

Software Design and Construction  
159.251

# **Software Configuration Management**

Amjed Tahir  
a.tahir@massey.ac.nz

# Daily issues in software development...

- Common scenarios:
  - Oops, I've just found a bug, should I **report** it or just **ignore** it?
    - **Reporting**: I've never seen something similar to this before, what **bug type** is this? And where is it coming from? Is it the code I wrote or is it coming from someone's else code?
  - This bug has been fixed before, why it has **reappeared** again?

# Have you heard of these stories before?

- I cannot find the latest version of this program or document?
- *A disaster*: the latest version of the code was **overwritten** by an old version!!
  - I've lost all my latest changes – two days of work!!!
- Nobody knows **which version** of the program is final.



# Have you heard of theses stories before?

- Design document is **out of sync** with programs.
- I don't know if **all the changes** that were suggested have been **incorporated**

# What's needed?

- .....a mechanism to handle these issues.
- Change is inevitable when software systems are built.
  - changes is a daily routine.
- Bugs are part of the development cycle- you first find (***detect***) them , and then try to ***fix*** them.
  - And it would be even better if you can ***predict*** them before they happen!
- Not only bugs, all ***issues*** should be managed and tracked.

# What is *Software Configuration Management (SCM)*?

- Process of **identifying and linking** the components that make up the product
  - Examples: source code files, database files...etc
- *Configuration Management help in*
  - **Controlling changes** throughout the development life cycle.
  - **Maintain multiple versions** of single application that is developed by multiple people/teams.
  - **Can roll back** to previous versions, when necessary.
  - **Maintain links (relationships) between various items** (i.e. notify when a change has been made to a dependent item).

“Software configuration management is the art of identifying, organizing, and controlling modifications to the software being built by a programming team. The goal is to maximize productivity by minimizing mistakes”

Babich, W.A. (1986). Software Configuration Management, Coordination for Team Productivity. 1st edition. Boston: Addison-Wesley

# Misconception about *SCM*

- *SCM* is not only version control
- *SCM* is not only source code management
- *SCM is not* only for the coding stage of development



# Configuration Management Process

- Identification of configuration items
  - Written code & derived (using external libraries), design doc, manual, ...
- Set up the environment (i.e., tools)
- Record all changes and requests in a CM database.
- Change control policy
  - formally defined policy – e.g., who is allowed to change what?
- Status accounting
  - management checks progress

# Typical Software Configuration Items

- Requirement specifications
- Design specifications
- Source code
- Test cases, test data, and recorded results
- User guides and installation manuals
- Executable programs
- Standards and procedures
  - (e.g. Java design guidelines)

# Three important concepts in SCM

- Version:
  - A well-defined state of a configuration item at a given point of time.
- Revision:
  - *Change* to a version that corrects only errors in the design/code, but does not affect the documented functionality.
- Release:
  - The *formal distribution* to users of an approved version.

# Configuration Management Process

- Identify a tool for SCM
- Identify configurable software *items*
- Identify *baselines*
- Define and assign roles
- Define change criteria
  - Consider Issue Based Information System (IBIS) technique
- Define release criteria

# Change Management

- Change management: handling of change requests.
  - A change request leads to the creation of a new release.
- General change process
  1. The change is ***requested*** (this can be done by anyone, including users and developers).
  2. The change request is ***assessed*** against project goals.
  3. Following assessment, the change is *accepted* or *rejected*.
  4. If accepted, the change is assigned to a developer and implemented.
  5. The implemented change is audited.
- Complexity of change management process varies with size of project.
  - Small projects can perform change requests informally and fast, while complex projects require detailed change request forms and the official approval of one or more managers.

