NOVA IMS Information Management School

MGI

Mestrado em Gestão de Informação
Master Program in Information Management

Analysis of requirements and technologies to migrate software development to the PaaS model

Fabiano Rosa

Dissertation submitted as partial requirement for obtaining the Master's degree in Information Management

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

ANALYSIS OF REQUIREMENTS AND TECHNOLOGIES TO MIGRATE SOFTWARE DEVELOPMENT TO THE PAAS MODEL

by

Fabiano Rosa

Dissertation submitted as partial requirement for obtaining the Master's degree in Information Management, with a specialization in Information Systems and Technologies Management.

Advisor: Vítor Manuel Pereira Duarte dos Santos, PhD

November 2018

ABSTRACT

Software development has been evolving during the last years and, more and more, the software architecture to support this development has become more complex to meet the new requirements and new technologies. With the new cloud computing architecture and models, IT departments and ISV are developing new applications and moving the traditional software architecture to the cloud. In this context, Platform as a Service (PaaS) model can provide software development services and components within a new architecture for building a new generation of software with all benefits of cloud, like scalability and elasticity.

However, currently, most companies have significant challenges to adapt and change its software development process to use the PaaS architecture and the cloud services. In this dissertation, it will first be identified and analyzed the changes and challenges for develop software with the PaaS architecture. Afterwards, will be analyzed and identified the requirements in a traditional software development and architecture (on premise) to development new software or adapt the existents software with the PaaS.

KEYWORDS

Cloud Computing; PaaS; Software Development; Software Architecture; Enterprise IT; Migration

INDEX

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

ALM	Application Lifecycle Management
API	Application Programing Interface
B2B	Business to Business
BI	Business Intelligence
BPM	Business Process Management
CEP	Complex-event processing
DBMS	Database Management Systems
DSR	Design Science Research
ESB	Enterprise Service Bus
IaaS	Infrastructure as a Service
IDE	Integrated Development Environment
IEC	International Electrotechnical Commission
IMC	In-memory Computing
IoT	Internet of Things
ISO	International Organization for Standardization
ISV	Independent Software Vendor
IT	Information Technology
NIST	National Institute of Standards and Technology
PaaS	Platform as a Service
RDBMS	Relational Database Management System
SaaS	Software as a Service
SDLC	Software Development Life Cycle
SDK	Software Development Kit
SLA	Service Level Agreement
VDI	Virtual Desktop Interface

1. INTRODUCTION

Cloud computing has been modifying the software development and the traditional IT and business, introducing new ways to develop software with more scalability and agility with lower costs, allowing companies from startups to big corporations to create new business models or improve the existents ones.

Due the grown of cloud computing adoption, PaaS has become an important part of cloud economy and has been showing the potential of a service model with a platform that support the entire lifecycle of an application, from development, test, deploy and operations, with components, tools and integrated services (Beimborn, Miletzki, & Wenzel, 2011).

Many organizations are planning and migrating their on premise software to the cloud (Pahl & Xiong, 2013), starting with the IaaS and SaaS model, but they are facing some challenges and difficulties, mainly in the PaaS model, related to the complexity in integrating the legacy and internal systems from their IT area to the new PaaS architecture model and services on the cloud.

1.1. BACKGROUND AND PROBLEM IDENTIFICATION

The Platform as a Service (PaaS) is a cloud service model that provide capabilities to deploy onto cloud applications created with languages and tools supported by the provider and use and integrate application infrastructure service in the application to the cloud consumers.

In the PaaS model, the foundation infrastructure (network, servers, operating systems, storage, etc.) are manage by the cloud provider, but the cloud consumer has control over the deployed applications, configurations and lifecycle (Mell & Grance, 2011).

PaaS delivers application infrastructure capabilities (middleware) to the software development process, such as runtime and development components, as cloud services like:

- Application Development, Data, Workflow, etc.
- Security Services (Single Sign-On, Authentication, etc.)
- Database Management Systems (DBMSs)
- Directory Services
- Integration middleware
- Business process management (BPM) platforms
- Rules engine
- Complex-event processing (CEP)
- In-memory computing (IMC) platforms

According the Harvey Nash / KPMG CIO Survey (KPMG, 2016), the adoption of PaaS is predicted to grow from 32% in 2017 to 56% in 2020, represented in Figure 1, indicate that the PaaS adoption will be fastest-growing of cloud platforms and is now the mainstream, accelerating as enterprises shift software and platform operations to the cloud.

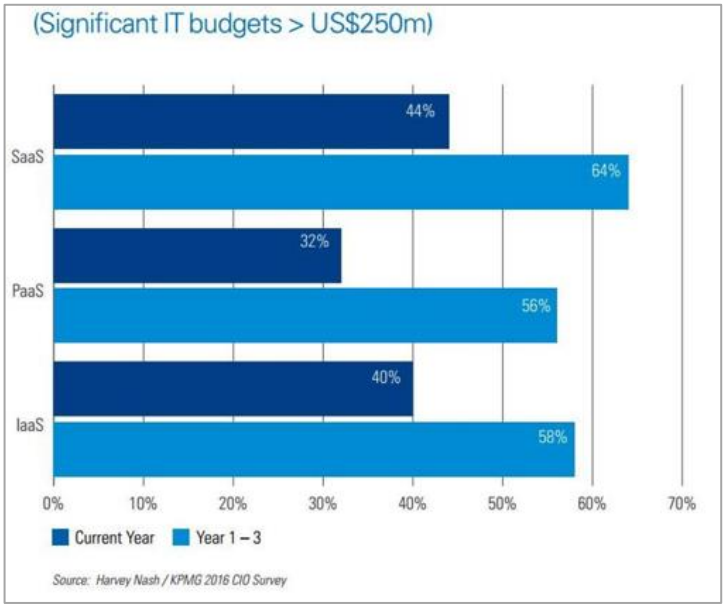

Figure 1 – Cloud investment by type (KPMG, 2016)

The cloud computing and PaaS model architecture brings new requirements and complexity to the software development process, since the new technologies, services and tools available in PaaS, to the adaptation of the traditional software engineering and development process to use new techniques like DevOps, native cloud application model and hybrid cloud deployment (on premise to cloud integration), until the new methodologies, like Agile, used in this new software development scenario to be considered in the traditional software project management practices.

In this fastest-growing scenario, the ISV and IT department of companies need to be able to evaluate the PaaS model and architecture to adapt your software development and architecture, usually complex with many legacy systems, many integrations and different technologies to this new software development scenario.

This brief exposure establishes the ground for this current work and brings us to the main objectives which will be described in the next section.

1.2. Study Relevance and Importance

The proposed project work is justified by a rational of three axes, namely: Technology; Academic Literature; and, Business.

Technology - The software development, deployment and lifecycle management is being transformed by Cloud computing, and the PaaS model have a huge opportunity and potential to support this transformation, introducing new technologies, services and facilities for companies in their application development (Cohen, 2013), and this study will have a theoretical application based on the needs of PaaS model.

Academic Literature - Recent research review on cloud computing, PaaS and migration to the cloud concludes that companies often has difficulties and challenges to move to the cloud model mainly related to misconceptions about benefits and risks, besides the unclear expectations from ISVs to

implement software in PaaS (Fowley, Elango, Magar, & Pahl, 2017). This work aims enrich the available academic literature, reviewing and creating new knowledge about software development in PaaS.

Business - In a scenario with a fastest-growing PaaS adoption prediction (KPMG, 2016), this study can contribute to the advancement of knowledge of PaaS software development, helping companies, ISV and IT departments to planning and mitigate the risks of the migrations and new developments in cloud model.

After the completion of this work, the recommendations resulted can be used as reference or guidelines for future cloud migration projects, helping to reducing costs and setup time. With the fastest-growing PaaS adoption the need of a reference architecture or guidelines to support the companies will be fundamental to guide the cloud migration, reducing errors, rework or an inadequate cloud architecture.

1.3. STUDY OBJECTIVES

The main purpose of this work is to understand and evaluate the requirements and changes in software development process using the PaaS model, trying to help the IT development areas, ISV and companies to plan and migrate your traditional development software and architecture to operate in this cloud model.

The main question to be answered in this study: **What are the requirements and technology changes to be considered in software development to migrate to the PaaS model?**

For this purpose, specific goals will be target to accomplish the main objective of this work:

1. Study of the cloud computing and PaaS model, architecture and requirements to develop software in cloud.
2. Study of the information systems and their relation with information technology and business applications.
3. Study of the methodologies, tools and technologies required in software engineering for software development in cloud.
4. Identify and analyze the main concerns in software development related to migration application development from on premise to PaaS.
5. Propose recommendations to use the PaaS model in development software, including the required technologies and architectures.
6. Validate the proposal with an interview with specialists in software development and cloud computing.
7. Register and document the conclusions.

2. LITERATURE REVIEW

In the literature review are presented the related works with the proposal research, focusing on the cloud computing concepts, information systems, software development and migration to the PaaS model.

2.1. CLOUD COMPUTING

2.1.1. Cloud Computing Definition

Cloud computing, one of the latest information technology (IT) innovations, is still taking shape and gaining maturity, and at this stage different people, from the technology area to the sales area, have different views and concepts about it, based on your interests and how to apply and use cloud computing in your daily life (Ruparelia, 2016). Although the different definitions of cloud computing, the National Institute of Standards and Technology (NIST), and the International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC), which publish technology standards, have a publication regarding the definition of cloud computing at NIST Special Publication 800-145 and ISO/IEC 17788:2014(E), respectively, where the NIST's publication is generally well accepted (Mogull et al., 2017).

The NIST's publication defines cloud computing as:

> *"Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management e ort or service provider interaction.",* (Mell & Grance, 2011).

Likewise, the ISO/IEC's publication defines cloud computing as:

> *"Cloud computing is a paradigm for enabling network access to a scalable and elastic pool of shareable physical or virtual resources with self-service provisioning and administration on-demand.",* (ISO/IEC, 2014).

In short, cloud computing represents a disruptive technology with potential to increase agility, collaboration, availability and scalability through a new operation model and with a set of technologies for managing computing resources and shared pools. This allows for optimized and efficient computing that results in reduced costs. In the cloud computing model, components can be orchestrated, provisioned implemented and decommissioned rapidly and with scalability, thus allowing to allocate and consume resources on-demand (Mogull et al., 2017).

The cloud computing model proposed by NIST, with the five essential characteristics, three cloud service models, and four cloud deployment models, is represented in the Figure 2.

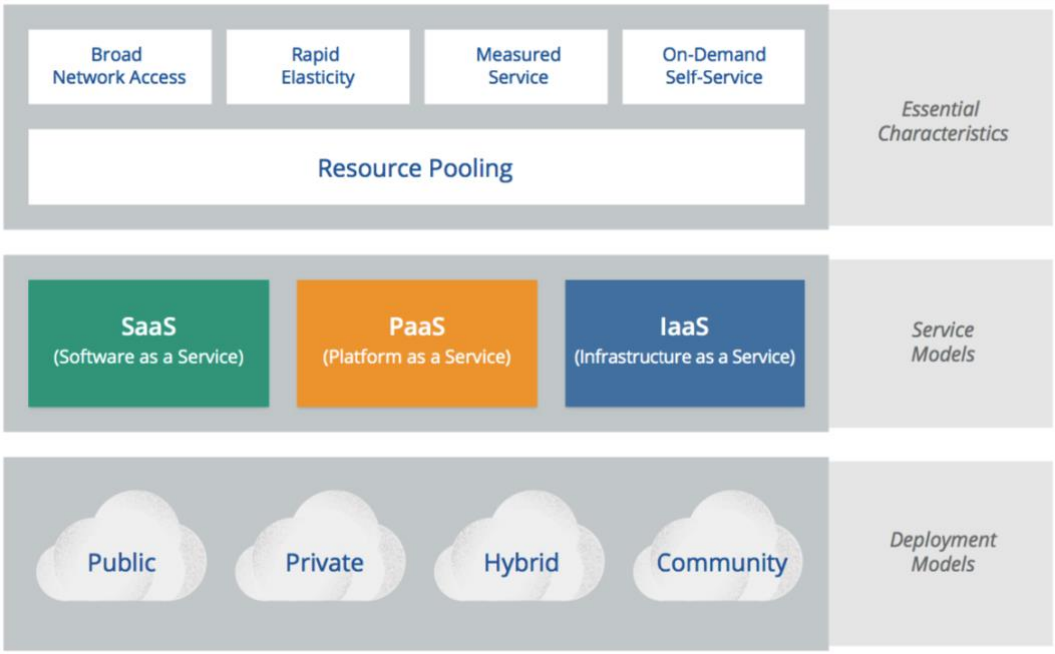

Figure 2 - Cloud computing model from NIST (Mogull et al., 2017)

2.1.2. Essentials Characteristics of Cloud Computing

To considered as cloud computing, it is necessary at least the five essentials characteristics described below. The lack of these characteristics disregards it as cloud computing (Mogull et al., 2017).

- **Broad Network Access:** to access the resources over the network, is not necessary a direct physical access, so the network can be abstracted in the cloud service.
- **Rapid Elasticity:** the resources from a pool can be expanded or contracted by consumers (provisioning and provisioning) usually automatically, giving the flexibility to consume resources according the demand, like adding virtual server when the demand increases and releasing it when it is no longer needed.
- **Measured Service:** gives the service consumer the opportunity to measure what is provided, so using and paying only the services that were consumed, like the paying for what was used in the water and electricity business model.
- **On-Demand Self-Service:** allows service consumers to provision the resources from a pool themselves, without a human intervention in an automatized way.
- **Resource Pooling: i**s the most relevant characteristic, because gives the provider to abstract resources and to collect them into a shared pool, and these resources can be allocated to different consumers.

A consolidate view on the characteristics of cloud computing, with a parameter indicating the purpose of each characteristic, is proposed by Ruparelia (2016) in Table 1.

Characteristic	Description	Parameter
Broad network access	Consume services from anywhere	Where
On-demand self-service	Consume services when you want	When
Resource pooling and virtualization	Pool the infrastructure, virtual platforms, and applications	How
Rapid elasticity	Share pooled resources to enable horizontal scalability	How
Measured service	Pay for the service you consume as you consume it	How much

Table 1 - Characteristics of cloud computing (Ruparelia, 2016)

2.1.3. Service Models of Cloud Computing

The cloud computing provides the service model as Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS), where each of them provide a specific level of abstraction, that allows the service consumers to build and deploy systems with a reduced effort. In each model, the IT tasks from a traditional on premise structure can be abstracted and automatized in each related model of cloud computing, such as build and install software, manage and operate servers and ensure the required security levels (Kavis, 2014).

