

Decision Dials

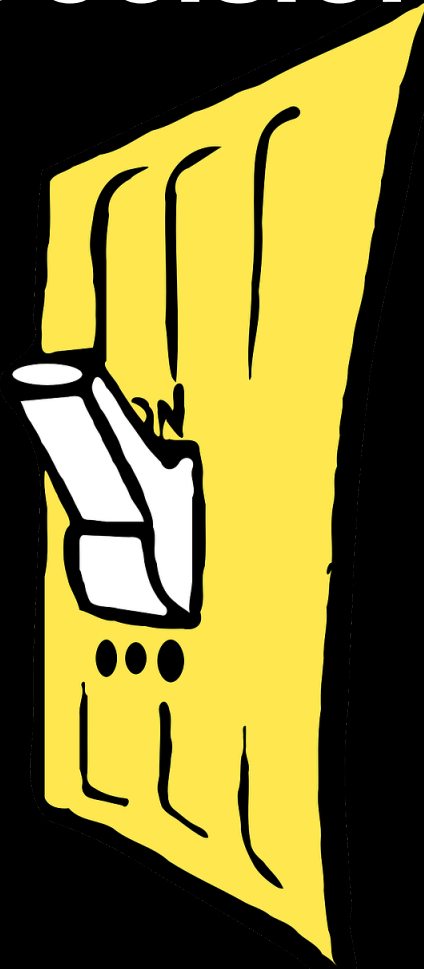
Venkat Subramaniam

venkats@agiledeveloper.com

[@venkat_s](#)

Decisions

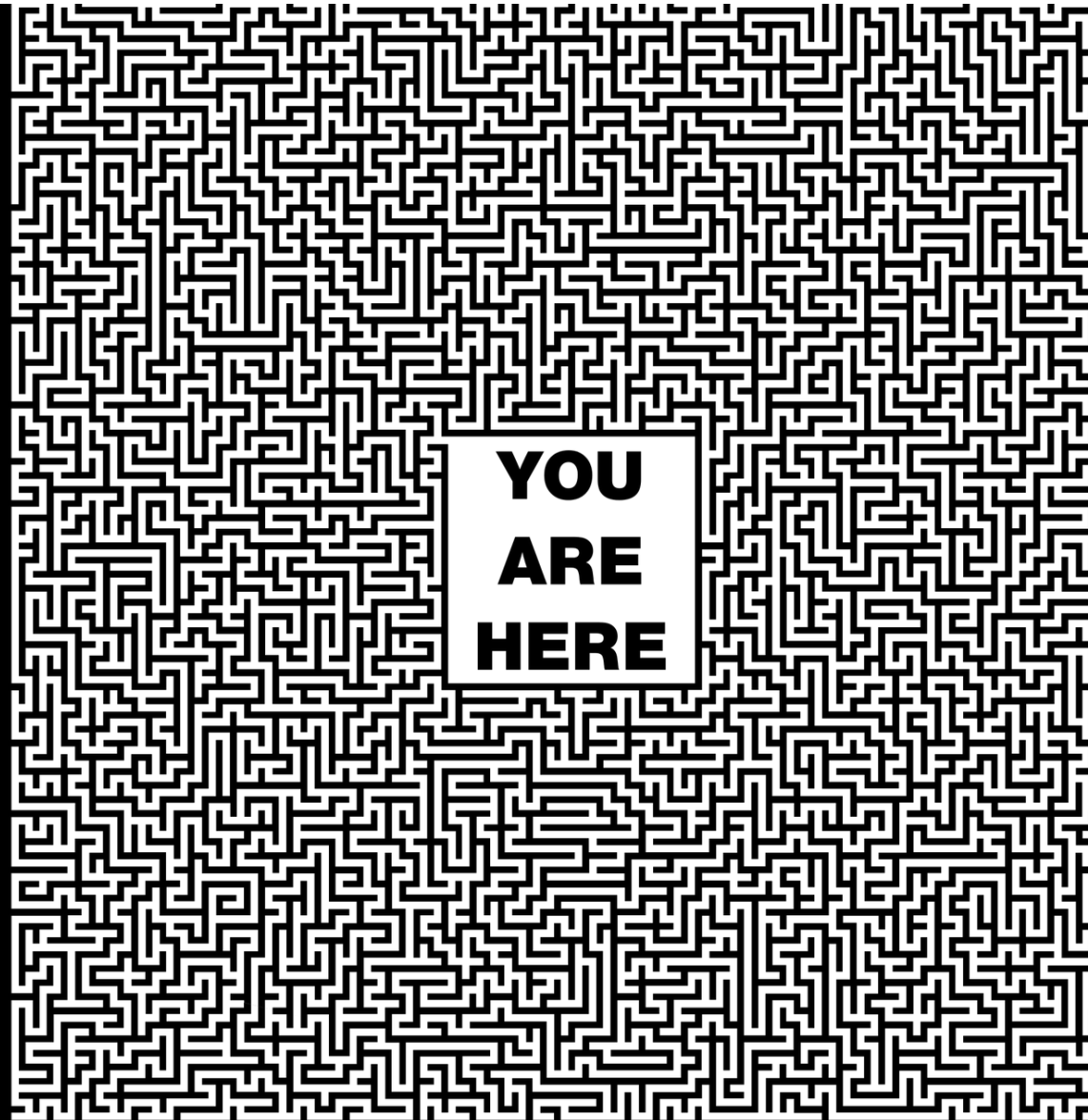
Decisions



Decisions are Hard



**Environment,
where you
are, can
make it hard**



We need to have the will to change our decisions

Those who compare building construction to software development understand neither.







Some Decisions are Highly Critical

Some Decisions are Highly Critical



The *over* orientation



The *under* orientation

https://en.wikipedia.org/wiki/Toilet_paper_orientation

Some Decisions are Highly Critical

Inc.

NEWSLETTERS

SUBSCRIBE



PRODUCTIVITY

The Correct Way to Hang Toilet Paper, According to Science

Hanging it the wrong way
can result in absenteeism, worker's comp payments and
even business-busting lawsuits. 