# Version Control

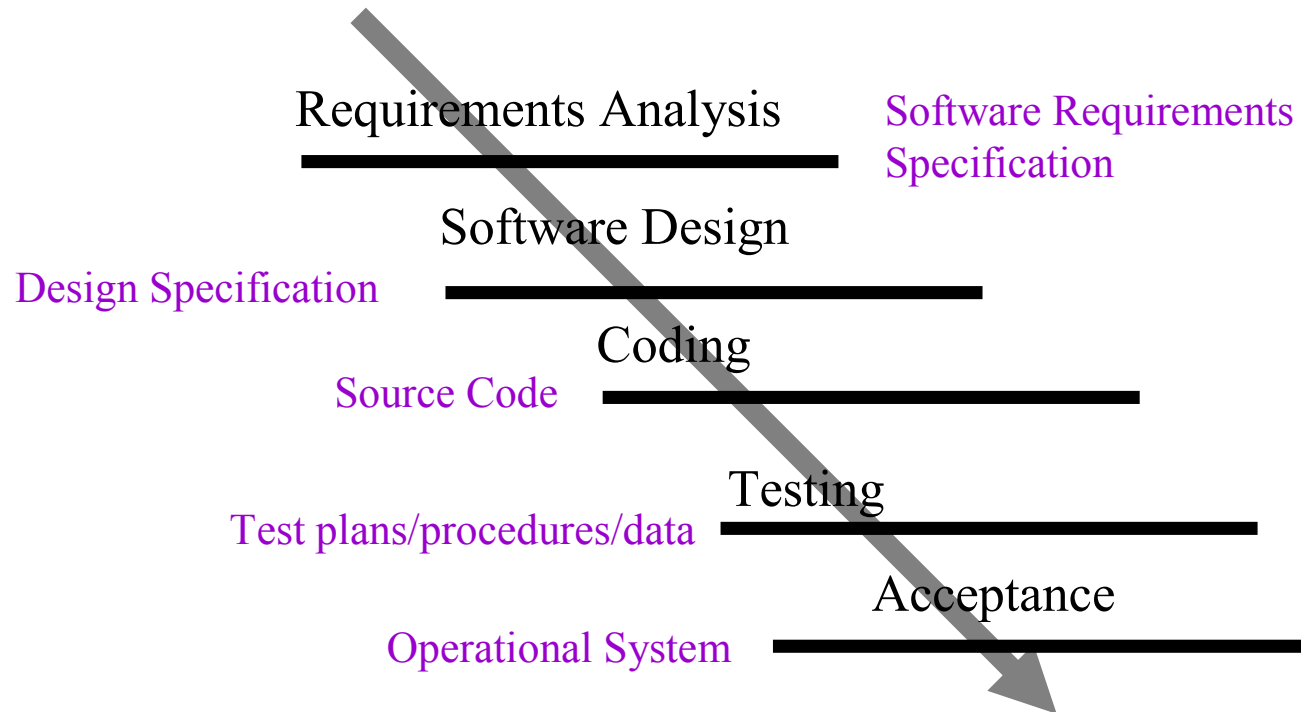
- Version control combines procedures and tools to manage different versions of the software, while production.
- A **version control system** records changes to a file or set of files over time so that you can recall specific **milestone** later.

See the previous Lecture on VC!

# Baseline

- A baseline is a ***snapshot*** of the system that has been formally reviewed and agreed upon (approved).
- It is an approved revision of a document or source code file from which subsequent changes can be made.
- It serves as the basis for further development and be changed only through a formal change control procedure.
- In many cases, a *Baseline* represents the most *stable version* of the software.

