



# Separation Anxiety

Designing for Change

Jon McBee  
CLA, CLED, CTD, CPI, LabVIEW Champion, Scrum Master





What concept do you value most within the context of software development?



# Problem Decomposition

# Separation Anxiety



## Reflecting on GDevCon #2

By *James McNally* on Aug 22, 2019 in *Events*

GDevCon #2 is wrapped. I'm really proud of the event we put on this year and its got me really excited about where we can take it next year. We've got plenty to think about and I hope that we can top it again.

Our target from the start has been that this has to be valuable to all members of a team. The CLA summits are often a place to pontificate about architecture and design but that is of no interest to someone getting going with LabVIEW or working on an established framework but with engineering challenges and deadlines to meet.

Our target from the start has been that this has to be valuable to all members of a team. The CLA summits are often a place to pontificate about architecture and design but that is of no interest to someone getting going with LabVIEW or working on an established framework but with engineering challenges and deadlines to meet.

This year I feel we got this much better. I believe that showed in the variety of levels, engaging presentations and useful takeaways.



# Working code is not enough

- Robert C. Martin (Uncle Bob)

# Working code is not enough

## Tactical Programming

- Move fast and break things
- Get something working (often before a deadline)
- Hero from managements point of view

~~• Technical Debt~~



# Tactical Debt

## Strategic Programming

- Problem decomposition
- Separation of concerns
- Emphasis on continuous investment in the infrastructure of the system

