

Migration to PaaS Clouds – Migration Process and Architectural Concerns

Claus Pahl

IC4 – the Irish Centre for Cloud Computing and Commerce
Dublin City University
Dublin 9, Ireland
Email: Claus.Pahl@dcu.ie

Huanhuan Xiong

IC4 – the Irish Centre for Cloud Computing and Commerce
Dublin City University
Dublin 9, Ireland
Email: Huanhuan.Xiong@dcu.ie

Abstract—In the cloud computing technology stack, infrastructure has matured more than platform or software service technologies with respect to languages and techniques used for architecting and managing respective applications. Platform-as-a-Service (PaaS) emerges as a focus for the near future that we will focus on. We look at software architecture and programming concerns in the context of migration to PaaS solutions, i.e. the transition of platform systems from on-premise to cloud solutions. We investigate best-practice approaches in cloud-aware coding in the form of patterns and formulate these as a migration process. While one-to-one mappings of software from on-premise to cloud platforms are possible, statelessness and data externalisation from stateful sessions and applications emerge as solutions if cloud benefits such as elasticity and performance are aimed at.

I. INTRODUCTION

PaaS Migration is the process of moving from the use of one software operating and deployment environment to another environment, e.g. from an on-premise solution to the cloud – which is our focus here. As for the IaaS or SaaS-level, where server migration or data loaders help to automate the process, supporting techniques would be beneficial, but, due to the complexity of applications in platforms, have currently less of an impact. The architecture complexity of applications for development and deployment in PaaS is often underestimated. Commonly used migration methods are question catalogues (typical are up to 100 questions that help to capture current and envisioned development architectures) used manually by PaaS provider consultants in the early stages of the migration process [16], [5]. Sometimes these questions are organised into decision trees to guide and focus their application. Based on the broader architectural settings determined through the questions, a concrete migration is implemented. The suggested standard solution often used for PaaS migration is to replicate an existing on-premise servers and applications in a virtual world, i.e. the software running and being maintained elsewhere. However, this might result in an oversized resource consequently being utilised, but not delivering performance gains (and other benefits) that the cloud allows. Current desktop virtualisation is an example of such a simplification. The problem is recognised, but current solutions for PaaS migration essentially avoid benefits and oversimplify. In order to develop (or adapt) software for cloud-based development and deployment, cloud-specific architecture and programming techniques need to be followed. While programming for virtualised environments has been investigated, we specifically look at the PaaS migration context.

- Development using a PaaS environment requires a number of major changes regarding the architecture design and software development approach. Firstly, statelessness is a requirement for resources such as VMs to be deployable without data. Secondly, as a consequence, data externalisation is required to prepare for scale-out, which necessitates externalisation for an efficient management of elasticity requirements.
- Another aspect apart from statelessness and data externalisation is no-SQL data representation. Most applications are developed around a relational representation of data in SQL-based databases. Data to be migrated also needs to be analysed in terms of its correctness and relevance. Validation and cleaning can avoid irrelevant, outdated and incorrect data to be migrate and using unnecessary resources. Furthermore, legacy systems management including component de-commissioning is an architectural concern.

Thus, the software architect or developer is the key to the solution, as s/he needs to implement these techniques. Skills shortage can be a black-spot – training needs arise for the software developers.

We introduce a PaaS migration process as our core contribution. For a migration process template, we capture concerns, migration focus and expectations, but also possible problems and issues stakeholders might not be aware of. A generic migration process is outlined as a template. This part is based on an empirical evaluation of PaaS migration use cases, involving expert interviews and focus groups.

II. ASSUMPTIONS AND BACKGROUND

Cloud migration can be categorised in terms of the cloud stack layers [25]. Migration layers shall here be aligned with the user-oriented service models SaaS, PaaS and IaaS [14], [19], as each deals with software entities of different types with different manipulation and processing needs.

- SaaS. This covers a range of classical enterprise applications such as payroll or accounting to more technical software supporting for instance manufacturing processes. Companies look for migration solutions to move their existing on-premises software to a cloud environment.

