

Governing Agile Delivery

Dan North

@tastapod

What is governance?

1. What are we spending our money on?
2. How is it going?

Governance is about managing investment risk

How does delivery usually work?

Initiation

- Definition and scope
- Tendering and procurement
- Budget approval

Execution

- Design
- Development
- Testing



Planning

- Work breakdown
- Resourcing
- Scheduling

Release

- Deployment
- User acceptance
- Transition to live

How do we govern this?

Initiation

- Proposition review
- ITT
- Contracts
- Budget review
- Payment schedules

Execution

- Steering group(s)
- Stage gate reviews
- Documentation
- Sign-offs
- Hand-overs



Planning

- Project review board
- PMO sign-off
- Financial controls
- Review schedule

Release

- Change Advisory Board
- Release documentation
- Go/No-go
- User sign-off

Why do we do it like this?

Each stage gate gives us assurance

- Because there is usually a lot at stake!

We make **big investments**

- or as some may say, big bets

We plan and budget on an **annual basis**

*We often make **multi-year commitments***

Multi-year commitments are risky!

We fear things going wrong:

- Cost overruns
- Schedule overruns
- Delivering the wrong thing
- Operational instability
- Supplier instability

We believe we can front-load the risk

We just need to plan in enough detail. What could possibly go wrong?

Governance models are designed top-down

Portfolio

- Steering group
- Exec committee



Programmes

- PMO
- Programme board



Projects

- Project manager
- Project sponsor

Agile methods are bottom-up

Team-scale: 7 ± 2 peopleTM

Interdisciplinary, or
cross-functional

Working in small time boxes

Releasing software frequently

Collaborating closely with the user

Planning in small time boxes,
based on feedback

Funding is usually incremental

*Agile is about teams delivering **small slices** and acting on **feedback***

Agile methods are bottom-up

CEO



Team

Team

Team

Team

Team

Team

Team

Team

Agile* Adoption* Patterns* - Richard Durnall

You are here 

1. The people break
2. The tools break
3. The governance breaks
4. The customer breaks
5. The money breaks
6. The organisation breaks

** Not agile, not adoption, not patterns, but still...*

Agile doesn't have an opinion about large programmes

Agile methods emerged to address delivery in the small

Delivery at scale is a substantively different problem

- $80 \neq 8 \times 10$

Multiple agile teams can collaborate on the same programme of work

- Needs strong technical direction
- Needs strong product management
- Needs strong (Lean) programme management

Governing principles: "Freedom within a framework"

Governance is about managing
investment risk

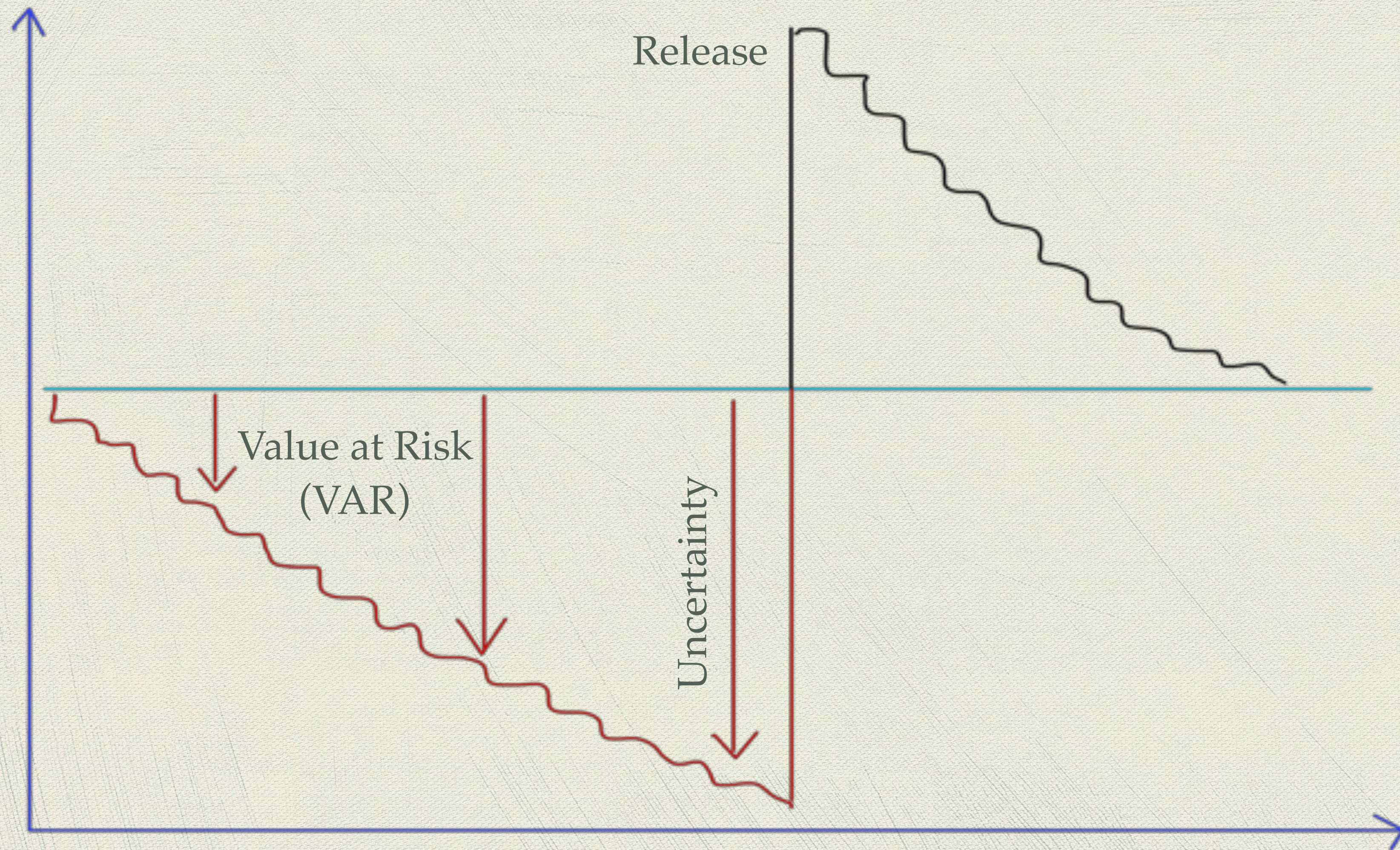
Return on Investment is a blunt instrument