~~• Technical Debt~~

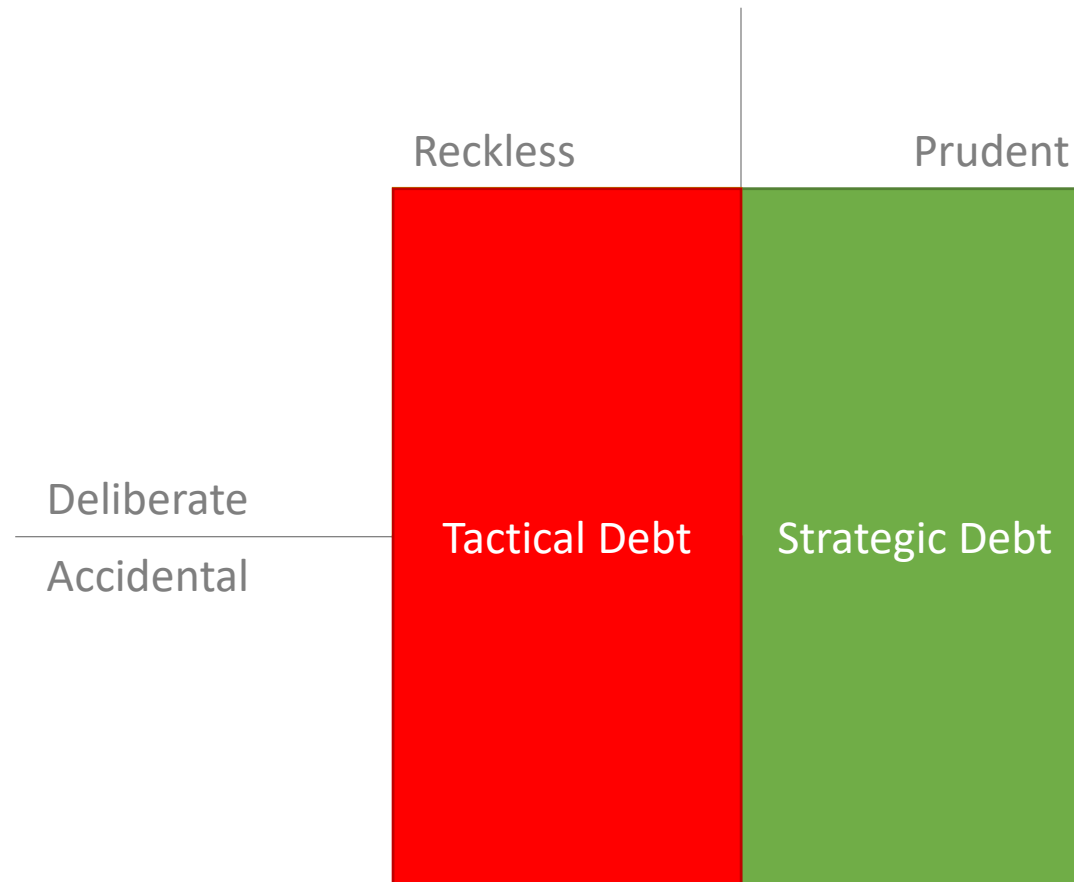


# Strategic Debt

# Technical Debt Quadrants



# Technical Debt Quadrants

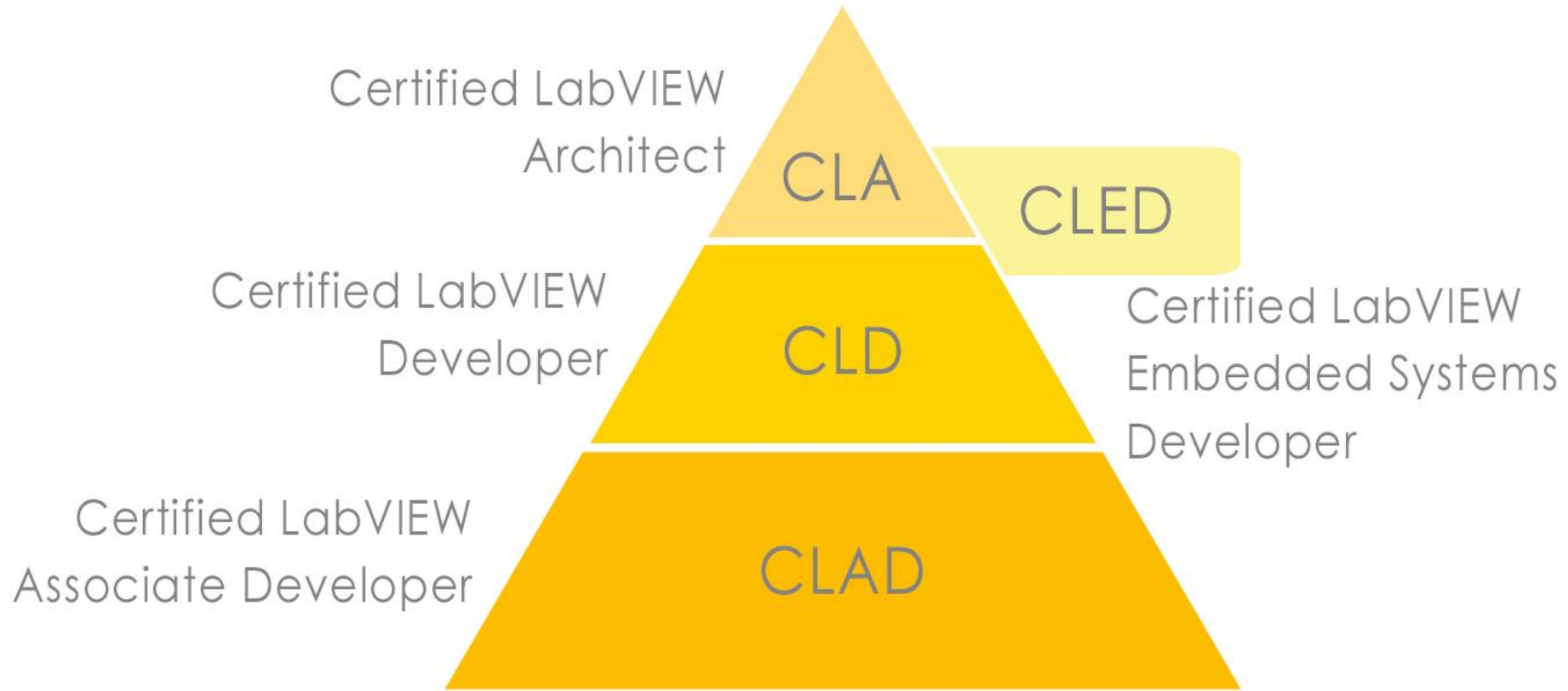






Any tactical tornados in the room?

# Tactical



# PRIN-CI-PLE <sup>/'prɪnsəpəl/</sup>

A rule or belief governing one's personal behavior.

1. It is more important to reduce the effort of maintenance than it is to reduce the effort of implementation
2. The effort of maintenance is proportional to the complexity of the system



# Separation of Concerns

“Separation of concerns... even if not perfectly possible is yet the only available technique for effective ordering of one’s thoughts”

