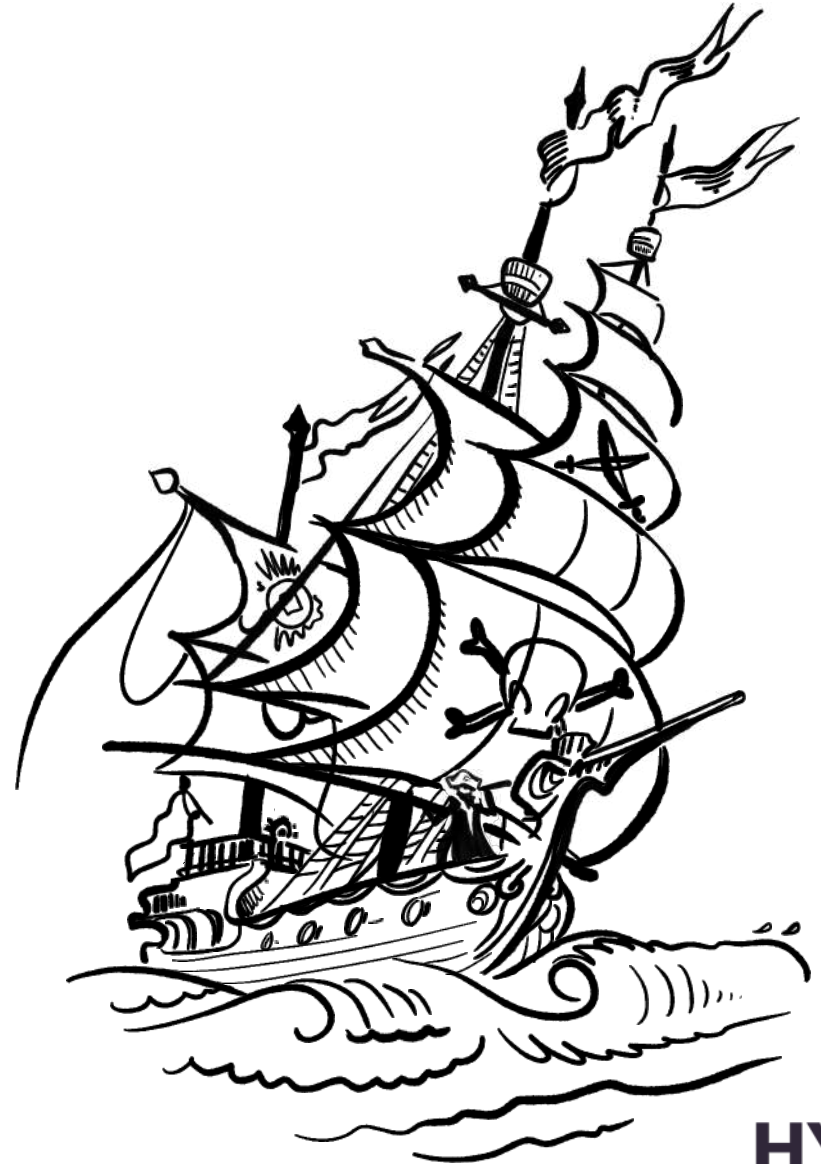


Issue 3 . Nov 22

Tiny Tales from the Frontline

—
A collection of 'pirate'
thoughts inspired
by our work



HYPR

PIRATE SHIPS



ARE YOU ON ONE?

Pirate ships were usually crewed by charismatic people who valued their liberty and despised being led by coercion. They turned to piracy to break the rules and avoid the strict discipline and harsh punishments of life on merchant ships. Because they were not likely to put up with a leader who ruled by rod alone, the captains of those pirate ships were often elected to the position by the crew.

This, of course, produces some very interesting group dynamics with significant implications for decision making. While this is an analogy from history, the scenarios are there in our working lives today. And it made us think about how people deal with them...

The pirate ship captain

-

A senior executive felt his team wasn't telling him the whole truth about things. He asked an outsider to explore on his behalf what was found to validate his thinking. But he wasn't prepared to take any action because he knew certain people in the team were 'keys' – influential, knowledgeable people who were in control. Given that he now knew enough about the situation and the ship was still broadly sailing in the right direction, he decided to keep the 'keys' happy.

A leader's life often entails accepting some insecurity, exercising discretion and occasionally letting something break so that the crew learn how to fix it (if the ends are worth it). A pirate ship captain for sure. But what if the ship was heading towards the rocks?...





Avoiding the rocks

-

Pirate crews were, by their nature, 'fast and loose'. They craved the next adventure, the next bounty and getting there quickly was all that mattered. The Captain was always wary of balancing that against the perils of shortcuts in uncharted waters.

