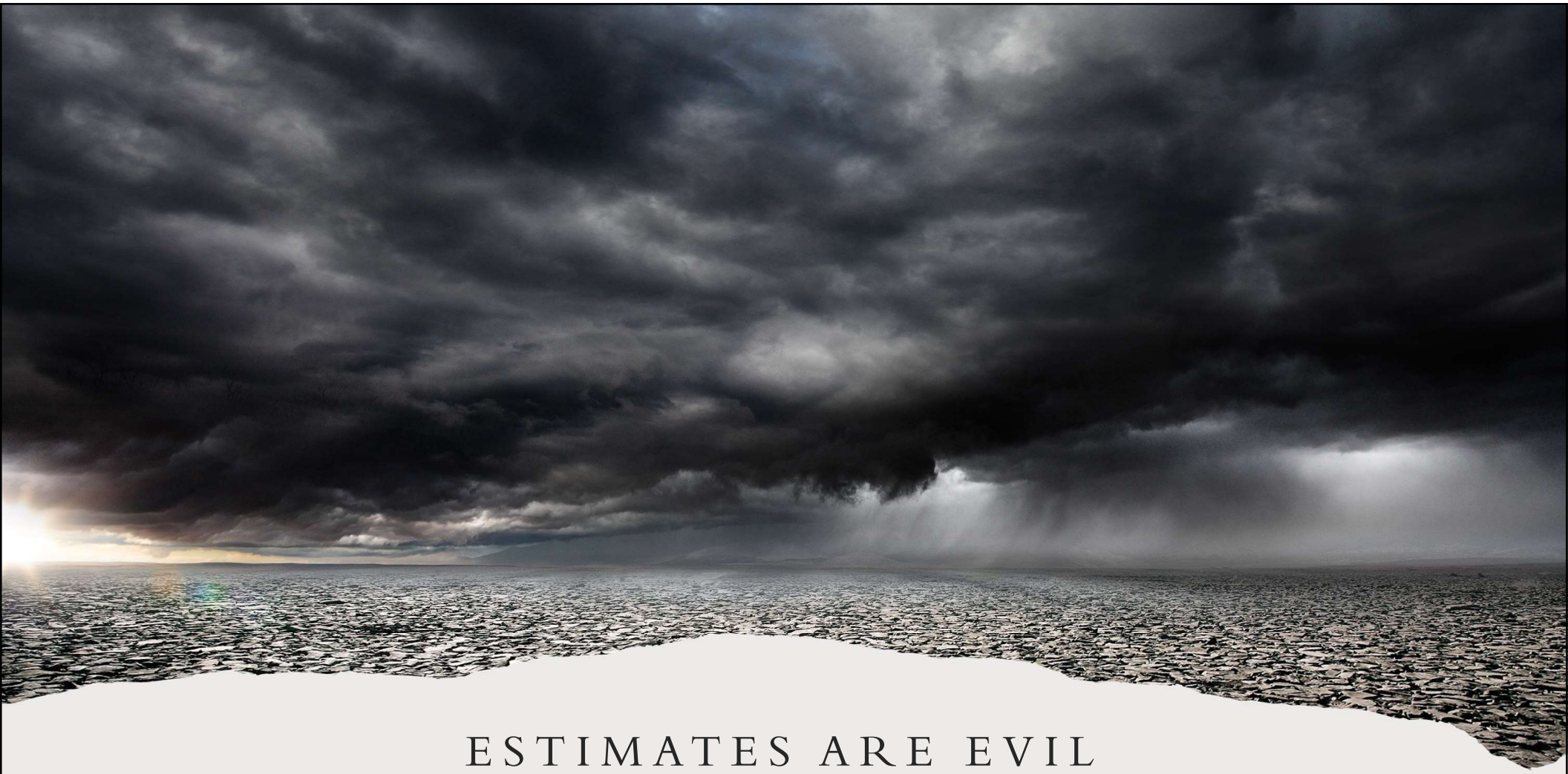


ESTIMATES ARE EVIL

but can be used for good

Jennie.ocken.org

PRODUCT MANAGERS ARE EVIL

- Different sizes of company
  30 <-> 94,000 (~800 in IT)
- Different Business Pressures
  SaaS product <-> Internal business applications
  Direct sales of development time <-> Development to create services efficiency
- Different business structures
  VC funded startup <-> Privately owned <-> Publicly traded
- Different technical challenges
  Mainframe forklift <-> Startup “greenfield”
  Monolith <-> Microservices
- Different Roles
  Sales/Marketing <-> Engineering department head
  Single team embed <-> Head of Product
- 15+ years in Product Management

“ARE WE THERE YET?”