- Edsger Dijkstra

# Separation of Concerns



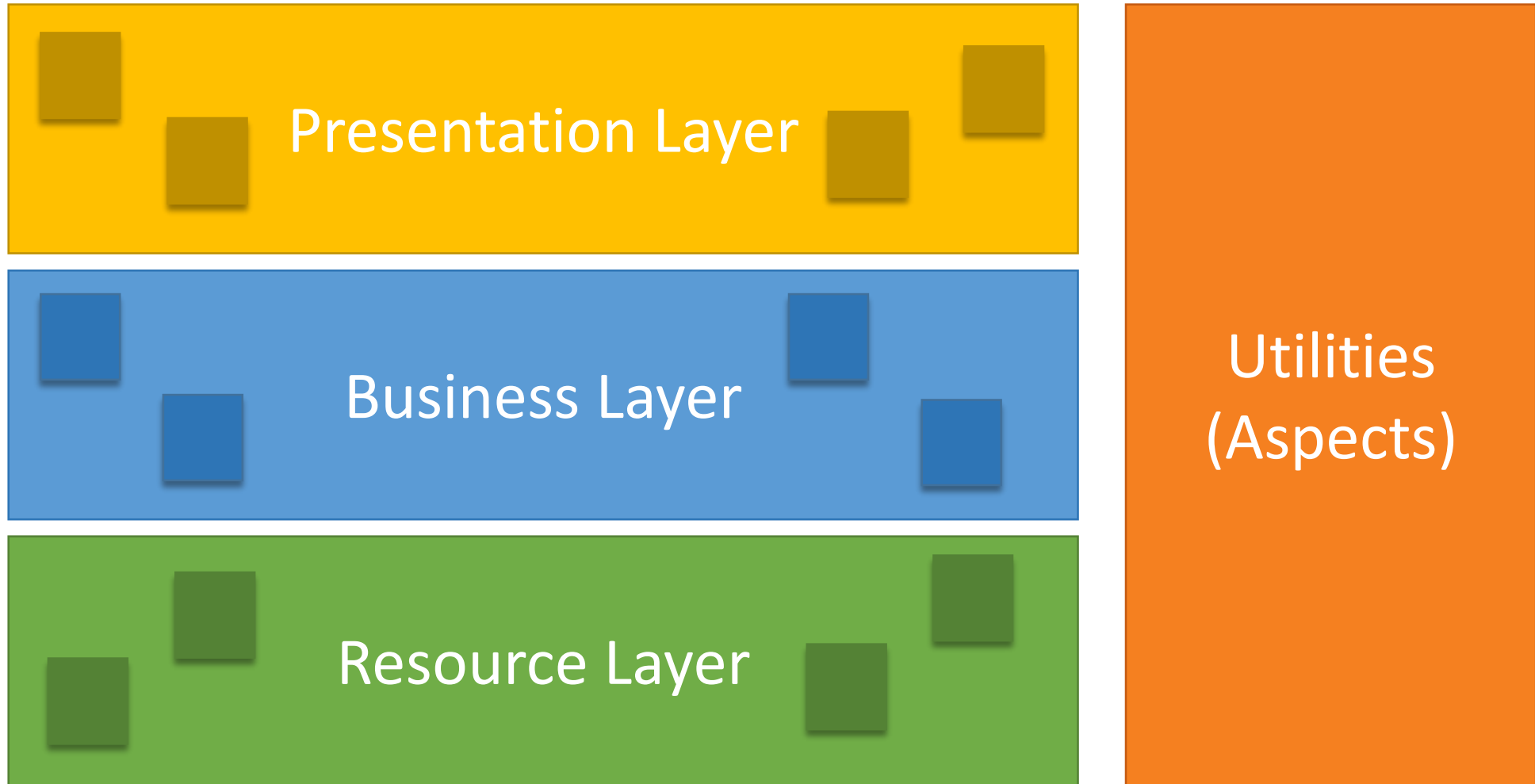
- The goal is not to reduce a system into its indivisible parts, but to organize the system into elements of non-repeating sets of cohesive responsibilities
- The overall goal of Separation of Concerns is to establish a well-organized system where each part fulfills a meaningful and intuitive role while maximizing its ability to adapt to change



