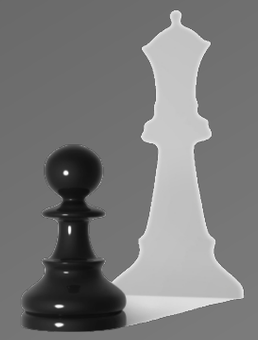
The Effective Architect

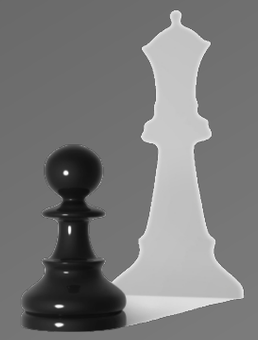
“Hello Human Intelligence, meet The Crunch”

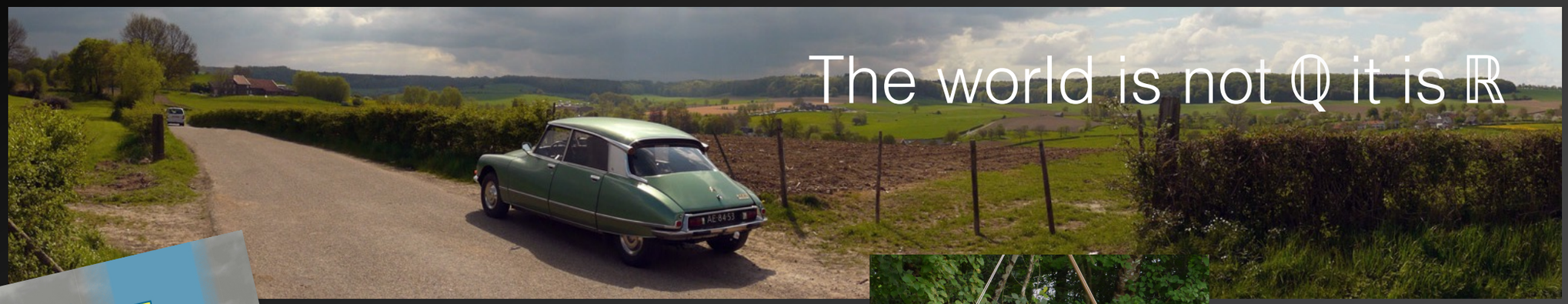
Gerben Wierda
Version: 20240917



This PDF is a Deck that has been produced from the slides as presented during the presentation. They have been adapted to remove graphic effects. Some slides that were skipped (lack of time) during the presentation have been included. Some slides have been augmented with the point I made orally at the time. Many slides have links to more extensive underpinnings of the statements. Enjoy.

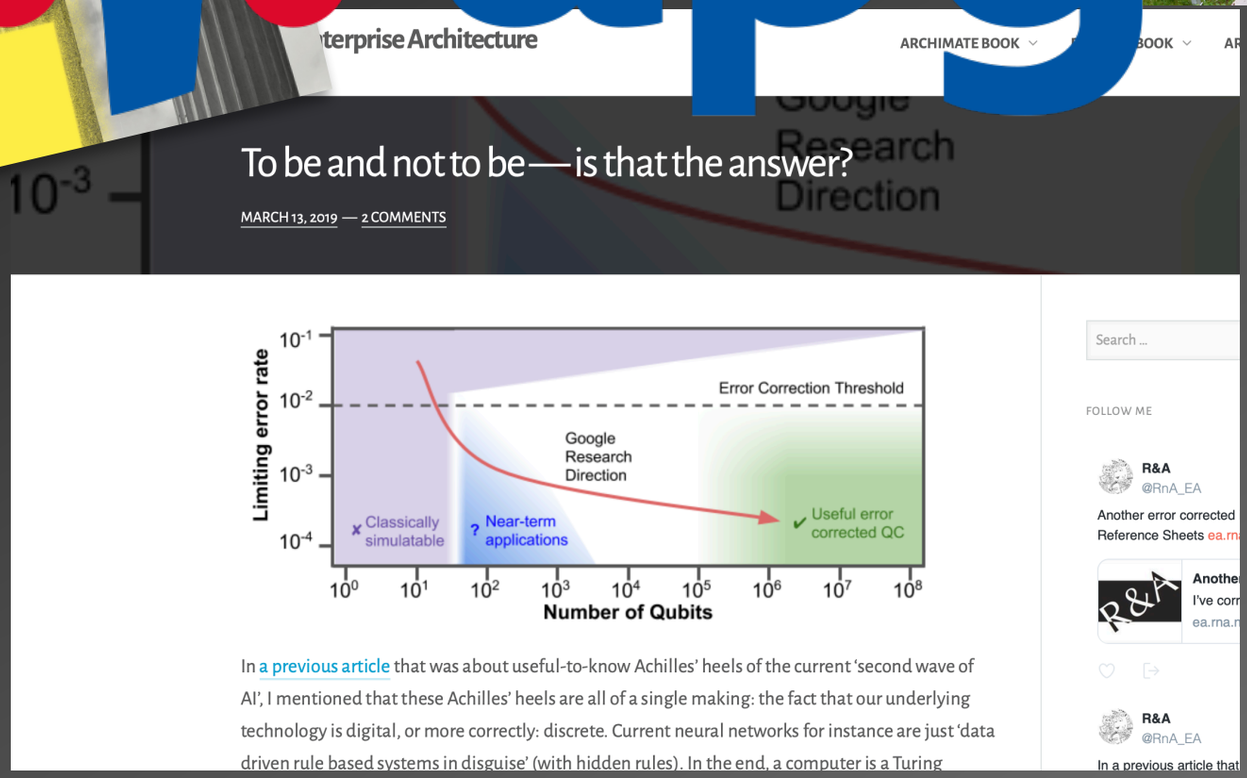
Gerben Wierda
Version: 20240917

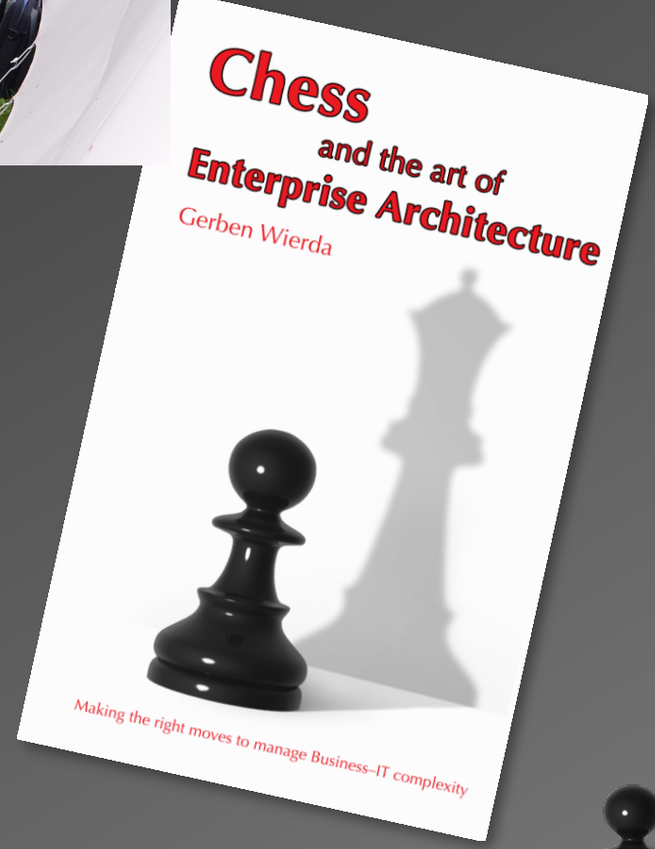


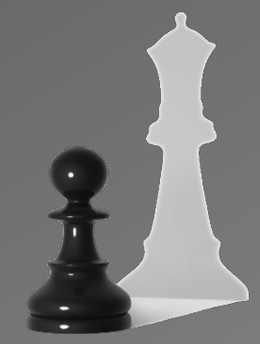

The world is not ℚ it is ℝ


Mastering ArchiMate
Edition 3.1
By Gerben Wierda
A Serious Introduction to the ArchiMate® Enterprise Architecture Modeling Language, Version 3.1

apg

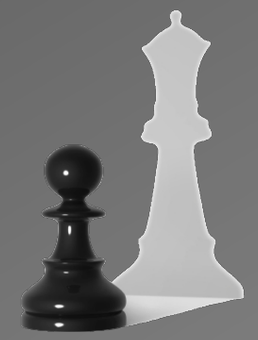

terprise Architecture
ARCHIMATE BOOK
BOOK
To be and not to be—is that the answer?
MARCH 13, 2019 — 2 COMMENTS
Limiting error rate
Error Correction Threshold
Google Research Direction
Classically simulatable
Near-term applications
Useful error corrected QC
Number of Qubits
In a previous article that was about useful-to-know Achilles' heels of the current 'second wave of AI', I mentioned that these Achilles' heels are all of a single making: the fact that our underlying technology is digital, or more correctly: discrete. Current neural networks for instance are just 'data driven rule based systems in disguise' (with hidden rules). In the end, a computer is a Turing
Search ...
FOLLOW ME
R&A
@RnA_EA
Another error corrected
Reference Sheets ea.rn
In a previous article that


Chess
and the art of
Enterprise Architecture
Gerben Wierda
Making the right moves to manage Business–IT complexity



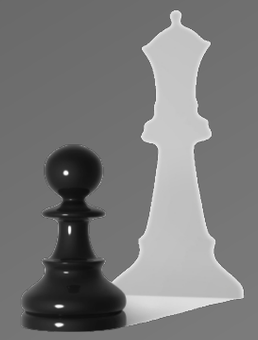
Who is Gerben Wierda

- ~~An actual day-to-day architect in the complex realities of a large enterprise~~ (APG — **NOTE**: Views presented are my own, not APG's)
- Independent author who writes 'from the muddy trenches of architecture in real enterprises' — realist, thinks Uncle Ludwig was mostly right.
  - Books: Mastering ArchiMate and Chess and the Art of Enterprise Architecture, Main blog: https://ea.rna.nl/
- Working for APG since 2007 in various roles
- Before: Lead Architect of the Judiciary in The Netherlands
- Before: Head of Digital Technology of the Netherlands Forensic Institute
- Before: Scientific staff member of the Dutch Council for Science & Technology Policy (government, advises parliament and government)
- Before: various private sector IT-jobs, mostly software engineering
- M.Sc. Physics with minor in Computer Science (1988, University of Groningen) and an MBA (2004, RSM Erasmus)



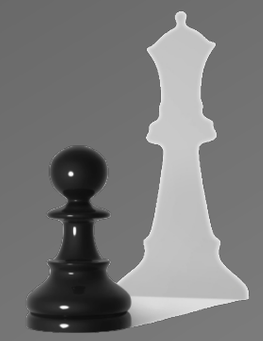
Who is Gerben Wierda

- ~~An actual day-to-day architect in the complex realities of a large enterprise~~ (APG — **NOTE**: Views presented are my own, not APG's)
- Independent author who writes 'from the muddy trenches of architecture in real enterprises' — realist, thinks Uncle Ludwig was mostly right.
  - Books: Mastering ArchiMate and Chess and the Art of Enterprise Architecture, Main blog: https://ea.rna.nl/
- Working for APG since 2007 in various roles
- Before: Lead Architect of the Judiciary in The Netherlands
- Before: Head of Digital Technology of the Netherlands Forensic Institute
- Before: Scientific staff member of the Dutch Council for Science & Technology Policy (government, advises parliament and government)
- Before: various private sector IT-jobs, mostly software engineering
- M.Sc. Physics with minor in Computer Science (1988, University of Groningen) and an MBA (2004, RSM Erasmus)



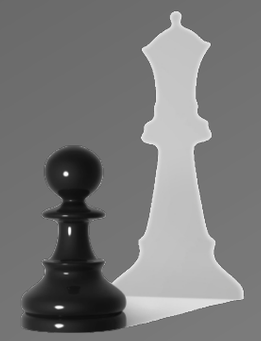
~40 years in one go

What usable insights — about IT and about humans — did I gather from forty years of being at the practical front-end of the IT-Revolution?



