

Available online at www.sciencedirect.com**ScienceDirect**

Procedia CIRP 22 (2014) 81 – 86

www.elsevier.com/locate/procedia

3rd International Conference on Through-life Engineering Services

Mitigating the Risk of Software Obsolescence in the Oil and Gas Sector

John Ahmet Erkoyuncu*, Stanley Ononiwu, Rajkumar Roy

*Manufacturing and Materials Department, Cranfield University, Bedford, MK43 0AL, UK.** Corresponding author. *E-mail address:* j.a.erkoyuncu@cranfield.ac.uk

Abstract

This paper focuses on how on-going energy demand, shift in water depth and heavy crude oil type are creating huge obsolescence issues especially software in the offshore oil and gas industry. The aim is to identify and quantify software obsolescence using a novel framework developed to evaluate major software types, their associated obsolescence impact and risk exposure, cost implications, and resolution qualification. Validation of the framework confirmed the role of the framework for life cycle support and guidance related to software obsolescence. The tool can be used for engineering and procurement contracts thereby reducing capital and operating expenditure

© 2014 Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/3.0/>).

Peer-review under responsibility of the Programme Chair of the 3rd International Through-life Engineering Conference

Keywords: Software obsolescence, risk management, oil and gas sector

1. Introduction

Obsolescence can be defined as the loss or impending loss of parts or suppliers of parts or raw materials [1]. Obsolescence can also be defined as the state of being no longer needed because something newer or more efficient has been invented [2]. Obsolescence is the unavailability of parts, or services that were previously available, and they could have been withdrawn for many reasons: technical, financial, legal and others. It affects components which need to be maintained for long periods of time which can be 30 years or more. There are a variety of types of obsolescence including: skills, software, mechanical components, electronic components, testing equipment, documentation, tools, processes, materials and people [3].

Almost every project across manufacturing industries have software with various degrees of complexity and dependencies. Software is defined as programs, procedures, rules, data and documentation associated with programmable aspects of system hardware and infrastructure [4].

To address these challenges, in this paper a case study of key software types in the oil and gas industry is used to develop a roadmap tool for the risk evaluation of factors influencing software obsolescence. From the evaluation study, approaches to manage the whole life cycle cost of obsolescence in the oil and gas industry is realised. This study aims to address the following key questions in association with the oil and gas industry:

- What are the drivers of software obsolescence?
- What are the key components and software affected by software obsolescence and to what degree?
- How can the risk of software obsolescence be evaluated?
- How to identify and quantify software obsolescence using a cost estimating framework?
- How can to mitigate or manage software obsolescence from engineering procurement companies (EPCs)?

Section 2, covers the methodology that was followed for the study. Section 3, presents the literature review, whilst focusing on software obsolescence and the identification of research gaps. Section 4 focuses on the current practice and

challenges in the oil and gas sector. Section 5 presents an overview of the developed framework to manage software obsolescence. Section 6 covers results from validation and Section 7 presents the conclusions and future work.

2. Methodology

The research methodology involved three phases. The initial phase focused on interviews to rank obsolescence risks the oil and gas industry i.e. software, mechanical, and electrical. As an outcome of the study software ranked the highest risk. Additionally, literature review was conducted around software obsolescence. This phase also involved drafting a questionnaire and conducting semi-structured interviews (face-to-face and online via WebEx) that helped to unravel the degree of software obsolescence. A face-to-face questionnaire was administered to industrial collaborators with an Obsolescence Lead (over 15 years of experience of which 3 years spent managing obsolescence issues) and head of I.T. department (over 10 years of experience). In phase one, the interviews lasted over five hours. The first part was used to present the aims and purposes of the interview focused on describing the aim and objectives of the study and subsequently, the semi-structured interview was followed and aimed to identify the current practice and challenges.

