Software product evolution for Intellectual Capital Management: The case of Meta4 PeopleNet

Ricardo Colomo-Palacios [a], Eduardo Fernandes [b], Pedro Soto-Acosta [c,*], Marc Sabbagh [b]

[a] *Computer Science Department, Universidad Carlos III de Madrid, Av. Universidad, 30, 28911 Leganés, Madrid, Spain*
[b] *Meta4 Spain, Centro Europa Empresarial, Edificio Roma, C/. Rozabella, 8, 28290 Las Rozas, Madrid, Spain*
[c] *Department of Management & Finance, University of Murcia, Campus de, Espinardo, 30100 Espinardo, Murcia, Spain*

A B S T R A C T

Software must evolve in order to adapt to new demands and scenarios. In the case of packaged software that can be customized, this evolution implies the maintenance of both out-of-the-box and customized functionality alike. New user interface technologies provide a way to interact with applications that software vendors are forced to incorporate in order to meet new user and market requirements. Packaged software vendors are facing software evolution processes in order to attract more customers and conserve the existing ones. This case analysis describes the evolution of packaged software, Meta4 PeopleNet, towards the addiction of Rich Internet Applications features to a Human and Intellectual Capital Management tool. The findings and lessons learned presented in this case study provide useful insights for packaged software vendors facing software product evolution.

1. Introduction

Since software change is inevitable, all software systems must evolve to meet the ever-expanding needs of its users (Williams & Carver, 2010) and the changes in environments and organizations (Ciraci, van den Broek & Aksit, 2007). Software evolution is the subject of many studies both in academia and in industry and, indeed, a major part of current software development is devoted to software maintenance (Pessemier, Seinturier & Duchien, 2008). In software products (packaged software), this evolution is a key element to gain competitive advantage in an ultra competitive market. Software products are programs that are used by more than one organization (Unphon & Dittrich, 2010). They are long-living; often evolving over several decades. In this sense, a software product is the result over time of a set of activities, which appeal to various competences and knowledge (Alija & Kaba, 2008). Many of the organisations dedicated to the development of software packages face the problem of having to migrate or reengineer their products, adapting them to new technologies and functionalities because changes in business processes are almost always linked to changes in systems and technology (Lientz and Rea, 2001).

Software evolution is not a simple problem. As software products evolve, complexity shifts from lower to higher structural levels (Sangwan, Vercellone-Smith & Laplante, 2008). The evolution of software packages seems to be even complex and full of problems. In this sense, problems affecting software development are magnified in an organization whose mission is to develop packaged software (Dubé, 1998). This is the focus of this case study. The main objective is directed to present an example of the evolution of a software package aimed to adapt a software solution to new and demanding market requirements.

2. Company background

Meta4 is one of the world's leading providers of solutions for the management and development of human and intellectual capital (HICM). Founded in 1991, Meta4 has more than 1300 clients in 100 countries and these figures mean that Meta4 has become one of the top three providers worldwide for Human Resources software as well as the second Spanish software vendor within the top 100 European software companies for 2009 on the Truffle 100 index. More than 18 million employees are managed using Meta4 software. The revenues of the company in 2009 reached 47.3 million Euros with a 13% growth compared to the year before. More than 60% of this revenue comes from international sources.

Meta4 has branches in eleven countries, although the headquarters of the company is located in Madrid, Spain. Its HCM solution has a strong presence in Spanish and Portuguese-speaking countries as well as in France. It has also expanded into other markets, including the UK, Scandinavia and the US According to Gartner (2009), approximately 75% of Meta4's customers implement the solution

1

on-premises. However, Meta4 also offers options for hosting, and for a subscription license (approximately 15% of their customers have chosen a subscription license).

3. The project

This section begins describing the scenario before the project started and, then, the project scope and objectives are depicted.

3.1. The previous scenario: PeopleNet7

From an historical perspective, the first version of PeopleNet was designed around 1997 as an evolution of previous solutions from a philosophical and technological perspective. Focused on Human and Intellectual Capital Management (HICM), the PeopleNet7 solution provided support for key human capital management activities such as: workforce acquisition, workforce administration, organization management and planning, workforce time management, payroll, compensation management, skills and competency management, performance management, career and succession planning, motivation...

