



KRS Techniques in Testing

K.Kalyan Dass

CH. Revathi Gupta

CH. Surya Teja

UNDERGRADUATE STUDENT
 DEPARTMENT OF IT
 MRIET
 kalyan.dass3@gmail.com

UNDERGRADUATE STUDENT
 DEPARTMENT OF IT
 MRIET
gupthachamala0928@gmail.com

UNDERGRADUATE STUDENT
 DEPARTMENT OF IT
 MRIET
 suryatrendz9410@gmail.com

Abstract— This paper deals with the introducing of new unit code testing technique called KRS unit code test, and new test approach called KRS approach, the combination of KRS unit code test and KRS approach is called KRS technique of testing. The major part of the paper is introducing a new way of unit code testing by clubbing of unit code test and Integration test into one unit. In this testing the main aim is to reduce the errors occur in the code and while integration. While in the KRS test approach we try to increase the efficiency of the testing with the help of KRS unit code test. This new approach is built only by considering the KRS unit code test.

I. INTRODUCTION

In these present days testing is made compulsory for organization, our goals in software are increasing but the procedure we follow in testing are not changed. Any project that deals with software is having 5%-10% left out. So our team has come forward with a new proposal.

This KRS unit code testing mainly deals with the coding. This technique is the first technique where unit and integration testing is combined in one platform. Now let us move on to the challenges that have made us to develop these techniques.

II. WHAT ARE THE CHALLENGES THAT MADE KRS UNIT CODE TEST TO BE BUILD

- Automated is the present technique we follow in the unit code testing, in this the major problem is it is time consuming process.
- It is the lengthy process and also difficult to understand.
- This technique completely requires human to solve the errors but not even helping the programmer to know the where the error is.

III. KRS UNIT CODE TEST

- Let us assume that there is a software company preparing a project on xyz software, it might require 3 different languages. This project is assigned to 3 software programmers.

If these software programmers develop the software on one independent platform, there is a chance to avoid 20% of the errors which occur while integration.

A. BLOCK DIAGRAM OF UNIT CODE TEST

Language Selecting Area:

In this block, selection of the programming language is done on the condition of “Optimal Feasible Solution” for the project by software programmer.

NOTE: Make sure of selecting the particular language.

Block diagram for KRS Unit Code Test:

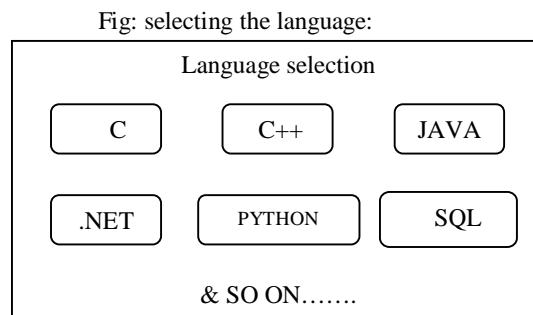
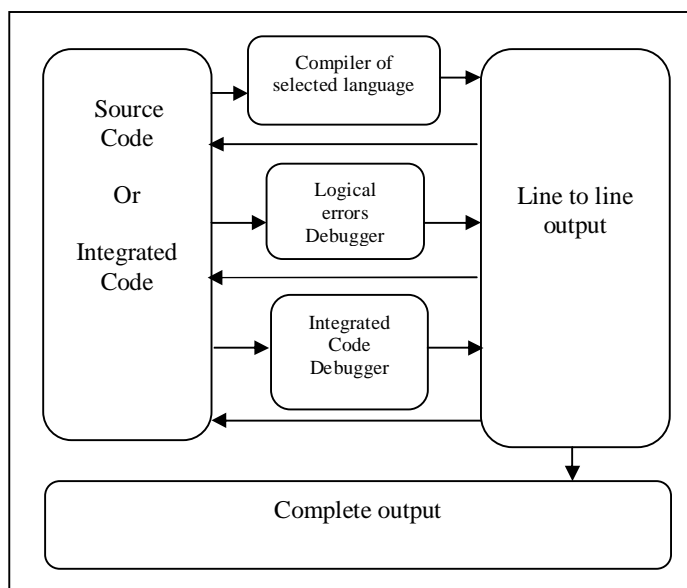


FIG: coding and testing for KRS Unit Code Test



The selected programming language includes the respective compiler into the compiler block.

