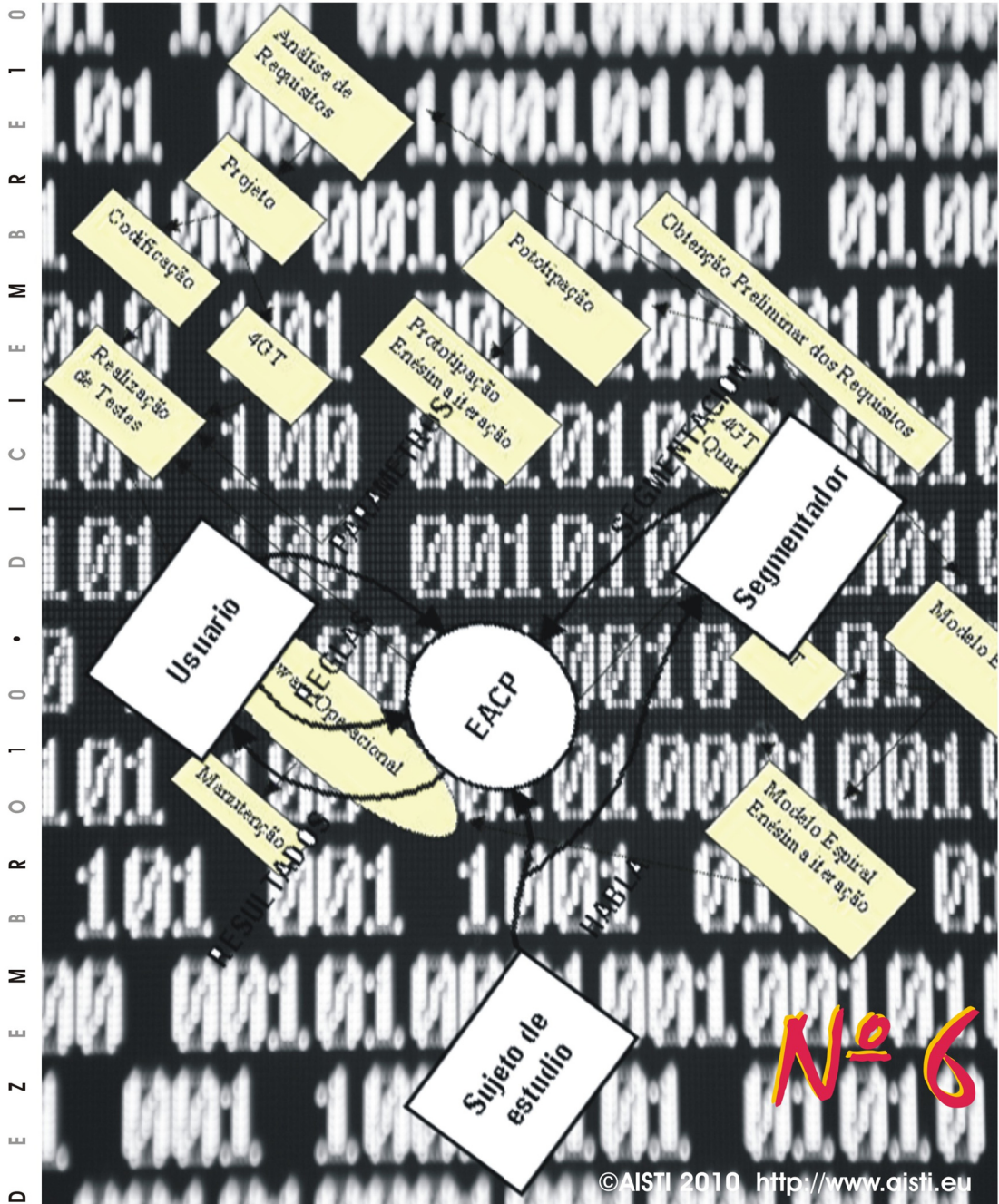




Revista Ibérica de Sistemas e Tecnologias de Informação
 Revista Ibérica de Sistemas y Tecnologías de Información



Edição / Edición

Nº 6, 12/2010

Tiragem / Tirage: 500

Preço por número / Precio por número: 12,5€

Subscrição anual / Suscripción anual: 20€ (2 números)

ISSN: 1646-9895

Depósito legal:

Indexação / Indexación: EBSCO, LatinIndex, Scielo

Propriedade e Publicação / Propriedad y Publicación

AISTI – Associação Ibérica de Sistemas e Tecnologias de Informação

Rua de Lagares, 523 - Silvares, 4620-646 Lousada, Portugal

E-mail: risti@aisti.eu

Web: <http://www.aisti.eu>

Ficha Técnica

Director

Álvaro Rocha, Universidade Fernando Pessoa

Coordenador da Edição / Coordinador de la Edición

Jose Antonio Calvo-Manzano Villalón, Universidad Politécnica de Madrid

Conselho Editorial / Consejo Editorial

Carlos Ferrás Sexto, Universidad de Santiago de Compostela

Jose Antonio Calvo-Manzano Villalón, Universidad Politécnica de Madrid

Luís Borges Gouveia, Universidade Fernando Pessoa

Manuel Pérez Cota, Universidad de Vigo

Maria Manuela Cruz-Cunha, Instituto Politécnico do Cávado e do Ave

Ramiro Gonçalves, Universidade de Trás-os-Montes e Alto Douro

Secretariado Editorial

Avelino Victor, Instituto Superior da Maia e Instituto de Informática do Porto

Paulo Teixeira, Instituto Politécnico do Cávado e do Ave

Conselho Científico / Consejo Científico

Adolfo Lozano-Tello, Universidad de Extremadura

Aldemar Santos, Universidade Federal de Pernambuco

Ana Paula Afonso, Instituto Superior de Contabilidade e Administração do Porto

Antoni Lluís Mesquida Calafat, Universitat de les Illes Balears

António Godinho, ISLA-Gaia

Antonio Guevara, Universidad de Málaga

Antonia Mas Pichaco, Universitat de les Illes Balears

Armando Mendes, Universidade dos Açores

Arturo J. Méndez, Universidad de Vigo

Carlos Costa, Universidade de Aveiro

Carlos Rabadão, Instituto Politécnico de Leiria

Carlos Vaz de Carvalho, Instituto Politécnico do Porto, ISEP

Carmen Galvez, Universidad de Granada
Claus Kaldeich, Polytechnic University of Namibia
Eduardo Destefanis, Universidad Tecnologica Nacional da Argentina
Eduardo Luís Cardoso, Universidade Católica Portuguesa - Porto
Feliz Gouveia, Universidade Fernando Pessoa
Fernando Bandeira, Universidade Fernando Pessoa
Fernando Diaz, Universidad de Valladolid
Francisco Restivo, Universidade do Porto, FEUP
García Pérez-Schofield Baltasar, Universidad de Vigo
Gonçalo Paiva Dias, Universidade de Aveiro
Henrique Gil, Instituto Politécnico de Castelo Branco
Ivan Garcia, Universidad Tecnologica de la Mixteca
Jaime S. Cardoso, INESC e Universidade do Porto, FEUP
Javier Garcia Tobio, CESGA-Centro de Supercomputacion de Galicia
Jesús Pardillo, Universidad de Alicante
João Barroso, Universidade de Trás-os-Montes e Alto Douro
João Paulo Costa, Universidade de Coimbra
João Sarmento, Universidade do Minho
João Tavares, Universidade do Porto, FEUP
Joana Maria Segui Pons, Universitat de les Illes Balears
Joaquim Cezar Filipe, Universidade de São Paulo
Joaquim Reis, Instituto Superior de Ciências do Trabalho e da Empresa
Jörg Thomaschewski, University of Applied Sciences OOW - Emden
José Adriano Pires, Instituto Politécnico de Bragança
Jose M Molina, Universidad Carlos III de Madrid
Juan Carlos González Moreno, Universidad de Vigo
Juan José de Benito Martín, Universidad de Valladolid
Juan Manuel Fernández-Luna, Universidad de Granada
Juan-Manuel Lopez-Zafra, Universidad Complutense de Madrid
José Silvestre Silva, Universidade de Coimbra
Luciano Boquete, Universidad de Alcalá
Luis de Campos, Universidad de Granada

Luis Fernandez-Sanz, Universidad de Alcalá
Luís Paulo Reis, Universidade do Porto, FEUP
Luisa María Romero-Moreno, Universidad de Sevilla
Magdalena Arcilla Cobián, Universidade Nacional de Educación a Distancia
Marco Painho, Universidade Nova de Lisboa, ISEGI
Maria Clara Silveira, Instituto Politécnico da Guarda
Maria Helena Monteiro, Universidade Técnica de Lisboa, ISCSP
María J. Lado, Universidad de Vigo
Maria João Castro, Instituto Politécnico do Porto, ISCAP
Martin Llamas-Nistal, Universidad de Vigo
Mercedes Ruiz, Universidad de Cádiz
Miguel de Castro Neto, Universidade Nova de Lisboa, ISEGI
Miguel Mira da Silva, Universidade Técnica de Lisboa, IST
Montserrat Sebastià, Universitat de Barcelona
Nuno Ribeiro, Universidade Fernando Pessoa
Orlando Belo, Universidade do Minho
Paulo Pinto, Universidade Nova de Lisboa, FCT
Paulo Silva, Universidade Lusíada de Famalicão
Pedro Nogueira Ramos, Instituto Superior de Ciências do Trabalho e da Empresa
Pedro Pimenta, Universidade do Minho
Pedro Sanz Angulo, Universidad de Valladolid
Sergio Gálvez Rojas, Universidad de Málaga
Thomas Panagopoulos, Universidade do Algarve
Tomás San Feliu Gilabert, Universidad Politécnica de Madrid
Vitor Santos, Universidade de Trás-os-Montes e Alto Douro
Xose A. Vila, Universidad de Vigo