The Infrastructure as a Service (IaaS) model provides the basic infrastructure resources, like processing, memory, storage and network as a virtualized or physical server. The cloud consumers are free to install, configure and manage operations systems, software and applications themselves, since the cloud provider is responsible to provide the required IT resources for that. The Platform as a Service (PaaS) model provide a set of tools, software and resources that allows the cloud consumer build and run applications on the top of a configured and managed infrastructure, abstracting the IaaS layer. In this model, the cloud provider is responsible for the basic infrastructure and also for the software, tools and resources, providing to the cloud consumers a complete "stack" for the software development on the cloud. Software as a Service (SaaS) model offer one or more applications to the cloud consumers, where these applications are hosted and managed by the cloud provider. In this model, the cloud consumers have a high level of abstraction related with the infrastructure (IaaS) and the application platform (PaaS), not having to manage this resources, but rather just configure and use the application. (Bond, 2015).

The different cloud computing service models (IaaS, PaaS and SaaS) can be categorized according the layers of a traditional application stack which is used by the IT, described below, and the relation with application stack items and the cloud computing service models are represented in Figure 3 (Fehling, Leymann, Retter, Schupeck, & Arbitter, 2014).

- **Physical hardware:** the physical IT infrastructure that contains the servers, storage and network connections that are used in a traditional data center.
- **Virtual hardware:** the physical hardware components that are abstracted in virtual components and that can share the same physical hardware components between multiples virtual counterparts.

- **Operating system:** software that provide basic functions to access network, filesystems and external devices, like printers and mouse. Microsoft Windows, Linux and MacOS are examples of operating systems.
- **Middleware:** environment for installation and execution of custom data, processes and applications, that contains the required software and operating system from running programming languages, such Java, .NET and Pyton, to provide a more complex execution environment, like the applications servers such as JBoss, IBM Websphere, Oracle, and databases like IBM DB2, MySQL, SAP HANA and PostgreSQL.
- **Application software:** custom applications that provide to the end users functionalities such as customer relationship management (CRM) and enterprise resource planning (ERP).
- **Business process:** the process, like order processing, credit approval and billing, of the company that are support by the applications.

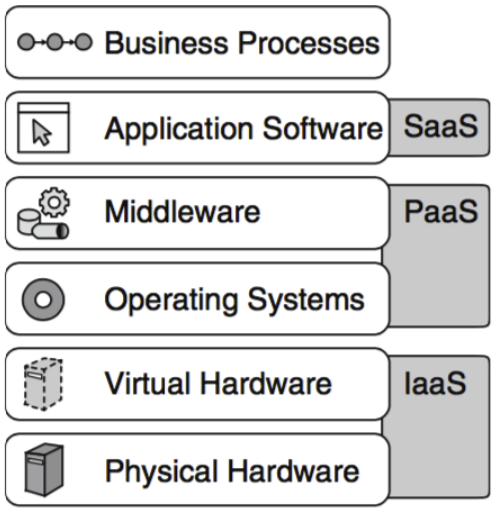

Figure 3 - Application stack and associated cloud service models (Fehling et al., 2014)

2.1.4. Deployment Models of Cloud Computing

A cloud computing environment can be deployed and hosted in a cloud provider in different forms, independent of the service model offered by the cloud provider. The cloud deployment models are defined according the target user groups and the IT resources sharing needs, and it can be categorized regarding the accessibility: public when is generally accessible to everyone, private when is accessible only by a single institution, community where the access is controlled by a specific group or institution and hybrid it combines the others models (Fehling et al., 2014).

Although the definition of deployment models was confusing at the beginning of cloud computing industry, these models are still categorized based on the NIST definition, and in a more modern representation, it can include a new deployment model, as described below (Bond, 2015).

- **Private cloud:** operated for a single organization, when it can be managed by the organization or a thirty party. This model is more flexible to manage and customize and usually is deployed in an existent on-premises datacenter.
- **Public cloud:** offered to a general public by a cloud provider, that operates and manages the computing resources. As these resources are shared to different consumers, the customization is limited.
- **Hybrid cloud:** a combination of two or more deployment models. A private cloud that is connected to one or more public cloud services is a common example of a hybrid cloud.
- **Community cloud:** provided for an organizations or community of user with shared concerns or interests, usually when it is needed share mission, policies, governance and security requirements.
- **Virtual private cloud:** the latest model which is a combination of public and private cloud, that allow provide a private cloud (virtual) to a dedicated customer within a public cloud, bring some advantages and flexibility to the cloud consumers related to the price, customization and security.

The relation between the different cloud computing deployment models and how they may be connected in an enterprise cloud consumer context is represented in the Figure 4.

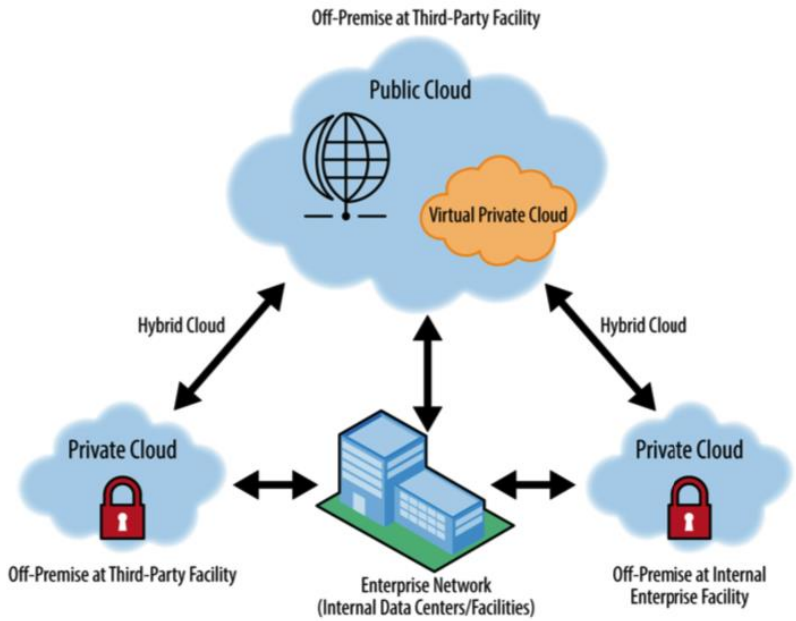

Figure 4 - Private, public and hybrid cloud integration (Bond, 2015)

2.2. Information Systems

More and more, Information Systems (IS) have become a vital component to the success of the companies and organizations, and it is integrated into our daily business activities like accounting,

finance, operations management, marketing human resource management and other key business functions. Also, Information technologies (IT) are playing a vital role in business, supporting and working together with the Information systems, improving the efficiency and effectiveness of the business process, managerial decision making and workgroup collaboration (O'Brien & Marakas, 2010).

An information system (IS) can be understood with a simple definition, like O'Brien & Marakas defined:

> *"Any organized combination of people, hardware, software, communications networks, data resources, and policies and procedures that stores, retrieves, transforms, and disseminates information in an organization. People rely on modern information systems to communicate with one another using a variety of physical devices (hardware), information processing instructions and procedures (software), communications channels (networks), and stored data (data resources) ", (O'Brien & Marakas, 2010).*

2.2.1. Roles of Information Systems in Business

Although there are a huge number of software applications for different areas and functions, there are three fundamental roles of business applications of information systems that support business process and operation, decision making and competitive advantage of the organizations, as represented in Figure 5 (O'Brien & Marakas, 2010).

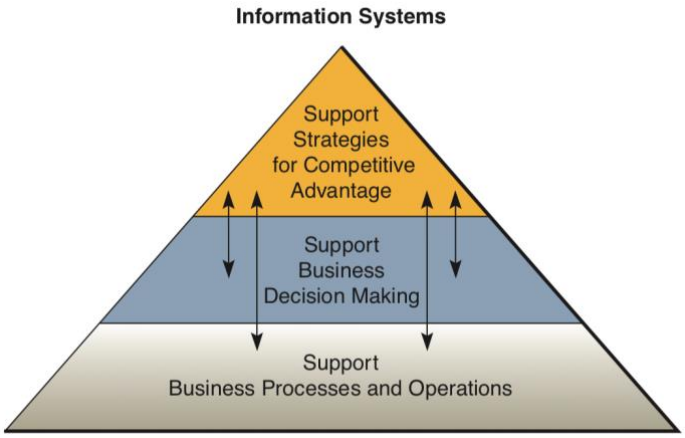

Figure 5 - Fundamental roles of business applications of IS (O'Brien & Marakas, 2010)

The three fundamental roles of information system in a business enterprise context can be implemented and explained with a typical retail store example (O'Brien & Marakas, 2010):

- **Support of business process and operations:** a retail store can use an information system (computer-based) to record customer purchases, inventory tracking, employees' payments and sales trends evaluations.
- **Support of decision making by employees and managers:** store managers and other business professionals can make better decisions through the information

systems, supporting them for example, to decide what lines of merchandise needs to be added or removed.

- **Supporting of strategies for competitive advantage:** a retail store can gain a strategic advantage over the competitors with a innovate application of information technologies, like using touch-screen kiosks in the stores that are connected with the e-commerce web site for online shopping.

2.2.2. Types of Information Systems

Nowadays, the application of information systems in the business world can be classified in different ways and based on different concepts. The several types of information systems can be classified and grouped by operation support or management support, and each on can derivate specialized types based on the role that each one plays in the operations and management of a business (O'Brien & Marakas, 2010).

Based on the Laudon & Laudon, 2018 classification, the information systems can be defined in three main groups:

1. **Transaction processing systems (TPS):** kind of information system that performs and records the daily routine transaction required to run the business. The main purpose of the transaction process systems is to bring answers to routine questions and to track the flow of transactions across the organization. Using transaction process systems, operational managers can keep track of some activities and transactions, like as sales, receipts, cash, deposits, payroll, credit decisions, flow of materials and others (Laudon & Laudon, 2018). The Figure 6 illustrates a transaction process system for payroll processing.

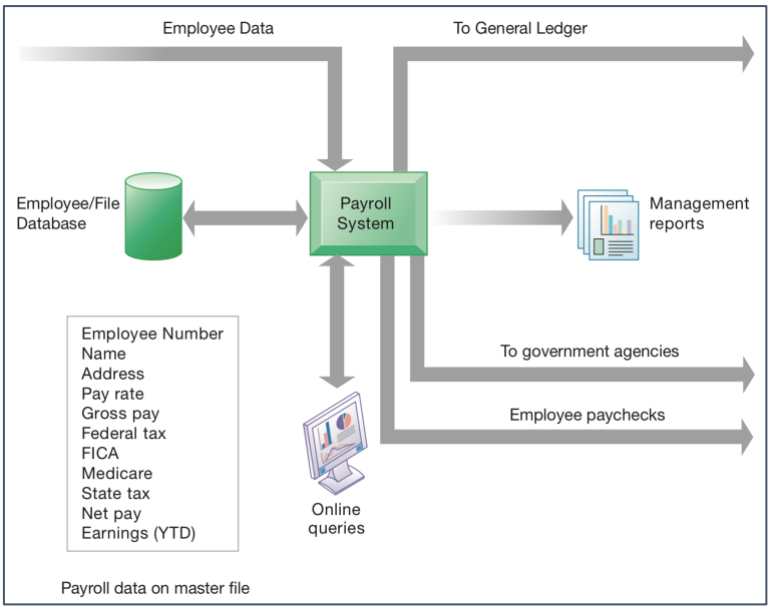

Figure 6 – Example of TPS for payroll processing (Laudon & Laudon, 2018)

A payroll information system keeps track of employee's payments and record in a single transaction employee information like name, social security number and total of hours worked in a specific period. Once this transaction is record in the system, it updates the master file (database) that will maintain the employee information for the organization. Management reports can be created with this data in different ways to send information to the employees (paychecks) and to government agencies. With the TPS, managers can monitor the status of internal operation and the relation with the firm and the external environments, in case of payroll system, the system can supplies data to others account TPS, like a company's general ledger system, which is responsible for recording the income and expenses of the firm and producing reports like balance sheets and income statements (Laudon & Laudon, 2018).

2. **Systems for Business Intelligence (BI):** focus on delivering information to support management decision making, with data and tools for organizing, analyzing and providing access to data to help managers and others key management users. Business Intelligence systems can support different levels or categories of management in an organization, addressing the decision-making needs of all levels. For the middle management, the BI systems are known as **management information systems (MIS)**, which helps the middle managers to monitoring and controlling activities, also taken decision-making based on the activities and information. Besides monitoring and controlling the business, this information is used to predict future performance of the organization, like production, sales and orders. The company's basic operations are summarized and reported through MIS using data supplied by transaction processing systems (Laudon & Laudon, 2018). Figure 7 shows how the managers can access reports with organizational data from MIS, and the TPS supplies summarized data to the MIS reporting system.

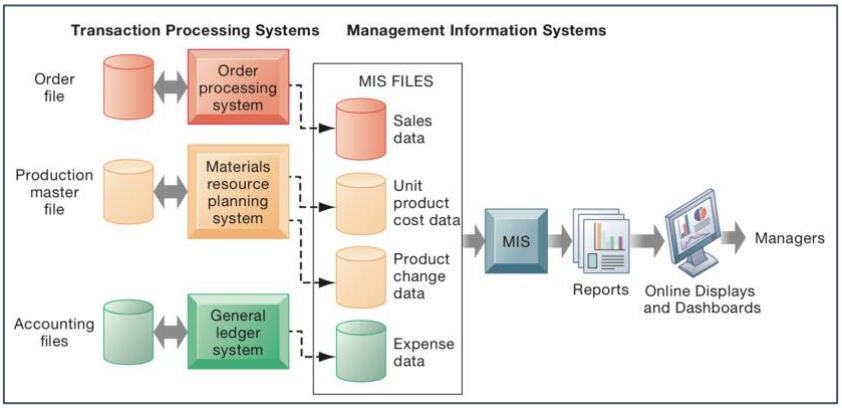

Figure 7 - MIS reporting system consuming data from TPS (Laudon & Laudon, 2018)

The other type of business intelligence system, the **Decision-support systems (DDS)**, focus on supporting more non-routine decision making to the managers in problems that are unique and rapidly changing, such as the impacts on production schedules if the sales double in a specific period or what the return on investment if a factory schedule delays by a few months. A Decision-support system can be operated by a manager in a powerful desktop computer, where he can enter data and obtain estimating information from different source of data combined in analytical models, as represented in Figure 8 (Laudon & Laudon, 2018).

