

Cloud New Trend

Sanchit Jain
24th Feb, Wednesday
10:00 AM to 12:00 PM



Speakers



Sanchit Jain

Lead Architect - AWS at Quantiphi
AWS APN Ambassador

FOLLOW ME



New Requirements in Industry





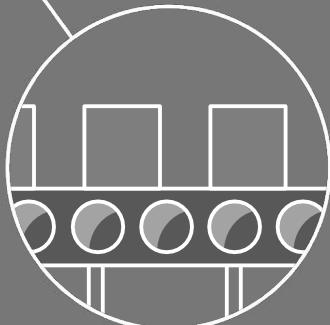
Marketing
analytics



Employees at work



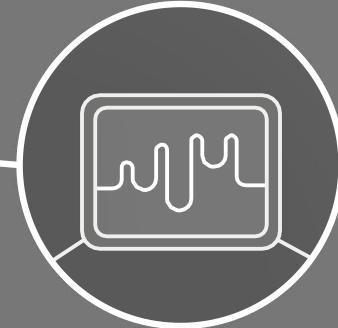
Sales channels



Factories + supply
chain



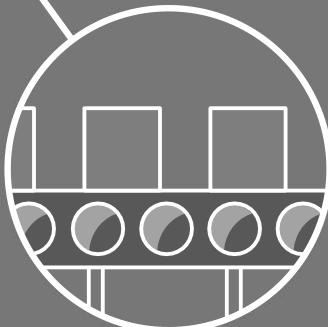
Marketing
analytics



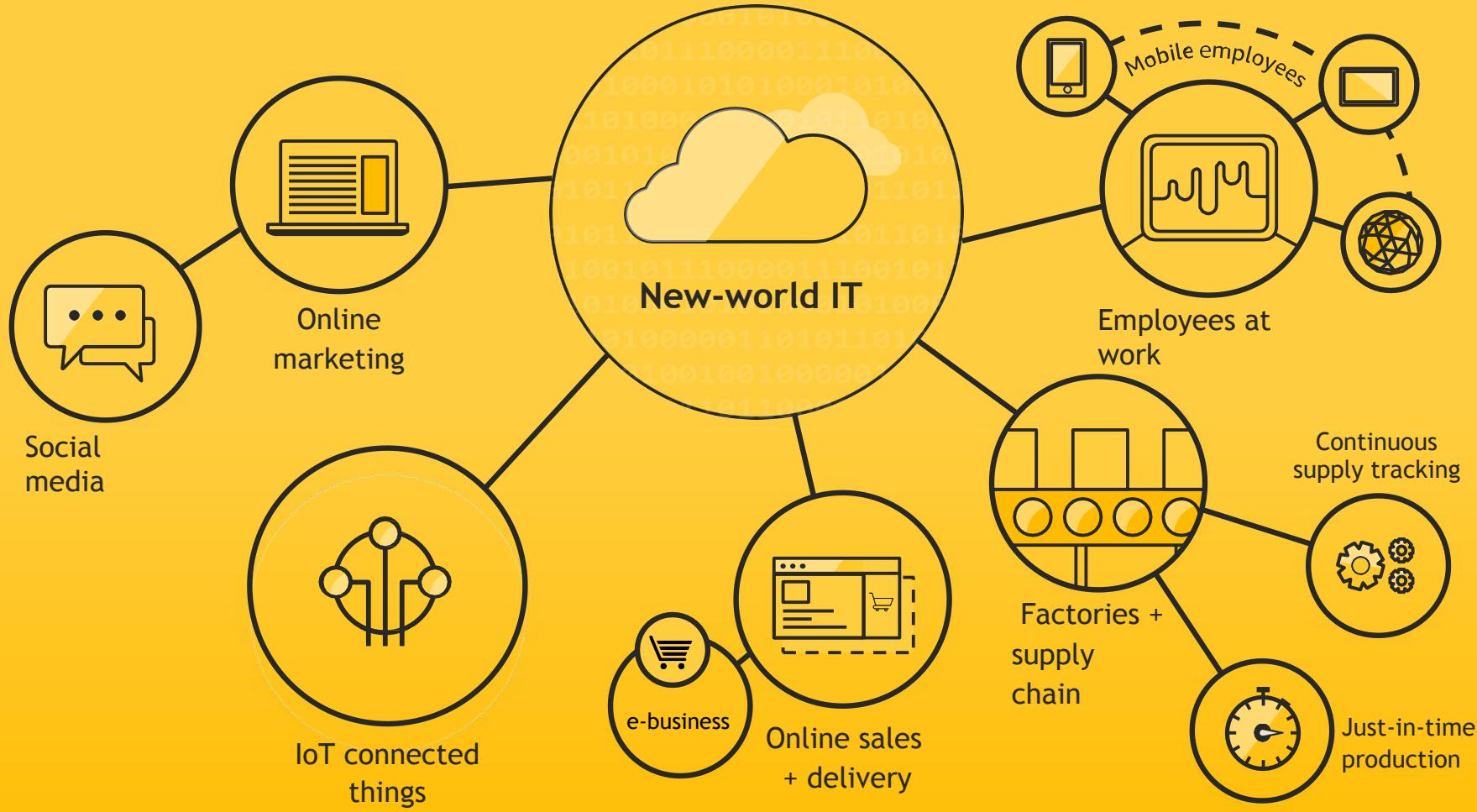
Employees at work



Sales channels



Factories + supply
chain



New Needs!



- Personalization
- Customer analytics
- New channels direct to customer
- More things, more scale, rapid change

Blocker for Innovation



Blockers for innovation



Culture



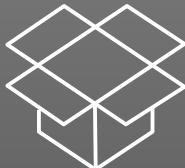
Leadership
systems and
feedback

Skills



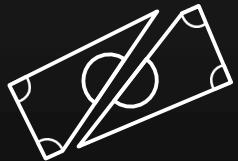
Training
and
compensation

Organization



Silos project to
product

Risk



Finance and board
level concerns

Blockers for innovation



Culture



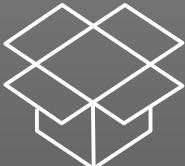
Leadership
systems and
feedback

Skills



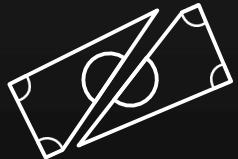
Training
and
compensation

Organization



Silos project to
product

Risk



Finance and board
level concerns

Leadership systems and feedback problems



- Centralized, slow decision-making
- Lack of trust
- Inflexible policies and processes



So how we can break it?

Culture



- Values are what we value
- High performance
- Freedom and responsibility
- Context, not control
- Highly aligned, loosely coupled
- Pay top of market
- Promotions and development

Blockers for innovation



Culture



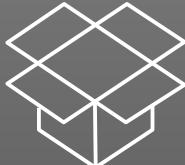
Leadership
systems and
feedback

Skills



Training
and
compensation

Organization



Silos project to
product

Risk



Finance and board
level concerns

How we can motivate?

Training and compensation



- Train existing staff on cloud tech
- Fund pathfinder teams
- Be prepared to create incentives to keep the best people after training!

