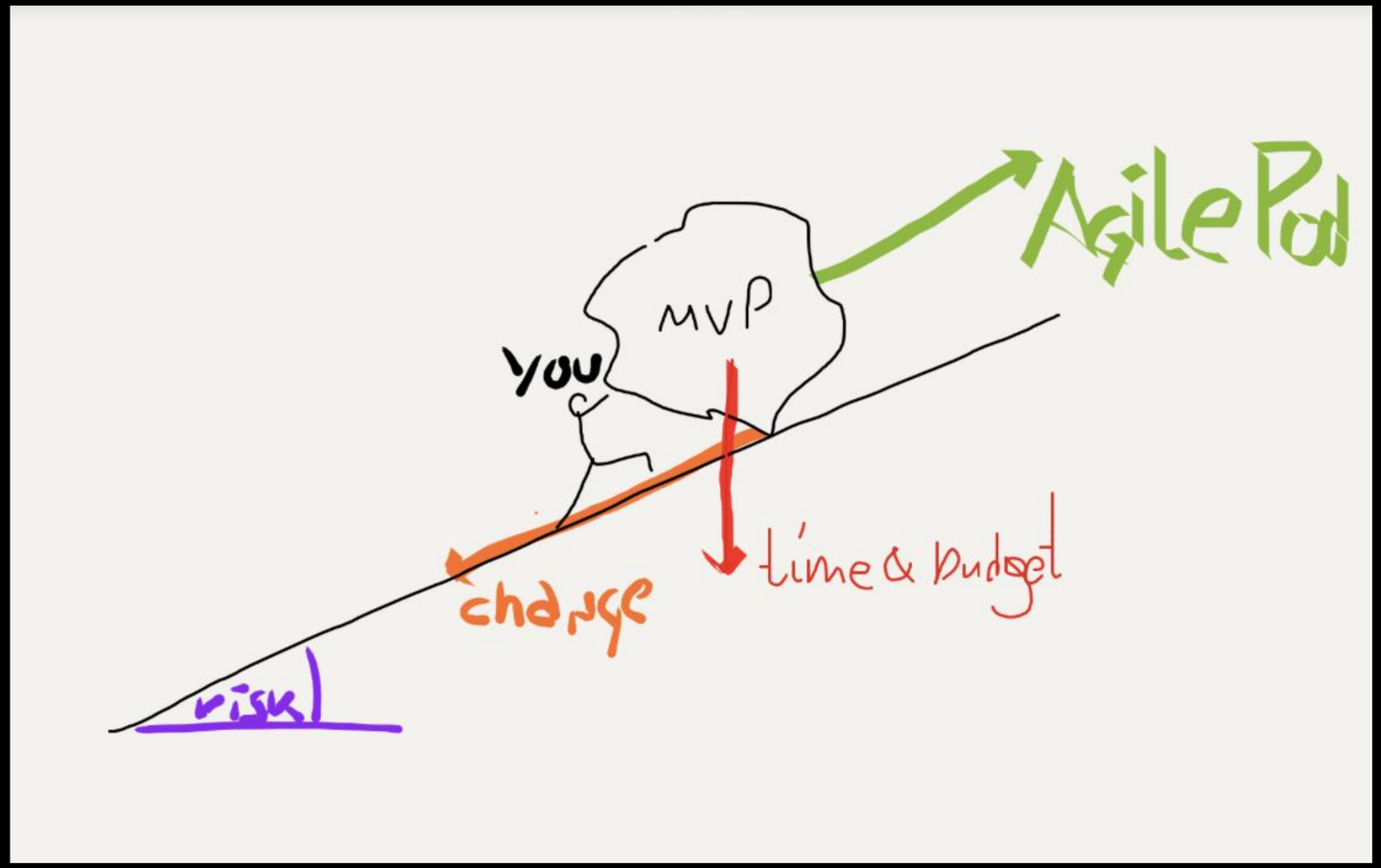




Antifragility

Antifragile system "loves erros". Software engineers do not.