BY GEOFFREY JAMES, CONTRIBUTING EDITOR, INC.COM @SALES_SOURCE

<https://www.inc.com/geoffrey-james/the-correct-way-to-hang-toilet-paper-according-to-.html>

Some Decisions are Highly Critical

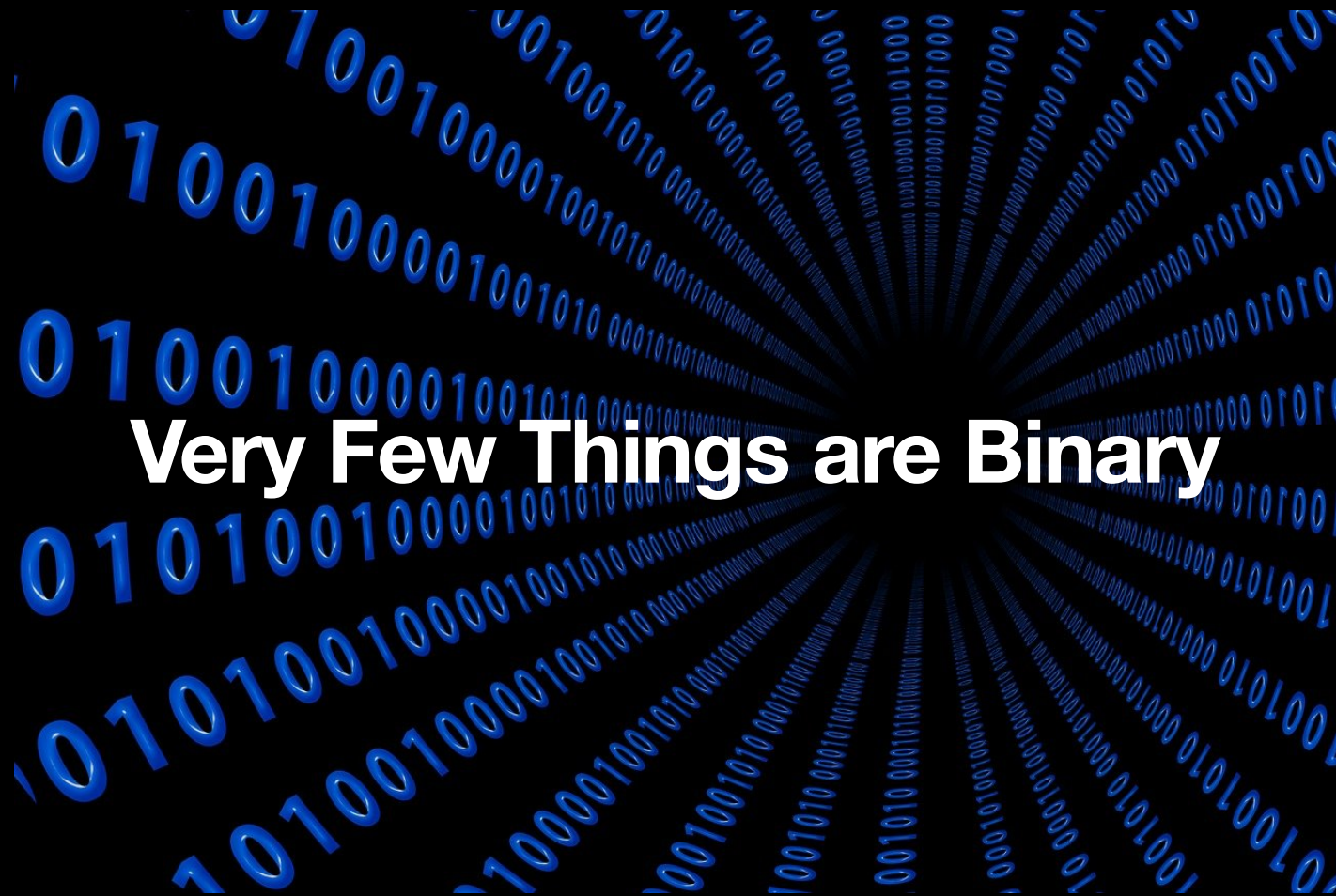


The *over* orientation



The *under* orientation

https://en.wikipedia.org/wiki/Toilet_paper_orientation



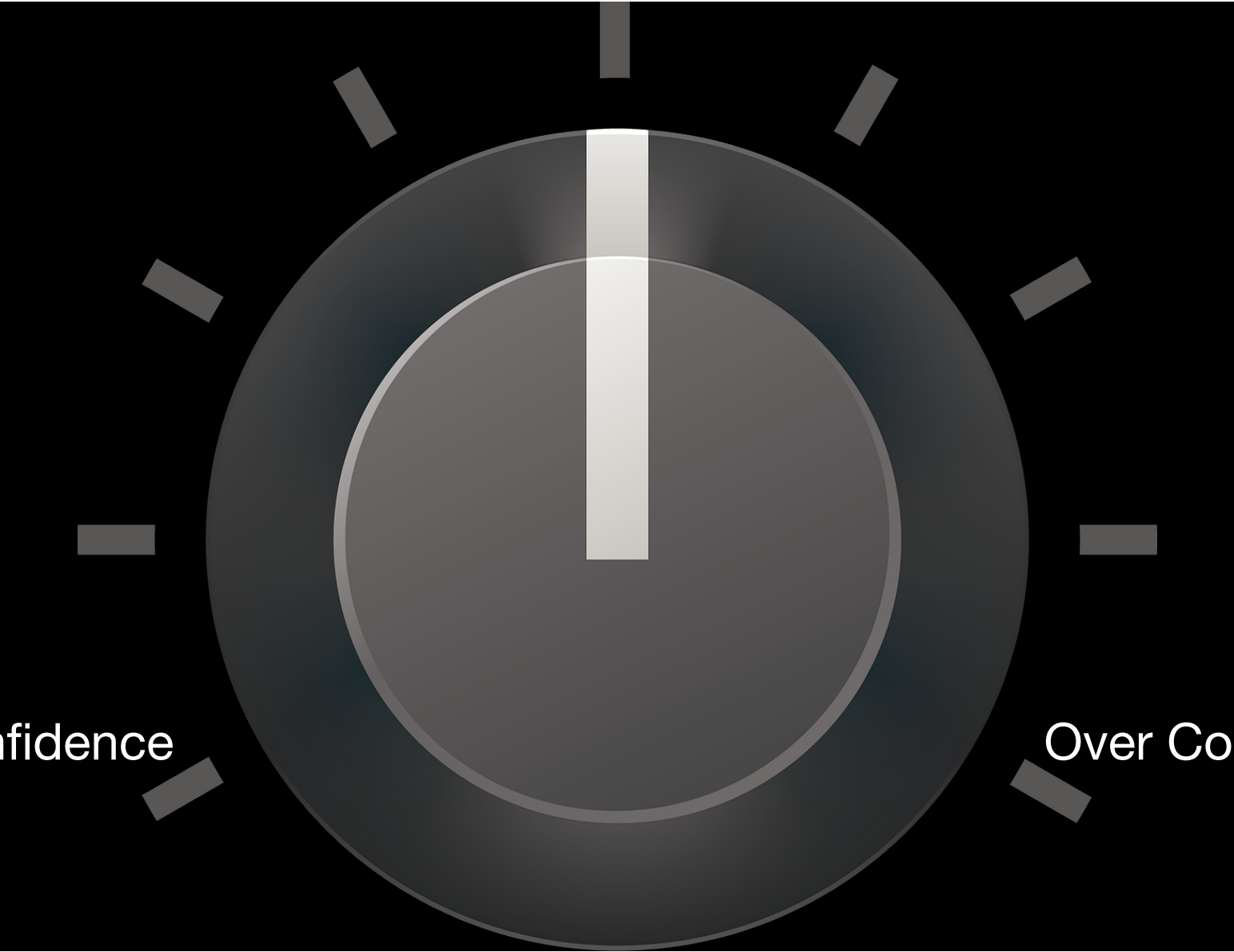
Very Few Things are Binary

Ego

Good or Bad?

