

# "AGILE" ESTIMATION

Is it possible to be good at  
estimating large projects?

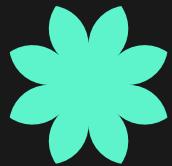
YES

andy.miller@gravie.com

onetribeyoyo@gmail.com

github.com/onetribeyoyo/osn-2024\_agile-estimation





gravie

# WHY DO WE DO THIS?

# WHY DO WE DO THIS?

- 'cause you love to estimate?

# WHY DO WE DO THIS?

- 'cause you love to estimate?
- 'cause you love arguing about how long something is gonna take? (or cost?)

# WHY DO WE DO THIS?

- 'cause you love to estimate?
- 'cause you love arguing about how long something is gonna take? (or cost?)
- 'cause ya really like sleepless nights tossing and turning about whether or not it's "right"?

# WE ESTIMATE...

- because it's important that our sponsors know what they're in for,
- to reduce surprises during the development effort,
- and to give us time to focus on the fun stuff.



# HOW?

Let's start by getting good with "smaller" efforts...

Keeping this very simple...  
...and estimate by counting.

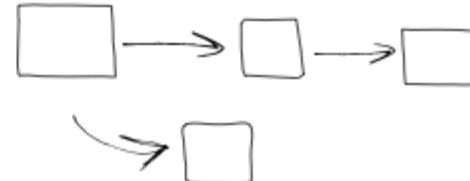
# 4 things to count

Who?

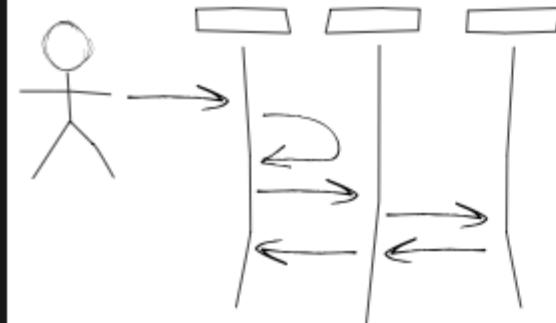
Needs to do what?



What are the elements of the system?



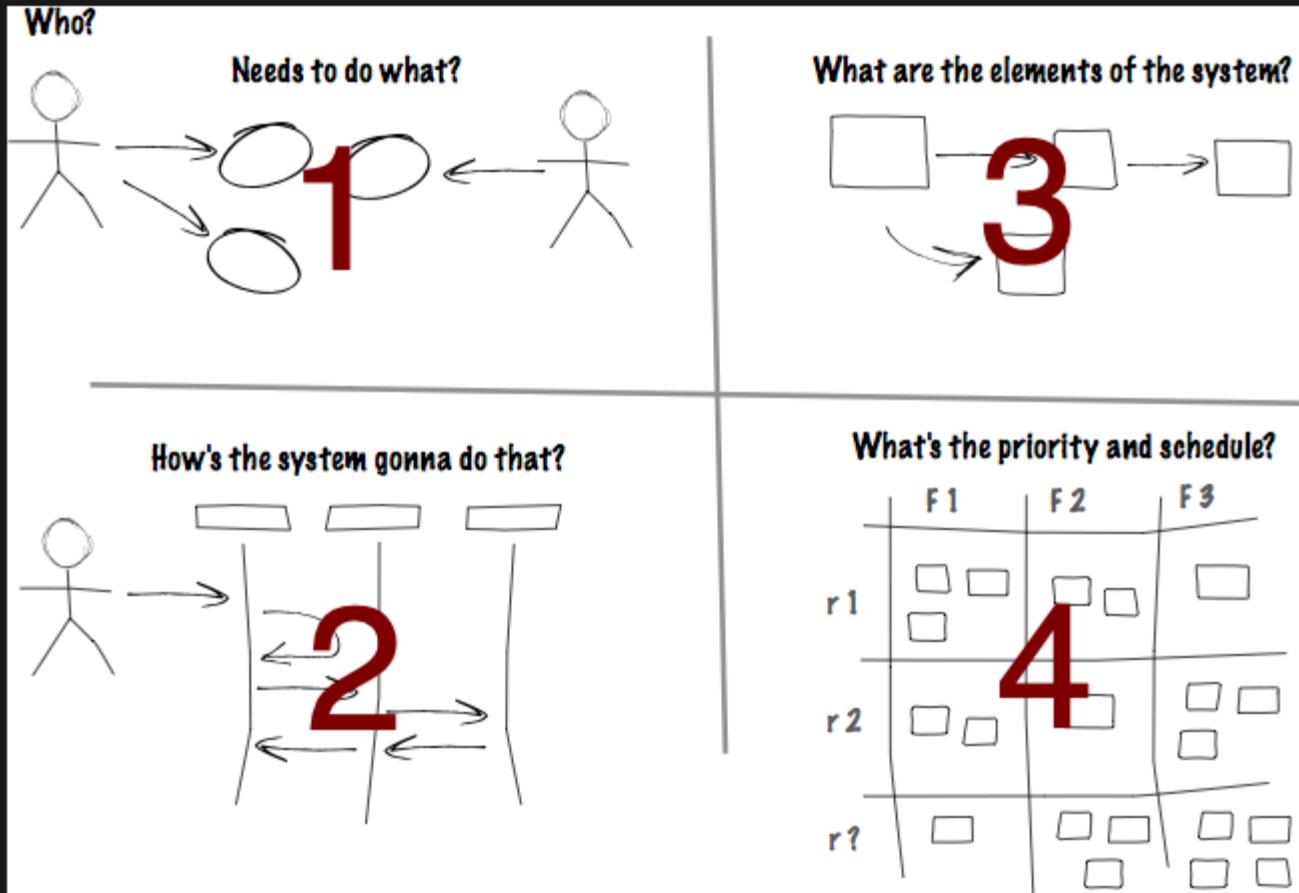
How's the system gonna do that?