Phase two of the methodology involved a case study to capture the causes and effects of software obsolescence. This involved developing fishbone diagrams. A real life project was examined through documents, and further interviews were held (a total of four hours with the above mentioned participants) to fill the fishbone diagrams that capture the key software drivers and builds links to demonstrate conceptually the impact of obsolescence.

Phase three involved framework development and a prototype tool (embedded in MS Excel) was developed. As part of this phase three interviews (each lasting one hour with the obsolescence lead) were administered and all the responses were used to initially develop and later modify and update the prototype tool which can be used to address whole life cycle cost of software obsolescence. This phase also involved validation and verification. These phases have been presented in Figure 1.

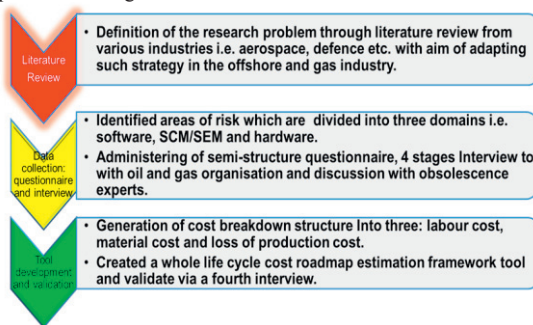


Fig. 1. Research methodology

3. Literature review

3.1. Obsolescence

Cretenet [5] and Aker [6], list the various reasons for obsolescence occurrence as follows:

- Technological development: new technology replaces the existing technology.
- New functionality and requirement on the system.
- Innovative life cycles of components shorter than the system life cycle.
- The original equipment manufacturer (OEM) no longer finds the product viable to produce.
- The OEM is no longer in business for any reason.
- Changes in legislation within the industry such as Restriction of Hazardous Substance and Waste Electrical and Electronic Equipment (RoHS).

3.2. Software obsolescence

There are divergent views about software obsolescence. A particular view claims that software cannot fall into an obsolete condition since it does not go through degradation therefore it will not need replacement, another argues that software does not fail per se either it works or not. In response to this view [3] asserts that applying the same idea of degradation to software obsolescence like mechanical or Electronic Electrical Electro-mechanical (EEE) components obsolescence will be a misconception. Again, he went on to identify the uniqueness of the software obsolescence and acknowledging this problem will be important.

Knowing that software can be obsolete allows the maintenance and support of the system to be given room for better understanding that software obsolescence has similarities with hardware obsolescence. For instance, when hardware components become obsolete and no last-time buy opportunity is available, maintaining the original planning becomes difficult. The same applies to software obsolescence, analogically and by implication the software is prevented from been maintained.

Romero-Rojo [3] stressed that internal or external factors can also trigger software obsolescence both within the development environment and at the target environment. Sandborn, et.al [7], stressed that there are many methodologies, strategies and tools that tries to mitigate and manage electronic components obsolescence i.e. MOCA, PRICE-S and COCOMO, TOMCAT and EEE-FORCE. Traditionally, the current practice focuses mainly on the hardware life cycle thereby neglecting software life cycle costs which can also drive the total life cycle cost.

Merola [8] is of the view that software obsolescence has become a major concern in the United States and in the United Kingdom, largely because it has not been given the type of proactive attention it deserves. Also Sandborn [9] added that electronic component obsolescence has a multiplier effect on software programmes. Note, hardware replacement will substantially cause supporting software program not to support the particular hardware. Software obsolescence may also be caused due to human knowledge/ergonomics level or

when the supplier withdraws support. Figure 2 presents the different types of software obsolescence.

