

# 技术雷达

2013年5月

由ThoughtWorks技术咨询委员会编写

[thoughtworks.com/radar](http://thoughtworks.com/radar)



# 有什么新鲜事？

以下是本版强调的趋势。

- **拥抱下降的边界**--无论你喜欢与否，边界正在你周围下降。我们选择通过研究无边界企业、云中的开发环境和网真共处等概念来拥抱这一点。
- **将成熟的实践应用于那些不知何故错过的领域**--我们并不确定原因，但我们行业中的许多人都错过了一些想法，如捕捉客户端JavaScript错误、移动端的持续交付、NoSQL的数据库迁移和CSS的框架。
- **分析学的轻量级选择**--数据科学和分析学不仅仅是为拥有该领域博士学位的人准备的。我们强调协作式分析和数据科学，所有开发人员都了解基础知识并与专家密切合作在必要的时候。
- **基础设施即代码**--连续交付和DevOps提升了我们对基础设施的思考。将基础设施视为代码的意义和对新工具的需求仍在不断发展。

ThoughtWorks对技术充满热情。我们建立它，研究它，测试它，开放它，写下它，并不断致力于改善它--为所有人。我们的使命是倡导软件的卓越性和革新IT。我们创造并分享ThoughtWorks技术雷达，以支持这一使命。ThoughtWorks技术顾问委员会，由ThoughtWorks的高级技术领导人组成，负责创建雷达。他们定期开会，讨论全球技术战略为ThoughtWorks和对我们行业有重大影响的技术趋势。

该雷达记录了技术咨询委员会的讨论结果，其格式对从首席信息官到开发人员等广泛的利益相关者提供了价值。这些内容旨在作为一个简洁的总结。我们鼓励你去探索这些技术的更多细节。雷达是图形化的，将项目分为技术、工具、平台、语言和框架。当雷达项目可以出现在多个象限时，我们选择了看起来最合适的一个。我们进一步将这些项目分成四个环，以反映我们目前对它们的立场。这些环是：

- **采纳**。我们强烈地感觉到，行业应该采用这些项目。我们在我们的项目中适当地使用它们。
- **审判**。值得追求。了解如何建立这种能力是很重要的。企业应该在一个能够处理风险的项目上尝试这种技术。
- **评估**。值得探索，目的是了解它将如何影响你的企业。
- **保持**：谨慎行事。

自上次雷达以来，新的或有重大变化的项目用三角形表示，（-）而没有移动的项目用圆圈表示（-）。每个象限的详细图表显示了项目的移动情况。我们感兴趣的项目远远多于我们可以合理地装入这么大的文件，所以我们从上一个雷达中淡化了许多项目，以便为新项目腾出空间。淡化一个项目并不意味着我们不再关心它。

关于雷达的更多背景，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 (CTO)  
马丁-福勒 (首席科  
学家) 巴德利-贾纳  
基拉曼 达伦-史密斯

Erik Doernenburg  
Evan Bottcher  
Hao Xu  
Ian Cartwright  
James Lewis

Jeff Norris  
Mike Mason  
Neal Ford  
Rachel Laycock  
Ronaldo Ferraz

萨姆-纽曼 斯科特-  
肖 斯里哈瑞-斯里  
尼瓦桑  
蒂亚古-帕拉尼萨米

# 雷达

## 技术

采纳

- 1 作为文件的 集合体
- 2 自动部署管道
- 3 游击队测试
- 4 过程中的验收测试
- 5 移动网络上的移动测试
- 6 作为一等公民的性能测试
- 7 异步编程的承诺
- 8 Windows基础设施自动化

试用

- 9 分析测试运行
- 10 蓝绿色的部署
- 11 通过远程呈现实现同地办公
- 12 针对移动设备的持续交付
- 13 NoSQL的数据库迁移
- 14 边缘侧 包括用于页面组成的
- 15 HTML5存储而不是cookies
- 16 日志作为数据
- 17 微服务 移动优
- 18 先
- 19 无边界的企业
- 20 响应式网页设计
- 21 语义监测

评估

- 22 捕获客户端的JavaScript错误
- 23 协作分析和数据科学
- 24 云中的开发环境
- 25 专注于平均康复时间
- 26 机器图像作为一个构建工件
- 27 尽量减少应用程序的配置
- 控股
- 28 详尽的基于浏览器的测试

## 平台

采纳

- 29 弹性搜索
- 30 梦之城\_梦之城娱乐\_梦之城国际娱乐  
\_梦之城国际娱乐平台
- 31 Neo4J
- 32 睿迪思(Redis)
- 33 短信和USSD作为一种用户界面

试用

- 34 BigQuery
- 35 云中的持续集成
- 36 辅导员 (Couchbase
- 37 Hadoop 2.0
- 38 Node.js
- 39 淘宝网
- 40 洛克菲勒云
- 41 Riak

评估

- 42 蔚蓝

持有

评估

审判

采纳

- 43 卡拉特拉瓦
- 44 卫星导航系统(Datomic)
- 45 PhoneGap/Apache Cordova
- 46 用于NoSQL的PostgreSQL
- 47 Vumi



