



敏捷集成：

企业架构蓝图

电子书

作者：Steve Willmott 和 David Codelli

编辑：Deon Ballard

目录

计划已经消亡：组织与敏捷性.....	4
敏捷性所依赖的基础架构.....	6
分布式集成.....	7
容器.....	9
API.....	10
敏捷集成的架构.....	12
团队实践.....	12
基础架构的框架.....	12
敏捷组织与文化.....	14
总结：实现敏捷集成.....	18

计划已经消亡：组织与敏捷性

敏捷性所依赖的基础架构

分布式集成

容器

API

敏捷集成的架构

团队实践

基础架构的框架

敏捷组织与文化

总结：实现敏捷集成

企业的成功越来越依赖于应对变化的能力。随着新的颠覆性企业进入市场，而且技术的发展改变了消费者的期望，企业更改计划的周期相比以往越来越短。现代软件架构和流程可以使企业更有效地应对这种变化，成为市场中的赢家。

名为“敏捷集成”的全新架构框架将三项重要的架构能力结合在一起：容器、分布式集成和应用编程接口 (API)，并阐明了这些关键能力如何提升敏捷性，为企业内的新流程提供强大的竞争优势。

旅游和酒店等行业已经通过新的业务开展方式实现了转型，现在已经能够提供新的服务，而且消费者与服务的交互方式也与以前不同。受到新技术的推动，外加企业和客户交互式思维方式的影响，这种颠覆性变化正不断扩展到其他重要行业，比如金融服务业和政府机构。这些挑战正在推动现有企业和机构彻底转变自己的 IT 技术，以提供这些新的服务。

为了立于不败之地，企业必须快速规划并对软件系统执行更改。

要想以当前所需的速度交付软件，企业需要敏捷的基础架构作为基础。这里的敏捷指的不是敏捷软件开发，而是它的传统含义——灵活且能够快速移动。

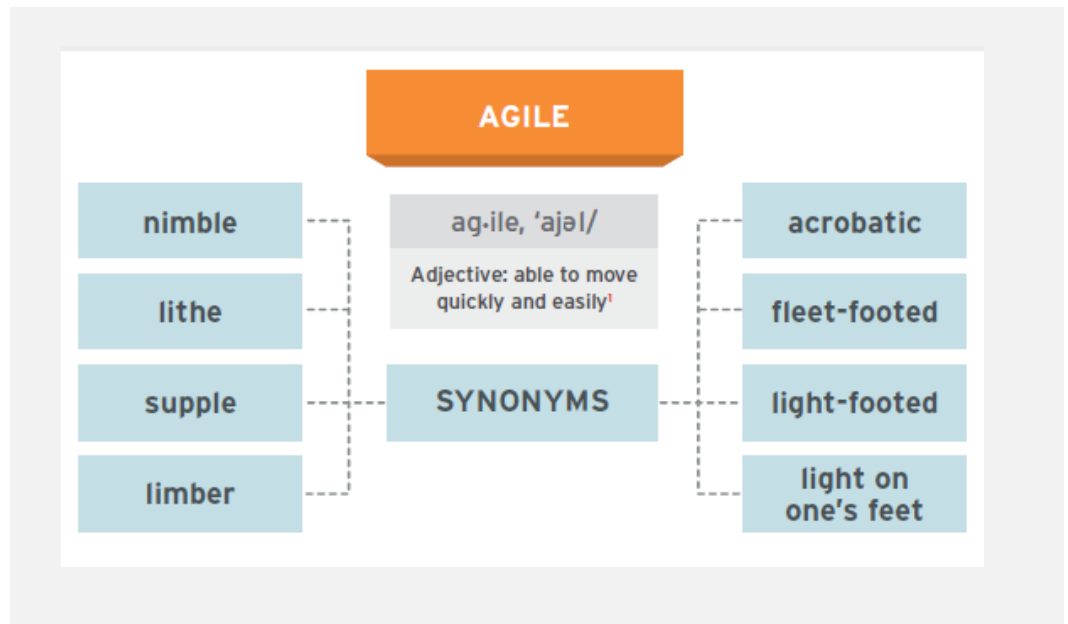


图1.敏捷的定义

¹ 牛津英语词典

“要想持续赢得、服务和留住客户，敏捷性必不可少，这要求交流系统和记录系统之间的接口变得更加敏捷——不仅表现在扩展能力上，也表现在快速适应能力上，例如为现有 API 添加新属性，在将来提供更多的上下文等。”

HENRY PEYRET
THE FORRESTER GROUP

Peyret, Henry.
“TechRadar™: 集成技术。
2015 年第 2 季度。”
Forrester Research 公司。
2015 年 6 月 23 日。

迄今为止，敏捷方法主要应用于软件开发，目的是改进并精简应用的创建方式。DevOps² 实践试图将这种方法应用到这些应用的部署中。

然而，一般情况下，DevOps 本身仅主要针对企业自己开发的新软件应用。

基础架构在敏捷性方面进一步发展，创建了一个涵盖所有 IT 系统的环境，包括传统软件。敏捷基础架构理念旨在解决现有系统、不同数据类型、数据流和客户期望的复杂性，并找到一种方法将这些方面统一在一起。从核心来讲，这是一个集成问题。

与需要三个月分阶段采用手动验证步骤推出新产品的企业相比，能够在一夜之间更改定价或者一夜之间推出新产品的企业无疑更有优势。

我们称之为敏捷集成。集成不是基础架构的一个子集，而是基础架构的一种概念性理念，其中包含数据、应用、硬件和平台。集成技术与敏捷和 DevOps 技术的结合可以创建一个平台，让您的团队能够根据市场需求快速做出改变。