- **PaaS.** At this layer, customers do not manage their virtual machines, but are concerned with middleware components, their development and deployment. They develop and deploy applications within an existing API or programming language. PaaS Migration is the process of moving from the use of one software development and deployment environment to another environment.
- **IaaS.** The IaaS layer is concerned with compute, storage or network capabilities that are provided. IaaS migration offers compute services, which often includes moving an in-house server to a (public or private) cloud.

Cloud migration is the process of partially or completely deploying an organization's digital assets, services, IT resources or applications to the cloud. The cloud migration process may involve retaining some IT infrastructure on-site. In such a scenario, the existing system may be fused with a partial cloud solution that can be hosted by a third party over the Internet for a fee. The cloud component of this hybridised system can transition from on-premise to several cloud providers allowing businesses to choose the most cost effective solution. However, the migration process involves the risk of accidentally exposing sensitive business critical information. Therefore, cloud migration requires careful analysis, planning and execution to ensure the cloud solution's compatibility with organizational requirements, while maintaining the security and integrity of the organization's IT systems.

III. PAAS MIGRATION PROCESS

In order to analyse the PaaS migration needs, we summarise a PaaS migration process here. The following definition of a cloud migration process is assumed:

A cloud migration process is a set of analysis, planning and re-engineering activities to support on-premise to cloud migration. Cloud migration processes define a comprehensive perspective, capturing business and technical concerns.

Thus, initial requirements and expectation elicitation are part of the processes, as are tools for automated migration of IT artefacts or plans for deployment of new cloud services and decommissioning of old infrastructure.

As part of our research, we have investigated several case studies. Based on expert interviews and focus groups we carried out with five migration experts (senior developers and consultants with at least 12-15 years of experience, including a minimum of 3 years in cloud technologies, who have been involved in more than 20 projects each), we define a number of top-level activities that are generally performed as discrete, sequenced steps in migration cases. We select aspects that help us to categorise the main migration issues:

- **Concern:** the concern of the respective activity, typically consultation, analysis or implementation.
- **Focus:** differentiates whether focus is technical or business-oriented in nature (determines the stakeholder).

- **Vision:** migration benefits that potential users are aware of (their 'vision').
- **Ignorance:** migration benefits and also pitfalls that potential users are not aware of (their 'ignorance').

The two case studies described here are based on expert interviews with senior migration experts. Due to the experience of the experts, these act as multipliers and consequently guarantee a high degree of validity.

A. PaaS-level Migration

PaaS solutions provide support for both the development and deployment of software application in the cloud. We will particularly focus on development here, with ISVs (independent software vendors) supporting medium or large organisations with their development. This does provide a complex PaaS setting, with the following stakeholders involved:

- **PaaS solution provider:** e.g., a multinational company also providing the consultancy support
- **ISV:** the direct user of the PaaS development and deployment solution
- **consumer:** the ISV customer who consumes the ISV's cloud-provided software

The results here originate from PaaS migration case studies. Specifically, we considered a PaaS solution by a major established multinational, providing a platform based on globally distributed datacentres. We focused on ISVs that are SMEs (small-to-medium size enterprise). The concerns in Figure 1 and the process in Figure 2 are extracted from this setting.

While Figure 1 characterises concerns that allow us to categorise PaaS migration as a distinguishable process, Figure 2 outlines the main structure of the migration process, with the following individual steps:

- 1) **Consultation:** The transfer is often forced by an ISV's end customer. As the latter are the ISV's PaaS customers, as a consequence, a SaaS/PaaS alignment is required, making the situation more complex than the previous SaaS case. Also, PaaS providers are often companies that work with partners (consultancies), which can be involved at all levels.
- 2) **Infrastructure:** The technology focus is on determining the elements of the existing IT development infrastructure and any dependencies between the components. A dependency analysis (e.g. network scanning to detect dependencies) needs to be carried out – includes applications and machines. Already virtualised solutions are easier to move.
- 3) **Development:** Development using a PaaS environment requires major changes regarding the architecture design and software development approach:
 - Statelessness is a requirement for VMs and applications to be deployable without data.
 - As a consequence, data externalisation is required to prepare for scale-out, which necessitates externalisation for an efficient management of elasticity requirements.

a) PaaS Migration Steps:1) **Consultation with ISV CEO**

Focus	Business.
From-To	FROM classical licensing model TO SaaS.

