Vol. 02, No. 01 (2020)
23 DECEMBER 2020

E-ISSN: 2582-7502
JS International Journal of Multidisciplinary Research (JSIJMR)
www.researth.iars.info/index.php/JSIJMR
www.jsu.edu.in

Listen to Customer Cries, Before the Software dies

[*1]Jain, Deepak

[*1]Hughes Communications India Limited

ARTICLE DETAILS

Article History

Article is originally taken from primary/contacting (co)author as an original research manuscript submitted to JS International Journal of Multidisciplinary Research (JSIJMR), published by JS University Shikohabad, U.P. on ***JCONSORT: Consortium of Research Journals****, managed by Nirmal Integrated Consultancy in association with International Association of Research Scholars (IARS).*

Keywords

Software Retirement, Software Maintainance, Software Death

[*]Corresponding Email

iars.researth@gmail.com; jsijmr@iars.info

ABSTRACT

The software marketplace is in a constant state of change with new and disruptive markets emerging while some traditional markets fall by the side. The software development market is also characterized by an irrevocable trend towards greater software complexity -- complexities that not only complicate the community of software developers but also create financial pressures on them as well as their customers. Software technology is changing at such a high rate that software bought this morning gets outdated by next evening. In such a scenario, many a times consumer of the product suffers for paying for an older software unknowingly. It suffers with the system when developer organization of such old systems stops providing technical supports on them stating that newer versions are available in the market to solve their problems and hence the customer should discard the system and purchase a new one. In this paper, the author proposes that software has finite lifetime regardless of its type and development platforms and it must be dismantled before it dies. The process of dismantling the software is treated as its retirement. Author brings into light the current poor scenario of software retirements and desires of consumer community in favor of code of ethics for software developing organizations.

1 Introduction

Human usually get retired from their jobs after a certain life. Maximum age of their working is usually defined in the policy documents or contracts of working. Not only human but also machines and even buildings get retired at a certain age. Bridges may be the best example for this. While making of a bridge, usually its life is estimated and warnings of discontinuation of their use afterwards are cited. It is common and obvious to understand that human as well as machines and buildings have a finite life after which they come to a state where they should be dismantled from their jobs. Discontinuing their use does not mean that they cannot serve their purpose any more, but it in fact means that risk in continuing their use has increased to an unaffordable level.

Similar behavior is observed with software products also. According to changes in its environment and expectations of the consumer, the software products need modifications in them. These modifications in the product give adverse effects in its reliability and quality of services. Also, the quality of design of the software is degraded. Software slowly reaches a state where any more modification in it may result to a crash of the system. This state may be treated as retirement or exhaust state of the software.

Figure 1 - Everything retires, be a human or machinery or building.

This paper discusses need of proper planning of retirement of software and adopting software retirement as a milestone in their life cycle as similar as in case of human employees so that the customer of the software need not to pay for losses due to untimely crash of software.

2 CONCEPT OF SOFTWARE RETIREMENT

Software retirement is a post development stage or activity in the software life cycle that is to be executed in a formal manner and with sincere efforts as any carelessness in the procedure may cause big harms to the consumers of the software. Dynamicity of software and hardware technology demands quick changes and enhancements in the systems.

Any post development alteration in the software degrades it reliability and give adverse effects on its quality of service. This is called aging of software. Amendment of the system is limited by its enhanciability which is the extent to which the system may be enhanced with affordable efforts and controllable risks. Enhanciability of the software hence limits permission to make any alteration in it. In the early chapters of this book, it is explained that software shows deteriorating behavior for any modification. The extent of deterioration and cost of normalization increases exponentially with age of the software. Finally, the system reaches its state of saturation where any more changes may result in complete crash of it or the alterations cause unaffordable costs and resource consumptions. Such state expects the software to be declared “dead” or “exhaust” or “retired”, and to stop further working on/with it.

