



技术雷达

2013 年 5 月

由 ThoughtWorks 技术顾问委员会编写

thoughtworks.com/radar

ThoughtWorks®

有什么新鲜事？

以下是本期重点介绍的趋势：

- 拥抱下降的界限——不管你喜不喜欢，你周围的界限正在下降。我们选择通过研究无周边企业、云中的开发环境和远程呈现的协同定位等概念来接受这一点。
- 将行之有效的实践应用到不知何故忽略了它们的领域——我们不确定为什么，但是我们行业中的许多人都忽略了一些想法，例如捕获客户端 JavaScript 错误、移动设备的持续交付、NoSQL 的数据库迁移以及 CSS 的框架。
- 用于分析的轻量级选项——数据科学和分析不仅仅适用于拥有该领域博士学位的人。我们强调协作分析和数据科学，所有开发人员都了解基础知识，并与专家密切合作必要时。
- 基础设施即代码——持续交付和开发运维提升了我们对基础设施的思考。将基础设施视为代码的含义以及对新工具的需求仍在不断发展。

思想工作者对技术充满热情。我们构建它，研究它，测试它，开源它，撰写它，并不断致力于改进它——为了所有人。我们的使命是支持优秀的软件，并对其进行革新。我们创建并共享 ThoughtWorks 技术雷达来支持这一使命。ThoughtWorks 技术顾问委员会是一个由 ThoughtWorks 的高级技术领导者组成的小组，负责创建雷达。他们定期会面讨论全球技术战略。

对于 ThoughtWorks 和对我们的行业有重大影响的技术趋势。

雷达捕捉技术顾问委员会讨论的结果，以一种格式为从首席信息官到开发人员的广泛利益相关者提供价值。该内容旨在作为一个简明的总结。我们鼓励您探索这些技术的更多细节。radar 本质上是图形化的，将项目分为技术、工具、平台和语言&框架。当雷达项目可能出现在多个象限时，我们选择看起来最合适的一个。

我们进一步将这些项目分为四个环，以反映我们目前的立场。这些环是：

- 采纳：我们强烈认为行业应该采纳这些项目。我们在项目中适当的时候使用它们。
- 审判：值得追求。理解如何建立这种能力是很重要的。企业应该在能够应对风险的项目中尝试这种技术。
- 评估：值得探索，目标是了解它将如何影响您的企业。
- 保持：小心行事。

新的或自上次雷达以来有显著变化的项目表示为三角形（▲），而未移动的项目表示为圆形（◆）。每个象限的详细图表显示了项目的移动情况。我们感兴趣的条目远远超过了我们在这个大小的文档中所能容纳的数量，所以我们从最后一个雷达中淡出了许多条目，为新条目腾出了空间。淡化一个项目并不意味着我们不再关心它。

有关雷达的更多背景信息，请参见 <http://martinfowler.com/articles/radar-faq.html>

贡献者 ThoughtWorks 技术顾问委员会由以下人员组成：

丽贝卡·帕森斯(首席技术官)

马丁·福勒(首席科学家)
巴德里·贾纳
基拉曼·达伦·史密斯

埃里克·多恩伯
格埃文·博彻·徐昊
伊恩·卡特莱
特·詹姆斯·刘易斯

杰夫·诺里斯
迈克·梅森
尼尔·福特瑞秋·雷考克·罗纳尔多·费拉兹

萨姆·纽曼·斯科特·肖·斯里
哈里·斯里尼瓦桑
Thiyagu Palanisamy

雷达

技术

采取

- 1 作为文档的集合
- 2 自动化部署管道
- 3 游击测试
- 4 过程中验收测试
- 5 移动网络上的移动测试
- 6 作为一等公民的表现测试
- 异步编程的 7 个承诺
- 8 Windows 基础架构自动化

试验

- 9 分析测试运行
- 10 蓝绿色部署
- 11 远程呈现协同定位
- 12 面向移动设备的持续交付
- 13 NoSQL 的数据库迁移
- 14 用于页面排版的边缘部分
- 15 HTML5 存储，而不是作为数据的
- 16 cookies 日志
- 17 微服务移动优先
- 18
- 19 无边界企业响应网页设
- 20 计
- 21 语义监控

评定

- 22 捕获客户端 JavaScript 错误
- 23 协作分析和数据科学
- 24 云中的开发环境
- 25 关注平均恢复时间
- 26 作为构建工件的机器映像
- 27 最小化应用程序配置

保持

- 28 基于浏览器的详尽测试

平台

采取

- 29 弹性搜索
- 30 MongoDB
- 31 Neo4J
- 32 雷迪斯
- 33 短信和 USSD 作为用户界面

试验

- 34 BigQuery
- 35 云中的持续集成
- 36 沙发底座
- 37 Hadoop 2.0
- 38 节点. js
- 39 OpenStack
- 40 Rackspace 云
- 41 里亚克

评定

- 42 蔚蓝的
- 43 卡拉特拉瓦
- 44 原子的
- 45 PhoneGap/Apache Cordova
- 46 NoSQL 的 PostgreSQL
- 47 武米
- 48 Zepto. js

保持

- 49 大型企业解决方案
- 50 单体基础设施
- 51 WS-*

举行评估试验采用 2

新的或移动的
无变化



工具

采取

- 52 D3
- 53 嵌入式 servlet 容器
- 54 弗兰克
- 55 格拉德勒
- 56 石墨
- 57 不可变服务器
- 58 努格特
- 59 普萨克

