# 运行效率

极简设计，概念最少化，极薄封装，从架构上保障性能最大化。

jfinal 有很多大型项目已经运行多年，例如 trafree.com ，每天千万级别的请求，去哪儿和淘宝旅行都用的他们的 API。

# 开发效率

开发效率极高

# 学习成本

学习成本低，但比较于市面上的开源框架，jfinal的使用方式有较大的差别。

# 运维成本

与spring和hibernate比较没有什么优势。

# 稳定性

封装层次较低，程序目前较为稳定

# 版本迭代

目前为单人单独进行程序迭代，bug修复和新功能迭代较为及时。有发展前景

# 安全性

暂未爆出有严重的系统漏洞和信息安全问题。

# 适用性

有一个比较普遍的误解认为 jfinal 只适合做小型项目，其实 jfinal 更加适合做大型项目，只需要引入业务层就好，有这种误解可能是看到 jfinal demo 中没有业务层造成的。

现在炒得比较热的微服务，也即大型项目拆分成小型服务做分布式 + 集群的方式，这个其实在很多年前巨头公司都是这样做的，本质就是老祖宗说的分而治之，jfinal 作为一个 web + orm 框架，可以处于这些微服务之中，无论你多大的项目 jfinal 都可以用。

Jfinal已经支持传统Java项目、web项目、微信、微服务等。

但是目前Jfinal还处于一个初始阶段，没有一个成熟的商业化使用规范，甚至没有一个专业的综合新能参数。仅仅靠一个人去维护整个工程，很难达到一个理想的高度。目前市面上大多以一些追求开发效率为主的开发者在使用Jfinal。每当一种新技术产生的时候，都会伴随着开发者学习使用，从公司层面上来说是不会轻易使用jfinal的。从软件工程的设计来讲，也不会轻易使用jfinal的。与spring boot相比，现在spring 高度家族化，几乎能涵盖Java能涉及的所有方面，稳定性、安全性、主流程度都非常高。

虽然jfinal的未来很好，但是目前来说还不推荐贸然使用。