GOOD GODS BUT WHY: ENABLING DECISIONS

- Converting time into money

  "If it costs 1 sprint to build it and I can sell 5 licenses…"

- Confronting flexibility vs predictability

  "Can't we just slip this one piece of scope in?"

- Predicting against uncertainty

  "But Jennie, we need to present a compelling roadmap"

- Converting team performance into delivery

  "Huh, I didn't think about approaching it that way"

BUSINESS DECISIONS: COSTING

Selling My Time Directly

- "Billable" Hours – cost of goods sold
- "Billable" Hours – opportunity cost
- "Cost plus" – what are materials in software?

Selling a Good My Time Goes Into

- Licensing – how many more seats
- Freemium and Product Lead Growth – stepping up engagement
- Internal Services – making other people more efficent

BUSINESS DECISIONS: PRICING

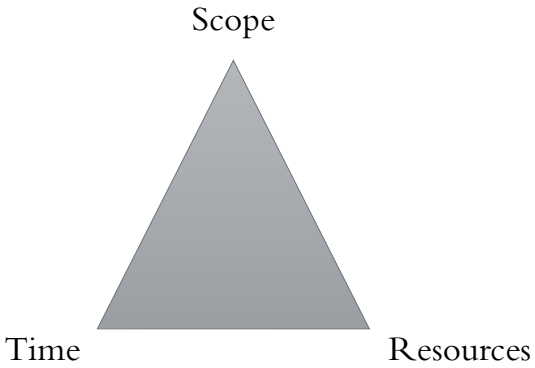
Revenue	*Loss Leader*	*Trouble*
Value	Value	Price
Price	Cost	Value
Cost	Price	Cost

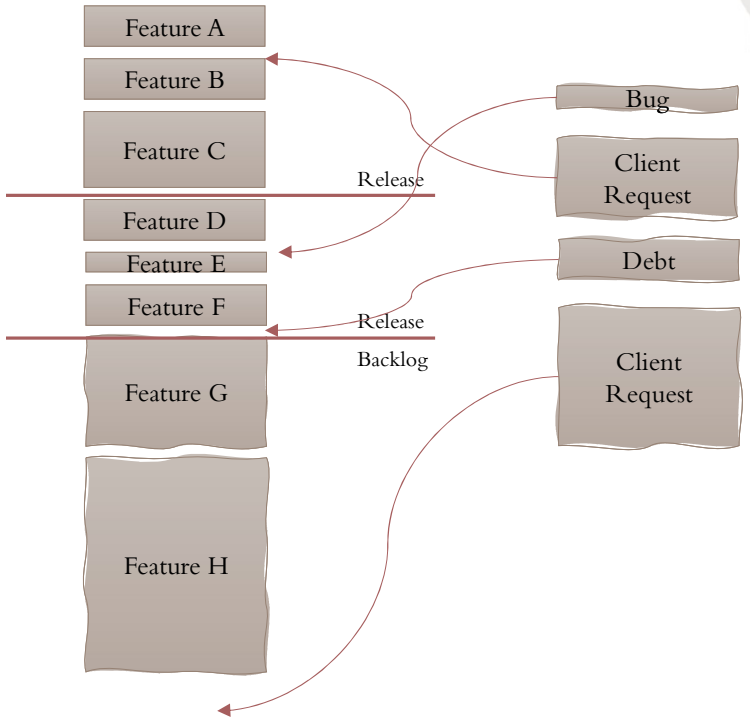
PRODUCT DECISIONS

Debt Paydown

Cost

Value





TECHNICAL DECISIONS

- Where do we want to start? What kinds of things do we need to learn?

  Estimates as a way into an approach discussion

- What happens if we tear into this wall? How do we recover when something goes wrong?