计划已经消亡：组织与敏捷性

红帽公司 CEO Jim Whitehurst 在 2017 年红帽峰会上做的主题演讲中指出：“众所周知，计划已经消亡。在一些不太了解的环境中，计划的效率非常低下。”³ 随着业务环境的运行速度逐步加快，而且重大变更持续发生，计划会迅速被打破，而受限于一行动方案可能会付出巨大代价。

这意味着您掌握的信息越少，或者您的环境越不稳定，计划的价值就越低。

您不知道自己不知道什么

基础架构规划通常是一个长期的过程，有时跨越多年时间。如果尝试制定多年计划，可能会扼杀企业随着市场变化而创新或转型的能力。Jim Whitehurst 提到的计划“消亡”最终可以归结为更快制定计划和执行计划的能力，这会令计划的期限缩短，有利于孕育新的计划。

当团队已经习惯于 6 个月甚至 24 个月的开发周期时，这种快速变化可能带来巨大的挑战。对于一些采用更传统结构的企业而言，当他们必须以全新的方式与正在开辟市场的初创企业进行竞争时，这一问题就会加剧。Netflix 与 Blockbuster，或者 Uber 与传统的出租车服务企业，就是很典型的例子，但初创企业的颠覆性影响可追溯到信息时代的初期，即从 1993 年的亚马逊或 20 世纪 80 年代的个人电脑开始。

² 采用 DevOps 实现更快创新 <https://www.redhat.com/en/insights/devops>

³ Jim Whitehurst 在 2017 年红帽峰会上的主题演讲
<https://www.cbronline.com/news/enterprise-it/software/red-hat-ceo-planning-know-dead/>

表 1：不同行业的颠覆者

行业	传统服务	颠覆者	影响
交通	出租车、公共交通	Uber、Lyft	创造统一的客户体验，而这是小型本地企业难以做到的
财富管理	投资公司	自动基金管理	将基金管理的差异因素从人员转变为算法
零售	实体购物	Amazon	将购物习惯从线下转变为线上购物
搜索引擎	Google、基于浏览器的搜索	语音搜索	影响 Google 的主要市场渠道，并且允许新进入者的加入

初创企业和颠覆者的优势在于他们可以自由构建基础架构、团队、应用、架构、甚至部署流程。他们不仅有创新的想法，而且关键是他们能够执行这些想法，因为不受传统基础架构（或者如 Rachel Laycock 所称的“守旧人”）制约，⁴所以可以实现敏捷运营。

除了构建新设施的能力之外，这些企业还构建可以应对变化的系统。软件基础架构是其差异化能力的一部分，并且系统的几乎任何部分都可以根据不断变化的市场需求而更换、更新或删除。随着初创企业走向成熟，一些企业的适应能力下降，但最好的企业不惜一切代价确保自身能力应对变革。

迎接挑战

要在快速变化的环境中取得成功，整个 IT 基础架构必须以敏捷的方式运行。

变革需要在两个层面上发生：

- 从组织和文化角度为敏捷流程提供支持——从架构设计到团队沟通。
- 能够提供快速升级、添加和移除能力的技术基础架构。

技术和文化变革不会带来敏捷性，而是敏捷性的基础。

eBay 产品经理 Marty Cagan 对每个例行项目都实行了他称之为“收税”的做法——从每个例行项目中省出一些时间和资源，以开展新的基础架构项目。⁵这使得新项目和创新能够优先实施。

4 Rachel Laycock, [“持续交付”]红帽峰会下午分会 - DevNation2016。2016 年 7 月 1 日，加利福尼亚州旧金山。
<https://youtube.com/watch?v=y87SUS0fgTY>

5 Cagan, Marty. “灵感：如何创建客户喜爱的产品”。WileyPress, 2017 年

“如果您无法在产品上市时间和敏捷性方面击败竞争对手，您就会陷入困境。推出新功能始终是一场赌博。如果您幸运的话，有 10% 的机会将获得理想的收益。因此，越快将这些功能推向市场并加以测试，您的收益就越高。有时，您也可以更快获得投资回报，这意味着企业可以更快开始赚钱。”

GENEKIM
凤凰项目

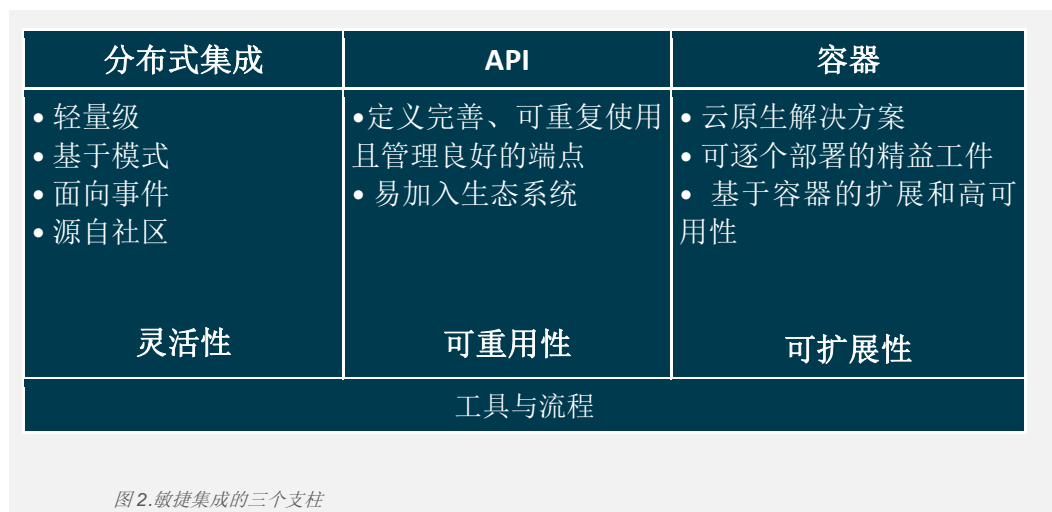
GeneKim、KevinBehr 和 George Spafford。“凤凰项目：一本关于 IT、DevOps 以及帮助企业成功的小说”。俄勒冈州波特兰：IT RevolutionPress, 2013 年。