33

35

40

46

48

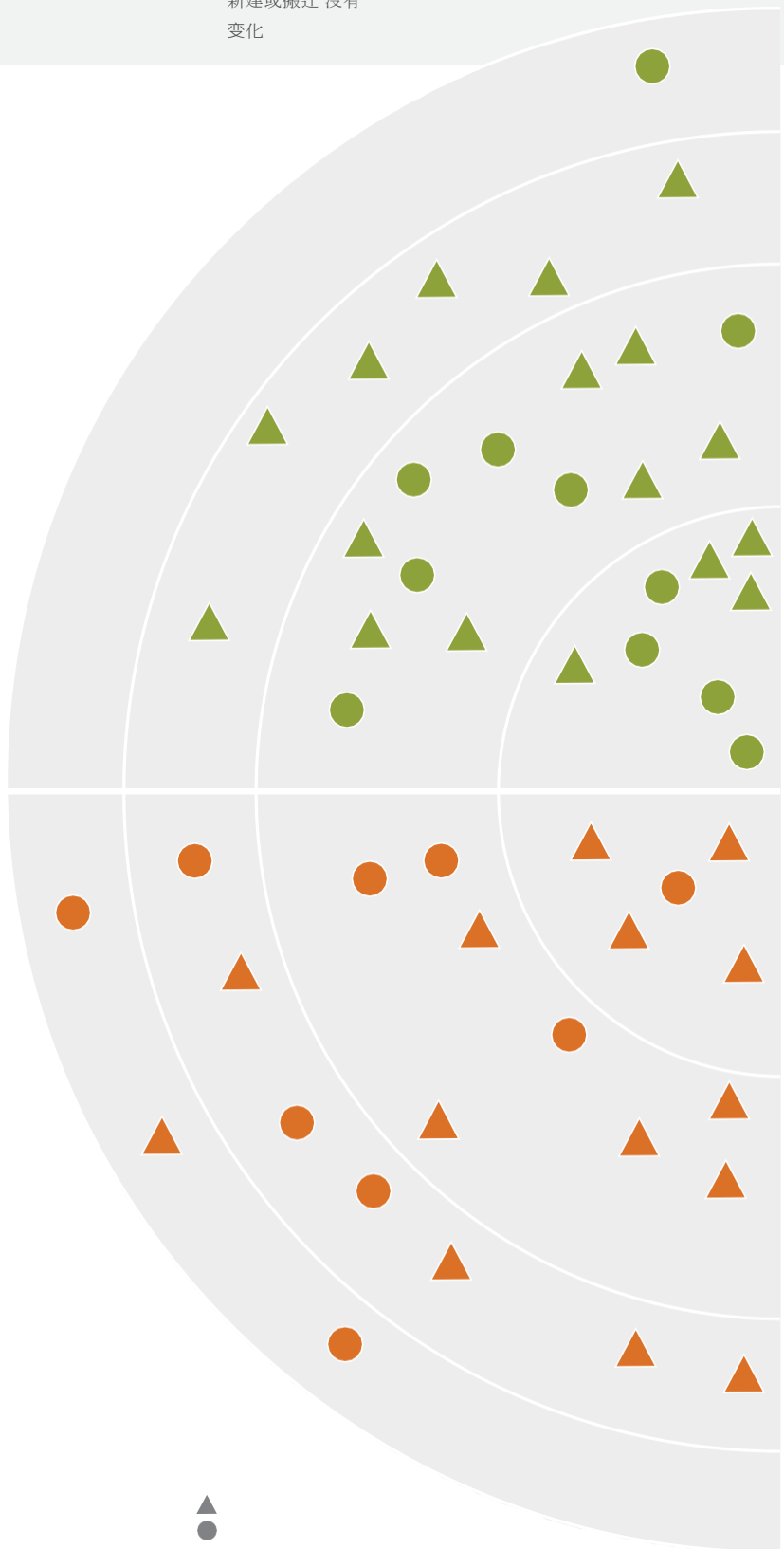
48 Zepto.js  
控股

49 大型企业解决方案

50 单一的基础设施

51 WS-\*

新建或搬迁 没有  
变化





## 工具

### 采纳

- 52 D3
- 53 嵌入式servlet容器
- 54 弗兰克
- 55 渐进式
- 56 石墨
- 57 不变的服务器
- 58 淘宝网
- 59 AAA

### 试用

- 60 Apache Pig
- 61 加特林
- 62 杰基尔
- 63 蝗虫
- 64 Logstash & Graylog2
- 65 PhantomJS
- 66 木偶图书管理员和厨师图书管理员
- 67 TestFlight & HockeyApp

