

The CREATE Framework

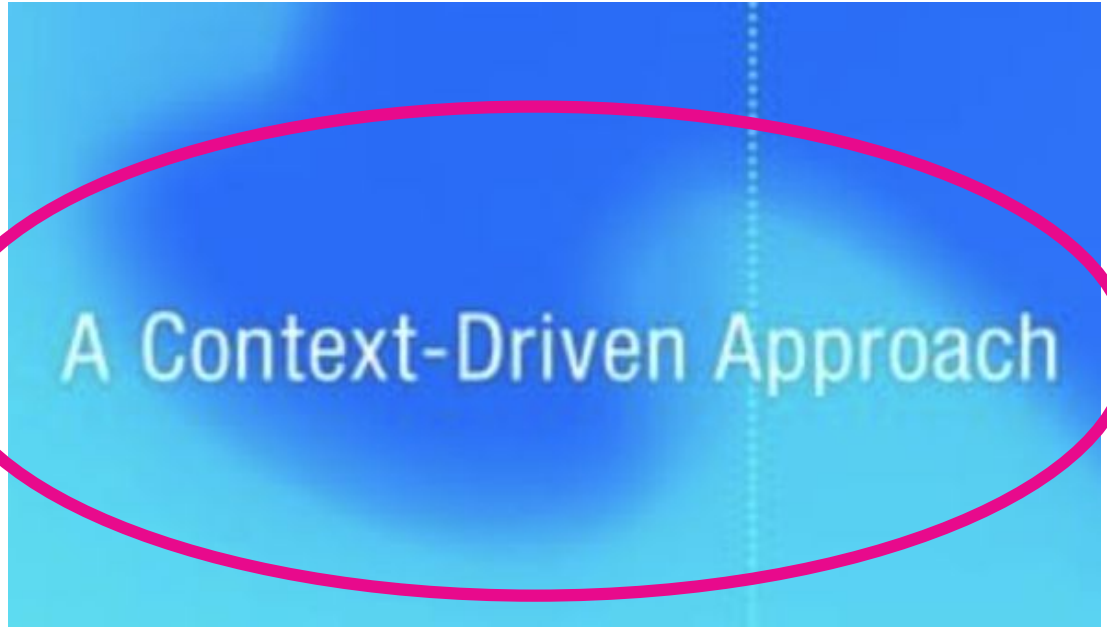
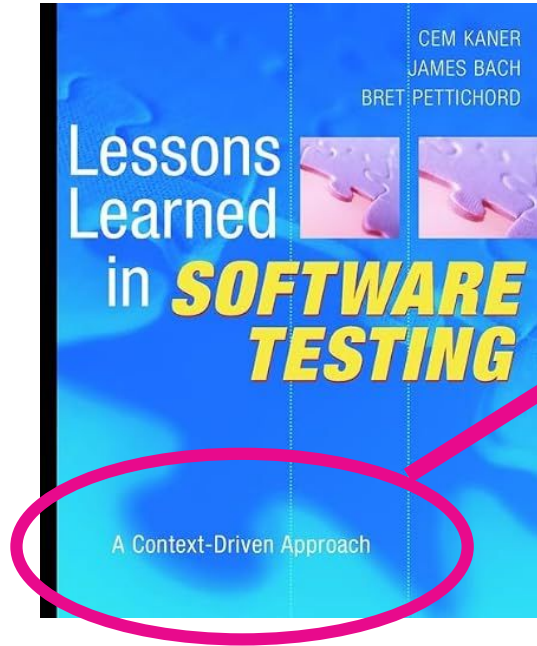
how to reason about context

Elisabeth Hendrickson
CAST 2026

About Context-Driven Testing

(a short historical digression
before we really get started)

Flashback 2001 – a movement gets a name



7 Context-Driven Principles

see context-driven-testing.com

1.

The value of any practice
depends on its context.

context-driven-testing.com

2.

There are good practices in context,
but there are no best practices.

context-driven-testing.com

3.

People, working together, are the most important part of any project's context.

context-driven-testing.com

4.

Projects unfold over time in ways that
are often not predictable.

context-driven-testing.com

5.

The product is a solution. If the problem isn't solved, the product doesn't work.

context-driven-testing.com

6.

Good software testing is a challenging
intellectual process.

context-driven-testing.com

7.

Only through judgment and skill, exercised cooperatively throughout the entire project, are we able to do the right things at the right times to effectively test our products.

context-driven-testing.com

About Context

say it with me...

How long will it take to...?

it depends

What's the best way to...?

it depends

Should I...?

it depends

How do I...?

it depends

What does it depend on?

context

target customer segment

state of the org

company strategy

company policies

competitive landscape

tech stack

OK so what is **context**?