2) **ISV PaaS Infrastructure Assessment and Requirements**

Focus	Technical.
From-To	FROM local TO virtualised (self-hosted, or better 3rd party-hosted) TO public configuration (3rd-party hosted, data centres).

3) **ISV Developer and Software Development**

Focus	Technical.
From-To	FROM Traditional OO/SC/Server Architectures TO cloud PaaS architecture.

4) **ISV Provisioning**

Focus	Business.
From-To	FROM Installation TO PaaS access channels (for clients).

b) PaaS Migration – Vision & Ignorance:

Vision: costs, sales and marketing are the drivers for an ISV to adopt cloud development. Value proposition is that hosting is outsourced, i.e. no more management of infrastructure is required.

Ignorance: while in general benefits and concerns such as security or failure are clear (and would be covered in SLA negotiations), some major changes result as implications that are not fully understood. This includes changed cash flow from a reliable up-front licensing model to a more unreliable pay-as-you-go or post-usage billing. Another major aspect overlooked is a required skills change. This applies to IT managers in particular, e.g. in relation to security technology like firewalls, but also the developers themselves, as we will discuss later on.

Fig. 1. PaaS Migration – a) Steps and b) Vision & Ignorance.

Consequently, this requires the developer to change development approaches, possibly in a significant way depending on the current development approach.

- 4) *Provisioning*: The business focus reflects the transfer of cloud advantages from the ISV (at the PaaS-level) to the client (at the SaaS-level). Corresponding access channels to the new cloud deployment platform need to be provided.

Tool support and other supporting techniques would be beneficial, but due to the increased complexity (compared to most structured data for SaaS or more standardised structures of VMs and their manipulation for IaaS) would have less of an impact [8]. Commonly used are question catalogues (typical are around 100 questions that help to capture the current and the envisioned development architectures) used manually by PaaS provider consultants in the early stages of the migration process. Sometimes these questions are organised into decision trees to guide and focus their application [28]. These help to organise the migration process and identify potential problems or can highlight possible improvements. However, the concern is more the organisation of the migration process and the identification of software problems, rather than their rectification.

Some other observations are noteworthy. Often, as our interviews have established, the business stakeholders can be difficult to convince, while it is easier with IT staff. This indicates that changed cash flow is initially perceived as more of a problem than IT development and re-skilling. Despite this observation, the architecture complexity (Stage 3) is often underestimated by the migration client, as migration consultants have reported.

Success criteria that are applied here are expenditure-based and end-customer numbers (of ISVs) as metrics, indicating the primary driver of the migration. Often, the migration is partial, meaning that an integration of on-premise legacy and cloud solutions is necessary. However, these concerns will not be further explored here. Quality improvements of assumed full migrations are possible through re-architecting and re-programming techniques such as code transformation for statelessness.

Our observations from our empirical PaaS migration process analysis can be summarised as follows:

- A basic solution – without quality gains – would replicate the existing server in a virtual world, i.e. essentially just running and maintaining applications elsewhere in the cloud, but not delivering performance gains (and other quality improvements) that the cloud allows. An oversized resource is the consequence of this approach. Current desktop virtualisation is a such a simplification. The problem is recognised, but solutions are essentially aims to avoid benefits and simplify.
- Stateless programming and the separation of data from processing have already been mentioned [29]. Another aspect apart from statelessness and data externalisation refers to data as well [12]: no-SQL data representation. Most applications are developed around a relational representation of data in SQL-based databases. This is, however, not ideal in federated, scalable cloud solutions. These are optional, i.e. there is no need unless scale-out and resource efficiency is required.
- As part of a migration process, the architecture and