#0

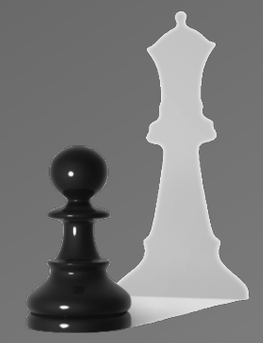
Prologue



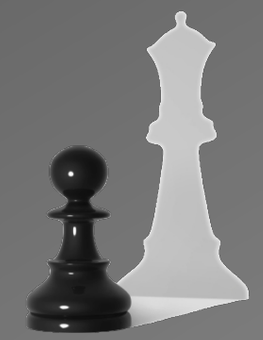
Two Aspects

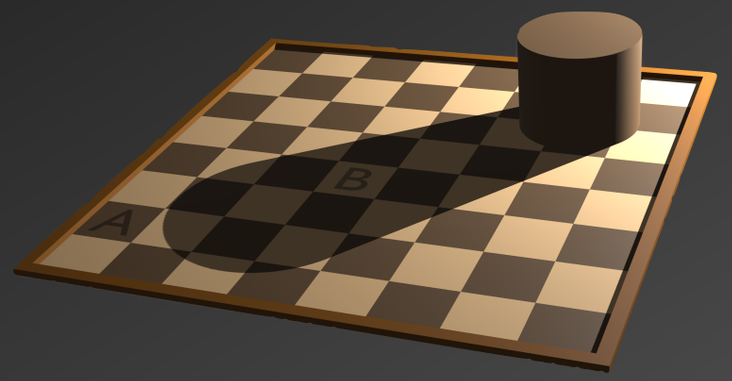
1. Fundamental properties of IT that are crucial

2. Fundamental properties of humans, organisations, and society, that are crucial



Play the accompanying movie here, called ‘minimalShadowChecker’





People call this an ‘optical illusion’ (i.c. *checker shadow illusion*):

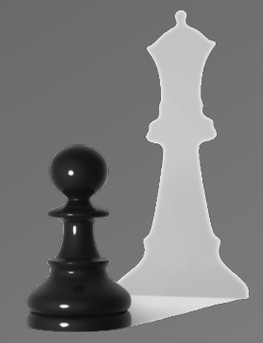
“A seems darker than B, but that isn’t true because they actually have exactly the same colour”

But it isn’t an *illusion* at all, it is an essential useful property of our intelligence:

Our brain continuously tries to ‘calculate’ the underlying *reality* (and that reality is: “there is a shadow, square A is ‘in reality’ darker than square B”, the identical colour that reaches our *retinas* is **not relevant**)

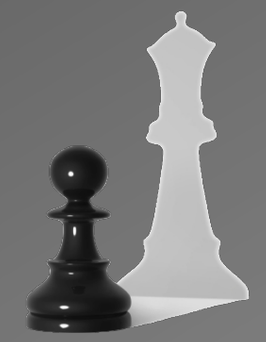
We look as it were ‘deep’ (through the shadow)

(So: realistic painters are brilliant at ‘superficially looking’)



#1

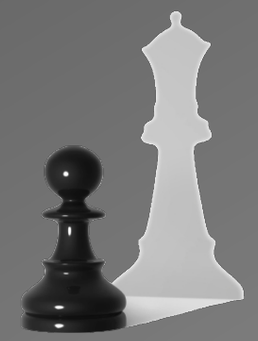
The Mountain of Joy and The Valley of Despair



It’s Getting Harder

“Large complex logical landscapes are increasingly hard to change. Large transformations in/with IT are slow, costly, and often fail. Why?”

https://ea.rna.nl/2020/02/11/a-tipping-point-in-the-information-revolution/

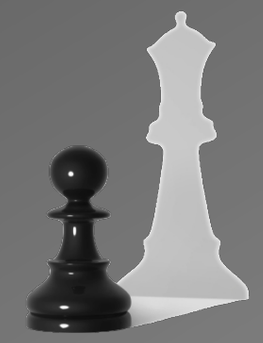


IT-heavy ‘landscapes’ are becoming harder and harder to change because of a combination of:

- logic essentially being rigid, dumb, brittle
- the landscape being a ‘loosely’ coupled spaghetti of massive amounts of logic

Perspective: All that machine logic — IT — shows something that looks like ‘mass’ (inertia, resistance to change*)

*) Note, this is the ‘resistance’ concept from physics, not from psychology. Note also: *change* is hard, *adding* is easy-ish.



Limitations

The limits of physics and the limits of humans constrain machine logic in various ways:

- We humans are bad at logic (better at frisbee)
- We humans aren’t overly intelligent anyway
- Physical limitations on performance: speed of computers and networks, size of computers, bandwidth, latency
- Physical limitations on the lifetime of hardware
- Limitations of logic itself (AI!): The world is not $\mathbb{Q}$ it is $\mathbb{R}$

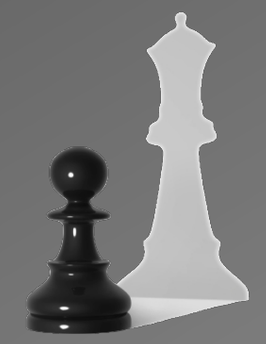
Examples of Effects of Limitations

That beautiful abstract architecture pattern just doesn’t perform and that makes it unusable in a practical sense, often leading to ugly workarounds

Something will only work if you have real AI — and you don’t

Lifecycle events complicate the evolution of your landscape

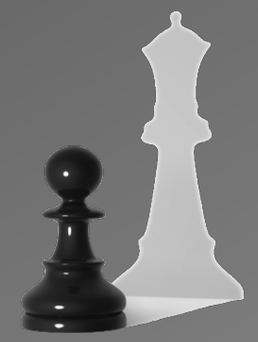
That beautiful architecture governance structure you’ve designed only works with perfect humans



Remember the Singularity?

"Really, the aforementioned singularity is a silly simplistic early trend extrapolation. What we are in reality looking at in IT is almost the opposite: a 'complexity crunch'."

https://ea.rna.nl/2021/09/24/should-you-derive-your-it-strategy-from-your-business-strategy-probably-not-too-much/



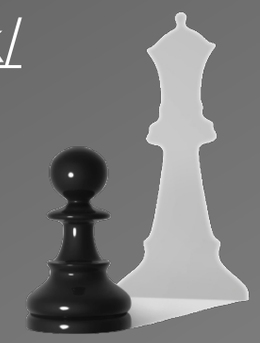
The Real 2-Speed IT

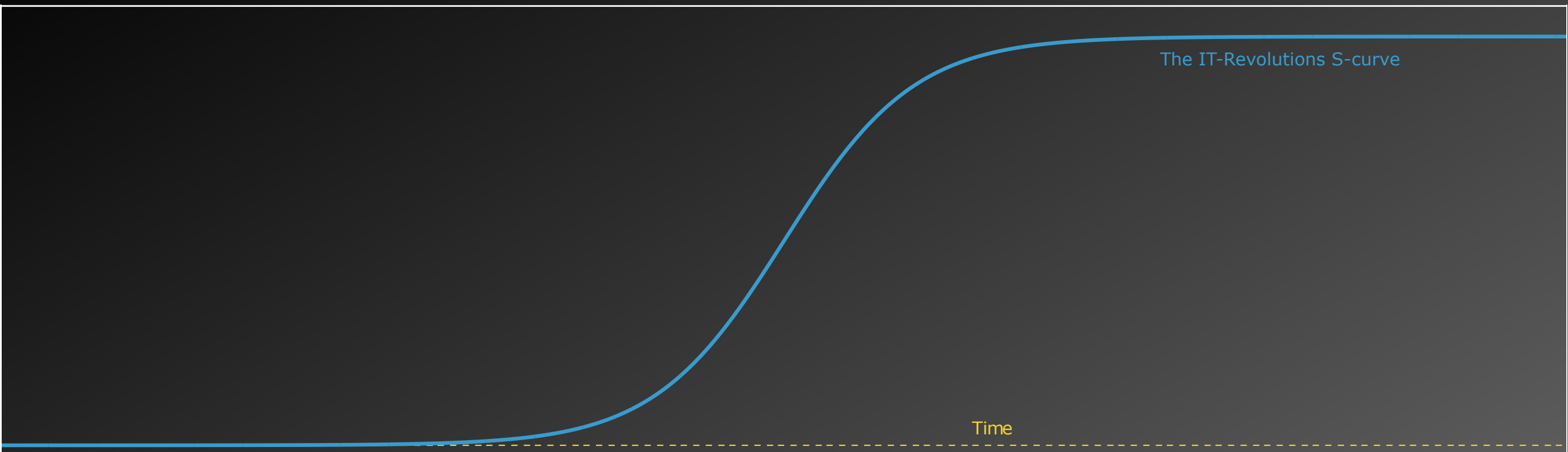
New is easy-ish, change is hard

Complex logical landscapes — i.e. IT — have *inertia*.

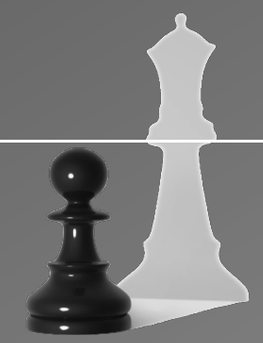
Note: ‘architecture’ in our landscapes is *that what is hard to change*

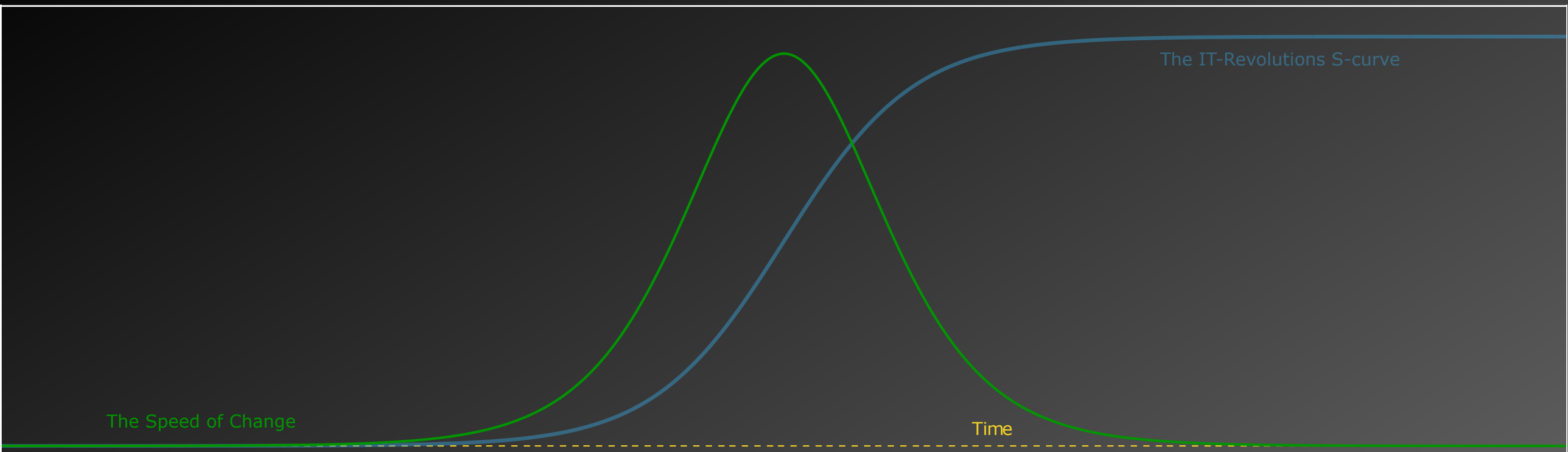
https://ea.rna.nl/2021/05/28/yes-2-speed-it-is-real-but-not-like-you-think/



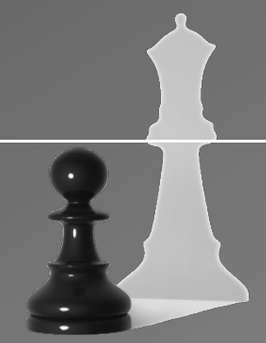


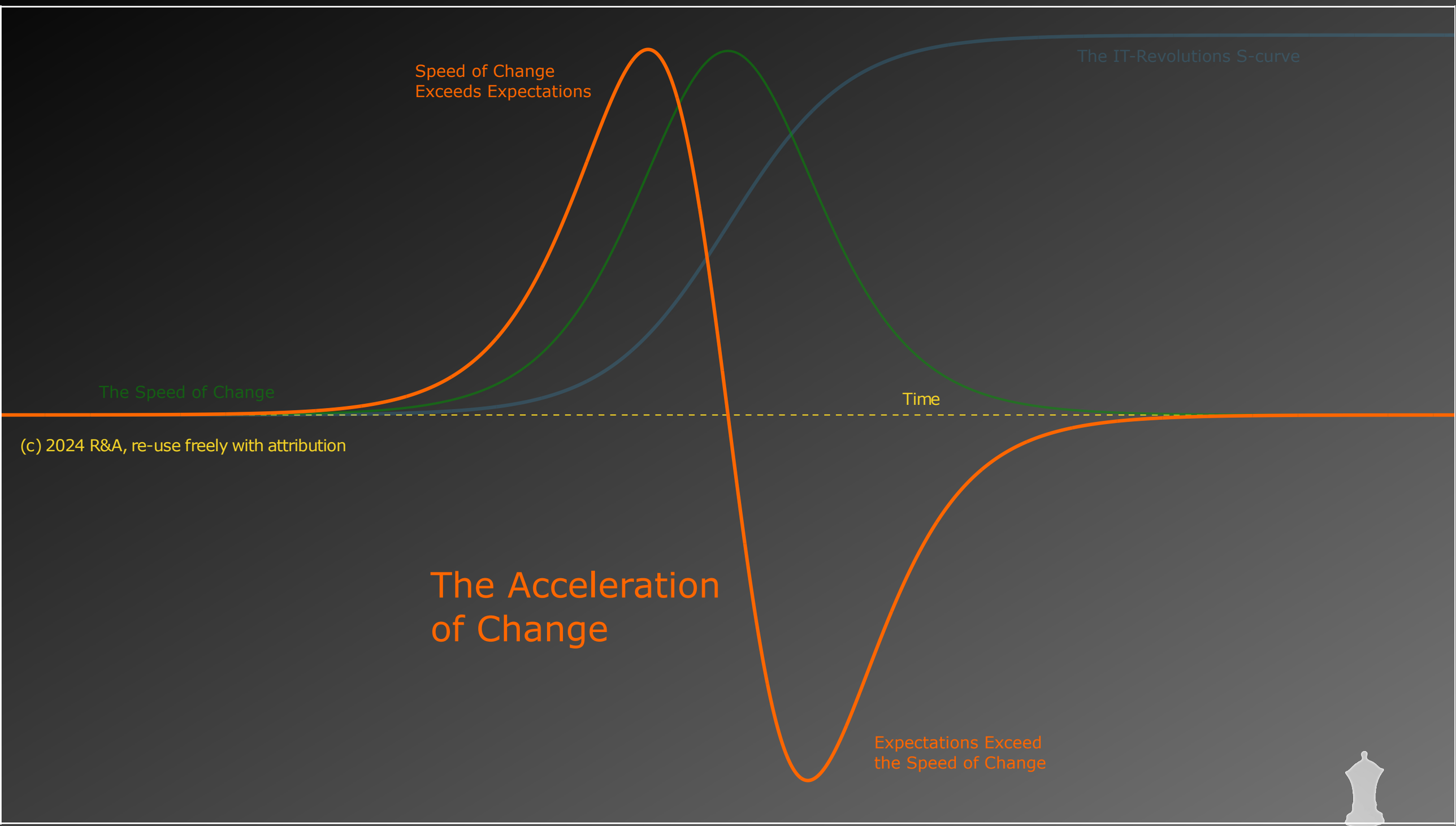
https://ea.rna.nl/2021/07/31/dont-become-an-enterprise-it-architect/



