

The Fourth International Conference on Through-life Engineering Services

Towards Software Performance Monitoring: An Approach for the Aerospace Industry

Raúl González Muñoz^a, Essam Shehab^{a*}, Martin Weinitzke^b, Rachel Bence^c, Chris Fowler^c, Paul Baguley^a

^aDepartment of Manufacturing, School of Aerospace, Transport and Manufacturing, Cranfield University, Cranfield, Bedfordshire, MK43 0AL, United Kingdom

^bAirbus Operations, Hamburg, 21129, Germany

^cAirbus Operations, Filton-Bristol, BS99 7AR, United Kingdom

* Corresponding author. Tel.: +44 79 50554 084. E-mail address: e.shehab@cranfield.ac.uk

Abstract

Software applications are becoming one of the most valuable assets for companies, providing critical capabilities and functionalities to perform a wide range of operations in the industry. This paper aims to provide a view on software application portfolio monitoring and its integration into business intelligence systems for aerospace manufacturing companies. The key research question addressed is how critical software has become for aerospace industry and how software applications could be monitored. This question has been addressed by conducting an in depth review of current literature and by interviewing professionals from different aerospace companies. The results are a set of key findings regarding software impact in aerospace industry, and a monitoring proposal based in a traditional business intelligence architecture.

By incorporating condition monitoring methodologies into the software application portfolio of the enterprise, benefits in maintenance budget allocation and risk avoidance are expected, thanks to a more precise and agile way of processing business data. Additional savings should be possible through further application portfolio optimisation.

© 2015 The Authors. 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/4.0/>).

Peer-review under responsibility of the Programme Chair of the Fourth International Conference on Through-life Engineering Services.

Keywords: Software Lifecycle; Condition Monitoring; Business Intelligence; Software Maintenance; Aerospace

1. Introduction

During the last decades software has been acquiring an increasingly important role in how industry works, providing critical capabilities and functionalities to companies from a wide range of fields.

Nowadays most of the projects, if not all of them, within manufacturing industry, make usage of software with different degrees of complexity and dependencies. Software is defined as programs, procedures, rules, data and documentation associated with programmable aspects of system hardware and infrastructure [1].

The current industry landscape, as a result of this technological change, offers new opportunities for competitive

advantage, where integration and responsiveness are key aspects, driving the value chain of the organisations involved [2].

Aerospace industry, as one of the major players in the manufacturing field, has been actively integrating Information Technologies in their business processes, being commonly an early adopter of those technologies within the manufacturing scene.

Although several research and studies have been undertaken on condition monitoring for engineering assets, there is a lack of research projects on monitoring of software application portfolios and how these methodologies could be integrated within the information systems of the organisations.

2. Methodology

The research methodology involved three main phases. The initial phase focused in literature review within the fields of software lifecycle, application portfolio management and business intelligence. At the same time, a set of face to face formal interviews was conducted with industry experts from different aerospace companies.

The second phase involved the identification of the software criticality in aerospace industry, derived from both the literature review and the interviews.

The third and final phase concerned the development of a proposal for software application monitoring, based in previous research and valuable information gathered from industry experts. This was presented and validated at a final interview with a key expert from industry. These phases are illustrated in Fig. 1.



Fig. 1. Schematic Depiction of the Research Methodology

3. Application Portfolio Lifecycle

Application portfolio refers to an organization's collection of software applications and software-based services, which it uses to attain its goals or objectives.

From the perspective of aerospace companies, as well as most of the manufacturing industry, software lifecycle is understood mostly in operative terms, as “for IT operating budgets, enterprises spend two-thirds or more on ongoing operations and maintenance” [3]. Hence, such companies will be exposed to software mainly during its service period, which is once the applications are deployed into the target environment as part of a portfolio, providing support to different business activities [4].

Development phase is usually not a focal point of attention in this case, as most of the software being used by aerospace companies is commercial off-the-shelf (COTS), although defense field is a bit more balanced in this aspect [5,6]. Governance, which involves making sure the applications meet the business needs, has also a relevant role, though Operations is still the most relevant aspect.