Blockers for innovation



Culture



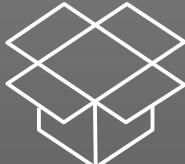
Leadership
systems and
feedback

Skills



Training
and
compensation

Organization



Silos project to
product

Risk

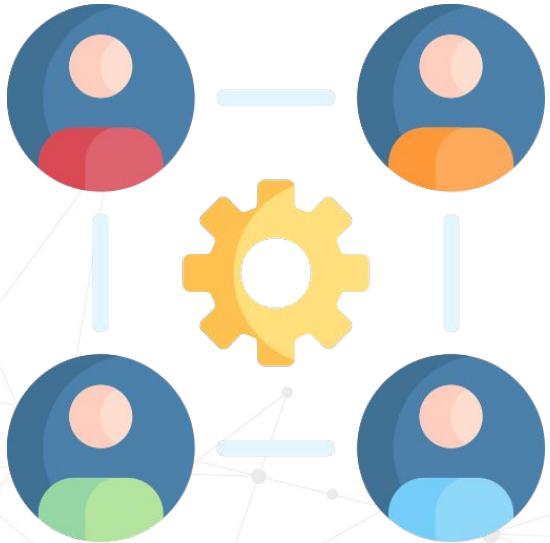


Finance and board
level concerns



How we can improve the bonding?

Move from projects to product teams



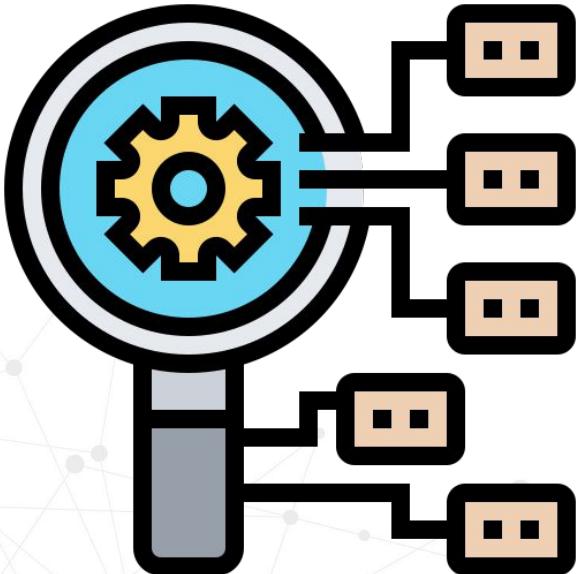
- Long-term product ownership
- Continuous delivery
- DevOps and “run what you wrote”
- Reduce tech debt and lock-in

Integrate business with DevOps



- Business - *Budget, headcount, goals*
- Product - *Customer input, roadmap*
- Development - *Continuous delivery of features*
- Operations - *Automated global support*

Retrospective and learning



- Monday – Service team
 - Review last week's operations dashboards
 - Review last week's revenue/growth/goals
- Tuesday – Groups of services
 - Roll-up reviews
- Wednesday – CEO/VP-level view of everything
 - Review all operations, spin the wheel*, learnings

Blockers for innovation



Culture



Leadership
systems and
feedback

Skills



Training
and
compensation

Organization



Silos project to
product

Risk



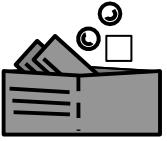
Finance and board
level concerns

Finance – Capex vs. opex



- Capitalized data center to expensed cloud
- Capitalized development, expensed operations, to combined DevOps
- Plan ahead, don't surprise the CFO or your shareholders!

Finance – Board-level concerns



Compensation
policy



Executive
succession



Oversight of
finance



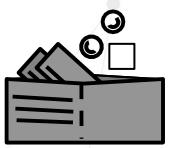
Oversight of
risk



Oversight of
strategy

Finance – Board-level concerns

All of these affect the ability to innovate and build long-term value



Compensation
policy



Executive
succession



Oversight of
finance



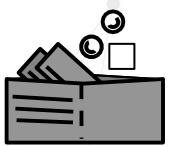
Oversight of
risk



Oversight of
strategy

Finance – Board-level concerns

All of these affect the ability to innovate and build long-term value



Compensation
policy



Executive
succession



Oversight of
finance



Oversight of
risk



Oversight of
strategy

Many board best practices *reduce innovation*

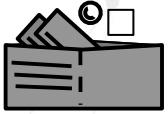


So what's the connection?

Connect the dots...

Finance – Board-level patterns

Short term focus



Compensation policy



Executive succession

Capitalization issue



Limited investment



Oversight of finance

Non-technical founder



Oversight of risk

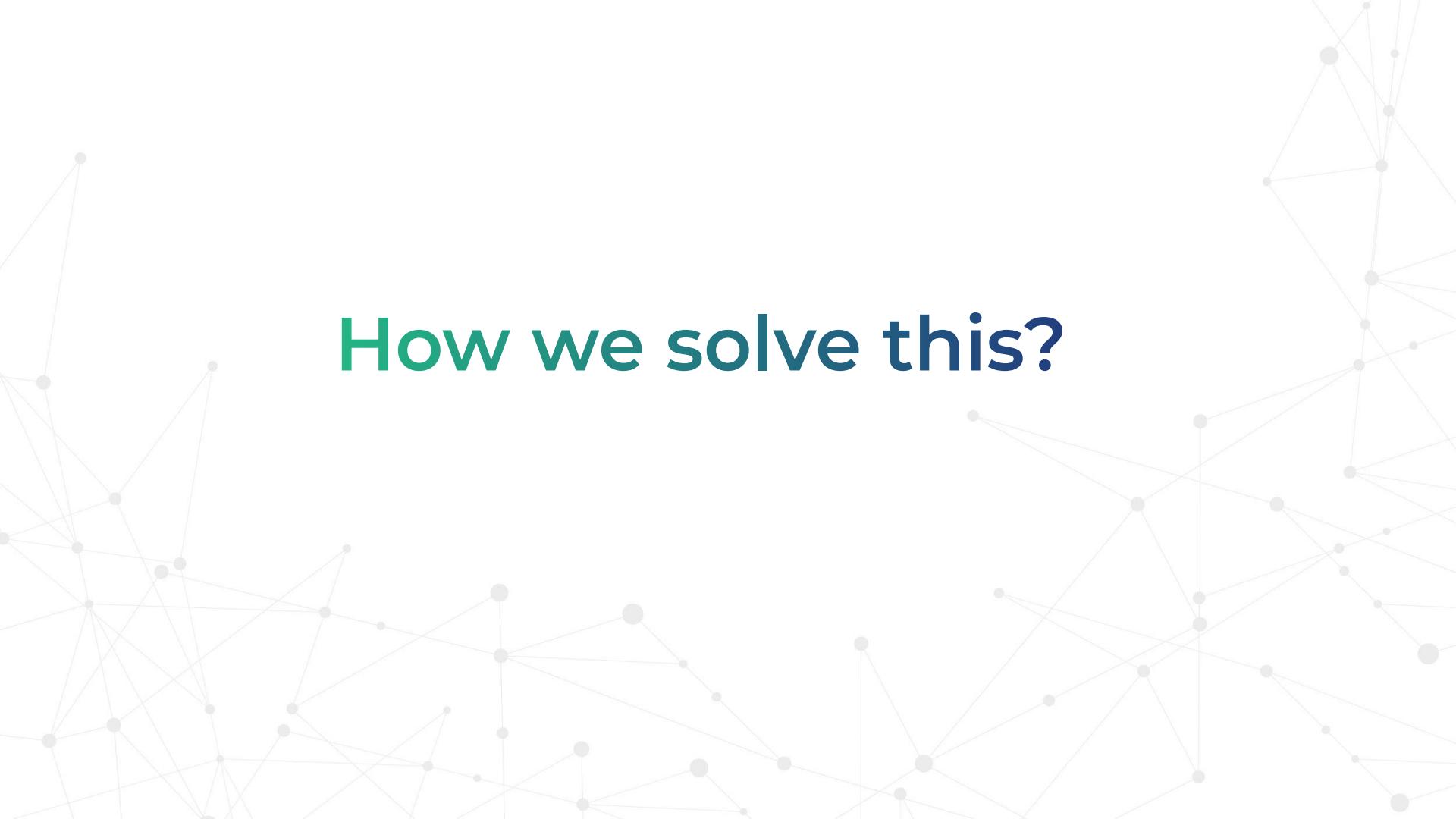
Risk adverse

NO FAILURES

But PLEASE be
more
innovative



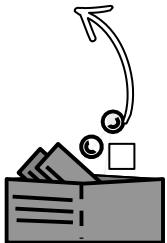
Oversight of
strategy



How we solve this?