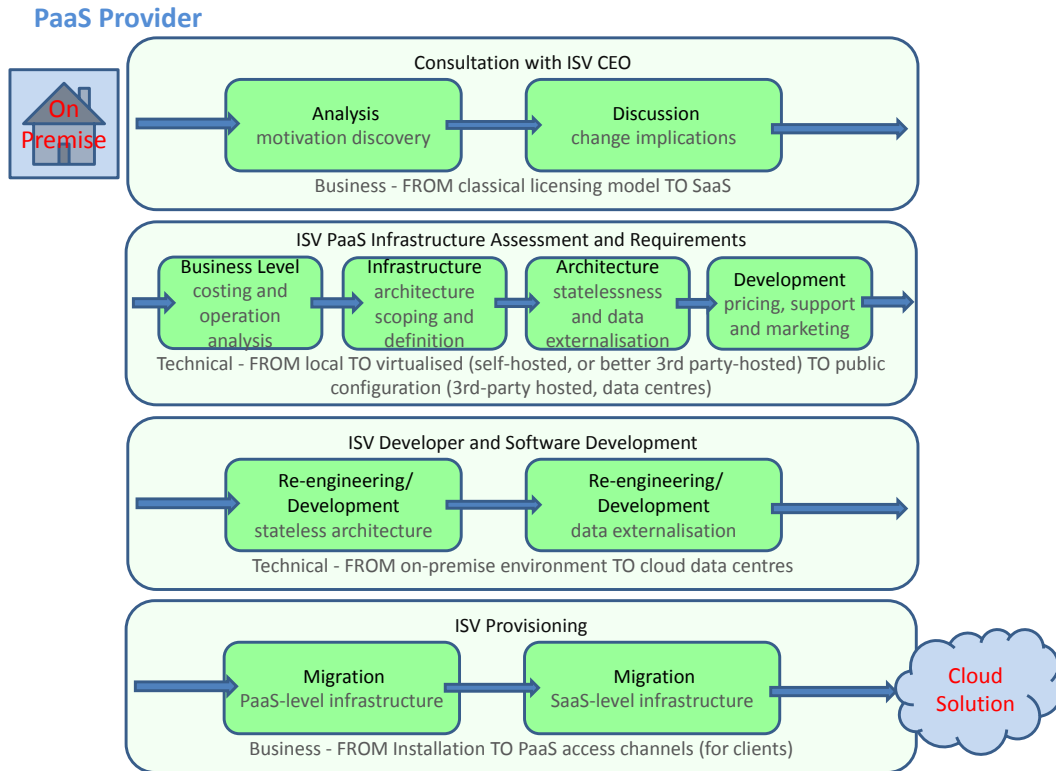


Fig. 2. PaaS Migration – Process Perspective.

programming solutions need to be embedded into:

- methodological process support, e.g. through enhanced questionnaires (about infrastructure used, complexity of programs and data, etc),
- analytical and diagnostic tools supporting dependency analysis and other preparations for re-engineering and refactoring [15], [31].

Overall, a productivity support system (an architecture decision support system) that aims at scalability (scale-out) and performance as a cloud benefit would be the ultimate aim.

- Technology solutions might be hampered by skills shortages, resulting in training needs.

This identifies possible software architecture and engineering techniques (through software re-engineering or refactoring) that would help to achieve possible cloud benefits. It also identifies the forms in which this support can be facilitated, i.e., methodological and tool support.

B. Systems Integrator

We look at another case study complementing the first in-depth one (PaaS-level, multinational, proprietary product, SME client). This second investigation that goes beyond the focus of the first, has a validating role for us as a number of concerns already addressed above will reappear here. Again, the focus is on-premise to cloud PaaS migration. The company has acted as a consultant supporting a service solution provider, for instance an ISV-SaaS provider of accounting systems using a PaaS platform like Google Apps. Systems integration refers to

the utilisation of infrastructure technologies for communication and processes to be supported [6].

In the following, we reflect on observations of a systems integrator across a number of migration projects of different types and complexity.

- 1) *Customer – Business Analysis.* As before, the main expected benefit and driver is cost reduction. Regarding the PaaS migration, there is the confirmation that cash flows change, but equally important, there is no hybrid between a software licence model and a SaaS model. A clear decision needs to be taken by an ISV providing applications via SaaS. The initial consultation with the client involves a discussion covering the following aspects and later steps of a migration process: feature requirements determination, user acceptance testing, data migration, go-live discussion and contract discussion.
- 2) *Customer – Architecture Design.* This step aims at the determination of the architectural scope of the system to be migrated. As in other complex cases, an incremental execution of the technical migration is the preferred solution, e.g. starting with the top 5 out of 50 features, adding features in regular extensions. For a consultant, platform (and provider) selection is the first major decision. Tradeoffs need to be considered. For instance at the PaaS level, Microsoft Azure supports a variety of development languages such as Python and, of course, the .NET platform, but is less advanced in terms of marketplace solutions. Another dimension in

choosing a provider are the different product lines, such as premium, standard and free, offering choice to the client, but also affecting the finances.