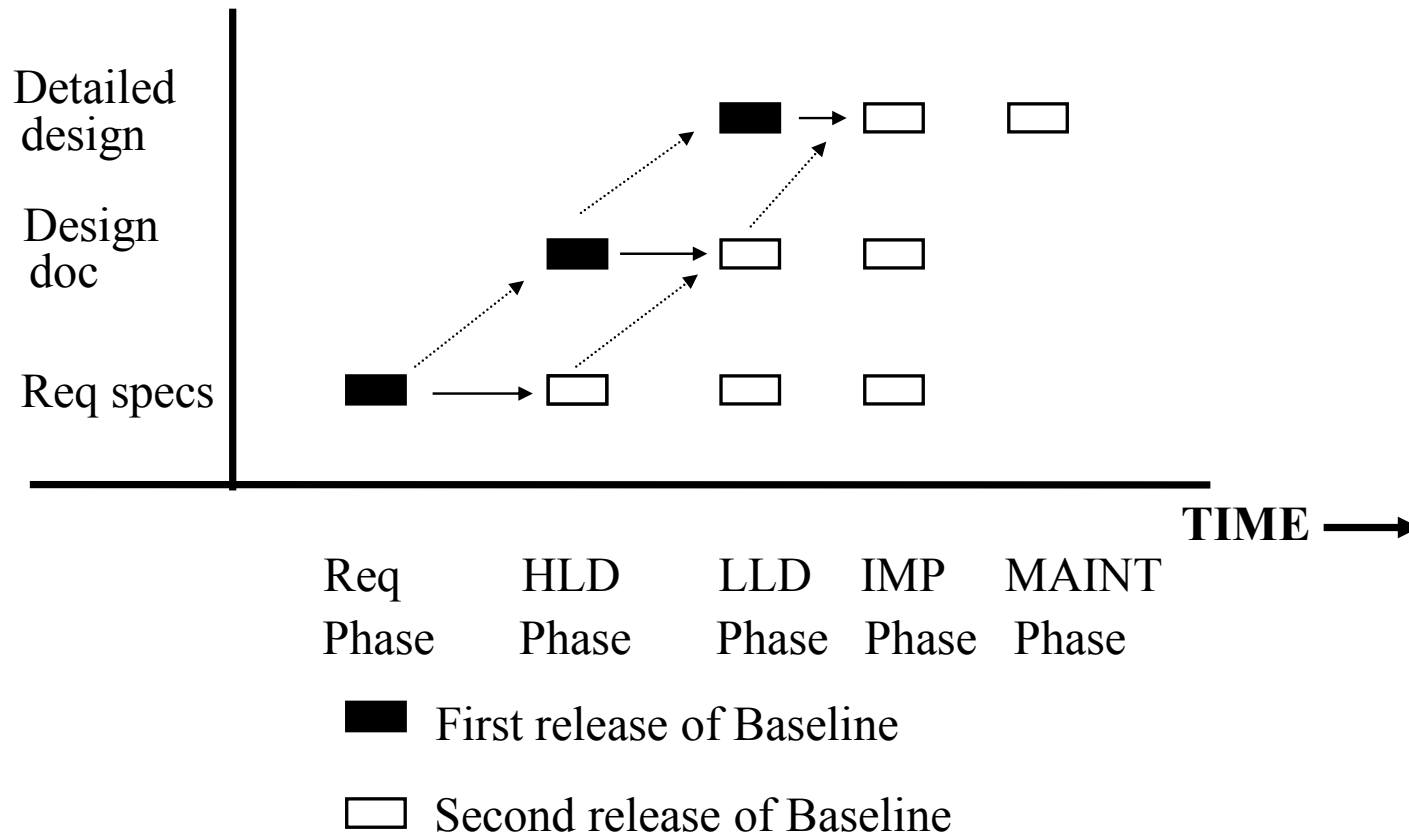
# Baselines





# Baselines in an iterative process

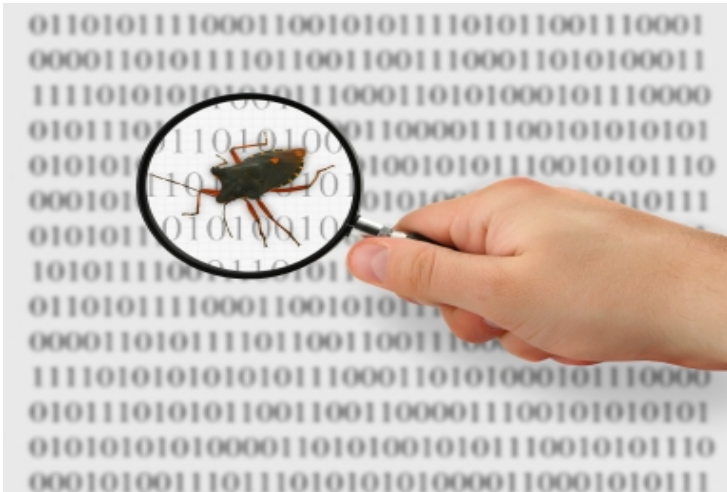
## BASE LINES



# Change control

- **Change control** is vital in any software development.
- One of the most effective ways to control changes is through ***change requests***.
- A ***change request*** is submitted and evaluated to assess technical merit, potential side effects, overall impact on other configuration objects and system functions, and the projected cost of the change.

# Issues and Bugs Tracking and Control



- Similar to *Change Management*, **issues** and **bugs** should be managed.
- Otherwise,
  - they will be reported multiple times
  - they will be reported in the wrong places
  - And they will be reported but will not be fixed!!
- All **issues** and **bugs** should be reported, reviewed and fixed (if approved).
- Different types, *similar* structure:
  - Also known as issue tracker, bugs tracker or faults management system.

# Bugs Management

- What's a software bug?
  - A software bug is an error, failure or fault in a software program that causes it to produce an ***incorrect*** or ***unexpected*** result, or to ***behave in unintended ways***.
- Bugs are reported by either *developers* or *end-users*.
- A ***bug report*** provides details of reported bugs in a program.

