



4. Bug Tracking with Bugzilla

왜 버그 트래킹이 필요한가?

- **제품의 문제점을 발견하고 해결하기 위한 과정**
 - 문제 발견 / 증상 규격화 / 원인 규명 / 재구현 / 문제 해결
- **가장 적합한 커뮤니케이션 방식 부재**
 - e-mail: 모두 참여 어려움. 변경 과정을 추적하기 힘들
 - Phone, IM: 모두 참여 어려움. 금방 잊어 먹음
 - Wiki: 공지나 변경 사항 알림 힘들
- **버그 처리를 위한 새로운 도구가 필요**
 - 버그에 대한 변경과정을 추적할 수 있어야 함



버그 트래킹 도구

Tool	Lang	Ver	Lic	Rev Date	Cust	Temp	Search	RSS	Not	Rep	Hist	Attach
Mantis	PHP	1.0.6	GPL	1/27/2007	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Request Tracker	Perl	3.6.3		1/27/2007	Yes	Yes	Yes	Yes	Yes		Yes	Yes
Scarab	Java	1.0b20	Tigris	1/27/2007	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Bugzilla	Perl	2.23.3	CC	1/27/2007	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Eventum	PHP	1.7.1	GPL	1/27/2007	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
PloneCollector-NG	Python	1.2.9	LGPL	1/27/2007	Yes		Yes	Yes	Yes	Yes	Yes	Yes
BugTracker.NET	C#	2.2.4	GPL	1/27/2007	Yes	No	Yes	No	Yes	Yes	Yes	Yes
ITracker	Java	2.4.2	LGPL	1/27/2007	Yes		Yes		Yes	Yes	Yes	Yes
OTRS	Perl	2.1.4	GPL	1/27/2007		Yes	Yes		Yes		Yes	Yes
Roundup	Python	1.3.2	Python	1/27/2007	Yes	Yes	Yes		Yes		Yes	Yes
Trac	Python	0.10.3	BSD	1/27/2007			Yes	Yes		Yes	Yes	Yes

출처: <http://www.michaelflanakin.com/Articles/Comparisons/WebBasedIssueTrackers/tabid/198/Default.aspx>



버그 트래킹 도구 기능

■ 버그 처리 기능

- 다양한 항목에 대한 버그 제출/검색 기능
- 패치(Patch), 첨부 파일 제공 기능
- 변경 이력 및 이에 대한 메일링 기능
 - Bugzilla

■ 이슈 처리 기능

- 단지 버그 뿐만 아니라 기능상 개선 사항 포함
- 마일스톤에 관련된 이슈에 대한 처리도 가능
 - Mantis, Trac, Issue Tracker



버그질라 (Bugzilla)



- Mozilla 프로젝트에서 개발한 오픈 소스 버그 트래킹 시스템
 - <http://www.bugzilla.org>
- 고급 리포팅 기능 탑재
- 웹 기반 인터페이스
- 주요 사용 프로젝트
 - GNOME bugs: <http://bugzilla.gnome.org>
 - mozilla bugs: <http://bugzilla.mozilla.org>
 - StarOffice bugs: <http://www.openoffice.org/issues>



단계 1. 버그 질라 가입

- 계정 만들기
 - 주로 이메일 계정을 이용한다.
 - 버그에 대한 모든 기록이 전달 되므로 버그 처리용 이메일을 별도로 사용하는 것이 좋다. (Gmail은 트래킹 용이)
 - 가입 후 이메일로 비밀번호 전달 됨.
- 관리자로부터 권한 얻기
 - 대개의 경우 권한이 필요 없는 경우가 많음.
- 제출할 버그 검색 및 조회 하기
 - (없으면) 버그 올리기

단계 2. 버그 검색 하기

- 버그를 보고 하기 전에 꼭 검색을 한다.
 - 자신과 같은 버그가 있는지 검색을 한 후 이것이 현재 해결되었는지 (해결 중인지) 여부를 확인 한다.
 - 해결 된 경우, 다음 릴리스 까지 대기중일 수 있다.
- 검색에 넣을 항목을 적게 한다.
 - 중요한 키워드만 입력하여 되도록 많은 버그를 살펴 본다.
- 다양한 버그 항목들이 있는 긴 목록을 사용한다.
 - 버그 보고를 할 때 사람들에 따라 다르게 인식하기 때문에 설명이 달라진다.