Editorial

Es un placer participar en la edición de este número 6 de la Revista Ibérica de Sistemas y Tecnologías de la Información (RISTI), dedicado a la temática de Ingeniería de Software (IS). Se han recibido numerosas contribuciones (49) para su evaluación por parte del Consejo Científico, aceptándose solamente 7 artículos. Por tanto, la tasa de aceptación ha sido de un 14%. Se han tenido que rechazar artículos muy buenos, y que desde aquí animo a los autores a que vuelvan a enviarlos a revistas y congresos relacionados no ya con la Ingeniería de Software, sino con las Tecnologías de la Información en su sentido más amplio.

En los últimos años hemos asistido a un crecimiento de la demanda de software, que se ha venido aplicando en la resolución de tareas cada vez más complejas y de mayor valor añadido. En estas circunstancias, nos podemos hacer la siguiente pregunta: ¿la industria del software está preparada para entregar el software que se necesita en los próximos años, de acuerdo a las exigencias que los clientes imponen? Desgraciadamente, para muchos investigadores, la respuesta es no. El proceso de desarrollo de software dista mucho de ser un proceso maduro porque no es escalable y no puede satisfacer la demanda.

Actualmente sigue existiendo un consenso en la industria sobre que el desarrollo de un producto tan complejo como es el software, debe ser realizado con la ayuda de un proceso de ingeniería y gestión, y con métricas que permitan predecir de forma efectiva los niveles de riesgo (hablando en términos de coste, plazos y defectos, principalmente). La llamada “crisis del software” del año 1969 perdura hasta nuestros días, estando presentes aún los problemas relativos al fracaso de los proyectos. Los principales factores que afectan al fracaso de un proyecto software son: requisitos incompletos, carencia de implicación del usuario, carencia de recursos, expectativas no reales, falta de apoyo ejecutivo, cambios en los requisitos, falta de planificación, falta de infraestructura, falta de gestión, falta de cultura de Tecnologías de la Información, y otros.

Los artículos publicados en este número de RISTI tratan de cubrir algunos de estos factores. Los artículos se presentan siguiendo el orden lógico del ciclo de vida.

El primer artículo describe una metodología de desarrollo centrada en el usuario, y que ha sido aplicado con éxito al campo del software educativo. Se presentan sus fases, y los procedimientos y técnicas aplicables.

El segundo artículo propone una arquitectura de software para sistemas distribuidos y colaborativos. Esta arquitectura ha sido aplicada al desarrollo de entornos distribuidos de aprendizaje, siguiendo un modelo de arquitectura por capas, facilitando la reutilización de componentes.

El tercer artículo presenta una herramienta para la generación automática de aplicaciones web. Para ello, se basa en los principios de la ingeniería dirigida por modelos, presentando la definición y codificación de las transformaciones entre modelos y de modelo a texto.

El cuarto artículo está orientado a las pruebas y presenta una herramienta para especificar casos de prueba a partir de casos de uso. Además, esta herramienta incorpora elementos para verificar y validar la trazabilidad entre las pruebas y las fases del ciclo de vida anteriores a las pruebas.

El quinto artículo se centra en la gestión de proyectos, uno de los aspectos primordiales en cualquier empresa. Concretamente, se presenta una aproximación a la asignación óptima de recursos, de forma que se conozcan los recursos principales y de soporte que se deben asignar a las actividades del proyecto para obtener los mejores resultados y de forma más económica.

El sexto artículo fue seleccionado como el mejor artículo de las Jornadas Iberoamericanas de Ingeniería del Software e Ingeniería del Conocimiento (JIISIC 2010), celebrado en Mérida, Yucatán (México), celebrado del 10 al 12 de Noviembre de 2010. Este artículo pone de manifiesto como llevar a cabo la implantación práctica de MoProSoft (Modelo de Procesos para la Industria del Software en México). Para ello, se propone una estrategia y se presenta un caso de estudio.