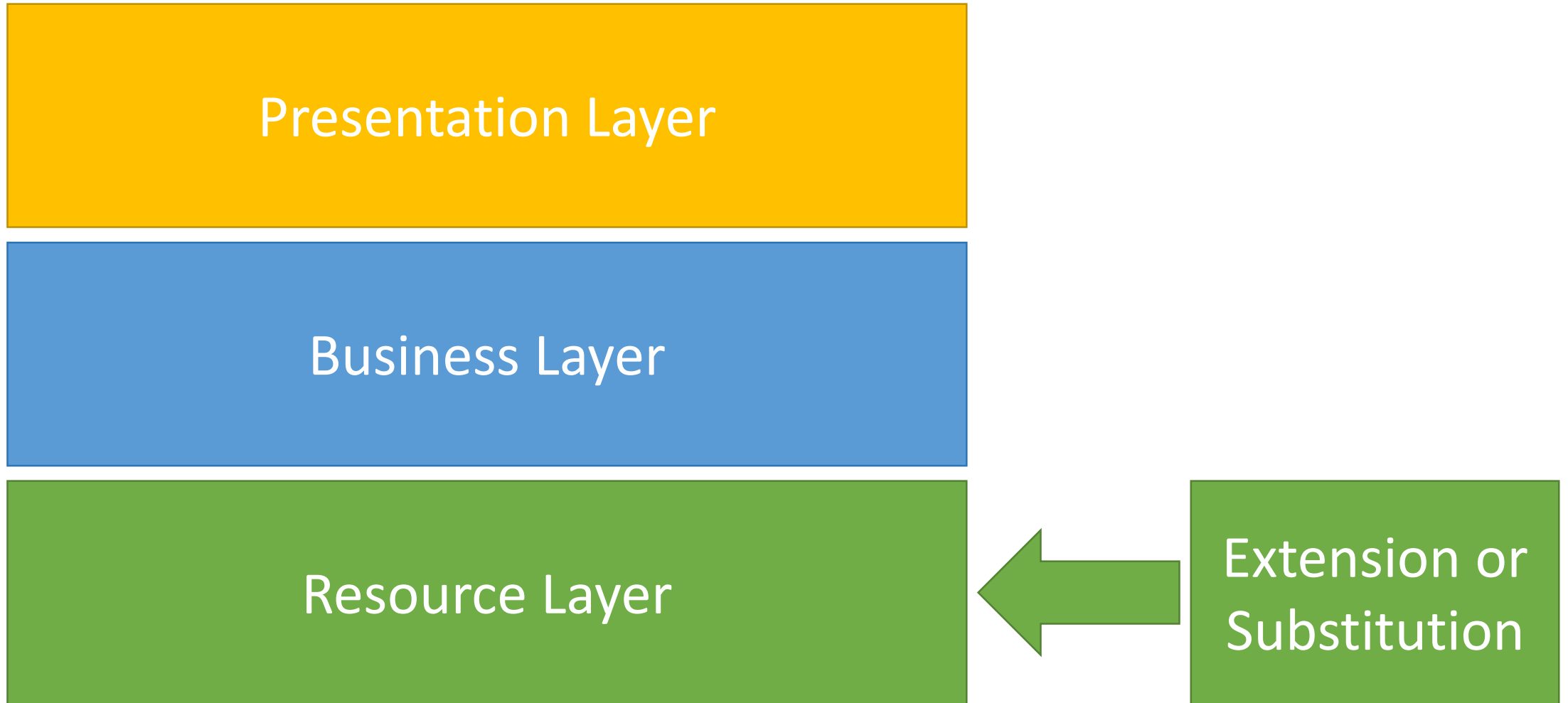
# Value of Separation of Concerns

1. Lack of duplication and singularity of purpose of components makes the system easier to maintain
2. The system as a whole becomes more stable as a byproduct of increased maintainability
3. Strategies required to ensure each component only concerns itself with a single set of cohesive responsibilities result in natural extension points
4. The decoupling that results from the cohesiveness of components makes components reusable