Low Confidence





A circular gauge with a dark gray outer ring and a lighter gray inner circle. A vertical white needle points to the top of the gauge. There are eight tick marks around the outer ring: one at the top, one at the bottom, and two on each side. The left side is labeled 'Low Confidence' and the right side is labeled 'Over Confidence'.

Low Confidence

Over Confidence



How are you?

Terrible





Terrible

Great, Awesome
Pretty (darn) Good



Wanna be Pessimistic or Optimistic?



**A Pessimist and an Optimist
walk into a bar**





**How are
things?**

**It can't
get any
worse**



**How are
things?**

**It can't
get any
worse**

Yes, it can!

**How are
things?**



Should You Say Yes or No?

The more Yes you say, the less you will deliver

Novice

Expert

Gain Experience

Need Opportunities

Novice

Expert

Gain Experience
Need Opportunities

Keen to Succeed
Maintain Reputation

Novice

Expert

Say Yes, *eagerly*

Gain Experience

Need Opportunities

Keen to Succeed

Maintain Reputation

Novice

Expert

Say Yes, *eagerly*

Begin to say No, *nicely*

Gain Experience

Keen to Succeed

Need Opportunities

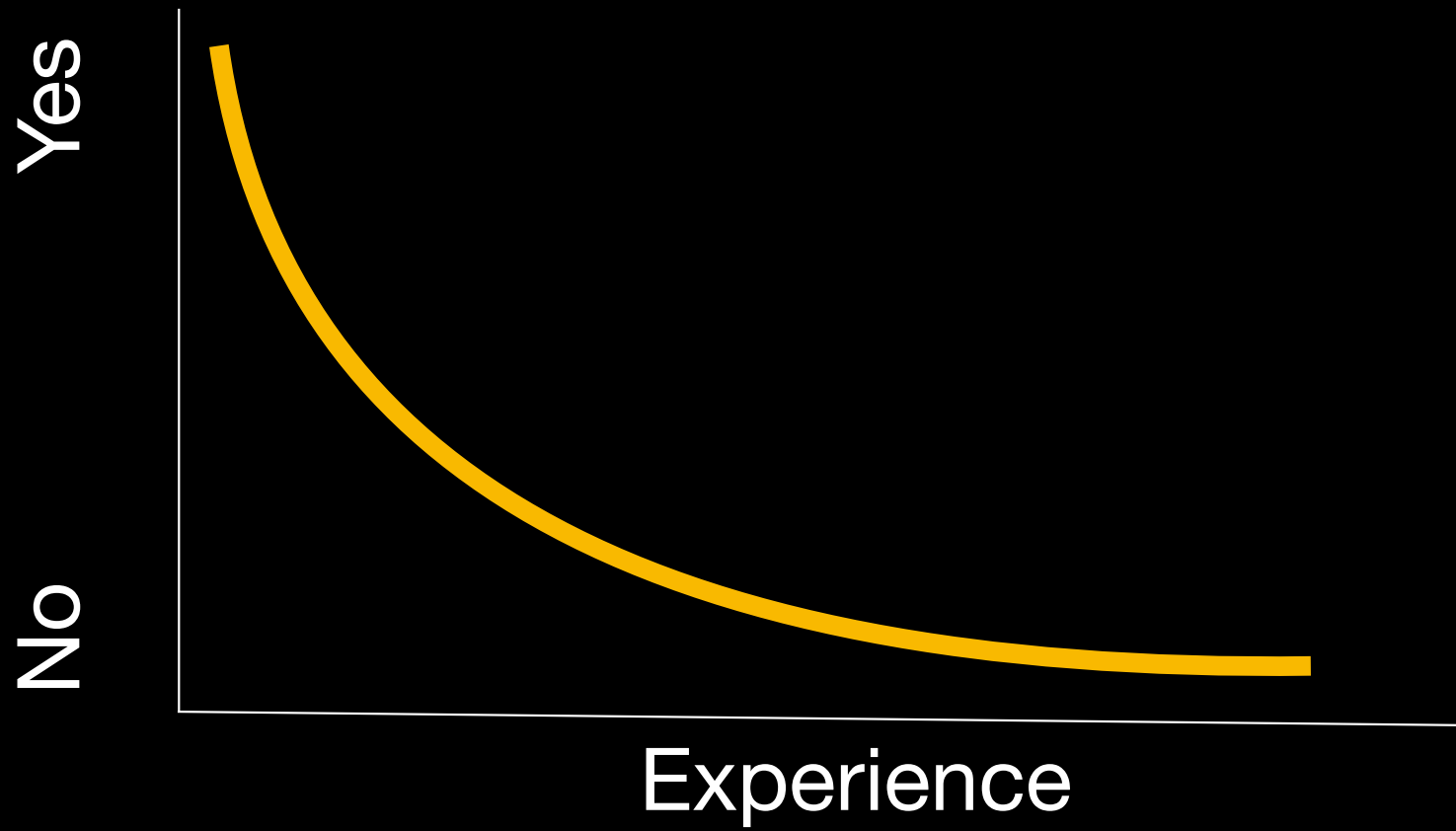
Maintain Reputation

Novice

Expert

No?

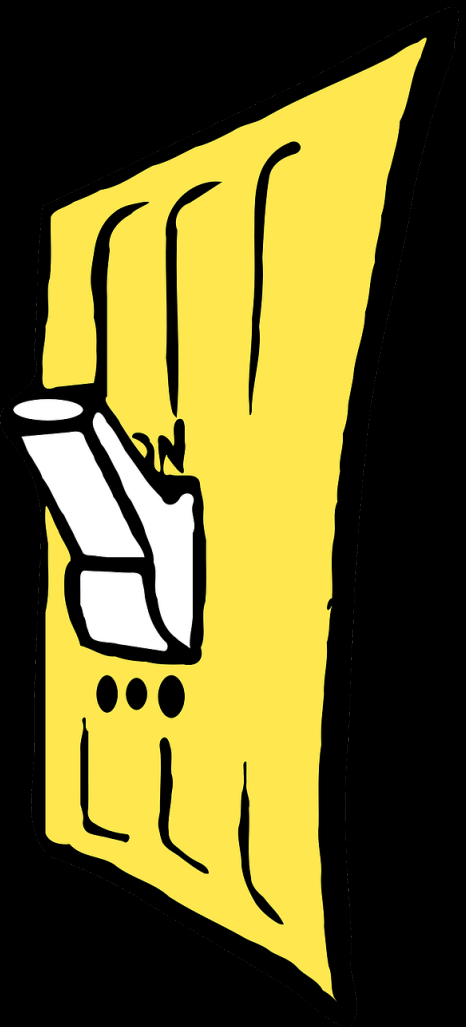
You're saying NO to the task, not to the person

