数字化转型已经成为传统企业的必然选择。随之而来的业务场景、用户习惯和行为在迅速变化，许多传统行业线上业务出现急速增长。

金融行业的移动支付、互联网理财等，汽车制造行业的营销、电商、售后服务等线上业务比例迅速提高。

IT团队业务开发、迭代都以每月、甚至每周来计，需要7\*24小时响应，这些给系统开发和运维带来极大挑战。

传统行业对于IT效率的变革需求以及业务模式的创新导致系统更新频繁，应用复杂度也急剧上升，传统架构不堪重负。

因此，架构转型成为了一些企业数字化过程中的最大难点。

近期, 为了准备雪浪大会, 钛媒体记者走访调研了近200家制造企业, 深入一线去了解他们的真实痛点。

**调研结果显示, 在系统支撑方面, 排在前四的难题分别为:**

**系统复杂性越来越高、运维管理复杂度高；**

**打造一支全栈运维团队困难；**

**线上访问压力大；**

**设备采购维护成本高。**

ThoughtWorks是一家技术类咨询公司。

这家公司长期致力于从全球不同的区域，服务于不同的客户，在不同的业务模式下面做不同的项目。

成功的案例包括德国戴姆勒公司的营销、电商、售后服务如何由消费者主导实现数字化在线、美国视频网站Netflix的系统架构演进式发展等。

通过全球项目经验的积累和实践!!!

ThoughtWorks总监级咨询师Neal Ford提出演进式架构的概念, 将“可演进性”作为新的架构特征加入到系统中，让它在系统演进时为其他特征(比如上文提到的业务需求、性能、安全性、可扩展性等)提供保护。

这便是演进式架构，它使我们可以兼顾多个架构维度进行引导式的增量变更。

为企业传统架构的迭代升级提供一种新的思考方式!!!

在不破坏原有IT架构的前提下进行“演进”!

在传统企业软件开发流程中, 经常需要面临改动, 有来自用户需求的改动, 有来自市场的改动以及为了一些潜在机会而产生的改动等!!!

这些改动要求企业能够快速做出调整。但不幸的是，事情并不总是如我们所愿。

这是因为企业的业务应用经过多年IT建设，系统非常庞大，难以更改。例如企业传统SAP、汽车制造企业里传统DMS以及MES等，要改动其中任何一小部分，都需要重新部署整个应用。敏捷开发和快速交付更无从谈起。

另外，传统企业在长期的IT建设过程中，通常大量使用外包团队，这导致采用的技术栈之间差异较大，统一管控和运维要求更高。需要运维7\*24小时全天候值守、在线升级，并快速响应。

而在此时脱颖而出的微服务架构，让传统企业眼前一亮，它具备诸多优势，例如独立开发、独立部署、独立发布，去中心化管理，支持高并发高可用，支持丰富技术栈，企业可以根据需要灵活技术选型。

NealFord认为，微服务是支持演进行为的众多架构之一。

微服务是后DevOps革命时代出现的第一种全新架构风格。

它是第一个全面拥抱持续交付的工程实践，也是演进式架构家族的一员。演进式架构以支持增量的、非破坏的变更作为第一原则，同时支持在应用程序结构层面的多维度变化, 不过, 微服务仅仅是支持某些演进行为的众多架构之一。

而演进式架构则在微服务架构基础上增加了两点思考——既要满足企业应用的真实场景，还要符合技术的演进。

**首先，创建演进式架构，应当满足企业应用的真实场景。**

这也回答了我们文章的问题——为什么演进式架构被誉为最好的迭代模式？因为这种架构不会破坏原有传统软件包，而是将传统IT架构演进到微服务架构上。

NealFord在《演进式架构》中也提到，“对于一个大的软件包，一个大的单体的应用，如果做微服务转型，肯定不是把这个大的单体应用直接干掉不要，建一个新的微服务平台出来，

而更倾向于一种做法, 即我怎么能够从SAP、DMS这样一个传统软件包的模式, 一步步的把它演进到一个微服务架构上，这是演进式架构要解决的一个重要问题”。

原本对传统企业来讲，SAP、DMS以及MES这类传统软件包等本身是封闭的，企业购买这样的软件包最大的一个出发点是来降低自己的IT成本去实现IT能力。但是在数字化转型的大趋势下，创新又成为企业最主要的诉求。

ThoughtWorks从技术的角度出发，认为更应该照顾企业真实的应用场景，定制化地开发软件, 或者说, 从定制化的软件平台下开发出独特的业务模型。

**同时，演进式架构符合技术的演进。**

NealFord认为, 微服务本身是一种架构的模式或者是一种架构的风格, 而这种架构风格正好是演进式架构的一些原则和实践的最佳体现!!!

