

白皮书

现代化 JAVA 应用平台

面向云和现代生态系统设计工作负载

72% 的 IT 经费被
用于项目维护。²

67% 的 CEO
将数字变革作为
业务战略的核心。¹

59% 的 CIO 担心
其当前 IT 人力资源
难当未来重任。²⁰

40% 的企业计划
在未来 3 年内对 IT 基础
架构进行现代化改造。¹⁸



红帽官方微博



红帽官方微信

cn.redhat.com

执行摘要

许多企业正处于战略规划的十字路口。根据 IDC 分析师的研究,¹三分之二的 CEO 将围绕数字变革计划制定企业战略。数字变革的目的不仅是提高现有职能效率，更是为了能够尝试新方法，以全新的方式、更有效地使用现有数据。

然而，企业当前 IT 支出的 72%²乃至 IT 资源工作所关注的重心都放在了现有系统的维护上，造成了当下需求和未来目标之前的紧张关系。

据 IDC 分析,³ 中间层应用是数字变革战略的核心，因为它们提供了数据整合、消息传递和应用编程接口 (API) 管理等重要功能。中间层应用还为应用开发和管理提供了平台，能够处理传统企业应用和云原生的分布式应用。

具体而言，基于 Java™ EE 的应用平台可以为时下的技术和云原生应用提供支持：

- 利用现有的人员和专业知识来运用新技术。
- 保留传统应用和重要数据。
- 开发能与现有环境并存的新应用。
- 实施新的流程和架构。

每家企业都有不同的数字变革前景。在制定自己的独特战略和目标时，企业需要遵循一套核心原则，帮助指引决策方向，利用现有的 IT 资源来达到最大成效。

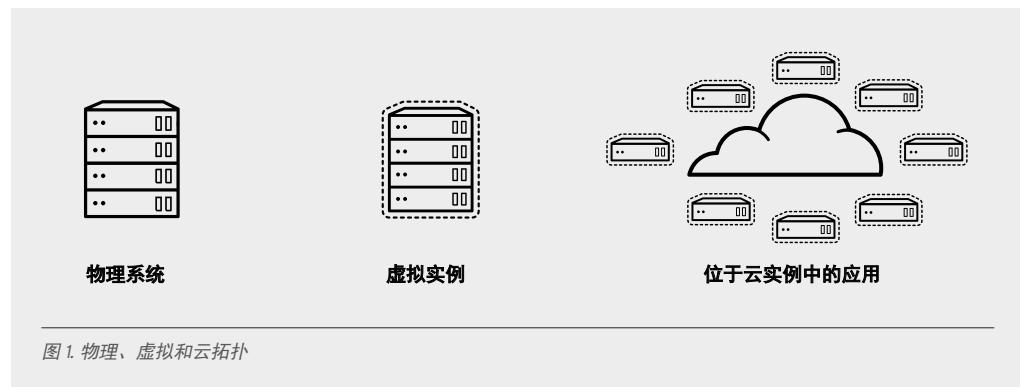
¹ Frank Gens. “IDC FutureScape: Worldwide IT Industry 2016 Predictions — Leading Digital Transformation to Scale”。IDC, 2015 年 11 月。

² Minda Zetlin. “How to Balance Maintenance and IT Innovation”。ComputerWorld, 2013 年 10 月 21 日网络来源。

³ Maureen Fleming. “New Middle-Tier Competencies Enabling Digital Transformation”。报告。IDC, 2016 年 6 月。网络来源。红帽赞助。

未来展望：平台、流程和架构

IDC 分析师 Al Hilwa 在一场关于软件开发现状的演讲中指出⁴，优秀的软件是一个软件架构、开发人员流程和开发人员技能的有效整合。IT 和运营部门在这三个方面都面临着重要变革。随着云和分布式计算为基础架构带来了新的规模经济，架构、流程和平台这三大基础也正在逐渐朝着高效利用云环境的方向发展。



平台：云

云计算将是下一波 IT 创新的核心基础架构。分析机构 IDC 估计，到 2020 年，基础架构总支出中将有 70% 花在云服务上。⁵

云的关键效用在于能对环境变化做出迅速响应。虚拟化是 IT 领域的一大推动因素。它能将运行环境从物理环境中抽离，令操作系统完全独立于物理系统，使得同一硬件上能够安装运行多个操作系统实例。云将环境进一步细分，把运行的应用从底层操作系统或物理环境中抽离出来。

基于云的服务可以跨分散节点存在，而不是被严格分配到某个特定物理系统，所以它能够提供冗余性（当一个节点失效时，其他节点可以接管其工作）和扩展性，可以根据特定的服务创建和部署新的实例。

