# Assessing the Readiness to Move into the Cloud

Leire Orue-Echevarria, Juncal Alonso, Marisa Escalante, and Stefan Schuster

TECNALIA. ICT / European Software Institute Divison
Parque Tecnológico Ed. #202. E-48170 Zamudio. Spain
{leire.orue-echevarria,juncal.alonso,marisa.escalante,
stefan.schuster}@tecnalia.com

**Abstract.** The race to keep software compatible and optimal with respect to the latest trends is hard. 90% of software cost can be due to maintenance, and 75% on developing new features to stay competitive and relevant. The industry progresses through periods of incremental development interspersed with true paradigm shifts. Legacy software must keep up the pace.

At present we are experiencing one of these paradigm shifts, as remarked by the EC [1] "The speed of change in Internet technologies continues to be impressive. Software is becoming more and more pervasive: it runs on the devices that we use every day ... [opening] a new world of possible applications". Today, technological and business model innovation generates large demand for the transition of legacy software towards modernization. However, software modernization is not a trivial issue and if improperly done, it dangers the business continuity and sustainability.

This means that for any company meditating about the transition to the new paradigm of cloud computing, there is a need to have at its disposal an innovative and combined technical and business analysis on the maturity and prospect of the legacy application. The major target of this process is to identify in advance the perspectives of the migration and pre-evaluate the performance and business benefits with relation to the cost of the process. For the first time, the business value will be directly attached to the technical performance.

This paper presents this aforementioned approach, being currently developed and tested, in order to assess the maturity of an application and the convenience of migrating to the new cloud computing paradigm or not, based on quantitative indicators while always ensuring the company's business continuity. Following this approach, questions such as cost and effort of the migration, impact of new business models in the company or return of the investment will be provided in advance of tackling the actual modernization.

## 1 Introduction

New developments in the way services and applications can be delivered over the Internet have opened up huge opportunities to software vendors. The Internet is not only getting faster and thus data is transferred in a quicker manner but it is also becoming more reliable in what concerns transactions among customers and providers. This is making possible the offerings of basic IT appliances such as servers for storage or computing clusters as a service, i.e. providers provide the hardware and infrastructure and clients provide the data. The decoupling of responsibilities accelerates the development of new service platforms and software products.

M. Yousif, L. Schubert (Eds.): Cloudcomp 2012, LNICST 112, pp. 12–20, 2013.

Due to the fact that the innovation rate is accelerating, software products in the Internet era need also to constantly evolve. Take a look for instance the last five years and how the way we work has changed thanks to the breakthrough of cloud computing, smartphones or social networks. Innovations in the technological space affect the systems that the software has to support or needs to adapt to. Innovations in the business space also affect the licensing usage and delivery model. Software products have to be improved with regard to these new circumstances but without disrupting the business continuity of existing customers.

However, managing software modernization is still a significant challenge in today's software life cycle. This challenge is usually considered as inevitable, unpredictable, costly, technically difficult, time-and resource-consuming, and poorly supported by tools and techniques or formalisms. The complete lifecycle of software, from requirements to run-time and delivery has to be re-adapted to the new technological and business conditions, requirements and challenges, since there is an increasing need for tools/means to support software evolution and adaptation as a key value for next generation service based software modernization.

The first challenge that all companies face is **the decision whether to migrate their existing products or to start from scratch**. Current open [6] [7] [8] [9] and proprietary migration methodologies [10] [11] [12] [13] [14] to service based software mostly begin with a big bang approach or perform a feasibility analysis well advanced the migration process. Questions such as cost and effort of the migration, impact of new business models in the company or return of the investment need to be answered **before** tackling the actual modernization. If the estimates they obtain suit their expectations and they finally decide on the migration to a service-based software, reusing as much as possible from the old one, they will face further challenges and difficulties, not only with respect to the usage of new technologies, or architecture but also with respect to assumptions that companies usually take for granted and then are no longer valid.