“比如业界经常做微服务转型的时候会说, 微服务本身的力度或者微服务本身的范围应该是什么样的, 而我认为演进式架构可以帮助企业将一个微服务的架构在力度的层面、在整个架构体系上能够不断地去演进.”

NealFord向钛媒体表示, 这种技术演进方向有两个:

“第一个方向是这种增量式的演进，让系统能够做到增量式的演进或者增量式的变更，而这本身作为演进式架构定义中的第一个维度, 它是天生要解决这个问题。

第二个方向, 它叫做指引式的演进或者向导式的演进, 向导式的演进就是说我在一个系统里面希望能够提升它的性能, 我怎么能向着我们想要的性能方面去演进,

所以它定义了一个适应度函数(fitness function), 通过适应度函数可以帮助我们的架构的开发人员明确地认知我现在想要的这个方向是不是我现在架构所演进的方向, 是不是我想要的方向,

然后它可量化地告诉我的开发者, 我离现在这个目标到底还有多远, 它是这样一种模式。

所以面对企业不断增加的应用需求, 它天生可以解决企业增量式、迭代式开发的一种诉求。”

然而，微服务也不可避免地表现出它的劣势——它会带来系统的各种复杂度、运维要求更高等诸多难点。对于团队来说，搭建微服务架构上手难，运维效率低，运维成本高。

另外, 创建微服务还可能带来团队之间沟通上的冲突---微服务需要与DevOps(Development&Operations)同步推进。

当微服务被分割成一个个独立的业务模块后, 服务间通信接口设计非常重要。

如何科学地将系统部署到服务器上, 保证各个服务高效运行, 更是难点。

演进式架构符合传统企业转型需求

当我们谈演进式架构的时候，其实是有很多不同的原则。

这些原则都很严格，而微服务某种程度上展现出了一些演进式架构的原则。

例如我们在谈微服务的时候，主要是想用它来快速应对频繁的需求变更，因此微服务所承载的期望便是让我们能尽可能快的适应变化，演进式架构中所倡导的进化性与此不谋而合。

NealFord补充道, 关于可进化性存在一个理解误区——开发者要非常聪明的想到所有可能出现的变化，并且为所有的这些可能的变化写好代码，哪怕到头来根本就没变。

但其实，可进化性并不同于可维护性, “不是说要怎么预测未来，而是一种随时准备响应变化的状态, 而不管你是否提前就设想好了这些变化。”

在今天来看, 其实演进式架构本身是有一定基础的, 当企业想采用演进式架构这种模式的时候, 需要以敏捷的开发模型、敏捷的开发方式、持续交付的开发基础设施或者是持续交付的开发实践以及DevOps的实践来作为整个架构的基础。

如果企业没有这样的基础, 实际上是很难做到演进式架构的, 而演进式架构本身它也是要求在整个企业开发的流程和模型上需要敏捷、持续交付，要去做DevOps。

据Neal Ford介绍, 目前在业界采用演进式架构企业不仅有美国视频网站Netflix公司, 还有很多ThoughtWorks参与或者是合作的公司都在采用演进式架构中的一些实践。

只不过在这之前并没有把它定义成演进式架构或者是适应度函数, 而是把它定义成fitness function。

而在这些传统企业中, 很多企业是通过持续交付流水线，通过对于架构、对于测试体系的一些追求, 对于架构本身的转型, 一直在实践着演进式架构的模式。

因为它们恰恰满足了演进式架构的一些特点, 例如:

**第一是模块化和耦合。**

边界划分明确的组件，显然可以给希望做出非破坏性变更的开发人员以更大的便利。

而毫无架构元素的混乱架构就无法做到演进式变更，因为它缺少模块化。

另外不适当的耦合将变更导向难以预料的方向，从而阻碍演化。

而演进式架构都支持一定程度的模块化，这种模块化通常体现在技术架构层面（例如经典的分层架构）。

**第二是围绕业务能力组织。**

现在越来越多的成功架构都以在领域架构层的模块化为特色。

基于服务的架构与传统的SOA主要区别在于模块划分的策略，SOA是严格按照技术层进行模块划分，而基于服务的架构则倾向于按业务领域划分。

**第三是试验。**

试验是演进式架构给商业交付带来的最大价值之一。

从操作角度来讲，可以采用A/B测试等常见的持续交付实践对应用进行低成本的、微小的变更。

微服务架构常常是围绕服务之间的路由来定义应用程序的。

通常微服务架构围绕服务之间的路由来定义应用程序, 允许同一个服务的多个版本同时运行。

这反过来也使得试验和现有功能的逐步替换成为可能。

最终，这使得企业业务可以花更少的时间去猜测待办故事项，从而投入到假设驱动开发中。

而在国内, 通过钛媒体记者近期的走访发现, 大部分传统企业架构还不能实现敏捷开发、持续交付等, 要实现数字化转型, 首先就需要对企业传统架构做转型!

