Regression Testing based Requirement Prioritization of Desktop Software Applications Approach

Varun Gupta\textsuperscript{1}, D.S Chauhan\textsuperscript{2} and Kamlesh Dutta\textsuperscript{3}

\textsuperscript{1}Research Scholar, Uttarakhand Technical University, Dehradun
\textsuperscript{2}Vice Chancellor, Uttarakhand Technical University, Dehradun
\textsuperscript{3}Associate Professor & Head, National Institute of Technology, Hamirpur

\textsuperscript{1}varun13\_cse@yahoo.co.in, \textsuperscript{2}pdschauhan@gmail.com, \textsuperscript{3}kd@nith.ac.in

Abstract

The delivery of new increments involves decision makings at various levels like decision aspect prioritization, requirement priorities and selections of regression test cases. The pressure to deliver high quality low cost software by employing limited resources and satisfying mass market invokes the need to have less dependence on regression testing activity. Earlier Proposed requirement prioritization technique categorizes the type of changes and the type of new implementations and then decides the neglection or minimization of test cases. Further minimizations are also possible due to existence of sound regression testing techniques. By focussing on common problems related to the development of mobile and embedded type software applications, one could have great improvement in software development quality and practices. The proposed requirement prioritization technique is applied in evolutionary mobile software application development and results were quite promising. This paper aims to have a deep insight into the ability of the proposed technique in minimization of regression testing of desktop software applications application. The results are very promising thus indicating the usefulness of proposal in low cost development of both mobile compatible and desktop software applications. The novel contribution of this paper lies in making the incremental development timely and low cost activity by making requirement prioritization effective and future increments less dependent on regression testing without having any compromise with the desktop software quality. This had been a result of analysis of proposed prioritization technique on desktop software application.

Keywords: Re prioritization, Regression Testing, Desktop Software

1. Introduction

Desktop applications are the standalone (offline) software applications that are executed on a single computer, laptops or tablets. Standalone refers to the ability of such applications to get executed without any need for network connectivity with server computing nodes. Such offline applications are thus supplied as an executable application which then is executed after being installed on each computing node. Such software’s are developed with computing node specifications in mind. On the other hand web applications are installed on server node and client nodes accessing the web pages through an internet browser. Thus there is no need for installation on any client computing nodes. Such applications could be made mobile compatible thus providing any time and any place data access. Each of these applications is developed for billions of users that expect desired functionalities to be delivered timely and at low costs. This means that efforts are to be made for timely and low cost delivery of these softwares that satisfies its intended mass market customers. This would be achieved by
incrementally developing software applications by using effective requirement prioritization and regression testing process. Also it would be better if the requirement prioritization process be based on activities that optimize the regression testing activity. This would enhance the quality of each increment by focussing on delivery times and costs associated with development.

In earlier work (Gupta et al., 2013a), the proposed requirement prioritization method that was based on analyzing the effects of requirement selection on regression testing was evaluated in Mobile Application (or just Mobile App). This mobile application is notification system dealing with sending messages among user share groups using the android based Aakash tablets networked in NIT campus. In order to analyze the efficiency of the proposal, it was decided to implement the new requirement and also to perform the necessary changes in already implemented ones. This requires new test cases to execute new requirements and reuse of available test suites to take into consideration generated ripple effects. The results were very promising since it leads to neglection of test cases to consider the ripple effects since changes were not related to variable usage and definition. Also the test cases executing the newly implemented requirements were selected and minimized using regression testing techniques as proposed in (Gupta and Chauhan, 2011c; Gupta et al., 2012c). This happened because newly added requirement is independent of already implemented ones.

There is an increase in the development of Mobile Apps and hence the burden on area of Mobile Software Engineering to effectively address the problems of Mobile Software development. Software developer must have expertise in the development of the desktop applications that could be optimized to be made compatible for use with the Mobile Units as either naive or web applications. Thus the proposal should work perfectly well with both desktop and Mobile Applications.

Incremental development of the software provides its customers the advantage of the timely delivery of application in increments. First increment implements highest importance requirements while delaying the implementation less priority requirements (Gupta and Srivastav, 2010). The activity called requirement prioritization needs to be undertaken by software analysts in order to create the implementation ordering on a set of requirements. An effective prioritization technique that is scalable enough to handle the diversity of a large number of stakeholders and large number of requirements would enhance the quality of increments of desktop software applications.