试验

- 60 阿帕奇猪
- 61 格林机关枪
- 62 吉基尔博士
- 63 蝗虫
- 64 Logstash & Graylog2
- 65 幻象
- 66 木偶图书管理员和厨师图书管理员
- 67 试飞和霍克亚普

评定

- 68 基于浏览器的模板
- 69 法拉第
- 70 高起鳞癣
- 71 图标字体
- 72 看版台
- 73 章鱼

保持

- 74 反应式延伸。网
- 75 黎曼
- 76 扫雪机分析
- 77 UIAutomator
- 78 重量级测试工具
- 79 专家
- 80 TFS

语言和框架

采取

- 81 Clojure
- 82 CSS 框架
- 83 Jasmine 与 Node.js 配对
- 84 斯卡拉
- 85 西纳特拉

试验

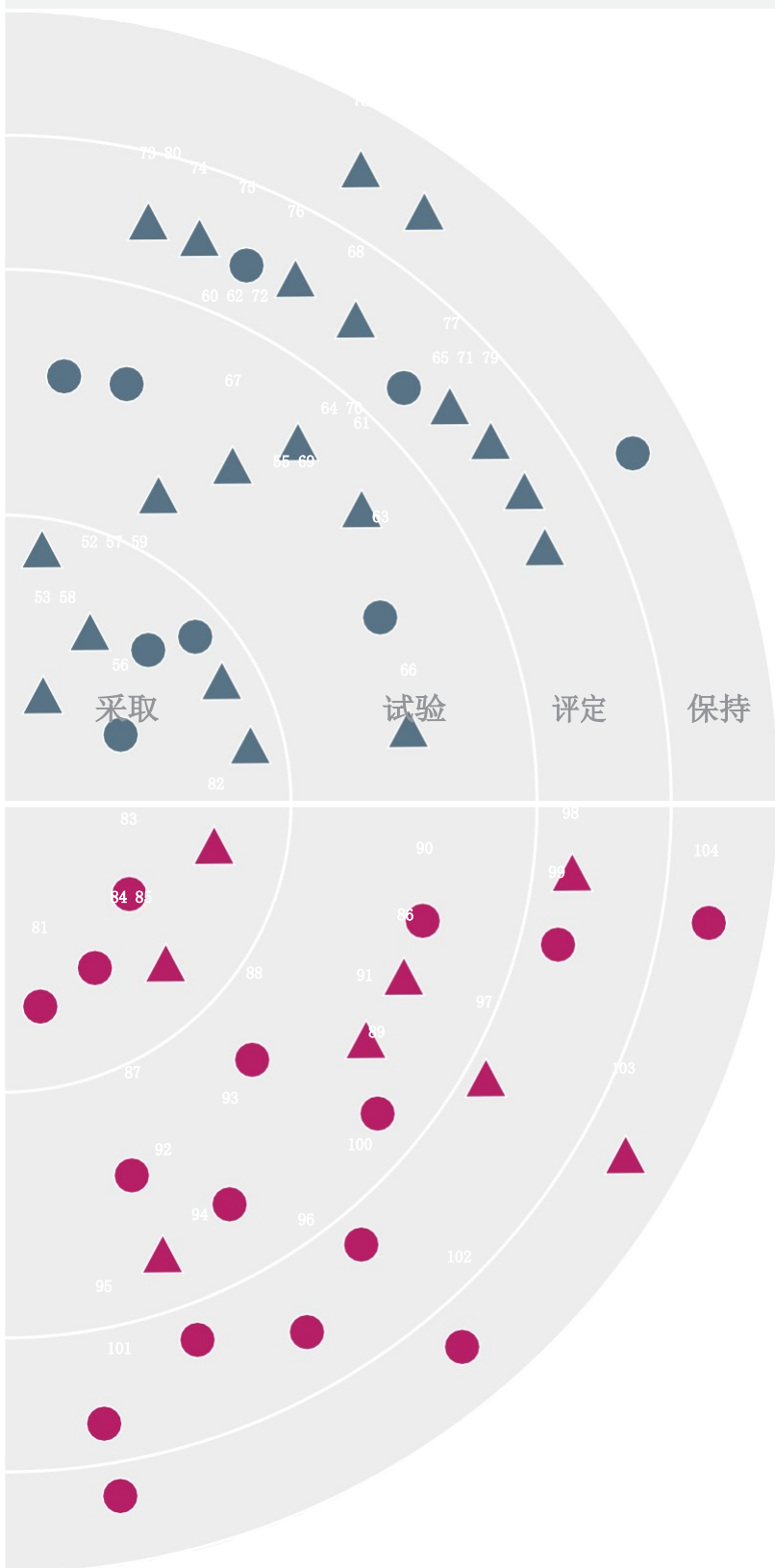
- 86 咖啡脚本
- 87 下拉向导
- 88 用于脱机应用程序的 HTML5
- 89 作为平台的 JavaScript
- 90 JavaScript MV*框架
- 91 游戏框架 2
- 92 Require.js & NPM
- 93 Scratch, Alice 和 Kodu

评定

- 94 ClojureScript
- 95 小妖精
- 96 左上臂
- 97 南茜
- 98 OWIN
- 99 RubyMotion
- 100 Twitter 引导