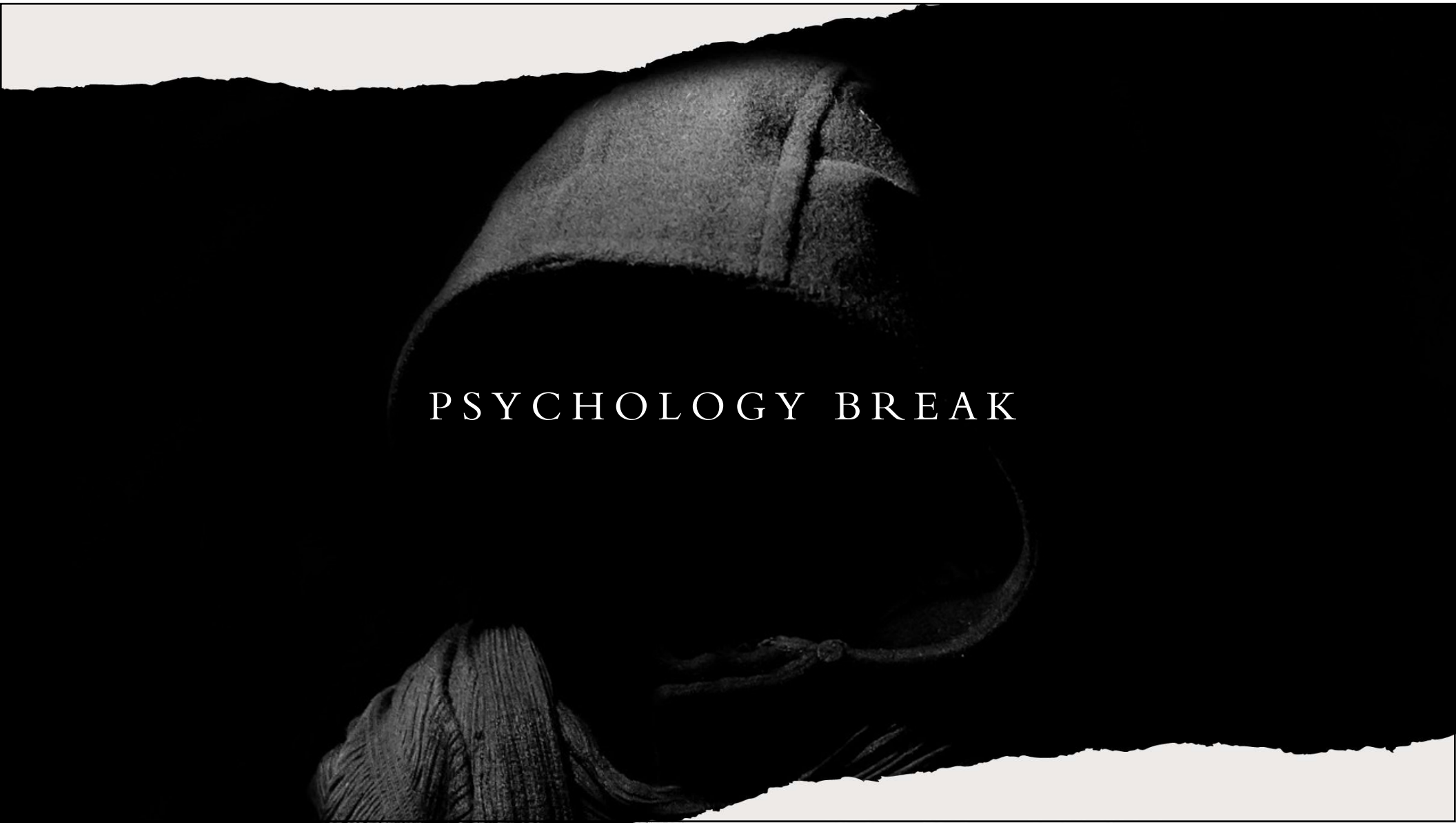
  Spikes and when things go wrong

- How do we balance skills across our team?

  Jon Expert vs Aaron Middle Tier vs Steven Beginner

- How do we divide up the work?

