Managerial & Technical Insight in Knowledge Management

George E. Vranopoulos[a*] *and Athanassios A. Triantafyllidis*[b]
[a]eProject,
[b]The American College of Greece
[*]Corresponding Author: VranopoulosGerogePublic@gmail.com

ABSTRACT

Corporate Knowledge is an asset but its management is no trivial task. In this publication the voyage towards a Corporate Knowledge Management system for a large organization (more than 15000 employees and 1000 location) will be presented. Managerial aspects along with technical issues will be detailed. The project management techniques used, technical decisions, corporate strategy along with Business and Technical challenges accompanied by their resolution and courses of action will provide the guidelines for the successful implementation of a Knowledge Management System.

Keywords: Knowledge Management, SOLR, Big Data, Variety, Corporate Knowledge

1. INTRODUCTION

One of the benefits of current technology is that it enables, especially corporations to accumulate large volumes of data which serve as corporate knowledge. However, accessing this knowledge is a challenge. The corporation must set objectives (conceptual framework), define the Knowledge Management methodology and provide employees with easy access in order for them to utilize this Knowledge pool (Mentzas, Apostolou, Young, & Abecker, 2001). Higher management commitment is essential for the successful implementation of knowledge management. (Stemberger, Manfreda, & Kovacic, 2011; Sumner, 1999).

In an organization with more than 15.000 employees accessing the right information at the right time is essential, but poses several challenges (Fischer, 2012). Some of the concepts that need to be considered include, identifying knowledge sources, classifying information, cleansing metadata and identifying a presentation / searching mechanism.

This case study presents the implementation of a "Knowledge Mechanism" in a large organization (> 15.000 employees). It will be reviewed under the prism of Managerial and Technical issues identified and resolved. With respect to the managerial issues, higher management's commitment, resistance to change and personnel involvement will be discussed. Additionally, infrastructural technical aspects and issues raised by Variety will be addressed.

Knowledge is accumulated fast with respect to volume, constantly changes and comes in different forms. These characteristics, direct to the Big Data ecosystem.

Big Data is primarily defined in terms of V's, based on the first definition by Laney (Laney, 2001):

- **Volume**: Refers to the amount of data being created and stored (Khan, Uddin, & Gupta, 2014) in the digital universe.
- **Velocity**: In Big Data environments the speed in which data change is quite high.
- **Variety**: This characteristic has to do with the data itself and the flavours it can pertain. Sensors, IoT (Internet of Things), database records, video and audio have different formats and standards, let alone the fact that in many cases alternative communication protocols must be used to disseminate the data streams.

Corporate sites, e-mails, corporate documents (policies, announcements, etc.), product definitions and brochures, corporate informational databases and Frequently Asked Questions come into perspective when Corporate Knowledge is investigated (Dieng, Corby, Giboin, & Ribière M, 1999). It is evident that these are stored in different formats and thus Variety, which constitutes the biggest challenge in Big Data environments (Baker, 2015), comes into play.

This paper presents, in terms of Knowledge Management, the current situation and efforts towards the implementation of an innovative portal that would disseminate and make the corporate knowledge pool effortlessly available and searchable the corporate knowledge pool.

2. IDENTIFY THE NEED

During executive meetings, top and higher level management expressed their concerns that they were losing valuable time trying to access existing knowledge in the intranet such as consultation on legal matters, etc. without success.

The CIO called an IT meeting and expressed his concerns about the organization's knowledge management, with respect to presentation and search engine usability. The task of identifying state of the art technologies to address the issue was assigned to the IT Research and Development department.

Information Technology is an essential, but not sufficient, part of the knowledge revolution (Rollo & Clarke, 2012). This is why the CIO also requested the involvement of several other departments (see Departmental Involvement).

A kick-off meeting was arranged and representatives from the departments attended in order to "set the bases" for the project. The project's mission statement was the development of an innovative information system to store, classify and present the existing knowledge within the organization. But once attempting to define further action, several questions were raised that could not be easily answered:

- What should be included in this hub?
- How will the information stored will be created and maintained?
- Who is going to be the owner?
- Should Waterfall, SCRUM or a "best-breed" project management methodology used?

Based on answers to the aforementioned questions, managerial and technical aspects of the project were to be investigated. In identifying and answering the questions, it was agreed that a cross-functional team would undergo an investigation and depict current practices and existing systems.

3. CURRENT SITUATION

Before implementing any project it is essential to understand the "Current Situation," that is, to investigate what is available and how it is utilized. Mapping the corporate knowledge sources is the first step in the knowledge organizations process, whilst the next steps involve evaluation and classification (Apostolou & Mentzas, 1999). In this chapter the existing corporate structures and the personnel handling knowledge along with the technological infrastructure employed will be presented.

3.1. Knowledge "Handling"

Knowledge was identified to be vested on different departments based on their expertise. The information and respective knowledge, was in departmental "silos" since interdepartmental exchange of information was limited. It was also recognized that each department utilised different means of storing and communicating knowledge.

In understanding the different approaches three departments with different knowledge management processes will be presented.

Legal Department

Requests are addressed to the head of the department and then assigned to respective lawyers by forwarding the request mail. Each lawyer in turn will reply to the initiator of the request with the appropriate proposal.

In this case knowledge is lost. In many cases rework is done since prior responses to requests cannot be accessed by co-layers. Even worse, since law is a science which involves interpretation, in certain cases the recommendations from different lawyers contradicted each other.

Information Technology Help Desk

Every request is handled through a Ticketing system. If the agent considered this was an "important issue," based on his experience and the propensity of reoccurrence, he would log the problem and its accepted solution into a WIKI developed and maintained by the department. This practice resulted in frequent knowledge loss since recording was vested upon agents. Another issue was that the WIKI and the Ticketing systems have no means of integration thus it is impossible to identify and correlate the original problem with the proposed solution.

Additionally, the information held in the WIKI was highly technical, since the intended audience was other IT support agents, making it unusable to common users.

Commercial Help Desk

Requests are sent via corporate e-mail to a shared mailbox which is viewed by the department's employees enabling requests to be processed on a first-come first-served basis. Furthermore, certain requests that were wrongfully addressed to the IT Help Desk were also reviewed via the respective ticketing system

All requests are recorded in a spreadsheet which serves two functions; a) Enable employees to know how to handle what in order to get the next request from the "stack;" b) Serve as a basis for further analysis with respect to reoccurrence.