2.1 Importance to understand software retirement

Software, these days are the major driving factor in almost all needs of business and home. Have it be a monitoring system or finance tracking system, software plays important role in all. Software need modifications and updates to comply with its changing environment and meeting changed needs of consumer. Post development leave aging signs in software disturbing its design and structure. Working with aged software involves a risk of crisis due to unplanned crash of the system due to any error in software.

Hence it is recommended to decommission the aged software well in time in a planned manner to save the losses of crash. This makes it important to understand the concept of retirement of software and its process of execution.

2.2 Certainty of Software Death and Retirement

All software must reach to its state of retirement or death in a finite time span. Software life cycle normally follows the steps from requirement engineering through designing, coding and testing and usually ends into its maintenance phase. The enhanciability of software limits the permissions to modify it and in turn limits its life. In a finite time, the software must reach to an extent where the cost of enhancement shoots unaffordable and the failure rate rises uncontrollable. Hence, in a finite time, software reaches its retirement. Software retirement or exhaust indicates the time of retiring the software and suspension from usage or new deployments. Limited enhanciability of the software ensures certainty of its death. In this manner, software life cycle should begin with an idea or need that can be satisfied fully or partly by any software product and it must end with the retirement of the software product.

3 CURRENT SCHENARIO OF SOFTWARE RETIREMENT

Software development companies develop, test, install and provide training for the user to work on the software product. During its use, the developing organization is responsible for debugging the software for the new discovered errors and bugs in the software those were left undiscovered during its testing. In order to enliven their relations with customer, the developing company may even update the software for a limited course of time, accommodating the changes in user requirements and technical upgradations in its environment.

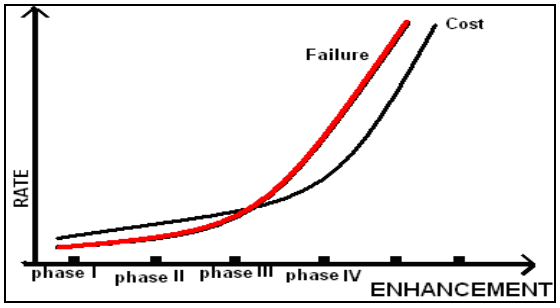

Figure 2 - Rise in cost & failure rate with enhancement

The companies are found to show fair interest in the software and trying to provide their best in order to meet the consumer expectations and their satisfaction for a course of time. But unfortunately it does not last long. Slowly and gradually, the company looses its interest to enhance the system as it accounts for big hassles and long insecure efforts. Working on aged software involves higher risks of failure.

This should be taken ethical on company side to declare their software dead when it reaches to the state of “Exhaust”.

Ethically, any developing company is expected to announce a formal retirement of its product when it reaches to its state of exhaust. Declaration about decommissioning of software, if made from customer or user side is one thing that may not be that objectionable because in this case, the stakeholder of the product or the consumer is not in any hidden or unannounced harms. But if decision about retirement of a software project comes from the developing

organization, i.e. when the developing company decides to stop production and support of any of its product, then the consumer may suffer undeclared immediate or longterm losses because of unplanned or unannounced informal retirement of the product. Due to such informal retirement of the software, the consumers suffer the loss of information and money.

Software companies generally do not give any formal declaration for exhaust of any of its software products although it happens for all software. In some (one in thousands) cases, even if some companies announce the retirement of any of its product, they do not publish any formal report for this that would contain the information about the facts like major reasons of retirement, technical limitations the product is facing and the focus or planning of retirement of the product. Generally, companies silently stop producing and maintaining its so called retired product because updating it seems to be too costly to afford and the failure rate gets increased frenziedly. Companies and their dealers usually retire their products silently to preserve their state of market. This may be termed as a silent death of the software or even a breed of it.

The problem becomes even worse in case of retail and original equipment manufacturer (OEM) and market software product where the product is used by mass community. In such a case when the information is not published formally, users across the world suffer. The similar problems occur with manufacturers of hardware computer parts too. Many times, companies stop manufacturing some particular hardware or any of its models. At the same time, they stop updating its software drivers too.