- 3) *Customer – Mapping and Moving*. The final stage addresses a number of business and technical concerns. This go-to-market stage needs an IT team to be prepared and put in place to carry out the actual technical migration. Architecture emerges again as a critical concern, which we have already noted. Architecture problems do occur. Improperly architected solutions in terms of integration or quality concerns need to be targeted. There is often a need (but also an opportunity) to redesign a cloud architecture solution (e.g. storage costs are often underestimated, which either requires unnecessary garbage data to be removed or a differently configured solution to be used in order to improve performance and reduce resource utilisation). Possibilities for configuration that cloud solutions offer are often, at least initially, not considered. Changes in the cloud solution architecture would anyway require an agile approach to architecture re-engineering as some problems might only emerge throughout the migration itself. Legacy management is another architecture concern to be looked at from the outset. Decommissioning of systems components needs to take place. Backup systems are useful at early migration stage, but provide consistency problems later on and incur costs.

Overall, the concerns from the first case study are confirmed in terms of expectations, problems and migration support activities – although at least at the surface, the activities are slightly differently grouped. This case study also validates the principle activities in the migration process and re-establishes architecture as the critical technical concern.

IV. RELATED WORK

Related research in cloud migration support in terms of methodologies and techniques focuses for instance on model-driven methodology and tools for reuse and migration of on-premises applications to the cloud. To support the migration, the REMICS project – an EU research project under the FP7 framework – enhances the OMG Architecture Driven Modernization (ADM) methodology with specific methods, meta-models and tool support, including knowledge discovery, patterns and transformations for SOA and cloud computing. The REMICS project looks at a model-driven solution to support primarily SaaS-level application migration [?], [17], [23]. Part of the work concentrates on the opportunity for software modernisation – a challenge, but also opportunities that we have also identified. Their architecture-driven modernisation extracts the legacy architecture in order for it to form the basis of the new cloud solution. Automated translation of business process, component and test specifications preserves existing features, also allowing weaving in new architectural elements in a coherent way. A defined process governs the sequence of activities. Similar architecture-driven change and evolution support and architecture manipulation has been presented in [1], [24], [2]. Our concern here was less to provide a concrete software architecture solution – rather to look at concerns from a broader perspective. We have identified, based on empirical

evidence, the challenges for a cloud PaaS-oriented migration technique.

We defined PaaS migration activities as a structured process. Some practical guidelines exist, published by cloud solution providers to aid the migration to their own products:

- Amazon provides whitepapers for its Amazon Web services solutions (including e.g. EC2) [30]. These AWS whitepapers outline the target architecture, their interfaces and also migration aspects.
- Similar documents are published by other companies, such as IBM [4], Salesforce [27] or Oracle [21], [22]
- Microsoft as another major provider in the PaaS space also provides whitepapers [16]. Additionally, technology evangelists like David Chappell provide material in various media types [5].

However, a generalisation of migration processes beyond individual solutions is lacking.

Data migration [18] emerges as one of the more mature migration concerns. Other aspects that can be tool supported are cost estimation [26], which can also alleviate initial concerns.

V. DISCUSSION AND CONCLUSIONS

Migration of IT infrastructures from on-premise solutions to the cloud raises currently a range of questions. Common procedures do not exist and tool support is not available for most concerns. Migration experts rely on their own experience and some basic tools to facilitate the process. We have aimed to identify commonalities in the PaaS migration process, using the cloud IaaS/PaaS/SaaS layers as the primary differentiation factor to extract PaaS-specific concerns and represent them as a generic migration process. Distinguishing between different SaaS application categories (classical ERP versus technical software) make sense as well as distinguishing more clearly between PaaS development and deployment or IaaS compute, storage and networking capabilities.