Reoccurring requests are investigated upon and a Frequently-Asked-Question document is formulated or an existing one is amended. The document is then uploaded to the intranet in a tree structure. Bimonthly the FAQ knowledge database is evaluated in terms of user access and revised for outdated content.

In this case the process of knowledge management is more structured and knowledge is disseminated across the organization. Structure, classification, relevancy and use are monitored and maintained through a structured process.

3.2. Technological Infrastructure

With respect to Corporate Knowledge Management each department, based on its own procedures and employees IT skills, employed a different infrastructure in recording and handling knowledge. Systems wise, there was no single repository where knowledge was maintained attributing to the intensification of the Variety effect.

Another aspect that has to be addressed is the Disaster Recovery (DR) and backup methodology for the accumulated knowledge. Knowledge is recognized as an asset and probably the only one that grows with time (Gupta, Iyer, & Aronson, 2000). In view of this realization, there has to be a procedure that will outline the way that this information is backed up and a detail plan for DR and Business continuity. These procedures were unavailable in the current situation due to the number of different systems.

Following, an enumeration of repository infrastructures identified is presented:

- **Corporate e-Mailing System**, data are only located within the user or group mailboxes.
- **Folders**, data are stored on local hard disk folders or network shared drives.
- **Intranet**, FAQ documents and corporate communication are uploaded to the Intranet web site.
- **Proprietary systems**, information is uploaded and maintained in custom departmental Content Management Systems (CMS) or Business Process Management (BPM) special software.
- **Open-Source CMS**, like WordPress, Joomla and Drupal where also installed and used. None the less these systems were partially governed by the IT department since they were not considered mission critical systems.
- **Open-Source Wiki's** like DokuWiki, PmWiki, TWiki and MoinMoin were also utilised primarily by the IT departments for problem and bug recording, investigation and resolution.
- **Commercial Software**, like JIRA, Confluence and Coveo were also utilized by departments to record findings and the proper course of actions towards resolution.

The most important IT systems challenge was the partial involvement of Group IT in validating the respective solutions. Due to this "un-involvement" several systems were installed without adhering to the corporate technological roadmap and without the proper procedures for maintenance and backup.

3.3. Lessons Learned

From the investigation performed in the existing knowledge management process it was identified that in essence there are multiple and different approaches. This was an indication that a Corporate Knowledge Management strategy should be formulated

With respect to technology, almost every available infrastructure was utilised in storing knowledge. However in some cases knowledge was not recorded.

It was obvious that measures should be taken in standardising knowledge "handling" with the implementation of a single unified repository and procedures that would govern the process across the organization.

4. EMBARKING ON A TRIP TOWARDS KNOWLEDGE

People are the main enablers in the Knowledge Management ecosystem and Information Technology is the fundamental tool (Yeh, Lai, & Ho, 2006). Based on this assumption, in this chapter the "people asset" will be investigated and presented.

4.1. Management Commitment

The senior executive general manager in cooperation with the Group CIO formed a multi-disciplinary (see Departmental Involvement) committee to investigate, evaluate and implement a new approach to Knowledge. Regular meetings where scheduled biweekly to evaluate progress.

The active involvement of executive and higher management signified the project's importance and management's commitment for innovation, thus abolishing an integral IT project's risk regarding employees support (Baccarini, Salm, & Love, 2004). Smaller teams, working simultaneously but always under the loose supervision of the steering committee in compliance with SCRUM, were formed.

4.2. Departmental Involvement

An important factor, in the successful implementation of a Knowledge Management initiative, is the development of an organizational infrastructure which will facilitate the establishment of organizational groups whose members will have the necessary skills to create, distribute and prioritize the knowledge efforts (Davenport, Long, & Beers, 1998). In view of this required organizational reform and identification, the involved departments and their role will be presented.

IT Department

For the IT Department representatives from Infrastructures, Software Development and R&D teams were allocated to the implementation taskforce. Their roles, examined later on, are distinct but integrated and coordinated. Since R&D was the driving force in researching the appropriate technologies, its representative was also assigned with the task of coordination and overseeing the proper implementation and integration of the new system within the company's infrastructure.

The Infrastructures representative was responsible for the following tasks:

- **Infrastructure Sizing**. Based on the requirements posed by the selected software and the anticipated usage (users' concurrent access), the hardware in terms of computational requirements should be identified and installed. Since the company employed a virtualized servers' infrastructure, it was decided to install a single server with both the repository and the application server. In case of performance deterioration more resources (CPU's, RAM) would be allocated or the server would be cloned.
- **Software Development**. In close cooperation with R&D, the aspects of existing sources connectivity to the new infrastructure were investigated. The technologies provided by the new infrastructure, identified by the R&D, were cross-referenced with the organizations development tools and standards. A technological blueprint was drafted in outlining the interoperability guidelines along with the front-end implementation tools to be utilised.
- **Research & Development (R&D)**. R&D was responsible to investigate state of the art technologies and future trends in the respective context. Once the available tools and infrastructures were identified R&D was responsible to present the respective environments to all involved parties in identifying the most prominent alternatives. Upon identification of the two most "convenient" alternative ecosystems, R&D would prepare a Proof-of-Concept (POC) by implementing small scale environments that would demonstrate the functionality and test "fit" with the company's existing infrastructures. Finally the R&D was responsible to give direction and guidance to any technical problem identified along the way with respect to installation, connectivity and software development on the new environment.

Organization Department

The Organization department is responsible for all the content available in the company's intranet and for the formulation of all corporate procedures. Any corporate communication between the branches and the organization's back-offices has to be filtered and eventually published by the respective department.

In this project, the Organization department was responsible to evaluate the existing knowledge sources and indicate which should be included in the new initiative. Content was evaluated and proposals for the respective classification and structuring were formulated

The investigation and evaluation of the existing sources was a very challenging task since there were no standardized procedures in classifying content and the existing "keywords" were arbitrarily attributed to content (details available in 6.3 Variety Issues).

The Organization department was responsible for the quantitative analysis of the system's usage. In this analysis data concerning response rates, access, usage and misleading results were collected and investigated upon.

Being the "most knowledgeable" with respect to knowledge content, the Organization department, was also tasked with the preliminary evaluation of the system in verifying that the classification, search and presentation subsystems functioned correctly.

Marketing Department

In the development of the front-end software (see 0 Frontend), it was imperative that the Look and Feel would comply with the corporate standards

The Marketing department was involved in incorporating and integrating the existing look and feel into the new system. Corporate resources like the logos where sized and handed over to IT; new images and icons were created; Cascading Style Sheets (CSS) where developed in identifying fonts, character size, colours etc. In identifying styles, the Marketing department underwent an investigation on the available Search Engines (Google, Yahoo!, Bing, etc.) for inspiration.