# Types of Bugs

- Some bugs types
  - Logic bugs
    - Infinite loop
  - Syntax bugs
    - E.g., using `x=y` instead of `x==y`
  - Resource bugs
    - ***Stack*** or ***buffer*** overflow.
    - Memory leaks
  - Performance bugs
    - Processor not able to handle computational tasks

**Read:** a list of well-know software bugs:

[https://en.wikipedia.org/wiki/List\\_of\\_software\\_bugs](https://en.wikipedia.org/wiki/List_of_software_bugs)

# Examples

## Well-known bugs

- Y2K bug – time tracking
  - A bug that have caused problems when dealing with dates beyond December 31, 1999

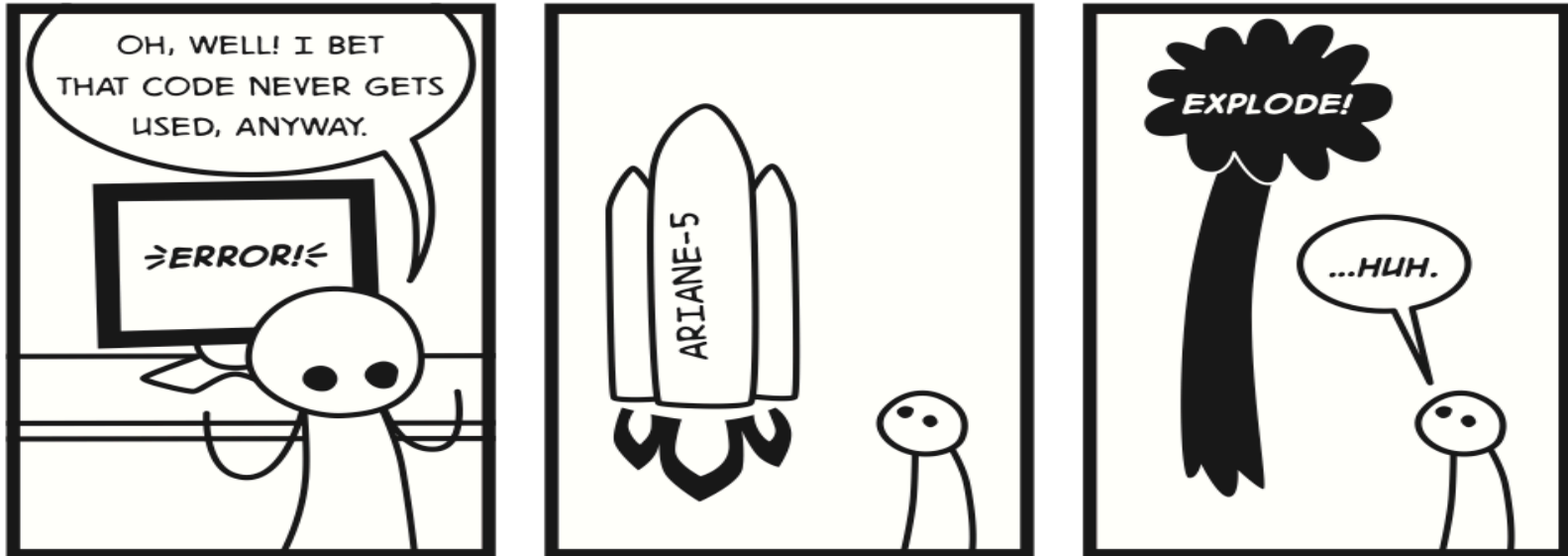


An electronic sign displaying the year incorrectly as 1900 on 3 January 2000 in France - [Wikipedia](#)

- Therac-25 - Radiation therapy
  - A software bug that resulted in a massive overdoses of radiation that killed and injured several people.