Finance – Successful board-level patterns

Focus on fewer,
higher paid staff
and more ownership



Compensation
policy

Manage capex/opex
expectations



Executive
succession

Invest in
differentiators



Oversight of
finance

Get out of the way
of innovation



Oversight of
strategy



Oversight of
risk

Smaller projects with
incremental returns

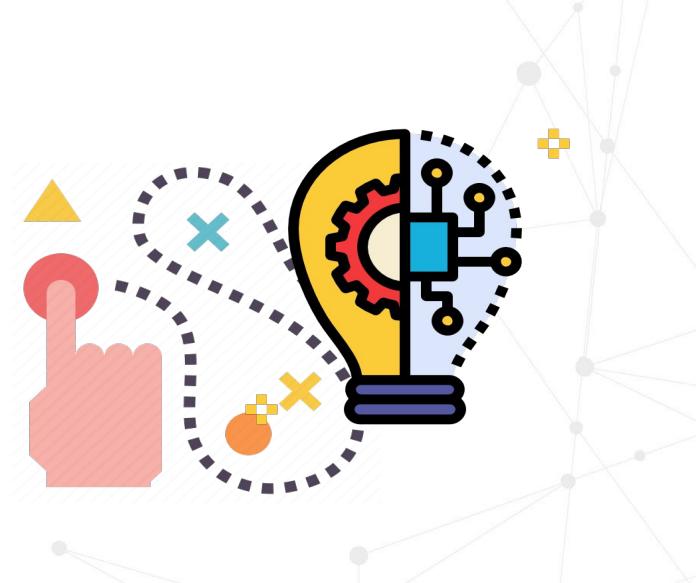
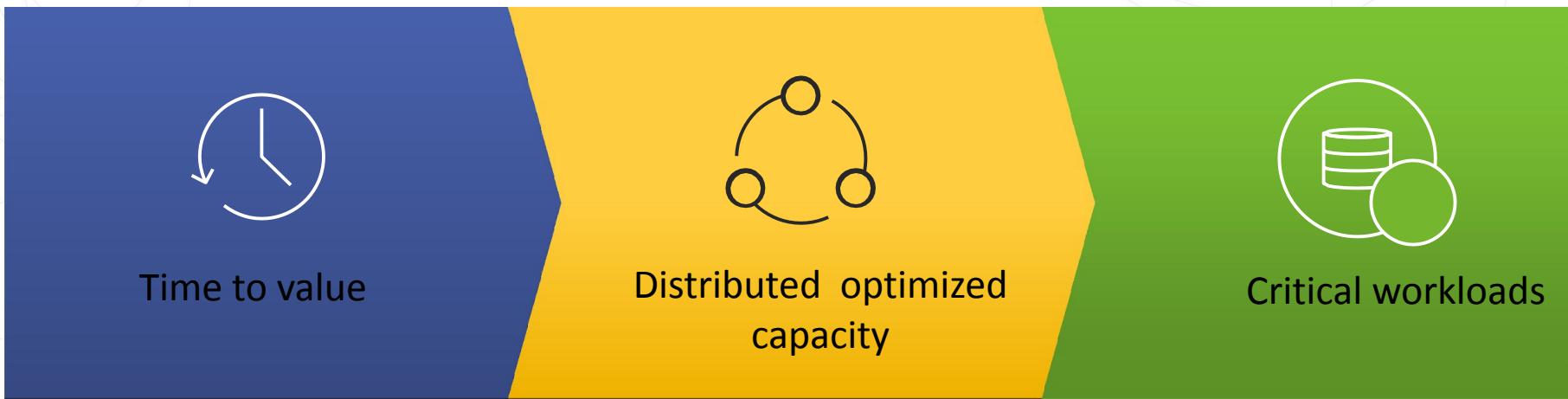
Time to value

Include technical
competence as
a must have

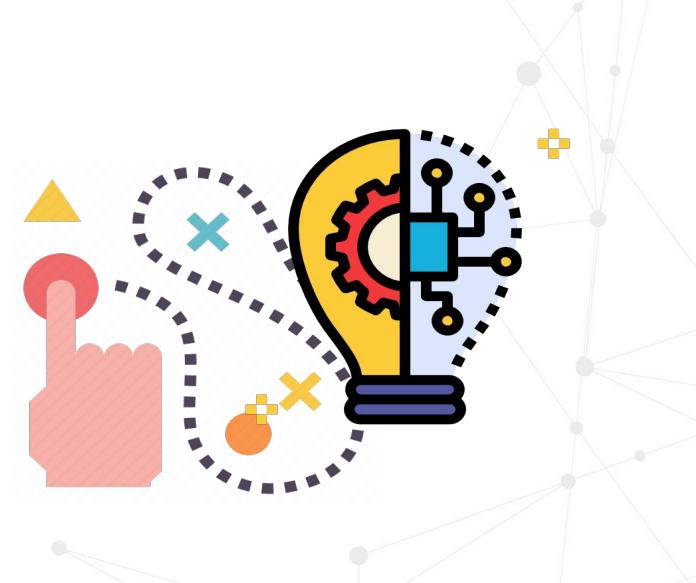
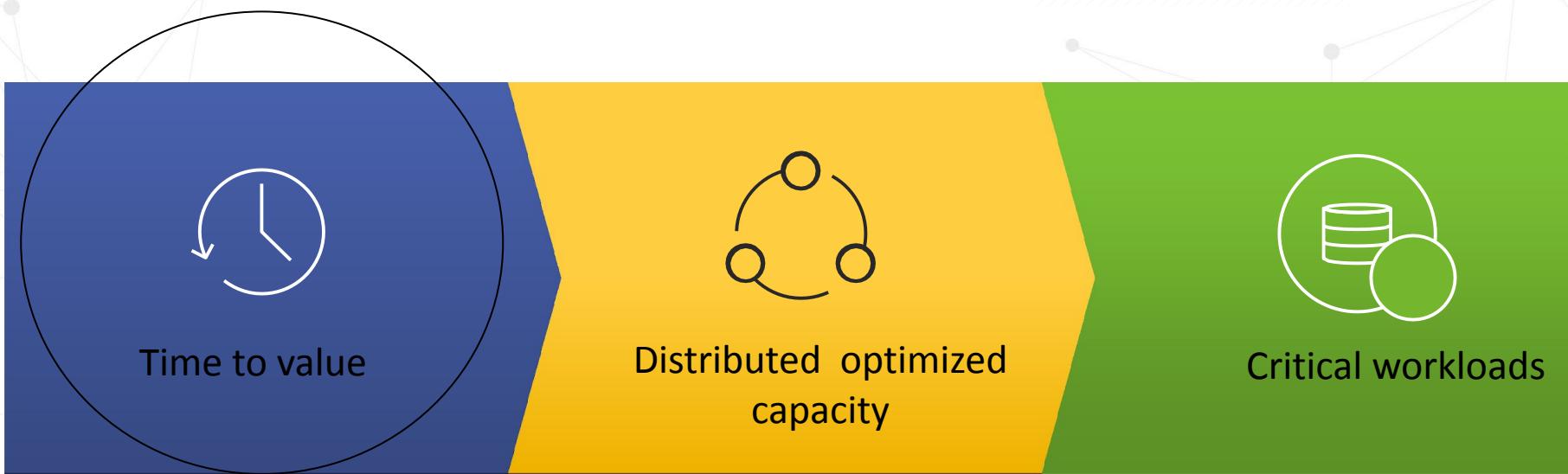
Pathway for Innovation