Human Resources Department

In order for the initiative to be embraced by the employees and utilise this new tool of knowledge, resistance to change should be minimised. Since it was impossible to get all employees involved in the project, due to their large number (more than 15.000) and geographic disbursement (more than 1,500 brunches scattered all over the country), alternatives were identified.

HR would implement two e-Learning courses. The first would be responsible to "acquaint" employees with the concepts and benefits of knowledge management while the second, would be an elaborate interactive presentation of the new system.

By implementing a "Top 10 UI" (see 0 Frontend) and a custom logging mechanism (see 0 Access – Authentication & Authorization) HR could analyse "what was mostly searched" and "what was mostly viewed". This way insight was gained in troublesome areas where corrective actions should be made. In the very beginning of the system, since it coincided with the deployment of a new product, it was identified that the employees were not adequately informed since the respective product's name ranked first in all search conditions.

Another area of involvement for HR was the measuring of the effectiveness of the system in relation to users' perception. The department prepared an electronic survey to measure employee satisfaction and ease of use of the new system. Based on the results adjustments were implemented in front-end look and feel, in the results structuring and in the results representation.

4.3. Plot a Course

In attaining the expected dissemination of knowledge and realizing the competitive advantage, the "Knowledge Refinery" process was exercised. This process includes the following five stages (Zack, 1999):

- **Acquisition**. Information and knowledge created within the organization alongside with external resources should be acquired.
- **Refining**. Before storing knowledge, it will have to go through the process of value-adding, which includes cleansing, labelling, indexing, sorting, abstracting, standardising, integrating and re-categorizing.
- **Storage and Retrieval**. In this stage upstream repository and downstream knowledge distribution should be bridged.
- **Distribution**. The mechanisms to make use of the content and access the knowledge repository should be identified.

- **Presentation**. End-users' knowledge capitalization based on the knowledge repository should be the ultimate goal. Mechanisms for flexible arrangement, selection and integration should be present in order for the environment to be easy enough for the users' to use.

Based on this approach, diagrammatically shown in Figure 1, the Technical Infrastructure and the Implementation Process will be discussed in the chapters to follow.

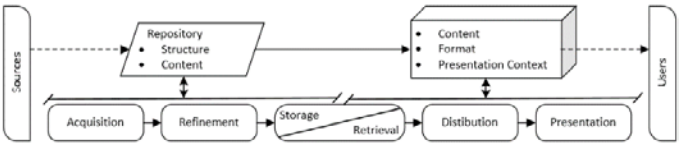

Figure 1. "Knowledge Refinery" process.

5. TECHNICAL INFRASTRUCTURE

Before presenting the actual solution and the corresponding implementation process, the underling infrastructure will be presented. Apart from the final ecosystem implemented, the alternatives evaluated will also addressed.

5.1. The Selection Process

In order to identify the system to be implemented the following options were examined: a WIKI, a Content Management System (CMS) and a Search Engine.

The WIKI was found unsuitable because in the corporate world wikis are utilised in ad-hoc problem solving within a distributed knowledge environment, whilst they are considered a misfit for applications that exhibit characteristics like formalized content and infrequently changed content (Wagner, 2004).

One of the most important features presented by CMSs in the corporate world is security and accountability since it is of utmost importance for the organization to know when a document was changed and by whom (Mescan, 2004). However, a CMS content should be authored and in this case it was imperative to utilize existing resources.

Rating Process

Having defined the available Knowledge Management systems to the steering committee, the actual process and variables attributing to the selection process were mapped on a balanced scorecard which was used to rate the available options.

The Balanced Scorecard

The actual scorecard was composed of 7 worksheets and more than 200 metrics. For simplicity, the main categories and some basic metrics will be discussed:

- **Functional Requirements**, refer to the required functionality of the system which is identified and its features are catalogued. Sample metrics include:
  - User interface simplicity
  - Categorization Capabilities
  - Indexing mechanisms
  - Available Interphases
  - Multilingual support
- **Technical Requirements**, need to be in compliance with the corporate technological roadmap. Consequently metrics concerning the technological compliance of the evaluated systems were outlined. Sample metrics include:
  - Active Directory Integration
  - Client Server vs. Web architecture
  - Network usage
  - Storage infrastructure

- **Vendor Commercial**, where the vendor that provided the systems was measured. Sample metrics include:
  - Local Geographical Presence
  - Company Size
  - Available support methods and media
  - Proposed Implementation / Installation Plan
  - SLA options
  - System Openness
- **CapEx & OpEx**, Capital Expenditure and Operational Expenditure were investigated on a 5-year basis. Sample metrics include:
  - Hardware Acquisition (CapEx).
  - Hardware Maintenance (OpEx).
  - Software Licensing (CapEx).
  - Software Support (OpEx).
  - Start-up Consultancy (CapEx).
  - Training (OpEx).

Based on the metrics and their respective weights two aggregate ratings for each proposed system / solution was calculated. Namely a) the first referred to the Technical Evaluation which measured the proposed solutions in respect to fulfilling requirements and IT infrastructure compliance while b) Financial Evaluation was more concerned with costs and Return On Investment (ROI).

Due to the economic recession experienced in Greece (Kouretas & Vlamis, 2010), it was of highest importance to minimize expenditure with the best outcome. In view of this financial restrictor it was only natural to mostly investigate open-source solutions.

Before the presentation of the selected solutions for the Proof-Of-Concept, almost all systems in the Gartner Magic Quadrant for Enterprise Search (Andrews & Koehler-Kruener, 2013) and the "Top" Open source Big Data Enterprise Search Software (IManuel, 2014) were also addressed (see Figure 2 & Figure 3) in identifying the systems to be further investigated.

Figure 2. Magic Quadrant for Enterprise Search

1. Apache Solr
2. Apache Lucene Core
3. Elasticsearch
4. Sphinx
5. Constellio
6. DataparkSearch Engine
7. ApexKB
8. Searchdaimon ES
9. mnoGoSearch
10. Nutch
11. Xapian

Figure 3. Top 11 Open Source Big data Enterprise Search Software

5.2. The Investigated Systems

It was identified, through the respective analysis depicted on the functional requirements, that Search and Indexing mechanisms should be investigated. This was primarily done in order to be able to synthesize a repository based on the existing sources. It was also agreed that since not all sources could be incorporated in their current format certain existing repositories

should be transformed. The policy would utilize existing procedures in populating and creating the repository but under the supervision of a single department responsible for the classification of knowledge.