这种轻量基础架构能为高度分散的架构模式（如微服务）提供支持。而云环境的另一大优势在于能够最大化资源利用率，提升运营效率，节省更多成本。

⁴ Al Hilwa. "The New Developer Landscape — Understanding the Modern SoftwareDeveloper". 2016 年 3 月。IDC 活动演讲。

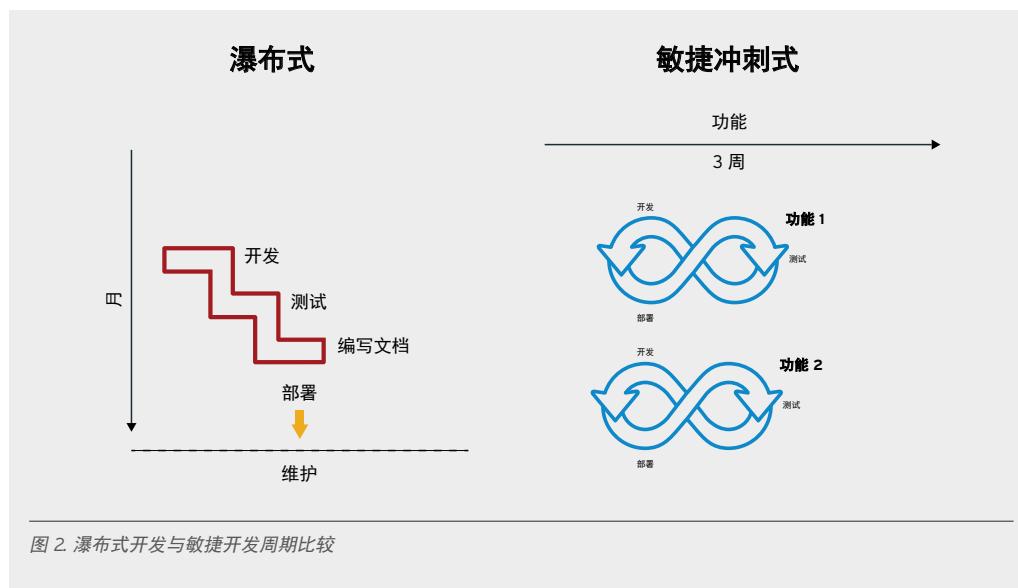
⁵ 出处同上，Frank Gens。

流程：DevOps 和敏捷开发

敏捷开发、持续整合/持续交付 (CI/CD) 和 DevOps 是相辅相成的概念，它们对现代 IT 团队的组建和履职至关重要。

敏捷开发是项目规划与执行的一种方法。根据“敏捷软件开发宣言”⁶ 中的定义，敏捷方法具有四大核心原则：

- 以人员/团队为先。
- 与外部团体协作。
- 响应环境变化。
- 创造易于理解的软件。



敏捷开发流程将项目划分为多个可以管理的任务，追求在任务之间快速更迭，逐一交付可以工作的软件功能。这与瀑布式项目所采用的受拘束的、渐进式方法截然不同。敏捷方法日渐流行，根据 Gartner 最近的一项评估，瀑布式项目的比例已掉到 50% 以下。⁷

敏捷方法给项目开发带来的一大改变是让其他团体参与到整个规划和开发过程中，而不是开发完成之后才介入。在冲刺结束时，软件必须能够正常运行。为此，软件的测试工作必须与开发工作同步进行。敏捷团队将测试包含在开发周期内。这种持续测试的理念变成了持续整合，而快速发布则变成了持续部署。由此形成接连不断的流程，而不是一个个离散的阶段。

⁶ Kent Beck 等人，2001 年。“Manifesto for Agile Software Development”。Agile Alliance。

⁷ Nathan Wilson。“Modernizing Application Development Primer for 2016”。Gartner，2016 年 1 月 14 日。网络来源。

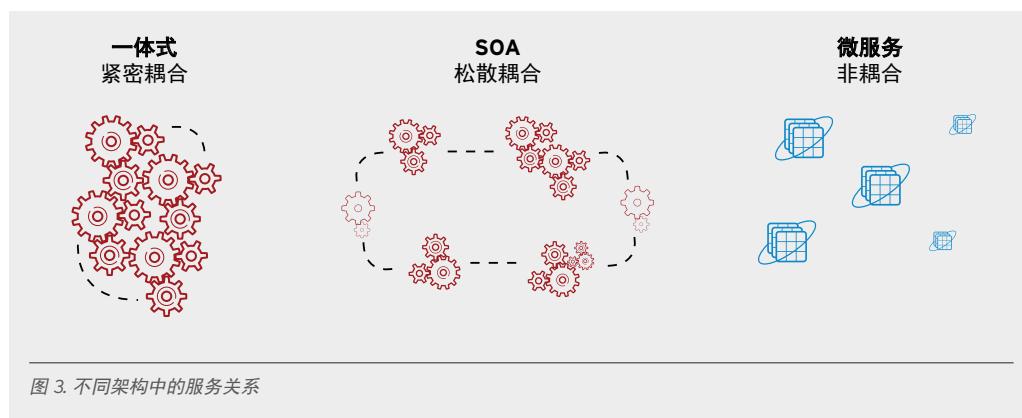
不过，应用的生命周期并非在交付之时就画上句号。应用还需要部署和维护。这一工作通常落入运营和IT部门的工作范畴，而不是开发部门。如果这些团队过于分离，开发人员可能不了解运营环境，或者运营团队不明白产品的战略目的。这种理解上的隔阂可能会造成软件或基础架构性能低于预期。而将开发与运营合为一体，即所谓的DevOps（开发运营）。

