2008

The Impact of Operating System Obsolescence on the Life Cycle of Distributed Teams

Herbert J. Mattord
Nova Southeastern University, mattord@nova.edu

Tridib Bandyopadhyay
Kennesaw State University, tbandyop@kennesaw.edu

The Impact of Operating System Obsolescence on the Life Cycle of Distributed Systems

Herbert J. Mattord, CISM, CISSP
Nova Southeastern University
mattord@nova.edu

Tridib Bandyopadhyay
Kennesaw State University
tbandyop@kennesaw.edu

ABSTRACT

Operating System obsolescence is widely considered an important factor when architectural choices are made during the planning phase of systems development and maintenance. In this work, we seek to understand the importance that planners actually ascribe to this attribute in practice, and ask probing questions to the managers, developers, and analysts of systems in industry in the form of a survey. Initial results suggest that obsolescence is neither perceived as a critical factor during planning for systems development, nor is it viewed as a major contributor to the total cost of operation of client/server systems. However, the survey does identify that organizations have a number of valid and functional coping strategies when obsolescence does affect systems operations.

Keywords

Obsolescence, operating Systems, software aging, distributed systems, coping strategies

INTRODUCTION

Standard technological, economic and accounting prudence suggest that ‘obsolescence’ of a system is an important consideration during the selection and design phase. Among others, risks of obsolescence manifest in technologically orphaned/deprecated components where vendor support is less available until it is no longer offered. This may be found to result in reduced compatibility and higher likelihood of security vulnerabilities. Information systems developers should also be concerned with Operating System (henceforth OS) obsolescence since it could affect the total cost of ownership for distributed systems. Insofar as information systems are mandated to provide efficient, effective, robust and reliable data/information processing capabilities, obsolescence is an issue deserving high attention [Whelan 2002]. However, general observation indicates that obsolescence of OS is a low priority consideration while choosing the platform components of a distributed system in practice. While it is not known with certainty which strategies are most effective in dealing with OS obsolescence, discovering what is the current practice is a good first step towards identifying reliable approaches to combat such OS obsolescence. The objective of this study is to explore i) the current degree to which eventual obsolescence influences platform selection in distributed systems at the beginning of their life cycle (design phase) and ii) the coping strategies (if any) employed by the organizations when distributed systems’ maintenance encounters obsolescence issues, as systems near the end of their planned existence.

Utilizing a survey of 77 IT/IS practitioners and academics, we show that i) there is an awareness of the issue of O/S obsolescence, ii) a high percentage of the respondents do not think that that O/S obsolescence is a critical factor in the development and operation of client/server and distributed systems, and iii) those organizations which have been affected by O/S obsolescence have well-developed coping strategies to deal with the issue. In what follows, we first review the extant literature on obsolescence issues, and then report the survey methodologies, target group, and our results. Finally we discuss the insights from the survey, and provide our conjectures and directions for future research.

EXTANT RESEARCH

The platform selection in the design and development of distributed applications is often approached from the current organizational architectural framework: organizations seek to maximize the utility of their IT investments [Whelan 2002]. This is often observed as a directive to developers to create systems that use current infrastructure options when developing and/or maintaining client server systems. While choosing the various components that comprise a distributed system the choice of platforms both influences and is influenced by the other components. The components of a distributed system in addition to platform selections are 1) application systems and the services they deliver, 2) networks made from network hardware devices and the network operating systems and protocols on which they depend, and 3) middleware software and services that enable the communications between the parts using the network protocols which enable system functionality [Coulouris, Dollimore, and Kindberg 2005]. All of this is occurring in a context that is usually aware that systems are envisioned, created, deployed, operated maintained and then retired in a cyclical life-cycle which is true for most systems but, for which there are exceptions.

Criteria for selecting components in agent-oriented distributed system are interoperability, usability, reliability, scalability, and performance [Biswas 2007]. Not mentioned directly, but inferred in the same discussion is the need for technical currency with peer components, which, once extended, asks one to consider the susceptibility to threats from obsolescence. The elements of distributed systems are mostly made up of software-based elements. Software is not immune from failure as it ages its way from creation to retirements [Jones]. Coupled with the use of wrapping approaches that conceal aging host-based application in seemingly modern server objects that provide application support to distributed systems, this can pose emergent issues to unprepared system maintenance professionals. Concern about platform obsolescence is not limited to application compatibility and aging either. While not directly a part of the operating systems, CORBA (once a dominant force among the early adopters of client/server systems) is a closely related software component and the experience of this aging software element is informative as to how obsolescence can impact distributed systems (Henning 1996).