保持

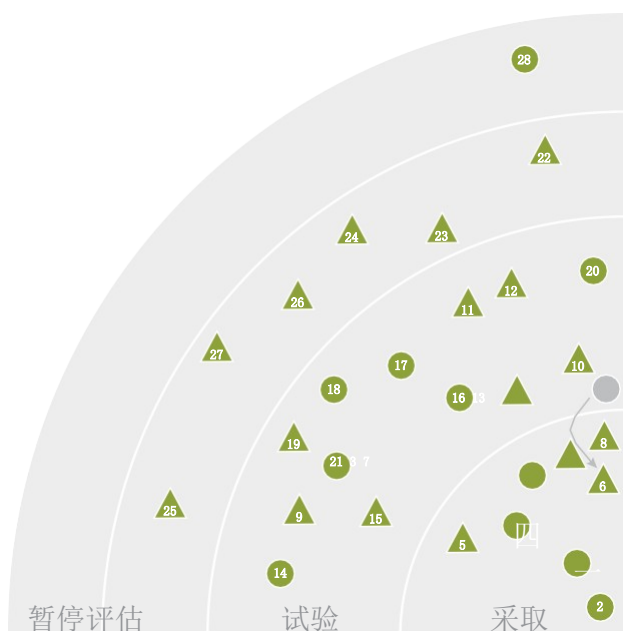
- 101 骨干网。js
- 102 基于组件的框架
- 103 手写 CSS
- 104 存储过程中的逻辑



多年来，团队和组织已经看到了围绕技术学科孤立专业知识的危险。虽然我们重视高级应用程序专家的意见，但开发人员应该具备用户界面、数据库和数据科学(最新的行业宠儿)的基础知识。虽然高级应用程序需要深厚的专业知识，但我们正在推动协作分析和数据科学，所有开发人员都可以使用基本的统计分析和工具来做出更好的决策，并在事情变得复杂时与专家密切合作。

技术趋势打破了过去围绕企业 IT 网络的花园围墙，导致了一个无边界的企业。员工经常使用自己的消费设备通过以下方式访问公司数据

云服务和 web APIs，通常在组织不知情的情况下。随着设备的持续激增和更多应用程序迁移到云，企业被迫重新思考关于数据访问和网络安全的基本假设。



云中的开发环境允许您完全外包开发基础设施，让您的团队只剩下笔记本电脑和互联网连接。通过使用私有 GitHub 存储库和 Snap CI 在云中的持续集成等同类最佳服务的组合，您的团队可能再也不需要为基础架构而麻烦内部 IT 了。

越来越多的廉价或免费视频会议的质量和选择范围正在为分布式团队带来一种新的工作方式。始终在线的视频连接有助于通过远程呈现营造一种共处感，即使团队按地理位置分布。这已经成为我们一些离岸交付中心的事实标准。我们也看到越来越多的屏幕共享工具，如 ScreenHero，用于远程配对。我们要提醒那些寻找银弹来消除对物理协同定位的需要的人。面对面交流所产生的理解和共鸣是无可替代的。

当开始使用新工具、管理不同环境的部署，或者试图了解为什么应用程序在不同的地方有不同的行为时，应用程序配置可能是一个痛苦的来源。我们热衷于最小化应用程序配置，试图确保应用程序以最小的配置合理地开箱即用。

大多数虚拟化技术都提供了一种从映像启动机器的方法。通过在构建管道的早期创建一个机器映像作为构建工件，并随着它通过进一步的测试在管道中提升它，您可以可靠地将通过测试的机器部署到生产中。这种技术消除了雪花服务器反模式的大部分原因。


蓝绿色部署是一种执行软件升级的模式。通过在生产应用程序堆栈的相同克隆上设置最新版本的应用程序，只要测试套件和业务部门认为合适，就可以近乎即时地将流量从当前的生产堆栈切换到新的生产堆栈。虽然这是一个古老的技术、基础设施自动化和云中的资源使它值得重新考虑。

采取
作为文档聚合
自动化部署管道
游击测试
过程验收测试
移动网络上的移动测试
作为一等公民的表现测试
异步编程的前景
Windows 基础架构自动化

试验
分析测试运行
蓝绿色部署
远程呈现协同定位
面向移动设备的持续交付
NoSQL 的数据库迁移
用于页面排版的边缘部分
HTML5 存储而不是 cookies
日志作为数据
微服务

移动优先
无边界企业
响应式网页设计
渗透监控
捕获客户端 JavaScript 错误
协作分析和数据科学
云中的开发环境
关注平均恢复时间

作为构建工件的机器映像
最小化应用程序配置
保持
基于浏览器的详尽测试



以前, Chef 和 Puppet 等工具缺乏对 Windows 的支持, 导致需要大量 Powershell 脚本来实现简单的基础架构自动化任务。在 Windows 上实现同样的自动化水平比在 Unix 上更具挑战性。然而, 在过去的 12 个月中, 对 Windows 的主厨和傀儡支持都有了显著的改进。这种支持与 Powershell 的内在能力相结合, 使 Windows 基础结构自动化变得非常可行。

HTML5 存储, 也称为本地存储或 web 存储, 是一种在现代浏览器中存储客户端数据的机制, 包括 iOS 和 Android 移动浏览器。我们建议在几乎所有情况下都使用 HTML5 存储来代替 cookies。HTML5 存储可以容纳高达 5MB 的数据, 而 cookies 限制为 4KB。Cookie 数据是在每个请求中传输的, 这会降低应用程序的速度, 并可能通过不安全的 HTTP 连接公开数据。相比之下, HTML5 存储数据安全地保留在浏览器中。Cookies 应该保留用于存储会话 ID 这样简单的小数据。

使用承诺进行异步编程是一种古老的技术, 也称为未来。鉴于 JavaScript 在客户端和服务端端的广泛使用, 它重新引起了人们的兴趣。这种技术消除了深度嵌套的回调、标志和轮询器的使用, 并拥有来自 jQuery 等库的一流支持。开发非常复杂的 JavaScript 代码库的团队应该利用这一点。