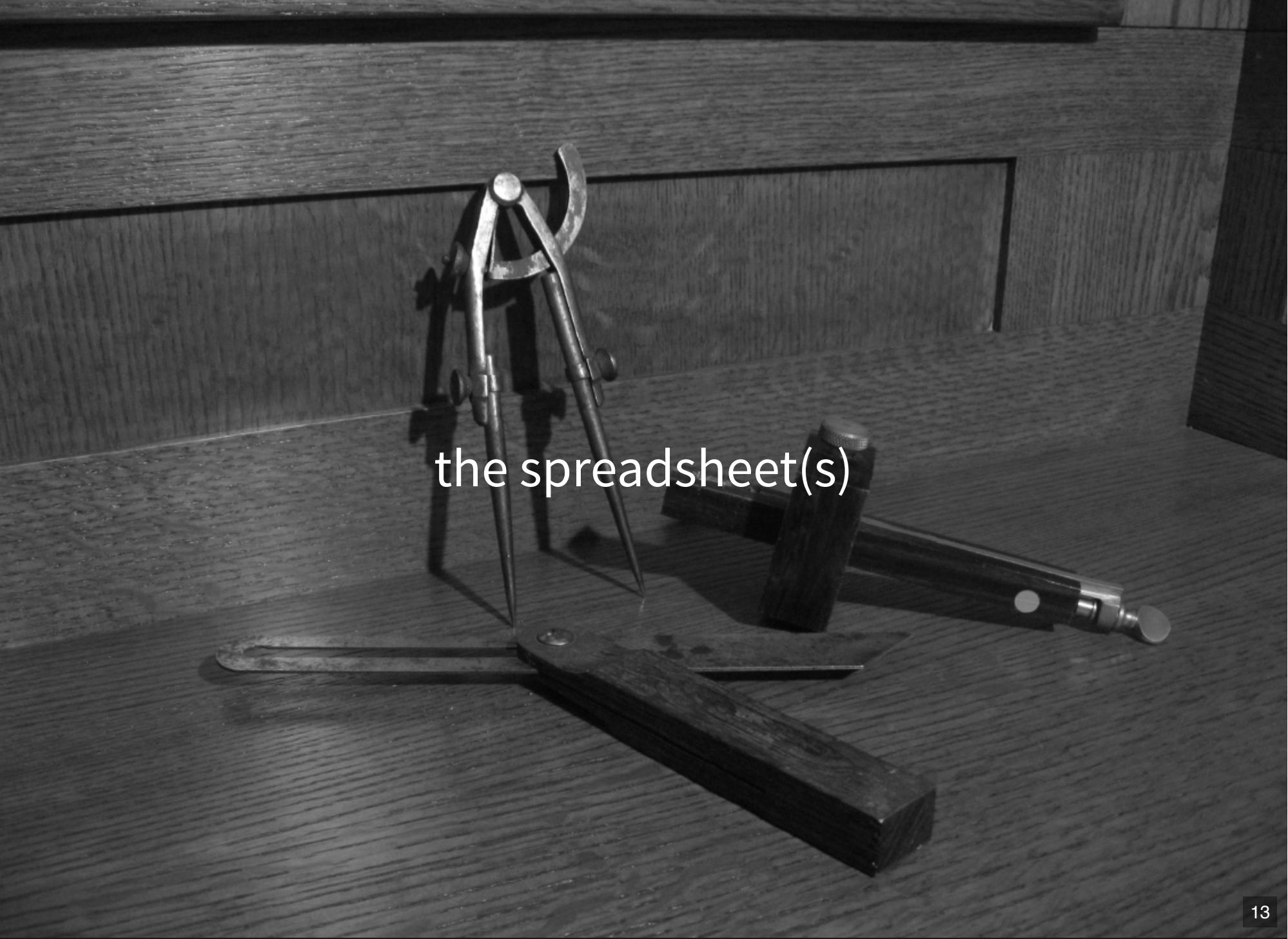


What's the priority and schedule?

	F1	F2	F3
r1	□ □	□ □	□
r2	□ □	□	□ □
r?	□	□ □ □	□ □ □

# 4 things to count





the spreadsheet(s)

Easy? Right?