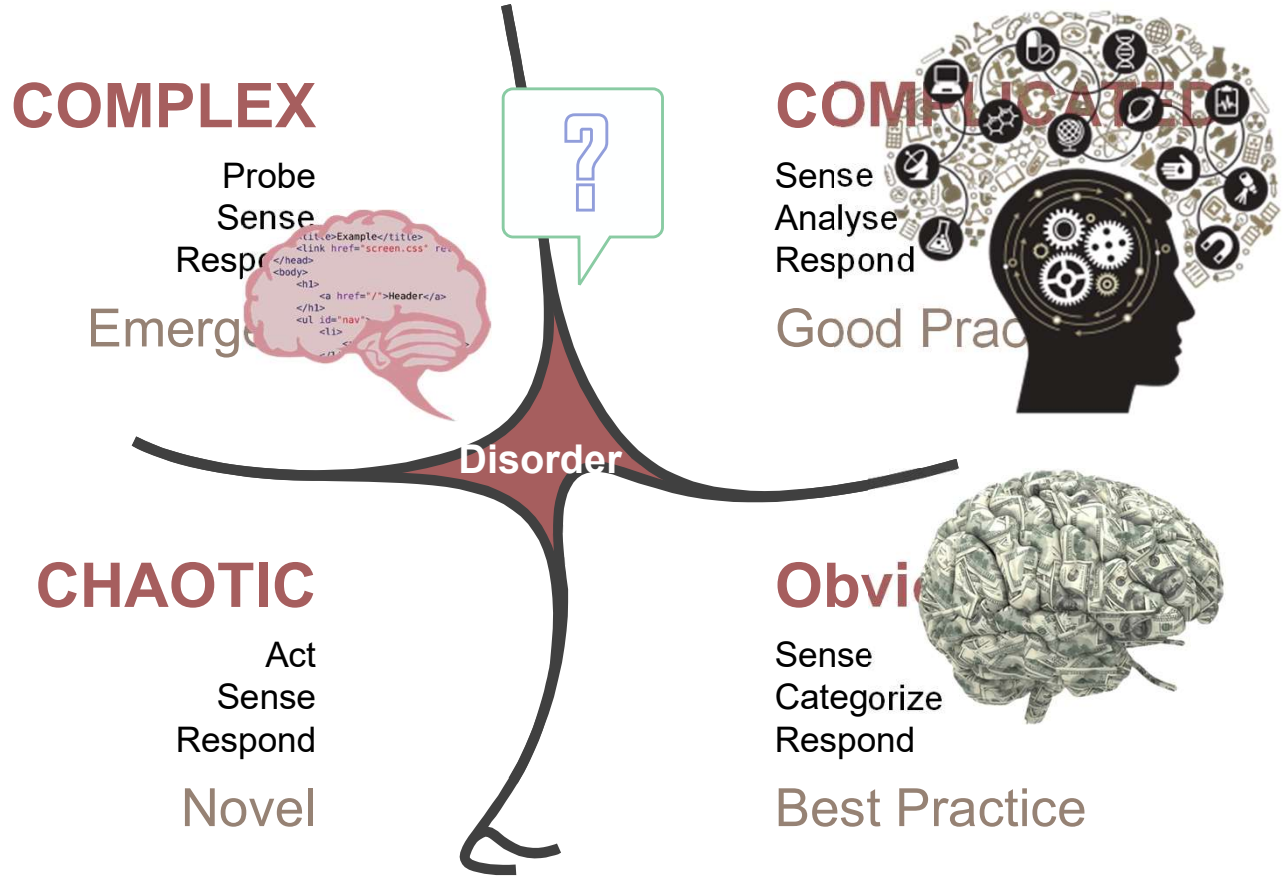
  Bathroom job vs Kitchen job


PSYCHOLOGY BREAK

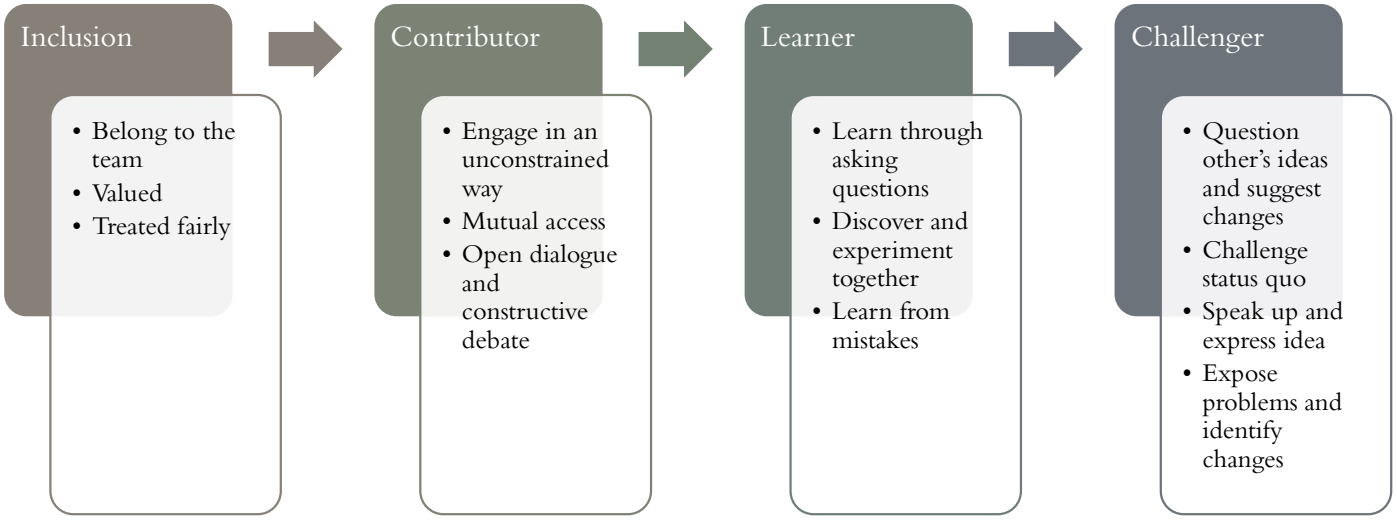
CYNEFIN

- Developed in 1999 and first published in the Hard Business Review in 2007 by David Snowden and Mary Boone
- Welsh for place or habitat

CYNEFIN


COMPLEX
Probe
Sense
Respo
Emerge
COMPLICATED
Sense
Analyse
Respond
Good Prac
Disorder
CHAOTIC
Act
Sense
Respond
Novel
Obvi
Sense
Categorize
Respond
Best Practice

PSYCHOLOGICAL SAFETY



HIGH SAFETY

- High trust
- Learn together
- Challenge each other
- Innovative

LOW SAFETY

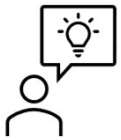
- No risk taking
- Fear driven
- Blame culture
- No problem solving

HOW TO CREATE PSYCHOLOGICAL SAFETY

- Make it an explicit priority
- Facilitate everyone speaking up