Another fundamental principle in selecting the appropriate system, primarily directed by the ease of use, was the decision to utilize free-text-search mechanisms instead of tree-categorized structures.

A Proof-Of-Concept (POC) was implemented and knowledge resources from the silos repositories were pulled together in forming the corporate knowledge repository for the two systems that will be described.

The actual strengths and weaknesses identified in the POC will be briefly presented instead of the features available by the products. This is done since the features are relatively common across both systems and can be easily identified in the respective product sites. It is of importance to dwell on the experience gained though the actual installation, integration and hands-on usage of the systems.

The tasks on which the analysis will be detailed are the following:

- **Product Installation**. Evaluation of the respective documentation, installation and basic functionality.
- **Product Features actual performance**. Indexing various resources (text, documents, web sites, etc.) in acquiring content and metadata.
- **Interoperability** (Back-End), meaning the available connectors or means of acquiring information from different repositories (file systems, web sites, databases etc.). Extensibility, load-balancing and failover were also referenced.
- **User Interface** (Front-End), ease of customization, client server communication protocols and structures, security and accuracy were investigated.
- **Support,** availability of resources in relation to problems and implementation of features.

Two open source solutions were examined:

Searchdaimon ES which “is an open source search engine for corporate data and websites. It is as simple to use as your favourite Internet search engine, yet it has the added power of delivering results from numerous systems with standardised attribute navigation. Searchdaimon is easy to install, open and scalable.” (Searchdaimon AS, 2016)

SOLR which “is the popular, blazing-fast, open source enterprise search platform built on Apache Lucene™. SOLR is highly reliable, scalable and fault tolerant, providing distributed indexing, replication and load-balanced querying, automated failover and recovery, centralized configuration and more. SOLR powers the search and navigation features of many of the world's largest internet sites.” (The Apache Software Foundation, 2016)

5.3. The Implemented Components

In the final decision on which system should be installed, features of the systems and evaluation of the actual implementation through the POC were performed. The ease of deployment and integration, procedures in importing data, ease of usage, customizations and extendibility, indexing capabilities, classification structures and features along with results accuracy were tested on actual installations. Input from all departments involved was collected with respect to technological issues along with user experience and ease-of-use concerning the maintenance of the repository.

Finally and in addition to the above stated facts, it was decided that SOLR would be the solution to be implemented due to the importance of its ability to use several stemming algorithms as well as also support of Greek accented characters. On the other hand,

Searchdaimon ES exhibited inefficiencies in parsing documents from newer versions of Microsoft Office products. Finally, SOLR is written in Java in contrast to Searchdaimon ES which is written in C and C++ making SOLR the best choice with respect to maintenance complexity.

Backend

The SOLR 5.1.0 indexing infrastructure mechanism was installed and customised. Two cores, one for documents originated from an SQL Server and another for documents residing in a file server were parameterized.

Both cores contained the same fields as far as data and indexing was concerned and the following mechanisms were configured and utilized:

- **Data Import Handler** (DIA) with the use of Java Database Connectivity (JDBC) in order to retrieve information from the SQL Server.
- **Faceting**, in order to represent document categorization based on assigned key-words.
- **Spell Check Component** was configured for several indexed fields, in order to provide with "as you type" suggestions based on the content of the repository.
- **Request Handlers** were defined in order to facilitate the suggested feature.
- **Elevator** component was parameterised in order to highlight certain documents considered as "more-important".
- **Sharding** was utilised in merging results from both cores into a single result list. Towards this configuration a third empty core, with the same field structure, was used as a "proxy."
- **Tika Entity Processor** was configured in retrieving metadata from documents (pdf, MS Word, MS Excel, MS PowerPoint, etc.) and in parsing the actual content of the documents.

Frontend

For the frontend, it was decided to utilise browser enabled technologies in an attempt to simulate the functionality of Google with which almost all users were familiar. JQuery, JSON and Ajax were selected in developing a flexible and efficient application. The main problem in delivering the application within the corporate compliance framework was security integration, which is not provided by SOLR. The organization has in place a user authentication and authorization mechanism utilising Microsoft Active Directory (A.D.) infrastructures. SOLR is natively installed on a Jetty application server. In utilising the A.D. and the respective Jetty installation, it was identified that the best alternative would be to install a proxy mechanism in IIS that would enable URL rewriting to handle access and respective application privileges.

Four applications where developed:

- **Search User Interface (UI)**, which was designed in a Google like manner in order to search and view results.
- **Admin UI**, which was designed to provide to the system Business Administrators the functionality of loading data in the cores. This application was developed, instead of utilizing the built-in functionality of SOLR admin, as a security requirement for business users to be able to refresh content without being able to manage the infrastructure.
- **TOP 10 UI**, which was developed for selected users that needed to know which terms were queried and which documents were accessed in terms of popularity and time.
- **Search UI for Mobile**, an application utilising responsive design and alternative representation modes and functionality that suited the mobile world best.

JQuery was selected since it represents a widely accepted framework and was very close to the corporate development standards. JSON was preferred for the communication of the front-end with SOLR engine, via the REST-like API, instead of XML since it is highly integrated with JavaScript (Mikoluk, 2013). Since SOLR query responses are very fast, is

was decided to utilise the Ajax functionality that JQuery framework supplies in enhancing the user experience. The user does not have to navigate back and forth within different pages and pagination was implemented based on user scrolling.

6. IMPLEMENTATION PROCESS

IT projects tend to fail due to three reasons, namely: a) 87% of failures are attributed to exceeding initial implementation time estimates by 30%; b) 56% of failures are attributed to cost exceeding budget by 30%; c) 45% of failures are attributed to unrealized benefits (Whittaker & Whittaker, 2006). Thus all efforts are focused in delivering a system that will meet the defined needs, on time and within budget.

6.1. Project Management

In delivering a system there are several project management techniques utilized within the system development methodology (SDM) with the Waterfall, Structured Model and Object-oriented Systems Development representing the traditional approach and SCRUM representing the agile – new generation – approach (Chan & Thong, 2009). Recently Agile Methodology is gaining practitioners' widespread acceptance since its principles direct towards delivering results very quickly especially when requirements tend to change frequently (Mahalakshmi & Sundararajan, 2013) which is the case in any innovative initiative. In this case, it was very important to follow the process closely and deliver parts of the system in an incremental manner so that the whole implementation was closely monitored.