# HOW THIS STARTED FOR ME

- Spurgat's challenge

# HOW THIS STARTED FOR ME

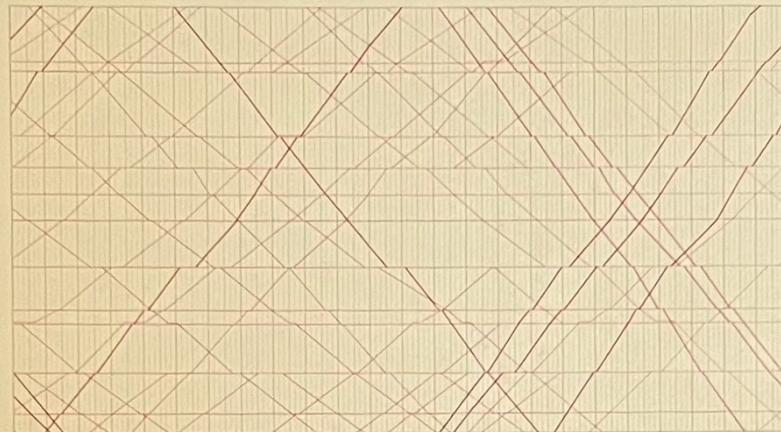
- Spurgat's challenge
- Tufte's advice

# HOW THIS STARTED FOR ME

- Spurgat's challenge
- Tufte's advice
- Karner's research

# HOW THIS STARTED FOR ME

- Spurgat's challenge
- Tufte's advice
- Karner's research
- Greenspun's innovation



SECOND EDITION

# The Visual Display of Quantitative Information

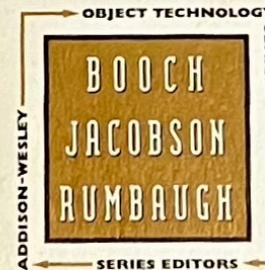
EDWARD R. TUFTE

# APPLYING USE CASES

## A PRACTICAL GUIDE

GERI SCHNEIDER  
JASON P. WINTERS

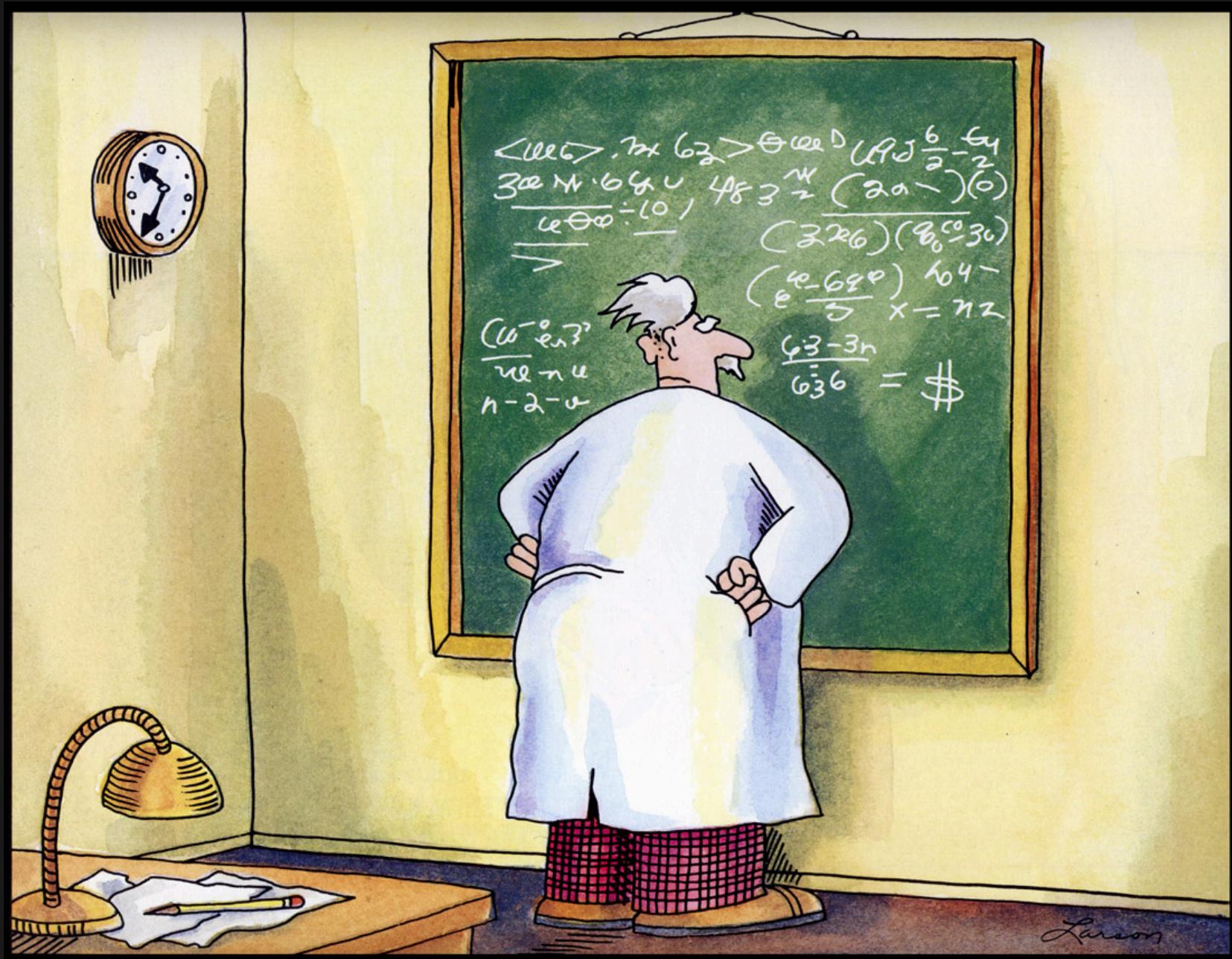
Foreword by Ivar Jacobson



check out those complexity factors in the spreadsheet!