Pathway for innovation



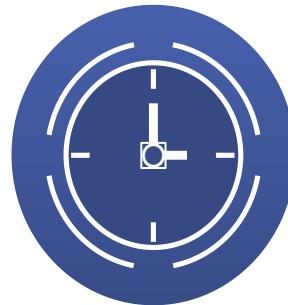
Pathway for innovation



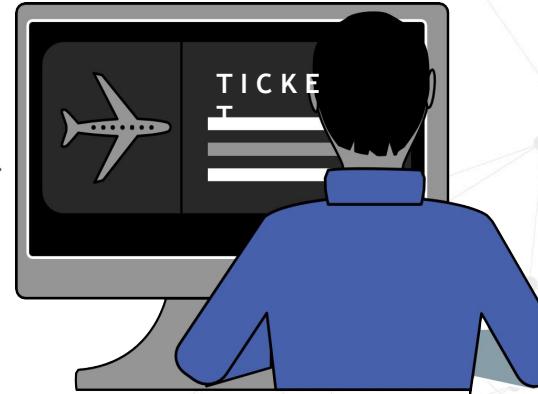
What is the fundamental metric for innovation?



Do some work



How long?



Value to a customer

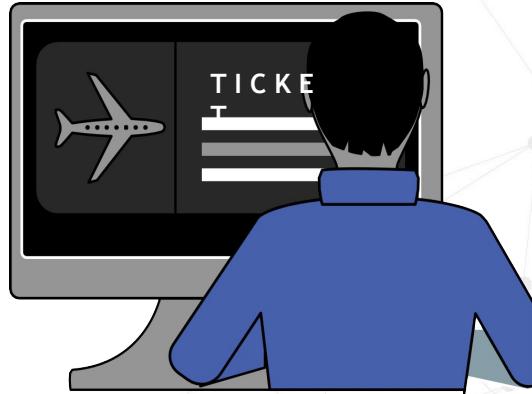
What is the fundamental metric for innovation?



Do some work



Months?



Value to a customer

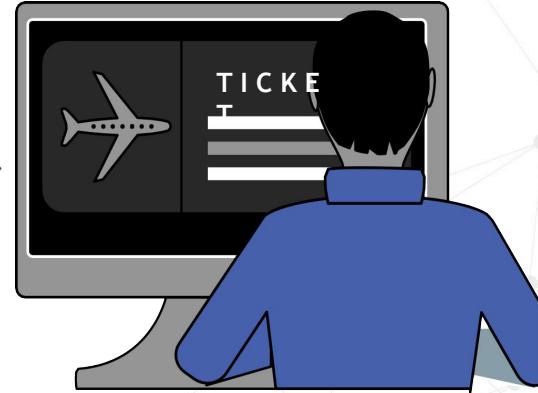
What is the fundamental metric for innovation?



Do some work



Days?

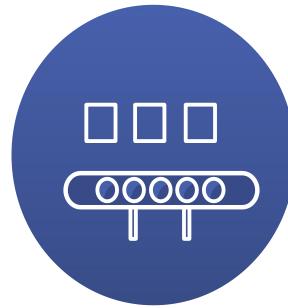


Value to a customer

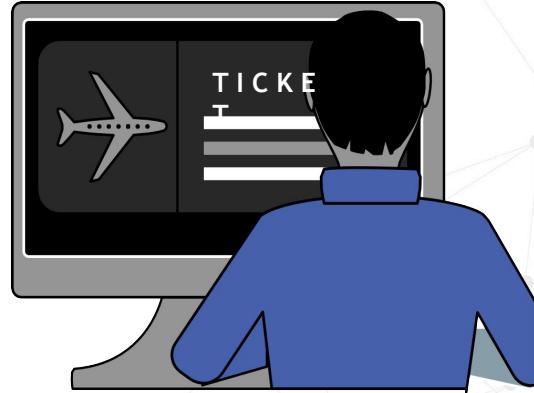
What is the fundamental metric for innovation?



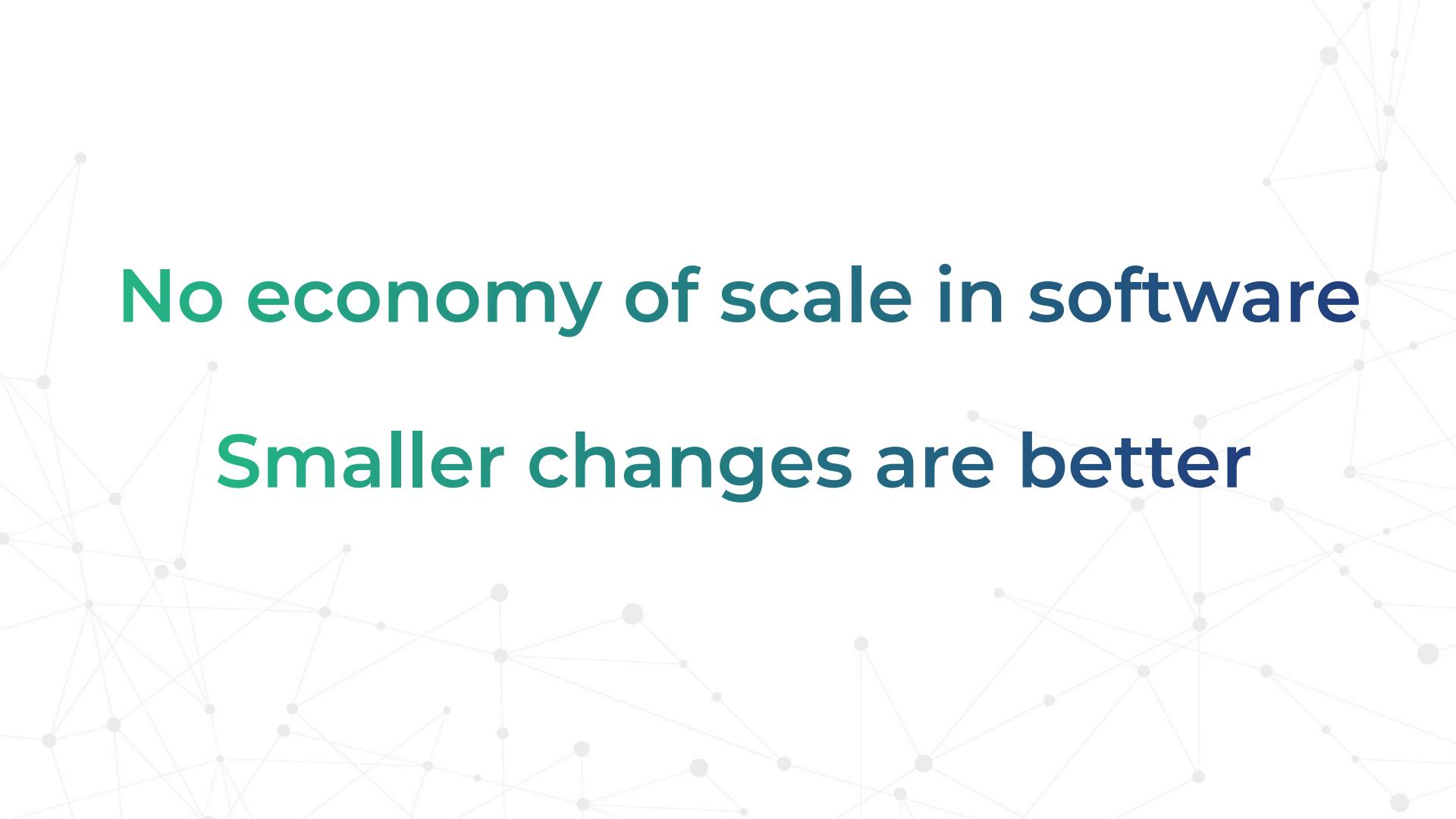
Do some work



Minutes?