The persistent relevance of the local operating system remains high [Lepreau, Ford, and Hibler 1996], and operating systems remain an integral component in the development and functioning of the client/server biosphere. The platform, made up of the system hardware and the operating system (plus enabling middleware) serves as the foundation for all client/server functionalities at the client. While the need for an operating system remains critical, the need for any specific operating system has come to become less so: the evolutionary path of the two dominant personal computer (PC) operating systems has seen the realization of the ubiquitous personal computer platform. Actually, availability and platform compatibility of commercial off-the-shelf systems (COTS) has become a higher determinant than the needs of the client/server system designer in reality.

The capability of operating systems has changed over time as well [Peng, Li and Mili 2007]. It is useful to note that in the context of distributed systems, we need to differentiate the platform among the individual aspects, viz. hardware, and operating system version. For example, the capabilities of each component in a platform using Intel processor and UNIX operating systems tend to evolve over time. A distributed system is designed to work with specific platforms, under a specified hardware environment and a specified operating system (even to the level of the version). Each of these elements can evolve over time (in varied degrees), and since operating system and hardware manufacturers are known to deprecate supported versions over time, it is entirely possible for a system to come to rely on platforms where one or both of its elements are unsupported by the manufacturer.

Traditionally, the impact of obsolescence has been handled in the field of economics. Langlois provide a treatise on the basis, impact, and manifestation of obsolescence in business [Langlois 2007]. Detailed ramifications of technological obsolescence in the broader areas of industry have been explored in numerous works [Williams 2002, Slade 2006, and Kressel 2007 among others].the survey of practitioners

We undertook a survey project in order to gain a first hand appreciation about i) how practitioners view OS obsolescence, and ii) how they strategize to cope with its impacts. Towards that, we created[1] a survey instrument designed to collect basic demographic information and organizational facts/initiatives about OS obsolescence, but no personally-identifiable information. After requisite IRB approvals, the survey was offered to a community (privy to the authors' network) in the social-networking service LinkedIn.com to collect input from self-selected respondents. Our guiding goals in this survey were two-fold: ascertain the current awareness/understanding among practitioners as to how operating systems obsolescence affects the complete life cycle of distributed systems, and what strategies, initiatives, and/or controls (if anything), are currently in place to ameliorate the impacts of operating system obsolescence during the maintenance phase of existing distributed systems. In the following subsections, we provide a summary account of the survey including the demographics and organizational roles of the respondents, and their responses.

Respondents

We solicited a group of approximately 260 peers from the social-networking site LinkedIn.com to participate in the survey through a link to SurveyMonkey.com, where we hosted the survey instrument. While 77 individuals made at least an initial

[1] We have utilized the available toolsets at SurveyMonkey.com.

connection to the survey, 63 chose to complete the majority of significant questions. It should be noted that the final section of the survey instrument which asked questions about coping mechanisms received responses from only 45 individuals. Although a response rate of 29.6% is quite strong, we do caution that the size of this self-selecting group was not sufficiently large, and that the targeted groups were not sufficiently diverse to dispel all concerns about reaching any form of generalized conclusion. Instead, we suggest that this survey assessment be treated as a peek into the complex topic of OS obsolescence in the design and maintenance of the life cycle of distributed systems, and also as a stepping stone towards developing a more complete instrument, surveyed on a more expansive and statistically robust participant pool.

The first question in the survey was meant as a qualifying gateway to limit responses to those who are engaged in systems design, development or support of client/server or distributed systems. Of the 77 respondents to the survey, 60 were qualified to continue the survey (Table-1). 19 of these 60 respondents or roughly 32% declared themselves to be architects or developers[2]. 58 individuals indicated that they were mostly from medium and large organizations (Table 2).