dependencies

current process

current code quality

funding / runway

other priorities

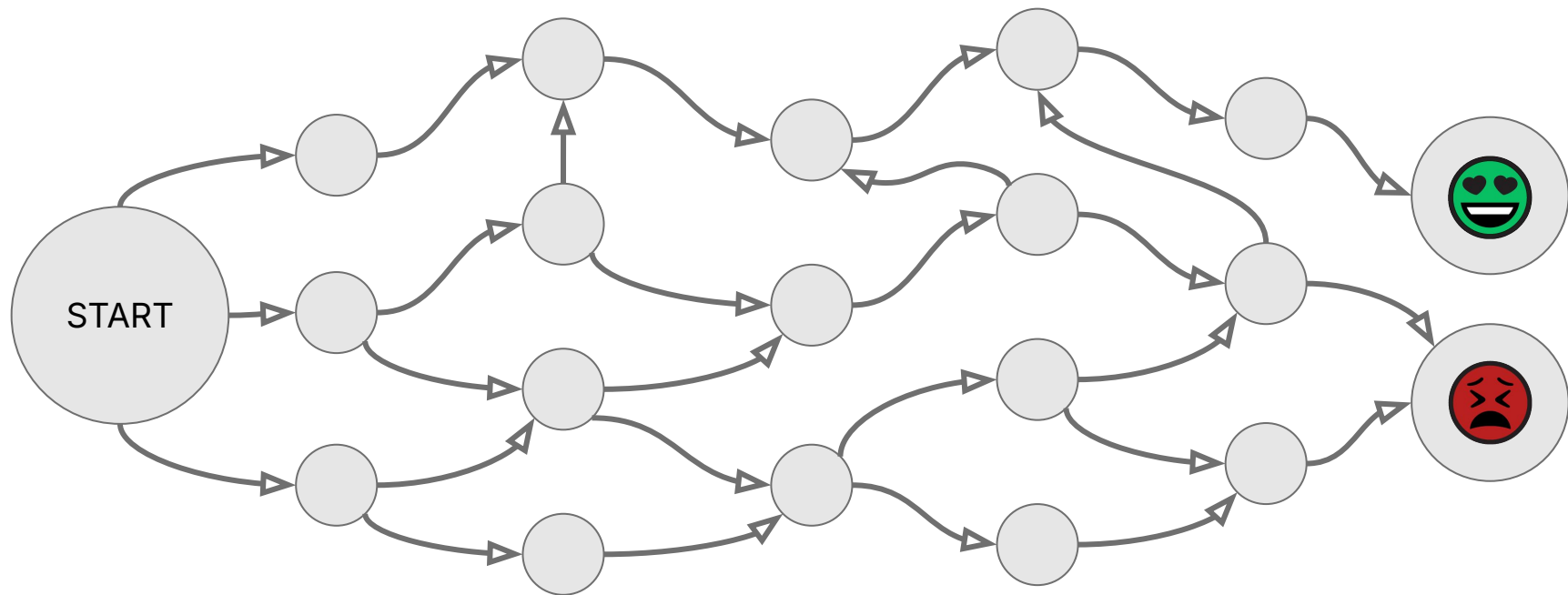
state of the team

Your context
is the cumulative effect
of every past decision.

“Things are the way they are because
they got that way.”

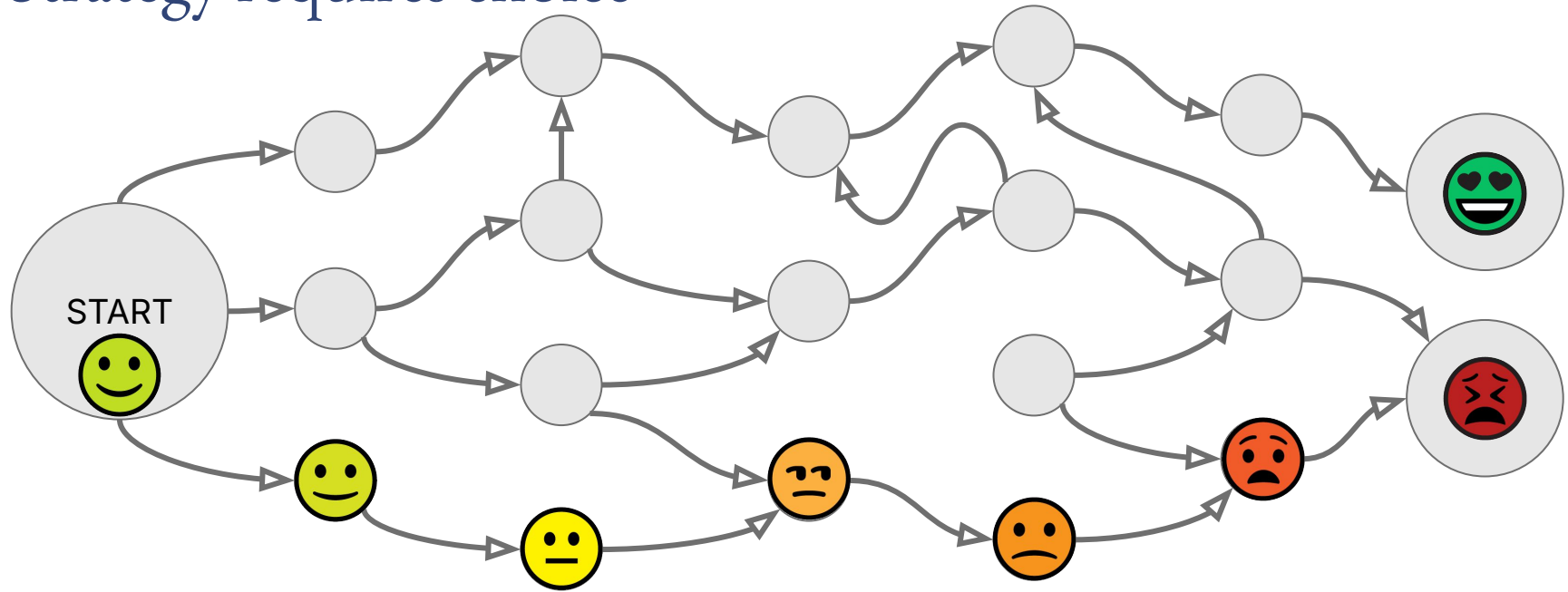
Gerald M Weinberg
quoting Kenneth Boulding
in *Secrets of Consulting*

Think of your context like a board game...