During the implementation of next increment, the requirement prioritization technique had to create the implementation ordering among already ordered requirements (that are yet to be implemented), changed and new emerged requirements. The final selected set is then implemented. This new increment is then tested by creating the test cases for new and changed requirements and minimization of already executed test cases by analysis of changed code to counter ripple effects. This activity that deals with retesting of already tested code as a result of changes in source code is called regression testing. Thus quality of new increment is the function of quality of prioritization technique, test case designing technique and regression testing techniques.

This paper is based on the objective to analyze the effectiveness of the requirement prioritization technique as proposed in (Gupta et al., 2013a) on desktop software.

2. Earlier Proposal

There had been an increase in the number of mobile users and applications. Increase in the number of mobile users means an increase in the expectations and hence the complexity of mobile applications. Building complex application by involving billions
of users as stakeholders is quite impossible. Pressure of development of high quality software that can be delivered within time and satisfies mass market is the main aim of any mobile or desktop software application developer. The proposed method targets those problems that are common to both mobile and embedded type software’s. It worked perfectly well with mobile applications and the same would be expected for desktop software applications. This paper highlights the reduction in regression testing as a result of applying proposed requirement prioritization technique in embedded type software as an experimental tool.

The proposed method is basically a decision making method involving large number of decisions at various stages of development. One of the decisions is to be made regarding the selection of the prioritization aspects and other decision pertains to selection of requirements by allotment of priorities. Proposed method replied on using the decision aspect prioritization method as proposed by (Gupta et al., 2012a) and requirement prioritization method as proposed by (Gupta and Srivastav, 2011a) for allotment of priorities against the prioritized decision aspects. In order to take into consideration the effects on regression testing, dependency count is computed using an Analytical hierarchical process employing pair wise comparisons. Finally, the requirements with a priority value greater than threshold value and dependency count of greater than 30% are selected for implementation. During implementation of new versions, delayed requirements are implemented and changes in source code are implemented resulted as a result of the changed requirements. By analyzing the type of changes and the type of new requirement implementation, the decision could be made regarding the neglection or minimization of test cases. Further minimization could be obtained by employing the regression testing techniques as proposed by (Gupta and Chauhan, 2011c; Gupta et al., 2012c). This approach aims at minimizing the regression testing activity thereby reducing the cost associated with testing and hence development costs.

3. Dynamic Reprioritization Model

In order to discuss the proposed requirement prioritization technique, case study of live system developed under supervision of corresponding author entitled “SOFTWARE for ABC Organization” is employed. This project was developed in Computer Programming Laboratory of National Institute of Technology, Hamirpur. Key features provided by this software include directory management, staff management, new recruitments etc. Bill management, payroll management and the profit/loss of the company can also be calculated by this software. Any information of a customer can be modified or deleted or can also be searched. The requirements of this software were hierarchically related and given by Figure 1:
After students implemented the project, they were asked to get evaluated during project presentation. Although students prioritized requirements using models proposed by (Gupta, Chauhan and Dutta, 2012a; Gupta and Srivastava, 2011), yet some changes were suggested by the evaluation committee. One of the changes to be implemented was to perform short listing not on the basis of “age”, “percentage” or “Cumulative Grade Point Index (CGPI)” at Bachelor or Masters Level but on the basis of calculation of Academic Performance Index (API) by combining criteria like educational qualification, percentage, experience, publications etc. This happened because the organization requires to shortlist new applicants that satisfies above mentioned multi objective criteria’s. Required objective values are added up using a formulae yielding single value called API. Although the software is about to enter its second release, yet we can consider the case of implementation of new requirements. The earlier development team thought that password protection through encryption would be quite challenging due to limited development time, but later it was decided to at least provide password protection without any encryption. So students implemented this new requirement of password and tested the whole system using the proposed algorithm. So following were tasks to be completed:

(1) **New requirement**: Software must provide security by enabling password protection.

(2) **Changed Requirement**: Short listing by calculation of Academic Performance Index (API). This happened because the organization requires to shortlist new applicants that satisfies multi objectives like good percentages, large number of publications, highly experienced etc. All these required objective values are added up using a formulae yielding single value called API.