단계 3. 신규 버그 제출

- 1) 버그를 보고할 product 를 고른다
- 2) 한 줄 정도 summary description 를 제공 (키워드 포함)
- 3) 버그를 재현 하는 details 기입:
 - 버그를 재현할 수 있는 단계별 설명 제공
 - 버그로 인해 나타날 수 있는 문제 설명
 - 버그가 아닐 경우 나타나야 하는 정상 동작 설명
- 4) component 항목 선정
- 5) severity와 priority 단계 설정
- 6) 해당 Version 및 OS 선정

버그 제출 방법 (1)

Bugzilla – Enter Bug: _test product

[Home](#) | [New](#) | [Search](#) | | [Reports](#) | [My Requests](#) | [My Votes](#) | [Preferences](#) | [Log out](#) channy@gmail.com

Before reporting a bug, please read the [bug writing guidelines](#), please look at the list of [most frequently reported bugs](#), and please [search](#) for the bug.

Reporter: channy@gmail.com

Version:

Product: test product

Component:

Severity: normal
Priority: P2
Target Milestone: ---

Platform: PC
OS: Windows XP

Initial State: UNCONFIRMED

Assign To:
QA Contact:
Cc:

Flags:
another-flag ()
another-flag2 ()
blocker

Requestee: ()



버그 제출 방법 (2)

URL:

Summary:

Description:

Attachment:

Keywords: (optional)

Depends on:

Blocks:

버그 제출 방법 (3)

- Summary(제목)에는 중요 키워드를 포함한 정보 제공
 - Good: "Cancelling a File Copy dialog crashes File Manager"
 - Bad: "Software crashes"
 - Bad: "Browser should work with my web site"
- Description(상세 정보) 제공
 - 버그가 어떻게 생기는 지 상황 재현 설명
 - 한 가지 문제에 하나의 버그만 제출
 - 프로그램을 종료 시킬 정도의 버그라면 Stack Trace를 첨부

버그 제출 방법 (4)

- **Description 항목에 기술할 내용**
 - Overview
 - 제목에 대한 좀 더 상세한 설명
 - Steps to Reproduce
 - 버그를 재현할 수 있는 최소한의 단계별 설명. 가급적 1), 2), 3) 으
로 번호 넘버링
 - Actual Results
 - 과정을 거친 후 실제로 동작하는 문제점 기술
 - Expected Results
 - 버그가 없었다면 동작해야 하는 기능 기술
 - Build Date & Platform
 - 이 버그를 만난 제품의 빌드 일시와 운영 체제
 - Additional Builds and Platforms
 - 다른 운영 체제나 빌드에서 일어 나지 않은 경우에 기술



버그 제출 상세 기능

- Attach (첨부 기능)
- Severity 단계 지정
- Priority 단계 지정
- Keyword 설정

Attach (첨부)

- 버그를 설명하기 위한 ‘스크린샷’이나 ‘Testcase’등을 제공할 때 사용할 수 있음
 - 원래는 ‘Patch’를 제공하기 위해 주로 사용됨
- ‘Create a New Attachment’를 통해 원하는 항목을 기입
 - Path: 자신의 컴퓨터에서 올릴 정보 제공
 - Content-type: 텍스트, 이미지, 동영상 등
 - Description: 첨부 파일에 대한 설명
- **Obsolete**
 - 새로운 첨부 파일이 이전 첨부 파일 보다 갱신된 것인 경우 체크 (이전 항목 strike 표시됨)

yahoo-kr.xml patch to fix search suggestion url and parameter. (504 bytes, patch) 2006-09-10 18:29 PDT, Channy Yun	<i>no flags</i>	Details Diff
yahoo-kr.xml patch to fix search suggestion url and parameter. (504 bytes, patch) 2006-09-10 18:29 PDT, Channy Yun	l10n: review+ l10n: approval-l10n+	Details Diff
Add an attachment (proposed patch, testcase, etc.)		Hide Obsolete (1) View All

