Row together
Row in the right direction
Row faster

Jason Yip
@jchyip
jcyip@thoughtworks.com
http://jchyip.blogspot.com

My first job, as part of engineering internship, was at a Canadian defense contractor working as a vehicle platform test engineer, that is, I would create and execute tests with the system installed in various military vehicles.

This was my first real involvement in a large-scale project and I don’t really remember having any strong expectations of how things like this should be run.

Make sure these

Uh... development is "complete" but nothing seems to be actually working?

What are "Hoshin goals"?

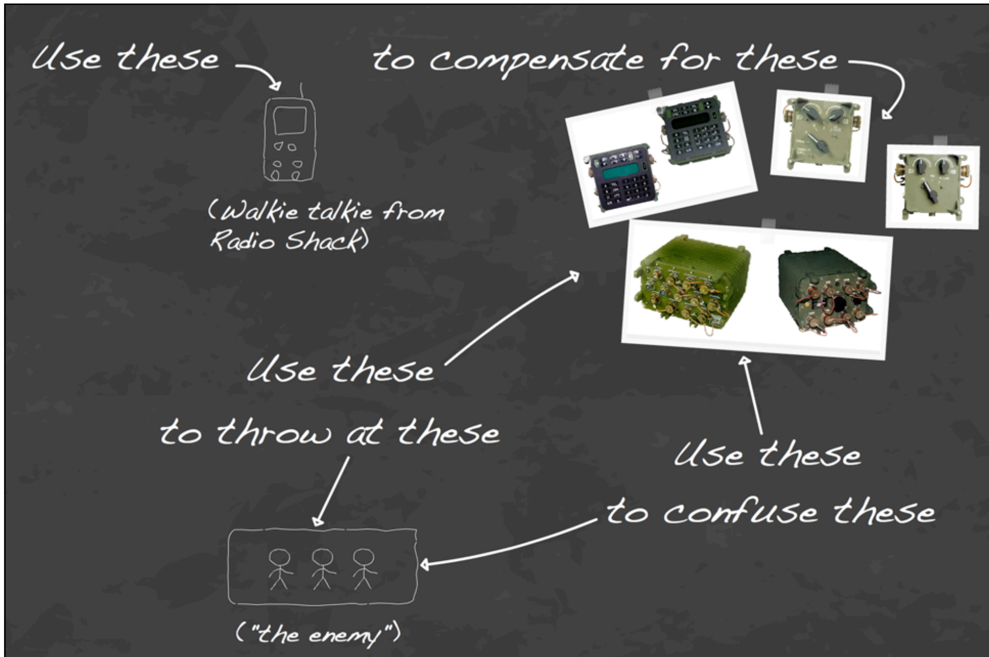
work when installed in these

So this was my first introduction to systemic failure...

Nothing actually worked -> tests were failing but “development is officially over”; first time I encountered Hoshin goals (aka strategic goals) but none of it had any bearing to the reality on the ground

The joke was that we should suggest that soldiers throw the devices at the enemy as they were pretty heavy; or that we could waste their time trying to reverse engineer things that didn’t actually work

So what was going on here? Why didn’t this work?



Use these

(Walkie talkie from Radio Shack)

to compensate for these

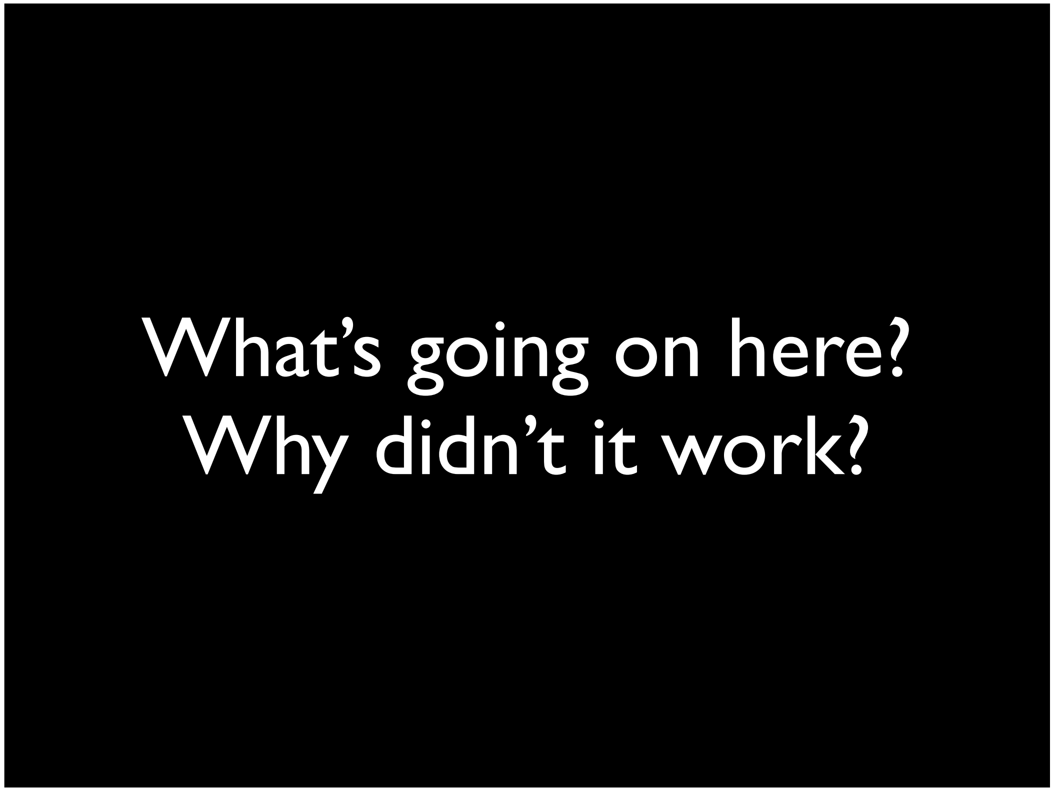
Use these

to throw at these

Use these

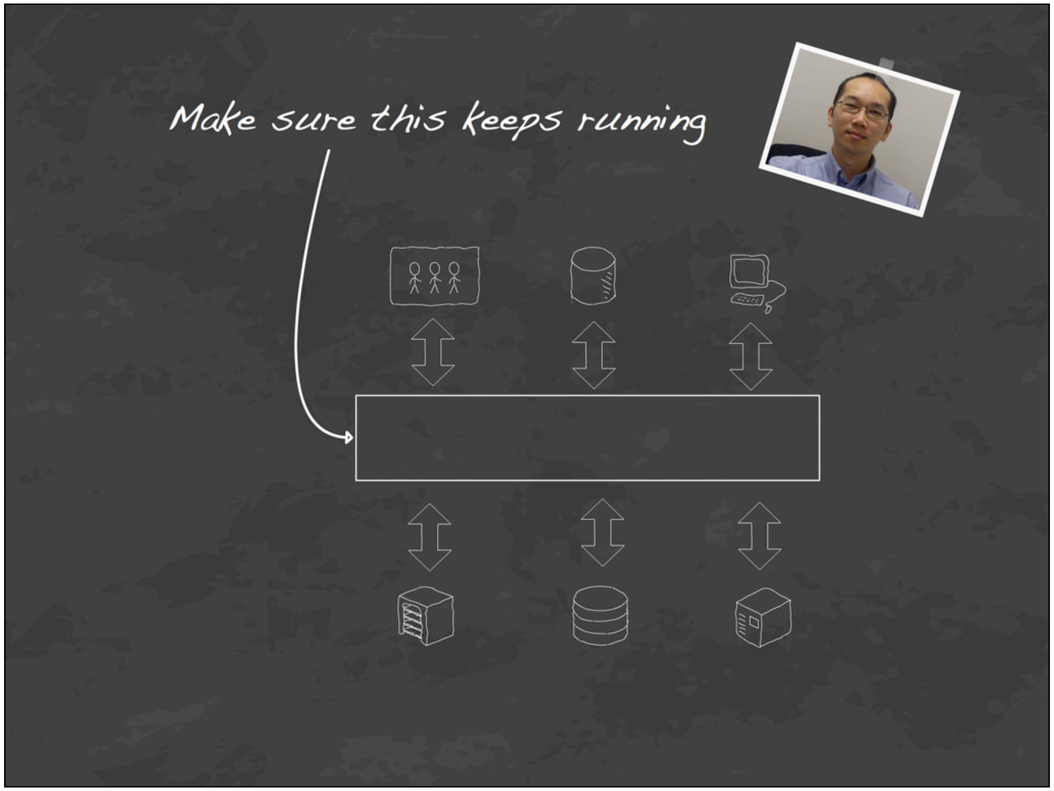
to confuse these

("the enemy")



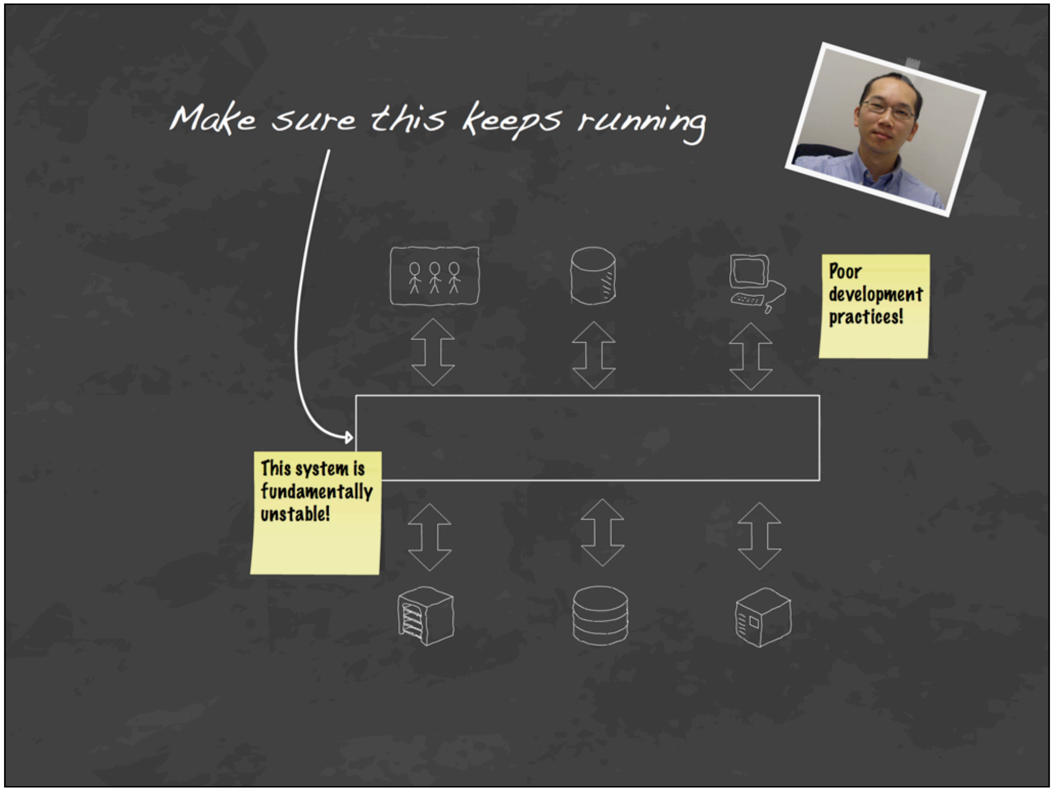
What's going on here?
Why didn't it work?

I remember thinking that the approach taken by the defense doesn't work and would never work for anyone. But why? What is the nature of the failure? Why is it surviving?