Finalmente, el séptimo artículo está orientado a la mejora de proceso software. En concreto, nos presenta una nueva forma de llevar a cabo la mejora de procesos cuando ésta se realiza en un entorno de referencia multimodelo, de forma que se minimice la resistencia al cambio.

No me queda más que dar la enhorabuena a todos aquellos autores cuyos artículos han sido publicados en este número de RISTI.

Jose A. Calvo-Manzano Villalón

Universidad Politécnica de Madrid

Índice

Metodologia Híbrida de Desenvolvimento Centrado no Utilizador aplicada ao Software Educativo	1
<i>António Pedro Costa, Maria João Loureiro, Luís Paulo Reis</i>	
Uma proposta de Arquitetura de Software para Construção e Integração de Ambientes Virtuais de Aprendizagem	17
<i>Amanda Monteiro Sizo, Adriano Del Pino Lino, Eloi Luiz Favero</i>	
M2DAT-HYMO: una herramienta basada en MDA para la generación automática de aplicaciones Web a partir del modelo del hipertexto	31
<i>Feliu Trias Nicolau, Iván Santiago Viñambres, Juan Manuel Vara, Valeria de Castro</i>	
SAM - Sistema Automatizado del Método MECAP para Especificar Casos de Prueba	45
<i>Edumilis Méndez, María A. Pérez, Kenyer Domínguez, Luis E. Mendoza</i>	
Um Algoritmo Genético para Programação de Projectos em Redes de Actividades com Complementaridade de Recursos.....	59
<i>Helder C. Silva, José A. Oliveira, Anabela P. Tereso</i>	
Lecciones Aprendidas en la Implantación de MoProSoft en una empresa escolar: caso AvanTI	73
<i>Angélica Astorga, Brenda Flores, Gloria Chavez, Mónica Lam, Araceli Justo</i>	
Experiencia en la Mejora de Procesos de Gestión de Proyectos Utilizando un Entorno de Referencia Multimodelo	87
<i>Cuevas Gonzalo, Mejia Jezreel, Muñoz Mirna, San Feliu Tomás</i>	

Metodologia Híbrida de Desenvolvimento Centrado no Utilizador aplicada ao Software Educativo

António Pedro Costa¹, Maria João Loureiro¹, Luís Paulo Reis²

apcosta@ua.pt, mjoao@ua.pt, lpreis@fe.up.pt

¹ CIDTFF - Centro de Investigação Didáctica e Tecnologia na Formação de Formadores, DE/UA
Departamento de Educação, Universidade de Aveiro, Campus de Santiago, 3800-193, Aveiro, Portugal.

² LIACC - Laboratório de Inteligência Artificial e Ciência de Computadores da Universidade do Porto,
DEI/FEUP - Departamento de Engenharia Informática, Faculdade de Engenharia da Universidade do Porto,
Rua Dr. Roberto Frias s/n, 4200-465, Porto, Portugal.

Resumo: Este artigo descreve a Metodologia¹ Híbrida de Desenvolvimento Centrado no Utilizador (MHDCU). Trata-se de um processo de desenvolvimento simples, iterativo e incremental que tem como “alicerces” princípios do Design Centrado no Utilizador (DCU), especificados na International Organization for Standardization - ISO 13407. Na sua base encontra-se a estrutura disciplinada de processos de desenvolvimento, bem como práticas e valores dos métodos ágeis de desenvolvimento de software. O processo é constituído por 4 fases principais: planeamento do guião didáctico, design do storyboard, implementação e manutenção/operação. A prototipagem e a avaliação são realizadas de modo transversal a todo o processo. A MHDCU está ser implementada numa Pequena e Média Empresa (PME) de desenvolvimento de recursos educacionais. O primeiro recurso que teve por base esta metodologia foi o Courseware Sere – O Ser Humano e os Recursos Naturais.

Palavras-chave: Metodologias de Desenvolvimento de *Software Educativo*; Engenharia de *Software*; Métodos Ágeis; Metodologia Híbrida de Desenvolvimento Centrado no Utilizador; Design Centrado no Utilizador.

Abstract: This paper describes the Hybrid User Centered Development Methodology (HUCDM). This methodology is a simple, iterative and incremental development process. The methodology is based on structured disciplined development processes, on principles of User Centered Design (UCD) processes, specified in the International Organization for Standardization - ISO 13407, as well as on practices and values of agile methods for software development. The process consists of 4 main phases: planning of educational guidelines, storyboard design, implementation and maintenance/operation. The prototyping and evaluation are carried out in order to cross the entire process. The HUCDM is being implemented in a Small and Medium Enterprise (SME) of educational

¹ Do grego *methodos* = 'caminho para chegar a um fim' e *logia* = 'estudo de'

resources development. The first resource that was based in this methodology was the Courseware Ser_e - The Human Being and the Natural Resources.