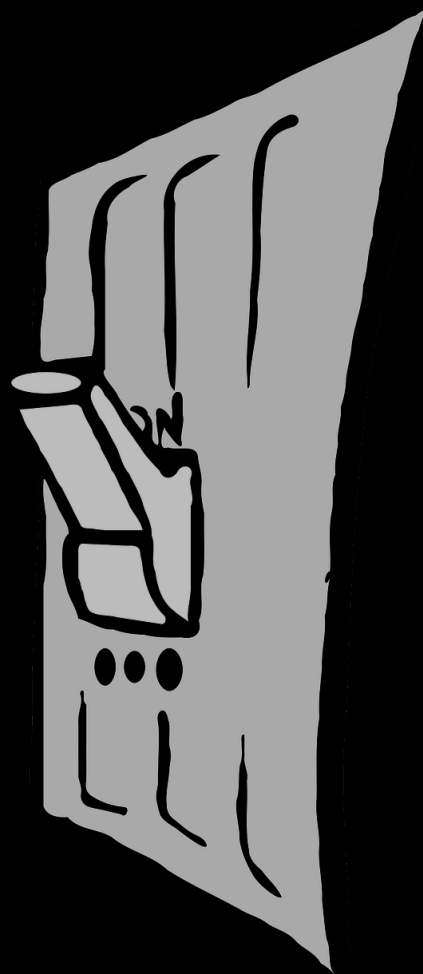




**Living is an act
of evaluating
trade offs**

You can rarely decide things in isolation













Local Optimization



Global Optimization

The Toyota Way

The system is summarized in 14 principles:^[7]

1. "Base your management decisions on a long-term philosophy, even at the expense of short-term financial goals."

https://en.wikipedia.org/wiki/The_Toyota_Way

Survive in the short term

Succeed in the long terms

Be sustainable



**Living is an act
of evaluating
trade offs**



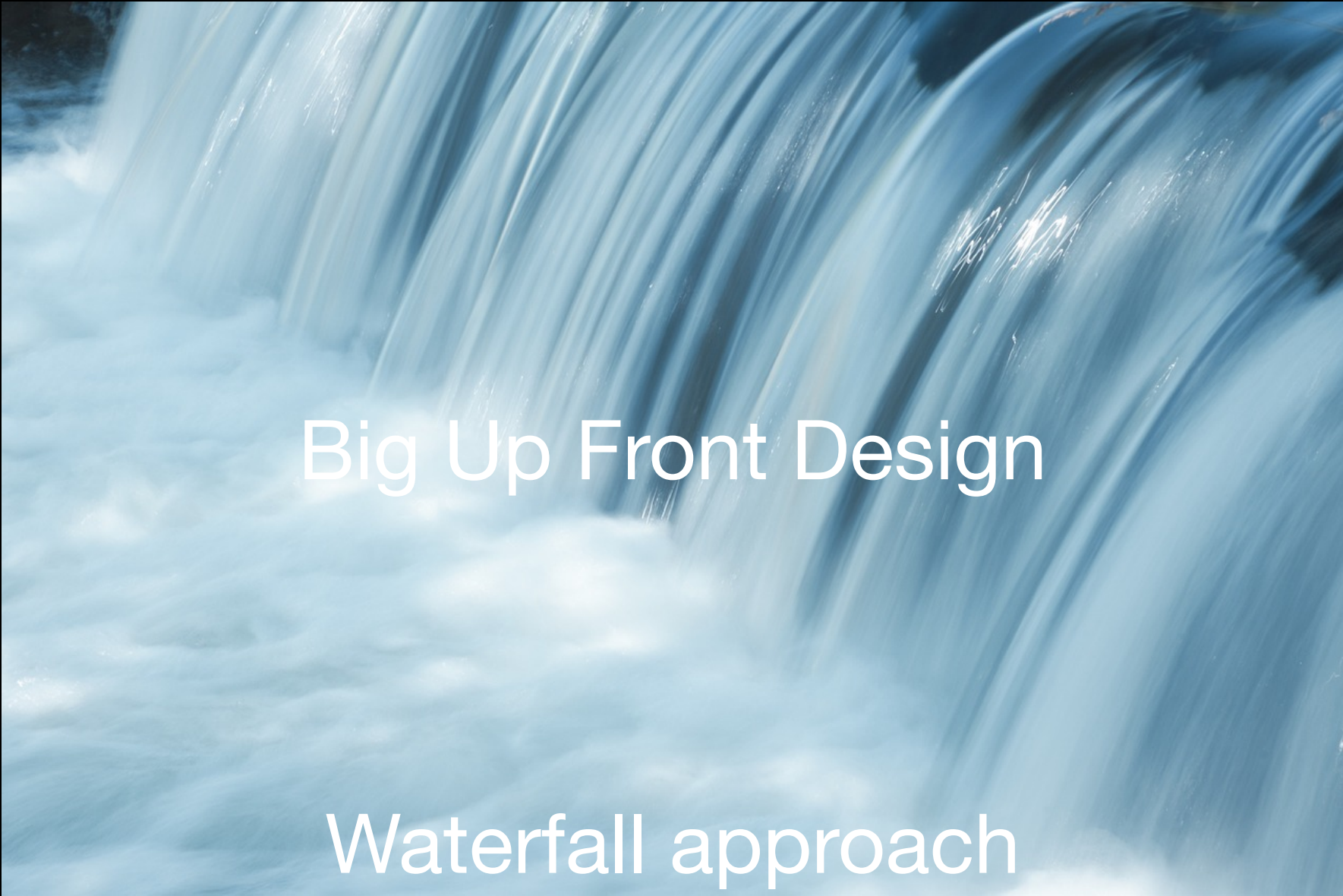
**Living is an act
of evaluating
trade offs**

**Programming is how
we make a living**

The Evolution of Software Development

A close-up, long-exposure photograph of a waterfall. The water is captured in motion, creating a series of vertical, blurred streaks that flow from the top left towards the bottom right. The water has a deep blue color, and the base of the waterfall is filled with white, frothy foam. The overall effect is one of dynamic movement and natural beauty.

Waterfall approach

A long-exposure photograph of a waterfall, creating a smooth, flowing blue and white curtain of water. The water is captured in motion, with streaks of white foam and blue water cascading down. The background is dark, making the water stand out.