Flash foward, now a university graduate, MSc dropout working on an operations team with the responsibility to keep this bit of middleware infrastructure running.

First time doing operations. First time with a pager, eventually even got one of the early Blackberry pagers.



Make sure this keeps running
Poor development practices!
This system is fundamentally unstable!

System constantly failing. We were getting paged at all hours.

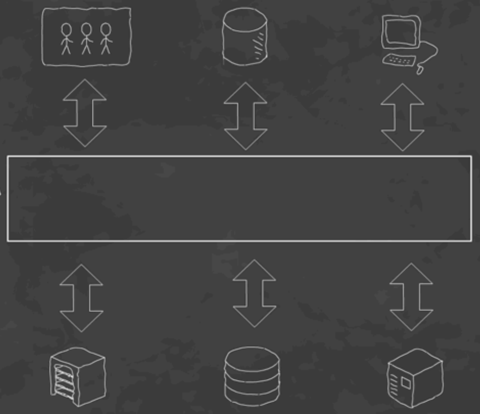
No version control, seat-of-the-pants code-fix-pray development.

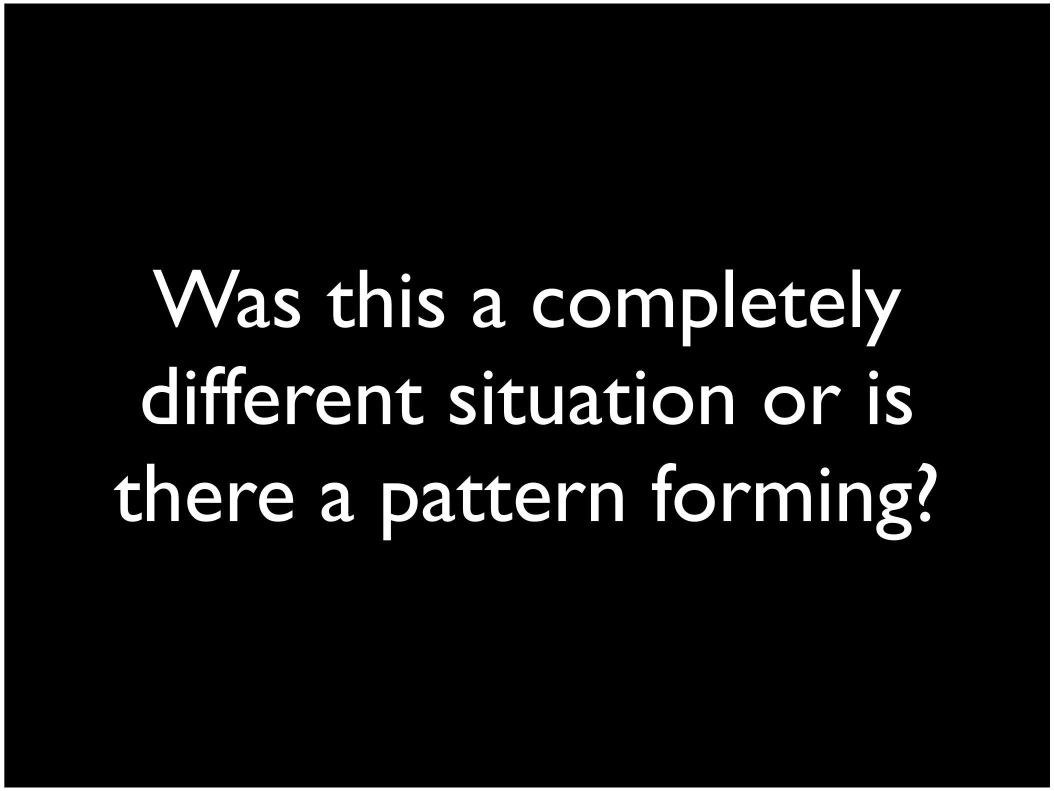
Minimum 2 hour charge mind you so we were billing a lot but it was pretty obvious that there was something fundamentally wrong with both the system as well as the development approach.

Use these

+ (simple e-mail handler)

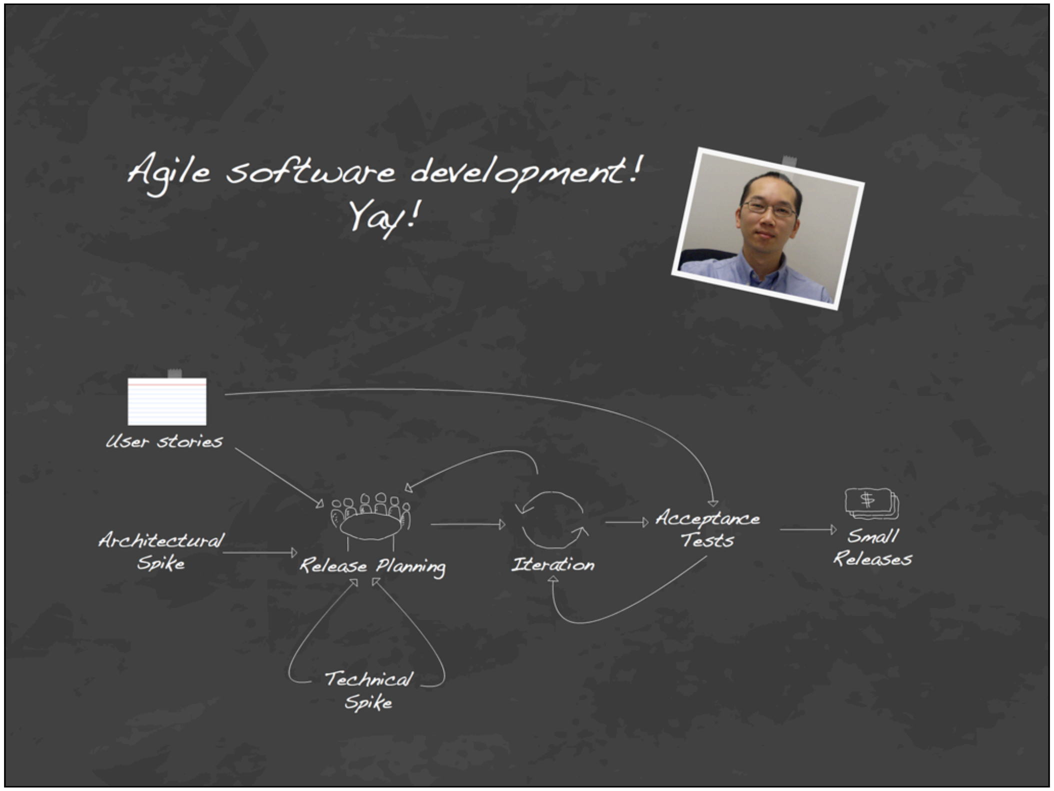
to auto-restart this





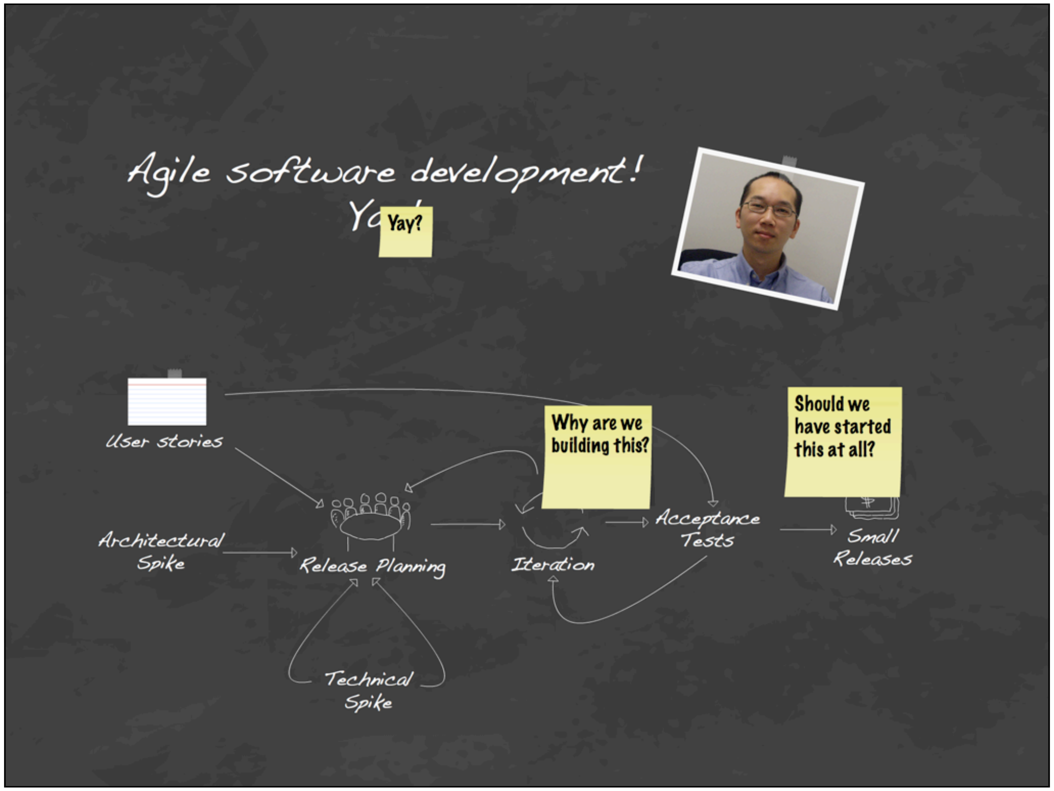
Was this a completely different situation or is there a pattern forming?

This is not something I remember thinking about at the time. Probably due to fatigue from responding to so many pages. Which is itself interesting...



Agile software development!
Yay!

Eventually moved to Agile delivery projects. All problems solved right?


Agile software development!
Yay!
Yay?
User stories
Architectural Spike
Release Planning
Technical Spike
Iteration
Acceptance Tests
Small Releases
Why are we building this?
Should we have started this at all?

Not exactly.

Hope that these

(Customers / Product Owners)

are NOT relying on a model that looks like this

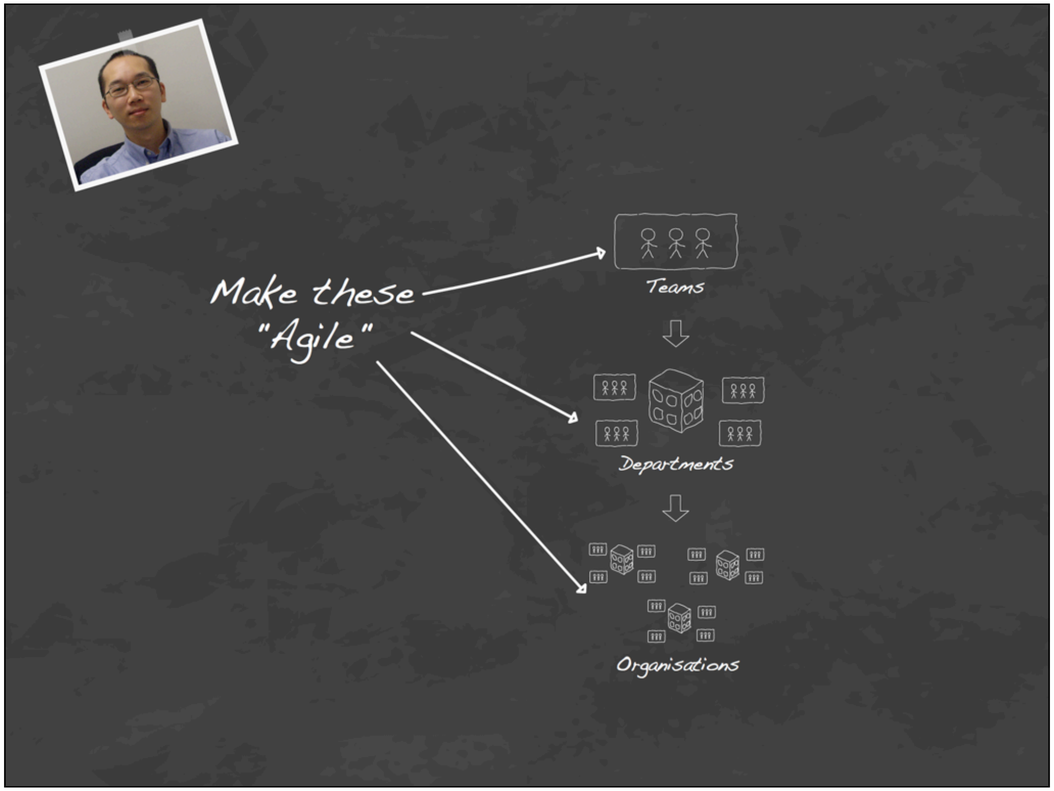
Phase 1: Steal

Phase 2: ?