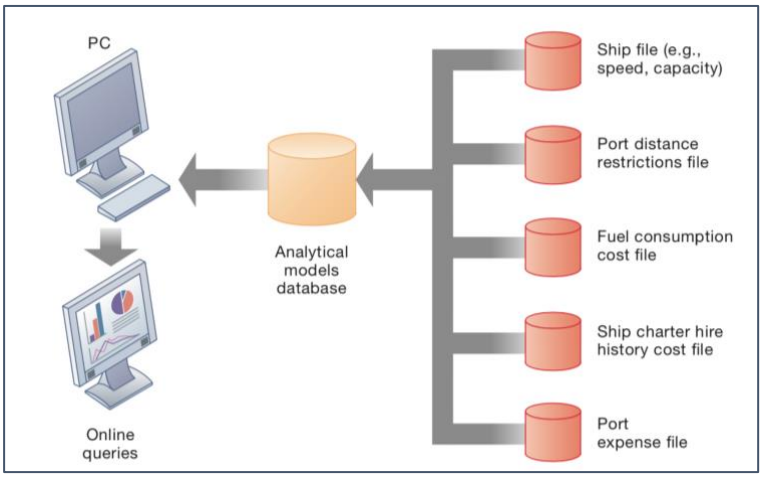

Figure 8 – Decision-support system sample (Laudon & Laudon, 2018)

Business intelligence systems also support the senior management to take decision-making focusing on strategic assumptions and long-terms trends. The **Executive support systems (ESS)** help the senior managers addressing nonroutine decisions, evaluation and insights, through simulations and predictions, such as the projection of production in the next years and the long-term industry cost trend, for instance. The ESS are delivered to senior executives through a portal, usually a web interface, allow the personalized business content to be filter, compress and track, where the data source is from external events and internal MIS and DSS. An executive support system can be provided as a digital dashboard, delivering comprehensive and accurate information to the senior manager using a graphical overview of the information, as represented in Figure 9 (Laudon & Laudon, 2018).

Figure 9 - Digital dashboard for strategical decision-making (Laudon & Laudon, 2018)

3. **Enterprise applications:** systems that reach functional areas in a company, focusing on execute business process across the organization, including all levels of management. These applications focus on efficient management of resources and customer service, integrating and coordinating groups of business process thus helping business process become more flexible and productive. The main enterprise applications are: enterprise systems (ERP), supply chain management, customer relationship management systems and knowledge management systems. Each of them group and integrate a set of functions and business process, thus the organization performance can be enhancement as a whole. Figure 10 represents an enterprise application architecture where the business process and functional areas are integrated across the entire organization (Laudon & Laudon, 2018).

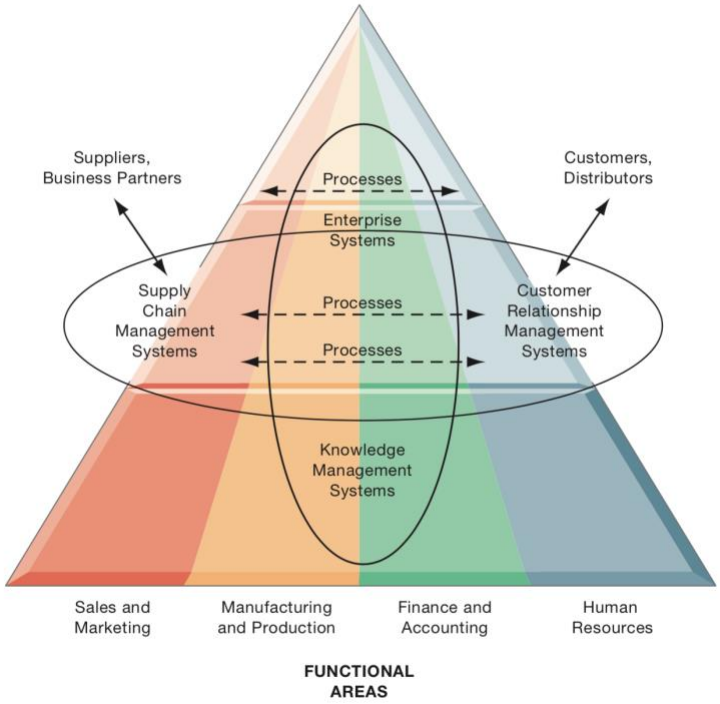

Figure 10 - Enterprise application architecture (Laudon & Laudon, 2018)

Enterprise systems or **enterprise resource planning (ERP)**, are used in organizations as a single software system to integrate business process between the main functional areas, such as manufacturing and production, finance and accounting, sales and marketing and human resources. This allow that the fragmented information from many different systems is stored in a centralized and comprehensive data repository and can be used by different areas of the business in the organization. A typical scenario when the business process performs among the different functional areas of the organization is when a customer places an order, so the order data flow will affect the others parts of the company, like trigger the warehouse to pick the ordered product, send the customer invoice by the accounting department and track the progress of the order by the customer service department (Laudon & Laudon, 2018).

For manage the relationship between the firm and their suppliers, a **supply chain management (SCM)** systems is used for source, produce and deliver goods and services efficiently. This system helps the organization share in an efficient way information about orders, production, inventory levels, and delivery of goods and services between suppliers, purchasing firms, distributors and logistics companies. The main focus is to optimize the production chain with the right quantity of products and the lowest cost (Laudon & Laudon, 2018).

Customer relationship management (CRM) systems are used by organizations to manage the relationship between the firm and their customers, providing information related with all business process that treat with customers in sales, marketing,

customer satisfaction and retention. The main gains expected from a CRM system are increase sales and improve the services to the customers (Laudon & Laudon, 2018).

Some organizations differentiate themselves from other companies using better the knowledge about how to create, produce and delivery products and services, using the **knowledge management systems (KMS)**, where it helps to improve the capturing and applying of knowledge and expertise. As in most cases, the organization knowledge is unique and difficult to imitate, it turns into a strategical information to help the company to be more competitive and differentiate in market in a long-term. (Laudon & Laudon, 2018).

2.3. SOFTWARE DEVELOPMENT

Over the past 50 years, software became the most important technology in the world and the key for the computer-based systems evolution. To support the advance of the software, the Software Engineering, with its process, methods and tools, has been supporting the software deployment to be carry out quality and timeliness (Pressman, 2009). But still today, the biggest challenge of software engineering remains to understand, satisfy and solve the needs of users or clients with software (Jalote, 2005).

Thus, process is the foundation layer for software engineering with activities to be carried out regardless complexity, size or domain in a software development. A generic process framework for software engineering comprise these well-defined phases with a set of activities (Zaigham & Saqib, 2013):

- **Communication phase:** focus on requirements gathering and specification and the project initiation.
- **Planning phase:** preparation of the schedule, effort estimation and task duration of the project, tracking the planned schedule across the subsequent phases.
- **Modelling phase:** contains the project analysis and design, with the requirements modeling, prototype, blueprint and fine grain level details needed for the software coding.
- **Construction phase:** software codification with design implementation, unit testing, code integration and the final product (application).
- **Deployment phase:** delivery, supporting and maintenance of the deployed software.

A systematic sequential approach to software development which contemplates the phases and activities described above is proposed by the waterfall model, sometimes called the classic life cycle, is represented in Figure 11 (Pressman, 2009). In this model, the main activities

required in a software development, is considering in each related phase, in this way supporting the entire Software Development Life Cycle (SDLC).

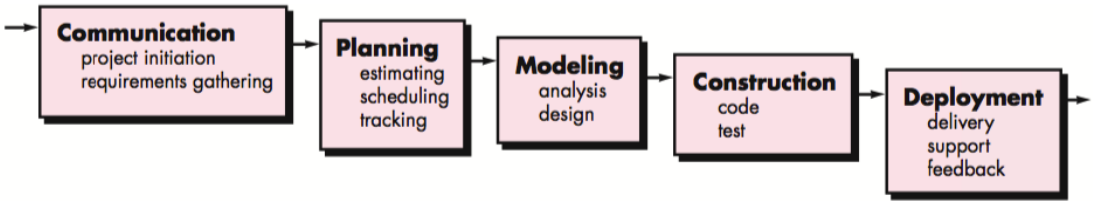

Figure 11 - The waterfall model (Pressman, 2009)

2.3.1. Software Architecture

When the word *architecture* started to be used in the software development, it was referenced because of the similarities with the construction industry, where to do the architecture of a building requires a lot of planning before takes place the construction. So, it was first used in the Waterfall software development process model to design the software components and elements, dependencies and relationships to build a software before any code to be written (Ingeno, 2018).

The standard definition from ISO/IEC/IEEE 42010 publication defines software architecture as:

> *"Fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution.",* (ISO/IEC/IEEE, 2011).

Also, the ISO/IEC/IEEE 42010 standard definition describe the following main aspects of software architecture:

- It is a fundamental part of a software system.
- The environment of a software system and its operation is taken in consideration to define the software architecture.
- The software architecture documentation describes the architecture of the software system and communicate to stakeholders how this architecture meets the system requirements and needs.
- One or more concerns of the stakeholders are covered by the architecture views described in the software architecture documentation.

Among the more commonly used software architecture used in software development projects, the main ones are:

- **Client-Server**

  As known as a two-tier architecture, the client-server architecture is a layered architecture where client and servers communicate with each other directly: the client request some resource or call some procedure or service and the server responds to the client requests. The **client** layer is the application part that contains the user interface code and some logic and the server layer contains a database, usually a relational database management system (RDBMS)

with the application data and the business application logic. In the **server** side, the business application logic is implemented with database stored procedures and database triggers, and its logic is directly related to the data model in database (Ingeno, 2018). This architecture is commonly represented as the Figure 12.

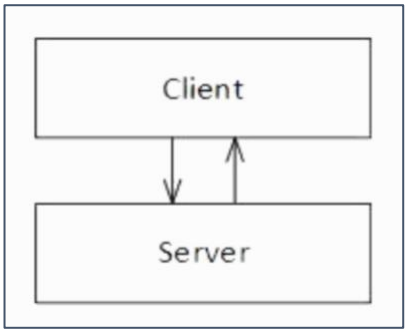

Figure 12 - Client-Server architecture (Ingeno, 2018)

Besides some stored procedures advantages, like database optimization and security management, this software architecture has some limitations related to coding construction because the business logic could be not centralized and so spread between the client and the server (Ingeno, 2018).

- **N-Tier**

The n-tier architecture, also known as a multitier architecture, rise with the web applications, where the most common variant used is the three-tier architecture, and the main driver to start to use this architecture was the needs to shift the client-server architecture to a new architecture that could support the requirements and needs of the web application, mainly because of the use of the web browsers, where the client logic needed to move from a client to another layer (Ingeno, 2018). The three-tier architecture defined by the 3 distinct layers (presentation, business and client) is represented in Figure 13:

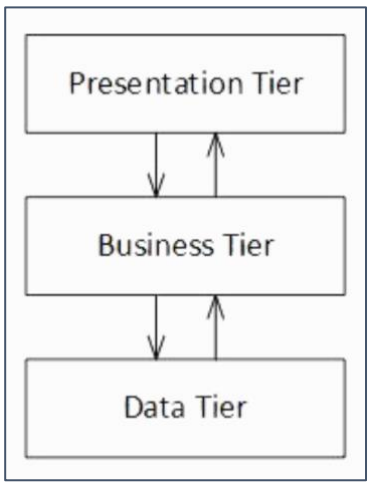

Figure 13 - N-tier architecture represented as three-tier architecture (Ingeno, 2018)

The **presentation tier** is the application user interface and provides a visual design for the end user interact and read and input data and usually contains the logic to render the interface and

some basic input validations to minimize user input mistakes and ensuring the correct types and formats of the data presents to the user. The **business tier** provides the implementation of the business application logic, including some logics as business rules, validations and calculation logic, also the business entities that represents the domain entities for the application. This tier serves as an intermediary between the presentation and data tiers, providing services and data to the presentation that it retrieved and manipulated from the data tier. The **data tier** is responsible for access and manage the data, usually through a persistent storage, such as a RDBMS or abstracted by an object-relational mapping layer that mapping relational tables (entities) to objects for using in the application code implementation. (Ingeno, 2018).

- **Monolithic**

A monolithic architecture is commonly implemented in applications that work as a single and self-contained unit, where the components of the monolithic application are inter-connected and interdependent, therefore the components and code are tightly coupled. In this software architecture, the different layers and concerns, like use interface, business logic, authorization and database access are kept together in the application (Ingeno, 2018).

The applications that are built following the monolithic architecture are considered as a single unit, even its usually contains the three main parts: a client-user interface (HTML pages that run in a web browser), a database (many tables in a relational database management system) and a server-side application that handle the client requests (HTTP), execute the business logic and retrieve and manipulate data from the database (Lewis & Fowler, 2014). A typical monolithic architecture is represented in Figure 14.

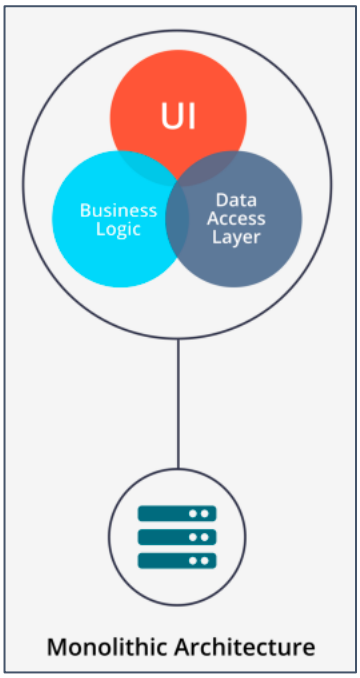

Figure 14 - Monolithic architecture (Malav, 2016)

But for modern applications that are projected and implemented focusing the cloud, this software architecture bring a lot of problems related with deployment, scaling, huge code base

to be maintained, hard to delivery new features and problems with the coupling between components (Malav, 2016).

- **Microservices**

The term *microservices* has become a buzzword and has been gaining focus in IT recently for describing multi-nodes distributed software architectures. Microservices Architecture stand in contrast to monolithic architecture, which focus to build all of the functionality of an application in a single unit deployable, because aims to describe a system with many different components running in separated process and communicating each other over a defined API (Burns, 2018). The smaller services and the communication and integration between the components in a Microservice Architecture is represented in Figure 15.

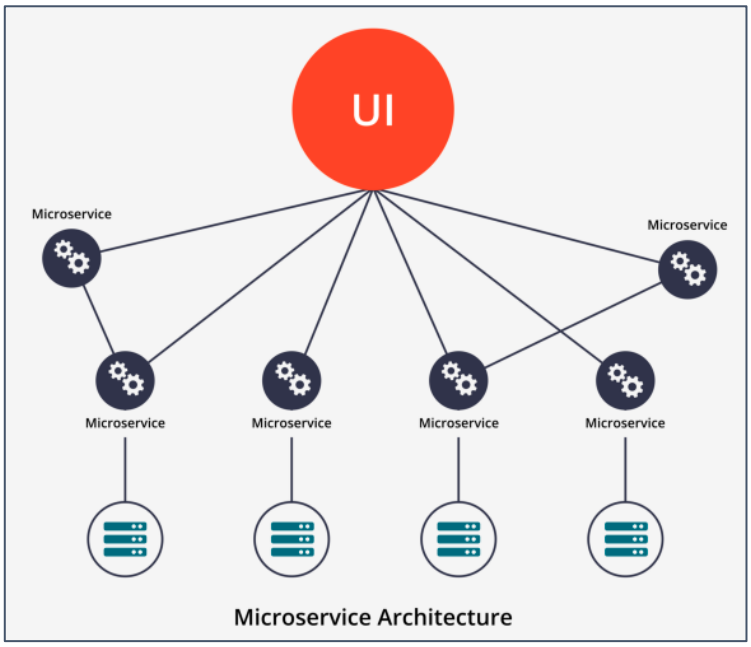

Figure 15 - Microservice Architecture (Malav, 2016)

The approach of the Microservice Architecture (MSA) is to building software with the business domain models decomposed into smaller, consistent and bounded-context implemented as services, in an isolated and autonomous way with a communication that's provide a piece of business functionality between them. With this architecture approach, the implementation of the microservices can be operated by small teams with a certain autonomy, so the impacts with internal changes in the services are minimal across the system (Posta, 2016). The independence and agility in the implementation of microservices by different teams are represented in Figure 16.