It says nothing about *when* the return occurs, or *how much* is at risk

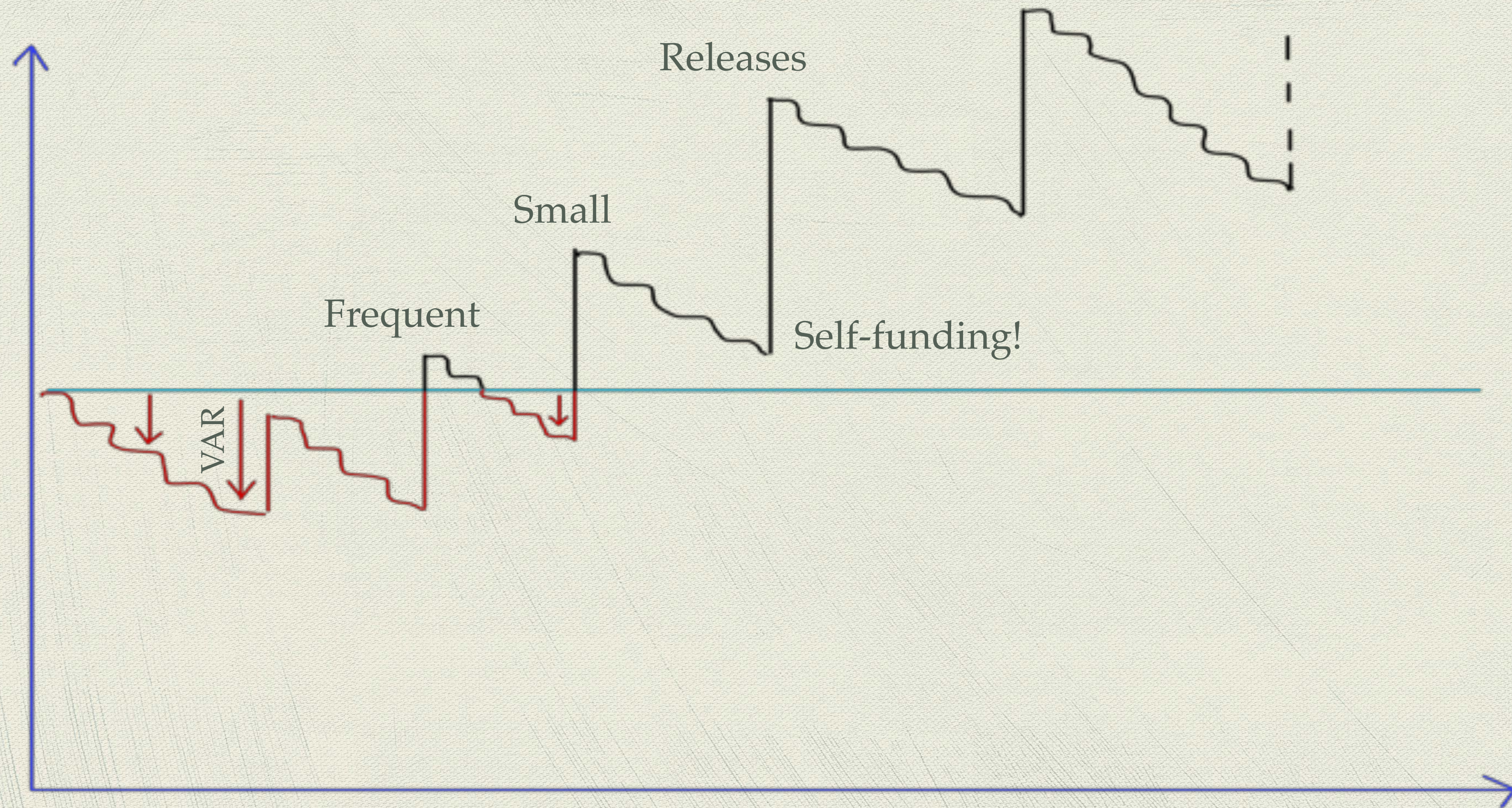
Risk-Adjusted Return on Capital (RARoC) is more useful