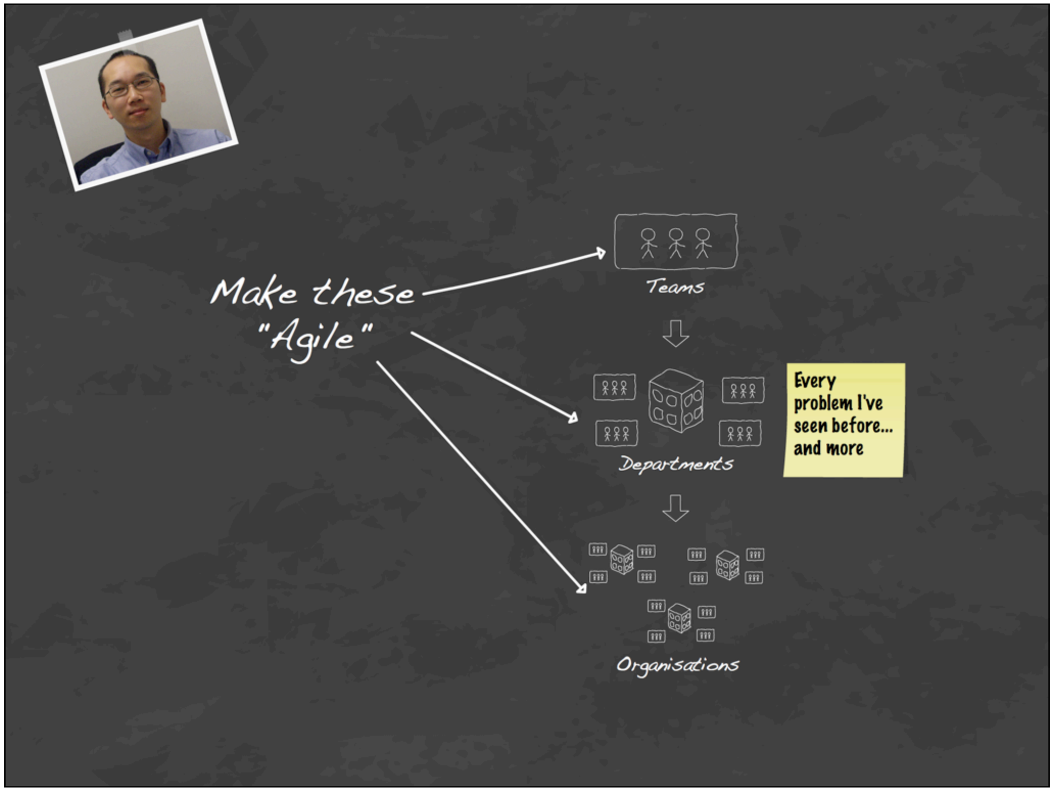
Phase 3:

Hmmm... new problems popped up

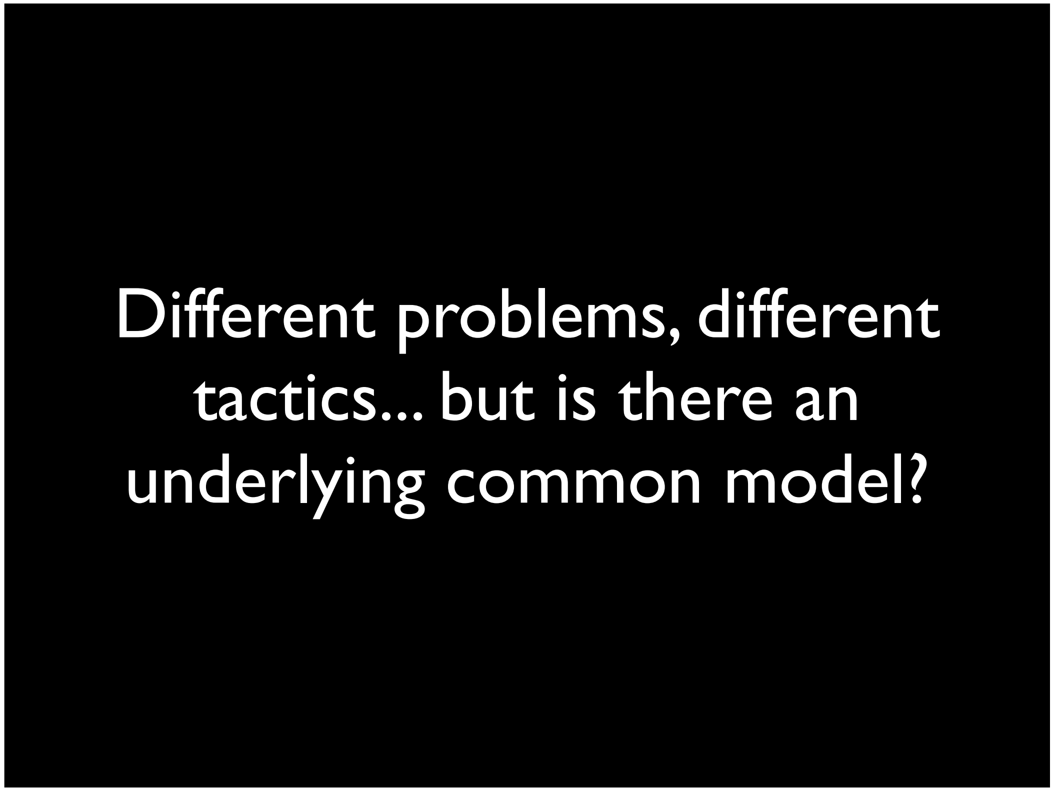
This led to the development of inception techniques described quite well in JR’s The Agile Samurai... but is that addition enough? what’s going on here?



These days I do things more aptly described here and trending toward the organisational end.



... and every problem I’ve seen before (plus a few new ones) are showing up



Different problems, different tactics... but is there an underlying common model?

The experience in all of these situations is different, for sure. If I break down the activities for each day they look quite different. It feels quite different but...

... are all these things disconnected and independent or is there a way to describe them using a common model?

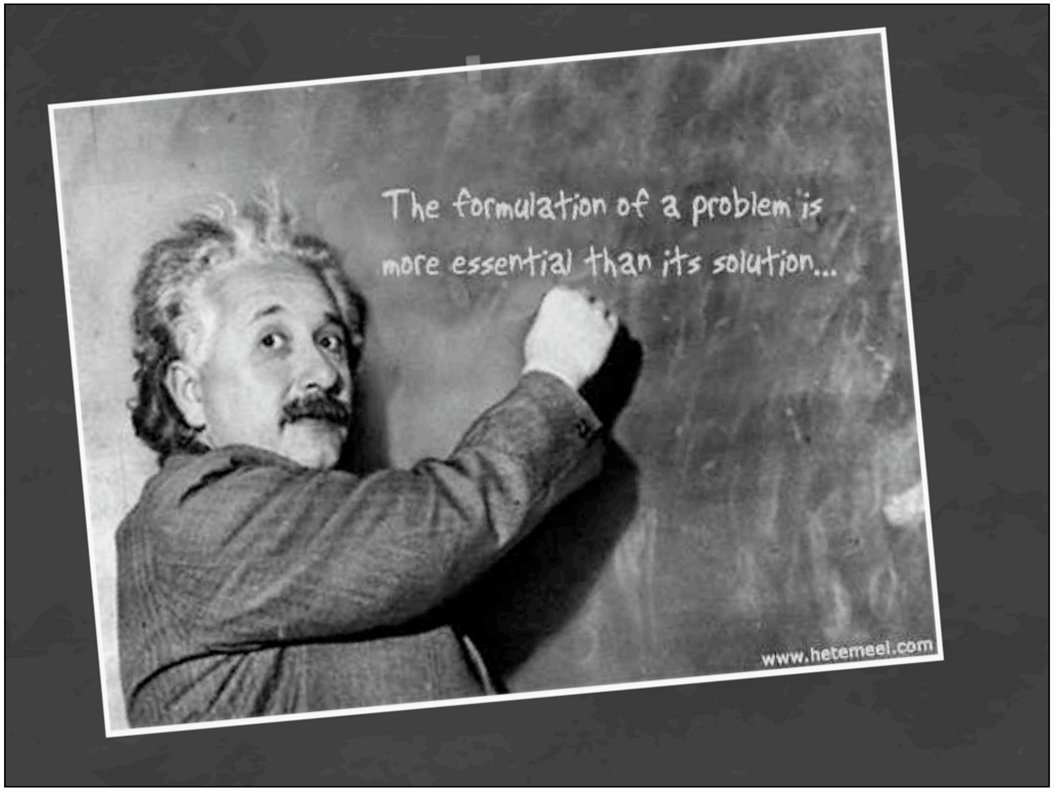
Can we have a Grand Unified Theory? Or at best, will we simply have a recipe book of problem and solution patterns.

Am I a blind man trying to understand an elephant?

Is the strategic / big picture perspective the correct perspective or simply looking at a different part of the elephant?

I think that I've been looking at different parts of the same elephant.

It is very easy to think that the immediate problem I'm facing at the moment is the most important thing but that's just observation bias. I need to remember what it was like when I was on the ground, remember what it was like talking to leaders, remember what it was like trying to get people to work together. Remember all of it.

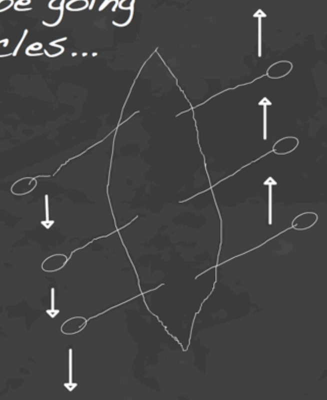


The formulation of a problem is more essential than its solution...

So what I’m going to attempt to do today is to formulate the underlying nature of the problem in order to connect it all together. I’ll hint at what I’m currently thinking about in terms of dealing with it, but the primary goal here is to grasp the situation.