架构：微服务

一直以来，企业软件通常非常庞大。在过去，单个应用要应付整个企业的高峰负载。而且，它所需的所有功能都得由它自己来执行。服务仅仅是单个一体式应用中的功能而已。比如，数据库就是一种典型的一体式应用，一个数据库可以为整个企业提供服务。

随着企业的计算需求愈加复杂，一体式应用的维护要求也在抬高。即使对一体式基础架构进行细微的更改，也会有难度，因为牵一发而动全身。这便促成了一种新的架构方法的诞生，即面向服务的架构(SOA)。面向服务的架构中不采用包揽一切的单一应用，而是交由多个不同的应用来提供相关功能，利用一种整合模式将这些应用松散地耦合在一起，就如企业服务总线(ESB)一般。

另一方面，SOA也增加了整体环境的系统复杂性。虽然架构的某些特定方面得到了简化，例如引入新组件或进行升级时较为方便；但是，如果无法清楚理解组件间的交互关系，SOA也可能会进一步加剧整个环境的复杂性。



虽然SOA的方向没错，但如今已有一种更加成熟的模式应运而生，称为微服务架构。微服务架构与SOA模式类似，也是采用专业化且松散耦合的服务形式，但微服务的细分更进一步。微服务架构以非常清晰的方式定义服务：

- 单一而明确的用途
- 参数定义恰当
- 多语言版本实施

仍然排名第一

“Java 的前景依然被看好。这是因为，语言和平台的变化不会一蹴而就。令人振奋的技术有许多，大部分为开源技术，但能够取代 Java 的还未诞生。与微服务相关的工作中，许多都基于 Java。在采用开放式混合云计算的现代框架下，企业级 Java 正在得到重塑。优秀的成果不断出现，而且都是在开源范畴内。”

RICH SHARPLES
中间件产品管理高级总监
红帽

在微服务架构中，服务采用通用消息传递框架，如具象状态传输 (REST) API，使各种服务进行相互通信，而不需要繁琐的数据转换或额外的整合层。

这种消息传递框架使企业能够更快交付新的功能和更新，甚至可以说正是这种框架促进了这种良性转变。微架构中的每一个服务都是离散的。一个服务可以被取代、增强或丢弃，而不会影响架构中的其他服务。这种轻量架构有助于优化分布式资源或云资源，而且能够支持个别服务的动态扩展。

康威定律和未来发展

云、DevOps 和微服务有一个共同特质，即分布复杂性。1967 年，软件开发员 Melvin Conway (康威)⁸发现，软件的设计模式与开发团队的沟通结构有着紧密联系。沟通越是死板、不透明或不完善，软件的执行就越糟糕。

无论你的公司处于数字变革的哪个阶段，最终结果都会反映出你们 IT 团队的文化。因此，企业在攻克技术难关之前，需要先对自身的文化进行反思，⁹要点如下：

- 在团队之间建立易于理解的沟通模式。
- 最大程度消除团队之间的沟通障碍。
- 提倡灵活的基础架构和办公室文化。

与任何复杂的分布式架构一样，微服务也依赖于坚实的技术基础。

技术上而言，没有基于云的基础架构就没有微服务。¹⁰组织结构上而言，如果没有牢靠的 DevOps 或敏捷工作环境，缺少协作性高、沟通力强的跨职能团队，微服务架构就注定会失败。

JAVA EE 的过去与未来

如果未来我们面对的是基于云的轻量架构所带来的生产力和复杂性，那么您现在的 IT 部门需要如何规划以适应这一未来趋势？

Java 诞生于 1995 年，现已发展为世界上最流行的编程语言。¹¹ Oracle 估计全球 Java 开发人员的数量超过 900 万，¹² 占全世界 1100 万专业开发人员的近 82%。¹³

⁸ Melvin E. Conway. “How do Committees Invent?”. *Datamation*, 14 (5): 28-31. 1968 年 4 月。