Instead we can also use common programming language like python.

○ Logical error testing :

Using of **Basis path** testing and **Control structure** testing, we can find 60-70% logical errors and solve the bugs.

It's not possible to find logical errors completely, but the above point is an attempt to find the errors up to 60% of logical errors, by using **artificial intelligence** can solve the bugs up to 100%.

○ Integration Code Debugger:

In Integration Code Debugger, Code debugger and a new type of language are included where it can execute any type of languages and produce output. The basic theme of this language is linking of front end and back end languages and rectifying those errors.

Here we require different concepts in Integration language to find integration errors. The below concepts are considered to solve integration errors:

- Use of array concept to solve the variable declaration bug.
- Use of OOP concept like **class, object** can solve the function calling problem.

Instead of using new language we can also use python language for rectifying the integration errors were it support various programming language.

○ Source code definition:

In this block programmer will write the code into this block as per the selected programming language.

○ Line to line output:

In this block the output is displayed line by line for the source code.

○ Complete output:

In this block the complete output of the source code or integrated code is displayed.

NOTE: The technique of testing used in KRS unit code test is **White box** and **Black box testing** which is not implemented externally but it is **done internally**.

The procedure of KRS unit code test:

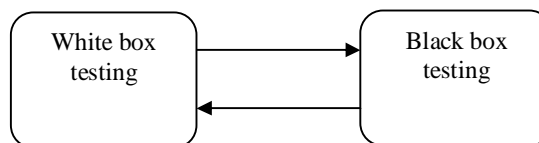
STEP 1: (for Code Developer)

- In the first initial step we select the programming language which is **feasible** to software project.
- Coding is done source code block.
- With the help of selected compiler, compiling is done to the code.
- It will display the line to line output for the source code in the line to line output block,
 - If any bugs occur, it is redirected to source code for debugging the bugs; the programmer can easily identify the bugs in the source code.
 - If no bugs occur in the source code then the execution process goes onto Logical error Debugger block.
- In Logical error Debugger block.
 - If no bugs occur, it will display the output in complete output block and goes to STEP 2.
 - If any bugs occur, it is redirected to source code for debugging the bugs.

STEP 2: (for Project Leader)

- The code received from all code developers is integrated in the integrated code block.
- The integrated code is sent to integrated code debugger
 - If no bugs occur, it will display the output in complete output block.
 - If any bugs occur, it is redirected to integrated code for debugging the bugs.
- The code is then sent to KRS test approach.

KRS UNIT CODE TEST ARCHITECTURE:



KRS test approach

In major problem with the test cycle used for testing(spiral model):

- It is the repetitive procedure where the code has to go different types of testing again and again.
- It is a time consuming process.

By using KRS test approach we can overcome the above problems

- KRS Test code result:

In this part of cycle, customer will contact with the software programmer and gives complete details of the project in the written format, so that there will be no mismatch between developer and the customer. In this phase there will be Formal Technical Review (it means meeting between project team members, project leader, project manager, reporter and customer). After this review meeting all the programmers are assigned with the particular programming task by project leader.

Completion of programmers task the project leader handles the task of integration of code and checks the overall errors after integration.