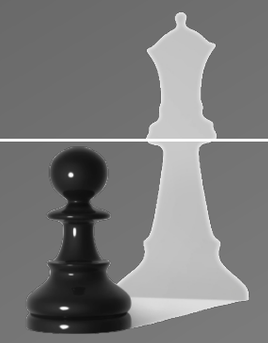


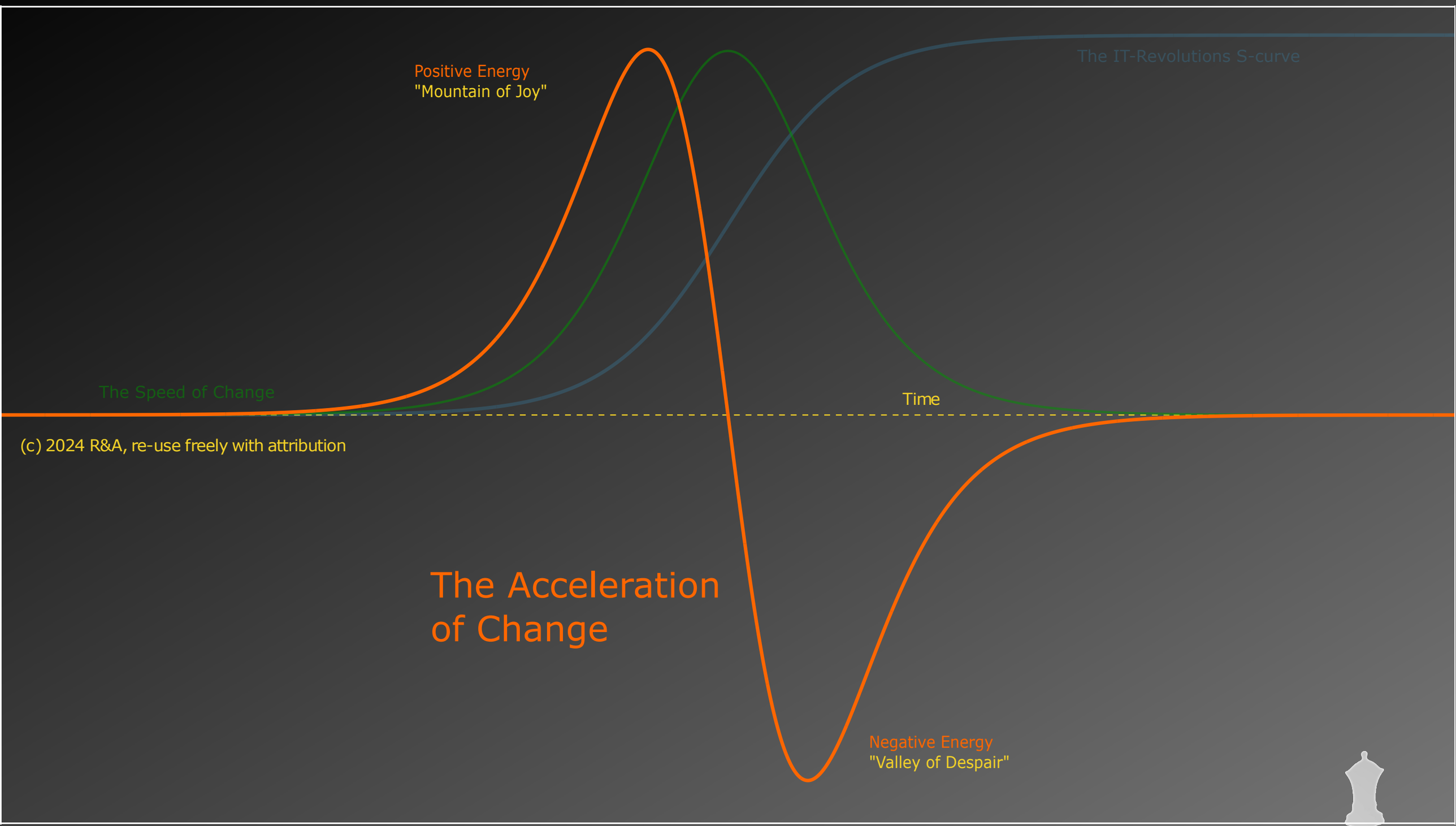
https://ea.rna.nl/2021/07/31/dont-become-an-enterprise-it-architect/



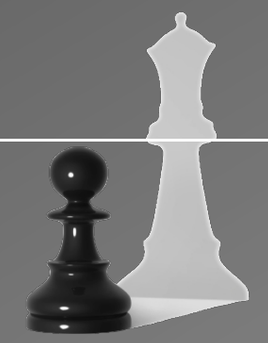


https://ea.rna.nl/2021/07/31/dont-become-an-enterprise-it-architect/





https://ea.rna.nl/2021/07/31/dont-become-an-enterprise-it-architect/



Positive Energy
"Mountain of Joy"

The IT-Revolutions S-curve

The Speed of Change

Time

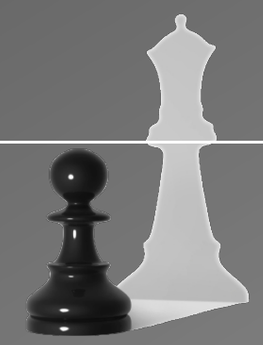
Don't take this too seriously

(it wasn't all fun in the 90's and 00's…)

The Acceleration of Change

Negative Energy
"Valley of Despair"

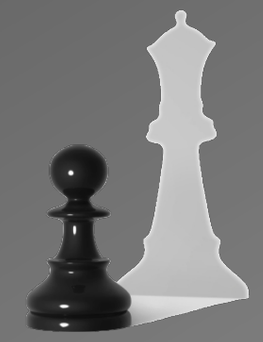
https://ea.rna.nl/2021/07/31/dont-become-an-enterprise-it-architect/



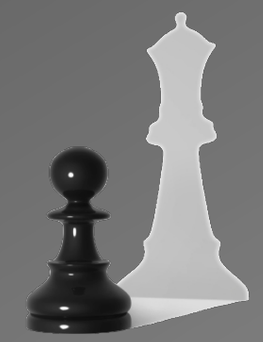
Don't become an Enterprise/IT Architect...

JULY 31, 2021 — 15 COMMENTS / EDIT

... really: don't. It's going to be nasty business.

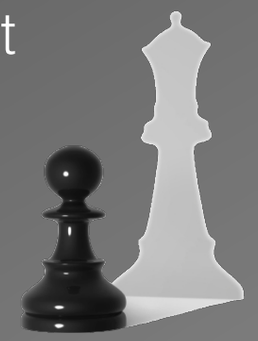


There still are ‘happy’ areas/niches of ‘accelerating change’ (e.g. today: ‘containers’ and ‘infra as code’), but overall, we might already be decelerating



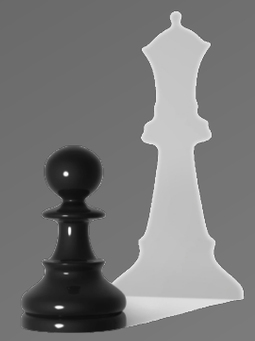
IT-Revolution: gaining productivity (speed, power, volume) at the expense of agility

(Also seen: lowering price at the expense of quality — i.e. the ‘satisficing’ effect of GenAI — but this is not as strong a law as the productivity/agility effect)



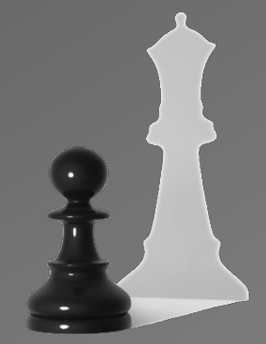
The Challenge

Fighting ‘the crunch’ is a challenge, so it can still be fun (both realism and ignorance work)

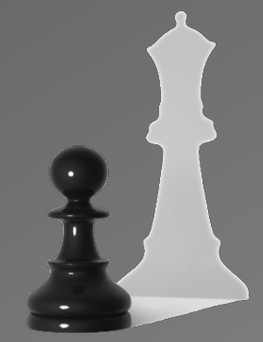


#2

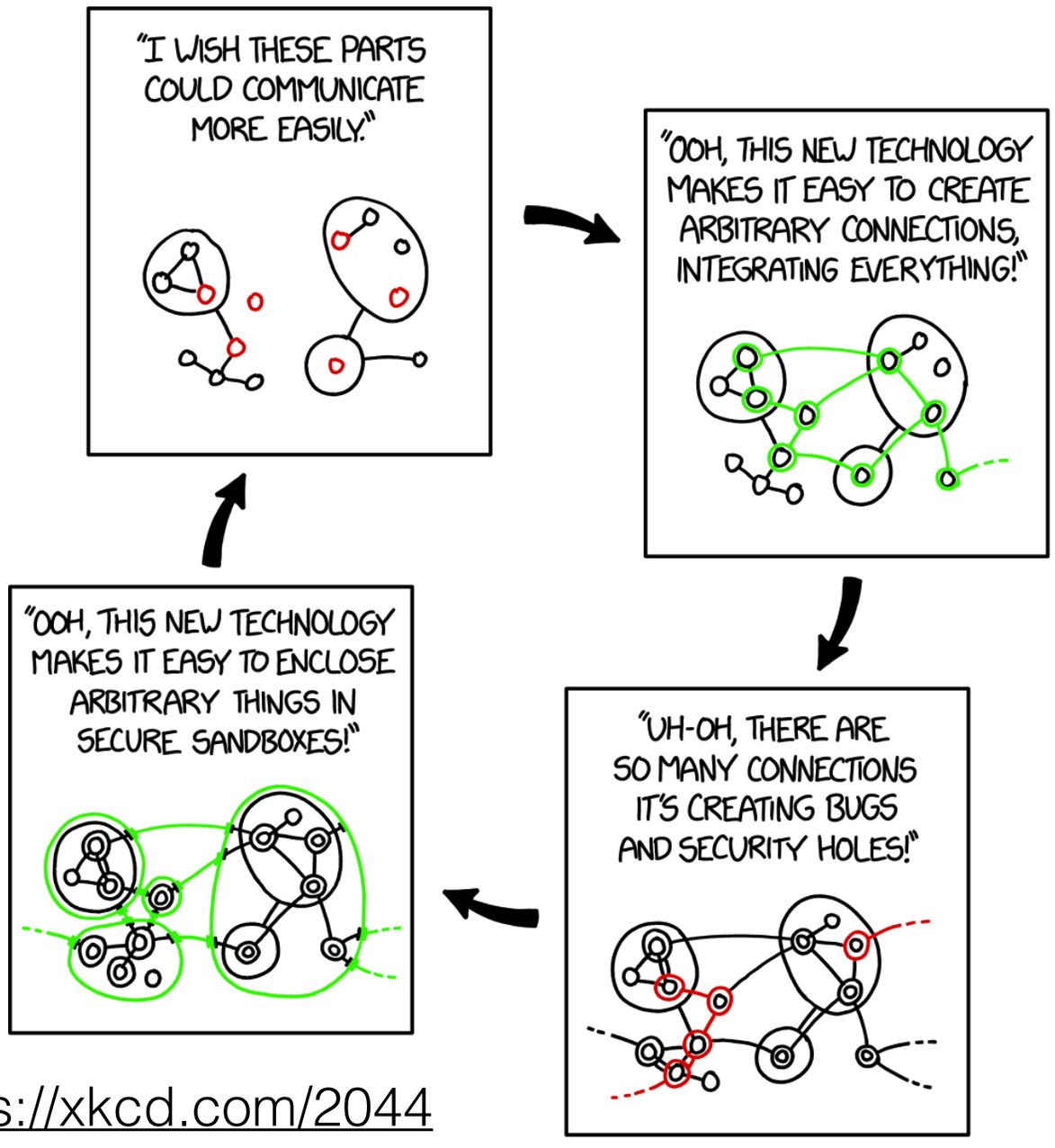
Half a Century of Reacting to ‘The Crunch’



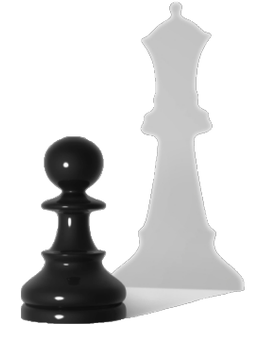
1. We standardise/encapsulate (agility++)
2. We grow (productivity++, agility--)
3. Return to 1



There is — of course — an XKCD Comic for that…



https://xkcd.com/2044

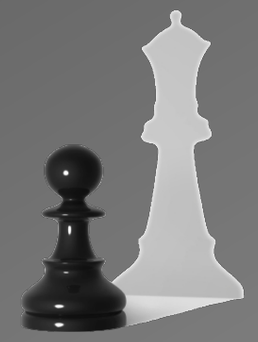


Encapsulation doesn't remove anything, it just reorganises it into manageable chunks and hides it.

Encapsulation *assumes* independence.

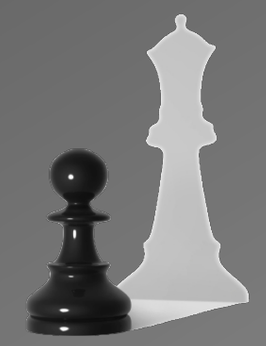
E.g. service orientation succeeds in technical decoupling, but not in logical decoupling

The risk of the hidden stuff rearing its ugly head remains — think log4j.