捕获客户端 JavaScript 错误有助于我们的交付团队识别影响用户体验的特定于浏览器或插件配置的问题。在过去的一年中, 许多服务提供商开始支持这一要求。除了将这些错误存储在应用程序数据存储中, web 应用程序还可以将这些数据记录到 web analytics 或 New Relic 等现有监控工具中, 以减轻存储需求。

随着 HTML5 模糊了传统本地应用和网络应用之间的界限, 我们开始尝试为移动设备持续交付。TestFlight 等服务允许您每天多次将原生应用部署到真实设备上。使用完全或部分基于 HTML5 的应用程序, 无需向应用程序商店提交新应用程序, 即可部署更改。如果您的组织有一个企业应用程序商店, 您可以轻松地向上推送版本。虽然在移动设备上实现 CD 的技术在不断改进, 但我们注意到测试实践却落后了。成为成功的你
将需要增加你对自动化测试的关注, 以确保一旦它到达设备, 一切都实际工作。

我们越来越多地看到移动应用程序在开发和测试期间工作得非常好, 但在现实世界中部署时却遇到了麻烦。在移动网络上的移动测试揭示了你的应用在各种条件下的表现。您可能会使用 3G 或 LTE 进行测试, 或者故意使用接入点过载的劣质 WiFi 网络。测量您的目标环境的网络性能, 然后使用延迟和丢包诱导工具模拟这些条件。此外, 有时有必要检查您的设备和软件如何通过 Wireshark 等工具使用网络。

NoSQL 数据存储继续成为主流, 团队应该认识到 NoSQL 数据库迁移的需求。特别是对于隐式或动态模式, 您可能希望随着时间的推移重新配置数据。有几种方法, 例如在部署新版本的应用程序时运行显式迁移, 或者在加载和处理文档时在代码中使用动态迁移。

失败的测试揭示了产品代码中的错误。然而, 分析其他属性的测试运行可以揭示有趣的信息。一个简单的例子是监控哪些测试经常失败, 并在构建管道中尽早运行它们以获得快速反馈。类似地, 跟踪其他属性, 比如测试执行时间和长时间运行测试与快速测试的比率, 可以提供可操作的度量。

在以前的雷达中, 我们建议在更长的旅程中安排自动化验收测试, 在我们所谓的语义监控中, 在生产环境中连续运行这些测试。我们仍然相信, 对于团队可以预见的场景, 这是一项重要的技术

提前。这种方法的一种变体, 尤其是在初创公司中, 是减少测试次数, 同时增加监控和自动警报。这将重点从避免可预见的问题转移到缩短所有问题的平均恢复时间。

虽然单元和验收测试作为标准开发实践被广泛接受, 但是这种趋势并没有延续到性能测试领域。目前, 通用工具驱使测试人员创建抛弃型代码和点击脚本心态。将性能测试视为一等公民能够创建覆盖更多功能的更好的测试, 导致创建和运行性能测试的更好的工具, 从而产生可维护的并且本身可以被测试的测试套件。



在以前的radars中，我们已经讨论过嵌入式servlet容器，现在这些已经被我们的项目广泛采用。SimpleWeb和Webbit之类的工具将这种简单的嵌入式方法进一步发展，提供了原始的HTTP服务器功能，而无需实现Java Servlet规范。与此同时，最流行的Java应用服务器Tomcat越来越多地用于嵌入式系统，微软也为.NET框架，进一步强调了这一趋势。

D3作为一个用于在浏览器中创建丰富可视化的库，继续获得关注。以前，它有些低级，与不太复杂、更有针对性的库相比，创建常用的可视化需要更多的工作。自从上一个radar以来，像用于制图的黄包车和用于浏览器内数据集探索的Crossfilter这样的库已经帮助D3变得比以前更容易访问。

我们看到一些JavaScript框架采用基于浏览器的模板，将更多的布局工作转移到客户端。虽然这种方法在很多情况下都很有用，但它确实会引入一些操作问题，包括缓存、性能和搜索。我们认为应该仔细评估这些工具，以确保它们适合目标部署环境。

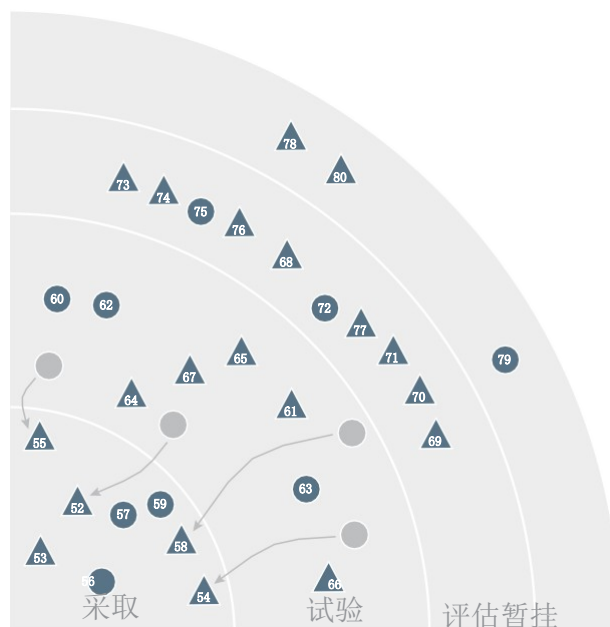
通过将IObservables和IObservers放在与IEnumerableables和IEnumerators平等的地位上，Rx for.NET允许开发人员使用他们现有的LINQ(语言集成查询)运算符知识，通过可观察事件流的公共底层抽象来查询和编写异步操作和基于事件的代码。微软还发布了RxJS，将反应式编程的优势引入JavaScript。他们开源了整个Rx框架，使其对Windows富客户端应用程序和单页JavaScript应用程序非常有用。