**Philip and Alex's Guide to Web Publishing**





# WHAT DO I MEAN BY "AGILE"?

# WHAT DO I MEAN BY "AGILE"?

- effort vs time

# WHAT DO I MEAN BY "AGILE"?

- effort vs time
- expect the problem to change

# WHAT DO I MEAN BY "AGILE"?

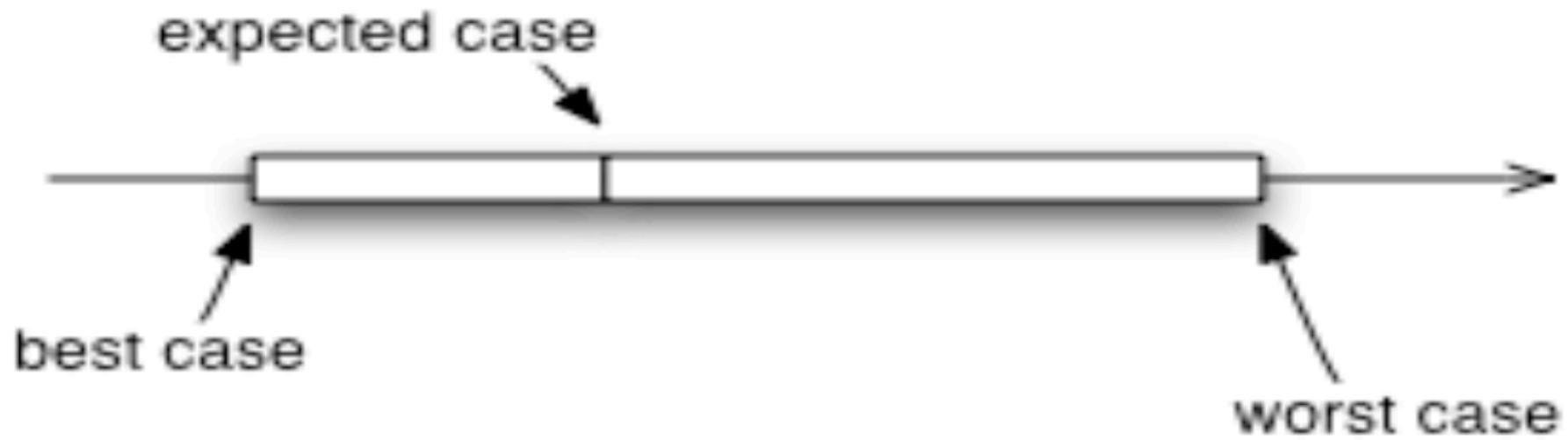
- effort vs time
- expect the problem to change
- expect the solution to change

How do we deal with all this change?