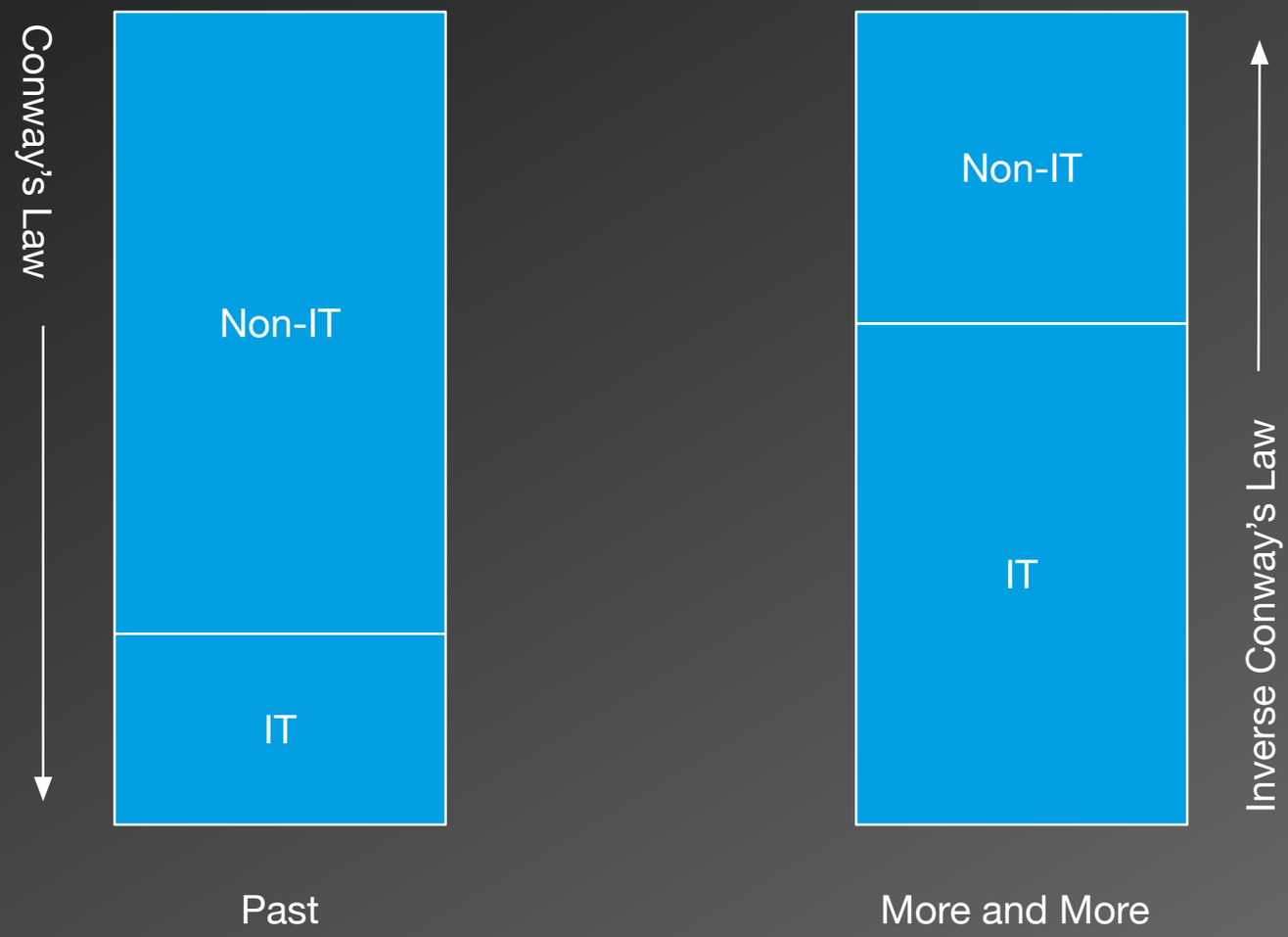


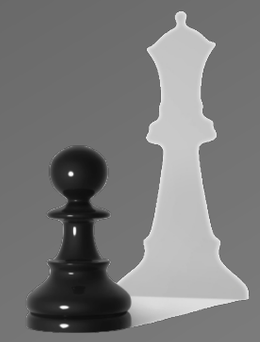
Historical Examples of ‘Encapsulation’

Functions (system code, 1960s–1970s)
Object-orientation (system code, 1970s–1990s)
Service orientation (system landscape, 1990s–2000s)
Agile Development (organisation, 2000s–2010s)
Containers (system landscape, 2010s–…)
BusSecDevOps (organisation, 2020s–…)



Inverse Conway’s Law

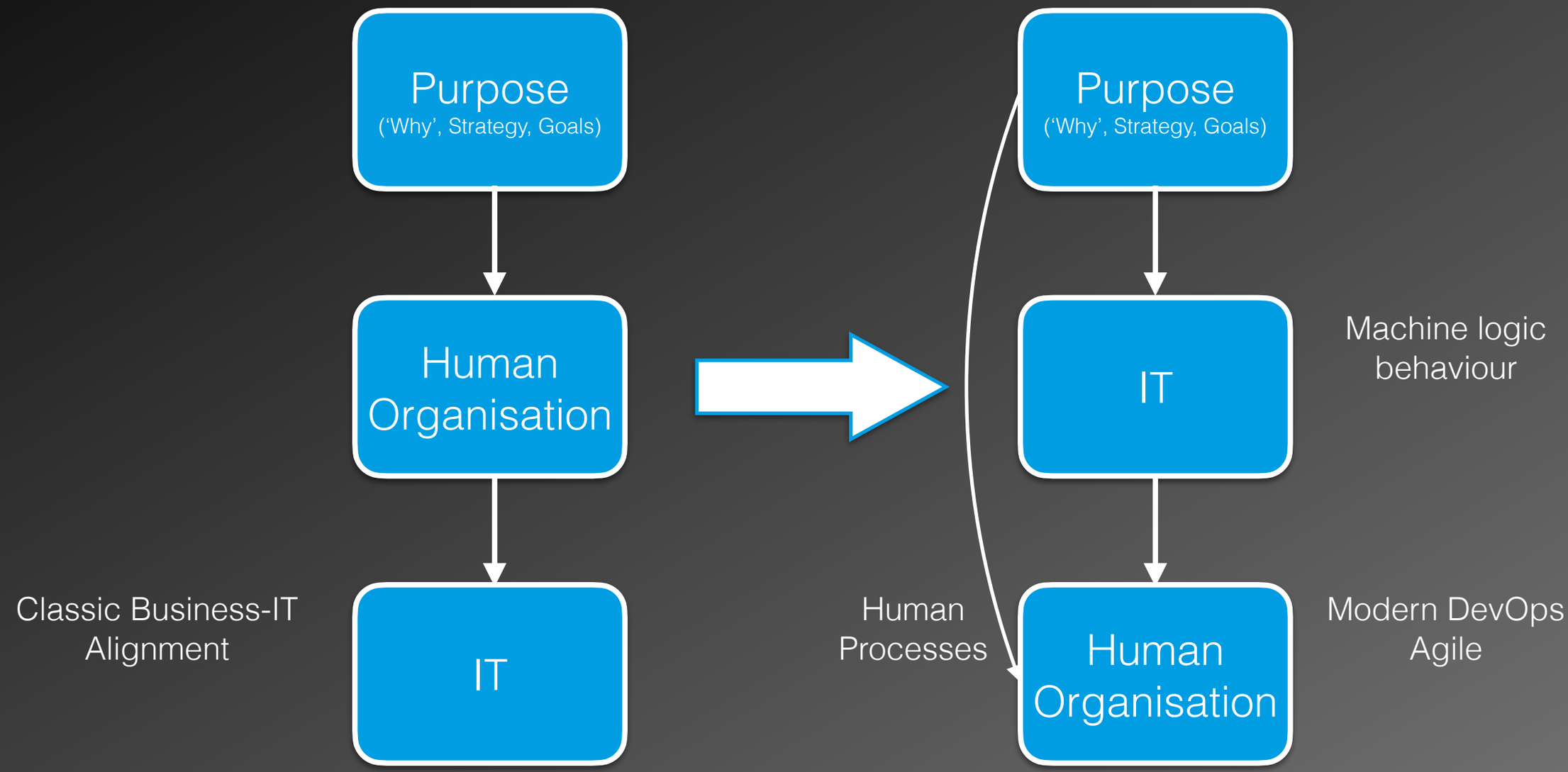


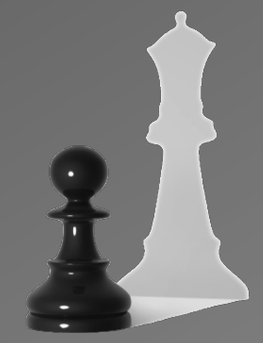


’Digital Transformation/Organisation’

Classical Company

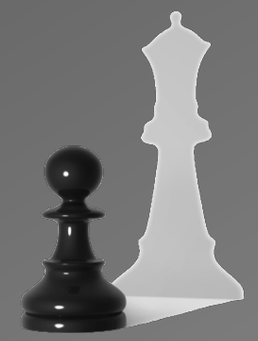
‘Digital’ Company





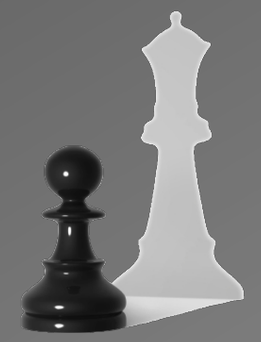
The next level — encapsulation of business units, departments, and the like — may be in conflict with the necessary *coherence* of organisations

E.g.: ask yourself: how important is it that all your people are part of the same Identity and Access Management (IAM) solution?



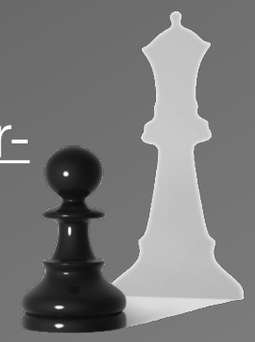
#2.1

Abstraction

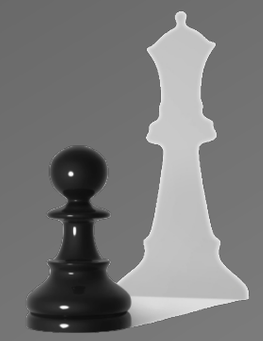


Capability Maps, Business Functions, IT Strategy is important, but it is maybe 5% of the effort — most of the effort needs to be spent on the shop floor where real design decisions are being taken and implemented, few of which can be effectively governed top-down

https://ea.rna.nl/2021/09/24/should-you-derive-your-it-strategy-from-your-business-strategy-probably-not-too-much/

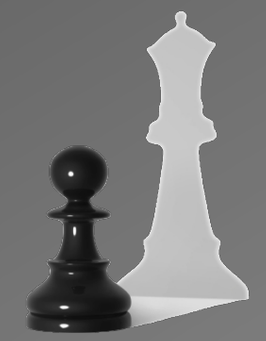


IT Strategy/EA effectiveness through planning and principles is often a *seductive illusion*

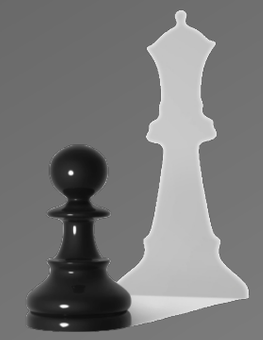


What makes a design choice into a principle?

‘Comply or explain’

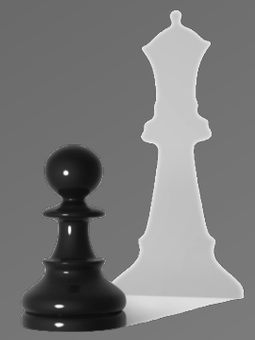


’Comply or explain’ is **a contradiction**



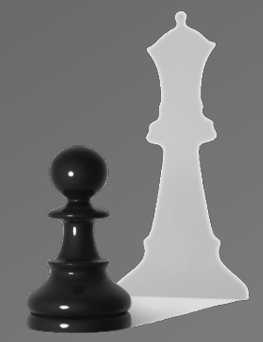
Architecture principles are unnecessary for a setup with good architects (they don’t need to be told of their positive side) and they are dangerous when used by poor architects (who will not recognise the negative side)

A heavy use of principles is a sign of a weak design culture (‘it is the law’)



Architecture principles are **toxic**

https://ea.rna.nl/2018/08/30/architecture-principles-considered-harmful-2/

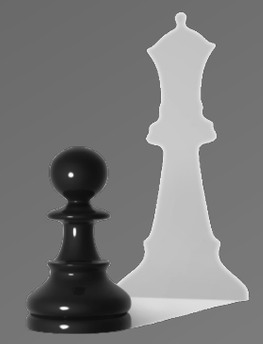


The real IT decisions are those that are actually implemented

Enterprise/Digital Architecture that cannot be shown to effectively influence design decisions on the shop floor is *meaningless*

EA functions (teams, departments) often have a short lifespan (2-3 years between reorganisations). Some may survive for a long time because of fulfilling a ‘shamanic role’.

https://ea.rna.nl/the-summary/

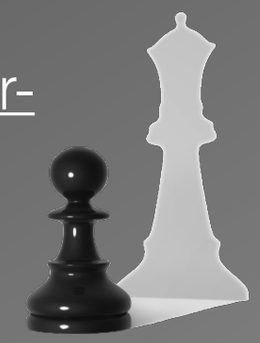


Architecture changes much more slowly than strategy (because of IT's inertia)