Keywords: Educational Software Development Methodologies; Software Engineering; Agile Methods; Hybrid User Centered Development Methodology; User Centered Design.

1. Introdução

O desenvolvimento de *software* é uma actividade de elevada complexidade, sendo que na maioria dos casos ocorre sem ser devidamente planeado, suportado por decisões de “curto prazo” (Fowler, 2005). Esta abordagem poderá funcionar para pequenos pacotes de *software*, mas à medida que o sistema cresce, cresce também a dificuldade de lhe adicionar novas funcionalidades. Complementarmente, Shneiderman & Plaisant (2005) afirmam que 60% dos projectos de desenvolvimento de *software* falham na definição dos objectivos. Este problema surge nomeadamente porque na maioria dos projectos existe falta de comunicação entre os elementos da equipa e, entre elementos da equipa e os utilizadores finais. Portanto, a escolha do método adequado para o desenvolvimento de um *software* terá vantagens em termos de qualidade, económicas e competitivas, mas caso seja seleccionada um método menos adequado, o mais provável será o projecto ultrapassar os limites temporais, existindo falhas e consequentemente problemas económicos (Toth, 2005).

Neste artigo é efectuado um breve enquadramento teórico sobre a evolução das metodologias de desenvolvimento de *software*, referindo alguns exemplos de métodos utilizados no desenvolvimento de pacotes de *software* educativo. Posteriormente, é efectuada uma breve descrição ao DCU, baseada na exposição dos pressupostos que o constituem e a sua importância no desenvolvimento de *software* educativo. Seguidamente, é apresentado o Courseware Ser_e - O Ser Humano e os Recursos Naturais e a descrição da MHDCU, caracterizando as fases, procedimentos e técnicas utilizadas. Para finalizar, é realizada uma breve reflexão em torno da MHDCU e respectivas conclusões.

2. Metodologias de Desenvolvimento de Software Educativo

Os primeiros métodos (designados na literatura como disciplinados, tradicionais ou clássicos) derivaram do processo mais comum do desenvolvimento de *software*. O método em cascata de água ou ciclo de vida de desenvolvimento de *software*, surgiu nos anos 70 e é-lhe atribuído o facto de servir de base teórica para outros métodos, sendo por vezes designado como um método genérico para o desenvolvimento de *software* (Sommerville, 2007). Porém, segundo Larman e Basili (2003), o método iterativo e o método incremental já remontam os anos 50, existindo exemplos concretos de projectos nos anos 70. Nos anos 80 surgiram, entre outros, os métodos em espiral e de prototipagem, e nos anos 90 os métodos ágeis, exemplos reais de integração das abordagens iterativas e incrementais (Abbas, 2006 ; Boehm, 2002 ; Boehm & Turner, 2003,s.d. ; Larman & Basili, 2003 ; Miguel, 2003 ; Paelke & Nebe, 2008 ; Sommerville, 2007). Os métodos ágeis, como o *Extreme Programming* (XP), o *Scrum* e o *Crystal Clear*, através do envolvimento dos utilizadores no processo de

desenvolvimento, procuram fornecer e dar prioridade aos novos requisitos do *software* e avaliar as iterações do mesmo. Dão enfoque ao papel das pessoas, sendo que as competências da equipa de desenvolvimento devem ser reconhecidas e exploradas. Os elementos da equipa são livres para utilizar os seus próprios métodos de trabalho sem serem prescritos processos (Beck, 2000 ; Bergin et al., 2004 ; Keith, 2002 ; Paelke & Nebe, 2008 ; Petersen, 2008 ; Sommerville, 2007).

Especificamente no concerne ao desenvolvimento de *software* educativo, os métodos descritos na literatura têm na sua génese premissas dos métodos supracitados. Exemplo disso, são os 3 métodos utilizados no desenvolvimento dos seguintes pacotes de *software* (ver tabela 2): o Univap Virtual (Bicudo et al., 2007), o Use Case (Castro & Aguiar, 1999) e Softvali (Benitti, Seara, & Schlindwein, 2005).