The technology behind the solution evolved from Client Server 2 tier (1994) and Client-server multitier (1998) or HTML portals (2000). Before the project started, some Rich Internet Application (RIA) experiences had been introduced in the product, although the scope of these features was restricted.

From the User Interface point of view, PeopleNet provides out-of-the-box forms to support HCM processes. However, the platform gives the opportunity to users to change predefined functionality or add new forms by means of a utility included. This utility is a "what you see is what you get (WYSIWYG)" tool in which form displayed during editing is very similar to the final output. This tool translates the form displayed in the editing process into a pseudo-XML format called OBL. The OBL representation is stored into the database and, once a form is invoked, a processor hosted in the client interprets the OBL code and draws the form.

Although PeopleNet's UI control are richer than common desktop applications, the advent of Windows Presentation Foundation (WPF) and its set of visual features claims for an updated version in which, on the one hand, new visual features could be included, and on the other hand, could enable the distribution of thinner and logic oriented clients.

3.2. Bringing new visual interfaces to PeopleNet: Rich Internet Application

The main objective of the project was to include RIA characteristics in the new version of PeopleNet in order to make the most of these characteristics. The adoption of RIA in the new version was a key feature in the new product release. RIA – which have many of the features and functionality of desktop software even though they run online – have become increasingly important and popular (Lawton, 2008). RIAs are breaking through the market, offering better responsiveness and a more extended user experience (Meliá, Gómez, Pérez, & Díaz, 2010). RIAs provide most of the deployment and maintainability benefits of Web applications while supporting a much richer client user interface (Linaje, Preciado, & Sánchez-Figueroa, 2007). Given some architectural similarities between PeopleNet and the Microsoft approaches, it was aimed to add RIA support to its applications using WPF, a graphical subsystem for rendering user interfaces in Windows-based applications that employs Extensible Application Markup Language (XAML), a derivative XML-based language, to define UI elements, data binding, event handling, and other features. Taking this into account, one of the main challenges of the project was to provide a way to translate all current user interfaces (both out-of-the-box and user-designed) defined in OBL to standard XAML. The second challenge is to give a solution to display both traditional OBL UI descriptions and new interfaces defined using XAML editors (either included in Microsoft Visual Studio or third parties).

The final objective regarding the inclusion of RIA via WPF was to give the chance to customers' developers and third parties to use XAML to design PeopleNet user interfaces by means of any XAML designer. Another consequence of this new feature was that user interface engine was going to be executed outside the PeopleNet environment, thus, providing a clear separation between the user interface and the business logic and making PeopleNet a lighter platform.

4. Lessons learned

As suggested by Erdogmus (2008), software processes can be placed inside a triangular map according to their emphasis relative to three aspects, represented by the vertices: people, technology, and rigor. The lessons learned during the different phases of the project can be classified into these three categories.

4.1. Personnel issues

Clearly, software development is an intellectual endeavour, and serious software development is a team activity (Erdogmus, 2008).The first important conclusion about the case in the people vertices is the importance of the role played by the project manager during the process. According to Ebert (2007), the success of any product depends on the skills and competences of its product manager. The role of software product manager has been addressed in the literature (e.g. Barney, Aurum & Wohlin, 2008; Fricker, Gorschek, Byman, & Schmidle, 2010; Karlsson, Dahlstedt, Regnell, Natt och Dag, & Persson, 2007; Lehtola & Kauppinen, 2006; Trienekens, Kusters, Kriek, & Siemons, 2009; Weerd, Brinkkemper & Versendaal, 2010) although the product manager's role in software engineering has so far not been summarized in a comprehensive perspective (Ebert, 2007). The product manager is responsible, among other things, for product definition, product release and product lifecycles, creating an effective multifunctional product introduction team and – above all – preparing and implementing the business case (Ebert, 2007). In our particular case, software product manager was a person with huge experience in the company. He was the man in charge of balancing the so called Bermuda triangle of software companies sales/marketing, product management and technical management. This role is quite important in the case of Meta4, since all important decisions about the product are made by this person, either as a single decision-maker or together with other colleagues. He guaranteed the joint of the product and the project.