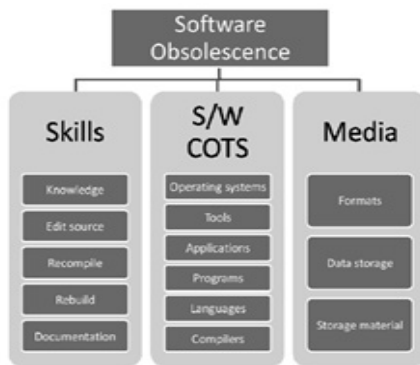


Fig. 2. Types of software obsolescence (Source: Romero Rojo et al., [10])

3.3. Risk mitigation

Obsolescence risk assessment is essential to manage obsolescence effectively. Although there are standards to manage obsolescence, such as the JSP886 Vol. 7 Part 8.13 and EN 62402:2007, none of them define explicitly a process to assess the risk of obsolescence that can be systematically used by industry. Risk mitigation is facilitated by evaluating the likelihood of obsolescence and its impact, which requires quantitative analysis. Figure 3 explains a number of steps that could be considered in an obsolescence management plan (OMP). An OMP is a proactive approach/methodology to address obsolescence and help to predict, detect, quantify, mitigate and report in the most cost effective way and timely manner and guidance for obsolescence management for system life cycle support [11].



Fig. 3. Offshore Oil and Gas obsolescence mitigation strategy [12]

3.4. Component Obsolescence Group: Software obsolescence

Component Obsolescence Group (COG) is a special interest group of like-minded professionals, from all levels of the supply chain and across all industries and relevant Government agencies, concerned with addressing and mitigating the effects of obsolescence. Parkinson [13] states the need for standards to assist with managing software

obsolescence with increased project lifecycles. Gowland [14] state the areas to consider for software obsolescence as:

- Source code: Applications, Board Support Packages (BSPs, embedded firmware, Programmable Logic Device (PLD) data, build data and documentation
- Software Tools: Compilers, build utilities etc.
- Operating Systems: Evolve (not always compatibly) or vanish
- Hardware Platforms: Continually evolving for better performance

3.5. Research gaps identified

- Lack of methods and tools to quantify and manage the effects of software obsolescence and incremental update in the oil and gas industry.
- Lack of tools to estimate the cost of software obsolescence.
- Lack of approaches proposed for risk management of software obsolescence.

4. Identifying current practice and challenges: Oil and Gas sector case study

The case study involved Cranfield University and an oil and gas engineering procurement company (EPC) based in the UK. As a global leading EPC company and in forefront of addressing obsolescence issues, software obsolescence has significantly impacted project budgets with high cost of mitigation approach i.e. reactive approach in addressing it. For this reason the case study aimed to determine strategies for effectively managing and mitigating software obsolescence issues. EPC provided relevant data for this study. A list of software used in the offshore oil and gas industry was identified as follows:

- Microprocessor
- Computer or industrial programmable computer (IPC)/ programmable logic controller (PLC).

4.1. Obsolescence identification

The obsolescence type is identified as “software” which is the highly vulnerable in the oil and gas industry. Also, prediction is a key to identifying which software type is vulnerable. The list below is a guideline used when identifying obsolescence risks:

- Life cycle assessment,
- Software Industry trends in relation to oil and gas industry trends,
- Market forecasts,
- Incremental upgrade costs,
- Reliability issues,
- Safety i.e. cyber security etc.,
- Complexity i.e. level of complexity based on component/software interaction high, medium or low,
- Commonality i.e. alternative, FFF,
- Changing performance requirements,

- Mean time between failures,
- Accredited software supplier availability/support,
- Tailor made software design or manufacturing processes

4.2. Obsolescence risk assessment

Risk assessment takes in to account the impact and the likelihood that is experienced from obsolescence. Where:

P= probability of failure or probability of obsolescence

C= consequences of failure added to the consequences of obsolescence

$$\text{Risk} = P_{\text{failure}} \times P_{\text{obsolescence}} \times (C_{\text{failure}} + C_{\text{obsolescence}}) \quad (1)$$

For this case study and considering the general nature of the software risk assessment two key attributes will be used namely: criticality of software and likelihood/rate of software obsolescence occurring. Table 1 shows the characteristics for the software criticality features, which have been ranked as part of the case study.