⁹ DevNation Afternoon General Session 大会。Rachel Laycock. 加利福尼亚旧金山，2016 年 6 月 27 日。Performance. <https://youtu.be/EC2rk9Jh5Ps>

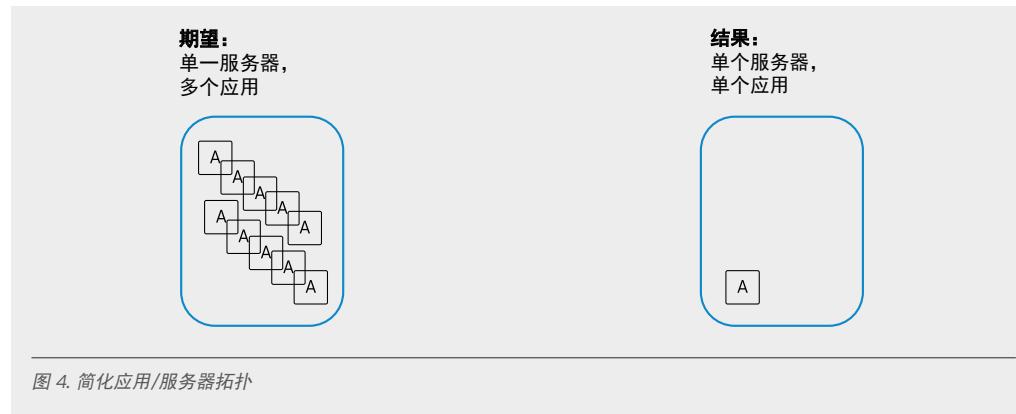
¹⁰ Frank Gens. “IDC FutureScape: Worldwide IT Industry 2016 Predictions — Leading Digital Transformation to Scale”. IDC, 2015 年 11 月。

¹¹ “TIOBE Index for August 2016”. 2016 年 8 月。网络来源。

¹² Timothy Beneke 和 Tori Wieldt. “8JavaOne 2013 Review: Java Takes on the Internet of Things”.

¹³ Al Hilwa. “The New Developer Landscape — Understanding the Modern SoftwareDeveloper”. 2016 年 3 月。IDC 活动演讲。

Sun（现已并入 Oracle）通过制定一系列通用运算标准、一套 API 和一个应用运行时环境，构建了 Java 编程语言。它们统称为 Java Enterprise Edition (Java EE)。实施 Java EE 规范的服务器就是 Java 应用平台，这些应用平台是许多 IT 开发环境的核心。Java 采用服务器加客户端编程模型，能够很好地适应早期的互联网框架以及后来的企业级应用。



21 世纪初发展应用平台时，大多数 IT 架构是在一体式应用基础上构建的。Java EE 应用平台原本要在单个集中位置上托管多个 Java 应用。而在实际应用中，由于流量分布、网络带宽和延迟、冗余性以及组织结构分散等原因，一个应用平台通常只能托管一个应用。Gartner 将这样的 Java 应用平台称为“超级平台”，¹⁴ 它们容易造成 IT 资源浪费。

Java 作为语言和平台具有很强的弹性，其中一大原因在于它具有很强的适应能力。Java 和 Java EE 由一个称为 Java Community Process 的社区进行定义。这个社区积极投身于 Java EE 的开发和相关社区项目，如 MicroProfile、Wildfly Swarm 和 Node.js。

目前，这一社区正在引领 Java 和 Java 平台向着实现云原生开发和部署轻量级互联应用的方向发展。Java EE 6 引入了配置文件的概念，既有一个与传统的全尺度 Web 服务器匹配的完整配置文件，又有一个适用于轻量应用的 Web 配置文件。配置文件和模块化特性在 Java EE 7 得到了拓展，而且还得到了几大社区项目的进一步推动。

- MicroProfile 正在考虑设计一种适用于新型 Java 配置文件的规格，它要比 Web 配置文件小，也更适合微服务，专注于事务处理和消息传递等功能。
- Wildfly Swarm 将 Java 应用当做容器镜像，把所有必要的库和依赖项捆绑到一起并塞入单个 jar（一种独立 Java 存档）文件中，不包含任何不必要的内容。

¹⁴ Nathan Wilson. “Modernizing Application Development Primer for 2016”. Gartner, 2016 年 1 月 14 日。网络来源。