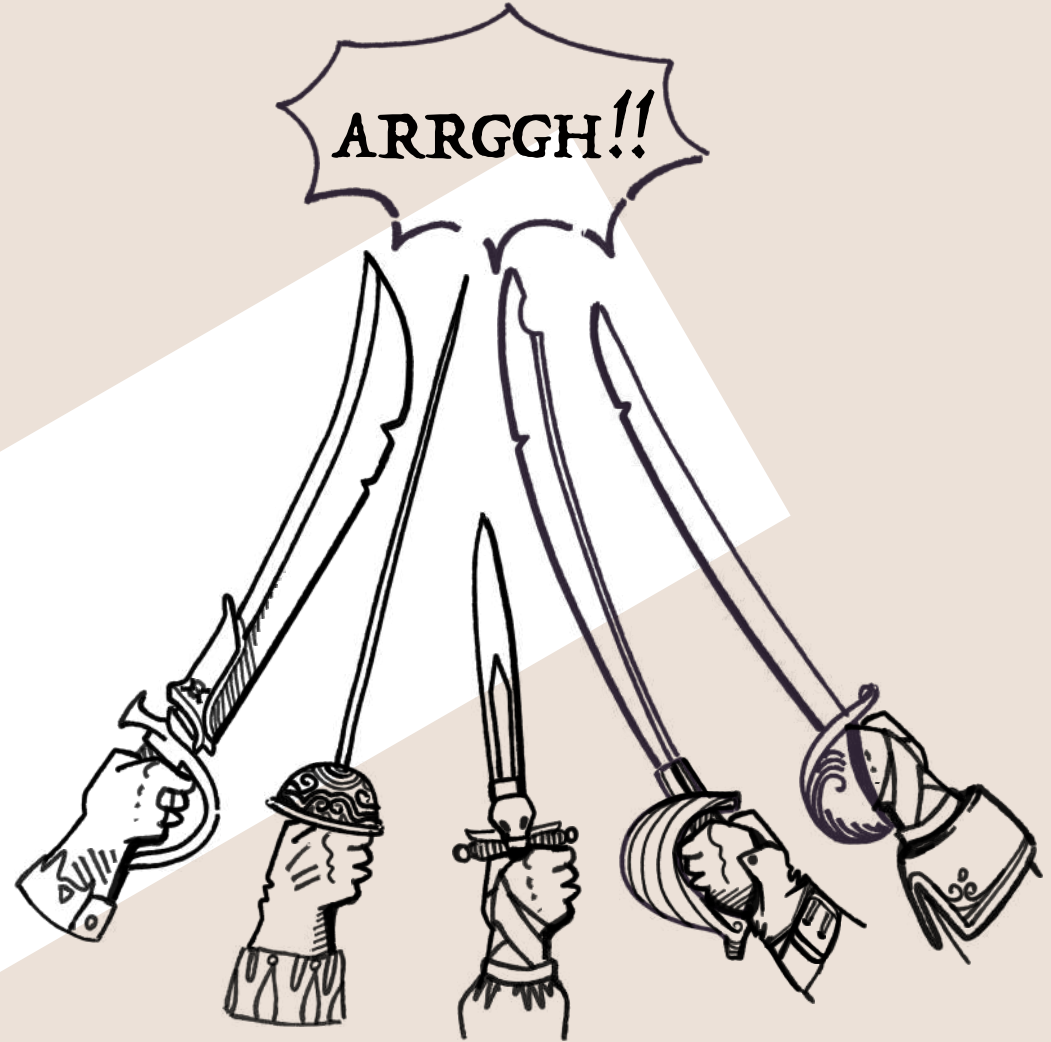
Senior execs usually know where the rocks might be. But teams often keep their heads down, below the waterline, focusing on the things *they* think matter. So the exec needs to help them look up, remind them of the bigger 'North Star' and help them reframe their view of progress. Navigating is never in a straight line.

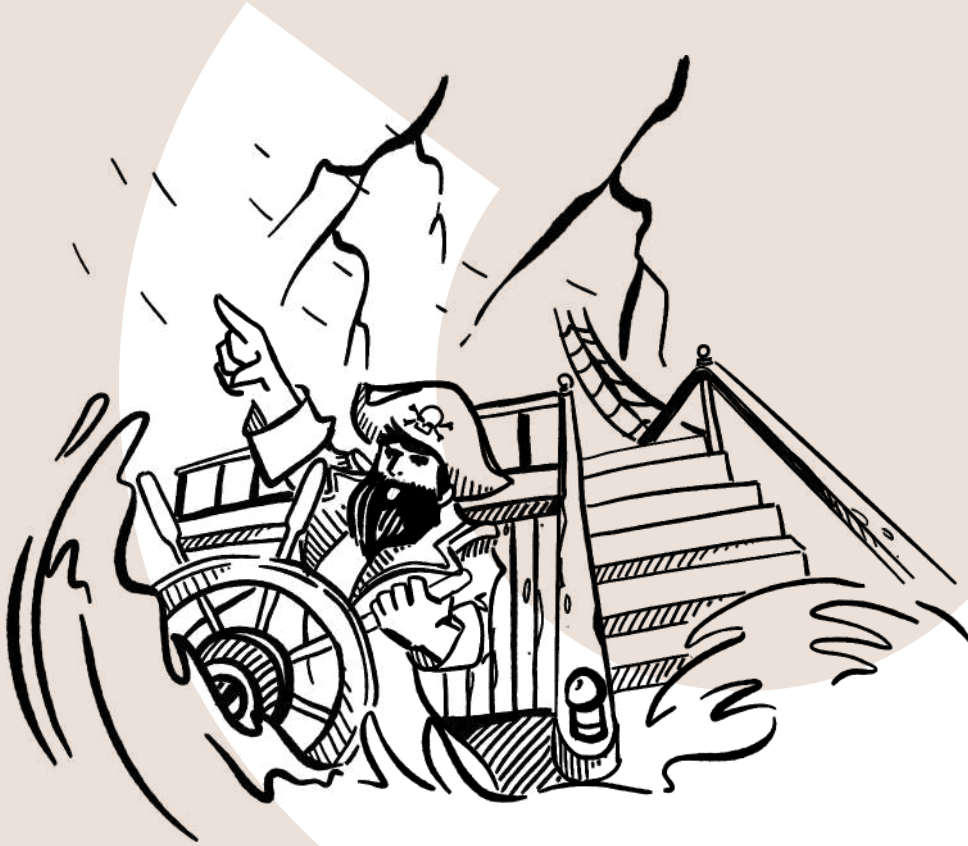
But how might you avoid the situation in the first place?...