In realising the advantages provided by both methodologies the Water-SCRUM-fall methodology was employed, which is the reality for most agile organizations (Bannink, 2014):

- **Self-Adjusting**. In this case where innovation is involved it is of outmost importance to be able to evaluate the project in a progressive way and take corrective action when needed. With the delivery of partial components and sub-parts, the progress could be easily monitored and in case of delays, resource allocation adjustments were triggered.
- **Self-Monitoring**. By delivering parts of the system to the interdisciplinary committee, the process incorporated a mechanism to identify possible diversion from the expected outcome by implementing an early-warning system to trigger adjustments when required.
- **Extended Involvement**. In order to succeed, several corporate departments with diverse expertise were involved. As a result an extended number of participants contributed towards the successful completion of the project.
- **Effective Resource management**. In forming small teams of experts to tackle the project stack, the human resources were effectively involved and lengthy large-group meetings were avoided. Additionally, since the departments were not in the same location, remote collaboration was more effective due to the small size of each team.
- **Upfront work required for governance**. In a highly regulated environment, it is of outmost importance that governance is applied into any project.

With the utilization of a structured approach as outlined by the Waterfall model in conjunction with the use of sprints from the SCRUM model, the organization was able to adhere to all governance requirements in addition to "fast-track" development and implementation. The data governance, data stewardship and data security along with the organizations traditional analysis and documentation policies were utilized in defining and documenting the project. The SCRUM multidisciplinary sprints were then used in discovering applicable solutions and continuously presenting, evaluating and adjusting the initiative towards a successful implementation.

6.2. The "Knowledge Base"

In this chapter the actual infrastructure, techniques and configurations will be detailed in an attempt to present all technological aspects and challenges along with their implemented solutions.

This chapter might be a-little technical for business readers but is included in to help readers understand the implications and challenges posed by business and corporate requirements on the technological implementation.

Indexing & Searching (SOLR)

Different systems must be used as information sources. Sources must be identified but results must be combined.

Different cores where utilised in loading and querying the SOLR repository in order to segregate information sources. Since querying in normally performed against a single core, an alternative was sought in aggregating result data. SOLR sharding feature was selected, although it's primarily used in aggregating partitioned cores (see sample query in Figure 4). Experimentation and testing identified that if client side scripts were used in combining data from multiple queries, performance was unpredictable and in many cases the aggregation was erroneous.

```
shards.qt=/select&shards.info=true&shards=localhost:8983/solr/FAQ,loca
lhost:8983/solr/YS
```

Figure 4. SOLR sharding Request

Different sources and formats must be loaded into the indexing mechanism.

In importing documents from diverse sources (databases, intranet web-sites, filesystems, etc.) and different formats (Microsoft documents, pdf, xml, etc.) it was imperative to set a common bases. Based on the common fields identified across all structures, a SOLR schema was constructed and was applied to all cores in accordance to sharding requirements. Since different indexing and querying algorithms had to be implemented on the same content, e.g. title information, fields were duplicated (see Figure 5).

```
<field name="title_el" type="text_el" indexed="true" stored="true"
multiValued="true"/>
<copyField source="title" dest="title_el" />
<field name="suggest_title_el" type="text_el_suggest" indexed="true"
stored="true" multiValued="true"/>
<copyField source="title" dest="suggest_title_el" />
```

Figure 5. "Title" Filed Definitions

The user might make spelling mistakes, thus the mechanism must be able to find the correct correlations and respective synonyms or derivatives.

SOLR provides stemming algorithms but several tests had to be performed before deciding which algorithm suited the data best. Stop words, that is words that should not be indexed e.g. and, or, the, etc., for Greek and the specific industry were devised and applied. Synonym lists were also devised with the difference that since Greek is a very "rich language," synonyms identification was limited to the terms and technical acronyms used in the specific industry (see sample configuration in Figure 6).

```
<analyzer type="query">
    <tokenizer class="solr.WhitespaceTokenizerFactory"/>
    <filter class="solr.SynonymFilterFactory"
synonyms="lang/synonyms_el.txt" ignoreCase="true" expand="true"/>
    <filter class="solr.StopFilterFactory" ignoreCase="false"
words="lang/stopwords_el.txt" />
    <filter class="solr.GreekLowerCaseFilterFactory"/>
    <filter class="solr.GreekStemFilterFactory"/>
    <filter class="solr.BeiderMorseFilterFactory"
nameType="GENERIC" ruleType="APPROX" concat="false"
languageSet="english,greek"
</analyzer>
```

Figure 6. SOLR filters

Access – Authentication & Authorization

The search engine must be available to all employees to use but: a) some must be able to administer content. b) Partners and external contractors that have access to the company's intranet must be restricted from querying. c) Existing authentication and authorization infrastructure must be used.

In resolving SOLR's weakness in security the authentication and authorization would be handled by the IIS which is integrated with Active Directory, while all access to the SOLR configuration web site should be available only via the server's web browser. In implementing this concept the following customizations were performed:

- **Server Firewall**. The web port utilised by the SOLR admin tool (usually 8983) was denied for all clients apart of the localhost with the use of a firewall.
- **IIS Rewrite Rules**. The web.config of the IIS site was customized with all entries that needed to be exposed to the intranet. The application UI, SOLR search, SOLR suggest and SOLR data import URLs were defined in pattern matching rules that actually redirected to the Jetty application server. A couple of rules as examples can be viewed in Figure 7.

```
<rule name="Solr-select" stopProcessing="true">
        <match url="solrapp/([A-Z-]+)/select(.*)" />
        <action  type="Rewrite"
                 url="http://localhost:8983/solr/{R:1}/select{R:2}" />
</rule>
<rule name="Solr-Core-DataImport" stopProcessing="true">
        <match url="solrupd/([A-Z-]+)/dataimport(.*)" />
        <action  type="Rewrite"
                 url="http://localhost:8983/solr/{R:1}/dataimport{R:2}" />
</rule>
```

Figure 7. IIS Rewrite Rules

- **IIS Location Security**. In preventing unauthorised access and in essence prevent access to some users in certain URLs the security rules utilised active directory groups in relation to each and every redirection rule. Samples of such rules can be viewed in Figure 8.

```
<location path="solrupd/YS/dataimport">
  <system.webServer>
    <security>
      <authorization>
        <remove users="*" roles="" verbs="" />
        <add  accessType="Allow"
              roles="Solr _Organization_Users" />
      </authorization>
    </security>
  </system.webServer>
</location>
```

Figure 8. IIS Location Security

A mechanism should exist in identifying what keywords were used in searching and which material was accessed.