### 评估

- 68 基于浏览器的模板设计
- 69 法拉第
- 70 Hystrix
- 71 图标字体
- 72 灯台
- 73 章鱼

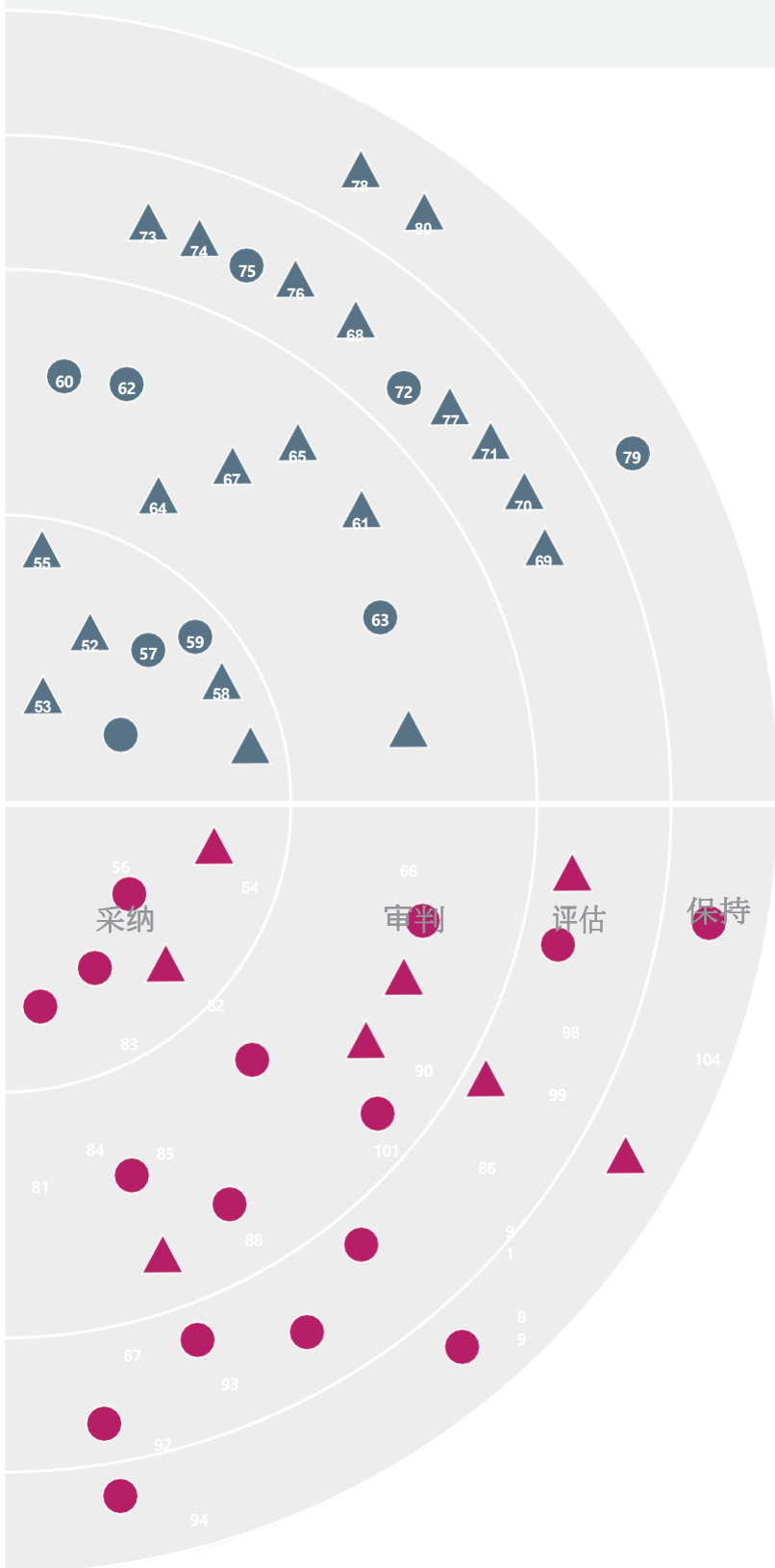
- 74 .Net的反应性扩展
- 75 黎曼
- 76 扫雪机分析

- 77 仪表盘上的 "UIAutomator"。

### 控股

- 78 重量级的测试工具
- 79 雯雯
- 80 TFS

## 语言和框架



采纳

- 81 Clojure
- 82 CSS框架
- 83 茉莉花与Node.js的搭配
- 84 苴麟岔语
- 85 西纳特拉

试用

- 86 咖啡色脚本
- 87 滴水观音
- 88 用于离线应用的HTML5
- 89 作为一个平台的JavaScript
- 90 JavaScript MV\*框架
- 91 游戏框架2
- 92 Require.js & NPM
- 93 划痕，爱丽丝，和科杜

评估

- 94 ClojureScript
- 95 格雷姆林
- 96 吕阿
- 97 南希
- 98 OWIN
- 99 润彩客网
- 100 推特Bootstrap

控股

- 101 Backbone.js
- 102 基于组件的框架
- 103 手写的CSS
- 104 存储程序中的逻辑



# 技术

多年来，团队和组织已经看到了围绕技术学科孤立的专业知识的危险性。虽然我们重视专家对高级应用的投入，但开发人员应该具备用户界面、数据库和数据科学的基本知识，这是最新的行业宠儿。虽然高级应用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**，所有的开发人员都使用基本的统计分析和工具来做出更好的决定，并在事情变得复杂时与专家密切合作。

技术趋势已经打破了过去围绕企业IT网络的园墙，并导致了一个**无边界的企业**。员工经常使用他们自己的消费类设备，通过以下方式访问企业数据

云服务和网络API，往往在企业不知情的情况下。随着设备的不断扩散和更多的应用程序转移到云端，企业正被迫重新思考关于数据访问和网络安全的基本假设。

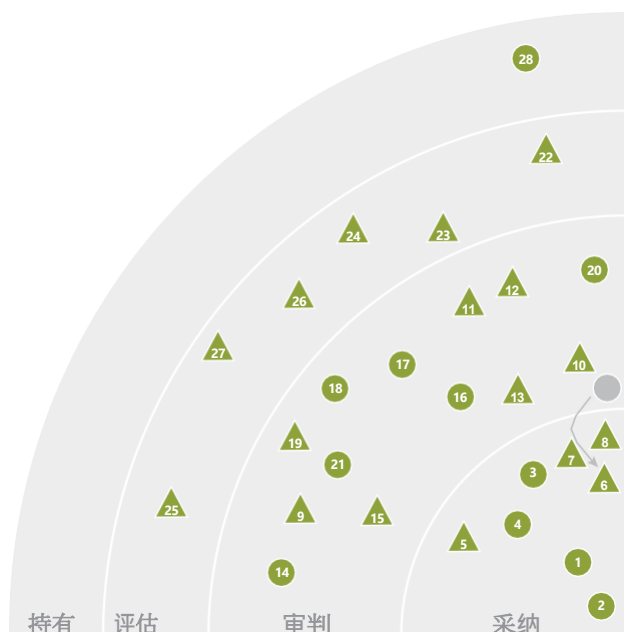
**云中的开发环境**允许你完全外包开发基础设施，让你的团队只需要笔记本电脑和互联网连接。通过使用最佳服务的组合，如私人GitHub存储库和Snap CI的云端持续集成，你的团队可能永远不需要再为基础设施而困扰内部IT部门。

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可以**通过远程呈现**帮助创造一种**共处的感觉**，即使是在团队在地理上的分布。这正在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地使用屏幕共享工具，如ScreenHero，用于远程配对。我们要提醒那些寻找银弹以消除对物理同地的需求的人。面对面的沟通所产生的理解和共鸣是无可替代的。

当开始使用一个新的工具，管理部署到不同的环境，或者试图理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。我们是最小化**应用配置**的忠实拥护者，试图确保应用在开箱后以最低的配置合理地工作。

大多数虚拟化技术提供了一种从图像中启动机器的方法。通过在构建管道的早期创建**机器镜像作为构建工件**，并在它通过进一步的测试套件时通过管道进行推广，你可以可靠地将通过测试的确切机器部署到生产中。这项技术消除了雪花服务器反模式的大部分原因。

**蓝绿部署**是一种执行软件升级的模式。通过在生产应用堆栈的相同克隆上设置你的应用的最新版本，一旦测试套件和业务确定是合适的，流量就可以近乎即时地从当前的生产堆栈切换到新的堆栈。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。