几个ThoughtWorks团队指出了Faraday的用处，Faraday是一个Ruby HTTP客户端库，它为各种适配器提供了一个公共接口，并与Rack中间件很好地集成在一起。

第三方库管理的包系统继续获得所有平台的认可和特性。我们称NuGet是最近加入的一个项目，Chocolatey NuGet的加入体现了围绕这一重要的敏捷工程实践出现的进步和能力。

应该采用Windows基础设施自动化，但是它仍然比Unix平台上的自动化更困难。像Chef和Puppet这样的工具正在增加它们的支持，但是也有一些特定于Windows的解决方案正在开发中，比如章鱼。Octopus允许自动部署您的ASP.NET应用程序和Windows服务，并减少对PowerShell的依赖。它可以与使用Octopack的NuGet和TeamCity一起使用，以创建完整的构建、打包和部署管道。

Puppet和Chef都必须处理共享社区贡献的模块和常用服务和任务清单的问题。Puppet Forge和Chef's Cookbook repository都有所帮助，但人们最终将这些食谱复制并粘贴到他们自己的代码库中，阻止他们利用后来的错误修复和改进。木偶图书管理员和主厨图书管理员试图通过使声明模块依赖变得容易来解决这个问题，包括从这些社区站点拉进已知版本的代码。



采取	试验	评定	保持
52 D3	60 阿帕奇猪	68 基于浏览器的模板	76 扫雪机分析
53 个嵌入式servlet容器	61 格林机关枪	69 法拉第	77 UIAutomator
54 弗兰克	62 吉基尔博士	70 高起鳞癣	
55 格拉德勒	63 蝗虫	71 图标字体	保持
56 石墨	64 Logstash & Graylog2	72 看版台	78 重量级测试工具
57 不可变服务器	65 幻象	73 章鱼	79 专家
58 努格特	66 木偶图书管理员和厨师图书管理员	74 反应式延伸。网	80 TFS
59 普萨克	67 试飞和霍克亚普	75 黎曼	



管理分布式系统中的依赖关系可能会变得复杂，这也是越来越多的人在转向更细粒度的微服务时面临的问题。Hystrix 是网飞 JVM 的一个库，它实现了处理下游故障，提供连接的实时监控，以及缓存和批处理机制，使服务间的依赖关系更有效。

TestFlight 和 HockeyApp 都允许您在没有应用商店的情况下管理移动应用程序的部署，从而使用户测试更加容易。它们提供事故报告和分析功能，以收集现场数据。HockeyApp 支持 iOS、Android、& Windows Phone，TestFlight 支持 iOS 和 Android。我们已经成功地使用这两种工具来帮助交付移动应用。这显然是一个快速发展的领域。

Frank 是一个开源库，允许用 Cucumber 编写 iOS 功能测试，并在远程设备上执行。这填补了一个重要的空白，验收测试驱动的开发在以前是麻烦和笨拙的。

UIAutomator 看起来是测试 Android 用户界面最有前途的工具，它允许在测试过程中对组件进行细粒度控制，并有助于在多种设备上进行测试。

随着具有多种形状因子和像素密度的设备的兴起，以各种比例呈现高质量图标的问题变得重要起来。图标字体通过使用浏览器对 WebFonts 和 SVG 的支持来解决这个问题，而不是缩放图像或维护不同的图标集。和往常一样，在大量使用 SVG 时，要注意移动设备上的功耗和旧设备上的性能。

由于我们构建的系统比以往任何时候都涉及更多分布在更多机器上的细粒度服务，因此如何聚合信息以轻松识别和解决问题的挑战比以往任何时候都更加紧迫。Logstash 是一种简单的方法，可以在源代码中解析和过滤日志，然后将它们转发到一个聚合点。尽管 Logstash 提供了一些搜索和过滤功能，但 Graylog2 通常与提供更全面的查询和报告功能结合使用。

我们看到了 Snowplow Analytics 的巨大前景，这是一个开源网络分析平台，基于开放数据原则和云存储，从常规网络分析中获取智能信息。

我们看到人们对围绕 PhantomJS 的 ThoughtWorks 项目感兴趣，PhantomJS 是一种无头的 web 测试工具，允许对现实目标进行功能测试。

Gatling 是自动化性能测试领域的另一个新玩家。它类似于 Locust，比 JMeter 和 Grinder 等较老的选项重量轻得多。DSL 构建在 Scala 之上，提供了丰富的现成功能，包括易于配置的数据馈送和响应断言。在需要定制的情况下，很容易使用 Scala 来提供扩展。通过 Highcharts 默认生成的大量动态数据视图增加了它的吸引力。

许多已经转向更敏捷的工作方式的组织继续使用重量级的测试工具。这些工具的问题使得它们不适合快速移动的软件交付。大型复杂的工具有很高的学习曲线，需要专业技能和培训团队自己很难测试。由于其他团队的参与，这通常会导致每个版本不必要的开销。昂贵且有限的软件许可证使得这个问题更加严重。一些重量级的工具使用“模型驱动”的方法，试图精确地建模应用程序的使用模式，这导致昂贵的测试脚本维护和开发时间被“误报”所浪费我们很少看到简单的开源解决方案不能以更少的时间、精力和金钱提供所需的信心水平的情况。