Another issue is the importance of counting on with key people during the software evolution process. The importance of people in software processes has been outlined in the literature (e.g. Chang, 2010; Colomo Palacios et al., 2010; Naranjo-Gil, 2009; Trigo et al., 2010). In the case explained in this paper, the whole project was designed to phase in professionals in key stages (integration, software core change...). The nominal productivity of these professionals doubled that of other professionals involved in the project. These individual differences, that are not new for software practitioners and researchers alike, had a deep impact in the success of the project.

Furthermore, another personnel issue is concerned with the participation of an external stakeholder in the development process, in this case, Universidad Carlos III of Madrid (Uc3m). Although Meta4 holds an active R&D (Research and Development)

department, the participation of Uc3m in the process brought an outside of the company vision that was very positive for the project. As suggested by Dhungana, Grünbacher, Rabiser, and Neumayer (2010), software evolution development teams require a mix of skills and roles. However, there were also some other issues related to this participation. The first approach to perform this collaboration was directed to build large teams (up to 10 members) and plan their work for a few months. Due to the low productivity of this approach, smaller teams (4 members) were formed and the schedule was changed to provide the team with a long term scenario. The differences in productivity obtained were very significant. The roots of this change could be found in Meta4's strong organizational culture of that leads to a low rotation of personnel in the R&D department. Lewis, Belliveau, Herndon, and Keller (2007) reported that membership change could tend to expect socially isolated members and that newcomers considered visitors (expected to remain in the group for only a short time) could receive less support in personnel integration. Apart from that, the culture of software vendors has been reported as highly individualistic in the literature (e.g. Carmel & Sawyer, 1998; Dubé, 1998). For instance, Dubé (1998) reported that team integration within these environments could be unsuccessful. Thus, designing a team aimed to interact for a long time (2 years) could provide better integration results and softer interpersonal relationships.

4.2. Rigor

Scalable business need managerial and procedural rigor (Erdogmus, 2008). From this point of view, one main conclusion can be found. This conclusion is the need of an established methodology to guide the process. The special needs of packaged software process model have been pointed out from the literature since early nineties (e.g. Carmel & Becker, 1995). To cope with these special needs the method presented in García-Crespo, Colomo-Palacios, Gómez-Berbís, and Ruano-Mayoral (2009) was adapted to the case in hand. The methodology, developed by Uc3m, employs an approach that focuses on the continuous improvement of the process, guiding the whole software evolution process. This approach was implemented in the model by means of continuous inspection and supply. The adoption of the methodology resulted to be successful. However, in order to adapt it better to Meta4's operations, a more agile version of the method was designed and implemented. This method included the observation of, among others, many well grounded in house software engineering practices, including practices such as requirements engineering, coding standards, code inspections, regressive unit testing among many others.

4.3. Technology

Regarding the development of the project its technology, the first lesson learned is related to the ways of translating OBL format into XAML. After testing several prototypes that included different solutions to the matter, the team decided to pursue a solution in which the code is stored in a database using OBL format, before editing it with a XAML editor. This solution implies that an on-demand mechanism must be included in the client and that such mechanism should include a parser to convert OBL in XAML. One important issue about this conversion is the necessity of minimizing the code when it is converted to XAML. The purpose of this is not other than producing a highly standardized XAML. To achieve this objective, OBL is required to be transformed to XAML after three consecutive steps: first, OBL is transformed into a XML execution; second, this XML execution is transformed into a XAML visualization (without transformed code); and finally, the XAML visualization is converted into a fully operational XAML (with minimized code).

The second lesson learned is about the overall performance of the solution. The decision of keeping the form stored in OBL implies that an on-demand conversion process must be designed. The OBL format has many features not supported in the standard XAML representation. Thus, keeping the original OBL format has many advantages but it also implies the need of conversion and, then, the addition of an extra functionality. The conversion must be fast enough to produce a display of the form in an adequate timing. The conversion process was the centre of the debate around the project for months. Once a decision was taken and a first approach to the solution was tested successfully, several project cycles were designed to expand the functionality of the presentations stored in OBL and, more recently, to shape conversion and overall application performance.

