

# 코드 품질 관리 Part 1

## : 읽을수록 좋아지는 코드 품질의 비결

2013. 1. 21. (제40호)

### 목 차

- I. 코드 결함의 악순환
- II. 코드 리뷰와 지원도구
- III. 탠덤의 역설
- IV. 클린코드(Clean Code)

소프트웨어공학센터  
경영지원TF팀

## I. 코드 결함의 악순환

소프트웨어 개발 회사에서 흔하게 경험하게 되는 코드 결함의 악순환의 시나리오가 하나 있다. 그 시나리오의 시작은 이렇게 시작된다.

시장에서 새로운 기술 환경과 고객의 욕구를 충족시켜줄 수 있는 제품을 많은 노력을 통해 출시하였다. 수많은 데모와 지원을 아끼지 않은 덕에 고객은 하나 둘 늘어나고 고객의 환경에 맞춰 기능들이 추가되기 시작한다. 시간이 지나면서 초기 개발자는 하나 둘씩 바뀌어가고 제품은 버전별로 조금씩 다른 브랜치(branch) 버전이 늘어난다. 제품 업그레이드 이후 결함과 요구사항이 늘어나고 개발 언어와 시스템 업데이트도 유지보수에 큰 부담이 되고 있다. 제품의 성능과 품질상의 이슈들로 문제점을 해결하는데 복잡도가 높아지더니 심각한 결함으로 유지보수에 빨간불이 들어온다.

고객사에 업그레이드된 제품에서 원인 파악이 되지 않는 문제들로 내부 개발자들이 불려나가기 시작한다. 파견 나간 개발자도 문제를 좀처럼 해결하지 못하고 결국 외부 인력까지 동원하여 문제점을 분석시켜보지만 그들도 확실하게 원인을 짚어내지는 못한다. 고객은 시스템을 철거한다, 손해를 배상하라며 으름장을 놓고 이를 수정하려면 임시조치가 아니라 핵심 엔진의 수정이 불가피하다는 결론에 도달한다.

재개발을 놓고 이런저런 토론과 회의가 이어지고 고민에 찬 관리자에게 의욕적인 몇몇 개발자들이 시간을 주면 빠른 시일 내에 재개발을 완료하겠다고 앞장선다. 어차피 현재 상태로는 유지보수도 불가능하다고 하니 큰마음을 먹고 재개발로 방향을 잡아본다. 처음에 약속했던 시간에 다시 몇 배의 시간을 더하더니 거의 완료되었다는 보고만 있을 뿐 재개발은 지연된다. 이미 접기에도 너무 늦은 상황이다 보니 확실한 종료일자는 단단히 못박아두고 인력을 좀 더 투입한다. 새로운 지원에 대한 부담감으로 날밤을 지새우면 드디어 재개발이 완료된다.

기존 제품보다 향상된 성능을 갖춘 신규 제품 위에 새로운 기능들을 추가하는데 전에 없던 문제점들이 발견된다. 안정화를 위해 재개발을 담당했던 핵심 개발자들은 다시 바빠지고 나머지 개발자들은 기존 제품과 새로운 제품 사이에서 혼란스러움만 가중된다. 시간이 지날수록 관리자는 처음의 그 자리로 돌아가는 느낌을 지울 수 없게 된다.

이러한 코드 결함의 악순환은 왜 반복되는 것일까? 코드 결함의 악순환을 방지하려면 개발자의 코드를 그대로 방치하지 말아야 한다. 관리자들은 코드는 관리하지 않고 눈에 보이는 화면과 문서를 통하여 관리하려고 하는데 이는 마치 겉모습과 옷차림만으로 그 사람을 확인하려는 것과 같다. 화려한 옷 안에 숨겨져 있는 몸 안의 기관과 뼈, 근육, 혈관 상태 등을 살펴보지 않고 겉모습만으로 그 사람의 건강함을 판단할 수 없듯이 코드를 관리하지 않고 시스템의 건강함을 확보할 수는 없다.

## II. 코드 리뷰와 지원도구

코드 리뷰를 통하여 결함을 보다 빨리 발견하여 수정 비용을 절감할 수 있을 뿐만 아니라 코딩 표준 준수와 학습효과 증가 등 많은 장점에도 불구하고 국내 소프트웨어 개발 조직에서 코드 리뷰를 잘 수행하는 경우는 드문 편이다. 왜 그런 것일까?