Tabela 2 – Fases fundamentais no desenvolvimento de *software*

Fases de desenvolvimento (Sommerville, 2007)	Univap Virtual	Use Case	Softvali
Especificação	Análise Planeamento	Preparação	Concepção
Concepção e implementação	Pré-produção	Prototipagem (análise, projecto, implementação, teste e avaliação)	Elaboração construção
Validação	Produção Pós-produção		Finalização
Evolução	-----	Implantação	Viabilização

Apesar de surgirem com designações diferentes, a análise, o *design* e a implementação são fases que se podem identificar nestes processos e que advêm dos primeiros métodos de desenvolvimento. Na fase referente à análise é efectuado o levantamento dos requisitos do *software*, são definidos os objectivos educacionais e o público-alvo a que se destina o *software*. No processo do Univap Virtual, nesta fase é ainda efectuado o levantamento de informação científica sobre a temática do recurso e, no Use Case é realizado um estudo da área educacional do recurso a ser desenvolvido e consequentemente o levantamento dos requisitos necessários, para que este corresponda às necessidades dos utilizadores.

3. O Papel do Design Centrado no Utilizador

O DCU serve para descrever os processos de um projecto em que os utilizadores finais têm influência na forma como este é conduzido. Alguns métodos de DCU sondam os utilizadores sobre as necessidades que estes possuem em determinada área educacional, envolvendo-os em partes específicas do processo de desenvolvimento. Por outro lado, existem métodos em que os utilizadores têm uma maior presença, integrando a equipa, isto é, são envolvidos como elementos durante todo o processo (Abrás, Maloney-Krichmar, & Preece, 2004).

O DCU é descrito na ISO 13407 (1999) - *Human Centered Design Process for Interactive Systems* e na ISO/TR 18529 (2000) – *Ergonomics of Human-System Interaction*. Estas duas normas descrevem uma situação ideal onde não existem quaisquer obstáculos à utilização dos pressupostos DCU, exceptuando a possível falta de competências por parte da equipa de desenvolvimento (Svanaes & Gulliksen, 2008).

Autores como Facer & Williamson (2004), entre outros, reforçam que o DCU é uma metodologia que combina, entre outros aspectos, a participação do utilizador e avaliação formativa de protótipos. De acordo com a norma ISO 13407 (1999), os projectos DCU são regidos por quatro princípios: i) constituição de uma equipa multidisciplinar; ii) a interacção entre o utilizador e o sistema; iii) o envolvimento activo dos utilizadores e; iv) a iteração de soluções de projecto.

Os três métodos de desenvolvimento de *software* educativo supracitados alicerçam-se em pressupostos DCU, entre os quais, a constituição de equipas multidisciplinares, organizadas por profissionais da área da educação (investigadores de psicologia e pedagogia), profissionais da área da informática, especificamente da área de engenharia de *software* e programadores, designers com conhecimentos de usabilidade e por fim os professores e os alunos.

Um outro método que incorpora pressupostos DCU, é o *Logical User-Centred Interactive Design* (LUCID), em que Kreitzberg (1996) identifica seis fases: visionamento, descoberta, fundamentação do projecto, detalhe do projecto, construção e lançamento. Tal como a maioria dos métodos que tem por base princípios DCU, LUCID emprega a prototipagem rápida e o teste de usabilidade iterativa.

Com base no que foi descrito nesta secção, concordamos com o relatório “*Quality Framework for UK Government Website Design: usability issues for government websites*”, quando defende que o DCU é um complemento para os métodos de desenvolvimento de *software*, não sendo um substituto dos mesmos (e-Envoy, 2003).

4. Metodologia de desenvolvimento do Courseware Ser_e: uma abordagem possível para PMEs

O *Courseware Ser_e* – O Ser Humano e os Recursos Naturais é um recurso desenvolvido através de uma parceria entre a Universidade de Aveiro e a Ludomedia – Conteúdos Didácticos e Lúdicos, empresa de desenvolvimento de *software* educativo.

4.1. Apresentação do Courseware Ser_e

O *Courseware Ser_e* integra várias tipologias de *software* (simulações, inquérito, pesquisa,...) com actividades didácticas especificadas em guiões de exploração, tanto para o professor, como para os alunos. Como se depreende a partir dos seus propósitos (promover a compreensão do impacte que a actividade humana tem sobre os recursos naturais e sensibilizar de que o futuro da Humanidade passará pela adopção de atitudes e comportamentos mais conscientes e responsáveis, nomeadamente no que respeita às fontes de energia utilizadas, em particular o petróleo e a floresta), visa uma abordagem à relação entre a actividade humana e a exploração dos recursos naturais, bem como das consequências ambientais, sociais e económicas desta exploração (Sá et al., 2010 ; Sá et al., 2009).

O *courseware* foi pensado para a utilização, em sala de aula, por alunos do 1º e 2º Ciclos do Ensino Básico (preferencialmente a partir dos 8 anos), particularmente dos 3º aos 6º anos de escolaridade, com a orientação dos respectivos professores, embora a sua exploração possa ser adaptada a outros níveis de escolaridade, bem como a outros contextos.