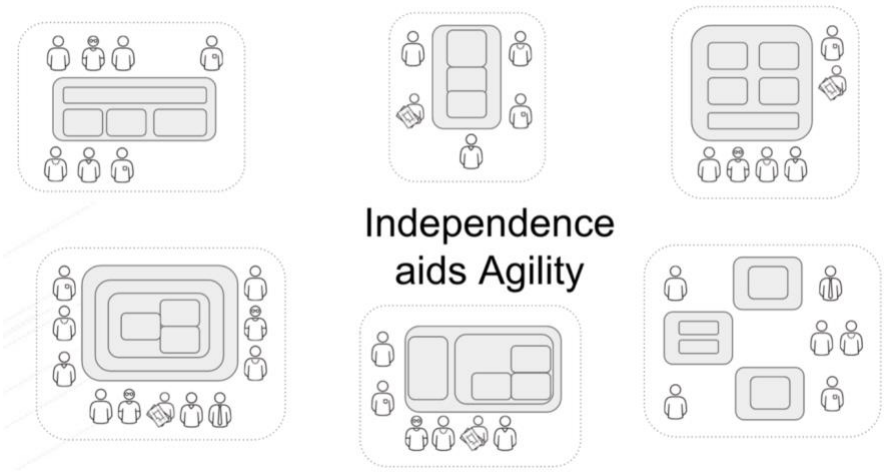

Figure 16 - Small and independent teams with Microservices (Posta, 2016)

In some aspects, Microservices Architecture is the natural evolution of Service Oriented Architecture (SOA), but with some differences between these architectures. The main characteristics that differentiate Microservice Architecture are: services are small, independent and loosely coupled; each service can be managed by a small development team and have a separate codebase; the teams can release a new version of a specific service without rebuilding and redeploying the entire application; the data or the external state of a service is persisted and maintained by the own service; services using a well-defined API to communicate with each other and each services can be implemented with your own technology stack, libraries and frameworks (Microsoft, 2017).

2.3.2. Software Development Life Cycle in the Cloud

When it develops applications for the cloud environment, the traditional software development life cycle is followed by the same phases for the cloud development, but with specifics tasks that target the cloud infrastructure, requirements and architecture. In the cloud context, the software development required besides the software development team, which implements the application for the end users, the cloud provider to provide and maintain the cloud infrastructure to host the application. The specific phases and tasks for cloud are described below, followed by the relation between the traditional software development life cycle and the cloud development life cycle is represented in Figure 17 (Joshi & Pandey, 2017).

- **Cloud Requirement (Requirement):** based on the non-functional requirements of the application, the cloud service provider evaluate if the cloud platform (IaaS, PaaS or SaaS) requirements, like resource availability, maximum number of access of users, downtime/uptime, services, components and tools, will attend the project and application needs.
- **Analysis of Cloud requirement (Analysis):** analyze the software specifications and details to create a plan to develop the application in the cloud. Based on this analysis, a risk analysis and a migration plan are made for the project, including the monitoring, management mitigation, schedule and costs analysis.

- **Cloud Design (Design):** the cloud service provider does a design considering the load balance and performance concerns for the application in a cloud environment, based on the development blueprint done.
- **Implementation (Development):** the cloud service provider implements the code onto web services and components on the cloud platform.
- **Testing:** after the unit tests was done by the development team, the cloud service provider does the integration tests on the cloud platform.
- **Deployment/Maintenance:** the cloud service provider takes control over the application delivery and deploy. The support and maintenance are done based on the Service Level Agreement (SLA) established between the solution provider and the cloud provider.

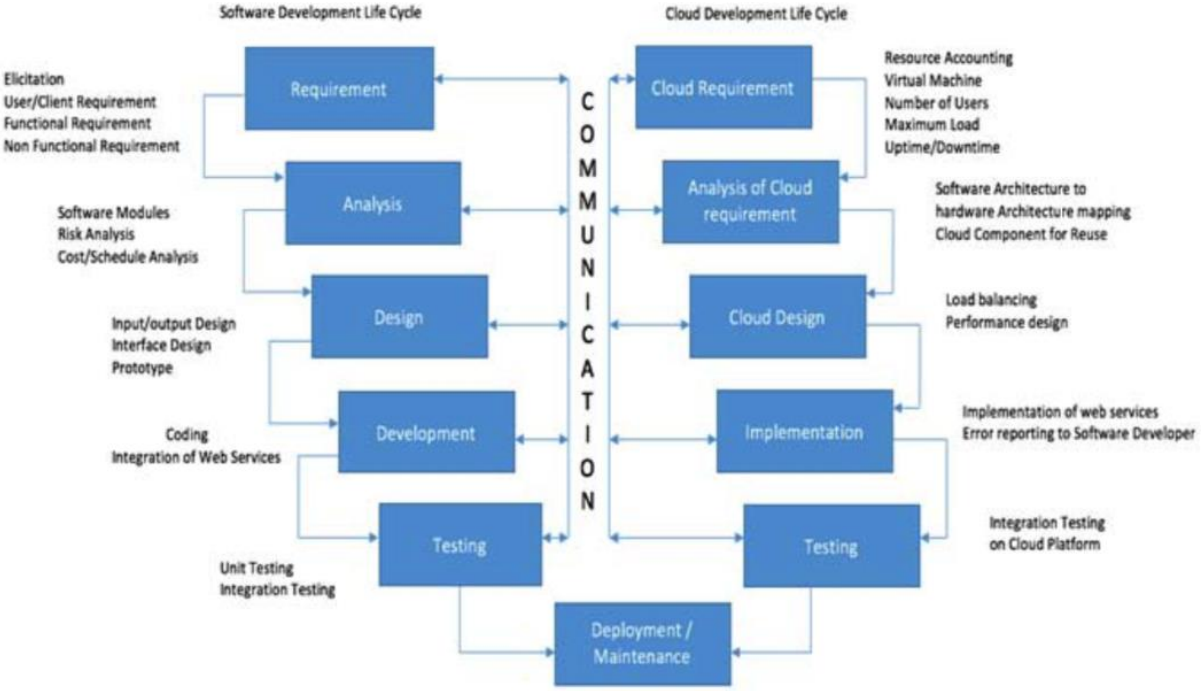

Figure 17 - Software development in cloud (Joshi & Pandey, 2017)

2.4. SOFTWARE DEVELOPMENT WITH PAAS

Given the evolution of the cloud computing architecture and technologies, and the benefits of its automated elasticity and new tools, cloud computing is transforming the software development, since the deployment to the lifecycle management of the software. Many large enterprises have a complex and costly IT portfolio to manage, usually with 3.000 to 5.000 applications, and with the cloud computing solutions, companies can reduce the maintenance costs of these applications moving the legacy applications, which are less critical and with lower value to the business, to an application platform in the cloud, or a Platform as a Service (PaaS) (Cohen, 2013).

In the Platform as a Service (PaaS) service model, cloud providers deliver a platform to the cloud consumers that allows to develop, run and manage applications in a cloud environment, without the need to maintain a own required infrastructure for the software development software (Violino, 2017).

2.4.1. Platform as a Service (PaaS)

Considered as the next level of abstraction of the cloud stack, after the Infrastructure as a Service (IaaS), which target the basic IT infrastructure, Platform as a Service (PaaS) provides the functions from the application stack as services, allowing developers to design and build solutions using platform services for caching, asynchronous messaging, database, and more. In this way, developers don't need to reinvent the wheel implementing from scratch this "commodities" requirements and can focus on the business logic from the system (Kavis, 2014).

The NIST's publication defines Platform as a Service (PaaS) as:

> *"The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting environment.",* (Mell & Grance, 2011).

However, the Platform as a Service (PaaS) is the hardest service model to characterize due to the many ways to building services and the wide range of offerings of PaaS providers. An additional layer of services related to development frameworks, middleware, databases, messaging ad queuing is added to the PaaS as a integration layer, so the applications can be built to platform with the supported development languages and tools (Mogull et al., 2017).

Platform as a Service (PaaS) is also considered as "Application infrastructure" cloud services, which included runtime and development-time components as services for the software development, and contains technologies such as (Natis et al., 2016):

- Application servers
- Database management systems (DBMSs)
- Integration middleware
- Business process management (BPM) platforms
- Rule engines
- Complex-event processing (CEP)
- In-memory computing (IMC) platforms

A PaaS is frequently built as a platform on top of IaaS as a layer of integration and middleware where the consumers can use the PaaS through APIs, illustrated in Figure 18. As an example of this model, an instance of database management system can be deployed and executed on IaaS and exposed to the cloud consumers with API instead of network protocols, so the developers can consume it as a Database as a Service (Mogull et al., 2017).

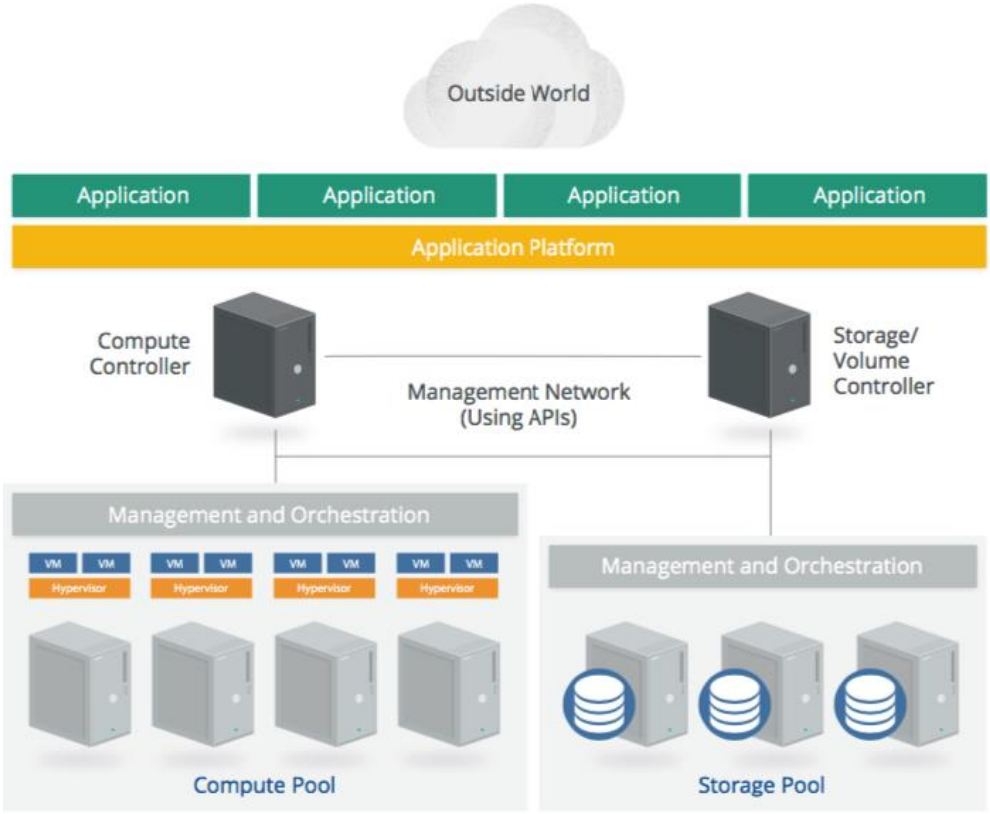

Figure 18 - PaaS built on top of IaaS (Mogull et al., 2017)

Some Platform as a Service (PaaS) offers can abstract the software execution environment, where the user explicitly has the control of the software installation, compilation and execution, providing a software platform with an SDK for the developers build applications and deploy them on the PaaS. In this way, the responsible for running the applications, process the external service requests and execute other dependent services is up to the platform, in addition to allowing share application servers and resources across different users. The Figure 19 show a PaaS sharing applications, process and resources between different users. (Wang, He, & Wang, 2012).

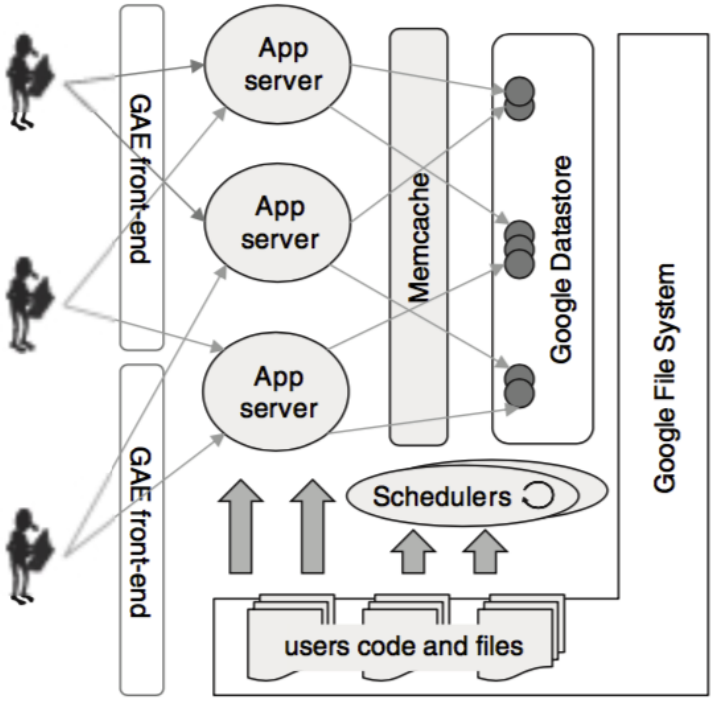

Figure 19 - PaaS sharing application resources between different users (Wang et al., 2012)

Another use for PaaS is to provide a set of technologies and a platform to develop and operate Software as a Service (SaaS) applications, and many PaaS platforms additionally provide services for marketing and sales opportunities to aggregate value in the software value chain. As represented in Figure 20, the central component of the PaaS platform is the Application Runtime Environment (ARE), which add requirement of scalability, reliability and security to the platform. The platform runs over an infrastructure, typically using an Infrastructure as a Service (IaaS) foundation, and over the database and middleware components required for the platform. For the application development and customization, an Integrated Development Environment (IDE) which supports multiples programing languages and includes a variety of libraries and tools for the entire development life cycle, such as modeling, implementation, testing and versioning. Besides the PaaS platform core components, additional services can be provided by the vendors to promote, monitor, track, manage and ensure the quality of the applications. (Beimborn et al., 2011).