This mechanism should record all keywords typed along with viewed document in informing about "most searched" terms and "most viewed" documents. Access was constrained with

the use of pre-mentioned IIS Proxying and data were collected through a custom logging mechanism utilising a new SOLR core. The core was populated directly for the front-end by capturing search terms and clicks. Asynchronous Ajax request were to the SOLR server making the process unnoticeable to the user.

User Interphase (UI)

A free text search mechanism with underlying tree structure.

The user should be able to search with the use of a free text input box. The results should be displayed as a list, much like the results shown from a Google search, but also be categorised in order for the user to easily narrow down the search. In implementing this concept, the functionality of faceting and filtering provided by SOLR were utilised. The return data were displayed in a results list and a custom tree-like structure was formulated for the users to easily select the available documents' categories. Since the faceting results provided by SOLR where not sorted in a parent-child relationship a custom JavaScript class was implemented with the use of recursion in processing the data and constructing the tree brunches. In the representation, the funcyTree plugin was finally utilised, instead of jsTree, with minimum processing time deviations irrespective of the processed branches.

Fast preview of the results related materials / content of documents.

Since there were different sources of documents and the respective applications server's workload and response speed varied, it was decided to incorporate the full document content in the SOLR engine and represent it as HTML to the users. Another important feature towards this implementation was the uniformity in respect to user experience. The document irrespective of its original format (pdf, Ms Word, etc.) was always displayed in the designated area within the browser window irrespective of the client's configuration.

Users might have forgotten to switch keyboard thus writing in English characters while typing Greek words – they do not want to retype everything.

It is very common for bilingual users to use the wrong keyboard language setting. A custom mechanism was devised in translating the users input, querying the SOLR and displaying the result in a "Did you mean" note. The user simply clicked the translated literal, which was presented as a hyperlink, and the search was executed with the corrected terms.

Seamless flow, the user must not have to change pages to see results and be able to filter the data with simple clicks.

In order to facilitate the user experience, a mechanism was devised so that once the user reached, by scrolling, the end of the currently displayed results list, the SOLR engine was automatically queried for the next set of data which were appended to the list. The inspiration for such a mechanism came from web sites like Facebook and LinkedIn which utilise similar mechanism in displaying result lists. In addition to that in order to process user filtering and enable multi-criteria selection, a parsing mechanism was implemented. The user interface provided "Tabs" representing the different origins of the data along with an "All" tab. Advanced settings were also provided in governing filtering options. Actually once the user clicked (selected) a tab / origin or selected a category for filtering, the SOLR mechanism was queried by applying the user choices.

The user must be able to easily identify the quality of results and view the respective relative document fragments.

In presenting the user with as much information possible in the respective results lists two methods were utilised. The first was the SOLR Highlight which provided the respective fragment in which the search term was identified. The second was the utilization of hovering messages in providing the user with information while keeping the actual data of the result list to a minimum length.

The system must direct the user towards available search terms based on what he/she previously typed.
The user starts typing and based on the respective input, once the user reached a limit of 3 characters, the application queried the SOLR engine for suggestions. This way the user can visualize available search closures already indexed while typing. It resembles the functionality provided by Google and it was implemented with the use of the spellcheck component available in SOLR. An important customization was that depending on the user preferences, full phrase vs. independent words, different suggestion mechanisms were queried utilising different indexing algorithms on different fields.

If the user knows the subject area - which is different than the available document categories - he/she is interested in, there has to be a mechanism in selecting it.
SOLR faceting is primarily used in "categorizing" the query results and not for filtering documents in advance. A filter can be applied only once the query is executed. In order to tackle this requirement an extra field was defined in the SOLR schema and if the user selected a subject area the query was accordingly parametrised. The categories and the subject areas were different and eventually a parsing mechanism was also implemented, similar to the filtering parser (see Seamless flow), in the query terms.

6.3. Variety Issues

Variety is the extremely wide range of sources and data types that exist in the digital universe (Mao et al., 2015). It is identified that Variety directly affects data integrity which in turn increases erroneous data (Khan et al., 2014).

Sources

Corporate Knowledge resides in several systems and is represented / disseminated in various ways. In the organization under investigation, as described in 3.1 Knowledge "Handling", there were quite a few systems handling Knowledge. The project team investigated almost all of them in their attempt to decide which one would be utilized. Purely documented systems and systems with limited usage were disqualified. At a later stage a sprint was assigned the task of in-depth analysis of the most prominent systems. The analysis revealed that even in the most structured sources (residing in a Relational Database) much diversification existed in the way data were stored. It was agreed to initially utilise two data stores the intranet and the FAQ's - the first residing on an RDBMS and the second on a file system.

Unique challenges were posed by both data sources as follows.

RDBMS

- Flags existed in the schema that indicated if the document was active or had been replaced by a new version. The challenge was that the responsible personnel neglected to update the respective statuses.
- Keywords were assigned to each document, nonetheless keywords assignment was superficially supervised and as a result the keywords did not correspond to the documents content in several occasions.
- Relations existed between lookup tables for the categorization of documents. The system used a display hierarchy and a topic assignment. The challenge was that the two structures where used interchangeably and as a result it was doubtful which one was used in every document.
- Different types of content utilized different fields and different lookup tables.

In the first two cases the resolution came from the business analysts which reviewed and in some cases re-categorised the documents. In order to support this new categorization, an external "override" system was developed which, before parsing the document, searched the database for an override and if it existed it abolished any other metadata available from other sources.

In the latter two, system analysts and programmers of the respective system were interviewed in identifying the optimal queries that would access the correct data for the respective documents.

File System

Categorization was supposed to be provided by the metadata of the respective documents. The challenge was that only few of the metadata were populated in the documents, and if assigned values their quality was questionable. The main reason was that users tend to copy documents and change the content while not updating the metadata. Metadata like title, keywords and category were purely populated. An interdisciplinary team within a sprint was assigned the task, following a rule based system that was developed where each empty metadata field was populated with alternative values for the indexing mechanism, e.g. for the title the filename was utilized. In addition to that, all documents were re-categorised and a delimited text was utilised in defining the depth of the branch at which the document should appear.