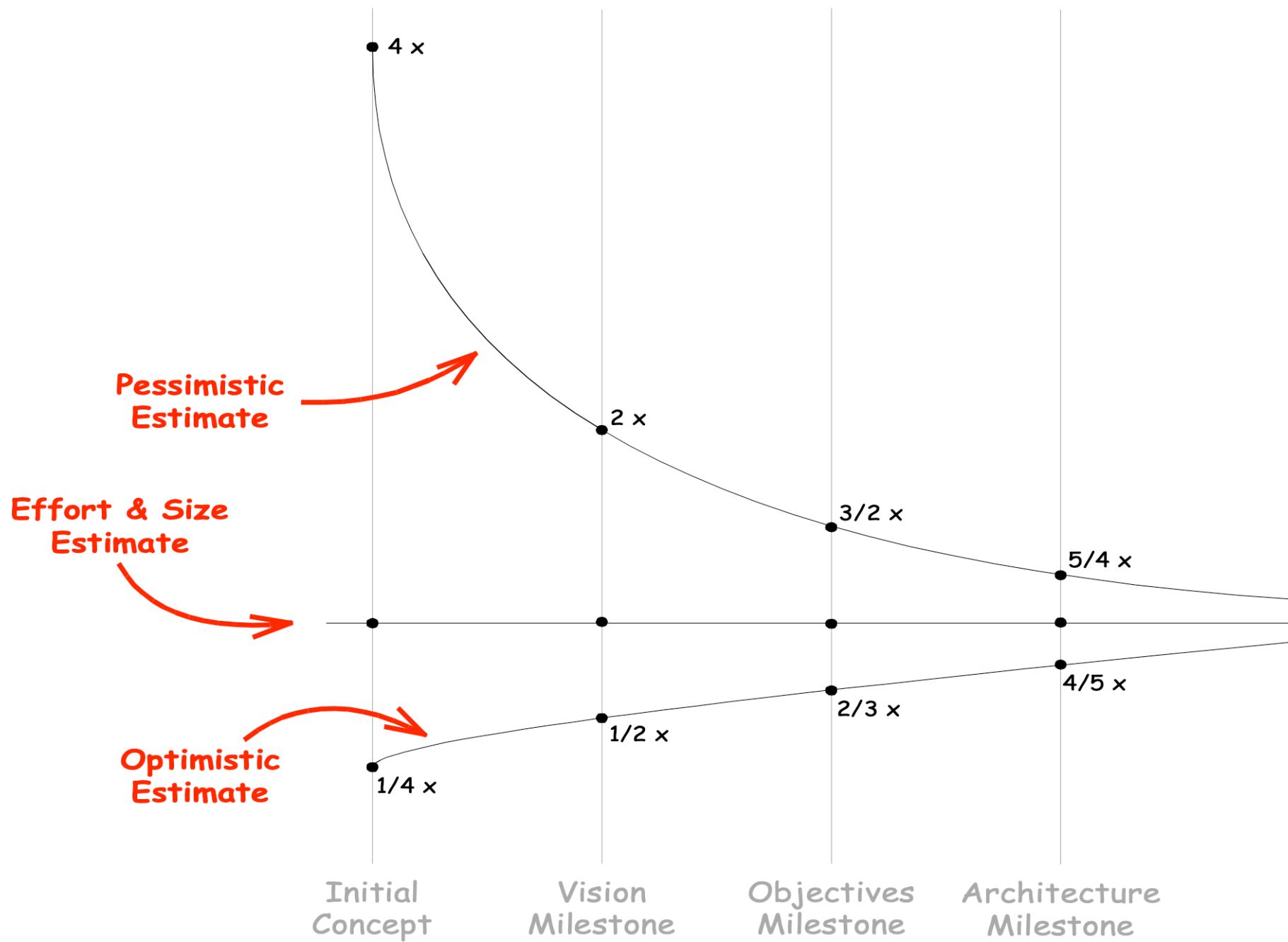
~~But consider~~

AND consider...



Things that push towards worst-case...  
...better be accounted for in the estimate







So...

...Now we've got an estimate

...How do we make this stick?  
(especially for large projects)

Learn how to build trust

( Are you being asked for an estimate  
or for a commitment? )

Learn how to "Influence"  
an understanding of the differences between

- An "Estimate"
- A "Plan"
- A "Schedule"
- A "Commitment"



COLLINS BUSINESS ESSENTIALS

ROBERT B. CIALDINI, Ph.D.

The Psychology  
**INFLUENCE**  
of Persuasion

Revised Edition

---

National Bestseller







Now we're ready to talk about big projects

( The only difference between big & small is the scope of the risk )

What is big?

## What is big?

- too big to trust your gut
- too big for a reliable sniff test

## What is big?

- too big to trust your gut
- too big for a reliable sniff test
- "failure" is gonna upset someone

A Secret: Don't ignore analysis

A Secret: Don't ignore analysis  
( it's one of those "worst-case" things )

# Another Secret: Choose decision makers

## Another Secret: Choose decision makers

- For design

## Another Secret: Choose decision makers

- For design
- For budget

And get into the same groovy...

And get into the same groovy...

- EVERYBODY, read the same book(s)

"IMPORTANT, TIMELY, INSTRUCTIVE, AND ENTERTAINING."  
—Daniel Kahneman, Nobel Prize-winning author  
of *Thinking, Fast and Slow*

# HOW **BIG** THINGS GET DONE

THE SURPRISING FACTORS  
THAT DETERMINE THE FATE OF EVERY PROJECT,  
FROM HOME RENOVATIONS TO SPACE EXPLORATION  
AND EVERYTHING IN BETWEEN

BENT FLYVBJERG  
and DAN GARDNER

THE SCIENCE OF LEAN SOFTWARE AND DEVOPS

# ACCELERATE

Building and Scaling High Performing  
Technology Organizations



Nicole Forsgren, PhD  
Jez Humble, and Gene Kim

with forewords by Martin Fowler and Courtney Kissler  
and a case study contributed by Steve Bell and Karen Whitley Bell

4th Edition • OVER 650,000 COPIES SOLD

THE BEST-SELLING GUIDE TO DEALING WITH THE  
HUMAN SIDE OF ORGANIZATIONAL CHANGE

# Managing MAKING THE MOST OF CHANGE Transitions



William Bridges, PhD *with* Susan Bridges  
Author of the best-selling *Transitions*

Foreword by PATRICK LENCIOMI, Best-selling Author of *The Five Dysfunctions of a Team*



And One More Secret: Confidence is Dangerous



So let's get good at this

...give yourself some legitimate confidence

Goal:

I want to be excellent at estimating the work of an  
single team!  
(smaller projects)

...and have a solid plan for coordinating them all.  
(bigger projects made of many smaller things)

Start by walking backwards

The easiest work to estimate... is something that is  
already complete

Aside: the simplest estimate for something new

Aside: the simplest estimate for something new

- How much money do ya got?

Aside: the simplest estimate for something new

- How much money do ya got?
- How much time do we have?

## Aside: the simplest estimate for something new

- How much money do ya got?
- How much time do we have?
- Ok, we'll spend all that

## Aside: the simplest estimate for something new

- How much money do ya got?
- How much time do we have?
- Ok, we'll spend all that
- ...done



Worst case? -- What am I missing?