## 采纳

- 1 作为文件的集合体
- 2 自动部署管道
- 3 游击队测试
- 4 过程中的验收测试
- 5 移动网络上的移动测试
- 6 作为一等公民的性能测试
- 7 异步编程的承诺
- Windows基础设施自动化

## 试用

- 9 分析测试运行
- 10 蓝绿色的部署
- 11 通过远程呈现实现异地办公
- 12 针对移动设备的持续交付
- 13 NoSQL的数据库迁移
- 14 边缘侧 包括用于页面组成的
- 15 HTML5存储而不是cookies
- 16 作为数据的日志
- 17 微服务

## 18 移动优先

- 19 无边界的企业
- 20 响应式网页设计
- 21 语义监测
- 评估
- 22 捕获客户端的JavaScript错误
- 23 协作分析和数据科学
- 24 云中的开发环境
- 25 专注于平均恢复时间

## 26 机器图像作为一个构建工件

- 27 尽量减少应用程序的配置
- 控股
- 28 详尽的基于浏览器的测试

以前，像Chef和Puppet这样的工具缺乏对Windows的支持，导致大量的Powershell脚本来实现简单的基础设施自动化任务。与Unix相比，为Windows实现同样水平的自动化更具挑战性。然而，在过去的12个月里，Chef和Puppet对Windows的支持都有了很大的改善。这种支持，再加上Powershell的内在力量，使得**Windows基础设施自动化变得极为可行**。

HTML5存储，也被称为本地存储或网络存储，是现代浏览器（包括iOS和Android移动浏览器）中存储客户端数据的一种机制。我们建议在几乎所有情况下使用**HTML5存储而不是cookies**。

HTML5存储可以容纳高达5MB的数据，而cookie则限制在4KB。Cookie数据在每个请求中都会被传输，这将降低您的应用程序的速度，并有可能通过不安全的HTTP连接暴露数据。相比之下，HTML5存储数据则安全地保留在浏览器中。Cookie应保留用于存储小的简单数据，如会话ID。

使用**承诺进行异步编程**是一种古老的技术，也被称为期货。鉴于JavaScript在客户端和服务端端的广泛使用，它重新获得了兴趣。这种技术消除了对深度嵌套的回调、标志和轮询器的使用，并得到了jQuery等库的一流支持。开发相当复杂的JavaScript代码库的团队应该利用这个优势。

**捕获客户端的JavaScript错误**有助于我们的交付团队识别影响用户体验的浏览器或插件配置的具体问题。在过去的一年里，一些服务供应商已经开始支持这一要求。除了在应用数据存储中存储这些错误外，网络应用可以将这些数据记录到网络分析或现有的监控工具，如New Relic，以卸载存储需求。

随着HTML5模糊了传统原生应用和网络应用之间的界限，我们开始尝试**为移动设备进行持续交付**。TestFlight等服务允许你每天多次将原生应用部署到真实设备上。

对于完全或部分基于HTML5的应用，可以在不向应用商店提交新应用的情况下部署更改。如果你的组织有一个企业应用商店，你可能会很容易地将构建推送给它。虽然对移动设备实施CD的技术在不断改进，但我们注意到测试实践是滞后的。要想获得成功，你必须

将需要增加你对自动化测试的关注，以确保所有的东西一旦到达设备上就能实际工作。

我们越来越多地看到移动应用程序在开发和测试期间运行得非常好，但当它们被部署到真实世界时却遇到了麻烦。**在移动网络上的移动测试**揭示了你的应用程序在各种条件下的表现。你可能会使用3G或LTE进行测试，或者故意使用接入点过载的不良WiFi网络。为你的目标环境测量网络性能，然后使用延迟和丢包诱导工具模拟这些条件。此外，有时还需要准确地检查你的设备和软件是如何用诸如Wireshark这样的工具来使用网络的。

NoSQL数据存储继续成为主流，团队应该承认**NoSQL的数据库迁移**的需要。特别是对于隐式或动态模式，你很可能希望随着时间的推移重新配置数据。有几种方法，比如在部署应用程序的新版本时运行显式迁移，或者在加载和处理文档时在代码中使用动态迁移。

失败的测试揭示了生产代码中的错误。然而，**分析测试运行的**其他属性可以发现有趣的信息。一个简单的例子是监测哪些测试经常失败，并在构建管道的早期运行它们以获得快速反馈。同样，跟踪其他属性，如测试执行时间和长期运行的测试与快速测试的比例，可以提供可操作的指标。

在以前的雷达中，我们建议将自动化验收测试安排在较长的旅程中，并在我们所说的语义监控中，针对生产环境连续运行这些测试。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

预先。这种方法的一个变种，特别是在初创企业中，是减少测试的数量，同时增加监测和自动报警。这就把重点从避免可以预见的问题转移到**减少所有问题的平均恢复时间**上。

虽然单元和验收测试被广泛接受为标准的开发实践，但这种趋势并没有延续到性能测试领域。目前，常见的工具促使测试人员创造被丢弃的代码和点击和脚本的心态。将**性能测试作为一流的公民**来对待，可以创建更好的测试，涵盖更多的功能，导致更好的工具来创建和运行性能测试，从而形成一个可维护的测试套件，并且本身可以测试。

# 工具

在以前的雷达中，我们谈到了**嵌入式Servlet容器**，现在这些容器在我们的项目中被广泛采用。像SimpleWeb和Webbit这样的工具进一步采用了简单的嵌入式方法，并提供了原始的HTTP服务器功能，而没有实现Java Servlet规范。同时，Tomcat，最流行的Java应用服务器，越来越多地被用于嵌入式设置，微软为.NET框架提供了自我托管的服务器，为这一趋势提供了进一步的支持。

D3作为一个在浏览器中创建丰富的可视化的库，继续获得吸引力。以前，它的水平较低，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工作。自上次雷达以来，像Rickshaw这样的图表库和Crossfilter这样的浏览器内数据集探索库已经帮助D3比以前更容易获得。

我们看到一些JavaScript框架采用了**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是有用的，但它确实引入了涉及缓存、性能和搜索的操作问题。我们认为应该仔细评估这些工具，以确保其适合于目标部署环境。

