Proceedings of the 8th International Conference on Applied Informatics
Eger, Hungary, January 27–30, 2010. Vol. 2. pp. 445–448.

Methodology for Assessment Validation of Platform Migration of Roboust Critical IT-systems

Ábel Garai

University of Debrecen

Nowadays several platform-migrations are running at blue-chip companies. The background factors may involve cost-efficiency and changing technical environment. An average migration of a mission-critical system consumes years and millions of dollars to execute. Therefore, it will be a good idea to measure - in advance - whether it would bring the expected results at the end of the migration. Furthermore, one should analyse a set of candidate platforms before choosing. Migration may involve alteration of hardware and software products. We can classify migrations from the functional or from technical point of view. Functional migration may involve user-interface, data- and program-code migration. The technical dimensions are hardware, architecture, system-software and development environment migration. Also, moving forward towards cloud computing, the whole contex for hardware architecture will be dramatically different far away from the stereotypes we have been accustomed to. The new technologies will call for new approach.

There are several roboust private and public institutions running their old mainframes, 'big irons' in the background to fulfill core business tasks. These tasks generally involve mission-critical activities. Mission-critical is an expression describing a role that is vital to the organization's daily successful operation. Examples for such mission-critical systems are billing, airport check-in and bank-card authorization systems. An important factor during the migration is that the ordinary business should continue, and there is no place for outage.

The London Stock Exchange is a good example moving from Microsoft to Linux. Owens and Minor substituted its legacy mainframe with a .NET environment. Exxon Mobil aims to shift its mainframe processes towards SAP. Deutsche Telekom plans to replace Mainframe with Linux.

Platform migrations have been conducted in the past decades, and are happening today, also. Several blue-chip companies are trying to 'normalize costs', which may involve getting rid of old, legacy technologies. The old technology represents mostly the mainframes today. What is wrong with the mainframe giants, where

IBM is the flag-ship? The licences are expensive, the knowledgeable personnel are retiring and expensive, furthermore, the technology offers only limited flexibility and conservative online interfaces.

How does a general migration project start? CIOs and application owners are under pressure to lower operating costs. One option is to look for a cheaper platform. Other driver may be technical modernization.

Where are these mission-critical systems generally used? A card-authorization system is very sensitive to work 24/7. Even a short outage, and several card-transactions will be blocked. The other classical example is the check-in system at an airport. We can image what a system failure could cause and how it would affect the business. The other general function lying on a mission-critical system is a billing system. Why is billing so important? Without having the bills issued and filed, no income will be received. Imagine, if the German landline telephone company would issue only five percent of the telephone bills incorrectly. It would imply more than two million customers. Just imagine when the following day more than two million upset customers call the helpdesk.

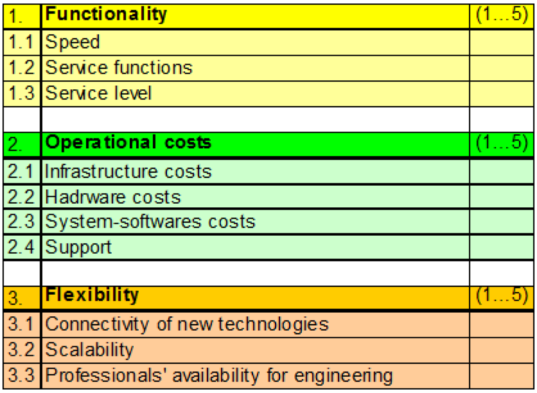
What are the general legacy and destination platforms? Generally the platform to be substituted is the Mainframe with IBM's z/OS as operating system. The usual object-platforms nowadays are Linux, Unix, SAP and Microsoft client-server solutions. Beside functionality, response-time and reliability are key-words. The roboust mission-critical core processes do not necessarily require super-computers, but they do need to keep up with the massive volume of transactions.

Where are these big irons and huge systems running? How does a company owning such a system look like? Banks, insurance companies and state institutions are good examples.

In my analysis the three major dimensions are functionality, operating costs and flexibility. These three criterions are used to describe a system, and to help to compare two or more systems.

The methodology stands on a scoring table. The scoring table consists of 3 key components: functionality, operational costs and flexibility. Each component has 3 to 4 subcomponents. These subcomponents act as the backbone of the full methodology. A number between one and five is to be associated to each subcomponent. The scale between one and five represents whether the new candidate system along the chosen dimension is significantly different from the present one. On the scale, number three represents that the present and candidate systems are equivalent from the chosen subcomponent's point of view. Four shows that the candidate system is better, but not significantly. Five represents that the candidate system is significantly better. Two means that the future system is somewhat poorer. In case of one, the candidate system is significantly worse.

The hypothesis is that the candidate system is significantly better than the present system. The phrase significantly is associated with a clearly defined threshold value.



1.	**Functionality**	(1...5)
1.1	Speed	
1.2	Service functions	
1.3	Service level	
2.	**Operational costs**	(1...5)
2.1	Infrastructure costs	
2.2	Hadrware costs	
2.3	System-softwares costs	
2.4	Support	
3.	**Flexibility**	(1...5)
3.1	Connectivity of new technologies	
3.2	Scalability	
3.3	Professionals' availability for engineering	

The method aims to achieve the following three characteristics: simplicity, measurability and objectivity. The system is intended to keep the calculation within extremely simple boundaries. In case of a multi-million euro project, which represents a normal budget on this field, the drafted short list of classification is very daring. As the list of the dimensions is intentionally very limited, the punctual definition of the scoring is elemental. Furthermore, some fine-tuning is possible by the weighting of the dimensions. The measurability proves that the outcome of the application of the method is mostly objective. Why does the result have only limited objectivity? The input step involves a person to classify the judgment whether the candidate and present systems are slightly or significantly different. The flexibility at this entry point limits the objectivity at the end.

There are several challenges in the methodology, also. There is chance to conclude contradictions as a result. For example, the technique at present stage may misinterpret if the candidate system provides higher service level but at a higher operating cost. In this case, a score will be higher in the functionality field; however another quantity will be lower in the operational costs part. The application of a potential threshold can also interfere with the result, showing something different as we would expect. There is a need to test the whole method at some initial cases: the mathematical outcome is to be challenged against what common sense would dictate. It is to be checked that the applied mathematical computing represents and follows the real and aimed logic of the model. The mathematic and scoring behind the technique should not show up with a l'art pour l'art process, but must represent the aimed evaluation method with a clear and valuable result. The improvement of the procedure requires the sophisticated application of weighting and setting correct thresholds.

Before 'going live', in course of the validation of the method, it is to be tested on an already completed platform migration. The requirement is that the result of the drafted methodology should point to the same conclusion as was done during the real decision.

The method provides a scientific process to compare how the same system runs on different platforms.

References

[1] ITILŽ Lifecycle Approach Based on ITILŽ V3 Suite - 5 Management Guides (english version), Authors: Jan van Bon, Arjen de Jong, Axel Kolthof, Mike Pieper, Ruby Tjassing, Annelies van der Veen, Tieneke Verheijen, ISBN: 9789087531089

[2] ISO 9126 Part 1-4, ISO/IEC, 2003

[3] COBIT 4.1 Framework, IT Governance Institute, 2007

[4] Workflows der Software-Migration, Rainer Gimnich - Andreas Winter, 2005

[5] http://www.computerworlduk.com/technology/networking/messaging/news/index.cfm?newsid=18429

[6] http://www.sdtimes.com/blog/post/2010/01/04/Modernizing-versus-Rip-and-Replace.aspx