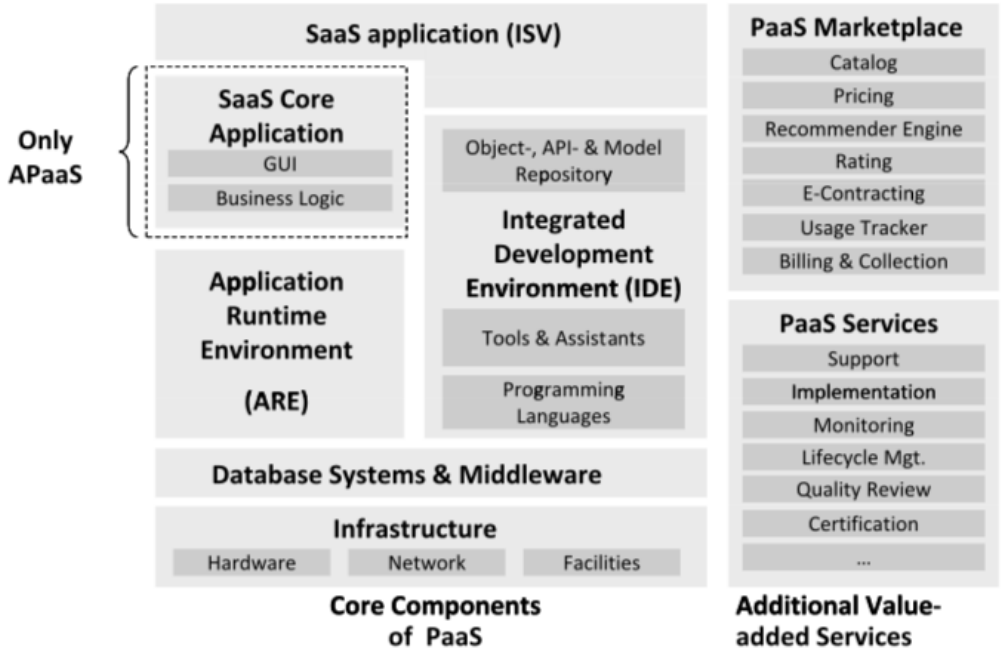

Figure 20 - PaaS stack for SaaS applications (Beimborn et al., 2011)

To close the topic, the characteristics of cloud computing, identified by NIST, for a Platform as a Service (PaaS) are described according Fehling et al. (2014):

- **Access via network (Broad network access):** customer can access the execution environment on the PaaS where the application is hosted and the development environments to build and test the applications via network, through the intranet or the internet.
- **On demand self-service:** a self-service portal or an API are offered in PaaS to customers to deploy their applications. The self-service portal can display the platform and applications availability, consumed resources and statistics for uptime and utilization.
- **Pay-per-use (Measured service):** the billing of the application hosted in the PaaS is performed based on the use of the provided functionality. The cloud provider can bill for the number of applications hosted, per message exchanged, per amount of services requested, etc.
- **Resource pooling (Resource pooling and virtualization):** offering pool resource on the middleware layer, but the sharing isolation depend on the nature of the PaaS offer, built on top of IaaS or native.
- **Rapid elasticity:** as the PaaS are characterized by their sharing capability on the middleware layer, the flexible scaling is delegate to a load balancer, which distributes the user request to the applications between the middleware instances on the PaaS.

2.4.2. Extended PaaS Service Models

Due to cloud computing industry evolution and the increased cloud customer adoption, new service models of cloud computing "as a service" were emerging, known as XaaS that refers to "anything" or "everything" as a service. These new service models, some of them listed in Table 2, that are related with the three original service models of cloud, are being deployed by the public cloud providers and into enterprise private cloud. (Bond, 2015).

Extended service model	Cloud service model	Description
Workplace as a Service; Desktop as a Service	IaaS or PaaS	Cloud service known as virtual desktop interface (VDI) which provides a desktop operational system, like Microsoft Windows or Linux, and installed software to the end users.
Application as a Service	PaaS	Use the VI technology to execute applications in the cloud for end users to consume in desktop or mobile devices.
Development/Testing as a Service (Dev/Test)	PaaS	Allow developers to perform development and testing of application in cloud-based servers with Application Lifecycle Management (ALM) tools included.
Database as a Service	PaaS	Services for databases, search engines and reporting offered as cloud services.
Data as a Service	PaaS or SaaS	Structured or business intelligence data offered in the cloud to consumers perform data analysis.

Table 2 - Cloud extended service models (Bond, 2015)

The Platform as a Service (PaaS) market is growing and becoming segmented with new extended service models for PaaS, and as reported by Gartner (Natis et al., 2016), in 2016 there were 20 specialized PaaS categories from vendors with paying customers, and some new categories will emerge and some categories will be integrated in new PaaS suites in the next years. This growing market of PaaS solutions reflects the evolution of the cloud computing and the search for new solutions in the cloud for software development. The main PaaS categories reported Gartner (Natis et al., 2016) in 2016 are:

- **Application Platform Services (aPaaS)**: is a cloud service focus on general-purpose business applications development, deployment and execution, supporting business logic and data handling for back-end services, web and mobile. This cloud service is composed by tools for development, management, composition and typically data persistence.
- **Business Analytics Platform Services (baPaaS):** cloud-based business analytics platforms offer capabilities to ingest data from different data sources, prepare data for analysis, visualize and analyze data, develop and publish dashboards or other Business Intelligence (BI) outputs. Also, can include advanced analytics offerings, like predictive modeling, machine learning and complex analytics process for budgeting, simulation and optimizations use cases.
- **Business Process Management Services (bpmPaaS):** delivery Business Process Management (BPM) capabilities as a cloud service including a graphical business process and rule modeling capabilities, a process registry and repository to handle the modeling metadata, a process execution environment and rule engine.
- **Business Rule Platform Services (brPaaS):** also referred to as decision management PaaS (dmPaaS), is a cloud-based service that aims to support the decision making by the business rules management. Business rules are implemented in the platform embedded and hidden the business logic from application code and are executed by a business rule engine.
- **Communications Platform Services (cPaaS):** cloud-based solutions that enable applications to integrate or improve communications functionalities, like telephony calling, SMS, MMS, Speech recognition, Mobile browsing, Video services, Conferencing, Recording and many others.
- **Database Platform Services (dbPaaS):** is a cloud service that provide any database management system (DBMS) or data storage engineered as a scalable, elastic and multitenant subscription. This category excludes data stores that run on a cloud infrastructure, but are not built as a cloud service, where the main requirement for DBMS in this category are elasticity and multitenancy.
- **Function Platform Services (fPaaS):** cloud-based service that provides a serverless execution environment for small and event-triggered functions, where it's possible run code without provisioning or managing servers with support to automatically scale process to support increasing and decreasing load. As a fPaaS function is not a complete application and actually is a step in a larger process, it can be comparable to a microservice.

- **Enterprise Horizontal Portal Services (Portal PaaS):** cloud-based service that provide an Enterprise Portal with core portal features like security, personalization, integration, content aggregation and presentation, whit the ability to run in shared, multitenant environments, including private and public cloud deployments.
- **Integration Platform Services (iPaaS):** provides a platform in cloud to support application, data and system-to-system integrations, using a mix of cloud-services, mobile apps, on-premises systems and Internet of Things (IoT) integrations. Capabilities found in enterprise service bus (ESB), data integrations tools, B2B gateways, message-oriented middleware, managed file transfer products and cloud integration interfaces are provided by the iPaaS.
- **Message-Oriented Middleware Services (momPaaS):** cloud-based services focus on provide communication between one part of an application to another or between different applications through the internet, in case of public cloud momPaaS, and between components in the same LAN with the private cloud model. Usually messages are exchange between the parts using common protocols, like AMPQ, MQTT, DDS and XMPP, JMS, WebSocket and APIs Rest.

2.4.3. PaaS solutions and technologies offerings

Given the growing demand for PaaS solutions and the wide segmentation of PaaS service model, there is an extensive listing of PaaS offerings, where it can be consulted in the website https://paasfinder.org. These PaaS offers can be classified by the customer requirements and demand: **PaaS cloud services** if the solution is delivered and maintained by a cloud vendor or **PaaS software** if the customer chooses to install and operate the PaaS solution on-premises. The more commonly used PaaS offerings is listed in the Figure 21 (Cloud Standards Customer Council, 2015).

PaaS Cloud Services	Public	Private	Open Source
IBM Bluemix https://console.ng.bluemix.net/	✔	✔	Cloud Foundry
HP Helion http://www8.hp.com/us/en/cloud/helion-overview.html	✔		Cloud Foundry
Microsoft Azure http://azure.microsoft.com	✔		
Google AppEngine https://cloud.google.com/appengine/	✔		
Amazon Elastic Beanstalk http://aws.amazon.com/elasticbeanstalk/_	✔		
Pivotal Web Services https://run.pivotal.io/	✔		Cloud Foundry
Heroku https://www.heroku.com/home	✔		
Red Hat OpenShift Online https://www.openshift.com	✔		OpenShift
PaaS Software			
Cloud Foundry http://docs.cloudfoundry.org		✔	Cloud Foundry
Red Hat OpenShift http://www.openshift.org/		✔	OpenShift
Windows Azure Pack http://www.microsoft.com/en-gb/server-cloud/products/windows-azure-pack/		✔	

Figure 21 - Most common offers of PaaS (Cloud Standards Customer Council, 2015)

As the PaaS solutions provides one or more programing languages runtimes for the software development, like Java, PHP, Node.js and Ruby, these wide offerings of PaaS often bring the flexibility to the developers to choose their preferred or official IT company language for implementing the applications. The chart in Figure 22 shows the main programing languages runtimes supported by the PaaS solutions and the percentage of use each, of which the Top 5 are Java, PHP, Ruby, Python and Node.js.

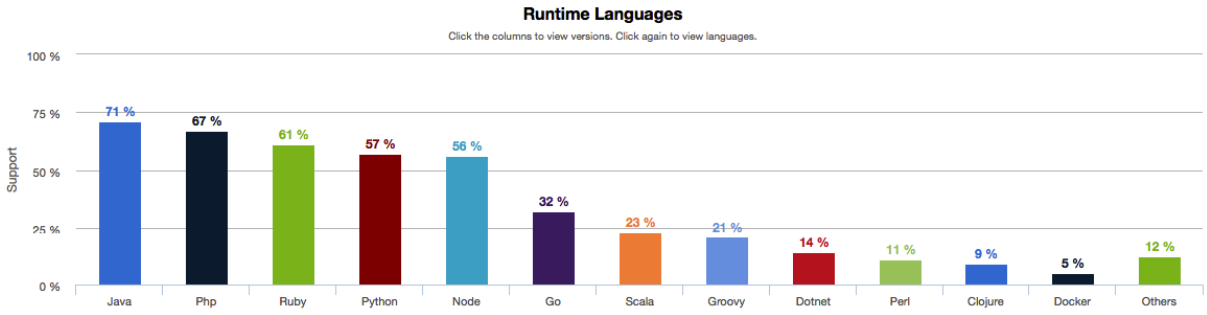

Figure 22 - Runtime languages supported by PaaS solutions (PaaSfinder, 2018)

2.5. MIGRATION

2.5.1. Migration of applications to Cloud

Companies from different industries and segments are taking advantage of cost savings, speed of deployment and scalability of the cloud computing to build and migrate applications to this new environment, besides cloud computing is being seen as a key enabler of business transformation for the companies, helping to engage customers and drive competitive advantage. Nevertheless, is crucial to plan and execute this migration to cloud in a strategic and methodical way (Council, 2013).

However, it is very important for a successfully application migration to cloud to analyze, mitigate and address the main concerns related to cloud computing (Council, 2013):

- **Security:** some risks related of security can be created when moving data and code to third party cloud provider.
- **Loss of control:** a third-party cloud provider will have the entire control of the hardware, software, security policies and the platform.
- **Integration:** internal systems of the companies will be to integrate with the systems in the cloud, bringing more complexity to the application architecture.
- **Availability and reliability of cloud applications:** different issues may appear in the cloud, related to server performance, configuration errors, networking design and application architecture.
- **Cloud service provider lock-in:** adopting one cloud service provider can lock-in the customer and the application with the specific cloud vendor platform and technologies.

Given the IaaS and SaaS maturity and market adoption, Platform as a Service (PaaS) is gaining the attention and is emerging as the new focus of the cloud computing consumers. Basically, the PaaS migration is the process of moving software development, operation and deployment from an on-premise environment to a cloud platform solution, but the architecture complexity and requirements of application development it becomes a challenge in the migration process, because is necessary to understand and follow the cloud-specific architecture and programing techniques to implement an application in the cloud (Pahl & Xiong, 2013).

In a scenario which start-ups and small business requires only a specific programing language to implement the application and run it on the provider platform, the adoption of PaaS is more common and less complexity. However, when we think of a Enterprise scenario, where the organization structure is more complex, with many different architectures, technologies stacks and more than one programing language, the adoption of PaaS is usually slowest (Kavis, 2014).

2.5.2. Migration roadmap and process for PaaS

To help enterprise information technology and business decision makers to analyze and plan the migration of applications to cloud computing, Council (2013) proposed a migration roadmap which it

focus on the transition of the customers data and applications to the cloud computing environments. The roadmap steps for the cloud migration are:

1. **Assess your Applications and Workloads:** to determine what application and data can or cannot be moved to a cloud environment and which deployment model (private, public or hybrid), is necessary assess the application and workloads from the organization. These aspects should be considered in the assessment:
    - Business (changes, risks, resistance, importance).
    - Application Lifecycle.
    - Application Architecture.
    - Data (governance, confidentiality, integrity and quality).
    - Technology.
    - Security.
    - Integration (system, data and presentation layer).
2. **Build the Business Case:** describes the current state and demonstrating the advantages of cloud computing adoption for the applications, including aspects of costs savings, shift of capital expenditures (CAPEX) to operational expenses (OPEX) , SLA, speed of deployment, elasticity, etc. The follow aspects need to be considered in the business case:
    - Cost Analysis:
        - On-going cloud service costs.
        - Service management.
        - License management.
        - Application re-designs.
        - Application deployment and testing.
        - Application maintenance and testing.
        - Application integration.
        - Cost of development cloud skills.
        - Human resources and talent management implications.
    - Service Levels:
        - Application availability.
        - Application performance.
        - Application security.
        - Privacy.
        - Regulatory compliance.
    - Business Impact:
        - Revenue impact.
        - Customer acquisition or engagement impact.
        - User satisfaction.

    - Time to market improvements.
    - Cost of handling peak loads.