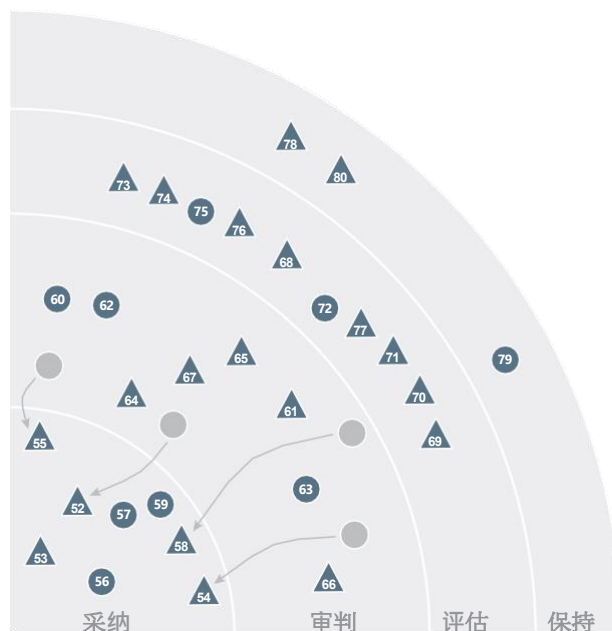
通过将IObservables和IObservers与IEnumerables和IEnumerators放在同等地位，.NET的Rx允许开发者使用他们现有的LINQ（语言集成查询）操作符的知识来查询和组成异步操作和基于事件的代码，使用可观察事件流的基础抽象。微软还发布了RxJS，将反应式编程的好处带到了JavaScript。他们开源了整个Rx框架，使其适用于Windows富客户端应用程序和单页JavaScript应用程序。

ThoughtWorks的几个团队指出了Faraday的用处，这是一个Ruby HTTP客户端库，在各种适配器上提供了一个通用接口，并与Rack中间件很好地整合。

用于第三方库管理的软件包系统继续在所有平台上获得认可和功能。我们把NuGet作为最近的一个项目，而Chocolatey NuGet的加入体现了围绕这一重要的敏捷工程实践所取得的进展和能力。

Windows基础设施的自动化应该被采用，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，但也有正在开发的Windows特定解决方案如Octopus。Octopus允许自动部署你的ASP.NET应用程序和Windows服务，减少对PowerShell的依赖。它可以与使用Octopack的NuGet和TeamCity一起使用，创建一个完整的构建、打包和部署管道。

Puppet和Chef都不得不处理分享社区贡献的模块和常用服务和任务的清单的问题。Puppet Forge和Chef的Cookbook库都提供了帮助，但人们最终还是将这些配方复制并粘贴到自己的代码库中，从而无法利用后来的错误修复和改进。Puppet-librarian和Chef-librarian试图解决这个问题，它使人们可以很容易地声明自己的模块依赖性，包括从这些社区网站拉入已知版本的代码。



| 采纳              | 试用            | 评估            | 保持                     |
|-----------------|---------------|---------------|------------------------|
| 52 D3           | 60 Apache Pig | 68 基于浏览器的模板设计 | 76 扫雪机分析               |
| 53 嵌入式Servlet容器 | 61 加特林        | 69 法拉第        | 77 仪表盘上的"UIAutomator"。 |
| 54 弗兰克          | 55 渐进式        | 56 石墨         | 79 不变的服务器              |



58 淘宝网  
59 AAA

# 工具

62 杰基尔  
63 蝗虫  
64 Logstash & Graylog2  
65 PhantomJS  
66 木偶图书管理员和厨师图书管理员  
67 TestFlight & HockeyApp

70 Hystrix  
71 图标字体  
72 灯台  
73 章鱼  
74 .Net的反应性扩展  
75 黎曼

**控股**  
78 重量级的测试工具  
79 雯雯  
80 TFS

管理分布式系统中的依赖关系可能会变得很复杂，而且随着向更细粒度的微服务的转移，越来越多的人面临着这个问题。

**Hystrix**是Netflix为JVM提供的一个库，它实现了处理以下问题的模式

下游故障，提供连接的实时监控，以及缓存和批处理机制，使服务间的依赖性更加有效。

**TestFlight**和**HockeyApp**都允许你在没有应用商店的情况下管理移动应用程序的部署，使用户测试更容易。它们提供崩溃报告和分析功能，以收集现场数据。**HockeyApp**支持iOS、Android和Windows Phone，而**TestFlight**支持iOS和Android。我们已经成功地使用这两个工具来帮助交付移动应用程序。这显然是一个快速发展的空间。

**Frank**是一个开源的库，允许用Cucumber编写的iOS功能测试，并在远程设备上执行。

这填补了一个重要的利基，在那里，验收测试驱动的开发以前是繁琐和笨拙的。

**UIAutomator**看起来是测试Android用户界面的最有前途的工具，它允许在测试期间对组件进行细粒度控制，并促进在多个设备上的测试。

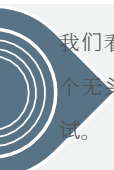
随着具有多种外形因素和像素密度的设备的兴起，在各种比例下呈现高质量的图标的问题已经变得很重要。图标字体通过使用浏览器对WebFonts和SVG的支持来解决这个问题，而不是按比例缩放或维护不同的图标集。一如既往，在广泛使用SVG时，要注意移动设备的功耗和旧设备的性能。

随着我们建立的系统涉及比以往任何时候都更精细的服务，如何获得信息汇总以方便识别和解决问题的挑战比以往任何时候都更加紧迫。

**Logstash**的出现是一种在源头解析和过滤日志，然后将它们转发到一个单一的聚合点的简单方法。虽然**Logstash**提供了一些搜索和过滤功能，但**Graylog2**经常被用来提供功能更全面的查询和报告。

我们看到了**Snowplow Analytics**的巨大前景，这是一个开源的网络分析平台，基于开放数据原则和云存储，从常规的网络分

析中得出智能信息。



我们看到ThoughtWorks项目对**PhantomJS**的兴趣，这是一个无头的网络测试工具，可以针对现实的目标进行功能测试。

## 工具

**Gatling**是自动化性能测试领域的另一个新玩家。它与Locust类似，比JMeter和Grinder等老的选项要轻得多。该DSL建立在Scala之上，提供了大量的功能，包括容易配置的数据源和响应断言。在需要定制的情况下，很容易将其放入Scala提供扩展。通过Highcharts默认生成的大量动态数据视图增加了其吸引力。