Big Up Front Design

Waterfall approach



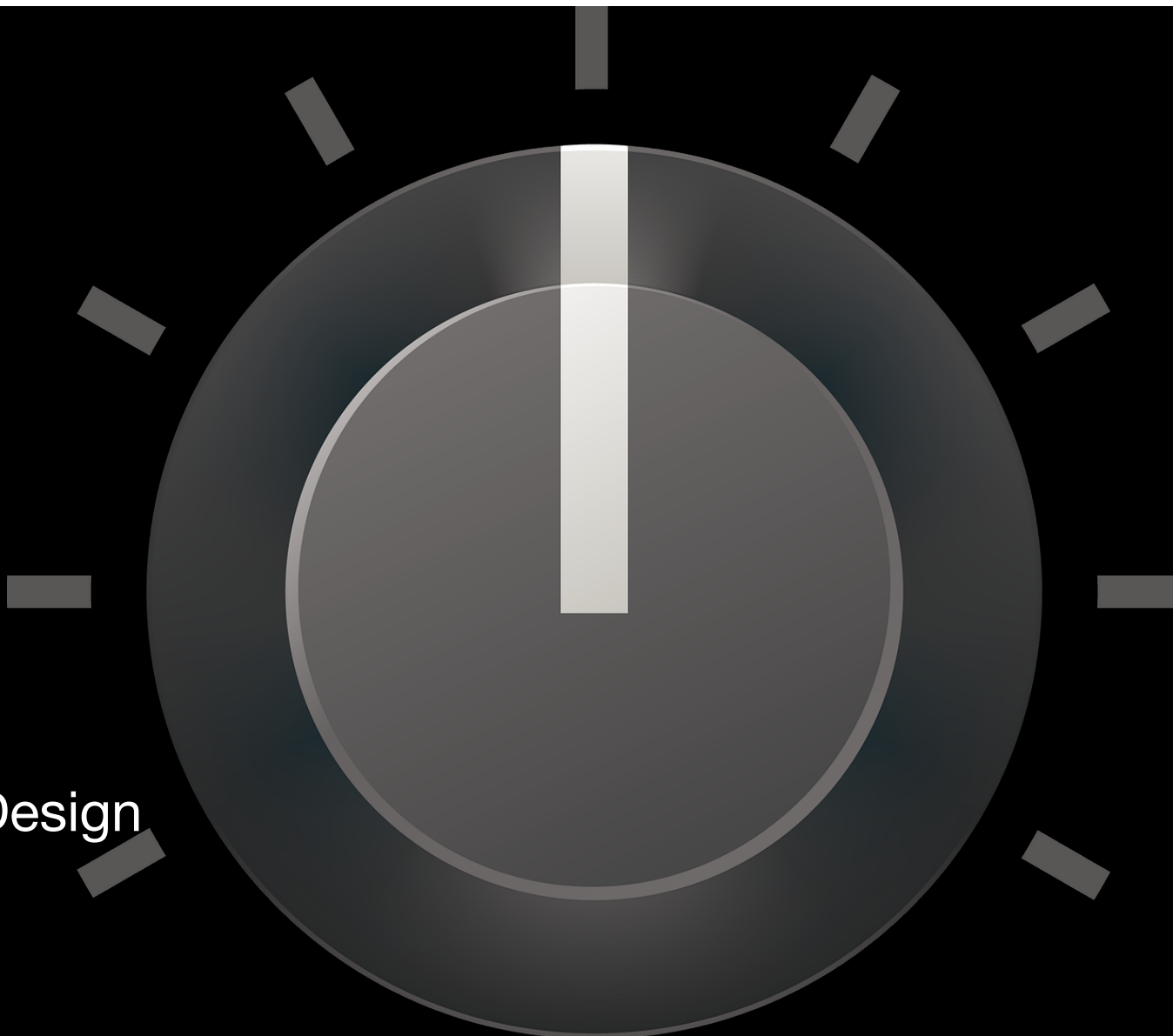
Agile Development

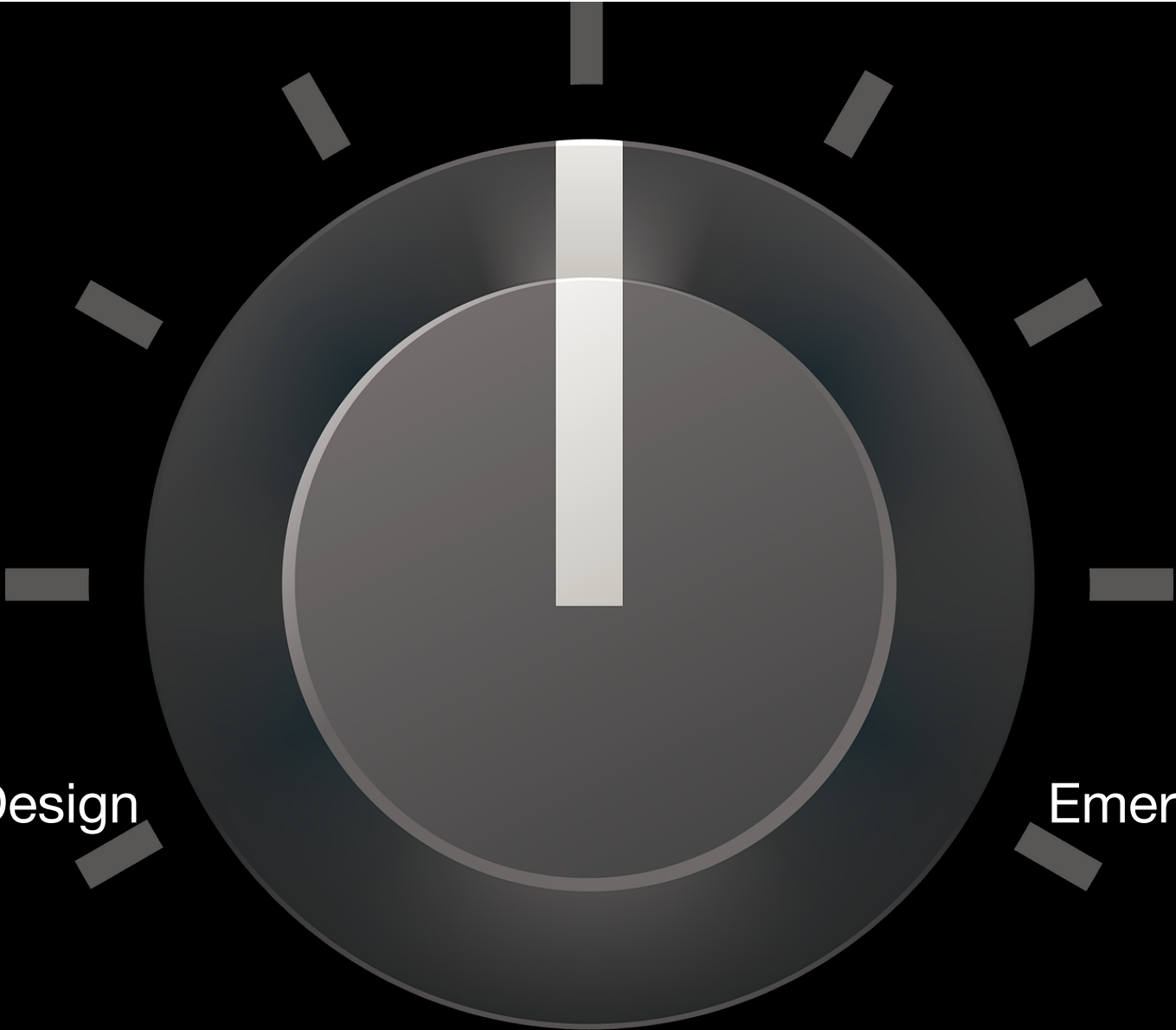
A photograph of a rugby scrum in progress on a green field. Two teams are locked together, pushing against each other. The team on the left wears dark blue jerseys with red accents, while the team on the right wears light blue and white striped jerseys. The players are in a low, crouched position, with their heads down and arms extended, exerting force. The background is a blurred green field.