3. **Develop the Technical Approach:** the application must be designed for one or more runtime environments in the PaaS service for the migration. The software stack required by the applications need to be evaluated to fit with the many PaaS offers and capabilities. These technical considerations must be considered in the approach:
    - Skills.
    - Security.
    - Integration.
    - Monitoring and Management.
    - Scalability.
    - Availability and Backup.
4. **Adopt a Flexible Integration Model:** the application owner needs to understanding the impact of the connections and integrations with other systems and applications. The integrations should be classified into these three types:
    - Process integration.
    - Data integration.
    - Presentation integration.
5. **Address Security and Privacy Requirements:** the most cloud customers concern issues in a cloud migration are security and privacy, and depending on the industry or sector, these concerns has high priority in the migration. The main security and privacy aspects to be considered are:
    - Steal of confidential data in the cloud provider.
    - Trust in the providers systems administrators.
    - SLA of security measures.
    - Impacts of potentials denial-of-service attacks.
    - Access and authorization of applications.
    - Shared responsibilities between the cloud provider and the cloud customer.
    - Control of malwares in the virtualization platform and cloud management software.
    - Isolation of malwares in multi-tenant environments (platform shared between different customers).
    - Data leak to domestic and foreign law enforcement agencies.

6. **Manage the Migration:** planning, execution and management of the application migration project by the IT department, tracking the project plan aspects such as tasks, durations, resources, costs and risks.

Also, to support the transition and migration of software development and applications to the cloud, Pahl & Xiong (2013) proposed a PaaS migration process, based on several case studies and expert interviews and focus groups, with well-defined top-level activities which can be performed in sequenced steps. The main migration issues were addressed in the process, such as the concern of the activities, the nature of migration (technical or business-oriented), the migration benefits and the potential pitfalls and risks in the migration. Also, the main stakeholders involved in the migration process was considered: the PaaS solution provider, the ISV (direct user of the PaaS for development and deployment) and the consumer of the cloud-provided software. The migration process proposed is represented in Figure 23 and the main structure and steps are:

1. **Consultation:** the ISV's PaaS customer working with the PaaS providers to analysis the requirements and motivations for the migration, and discuss the change implication and impacts.
2. **Infrastructure:** assessment of the main PaaS infrastructure aspects and requirements, such operation costs, infrastructure architecture, pricing, support and marketing.
3. **Development:** identification of the major change in the architecture design and software development regarding the PaaS platform and technologies.
4. **Provisioning:** transferring the cloud advantages from the ISV to the cloud customer, providing access channels to the PaaS platform.

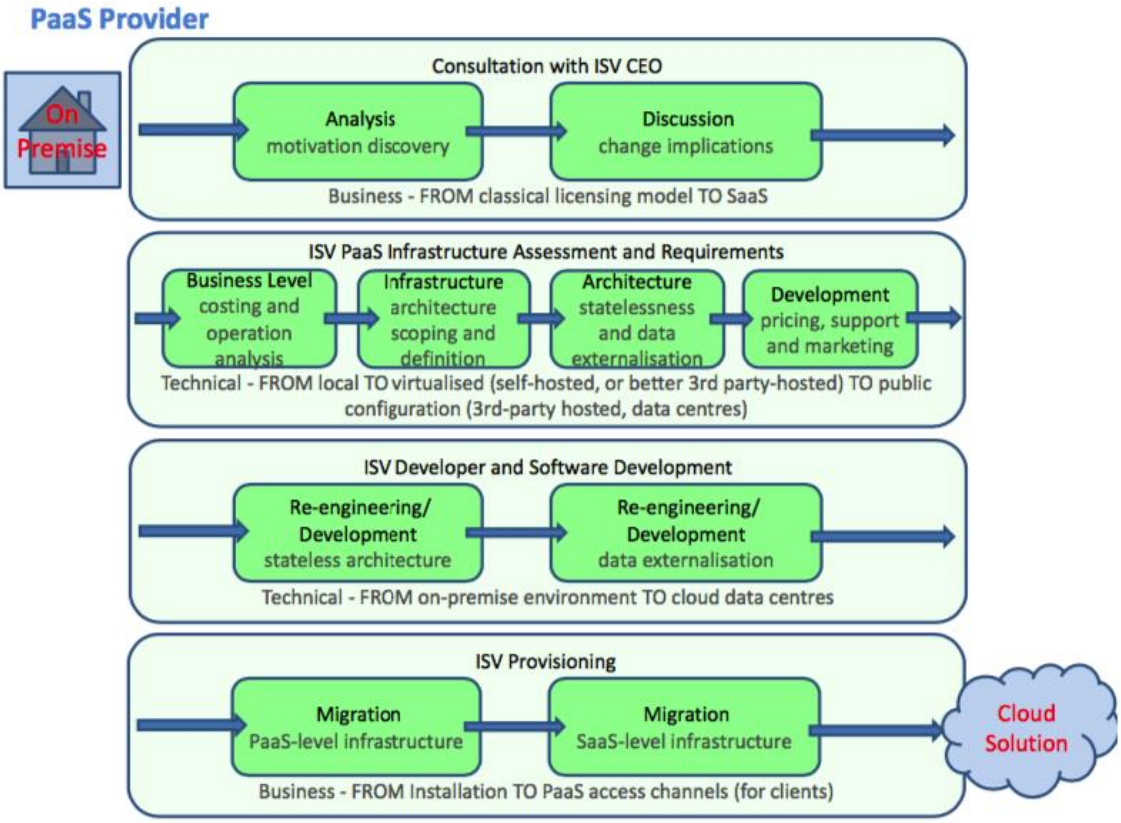

Figure 23 - PaaS migration process perspective (Pahl & Xiong, 2013)

3. METHODOLOGY

The creation of new knowledge in a conception process evolves the analysis of innovative artifacts with reflection and abstraction based on a well-defined process. In a scientific and academic production, is fundamental have a methodology to support this process creation (Hevner & Chatterjee, 2010).

The Design Science Research (DSR) is one of the methodologies that allows producing knowledge in a structured way, focusing in a real problem resolution, ensuring that the results obtained are scientists.

3.1. DESIGN SCIENCE RESEARCH PROCESS MODEL

The DSR methodology has some process model variations, but basically the goal of using DSR is producing a scientific knowledge contribution follow some process model steps, resulting in a proposal validated for some specialist group in the subject area. The Figure 24 represent this research process model with the inputs, process steps and desired outputs.

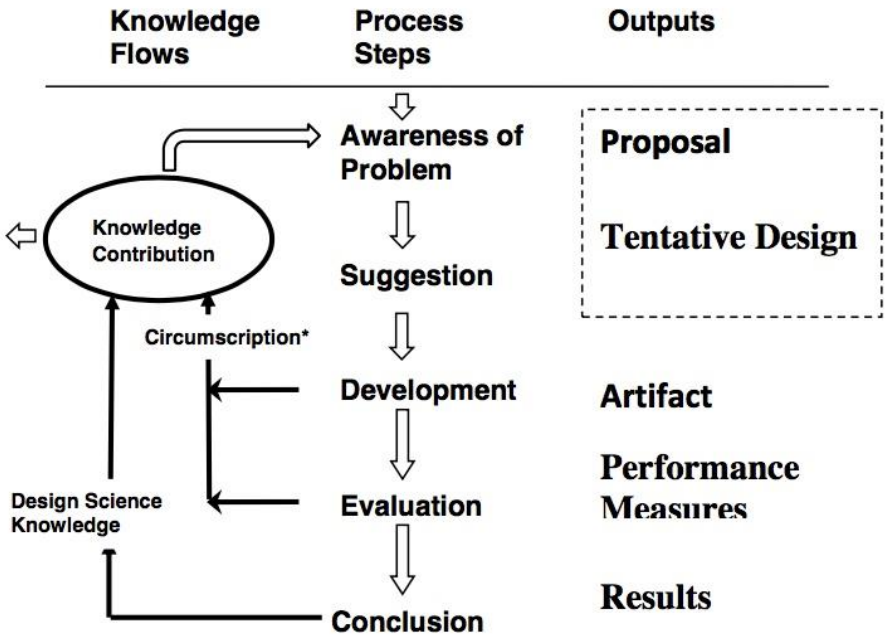

Figure 24 - Design Science Research Process Model (Hevner & Chatterjee, 2010)

The main process steps of DSR process model referenced in Figure 24 are described as follows:

Awareness of Problem: The process starts with an awareness of a research problem, where the goal is the identification and understanding of the relevant problem.

Suggestion: This step is essentially a creative step wherein creativity, imagination and previous knowledge are used to make inferences and discard hypotheses.

Development: In this step are development the artifacts, based on the previous step, with the goal to solve the problem identified.

Evaluation: The artifacts previously produced are evaluated based on an interview with specialists in the related subject area.

Conclusion: At the end of process, the obtained results are collected and communicated, allow your discussion. If the results are incomplete or insufficient compared to the awareness of problem, a new cycle can be stared in the DSR process model.

3.2. DEFINITION AND STRATEGY

According the DSR process model previously described, and which is used as a basis to guide this investigation work, the results based on the application of process model steps are described in sequence.

The Figure 25 describe the DSR process model applied in this work.

Figure 25 - DSR results applied in this work

From the “Awareness of Problem”, resulted the motivation of this work to understand the current challenges of software development on PaaS and try to answer the main question identified as main objective of this work.

In the second phase, “Suggestion, emerge the need to review the literature to understand the PaaS model and the main software development requirements.

In the third phase, “Development”, recommendations will be proposed for to achieve the objectives of this work.

The “Evaluation” phase aims to validate the proposed recommendation in interview sessions with invited specialists in software development and cloud computing, from organizations such as NOVA IMS, Accenture and Microsoft.

In the end, in the phase “Conclusion” the recommendations will be validated and reviewed based on the interview, and the obtained results will be documented and communicated. If in this phase, the awareness of problem was incomplete or not enough, a new DSR cycle can be started.

4. MODEL PROPOSAL

4.1. ASSUMPTIONS

The proposal and migration model are based on the concepts and fundamentals identified in the literature review study, which are related with the following assumptions:

1. There are different PaaS service models specialized to attend the different needs of the software development and migration to cloud;
2. Although there are several types of software, systems and applications, the migration model was focused on the main types of information systems that have a business content relevance;
3. There are different software architectures to use in software development, and each one has specific requirements and characteristics that should be evaluated according needs and requirements of the system;
4. Despite the software development for PaaS use the traditional Software Development Life Cycle (SDLC), is fundamental consider the cloud computing characteristics to adapt the software development to PaaS;
5. Migrate software development to PaaS need to have a defined roadmap and process to allow the migration in phases, although the migration process may be the same for all software development migration to cloud.

4.2. MIGRATION MODEL

The migration model for develop and migrate the software development to the PaaS cloud model, described in Table 3, was constructed based on two dimensions: the software architecture style and the type of information system that the software was built or will be build using one or more PaaS specialized categories.

The software architectures used in the model was selected based on the relevance and importance in the software engineering and in the current technologies used in software development. Also, the types of information systems were selected based on focus on business systems and in the most of legacy software implemented. For a specific software architecture and type of information system relation, was recommended one or more PaaS specialized models based on the literature review, and should be evaluated whether it will be necessary to use a PaaS specialized model or more than one combined to use PaaS in software development migration projects.

Software Architectures	Types of Information Systems					
	Transaction Processing Systems (TPS)	***Business Intelligence (BI)***	***Enterprise Applications***			
			ERP	***SCM***	***CRM***	***KMS***
Monolithic	(1) Application Platform Services (**aPaaS**) *Integration Platform Services (**iPaaS**)*	(2) Application Platform Services (**aPaaS**) Business Analytics Platform Services (**baPaaS**) *Integration Platform Services (**iPaaS**)*	(1) Application Platform Services (**aPaaS**) *Integration Platform Services (**iPaaS**)*	(1) Application Platform Services (**aPaaS**) *Integration Platform Services (**iPaaS**)*	(1) Application Platform Services (**aPaaS**) *Integration Platform Services (**iPaaS**)*	(1) Application Platform Services (**aPaaS**) *Integration Platform Services (**iPaaS**)*
Client-Server	(3) Database Platform Services (**dbPaaS**) *Integration Platform Services (**iPaaS**)*	(3) Database Platform Services (**dbPaaS**) *Integration Platform Services (**iPaaS**)*	(3) Database Platform Services (**dbPaaS**) *Integration Platform Services (**iPaaS**)*	(3) Database Platform Services (**dbPaaS**) *Integration Platform Services (**iPaaS**)*	(3) Database Platform Services (**dbPaaS**) *Integration Platform Services (**iPaaS**)*	(3) Database Platform Services (**dbPaaS**) *Integration Platform Services (**iPaaS**)*
N-Tier	(4) Application Platform Services (**aPaaS**) Database Platform Services (**dbPaaS**) *Integration Platform Services (**iPaaS**)*	(4) Application Platform Services (**aPaaS**) Database Platform Services (**dbPaaS**) *Integration Platform Services (**iPaaS**)*	(5) Application Platform Services (**aPaaS**) Business Process Management Services (**bpmPaaS**) Database Platform Services (**dbPaaS**) *Integration Platform Services (**iPaaS**)*	(6) Application Platform Services (**aPaaS**) Database Platform Services (**dbPaaS**) Enterprise Horizontal Portal Services (**Portal PaaS**) *Integration Platform Services (**iPaaS**)*	(7) Application Platform Services (**aPaaS**) Communications Platform Services (**cPaaS**) Database Platform Services (**dbPaaS**) Enterprise Horizontal Portal Services (**Portal PaaS**) *Integration Platform Services (**iPaaS**)*	(6) Application Platform Services (**aPaaS**) Database Platform Services (**dbPaaS**) Enterprise Horizontal Portal Services (**Portal PaaS**) *Integration Platform Services (**iPaaS**)*
Microservices	(8) Application Platform Services (**aPaaS**) Database Platform Services (**dbPaaS**) Function Platform Services (**fPaaS**) *Integration Platform Services (**iPaaS**)* *Message-Oriented Middleware Services (**momPaaS**)*	(9) Application Platform Services (**aPaaS**) Business Analytics Platform Services (**baPaaS**) Database Platform Services (**dbPaaS**) *Integration Platform Services (**iPaaS**)* *Message-Oriented Middleware Services (**momPaaS**)*	(10) Application Platform Services (**aPaaS**) Business Process Management Services (**bpmPaaS**) Business Rule Platform Services (**brPaaS**) Database Platform Services (**dbPaaS**) *Integration Platform Services (**iPaaS**)* *Message-Oriented Middleware Services (**momPaaS**)*	(11) Application Platform Services (**aPaaS**) Business Rule Platform Services (**brPaaS**) Enterprise Horizontal Portal Services (**Portal PaaS**) Database Platform Services (**dbPaaS**) *Integration Platform Services (**iPaaS**)* *Message-Oriented Middleware Services (**momPaaS**)*	(12) Application Platform Services (**aPaaS**) Business Rule Platform Services (**brPaaS**) Communications Platform Services (**cPaaS**) Enterprise Horizontal Portal Services (**Portal PaaS**) Database Platform Services (**dbPaaS**) *Integration Platform Services (**iPaaS**)* *Message-Oriented Middleware Services (**momPaaS**)*	(13) Application Platform Services (**aPaaS**) Enterprise Horizontal Portal Services (**Portal PaaS**) Database Platform Services (**dbPaaS**) *Integration Platform Services (**iPaaS**)* *Message-Oriented Middleware Services (**momPaaS**)*

Table 3 – Recommendations for software development and migration to PaaS

1. **Monolithic architecture and TPS, ERP, SCM, CRM and KMS information systems:** as a monolithic application is built in a single unit, these types of information systems can use the Application Platform Services (**aPaaS**) model to implement and deploy the system in the cloud, using an embedded database and one or more programing language runtimes to implement the application (UI, business logic and data access layer).