您选择何种方式实现现代化更新？

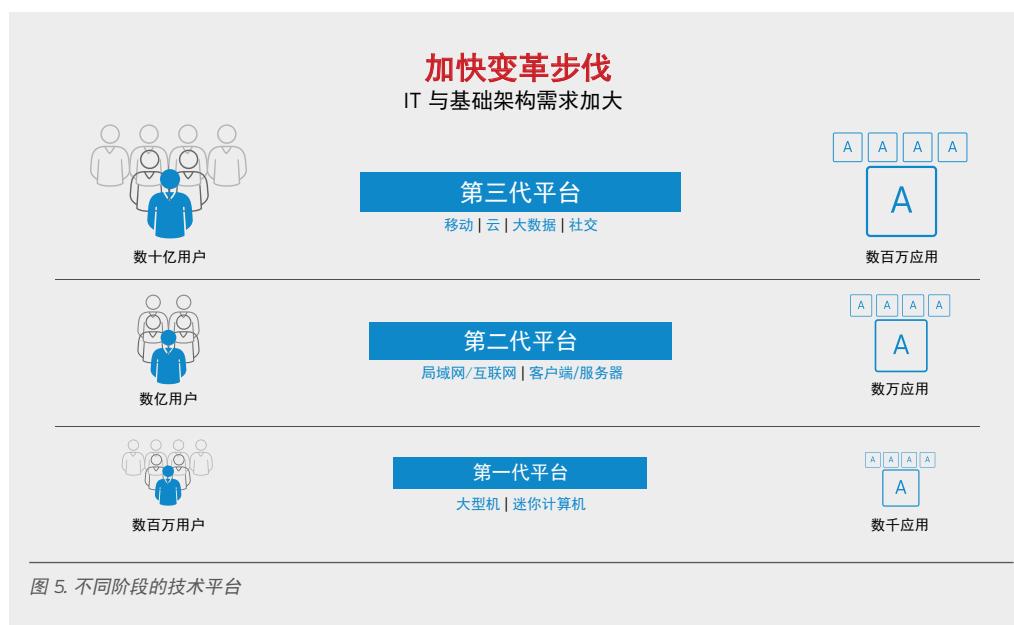
- 迁移至云端。
- 使用一种能在云端或数据中心发挥同等性能的应用平台。
- 借助你的业务策略，推动 IT 战略发展。
- 明确需要进行现代化改进的基础架构领域和具体方式。
- 创建新型团队及企业文化。
- 为开发人员提供合适的工具。
- 构建一个支持整合、扩展和互操作的 IT 生态系统。

与许多技术一样，Java 在适应云基础机构方面有一个关键优势：这一核心语言已被数百万开发人员理解和采用。虽然应用的架构和实施方法都是新的，但创建它们所需的技能却早已耳熟能详。

在变革中实现高效资源利用

IDC 把云称作第三代平台 IT 环境的核心。¹⁵ IDC 同时还为计算的演进定义了三个发展时期：¹⁶

1. 大型机和个人计算机
2. 基于互联网的事务和客户端/服务器架构
3. 以应用为中心的技术（如移动、社交、物联网和大数据）托管于云端



此第三代平台极大依赖于另外两种平台，但它的重点不在于以更高的效率完成同样的事，而是通过在这些层面上进行搭建来完成新的事务。

1. (利用 Java EE) 迁移至云端

云计算是所有现代 IT 环境的核心。原因在于云计算具有良好的可扩展性，特别是能够根据需要动态增加和减少节点，而这是传统的物理或虚拟系统无法企及的。

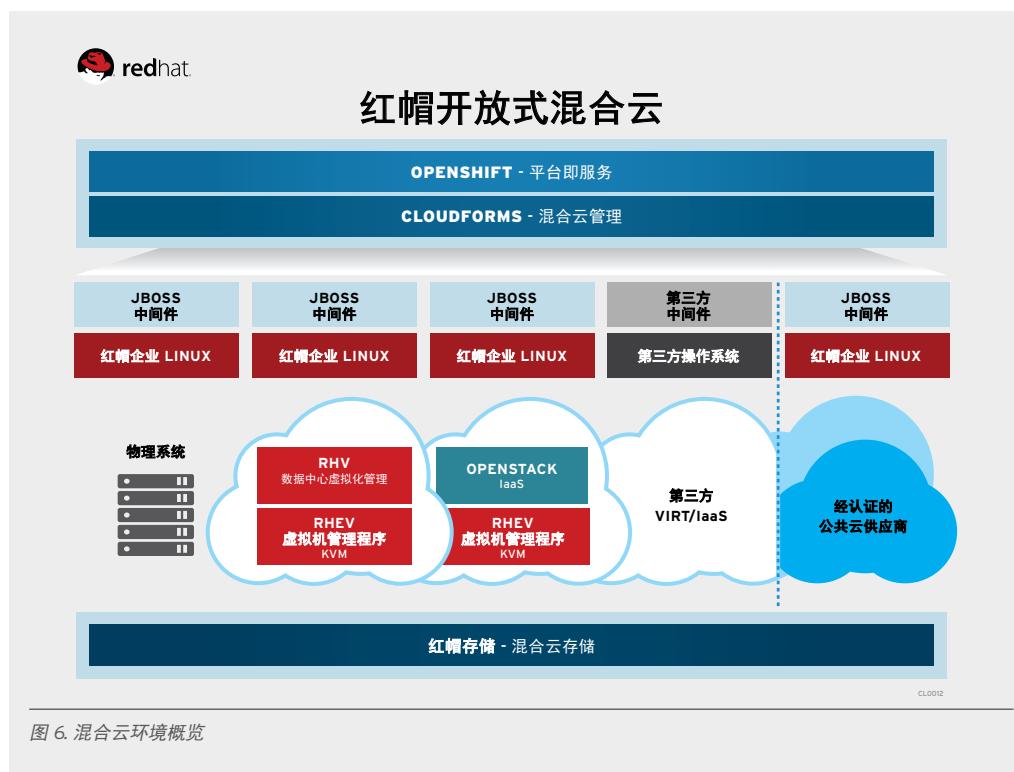
¹⁵ Frank Gens. "IDC FutureScape: Worldwide IT Industry 2016 Predictions — Leading Digital Transformation to Scale". IDC, 2015 年 11 月。