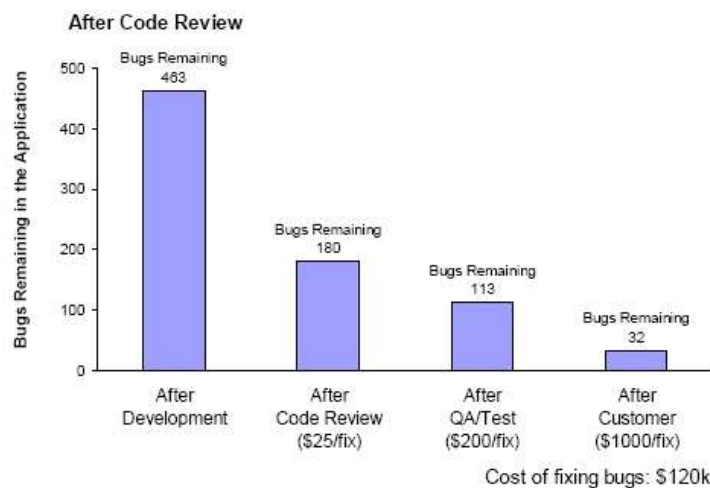
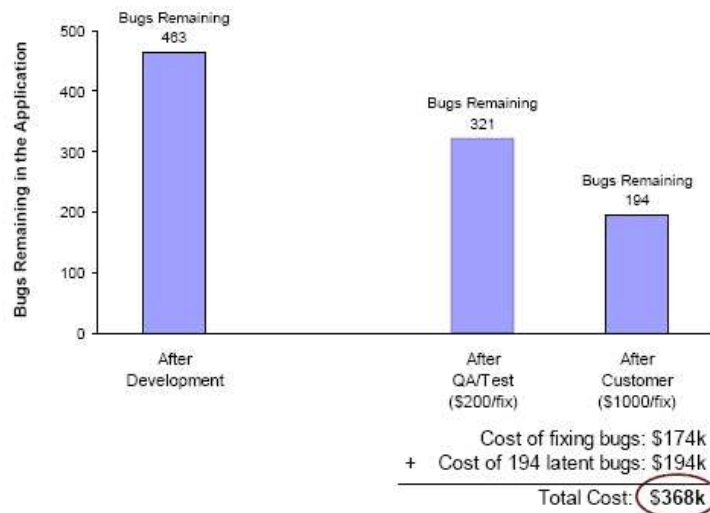
우선 대부분의 개발자들은 코드를 볼 시간이 없다는 말부터 한다. 코드 리뷰 과정보다는 개발할 기능과 수정할 결함들이 발등에 떨어지다 보니 그럴 여유가 없다는 것이다. 또한 개발자 사이에서 코드 리뷰는 좋은 학습 과정이지만 간혹 내 코드를 남에게 보이고 검토를 받는 것에 대한 거부감과 리뷰 과정에서 자존심에 상처를 주는 경우도 있다. 관리자 입장에서 코드 리뷰를 한다고 당장의 성과가 나오지 않을 뿐만 아니라 잘 이해하지 못하는 코드 수준까지 챙기지 못하는 점도 불편한 현실 중에 하나이다. 그런데 코드 리뷰를 하는 경우와 하지 않는 경우 과연 어떤 프로젝트가 개발 비용과 기간에서 보다 나은 결과를 얻을 수 있을까?

아래 <그림 1>은 프로젝트에서 코드 리뷰를 수행한 경우와 그렇지 않은 경우를 비교한 개발 사례이다. 코드 리뷰가 이루어지지 못했던 프로젝트는 개발이 완료 후 남은 결함(버그)수가 483개, 테스트 완료 이후 321개, 고객에게 전달된 이후에도 194개의 결함이 남아있다. 품질관리에서 1:10:100의 법칙으로 잘 알려진 것처럼 결함 비용은 수정하는데 초기에는 적게 들지만 후반으로 갈수록 수정하는데 비용이 증가하게 된다. 그래서 모든 결함을 수정하는데 총 368,000 달러가 사용되었다.