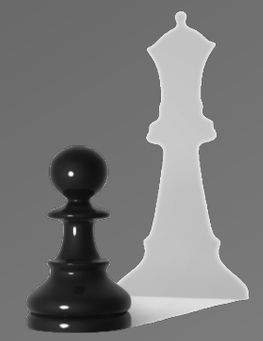
Enterprise/Digital Architecture that cannot be shown to effectively influence design decisions that support future ***strategic*** change is *damaging*

Your current business strategy is not a good basis for architecture

https://ea.rna.nl/2021/09/24/should-you-derive-your-it-strategy-from-your-business-strategy-probably-not-too-much/

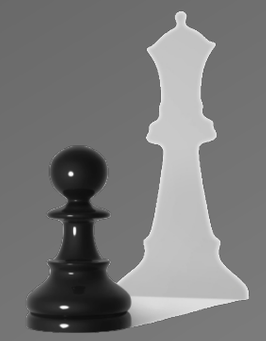


We will have become successful as designers when ‘legacy’ has become a *positive* term



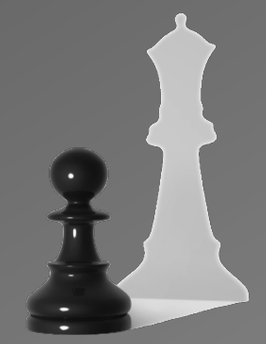
Abstraction really is: consciously leaving out irrelevant detail

(Note: people stumble over molehills, not over mountains)
(Adapted from: safety is consciously taking acceptable risk)



#2.2

There is no such thing as a ‘non-functional’



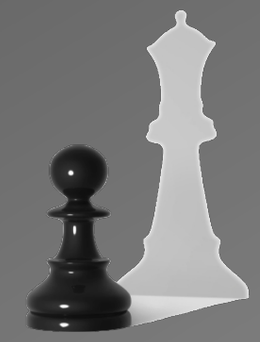
Blame the Mathematicians

(I like to quote Uncle Edsger, but boy, was he mostly wrong…)

Programming has been started by mathematicians.

IT (and then architecture) has long been based on the ideal mathematical idea of ‘(logical) function’ (transform inputs to outputs).

https://ea.rna.nl/2020/01/26/blame-the-mathematicians/



Workflows

License & Cost Management

Change Management (ITIL)

Logging & Monitoring

Change (Development & Deployment)

Disaster Recovery

Configuration Management (E.g. desired state)

The ‘Function’

CMDB

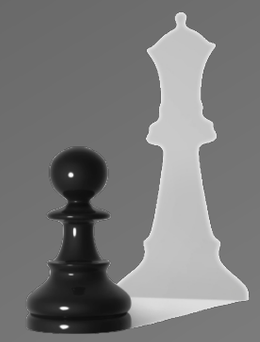
Identity & Access Management

Performance

Security

Risk & Control Transparency

And this is not even everything…



Workflows

License & Cost Management

Change Management (ITIL)

Logging & Monitoring

Change (Development & Deployment)

Disaster Recovery

A common form of myopia

Configuration Management (E.g. desired state)

The ‘Function’

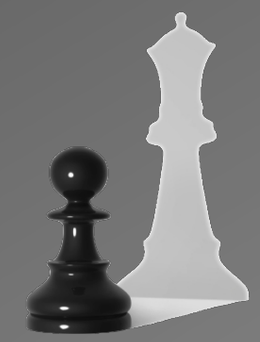
CMDB

Identity & Access Management

Performance

Security

Risk & Control Transparency

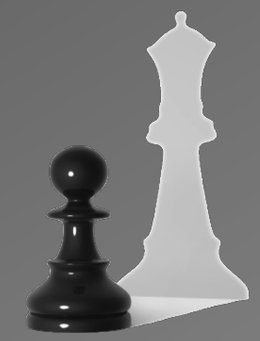


Architecture must be holistic or it is useless

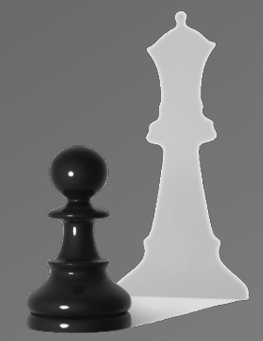
- Risk & Control is part of architecture
- Security is part of architecture
- Performance is part of architecture
- Systems & Resource Management are part of architecture
- Future Change is part of architecture
- Etc.

The pervasive logical/functional purity is a mental disease in IT because IT acts in the real world.

E.g. only in the imaginary world of mathematicians is continuity of IT behaviour (‘functions’) not ‘functional’



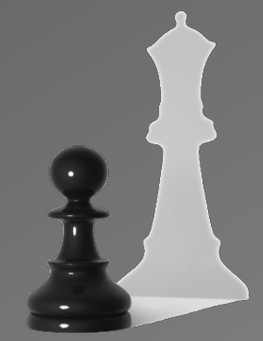
The concept of ‘non-functionals’ is **utter nonsense** — all of them are functional at the enterprise level.



#2.3

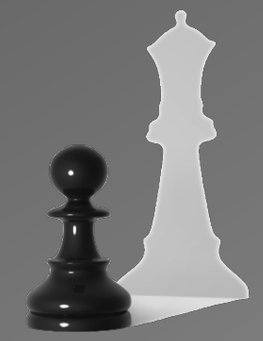
If you're dead, you don't move anymore

If you stop moving, you will die



LCM and removing ‘technical debt’ may feel like they have no functional advantage but at enterprise level they’re essential for your health

This is related to what ‘architecture’ in our domain *really* is: https://ea.rna.nl/2018/09/01/agile-teaches-us-the-true-meaning-of-architecture/

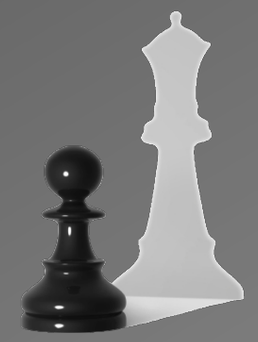


#3

Forget (Generative) Artificial ‘Intelligence’

Human ‘Intelligence’ requires your attention

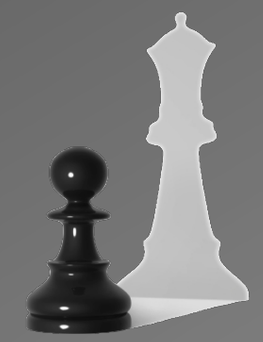
You need to convince people, right?



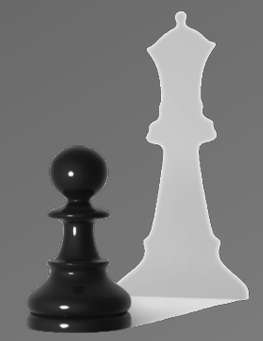
The higher up you go, the more the distance to reality — abstraction, encapsulation — starts to have a negative effect on decision making, e.g.:

the architect's Catch-22

molehills versus mountains



How humans decide is key in IT Strategy/Enterprise Architecture

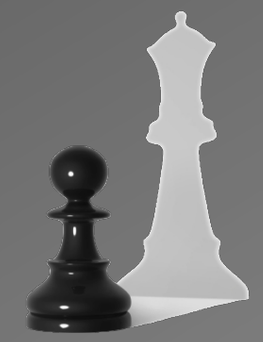


Our human intelligence has evolved as a variable mix of:

an optimal fit for individual success, and

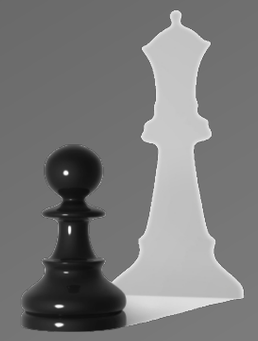
an optimal fit for tribe success (tribe size: “Dunbar’s Number”)

https://youtu.be/9_Rk-DZCVKE

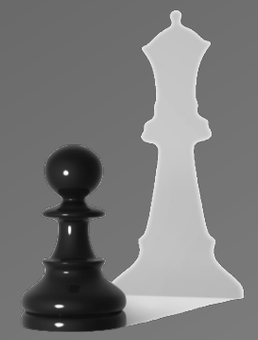


Human intelligence is optimised for speed and efficiency — not thoroughness

The key element that enables us to act quickly and efficiently is the conviction (also: assumption, belief, standard reaction)

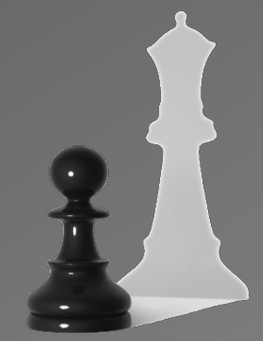


Most of human intelligence is ‘mental automation’

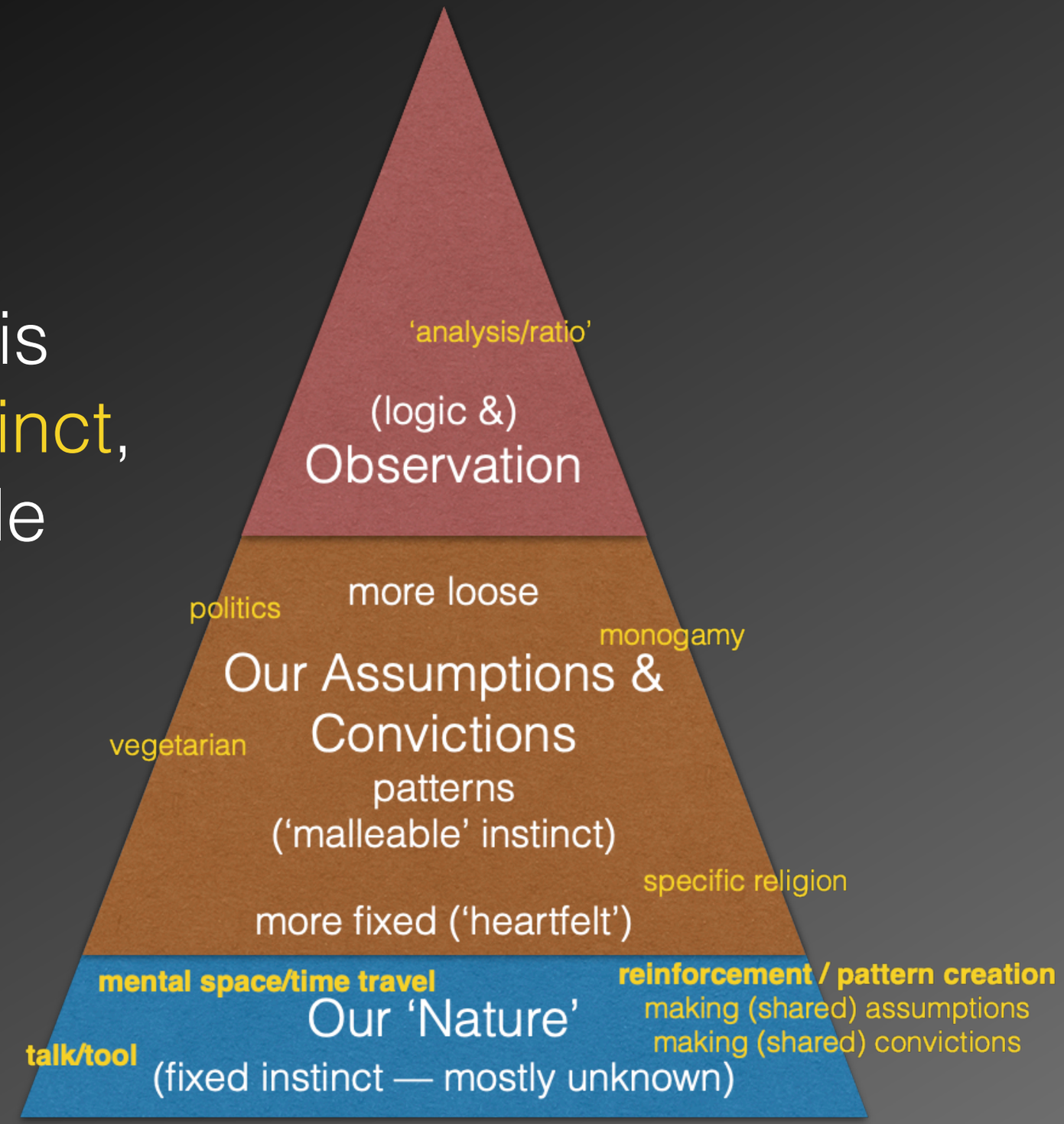


‘Mental automation’ comes with the same trade-off as any automation:

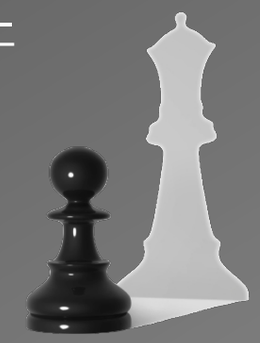
increased productivity paid for by a decreased agility



Human intelligence is (mostly) malleable instinct, realised by malleable hardware

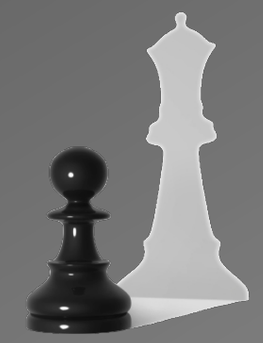


https://ea.rna.nl/2022/10/24/on-the-psychology-of-architecture-and-the-architecture-of-psychology/

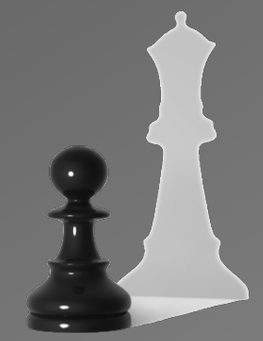


Note: We have built much larger structures than tribes (organisations, societies) mainly on ‘*shared* convictions’

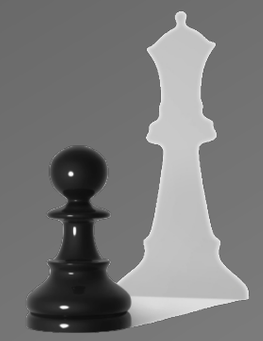
https://youtu.be/9_Rk-DZCVKE



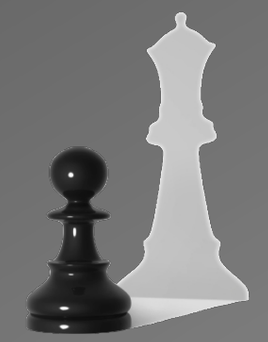
Summary: how humans make ‘design decisions’ is key for organisations — but humans aren’t very ‘thoughtful’, by nature. We mostly have automated (quick & dirty) assessments



we mostly have automated (quick & dirty) assessments

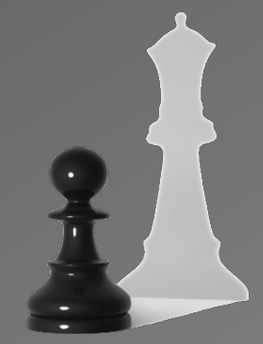


Like now

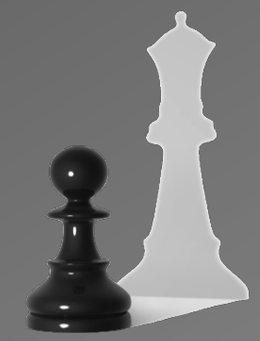


Like now

Take a moment to reflect. You read all those statements so far. How much time did it take you to have an opinion or a judgement? How much energy did you have to spend? Most of the time almost instantly and almost effortlessly. That is your *mental automation* — that is most of your mental activities — at work.

