Information Systems Evolution over the Last 15 Years

Magne Davidsen and John Krogstie

IDI, NTNU, Trondheim, Norway
Magne.davidsen@gmail.com, krogstie@idi.ntnu.no

Abstract. The information systems we see around us today are at first sight very different from those that were developed 15 years ago and more. On the other hand, it seems that we are still struggling with many of the same problems. To understand how we can evolve future ISs, we should have good understanding of the existing application portfolios. In this article we present selected data from survey investigations performed in 1993, 1998, 2003 and 2008 among Norwegian organizations on how they conduct information systems development and evolution. A major finding is that even if we witness large changes in the underlying implementation technology and approaches used, a number of aspects such as the overall percentage of time used for maintaining and evolving systems in production compared to time used for development is stable, and should be taken into account in the planning of information systems evolution for the future.

Keywords: Information systems evolution, maintenance.

1 Introduction

Modern information systems are the result of the interconnection of systems of many organizations, which are running in variable contexts, and require the capability to actively react to changing requirements and failures. Large changes in how we develop information systems have been witnessed over the last 15-20 years. The prevalent development methods, programming languages and general technological infrastructure have changed a lot. In the early nineties, one was going from mainframe solutions to a client-server, and then to an internet architecture for most applications. Usage of packages has been on the rise. Lately SOA, free and open source software, outsourcing and agile development are also expected to have impact. According to [9] one of the impacts on the state of IS-development is the increasing amount of time used for maintenance and support of systems, contrary to what is claimed to be the impact of e.g. the introduction of SOA.

In this paper, we present descriptive results from a survey-investigation performed in Norwegian organizations in this area during the end of 2008, comparing with similar investigations done in 2003, 1998 and 1993. We will first give definitions of important terms used within information systems evolution. We describe the research method, before the main descriptive results from our investigation are presented and compared with previous investigations. The last section summarizes our results.

B. Pernici (Ed.): CAiSE 2010, LNCS 6051, pp. 296–301, 2010.

2 Definition of Core Concepts

Maintenance is defined as the process of modifying a software system or component after delivery to production. It has traditionally been divided into three types: corrective, adaptive and perfective [8] inspired by, e.g. [21]. Corrective maintenance is performed to correct faults in hardware and software. Adaptive maintenance is performed to make the computer program usable in a changed environment Perfective maintenance is performed to improve the performance, maintainability, or other attributes of the program. This has been divided into enhancive maintenance [2] and non-functional perfective maintenance. Enhancive maintenance implies changes and additions to the functionality offered to the users by the system. Non-functional perfective maintenance implies improvements to the quality of the system. In addition to the temporal distinction between development and maintenance, we have introduced the concepts application portfolio evolution and application portfolio upkeep (previously termed functional development and functional maintenance [10]).

1. Application portfolio evolution: Development or maintenance where changes in the application increase the functional coverage of the total application systems portfolio of the organization. This includes development of new systems that support new areas and enhancive maintenance.
2. Application portfolio upkeep: Work made to keep up the functional coverage of the information system portfolio of the organization. This includes corrective adaptive and non-functional perfective maintenance and development of replacement system

We note that some writers provide more detailed overview of maintenance tasks [3, 9]. Jones [9] has in total 21 categories, also including user-support to be part of maintenance (a view shared with, Dekleva [4], but not with us).

3 Research Methodology

Our survey form was implemented in the SurveyMonkey web-tool and invitations were distributed by e-mail to 278 Norwegian organizations. The organizations were randomly selected from the list of member organizations of The Norwegian Computer Society – NCS. (NCS has currently around 1000 member organizations), using the same approach as in the previous investigations. The contents of the form were based on previous investigations within this area; especially those described in [7, 10, 12, 15, 16, 18, 22]. 79 responses were returned, giving a response rate of 28%. Out of these, 67 responses could be used for the analysis since the additional 12 responses were incomplete. This was a higher response rate than in previous investigations.

