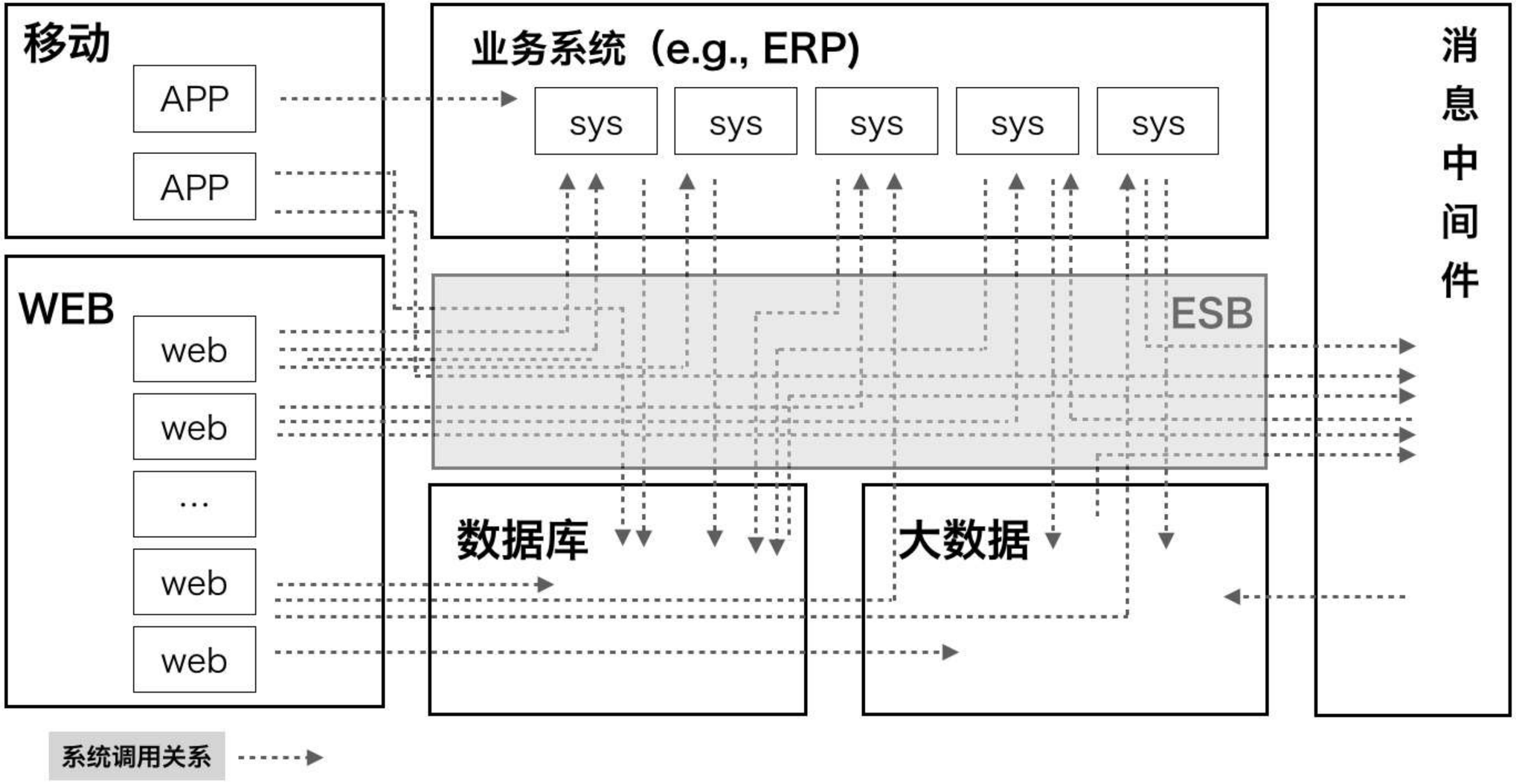

Dec 2017
为不确定性架构
肖然 @ DDD China
ThoughtWorks®

ThoughtWorks

这样的架构有什么问题?