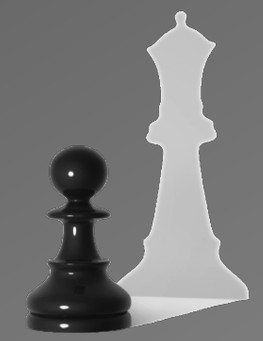


~~Our convictions come from our observations and reasoning~~



Our convictions come from our observations and reasoning, slightly

Our observations and reasoning come from our convictions, mostly

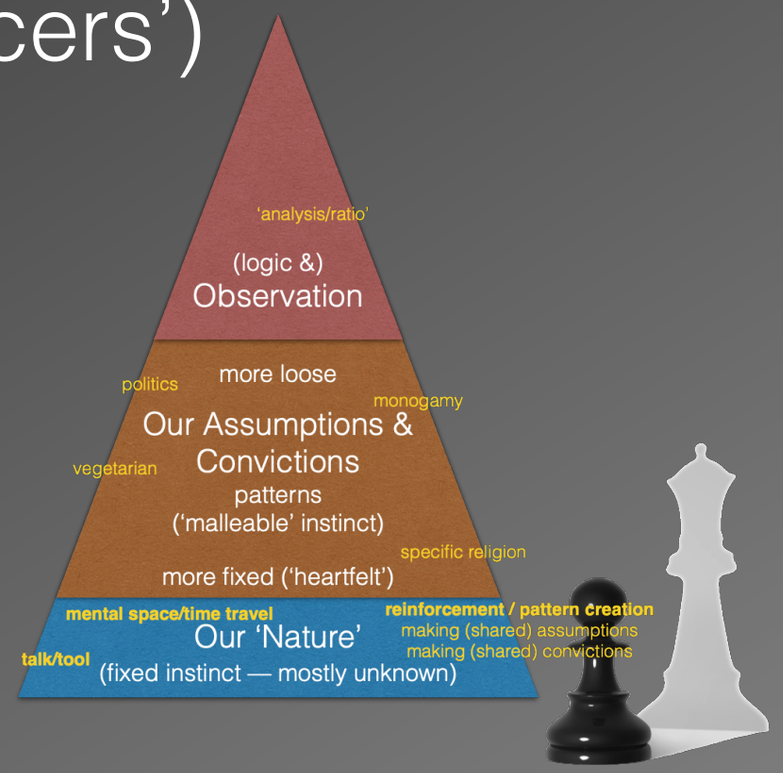


Our convictions (automations) are created from:

learning — personal experience

copying — repetition and close relations are strong creators (see ‘social media’, ‘influencers’)

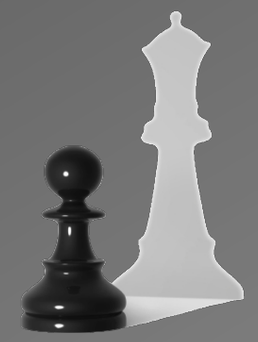
https://youtu.be/9_Rk-DZCVKE



Another key aspect of human intelligence is that it is **not** just the ‘functional’ handling of sensing (everything alive with sense organs does that)

Human intelligence it is built around a strong imagination — we can think of things that aren’t there (yet), we can think in *possibilities*, both — positive — opportunities and — negative — risk

It’s not just deduction and induction, it’s also C.S. Peirce’s *abduction*, as AI author Erik J. Larson has argued in *The Myth of AI*



RPA

Cloud '05

AI 80-90's

Rapid Anything

Cloud '20

Low-Code No-Code

Code Generat-ion

It can happen that (strong) convictions are mostly based on our 'imaginations'

Not just conspiracy theories: every IT silver bullet that has come and gone (and left a mark — or not)

AI 50-60's

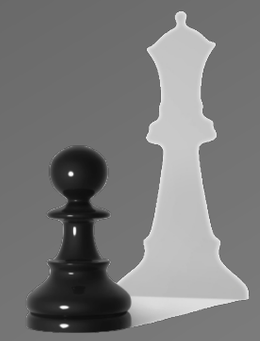
OOP

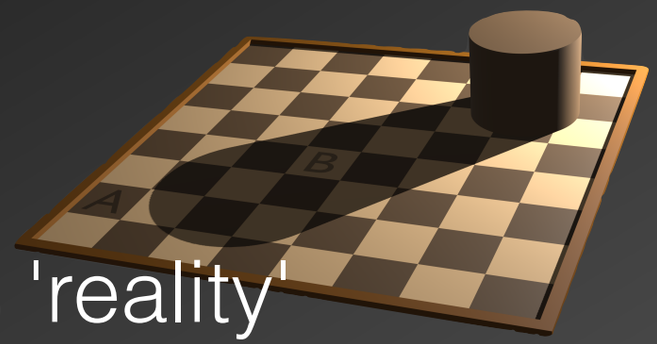
Block-chain

AI 10-20's

BRE

SOA/ ESB



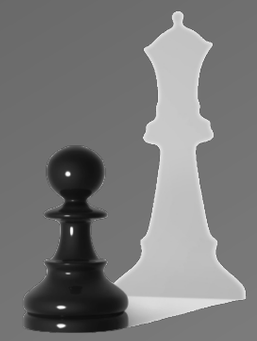


We humans accept what we experience as 'reality' largely on autopilot

- This applies not only to the senses but also to the information we receive. Our brains are dominated by a single mechanism.
- We are vulnerable

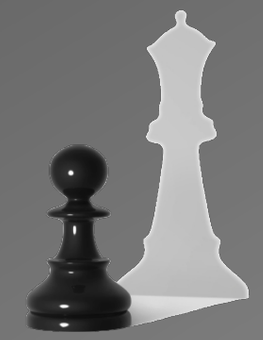
Moreover, we do not experience 'reality' but an efficient *estimate* of what reality is likely to be — an *interpretation*

- That interpretation depends mainly on what we are already convinced of
- We are — just below the surface — rather inflexible

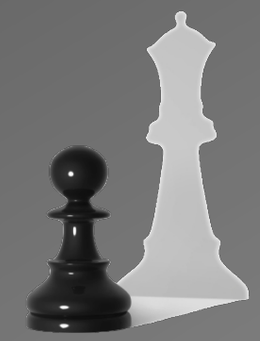


Our collaborations/collectives can be set up to be:

- *more* intelligent (e.g. a tribe, a team, ‘science’, value-based societies)
- *less* intelligent (e.g. a mob, a dictatorship, too much bureaucracy)



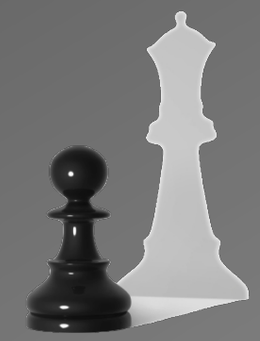
Success comes from *quality* collaboration
not from ‘parallel instinct’ (dumb)
not from ‘top-down enforcement’ (risky)
and least at all from conflict (counterproductive)



Summary: The Key Aspects for ‘Effective Architecture’ are:

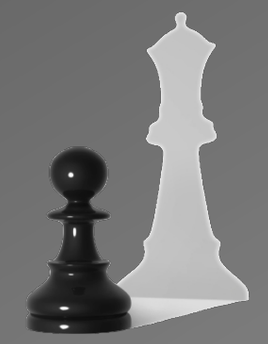
#1. Understanding Complexity Crunch
#2. Understanding Human Intelligence

(The business strategy is far less relevant than most assume)



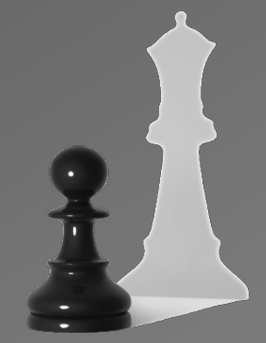
#4

Practical tips that (still) work



#4.1

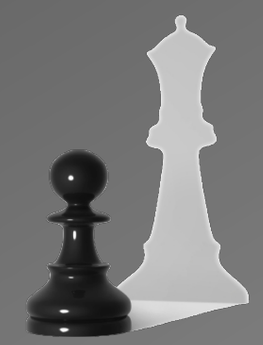
Information is weak, interaction is strong



The most influential architects:

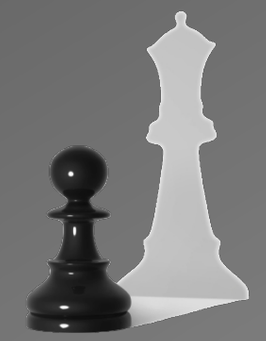
- Are a ‘close relation’ of the decision makers (e.g. have a permanent seat at the table, have frequent meetings)
- Produce information that is often used/consumed

The importance of the actual *quality* of architectural advise depends mostly on the audience
(we architects can be highly influential fools)



#4.2

Architecture Decision Making & Governance



Interaction is strong, information is weak

Collaborative checks & balances

‘Collaborative’ because:

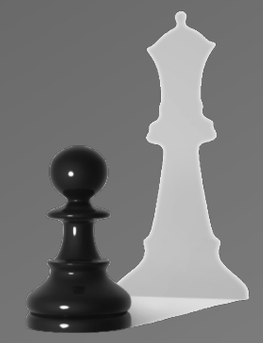
lack of insight in our ‘environment’ makes us blind

‘convictions’ about our own subject make us blind

simple enforcement makes us stupid

conflict makes us unproductive

‘Policing’ is risky and toxic

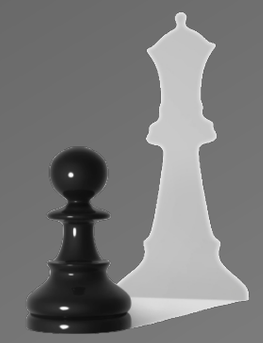


"~~I agree with that~~"

Consent-based decision making

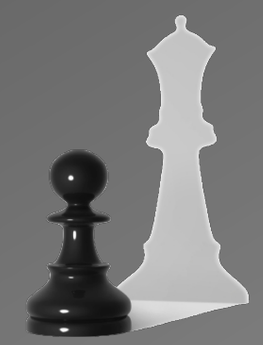
Consent-based, because our world is too complex for perfect solutions, and thus striving for perfect solutions unavoidably leads to conflict, which — like in nature — is costly and risky)

"I can live with that"



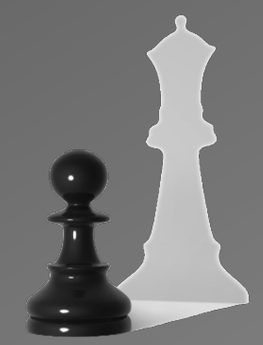
Decision making by people with enough engagement with the subject — fully relying on advisors doesn't really work

Talking of which…



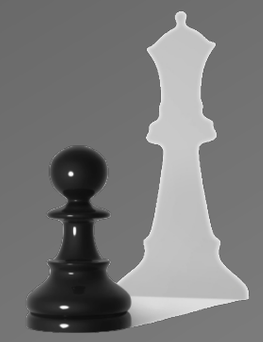
#Intermezzo (for Managers): How To Recognise and Deal with Bad IT/EA Consultants

(after Simon Wardley's *Ten basic rules for dealing with strategy consultants*)



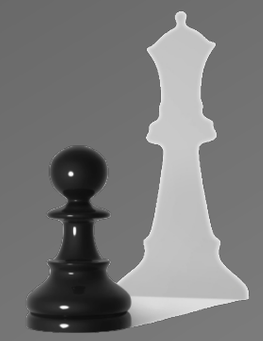
If you do not *really* understand what the prospective consultants are saying, but everything still sounds convincing:

Politely show them the way out, and disinfect with alcohol.