The third conclusion is the success in layers separation. In this new scenario in which business logic and layers presentation are separated, the flexibility of the approach finally produced better application performance and an optimized use of video capabilities.

The forth conclusion is the importance of code refactoring. Refactoring is an approach to improve the design of software without changing its external behaviour which means it always gives the same output with the same input after the change is applied (Fowler, Beck, Brant, Opdyke, & Roberts, 2001). To cope with the spiral software complexity, there is an urgent need for techniques that reduce software complexity by incrementally improving the internal software quality; in the case of object oriented software, this is done by means of refactoring (Mens & Tourwe, 2004). Refactoring can be used to improve software design by tidying up code, moving code to the right place or removing code (Counsell, Loizou, & Najjar, 2010). In the particular case of PeopleNet, although some aspects like software extensibility, modularity, reusability, complexity and maintainability were considered, efficiency was the main issue.

The final conclusion about technology is about application performance. Previous approaches to shape PeopleNet7's video performance were not applicable to the new one. The new peculiarities of WPF require new optimization methods and tools. To reach a valid application performance, specific task related to the use of cache, video memory usage and code optimization techniques were designed and implemented along the development process. This optimization process leaded to a final solution in which performance scores are more than acceptable.

5. Conclusions

This case study highlights some of the realities involved when packaged software vendors face software evolution. Findings from this case study revealed three different conclusions. First, the importance of people in product evolution processes. Second, the need of a methodology to guide the process in an accurate and consensuated way and, finally, the need of addressing the technological factors involved in the process. Although technological factors are very important and seem to be crucial for both the impact they provoke in the product as a whole and the approach for fulfilling requirements, the most important factor is people.

Following the path described by DeMarco and Lister (1987), authors believe that aspects like teamwork, group dynamics, nominal developer productivity and organizational factors, among others, are the main issues in software systems. In this sense, until today, software is developed by people. The impact of these factors is also crucial in software evolution. In this scenario, human-centricity (Peopleware in a nutshell) is a key aspect. A human-centric process emphasizes collaboration, recognizes the importance of effective leadership, and caters to the needs of creative professionals who take pride in their work (Erdogmus, 2008). In our particular case, the main finding could be the

importance of the effective leadership played by the product manager. That leadership comprised the use of both soft and technical skills. The importance of both is unquestionable but, in line with Erdogmus' (2008) arguments, technical orientation adds meat to human-centricity. In the particular case analyzed here, both fundamentals and software technology competences helped in several hard decisions, being even more crucial in the selection of the conversion-process setup, which was probably the harder decision taken in this project.

Authors hope that the findings presented in this case will not only shed new light on packaged software evolution, but also offer some clues and guidelines to both product managers and other managerial staff on the human and technical issues related to the evolution of software. The current work proposes two issues which should be explored in future research. In first place, the study of psychological and cultural factors determining the performance of different software workers in the context of software evolution should be further researched. In the second place, the study of the role performed by the product manager should be further investigates, since globalization tendencies and software development are constantly increasing (García-Crespo, Colomo-Palacios, Soto-Acosta, & Ruano-Mayoral, 2010; Hernández-López, Colomo-Palacios, García-Crespo, & Soto-Acosta, 2010).

Acknowledgment

References

Alija, S. A., & Kaba, A. B. (2008). Evolution support mechanisms for software product line process. *The Journal of Systems and Software*, 81(10), 1784–1801.

Barney, S., Aurum, A., & Wohlin, C. (2008). A product management challenge: Creating software product value through requirements selection. *Journal of Systems Architecture*, 54(6), 576–593.

Carmel, E., & Becker, S. (1995). A process model for packaged software development. *IEEE Transactions on Engineering Management*, 42(1), 50–61.

Carmel, E., & Sawyer, S. (1998). Packaged Software Teams: What makes them so special? *Information Technology and People*, 11(1), 6–17.