ThoughtWorks

这样的架构有什么问题?

微服务治理

- 服务发现
- 服务认证
- 负载均衡
- 健康度监控
- 服务配置
- 服务路由
- 流量整形
- 失败隔离与恢复

微服务设计

- 领域驱动设计
- 绞杀应用模式
- *RESTful*
- 断路器
- 命令与查询职责分离
- 守门人
- 消息总线
- 轻量级消息
- 补偿性事务
- 事件源
- 领导者选举

微服务交付

- 服务模版
- 服务组件测试
- 消费者驱动契约测试
- 部署流水线
- 基础设施即代码
- 服务容器化

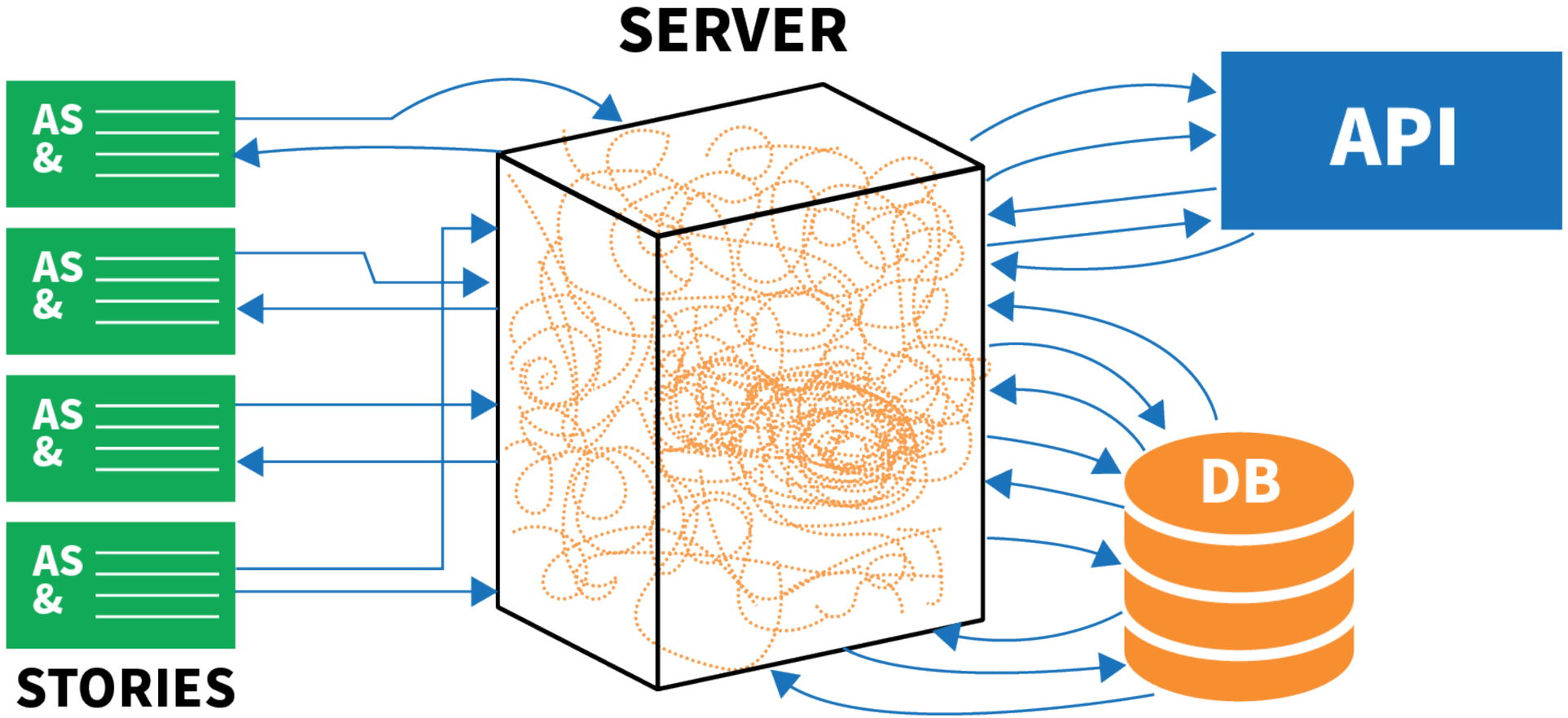
微服务组织

- 跨职能小团队
- 全生命周期职责
- 自助运维

我们忘记了时间

ThoughtWorks

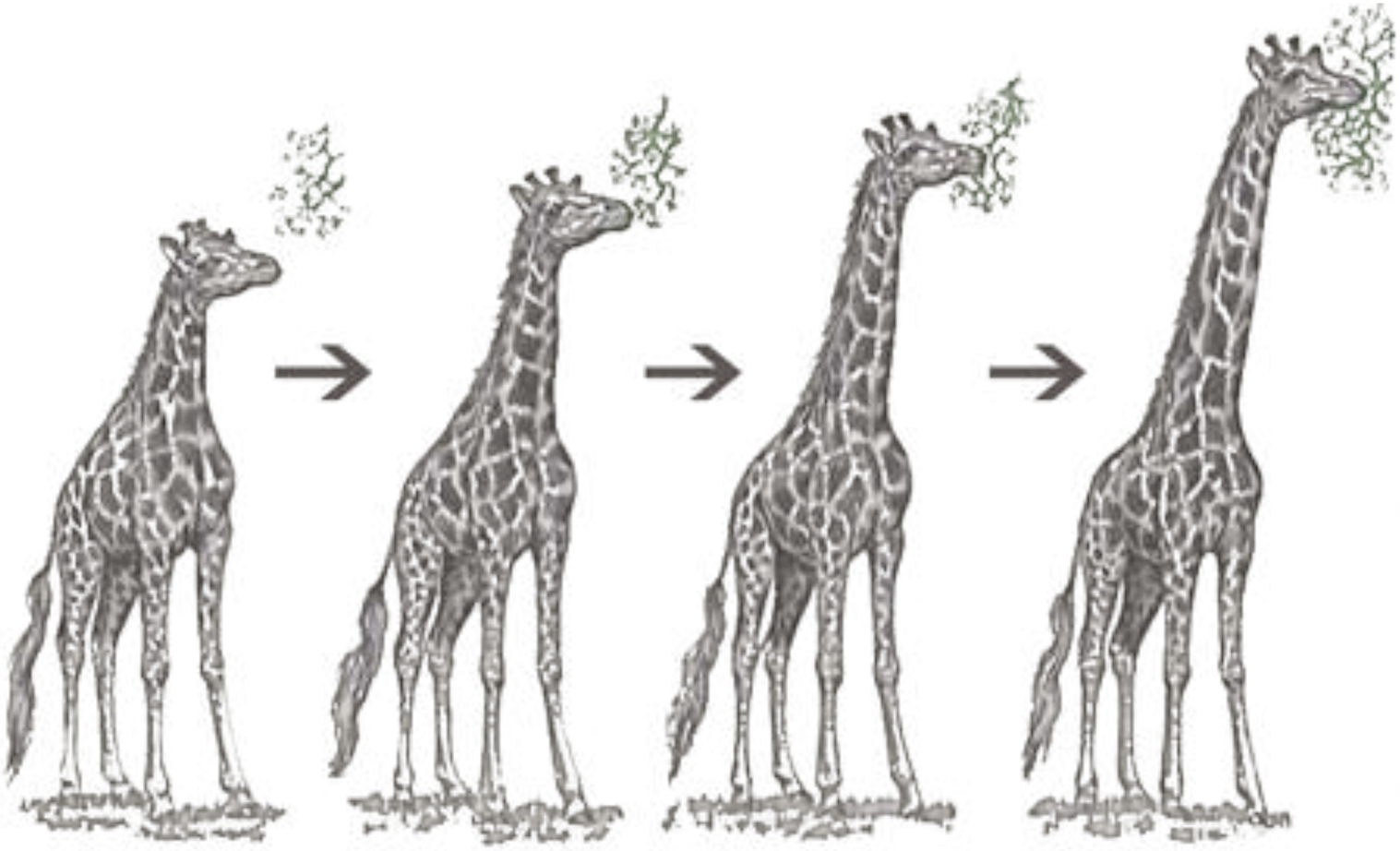
我们怎么解决这个问题?