# Separation of Concerns Strategy

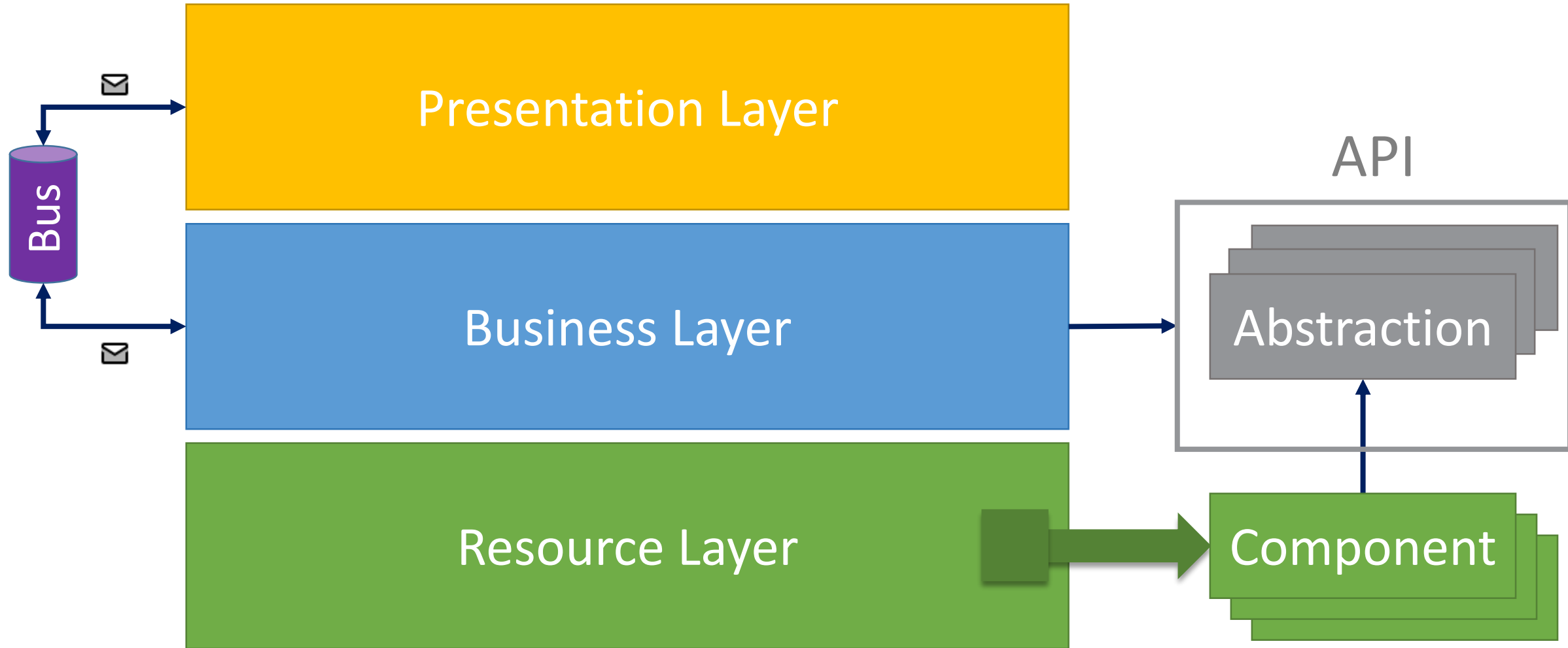


# Extending Concerns



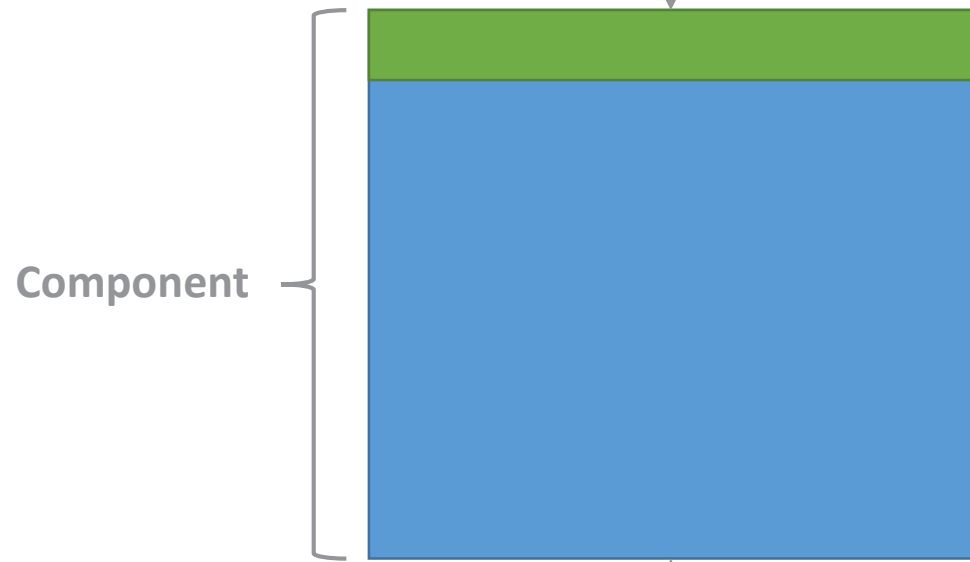


# Dependencies Between Concerns

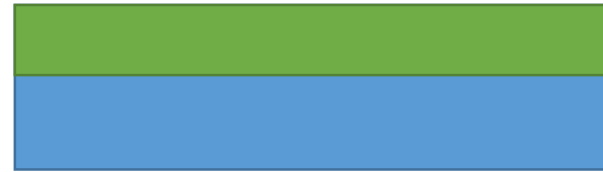


# API Design

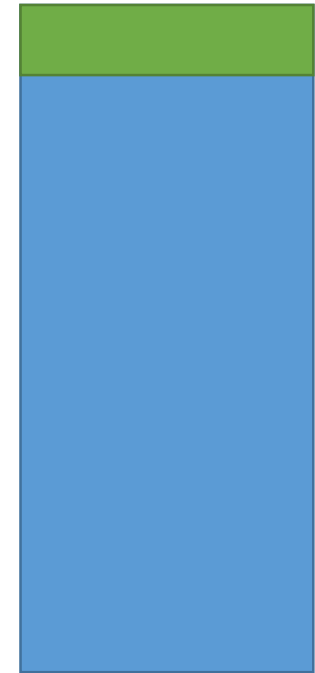