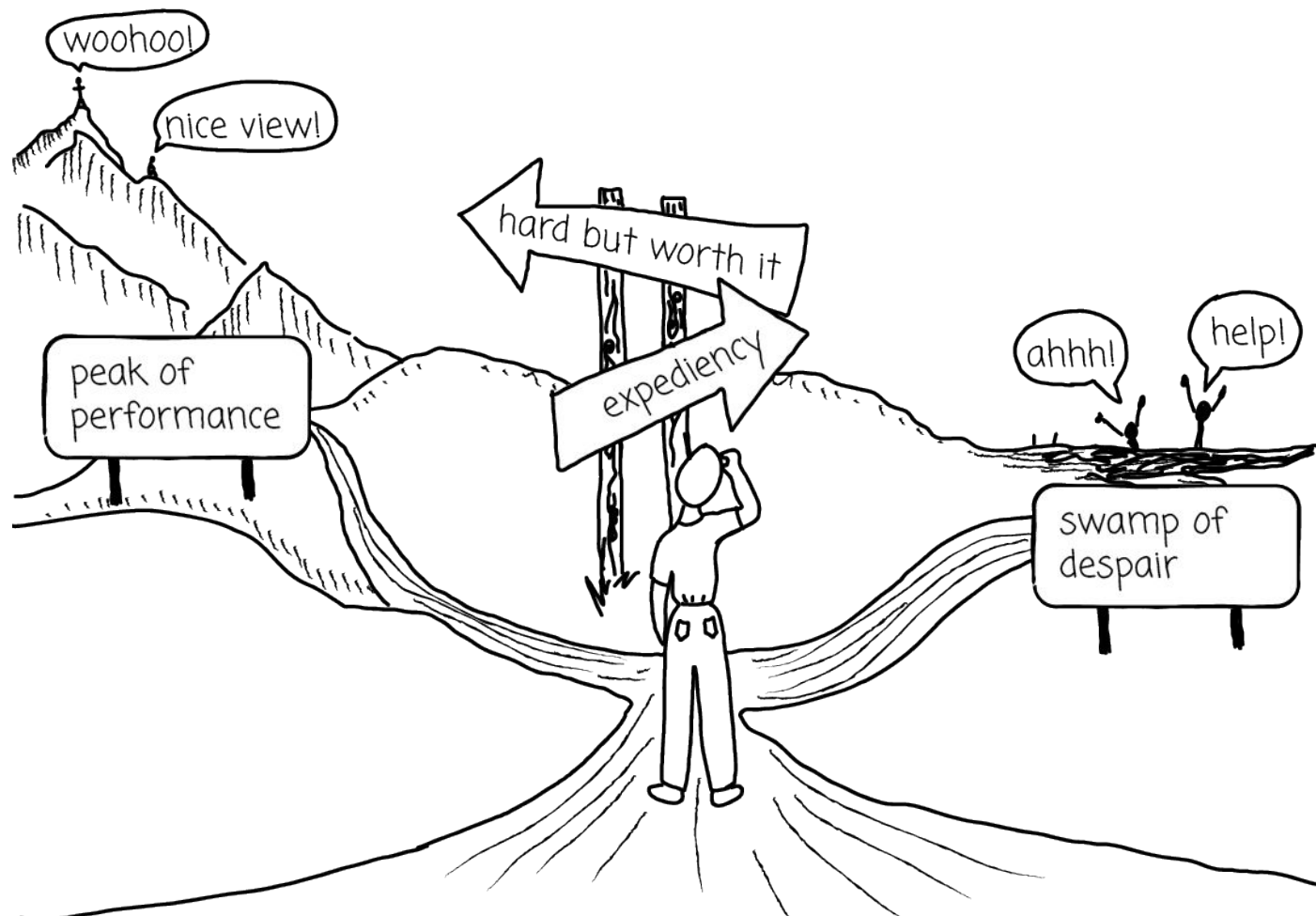
...past choices determine current options.

Strategy requires choice



Do your future self a favor

Favor decisions today that
give you better choices tomorrow
to steer your strategy into the future.



Recognizing context is not enough.
You have to be able to talk about
context to decide on a strategy.

The CREATE Model

6 dimensions of context to consider

capacity

How much can
we do?

risk

What could
wrong & how
bad would it be?

execution

How do we
work? How well?

adaptability

How quickly can
we change
course?

trust

How do people
treat each other?

economics

What's the
"bottom line"?