Table 1. Criticality ranking

Criticality	5-1 value
Safety i.e. cyber security	4
Functionality	3
Cost	5
Complexity level	5
Commonality	3
Average	4

Top down values from 5 to 1 are assigned to each software type against a list of departments involved i.e. engineering/IT, finance, procurement and logistics, where 5 is the most-critical and 1 is ranked as least-critical.

The type of software used by participating oil and gas industry is ranked against the implications of obsolescence. Table 2 below explains the likelihood attribute and follows what was used above on criticality to address likelihood.

Table 2. Likelihood ranking

Likelihood	5-1 Value
Mean time before failure (low)	3
In-service data availability	4
Upgrading/update cycle	3
Tailor made software program	5
COTS	3
Average	3

The likelihood and impact information is translated into a traffic light type decision support mechanism.

4.3. Obsolescence mitigation approaches

There are a number of software obsolescence mitigation approaches as observed in the case study:

- License downgrade —usually there exists a type of memorandum of understanding with software suppliers to allow the end user to extend the authorization to the use of older versions by procuring additional licenses of the newer edition and then applying the acquired license on the older versions;
- Source code purchase—here the patent holder may choose to authorize the end user to buy the source code for the software in this case usually it is embedded in the original contract;
- Third Party Support involves when a third party is authorized to continue to support the obsolete software program.
- Redevelopment: the software program is upgraded to be adapted with the newer version usually done by retesting the software to realigning which includes data porting, reconfiguration, revision and retraining.
- Re-qualifying: here legacy software packages may have to be subjected to fresh testing and requalification in the offshore oil and gas operational environment to validate workability.
- Re-hosting: in this case however, existing software is upgraded to adapt to a new development environment via technology porting. This may only be used for legacy software packages programmed specifically with languages and systems that are obsolete.
- Media Management: software archive is a key ingredient in software obsolescence, especially the method of storage and maintenance. This process faces some other difficulties like costs involved, type of media, and version control.
- Case resolution: Software obsolescence resolution costs should incorporate tracking different resolution metrics, version control and data-base management.

4.4. Obsolescence management challenges

Commercial off-the- shelf (COTS) usage is broadly employed with minimal control on the supply chain in the market. By this obsolescence problem has further been increased where the interest is to maintain long life system over many decades which is at variance with COTS software providers whose main interest is to keep the life cycle of their software lower as a marketing strategy.

The case study identified that the main challenge of software obsolescence is that it is generally overlooked, especially in obsolescence management plans (OMP), and when it is mentioned, it is usually only brief. This is even worse in the offshore oil and gas industry where in many cases there is not even an OMP. However, hardware i.e. EEE

components are the main focus and software obsolescence received minimal management strategy.

Beyond the lack of awareness of software obsolescence in the offshore oil and gas industry, there is the lack of tools (i.e. obsolescence monitoring tools like those used for EEE component QinetiQ/IHS) problem to address software obsolescence, making it almost impossible to forecast software obsolescence issues.

To address this issue there is need to raise the awareness level in the offshore oil and gas industry of software obsolescence. The cost of addressing it during the life cycle is equal or even higher when compared to hardware components.

5. Prototype tool for software obsolescence management

The focus of the prototype tool, embedded in MS Excel, lies in managing software obsolescence. One of its advantages is the ability to create a platform for systematic software obsolescence risk mitigation. It is a simple tool and flexible to be adapted. The following are the key steps and parameters used:

- Step 1: identify software type
- Step 2: identify the cause and effect of obsolescence in software types through fishbone diagrams
- Step 3: identify the impact of software obsolescence
- Step 4: risk exposure register
- Step 5: resolution qualification
- Step 6: finalized software obsolescence management plan

5.1. Identification of software obsolescence

The software can either be micro-processor or computer / programmable logic controller. The identification focuses on determining the relevance of each of these types to the system of interest.

5.2. Identification of cause and effect of software obsolescence

This step aims to establish a link between software and the knock on effect on other software and hardware (i.e. application code to expertise). From step 2, the likelihood of the elements impacting each other can be determined.