**Interface:** everything that must be known to users (*Cost*)



**Functionality:** fulfills promises of interface but hidden from users (*Benefit*)

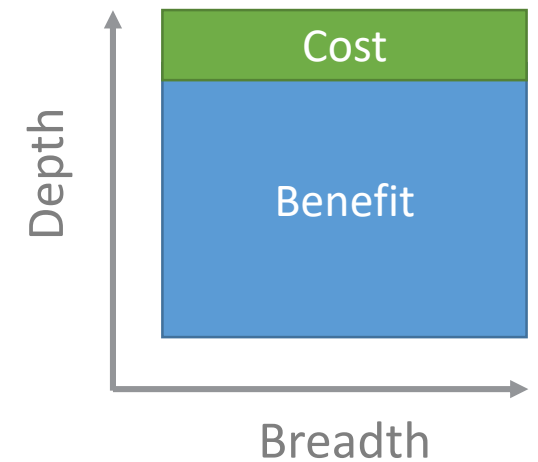
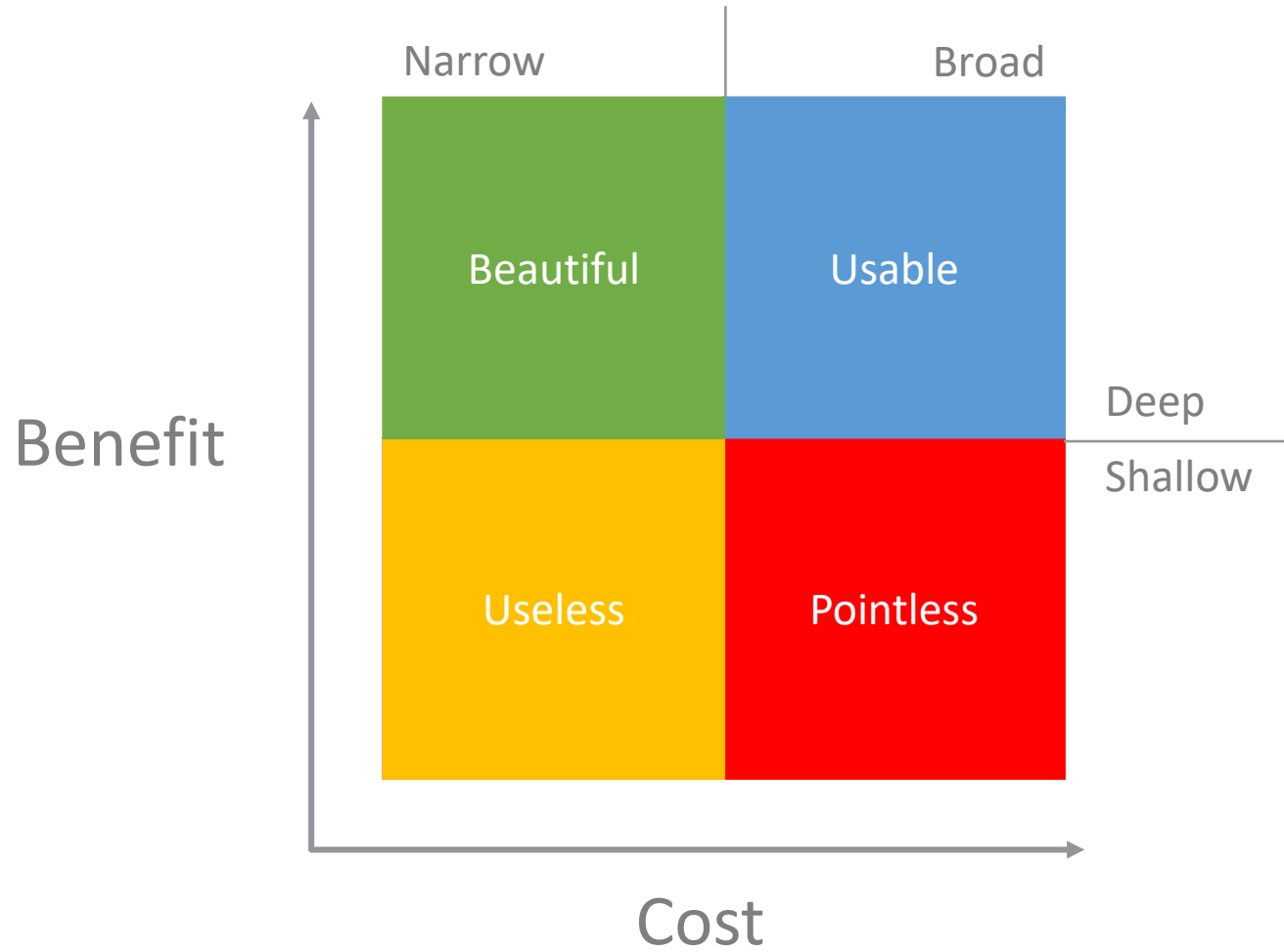