Let’s talk about row boats

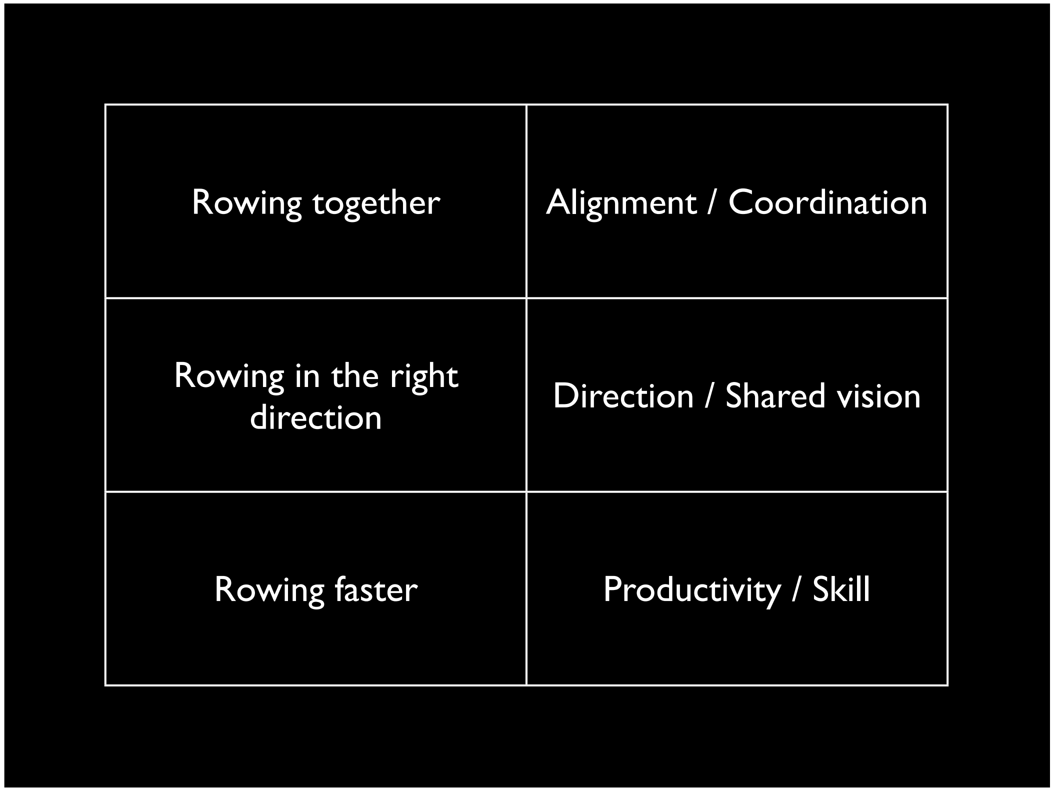
We seem to be going around in circles...



Does anyone know
where we're heading?

Are we there yet?

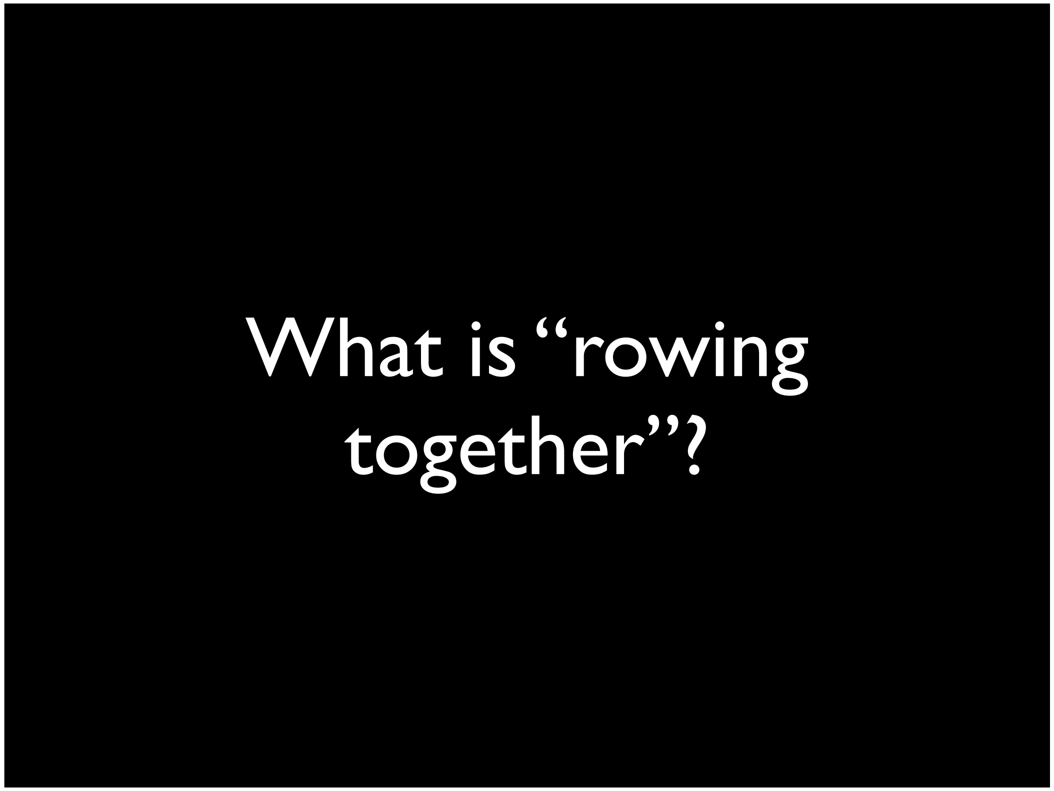
You don’t win boat races by only focusing on one aspect



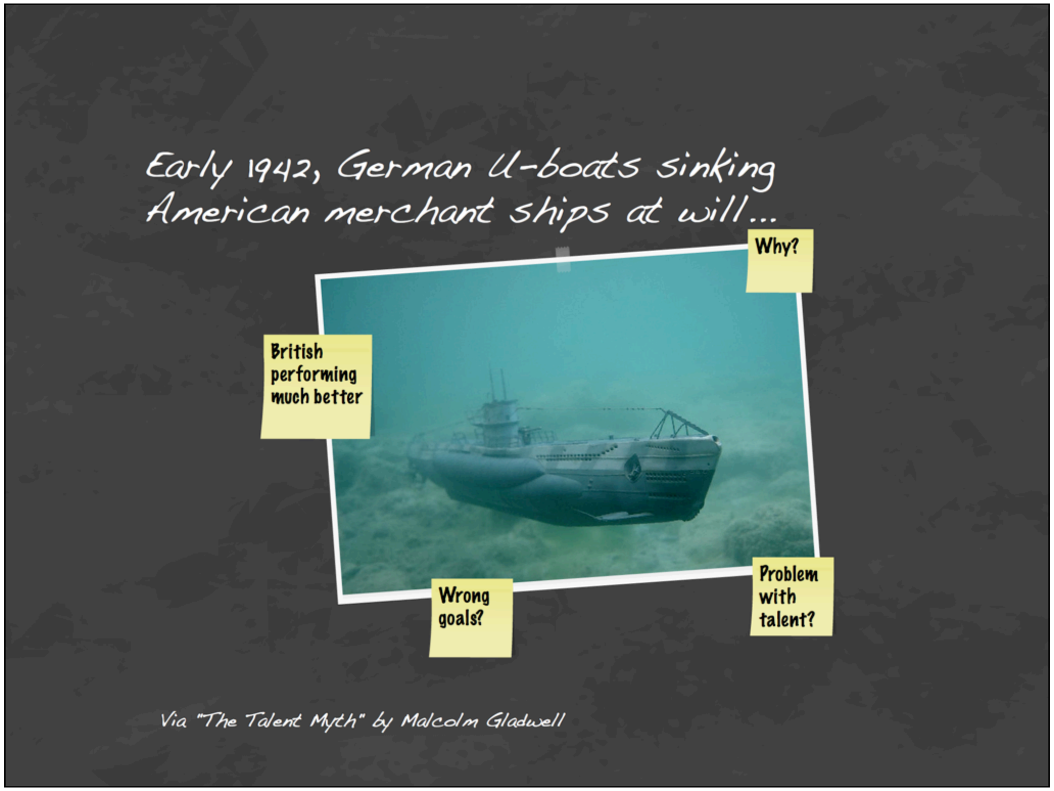
Rowing together	Alignment / Coordination
Rowing in the right direction	Direction / Shared vision
Rowing faster	Productivity / Skill

My claim is that every problem I’ve dealt with generally falls under these three distinct categories. And... given a new situation, I can more effectively and efficiently understand it by exploring each of these categories more explicitly.

Let me expand further.



What is “rowing together”?



Early 1942, German U-boats sinking American merchant ships at will...

Why?

British performing much better

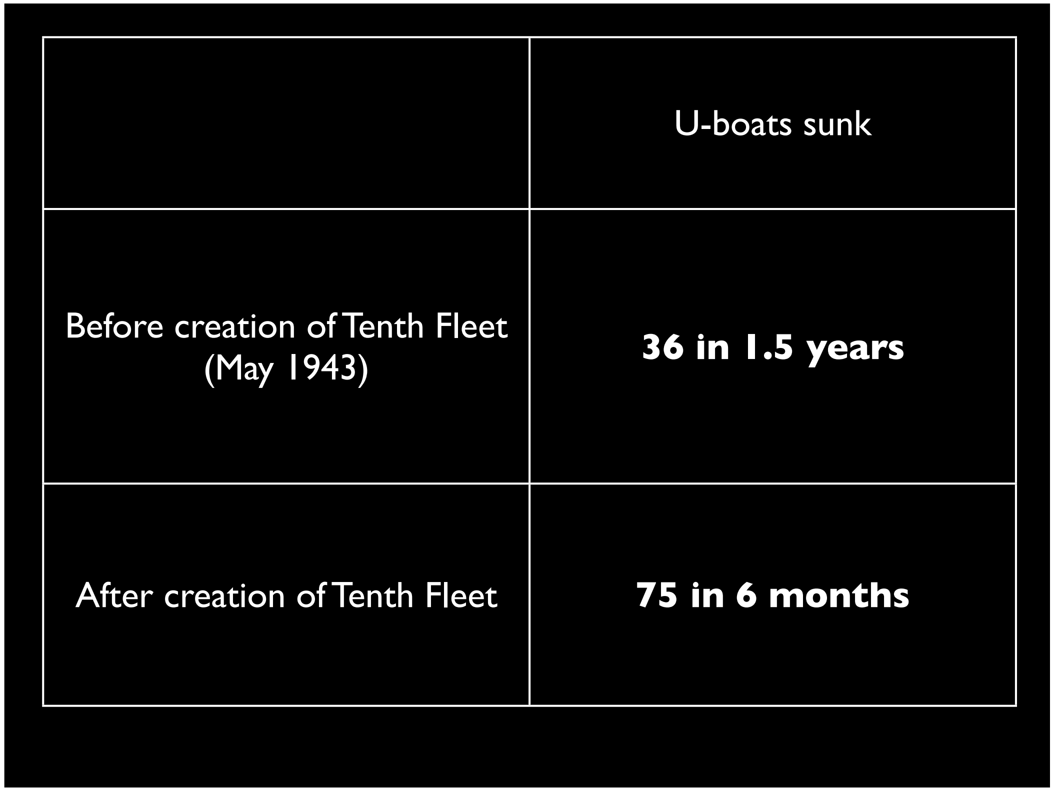
Wrong goals?

Problem with talent?