As illustrated in Fig. 2., the main activities performed during the operations phase are the deployment of updates, being these small improvements or bug fixes, and the monitoring of the application performance. Many aerospace companies outsource part of these activities to specialised contractors, which are the ones dealing in technical terms with software updates.

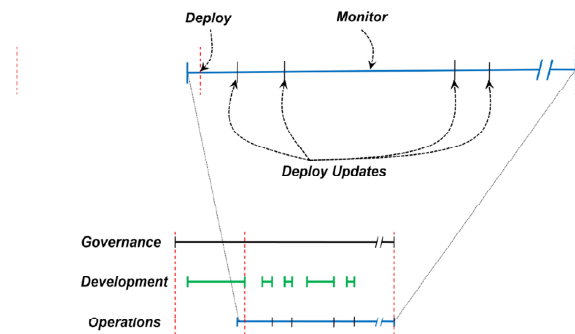


Fig. 2. Application Lifecycle Management. [4]

The importance of applying an effective monitoring strategy over the application portfolio has become far more essential now, as many business processes require the correct performance of specific software. At this point, the failure of an application could cause business disruption [7,8], having this a high cost for aerospace companies, in terms of money, time and image.

However, monitoring strategies should take into account the nature of software application portfolios. Software applications present a high number of dependencies among them as well as with other business assets, resulting in a system whose complexity will depend on the particularities of the applications the business requires [9].

As for system, it could be defined as “a combination of interacting elements organized to achieve one or more stated purposes” [10] or “a set of elements standing in interrelation among themselves and with environment” [11]. Hence, regarding the monitoring of software portfolios there is the need to control the underlying interactions among the applications. Failures in specific parts of the system could create unexpected cascade effects and put in risk the capabilities of the whole system, consequently causing delays or even suspension of business activities.

From an operational point of view, when a software application is deployed for the first time into the target environment, its starting capabilities will not be at its full potential. There will be an adaptation period of time to the working environment, in which the software applications are being deployed. These constraints can originate due to two main causes:

- The software application has been developed and launched recently, hence it is an early version the one being deployed. Thus, bugs and irregular performance are expected for a period of time.
- Adjustment to the set of intrinsic dependencies of the new system that is the working environment. Considering not just software applications within the portfolio, but other interacting assets as well.

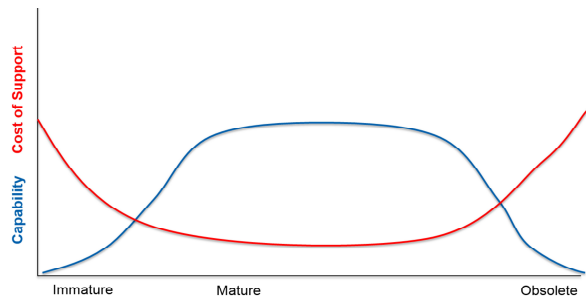


Fig. 3. Application Capability Cycle

After this early period, the capabilities of the software should increase with time, as shown in Fig. 3., until reaching eventually its full potential, providing a reliable performance and meeting the business requirements, hence being considered mature.

Nonetheless, the capabilities of an application may be changing through time, due to external conditions, obsolescence risk or the request of new features from the business.

Regarding software obsolescence, some people would argue that software applications cannot become obsolete as they are not affected by degradation (and hence do not require replacement) and can be easily replicated. Their misunderstanding is to try to employ the same reasoning to software obsolescence as to mechanical or electrical component obsolescence. It is required to comprehend the different nature of the software obsolescence issue. The significance of obsolescence is that it prevents from maintaining and supporting the system, hence creating a risk of disruption on the operations [12].

Depending on the capability status of the software application, its cost of support will vary, requiring a bigger effort while its immature or the obsolescence risk takes place. Due to the relevance of the financial figures dedicated by companies to deal with software portfolios maintenance [3], it is essential to understand what is the evolution of the software applications. This will enable organisations to provide the best possible support to business operations, avoiding or mitigating disruption.

