

探讨Java与Ruby语言迁移时的安全性[3] 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\\_244667.htm](https://www.100test.com/kao_ti2020/244/2021_2022__E6_8E_A2_E8_AE_A8Java_c104_244667.htm) 还有其他许多类似的风险。然而，你可以有效地降低使用Ruby语言的风险，比如采取绩效挂钩的风险预测。虽然开发和部署大型Ruby应用的相关知识积累仍然十分有限，但是你可以在适当的着眼点不断学习新的知识。对于PHP、Perl和Python等LAMP相关语言，业界有着非常丰富的知识积累。在应用部署机制、Web服务器以及非共享可拓展策略等方面都是一致的。在考虑参与开发的人手时，不要低估你通过对员工进行内部培训来建立高效团队的能力。对于使用Spring、Eclipse、Hibernate和WebWork进行Java开发的新手，我的训练计划常常是为Ruby on Rails开发者指定培训计划的五倍。如果你开始使用具有类似于Ruby特性的开发语言，比方说Perl，Python或Smalltalk，它们可以帮助你很好地起步。如果你打算从零开始培养一个程序员的话，培养一个使用Ruby的开发者，远比培训Java开发者使用最新的一大堆各种框架要合算的多。想一想那些众多的函数类库，有多少是你真正需要的？如果你需要分布式处理，双相提交，那么就使用Java编程。如果您需要与Microsoft Office的宏完美地整合，那么就使用.NET。但如果你想编写操作系统整合脚本，或编写基于数据库的绿色Web应用，那么Ruby则正是你所需要的。并且你可以经常编写要用到但手边没有的任何程序。我曾协助一家公司工作，他们在两个星期内编写了自己的数据库驱动程序，但仍然比完成项目其他工作所用的时间要多。我还认识一个使用Ruby在四小时内修补现有代码

，为程序拓展Oracle支持的开发者。Thoughtworks在很短的开发周期内就发布了RBatis，即Ruby版本的实体关系映射工具iBATIS。所以当你站在全局考虑时，会感觉到使用Ruby的风险往往被夸大了，尤其是在Java并没有带给你一切所需资源的时候。自己真正去尝试使用Ruby语言，是把这些风险纳入控制范围之内的最好方法。使用Rails开发一些实际的应用，并站在实践的角度上发言。而不要盲目迷信别人的说法。神话 vs 事实 Rails是银弹。人们曾经在Rails项目上失败过，并且还将会有更多失败的教训。如果你在具备必须技能的情况下使用它，你也将可能面临失败的命运。与之类似的说明是，如果Java语言不是导致失败的问题根源，那么Ruby将同样不会是你的答案。大多数软件开发问题的出现是与特定技术无关的。如果你正在遭受打击，Ruby on Rails的采用只能加快你遭受打击的速度。选择Ruby颇具风险，因为你无法预测到错误。采用任何新的语言，最主要的风险是你将预测到错误，并且错误停滞在使用的类库之中。这的确是一项相当重大的风险，但是这个问题决不仅局限于Ruby语言之中。在Java语言里，你需要就主要类库的使用做出决定，其中任何一个都可能带给你复杂臃肿的代码。你是否会为声明事物选择Spring或EJB 3等技术?Java的持久层架构是不是一个正确的选择，或者Hibernate就是最终的解决方案?关于Web MVC分层的正确选择是什么，是逐步衰落的Struts框架，还是其他更易用的框架?在Ruby语言之中，选择Web开发框架则相对简单许多。你将很可能与Rails一起工作。语言动态的特性同样各层之间的结构更为简化，通过特定的约定来使得开发配置比Java实现更为明晰。为Java项目招募人手总是更为容易。Java拥有

数量庞大的开发者群体，但是开发社区之间有着巨大的分歧。如果你想使用一个综合的Java工具集，你的选择是十分有限的。即使你选择了像Spring这样的流行框架，你的团队必须还要学会使用针对给定项目所需的各种类库。在这种情况下，Java的核心力量，过多的函数类库，将会给项目带来副作用。相反，大部分的Ruby开发者都知道Rails框架。此外，你通常需要更多的Java开发者去处理类似的任务。有时，招募Java的开发人员要容易得多。但有时，情况也并不是这样。Rails无法拓展。Ruby on Rails其实有很好的延展性。它的缓存模型非常强大，并且非共享的架构在LAMP社区中多次被证明是非常有效的。实际上，我们知道Ruby on Rails完全可以适应较大型应用的要求。我们不知道Ruby on Rails是否可以承受大规模的应用部署。没有固有的架构使我相信这是一条死胡同。对于典型的应用，总之错误的潜伏期是存在于数据库端。Rails的整合选项十分有限。Rails对于基于ReST的Web服务有着良好的支持。Ruby同样通过JRuby项目提供对于JVM的支持，以及提供对于微软的CLR运行时的支持。同时Ruby也提供了良好的消息传输支持。最后，为项目选择最好的工具将会帮助你始终处于良好的状态。优秀的开发团队可以在Java和Ruby项目上同时获得成功。总结：你可以承担什么样的角色？如果你正在考虑使用Ruby，那么在你身边将会有很多有用的信息。与其他同时在有效使用Java和Ruby的开发者交流。阅读关于开发框架的资料。查找从Java到Ruby的迁移资料。如果你并不想放弃Java，只是想寻找轻量级的开发体验，那么去了解一下那些可以为你带来更多相关体验的项目，比如说RIFE、JMatter或Wicket项目。如果你认为Ruby可能是一个

好的选择，那么要留心以下的建议：为项目选择合适的工具。Ruby on Rails并不是银弹，ROR是一个针对以数据库为后台的高度精简的Web应用开发环境。与新的数据库模式配合较好，或者你可以通过变更来适应Rails的各种固有优点。细心计划开发团队的热身阶段。你不需要在Monster.com站点投放广告并在三日之内为项目招募齐全开发人员。但你可能需要考虑培训你部分或全部的开发者，并且招募几个顶尖的Rails开发者，或是请求某些项目咨询来帮助你把项目启动。了解你使用传统方式的结合点。通常，项目中最头疼的部分是定义与外部系统的交互。你最初证明概念的工作需要与某些接触点交互，至少是要明确你在何处对项目感觉到满意。如果你还是不确定，那么做一个先行者，或是遵从保守派的观点。缓解风险最佳的方法总是优秀的判断能力。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](http://www.100test.com)