与基于 XML 和插件的工具(如 Ant 和 Maven)相比，基于语言的构建工具(如 Gradle 和 Rake)继续提供更细粒度的抽象和更大的灵活性。这允许他们随着项目变得更加复杂而优雅地成长。

我们继续看到团队在试图使用 TFS 作为版本控制系统时遇到生产力问题。想要实践频繁的代码检入(持续集成的核心部分)的团队发现它的重量级方法极大地消耗了生产力。这通常会导致团队更少的签入，导致更多有问题的合并。我们推荐使用 Git、Perforce 和 Subversion 等工具。



PostgreSQL 正在扩展成为 SQL 数据库的 NoSQL 选择。版本 9.2 包括存储 JSON 数据的能力，以及对 JSON 内容的完全查询能力文档。其他扩展允许用户以键/值对的形式存储和查询数据。这让你可以利用经过时间考验的数据库的底层存储和事务处理能力，而不依赖于关系数据模型。对于那些既想要 SQL 应用程序又想要 NoSQL 应用程序，但更喜欢他们已经知道如何支持的单一可靠基础架构的人来说，这是非常理想的。

即使是一个相对较小的网站也能产生大量的数据。一旦你添加了分析、业务指标、人口统计、用户资料和多种设备，它会变得势不可挡。许多组织使用数据仓库作为存储库，从组织的各个部分吸收数据。这里的挑战是这些经常变成“数据堡垒”甚至获得及时的业务指标都成了一个挑战，更不用说在整个数据集上运行探索性查询了。像基于云的 BigQuery 这样的技术有所帮助。现收现付模式和即席查询功能让您无需购买专业硬件和软件即可获得洞察力。数据驱动型企业应该数据掌握在决策者手中，而不是隐藏在技术壁垒和官僚主义背后。

对于适合文档数据库模型的问题，MongoDB 现在是最受欢迎的选择。除了易用性和可靠的技术实现，社区和生态系统也为这一成功做出了贡献。我们意识到了一些问题，当团队被 MongoDB 的流行所诱惑时文档数据库不是很合适，或者他们不了解内在的复杂性。然而，如果使用得当，MongoDB 已经在许多项目中证明了自己。

Redis 在多个 ThoughtWorks 项目中被证明是一个有用的工具，既可用于分布在不同国家的结构化缓存，也可用作数据存储。

Hadoop 最初的架构基于水平扩展数据和垂直扩展元数据的范例。虽然数据存储和处理由从节点处理得相当好，但是管理元数据的主节点是单点故障，并且限制了 web 规模的使用。Hadoop 2.0 对 HDFS 和 Map Reduce 框架进行了重大的重新架构，以解决这些问题。HDFS 命名空间现在可以使用同一集群上的多个名称节点进行联合，并以高可用性模式进行部署。MapReduce 已被 YARN 所取代，它将集群资源管理与作业状态管理相分离，并消除了 JobTracker 的规模/性能问题。最重要的是，这种变化鼓励在 Hadoop 集群上部署除 MapReduce 之外的新的分布式编程范例。

在过去的一年里，我们已经看到弹性搜索作为一个开源搜索平台逐渐被采用。它是一个基于 Apache Lucene 的可扩展、多租户、水平可伸缩的搜索解决方案。它允许通过基于 JSON 的 REST API 索引和检索复杂的数据结构。它提供了一个优雅的操作模型，可以自动发现集群中的对等点、故障转移和复制。弹性搜索可以通过一个插件系统来扩展，该系统允许添加新的功能和改变现有的行为。围绕这个工具的社区非常活跃如 Java、C#、Ruby 和 JavaScript 等语言中可用的客户端库的数量所示。



采取
弹性搜索

MongoDB
Neo4J

雷迪斯

短信和 USSD 作为用户界面

试验

BigQuery

云沙发库中的持续集成

Hadoop 2.0 Node.js OpenStack

35

36

37

38

39

Rackspace 云

里亚克

评定

蔚蓝的

卡拉特拉瓦

原子的

PhoneGap/Apache Cordova

NoSQL 的 PostgreSQL

武米

Zepto.js

保持

大型企业解决方案

单体基础设施

WS*



Node.js 是一个轻量级的 web 容器，是开发微服务和作为移动和单页面 web 应用程序的服务器器的一个强有力的选择。由于 node.js 的异步特性，开发人员转向 promise 库来简化他们的应用程序代码。随着 promises 在 node.js 社区中的使用逐渐成熟，我们希望看到更多针对 node.js 开发的应用程序。对于那些不愿意在生产中尝试 node.js 的团队来说，考虑 node.js 用于开发任务仍然是值得的，比如在浏览器外部运行 JavaScript 测试，或者从 CoffeeScript、SASS 等工具生成静态 web 内容。

Zepto.js 是一个轻量级的 JavaScript 库，主要基于 JQuery。该 API 与 JQuery 完全相同，尽管它不提供与 JQuery 的完全兼容性。Zepto 具有极大的压缩文件大小，在构建响应性 web 应用程序时是一个引人注目的选择。

PhoneGap，现在更名为 Apache Cordova，是一个让你使用 HTML、CSS 和 JavaScript 开发跨平台移动应用的平台。它通过一组 JavaScript APIs 抽象出特定于平台的本机代码，这些 API 在不同的移动平台上保持一致。Cordova 可用于多种平台，包括 iOS、Android、Blackberry、Windows Phone 和 WebOS。