Value to a customer



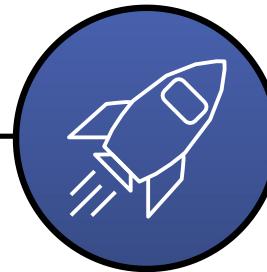
No economy of scale in software

Smaller changes are better

Lots of small changes



Automated
continuous-delivery
pipeline

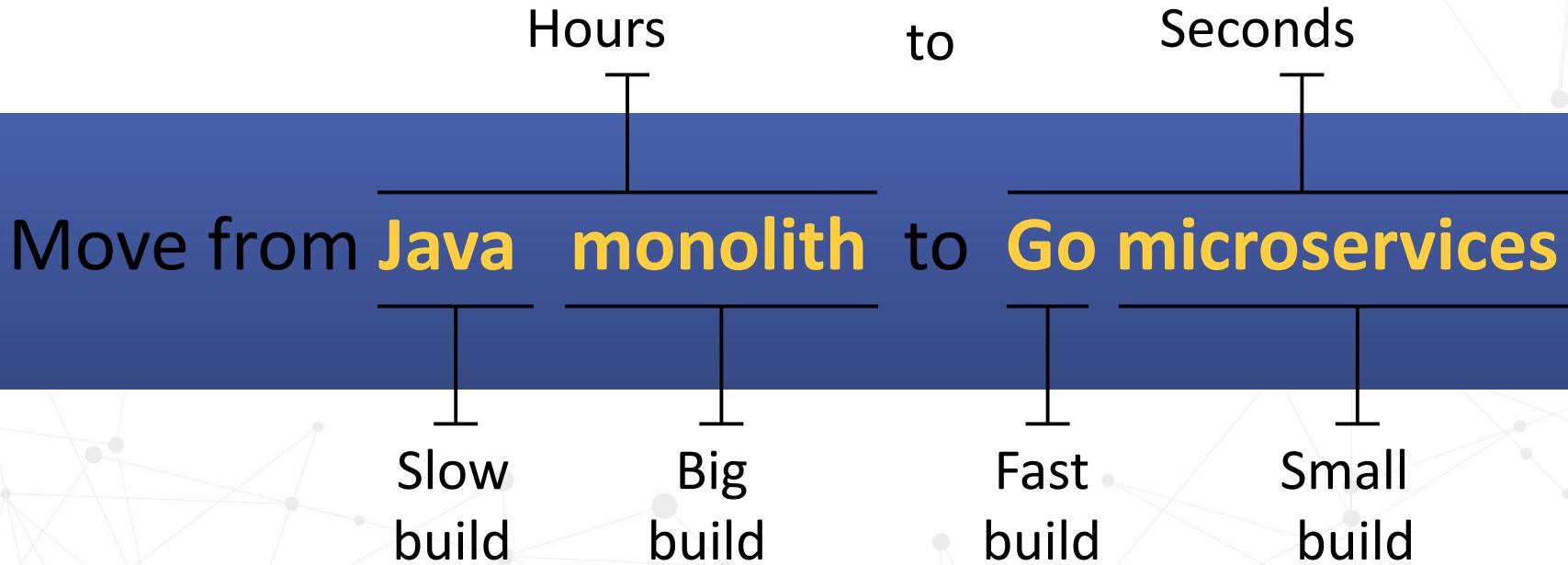


Tagging
feature flags,
A/B tests



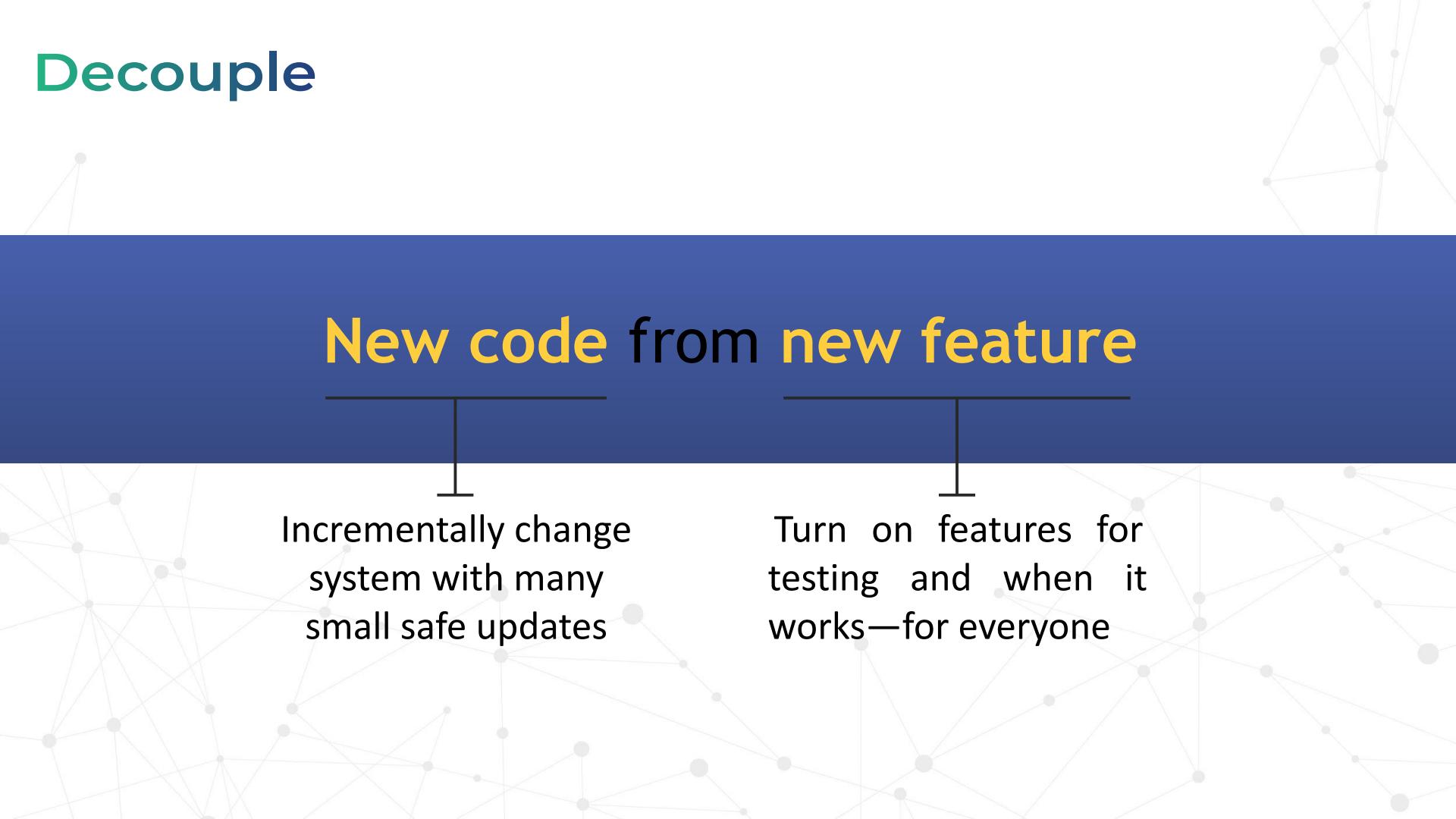
Rapid
cheap
builds

Lots of small changes

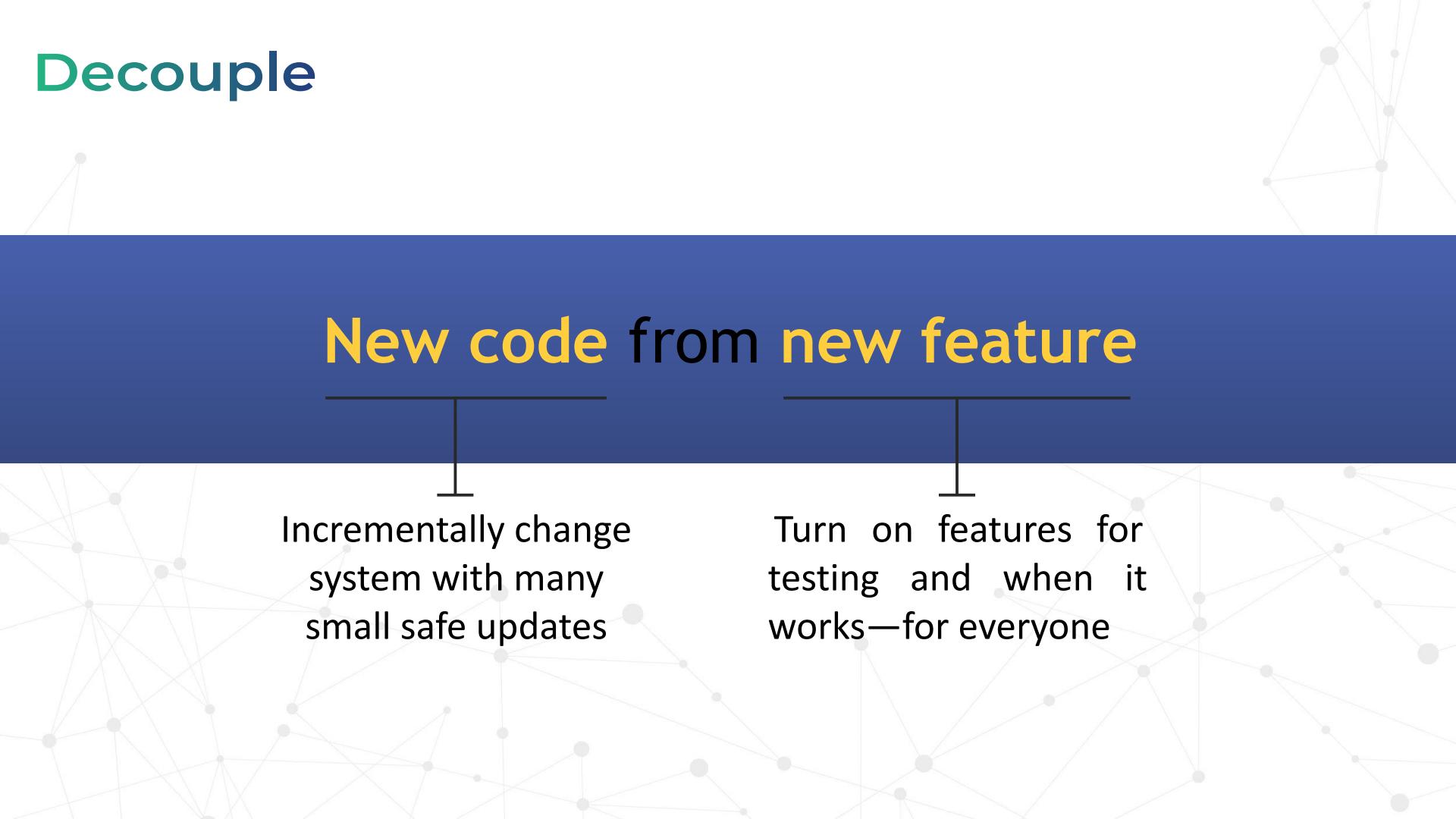


Decouple

New code from new feature

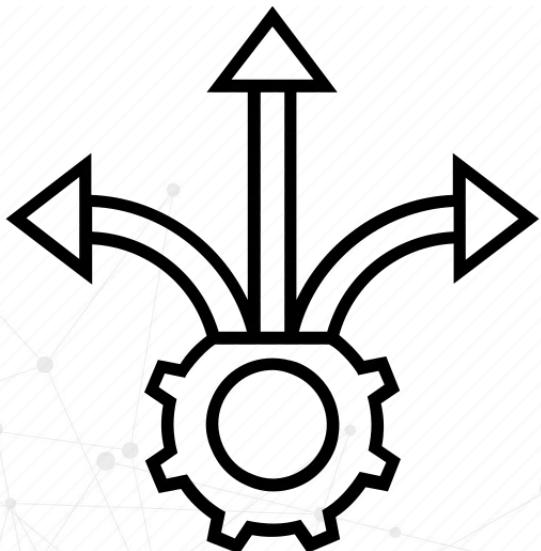