4. Software Monitoring

As mentioned earlier, an application portfolio is a system with a large number of components and dependencies. Thus, a monitoring mechanism is needed in order to provide the capabilities required to gather, store and assess data coming from all the application portfolio. For that purpose, data warehousing techniques need to be used.

The usage of these procedures will provide the foundations and serve as building blocks for an integrated information system, which would provide valuable business intelligence information.

Business intelligence refers to systems that “combine data gathering, data storage, and knowledge management with

analytical tools to present complex internal and competitive information to planners and decision makers” [13].

To elaborate business intelligence information, the integrated data is searched, analysed, and provided to the decision maker. Regarding structured data, analysts use Enterprise Resource Planning (ERP) systems, extract-transform-load (ETL) tools, data warehouses (DW), data-mining tools, and on-line analytical processing tools (OLAP). But a different and less sophisticated set of analytic tools is currently required to manage semi-structured data.

In the case of software application portfolios, the data is structured, which refers to “all data that does fit neatly into relational or flat files” [13].

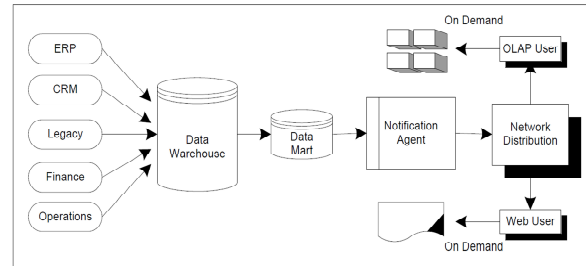


Fig. 4. Typical business intelligence architecture for structured data [13]

Typical business intelligence architecture for structured data centres on a data warehouse. The data is extracted from operational systems and distributed using Internet browser technologies, as seen in Fig. 4. The specific data required for business intelligence is downloaded to a data mart used by planners and executives. Outputs are obtained from routine push of data from the data mart and from response to inquiries from Web users and OLAP analysts. The outputs can take different forms including exception reports, routine reports, and responses to specific request. The outputs are sent whenever parameters are outside pre-specified bounds [13,14].

Such kind of systems involving the monitoring of software have been approached before, but understood as a mean to study the intrinsic complexity of information technologies [15] or to measure the performance of a software application from a software engineering point of view [16]. In contrast, this research work focuses in studying a constant monitoring strategy to assess the quality of the service provided during the application lifecycle, from the point of view of the organisation using the application, and assuming the likely inability to access the source code of COTS software.

5. Results

5.1. Current Practice and Business Impact

Most of the research around the area of software maintenance and lifecycle started in the defence field, usually involving aerospace sector but under a military approach [5,6]. However, the scope is being extended to the whole aerospace industry, and not just the military branch, due to the ample similarities in terms of development times and support

processes [12,17]. Furthermore, additional long-life assets are also the subject of research [18].

Thirteen participants from different aerospace companies and services related have been interviewed, as shown in Table 1. Participants ranged from department managers to graduates, and the work experience differed from 1 year to 35 years. Furthermore, the profiles of the participants were diverse, coming from varied educational backgrounds. This was intended in order to gather information from different points of view. Questions were asked to find out their understanding of:

- Criticality of software in aerospace industry and examples of that criticality
- Major risks in aerospace industry regarding software
- Software obsolescence
- Communication and reporting systems within the organisation
- Business intelligence and centralized information systems and its implementation in industry

Table 1. Experiences of participants.

Participants	Experience	Role
Participant 1	13 years	Service Operations
Participant 2	6 years	Service Manager
Participant 3	8 years	Systems Designer/Project Manager
Participant 4	3 years	Avionics System Engineer
Participant 5	1 year	Production Graduate
Participant 6	5 years	Project Manager
Participant 7	14 years	Application Health and Prevention Manager
Participant 8	1 year	Project Manager
Participant 9	35 years	System Engineering Project Manager
Participant 10	20 years	Service Pack Manager
Participant 11	3 years	License Management
Participant 12	30 years	Security and Obsolescence Manager
Participant 13	20 years	Service Pack Manager Delegate

Almost all the participants agreed that software applications were becoming more critical for the business activities. Only participant 9 was not very sure if the change through time was that clear. Participant 7 pointed out that even though software is more critical in general terms nowadays, the criticality changes within each aerospace project depending on its different phases. During the development phase the criticality focus is in the engineering applications while in the operations phase the focus would move to manufacturing applications.