- Establish norms for how failure is handled
- Create space for new ideas
- Embrace productive conflict

THE RAVAGES OF TIME

WHAT QUESTION ARE WE TRYING TO ANSWER?



Ideation

Epics

Is the cost of this product going to be worth the investment?

Months-Years
Gut Instinct

The Estimating Game

Discovery

Features

What is the right set of features for learning something from? What deliveries customer value?

Weeks-Months
High Level Requirements

Sprints or T-shirt size
Delivery Planning

Commitment

Stories and Tasks

What are delivering this sprint and are we all on the same page? What can I tell the clients about what's coming?

Days-Weeks
Planning

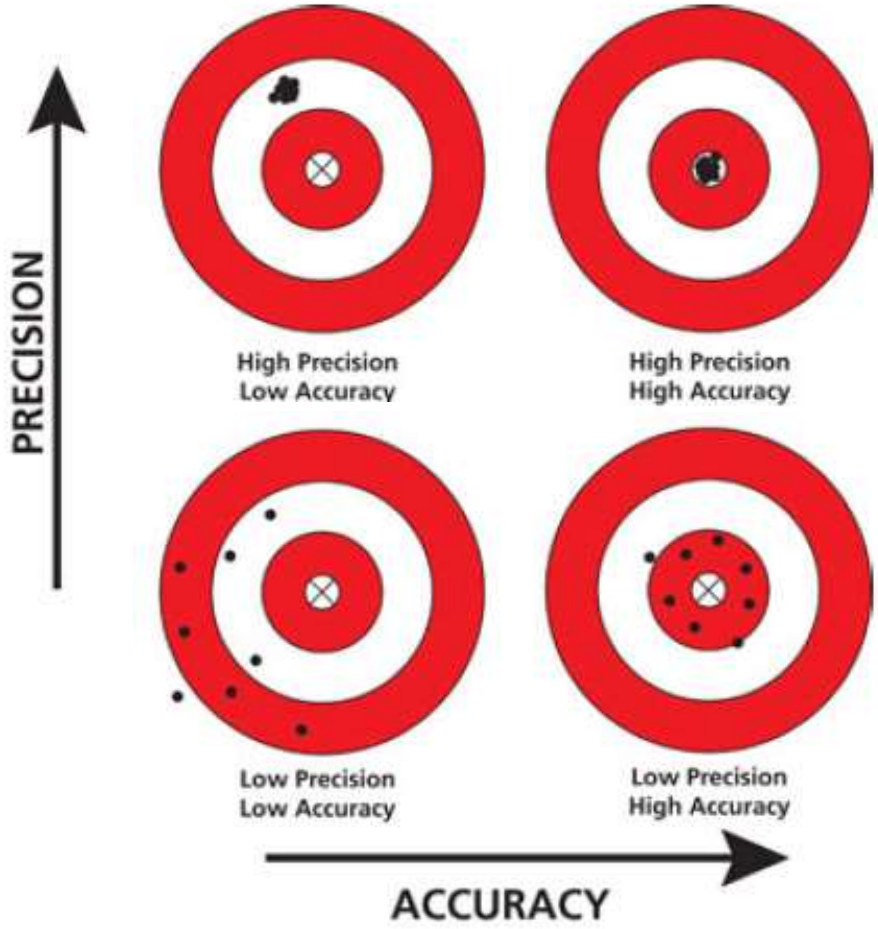
Sprints or Days or Hours
Points or IDD
Relative pointing