Care for your crew!

-

Most people responsible for leading teams land in them. They don't get to pick and choose the make up. So they might just find themselves leading a pirate crew! At the start of any new relationship, the minimum expected from everyone is mutual respect (even from pirates). Keeping and nurturing that respect is the aim. We're always joining new client teams and the one pattern we see time again is that leaders who care about people, who keep intimately interested in their team's work and the milestones they achieve, win through. If you're a crew member who's ever lost a leader to another ship and you wanted to jump there too, you know what we're talking about.





Time for mutiny? Time for empathy

-

Senior execs do a lot of invisible work that allows teams to get on with the delivery of value. Execs also have a lot of invisible things to deal with, such as politics, others' poor decisions or keeping the 'keys' happy. The captain might feel like they are balancing on a wobbly stack of crates behind the ship's wheel and steering is tough! Because of this, there will be times when bringing problems to the exec may cause other problems or extra stress. If you need to share a problem, be sure to show up with both the problem and a potential solution – it shows you're trying to help rather than acting the pirate and throwing the Captain off-balance. It shows empathy to an exec who is, of course, human too.

Opportunities
Dependencies



*President Eisenhower put it perfectly:
“Plans are of no particular value, but
planning is indispensable”. The process of
planning helps everyone explore and
digest risks, opportunities, dependencies,
resources, directions and goals and so
much more; it allows us to develop insights
into ‘why’ and ‘how’ and it’s crucial in
building relationships, trust and
alignment.*

*So if it’s the process which counts, just
make sure it actually does count. Unlike...*

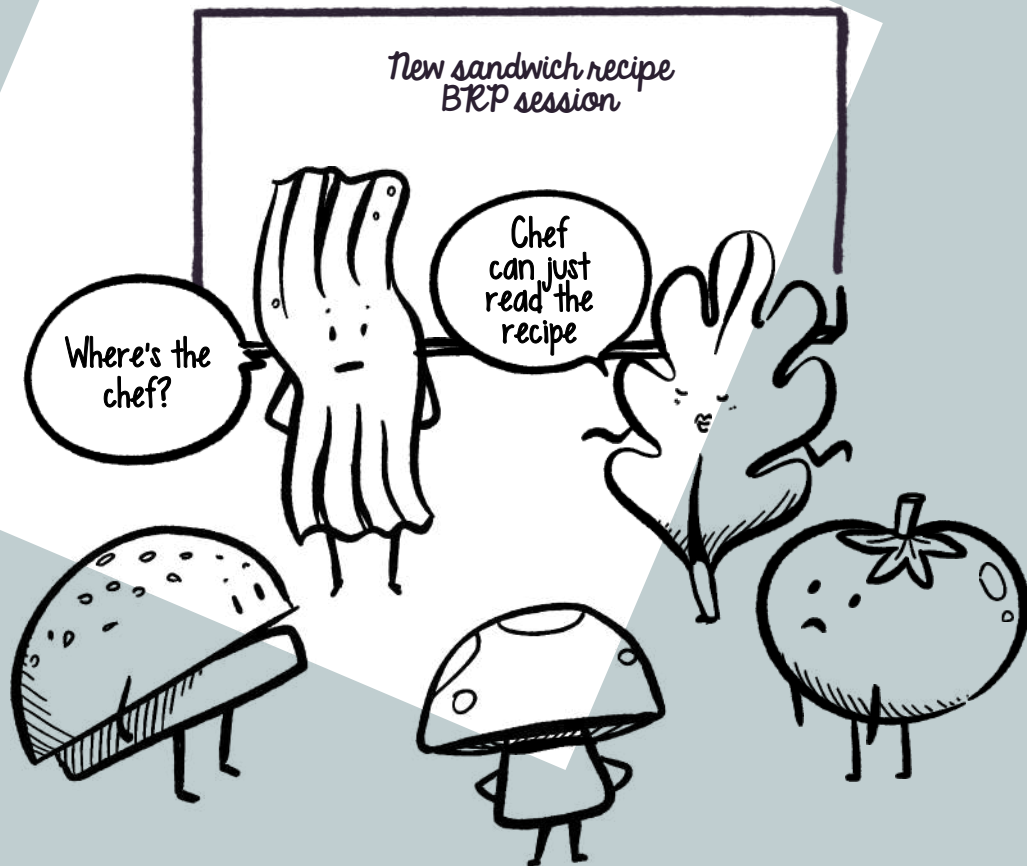
Big room, missing something

-

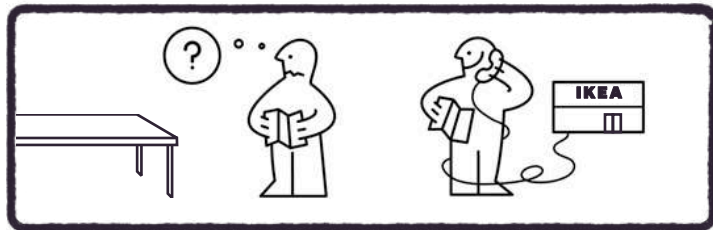
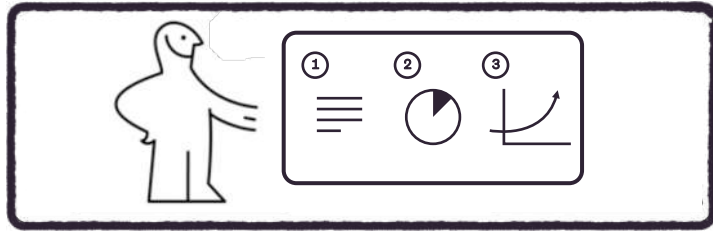
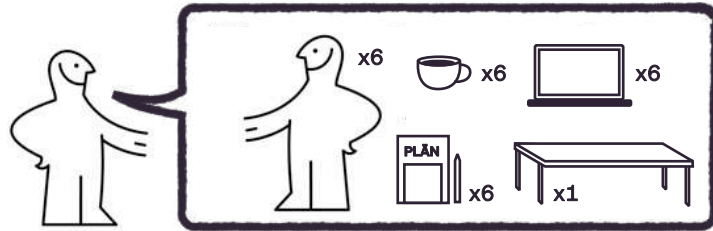
Q. When is a ‘big room planning session’ not a big room planning session?