Via "The Talent Myth" by Malcolm Gladwell

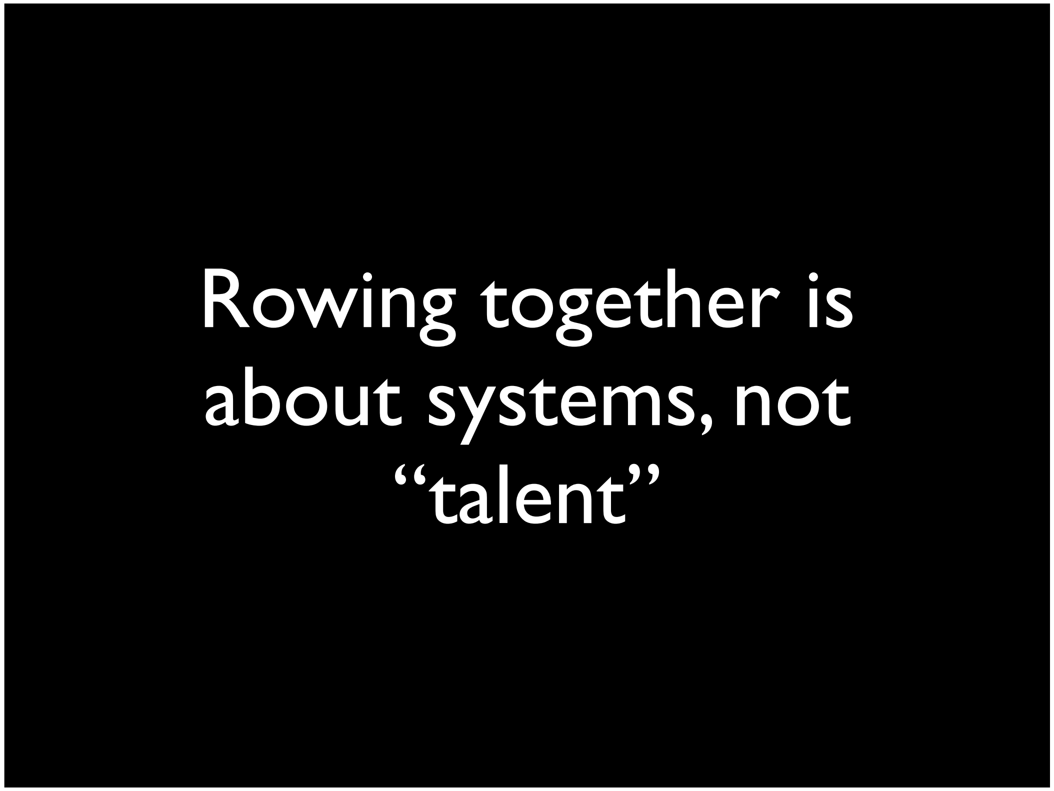
Via Malcolm Gladwell’s “The Talent Myth”

British much more effective due to superior coordination



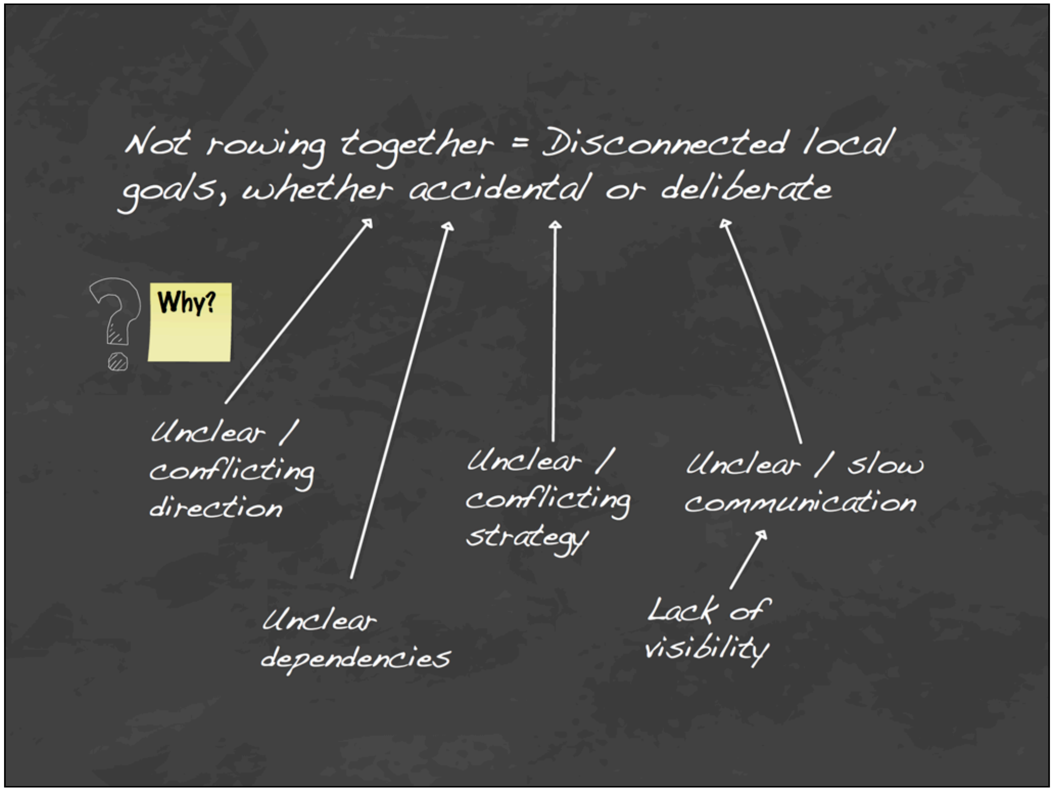
	U-boats sunk
Before creation of Tenth Fleet (May 1943)	**36 in 1.5 years**
After creation of Tenth Fleet	**75 in 6 months**

Not an issue of talent, technology or goals but rather coordination of efforts



Rowing together is about systems, not “talent”

It was not like the Americans didn’t have talented people. But those people could not overcome the system they operated in.



Not rowing together = Disconnected local goals, whether accidental or deliberate

Why?