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**。这些工具有一些问题，使它们不适合于快速移动的软件交付。大型复杂的工具有很高的学习曲线，需要专业的技能和培训，使得它对团队自己来说很难测试。由于其他团队的参与，这往往会导致每次发布都有不必要的开销。昂贵的和限制性的软件许可证使这个问题变得更加糟糕。一些重量级的工具使用 "模型驱动 "的方法，试图对应用程序的使用模式进行精确建模，这导致了昂贵的测试脚本维护和开发时间被 "假阳性 "所浪费。我们已经看到了一些情况，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。

与Ant和Maven等基于XML和插件的工具相比，**Gradle**和**Rake**等基于语言的构建工具长期以来一直提供更精细的抽象和更多的灵活性。这使它们能够随着项目的复杂化而优雅地成长。

我们继续看到团队在尝试使用**TFS**作为版本控制系统时遇到了生产力问题。那些希望频繁检查代码的团队，也就是持续集成的核心部分，发现其重量级的方法大大消耗了生产力。这常常导致团队检查的频率较低，造成更多的合并问题。我们推荐使用Git、Perforce和Subversion等工具。

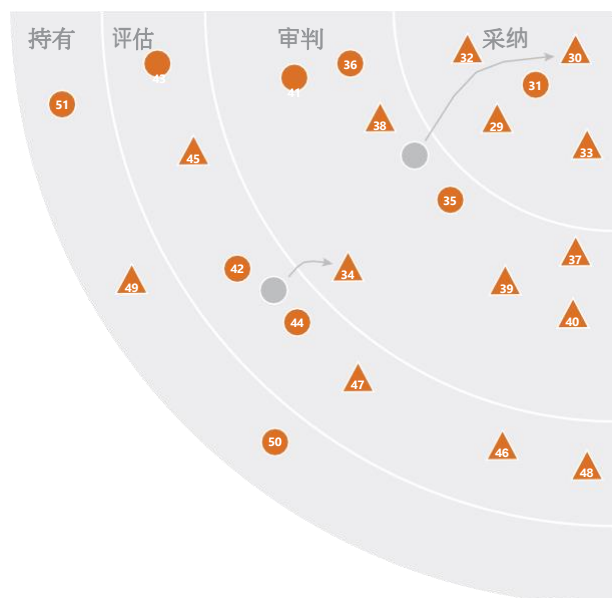
**PostgreSQL**正在扩展，成为SQL数据库的NoSQL选择。9.2版本包括存储JSON数据的能力，并具有对JSON内容的完全查询能力。

文件。其他扩展让用户以键/值对的形式存储和查询数据。这让你可以利用

在不被关系型数据模型所束缚的情况下，可以使用久经考验的数据库的底层存储和交易能力。这对于那些既想使用SQL又想使用NoSQL但又喜欢使用他们已经知道如何支持的单一可靠基础设施的人来说是理想的。

即使是一个相对较少的网站，也能产生巨大的数据量。一旦你加入了分析、业务指标、人口统计、用户资料和多种设备。它可能会变得不堪重负。许多组织将数据仓库作为一个存储库，从组织的各个部分吸纳数据。这里的挑战是，这些数据往往变成了"数据堡垒"。即使是及时获得业务指标也是一个挑战，更不用说在整个数据集中运行探索性查询。像基于云的**BigQuery**这样的技术可以提供帮助。现收现付的模式和临时查询的能力让你无需购买专业的硬件和软件就能获得洞察力。一个数据驱动的企业应该把

决策者手中的数据，而不是隐藏在技术障碍和官僚主义背后。



对于符合文档数据库模型的问题，**MongoDB**现在是最受欢迎的选择。除了易用性和坚实的技术实现外，社区和生态系统也为这一成功做出了贡献。我们知道，在一些问题上，团队被**MongoDB**的流行所诱惑，当

在这种情况下，他们认为文档数据库并不适合，或者他们不了解其固有的复杂性。然而，如果使用得当，**MongoDB**已经在许多项目中证明了自己。

**Redis**已经在ThoughtWorks的多个项目中被证明是一个有用的工具，它被用作分布在多个国家的结构化高速缓存和数据存储。

**Hadoop**最初的架构是基于横向扩展数据和纵向扩展元数据的范式。虽然数据存储和处理由从属节点合理地处理，但管理元数据的主节点是一个单点故障，对网络规模的使用有限制。

**Hadoop 2.0**对HDFS和Map Reduce框架都进行了重大的重新架构，以解决这些问题。**HDFS**名称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HA模式部署。**MapReduce**已经被**YARN**取代，它将集群资源管理与作业状态管理解耦，并消除了规模/性能问题的JobTracker。最重要的是，这一变化鼓励在**Hadoop**集群上部署除**MapReduce**之外的新的分布式编程范式。

在过去的一年里，我们看到**Elastic Search**作为一个开源的搜索平台被逐渐采纳。它是一个基于**Apache Lucene**的可扩展、多租户和水平可扩展的搜索解决方案。它允许通过基于**JSON**的**REST API**对复杂的数据结构进行索引和检索。它提供了一个优雅的操作模式，自动发现集群中的对等物，故障转移和复制。弹性搜索可以通过一个插件系统进行扩展，允许添加新的功能和改变现有行为。围绕这个工具社区是相当活跃的，诸如**Java**、**C#**、**Ruby**和**JavaScript**等语言的客户端库的数量就说明了这一点。





### 采纳

- 29 弹性搜索
- 30 梦之城\_梦之城娱乐\_梦之城国际娱乐\_梦之城国际娱乐平台
- 31 Neo4J
- 32 Redis
- 33 短信和USSD作为一种用户界面

### 试用

- 34 BigQuery
- 35 云中的持续集成 Couchbase
- 36 Hadoop 2.0
- 37 Node.js
- 38 OpenStack
- 39