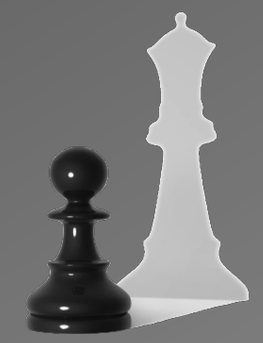
If prospective consultants try to convince you that hiring them is an easy fix for your problem:

Politely show them the way out, and disinfect with alcohol.



If prospective consultants try to convince you that they have an approach for *all* circumstances, it probably doesn't really work in *any* circumstance, so:

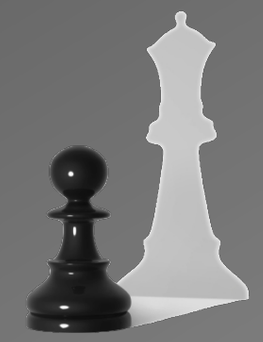
Politely show them the way out, and disinfect with alcohol.



“You should move to an event-driven customer-centric loosely-coupled bi-modal big data micro-services AI cloud micro-frontend container architecture, and you should liberate your data!”

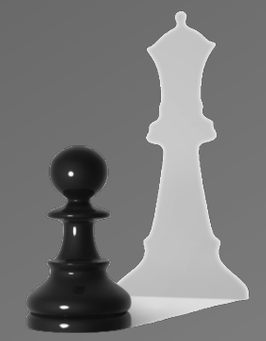
If prospective consultants are using to many buzzwords and phrases that you’ve seen or heard a lot already, they’re peddling simplistic silver bullets to the unawares (i.e. you). So:

Play a game of buzzword bingo, then politely show them the way out. Disinfect with alcohol.



As soon as prospective consultants have used the word ‘legacy’ (negatively) three times:

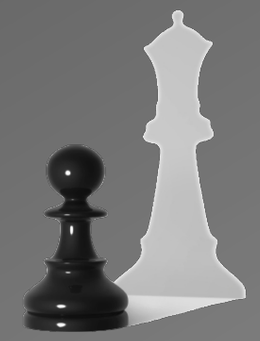
Politely show them the way out, and disinfect with alcohol.



If prospective consultants propose a setting where they ‘own’ the consultancy process, turn it around. A consultant — like an enterprise architect — is an advisor and should never be in charge. *You* should be

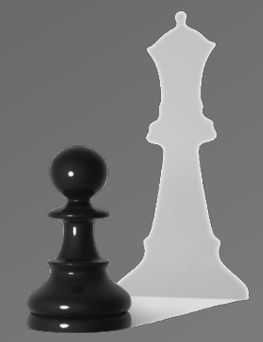
If they don’t play ball:

Politely show them the way out, and disinfect with alcohol.



If you find out that you have been infected, don’t wait.

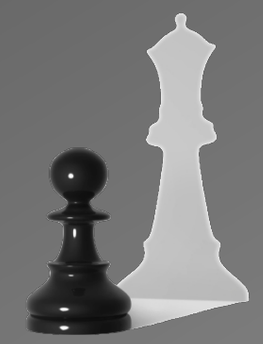
Politely show them the way out. Disinfect with alcohol.



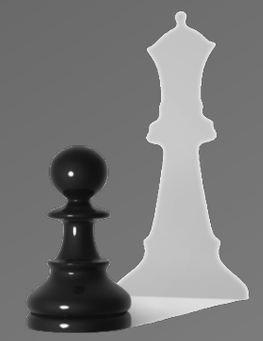
If you find out that you have been infected, don’t wait.

Politely show them the way out. Disinfect with alcohol.

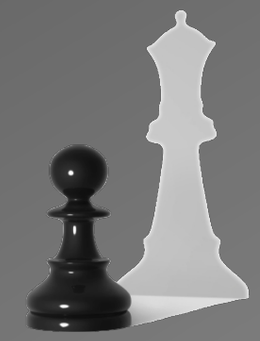
Lagavulin will do nicely.



My apologies to the good consultants, of which I have experienced my share as well



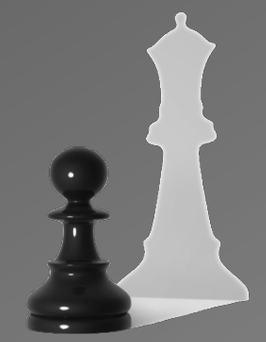
#4.3
The Most Effective Concept I Ever Introduced
(quite recently, actually)



The Problem With Ownership In Complex Landscapes

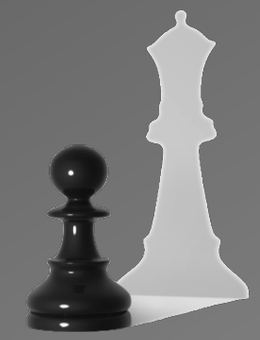
Who is responsible for systems being in control, when those systems exist in a web of dependencies, especially the ‘turtles all the way down’ dependencies of platforms?

In the past, the project was responsible for all *coordinated change* in the landscape within project scope, but in an Agile/DevOps world, we have loosely coordinated product owners.



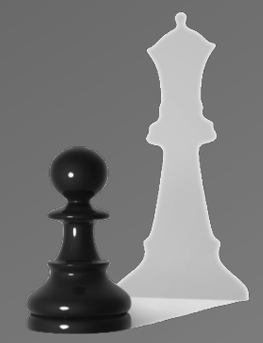
You own what you can change

(note: ‘can’ should be read as ‘may’, we’re not talking ‘reverse engineering’ etc. here)
(alternative: you own what you change)



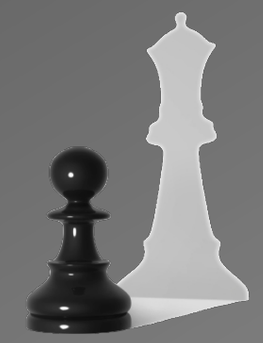
Example: CIA

	Confidentiality	Integrity	Availability
Business Process	3	2	3
Application			
Platform			



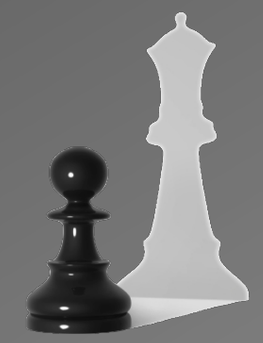
Example: CIA

	Confidentiality	Integrity	Availability
Business Process	3	2	3
Application			
Platform			



Example: CIA

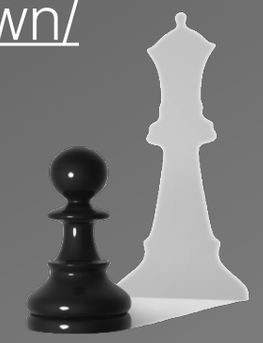
	Confidentiality	Integrity	Availability
Business Process	3	2	3
Application	3	2	**1**
Platform			



Example: CIA

	Confidentiality	Integrity	Availability
Business Process	3	2	3
Application	3	2	1
Platform	Supported: 2 Self: 1	Supported: 2 Self: 3	2

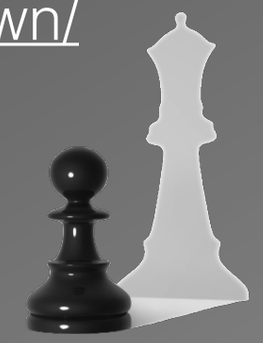
https://ea.rna.nl/2022/04/27/dev-test-production-its-turtles-all-the-way-down/



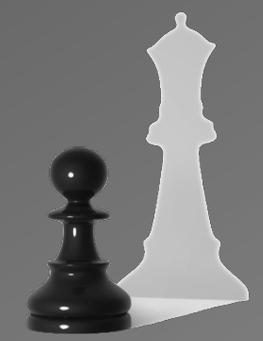
Example: CIA

	Confidentiality	Integrity	Availability
Business Process	3	2	3
Application	3	2	1
Platform	Supported: 2 Self: 1	Supported: 2 Self: 3	2

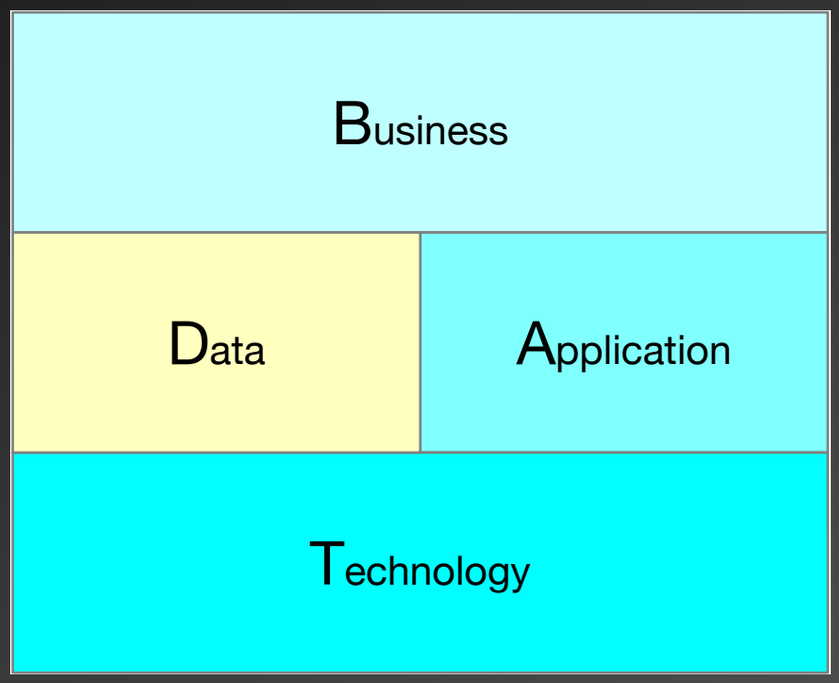
https://ea.rna.nl/2022/04/27/dev-test-production-its-turtles-all-the-way-down/