Table 1.
Respondent's Role

Role	Number	Percentage
A systems architect or designer of client/server or distributed systems	9	11.7%
A developer, programmer or software engineer involved in the creation of client/server or distributed systems	10	13.0%
A technical professional engaged in the support of client/server or distributed systems	21	27.3%
A manager of IT or Information Systems with responsibility for client/server or distributed systems	20	26.0%
None of the above	17	22.1%

Table 2.
Organization Size

Size of Organization	Number	Percentage
1-10 employees and/or contractors	6	10.3%
11-100 employees and/or contractors	10	17.2%
101-1,001 employees and/or contractors	15	25.9%
1,001-10,001 employees and/or contractors	8	13.8%
10,001-100,000 employees and/or contractors	9	15.5%
Over 100,000 employees and/or contractors	10	17.2%

When assessed based on industry, the largest segmented represented amongst the respondents was the High-tech category (31.6%), the second-most identified industry was that of University education with 8.8%, and the third-place in industry affiliation was a tie between Consumer Product and Insurance and Financial at 7% of responses each. All other responses were spread over nine additional categories.

Planning Factors for Systems

We asked respondents to identify their organizations' criteria towards selecting platforms used in client/server or distributed systems. The intent of this question was to collect an unbiased assessment of the relative importance of obsolescence among other criteria in the platform selection process. The survey preamble and all solicitation materials referred to a study on the factors that affect the client/server development process. The criteria for inclusion as a selection choice in the survey were synthesized from Coulouris, Dollimore, and Kindberg (2005), and from Biswas (2007), and the participants were asked to select only one level of importance for each of the criteria listed. The selection options were developed to provide the largest possible range of factors. A detailed presentation of the survey responses (factors) is show in Table-3.

The most frequently cited, and the most significant criterion in platform selection (based on weighted factor analysis) is *reliability*. 40 of the 51 respondents, (representing over 78% of responses) identified reliability as a 'critical' factor. Reliability also stood alone as the premier factor for consideration with 100% of respondents considering it critical/important. The second cluster of responses included *functionality, performance, securability* and *scalability*, and received 29 'very important' or 'critical' responses and was considered at least moderately important factors by all but 1 respondent. A third cluster of responses that included *interoperability, modifiability/extensibility,* and *reusability* was also considered 'moderately important' by most respondents but did not receive the degree of criticality of the earlier 2 factors, and was considered 'a minor factor' or 'not a factor' by a sizeable number of respondents. A fourth grouping of factors, made up of *autonomy,*

[2] This may be a limiting factor in terms of the collective perspective on how obsolescence influences the design and development process

susceptibility to *obsolescence*, and cooperative task processing were chosen as critical by only a few respondents and were noted as 'minor' or 'not a factor' by sizeable numbers of respondents.

Table 3.
Factors in Platform Selection

	Critical Factor (wt. = 5)	Important Factor	Moderate Factor	Minor Factor	Not a Factor (wt. = 1)	Weighted Factor (1-5)
Reliability	78.4% (40)	21.6% (11)	0.0% (0)	0.0% (0)	0.0% (0)	4.78
Functionality / Usability	56.9% (29)	33.3% (17)	9.8% (5)	0.0% (0)	0.0% (0)	4.47
Performance	54.9% (28)	37.3% (19)	7.8% (4)	0.0% (0)	0.0% (0)	4.47
Securability	54.9% (28)	33.3% (17)	11.8% (6)	0.0% (0)	0.0% (0)	4.43
Scalability	47.1% (24)	45.1% (23)	5.9% (3)	2.0% (1)	0.0% (0)	4.37
Interoperability	33.3% (17)	54.9% (28)	9.8% (5)	2.0% (1)	0.0% (0)	4.20
Modifiability / Extensibility	28.6% (14)	38.8% (19)	24.5% (12)	4.1% (2)	4.1% (2)	3.84
Reusability	23.5% (12)	37.3% (19)	27.5% (14)	11.8% (6)	0.0% (0)	3.73
Autonomy	5.9% (3)	45.1% (23)	29.4% (15)	9.8% (5)	9.8% (5)	3.27
Susceptibility To Obsolescence	6.0% (3)	36.0% (18)	36.0% (18)	20.0% (10)	2.0% (1)	3.24
Cooperative Task Processing	5.9% (3)	43.1% (22)	23.5% (12)	19.6% (10)	7.8% (4)	3.20
Aggregate weighted factor						4.00