Shallow Component



Deep Component

# API Design



# Create Awesome LabVIEW

Don't waste time re-inventing code that's already been written. Leverage the power of our community maintained packages to deliver bug free code faster.



With a registry containing 70 packages, you're sure to find some help.

## Checkout These Cool Packages

Popular

New

### @cs/lookup-table

A malleable, by-reference, lookup table API with in-placeness optimi...

### @cs/event-logger

Composed Event Logger is an extendable, ready to use LabVIEW eve...

### @cs/ci

composed-ci that takes the LabVIEW-specific configuration complexi...

### @cs/linked-list

A malleable, by reference, linked list

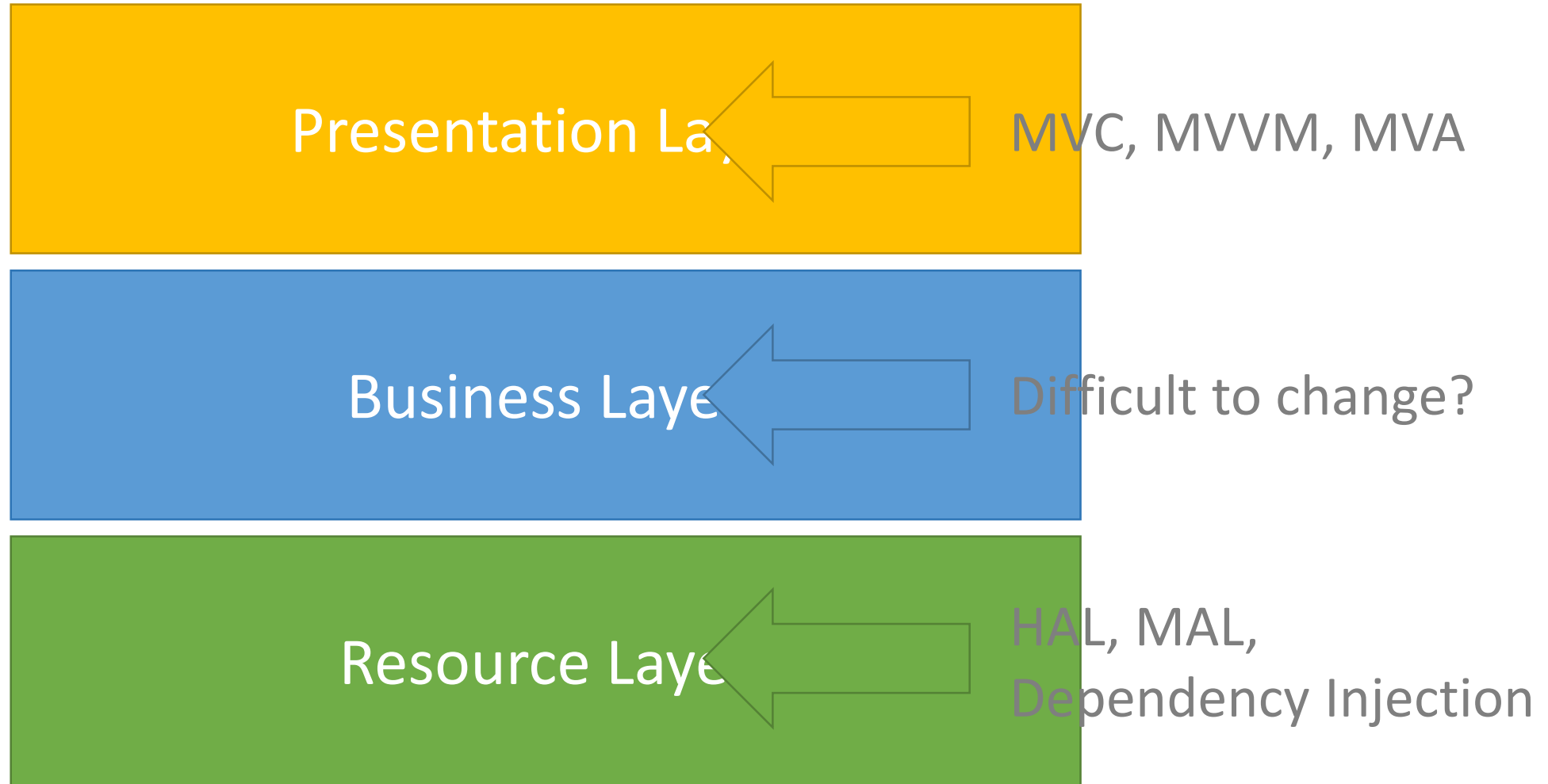
### @cs/variant

Extensions to LabVIEW variant functionality for everyday use


## Why GPM rules



# Plug GPM

# Separation of Concerns Strategy




  
Hi Jon,

It is 3 am...I can't sleep- due to stress about the status of our instrument. Lack of software support has been a primary reason for where we are today. Possibly you don't recognize how much danger our company is currently in. If we don't get our instrument running the way it should ASAP, there is a very good chance we will fail to raise additional capital and myself and the other 16  employees will not have a job by Christmas.

Whatever else you are working on right now...I'm pleading with you to reprioritize and make us your #1 concern right now. This doesn't mean that next week is okay. Let me repeat, next week is NOT okay. I need you working on  problems today, tomorrow, etc until software is no longer our biggest issue. I wouldn't be asking if I didn't fully believe  life wasn't at stake.

Please please please tell your other customers you are unable to support them this week due to a dire emergency. Please confirm you got this email and are working on our problems as soon as you've had your cup of coffee.

Thank you  


Sent from my 

# Let me tell you a story...



# An Example

<https://gpackage.io/packages/@cs/carwash>

<https://bitbucket.org/composedsystems/composed-car-wash>

10<sup>th</sup> ANNIVERSARY EDITION

REVISED AND UPDATED

# Talent Is Overrated

*What Really Separates World-Class Performers from Everybody Else*



NATIONAL  
BESTSELLER

GEOFF COLVIN

Senior Editor at Large, FORTUNE

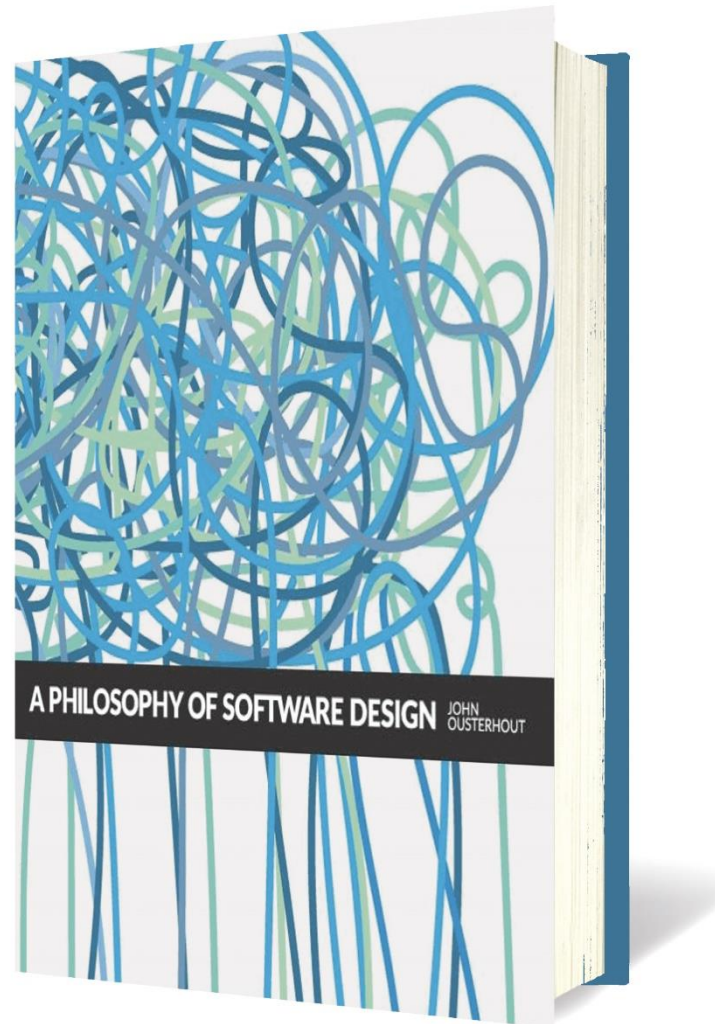
## Am I Being Reasonable?