반면에 코드 리뷰가 이루어졌던 프로젝트의 경우에는 개발 완료 이후 남은 결함 수는 483개로 동일하지만 코드 리뷰를 통해서 남은 결함이 180개로 줄어들었다. 테스트 이후 113개, 제품 릴리즈 이후에 남은 결함은 32개에 불과하다. 코드 리뷰에

개발자의 리소스가 추가되었음에도 불구하고 코드 리뷰로 조기에 결함을 제거하여 총 결함 제거 비용에는 120,000 달러가 사용되었다. 코드 리뷰가 없던 프로젝트와 비교할 때 무려 248,000 달러가 절약되는 것을 알 수 있다.

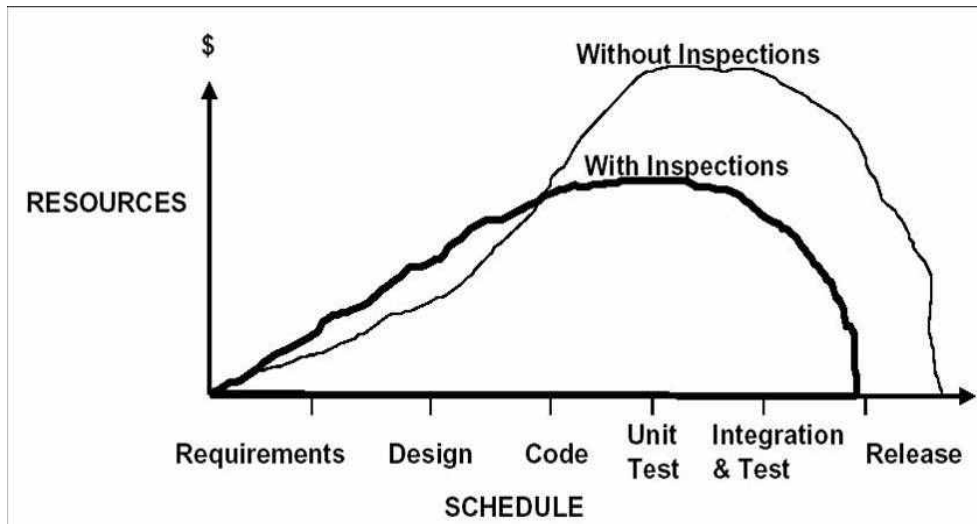
<그림 1> 코드 리뷰 수행 여부에 따른 총 결함 제거 비용 비교



(자료: Best Kept Secrets of Peer Code Review)

개발을 마치고 바로 테스트를 진행하는 것보다 코드 리뷰(동료 검토)를 수행한 뒤 테스트를 수행하는 경우가 결함수정 비용이 절감되는 것뿐만 아니라, 아래 <그림 2>에서 알 수 있듯이 개발 기간도 단축할 수 있게 된다. 이는 개발 이후 코드의 결함 제거가 빨라지면 안정화가 쉽게 이루어지기 때문이다.

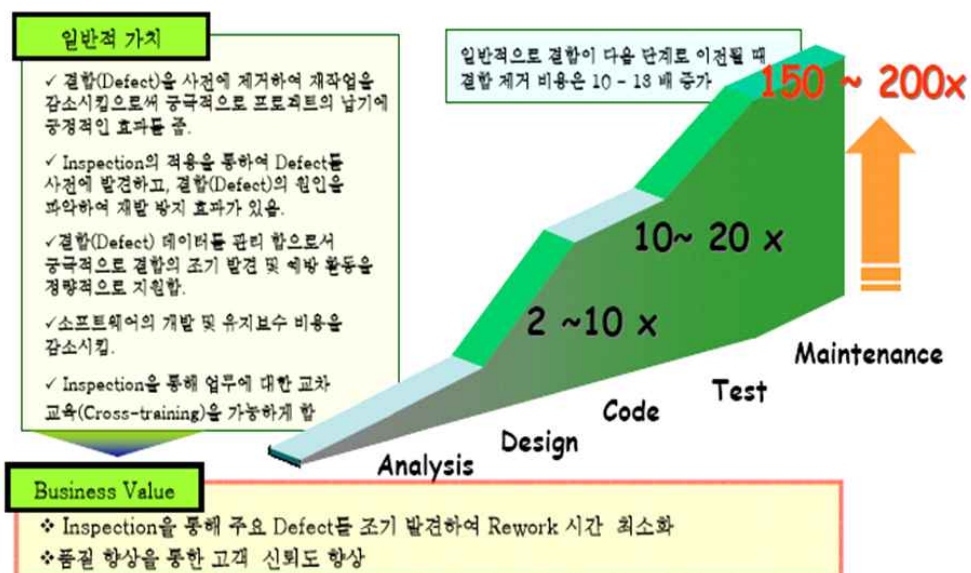
<그림 2> 개발기간의 단축 사례



(자료: Fagan, M. "Advances in Software Inspections", IEEE Transaction on SE, July 1986)