A. When the people who will be doing the work aren’t even in the room!

We heard on the grapevine of a BRP session that didn’t include the vendor developers “because they just needed to pull the work down”. A simple rule – in BRP, don’t make commitments on behalf of others. Bring them into the room!



PLÄN



Sorted. In advance

-

Q. When is any planning session not a planning session?

A. When someone walks into the room with ‘the plan’.

This happens regularly – and mostly because someone has good intentions and good preparation. They need to think, however, that all the wonderful information they used to create their plan is precisely the information that others need in order to create a better one. Together. If you’re the fount of knowledge, think of yourself as a sharer and enabler in planning. You will empower your team.

—

A. Because plans are, by definition, a model of the future.

need Agile to tell you that!).



We cross the river by feeling
the stones under our feet



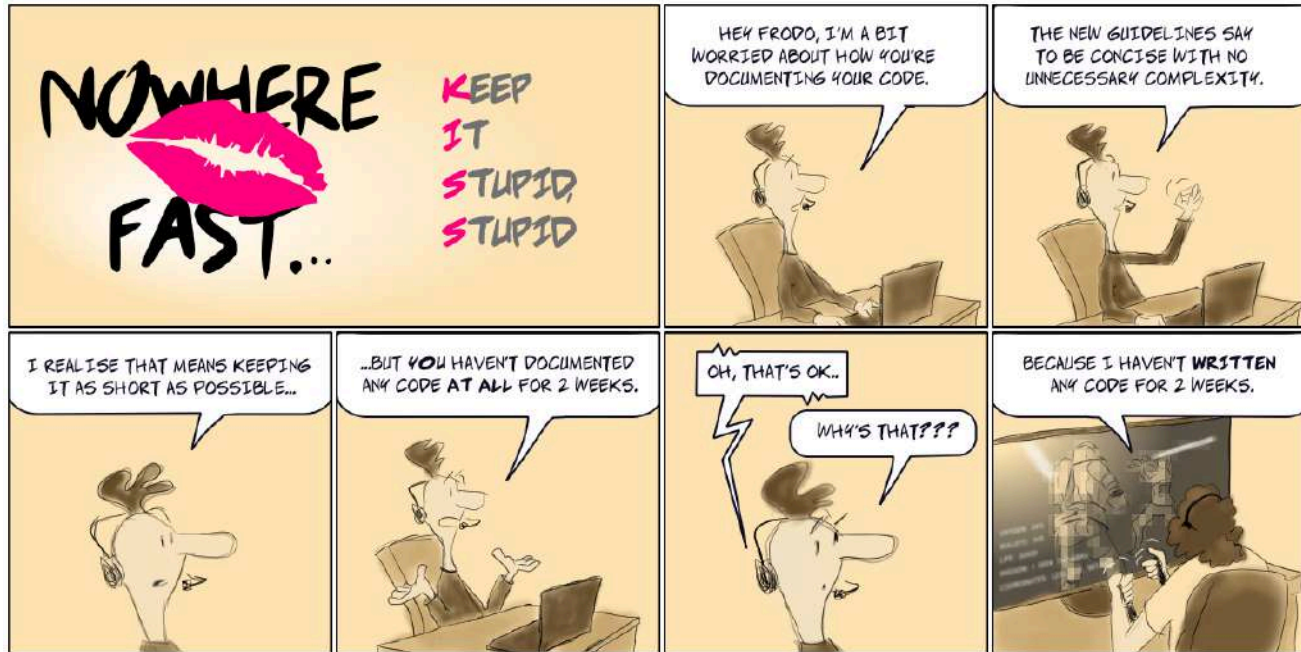
The only way to know is to do

-

There is a saying that 'we cross the river by feeling the stones under our feet'. We can see the other side of the river and we plan to cross it. But we cannot see where our next steps are going. We must explore our way by testing each step, by experiment. When the river is calm and low, we can see where our next steps are going to be for a set distance ahead. We still cannot see every step of the way to the other side of the river. When the river is moving fast and high, it is opaque with mud and we can only probe and feel. And there is heightened danger of being knocked over and swept away. We must test each stone before we put our weight on it to make sure it won't roll.

The idea of simple, predictable linear systems no longer works. 'Define Once, Execute Perfectly' is a fallacy. The future is unknown. The only way to know is to do.