The forms were filled in by people with long experience with IT-related work (average 17.5 years), most filling the role as IT director in the company. We will compare some of the results with the results of similar investigations in particular.

1. The investigation carried out by Krogstie, Jahr and Sjøberg in 2003 [12].
2. The investigation carried out by Holgeid, Krogstie and Sjøberg [7] in 1998.
3. The investigation carried out by Krogstie [10, 11] in 1993.

These surveys contain the results from 54, 52 and 53 organizations, respectively. A number of similar investigations of this type were performed earlier in the USA [16, 18, 22]. Comparisons with these surveys have been done in [12]. A number of later investigations have been done, but they only look on the distribution of maintenance tasks [6, 13, 17, 19], looking on the situation in a few organization.

4 Selected Results

First, we present some of the overall demographics of the survey. Similar results from our previous surveys conducted in 2003, 1998, and 1993 are included in parenthesis where the numbers are comparable. The mean number of main systems in the organizations was 7.8 (2003-4.5; 1998-9.6; 1993-10.3). The mean user population of these systems was 4661 (2003– 314; 1998-498; 1993-541). It is in particular the number of external users that has increased dramatically (also relative to number of employees in the organizations); the average number of internal users was 944. The average age of the systems was 4.9 years (2003-3.9; 1998-6.4; 1993-4.6).

In 1993, 58% of the systems were developed by the IS-organization, and only one percent was developed *in* the user organization. In 1998, however, 27% of the systems were developed by the IS-organization and 27% as custom systems *in* the user organization. In 2003 23 % of the systems are developed in the IS-organization, whereas in 2008 only 12% was developed in the IS organization. The percentage of systems developed by outside firms is higher (40% vs. 35% in 2003, vs. 22% in 1998 vs. 12 % in 1993). The percentage of systems developed based on packages with small or large adjustments is also comparatively high (41% vs. 39% in 2003 vs. 24% in 1998 vs. 28% in 1993). The new category we introduced in 1998, component-based development (renamed "use of external web services" in 2008) is still small (5%) although increasing (1.0 % in 2003, 0.4% in 1998) of the total systems.

From being dominant ten to fifteen years ago COBOL is almost not used anymore. The languages that are used in most organizations and for most systems are now Java (40%, 27% in 2003) and C++ (33%, 24% in 2003). Java was just starting to be in widespread use in 1998 and C++ was barely included in 1993. The percentage of organizations reporting to have COBOL applications has decreased from 73% in 1993 to 26% in 1998 to 5% in 2008.

94 new systems were currently being developed; 60 of these systems (64 %) were regarded as replacement systems. (2003-60%; 1998-57%; 1993-48%). 13% of the current portfolio was being replaced. (2003-13%; 1998-9%; 1993-11%). The average age of systems to be replaced was 7 years (2003 - 5.5; 1998 - 7.7; 1993 - 8.5).

Reasons for system replacements have changed slightly from earlier investigations. The most important reasons for replacement are need for integration and burden to maintain and operate, a bit surprising giving the relatively young age of the systems that are replaced. One area which is expected to influence the software development and maintenance landscape is Service Oriented Architecture (SOA) [14]. Transfer to SOA was very important as a reason to create replacement systems for only two organizations. Less than 20% of the organizations had started implementing SOA, and we could not find any significant impact on the use of SOA on maintenance figures.

Work on application systems was in the survey divided into the six categories presented in section 2. The same categories were also used in 1993, 1998 and 2003. We also asked for the time used for user-support and for systems operations which took up the additional time for the work in the IS departments.

In earlier investigation of this sort between 50% and 60% of the effort is done to enhance systems in operation (maintenance) when disregarding other work than development and maintenance. An exception from this was our study in 1998 that was influenced particularly by the amount of Y2K-oriented maintenance. The numbers for Dekleva [4] and those reported by Capers Jones [9] were also higher than this, but these also include user support as part of maintenance.

Table 1 summarizes the descriptive results on the distribution of work in the categories in our investigation, comparing to previous investigations.