# Reducing Risk: Coordinating multiple teams/streams

# Reducing Risk: Coordinating multiple teamsstreams

Learn from air traffic control

# Reducing Risk: Coordinating multiple teamsstreams

Learn from air traffic control

- Lateral + Vertical + Timing + Visual

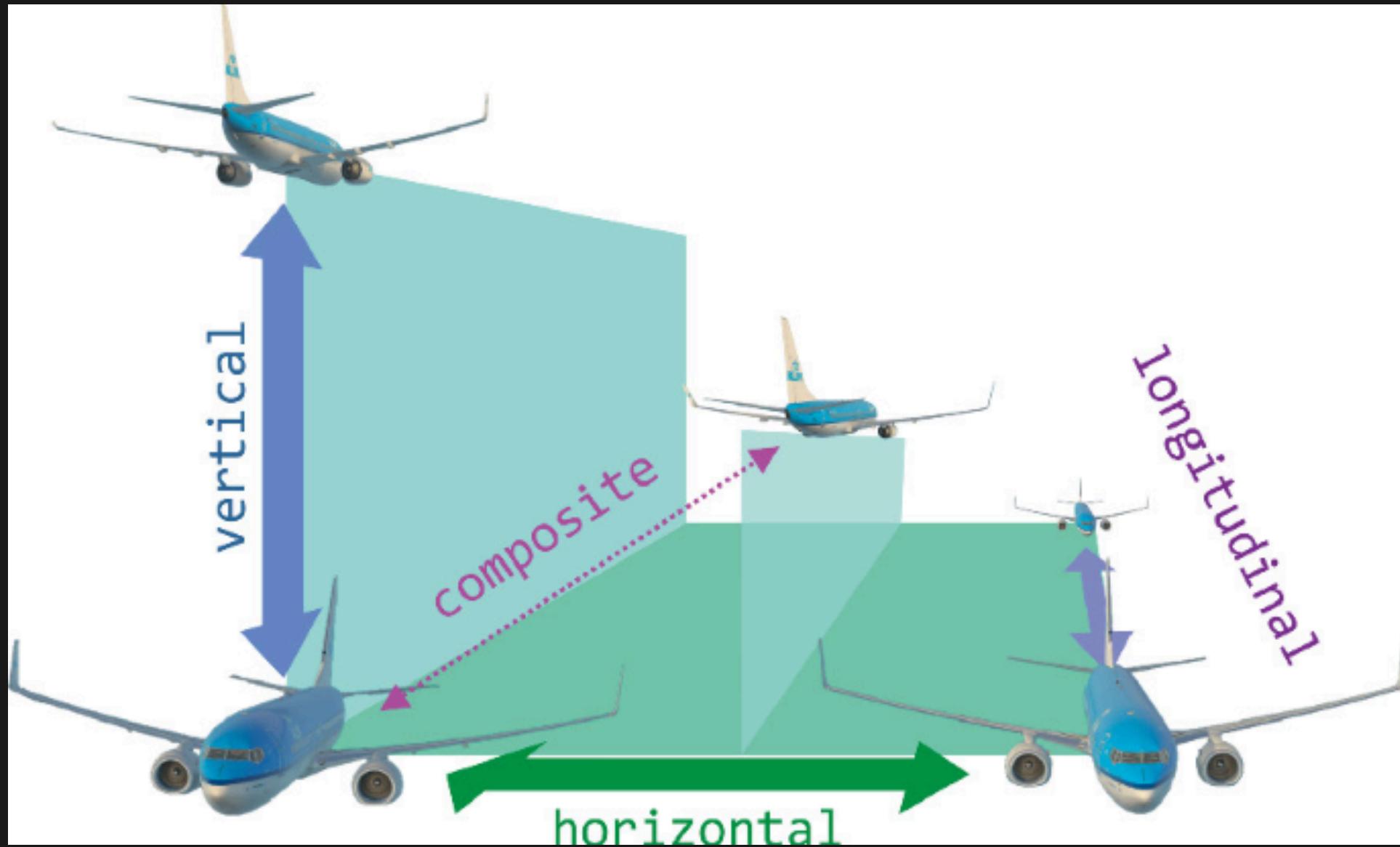
# Reducing Risk: Architecture

## Reducing Risk: Architecture

- so we can build it
- so we can maintain it
- so we can understand it

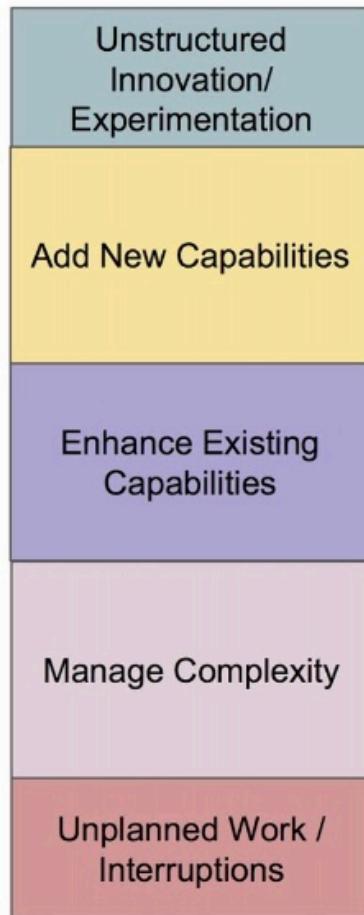
## Reducing Risk: Architecture

- so we can build it
- so we can maintain it
- so we can understand it
- so we can coordinate
  - Lateral + Vertical  
+ Timing + Visual