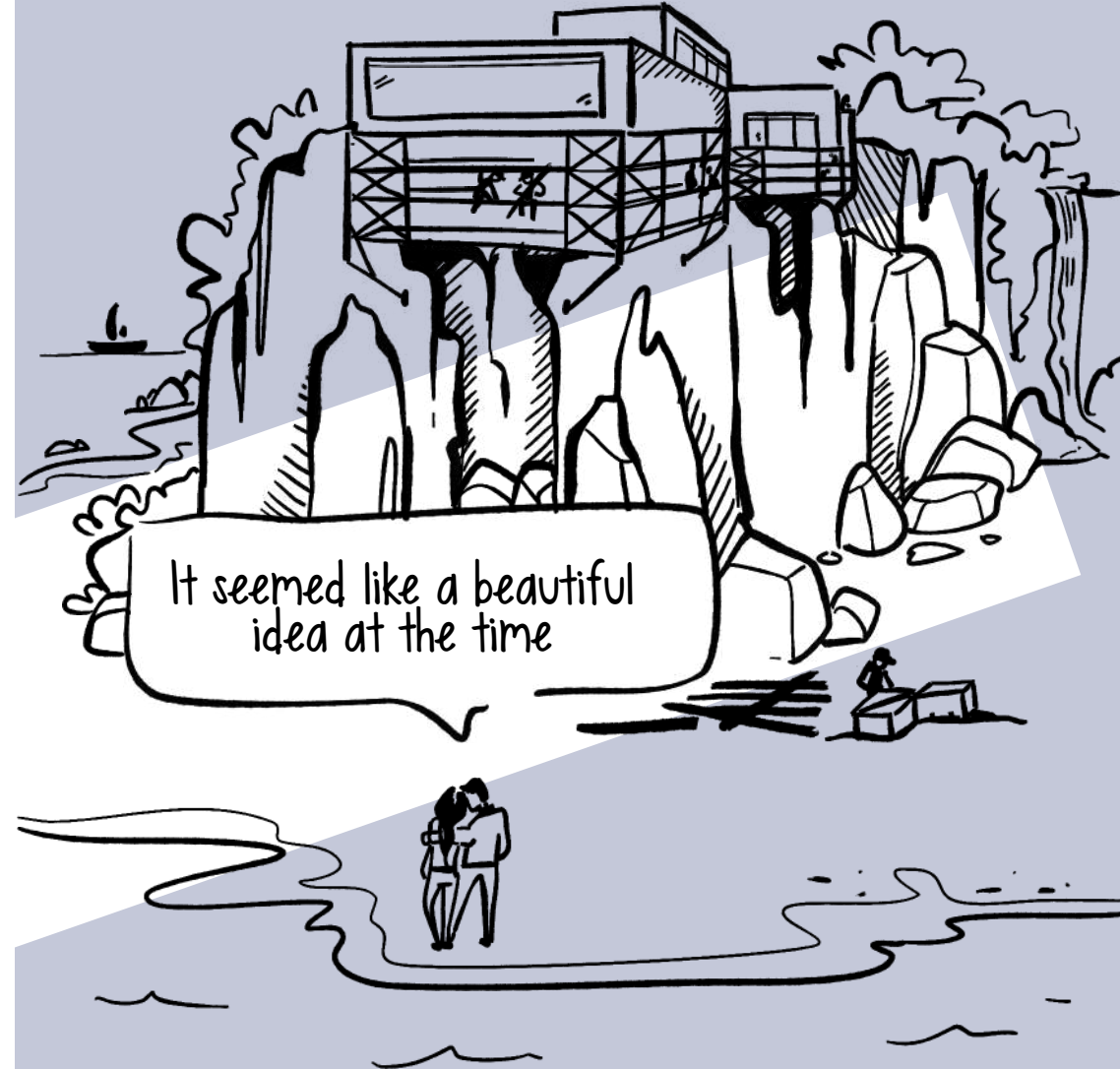
Code – a balancing act



There are many ways to 'skin a cat' and no one way will ever be perfect. So it is with code. No matter how beautiful and creative the solution, no amount of documentation will ever be good enough (or worth writing) to explain the complexities to others. What you leave out is as important as what you write in – and what you leave for others must be something they can work with. Coding on the client frontline is a delicate balancing act...

Future You hates today's You

Developers love the complexity of code, of being creative and having fun with a problem. We all get a kick out of it and we can point to it and feel really clever! But you have to think about whether Future You will hate Today's You. Because the code has to be maintainable for you and others in the future. The real cleverness belongs outside the code in how you look at, approach and solve the problem, using your cleverness and imagination to find the simplest possible solution. It's as much about what you end up not writing, as what you do.



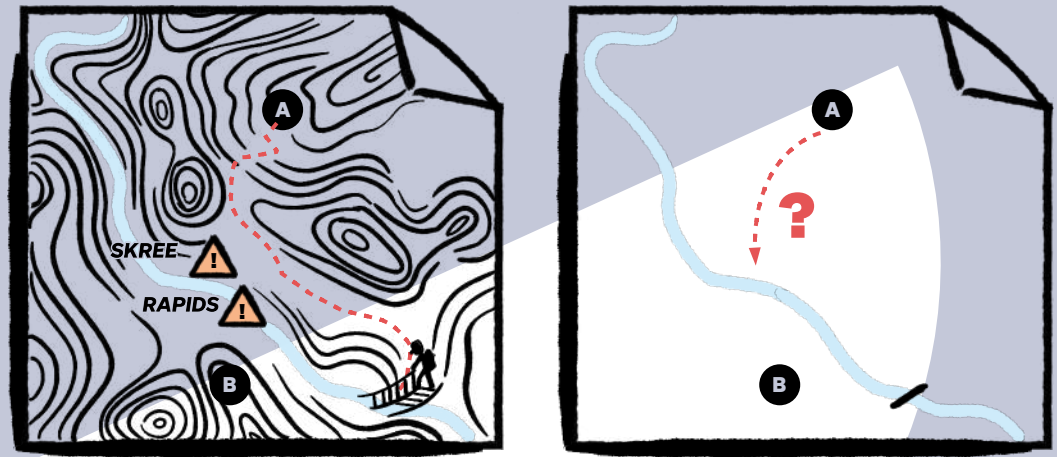
Accidental Complexity is no accident

-

Code is complicated and necessary if the software is going to work, be reliable and of high quality. This is 'essential complexity'. If you dumb down code too much with the aim of making it really easy to maintain, it may not be appropriate for the complexity that's needed for your product. This accidentally introduces complexity elsewhere. We're thinking about this all the time in our work – striking the balance. A good analogy might be:

Imagine you're on a walk in the country and have to get from point A to B. The map you're looking at is complicated, but it has everything you need to make that journey safely, easily and in good time. It has the essential complexity – simple as possible, but no simpler – necessary for the task at hand.

Now imagine it again, but this time your map has been made too simple, less complex: A large river at the bottom of a gorge, and the bridge that crosses it, have become a simple blue line with a mark for the bridge. You start walking to point B, taking a straight line because you assume the river can be forded anywhere. Then you meet the river, swear a bit, and walk off to find the bridge. Dumbing down the map has introduced accidental complexity.

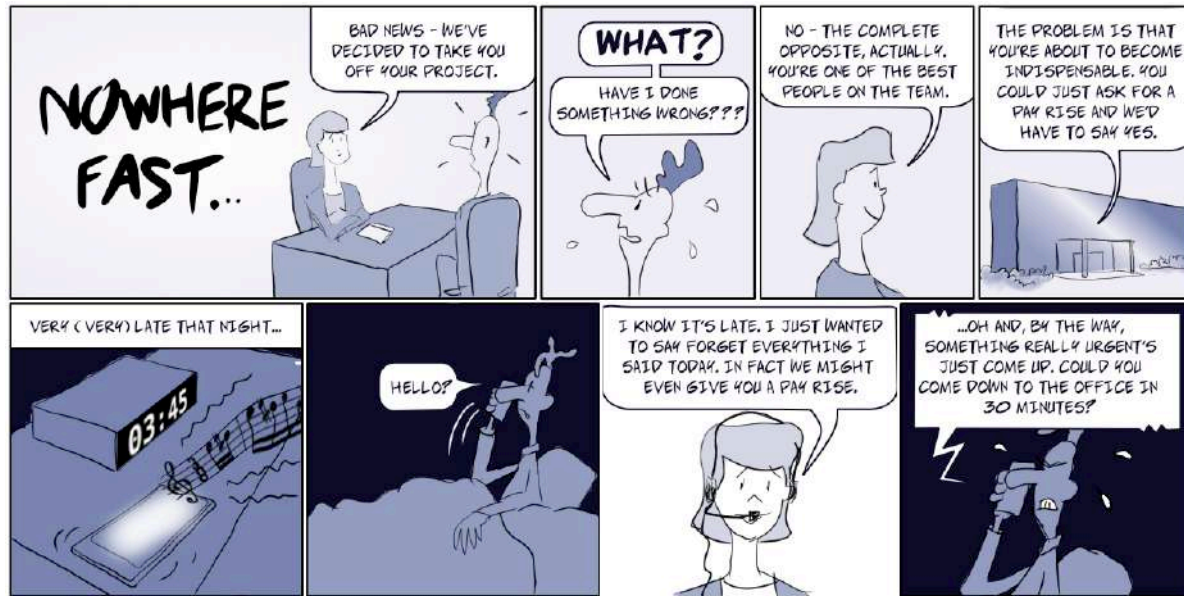


Making ourselves redundant

-

On any client engagement, our aim is to make ourselves redundant. A good test of that is how many calls we get for help after we leave. Our work usually involves deep technical skills which the client teams may not have, so we've got a duty of care to do two things. First, make choices that are best for them. We may find interesting ways to find solutions with code, but they may not be appropriate for the client teams who have to work with it. Secondly, we always think of the engagement as a learning journey for the client teams, upskilling them as we go and making a human impact. We're not trying to hold on to the client and hold on to the system for future reward. Saying goodbye should be a good thing. Measured by the number of calls we get.





Becoming irreplaceable is a risk to the business

-

“I was a software engineer for a business that told me ‘If you make yourself irreplaceable, we will fire you. It is harmful to our business for you to know so much that we can’t make do without you. So, if we see that happening, you’re leaving, because that’s professional incompetence’”. That should be applied to vendors too. (But what about the Pirate Ship Captain’s issue with ‘the keys’? See page 3).

