**代码审查综述和工具**

1. 代码审查定义

<http://zh.wikipedia.org/wiki/%E4%BB%A3%E7%A0%81%E5%AE%A1%E6%9F%A5>

1. 11个高效的同行代码评审最佳实践
2. [一次评审量要低于 200–400 行代码](http://www.ibm.com/developerworks/cn/rational/11-proven-practices-for-peer-review/#1. 一次评审量要低于 200–400 行代码)
3. [每小时低于 300–500 LOC 检查率的目标](http://www.ibm.com/developerworks/cn/rational/11-proven-practices-for-peer-review/#2. 每小时低于 300–500 LOC 检查率的目标)
4. [花足够的时间进行适当缓慢的评审，但是不要超过 60-90 分钟](http://www.ibm.com/developerworks/cn/rational/11-proven-practices-for-peer-review/#3. 花足够的时间进行适当缓慢的评审，但是不要超过 60-90 分钟)
5. [确定在评审开始之前代码开发者已经注释源代码了](http://www.ibm.com/developerworks/cn/rational/11-proven-practices-for-peer-review/" \l "4. 确定在评审开始之前代码开发者已经注释源代码了)
6. [为代码评审创建可定量化的目标，并获取制度，这样您就可以改进流程了](http://www.ibm.com/developerworks/cn/rational/11-proven-practices-for-peer-review/" \l "5. 为代码评审创建可定量化的目标，并获取制度，这样您就可以改进流程了)
7. [使用检查表，因为它能极大地影响代码开发者和评审者的结果](http://www.ibm.com/developerworks/cn/rational/11-proven-practices-for-peer-review/" \l "6. 使用检查表，因为它能极大地影响代码开发者和评审者的结果)
8. [确认缺陷得到了修复](http://www.ibm.com/developerworks/cn/rational/11-proven-practices-for-peer-review/#7. 确认缺陷得到了修复)
9. [培养良好的代码评审文化氛围，在这样的氛围中可以积极地评审缺陷](http://www.ibm.com/developerworks/cn/rational/11-proven-practices-for-peer-review/#8. 培养良好的代码评审文化氛围，在这样的氛围中可以积极地评审缺陷)
10. [警惕老大哥效应（Big Brother Effect）](http://www.ibm.com/developerworks/cn/rational/11-proven-practices-for-peer-review/#9. 警惕老大哥效应)
11. 评审一部分的代码，就算您不能全部完成，以从自我效能感（Ego Effect）中获益
12. 采用轻量级，工具支持的代码评审

详见：<http://www.ibm.com/developerworks/cn/rational/11-proven-practices-for-peer-review/>

1. Java代码审查工具

<http://fyting.iteye.com/blog/87489>

1. Visual Studio 代码审查工具

<http://www.devart.com/review-assistant/>

1. IBM Rational Team Concert

<http://www.ibm.com/developerworks/cn/rational/products/rtc/>