虽然 AWS 继续增加更多功能，但 Rackspace Cloud 已经成为存储和计算领域的一个有力竞争对手。一些用户可能看重 Rackspace 提供的更全面的客户支持，以及混合更传统的托管模式的能力。我们对此并不感到兴奋，因为 Rackspace 是我们的客户，我们很高兴开发了这个平台。我们已经成功地将 Rackspace Cloud 与其他几个客户一起使用，并期待它在更多的地理位置提供。

开源的 OpenStack 项目正在加速发展，并且在最近几个月成为一个更可行的平台来部署你自己的私有云。许多使 OpenStack 难以启动和运行的问题已经得到解决，并且新功能一直在增加。很明显，OpenStack 联盟及其成员 Rackspace、Redhat 和 HP 都致力于该项目，将其作为自己基于 OpenStack 的云服务的基础。

去年全球售出的所有手机中，58%是功能手机。在许多发展中国家，这一比例甚至更高。如果你的市场要求你在这些领域发展，你需要在发展的时候考虑到这个限制。这些手机使用短信和 USSD 作为用户界面。SMS 是一种由来已久的发送消息的技术，USSD 允许您在安全的会话中发送类似 SMS 的消息。你应该把 USSD 和短信视为另一个用户界面和 UX 平台，并把它们视为一等公民。

Vumi 是一个可扩展的开源消息引擎，通过移动设备上的节俭方法推动对话。Vumi 促进了公司与客户、医疗服务机构与患者之间的短信、即时消息和 USSD 互动，政府和公民，等等。Vumi 与电信公司集成，允许您在其上轻松构建应用程序。你只需要支付运营商费用。

“企业级”商业软件包提供的东西和实际需要的东西之间的差距正在扩大。对于面向互联网的应用程序来说尤其如此。

实践者为实践者编写创新的解决方案，这些解决方案能够真正扩展并轻松支持现代技术，例如连续交付。它们起源于许多互联网规模的公司，并被提炼为开源软件。大型企业解决方案通常会阻碍有效的交付，因为它们积累了臃肿、繁琐的许可限制，以及由检查表和想象中的需求驱动的特性集，远离大多数开发团队现实。



随着支持数据绑定、客户端模板、验证和其他功能的框架的出现，单页面 web 应用程序开发继续蓬勃发展。ThoughtWorks 项目中积极使用的 JavaScript MV* 框架包括 AngularJS、Knockout 和 Ember.js。每个框架都有拥护者和一些反对者。我们期待在这个充满活力的领域继续创新。

随着单页面和基于移动浏览器的应用程序进入主流，以及用于服务器端应用程序的 node.js 的持续增长，CoffeeScript 越来越多地被用来简化 JavaScript 代码库。作为一种编译成 JavaScript 代码用于运行时执行的语言，许多人担心调试 CoffeeScript 应用程序的难度。CoffeeScript 1.6.1 中源地图的引入正在帮助开发工具的生产者解决这个问题。我们预计，随着 Dropbox 等知名度很高的技术公司的领先，这将导致这种语言的进一步采用。

我们新的生产应用程序中继续使用 node.js，这再次强化了我们对于 JavaScript 代码和库的可靠打包的需求。节点包管理器 (npm) 是 node.js 生态系统的重要组成部分，也是一个有用的工具

用于打包 node.js 应用程序。拥有大量 JavaScript 或 CoffeeScript 的浏览器应用程序开发人员应该考虑 Require.js 来帮助构建他们的代码并在运行时加载依赖项。

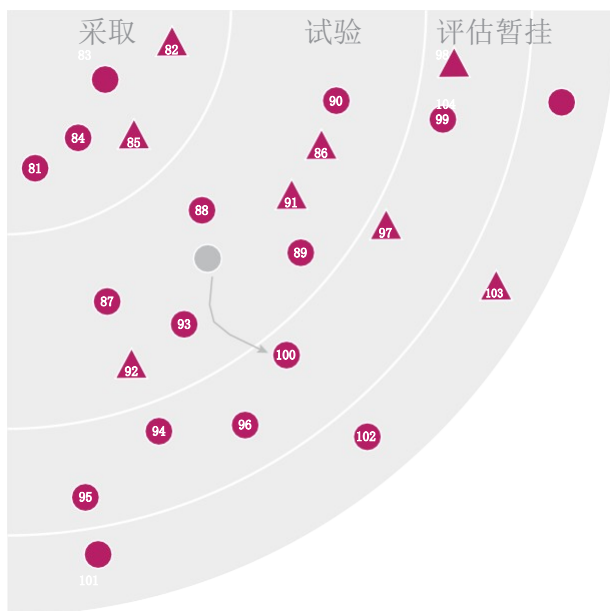
微框架正在成为处理客户端和服务端应用程序日益增加的复杂性的一种方式。

Sinatra 是服务器端领域中这种趋势的第一个例子，它揭示了一种轻量级 DSL 来构建易于组合的快速服务。其他语言也有类似的产品，包括 Spark for Java、Flask for Python、Sclatra for Scala、Compojure for Clojure 和 Nancy for .NET。

上一个丰富的开源 web 开发生态系统的发展速度减慢了。NET 平台已经过度依赖 IIS 和 ASP.NET 框架。OWIN 指定了一个开放的 HTTP 处理接口，将 web 服务器与应用程序分离，就像 Rack 为 Ruby 所做的那样

社区。我们对 OWIN 感到兴奋，因为它开启了新的可能性。NET web 开发工具由简单的、独立开发的模块组成。南希是这方面的完美例子。我们还希望它能增加在 .NET 平台。