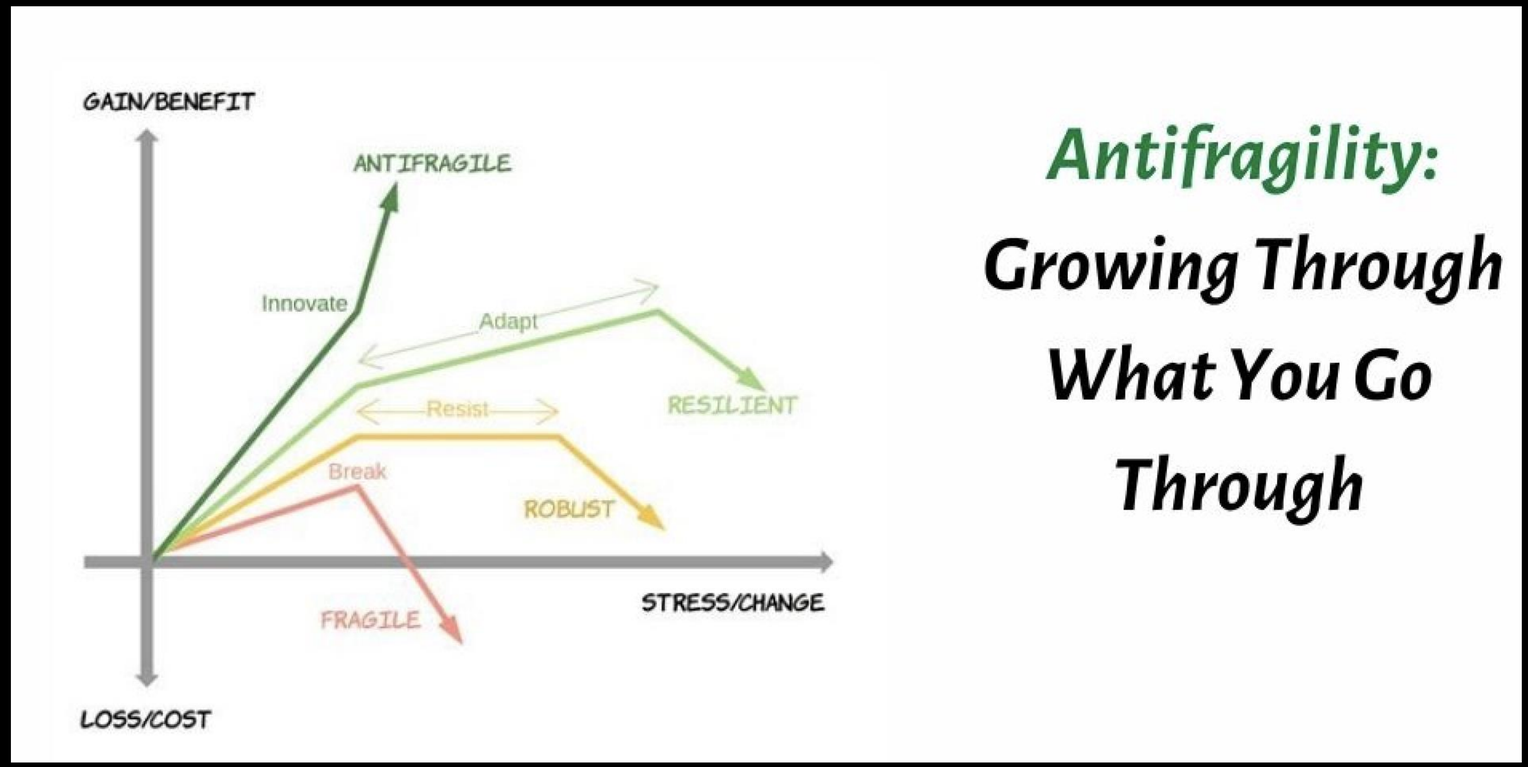
- **Fragility** is the tendency of the software to break in many places every time it is changed.
- **Antifragility** is a property of systems in which they increase in capability to thrive as a result of stressors, shocks, volatility, noise, mistakes, faults, attacks, or failures.
- Antifragility is fundamentally different from the concepts of **resiliency** (i.e. the ability to recover from failure) and **robustness** (that is, the ability to resist failure).