Incrementally change system with many small safe updates



Turn on features for testing and when it works—for everyone

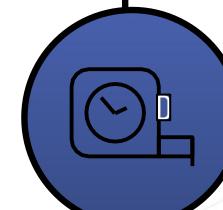
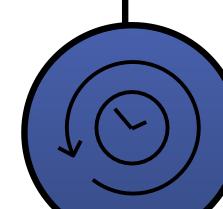
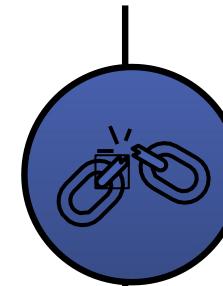
Change one small thing at a time



Easier to

Easier to

Easier to



tell if it breaks

roll back to
previous version

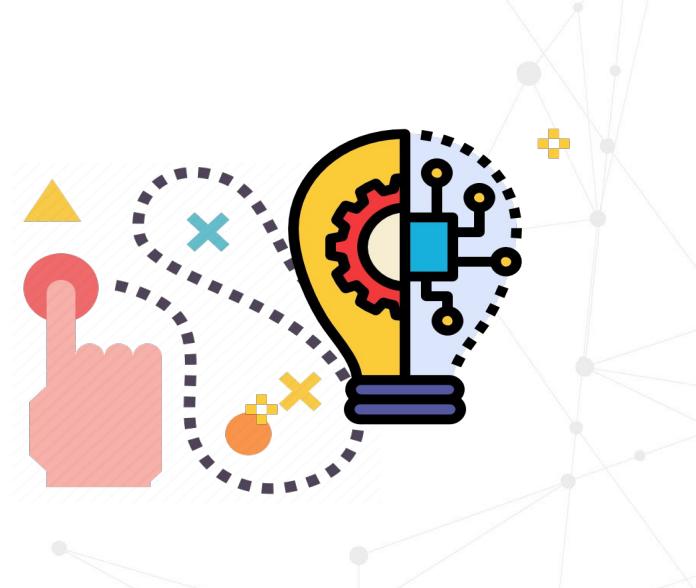
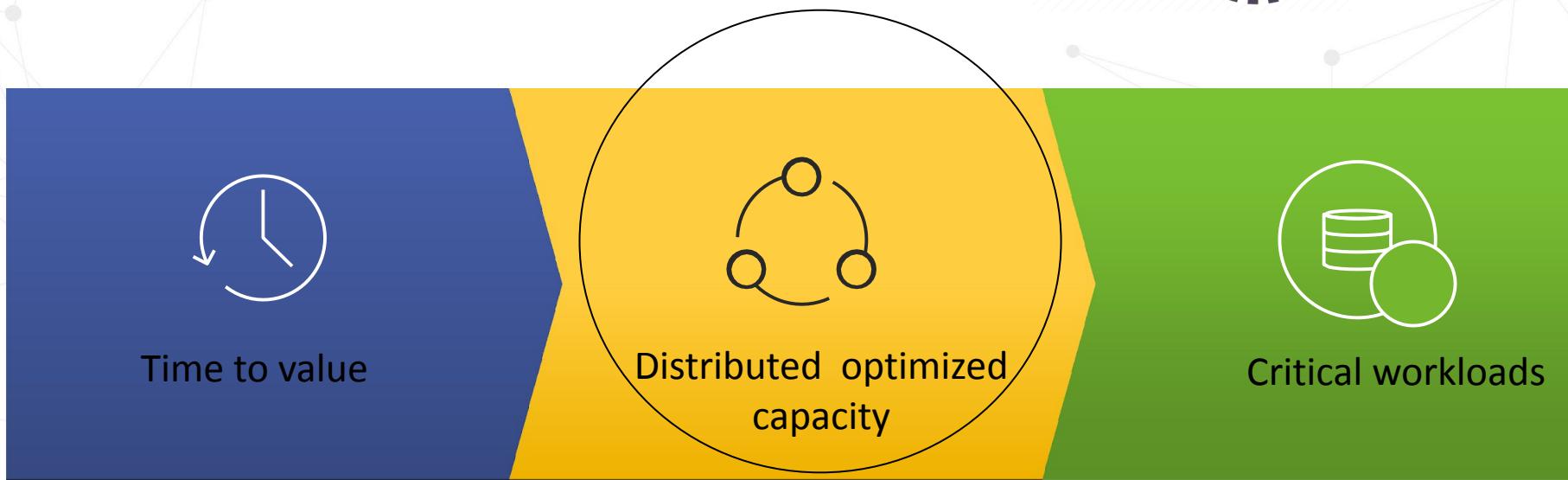
measure time
to value