About Problems

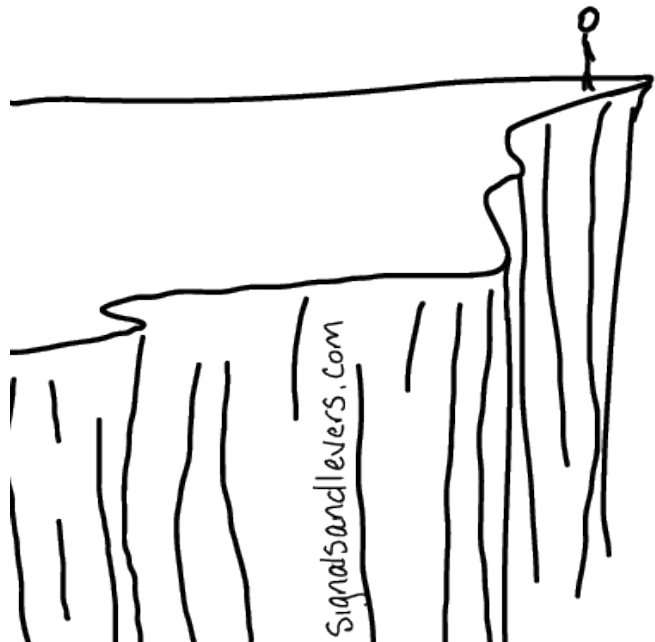
it's really hard to craft
a good problem statement

“A problem is the difference
between things as desired and
things as perceived.”

– Jerry Weinberg & Donald Gause
Are Your Lights On

perceived state

standing on one
side of a ravine

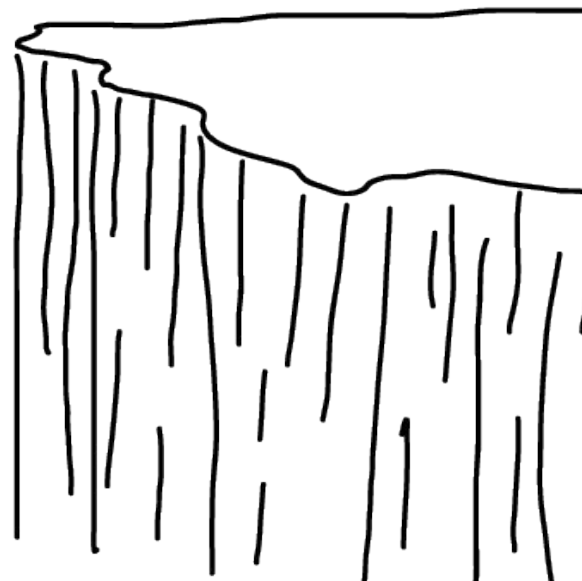


SignalsandLevers.com



desired state

be on the other
side of the ravine



problem

how to get to
the other side

(not the lack of a bridge,
or lack of a helicopter, or
lack of a giant pet eagle)

problem

“We want every team to be productive, but time & distance make it impossible to collaborate.”

Context...

- Large enterprise software product
- Multiple geographically distributed teams with time zone and language barriers
- Codebase was a monorepo where theoretically each team “owned” subdirectories but in practice there were overlapping responsibilities
- Waterfall process with long QA phase

capacity

split between
geographic areas

risk

critical
enterprise
infrastructure

execution

traditional
process

adaptability

not considered
important

trust

high within geos,
low across geos

economics

constrained
budget

Decision Point

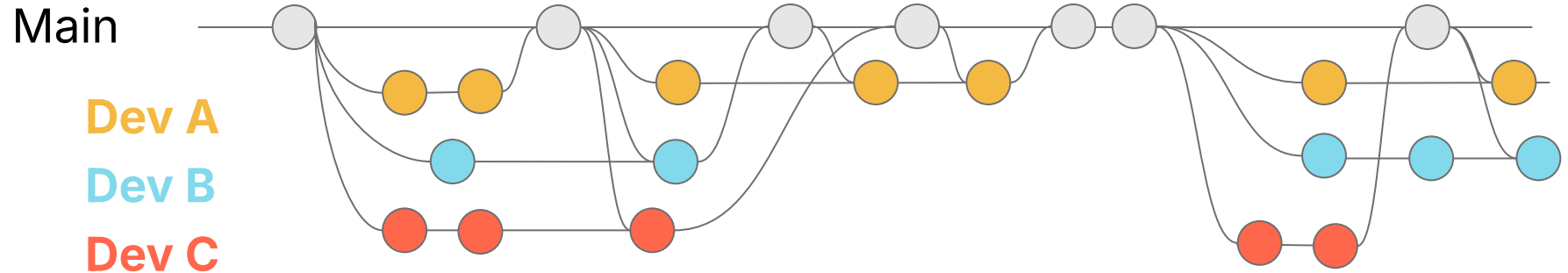
What branching strategy
should we use?

Option: Living on Main

Truly continuous integration.

Small batches, fast feedback.

BUT: not compatible with the QA process.