敏捷性所依赖的基础架构

如果在探索改进方案的过程中，不同的团队朝着不同的方向发展，那么大量新技术的引入往往无助于构建敏捷的基础架构。如果没有一套完整的总体目标，就很难确定哪些新能力会对组织的整体运营产生真正的影响。

敏捷集成的三个方面

敏捷集成理念需要三个主要技术的支撑。



- 1. 分布式集成。**以几十个总体集成模式来反映企业的工作和数据流。当这些集成模式在容器中部署时，可根据特定应用和团队所需的规模和位置来部署。这是一个分布式集成架构，而不是传统的集中式集成架构，而且它允许各个团队定义和部署敏捷性所需的集成模式。
- 2. API。**稳定、良好管理的 API 对团队、开发和运营之间的协作具有巨大影响。API 将关键资产封装在稳定、可重用的接口中，允许这些接口作为构件在整个企业、合作伙伴和客户中重复使用。API 可以与容器一起部署到不同的环境中，从而允许不同的用户与不同的 API 集进行交互。
- 3. 容器。**对于 API 和分布式集成技术而言，容器作为底层部署平台而运行。容器允许将准确的服务部署在特定的环境中，使开发、测试和维护工作能够轻松且一致地进行。由于容器是 DevOps 环境和微服务的主要平台，因此，将容器用作集成平台可以在开发和基础架构团队之间建立更加透明的协作关系。

这三种技术使 IT 基础架构更加敏捷，因为它们分别提高了抽象级别，使不同团队可以互相协作。使用带有 API 和分布式集成能力的容器平台，将集成的实施过程与集成本身分离。团队的敏捷性得以提高，因为 API 和分布式集成模式能够以易于理解的方式将特定资产打包，而无需理解或更改底层基础架构。

如果分别使用，每项技术都将为特定的集成挑战提供显著的敏捷性。如果结合使用，则可以带来乘数效应。技术有赖于文化：与 DevOps 实践结合，尤其是自动化和部署流程，可以让该技术的优势进一步增强。

分布式集成

当前 IT 系统面临的巨大挑战之一是需要连接来自不同企业的应用。集成服务的难度导致集中式集成中心变得日益复杂。这些中心通常以企业服务总线 (ESB) 的形式实施，已成为非常复杂的瓶颈，而且过于僵化，无法快速应对变化。

分布式集成实现了前几代 ESB 的多个相同技术目标，但更适应企业内的团队。与 ESB 一样，分布式集成技术也提供转换、路由、解析、错误处理和告警能力。不同之处在于集成的架构。

分布式集成架构将每个集成点视为独立且独特的部署项目，而不是较大的集中集成应用的一部分。然后，可以对集成项目进行容器化，并在本地部署给特定项目或团队，而不影响整个企业内部部署的其他任何集成项目。分布式理念可以提供敏捷项目所需的灵活性，而且通过使用底层容器平台，与敏捷或 DevOps 团队使用相同的工具链，从而提高团队使用自己的工具和日程表管理集成项目的的能力。从根本上讲，这将集成看作是一种微服务，⁶提高了开发和发布集成项目的速度。

与开发人员工具和流程保持一致至关重要。分布式集成的一个核心方面在于，它不是由一个部门的专业用户开发和管理而与软件开发流程分开部署的集中软件基础架构。通过一个通用的平台和工具来分发集成架构，整个项目的所有开发人员都可以接入这个基础架构，并且可以随时随地根据集成需求来支持轻量级部署。

⁶ 参考 Martin Fowler 的对微服务的定义：<https://martinfowler.com/articles/microservices.html>

“在软件中，如果某些方面令你感到痛苦，则减少痛苦的方法是更频繁地去做，而不是逃避。”

DAVID FARLEY

持续交付：通过构建、测试和部署自动化而实现可靠的软件发布

David Farley 和 Jez Humble, 持续交付：通过构建、测试和部署自动化而实现可靠的软件发布, Addison-Wesley Professional, 2010 年

表 2. 软件生命周期各个阶段的集成技术比较

生命周期阶段	ESB, 大多数集成平台即服务 (IPAAS)	支持分布式集成技术
版本控制	专有	Github 等
构建	专有	Maven 等
部署	专有	容器和其他 DevOps 工具
管理和扩展	专有	容器和其他 DevOps 工具

要使用 ESB, 除了在开发和运营环境中使用原本所需的工具外, 团队还必须在整个生命周期内使用该 ESB 附带的工具。这种限制会导致繁琐、低效和容易出错的操作。

依靠消息处理来增强集成能力

从架构上来讲, 分布式集成将集成视为微服务, 可以进行容器化处理, 易于在本地部署, 并且可以快速发布。

集成技术需要能够支持这种轻量级、基于微服务的架构。红帽® Fuse 允许用户将集成视为代码, 可以在任何地方运行, 包括在容器中。

此外, Fuse 与红帽 JBoss AMQ 捆绑在一起, 以提供消息基础架构。强大的消息基础架构可确保事件和数据在系统之间有效地路由。消息处理是微服务的重要架构工具, 因为它的异步性质不需要依赖关系。

通过提供更有效的路由、对多种语言和协议的支持、异步吞吐量和更好的数据管理, 集成和消息传递的这种组合提高了集成架构的整体性能。

“一种新的竞争形式（通常被称为数字化转型）正在推动企业重新思考 IT 架构，在本地基础架构、云端和物联网中重新分配工作负载，并通过互操作来支持业务战略和运营。所有这些变化都需要新的集成方法——我们称之为‘混合整合’”。

CARLEHMANN
451GROUP

Carl Lehman, 451 Research,
“集成 PaaS 和 API 在新的混合集成平台市场中的颠覆角色”。2017 年 7 月。<https://451research.com/report-long?icid=3862>.