Delivery

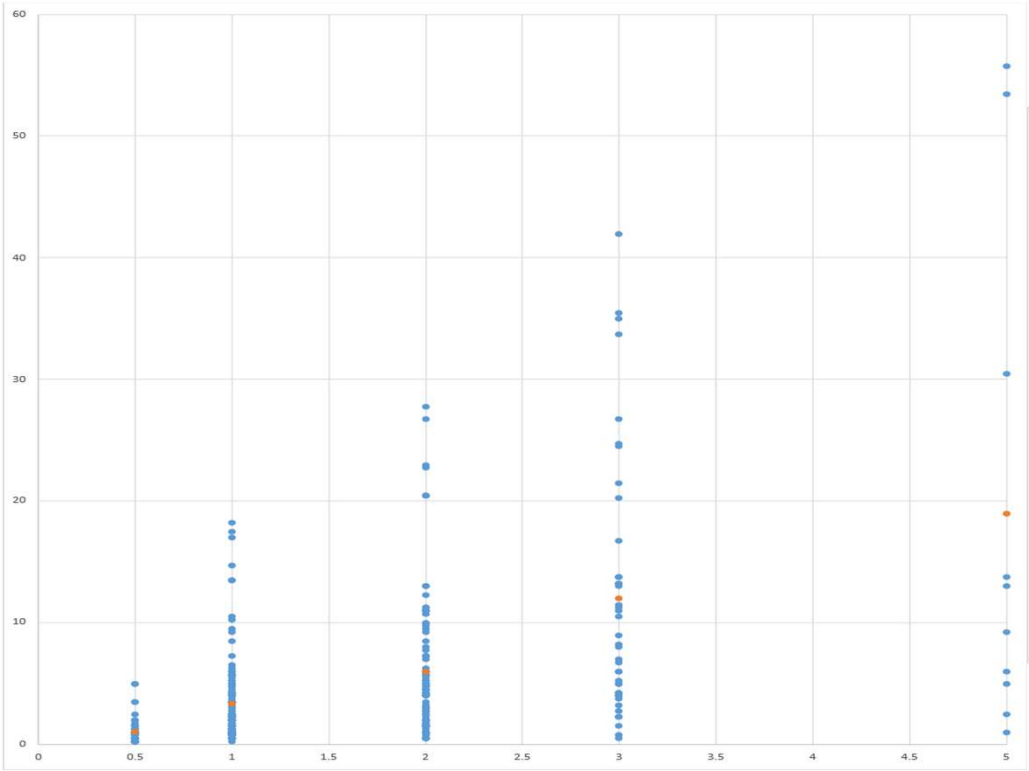
Tag or Release

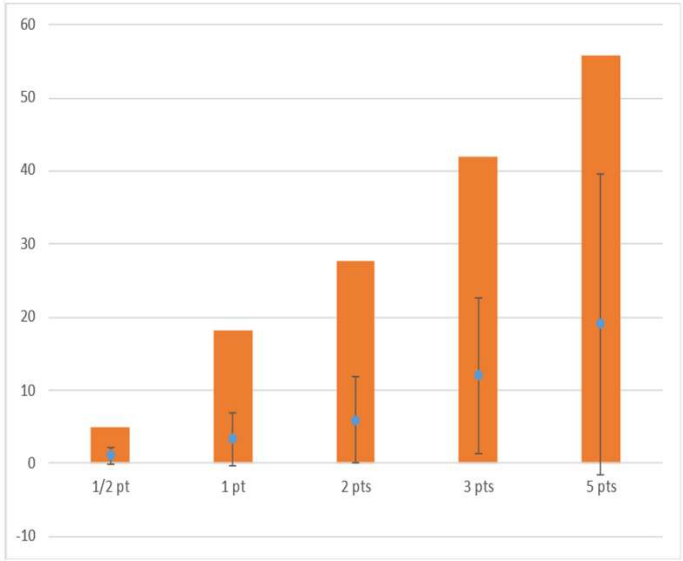
What goes into the release notes? What can I market?