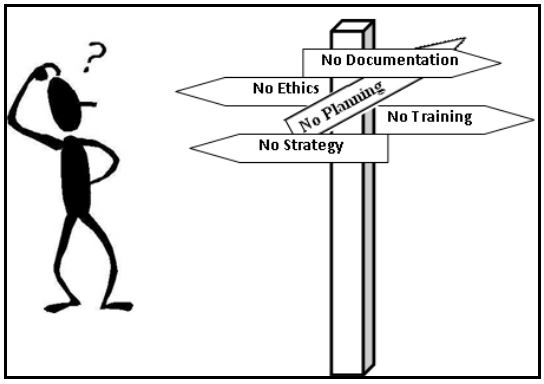

Figure 3 - Hurry & Mess: the current scenario of Software Retirement

A problem of unavailability of updated drivers for hardware modules to use them with newer operating systems is very common in user community. Operating systems change at a very fast rate e.g. WIN9X series products were quickly replaced by WINXP and WIN2000. But hardware does not vanish so fast and they need to be updated for their drivers. The problem persists with hardware that have been stopped in production and the companies show no interest towards the customers who paid them just a short time back, especially in case of hardware like internal modems and LAN cards.

4 CUSTOMER CRIES

It should be ethically noted that the customer (and also consumer in many cases) is charged a reasonably high amount as the cost of the software and license for their commercial and noncommercial usage. They also pay the costs of services of training and supports. During the process of snatching the tender of project from its competitors, the customer is assured for the best and uninterrupted timely services by the

developing companies. The customer is provided with descriptions of a number of past successes stories hiding all failures histories and their old products for which they have stopped working upon. One could never name any single company that ever mentions and addresses the problems of their customers of any of their old products that they had stopped working for. Also not many companies are working to help their customers get rid of these problems.

It may be expectable and acceptable and even bearable to replace such software or drivers (or even complete hardware & driver set in some cases of acceptable cost) with another suitable. Needless to say, the companies do not follow such strategies of replacing their old products except in some special cases of legitimate values. Moreover, when these companies execute exercises of replacing, they do not follow any formal and ethical manner. Consumer is charged for the new developments and only an inferior or no documentation at all is prepared about the reasons of replacing the software and analysis results of replacement. The consumer is never described about why the software is replaced and why the last one could not be enhanced as per its needs. Usually companies do not provide formal declarations about selected replacements too.

In future, it becomes hard to retrieve the reasons of replacement of the existing software with altogether new software and the process become tough to trace. Also, the consumer who has spent money on the development and maintenance of the previous software is expected to have a right to know why the company now has stopped providing updates to that software and what can be the cost of further upgradations.

The problem becomes worse in case of OEM and market products where the product is under usage of a bigger community across the globe. The companies are ethically required to publish the information about their decisions of no more working for enhancements of a product so that the user community may look for an alternative well in time. It is accepted that running any software beyond time boundaries and making it adaptable to all types of changes in its environment may not be feasible for a company and they need to take vindictive decision of stopping the production and maintenance of such products. But at the same time they need to put in mind the cost the consumer has to bear for that.

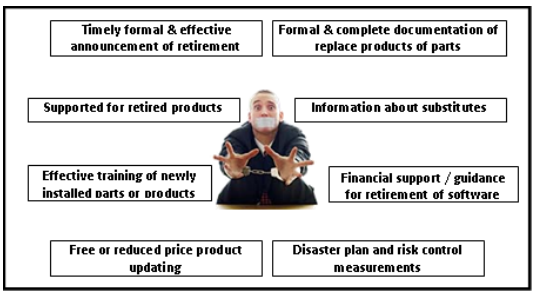

Figure 4 – Customer's Expectations

Further, the developing company should publish the formal information about exhaust or retirement of any of its product. A formal report of exhaust or retirement of the software needs to be published on the responsibility of the developing company. This report must clearly be describing key reasons for declaring the software retired and stop spending and working on it. In details, it should be enlightening the technical bottlenecks of the product in updating it to copeup with the current

technology expectations. Further, it should be suggesting some excellent alternatives of the product that may be used for replacing the retired product.