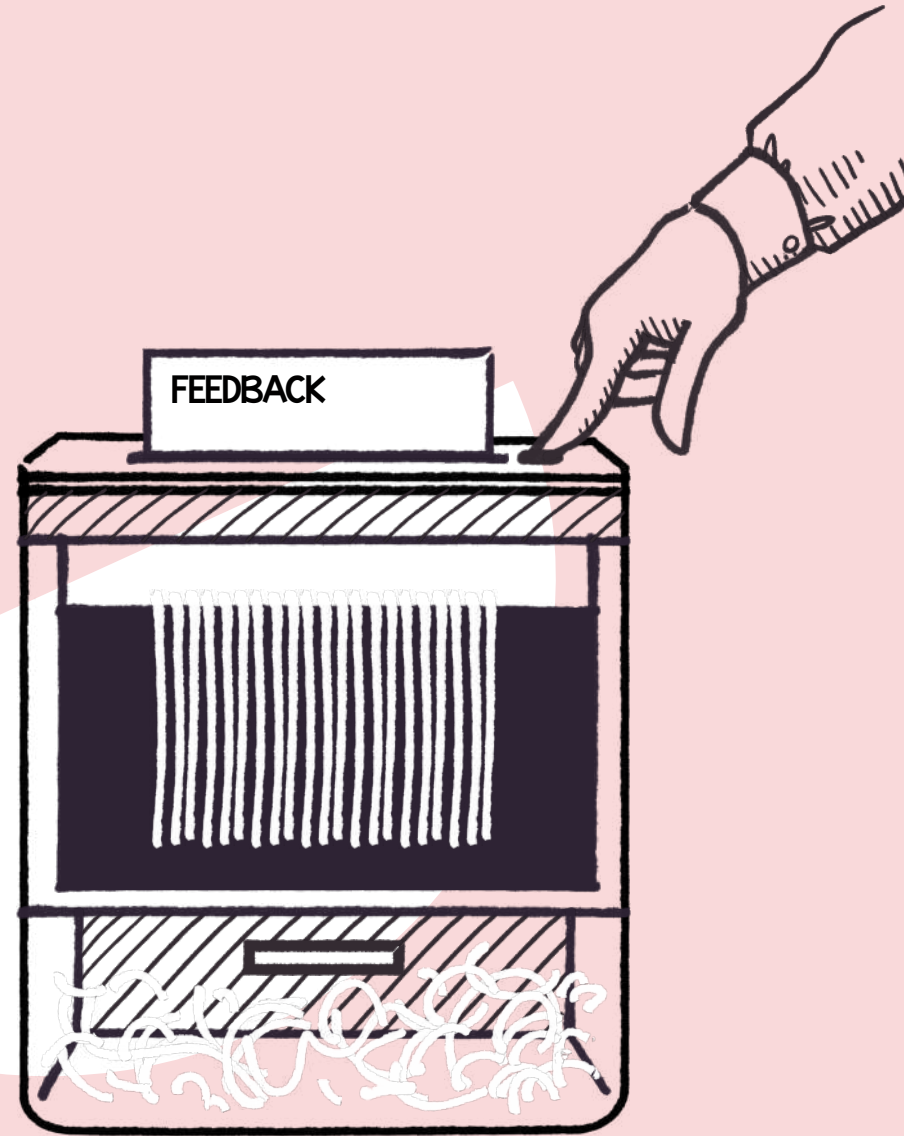


Random stuff

Kept in the dark

-

“I needed to get customer feedback on a product we were building so I reached out to the analytics team lead. He was wary about giving it because ‘it goes through management and they decide what that information means’. It seemed daft. So I proposed to our Product Owner that we build a widget at an upcoming hackathon and get the feedback directly. We did that and the support we got during the hack was fantastic. Sadly, management never went to the event but, just before we could put the widget into production, they found out and shut it down. This might obviously be about management not trusting us to interpret the data correctly, but really it’s about management self-interest and keeping negative feedback secret, as if that reflects badly on them. Negative but constructive feedback is, of course, more valuable than positive feedback and it’s the kernel of continuous improvement.”





Nothing is broken in a system that doesn't have goals

-

"Every system should have goals to which the work of teams is aligned. I recently asked, for example, 'What are the goals of this particular technology system?'. I got a blank. If you don't know what the goals of the system are, then you can be sure of two things – teams won't be aligned and you'll never know what's working or not. It's impossible to tell if something is 'broken' unless you know what you're trying to achieve and then, even if something is broken, whether it's even worth fixing at all. Fix goals for the system first."

Things aren't necessarily broken just because we hate them

-

Everyone hates 'procurement'. It's always laborious, painful and takes an age. The process itself also comes with very high costs for all parties. On the face of it, typical 'procurement systems' seem ripe for fixing. But are they really? If the goal of the procurement system is to deliver X, Y, Z and it's doing it, then you can't say it's broken. There are some things in working life that you'll never hear a good word about, no matter how good they are. Procurement systems are probably one of them.





Now that you all have individual targets, we will achieve our goals

KPIs don't fix a broken system. They break people

-

W. Edward Deming said that “the bulk of the causes of low quality and low productivity belong to the system and are beyond the power of the workforce”. Simply put, the system itself needs attention. It’s leadership’s responsibility to improve the system but, too often, the response is to place KPIs on teams and individuals lower down to deliver through it. Deming said something about that too... “Targets create adversarial relationships” and “If you use targets and objectives, your people will burn the company down to meet them”. What you want is for the system to function well and achieve its goals, not for anyone to be doing things that don’t benefit the system, just to hit a target.

“Surely you can just move the logo?”