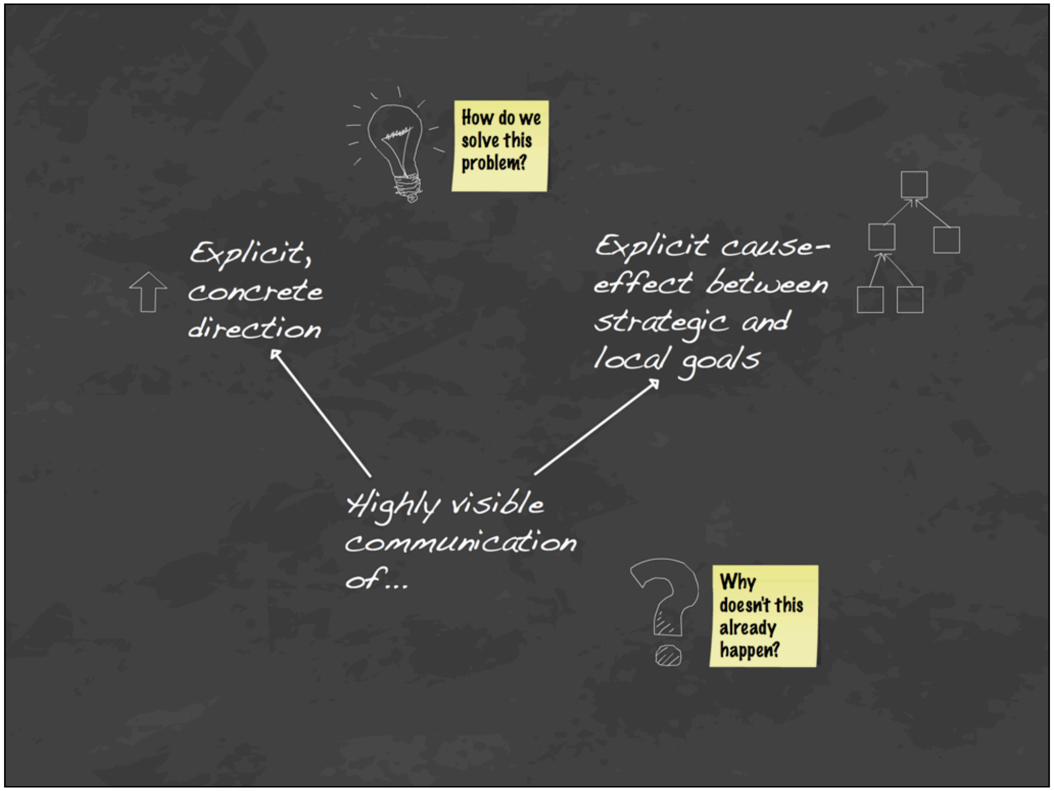
Unclear / conflicting direction

Unclear dependencies

Unclear / conflicting strategy

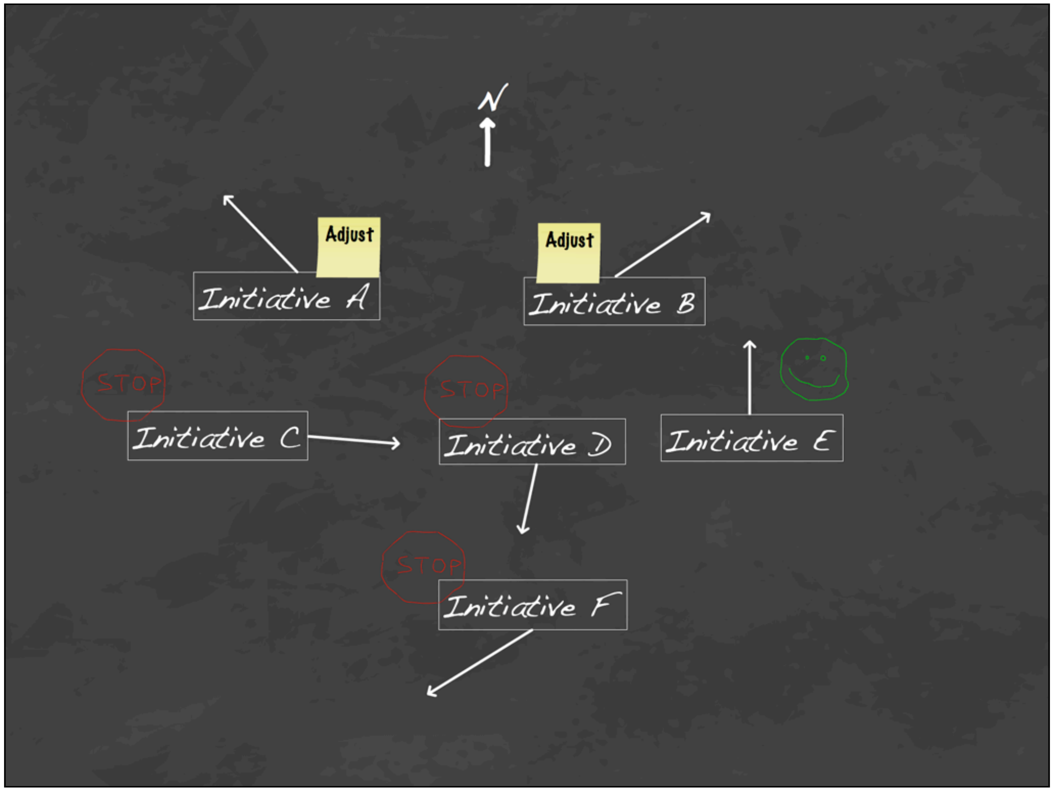
Unclear / slow communication

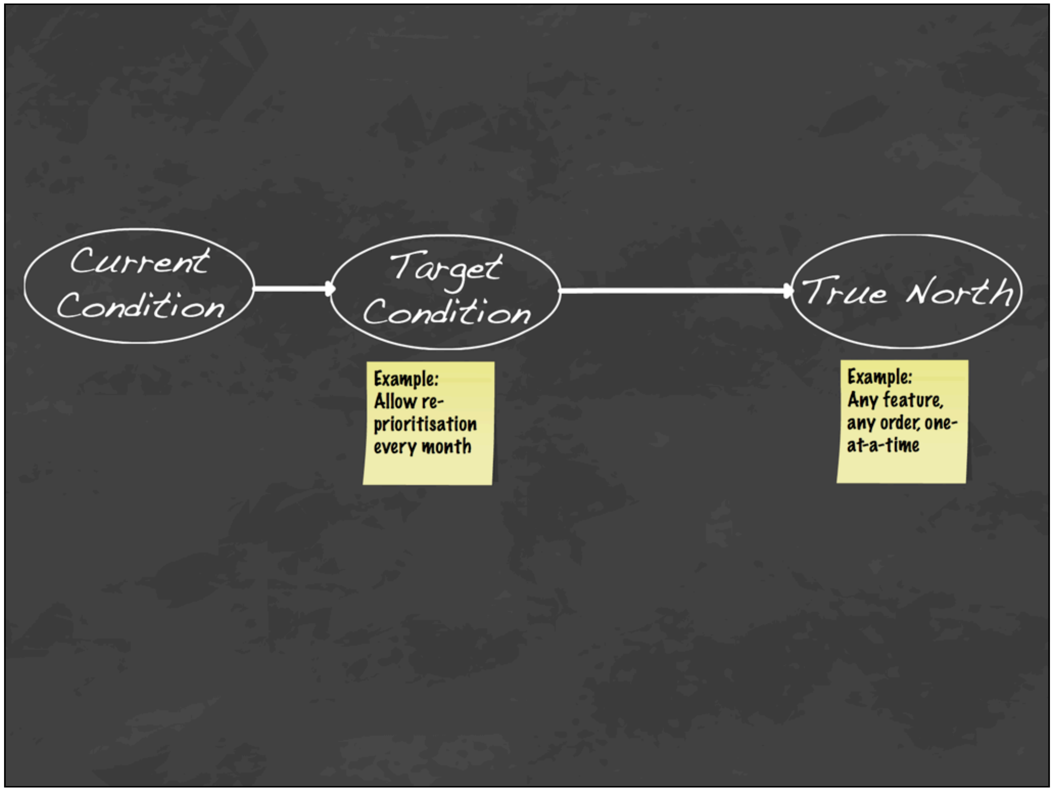
Lack of visibility


How do we solve this problem?
Explicit, concrete direction
Explicit cause-effect between strategic and local goals
Highly visible communication of...
Why doesn't this already happen?

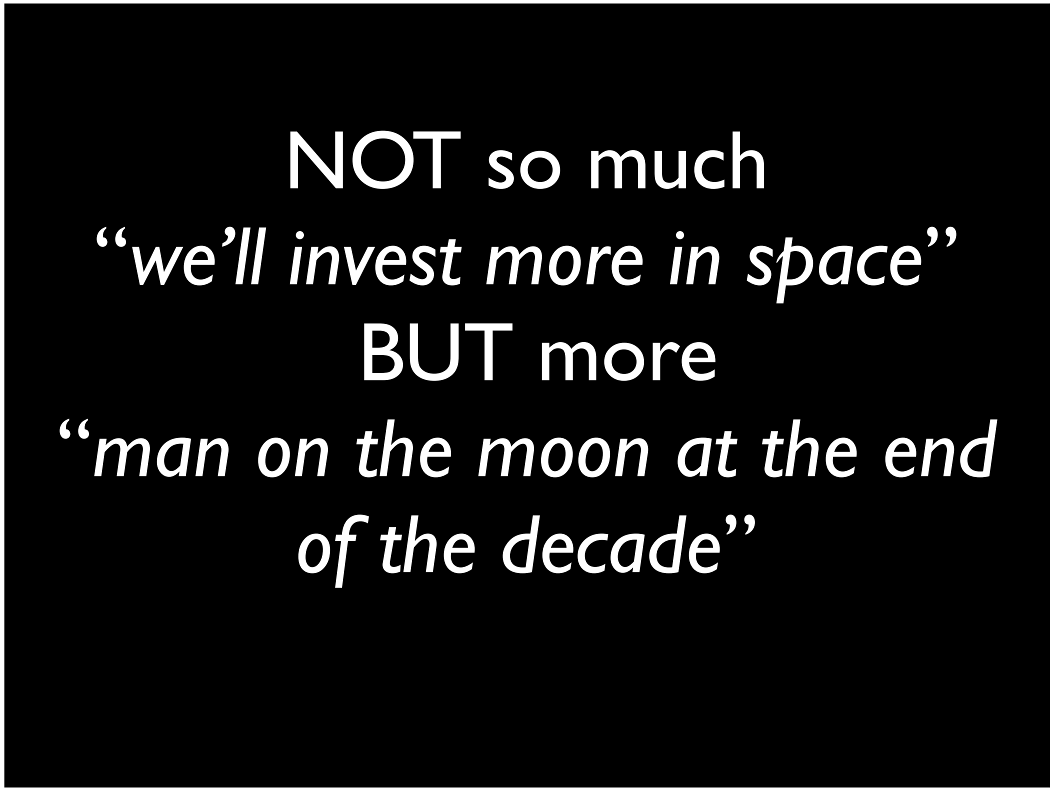
Countermeasure: Explicit, concrete vision + explicit, cause-effect relationship + highly visible communication of former <- Why doesn’t this happen?

What is “rowing in the right direction”?


N
Adjust
Initiative A
Adjust
Initiative B
STOP
Initiative C
STOP
Initiative D
Initiative E
STOP
Initiative F


Current Condition
Target Condition
True North
Example: Allow re-prioritisation every month
Example: Any feature, any order, one-at-a-time

Direction not goal; goals are achievable, direction does not have to be. Mike Rother true north style (Any Feature, Any Order, Released One at a Time)



NOT so much

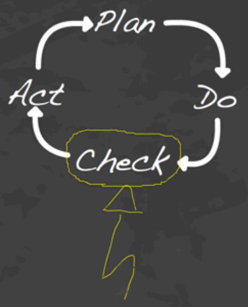
“*we’ll invest more in space*”

BUT more

“*man on the moon at the end of the decade*”

“Simplify enterprise Java”

Not just direction but the "right direction"