ACCURACY VS PRECISION



THE DEVIL IS IN THE POINTS





ROADMAPS AND DELIVERY PLANS

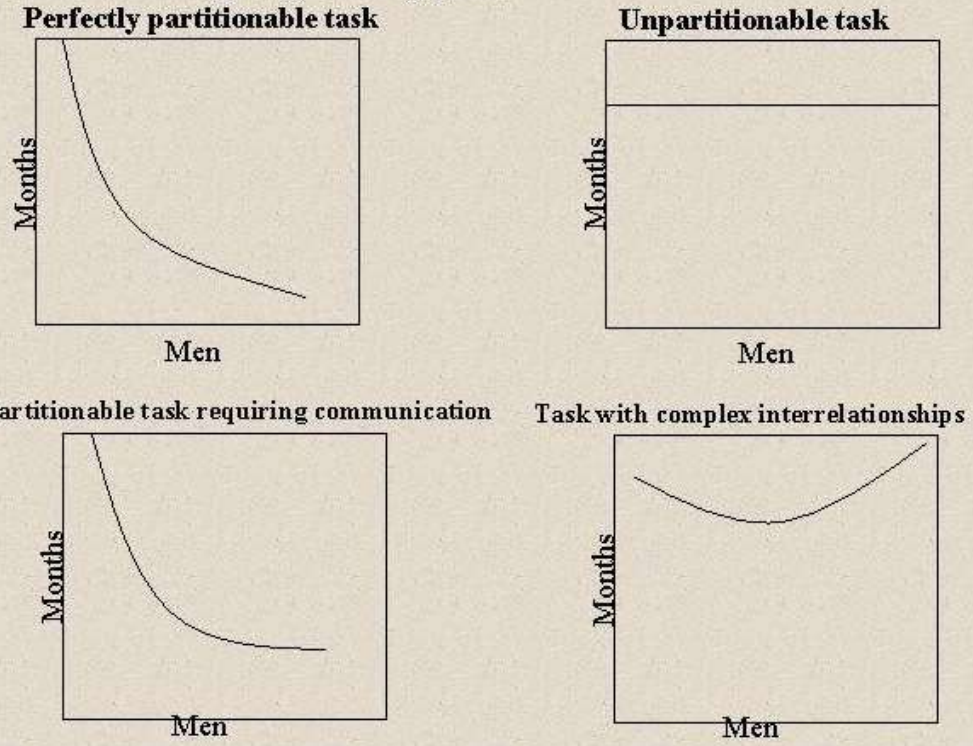
- A roadmap is a statement of mission and landscape
- A delivery plan is an expectation of when we will hit certain milestones

- If you give someone a time-based estimate, they will turn it into a gaunt chart
- If you don't give someone a time-based estimate, they will make up a timeframe and turn it into a gaunt chart

THE MYTHICAL MAN-MONTH

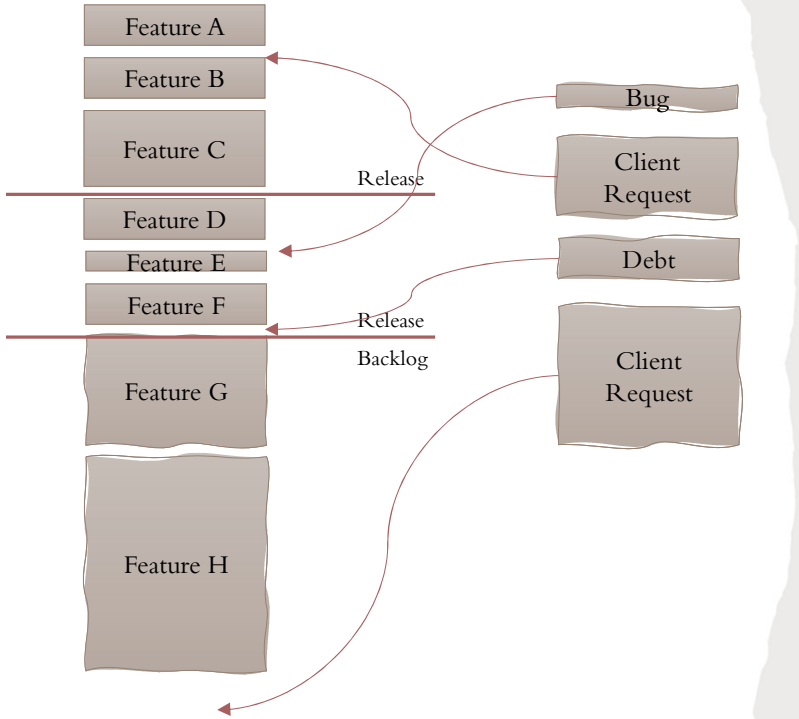
- Why estimation goes poorly
  - Assume all will go well ("programmers are optimists")
  - Assumes that men and months are interchangeable
  - Lack courteous stubbornness
  - Progress is poorly monitored
  - The response to delay is to add people
- Brook's experience with actual task breakdown
  - 1/3 planning
  - 1/6 coding
  - 1/4 component test and early system test
  - 1/4 system test, all components in hand
- Brook's law: Adding manpower to a late software project makes it later