Table 1. Distribution of the work done by IS-departments in percentage

Category	2008	2003	1998	1993
Corrective	8.2	8.8	12.7	10.4
Adaptive	6.2	7.3	8.2	4
Enhancive	11.3	12.9	15.2	20.4
Non-functional perfective	9.1	7.6	5.4	5.2
Total maintenance	**34.9**	**36.7**	**41,4**	**40**
Replacement	9.7	9.9	7.7	11,2
New development	11.4	12.6	9.5	18,4
Total development	**21.1**	**22.5**	**17.1**	**29.6**
Technical operation	23.7	23.8	23	NA
User support	20.1	17.1	18.6	NA
Total other	**44.0**	**40.8**	**41.6**	**30.4**

When disregarding other work than development and maintenance of application systems, the percentages are as follows: maintenance activities: 65, 7%, development activities: 34.3%. This is at the same level as in 2003. 63% of development and maintenance work was application portfolio upkeep, and 37% was application portfolio evolution. This is almost the same as in 2003 and 1998, which in turn was significantly different from the situation in 1993 where application portfolio upkeep- and application portfolio evolution respectively amounted to 44% and 56% of the work.

Looking in detail on the distribution of maintenance activities, a number of later studies have been looking at this in particular. As stated in [5] corrective efforts is time-consuming. As reported in [1], it appears to be very large differences reported in different studies. Whereas Lientz/Swanson [16] reported 60% perfective, 18% adaptive and 17% corrective maintenance when asking about selected systems from a large number of organizations (one per organization), Sousa [33] reported (based on a number of systems in one organization) 49% adaptive, 36% corrective, and 14 % perfective maintenance. [17] reported 53% corrective, 36% perfective and 4% adaptive maintenance, based on data on three open source products. [13] reported 62% perfective, 32% corrective, and 6% adaptive maintenance based on data from one application in production. Most interesting for comparison with other surveys is

looking at corrective, adaptive, and perfective maintenance, which appears to be much more stable than the numbers reported from others above when looking upon this across our investigations looking on the application portfolios of a number of organizations.

5 Conclusions and Further Work

There are a number of differences in the underlying technology, which is as expected. This is very clearly witnessed in the distribution of programming languages used, where procedurally languages like COBOL have to a large extent been suppressed by object-oriented languages like Java, C++ and C#. New architectural trends such as SOA have yet to make a noticeable impact on the use of resources. Another marked difference is that less and less of IT is done internally in organizations (this applies to development, maintenance, operations and use), which will make appropriate software evolution harder to do. On the other hand, even if most organizations outsource part of the IT-activities, most still do some of the activities in house. Overall percentage of time used for evolving systems in production compared to time used for development is remarkably stable on average (these numbers differs a lot from year to year within individual organizations). The same can be said about the rate of replacement, although slightly increasing, more than 60% of 'new' systems to be developed are actually replacement systems, constituting around 13% of the current application portfolio. Since more complex infrastructures are supporting the information systems, more and more of the resources are used for other tasks such as operations and user-support, less and less time is available for providing new information systems support in organization, although it seems to have plateau on 20% of the overall time, a level reached already ten years ago in Norway (i.e. earlier than indicated in [9]). Even when using new approaches where evolution is better supported by the system itself and the design of the information system consider evolution as an inherent property of the system, it will take quite some time until such technology is used broadly. Waiting for this, we have to keep the system operational and continuously evolving, addressing new needs currently not known.

Several of our results have spurred new areas that could be interesting to follow up on in further investigations, and we are currently performing several detailed case studies. A long-term plan is to do a similar investigation in 2013.