2. **Monolithic architecture and BI information system:** To implement BI information systems in cloud as a monolithic application, the Application Platform Services (**aPaaS**) can be used to build an application with BI capabilities from scratch or the Business Analytics Platform Services (**baPaaS**) model can be used to prepare data for analysis, visualize and analyze data, develop and publish dashboards or other Business Intelligence (BI) outputs.

3. **Client-server architecture and TPS, BI, ERP, SCM, CRM and KMS information systems:** Database Platform Services (**dbPaaS**) can be used to host the server logic and data for client-server applications and the respective information systems types, but as the application logic is implemented with database stored procedures and database triggers, the dbPaaS vendors and products offers need to be analyzed and evaluated, before any implementation or migration, to attend these requirements.

4. **N-tier architecture and TPS/BI information systems:** The Application Platform Services (**aPaaS**) and Database Platform Services (**dbPaaS**) can be used to implement a n-tier application for TPS and BI information systems, where the data layer can be hosted in dbPaaS and the presentation and business logic layer (and others layers if needed) can be implemented and deployed in an aPaaS as separated applications/modules.

5. **N-tier architecture and ERP information system:** Like the recommendation 4, the Application Platform Services (**aPaaS**) and Database Platform Services (**dbPaaS**) can be used to implement an ERP information system. Additionally, a Business Process Management Services (**bpmPaaS**) can be used to modeling and execute the business process related with the functional areas in ERP, such as manufacturing and production, finance and accounting, sales and marketing and human resources.

6. **N-tier architecture and SCM/KMS information systems:** Like the recommendation 4, the Application Platform Services (**aPaaS**) and Database Platform Services (**dbPaaS**) can be used to implements an SCM and KMS information systems. Also, an Enterprise Horizontal Portal Services (**Portal PaaS**) can be used to provide a corporate portal with the enterprise knowledge base for KMS systems and a Business-to-Business (B2B) portal for the interaction between the company and the suppliers, purchasing firms, distributors and logistics companies.

7. **N-tier architecture and CRM information system:** Like the recommendation 6, the Application Platform Services (**aPaaS**), Database Platform Services (**dbPaaS**) and Enterprise Horizontal Portal Services (**Portal PaaS**) can be used to implements a CRM information system. Besides, a Communications Platform Services (**cPaaS**) can be used to provide communications functionalities to enhance the communication and interaction with the customers.

8. **Microservices architecture and TPS information system:** In case of TPS information system, it can be implemented in microservices architecture using: Application Platform Services (**aPaaS**) to implement each microservice application, a Database Platform Services (**dbPaaS**) to be used by each microservice and encapsulate the business data domain, and a Function Platform Services (**fPaaS**) to implements small functions and procedures to be executed as a microservice by the system.

9. **Microservices architecture and BI information system:** Application Platform Services (**aPaaS**) and Database Platform Services (**dbPaaS**) can be used to implement each microservice of the BI information systems and a Business Analytics Platform Services (**baPaaS**) can be used together with the microservices to provide the analytical reports and dashboards.

10. **Microservices architecture and ERP information system:** Application Platform Services (**aPaaS**) and Database Platform Services (**dbPaaS**) can be used to implement each microservice of the ERP information system, and additionally can use a Business Process Management Services (**bpmPaaS**) for modelling and executing the business process and a Business Rule Platform Services (**brPaaS**) to encapsulate some business rules to be executed by the microservices in the system.

11. **Microservices architecture and SCM information system:** Application Platform Services (**aPaaS**) and Database Platform Services (**dbPaaS**) can be used to implement each microservice of the SCM information system and a Business Rule Platform Services (**brPaaS**) to encapsulate the business rules that can be consumed by the microservices. Additionally, an Enterprise Horizontal Portal Services (**Portal PaaS**) can be used to provide a B2B portal that is integrated with the microservices layer.

12. **Microservices architecture and CRM information system:** Like the recommendation 11, a CRM information system can be implemented in microservices architecture using Application Platform Services (**aPaaS**), Database Platform Services (**dbPaaS**), Business Rule Platform Services (**brPaaS**) and Enterprise Horizontal Portal Services (**Portal PaaS**). Also, a Communications Platform Services (**cPaaS**) can be used by the microservices to provide communications functionalities to the customers.

13. **Microservices architecture and KMS information system:** Information systems type of KMS can be implemented in a microservice architecture using Application Platform Services (**aPaaS**) and Database Platform Services (**dbPaaS**) to build each microservice of the system, and an Enterprise Horizontal Portal Services (**Portal PaaS**) can be used to provide a knowledge base portal that is integrated with the microservices layer.

For all recommended scenarios in the migration model (Table 3), the Integration Platform Services (**iPaaS**) was recommended for situations when the integration and exchange of information between the cloud applications and on-premise and legacy applications is required.

In the recommendations related with Microservices Architecture (8, 9, 10, 11, 12 and 13), a Message-Oriented Middleware Services (**momPaaS**) was proposed to support the communication and integration between the different microservices implemented in the respective system, thus

supporting the message exchange with some protocols like AMPQ, MQTT, DDS and XMPP, JMS, WebSocket and APIs Rest.

4.3. VALIDATION

The objective of the interviews is to validate the migration model proposed for software development in PaaS, based on what was investigated during the literature review and assumptions, with specialists in Cloud Computing and Software Development from academic institutions and IT companies. The specialists invited for the interview are:

- **Henrique José de Jesus Carreiro**, Director of IT Insight and invited lecturer of NOVA Information Management School (NOVA IMS), with post-graduation by NOVA and INSEAD, UC and FD UL, is a lecturer and consultant on Big Data, Cloud Computing, IoT and Blockchain. Experienced manager of significant and innovative technology businesses. Led the launch of Cloud Services at Microsoft and the first commercial Internet Services in Portugal, at Telepac.
- **Ricardo Soares**, Global Solution Architect from SAP Cloud Platform & SAP Leonardo at Accenture Portugal, with over 17 years of experience in software development, consulting and architecture with Microsoft and SAP technologies in several projects related to Retail, Consumer Goods, Travel Management, Internet, Communications and Mobility areas.
- **Nelson Luciano**, Data & AI Solution Specialist at Microsoft Portugal, with over 13 years of experience in architecture and technology strategy with Microsoft technologies in projects related with Financial Services and Healthcare, Public Sector, Telecommunications and Financial sectors.

The following 3 questions and the identification of each interviewee was considered fundamentals in the interview step and aims to validate the study, validate and improve the model and the work.

1. **Does it make sense to propose a model that systematize the different specializations of PaaS for software development?**

2. **Do you agree with the strategy followed in the presented migration model (Software Architecture/Information System Type/Recommendations)?**

3. **Do you have suggestions to improve the migration model proposed?**

4.4. DISCUSSION

Based on the interview answers collected from the specialists, presented in Annex I - Interviews, it is possible to verify that the migration model proposed in this work is valid and the recommendations presented in the model can be applied in software development scenarios with PaaS, according to what has been proposed in the migration model.

Regarding the question 1, which aimed to validate if it makes sense to propose the model presented, all of them agreed that the model proposed makes sense, and was highlighted the importance of have precise definitions of the PaaS specialized models, thus avoiding overlaps of concepts. Also, be careful about the definition of microservices architecture, because it is currently constantly changing. Likewise pointed out that this proposed model is just one more of several solutions based on model. Lastly, it

was mentioned that this model has great potential to grow and contemplate new types of information system and software architectures.

The strategy followed in the model was agreed by all interviewees, according the answers from question 2, and it was suggested that some examples of the application of recommendations to enrich the work should be described. It was pointed out that moving to the cloud could be to migrate existing applications or extend applications by implementing new features in the cloud, and that it is important to differentiate between developing from scratch or migrating to the cloud, as there are specific requirements to be considered in each situation. It was also suggested to consider the operations of the cloud applications in the model as a characteristic and also the analysis of information and metrics that can be obtained from PaaS to optimize the cloud applications for the cloud computing model.

In question 3 answers, the interviewees collaborated with interesting suggestions for improvements in the model, but there was no agreement on how to evolve the model, since one interviewee suggested keeping it as simple as possible and not include new types of information systems and new software architectures, while another interviewee suggested exactly the opposite, evolve the model to consider new types of information systems and new software architectures. In addiction was suggested to consider UI and UX aspects mainly in the microservices architecture recommendations, and Analytics as a category to consider aspects such as predictive models and machine learning in the analysis and recommendations of the model. Other suggestions were to consider the problem of vendor lock-in in the software development of applications in PaaS model and how to avoid it, and also expand the scope of the analysis in the model from just the technology architecture point of view to an enterprise architect point of view to consider strategical decisions that could produce a roadmap to support the decision of migrate or not the applications to the cloud.

Thus, analyzing the results of the validation of the proposed model by the specialists through interviews, it was clear that it can add value to the software development process for PaaS and thus support decision makers, managers and technical specialists in trying to choose the most appropriate PaaS models for the needs and requirements of their projects.

5. CONCLUSIONS

Every day, agility and innovation has become key factors for organizations to continue to grow and be competitive in the local and globalized market. Cloud computing comes increasingly playing a key role to support the IT departments and ISV to drive organizations to achieve these goals, supporting new software and hardware technologies, and new software development methodologies and software architectures.

More and more users are deploying strategical business applications in IaaS, PaaS and SaaS, thus making platform capabilities the center of the cloud innovation. PaaS has been evolving and specializing in specific platforms to attend the different software development needs and the IT market trends, like Big Data, Internet of Things, Mobility, Cloud Native and others.

However, with the continuous PaaS specialization and the new software development technologies that comes up every day, migrate legacy systems or development new systems in Cloud becomes complex and challenging in some situations, and was for these scenarios that the investigation produced in this document tried to give recommendations and solutions, with a migration model built based on the study of the concepts and technologies related to PaaS, to help IT departments and ISV to start entering in the PaaS software development with an initial blueprint.

5.1. SYNTHESIS OF THE DEVELOPED WORK

The main objective of this investigation work was to propose a model with recommendations, based on the requirements and technologies to develop and migrate software to PaaS model studied in literature review. The study was divided in 7 main phases:

- Study of Cloud Computing concepts;
- Study of Information System types and concepts;
- Study of Software Architectures and software development in cloud;
- Study of PaaS concepts, software development in this model and its particularities;
- Study of migration of application to PaaS;
- Building a migration model with recommendations to use PaaS in software development and migration to cloud;
- Migration model validation through interviews with specialist in cloud computing in Portugal.

Based on the study, was possible conclude that the cloud computing is taking an important role in IT and in software development, enabling to implement and to run applications with new architectures and concepts in an agility and competitive way, with lower costs and high availability and scalability. Was based on these premises that were discussed PaaS as an alternative for software development in cloud. Starting from basic concepts of Cloud Computing, the study was evolving to understand the information systems, software architectures, software development process and PaaS specific concepts to realize as the software development could be migrated to cloud.

It was noticeable that there is a great complexity involved when it comes to software development in the cloud, given the diversity of software architectures, information system types and PaaS

specializations required to implement an application, and a reference model can really help companies to start to develop software to cloud.

It was also clear that the software development process and life cycle has no significant changes when applied to the cloud development and PaaS model, then this topic was not considered relevant to be part in the construction of the migration model, thus not influencing in the recommendations for PaaS models and scenarios.

From the study, it was possible to define the main goals and objectives in the dissertation and thus elaborate an interview with questions focused in help to validate the model proposed and perceive the vision that the specialists, from IT companies and universities involved with cloud computing, have on the PaaS and software development subjects.

As a result of the research, the main conclusions obtained are firstly, PaaS is been used in the market and IT to help to companies to develop your applications and solutions, and that there are different specializations of PaaS available to support the software development in cloud. Secondly, cloud computing is becoming a key factor for innovation and agility in organizations. Finally, this study helps to bring new knowledge to the author related to PaaS and software development in cloud.

5.2. LIMITATIONS

Although Cloud Computing is not a recent technology, Platform as a Service has been gaining attention in the IT market in the last years, and consequently the platform has been evolving, also there are still few scientific and academic documents focused in this particular area. Thus, an exhaustive investigation was necessary in different sources of information, like research and advisory companies in IT, academic papers and thesis, books related with Cloud Computing and software development, producing this document with more impartiality and objectivity and contributing to the scientific community.

Gather experts in Cloud Computing area and getting an agenda for the interviews was a difficulty too, so the focus was on selecting at least one specialist from the university, one from a software consulting company and one from a software technology company, thus allowing to validate the migration model with different perspectives and experiences.

5.3. RECOMMENDATIONS FOR FUTURE WORKS

Cloud computing is constantly evolving, and despite IaaS and SaaS have been explored for some years and can be considered mature, more and more companies are evaluating and considering the PaaS model for implement your business applications and solutions in the cloud. So, it is very important that we know how to use and to take advantage of this cloud model in the software development are, choosing the right PaaS specialized model for the desired scenario with its several variables and variants, like as the type of application, type of architecture, different integrations, programming languages and tools.

Currently, we can notice that cloud computing has been revolutionizing the IT market and the way we build applications and solutions, and if we analyze the conclusions of this study, it is evident that there is a need to study and explore more the PaaS cloud model and its specializations, also the changes and adaptations required in the software development arising from this “new way” to build software.