## 2 Approach

The main objective of the approach presented in this paper, currently being developed and tested, is to provide a set of methods and techniques that will support companies on the assessment for the modernization of their software towards a Cloud delivery model, sustaining them on the migration strategy and providing the required tools to analyse the impact of the potential transformation of the software in the company.

The modernization of the software and its delivery will be analyzed under two different, but intertwined dimensions: one focusing on Technology (architecture, performance, reliability, data schema, and so on) and another one on Organisational & Business aspects (pricing model, market addressed, organisational processes, etc). This is rather significant since the business model offered by the organization (based on the delivery of software artefacts) will change from a product to a service. In the cloud world, decisions taken at business level constraint the technology and vice versa, for instance, a billing component (business related) needs a monitoring component of application use (technology related).

After the assessment, the assessed organizations will be able to visualize a maturity map where the position of their current business service is shown, as well as the

potential position (in terms of technology modernization and business model changes) once the migration takes place.

In addition, the modernization assessment will support the analysis of such initial and desired situations through a set of impact assessment tools. The main purpose of these tools is to establish a collection of objective and measurable metrics and indicators on which to estimate the feasibility of the migration. Furthermore, the figures will be presented in measurement units and concepts easily shared, recognised and acknowledged by stakeholders.

Summarizing, the main outcomes of this approach are:

- a **method for characterising the technical and business dimensions** of the current legacy application, in particular those concerns related with its modernisation towards a selected target,
- a set of common **metrics and indicators** that characterise relevant technical aspects of the legacy application and the business model before and after the migration takes place,
- a set of tools that will **automatically evaluate the figures** related to the modernization processes such as: resources and effort required, impact in the company processes, estimated ROI and payback, operational risks,
- a **modernization strategy** with the activities to carry out in case the organisation decides to continue with the modernisation process after the figures are analysed.

### A. Business and Technical Modernization Assessment

This step focuses on the characterization of the metrics and indicators (metrics weighed and combined) of the business and technical dimension of the legacy application and the company, such as the pricing model, the targeted market, the product sustainability, SLAs, legal issue, metrics that describe the legacy application source code and data schema complexity, compliance with baseline legacy technologies, gap estimation between legacy and target baseline, etc. Authors of this approach have not yet found a similar Business and Technical Modernization Assessment procedure in literature, neither a classification of applications from an architectural point of view, nor a business model and process one.

In order to perform this assessment several issues and knowledge are pre-required. Among them: artefacts and knowledge related to source code and architecture, development process, GUI, source environment and desired environment, source and target infrastructure, covered and uncovered non-functional requirements.

This assessment is to be executed in several steps:

**Step 1:** Fill in on-line questionnaires. Examples in Spanish can be found here [2]. These user friendly questionnaires can be answered by a person with a technical role, a person with a more business-oriented role or a person covering both roles. The main requirement is to have a good knowledge of the application in terms of architectural issues, programming language, security, SLA fulfilment, helpdesk, maintenance, privacy and trust practices, marketing, business model, pricing model, target platform model (private, public or hybrid cloud) and performance and reliability. The questions are related both to the current situation and the desired situation, that is, how the application and business model shall behave once the migration takes place.


**Fig. 1.** Assessment Questionnaires

**Step 2:** Based on the results attained in the questionnaires, an analysis is executed. Similar to the evaluation of quality criteria as motivated by the ISO 9126 quality model standard and used in the methodology of the "bidirectional quality model" [3] this approach measures several metrics by evaluating questions and checklists. The measured metrics are weighted following a certain criteria and aggregated into indicators of interest, which define a characteristic used to calculate the maturity of the business model and the maturity of the technology model, both before the migration and after the migration takes place.