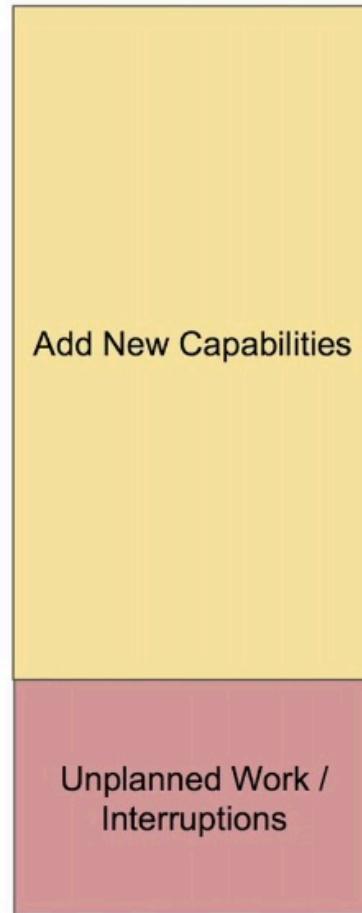


# Reducing Risk: Distractions (or necessities?)

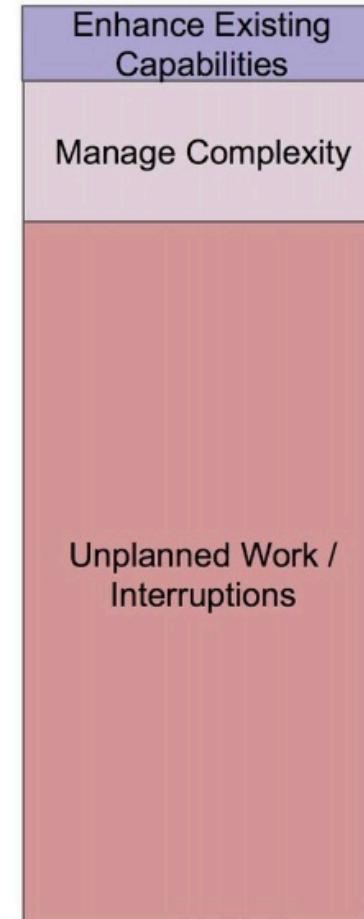
# John Cutler wrote:



**Do this ...**



**Not this**



**... or this will happen**

# Reducing Risk: The flexibility of your dev process

# Reducing Risk: The flexibility of your dev process

An effective process...  
...doesn't save the surprises for the end.

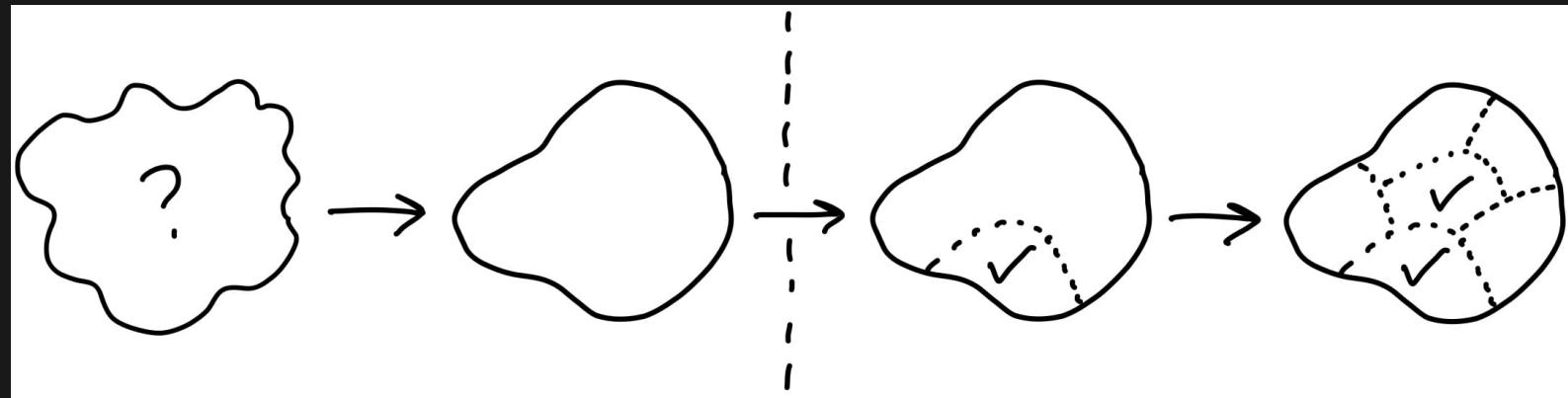
# Reducing Risk: The flexibility of your dev process

An effective process...  
...doesn't save the surprises for the end.  
...it spreads them out evenly over time

