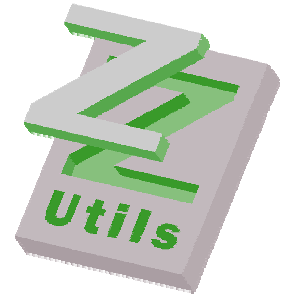


ZZUtils



Source Code Walkthroughs

A source code walkthrough often is called a technical code walkthrough or a peer code review. The typical scenario finds a developer inviting his technical lead, a database administrator, and one or more peers to a meeting to review a set of source modules prior to production implementation. Often the modified code is indicated after the fact on a hardcopy listing with annotations or a highlighting pen, or within the code itself with comments.

A code walkthrough is an effective tool in the areas of quality assurance and education. The developer is exposed to alternate methods and processes as the technical lead and database administrator suggest and discuss improvements to the code. The technical lead is assured of an acceptable level of quality and the database administrator is assured of an acceptable level of database performance. The result is better performance of the developer, his programs, and the entire application.

Despite all the benefits of source code walkthroughs, few organizations implement and enforce them as a shop standard. Many excuses are given, but each has a practical solution.

The bottom line is that code walkthroughs help to improve the performance of your applications and your developers, and **ZZTDOC**, a component of **ZZUtils**, improves the usefulness of code walkthroughs while reducing their cost. And the return on investment is immediate.

Benefits

Improved Code Quality

Improved code quality is ensured by the enforcement of coding standards.

Improved Application Performance

Improved application performance is ensured by the review of all database access paths by the DBA and technical lead, and the improvement/removal of questionable coding practices.

Improved Developer Performance

Improved developer performance is ensured by the mentoring of the developer by the DBA and technical lead, the discussion of coding style and technique. Peer reviews are an important component of a continuing training plan. How else can a developer hone his skills? Few training vendors offer formal sessions for developers with more than five years of experience, and fewer employers take advantage of those sessions.

Excuses

Here are a few of the many excuses offered for not enforcing code walkthroughs as a shop standard.

Volumes of data

Technical leads and database administrators are unwilling to spend time wading through large Natural and COBOL listings searching for a few simple source changes. Database administrators need to see quickly what database accesses were added or changed.

Deleted code

Deleted code cannot be reviewed, but leaving the code in place, converted to comments, can render illegible an otherwise well-structured module.

Distribution/logistics

More and more, development teams are dispersed over several geographic locations, making distribution of documentation more difficult. Even when the distribution concerns are resolved, it is difficult to ensure that all eyes are on the same page. When the developer asks everyone to turn to page 12 of module XXABDC99, invariably someone is looking at page 12 of module XXABCD99.

Manual effort

The amount of effort required by the developer is significant to create useful documentation for a technical walkthrough. The manual procedures involved are difficult, tedious, time-consuming, and error-prone.

Lack of Consistency

There must be consistency. If code is accepted by one reviewer but rejected by another, or accepted on one occasion but rejected on another, developers will become confused and frustrated. There must be consensus among the reviewers.

Once standards are published, developers can ensure compliance, allowing a much more positive and less time-consuming review process. Reviewers can direct more attention to unusual and complex coding techniques.

Developers are reluctant to be involved

Without training, experience, and focus, a technical lead can allow a code walkthrough to degrade into something resembling a lynching. Constructive criticism deteriorates into destructive criticism.

The process must be considered by all parties an opportunity to train the developer, enlighten the technical lead and database administrator, and maintain or improve the quality and performance of the application system. It should be a win-win situation for all persons involved.



Solutions

All but the last of the excuses described in the previous section are addressed by the **ZZTDOC** utility, a component of **ZZUtils**.

Volumes of data

For an updated module, a delta comparison is created. At a glance the reviewer can see whether the module's changes require a cursory review or in-depth investigation. IBM's SuperC source code comparison utility is used to create a full or context listing of the module. In either case, all changes are highlighted with shading for easy location in the source code listing. For a Natural module, an Adabas command listing is generated. This brief report includes the line number, file number, and file name of each Adabas command in the module.

Deleted code

By comparing the old version to the new, all insertions, changes, and deletions are identified without negative impact to the source module.

Distribution/logistics

The resulting report is a word processing document, easily distributed via e-mail or posted to a Web site. Although comprised of several mainframe reports, each with its own page numbering, the entire report has a contiguous page number in the footer. This is the page number referenced in the table of contents.

Manual effort

ZZTDOC can reduce the manual effort of documentation by 95%. Developers have been observed to need an average of five hours to create manually a ten-module report with the same pagination and formatting as the **ZZTDOC** utility. **ZZTDOC** typically reduces that effort to fifteen minutes.

Lack of Consistency

LEN Consulting LLC provides a list of criteria with the purchase of **ZZUtils**. The list includes a set of common criteria covering coding style, performance, and technique. The document is intended as a starting point for clients to customize with their own shop standards.

Developers are reluctant to be involved

The chairperson of the review meeting must be in charge, maintain control, and keep the group focused. He must not allow the meeting to degrade into a destructive mode. To do so he must be given management's support and the authority to control the meeting. The ultimate objectives of a code review must be quality improvement and customer satisfaction.

For assistance with your technical walkthrough policy, contact:

Ralph Zbrog
LEN CONSULTING LLC
562 208 5470 (USA)
RGZbrog@ZZUtils.com

ZZUtils is a trademark of LEN CONSULTING LLC.

Learn more about our products and services at www.ZZUtils.com/LENcon.html.

Natural and Adabas are registered trademarks of Software AG.