- Different projects types:



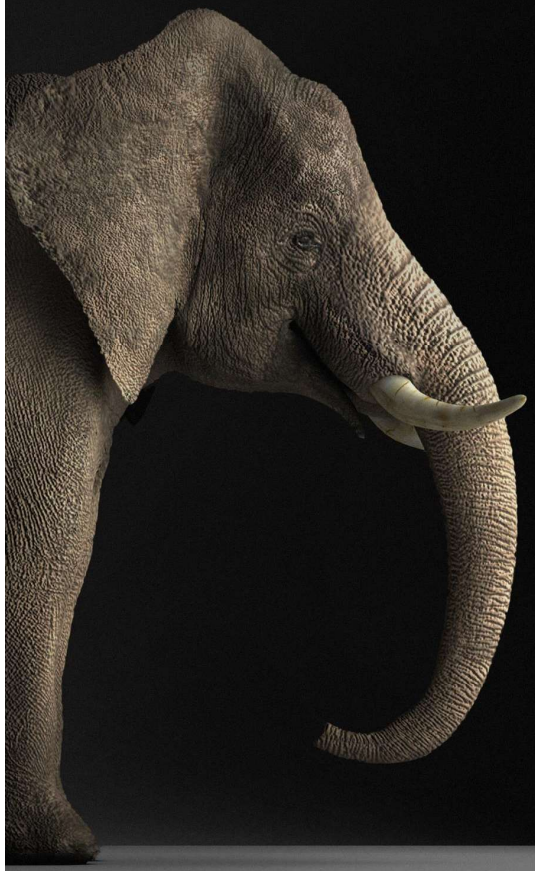
PLAYING THE GAME

- The further away something is, the more uncertainty we *should* have about it
  - Big to small
  - High uncertainty to low uncertainty
  - Doubling the doubled, padding the pad
- Know how the estimate is going to be used and who is going to see it
  - Development for hire vs software for direct sales vs internal services vs service support
  - The difference between an estimate and a cost



A WORD ON DEBT

- If you don't include these things in the up-front estimation time, you will be buying debt you will be paying with interest later:
  - User research and product market fit
  - Discovery and spikes
  - Testing
  - Documenting and releasing
  - Product marketing and training

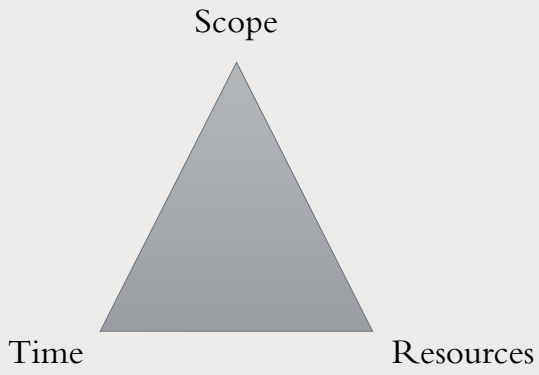


TERRIBLE IDEAS

- Comparing across teams
- Individual performance measurement (just as bad as lines of code)
- Translating points to hours (just do hours, you aren't doing anything with points anyway)
- Using estimates to price (it's a costing mechanism)
- Detailed break downs too early (during ideation or discovery)
- Too high level too late (aka the JFDI task)

ONE MORE THING

- Sometimes set the timeframe and resources instead of the scope



QUESTIONS

Jennie.ocken.org jennie@ocken.org @jennieocken