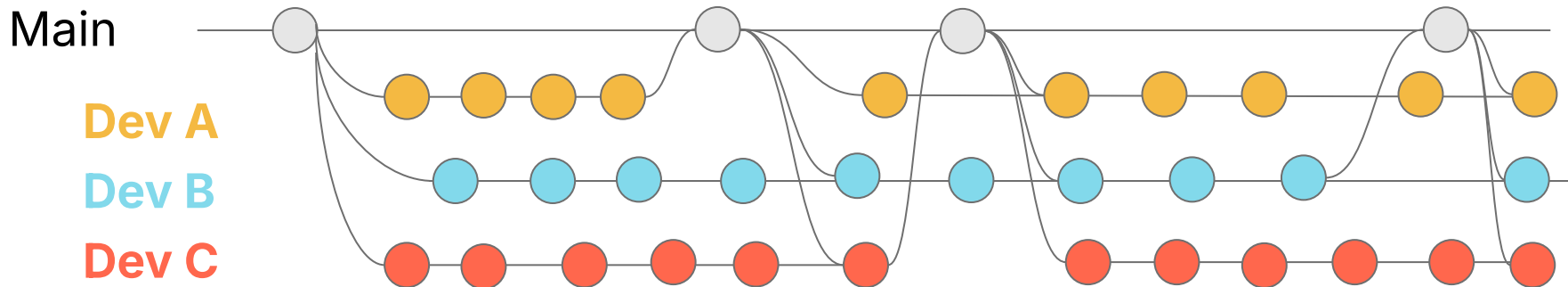


Option: Feature Branching

Big batches. Much merging.

More compatible with the QA process.

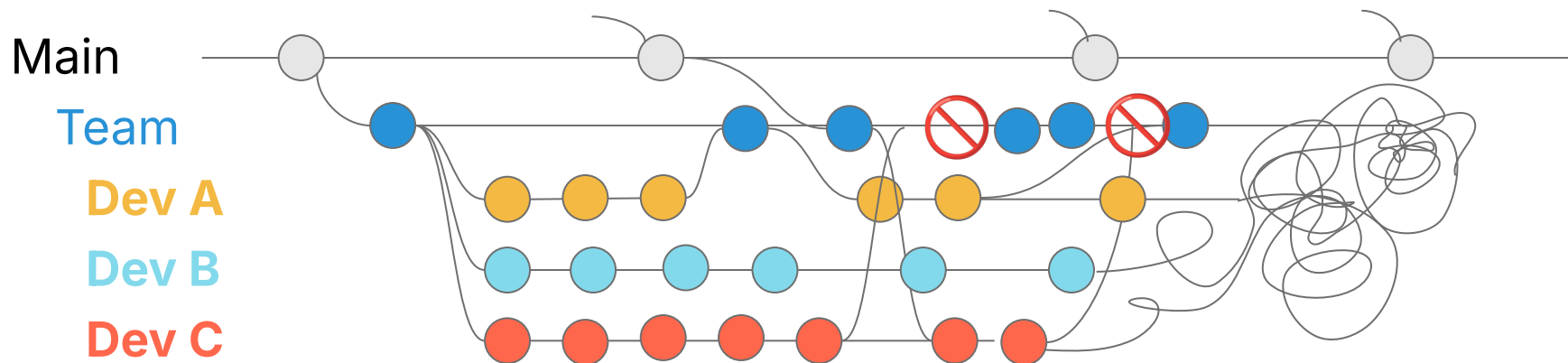
But still painful across timezone and language barriers.



Option: Long Running Team Branch

We can be agile locally.

Possible merge conflicts? Nah. It will be fine!



Consequence

Illusion of progress...for a while.
Then many months of work lost.

If you are very good, and very lucky, you get to choose which problems to have.