Do conjunto de recursos do *Courseware Sere* fazem parte: um *software* educativo (versão em CD-ROM e *online*, ver em: <http://sere.ludomedia.pt>), os Guiões de Exploração Didáctica para o Professor, os Guiões de Registo para o Aluno/Utilizador e o Manual do Utilizador. No Manual do Utilizador encontram-se informações relacionadas com a navegação nos ecrãs e os ícones utilizados no *software*.

O *software* está dividido em duas fases principais: Fase 1 – Petróleo e Fase 2 – Florestas, não sendo obrigatoriamente sequenciais, isto é, o professor/aluno poderá optar por qual das fases e actividade pretende iniciar a exploração.

Os guiões (figura 1(B)) foram desenvolvidos para servir de base à exploração do *software*. No Guião de Exploração Didáctica - Professor são propostas actividades, estruturadas da seguinte forma: 1) Finalidades da Actividade; 2) Contexto de Exploração; 3) Metodologia de Exploração. Os guiões destinados aos/às alunos(as) são compostos fundamentalmente por folhas de registos.



Figura 1 – (A) Exemplo de ecrã do *Courseware Sere*. (B) Guiões de exploração didáctica.

4.2. Metodologia de Desenvolvimento

Quanto à metodologia de desenvolvimento, a equipa tem procurado dar resposta a questões de investigação relacionadas com a implementação de metodologias de desenvolvimento de *software* educativo centradas no utilizador (Costa, Loureiro, & Reis, 2009b). Factores de qualidade, tais como, a usabilidade, o envolvimento dos utilizadores finais nas diversas fases de desenvolvimento e a constituição de equipas multidisciplinares, são alguns dos pressupostos do DCU em que se baseia a metodologia de desenvolvimento do *Courseware Sere*.

A equipa multidisciplinar foi constituída por elementos com diversas competências ao nível da Didáctica das Ciências (DC), da Tecnologia Educativa (TE), da Gestão de Projectos, do Design Gráfico, da Programação e da Usabilidade.

Tendo em vista reduzir o tempo e custo de desenvolvimento, duas das desvantagens do DCU (Abrás, Maloney-Krichmar, & Preece, 2004), a equipa optou por envolver o utilizador final (professores e alunos) apenas nas tarefas de avaliação do recurso. O recurso (incluindo o *storyboard*) foi também submetido a avaliação por parte de peritos exteriores à equipa (Costa, Loureiro, & Reis, 2010a ; Costa et al., 2009a ; Guerra, 2007), o que se considera incontornável, independentemente da metodologia adoptada.

A figura 2 sintetiza o processo de desenvolvimento do *Courseware Sere*, o qual se descreve de seguida.

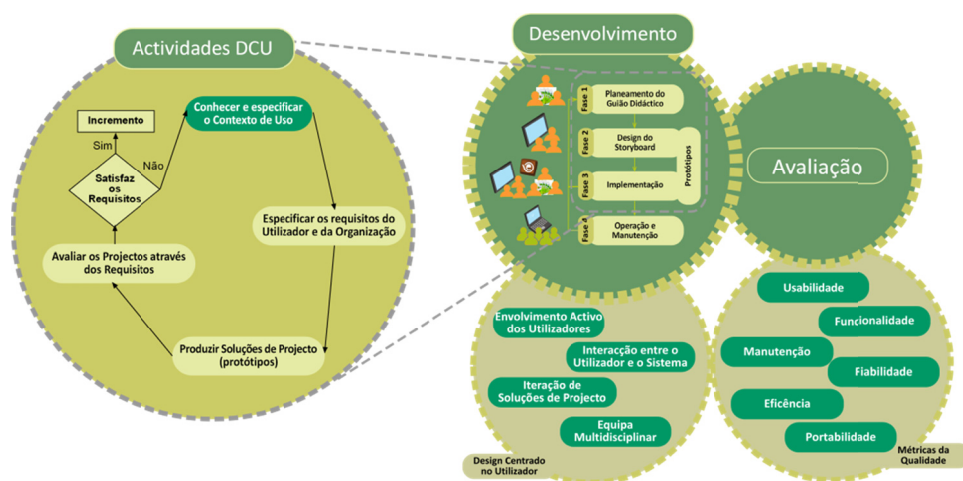


Figura 2 - Metodologia Híbrida de Desenvolvimento Centrado no Utilizador

4.2.1. Fases de Desenvolvimento da MHDCU

Fase 1, Planeamento do guião didáctico: compreendeu a realização de um documento por três peritos em DC e dois em TE com a definição do nível de ensino/público-alvo do recurso, da temática e dos propósitos didácticos, bem como aspectos relacionados com a arquitectura, a navegação e o desenho dos ecrãs do recurso, acima referidos. Esta fase compreendeu ainda o registo de marca e da patente, bem como, entre outros, acordos relativos aos direitos de autoria.