Agile delivery has (initially) **worse RoI**, but much better **RARoC**

Risk profile of traditional delivery



Risk profile of agile delivery



We need to rethink our underlying assumptions

Flow of value is more important than utilisation of workers

Busyness is not an indicator of progress, results are

Feedback is vital to reducing uncertainty

Lead time is the only game in town

Lead time is the only game in town

Reduce batch size

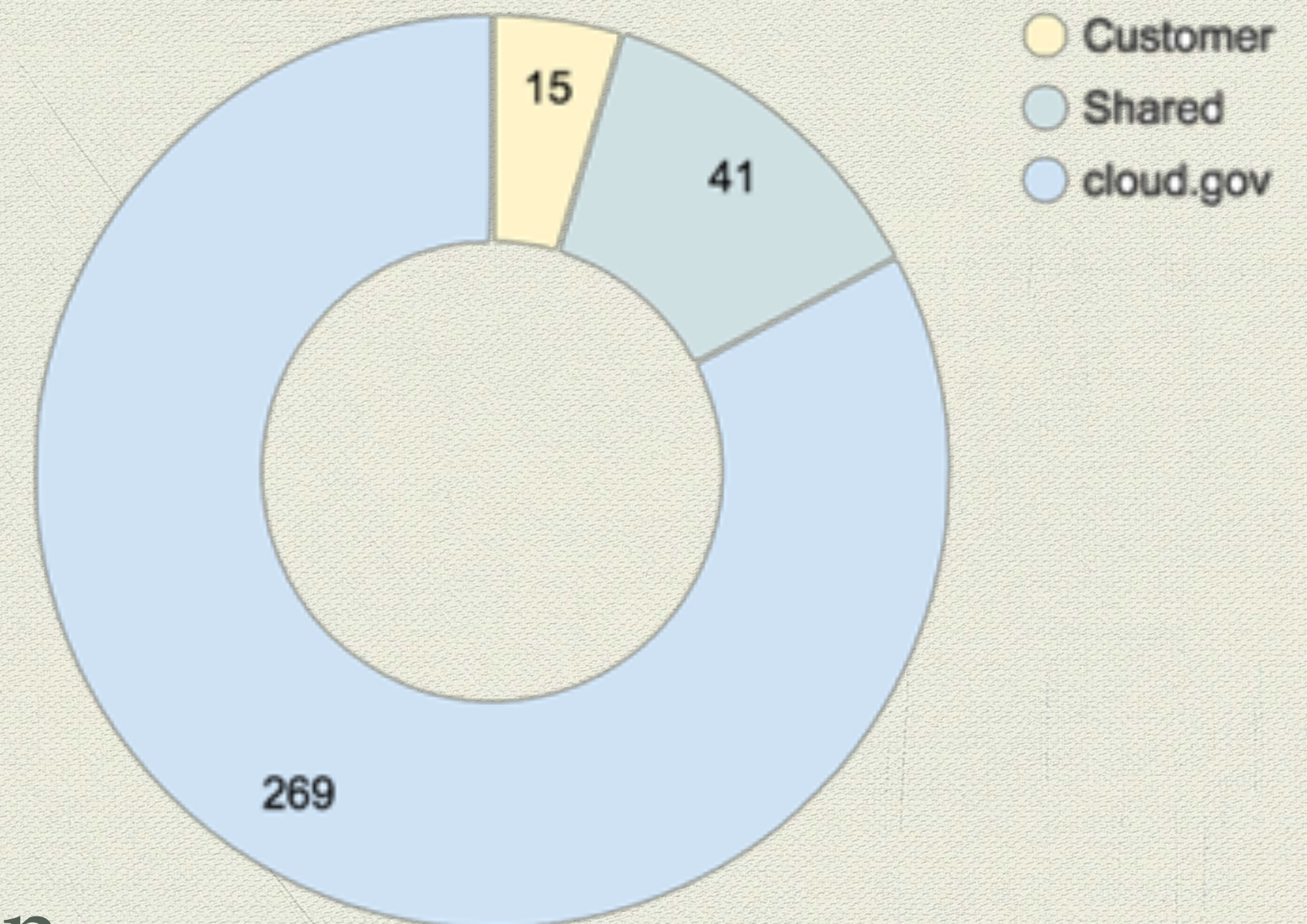
- which reduces variability to boot!

Keep work within the team

- including inspections and reviews
- including specialist knowledge and skills

Identify information you need to govern, then

- **VESSA**: Visualise, Eliminate, Simplify, Standardise, Automate
- until your process becomes self-evidencing



Wrapping up

Governance is about managing investment risk

Iterative delivery methods reduce risk and surface uncertainty

But your governance needs to change to benefit from this

All software development carries uncertainty

You can't eliminate uncertainty, but you can surface it early

Thanks for listening

<https://dannorth.net>

dan@dannorth.net

[@tastapod](#)

[@tastapod](#)