- 40 洛克菲勒云
- 41 Riak

### 评估

- 42 蔚蓝
- 43 卡拉特拉瓦
- 44 卫星导航系统(Datomic)
- 45 PhoneGap/Apache Cordova

- 46 用于NoSQL的PostgreSQL
- 47 Vumi
- 48 Zepto.js

### 控股

- 49 大型企业解决方案
- 50 单一的基础设施
- 51 WS-\*



**Node.js**是一个轻量级的网络容器，是开发微服务和作为移动和单页网络应用的服务器的有力选择。由于**node.js**的异步性，开发人员正在转向承诺库以简化他们的应用程序代码。随着诺言的使用在**node.js**社区中的成熟，我们希望看到更多的应用程序为**node.js**开发。对于那些不愿意在生产中尝试**node.js**的团队，仍然值得考虑**node.js**的开发任务，如在浏览器外运行JavaScript测试，或从CoffeeScript、SASS和LESS等工具生成静态网页内容。

**Zepto.js**是一个轻量级的JavaScript库，主要基于jQuery。它的API与jQuery相同，尽管它不提供与jQuery的完全兼容。由于文件大小被大大压缩，Zepto是构建响应式网络应用程序时的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程序的平台。它通过一组在不同移动平台上保持一致的JavaScript APIs抽象出平台特定的本地代码。Cordova可用于广泛的平台，包括iOS、Android、黑莓、Windows Phone和WebOS。

虽然AWS继续增加更多的功能，但**Rackspace云**已经成为存储和计算领域的一个可行的竞争对手。一些用户可能会看重Rackspace提供的更彻底的客户支持，以及混合更多传统托管模式的能力。我们对此并不感到兴奋，只是因为Rackspace是我们的客户，而且我们有幸开发了这个平台。我们已经在其他几个客户那里成功地使用了Rackspace云，并期待着它能在更多的地区提供。

开源的**OpenStack**项目正在积聚能量，最近几个月正在成为部署你自己的私有云的一个更可行的平台。许多使OpenStack难以启动和运行的问题已经得到解决，而且新的功能也在不断增加。很明显，OpenStack联盟及其成员，如Rackspace、Redhat和HP，都致力于将该项目作为他们自己基于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为这些地区开发，你需要在开发时考虑到这一制约因素。这些手机使用**SMS和USSD作为用户接口**。SMS是一种长期存在的发送消息的技术，而USSD允许你在一个安全的会话中发送类似于SMS的消息。你应该把USSD和SMS看作是另一个用户界面和用户体验平台，并把它们当作一流的公民。

**Vumi**是一个可扩展的开源信息传递引擎，在移动设备上通过节俭的方法推动对话。Vumi促进了公司和他们的客户、医疗服务和他们的病人之间的短信、即时通讯和USSD互动。政府和公民，以及更多。Vumi与电信公司整合，并允许你在其上轻松建立应用程序。你只需要支付运营商的费用。

企业级“商业软件包所提供的东西与实际需要的东西之间的差距正在扩大。这对于面向互联网的应用来说尤其如此。

真正能够扩展并轻松支持现代技术（如持续交付）的创新解决方案是由从业者为从业者编写的。它们起源于许多互联网规模的公司，并作为开源软件被完善。**大型企业的解决方案**往往阻碍了有效的

由于其累积的臃肿、繁琐的许可限制以及由检查清单和想象中的需求所驱动的功能集，与大多数开发团队的现实情况相去甚远，因此在交付方面存在着一些问题。

# 语言和框架

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Web应用开发继续蓬勃发展。ThoughtWorks项目中正在使用的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。我们期待着在这个充满活力的空间中继续创新。

基于单页和移动浏览器的应用扩展到主流用途，加上服务器端应用的node.js的持续增长，导致越来越多的人采用CoffeeScript来简化JavaScript代码库。作为一种编译成JavaScript代码在运行时执行的语言，许多人对调试用CoffeeScript编写的应用程序的难度表示担忧。在CoffeeScript 1.6.1中引入的源码图正在帮助开发工具的生产商解决这一问题。我们预计这将导致该语言在Dropbox等高知名度技术公司的带领下被进一步采用。

我们在新的生产应用中持续使用node.js，这再次强化了我们对JavaScript代码和库的可靠包装的需求。Node Package Manager (npm) 是node.js生态系统的重要组成部分，是一个有用的工具。用于打包node.js应用程序。含有大量JavaScript或CoffeeScript的浏览器应用程序的开发者应该考虑Require.js，以帮助构建他们的代码并在运行时加载依赖性。

微框架作为处理客户端和服务器端应用程序中日益增长的复杂性的一种方式正在出现。

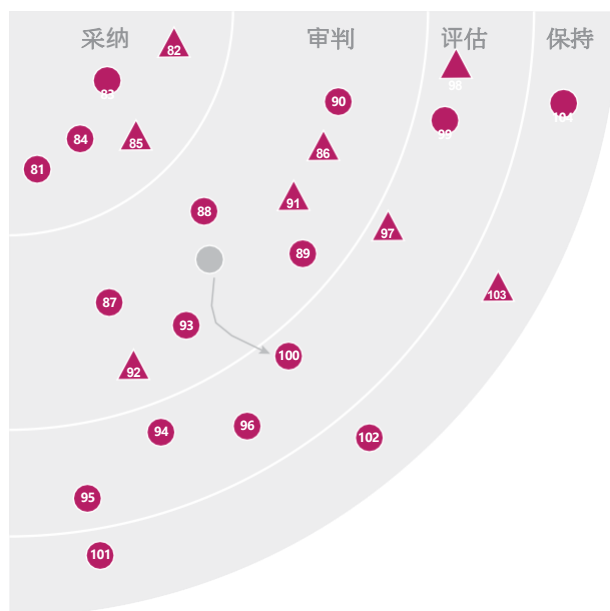
Sinatra是这种趋势在服务器端的第一个例子，它提供了一个轻量级的DSL，以建立可以轻松组成的快速服务。其他语言也有类似的产品，包括Java的Spark、Python的Flask、Scala的Sclatra、Clojure的Compojure和.NET的Nancy。