最近发布的 Play Framework 2.1.1 支持控制器依赖注入、异步、非阻塞 I/O、代码重载工作流、数据库迁移、资产流水线和灵活的部署选项，这使得它对开发人员更具吸引力。由于这个原因，Play 再次成为团队在 JVM 上构建 web 应用程序和服务时需要考虑的事情。一句话然而，注意，Play 采用了函数式编程风格，当使用 Java 语言时，仍然会转化为过多的静态方法，这些方法可能难以在运行的服务器之外进行单元测试。



采取

- 81 Clojure
- 82 CSS 框架
- 83 Jasmine 与 Node.js 配对
- 84 斯卡拉
- 85 西纳特拉

试验

- 86 咖啡脚本
- 87 下拉向导
- 88 用于脱机应用程序的 HTML5
- 89 作为平台的 JavaScript
- 90 JavaScript MV* 框架
- 91 游戏框架 2
- 92 Require.js & NPM
- 93 Scratch, Alice 和 Kodu

评定

- 94 ClojureScript
- 95 小妖精
- 96 左上臂
- 97 南茜
- 98 OWIN
- 99 RubyMotion
- 100 Twitter 引导

保持

- 101 骨干网.js
- 102 基于组件的框架
- 103 手写 CSS
- 104 存储过程中的逻辑



与 JavaScript 和 HTML 一样，CSS 是创建网站的核心技术。不幸的是，语言本身缺乏关键特性，这导致了高度的重复和缺乏有意义的抽象。虽然 CSS3 旨在纠正其中一些问题，但组成 CSS3 的模块在大多数浏览器中得到正确支持还需要几年时间。幸运的是，现在有一个使用 CSS 框架的解决方案，比如萨斯、SCSS 等等。由于他们的质量和支持，我们相信手写 CSS 的时代已经结束了，除了琐碎的工作。

CSS 框架简化了大规模 CSS 代码库的开发，而不必每次都从头开始。

因为有大量的框架，所以选择一个能够持续增强和维护的框架是很重要的

代码库的一部分，而不是帮助你快速入门的东西。基于插件的框架，例如

指南针，或者有特定焦点的，比如 Susy，在这方面要好得多。

Twitter Bootstrap 是一个流行的 CSS 框架，它允许你快速创建一个具有流畅和响应性布局的好看网站。在这一期的 radar 中，Bootstrap 从试验转向基于我们长期使用经验的评估。如果您希望替换或广泛定制应用程序的外观和感觉，Bootstrap 可能会面临一个挑战，因为它与 HTML 标记深度集成。这并不一定使它成为一个坏的选择，但在选择它而不是其他选择时，记住这些限制是值得的。

参考

阿帕奇科尔多瓦:<http://cordova.apache.org/>

BigQuery:<http://martinfowler.com/articles/bigQueryPOC.html>

客户端错误日志记录:<http://openmymind.net/2012/4/4/You-Really-Should-Log-Client-Side-Error/>

CoffeeScript 并不是一门值得学习的语言:<https://github.com/raganwald/homo-iconic/blob/master/2011/12/行话.md>

CoffeeScript 源地图:<http://coffeescript.org/#source-maps>

Dropbox 潜入 coffee script:<https://tech.dropbox.com/2012/09/Dropbox-dives-into-coffee-script/>

下拉向导:<http://dropwizard.codahale.com/>

法拉第:<https://github.com/lostisland/faraday>

格雷洛:<http://graylog2.org/>

主厨:<http://www.opscode.com/hosted-chef/>

JavaScript 错误报告:<http://devblog.pipelinedeals.com/pipelinedeals-dev-blog/2012/2/12/javascript-error-reporting-for-fun-and-profit-1.html>

日志存储:<http://logstash.net/>

手机使用:<http://www.idc.com/getdoc.jsp?containerId=prUS23982813#>

MongoDB:<http://www.mongodb.org/>

希框架:<http://nancyfx.org/>

<https://npmjs.org/>

<https://npmjs.org/doc/json.html>

章鱼:NPM:<http://octopusdeploy.com/>

OWIN:<http://owin.org/>

幻象:<http://phantomjs.org/>

PhoneGap:<http://phonegap.com/>

计划:<https://github.com/plans>

反应延伸:<https://rx.codeplex.com/>

要求:<http://requirejs.org/>

snap ci:<https://snap-ci.com/>

雪花服务器:<http://martinfowler.com/bliki/SnowflakeServer.html>

来源地图修订版3提案:https://docs.google.com/document/d/1ulrgaehqwryputovf1krlpiofze0b_2gC6fah0ky0k/edit

UIAutomator:<http://developer.android.com/tools/help/uiautomator/index.html>

USSD:http://en.wikipedia.org/wiki/Unstructured_Supplementary_Service_Data

Vumi:<http://vumi.org/>

为什么大家要么讨厌要么离开 Maven:http://nealford.com/memeagora/2013/01/22/why_everyone_eventually_hates_maven.html

ThoughtWorks 是一家全球 IT 咨询公司

ThoughtWorks——一家软件公司和一个由充满激情的个人组成的社区，其目的是革新软件创作和交付，同时倡导积极的社会变革。我们的产品部门 ThoughtWorks Studios 为渴望卓越的软件团队开发先锋工具；例如 Mingle、Go 和 Twist，它们帮助组织更好地协作并交付高质量的软件。我们的客户是有雄心壮志的人和组织；我们提供颠覆性思维和技术，帮助他们取得成功。在我们的第 20 个年头，大约有 2500 名 ThoughtWorks 员工——思想工作者——为我们在澳大利亚、巴西、加拿大、中国、德国、印度、新加坡、南非、乌干达、英国和美国的客户提供服务