跟上趋势

容器的采用率不断提高——但提高了多少？为什么？451 Research 预测市场的增幅达 250%⁷，但这是指支出，而非部署数量。实际部署数量不太容易衡量。红帽委托 Bain 进行的调查发现，目前大约 20% 的客户在生产环境中部署容器，而在开发和测试环境中的部署数量大致相同，超过 30% 的客户正在评估容器或运行概念验证。⁸

阻碍容器普及的部分原因在于企业对使用容器的意义不太明确。Enterprisers Project 概述了采用容器的四种不同模式：用作一般开发或部署平台，用作云本地或微服务平台，用在混合云中，或用于创新项目。⁹您使用容器的方式可能会影响您如何看待其采用情况。

对于敏捷集成，关键是创建一个支持现有操作的基础架构平台。该平台可以借鉴各种实施模式，但其核心是用作一个平台，作为新项目 and 现有服务的基础。

容器

虚拟化、云和容器是类似的技术，试图实现类似的目标。这些技术将软件的操作环境从物理硬件抽象出来，从而可以在硬件上堆叠更多实例，并更有效地管理利用率、扩展和部署。然而，它们以不同的方式应对这一挑战：虚拟化实现操作系统层的抽象化；云移除了永久专用服务器实例的概念；容器则定义了运行单个应用所需的操作环境和库。

容器技术所描述的方法更具说服力、更加轻量级，因而使容器成为现代化软件环境的理想工具。每个实例使用一个不变的定义，包括从操作系统到每个库的确切版本。这使得每个实例的环境都具有高度可重用性和一致性，这对于持续集成和持续交付 (CI/CD) 管道来说非常理想。此外，由于容器映像仅定义单个应用程序所需的内容，因此，容器与微服务相匹配，而容器编排也可以协调大型微服务基础架构的部署和管理。

轻量级和可重用性的结合使容器成为敏捷集成的理想技术平台。

7 451Research 信息图基于《云实现技术监测》报告，2017 年 1 月

https://451research.com/images/Marketing/press_releases/Application-container-market-will-reach-2-7bn-in-2020_final_graphic.pdf

8 Bain 调查：传统企业的数字化路径和容器角色。2016 年 11 月。

<https://www.redhat.com/en/resources/path-digital-containers>

9 <https://enterprisersproject.com/article/2017/8/4-container-adoption-patterns-what-you-need-know>

传统的集成方法具有高度集中的结构，ESB 位于基础架构的主要部署点。分布式集成和 API 管理都采用分散式架构，只将所需功能部署到特定位置或团队。容器作为这两种方法的底层平台，因为具有不变性，镜像和部署过程在各个环境中保持一致，因此可以快速部署或更换，而不存在模糊依赖性 or 冲突。

无论是对于集成还是 API，分布式架构的关键在于，需要一种用于设计和部署新服务的方法，而无需复杂的审批流程。

容器允许分布式集成和 API 都被视为微服务，为开发团队和运营团队提供了一个通用工具，并且能够使用快速开发流程和代管发布流程。

容器需要编排

每个容器代表单个服务或应用，就像微服务代表一项单独功能一样。微服务架构中可能有几十个甚至数百个单独的服务，而且这些服务在开发、测试和生产环境中都可重复使用。

由于实例数量众多，编排实例和执行高级管理任务的能力对于容器环境的有效性至关重要。

红帽 OpenShift 将 Docker 容器与 Google 的 Kubernetes 编排项目结合在一起，并包含集中式管理，例如实例管理、监控、日志记录、流量管理和自动化，而这在仅有容器的环境中几乎是不可能的。

红帽 OpenShift 还提供了方便开发人员使用的工具，如自助式目录、实例集群、应用持久性和项目级隔离。

这种组合平衡了运行要求，特别是稳定性和测试方面的要求，以及开发人员对于易用性和快速交付的要求。

API

大多数信息基础架构都包含数百甚至数千个系统、应用和资产，但这些系统可能很难交互，甚至 IT 管理员可能都无法知道哪些系统可用。

API 是可以采用集成技术连接所有资产的接口。API 是一组定义或规则，用于设置应用如何相互通信。

随着组织从基于集中集成技术中心的方法转向分布式方法，自助服务成为最优先考虑的方案。敏捷团队需要授权和自主性，以寻找、测试并使用公司内外开发的服务。强大的 API 能力为这些团队提供了这种授权和自主性。借助 API，团队可以实现所需的集成，同时企业可以确保安全、授权和使用策略得到适当管理和实施。API 为团队如何设计集成项目提供了参考。

API 不同于最终应用，它确定应用如何交互，然后由开发人员将这些 API 用作项目中的构块。API 为开发人员和团队提供了一种通用语言。企业组织甚至可以使用 API 建立共享和合作社区，从而为服务创建新的创新用途。