¹⁶ IDC. "IDC Predicts the Emergence of the DX Economy in a Critical Period of Widespread Digital Transformation and Massive Scale Up of 3rd Platform Technologies in Every Industry". 2015 年 11 月 4 日。网络来源。

使用 Java EE 等技术平台时，重要的是平台要能够跨越多种类型的环境运行，例如：

- 企业内部环境。
- 公共云（如 Amazon Web Services、Microsoft Azure 或 Google）。
- 私有云（如 OpenStack® 私有云）。
- 容器。
- 托管服务。

应用必须能够在所有环境中以相同的方式运行。一些应用平台可能无法在所有必要的环境中工作，或者在不同环境中的功能层次不齐。挑选合适的 Java EE 平台可以保障混合环境所需的互操作性。



除了混合环境的复杂性外，在多个环境中运行还会产生额外开支。Gartner 认为软件许可是 IT 部门缩减开支的一个潜在切入点，¹⁷ 也被其称之为最容易被忽视的一项开支。软件定价可能比较复杂，涵盖内部环境、虚拟环境或云的不同许可，外加各种技术支持的额外订阅服务。

¹⁷ Jim McGittigan 和 Sanil Solanki。“The Gartner Top 10 Recommended IT Cost Optimization Ideas, 2016”。技术编号：G00301094。Gartner, 2016 年 2 月 29 日。网络来源。

贴士：你的未来你做主。

并非所有 IT 基础架构都需要转为初创型、微服务架构才能实现数字变革的目标。迁移至云端、使用基于容器的部署，或者推出新型物联网或移动计划，都可能是你的正确战略规划选择。

2. 是否具备相应条件？

要想为你的环境选择最佳平台，就得参照你的战略目标权衡变革风险。每家公司的要求各不相同。数字战略应当体现公司特有的业务战略。例如：

- 你们公司的发展目标或战略方向是什么？
- 企业是否具备执行这一战略所需的技能和资源？
- 是否具备支撑这些应用运行的环境？

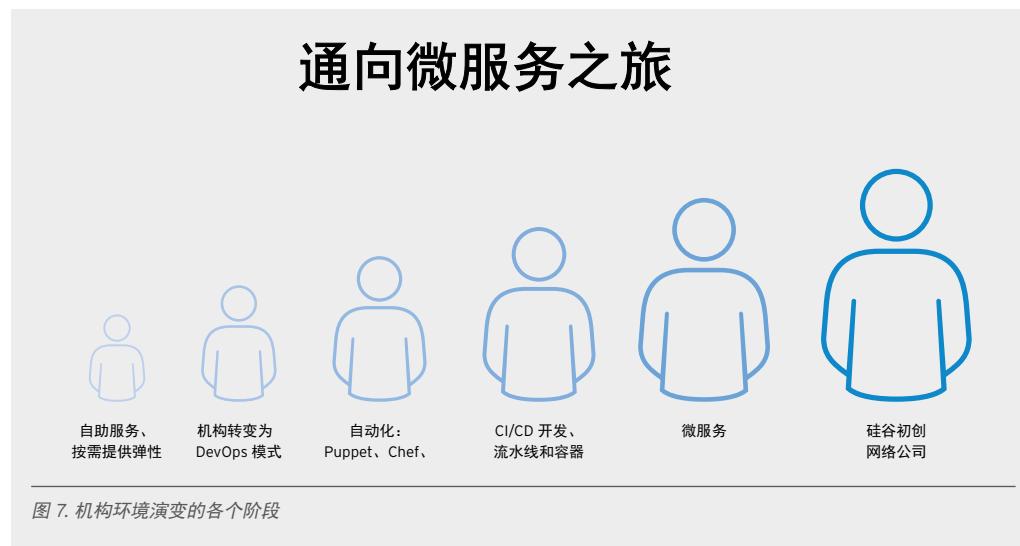


图 7. 机构环境演进的各个阶段

3. 决定现代化更新的内容和方式

IDC 的 Peter Marston 认为，数字变革的核心问题在于如何进行应用现代化更新。根据 IDC 的研究，大约 40% 的企业已经将应用现代化定为 IT 工作的头等大事，或者会在未来三年中做出这样的安排。¹⁸