Severity 단계 설정

Severity – 사용자에게 미치는 강도에 따라 구분

1. *blocker*: 개발을 중지하고 테스트 필요
2. *critical*: 데이터 분실이 일어날 정도로 심각한 경우
3. *major*: 기능 장애, 원하는 기능이 동작하지 않는 경우
4. *normal*: 단위 기능에 장애가 있는 경우
5. *minor*: 기능 상 장애가 미미한 경우
6. *trivial*: UI나 번역상 문제
7. *enhancement*: 새로운 기능이나 향상 요청

Priority 단계 설정

- **Priority : 버그를 처리 해야 하는 우선 순위**
 1. *immediate*: 즉시 바로 수정해야 하는 버그
 2. *urgent*: 다음 마이너 릴리스 전에 수정해야 하는 버그
 3. *high*: 다음 주요 릴리스 전에 수정해야 하는 버그
 4. *normal*: 일반적인 토론을 거쳐 수정할 버그
 5. *low* : 수정 되면 좋은 버그
- **참고**
 - Major Release: 주기적인 기능 상 제품 출시
 - Minor Release: 보안이나 기능 장애로 인한 제품 출시

키워드(Keyword) 설정

- **버그 묶음을 만들거나 현재 상황을 파악하는 데 필요한 항목**
- **처리 중이거나 처리할 버그들을 묶는 세트**
 - fixed1.0 : 처리한 버그들
 - verified1.0 : 처리 후 빌드 후 기능 확인한 버그들
- **특정 항목을 묶는 세트**
 - intl: 국제화 관련 버그
 - js1.5: 자바 스크립트 구현에 관련된 버그
- **사용 용도에 따른 세트**
 - testcase: 테스트를 위한 버그 목록
 - patch: 패치가 들어 있는 버그 목록

기타 기능

■ 투표(Vote) 기능

- 현재 버그에 대해 수정 요청을 하는 경우
- 투표수가 많을 수록 중요 버그로 갈 가능성이 높음

■ 참조자(Cc) 기능

- 현재 버그에 관여되어 있는 사람을 추가하는 기능
- 버그 변경 사항이 메일로 전달

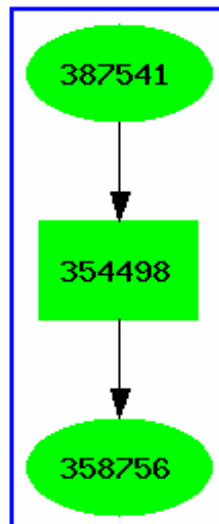
■ 버그 트리 (Depend on, Block) 기능

- 특정 버그 산하에 트리 형태로 버그를 묶는 기능
- 처리해야 할 버그들의 진행 사항을 파악 할 수 있음

Bugzilla@Mozilla – Dependency Graph for bug

[Home](#) | [New](#) | [Search](#) | |

Green circles represent open bugs.



Bug numbers:

Show the summaries of all displayed bugs

Orient top-to-bottom

Bugzilla@Mozilla – Dependency tree for Bug 348096





[Home](#) | [New](#) | [Search](#) | | [Reports](#) | [My Requests](#) | [My Vo](#)

Max Depth:

Bug 348096 does not depend on any bugs.