不同的 API 或 API 的不同子集可以提供给不同的受众。供应商的需求可能与内部开发团队或社区开发人员的需求不同。API 管理包括为应用和用户组设计 API，以及 API 生命周期的管理。API 越来越多地作为产品进行管理，不同的团队负责每个 API，但需要确保所有这些资源的一致性和易用性。

与分布式集成一样，容器可以作为开发、部署和管理 API 的平台，使 API 的开发与更大的开发和运营流程及工具保持一致。

适当的 API 平台可帮助您的开发人员完成更多工作

API 所能发挥的作用取决于其他人使用这些 API 的能力，无论是内部开发人员还是外部用户。红帽 3scale API 管理平台提供的工具可为所有用户提供帮助。它为开发人员提供了一个开发人员门户，用于协作创建 API，并提供了管理员门户，用于发布这些 API。

3scale API 管理平台通过提供身份验证，与主要云提供商集成并且在容器内运行，使得这些 API 可在外部使用。

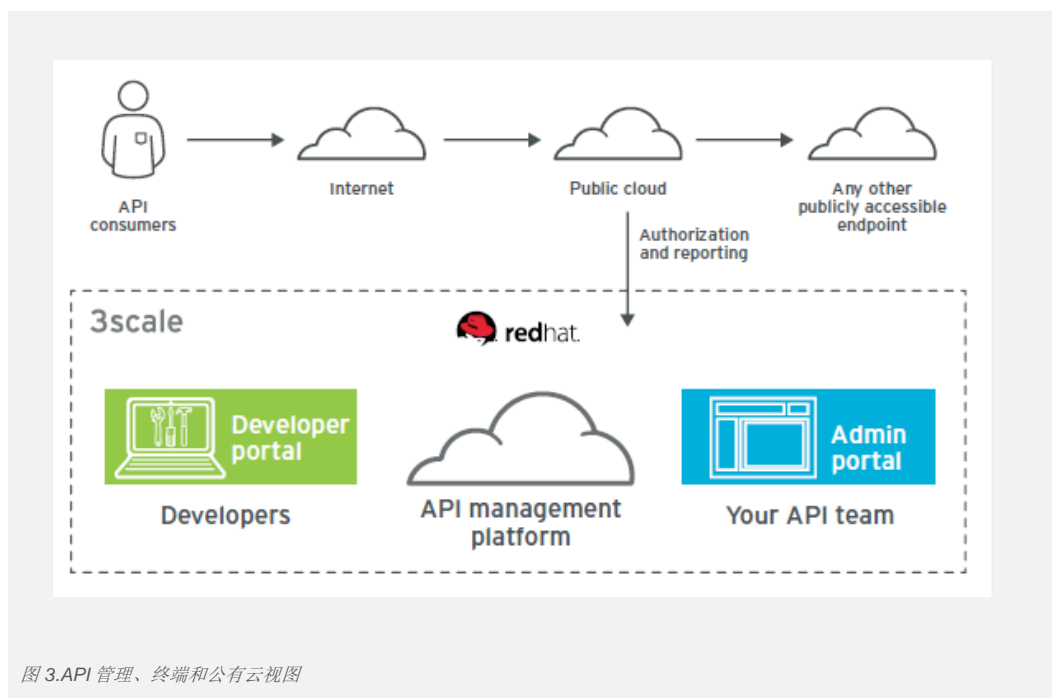


图 3.API 管理、终端和公有云视图

API 策略将 API 的设计与发布方式结合起来。3scale AP 管理平台（尤其是容器平台上的 3scale）提供了执行这一策略的方法。

敏捷集成所依赖的架构

团队实践

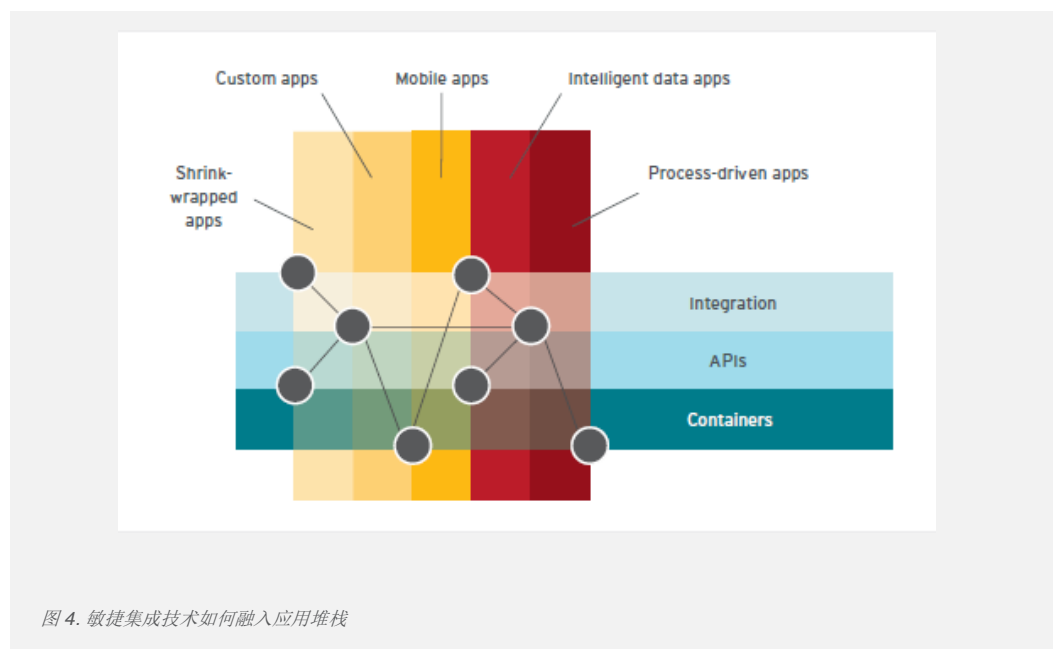
敏捷集成所依靠的技术如果作为可重用的能力提供给团队，将发挥最大的效用。

此处所说的能力是指授权群体能够以自助服务方式使用这些技术，轻松地遵循企业的指导准则，并获得最佳实践信息。信息架构师或 IT 管理员必须为各个团队定义清晰的流程，例如：