Antifragility

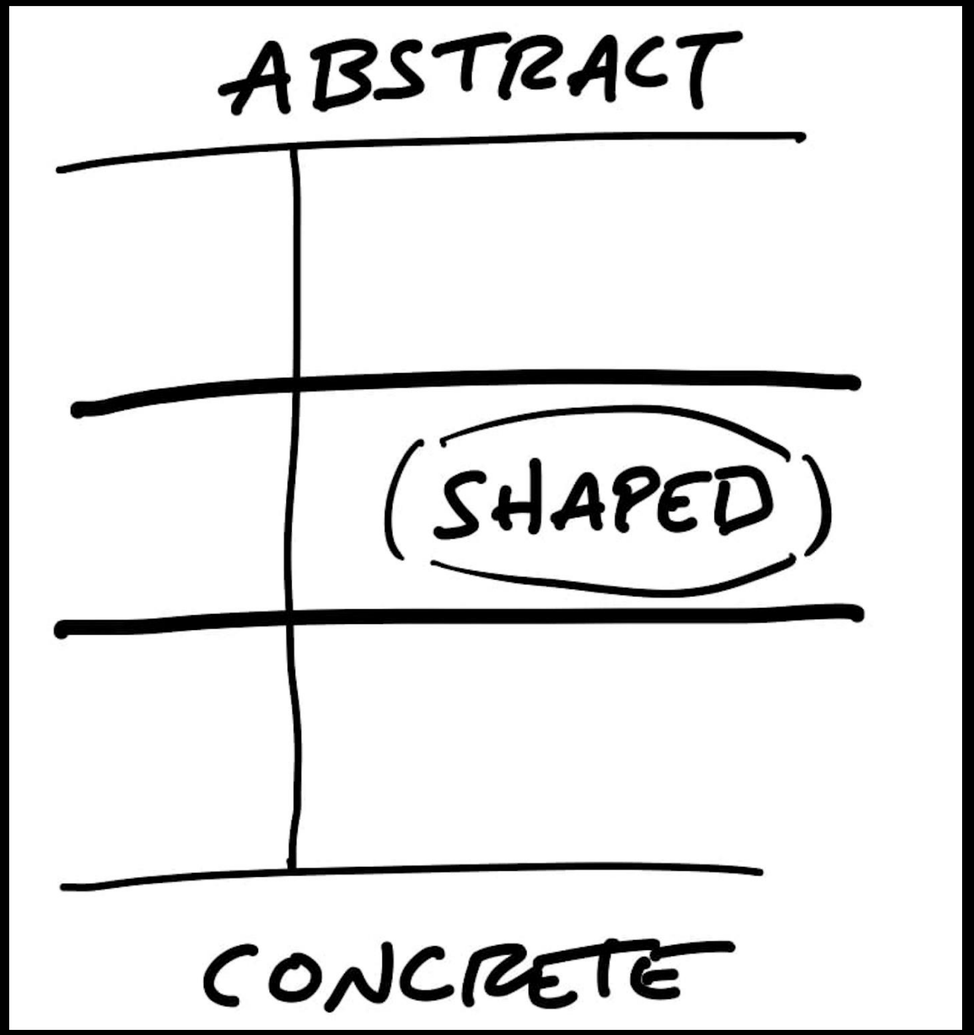
Kintsugi & Kaizen

To move on to becoming **antifragile** then requires a combination of low fixed obligations with making small bets that have asymmetric payoffs.