Chang, C. L. (2010). The study of the turnover of MIS professionals—The gap between Taiwanese and US societies. *International Journal of Information Management*, 30(4), 301–314.

Colomo Palacios, R., Tovar Caro, E., García Crespo, A., & Gómez Berbís, J. M. (2010). Identifying technical competences of IT professionals: The case of software engineers. *International Journal of Human Capital and Information Technology Professionals*, 1(1), 31–43.

Ciraci, S., van den Broek, P., & Aksit, M. (2007). A taxonomy for a constructive approach to software evolution. *Journal of Software*, 2(2), 84–97.

DeMarco, T., & Lister, T. (1987). *Peopleware: Productive Projects and Teams*. New York: Dorset House.

Erdogmus, H. (2008). Essentials of software process. *IEEE Software*, 25(4), 4–7.

Counsell, S., Loizou, G., & Najjar, R. (2010). Evaluation of the 'replace constructors with creation methods' refactoring in Java systems. *IET Software*, 4(5), 318–333.

Dhungana, D., Grünbacher, P., Rabiser, R., & Neumayer, T. (2010). Structuring the modeling space and supporting evolution in software product line engineering. *The Journal of Systems and Software*, 83(7), 1108–1122.

Dubé, L. (1998). Teams in packaged software development The Software Corp. experience. *Information Technology and People*, 11(1), 36–61.

Ebert, C. (2007). The impacts of software product management. *Journal of Systems and Software*, 80(6), 850–861.

Fowler, M., Beck, K., Brant, J., Opdyke, W., & Roberts, D. (2001). *Refactoring: Improving the design of existing code*. Upper Saddle River, MJ: Addison-Wesley.

Fricker, S., Gorschek, T., Byman, C., & Schmidle, A. (2010). Handshaking with implementation proposals: Negotiating requirements understanding. *IEEE Software*, 27(2), 72–80.

García-Crespo, Á., Colomo-Palacios, R., Gómez-Berbís, J. M., & Ruano-Mayoral, M. (2009). A project management methodology for commercial software reengineering. *International Journal of Project Organisation and Management*, 1(3), 253–267.

García-Crespo, Á., Colomo-Palacios, R., Soto-Acosta, P., & Ruano-Mayoral, M. (2010). A qualitative study of hard decision making in Managing Global Software Development Teams. *Information Systems Management*, 27(3), 247–252.

Gartner (2009). *Magic quadrant for Employee Performance Management Software*. http://www.gartner.com/DisplayDocument?id=873313.

Hernández-López, A., Colomo-Palacios, R., García-Crespo, A., & Soto-Acosta, P. (2010). Trust building process for Global Software Development Teams. A review from the literature. *International Journal of Knowledge Society Research*, 1(1), 66–83.

Karlsson, L., Dahlstedt, A. G., Regnell, B., Natt och Dag, J., & Persson, A. (2007). Requirements engineering challenges in market-driven software development—An interview study with practitioners. *Information and Software Technology*, 49(6), 588–604.

Lawton, G. (2008). New ways to build rich internet applications. *IEEE Computer*, 41(8), 10–12.

Lehtola, L., & Kauppinen, M. (2006). Suitability of requirements prioritization methods for market-driven software product development. *Software Process Improvement and Practice*, 11(1), 7–19.

Lewis, K., Belliveau, M., Herndon, B., & Keller, J. (2007). Group cognition, membership change, and performance: Investigating the benefits and detriments of collective knowledge. *Organizational Behavior and Human Decision Processes*, 103(2), 159–178.

Lientz, B. P., & Rea, K. P. (2001). *Breakthrough technology project management* (2nd ed.). Oxford, UK: Butterworth-Heinemann.

Linaje, M., Preciado, J. C., & Sánchez-Figueroa, F. (2007). Engineering rich Internet application user interfaces over legacy Web models. *IEEE Internet Computing*, 11(6), 53–59.

Meliá, S., Gómez, J., Pérez, S., & Díaz, O. (2010). Architectural and technological variability in rich Internet applications. *IEEE Internet Computing*, 14(3), 24–32.

Mens, T., & Tourwe, T. (2004). A survey of software refactoring. *IEEE Transactions on Software Engineering*, 30(2), 126–139.