https://18f.gsa.gov/2015/11/17/choose-design-over-architecture/

ThoughtWorks

大自然怎么解决这个问题?



适者生存

在*1809*年出版的《动物哲学》（*Philosophie Zoologique*）

为不确定性架构

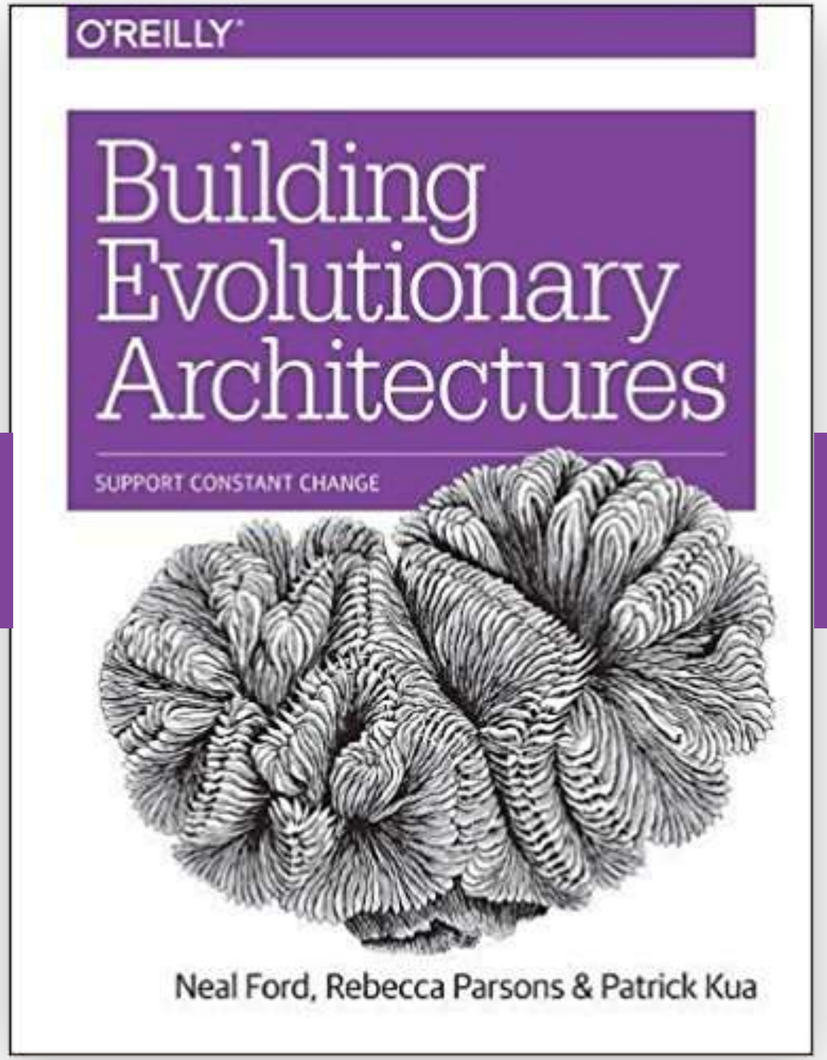
ThoughtWorks

没有终点的架构

Table 1-1. Partial list of "-ilities"