Trade-offs & U-Curves

Framing problems, finding a range of solutions

More People Doesn't Make Things Go Faster

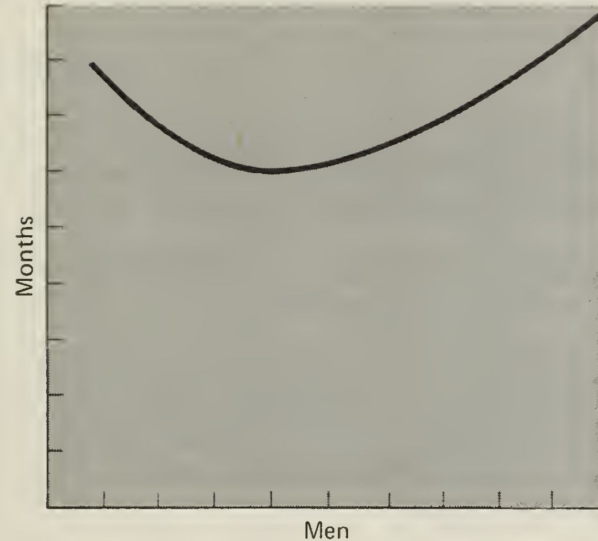
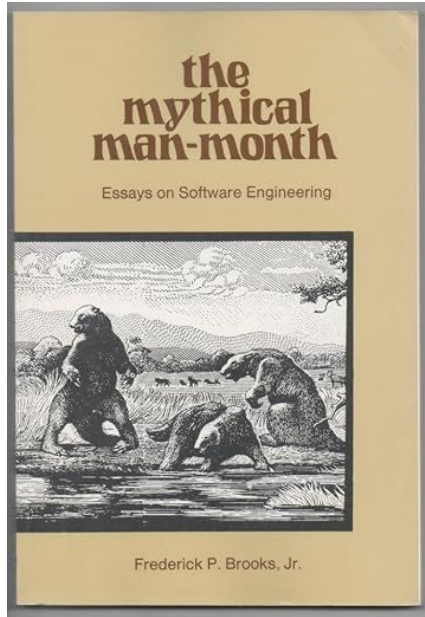
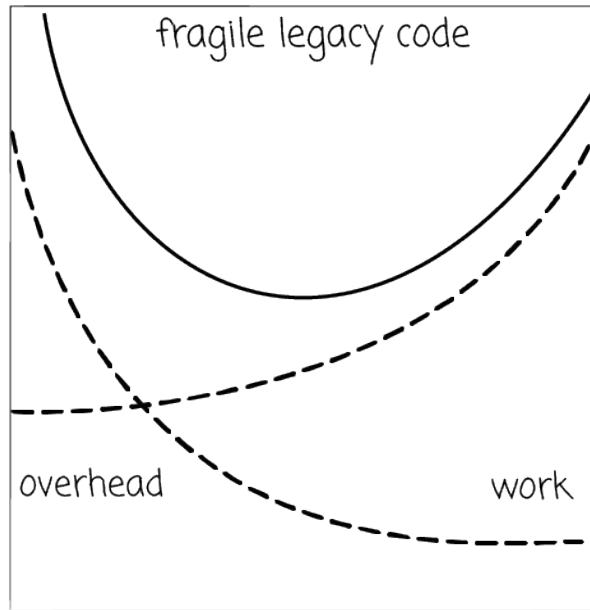
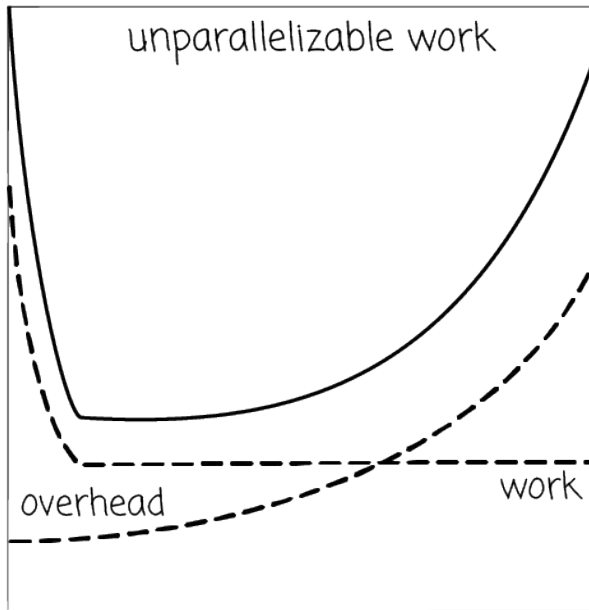
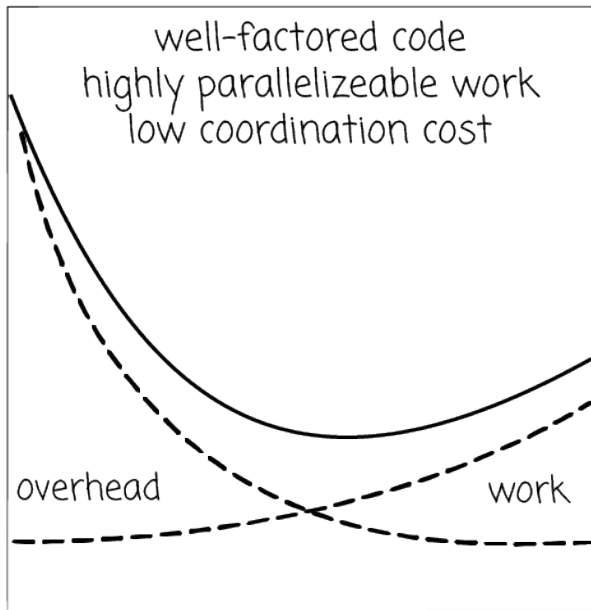


Fig. 2.4 Time versus number of workers—task with complex interrelationships

Context-Specific Variations on the Mythical Man Month

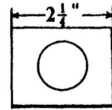
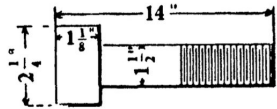


Happy Medium?



Whenever you have a tradeoff,
there is usually a sweet spot
between the extremes.
But how to find it?

Example: Batch Size



Article = A Stud

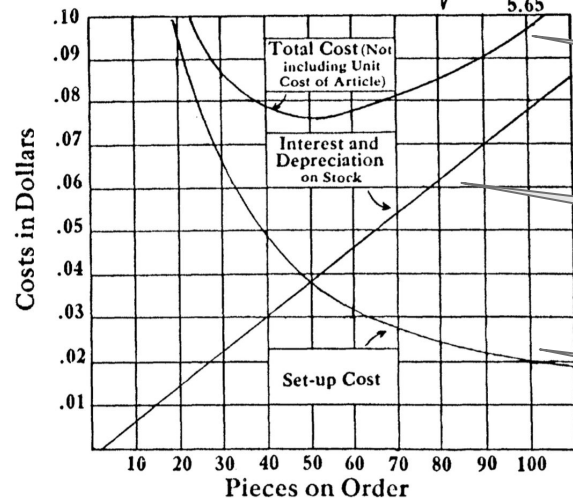
Material = Copper

Quantity per Month = $M = 30$

Cost per Piece. = $C = 5.65$ Dollars

Set-up Cost, = $S = 1.85$ Dollars

Manufacturing Quantity = $x = 48.5 = \sqrt{\frac{240 \times 30 \times 1.85}{5.65}}$