Naranjo-Gil, D. (2009). Management information systems and strategic performances: The role of top team composition. *International Journal of Information Management*, 29(2), 104–110.

Pessemier, N., Seinturier, L., & Duchien, L. (2008). A component-based and aspect-oriented model for software evolution. *International Journal of Computer Applications in Technology*, 31(1/2), 94–105.

Sangwan, R. S., Vercellone-Smith, P., & Laplante, P. A. (2008). Structural epochs in the complexity of software over time. *IEEE Software*, 25(4), 66–73.

Trienekens, J. J. M., Kusters, R., Kriek, D., & Siemons, P. (2009). Entropy based software processes improvement. *Software Quality Journal*, 17(3), 231–243.

Trigo, A., Varajão, J., Soto-Acosta, P., Barroso, J., Molina-Castillo, F. J., & Gonzalvez-Gallego, N. (2010). IT professionals: An Iberian snapshot. *International Journal of Human Capital and Information Technology Professionals*, 1(1), 61–75.

Unphon, H., & Dittrich, Y. (2010). Software architecture awareness in long-term software product evolution. *The Journal of Systems and Software*, 83(11), 2211–2226.

van de Weerd, I., Brinkkemper, S., & Versendaal, J. (2010). Incremental method evolution in global software product management: A retrospective case study. *Information and Software Technology*, 52(7), 720–732.

Williams, B. J., & Carver, J. C. (2010). Characterizing software architecture changes: A systematic review. *Information and Software Technology*, 52(1), 31–51.

Ricardo Colomo-Palacios is an Associate Professor at the Computer Science Department of the Universidad Carlos III de Madrid. His research interests include applied research in Information Systems, Software Project Management, People in Software Projects and Social and Semantic Web. He received his PhD in Computer Science from the Universidad Politécnica of Madrid (2005). He also holds a MBA from the Instituto de Empresa (2002). He has been working as software engineer, project manager and software engineering consultant in several companies including Spanish IT leader INDRA. He is also an Editorial Board Member and Associate Editor for several international journals and conferences and Editor in Chief of International Journal of Human Capital and Information Technology Professionals.

Eduardo Fernandes received his PhD in Physical Sciences from the European Space Agency (ESA)/Universidad Complutense de Madrid (1995). He is currently Chief Architect and Technology Product Manager at Meta4. In this position he is responsible for strategic decisions which affect the Meta4 product line, for example PeopleNet. His main role is analize technological market trends and technological frameworks to merge them to Meta4 products. Before reaches meta4 he was responsible for the development software team for the MiniSat-01 (ESA), the first Spanish mini-satellite (1997). While working on his master in physics he was responsible for the automation of the biggest telescope in Brazil, deeply involved in both software and hardware developments (1992).

Pedro Soto-Acosta is an Associate Professor of Management at the University of Murcia (Spain). He holds a PhD in Management Information Systems (MISs) and a Master's degree in Technology Management from the University of Murcia. He received his BA in Accounting and Finance from the Manchester Metropolitan University (UK) and his BA in Business Administration from the University of Murcia. He attended Postgraduate Courses at Harvard University (USA). His work has been published in journals such as the African Journal of Business Management, the European Journal of Information Systems, the International Journal of Information Management and the Information Systems Management. He is also an Editorial Board Member and Associate Editor for several International Journals and Chairman of various International Conferences.

Marc Sabbagh is Vice President of Meta4 R&D Worldwide since January 2004. He is responsible for a group of 200 developers based in Europe and the Americas, pri-

marily focused on driving evolution of Meta4's flagship product and researching the latest in technology and people management trends. During his tenure, Marc has been a key driver of Meta4's participation in a number of significant European Innovation Projects amongst them LIP, SWAP, ASP-Net and ONTOLOGGING. He joined Meta4 in 1997 as R&D and Services Director in France and then became Director of Worldwide Product Strategy and Management. Before joining Meta4, he was Technical Director for HR Access at IBM Global Services group in France. Marc is an electrical engineer from the Ecole Supérieure d'Electricité Paris, France.