accessibility	accountability	accuracy	adaptability	administrability
affordability	agility	auditability	autonomy	availability
compatibility	composability	configurability	correctness	credibility
customizability	debugability	degradability	determinability	demonstrability
dependability	deployability	discoverability	distributability	durability
mobility	modifiability	modularity	operability	orthogonality
portability	precision	predictability	process capabilities	producibility
provability	recoverability	relevance	reliability	repeatability
reproducibility	resilience	responsiveness	reusability	robustness
safety	scalability	seamlessness	self-sustainability	serviceability
securability	simplicity	stability	standards compliance	survivability
sustainability	tailorability	testability	timeliness	traceability

Evolvability



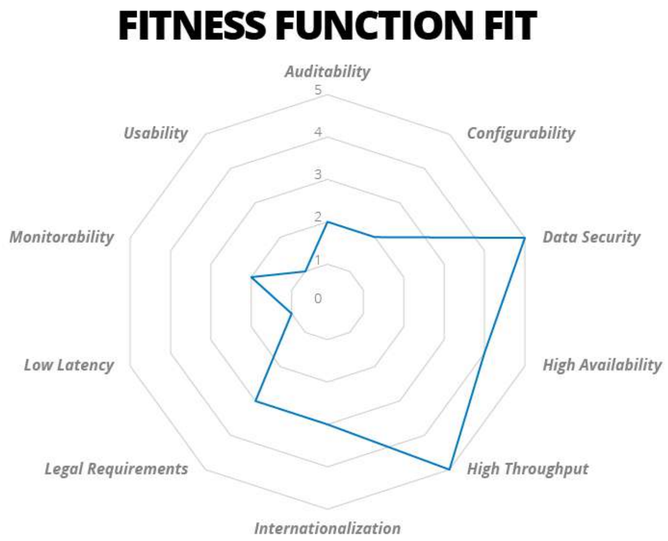
ThoughtWorks

适者生存：适应度方程

FITNESS FUNCTIONS

A **fitness function** is a particular type of objective **function** that is used to summarise, as a single figure of merit, how close a given design solution is to achieving the set aims.

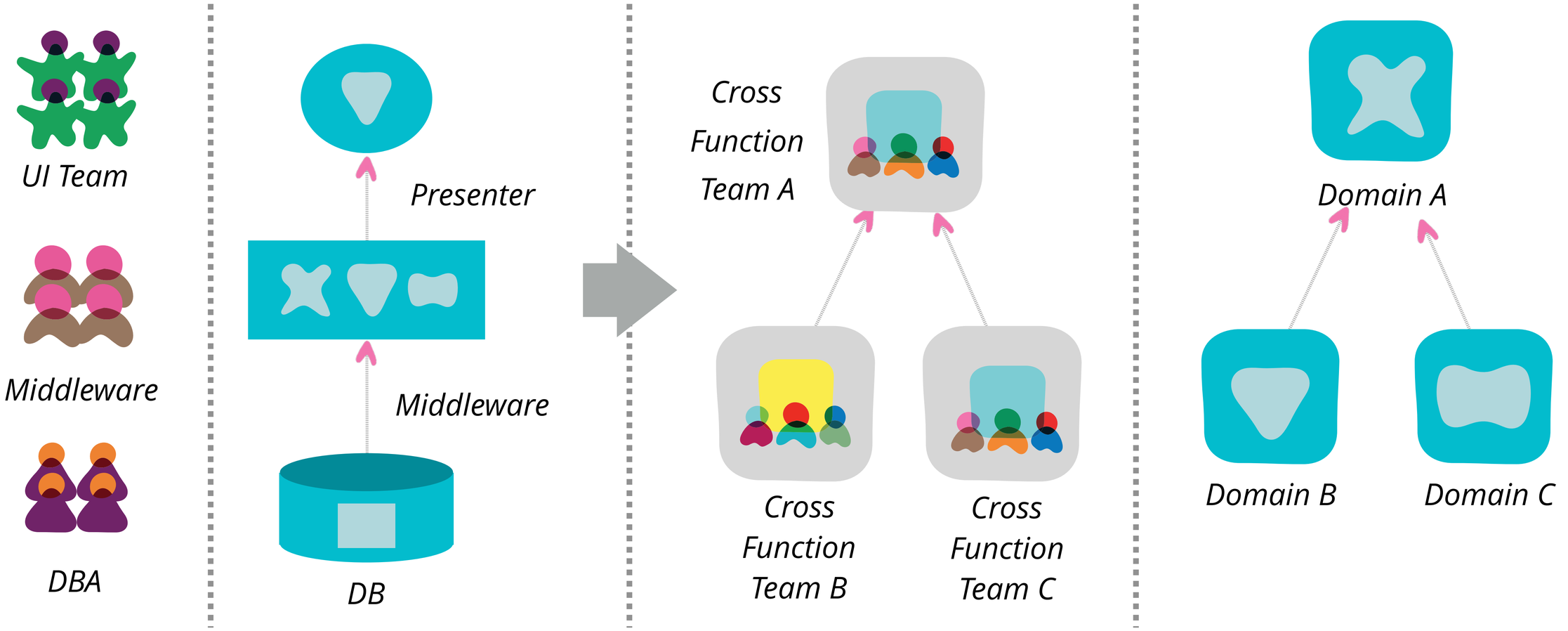
Fitness function - Wikipedia
https://en.wikipedia.org/wiki/Fitness_function