- Is it fair to expect everyone to write clean code?
- Is this technology stack too hard to learn?
- Is it possible to apply these ideas if you have an established code base?
- Other questions you may have?



# A Philosophy of Software Design

---

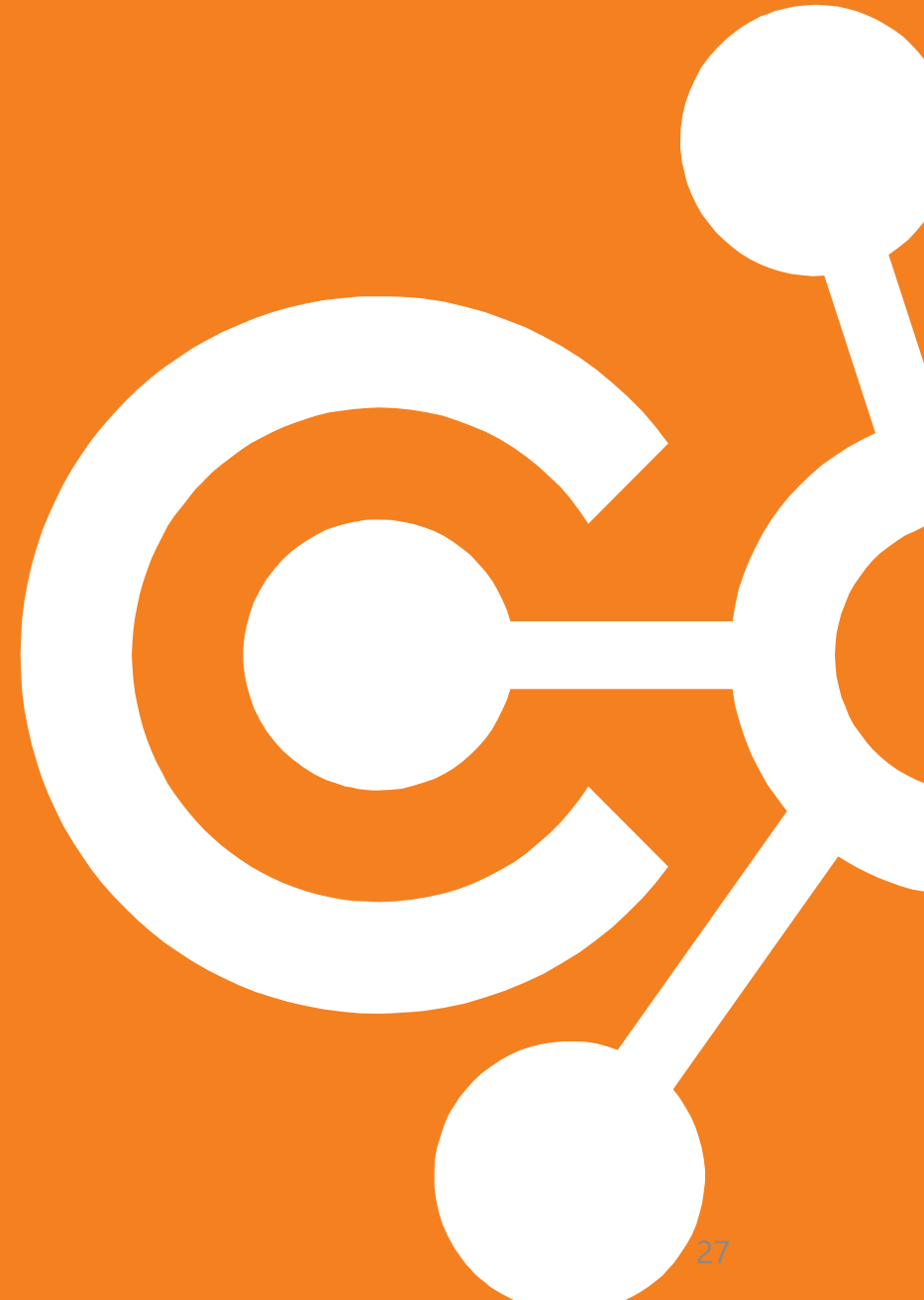




# Separation Anxiety

- Applying the principle of separation of concerns often involves advanced concepts and constructs which bring a certain level of complexity to the application beyond that of merely addressing the domain concerns of the application
- These techniques often lead inexperienced or more tactical-minded developers to characterize such designs as “overly-complex” or “over-engineered”
- Development teams should certainly not seek complexity for complexity’s sake, but the notion that avoiding advanced design concepts equates to avoiding complexity should be dispelled

# QUESTION & ANSWER





# Thank You

Stay In Touch With Us .....

67 West Street, Suite 401

Brooklyn, Ny 1122

000 000 0000

[info@composedsystems.com](mailto:info@composedsystems.com)

[Www.composedsystems.com](http://www.composedsystems.com)