- 提供广泛可用的使用指导准则。
- 在适当情况下执行使用规则和最佳实践，但除这些规则外，还允许自由尝试其他规则。
- 定义明确的流程，包括原型设计、测试、上线、更新和淘汰。
- 允许共享新部署和开发信息。
- 将基础架构团队作为自助服务能力的实现者和提供者，而不是迫使他们参与每一个流程。

例如，软件团队应该能够以完全自助服务的方式开发、测试和准备待发布的新 API，并有适当的流程来更新其他群组 and 文档。在发布或转入生产环境之前，可能会有与其他团队配合并进行交叉检查，但基础架构应尽可能实现这一流程的自动化。

基础架构的框架



容器、API 和集成密切配合，为企业的内部软件生态系统提供坚实的基础层，并在许多情况下为外部集成提供接入点。

不同类型的系统展现多个可重用的端点，而每个端点都可视为可重用的 API，而且许多运行在容器内，以实现可扩展性和易部署性。通过集成一组服务或收集企业不同部门的结果，集成项目可通过系统实现所需的转换、组合或内联业务逻辑。

在为最终用户应用提供服务之前，集成后的应用可进一步合并。

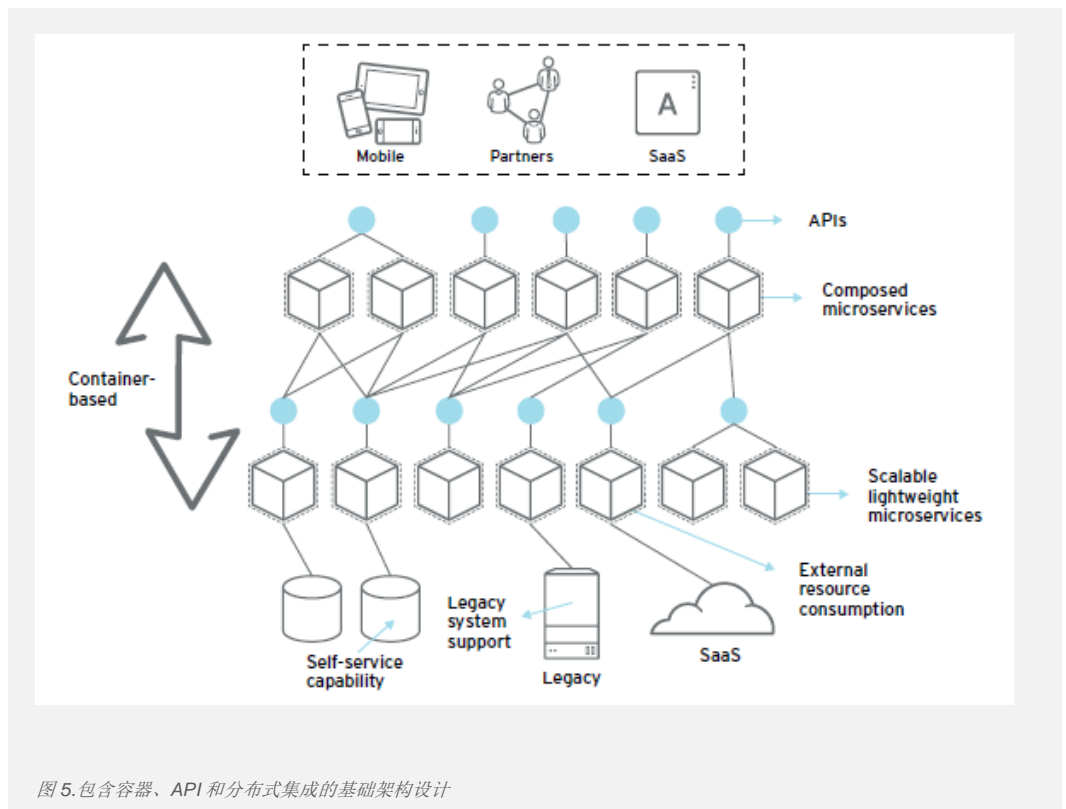


图 5. 包含容器、API 和分布式集成的基础架构设计

不存在这样的假设：所有系统将被分解为越来越小的片断，或者经过多层 API 抽象。这样的操作可能会降低效率，增加延迟或提高不必要的复杂性。在某些领域，通过保留现有传统 ESB 功能来保持特定应用之间的连接可能是正确的选择。分布式系统之间的依赖关系也需要使用适当的工具进行跟踪和管理。

然而，对于整个系统来说，针对容器、API 和集成而改变架构意味着可以为每个服务、集成点和客户交互做出正确的选择。例如，可以对大量入站请求进行安全检查，然后直接路由到正确的后端服务，而不会遇到单个 ESB 瓶颈。

在混合的分布式云环境中，许多相关的后端系统可能位于不同的物理位置。通过将本地接近的系统集成在一起以满足本地需求，这要比通过一个包含关键业务逻辑的单一中央集成系统传送所有内容更高效、更安全。