Total Cost

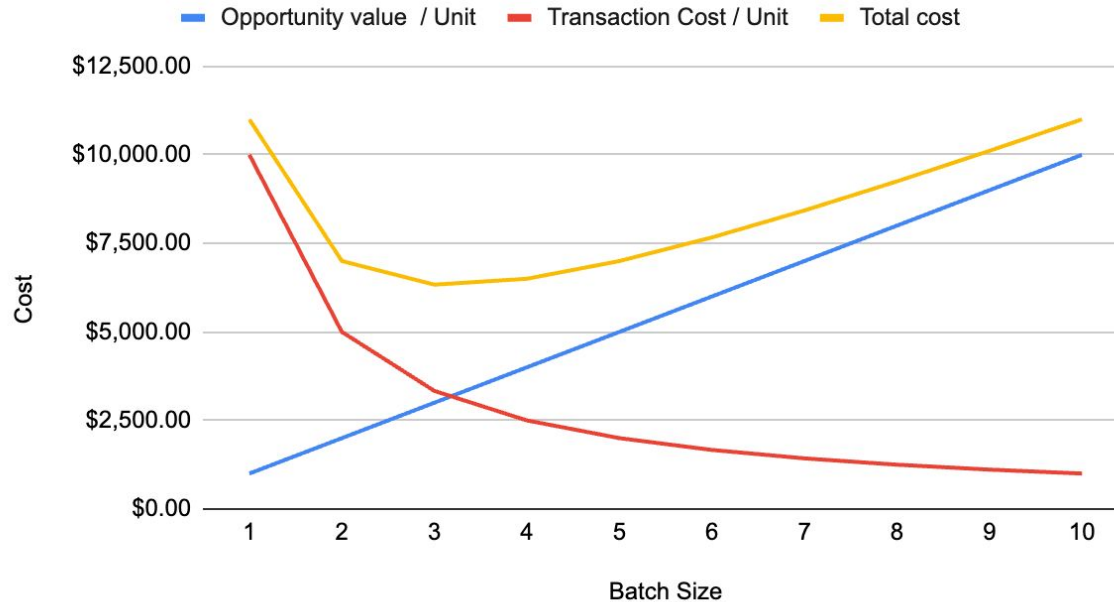
Holding Cost

Setup Cost

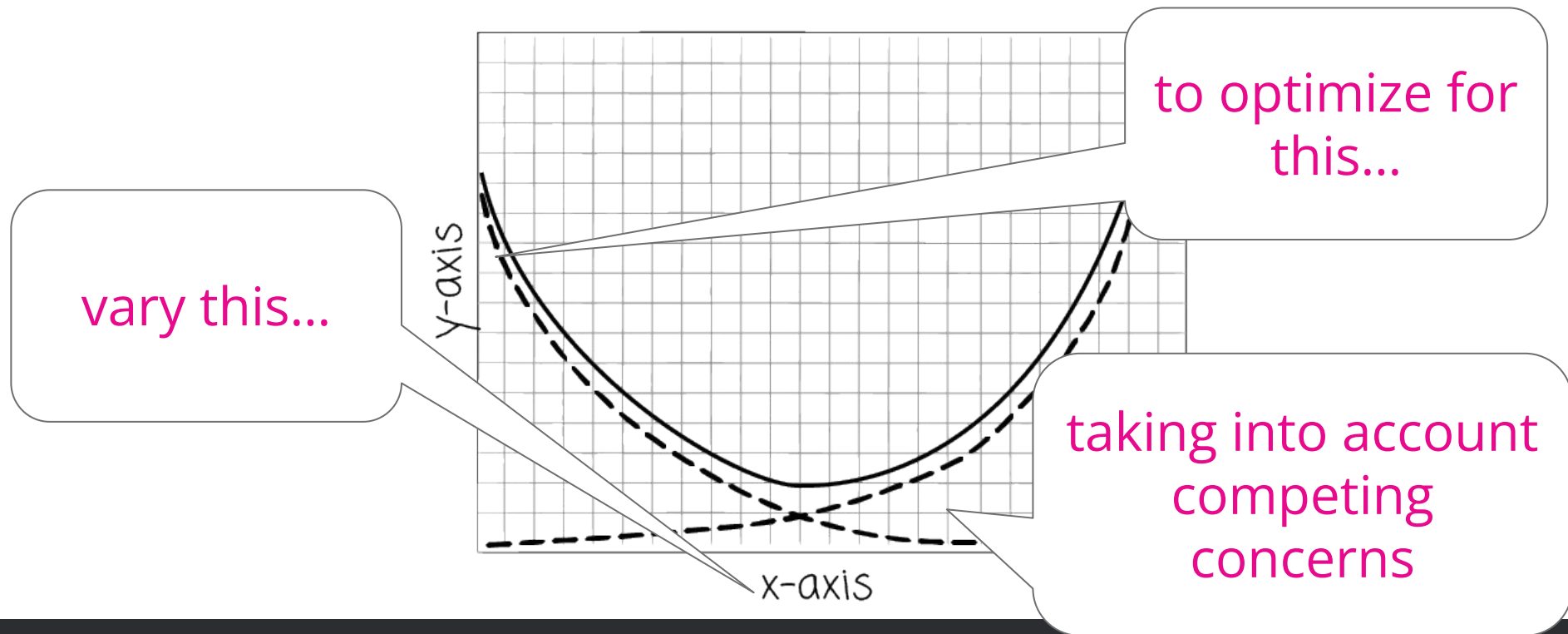
An example from Ford W. Harris's classic 1913 paper, "How Many Parts to Make at Once" originally published in 1913 (reprinted in the Operations Research journal, 1990)