# Ariane 5 - the most expensive bug in history!

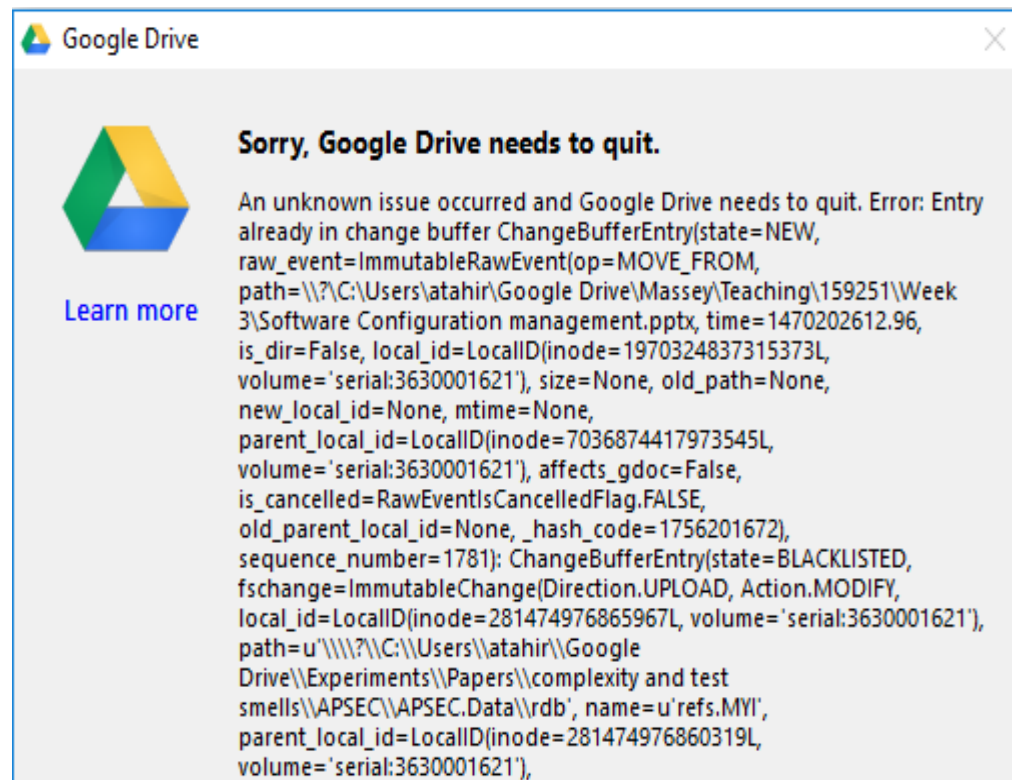


- A bug caused a data conversion from a 64-bit floating point number to a 16-bit signed integer value to overflow and cause a hardware exception
- Cost over US\$370 million!



# And while I was preparing this material....

## This happened!



# Bug Report

- A ***bug report*** is a document that include the list of issues and bugs in a software system.
- Most widely used as a software package.
- Details can be part of a bug tracker
- A typical issues/bugs tracker will provide details of the reported issue such as the title, name of the person reported the issue,

# Issues tracker

What can you report?

- Many issues can be tracked:
  - **Bugs:** error or failure in the software.
  - **Improvement:** a request or suggestion...
  - **Wish:** of a developer or user.
  - **Test:** test of an issue.

This usually done by developers or users.

# Issues Management in Software Development

- A developer should report all issues and bugs found.
- Several vital information should be included:
  - Automatically generated:
    - Bugs ID, name of developer, time, version number, repository number etc...
  - Manually entered info:
    - Type of bug, impact (if known), circumstances etc...
  - ***Bugs are then categorised based on:***
    - ***Class - Severity - Priority***

# Issues tracker

## What can you report?

Cont'd.

- You can also mark the statuses of the reported issue:
  - **Open:** has not been addressed yet.
  - **In progress:** currently under development.
  - **Resolved:** the issue has been resolved, but still open for discussion.
  - **Closed:** it has been resolved and discussion has been concluded.
  - **Reopened:** especially when an issue was marked as closed or resolved, but someone noticed that the issue still exist (e.g., bug reappeared).
- These options are monitored by a senior developer, team leader or quality assurance personal.