Micro-processor may be influenced by and/or have an impact on: support chips, hardware connectivity, software connectivity, expertise / skill, supplier, programming device, operating system, or application code. Through interaction with the case company further detailed reasoning for building relationships to identify the cause and effect was established.

Computer / programme logic controller may be influenced by and/or have an impact on: supplier, programming device, operating system, application code, expertise /skill, hardware connectivity, or software connectivity. Compared to micro-processor a link between support chips and computer / programme logic controller cannot be formed.

5.3. Impact of software obsolescence

In this step the likelihood of impact on cost and labour is established. Here the implication of the likelihood of software obsolescence occurring impact whole life cycle cost on the risk register. The cost impact is further divided into three areas as follows:

- Labour / hour(s)
- Material employed
- Loss of production

All of the above are to determine the complexity level which helps to analyse the degree of risk exposure. If the resolution approach involves labour/hour, material and loss of production this indicates a high risk software obsolescence. But if scaled down to labour/hour and material alone it indicates a medium risk. However, when labour/hour is all that is required to resolve the software obsolescence issue this is termed low risk. Figure 4 presents a snapshot of the framework where the focus is on identifying the impact of software obsolescence.

The screenshot shows a software impact assessment tool for a 'Computer / Programme Logic Controller'. It is divided into two main sections: 'Application codes' and 'Programming device'. Each section contains a table with columns for 'Likelihood', 'Impact (cost/labour)', and 'Risk Exposure'. The 'Impact' column is further subdivided into 'Labour', 'Material', and 'Loss of production'. The 'Risk Exposure' column is also subdivided into 'High', 'Medium', and 'Low'. The tables are color-coded with red and blue headers and rows.

Fig. 4. Software obsolescence impact assessment

5.4. Risk exposure register

A guideline is provided using a predefined list of mitigation strategies which asks some probing questions like “do you want to include this strategy?”. All answers to the probing questions are collected and synthesized through the excel tool. The collected result is a classification of various risk levels of all software types deployed or employed in the oil and gas industry. The high risk items are singled out for mitigation thereby giving room for the oil and gas industry to proactively address software obsolescence impact rather than reacting. Furthermore, this proactive approach is seen as vital by all levels of management. The traditional approach of reacting after the effect has proven to be very costly.

5.5. Resolution quantification

The resolution strategy is qualified against cost and appropriateness. Questions of interest include: how software obsolescence issue is addressed effectively reducing labour, material and loss of production cost, at what percentage increase will the resolution strategy decrease cost etc. Cost change composition has been added to help decision makers to manage associated risk.

5.6. Finalised software obsolescence management plan

The roadmap tool at a glance indicates the risk exposure level which informs proactive management of software obsolescence. This requires the involvement of various departments in the participating organisation i.e. procurement, IT, engineering, etc. to work together and ensure the effectiveness of this tool. As an outcome, the management team can adopt a sustainable process and also can push for a change in organisational culture to take cognizant of software obsolescence management plans. This also can be put forward to facilitate the communication regarding software obsolescence across the supply network.

6. Validation and verification

The validation focuses on software obsolescence management within the in-service phase which also considers the bidding stage. The validation was carried out in collaboration with the EPC partner and two participants were involved: Obsolescence Lead and Product Life Cycle Lead who has 15 years of experience and 3 years of top management position in obsolescence management issues.

A series of semi structured interviews, which totaled four hours, were conducted to review a number of aspects related to the decision support tool presented. The Obsolescence Lead agreed that the logic of the cost estimating process is valid. Furthermore, it was highlighted that the framework is in particular suitable for the bidding stage for early decision making when limited data is available. The Product Life Cycle Lead asserted that the use of the tool was intuitive and used a suitable language to avoid confusion and misunderstanding. Also, he expressed that the tool provides continuous monitoring of the product life cycle especially when developing products and solutions. The Obsolescence Lead stressed that tool provides a generic overview of risks faced and the resolution options. He also highlighted that the framework would easily adapt to other product types i.e. hardware and none electrical based products.