We have established a migration process framework outlining major steps and their concerns. This has served as a basis to extract critical problems in business and technical terms. We particularly have seen architecture emerging as a technical problem. The discussion of case studies featuring in the expert interviews we have conducted has highlighted the immaturity in terms of established procedures and availability of tools to support the architecture migration process. Important challenges arising from our observations include the importance of an adequate architecture design in terms of statelessness or data separation for the cloud environment, but also the implications in terms of changed business models.

Two classical software architecture concerns need to be embraced by a PaaS architecture migration process solution:

- architecture re-engineering, i.e. the adaptation and modification of a software solution to meet the requirements of elastic and resilient cloud provisioning,
- architecture ageing, i.e. the validation and correction of programming and data aspects aiming at incorrect and outdated structures.

Our aim was to define a conceptual framework. This framework outlines a generic PaaS migration process. Focusing on the architecture concerns in this process, the framework can organise re-engineering and refactoring techniques.

Our future work will further focus on architectural re-engineering, covering more experimental work to document the benefits in a more quantitative form, but also include an investigation into tool support for this problem. Some specific problems that have emerged need to be addressed. The incrementality of the required software re-engineering solution for complex migrations also needs to be considered. Incremental migration is needed for the scale of most on-premise to cloud migration projects. In addition, our investigation, which has focused here on a single deployment solution in the cloud, could be expanded to multi-cloud PaaS solutions – again a concern necessitated by large-scale migrations.

ACKNOWLEDGEMENTS

The authors are greatly indebted to Andrew Macadam and Kurt Claeys from Microsoft Ireland and Microsoft Belgium for discussions about migration to MS Azure, to Richie Bowden from RB Consulting and to Gavin Guinane from Salesforce.

The research work described in this paper was supported by the Irish Centre for Cloud Computing and Commerce, an Irish national Technology Centre funded by Enterprise Ireland and the Irish Industrial Development Authority.