So, as for future recommendations for academic literatures, the following topics can serve as a basis for evolving this study in others master and PhD thesis:

- Considerer others types of software architecture in the migration model proposed;
- Extend the scope of information system types in the migration model to consider others areas of software development, such as scientific applications, software utilities, software for education, entertainment and games;
- Describe reference scenarios of software development and migration for using the proposed recommendations;
- Understand the level of use of the recommendations proposed in this study in software development projects that already used PaaS by IT departments and ISV;
- Study new cloud computing models that aims the software development, like Container-based and Serverless Computing;

BIBLIOGRAPHY

Beimborn, D., Miletzki, T., & Wenzel, S. (2011). Platform as a Service (PaaS). *Business and Information Systems Engineering*, *3*(6), 381–384. https://doi.org/10.1007/s12599-011-0183-3

Bond, J. (2015). *The Enterprise Cloud*. (Brian Anderson, Ed.), *Technology* (First Edit). O'Reilly Media. Retrieved from http://shop.oreilly.com/product/0636920034124.do

Burns, B. (2018). *Designing Distributed Systems - Patterns and Paradigms for Scalable, Reliable Services*. *O'Reilly*. https://doi.org/10.1093/ecco-jcc/jjy097

Cloud Standards Customer Council. (2015). Practical Guide to Platform-as-a-Service Version 1.0, 32. Retrieved from http://www.cloud-council.org/deliverables/CSCC-Practical-Guide-to-PaaS.pdf

Cohen, B. (2013). PaaS: New opportunities for cloud application development. *Computer*, *46*(9), 97–100. https://doi.org/10.1109/MC.2013.323

Council, C. S. C. (2013). Migrating Applications to Public Cloud Services: Roadmap for Success. Retrieved from http://www.cloud-council.org/deliverables/CSCC-Migrating-Applications-to-Public-Cloud-Services-Roadmap-for-Success.pdf

Fehling, C., Leymann, F., Retter, R., Schupeck, W., & Arbitter, P. (2014). Cloud Computing Fundamentals. In *Cloud Computing Patterns* (pp. 21–78). https://doi.org/10.1007/978-3-7091-1568-8_2

Fowley, F., Elango, D. M., Magar, H., & Pahl, C. (2017). Software System Migration to Cloud-Native Architectures for SME-Sized Software Vendors (pp. 498–509). https://doi.org/10.1007/978-3-319-51963-0_39

Hevner, A., & Chatterjee, S. (2010). *Design Research in Information Systems* (Vol. 22). Springer, Boston, MA. https://doi.org/10.1007/978-1-4419-5653-8

Ingeno, J. (2018). *Software Architect's Handbook Become a Successful Software Architect by Implementing Effective Architecture Concepts.* Packt Publishing Ltd.

ISO/IEC/IEEE. (2011). ISO/IEC/IEEE 42010:2011, Systems and software engineering - Architecture description. ISO. Retrieved from https://ieeexplore.ieee.org/document/6129467

ISO/IEC. (2014). ISO/IEC 17788 -Y. 3500 SERIES Y- Information technology - Cloud computing - Overview and vocabulary. *ITU SERIES Y*.

Jalote, P. (2005). *An Integrated Approach to Software Engineering*. New York: Springer-Verlag. https://doi.org/10.1007/0-387-28132-0

Joshi, M., & Pandey, D. D. (2017). Comparative study of Traditional Software Development and Development on Cloud. *International Journal of Advanced Research in Computer Science (IJARCS)*, *8*(5). https://doi.org/http://dx.doi.org/10.26483/ijarcs.v8i5.3463

Kavis, M. J. (2014). *Architecting the Cloud: Design Decisions for Cloud Computing Service Models (SaaS, PaaS, and IaaS)*. Wiley; 1 edition (January 28, 2014).

KPMG. (2016). Journey to the cloud. Retrieved from https://assets.kpmg.com/content/dam/kpmg/xx/pdf/2017/02/the-creative-cios-agenda-journey-to-cloud.PDF

Laudon, K. C., & Laudon, J. P. (2018). *Management information systems: Managing The Digital Firm* (15th Editi). Pearson Education Limited. https://doi.org/10.1007/978-94-017-9618-7_44

Lewis, J., & Fowler, M. (2014). Microservices. Retrieved October 19, 2018, from https://martinfowler.com/articles/microservices.html

Malav, B. (2016). Microservices vs Monolithic architecture. Retrieved October 19, 2018, from https://medium.com/startlovingyourself/microservices-vs-monolithic-architecture-c8df91f16bb4

Mell, P., & Grance, T. (2011). The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology. *Nist Special Publication*, *145*, 7. https://doi.org/10.1136/emj.2010.096966

Microsoft. (2017). Cloud Application Architecture Guide. Microsoft Press. Retrieved from https://azure.microsoft.com/en-us/campaigns/cloud-application-architecture-guide/

Mogull, R., Arlen, J., Gilbert, F., Lane, A., Mortman, D., Peterson, G., & Rothman, M. (2017). The Security Guidance for Critical Areas of Focus in Cloud Computing v4.0 ("Guidance v4.0"). *Cloud Security Alliance*, *4.0*, 152.

Natis, Y. V., Baker, V. L., Dunie, R., Elliot, B., Feinberg, D., Guttridge, K., … Vincent, P. (2016). Platform as a Service: Definition, Taxonomy and Vendor Landscape, 2016. Retrieved February 2, 2018, from https://www.gartner.com/doc/3334517/platform-service-definition-taxonomy-vendor

O'Brien, J., & Marakas, G. (2010). *Management Information Systems* (10th Editi). McGraw-Hill Education.

PaaSfinder. (2018). Platform as a Service | Language Statistics. Retrieved February 2, 2018, from https://paasfinder.org/statistics/languages

Pahl, C., & Xiong, H. (2013). Migration to PaaS clouds - Migration process and architectural concerns. In *2013 IEEE 7th International Symposium on the Maintenance and Evolution of Service-Oriented and Cloud-Based Systems* (pp. 86–91). IEEE. https://doi.org/10.1109/MESOCA.2013.6632740

Posta, C. (2016). *Microservices for Java Developers*. (N. Barber & S. Conant, Eds.) (First Edit). O'Reilly Media, Inc.

Pressman, R. S. (2009). *Software Engineering A Practitioner's Approach 7th Ed - Roger S. Pressman*. *Software Engineering A Practitioner's Approach 7th Ed - Roger S. Pressman*. https://doi.org/10.1017/CBO9781107415324.004

Ruparelia, N. B. (2016). *Cloud Computing* (First Edit). MIT Press. https://doi.org/https://doi.org/10.7551/mitpress/9780262529099.001.0001

Violino, B. (2017). What is PaaS? Software development in the cloud explained. Retrieved February 1, 2018, from https://www.infoworld.com/article/3223434/paas/what-is-paas-software-development-in-the-cloud.html

Wang, H., He, W., & Wang, F. K. (2012). Enterprise cloud service architectures. *Information Technology and Management*, *13*(4), 445–454. https://doi.org/10.1007/s10799-012-0139-4

Zaigham, M., & Saqib, S. (2013). *Software Engineering Frameworks for the Cloud Computing Paradigm*. *Computer Communications and Networks*. Springer, London.

https://doi.org/10.1007/978-1-4471-5031-2

ANNEX I - INTERVIEWS

Interview 1:

Name: Henrique José de Jesus Carreiro

Company/Institution: NOVA Information Management School (NOVA IMS)

Role: Invited Lecturer

Experience in Software Development and/or Cloud Computing area: Henrique was responsible for the Development Platforms and Enterprise Solutions in Microsoft Portugal. Currently teaches curricular units related to Big Data, Enterprise Cloud and Mobility in NOVA IMS.

1. **Does it make sense to propose a model that systematize the different specializations of PaaS for software development?**
   "I think it makes sense the characterization and to characterize in a matrix format or in other multi-dimensional format, not in this case, and we could face it in a different way. For instance, when we are talking about Business Intelligence (BI), in your model you have just one layer, but we know that BI can have many dimensions/subdivisions/extensions, so we can have this model that is like a slice, thinking in the multi-dimension models, or we can have additional dimensions. So, yes, it makes sense have a characterization, but a characterization like that requires precision in the definitions, that is, it will have to be very well defined what we mean as aPaaS, iPaaS, dbPaaS, for example, and as the number of iterations is limited (5, 6 or 7), yet they will have to be defined for have any meaning, the definitions have to be complementary and mutually exclusive. That is, what is characterized in aPaaS did not overlap much with the other characteristics, because otherwise we are creating artificial divisions, so the definitions must be precise and based on literature. The definitions of information systems are based on systems already established and know and in terms of software architecture also. In case of Microservices and Serverless needs extra rigor and a strict definition, because it is an area that is changing where we have Lambda and K architectures with different characteristics. Of course, this will have to be useful for a company that is thinking about doing this transition, if we have a monolithic implementation, like a legacy CRM created 15 years ago with a monolithic architecture, how can we proceed to move to the cloud, of course, within the perspective of PaaS, because we are talking about migration and replatforming, but there are others alternatives to do it. It is important to highlight that this model is one solution of many others, because this is not the only solution and we can have other matrix (model). For example, you can take the monolithic application and put to run on IaaS, using a container or virtual machine."

2. **Do you agree with the strategy followed in the presented migration model (Software Architecture/Information System Type/Recommendations)?**
   "I agree as long as the scope is well delimited, because here (in the model) there is a subset of the types of information systems, a rather large subset of software architectures and the possible migration strategies. I think it is particularly useful if 2 or 3 application examples are presented and so enriches the work. Of course, it depends on the time and availability, but eventually you can tell

how this applies, what the process, the issues, etc. would be like. I think so, just to illustrate the matrix, because the matrix itself is useful but does not fully illustrate, and I think it would enrich if there were an illustration of one or two sample cases."

3. **Do you have suggestions to improve the migration model proposed?**
   "I'd rather not add more types of information system and software architectures, as I'm afraid of the model might be too complex, that is, I would not increase the complexity of the model, because when we are talking about microservices and CRM, we already have a lot of information. The only thing I would say the only thing I would say in terms of improvement was not to add more and see if any of these categories could possibly be compacted. I would apply some method to simplify and not complicate more the model."

Interview 2:

Name: Ricardo Soares

Company/Institution: Accenture Portugal

Role: Senior Manager/Solution Architect

Experience in Software Development and/or Cloud Computing area: Ricardo had experience as functional and technical consultant in software development projects with .NET, SAP, Mobile, SOA and BW technologies. For the last years has been working with Cloud Platform (PaaS, Cloud Foundry, Cloud Native, Microservices) technologies and solutions from SAP in software development projects.

1. **Does it make sense to propose a model that systematize the different specializations of PaaS for software development?**
   "Yes, it makes sense to separate the layers, the various parts in a software architecture and group them, so it makes sense to treat them, even because we have some providers in the market that are stronger in some services than others, so it makes sense to do this. Having this subcategorization of PaaS, with nomenclatures like database, integration, clearly makes sense. In terms of workflow or processes it also makes sense."

2. **Do you agree with the strategy followed in the presented migration model (Software Architecture/Information System Type/Recommendations)?**
   "Yes, I agree with this strategy. When we are talking about development, may be developing a new solution, and I think this matrix can help and assist in how to structure the types of service and which guideline and recommendation can be followed. But when we talk about a migration, or rather an extension, this model is not enough. For example, we might want to do a migration or an extension, which is different, because we can do an extension in cloud of an ERP system that already exists that is monolithic, and we want to extend the solution to have more functionality now, and this extension can be an integration as this can happen."

3. **Do you have suggestions to improve the migration model proposed?**

"About improvements, it makes perfect sense when we are talking about microservices segregating the user experience from everything else. What I would do would be to introduce a new category that is from UI (User Interface), where UI does not mean that it is just web pages, since it can be mobile or it can be a notification on the mobile phone, that is a form of interaction, and we can also call it UX (User Experience). Because the trend is that you develop a solution and you will decide according to the architecture that you want to follow, you will decide what types of services you want to adopt, which is what you have in the model, for example, for integration I would like to use Microsoft, because we already work with them and is the best in the market in our opinion; for machine learning or for database, we would like to use SAP, and so here we will have a mix, but when we are talking at the level of the service application this can also happen, as it goes to the client-server and n-tier also applies, and in the case of monolithic is not so much. It makes sense to separate UX from the core/logic/application, so you can segregate the aPaaS category to consider the UX in the recommendations. For UX, we are talking about mobile, single screen, and can even create several screens around the monolithic, though it does not make much sense. In addition, I would introduce another category which is the analytics, although it is related to database, in this case we are splitting the database in a data storage and analytics concepts. The analytics is more than a database, because it is understood that it is more in the sense of predictive models, machine learning, etc. For example, you can develop an ERP solution and may want to use an analytics solution. One thing is to create a BI solution, which is an application, another thing is to have an ERP/CRM/SCM and I want to introduce functionalities that are in the area of analytics, like machine learning, predictive models, real-time analysis, etc."

Interview 3:

Name: Nelson Luciano

Company/Institution: Microsoft Portugal

Role: Data & AI Solution Specialist

Experience in Software Development and/or Cloud Computing area: Nelson had experience as Consultant and Solution Architect in software development projects with Microsoft technologies focused on Data and Business Intelligence (BI) platform and currently is Data & Artificial Intelligence (AI) Solution Specialist in the pre-sales team.

1. **Does it make sense to propose a model that systematize the different specializations of PaaS for software development?**
   "Yes, it makes sense, and it's a great foundation that has the potential to grow and become even more comprehensive. It is clear that some types of software architecture and types of information systems have been chosen, but nowadays, in addition to the types of information systems that tend to grow, the truth is that there is a mix of several types of information systems in a single application. For example, SAP is a case of these, which mixes ERP with CRM, Microsoft also does the same with Navision and with CRM, so clearly this matrix tends to grow."

2. **Do you agree with the strategy followed in the presented migration model (Software Architecture/Information System Type/Recommendations)?**

"Yes, I agree and I think there is a topic here that may be interesting to uncouple, since it may be different from the development from scratch of an application and from migration, because migration typically brings conditions to the move to cloud, and it is interesting to have that, because is a type of analysis different when it comes to a migration, because it is not being built from scratch and when it is built from scratch already thinking about a cloud architecture, but when migrating is bringing the legacy (on premise) that often does not fit exactly with these services, and in case of migration, the iPaaS model is super relevant, because it is in the migration that we will have more cases of a hybrid architecture between cloud and on premise. Another topic that I find interesting to approach would be from an operational point of view, as it is not usually related to the software development lifecycle, but today with DevOps it is increasingly relevant, that is, to consider as a closed loop and not a timeline, and in cloud has long been this concept, which is how we in the software development lifecycle itself, we take advantage of the events that are generated by PaaS platforms, as an example of what happens in Formula 1 where the car is always providing telemetry data and the engineer is adjusting if the car is needed in real time. This is a super interesting subject when thinking about cloud, because in fact today, applications, even if they are built from scratch, have many events."

3. **Do you have suggestions to improve the migration model proposed?**

"I think this model tends to grow, to be added more types of information systems and eventually different software architectures, and it almost actually gives to disaggregate some of these types of information systems into smaller systems and begin to detail exactly what kind of services can be used in each case. I also find interesting to analyze, in the case here is represented from the point of view of technological architecture, a more strategic and corporate architecture vision, as we have in paper Enterprise Cloud Strategy (Microsoft), since one of the areas that focuses is as from the point of view of business architecture strategy of a company if it build a roadmap of evolution to the cloud and how does it evaluate if an application should or should not be moved to the cloud, for example, if it does replatform or rehost, it analyzes several factors like the information architecture of the application, which the roadmap of the application, because if it is discontinued in 1 year, does not make sense to move it to the cloud, the costs, whether or not advantage of cloud from the point of view of elasticity of this type of model. So, I think it could be another super interesting point of view to be explored. Also, another area to be explored is, as in cloud migration the company does not get vendor lock-in and as from the software development point of view it reaches that goal. Although this problem already exists in the on-premise model of software development, but cloud the customers are much more concerned about this, to be able to move applications between different cloud providers with the minimum impact."