References

1. Benestad, H.C., Anda, B.C.D., Arisholm, E.: Understanding software maintenance and evolution by analyzing individual changes: A literature review. Journal of Software Maintenance and Evolution: Research and Practice (2009)
2. Chapin, N.: Software Maintenance Types – A Fresh View. In: Proceedings of the International Conference on Software Maintenance (ICSM 2000), pp. 247–252 (2000)
3. Chapin, N., Hale, J., Khan, K., Ramil, J., Tan, W.-T.: Types of Software Evolution and Software Maintenance. Journal of Software Maintenance (13), 3–30 (2001)
4. Dekleva, S.M.: Software Maintenance: 1990 Status. Journal of Software Maintenance 4, 233–247 (1992)

5. Ghazarian, A.: A Case Study of Defect Introduction Mechanisms. In: van Eck, P., Gordijn, J., Wieringa, R. (eds.) CAiSE 2009. LNCS, vol. 5565, pp. 156–170. Springer, Heidelberg (2009)
6. Gupta, A., Slyngstad, O.P., Conradi, R., Mohagheghi, P., Rønneberg, H., Landre, E.: An Empirical Study of Software Changes in Statoil ASA - Origin, Priority Level and Relation to Component Size. In: Proceedings of the international Conference on Software Engineering Advances, ICSEA, Washington, DC, October 29-November 03, p. 12. IEEE Computer Society, Los Alamitos (2006)
7. Holgeid, K.K., Krogstie, J., Sjøberg, D.I.K.: A study of development and maintenance in Norway: Assessing the efficiency of information systems support using functional maintenance. Information and Software Technology 42, 687–700 (2000)
8. IEEE Standard Glossary of Software Engineering Terminology (1991)
9. Jones, C.: The Economics of Software Maintenance in the Twenty First Century (2006), http://www.compaid.com/caiinternet/ezine/capersjones-maintenance.pdf (last accessed February 2010)
10. Krogstie, J., Sølvberg, A.: Software maintenance in Norway: A survey investigation. In: Muller, H., Georges, M. (eds.) Proceedings of the International Conference on Software Maintenance (ICSM 1994), Victoria, Canada, pp. 304–313. IEEE CS Press, Los Alamitos (1994)
11. Krogstie, J.: On the distinction between functional development and functional maintenance. Journal of Software Maintenance 7, 383–403 (1995)
12. Krogstie, J., Jahr, A., Sjøberg, D.I.K.: A Longitudinal Study of Development and Maintenance in Norway: Report from the 2003 Investigation. Information and Software Technology 48, 993–1005 (2006)
13. Lee, M.-G., Jefferson, T.L.: An Empirical Study of Software Maintenance of a Web-Based Java Application. In: ICSM 2005 (2005)
14. Lewis, G.A., Smith, D.B.: Service-oriented Architecture and its Implications for Software Maintenance and Evolution Frontiers of Software Maintenance. In: FoSM 2008 (2008)
15. Lientz, B.P., Swanson, E.B., Tompkins, G.E.: Characteristics of application software maintenance. Communications of the ACM 21(6), 466–471 (1978)
16. Lientz, B.P., Swanson, E.B.: Software Maintenance Management. Addison-Wesley, Reading (1980)
17. Mohagheghi, P., Conradi, R.: An Empirical Study of Software Change: Origin, Acceptance Rate and Functionality vs. Quality Attributes. In: International Symposium on Empirical Software Engineering (ISESE 2004), pp. 7–16 (2004)
18. Nosek, J.T., Palvia, P.: Software maintenance management: Changes in the last decade. Journal of Software Maintenance 2, 157–174 (1990)
19. Schach, S.R., Jin, B., Yu, L., Heller, G.Z., Offutt, J.: Determining the Distribution of Maintenance Categories: Survey versus Measurement. Empirical Software Engineering 8(4), 351–365 (2003)
20. Sousa, H., Moreira, H.: A Survey of the Software Maintenance Process. In: Proceedings of ICSM 1998, Bethesda, Maryland, pp. 268–274. IEEE CS Press, Los Alamitos (1998)
21. Swanson, E.B.: The dimensions of maintenance. In: Proceedings of the Second International Conference on Software Engineering, San Francisco, USA, August 1976, pp. 492–497 (1976)
22. Swanson, E.B., Beath, C.M.: Maintaining Information Systems in Organizations. Wiley Series in Information Systems. John Wiley & Sons, Chichester (1989)