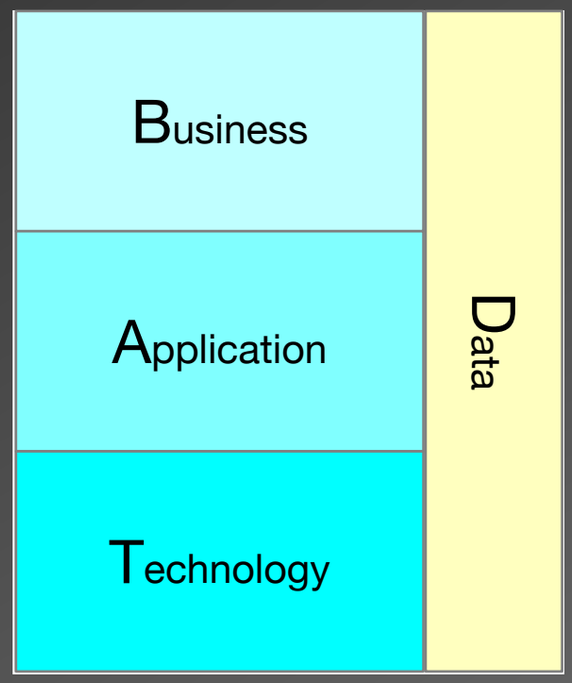


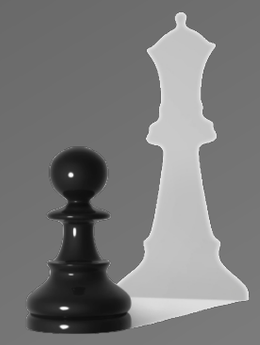
You own what you can change

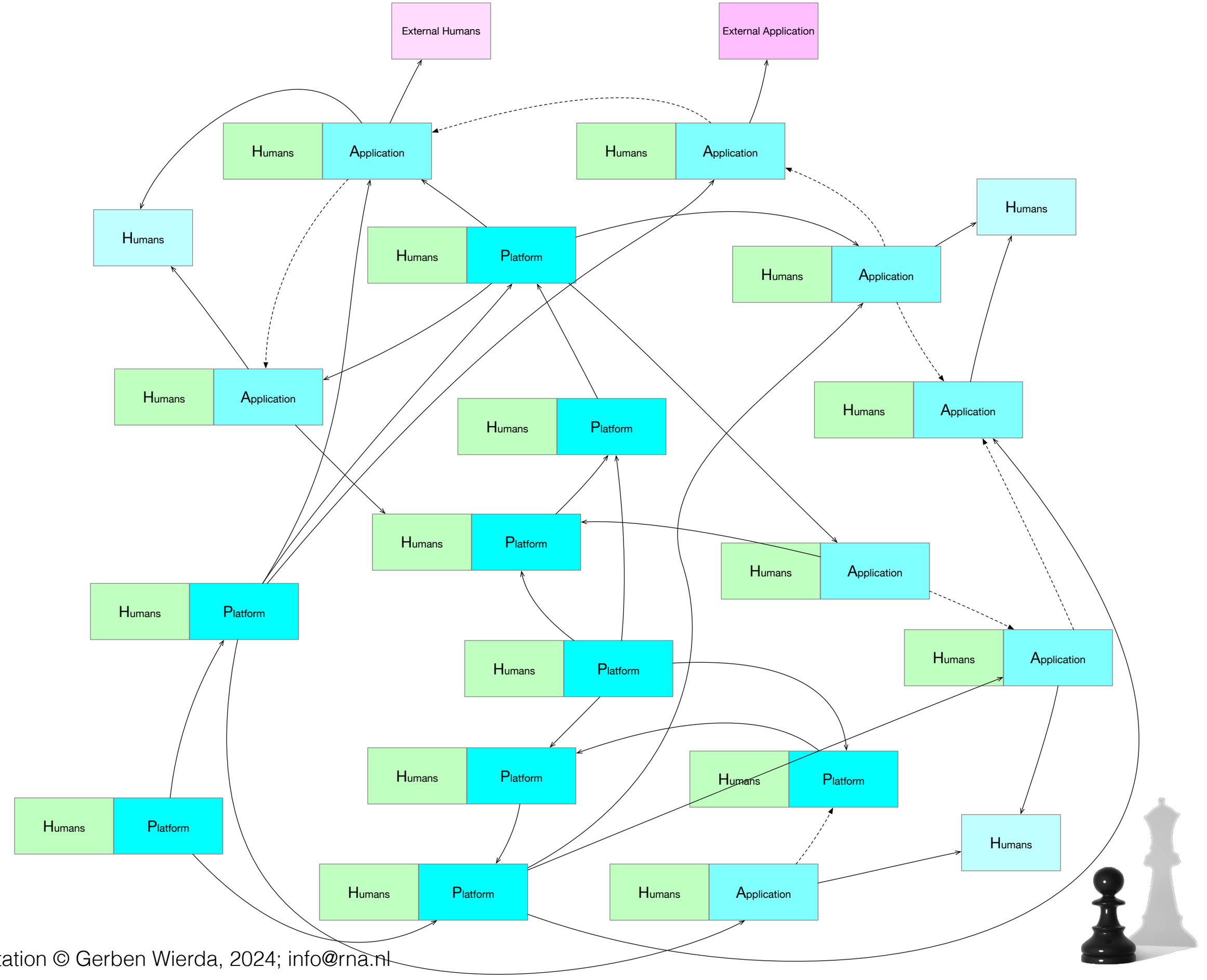


Seductive Oversimplification



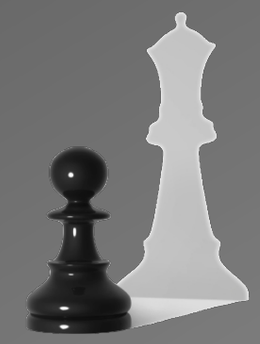






#4.4
The Most Effective Policy I Ever Wrote

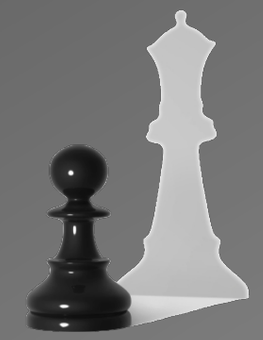
(A practical example of ‘you own what you can change’)



The main problems with Life Cycle Management

Change is hard and LCM feels ‘unproductive’

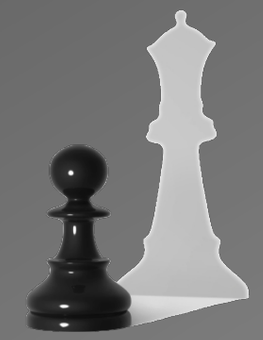
There is almost limitless variation of LCM from external providers (both commercial and open source)

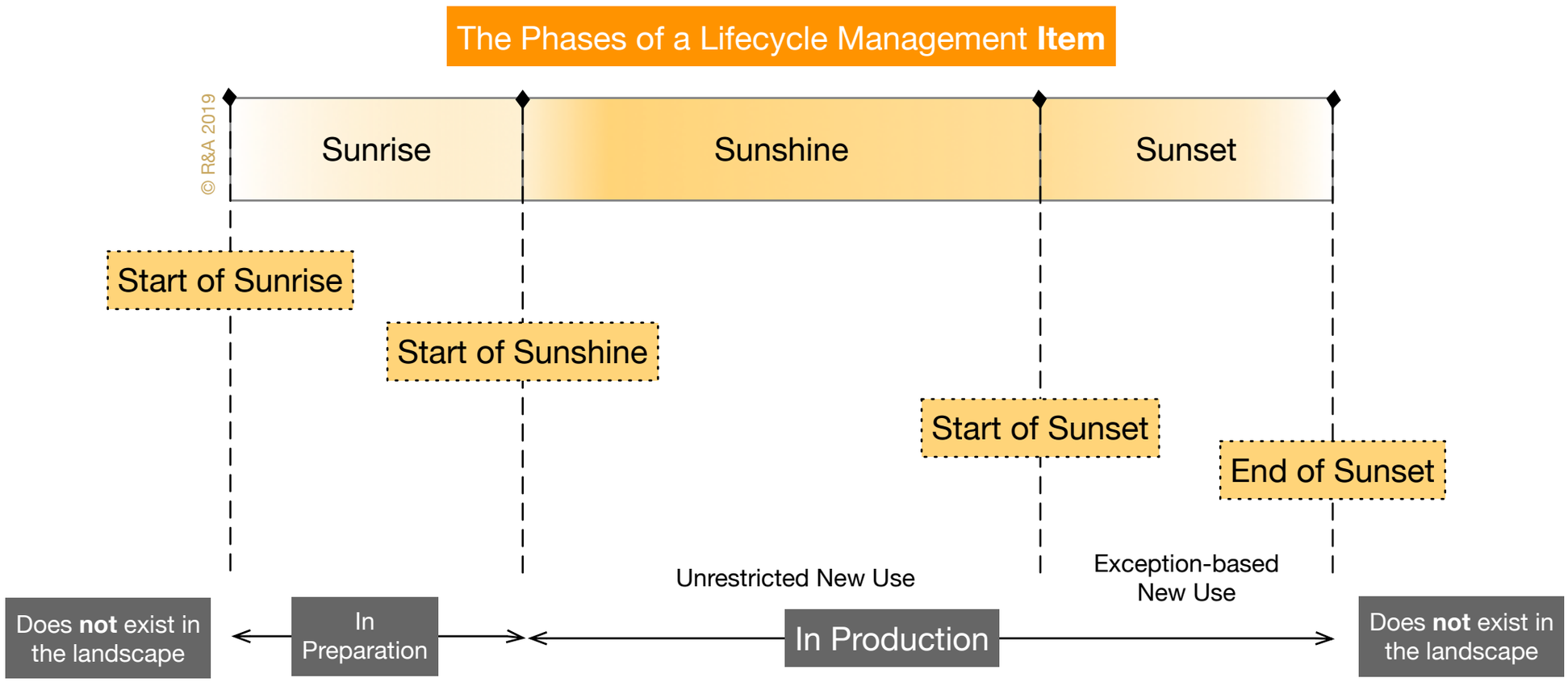


Technically, nothing stops you from running a Windows XP OS today

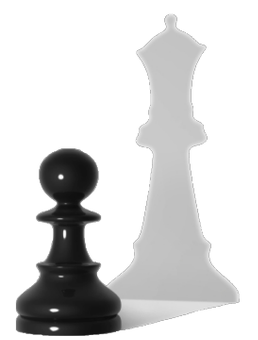
In reality: we decide what is in our landscape (and when), it is our decision

Here too: we own what we can change



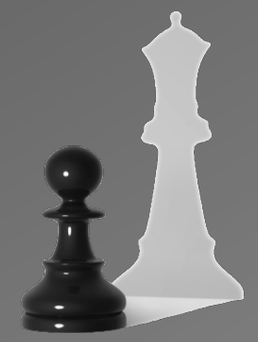


https://ea.rna.nl/2019/08/14/lifecycle-management-let-the-sunshine-in/



There are no ‘exceptions’ in this system with one exception 😀: you can get an exception for ‘new use’ during ‘sunset’

There is no exception for ‘end of sunset’, instead it means we have to move ‘end of sunset’ — after all everything required must stay too: know how, monitoring, backups, security, risk, etc.



End of Vendor Mainstream Support

End of Vendor Extended Support

Vendor Mainstream Support

Vendor Extended Support

Start of Specific Restrictions

The Phases of a Lifecycle Management **Item**

Sunrise

Sunshine

Sunset

Start of Sunrise

Start of Sunshine

Start of Sunset

End of Sunset

Unrestricted New Use

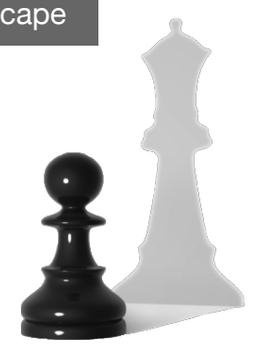
Exception-based New Use

Does **not** exist in the landscape

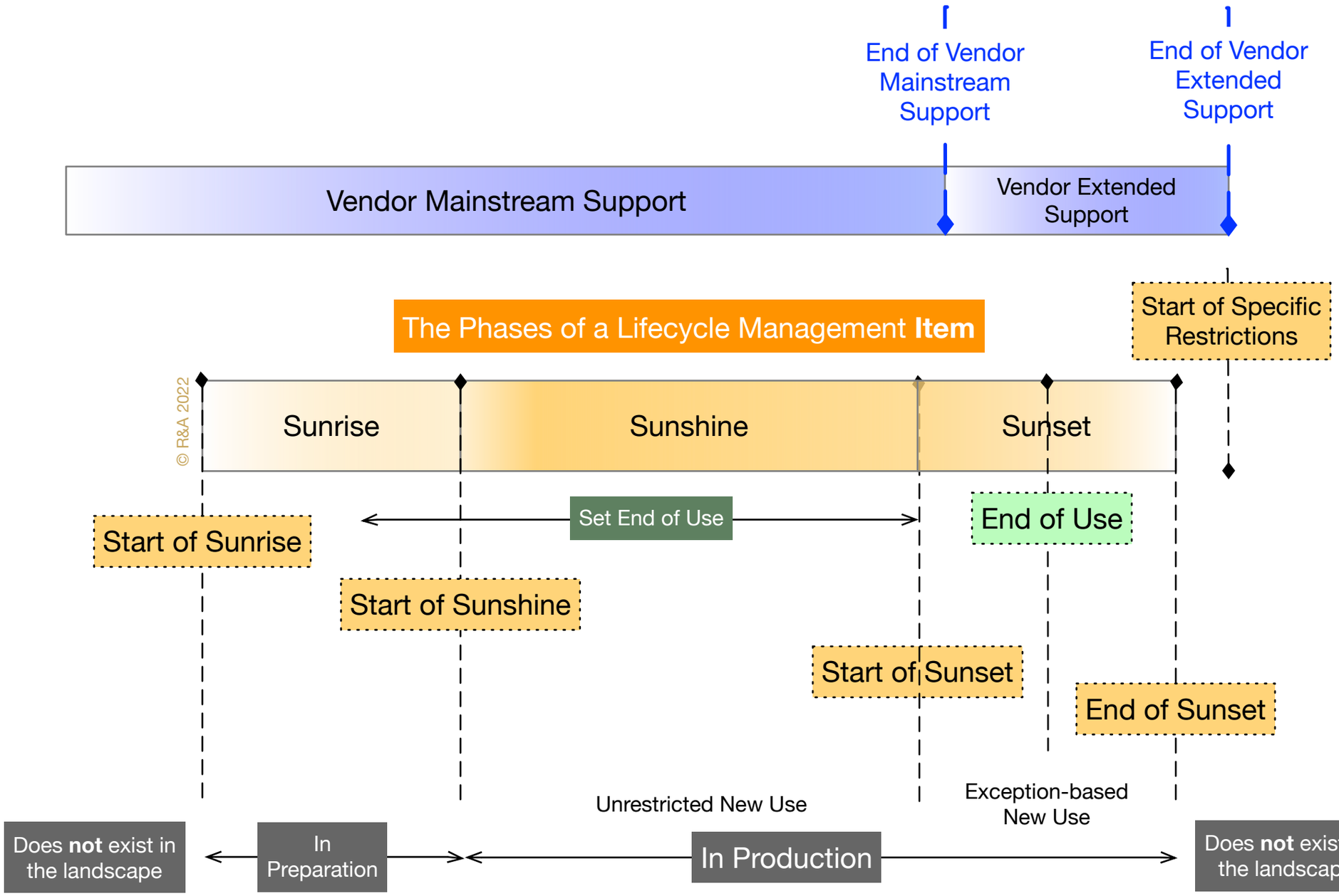
In Preparation

In Production

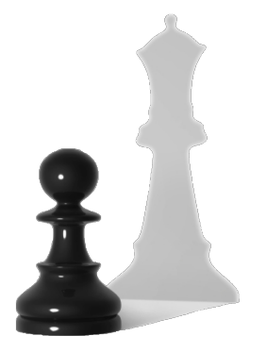
Does **not** exist in the landscape



https://ea.rna.nl/2019/08/14/lifecycle-management-let-the-sunshine-in/

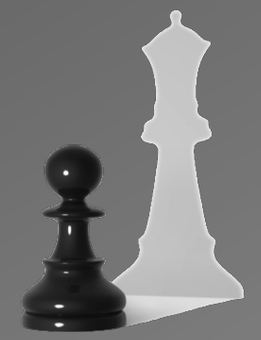


https://ea.rna.nl/2022/11/10/follow-up-the-missing-element-of-sunshine-life-cycle-management/



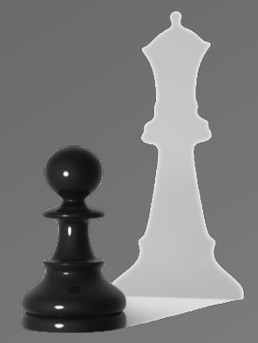
#Last

Epilogue

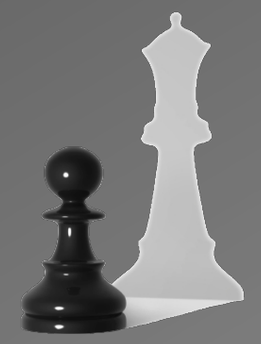


"In every field of inquiry, it is true that all things should be made as simple as possible – but no simpler. (And for every problem that is muddled by over-complexity, a dozen are muddled by over-simplifying)"

– Sydney Harris, Chicago Daily News, Jan. 2, 1964



The world is not $\mathbb{Q}$ it is $\mathbb{R}$



Your Turn

(I am outta here…)