No Design

Agile Development

Planned Design





Planned Design

Emergent Design

Evolutionary Design
Enough Up Front

Planned Design

Emergent Design



Design by Contract

Design by Capability

Conformance

Compilation checks

Fewer Runtime Failures

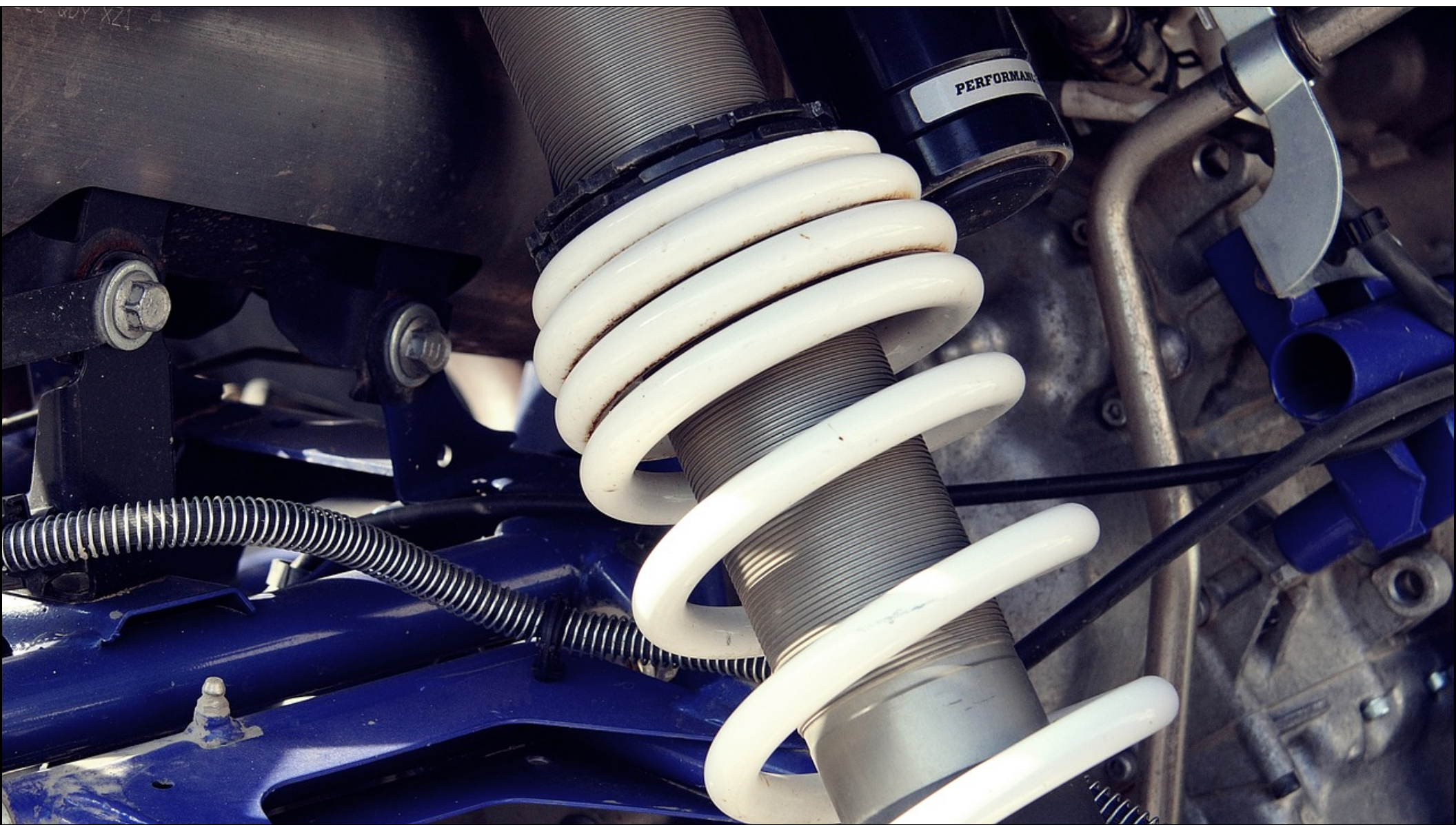
Design by Contract

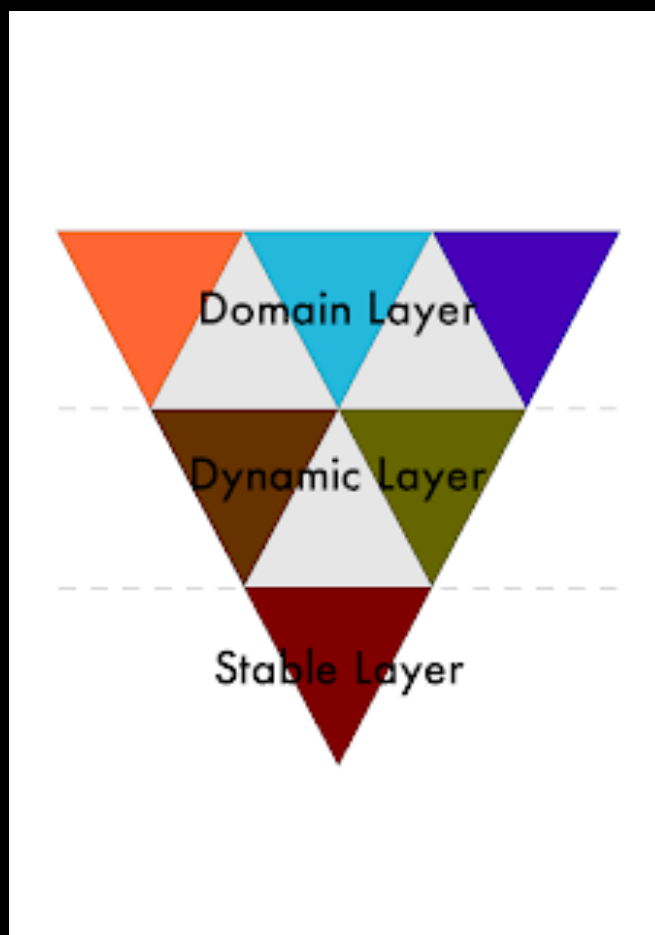
Flexible

Evolvable