**Step 3:** Presentation of the results in a graphical manner. The authors found that an adequate and visible way to do it is by means of a quadrant. An example of such a quadrant is shown in the next figure:

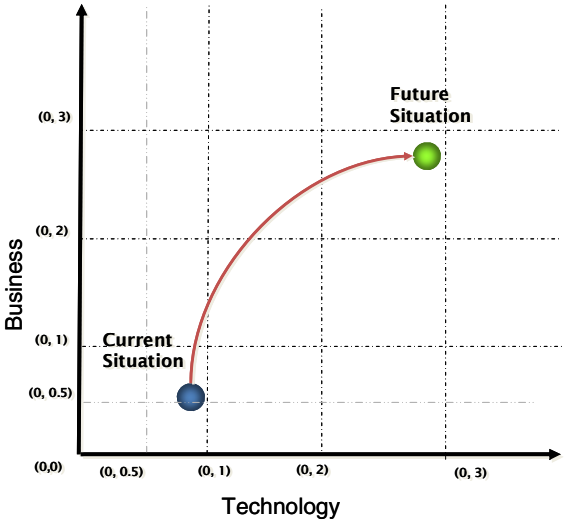


**Fig. 2.** Position of an application in the quadrant

The current maturity levels of the Technology and Business axis, as shown in the figure above, have been established based on the professional experience and State-of-the-Art studies [15] [16] [17] [18] from the authors. Nevertheless, the maturity levels will be accordingly updated with new achievements.

These maturity values represented in each axis have the following meanings:

Technology axis

- (0,0) Monolithic: interface logic, business logic, and data logic are in the same machine.
- (0,0.5) Client-server with a thick client (i.e. VB application), event driven. Code tightly coupled to the interface. DB is in the local network or on a server outside but all the logic remains in the same machine.
- (0,1) Client-server with a thin client (i.e. j2EE application, 2-n tier), with no usage of web services. Multiple DB instances.
- (0,2) Client-server with a thin client such as mozilla, opera, chrome or Internet explorer (i.e. J2EE application, 2-n tier), with usage of web services. A unique instance of the DB. Multiple instances of the application.
- (0,3) Client-server with a thin client, 1 DB instance, 1 DB application, n appearance customizations.

Business axis

- (0,0) License (instalment), support, updates, upgrades, maintenance are paid under a different fee model than the license. No helpdesk. No SLA. No upgrade protocol and procedures.
- (0,0.5) Most revenues are obtained from sales of licenses. Even though, there exist some sales (less than 10% of the total amount) that are made in a service form with a flat rate model.
- (1,0) Most revenues are obtained from sales of licenses. Between 10-20% are from the product sale as service with pay per use, flat rate, hybrid pricing models. SLA is being defined. Upgrade protocol and procedures are being defined.
- (2,0) More than 60% of the sales are from the product as a service. Helpdesk is institutionalized but not 24x7 and only in certain languages. Existence of SLA, upgrade protocol but upgrades are still seldom, legal department.
- (3,0) 100% of the sales are from the product as a service. Existence of a 24x7 helpdesk, multilingual, Marketing mostly done through the Internet (social media), SLA, upgrade protocol and procedures, Long Tail business model.

### B. Technical Feasibility Analysis

A possible way to discern whether the migration is possible or not, is a feasibility analysis. Existing migration methods to SOA like SEI's SMART [5] propose doing it mainly by means of questionnaires or interviews to key people leaving aside the use of supporting tools.

The feasibility analysis performed in this approach, however, centres it on several tools with the main purpose of providing numbers and graphical images that will ease the decision of whether to tackle the migration or not. These tools and their purpose are described next:

- **Code analysis:** The goal of such an analysis is twofold. On one hand, to represent the coupling of methods, types, classes, namespaces or assemblies in varying sizes according to common metrics like lines of code (LOC), McCabe's Cyclometric Complexity, CBO (coupling between objects), RFC (response for class) [19] RFC∞ [20] MPC (message passing coupling) [19] DAC (data abstraction coupling) and DAC1 [21] ICP (information-flow-based coupling) [22] COF (coupling factor), in order to obtain other useful views of code cohesiveness and complexity. These views can be presented in a variety of ways such as a matrix, a dependency graph or a tree. On the other hand, the second goal of this code analysis is to discern the dependency of the legacy software in 3rd party COTS and/or with other applications internal to the company.
- **High Level Modelling:** This activity has as main goal to obtain the understanding of candidate functions or modules that might be exposed as services in an easy manner. The best way to obtain this knowledge is by modelling the application with UML in its different views, seizing the power of Reverse Engineering Techniques. Also important at this stage is also to analyse the database schema, data and transaction model in order to select the best database architecture (RDBMS, NoSQL, a hybrid solution) and migration strategy towards the chosen one.
- **Effort estimation tool:** based on the desired target cloud platform, this tool provides an estimation of the work (effort) that will be needed to transform it to that target platform.

The main outcomes of this analysis are:

- a set of metrics and indicators that show how complex the code is and thus how much effort will be needed to transform the legacy application to a cloud oriented environment. These metrics will be used in the Business Feasibility Phase to extract the costs of the migration strategy,
- a classification of legacy artefacts to be considered in a posterior Reverse Engineering process, e.g.; source code, configuration files, data files, documentation, existing models, etc.
- based on the previous results, a taxonomy highlighting the main different types of legacy artefacts according to their corresponding characteristic and properties.
- effort calculation on the migration.

### C. Business Feasibility Analysis

The goal of this business feasibility analysis is to provide guidelines that will aid the management level take the decision whether to tackle the migration to SOA and/or Cloud based on objective economic parameters. In order to do so, a cost-benefit analysis is being developed (including ROI and Payback) that will cover the specific

issues related to this shift of business model, as well as means to calculate the impact and implications of changing business models in a company that is already sustainable.

Although a cost-benefit analysis has to be customized for each migration project, there are several common concepts that need to be analysed:

- Costs, divided in development costs and operational costs. Whilst the first kind focuses on how much it will cost to migrate an application compared to developing it from scratch, the second kind focuses on the costs related to training to employees with the new roles they are expected to have, the costs of the cloud provider, costs of regular updates and continuous maintenance, new marketing activities, as well as other structural costs.
- Revenues, considering not only the revenues from the business itself, that is, number of customers using the service, but also considering other issues that at first sight are not seen as direct revenues but that will eventually lead to that, such as:
    - (Reduced) Costs of no quality: More quality in the application as a service since upgrades are done more frequently and every time this happens, regression tests are performed. This is measured in terms of less rework (and human resources dedicated to it) and less time dedicated to solve customer complaints.
    - (Reduced) Costs in travelling for maintenance and installation.
    - (Reduced) Costs in marketing.
    - (Greater) margin by targeting new markets non reachable beforehand.

This analysis provides the management level with at least the following data:

- Net Present Value for the next five years.
- Return on Investment for the next five years.
- Payback in years.

However, not only economic factors are studied in this business analysis. The business processes within the company are also analyzed in order to determine the impact of the business model transition at process level. Cloud Computing Business Model implies the redefinition of old processes and the creation of new ones, such as how to control and maintain the SLA's, customer care, and other support processes but also those related to the business core, the software development, design and testing.

### D. Modernization Strategy

Once the metrics and indicators have been analyzed, the organization will decide on the convenience to continue with the modernization process or not.

If the organization decides to continue with the modernization process, a strategy indicating the activities to carry out will be defined. This strategy will provide the organization with the needed roadmap in order to achieve the desired maturity expressed in the questionnaires.

## 3 Conclusion

The presented work is currently being tested in eight different companies in Spain. Even though the sample is not big enough to ensure complete correctness, the approach has proven to be valid. However, there is still a lot of work to do and improvements to make. These include:

- The set of questions have proved to be valid for the eight cases in which the approach was tested and piloted. However, as the environment was quite controlled in terms of current and target business models and migrated products and technologies, it is clear that if the scope is widened, a new set of questions may arise.