Impact of Obsolescence

The next aspect of the study was to ascertain the platforms for which obsolescence is a consideration. We asked the study participants how far (critical/important/moderate/minor/not a factor) they considered obsolescence risk of the operating system as a factor for designing the system. A non-applicability (N/A) option was also available for those participants who may not have the need or use of a particular type of platform[3]. The results (Table-4) reveal that there is no significant difference in the way respondents view the impact of obsolescence on the various types of platforms. The calculated weighted factor for each platform type shows narrow variance - with highest sensitivity to obsolescence for 'database platforms' and the lowest sensitivity to obsolescence for 'authentication platforms'.

We then inquired about how obsolescence considerations influence the planning process for client/server and distributed systems in participants' organizations. We posed 2 questions: whether the respondent's organization considered obsolescence in estimating maintenance costs, and whether the participants actually believed that obsolescence had meaningful impact on the total cost of operations of client/server systems.

- Table-5 shows that obsolescence, in general, is not viewed as a critical factor in influencing maintenance costs. Once again, the distribution of responses exhibit little variation among the types of platforms (the weighted factor analysis indicate that all platform types receive only a 'moderate' impact assessment in estimated maintenance towards the system operation costs).
- Next, we asked whether (based on the respondents' experiences with client/server or distributed systems) the actual or threatened obsolescence of an operating system was a factor in the total cost of ownership over the life of existing systems. The grouped responses (Table-6) show that participants do not feel that OS obsolescence is much of a factor: an aggregated response of 2.8 on a 5.0 scale exhibit that OS obsolescence is perceived as less than a 'moderate' factor.

In essence, the impact of O/S obsolescence on the design of and planning efforts for client/server and distributed systems is not considered critical by most of the survey participants. All factors together, including OS obsolescence, have an aggregate weighted factor of 4.0 on a 5.0 scale, while that of the collective opinion of OS obsolescence is 3.24 - well below that of the aggregate level (Table-3). One interpretation of these measurements could be that OS obsolescence is somewhere between a 'moderate' and an 'important' factor, and that between these levels, the collective opinion is closer to 'moderate'.

[3] The platform choices were those articulated as essential platforms for systems development by Dennis, Wixom and Tegarden 2005.

Table 4.
Impact of Obsolescence on Platform Types

	Critical Factor (wt. = 5)	Important Factor	Moderate Factor	Minor Factor	Not a Factor	N/A (wt. = 0)	Weighted Factor (0-5)
For database platforms	25.0% (12)	37.5% (18)	20.8% (10)	12.5% (6)	4.2% (2)	0.0% (0)	3.7
For application server	16.3% (8)	38.8% (19)	22.4% (11)	16.3% (8)	2.0% (1)	4.1% (2)	3.5
For web server platforms	18.4% (9)	32.7% (16)	16.3% (8)	24.5% (12)	2.0% (1)	6.1% (3)	3.4
For client platforms	14.3% (7)	36.7% (18)	22.4% (11)	16.3% (8)	8.2% (4)	2.0% (1)	3.3
For authentication server	20.4% (10)	20.4% (10)	28.6% (14)	18.4% (9)	6.1% (3)	6.1% (3)	3.3
For other platforms	8.2% (4)	16.3% (8)	34.7% (17)	12.2% (6)	8.2% (4)	20.4% (10)	3.1
Aggregate weighted factor							3.4