NealFord表示, 对于企业传统架构的转型, 首先要把企业自己基础的工程实践能够搭建起来, 使原来采购软件包的模式能够向新的模式做转型。

“比如说在持续交付层面，如果一个企业不是在做持续交付的转型，那企业在做精益企业(或数字化)转型的时候, 第一步要做的事情就是要有一个持续集成的转型,

把自动化的基础设施能够搭起来。

这是基础能力, 不仅仅是向演进式架构转变, 更企业在做数字化转型时必备的一些能力。”

大数据、AI技术都将被引入演进式架构

“目前在演进式架构中已经引入了很多AI的实践，比如说在一些企业里面会用TensorFlow去写一个框架，帮助他去分析在网络传输这一层里面，

从安全的角度是不是有一些模式可以被快速地发现，有一些安全的点或者是有一些安全的攻击的行为是不是可以从网络传输这一层来用AI来做分析和发现。

它也希望未来可以通过把目前AI技术应用所处的层面逐渐往上提，提到不同服务之间传输信息的层面上、是不是可以利用AI找到这样的模式，能够帮助我们企业识别出来我的安全风险。”

Neal Ford认为，**在当下最时髦的AI技术，在未来可能会被更多地引入演进式架构中，加入到整个企业架构适应度函数的设计，或者是在架构演进的过程中利用AI的能力做更多的事情。**

“同样对于大数据大数据或者是数据型的项目, 演进式架构中可以应用在一些场景中。

比如说演进式架构里面去定义企业最关注的适应度函数、最关注的架构设计要素。

而每一个要素都可以通过适应度函数把它定义下来，我认为每个架构应该有自己非常独特的或者是特定的适应度函数，或者是这种架构规则的新的定义，就好像我们写unittest和写functional test是一样的。

因为你是根据某一个项目去写的，而不会把这个项目的规则直接搬到另一个项目上去。”

无论是传统架构还是演进式架构，系统本身在市场上是有竞争者的。

Neal Ford也坦率地说, 如果是利用这个系统直接去赚钱的企业或者是组织, 这样的企业会首先采用演进式架构的模式。

而往往一个企业当它先开始采用演进式架构这种模式的时候, 这种架构一开始对企业的影响并没有那么大。

“只是简单的在企业流水线上布一个适应度函数或者是布一个架构检查的脚本, 其实就已经在采用演进式架构当中的实践了, 但是显然这种实践对于人和组织来讲并没有什么太大的改变。”

“但是如果企业想充分发挥演进式架构带来的好处, 那对于整个团队的组织结构肯定是要发生一些变化。

比如说企业的组织结构应该能够更好地支撑业务的敏捷性转型、企业的组织结构应该更好地降低我在团队不同成员之间的波动,

我能够让团队是一个相对比较稳定的团队。

同时我的团队能够支撑DevOps这样一些实践的追求或者应用。”

这就像1967年康威提出的“康威定律”， 企业需要关注怎么组织团队结构，以及不同的团队结构可能影响到最终的系统形态。

只有团队对服务有很好的所有权意识，团队做出来的微服务才是这种松耦合的独立服务。

康威定律在1967年被提出来，但是真正被业界采纳其实是在近几年有了微服务以后，团队才开始采用康威定律。“所以说这件事情相对来看会有一个比较长的周期。

其实在业界已经有了这样一些原则在那里了，但是真正的被企业界所应用还是需要一定时间的。“Neal Ford表示。

  所有议题都将围绕着基于对近万家制造企业的数据调研，和超过200家企业面访所总结出来的制造企业最关心的问题展开。

雪浪大会以唤醒计划推动制造业和互联网的全面融合，致力打造制造业与科技对接的平台。

制造企业不仅可以充分了解、对接全球最新的科技资源，还可以发布自己的痛点问题，在更广范围内寻求解决方案；而科技公司除了展示自己的成果，更可以通过了解制造企业的需求，为自己的科技产品找到海量的制造业应用场景。