在.NET平台上，有一件事减缓了丰富的、开源的网络开发生态系统的发展，那就是对IIS和ASP.NET框架的过度依赖。OWIN指定了一个开放的HTTP处理接口，将网络服务器和应用程序分离开来，就像Rack为Ruby做的那样。

社区。我们对OWIN感到兴奋，因为它为由简单的、独立开发的模块组成的新的.NET网络开发工具提供了可能性。南希就是一个完美的例子。我们也希望它能增加在.NET平台上将网络应用程序部署为独立的、自我托管的服务的做法。

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、资产管道化和灵活的部署选项，使其对开发者更具吸引力。由于这个原因，Play重新出现在雷达上，成为团队在JVM上构建网络应用和服务时需要认真考虑的东西。一句话

然而，Play采用了函数式编程风格，当与Java语言一起工作时，仍然转化为大量的静态方法，可能很难在运行的服务器之外进行单元测试。



## 采纳

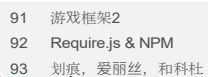
81 Clojure  
82 CSS框架

83 茉莉花与Node.js的搭配  
84 宜麟盆语  
85 西纳特拉

## 试用

86 咖啡色脚本  
87 滴水观音

88 用于离线应用的HTML5  
89 作为一个平台的JavaScript  
90 JavaScript MV\*框架



## 评估

|     |               |
|-----|---------------|
| 94  | ClojureScript |
| 95  | 格雷姆林          |
| 96  | 吕阿            |
| 97  | 南希            |
| 98  | OWIN          |
| 99  | 润彩客网          |
| 100 | 推特Bootstrap   |

控股

|     |                  |
|-----|------------------|
| 101 | Backbone<br>e.js |
| 102 | 基于组<br>件的框<br>架  |
| 103 | 手写的<br>CSS       |
| 104 | 存储程<br>序中的<br>逻辑 |



# 语言和框架

与JavaScript和HTML一样，CSS是创建网站的一项核心技术。不幸的是，该语言本身缺乏关键功能，这导致了大量的重复和缺乏有意义的抽象。虽然CSS3旨在纠正其中的一些问题，但构成CSS3的模块在大多数浏览器中得到适当支持还需要几年时间。幸运的是。现在有一个解决方案，那就是使用SASS、SCSS和LESS等**CSS框架**。由于它们的质量和支

持，我们相信，除了琐碎的工作之外，**手写CSS的日子已经过去了**。

**CSS框架**简化了大规模的CSS代码库的开发，而不必每次都从头开始。由于有大量的框架，选择一个能够持续**增强**和**维护**的框架非常重要。的代码库，而不是仅仅帮助你快速入门的东西。基于混合插件的框架，如指南针，或有特定重点的，如苏西，在这方面更适合。

**Twitter Bootstrap**是一个流行的CSS框架，它允许你快速制作一个具有流体和响应式布局的漂亮网站。在这一期的雷达中，**Bootstrap**根据我们长期以来的使用经验，从试用转回评估。如果你希望替换或广泛地定制你的应用程序的外观和感觉，**Bootstrap**可能会带来挑战，因为它与HTML标记深度集成。这并不一定使它成为一个糟糕的选择，但在选择它而不是可用的替代方案时，值得记住这些限制。

# 参考文献

**Apache Cordova:** <http://cordova.apache.org/>

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

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

**CoffeeScript不是一种值得学习的语言:** <https://github.com/raganwald/homoiconic/blob/master/2011/12/jargon.md>

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

**Dropbox潜心研究CoffeeScript:** <https://tech.dropbox.com/2012/09/dropbox-dives-into-coffeescript/>

**DropWizard:** <http://dropwizard.codahale.com/>

**Faraday:** <https://github.com/lostisland/faraday>

**Graylog2:** <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>

**Logstash:** <http://logstash.net/>

**移动电话的使用:** <http://www.idc.com/getdoc.jsp?containerId=prUS23982813#.UTTAZzCG2TU>

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

**Nancy Framework:** <http://nancyfx.org/>

**NPM:** <https://npmjs.org/>

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

**Octopus:** <http://octopusdeploy.com/>

**OWIN:** [owin.org](http://owin.org)

**PhantomJS:** <http://phantomjs.org/>

**PhoneGap:**

<http://phonegap.com/>

**Plans:**

<https://github.com/plans>

**反应性扩展:** <https://rx.codeplex.com/>

**RequireJS:** <http://requirejs.org/>

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

**Snowflake服务器:** <http://martinfowler.com/bliki/SnowflakeServer.html>

**源地图修订版3建议:** [https://docs.google.com/document/d/1U1RGAehQwRypUTovF1KRlpjOFze0b-\\_2gc6fAH0KY0k/edit](https://docs.google.com/document/d/1U1RGAehQwRypUTovF1KRlpjOFze0b-_2gc6fAH0KY0k/edit)

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

**USSD:** [http://en.wikipedia.org/wiki/Unstructured\\_Supplementary\\_Service\\_Data](http://en.wikipedia.org/wiki/Unstructured_Supplementary_Service_Data)

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

**为什么每个人不是讨厌就是离开Maven:** [http://nealford.com/memeagora/2013/01/22/why\\_everyone\\_eventually\\_hates\\_maven.html](http://nealford.com/memeagora/2013/01/22/why_everyone_eventually_hates_maven.html)

## ThoughtWorks是一家全球性的IT咨询公司

ThoughtWorks是一家软件公司，也是一个由充满激情的人组成的社区，其宗旨是彻底改变软件的创造和交付，同时倡导积极的社会变革。我们的产品部门，ThoughtWorks Studios，为那些渴望成为伟大的软件团队制造先锋工具；如Mingle®、Go™和Twist®，帮助组织更好地协作和交付高质量的软件。我们的客户是具有雄心壮志的人和组织；我们提供颠覆性的思维和技术，使他们能够取得成功。在我们成立的第20年，约有2500名ThoughtWorks员工--"ThoughtWorkers"--在澳大利亚、巴西、加拿大、中国、德国、印度、新加坡、南非、乌干达、英国和美国的办事处为我们的客户提供服务。