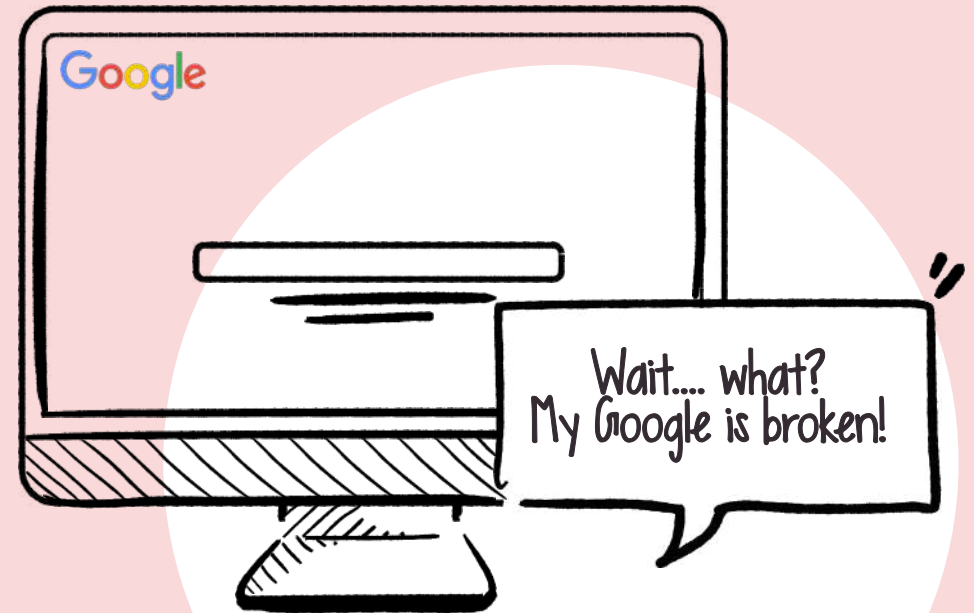
-

The Highest Paid Person’s Opinion (HIPPO) can be genius or lethal to value generation. Opinion outweighs data, hierarchy outweighs team and violence of a subtle kind is inflicted on those who actually bear the responsibility for delivery. For example...

We had a user interface (UI) with 10 million daily users. The front-end UI on our main device had very limited graphics resource and so the implementation was limited to a specific page layout template that ran across the entire UI. One day, the new CEO declared that the company logo must move to the top left of the interface on every page. This was months of work that would interrupt other valuable work. It made zero commercial sense. But the CEO was resolute – “JFDI!”. And, in the process, he destroyed the team’s motivation too.

Thankfully, the world has largely shifted towards experiment-led customer validation and discovery. Good leaders who might look for a change in something will present product engineering teams with a hypothesis to test and then wait to see the data.

Or, perhaps, even ask if there was any existing data that was relevant...



The customer is always right, right?

-

Years ago, we were working on a web portal (remember them... Yahoo, AOL, MSN?) with five million users a week. Our CEO received emails complaining that the Google search box – which we placed top-centre of the homepage – was getting in the way of the content they really loved. The CEO asked us to “move the search box below the hero content”.

So here’s the issue: Customer-centric design said to move the thing out of the way. But anyone who understood how Google search referral revenue worked would know this would seriously reduce how we made money on the site. We had the data to prove it.

The CEO exercised their HIPPO rights and said they didn’t care because the customers were right. Wrong! Or at least, it depends on the measure. So we ran an experiment, relocating the search box and deploying it to 10% of the audience. Search revenue dropped off a cliff for those users.

The lesson here is about discerning signal from noise and about using data not bias to make delivery calls. The HIPPO thought they were making a good call but the sample complaining was just a small subset of the audience. Two weeks later, the search box was back, front and centre.

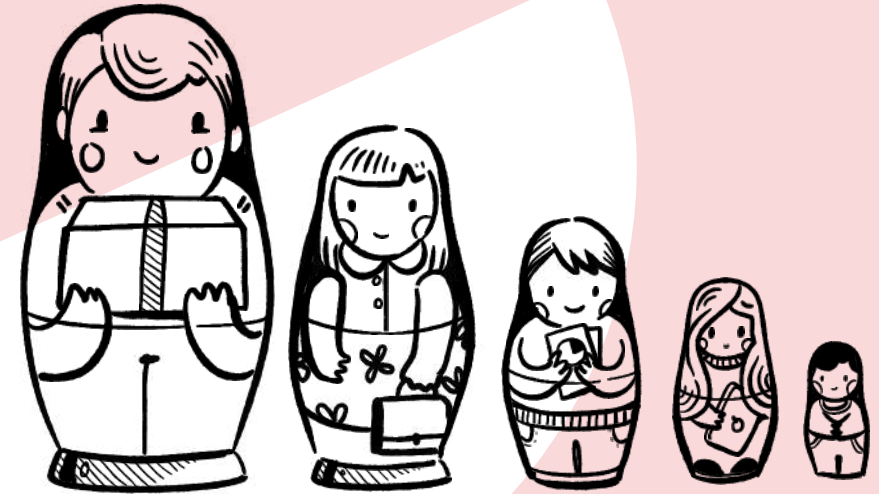


Why do we even have IT projects?

-

You should always consider the possibility of not doing big projects at all, especially rip-and-replace of core systems. The grand gesture to sweep it all away and start again is intellectual laziness, a cop out and a high stakes gamble.

You can be smarter about devising incremental ways forward such as the Strangler Pattern and Agile product-centric development. Aim to preserve agility, maximise resilience and minimise risk. The following books will inspire you: *Project to Product* by Dr. Mik Kersten and Shane Hastie and Evan Leybourn's book *No Projects*.



CONTRIBUTIONS AND CONTACT

This issue was inspired by Laché Melvin, Don Smith, Rob England, Benji Stephenson, Noah Cantor, Tony Luisi, Justin 'JT' Tomlinson, Ajay Blackshah, Vicky Price, Gillian Clark, Gareth Evans and Nick Allan.

Tiny Tales from the Frontline was edited by [Siân Hoskins](#).

Illustration and design by HYPR's Creative Director, Anne-Amelie Berdugo.

The Nowhere Fast cartoon character was created three decades ago by Robbie Kirkpatrick. We're so happy to meet him again through Robbie's (much older) pen.

FOR MORE FREE RESOURCES



Playbooks



Zoom backgrounds



Posters

HYPR

In Pursuit of a Smile :)



HYPR.NZ