In order to implement the new requirement related to password protection following modules were added:

---

**Figure 1. Hierarchically Related Requirements of Software**

[Diagram showing the hierarchy of software requirements with modules like Customer Record Management, Staff Management, Password Protection, Financial Management, and Recruitment Management.]

---
**Code for changing password**

```c
void change_pwd()
{
    int i,j;
    char ch, b[20], c[20], d[20] = {}, ch1;
    FILE *f,*f1;
    f=fopen("password.pdf","rb+" obviously a mistake; should be "rb+\n"");
    f1=fopen("password.dat","wb+");
    while(fscanf(f,"%s",b)!=EOF)
    {
        printf("\n\n\n\nENTER NEW PASSWORD \n");
        for(i=0;i<strlen(b);i++)
        {
            ch=getch();
            c[i]=ch;
            ch='*';
            printf("%c",ch);
        }
        printf("\n CONFIRM PASSWORD \n");
        fflush(stdin);
        for(j=0;j<strlen(b);j++)
        {
            ch1=getch();
            d[j]=ch1;
            ch1='*';
            printf("%c",ch1);
        }
        if(strcmp(c,d)==0)
        {
            printf("\n\n password changed successfully \n \n ");
            fprintf(f1,"%s",c);
        }
        else
        {
            printf("\n password doesn't match \n Older password sustains\n ");
        }
    }
    printf("\n\n\n\nGood Day\n");
    fclose(f);
    fclose(f1);
    remove("password.pdf");
    rename("password.dat","password.pdf");
}
```

**Code for checking password**

```c
char k3[20],k1[20]="default";
FILE *r;
r=fopen("pwd.bin","rb+");
fscanf(r,"%s",k3);
```
fclose(r);
char ch,ch1,c1,c;
printf("\n\n");
for(int j=0;j<80;j++)
{
    int x,count=0;
    char a[20],b[20],tri,p;
    FILE *f;
    f=fopen("pwd.bin","rb");
    fscanf(f,"%s",a);
    fflush(stdin);
    fclose(f);
    p=strlen(a);
    printf(" 
	 PLEASE ENTER PASSWORD 
");
    fflush(stdin);
    J: for(int k=0;k<p;k++)
    {
        tri=getch();
        b[k]=tri;
        tri='*';
        printf("%c",tri);
        fflush(stdin);
    }
    if(strcmp(a,b) == 0)     {
        //Code for menu driven application goes here
    }
}

This new requirement is only control dependent on existing code. Thus only the test cases executing this requirement will be required leading to neglection of other test cases. To implement the changes in existing requirement of short listing the applications, following changes are made to existing code.

**Old Code**

```c
void short_list( )
{
    char ch;
    int age,age1;
    float CGPI;
    FILE *f,*p;
    f=fopen("recruit.bin","r");
    p=fopen("shortlisted.data","w");
    K: printf("\n\n\t\tn SHORTLIST ON BASIS OF :-\n");
    printf("\n\t\tx B0\tx B0\tx B0\tx B0 SHORTLIST ON BASIS OF :-\n");
    printf("\n\t\tx B2\tx B2 1. AGE\n");
    printf("\n\t\tx B2\tx B2 2. CGPI\n");
    ch=getch();
    if(ch=='1')
    {
        printf("\n ENTER MINIMUM AND MAXIMUM AGE ");
        scanf("%d%d",&age,&age1);
        while(fread(&rc,sizeof(rc),1,f)==1)
```