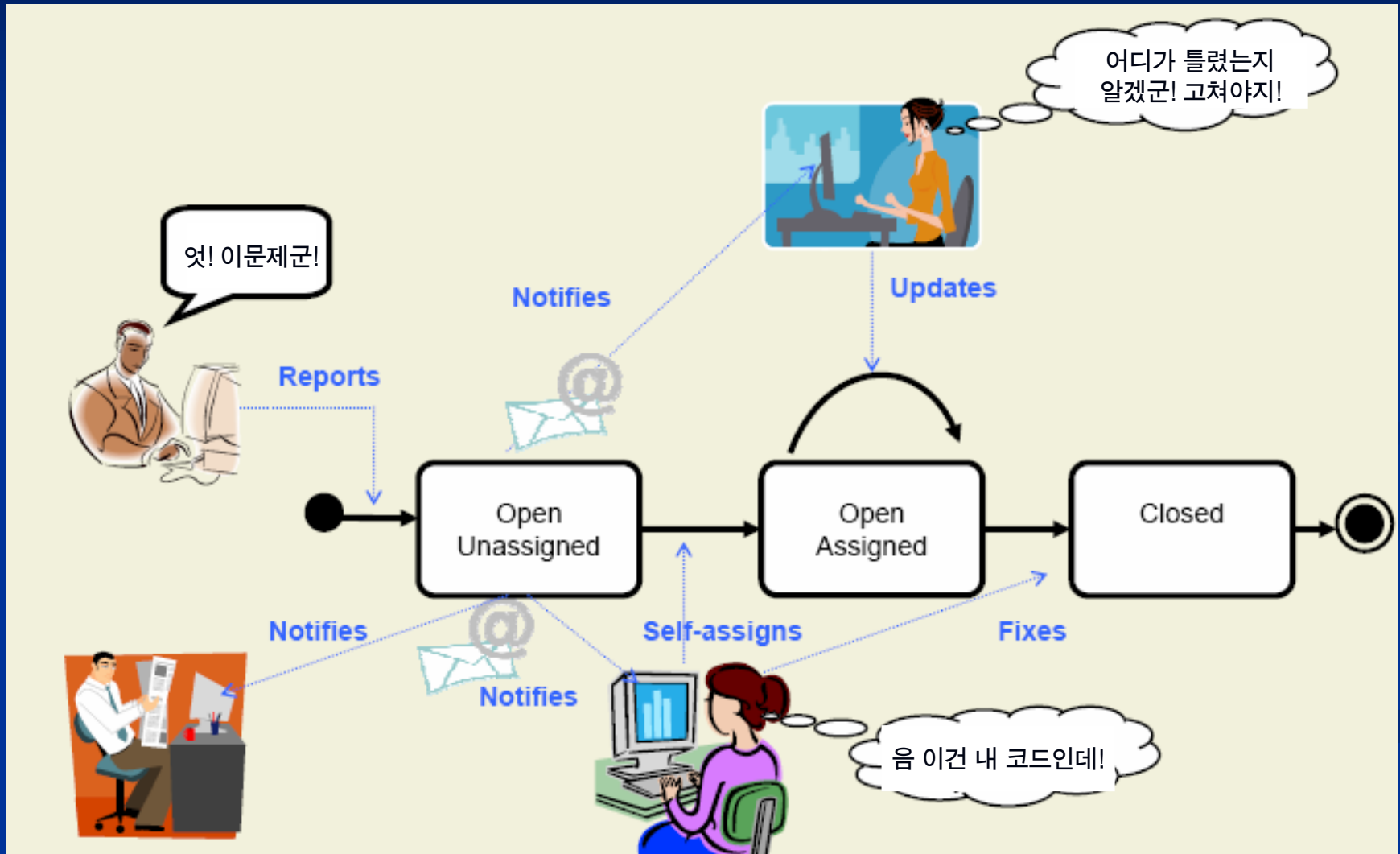
Bug 348096 blocks 3 bugs:

([view as bug list](#) | [change several](#))

-  **348096:** ru Search Feature Tracking for Firefox 2 
-  **347914:** Search Feature Tracking for each locale shipped w/ Fire
 -  **350051:** [ru] Ship Firefox 2 
 -  **350031:** Ship localized Firefox 2 

Max Depth:

버그 처리 과정



상태 관리

- 버그의 처리 과정은 “Status”를 통해 확인 가능
 - UNCONFIRMED: 기본 설정. 아직 확인하지 않은 상태
 - NEW: 버그를 확인 하고 담당자에게 지정하기 위한 단계
 - ASSIGNED: 담당자가 지정된 상태
 - NEEDINFO: 버그에 대한 더 자세한 정보가 필요한 경우
 - FIXED(RESOLVED): 버그가 고쳐진 경우
 - VERIFIED(CLOSED): 버그 수정이 실제 제품에도 반영된 경우
 - REOPEN: 버그가 다시 발생해서 수정이 필요한 경우
- 일반적인 처리 과정
 - NEW-> ASSIGNED -> (Patch) -> FIXED

기타 처리 과정

- **중복 버그인 경우**
 - Duplicated 상태로 변경 한 후, 중복 버그 번호 표시
- **버그가 아닌 경우**
 - Invalid 상태로 변경 한 후 버그 종료
- **고칠 필요가 없는 버그 인 경우**
 - WONTFIX 상태로 변경 한 후 버그 종료
 - 대개 정치적, 기술적 논쟁의 의해 프로젝트 내에서 결정된 사안의 경우



패치 생성 및 처리

■ 패치 생성 방법

- 패치는 Workspace와 Repository 사이의 변경 사항을 diff 해서 만든다.
- SVN 패치 생성 방법
 - `$ svn diff > filename.patch`
- CVS 패치 생성 방법
 - `$ cvs diff > filename.patch`

■ 패치 처리 과정

- 생성된 패치를 Attach를 통해 추가.
- 의사 결정 과정(Flags)을 거쳐 패치 처리 여부 확인.
- 커밋 시 반드시 버그 번호와 의사 결정자 정보를 제공
 - `$cvs commit -m "Bug 99 Hacking Bugs by Channy, a=channy, r=superman"`



패치 형식

서버 상의 파일 위치 및
최종 리비전 번호

Index: SearchDialog.dtd

=====
RCS file: /I10n/I10n/ko/mail/chrome/messenger/SearchDialog.dtd,v
retrieving revision 1.1.2.10

diff -u -r1.1.2.10 SearchDialog.dtd

--- SearchDialog.dtd 28 Jul 2006 03:28:53 -0000 1.1.2.10

+++ SearchDialog.dtd 13 Jul 2007 00:58:51 -0000

@@ -2,15 +2,15 @@

<!ENTITY searchHeading.accesskey "h">

<!ENTITY searchSubfolders.label "하위 폴더 검색">

<!ENTITY searchSubfolders.accesskey "e">

-<!ENTITY resetButton.label "삭제">

+<!ENTITY resetButton.label "초기화">

<!ENTITY resetButton.accesskey "C">

<!ENTITY optionsButton.label "설정">

<!ENTITY closeButton.label "닫기">

<!ENTITY openButton.label "열기">

파일 내에서 수정되어야
할 부분



플래그(Flags)

주의: 아무나 플래그를 사용하면 안됨.

■ 플래그란?

- 버그에 대한 중요한 의사 결정을 수행
- 의사 결정은 주로 상위 개발자들이나 프로젝트 리더들이 수행

■ 플래그의 종류

- Tree Blocking: 현재 버그가 특정 소스 트리에서 처리 해야하는 지 여부 (ex. blocking 1.1b)
- Attach Blocking: 현재 첨부(Patch)를 특정 소스 트리에서 수정 코드로 사용하는 지 여부(approval1.3b)

■ 플래그 사용 방법

- 의사 결정 요청: ?
- 허용: +
- 허용 안 함: -



Lab: 버그 질라 실습

- 테스트 버그질라 이용
- <http://landfill.bugzilla.org>
 - 1. 아이디 가입 및 등록
 - 2. 신규 버그 제출
 - 3. Assign 및 버그 처리
 - 4. 패치 제출 및 Review
 - 5. 버그 완료



각종 데모 사이트

■ Mantis

- http://www.futureware.biz/mantisdemo/my_view_page.php

■ Trac

- <http://www.hosted-projects.com/trac/TracDemo/Demo>
- 태터툴즈, 제로보드 XE에서 사용 중