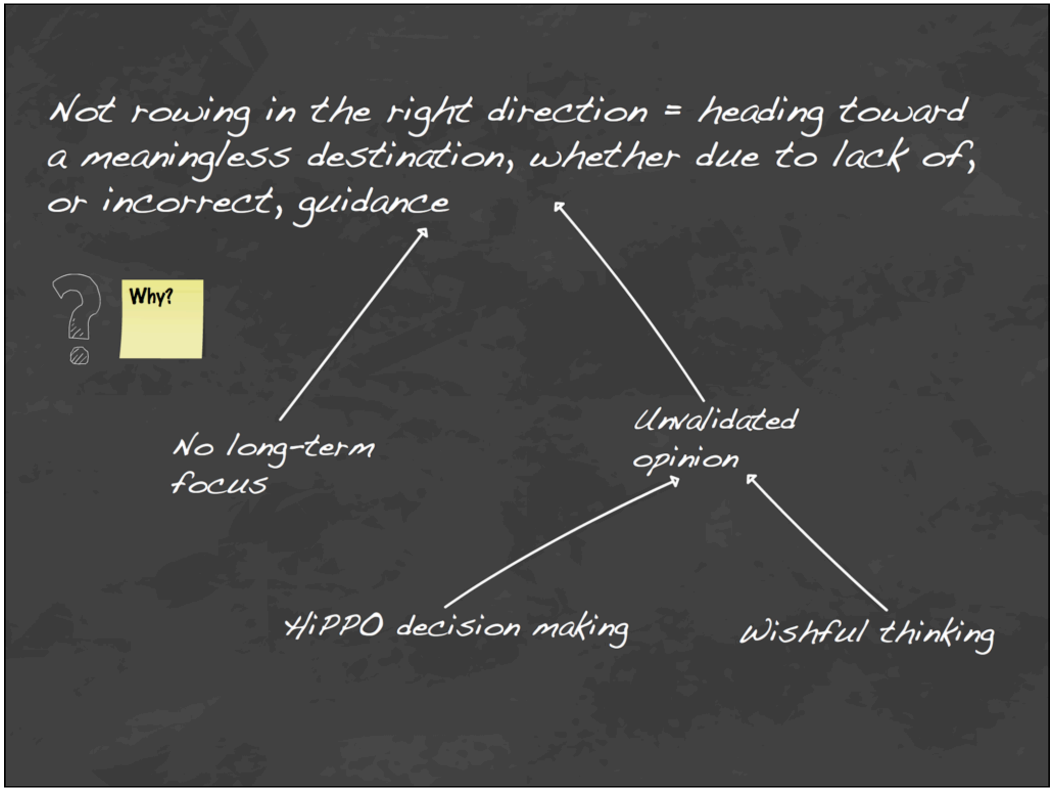


Plan

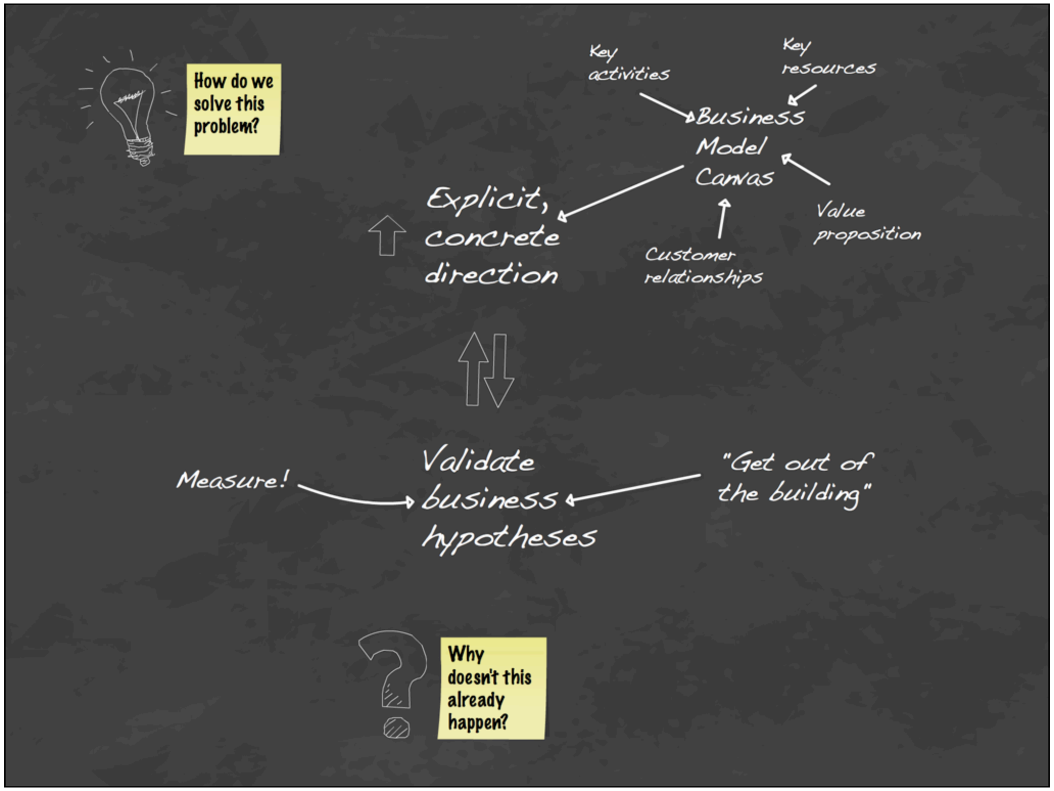
Do

Check

Act

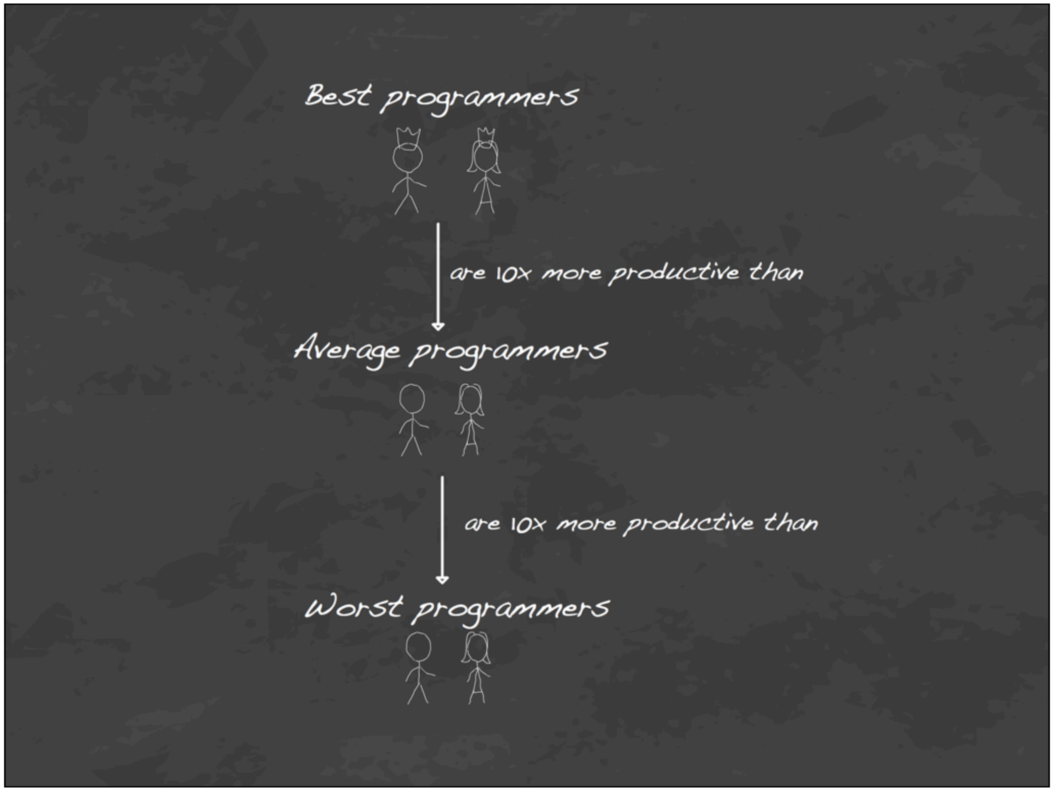

Not rowing in the right direction = heading toward a meaningless destination, whether due to lack of, or incorrect, guidance
?
Why?
No long-term focus
Unvalidated opinion
HiPPO decision making
Wishful thinking

Not rowing in the right direction = heading toward a meaningless destination, whether due to lack of or incorrect guidance
Short-term focus; HiPPO decision making; wishful thinking rather than validating with evidence


How do we solve this problem?
Key activities
Key resources
Business Model Canvas
Value proposition
Customer relationships
Explicit, concrete direction
Measure!
Validate business hypotheses
"Get out of the building"
Why doesn't this already happen?

Make direction explicit (business model canvas); validate business hypotheses

What is “rowing faster”?



Best programmers

are 10x more productive than

Average programmers

are 10x more productive than

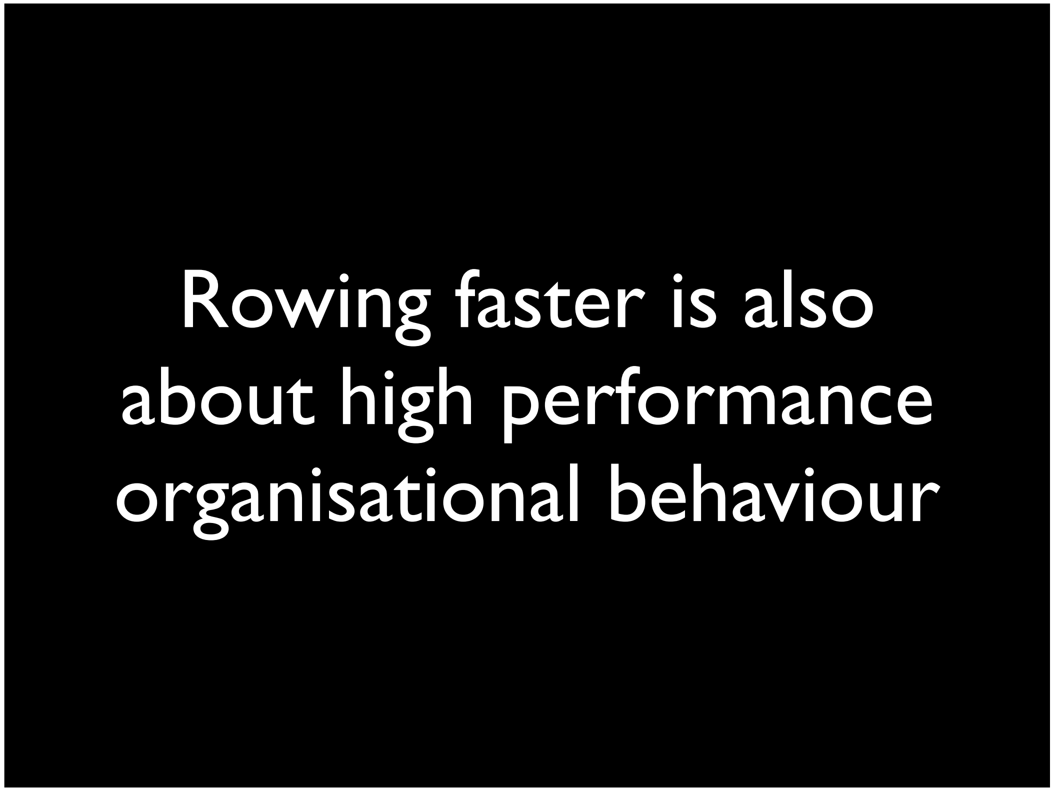
Worst programmers

Craftsmanship

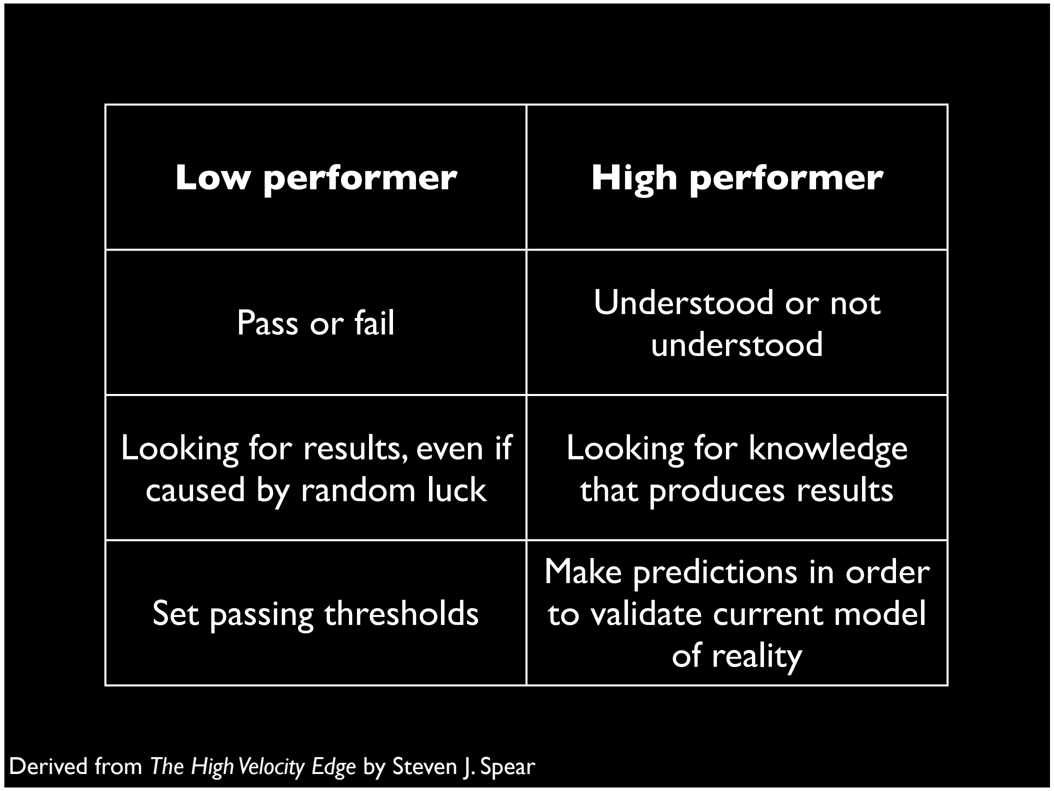
Mastery

Rowing faster is about high performance individual behaviour

10 000 hours



Rowing faster is also
about high performance
organisational behaviour

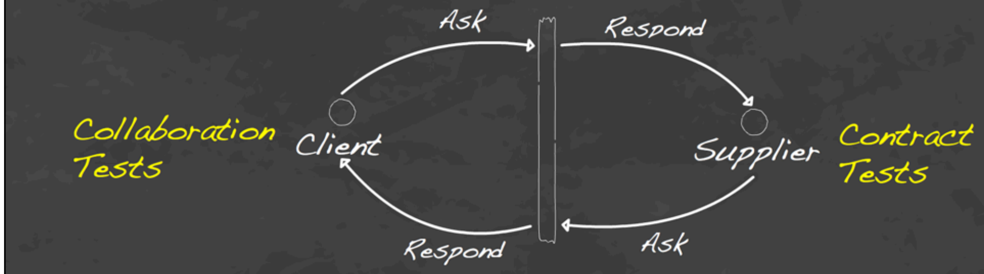


Low performer	High performer
Pass or fail	Understood or not understood
Looking for results, even if caused by random luck	Looking for knowledge that produces results
Set passing thresholds	Make predictions in order to validate current model of reality

Derived from *The High Velocity Edge* by Steven J. Spear

The difference between US Navy Naval Reactors vs NASA. Not pass or fail but understood vs not understood. Not detecting safe or unsafe but more about detecting pockets of ignorance. Make predictions that can be learned from. High performers have a model of reality that they are refining. It’s not random activity and reliance on luck.