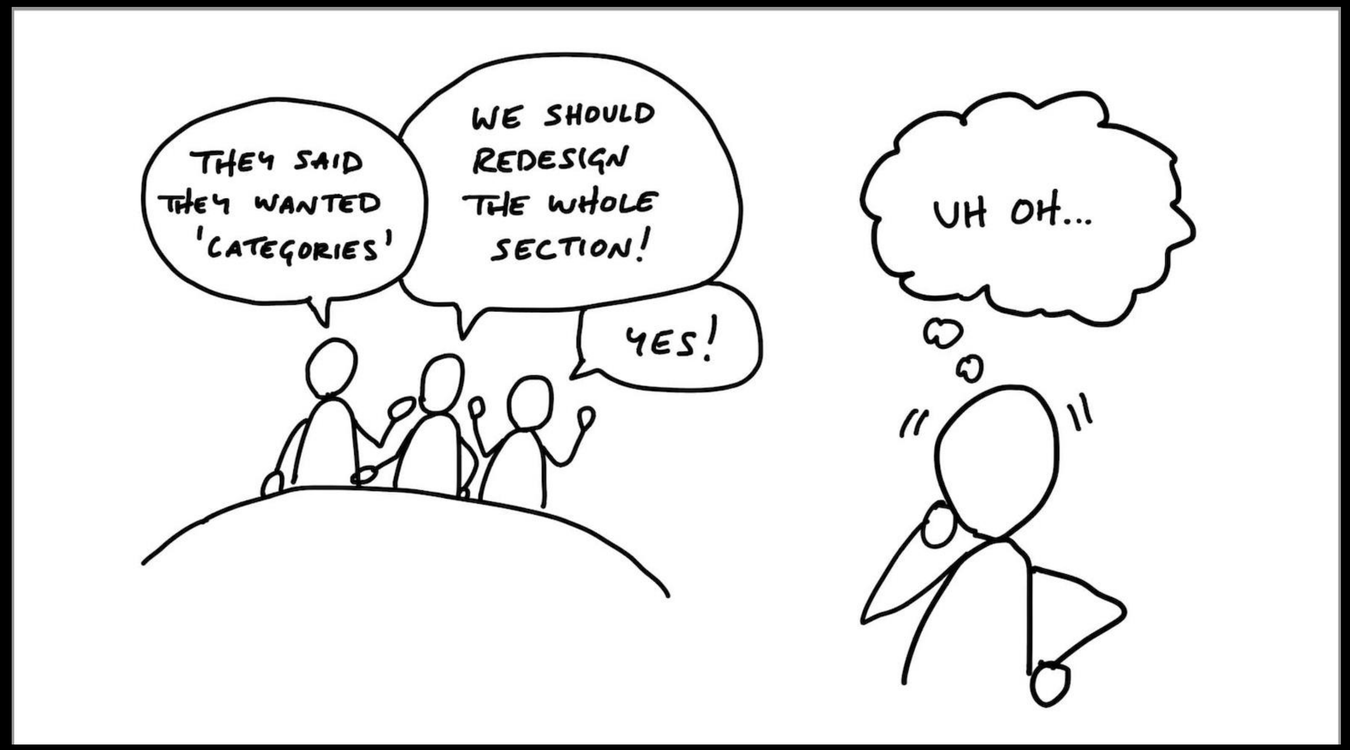
Shape Up

- wireframes are too concrete
- words are too abstract
- use "The OneThing" approach
- JOMO instead of FOMO



Setting the Appetite

An appetite is completely different from an estimate. Estimates start with a design and end with a number. Appetites start with a number and end with a design. We use the appetite as a creative constraint on the design process.



Sprints?

- 3 week Sprint
- .. but do Sprint Review at the end of every second Sprint
- work on DEV in Sprint N
- work on UAT in Sprint N+1
- do reviews on PrePROD