敏捷组织和文化

基础架构的生命周期与软件开发或运行的生命周期截然不同。开发周期是完成一个项目，然后转到下一个项目，效率意味着加快产品发布或者在指定时间内创建多少个新特性。即使对于更关注维护和稳定性的运营方式而言，更有效而且更快地应用安全补丁和更新、部署新服务或撤销更改仍然很有用。

但是，基础架构采用了截然不同的方法。基础架构倾向于在更长的时间段内运行，并且与不同的高度专业化小组合作，这与开展特定软件工程项目的跨职能团队完全不同。基础架构项目通常比软件项目的规模大得多，这意味着在较短发布周期内可能无法完成太多工作，或者可能导致结果不完整。如企业 IT 专业人士 **Andrew Froelich** 在《InformationWeek》中所写的那样，基础架构有一个无返回限制的点，特别是对于硬件和数据中心，但即使对于公有云，在达到某个时间点后，不能放弃项目并重新开始。¹⁰ 基础架构是永久的。然而，您可以根据基础架构的性能而调整所用的方法。

敏捷和 DevOps 等快速响应式迭代流程的优点对于开发和运营团队来说显而易见，而对于基础架构团队不太明显。然而，**Froelich** 在对基础架构敏捷性的优缺点进行分析时忽略了一个关键方面：基础架构团队与开发和运营团队的协调一致。**Rohan Pearce** 在《CIO》中介绍了要将基础设施团队转变为敏捷式工作单元，而不是职能团队。¹¹ **Telstra Enterprise Services** 团队让开发团队忽略内部系统，因为检查系统或进行更新的过程非常痛苦和复杂。通过调整工作组，他们的开发周期从 212 天减少到了 42 天。¹²

本例说明了基础架构团队如何更有效地服务内部团队，从而实现关键流程的改变。

敏捷集成技术可以支撑更灵活的基础架构。API、容器镜像和分布式集成已经成为软件基础架构中新的热点。

¹⁰Froelich, Andrew. “IT 是否应提高敏捷性? 利弊分析”。2015 年 10 月 6 日。

<http://www.informationweek.com/infrastructure/pc-and-servers/should-it-go-agile-the-pros-and-cons/d/d-id/1322448>

¹¹ Pearce, Ronan. “基础架构能否实现敏捷性?” 2013 年 6 月 20 日。

https://www.cio.com.au/article/465436/can_infrastructure_agile/

¹²<http://agilemanifesto.org/>

敏捷宣言为软件开发定义了四个核心原则。¹²在基于集成技术的敏捷基础架构中，这些原则可应用于集成战略。

1

个人和交互优先于流程和工具。

对于基础架构，讨论的重点是团队之间的交互。交互包括直接沟通、由 API 管理、消息传递和流量模式；系统层面的相互依赖性；以及测试和发布流程，例如 CI / CD 管道。

2

可运行的软件优先于全面的文档记录。

基础架构的性质要求其必须 24/7/365 全天候运行，宜采取逐步调整而不是重大改变。从这个意义上说，运行的基础架构始终是一种隐含的要求。作为基础架构战略，“运行”意味着基础架构组件在预期的性能范围内提供预期的最终用户行为。

3

客户协作优先于合同谈判。

对于基础架构系统，合同代表基础架构团队如何管理系统依赖性，如安全策略、服务等级协议，甚至已发布的 API。客户包括这些系统的内部和外部用户。敏捷性使这些用户能够对与系统相关的策略和界面潜在变化发表看法，并让他们看到这些变化更快得到执行。通过使用分布式集成，开发和部署集成项目的控制权直接交给了团队，这样扩展了协作范围。

4

响应变化优先于遵守计划。

这是技术支持流程的原则。对于基础架构，系统应该保持稳定，但像容器这样的新技术提供了一个弹性的平台。实例可以根据需求动态添加和删除，部署和更新可以自动执行，也可以在多个实例间协调变更项目。已公布的 API 定义提供了可重复使用的工具，使开发流程更加一致。这种方法构建了一个可适应变化的稳定平台。

图 6.敏捷宣言指出了软件开发的核心理念

敏捷集成使用技术来支持基础架构团队中的文化变革。它是基础架构战略的基础，并将基础架构技术及其团队，与开发和业务战略更紧密地结合在一起。

敏捷方法指出了软件项目中的一些关键部分，例如个人、构建和依赖关系。然后，它可以定义这些元素之间的关系。在将集成基础架构作为敏捷项目进行实施时，可以识别与敏捷方法类似的元素和关系，例如团队、容器镜像、API 和集成点。表 3 描述了其中的一些相似点。

表 3. 软件敏捷性和基础架构敏捷性要素的对比

项目	企业	说明
个人	团队	团队负责基础设施的特定部分。这样可以确定团队职责，例如团队管理的系统或 API、团队领导以及团队的目标。
模块	API	精心定义的接口 (API) 在一段时间内保持稳定，拥有自己的路线图，由特定团队运行，并创建针对企业内部的重要能力。
构建	容器镜像	发布基于经过测试、标记的可部署单元，并且可由任何有权访问的团队可靠地部署。这取代了一体化、有版本号的代码。
编译依赖关系	集成	此元素标识这些分布式系统中不同组件之间的集成和映射。这些集成点可以像系统中的其他任何部分一样进行管理、调试、淘汰、版本控制和测试。
构建测试	基础架构自动化	这是完整的生命周期管理，涵盖从测试软件构建、性能和用户需求到运行和监控多个系统的能力。