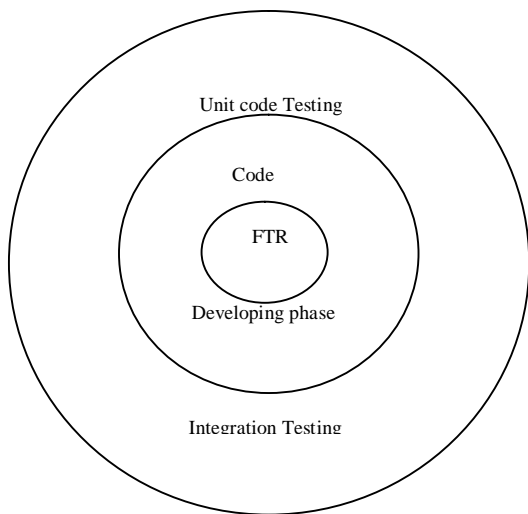


Fig: KRS code developing cycle

ii. Validation Testing:

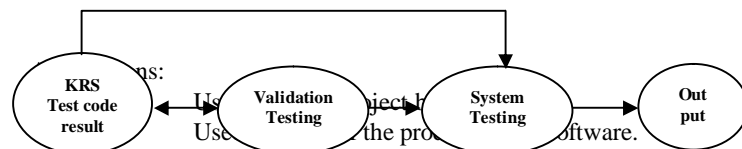
Here the project leader after completing the integration and debugging task, project leader will submit it to the project manager. Project manager analyze whole code and check whether it is satisfying:

- Simplicity
- Readability
- Accuracy
- Security
- Reliability- Recoverability
- Performance
- Resource utilization
- Changeability
- Portability

iii. System Testing:

The project manager will accompany with reporter and validates the codes whether it meets the customer requirements.

Fig: KRS test approach



Advantages:

- The major problem like debugging logical errors is sorted out.
- It is more user friendly
- It reduces the testing cost.
- It reduces the usage of testers.

Disadvantages:

- It is expensive.

Case study:

If the software company wants to build an artificial intelligence software

- In the initial step there will be Formal Technical Review (it includes meeting between programmers or code generators, project leader, project manager, reporter, customer). After this review meeting all the programmers are assigned with the particular task.
- Now, programmer selects the programming language like IPL, LISP, C++ etc which is feasible to the software.
- We are directed to the page which contains compiler of selected language, logical error debugger, integration compiler, source code block, line to line compiler or interpreter, and a output frame.
- We enter our source code into the source code block, after completing our coding, we compile it and we get output in the line to line compiler or interpreter,
 - a) If any bugs occur, it is redirected to source code for debugging the bugs; the programmer can easily identify the bugs in the source code.
 - b) If no bugs occur in the source code then the execution process goes onto Logical error Debugger block.
- In Logical error Debugger block.

- a) If no bugs occur, it will display the output in complete output block and then programmer sends the code for integration..
- b) If any bugs occur, it is redirected to source code for debugging the bugs.
- Now the code is in integration phase, the project leader integrates all the code and compiles it, if programmer finds any error in it can be rectified using a integration code debugger.
- After the integration the whole code is sent to Validation testing here the project manager associates with project leader, taking the FTR meeting into consideration the code is validates. Then it is sent to System testing.

- We can provide the more enhancement for this KRS Techniques in testing in order to provide accuracy.

References:

- [1] Roger S Pressman, Software Engineering, 6th edition, TATA McGRAW-HILL.
- [2] <http://www.makinggoodsoftware.com/2009/05/07/the-pyramid-of-code-quality-the-5-characteristics-of-good-code/>
- [3] C2NzV3_PvUtnFC8mriQf45YGACZXMroQElc35hBqdn6KrOAgAEAFQ99vo1vr_____AWDl6uODmA6gAZm8__ADyAEBqgQoT9Cyr_V3B8TE4uEgaaOPXUiashXSZddugsUjolXmZL_9wbHvsoYBfLgGAYAHz8OAD5AHAg
- [4] <http://msdn.microsoft.com/en-us/magazine/cc163962.aspx>
- [5] http://en.wikipedia.org/wiki/File:Test_Automation_Interface.png
- [6] http://en.wikipedia.org/wiki/Unit_testing

Conclusion:

- KRS Techniques in testing project is used for detecting the bugs and reducing the risk factor for end user.
- This project provides the Simplicity, Readability, Accuracy, Security, Reliability- Recoverability, Performance, Resource utilization,