Participants 3, 6, 10 and 11 mentioned several examples in which, the failure/malfunction of a certain software or the hardware that was supporting it, caused business disruption, generating the risk of delays in different projects, extra costs or security risks.

As for the major risks of software in aerospace industry, the participants pointed, by importance:

- Security: Either related to cyber-attacks and intellectual property protection or business disruption.
- Archiving/Obsolescence: Due to the long life of aerospace assets, there is the need to maintain a wide sort of data, files and software available for access. Big migration projects are performed in the sector periodically for this purpose.
- Complexity of dependencies among the system: The high amount of interactions and dependencies among software applications and other business assets may cause unexpected cascade effects in the system.

On the subject of software obsolescence, most of the participants had a quite accurate view on the topic. However, participants 1, 3, 8 and 11 had a more basic understanding of obsolescence, at least concerning the software field.

Regarding integration and communication across the departments, participants stated that the communication was acceptably good. This is partially due to the international nature of aerospace business operations. However, participants remarked that there was still silo effect in some aspects, largely due to bureaucracy, business perception too much based in own department area, and different geographical location.

With respect to the understanding of Business Intelligence, it was not very well known among the participants. Just half of the participants had a proper view on this matter. However, some of the participants had a perception on big data or data warehousing which corresponded quite well with the definition of business intelligence.

Lastly, concerning monitoring systems based in big databases, more than half of the participants perceived the value that would bring to the business, and would like to have such tool available. Yet, participants expressed concerns regarding the methodology used within that system to make the tool feasible, as well as to convince the business of the value provided by this system, in order to secure budget for its development and support.

As a result from these interviews, the key findings were as follows:

- Software is very critical for aerospace industry, having a big impact into the business operations
- Security, Obsolescence/Archiving and System dependencies are major risks for software in the aerospace field
- Data integration across different departments of the organisations is a present challenge
- There is a value perceived by a majority of professionals in the development of big monitoring systems, based in data warehousing/business intelligence

5.2. Proposed Approach

Currently, there are standards regarding the evaluation of software applications in terms of reliability, quality and service requirements [19,20,21]. However, these methodologies were designed to be a one-time assessment, not a proper constant monitoring process.

As a result of the study of the literature review and the interviews, a proposal for an approach to the monitoring of software applications, in conceptual terms, has been developed and illustrated in Fig. 5.

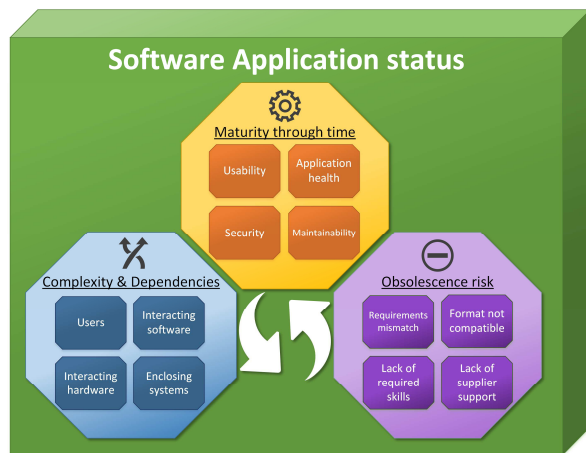


Fig. 5. Software application monitoring approach

Software applications should be monitored and evaluated based on three main aspects, namely maturity through time, obsolescence risk and complexity and dependencies.