REFERENCES

- [1] A. Ahmad, P. Jamshidi and C. Pahl. Graph-based pattern identification from architecture change logs. *Advanced Information Systems Engineering Workshops*, pp. 200-213. Springer. 2012.
- [2] A.A. Almonaies, J.R. Cordy and T.R. Dean. Legacy system evolution towards service-oriented architecture. *International Workshop on SOA Migration and Evolution*. pp. 53-62. 2010.
- [3] R. Barrett, L.M. Patcas, J. Murphy and C. Pahl. Model Driven Distribution Pattern Design for Dynamic Web Service Compositions. *International Conference on Web Engineering ICWE06*. Pages 129-136. Palo Alto, US. ACM Press. 2006.
- [4] M. Behrendt et al. IBM Cloud Computing Reference Architecture 2.0. [Online]. Available: https://www.ibm.com/developerworks/mydeveloperworks/blogs/c2028fdc-41fe-4493-8257-33a59069fa04/entry/chapter_13_cloud_computing_reference_architecture1?lang=en. 2011.
- [5] D. Chappell. How SaaS Changes an ISV's Business, Sponsored by Microsoft Corporation. [Online]. Available: http://www.davidchappell.com/writing/white_papers/How_SaaS_Changes_an_ISVs_Business-Chappell_v1.0.pdf. 2012.
- [6] J.K. Halvey and B.M. Melby. *Business Process Outsourcing: Processes, Strategies, and Contracts*. John Wiley & Sons. 2007.
- [7] IBM. Rational Software Development Platform – Migrate from VisualAge Generator 4.5 to the Enterprise Generation Language (EGL). [Online]. Available: <http://www.ibm.com/developerworks/rational/library/egldoc.html>. 2012.
- [8] H. In. Conflict Identification and Resolution for Software Attribute Requirements. PhD dissertation. University of Southern California. 1998.
- [9] S. Jin. Cloud Architecture Patterns. vCloud VMWare. [Online]. Available: <http://blogs.vmware.com/vcloud/2011/01/cloud-architecture-patterns-vm-template.html>. 2010.
- [10] K. Kontogiannis and Y. Zou. *Reengineering Legacy Systems Towards Web Environments. Managing Corporate Information Systems Evolution and Maintenance*. Idea Group Publishing. 2005.
- [11] E. Lee, B. Lee, W. Shin and C. Wu. A reengineering process for migrating from an object-oriented legacy system to a component-based system. *Proceedings 27th Annual International Computer Software and Applications Conference COMPSAC 2003*. pp. 336-341. IEEE. 2003.
- [12] J. Lu. Reengineering of database applications to EJB based architecture. In *Advanced Information Systems Engineering*. pp. 361-376. Springer. 2006.
- [13] Y. Maezawa, H. Washizaki and S. Honiden. Extracting Interaction-Based Stateful Behavior in Rich Internet Applications. *16th European Conference on Software Maintenance and Reengineering (CSMR) 2012*. pp. 423-428. 2012.
- [14] P. Mel and T. Grance. The NIST Definition of Cloud Computing, NIST Special Publication 800-145. NIST. 2010.
- [15] T. Mens and T. Tourwe. A survey of software refactoring. *IEEE Transactions on Software Engineering*, vol.30, no.2, pp.126-139. 2004.
- [16] Microsoft. Server and Cloud Platform. [Online]. Available: <http://www.microsoft.com/en-us/server-cloud/default.aspx>. 2013.
- [17] P. Mohagheghi, A.J. Berre, A. Sadovykh, F. Barbier and G. Benguria. Reuse and Migration of Legacy Systems to Interoperable Cloud Services – The REMICS project. In *Proceedings of Mda4ServiceCloud'10 at ECMFA 2010*. 2010.
- [18] S. Mohanty. Data Migration Strategy. [Online]. Available: <http://www.information-management.com/specialreports/20040518/1003611-1.html>. 2004.
- [19] S. Murtaza and R.A. Masud. An Extended and Granular Classification of Cloud's Taxonomy and Services. *International Journal of Soft Computing and Engineering*, 2(2), pp.278-286. 2012.
- [20] K. North. SQL, NoSQL or SomeSQL?. Dr. Dobb's Blogs. [Online]. Available: <http://www.drdoobs.com/database/sql-nosql-or-somesql/228701075>. 2010.
- [21] Oracle. SQL Developer Migration. [Online]. Available: <http://www.oracle.com/technetwork/products/migration/sqldeveloper/migration21-wp-131240.pdf>. 2010.
- [22] Oracle. After Migrating or Upgrading the Database. [Online]. Available: http://docs.oracle.com/cd/A87860_01/doc/server.817/a86632/migaftm.htm. 2012.
- [23] L. Orue-Echevarria, J. Alonso, M. Escalante and G. Benguria. Moving to SaaS: Building a migration strategy from concept to deployment. In: A.C. Ionita, M. Litoiu, G. Lewis (Eds). *Migrating Legacy Applications: Challenges in Service Oriented Architecture and Cloud Computing Environments*. IGI. 2012.
- [24] C. Pahl, S. Giesecke and W. Hasselbring. An Ontology-based Approach for Modelling Architectural Styles. *Proceedings European Conference on Software Architecture ECSA2007*. Springer. 2007.
- [25] C. Pahl, H. Xiong and R. Walshe. A Comparison of On-premise to Cloud Migration Approaches. *Proceedings European Conference on Service-Oriented and Cloud Computing ESOC 2013*. Springer. 2013.
- [26] RightScale. PlanForCloud. [Online]. Available: <http://www.planforcloud.com/>. 2013.
- [27] Salesforce. Developerforce Tools. [Online]. Available: <http://wiki.developerforce.com/page/Tools>. 2013.
- [28] M. Skilton and P. Gordon. Cloud Buyers' Decision Tree. The Open Group. 2010.
- [29] H.M. Sneed. Generation of stateless components from procedural programs for reuse in a distributed system. *Proceedings of the Fourth European Conference on Software Maintenance and Reengineering 2000*. pp.183-188. 2000.
- [30] J. Varia. Migrating your Existing Applications to the AWS Cloud: A Phase-driven Approach to Cloud Migration. *AWS Cloud Computing Whitepapers*. [Online]. Available: <http://media.amazonwebservices.com/CloudMigration-main.pdf>. 2011.
- [31] U. Zdun. Reengineering to the web: A reference architecture. *Proceedings Sixth European Conference on Software Maintenance and Reengineering 2002*. pp. 164-173. IEEE. 2002.