  Also, due to different time constraints, the analysis of the position of the products in the quadrant was performed manually. The idea is to automate this analysis as much as possible to deliver completely as a service over the Internet.
- Future research with the technical and business feasibility analysis. Currently, these are rather manual and thus, time costly. Supporting tools will be developed in the near future.

**Acknowledgement.** This work has been partly funded by the Basque Government project SMARTWORK in the Future Internet and by the Spanish Ministry of Science and Innovation through the project named mCloud: Advance migration to the cloud [5].

Further acknowledgment goes to partners of the ARTIST FP7 proposal.

## References

[1] http://cordis.europa.eu/fp7/ict/ssai/docs/call8objective1-2-brochure-web.pdf
[2] http://sg1.esilab.org/limesurvey/index.php?sid=29249&newtest=Y&lang=es
[3] Simon, F., Seng, O., Mohaut, T.: Code-Quality-Management. Dpunkt.Verlag, Heidelberg (2006)
[4] Orue-Echevarria, L., Alonso, J., Gottschick, J., Restel, H.: FROM SOFTWARE-AS-A-GOOD TO SAAS: CHALLENGES AND NEEDS. Developing a tool supported methodology for the migration of non-SaaS applications to SaaS. In: ICSOFT 2011 (2011)
[5] http://innovation.logica.com.es/web/mcloud
[6] Wu, B., Lawless, D., Bisbal, J., Grimson, J., Wade, V., O'Sullivan, D., et al.: Legacy systems migration - a method and its tool-kit framework. Paper presented at the Joint 1997 Asia Pacific Software Engineering Conference and International Computer Science Conference (1997)
[7] Khusidman, V., Ulrich, W.: Architecture-Driven Modernization: Transforming the Enterprise. OMG (2007)
[8] Warren, I., Ransom, J.: Renaissance: A Method to Support Software System Evolution Paper presented at the 26th Annual International Computer Software and Applications Conference (2002)
[9] http://www.sei.cmu.edu/reports/08tn008.pdf

[10] ftp://service.boulder.ibm.com/s390/audio/pdfs/G224-7298-00_FinalMigratetoSOA.pdf
[11] http://www.oracle.com/technetwork/middleware/soasuite/overview/wp-soa-suite-11gr1-2-129551.pdf
[12] http://download.microsoft.com/download/d/d/e/ddeb427d-dc05-4ab0-b47e-74f0a936d892/Real-World-SOA-At-The-Edge.pdf
[13] http://www.sap.com/platform/soa/adoptionprogram.epx
[14] http://media.amazonwebservices.com/CloudMigration-main.pdf
[15] Moyer, C.: Building Applications in the Cloud. Ed. Addison Wesley (2011)
[16] Reese, G.: Cloud Application Architectures. Ed. O'Reilly Media (2009)
[17] Sosinsky, B.: Cloud Computing Bible. Ed.Wiley (2011)
[18] Chate, S.: Convert your web application to a multi-tenant SaaS solution. IBM developerWorks (2010)
[19] Chidamber, S.R., Kemerer, C.F.: Towards a Metrics Suite for Object Oriented Design. In: Proceedings of OOPSLA 1991, pp. 197–211 (1991)
[20] Chidamber, S.R., Kemerer, C.F.: A Metrics Suite for Object Oriented Design. IEEE Transactions on Software Engineering 20(6), 476–493 (1994)
[21] Li, W., Henry, S.: Object-oriented metrics that predict maintainability. Journal of Systems and Software 23(2), 111–122 (1993)
[22] Lee, Y.S., Liang, B.S., Wu, S.F., Wang, F.J.: Measuring the Coupling and Cohesion of an Object-Oriented Program Based on Information Flow. In: Proceedings of International Conference on Software Quality, Maribor, Slovenia (1995)