7. Conclusion and future work

This research presented in this paper is novel in the oil and gas industry. This is in particular due to the traditional approach to obsolescence management is reactive and there is a lack of emphasis to take account of software obsolescence. The framework for software obsolescence management offers benefit with addressing software obsolescence risks and will enhance the evaluation process quicker as it aids in providing data for comparison of solutions and forces questions to be asked and evaluated where applicable. The research focused on software obsolescence because of the scale of the challenge and urgency of industrial support i.e. hardware, equipment etc. upgrade to meet present energy demand and operational demand. However, the following concludes this research by offering some recommendations for future initiatives and research:

- The oil and gas industry should develop the obsolescence awareness within the industry via different forums.
- Set up monitoring units in various departments especially software to proactively observe COTS software applications employed in key system and system development environment.
- Keep track of incremental upgrades so as to know when to initiate such inclusion into key environments or to find an alternative mitigation strategy i.e. life cycle extension via a third party.

Acknowledgements

This project was sponsored by an industrial partner from the oil and gas sector. The authors would also like to thank the EPSRC Centre in Through-life Engineering Services at Cranfield University.

References

- [1] Feldman K. and Sandborn P. Integrating technology obsolescence considerations into product design planning. In: ASME/IEEE International Conference on Mechatronic and Embedded Systems and Applications, presented at - 2007 ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, 4-7 September 2007, Las Vegas, Nevada, IDETC/CIE, IDETC/CIE 2007; 4: 981-988.
- [2] Waldman M.. Planned obsolescence and the R&D decision. *Rand J Econ* 1996; 27(3): 583-595.
- [3] Romero Rojo (2011). Development of a framework for obsolescence resolution cost estimation. PhD Thesis, Cranfield University, Cranfield, UK.
- [4] IEC 62402 - Obsolescence management – Application guide, Online: http://webstore.iec.ch/preview/info_iec62402%7Bed1.0%7Db.pdf
- [5] Cretenet A. Obsolescence of electronics potential impact on subsea control from an operators standpoint. In: Subsea control and data acquisition: experience and challenges, Conference Data Services, Pau, France, 2004; 55-60.
- [6] Aker Solution. 2009. *Managing obsolescence and new technology requirement*. Online: http://www.sut.org.au/perth/perth_events/Aker_Solutions_Low_power_S_EM_managing_obsolescence_060809.pdf
- [7] Sandborn P. Prabhakar V. and Ahmad O. Forecasting electronic part procurement lifetimes to enable the management of DMSMS obsolescence. *Microelectron Reliab* 2011; 51(2): 392-399.
- [8] Merola L. (2006). The COTS software obsolescence threat. In: Proceedings of the fifth international conference on commercial-off-the-shelf (COTS)-based software systems, IEEE Computer Society Washington, DC.
- [9] Sandborn P. Software obsolescence-complicating the part and technology obsolescence management problem. {IEEE} *Trans. Comp., Packag., Manufact. Technol.* 2007; 30(4): 886-888.
- [10] Rojo FJ R. Roy R. and Shehab E. Obsolescence management for long-life contracts: state of the art and future trends. *Int J Adv Manuf Tech* 2010; 49(9-12): 1235-1250.
- [11] Fishman A, Gandall N and Shy O. (1993). Planned obsolescence as an engine of technological progress. *J Ind Econ* 1993; 41(4): 361-370.
- [12] Josias C, Terpenney JP and McLean KJ. (2004). Component obsolescence risk assessment. In: Proceedings of the 2004 Industrial Engineering Research Conference (IERC), 15. Amherst MD
- [13] Parkinson P. (2008) Managing Software Obsolescence through Standards. In Component Obsolescence Group, Duxford, 5th February 2008.
- [14] Gowland A. (2002) Managing Software Obsolescence in the COTS Environment. Radstone Technology 2002