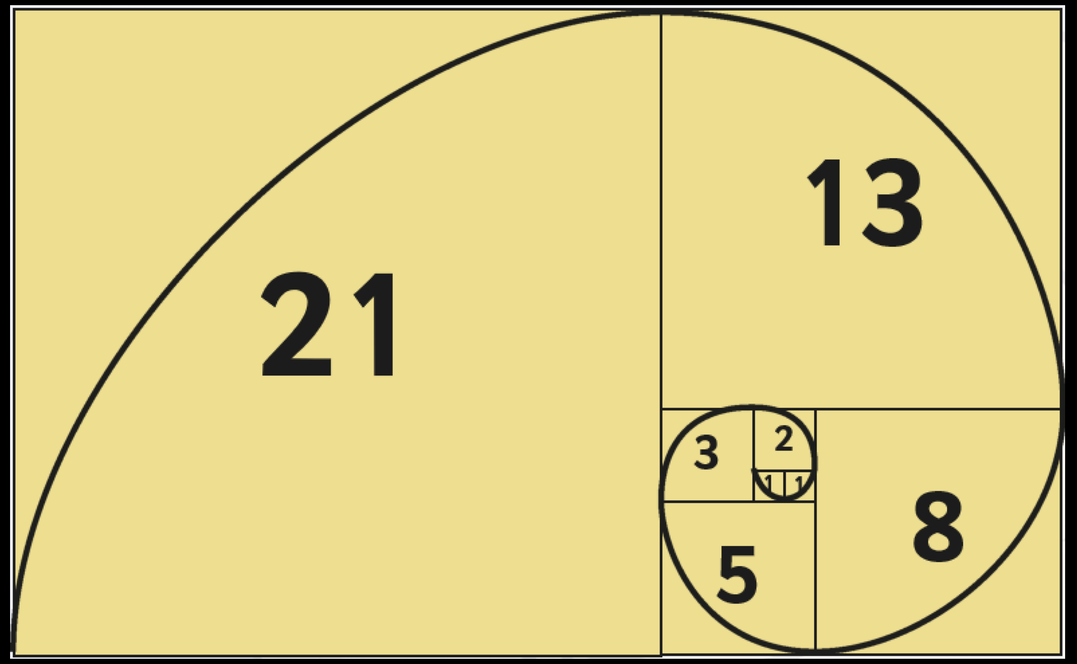
NOTASAP

The expectation of immediate response is everywhere. Real-time everything isn’t human-scale, yet that’s how so many work and communicate these days. Not us. We think urgency is overrated, and ASAP is poison. Real-time is the wrong time most of the time.

👉 https://37signals.com/

Budget?

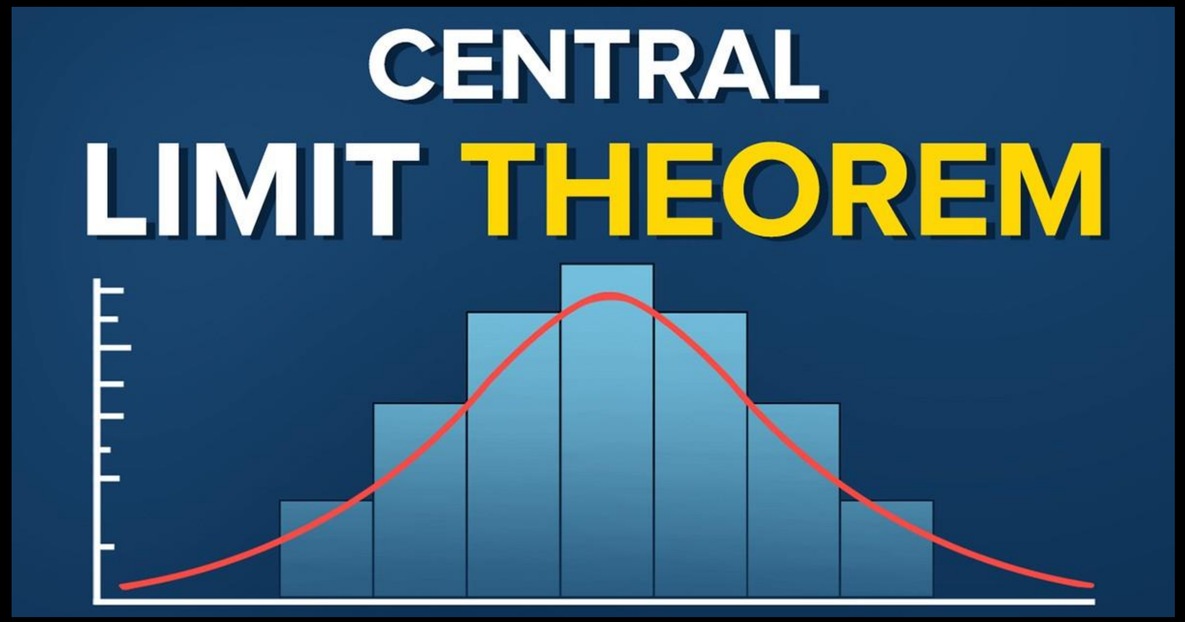
- pick a Fibonacci number that feel like You can afford to spend working on Demo version
- skip one number and the next is the time You allocate do make a version worth of formal User Acceptance Testing
- end the next one is the time You allocate to work on the Project
- pick weight (day, week, month, Sprint) of the number: that is Your complexity
- You got big value You don't like: split the Project in phases
- use minimal team