코드 리뷰라고 해서 검토 대상이 꼭 코드로 한정될 이유도 없다. 동료 검토(Peer Review)를 통해 요구사항 분석, 설계, 개발, 테스트에서 단계마다 주기적으로 산출물을 검토하는 것이 필요하다. 실제 많은 결함들은 요구사항의 모호함과 설계의 부정확함에서 기인한다. 요구사항이나 설계 과정에서 결함을 발견하여 수정을 한다면 개발이나 테스트 단계에서 보다 큰 효과를 발휘하게 된다.

<그림 3> 동료 검토의 가치



프로세스 개선의 표준 모델로 사용되는 CMMI(Capability Maturity Model Integration)에서 표준화 단계인 Level 3 이후 Level 4의 정량적 관리로 넘어가게 해주는 핵심 활동 중 하나가 바로 코드 리뷰 활동이다. 각 단계별로 동료 검토(peer review) 활동을 수행하여 결함을 수집하지 않는다면 Level 4의 핵심인 정량적 관리를 달성할 수 없다.

애자일(Agile)에서도 코드 리뷰의 장점을 인정하여 동료 개발(Pair programming)을 통해 실시간 코드 리뷰를 가능하게 만들어준다. 키보드를 주고 있는 드라이버(Driver)와 이를 관찰자(Observer)가 함께 프로그램을 작성하는 것이다. 동료 개발도 개발 단계에서 뿐만 아니라 설계와 테스트 단계에도 확장하여 적용할 수 있다.

코드 리뷰를 위한 공개된 도구들이 여러 가지가 있는데 그중에서 Review board (<http://www.reviewboard.org/>)는 리눅스와 윈도우 서버에 설치가 가능하고 CVS나 Subversion과 같은 코드버전관리 도구도 지원한다. 사용법도 간단해서 자신의 코드를 수정하고 review를 요청하면 검토자는 기존 코드와 변경된 코드를 보고 리뷰 의견을 전달해준다. 검토 의견에 따라 변경 코드를 커밋(commit)하던가 수정하여 다시 리뷰를 받는 과정을 진행할 수 있다.

<그림 4> 리뷰 보드(Review Board) 예제

Welcome, Soonsam(Sam) Hwang - [My account](#) - [Log out](#) - [Bugs](#) - [Report bug](#)


Review Board<sup>beta</sup>



[My Dashboard](#) [New Review Request](#) - [All review requests](#) [Groups](#) [Submitters](#)


<div>★ Starred Reviews0</div> <div>Outgoing Reviews0</div> <div>Incoming Reviews14</div> <div>To Me0</div> <div>Bugzilla1</div> <div>Finance13</div>	<div>All Incoming Review Requests</div> <table><tr><td><div>Summary</div></td><td>Submitter</td><td>Posted</td><td>Last Updated</td><td>...</td></tr><tr><td>☆ Disabling of rv in eptrecv</td><td>kvs</td><td>October 21st, 2009, 11:42 a.m.</td><td>1 day, 14 hours ago</td><td></td></tr><tr><td>☆ Disabling of rv in eptrecv_pf</td><td>kvs</td><td>October 21st, 2009, 11:41 a.m.</td><td>1 day, 14 hours ago</td><td></td></tr><tr><td>☆ [bug 3083068] getSpaceidFromService not using proxy/called multiple times.</td><td>apaz</td><td>October 8th, 2009, 11:41 a.m.</td><td>1 week, 6 days ago</td><td></td></tr><tr><td>☆ yoda_finance_data_component review</td><td>kvs</td><td>October 5th, 2009, 1:44 p.m.</td><td>2 weeks, 2 days ago</td><td></td></tr><tr><td>☆ Rewrote Lassie in a more c++ style</td><td>ajtd</td><td>September 23rd, 2009, 12:22 p.m.</td><td>4 weeks, 1 day ago</td><td></td></tr></table>	<div>Summary</div>	Submitter	Posted	Last Updated	...	☆ Disabling of rv in eptrecv	kvs	October 21st, 2009, 11:42 a.m.	1 day, 14 hours ago		☆ Disabling of rv in eptrecv_pf	kvs	October 21st, 2009, 11:41 a.m.	1 day, 14 hours ago		☆ [bug 3083068] getSpaceidFromService not using proxy/called multiple times.	apaz	October 8th, 2009, 11:41 a.m.	1 week, 6 days ago		☆ yoda_finance_data_component review	kvs	October 5th, 2009, 1:44 p.m.	2 weeks, 2 days ago		☆ Rewrote Lassie in a more c++ style	ajtd	September 23rd, 2009, 12:22 p.m.	4 weeks, 1 day ago	
<div>Summary</div>	Submitter	Posted	Last Updated	...																											
☆ Disabling of rv in eptrecv	kvs	October 21st, 2009, 11:42 a.m.	1 day, 14 hours ago																												
☆ Disabling of rv in eptrecv_pf	kvs	October 21st, 2009, 11:41 a.m.	1 day, 14 hours ago																												
☆ [bug 3083068] getSpaceidFromService not using proxy/called multiple times.	apaz	October 8th, 2009, 11:41 a.m.	1 week, 6 days ago																												
☆ yoda_finance_data_component review	kvs	October 5th, 2009, 1:44 p.m.	2 weeks, 2 days ago																												
☆ Rewrote Lassie in a more c++ style	ajtd	September 23rd, 2009, 12:22 p.m.	4 weeks, 1 day ago																												