Data Types

- The most common problem was the date's format. Depending on the language set settings, different formats were assigned. Since it was impossible to correct and normalise formats it was decided to use the metadata dates as text for reference and the filesystem dates (creation, alteration) as date fields.
- Another important problem was embedded HTML, within the documents and the respective database records. Values had embedded HTML tags in order for instance to underline a word. SOLR users must have faced this problem since there is built-in functionality in order to stripe HTML tags (HTMLStripTransformer). In addition to that a custom mechanism was developed in order to stripe none-standard HTML tags (e.g. <o:p>) and erroneous tags (e.g. <a/>).
- Another problem identified was that there was no standardization in the embedded hyperlinks. Most of them utilised relative paths. A client side mechanism was developed in normalizing the URLs by transforming the paths into absolute paths based on the respective source. The mechanism also hid any links that were references to emails (e.g. mailto:) and browser commands (e.g. _Toc, _Ref, _GoBack).
- Code-pages and character set posed a relatively small problem. Certain documents where not utilising UTF-8 and thus "stored garbage" in the data store. A mechanism was devised and before the execution of the import process an agent was executed that checked the files encoding and transformed it when required.
- Field lengths were also a source of problems. It was identified that in some cases the titles used (coming from filenames) exceeded 130 characters. In such cases the document was not indexed. Two mechanisms were developed: a) a warning system which scanned the source filesystems from lengthy filenames and sent an email to the business administrator; b) all fields were trimmed with the use of JavaScript during the document parsing phase in order to make sure that data adhered with the SOLR schema definition.

The above sources and data type Variety issues might seem trivial but it is important to understand that in most cases the problem was that data were missing or data were not displayed correctly (malformed HTML). The following were tested to identify the cause of the problem: import, parsing, indexing, query design and presentation rendering. SCRUM and the interdisciplinary teams facilitated efficient communication and 360º degrees perspective of raised issues towards the identification and eventual resolution of Variety challenges.

6.4. User Acceptance Test (UAT)

It is difficult to define the UAT phase on an on-going process. Different teams on different sprints were developing and constantly testing their partial deliverables. It was very important that independent interdisciplinary teams with the use of sprints tested all components and actually redesigned and adjusted the technical requirements. Changes were

easily implemented since independent, although interrelated, sprints produced efficient products towards the completion of the system.

It is important though to mention that the UAT and testing had the following phases:

- **System Test**. Done by the technical personnel in confirming that the program actually did what it was intended to do.
- **Functional Test**. Performed by the business super users in cooperation with the IT personnel which was intended to identify logical errors in the process.
- **Performance Test**. The system was stressed with the use of simulated simultaneous access and with abnormally high quantities, e.g. what happens if 1000 users make a query request simultaneously, what if all documents were to be returned within one query request, etc.
- **UI ease of use**. All levels of users, from super users to simple users, were presented UI alternatives e.g. positions of buttons, arrangement of element on the web page etc. in order to identify the most convenient combination.

Once the independent components were tested, in many cases, based on the results, the actual design and infrastructure were redesigned. All parts were put together in formulating the final system. At that stage an interoperability / integration test was performed just to make sure that components cooperated efficiently and the system was handed over to the users. In order to avoid any implementation failures the project team decided that the "go-live" process would utilise a phased-in approach. Gradually, departments were given access to the system and the system resources were closely monitored while a post release survey was also conducted in getting feedback from the users.

6.5. Going Public

The project team had managed to succeed in developing a high-tech project within budget and in time. Users were educated through an eLearning system.

The team subdivided the organisations' departments in relation to their population, intended/expected use of the system and average IT skills possessed by the respective employees. Each department was graded for each factor in a scale from 1 to 3. Then the order of release for each department was decided based on its grades. The process attempted to evenly distribute load and expected feedback.

7. CONCLUSION

In implementing an enterprise wide Knowledge Management system the organization adhered to the fundamental rule of Strategic Information Systems Planning (SISP), which is to identify a prominent system and cater for implementation (Lederer, 1988). Although the system encompasses state of the art technologies, the project was a solution to promote corporate welfare by disseminating knowledge.

By utilising a combination of Waterfall and SCRUM, all corporate, compliance and governance policies were followed and agility was also added. With the use of SCRUM the project team managed to have continuous control over frequently changing requirements. Additionally, an early-warning system was set in place to identify project bottlenecks and dead ends. With the independent multi-disciplinary teams implementing, testing and evaluating the outcomes constantly, the team was able to adjust and align the system to the user requirements.

The technical infrastructure challenges were significant due to integration and cutting-edge technologies employed. Nonetheless, the team was mostly engaged in resolving business issues such as which sources should be used, resolving quality of Data issues and data

classification and presentation. Furthermore, in managing interdepartmental involvement, higher-management sponsorship and involvement played a crucial role.

Another non-technical issue was “resistance to change.” People tend to be reluctant to use new systems even if the new systems “make their lives easier.” In this case the organization involved a substantial number of personnel in the UAT process thus enabling easier adoption to the system along with the important employees’ involvement and feedback.

Last but not least the organization has profited from the system since its employees had the means of identifying crucial resources for their day-to-day operations with an easy and accurate process. Call centre requests for already answered questions were minimized, while the process of handling request was standardised across the different departments. In addition to that, HR now had an accurate system from which it could identify education needs and sources of confusion directly from the “root source.”

REFERENCES

Andrews, W., & Koehler-Kruener, H. (2013). Magic Quadrant for Enterprise Search. *Gartner Inc.*, *2013*(April), 18.

Apostolou, D., & Mentzas, G. (1999). Managing Corporate Knowledge: A Comparative Analysis of Experiences in Consulting Firms. Part 1. *Knowledge & Process Management*, *6*(3), 129–138. http://doi.org/10.1002/(SICI)1099-1441(199912)6:4<238::AID-KPM64>3.0.CO;2-3

Baccarini, D., Salm, G., & Love, P. E. D. (2004). Management of risks in information technology projects. *Industrial Management & Data Systems*, *104*(4), 286–295. http://doi.org/10.1108/02635570410530702

Baker, P. (2015). Variety, not volume, biggest big data challenge in 2015 - FierceBigData. Retrieved September 2, 2015, from http://www.fiercebigdata.com/story/variety-not-volume-biggest-big-data-challenge-2015/2015-01-14

Bannink, S. (2014). Challenges in the Transition from Waterfall to Scrum–a Casestudy at Portbase. *Proceedings of the 20th Twente Student Conference on IT, Enschede, The Netherlands, January 24, 2014*. Retrieved from http://referaat.cs.utwente.nl/conference/20/paper/7427/challenges-in-the-transition-from-waterfall-to-scrum-a-casestudy-at-portbase.pdf