Not only this, the organizations are expected to provide a formal plan for disaster management against the probable (howsoever small) losses due to retiring of the software. They should produce a public report describing the expected areas of disaster for related fields. Generally companies prepare these reports, but they do not provide it for public or customer feedbacks, which are expected for the consumer.

5 ETHICAL RESPONSIBILITIES OF COMPANIES

Companies declaring retirement of any software must raise ethical responsibilities on their own for the probable losses due to that retirement. Software developing organizations must not declare the retirement of any software until unless a viable path of migration from the software exists for its consumers.

Organizations need to adopt softwareretirement as a milestone in life cycle of the software product. The developing companies, while announcing the retirement of any of their software product, should provide (if possible) a formal or approximate schedule for retirement of the software well in time that allows the consumer to settle and prepare its environment, infrastructure and other arrangement to minimize the losses. The companies must effectively discourage new deployments of a system that is either retired or whose early retirement is expected.

Also the company announcing the retirement of the software must specify clearly what it means by retiring the software and what all services will be stopped once the software is retired. They need to mention whether the consumer will be getting some suitable replacement of the product at some discounted or controlled costs. They should give the formal declarations for all the terms and conditions for the retirement of the software. The company is supposed to provide the information about the impacts of retirement of the software on other products of the company and its related systems. The users should be suggested with the remedy actions against the possible losses of information while opting for a replacement. The customers should be provided the suitable assistance with technical support for upgradation of their systems.

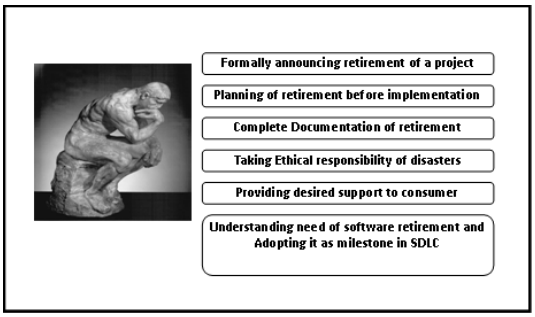

Figure 5 – Ethical Responsibilities of Seoftware Developing Organizations

6 CONCLUSION

Exhaust of software in a finite time is certain. Software should be retired or dismentalled before it exhausts or dies. Consumer expects formal retirement of software from its development companies. Formal retirement of software should be preplanned in the life cycle of the software and announced well in time with a focus of minimizing the loss to the consumer. Software developing organizations should

bear their ethical responsibilities for retirement of the software.

7 Acknowledgement

Acknowledging Prof. (Dr.) Gaurav Sharma for opportunity to write and publish on this topic and Prof. (Dr.) D. Jain for editing and formatting of manuscript. The International Association of Research Scholars (IARS) is to be acknowledged for their support in the publication of the study outcome.

8 Reference

[1]. Ambler, Scott W., 2006, Agile Modeling and the Rational Unified Process (RUP), AmbySoft Inc.,

[2]. Ambler, Scott W., 2006, Extending RUP with Retirement Phase, AmbySoft Inc.

[3]. Highsmith, Jim, 2001, Agile Software Development: The Business of Innovation, Computer, IEEE.

[4]. Jain, Deepak, 2006, Why need process: Software Development and Standards, DIAS Times, Delhi Institute of Advanced Studies, India.

[5]. Jain, Deepak, 2007, Software Development Process Models: A Questionnaire Approach, BPB Publications, India.

[6]. Jain, Deepak, 2009, Software Engineering Principles and Practic-es, Oxford University Press, India.

[7]. Jalote, P., 2005, An Integrated Approach to Software Engineering, 3rd ed., Narosa Publishing House, New Delhi.

[8]. Krasnen, Jerry, 2003, Total Cost of Development, American Technology International, DOI=http://www.embedded-forecast.com

[9]. Pressman, Roger S., 1997, Software Engineering: A Practitioner's Approach, 4th ed., Tata McGraw Hill.

[10]. Scott W., 2006, Enterprise Unified Process (EUP), DOI=http://www.enterpriseunifiedprocess.com/essays