- The maturity would be evaluated in terms of the usability of the application, how easy it is to maintain, the amount of incidents it causes and its security level.
- The obsolescence risk would be controlled through the analysis of the suppliers, the required skills to support it, the change of business requirements and the format of the data
- The complexity would be assessed based in the number of interacting software and hardware, the number of users and the number of sub-components within the applications

In order to elaborate a feasible monitoring strategy out of these three areas, a set of specific metrics, able to assess the available data, should be developed. These metrics should be able to be measured periodically in reliable terms and with a minimum effort from the organization.

Further integration of the model into big data warehouses would provide the required capabilities to automate the monitoring process. One of the main reasons for this approach is the diversity of the application portfolio, being the data of the applications gathered from different sources. An overview of the structure of this approach is shown in Fig. 6.

To provide an effective monitoring on the application portfolio, data related to the applications behavior, such as number of incidents, number of users, availability of the documentation, time between failures or security compliance, is gathered and stored in a data warehouse. That data is filtered afterwards for assessment purposes, in terms of maturity, complexity and obsolescence. An assessment, based in the allocation of certain scores within an agreed scale, is then performed, measuring each of the established

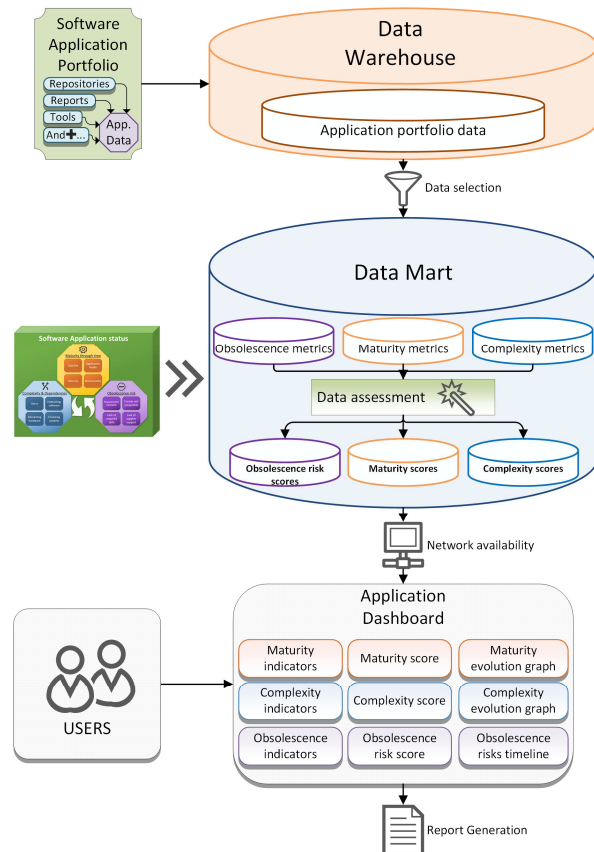


Fig. 6. Proposed monitoring system structure

metrics/indicators within maturity, complexity and obsolescence. As a result, a set of scores is provided, being the sum of all of them a final high-view unique score for each application. Those set of scores are delivered in the shape of a user interface, supporting the diverse information with graphs and visual aids. From this dashboard, the creation of reports in several formats should be possible, based on the requirements of the users.

As a result of this final integration, aerospace organisations would obtain an automated system, capable to assess the status of the software applications within the portfolio, in the aspects of obsolescence, complexity and maturity, during all the application lifecycle.

6. Conclusions and Further Work

Software applications are becoming increasingly critical for aerospace industry; every system and every manufacturing technique is supported by software. It is essential for the industry to establish proper monitoring strategies for software application portfolios, in order to enable a proactive approach and avoid or mitigate business disruption. This paper described, an approach to enable the monitoring of software applications, based in the assessment through the areas of maturity, complexity and obsolescence, and its possible integration into a data warehouse system. Further research and

development needs to be performed, in order to establish a detailed set of metrics based in the key indicators of software during its lifecycle. Last steps will involve integrating the resultant model with, the already mentioned data warehouse, automating the monitoring process. This will allow a testing and validation process, during which, upgrades and tweaks on the system are expected. Final delivery would be a software monitoring tool based in a typical business intelligence system architecture.