实现现代化有多种方式：

- 翻新或重新架构现有应用，以适应现代化环境。
- 将应用从一个环境迁移到另一个。
- 以新的应用取代现有应用。
- 创造新旧共存的环境。

您选择的方法应取决于您的公司希望在微服务发展之路上想要到达的目标阶段。

其中一个目标是减少 IT 部门用在维护上的资源。IT 部门的经费与工作在新旧项目之间的理想划分应是均匀分配。然而，事实是：近四分之三 (72%) 的 IT 经费花在了维护和运营项目上。大多数 CTO (63%) 认为这一比例太高。¹⁹

¹⁸ Peter Marston. “Ten Criteria to Use for Application Modernization Service Provider Selection”。报告编号：IDC #US41012716。IDC，2016 年 2 月。网络来源。

¹⁹ Minda Zetlin. “How to Balance Maintenance and IT Innovation”。ComputerWorld，2013 年 10 月 21 日。网络来源。

基于 Java EE 的基础架构有助于转移维护成本。新旧应用可以在同一基础上运行，即使一个应用为一体式，另一应用属于微服务也没关系。更为重要的是，Java 应用最终能够在不同环境间顺利迁移。

这种可移植性能够减少迁移的成本和风险，同时为应用现代化开辟了道路。如此一来，你的公司依旧可以循序渐进地改变基础架构。先获得云的成本优势，再利用容器的运行效率。在与你的战略吻合之前，不必全部投入于微服务或高度分布式架构。

4. 专注于自身团队发展

当前，CIO 们面临着一大难题是：总感觉缺乏能够胜任第三代平台工作的 IT 人才。《哈佛商业评论》关于 IT 人才危机的分析报告显示，59% 的 CIO 认为技能欠缺妨碍了他们有效应对 IT 或战略挑战。²⁰ 其关注的重点领域为 IDC 提出的第三代平台：

- 大数据和分析 (36%)
- 架构，企业架构 (27%) 或技术架构 (24%)
- 开发 (27%)
- 移动开发 (24%)
- IT 战略 (22%)

来自 Forrester Research 的 Nigel Fenwick 指出，IT 部门倾向于采用定制化解决方案实现每个 IT 目标，但这不一定是最佳的战略方法。“我们为了支持通用功能已花费了数百万美元进行软件定制。结果加剧了 IT 复杂性，加大了接口难度，而且削弱了 IT 敏捷度，还增加了成本。”²¹

不要尝试面面俱到，而应通过以下办法简化你的 IT 战略：

- 确定必须实现的两至三个核心功能，作为企业的战略目标。
- 对于所有其他需求，使用基于标准的解决方案，而非定制化解决方案。
- 选择易于维护的解决方案。

CIO 们普遍认为，最大的技能缺口通常跟流程相关，而非技术。使用开放标准有助于加快学习新技术，Java 等通用技术也有此功效。将 Java EE 应用平台作为开发平台部署，IT 部门就能利用现有的知识和经验开发聚焦于新领域的项目。这将扩大潜在的人才库，因为专业开发社区里最不缺的就是经验丰富的 Java 开发人员。

5. 为开发人员提供合适的工具

在技术平台中运用以开发人员为重的整合式工具，有助于让总体开发周期更加顺畅，特别是采用整合测试或自动化时。开发人员可直接受益的几个关键领域有：

²⁰ “I.T. Talent Crisis: Proven Advice from CIOs and HR Leaders”。技术。Harvard Business Review Analytic Services，2016 年 7 月。网络来源。红帽赞助

²¹ Minda Zetlin。“How to Balance Maintenance and IT Innovation”。ComputerWorld，2013 年 10 月 21 日网络来源。

- 面向 CI/CD (持续整合/持续交付) 的整合式测试模块
- 部署自动化
- 开发工具包
- 自定义类加载
- 高性能

即便细微的性能改进也能发挥巨大的功用。例如，缩短应用部署时的启动时间可以为开发人员每周节省数小时的时间，因为常规开发中开发人员或许每天都要屡次重新启动应用。

整合测试等其他功能则是 DevOps (开发运维) 或 CI/CD 不可或缺的。一家澳大利亚信息技术公司仅仅通过将加载测试模块融合到部署流程中，就使开发人员的工作效率提高了 15%。²² 开发、测试和运营之间的整合有助于提高代码质量，缩短开发生命周期。