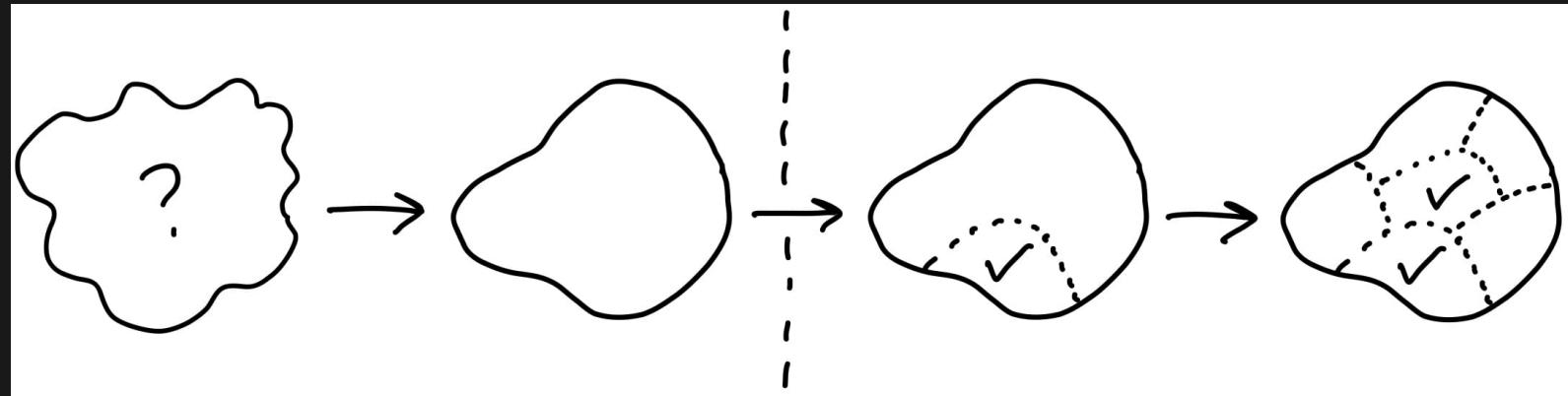


# "Shape Up" Process - Shaping + Betting + Building

# "Shape Up" Process - Shaping + Betting + Building



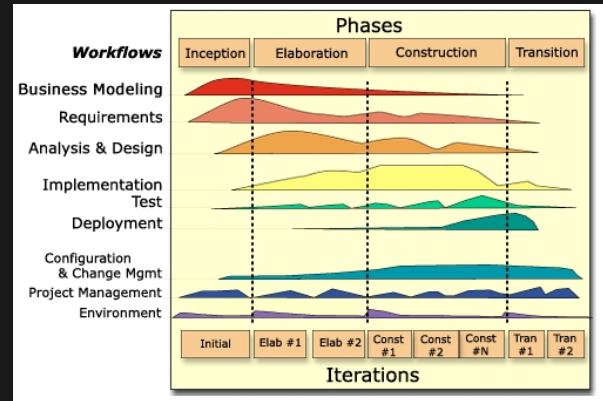
# "Shape Up" Process - Shaping + Betting + Building



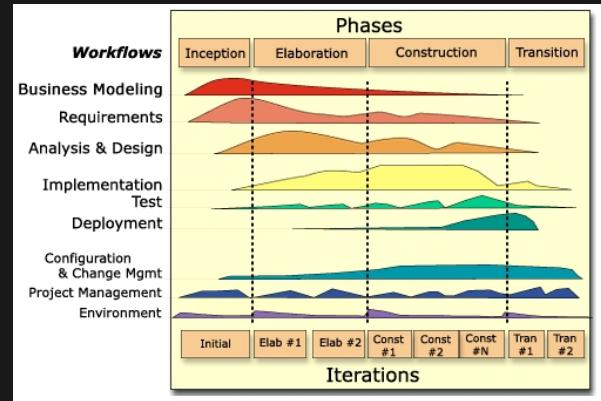
Good for teams; Not so effective  
for coordinating multiple teams

# "Unified" Process - Inception + Elaboration + Construction + Transition

# "Unified" Process - Inception + Elaboration + Construction + Transition



# "Unified" Process - Inception + Elaboration + Construction + Transition



Well, that's too complicated!  
( at least it doesn't ignore "transition" )

## "Irrational Unified" Process

- *Deception*
- *Confusion*
- *Recap and Documentation*
- *Destruction*
- *March 31*

## 6 Stages of A Project

1. Euphoria
2. Fear
3. Resignation
4. Search for the guilty
5. Punishing of the innocent
6. Praise and glory to the uninvolved



# WHEN TO SAY NO

When your reputation is on the line?

When it's too risky?

## Seven Deadly Sins (from "The Mess We're In")

1. Code you can't understand a week after you wrote it
2. Code with no specifications
3. Code that is shipped as soon as it runs and before it is beautiful
4. Code with added (unnecessary) features
5. Code that is very very fast very very very obscure and incorrect
6. Code that is not beautiful
7. Code you wrote without understanding the problem

## Extending legacy code (more from "The Mess We're In")

- Programmers who wrote the code are dead
- No Specification
- Written in archaic languages which nobody understands
- "it works"
- Management thinks modifying legacy code is cheaper than a total re-write

When it's too complex

...combined with one giant estimate  
...and no architecture

# "AGILE" ESTIMATION

Yes, it is possible to be good at estimating large projects.

# References

- *Accelerate; Building and Scaling High Performing Organizations*, Forsgren, Humble, and Kim
- *Applying Use Cases*, Schneider & Winters, with Karner's research
- *How Big Things Get Done*, Flyvbjerg & Gardner
- *Influence; The Psychology of Persuasion*, Robert Cialdini
- *Managing Transitions*, Robert Bridges
- *The Mess We're In*, Joe Armstrong, Strange Loop
- *Philip and Alex's Guide to Web Publishing*, Philip Greenspun
- *The Visual Display of Quantitative Data*, Edward Tufte