Benefits of small changes

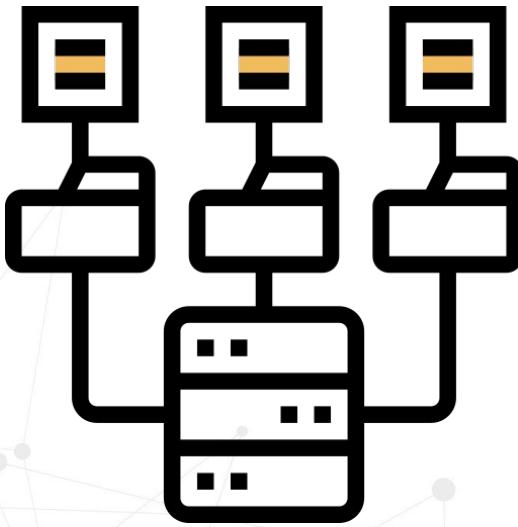


- Less risk
- Faster problem detection
- Faster repair
- Less work in progress
- Less time merging changes
- Happier developers
- Faster flow

Pathway for innovation

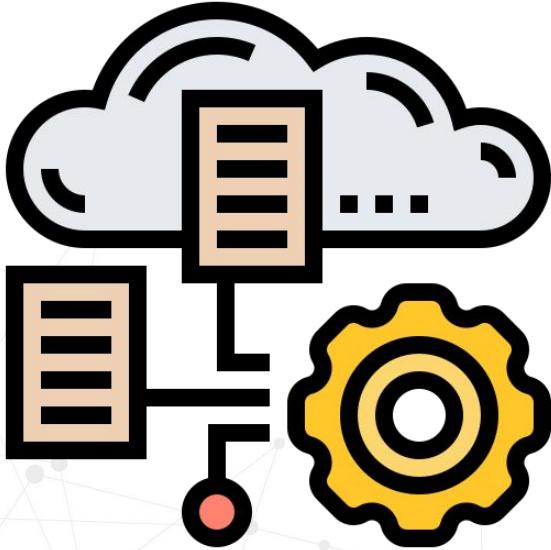


Distributed optimised capacity



- Highly scaled
- Distributed for availability
- Cost-optimized high utilization
- Cloud-native architecture

Cloud-native principles



- Highly scaled
- Distributed for availability
- Cost-optimized high utilization
- Cloud-native architecture