Table 5.
Impact of Obsolescence on Estimated Maintenance

	Critical Factor (wt. = 5)	Important Factor	Moderate Factor	Minor Factor	Not a Factor	N/A (wt. = 0)	Weighted Factor (0-5)
For database platforms	19.1% (9)	27.7% (13)	29.8% (14)	17.0% (8)	6.4% (3)	0.0% (0)	3.4
For application server	14.9% (7)	31.9% (15)	25.5% (12)	17.0% (8)	6.4% (3)	4.3% (2)	3.3
For client platforms	14.9% (7)	29.8% (14)	21.3% (10)	23.4% (11)	8.5% (4)	2.1% (1)	3.2
For authentication server	17.0% (8)	25.5% (12)	21.3% (10)	25.5% (12)	6.4% (3)	4.3% (2)	3.2
For web server	8.5% (4)	27.7% (13)	34.0% (16)	19.1% (9)	6.4% (3)	4.3% (2)	3.1
For other platforms	8.5% (4)	21.3% (10)	23.4% (11)	19.1% (9)	8.5% (4)	19.1% (9)	3.0
Aggregate weighted factor							3.2

Table 6.
Obsolescence Impact on TCO

	Critical Factor (wt. = 5)	Important Factor	Moderate Factor	Minor Factor	Not a Factor	N/A (wt. = 0)	Weighted Factor (0-5)
For database platforms	14.6% (7)	22.9% (11)	29.2% (14)	10.4% (5)	22.9% (11)	0.0% (0)	3.0
For application server	16.7% (8)	16.7% (8)	29.2% (14)	12.5% (6)	20.8% (10)	4.2% (2)	3.0
For client platforms	14.6% (7)	18.8% (9)	27.1% (13)	16.7% (8)	20.8% (10)	2.1% (1)	2.9
For authentication server	12.5% (6)	16.7% (8)	22.9% (11)	12.5% (6)	31.3% (15)	4.2% (2)	2.7
For web server	6.3% (3)	16.7% (8)	31.3% (15)	16.7% (8)	25.0% (12)	4.2% (2)	2.6
For other platforms	4.2% (2)	10.4% (5)	29.2% (14)	16.7% (8)	25.0% (12)	14.6% (7)	2.4
Aggregate weighted factor							2.8

Table 7.
Coping with Obsolescence

Strategy	Client		Server	
	Count	Percent (multiple responses allowed)	Count	Percent (multiple responses allowed)
Rolling upgrades with thorough regression testing for server platforms as planned maintenance to system	32	71.1%	32	69.6%
Extended maintenance agreement from O/S manufacturer	16	35.6%	24	52.2%
Use open source O/S with investment in staffing resources to self maintain	7	15.6%	13	28.3%
Other	5	11.1%	8	17.4%
No provision made for O/S obsolescence	8	17.8%	5	10.9%

Coping With Obsolescence

Finally we asked how participants (and their organizations) had acted to cope with the impacts from OS obsolescence. Table-7 combines these responses and groups them in two pairs of columns: those with client, and server platform OS obsolescence. Based on the responses, it appears that most organizations rely on rolling upgrades and sound regression testing to make sure that they stay current with the current practices and that version is not caught by OS obsolescence. It is also interesting to note that between client and server systems, the former is more likely to have no provisions made for OS obsolescence.

Respondent also provided explanation regarding their choice of 'other' (client and server platform obsolescence) coping strategies (Table-8). Noteworthy among these are the use of virtualization to operate deprecated systems on incompatible hardware, development of layered defenses to enable otherwise insecure systems to continue operation, and migration to systems that implement middleware and application integration.

Table-8
'Other' coping strategies as expressed by respondents

Client side	Server side
• In-house support of deprecated operating systems without manufacturer support • Use of virtualization to enable continued operation of legacy applications at the client level while allowing for upgrades to client hardware platforms • "Just kept our fingers crossed that things would keep running as is."	• Use a process of migration of some services to more stable and better supported operating systems • Using a development practice that made use of middleware and integration to extend the life of some services by allowing them to operate behind middleware agents • Developing an overarching security strategy of multiple, layered defenses which allows the use of a deprecated operating system in a protected zone of an internal, trusted network • Using virtualization to allow an extension of the life of deprecated operating systems when compatible hardware if no longer cost effective.