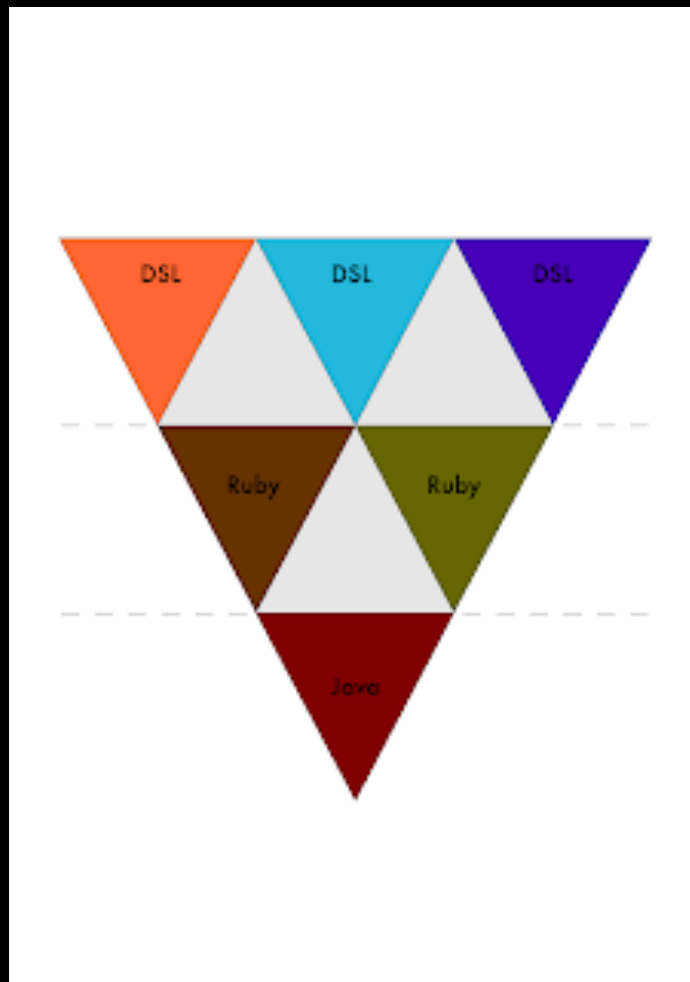
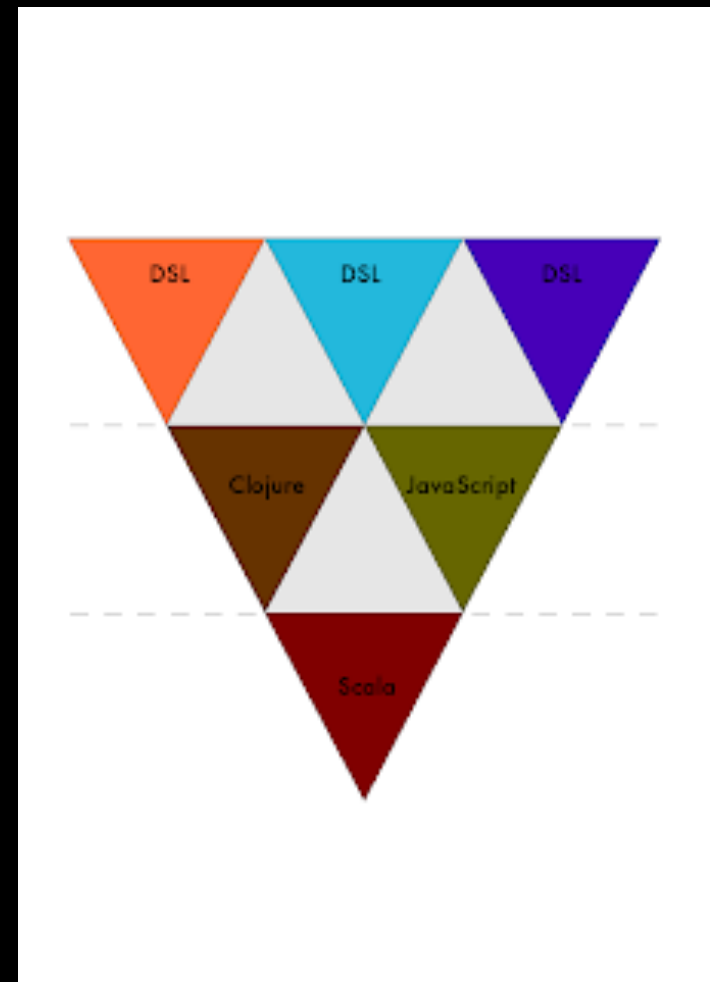
Extensible

Design by Capability





Ola Bini's Language Pyramid



Ola Bini's Language Pyramid

Design Principles

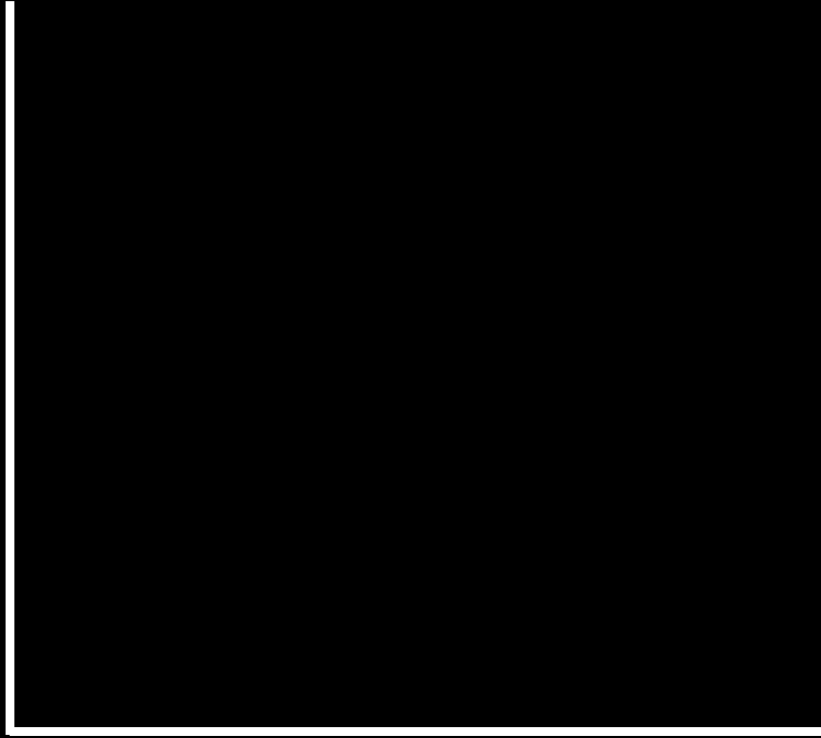
Should we favor Reuse?

Should we increase or decrease coupling?

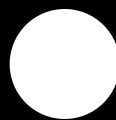
Should we increase or decrease duplication of code?