Batch size in software development

U-curve Example - Batch Size Total Cost



Constructing a U-Curve



How to make a U-Curve

1. State the problem.
2. Identify what you are trying to optimize for (cost, time, risk, etc.).
This becomes the Y axis.
3. Identify tradeoff variables. What affects the thing you're trying to optimize for?
4. Describe 2 (or more, but 2 is enough) relationships between the tradeoff variables and Y axis that contribute to the total.
5. Draw the chart.
6. Iterate until done.

Implicit Tradeoffs

when a decision to optimize in one dimension sacrifices in another

economics

What's the
"bottom line"?

VS

risk

What could
wrong & how
bad would it be?

execution

How do we
work? How well?

VS

adaptability

How quickly can
we change
course?

Mistaken Identity

how to avoid confusing one
dimension of context for another

problem

“We’re not shipping anything.”

is it...

capacity

How much can
we do?

or

execution

How do we
work? How well?

?

“...in order to describe a **wicked problem** in sufficient detail, one has to develop an exhaustive inventory of all conceivable solutions ahead of time.”

– Horst W. J. Rittel & Melvin M. Webber
“Dilemmas in a General Theory of Planning”
Policy Sciences 1973

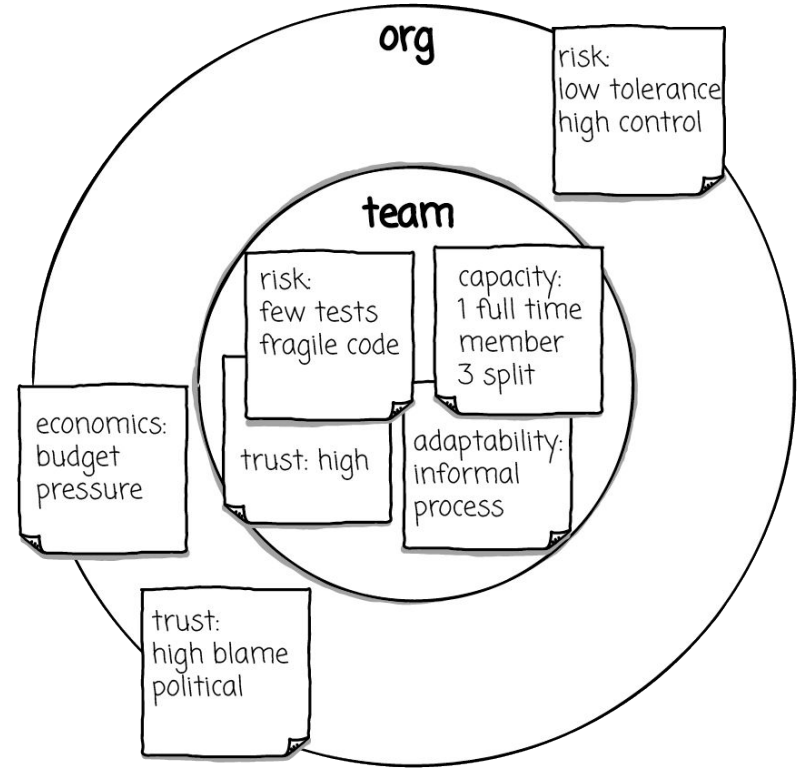
Context Always Changes

what worked over there then might not work here
and now, but what didn't work just might

Same company, different context

Different parts of a company have to optimize for different things, and make different tradeoffs.

That's not (necessarily) a lack of alignment. It can be a legitimate difference in their respective missions and constraints.



time changes everything

Right +  = Wrong

Wrong +  = Right

Want better decisions?



communication about context in
concrete detail across all dimension

capacity

How much can
we do?

risk

What could
wrong & how
bad would it be?

execution

How do we
work? How well?

adaptability

How quickly can
we change
course?

trust

How do people
treat each other?

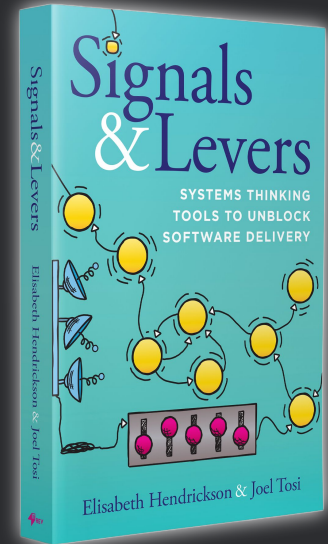
economics

What's the
"bottom line"?

Thank you!

The CREATE Framework: How to Reason about Context

slides &
resources



*coming Sept 2026
from IT Revolution*