Traditional Development



Design a prototype



Carve from modeling clay



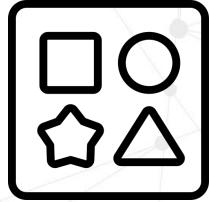
Make molds



Produce injection-molded parts



Assemble parts

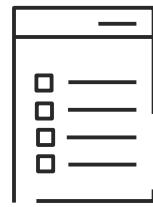


Sell finished toy

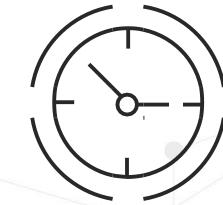
Rapid development



Big bag of blocks

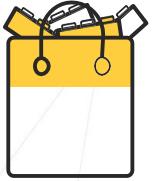


Instructions



A few hours

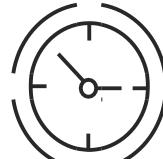
Rapid development



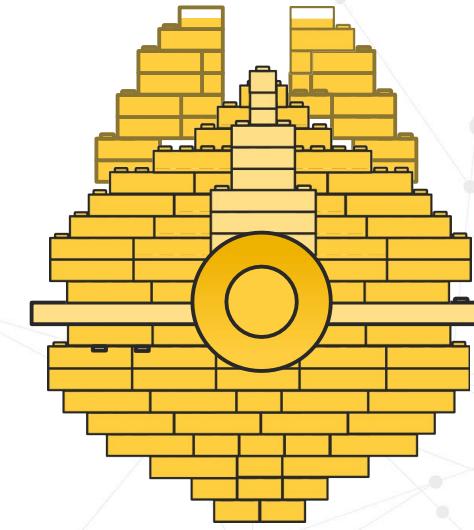
Big bag of
blocks



Instructions

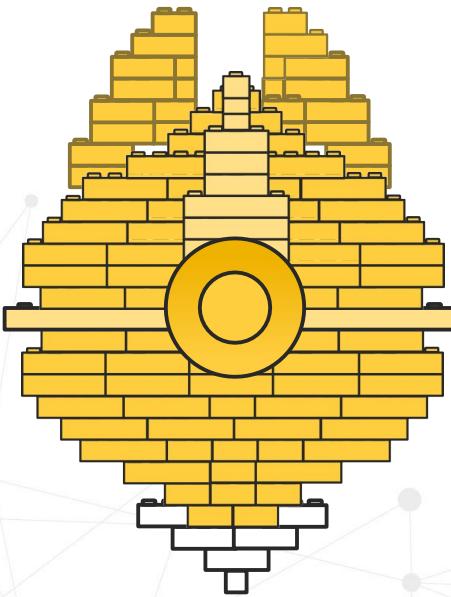


A few
hours



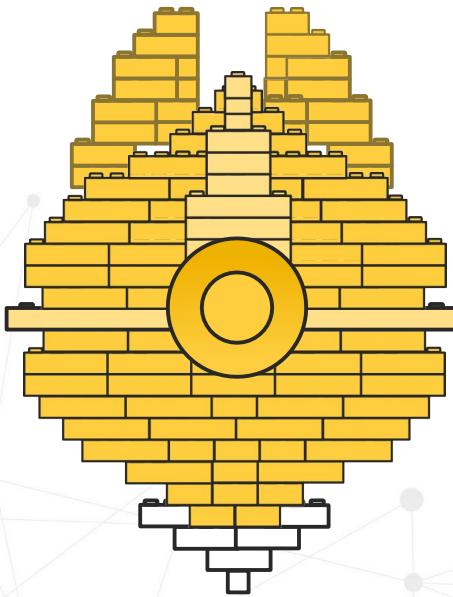
A finished toy

Rapid development



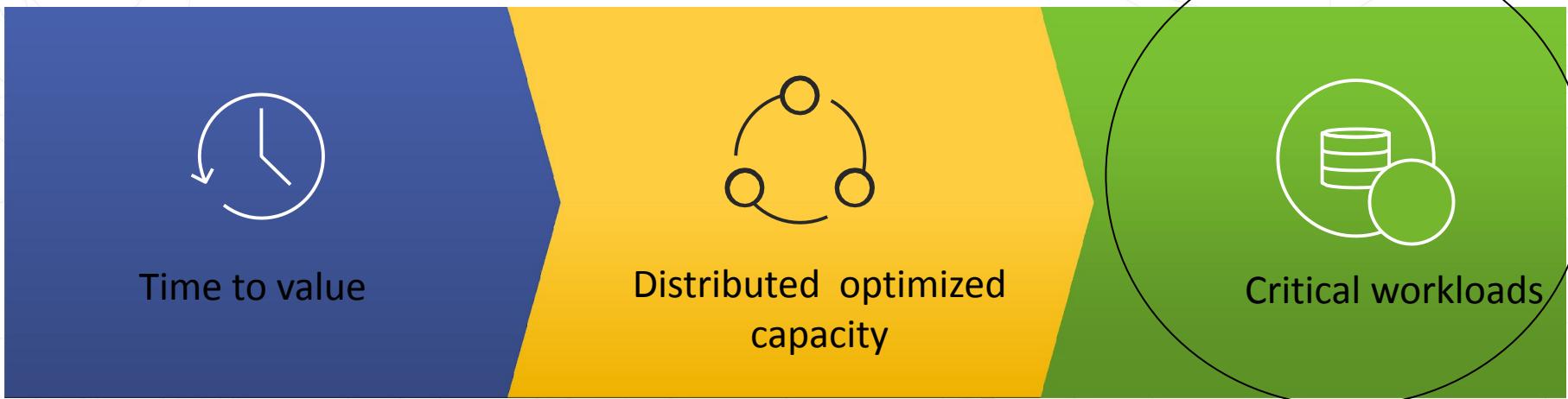
- Lacks fine detail
- Recognizable, but not exactly what was asked for
- Easy to modify and extend

Optimization

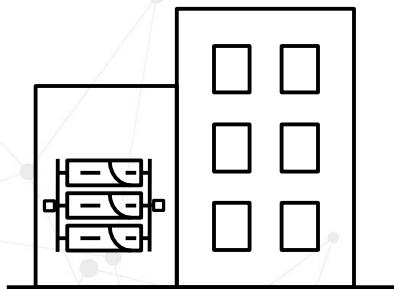


- Take a group of Lego bricks ...
and form a new custom brick
- A more specialized common
component

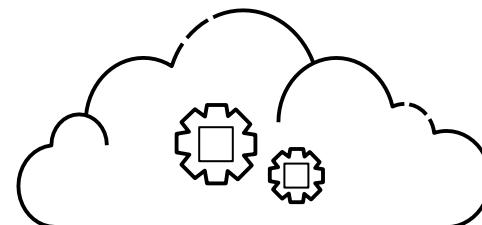
Pathway for innovation



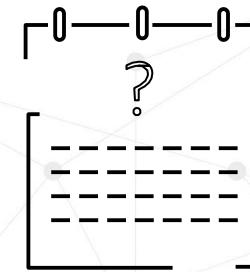
Resilience



Disaster
recovery



Chaos
engineering



Resilient
critical systems

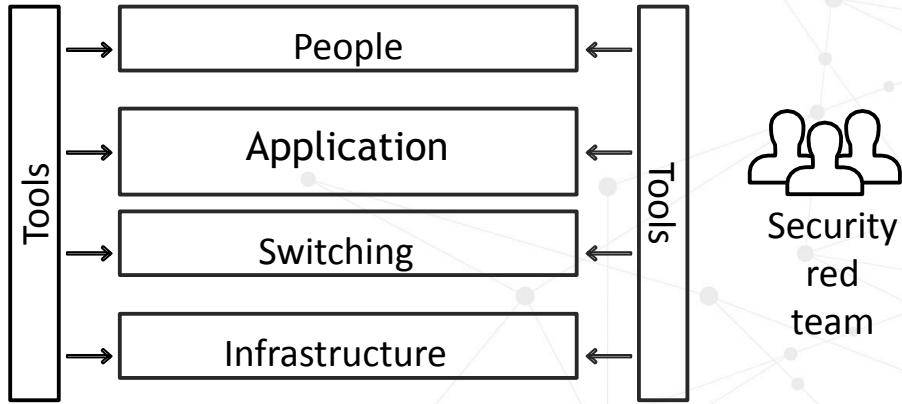
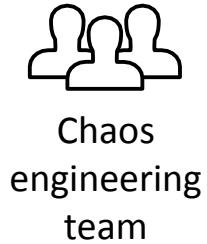
Resilience

Chaos architecture

Four layers

Two teams An
attitude—

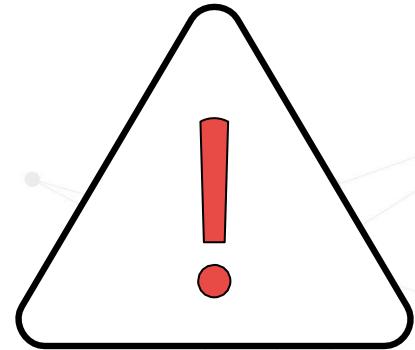
Find the weakest link



Resilience

Failures are a system problem—lack of safety margin

Not something with a root cause of component or human error



Resilience



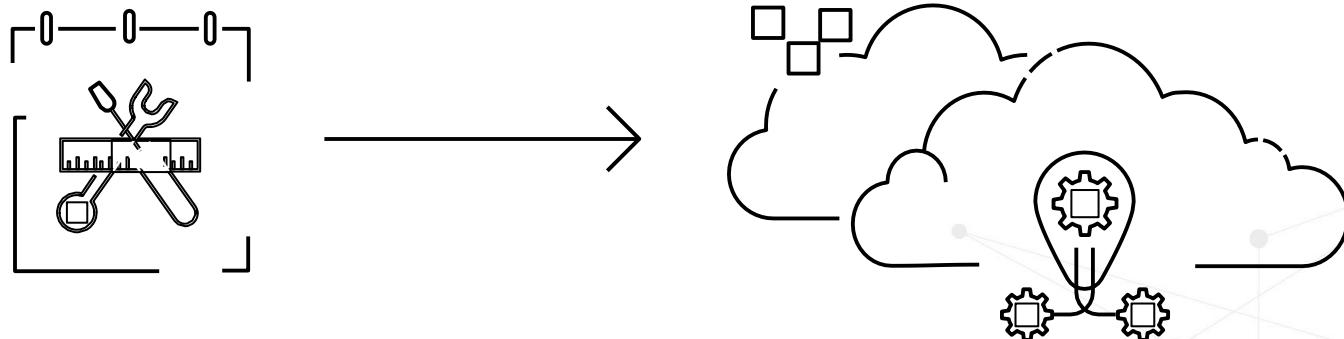
**Cloud provides the automation that leads to
chaos engineering**

Resilience



As data centers migrate to cloud, fragile and manual disaster recovery processes can be standardized and automated

Resilience



Testing failure mitigation will move from a scary annual experience to automated **continuously tested resilience**



A network graph consisting of numerous small, semi-transparent gray circular nodes scattered across the frame. These nodes are interconnected by a web of thin, light-gray lines, creating a sense of a complex, distributed system or community.

THANK YOU