소셜 소스 코드 공유를 목적으로 하여 제공되는 깃헙(github) 협업 개발 서비스를 통하여 코드 리뷰를 웹 기반에서 하는 것도 가능하다. 개발자 서로에게 투명하게 코드를 오픈하고 동료 간의 리뷰를 통하여 원활하게 피드백을 받을 수 있어 개발자의 실력 향상과 동기부여에도 큰 도움이 된다.


## <그림 5> Github(소셜코딩 관리도구) 예제



SOCIAL CODE HOSTING



**schacon**
[account](#) | [profile](#) | [guides](#) | [log out](#)  
 103
 repositories: [all](#) | [search](#)


**schacon**







GitHub Role: Gizzard  
 Name: Scott Chacon [edit](#)  
 Email: [schacon@gmail.com](mailto:schacon@gmail.com) [edit](#)  
 Blog: <http://jointheconversation.org> [edit](#)  
 Company: Logical Awesome [edit](#)  
 Location: Redwood City, CA [edit](#)  
 Followers: 123  
 Public Repos: 39  
 Member Since: Jan 27, 2008

**asgit**  
 ActionScript Git library and browser implementation  
 Created May 27, 2008  


**dotgit**  
 A Git implementation for .NET written in C#  
 Forked from [pheew/dotgit](#) December 15, 2008  


**Following**  


**Public Activity**

-  **schacon pushed to master at [schacon/schacon.github.com](#) 39 minutes ago**  
 HEAD is [5692b7b69b56cff9ea3c9bbd879644dc33a01260](#)  
 **schacon committed [5692b7b6](#):**  
 added github ribbon
-  **schacon committed to [schacon/schacon.github.com](#) 39 minutes ago**  
[5692b7b69b56cff9ea3c9bbd879644dc33a01260](#)  
 added github ribbon
-  **schacon pushed to master at [schacon/schacon.github.com](#) 4 days ago**  
 HEAD is [c64f4f2ea8127d5b08a2ebffa1a147c6ac7f9b2f](#)  
 **schacon committed [c64f4f2e](#):**  
 added a link to my second page
-  **schacon committed to [schacon/schacon.github.com](#) 4 days ago**  
[c64f4f2ea8127d5b08a2ebffa1a147c6ac7f9b2f](#)  
 added a link to my second page

### <참고 자료>

1. Fagan, M. "Advances in Software Inspections", IEEE Transaction on SE, July, 1986
2. Best Kept Secrets of Peer Code Review, <http://goo.gl/auK6f>
3. Review board, <http://www.reviewboard.org/>
4. 사람을 위한 자동화, 이클립스 플러그인으로 코드 품질 높이기, IBM DeveloperWorks
5. 소프트웨어 프로세스 이야기, <http://swprocess.egloos.com>
6. 인스펙션 강의 자료(핸디소프트, 황순삼)