DISCUSSION AND CONCLUDING REMARKS

Our initial survey of current IT practitioners suggests that operating system obsolescence is not a critical, or even an important factor to most in the field of client/server systems development. The most critical factor (for an operating system) for client/server systems developers appears to be ***reliability*** which is regarded as of utmost importance by the respondents. This appears to stem from the mandate that IS has in terms of providing seamless and efficacious support to the business processes. Another group of critical factors comprising the core functionality of client/server systems (viz. functionality, performance, securability and scalability) closely follows reliability in importance but does not compare (with reliability) with the same degree of criticality. The third group of factors reflects desirable niche/non-functional capabilities such as interoperability, modifiability, extensibility and reusability and is of keen interest to a few respondents, which is, however, not shared by the first two groups. A final group reflects factors that are sometimes considered important or critical by fewer than half of the respondents, and even though these remain desirable characteristics for client/server systems platforms; they are not widely viewed as critical or important.

We observe that impact of obsolescence is perceived to be the highest for database platforms, followed by application platforms. The same emphasis was evident on database and application platforms in terms of the total cost of systems (as impacted by obsolescence).

Although respondents do not appear to ascribe much significance to obsolescence as a factor of consideration during the planning and design phases, it is evident that several measures have been utilized/thought-of in the organizations to combat obsolescence of platform in the server or client side. Certain insights are noteworthy from these measures: obsolescence-triggered security issues are of concern, and a layered defense strategy is appropriate; obsolescence could be successfully hidden behind contemporary middleware; and virtualization could be an effective way to extend the lifeline of deprecated systems.

Overall, the outcomes of the survey lead us to conjecture that practitioners are accustomed with the evolving nature of the IS/IT field in general, that they treat such uncertainties as business risk and take obsolescence issues in their stride. We think that this could be the reason why they expect to assuage such concerns during the maintenance phase as a contingent requirement rather than taking painstaking considerations during the design phase. Having said so, it is also incumbent upon us to accept that our survey did not include economics, risk management or accounting personnel of these organizations, who may have responsibilities/embedded decisional criteria in their processes of approval and budget allocation of IS/IT projects. As regards possible directions for future research options include 1) identifying data collection mechanisms that could assure that opinions are holistic across professionals involved in project approval and planning in order to more effectively locate and identify obsolescence (and countermeasure) concerns of distributed systems, and 2) investigations in to how virtualization technologies are used as a control strategy for obsolescence, and what effect the various virtualization approaches available can have to control systems for risks of technical obsolescence.

REFERENCES

1. Biswas, P. K. "Toward Agent Oriented Conceptualization and Implementation," *Architectural Design of Multi-Agent Systems: Technologies and Techniques.* Information Science Reference, Kershey, NY, 2007, pp. 15-16.
2. Coulouris, G., Dollimore, J., & Kindberg, T. *Distributed systems: Concepts and design. (4th ed.).* Pearson Education Limited, Harlow, England, 2005.
3. Dennis, A., Wixom, B., & Tegarden. D. Systems Analysis and Design with UML Version 2.0: An Object-Oriented Approach: Second Edition. John Wiley & Sons, Inc., 2005, 265-267.
4. Henning M. "The Rise and Fall of CORBA," *ACM Queue*, 4:5, 1996, pp. 28-34.
5. Jones, C. "Geriatric Issues of Aging Software," *Software Productivity Research, Inc.* Available at www.compaid.com/caiinternet/ezine/capers-maintenance.pdf.
6. Kressel, H. and Lento, T.V., *Competing For The Future: How Digital Innovations Are Changing The World.* Cambridge University Press, Cambridge, England, 2007.
7. Langlois, R. N. *The Dynamics of Industrial Capitalism.* Routledge, London, England, 2007
8. Lepreau, J., Ford, B. and Hibler, M. "The Persistent Relevance of the Local Operating System to Global Applications," Proceedings of the 7th workshop on ACM SIGOPS European workshop: Systems support for worldwide applications, Connemara, Ireland: September 1996, pp. 133-140.
9. Peng, Y., Li, F. and Mili, A. "Modeling the Evolution of Operating Systems: An Empirical Study", *Journal of Systems Software* 80:1, 2007, 1-15
10. Slade, G. *Made to Break.* Harvard University Press, Cambridge, MA, 2006.
11. Whelan, K. "Computers, Obsolescence, and Productivity", The Review of Economics and Statistics. 84: 3, 2002.
12. Williams, R. Retooling: A Historian Confronts Technological Change. The MIT Press, Cambridge, MA, 2002.