Acknowledgements

This research project is funded by Airbus and Cranfield University. The author would like to gratefully acknowledge the support and assistance from many staff of Airbus and several aerospace organisations during the research.

References

- [1] IEC 62402 - Obsolescence management – Application guide, Online:
- [2] Porter, M. E. (1985). "Competitive Advantage: Creating and Sustaining Superior Performance". Free Press, no. 5, pp. 164-200.
- [3] Forrester Research, 2009 "The State Of Global Enterprise IT Budgets: 2009 to 2010"
- [4] Chappell, D., (2010) "What is Application Lifecycle Management?", David Chappell & Associates, <http://www.davidchappell.com/>
- [5] Merola L. (2006). The COTS software obsolescence threat. In: Proceedings of the fifth international conference on commercial-off theshelf (COTS)-based software systems, IEEE Computer Society Washington, DC.
- [6] Rajagopal. S., Erkoyuncu, J.A., Roy, R., Software Obsolescence in Defence, Procedia CIRP, Volume 22, 2014, pp 76-80, ISSN 2212-8271.
- [7] Snedaker, S., 2013 "Business Continuity and Disaster Recovery for IT Professionals", Syngress, Second edition, no. 5
- [8] Swanson M, Bowen P, Phillips AW, Gallup D, Lynes D. "Contingency planning guide for federal information systems". 2010 [Retrieved May 26, 2013], from National Institute of Standards and Technology.
- [9] Waite, E.J., AIT — Advanced Information Technology for Design and Manufacture K. Kosanke, J.G. Nell (Eds.), Enterprise Engineering and Integration: Building International Consensus, Springer-Verlag, Berlin (1997), pp. 256–264
- [10] ISO/IEC 15288:2002, Systems engineering - System life cycle processes.
- [11] Bertalanffy, L.: General System Theory – Foundations, Development, Applications. 14th edition 2003, George Braziller Inc. New York, 1969, p. 252.
- [12] Romero Rojo, F.J., Roy, R., Shehab, E., Cheruvu, K., Blackman, I. and Rumney, G.A. (2010). Key Challenges in Managing Software Obsolescence for Industrial Product-Service Systems (IPS2). The 2nd CIRP Industrial Product-Service Systems (IPS2) Conference, Sweden, 14th–15th April 2010.
- [13] Solomon, N., (2004) "Business Intelligence", Communications of the Association for Information Systems: Vol. 13, Article 15
- [14] Ranjan, J., (2009) "Business Intelligence: Concepts, Components. Techniques and Benefits", Journal of Theoretical and Applied Information Technology: Vol. 9, No. 1.
- [15] Leukert, P., Alliet, B., Vollmer, A., and Reeves, M., 2011 "IT Complexity metrics – How do you measure up?" The Capco Institute Journal of Financial Transformation, vol. 34, pp. 11-15.
- [16] van Hoorn, A., Waller, J., and Hasselbring, W. 2012. Kieker: a framework for application performance monitoring and dynamic software analysis. In *Proceedings of the 3rd ACM/SPEC International Conference on Performance Engineering* (ICPE '12). ACM, New York, NY, USA, 247-248.
- [17] Romero Rojo, F.J., Roy, R. and Shehab, E., Obsolescence Management for Long-life Contracts: State of the Art and Future Trends. International Journal of Advanced Manufacturing Technology, 2009, Volume 49, Issue 9-12, pp1235-1250
- [18] Erkoyuncu, J.A., Ononiwu, S., Roy, R., Mitigating the Risk of Software Obsolescence in the Oil and Gas Sector, Procedia CIRP, Volume 22, 2014, Pages 81-86, ISSN 2212-8271.
- [19] ISO/IEC 25010:2011, "Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuARE) – System and software quality models".
- [20] Kaur, G., Bahl, K., Software Reliability, Metrics, Reliability Improvement Using Agile Process. IJISSET, Volume 1, Issue 3, May 2014.
- [21] Quyoum, A., Din Dar, M.U., Quadri, S.M.K, Improving Software Reliability using Software Engineering Approach- A Review. International Journal of Computer Applications, Volume 10, No.5, November 2010.