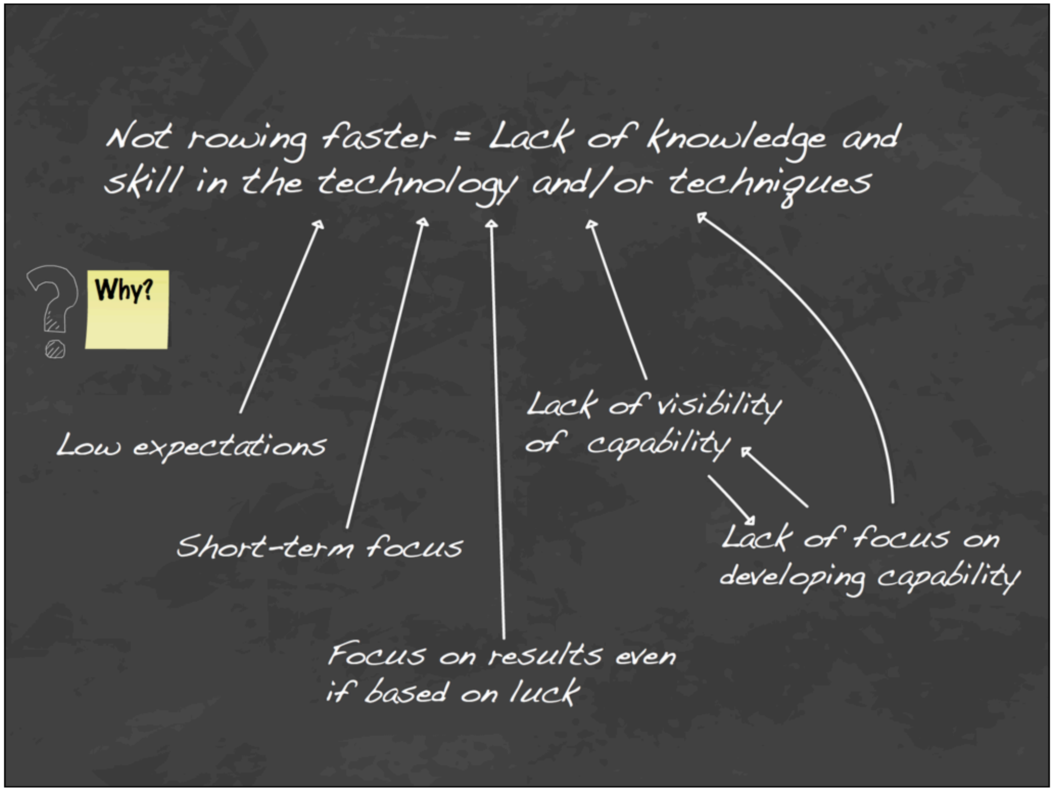
Integration Tests are a SCAM!



Understand and validate how components work!

Don’t be willing to accept ignorance about the contract of a component you rely on

High performance mindset


Not rowing faster = Lack of knowledge and skill in the technology and/or techniques
Why?
Low expectations
Short-term focus
Focus on results even if based on luck
Lack of visibility of capability
Lack of focus on developing capability

Low expectations; lack of focus on developing capability; no explicit visibility of capability; short-term focus; too much focus on results even if only lucky

A question for Takeshi Kawabe

“ What would you suggest should be done with a software development team where there is a significant difference in skill levels, let's say up to 10x difference?”

not "second worst performer"

"**Set the best performers as the standard. Pair people with the masters in a master-apprentice model. Find other suitable jobs for those without aptitude.** Like professional baseball players, you need to practice every day to be a professional. Software development is a team activity and **teams are only as strong as their weakest link.**"

How do we solve this problem?

Set the best performers as the standard

Master-apprentice model

Explicit time and support for practice

Lower the threshold for incidents

Raise the threshold for incident response

Why doesn't this already happen?

How well does this model fit with my experiences?

Rowing together	Alignment / Coordination
Rowing in the right direction	Direction / Shared vision
Rowing faster	Productivity / Skill

What problems existed at the defense contractor?

- **Rowing together?** Despite all the “Hoshin goals”, no. It’s like a portion of the boat just stopped rowing because their part was done.
- **Rowing in the right direction?** Probably
- **Rowing fast enough?** No. Typical problem with defense contracts. Way too slow to respond to events.

What problems existed in the operations support role?

- **Rowing together?** Seemed to be.
- **Rowing in the right direction?** No. Wishful thinking about the effectiveness of middleware and no intention to validate.
- **Rowing fast enough?** No. Did not even have basic knowledge about "rowing" (e.g., using a shared network drive rather than a VCS). Code-and-fix approach.

What problems typically exist on Agile delivery and transformation projects?

- **Rowing together?** Typically have problems with local optimisation for subset teams
- **Rowing in the right direction?** Sometimes unclear on the ground what the direction is. Typically direction is not validated.
- **Rowing fast enough?** Typically not as fast as they could be. Most places do not deliberately focus on developing skills.

Vision is everything; the rest is details

Alignment is everything; the rest will fall in line

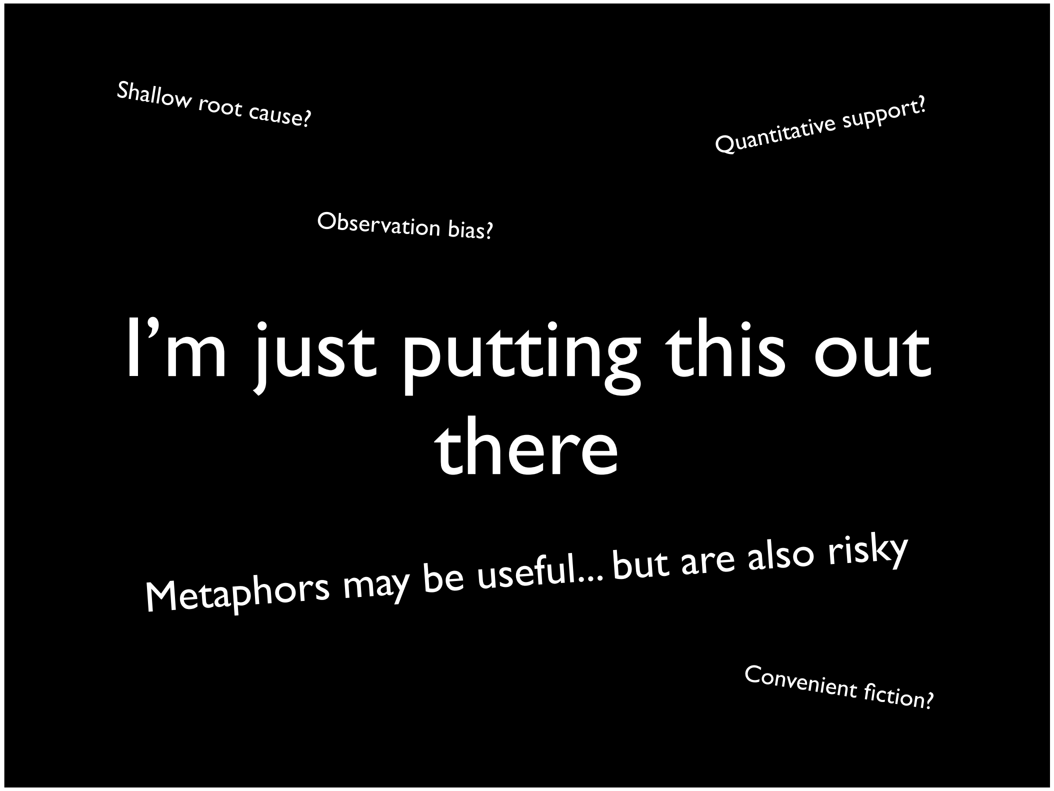
Vision and alignment are easy; the hard part is developing the skill to be hyperproductive

It's all one elephant

Direction does not excuse failing to ensure people can coordinate

Direction and alignment does not excuse lazy work habits

Disciplined craft does not excuse lack of any meaningful purpose



Shallow root cause?

Quantitative support?

Observation bias?

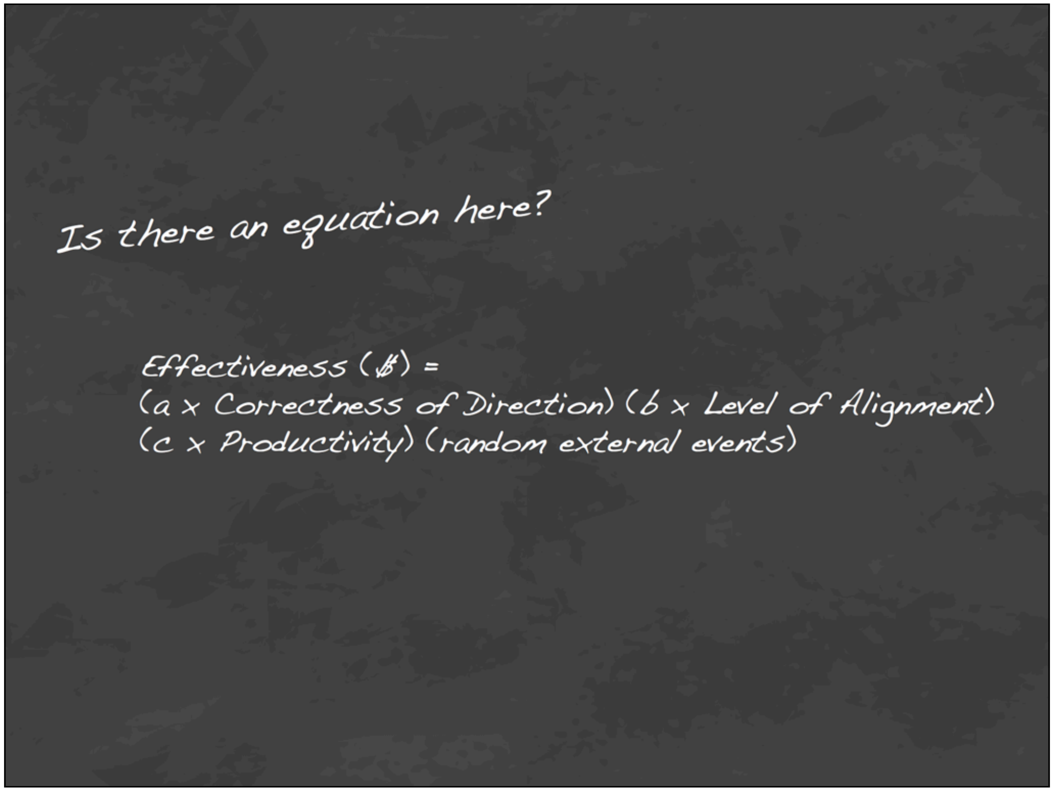
I’m just putting this out there

Metaphors may be useful... but are also risky

Convenient fiction?

This is the start of the story. There is a lot here that I’m not sure about. The root cause analysis is not deep enough. I don’t have enough quantitative support. I’m wary of observation bias. Is this just convenient fiction or is it useful to actually help understand situations?

I think this will be useful to help me more assess situations more effectively and quickly. At least the mnemonic will help. I’d appreciate any questions, comments, and challenges. How can I make this better?



Is there an equation here?

Effectiveness ($) =
(a x Correctness of Direction) (b x Level of Alignment)
(c x Productivity) (random external events)

Estimate mulipliers using things like 64% of features not used, best are 10x better than average, etc.

Photographs

- Underpants:
  - http://www.flickr.com/photos/richardoyork/2273413333/
- Elephant parts:
  - http://www.flickr.com/photos/archetypefotografie/3632454965/
  - http://www.flickr.com/photos/debbiehostetler/2456981718/
  - http://www.flickr.com/photos/michaelhall/155809221/
  - http://www.flickr.com/photos/aussy_greg/255942923/
- U-boat:
  - http://www.flickr.com/photos/mateus27_24-25/2329507830/

Thanks for listening

Jason Yip
jcyip@thoughtworks.com
@jchyip
http://jchyip.blogspot.com