Chan, F. K. Y., & Thong, J. Y. L. (2009). Acceptance of agile methodologies: A critical review and conceptual framework. *Decision Support Systems*, *46*(4), 803–814. http://doi.org/10.1016/j.dss.2008.11.009

Davenport, T. H., Long, D. W. De, & Beers, M. C. (1998). Building Successful Knowledge Management Projects. *Sloan Management Review*, *39*(Winter), 43. http://doi.org/10.1016/j.ygeno.2009.01.004

Dieng, R., Corby, O., Giboin, A., & Ribière M. (1999). Methods and tools for corporate knowledge management. *Internation Journal of Human Computer Studies*, (51), 567–598. http://doi.org/10.1006/ijhc.1999.0281

Fischer, G. (2012). The "Right" Information, at the "Right" Time, in the "Right" Place, in the "Right" Way, to the "Right" Person. *Proceedings of the International Working Conference on Advanced Visual Interfaces - AVI '12*, 287. http://doi.org/10.1145/2254556.2254611

Gupta, B., Iyer, S., & Aronson, J. E. (2000). Knowledge management: practices and challenges. *Industrial Management & Data Systems*, *100*(1), 17–21. http://doi.org/10.1108/02635570010273018

IManuel. (2014). Top 11 Open Source Big data Enterprise Search Software - Predictive Analytics Today. Retrieved March 1, 2016, from http://www.predictiveanalyticstoday.com/top-open-source-big-data-enterprise-search-software/

Khan, M. A. U. D., Uddin, M. F., & Gupta, N. (2014). Seven V's of Big Data understanding Big Data to extract value. *Proceedings of the 2014 Zone 1 Conference of the American Society for Engineering Education - "Engineering Education: Industry Involvement and Interdisciplinary Trends", ASEE Zone 1 2014*. http://doi.org/10.1109/ASEEZone1.2014.6820689

Kouretas, G., & Vlamis, P. (2010). The Greek crisis: Causes and implications. *Panoeconomicus*, *57*(4), 391–404. http://doi.org/10.2298/PAN1004391K

Laney, D. (2001). 3D Data Management: Controlling Data Volume, Velocity, and Variety. *Application Delivery Strategies*, *949*, 4.

Lederer, B. A. L. (1988). The Implementation of Strategic Information Systems Planning Methodologies. *MIS Quarterly*, (September), 445–461. http://doi.org/10.2307/249212

Mahalakshmi, M., & Sundararajan, M. (2013). Traditional SDLC Vs Scrum Methodology – A Comparative Study. *International Journal of Emerging Technology and Advanced Engineering*, *3*(6), 2–6.

Mao, R., Xu, H., Wu, W., Li, J., Li, Y., & Lu, M. (2015). Overcoming the Challenge of Variety : Big Data Abstraction , the Next Evolution of Data Management for AAL Communication Systems, (January), 42–47.

Mentzas, G., Apostolou, D., Young, R., & Abecker, A. (2001). Knowledge networking: a holistic solution for leveraging corporate knowledge. *Journal of Knowledge Management*, *5*(1), 94–107. http://doi.org/10.1108/13673270110384446

Mescan, S. (2004). Why Content Management Should Be Part of Every Organization's Global Strategy. *Information Management Journal*, *38*(4), 54. Retrieved from http://www.arma.org/bookstore/files/Mescan.pdf

Mikoluk, K. (2013). JSON vs XML: How JSON Is Superior To XML. Retrieved November 27, 2015, from https://blog.udemy.com/json-vs-xml/

Rollo, T. C., & Clarke, T. (2012). Corporate initiatives in knowledge management.

Searchdaimon AS. (2016). Searchdaimon Open Source Enterprise Search. Retrieved March 1, 2016, from http://www.searchdaimon.com/

Stemberger, M. I., Manfreda, A., & Kovacic, A. (2011). Achieving top management support with business knowledge and role of IT/IS personnel. *International Journal of Information Management*, *31*(5), 428–436. http://doi.org/10.1016/j.ijinfomgt.2011.01.001

Sumner, M. (1999). Critical success factors in enterprise wide information management systems projects. *Proceedings of the 1999 ACM SIGCPR Conference on …*, SIGCPR '99 Proceedings of the 1999 ACM SIGCPR conf. http://doi.org/10.1145/299513.299722

The Apache Software Foundation. (2016). Apache Solr. Retrieved March 1, 2016, from http://lucene.apache.org/solr/

Wagner, C. (2004). Wiki: a Technology for Conversational Knowledge Management and Group Collaboration. *Communications of the Association for Information Systems*, *13*, 265–289. http://doi.org/10.1007/s10115-009-0247-6

Whittaker, B., & Whittaker, B. (2006). What went wrong ? Unsuccessful information technology projects. *Information Management & Computer Security*, *7*(1), 23–30.

Yeh, Y.-J., Lai, S.-Q., & Ho, C.-T. (2006). Knowledge management enablers: a case study. *Industrial Management & Data Systems*, *106*(6), 793–810. http://doi.org/10.1108/02635570610671489

Zack, M. H. (1999). Managing Codified Knowledge. (cover story). *Sloan Management Review*, *40*(4), 45–58. http://doi.org/Article

ABOUT THE AUTHORS

George E. Vranopoulos received the B. Sc. degree in Business Administration, Major Management Organizational Behaviour – Second Major Computer Information, in 1994 from the American College of Greece. He held the M. Sc.(with distinction) in Information Systems Management from the University of Stirling in 1996. He is a Ph.D. student in University of Plymouth, School of Computing Electronics and Mathematics. Over the last years he held senior managerial positions in the Banking sector specializing in Business Intelligence. He taught in the University of Stirling and the American College of Greece courses in Computer Information Systems discipline.

Athanassios A. Triantafyllidis was a part-time faculty member at DEREE from 1993 to 2004. Throughout his career, he served as a consultant, trainer and developer in several domestic and multinational corporations while simultaneously creating his own electronic commerce business. In 2004, he shifted his career to serve as a full-time professor within the CIS Department teaching courses across the curriculum. He currently teaches Introduction to Information Systems, Business Information Systems, Electronic Commerce and Analysis and Design of Information Systems. He is currently a doctoral research candidate with Plymouth University at the Centre for Security, Communications and Network Research. His current research interests are focused on Virtual Learning Management Systems productivity.