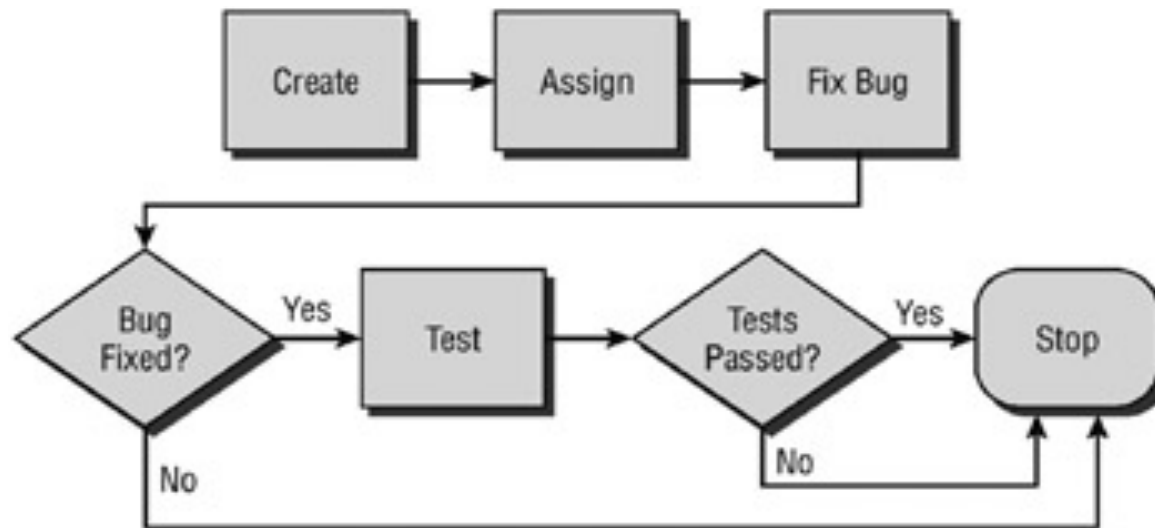
# Bug Management in Software Development

- A developer should report all issues and bugs found.
- Several vital information should be included:
  - Automatically generated:
    - Generated number, name of developer, time, version number, repository number etc...
  - Manually entered info:
    - Type of bug, impact (if known), circumstances etc...
  - ***Bugs are then categorised:***
    - ***Class - Severity - Priority***

# After they are reported....

- Once reported, bugs get reviewed by team leaders, senior developers or quality assurance team.
- Each issue will be evaluated, and it's impact will be rated.
  - Some bugs will be prioritized based on their immediate impact or the severity of their impact (minor or major).
    - Some tracking systems use a scale of 1-5 to rate bugs/issues.
- Then someone (or a team, depending on the bug) will be assigned to work on fixing the bug.

# Bugs fixing workflow





# Bug Tracking Systems

- A ***bug tracking system*** is a software application that keeps track of reported software bugs.
  - Usually comes as a part of the issue tracking system
- Allows developers to report and locate bugs that they need to resolve.
- Some software project management tools provide this as one of their functionalities.
- Several version control systems also provide bug tracking system.
- **Example of bug tracking system:** Bugzilla and JIRA

# Bugzilla

- Open source bugs tracker.
- Originally designed by Netscape in 1998.
- Some of the open-source projects that uses bugzilla:
  - Mozilla.org projects,
  - Linux kernel,
  - Eclipse,
  - FreeBSD,
  - Apache (some projects!),
  - Red had,
  - LiborOffice

The “zarro Boogs found” expression!

# To operate Bugzilla...

- To use Bugzilla for your project, you'll need the following:
  - A compatible database management system
    - Such as MySQL, SQLite, Oracle or PostgreSQL,
  - A compatible web server
  - A suitable release of Perl 5
  - A suitable mail transfer agent, or any SMTP server

# Bugs Tracking System

## Example 1- Mozilla Firefox

- Reported in [Bugzilla](#)

Bugzilla@Mozilla New Account | Log In | Forgot Password **mozilla**

Home New Browse Search  Search [help] Reports Product Dashboard

bugzilla.mozilla.org will be unavailable because of maintenance work on Saturday May 14th, 2016, starting at 7:00AM PDT (2:00PM UTC). Work should be complete around three hours later.

Fri May 13 2016 20:09:47 PDT

Component: Document Navigation Product: Core Resolution: ---

This result was limited to 500 bugs. [See all search results for this query.](#)

