

探讨Java与Ruby语言迁移时的安全性[1] PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/244/2021_2022__E6_8E_A2_E8_AE_A8Java_c104_244669.htm 通常来说，“使用Ruby具有风险”是一种普遍的看法，这存在一定的原因。因为使用新的语言天生是有风险的。随着Ruby on Rails逐步进入到主流的开发领域中，这样的风险将会随时间逐渐降低，因为有逐步增长的开发者群、组件(或称作gems和plug-ins)相关的书籍、以及业务合作伙伴与你沟通交流。但同时你也可以听到主流的观点指出“使用Java是安全的”。对于这种的观点，我持有强烈的反对意见。随着语言的膨胀，这样的风险通常也会增长。为了便于理解在目前在这些观点上正发生什么变化，投入点精力去研究Java最初的应用情况是值得的。新技术采用概况 许多分析家拥有技术应用所需的描述模型。其中最为流行的模型是定义在Ruby的Web开发框架Iowa中，用来描述农产品的应用，稍后在一本由Geoffrey A. Moore写作的名为《跨越鸿沟》(Crossing the Chasm)的书中，被用来描述技术内容。在书中，Moore分析了技术应用周期中存在着的五个截然不同的群体：技术专家。这个群体倾向于采用新的技术。任何一种有前途的技术都会引起这个群体的注意。先行采纳者。不管这项技术是否在主流技术中取得成功，这个群体都将会采用新的技术来提升竞争优势。实用主义者。一旦新的技术进入主流应用，或是有足够陡峭的增长曲线来保证技术将得到广泛采用，那么实用主义者就会积极采用新的技术。保守派。只有新技术成为必须的时候，他们才会考虑采用新的技术。怀疑论者。这个群体可能很晚才会采用新的技术，或者

也可能永远只使用某一特定技术。 Moore指出，技术应用的 关键之处在于团队中是否存在实用主义者。因为实用主义者 需要新技术大规模的应用，这个中间群体希望看到其他务实 派在团队做出承诺之前就使用新的技术。这是一个类似于《 第二十二条军规》书中所描述的现象，因为务实派们都会相 互依赖的存在。出于这样的原因，在先行采纳者排列在技术 专家之后和务实派之前，你会经常在市场接受度曲线中看到 一种下降的趋势。 Moore将这种下降称之为鸿沟倾向，并且 这种想法应出于围绕任何新技术的风险讨论的中心。 Moore 解决方法是，把重点放在跨越鸿沟的过程中。通常来说，你 很难通过一个巨大的飞跃跨过鸿沟。你需要有一个目标明确 的细分市场。 Java技术首先通过Applet程序进入网络客户端， 之后转向服务端的计算、移动终端、以及其他类似于移动计 算以及企业架构的应用，最终为网络带来强大冲击。在《超 越Java》一书中，我认为存在于程序设计语言之间的鸿沟特别 严重。我们大多数人都认识到在Lisp语言上投入精力将大幅提 高生产率，但是同时也会导致更难找到合适的程序开发人员 、教学资源、类库以及组件等。同时我们还将不得不付出更 多的花费来进行一些必要的整合工作。出于这个原因，大众 市场只会以大约每十年的时间周期更换主流的编程语言。在 服务端编程语言方面，可以清晰看到这种趋势的存在

。 COBOL和Fortran语言出现于1954年到1961年之间。 C语言 则诞生在上世纪70年代初期， C是出现在上世纪80年代中期 ， Java语言则出现在1996年。我应当把C#语言算做整合高效 的Java语言克隆版本，虽然这样的说法可能会引发一些争辩。 许多其他的语言在此阶段中诞生，但是上述语言仍旧没有一

个能够占据统治地位。伴随的风险是阻碍新编程语言被广泛采用的最重要原因。Java的风险概况 使用Java语言曾经需要克服很大的风险。当时，大多数服务端的编程都在使用C语言。C是一门高效的操作系统语言，非常适用于应用程序开发。C语言家族在这方面的表现相当出色，因为客户机/服务器端编程以及用户界面开发需要程序性能与适应性良好地结合在一起，在当时其他的编程语言都无法符合这样的要求。为了克服伴随采用新编程语言而来的风险，Java需要以下的三个条件均成立：C开发者不得不经历一番辛苦的学习过程。指针的存在(由于缺少编译时的安全性)导致各种各样难以消除的缺陷。内存管理使得内存泄漏成为家常便饭。C对于大多数程序开发者来说，显得过于复杂。这些问题增加了针对于C语言的风险评估。Java需要解决一些C语言无法处理的工作。Java语言所具有简洁、灵活的特性以及众多C所不包括的类库支持。这些要素减少了针对于Java语言的风险评估，并可以保持开发团队小型化最终从根本上提高生产力。Java需要一个催化剂。随着网络爆炸，Applet应用普遍被嵌入在NetScape浏览器中，使得C语言开发者不得不转向去开始使用Java语言。C因为和Java语法的类似，可以简单地进行过渡。Java得以迅速获得数量庞大的用户群，并且在同微软的竞争中逐步提升这样的过渡。Java的膨胀要比我们以前所见的任何一次技术浪潮都要迅速，同时也可能比我一生所见的任何技术都要庞大，然而Java的发展蓝图却一直保持清晰。为了建立新的语言，原有的语言已不适应开发者的需求，新的语言必须要克服原有语言的缺陷，并最终以某些催化效应迅速聚集起数量庞大的用户群。 100Test 下载频道开通，各类考试题

目直接下载。详细请访问 www.100test.com