1500 characters with color, size and location attributes to create an image that matches Mona Lisa. Final fitness came to 90.6% with 1700 generations. Source: https://www.youtube.com/watch?v=TManzvC9pi8

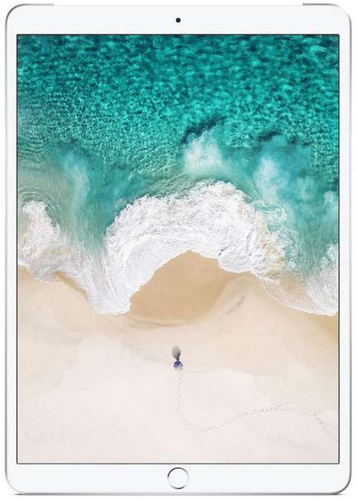
ThoughtWorks

架构和组织共同进化

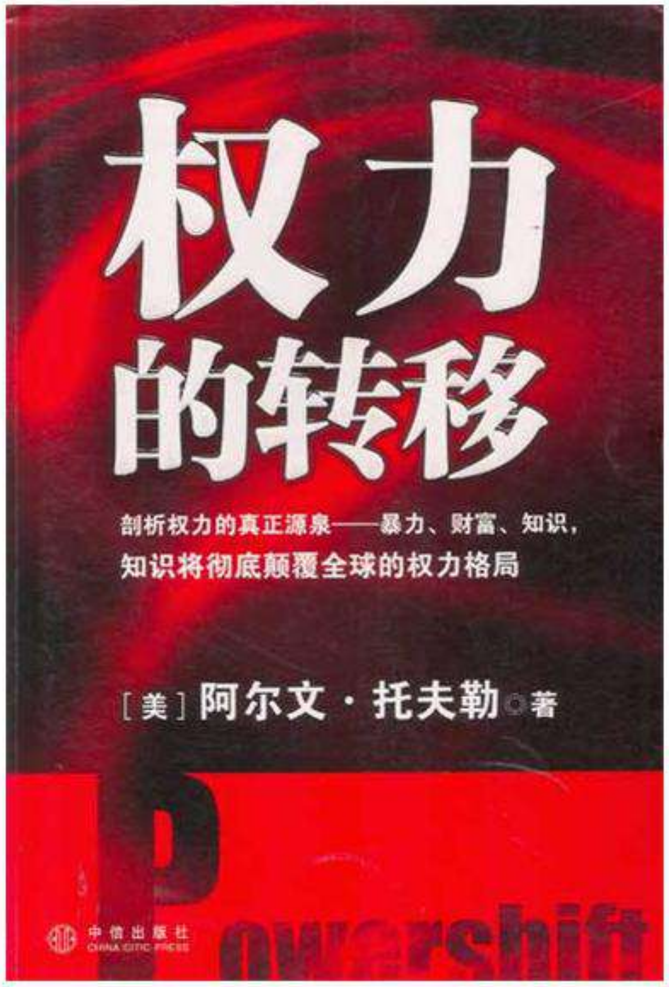


ThoughtWorks

应该怎样进化?



不确定中的确定



为不确定性架构

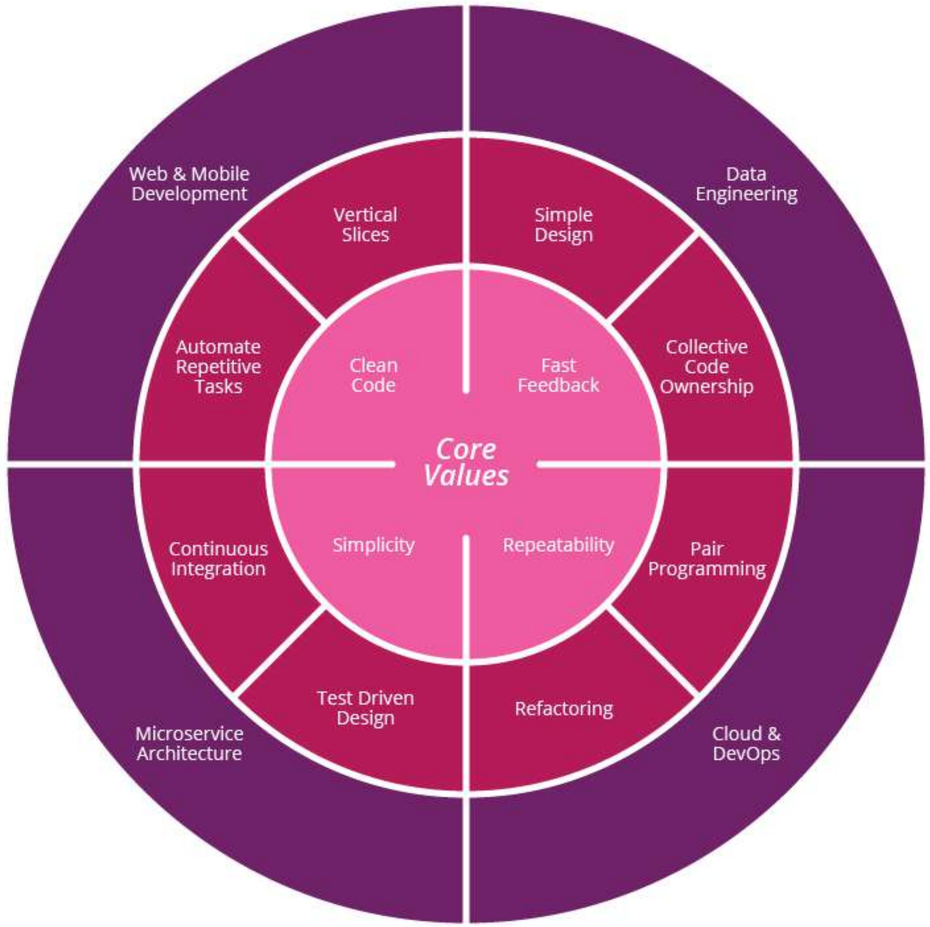
怎样适应进化?

快速反馈

重复可见

追求简单

整洁代码



https://www.thoughtworks.com/insights/blog/what-are-our-core-values-and-practices-building-software

考虑进化，做一个有节操的工程师！

Dec 2017

肖然 *@ DDD China*