6. 打造生态系统

IDC 分析师 Maureen Fleming 表示，数字变革的力量在于能够整合一切资源。²³ 而第三代平台具有可以协同工作的多个层次技术，例如：

- 虚拟化、公共云和私有云。
- 容器和编配。
- 数据虚拟化。
- 内存缓存和存储。
- 多个消息传递协议。
- 适用于不同数据来源的不同数据格式。
- 管理和部署工具。
- 测试自动化。
- 业务流程自动化。

事实上，当人们设计一款适应数字变革的平台时，该平台只是整个计划的一个组成部分，而非全部。真正的目标应该是让所设计的平台和其上运行的各个应用在一个更大的生态系统中运作。

IDC 的 Peter Martson 建议，应用现代化的战略目标越大，就要寻找生态系统覆盖面越广的平台提供商提供支持。²⁴ 能够在多种技术上提供经验和指导的解决方案提供商可以带来各种形式的帮助，比如架构规划和云置备等。

²² Maureen Fleming 和 Matthew Marden。“The Business Value of Red Hat JBoss Enterprise Application Platform”。技术编号：257256。IDC，2015 年 7 月。网络来源。红帽赞助。

²³ Maureen Fleming。“Integration Is a Core Competency of Digital Transformation”。技术编号：IDC #US41293916。IDC，2015 年 5 月。网络来源。红帽赞助。

²⁴ Peter Marston。“Ten Criteria to Use for Application Modernization Service Provider Selection”。报告编号：IDC #US41012716。IDC，2016 年 2 月。网络来源。

总结

数字变革是一种 IT 战略方法，旨在将 IT 基础架构和数据转化为一种可提供给客户的潜在产品。要实现数字变革，企业就需要改变视角，不再将企业应用视为单纯服务于内部职能的工具，而是将其视为潜在的产品，并积极寻找使用数据和数据来源的新方法，和探索与客户交互的新途径。

实现数字变革须满足三点要求：

- 制定清晰的公司目标和战略
- 建立跨不同职能团队的高效协作和沟通方式
- 采用有序成熟的分布式架构

Java 应用成为企业的核心技术已有二十载。如此丰富的数据、功能和知识对企业而言非常重要。利用基于 Java 的应用，企业可在任何一个能够实现云原生架构、同时无缝支持现有应用的平台上实现现代化开发。传统的企业级、一体式应用和基于云的应用部署两者结合，可让组织充分利用现有的知识和资源，同时积极采用更新颖的应用模型。

运用云友好型 Java 应用平台可为企业带来多个方面的益处：

- 消除 IT 部门内技能欠缺或技术缺口对企业产生的影响。
- 拓展潜在资源。
- 为现有工作负载提供衔接迁移策略，同时允许在新环境中进行开发。
- 为引入其他技术以实施整合、数据管理或其他战略计划提供适合环境。

一款理想的 Java 应用平台不仅能管理现有的企业应用，也可以成为高层管理者正在规划的数字变革实现手段，还能最大化利用现有 IT 资源，同时进行关键性项目维护。

关于红帽

红帽是世界领先的开源解决方案供应商，依托社区力量为客户提供稳定可靠及高性能的云技术、Linux、中间件、存储和虚拟化产品。红帽还提供屡获殊荣的支持、培训和咨询服务。作为紧密连接全球企业、合作伙伴和开源社区的中心，红帽致力于通过为广大客户提供实用、创新型技术产品，有效释放其宝贵资源以推动业务增长，并为未来 IT 发展奠定坚实基础。

红帽产品组合 更多信息，请访问 redhat.com/zh

销售及技术支持

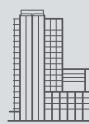
红帽软件（北京）有限公司

800 810 2100
400 890 2100

北京市朝阳区东大桥路 9 号侨福芳草地大厦 A 座 8 层 邮编: 100020
8610 6533 9300

版权所有 © 2016 Red Hat, Inc. Red Hat, Red Hat Enterprise Linux, Shadowman 徽标和 JBoss 是 Red Hat, Inc. 在美国和其他国家/地区注册的商标。Linux® 是 Linus Torvalds 在美国和其他国家/地区注册的商标。

OpenStack® 文字商标和 OpenStack 徽标是在美国和其他国家/地区注册的商标/服务标志或 OpenStack 基金会的商标/服务标志，拥有 OpenStack 基金会的许可才能使用。我们不隶属于 OpenStack 基金会或 OpenStack 社区，也未获取他们的支持和赞助。



红帽官方微博



红帽官方微信

cn.redhat.com
INCO441724_1016