将敏捷原则应用于基础架构计划

大多数变更管理方法要求所有子系统都有全面的文档记录。该文档必须详细介绍系统的各个方面，包括监控方法、性能参数、负责的团队。敏捷原则需要协作和适应性，这与需要大量文档的变更管理流程相冲突。

定义一套可用于评估变更请求和计划的准则和标准，而不是试图规范地定义所有潜在的利益相关者、变更和系统组件。应考虑以下问题：

- 用户预期的端到端体验是什么？
- 每个成员（每个团队、API 和系统）如何帮助逐步增强这种体验？
- 如何定义监测和警报，为保持服务等级需要设置哪些参数？
- 验证预期行为需要哪种自动化测试？
- 为了在不中断用户体验的情况下测试和部署子系统的新版本，团队可以使用的发布管道是什么？
- 组件服务中的故障会对整个系统的服务等级产生什么影响？

敏捷基础架构内的变更管理应该更多关注持续协作，而不是合同。

成功的几率有多大？

您的 IT 项目有多大的成功可能性？首先，这取决于您是否了解成功的标准——是否做到符合规范、增加客户的采用率，还是仅仅发布项目就撒手不管了？项目管理培训集团 4PM 将成功定义为按预算、按时和按规格完成项目。¹³ 根据这个定义，4PM 估计大约 70% 的 IT 项目会失败。¹³ 这一数字已经开始改变。项目管理协会最近的调查显示，与过去五年相比，更多的项目达到了计划的目标。¹⁴ 这归功于 IT 和业务团队之间更密切的配合，从而有助于更好地了解战略和客户需求。⁸

之所以能达成这种战略一致性，原因之一在于组建了敏捷团队。敏捷性有利于协作和反馈，以整体视角看待问题和系统，发现创新方法。

¹³ 4PM.com, “项目为何频繁失败”。2015 年 9 月 27 日。

<http://4pm.com/2015/09/27/project-failure/>

¹⁴ Florentine, Sharon, “IT 项目的成功率总算提高了”。2017 年 2 月 27 日。

<https://www.cio.com/article/3174516/project-management/it-project-success-rates-finally-improving.html>

拥有共享技术堆栈后，讨论的焦点从独立代码转移到了系统及其相互依赖关系。这是系统级思维，它将整个软件基础架构作为单个系统，包括内部开发的软件、供应商系统以及它们之间的连接。API 和消息系统可以跨越整个基础架构，并努力实现软件系统的统一性。

由于 API 和分布式集成可由单个开发或运营团队开发和理解，因此对团队职责会有更清楚的认识。集成本身也可以更好地被理解，原因是负责开发和部署的团队认识到了系统和应用程序之间的相互依赖关系。

将集成作为基础架构的基础，然后为不同的团队分配集成责任，这创建了一种更适合采用敏捷方法的基础架构环境。

总结：实现敏捷集成

敏捷性是一个过程，而不是单个项目。

企业应对市场变化的能力从未像现在这样重要，从很大程度上讲，IT 系统必须提供这种能力，这样才能快速启动新服务或更新现有服务。重新思考 IT 基础架构比以前更加重要，因为这是数字化服务的基础。

由于需要降低风险并保持稳定性，基础架构团队过去需要完成一个冗长的调整过程。然而，基础架构的理念可以从基于硬件或平台转变为基于集成。集成不是基础架构的一个子集，它是基础架构的概念性方法，包括数据、应用以及硬件和平台。

我们将此方法定义为敏捷集成，这是使用集成技术来创建更灵活、更具适应能力的基础架构的一种方式。敏捷集成有三个技术支柱：

- **分布式集成**：使用消息和企业集成模式来集成数据和系统。这些集成项目根据需要，在项目和接触点之间被分解为一个个由团队驱动的小集成项。
- **内部 API 管理**：创建了一组可重用的接口，以便开发团队能够与应用和系统进行交互。API 为应用的交互提供了指导和结构。
- **容器**：允许集成项目与开发和运营项目紧密结合，并使得开发、测试和发布的集成与使用 DevOps 方法执行的软件项目类似。

E-BOOK 电子书 敏捷整合：企业架构蓝图

技术必须用于支持文化变革，这意味着要使基础架构团队更加敏捷，而不仅仅是他们的软件。随着基础架构团队与敏捷原则保持一致，可以逐渐引入技术来支持这些变革。没有一个项目可以将整个企业改造得具有敏捷性。更加有效的方法可能是实施一种敏捷集成技术或改变业务的一个领域，然后在更大范围内逐步扩展这些变化。

增强 IT 基础架构对变化的响应能力是一个长期的战略目标。无需在整个企业内开展彻底的变革即可实现进步，甚至可能没有必要尝试进行单独的变革然后逐步推广。

在技术和组织层面，敏捷集成提供了一个框架，可帮助重塑 IT 基础架构。

关于红帽

红帽是全球领先的开源解决方案提供商，公司采用基于社区的理念提供可靠且高性能的云、Linux、中间件、存储和虚拟化技术。红帽也提供屡获殊荣的支持、培训和咨询服务。作为全球企业、合作伙伴和开源社区网络的连接中枢，红帽帮助创建相关的创新技术，将资源用于推动增长，并且帮助客户为满足未来的 IT 要求做好准备。



facebook.com/redhatinc
[@RedHat](https://twitter.com/RedHat)

linkedin.com/company/red-hat

redhat.com
f11423_0518

北美洲
1 888 REDHAT1

欧洲、中东和非洲
00800 7334 2835
europa@redhat.com

亚太地区
+65 6490
4200apac@redhat.com

拉丁美洲
+54 11 4329 7300
info-latam@redhat.com