14
```c
if((rc.age<age1)&&(rc.age>=age))
{
    fwrite(&rc,sizeof(rc),1,p);
}
else if(ch=='2')
{
    printf("n ENTER MINIMUM CGPI REQUIRED");
    scanf("%f",&CGPI);
    while(fread(&rc,sizeof(rc),1,f)==1)
    {
        if(rc.CGPI>CGPI)
        {
            fwrite(&rc,sizeof(rc),1,p);
        }
    }
}
else
{
    char ch1;
    printf("n WRONG CHOICE.......PRESS 1 TO GO BACK, 2 TO EXIT");
    ch1=getch();
    if(ch1=='1')
    {system("els");
        goto K;
    }
    else
    {exit(0);
    }
}
fclose(f);
fclose(p);

**Modified Code**
void short_list()
{
    char ch, name[25], education[12], Phd;
    float exp;
    int n,maxage,percentageu, percentagep, Ij, Nj, Ic;
    float api;
    FILE *p;
    p=fopen("shortlisted.data","w");
    printf("n ENTER NUMBER OF APPLICANTS TO BE ENTERED IN THIS SESSION");
    scanf("%d",&n);
    for(int i=0;i<n;i++)
    {
        printf("n Please enter your age, name, education qualification, percentage at UG");
    
```
printf("n, Percentage at PG, whether PhD or not, experience (in years), papers in I.Journals, N.Journals, I.Conferences");
fflush(stdin);
scanf("%d",rc.age);
fflush(stdin);
scanf("%s",rc.name);
scanf("%s", education);
fflush(stdin);
scanf("%f%f", percentageu, percentagep);
fflush(stdin);
scanf("%c", Phd);
fflush(stdin);
scanf("%d%d%d%d",exp, Ij,Nj, Ic);
printf("Enter maximum age allowed as per advertisement");
scanf("%d", maxage);
if(rc.age<=maxage)
{
 if(Phd == 'y') //add 20 marks for being PhD.
 rc.api = percentageu + percentagep + 20 + Ij + Nj + Ic + exp;
 else //No need to add 20
 rc.api = percentageu + percentagep + Ij + Nj + Ic + exp;
}
else
{
 printf("Sorry your age exceeds the maximum allowed");
}
float minapi=156.0;
while(fread(&rc,sizeof(rc),1,p)==1)
{
 if(rc.api>=minapi)
 {
 fwrite(&rc,sizeof(rc),1,p);
 }
}
fclose(p);

As shown above, implementation of changes in the requirement satisfies two of the cases where changes are not related to changes in functionality and changes are not related to variable usage and definition, thus modules of the software employing services of this module are unaffected. In this case only test cases executing these changes are to be executed without a need for testing other requirements. Cumulating the above results, related to new or changed ones, only required test cases are those that execute the new requirement.

6. Conclusion & Future Work

The proposed technique is quite beneficial for both embedded and mobile applications in terms of reduction in development cost by minimization of regression
The proposed technique is basically a framework where various proven techniques for decision aspect prioritization, requirement prioritization and regression test suite minimization are used along with new methods of requirement prioritization based on analysis of effects on regression testing. Application of proposal on both mobile and desktop software application apps resulted in very promising results in terms of reduction of regression testing activity. In the near future, it is expected to be tested on various web applications including social networking sites. Further, it is expected that the proposal will be evaluated on complex desktop software’s with diverse characteristics, using different test case designing approaches. This evaluation will highlight future research directions.

References


Authors

Varun Gupta, he is a research scholar at the Uttarakhand Technical University, Dehradun. He is teaching at Jawaharlal Nehru Govt. Engineering College, Sundernagar and earlier worked with National Institute of Technology, Hamirpur and Indian Institute of Technology, Mandi. He holds a Masters of Technology and Bachelor of Technology in Computer Science & Engineering. His area of interest is Software Engineering (Requirement Engineering, Requirement prioritization, Global Software Development, Software Testing etc). The author has authored numerous research papers in peer reviewed, referred International Journals and various International and National Conferences that were held in India and Abroad.
Prof. Durg Singh Chauhan, he received a Ph.D. degree from Indian Institute of Technology, Delhi in 1986, and did his postdoctoral work at Goddard space Flight Centre, Greenbelf Maryland, USA (1988-91). He is currently working as Vice Chancellor, Uttarakhand Technical University, Dehradun. Dr. Chauhan also served as Vice Chancellor of three other Indian universities. He had been nominated chairman/member of various committees of national importance. He got the best Engineer honor of the institution of Engineer in 2001 at Lucknow. He has authored two books and published and presented 150 research papers in international journals and international conferences. He is Member of prestigious organizations like IEEE, ACM, SIAM and Fellow of Institution of Engineers.

Dr. (Mrs.) Kamlesh Dutta, she is Associate Professor and Head in the Department of Computer Science & Engineering at the National Institute of Technology, Hamirpur (Himachal Pradesh) INDIA. She obtained her doctorate from Guru Gobind Singh Indraprastha University, Delhi, INDIA and her master’s degree in Computer Science & Engineering from Indian Institute of Technology, Delhi, INDIA. Her major research interests include Software Engineering, Artificial Intelligence and Natural Language Processing. She has published more than 80 research papers in national and International Journals and Conferences.