ID	Product	Comp	Assignee	Status	Resolution	Summary ▼	Changed
435323	Core	Document Navigation	nobody	NEW	---	"Address Not Found" and similar pages should have a "Work Offline" button next to "Try Again"	2009-11-21
822085	Core	Document Navigation	nobody	NEW	---	"Assertion failure: failedURI (We don't have a URI for history APIs.)" with onunload navigation to error page	2012-12-17
746880	Core	Document Navigation	nobody	NEW	---	"Assertion failure: GetReadyStateEnum() == nsIDocument::READYSTATE_COMPLETE (Bad readyState)"	2016-03-21
1000351	Core	Document Navigation	nobody	NEW	---	"Assertion failure: mBlockDOMContentLoaded" with XMLDocument.load	2015-01-06
443655	Core	Document Navigation	nobody	NEW	---	"ASSERTION: A frame but no DOM element!?" removing two frames, one of which has an onunload that navigates the other	2015-10-23
645229	Core	Document Navigation	bugs	NEW	---	"ASSERTION: Adding a child where we already have a child? This may misbehave"	2015-01-31
958976	Core	Document Navigation	nobody	NEW	---	"ASSERTION: Adding a child where we already have a child? This may misbehave" when searching on Youtube.com	2015-04-27
682644	Core	Document Navigation	nobody	NEW	---	"ASSERTION: aIndex is out of range" again	2011-08-28
610952	Core	Document Navigation	nobody	NEW	---	"ASSERTION: aToIndex is out of range" in nsSHistory::EvictWindowContentViewers	2010-11-10
866653	Core	Document Navigation	nobody	NEW	---	"ASSERTION: ContentViewer Initialization failed" after thousands of window.Notification calls	2013-04-29
621375	Core	Document Navigation	nobody	NEW	---	"ASSERTION: Entry added to loadgroup twice, don't do that"	2012-01-06

# JIRA

- JIRA issue tracker is part of the JIRA software
- JIRA is an industrial-based tools from Atlassian, Inc.
- Integrates well with other JIRA software such as Bitbucket.
- Some of the projects/organisations that uses JIRA:
  - **Open-source:** JBoss, Fedora Commons, Spring Framework and several Apache projects (such as *Commons* libraries)
  - **Industry:** NASA JPL and Audi...

# Bugs Tracking System

## Example 2- Apache Hadoop

- Reported in [JIRA](#)

The screenshot displays the Apache JIRA web interface. At the top, the Apache Software Foundation logo is on the left, and navigation links for Dashboards, Projects, Issues, and Agile are in the center. A search bar and a 'Log In' link are on the right. The left sidebar contains a 'FILTERS' section with a 'New filter' button and a list of filters: 'Find filters', 'My Open Issues', 'Reported by Me', 'Recently Viewed', and 'All Issues'. Below this is a 'FAVOURITE FILTERS' section with a message: 'You must be logged in to view favourite filters.' The main content area shows a search results page for 'Hadoop HDFS'. The search criteria include 'Type: All', 'Status: All', 'Assignee: All', and 'Contains text'. The results are ordered by 'Due Date'. The first result is 'HDFS-2264' with the title 'NamenodeProtocol has the wrong value for clientPrincipal in KerberosInfo annotation'. The bug is marked as 'CLOSED' and 'Fixed'. The details section shows the bug type as 'Bug', priority as 'Major', and affects version as '2.0.2-alpha'. The component is 'namenode' and the target version is '2.0.3-alpha'. The assignee and reporter are both 'Aaron T. Myers'. The bug was reported on '16/09/17'.

APACHE SOFTWARE FOUNDATION  
http://www.apache.org/

Dashboards ▾ Projects ▾ Issues ▾ Agile ▾

Search [?] Log In

FILTERS <<

New filter

Find filters

My Open Issues

Reported by Me

Recently Viewed

All Issues

FAVOURITE FILTERS

You must be logged in to view favourite filters.

16/09/17

Search Save as

Export ▾ Tools ▾

Hadoop HDFS ▾ Type: All ▾ Status: All ▾ Assignee: All ▾ Contains text More ▾ Q Advanced

Resolution: Unresolved, Fixe... ▾ × Fix Version: 1.3.0 ▾ ×

Order by Due Date ▴ ▾

HDFS-2264  
NamenodeProtocol has the w...

HDFS-4779  
Backport HDFS-2576 to branc...

HDFS-4794  
Browsing filesystem via webui...

HDFS-4944  
WebHDFS cannot create a file...

HDFS-4963  
Improve multihoming support ...

HDFS-5056  
Backport HDFS-1490 to branc...

Agile Board

Export ▾

Details

Type: Bug Status: CLOSED

Priority: Major Resolution: Fixed

Affects Version/s: 2.0.2-alpha Fix Version/s: 2.0.3-alpha, ... (1)

Component/s: namenode

Labels: None

Target Version/s: 2.0.3-alpha

Hadoop Flags: Reviewed

People

Assignee: Aaron T. Myers

Reporter: Aaron T. Myers

Votes: 0 Vote for this issue

Watchers: 48 Start