DRY/Reuse

Coupling



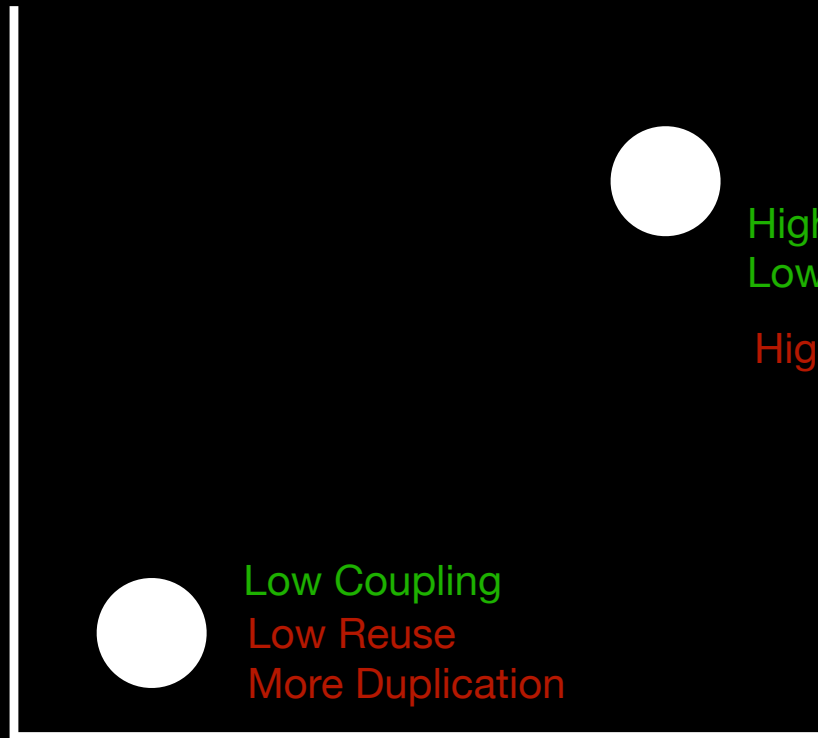
DRY/Reuse



Low Coupling
Low Reuse
More Duplication

Coupling

DRY/Reuse



Low Coupling
Low Reuse
More Duplication

High Reuse
Low Duplication
High Coupling

Coupling

An Insurance Company with a Unified Data Model

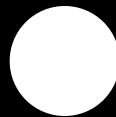
The directive to the developers: Don't Screw Up

A black and white photograph of a young child, likely a toddler, with their face partially obscured by their hands. The child's eyes are closed, and their expression suggests they are crying or feeling distressed. The background is dark, making the child's face the central focus. The text is overlaid on the right side of the image.

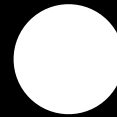
**Code Stability
took precedence
over Business
Agility**

DRY/Reuse

Encapsulation



High Business agility
Low Reuse
More Duplication



High Reuse
Low Duplication
Low Business agility

We're not saying Duplicate with vengeance

Reduce Duplication, but

Prioritize Encapsulation

To provide better *business agility*



**evaluate
trade offs**

Raw Perceived Performance

Easy to Parallelize

Exception Handling

Easy to Maintain

Very familiar

Reduced Accidental Complexity

Imperative Style

Functional Programming



Maintainability

Performance



Scalability

Performance

Don't ask "is the code fast?"

~~Don't ask "is the code fast?"~~

Ask, "is the speed adequate?"



Tweet



Venkat Subramaniam 🇺🇦 ✌️

@venkat_s



The end result of worrying about performance often is the worry and not the performance.



Tweet



Venkat Subramaniam 🇺🇦 ✌️

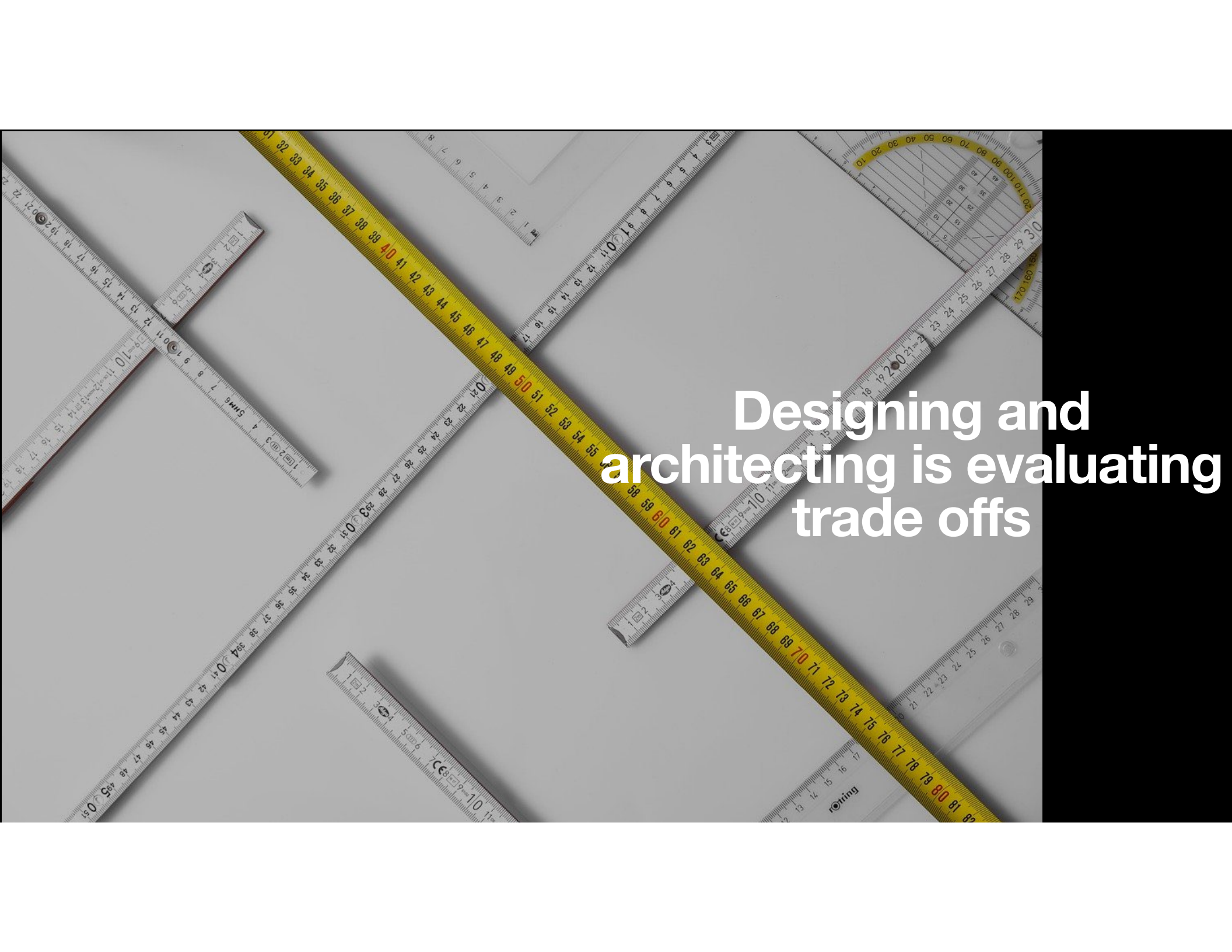
@venkat_s



Those who sacrifice quality to get performance may end up getting neither.



**Where performance is
truly a concern,
write tests for that first**



Designing and
architecting is evaluating
trade offs



**Identify the Essential
Characteristics**

Prioritize what is really important



Evaluate the trade offs

Don't Make binary decisions

Decision making is an Optimization Problem

Decision making is an act of considering compromises

The most important things to document?



Why?

The most important things to document?

The most sensible answer we can give?

It depends



The most sensible answer we can give?

Decision Dials

Thank you

Venkat Subramaniam

venkats@agiledeveloper.com

@venkat_s