Art of Wu Wei

don't stress: the project IS out of control

- Discovery vs Invention
- Do not Force
- .. but put some pressure
- happy people do great stuff!

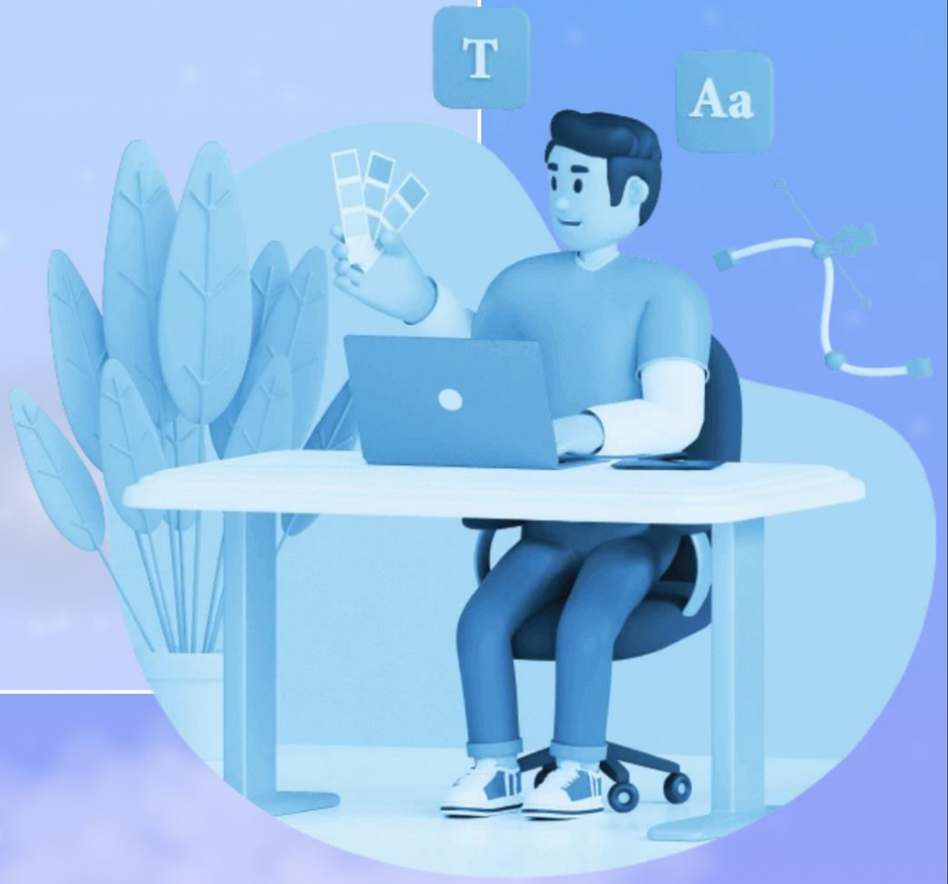


MQ SOFT

Making Quality Software is Hard

You need end to end software development, outsourcing or consulting?

We can help



Thank You