Fase 2, Design do *storyboard*: nesta fase harmonizaram-se as ideias preliminares das actividades didácticas e do conteúdo disciplinar, definidas na fase anterior, com os aspectos de interacção do *software*, particularmente a navegação e interface, com a colaboração de um designer e de um programador da empresa. Como Bassani, Passerino, Pasqualotti & Ritzel (2006) e Carvalho (2003), consideramos que o desenho dos cenários resultantes desta fase foi essencial para se compreender o contexto de utilização do recurso e para representar algumas das situações interactivas do *software*.

Fase 3, Implementação do recurso: esta fase foi dividida em duas subfases que decorreram em simultâneo:

- a parte educacional - requereu a especificação em detalhe de aspectos, para além dos já especificados no *storyboard*, como a animação inicial e os guiões do professor e do aluno.
- a parte técnica - correspondeu ao *design* e programação do *software* e do respectivo manual do utilizador.

Durante esta tarefa, a equipa multidisciplinar testou e ajustou o conteúdo dos guiões à exploração que se pretendia dos ecrãs do *software*. Esta tarefa envolveu a colaboração permanente de todos os elementos, realizada quer presencialmente quer *online*.

Protótipos: os protótipos foram desenvolvidos colaborativamente, entre todos os membros da equipa. Entre outros, a equipa identificou aspectos na interface que tiveram implicações na arquitectura do *software*, que, em alguns casos, levou a alterações nos guiões educacionais do recurso. A prototipagem do *software* foi também usada, no processo de desenvolvimento, de forma a explorar algumas soluções de *software* em particular.

Durante o desenvolvimento do recurso, a equipa recorreu a três tipos de protótipos, como se pode visualizar na figura 3:

- (A) protótipos em papel (*early paper prototypes*);
- (B) ecrãs chave (*key screens*);
- (C) protótipos programados (*running prototypes*).



Figura 3 - (A) Protótipo em papel de um cenário da fase 2 e de uma das personagens. (B) Protótipo do ecrã da escolha das personagens e de um ecrã de uma das actividades. (C) Protótipos programados de dois ecrãs: um da fase 1 - petróleo e um da fase 2 – floresta.

Avaliação: pretendendo-se avaliar tanto o recurso como o seu processo de desenvolvimento, esta fase é transversal a todas as fases acima indicadas. A primeira versão do recurso, também foi alvo de avaliação, por parte de: a) Professores - na avaliação feita por professores, o questionário para avaliação técnica e didáctica do *Courseware Ser*, foi respondido em *workshops* (sessões práticas com a duração máxima de 120 minutos, em que os professores em grupos de dois a três elementos, exploram duas actividades de uma das fases do *courseware*) por parte de um grupo heterogéneo de potenciais utilizadores do recurso (Costa et al., 2009a ; Guerra, 2007); b) Alunos - relativamente à avaliação efectuada pelos alunos, foi respondido um questionário de avaliação técnica e didáctica, após a utilização do recurso em contexto de sala de aula (em blocos de 90 minutos, os alunos em grupos de três a quatro elementos, exploram as actividades do *courseware*, devidamente planificadas pelo professor), tratando-se de uma avaliação controlada (Costa, Loureiro, & Reis, 2010b).

Fase 4, Operação e Manutenção: esta fase incluiu a correcção de erros, técnicos e educacionais, que não foram detectados nas fases iniciais do ciclo de vida do processo de desenvolvimento do *courseware*. Desta forma, é possível melhorar o *software* e implementar novas funcionalidades através de novos requisitos que são detectados durante este processo (Miguel, 2003 ; Sommerville, 2007). Foram tidos em consideração três tipos de manutenção: correctiva, perfectiva e preventiva.

A MHDCU também teve por base alguns princípios dos métodos ágeis, tais como: a) manter a simplicidade, foi desenvolvido o essencial de forma a responder aos requisitos actuais; b) a equipa (essencialmente os programadores) procurou corrigir e melhorar o código do *software* continuamente; c) a entrega foi incremental, sendo que cada ecrã do *software* era independente dos outros ecrãs. Enquanto soluções de projecto eram testadas/validadas/avaliadas, outras eram desenvolvidas com base nos requisitos.

4.2.2. Procedimentos e Técnicas da MHDCU

Para agilizar o processo de desenvolvimento e partindo do princípio que o trabalho colaborativo decorre simultaneamente em dois estados, presencial e *online*, a MHDCU incorpora o Procedimento de Verificação e Validação (PVV), representado no *workflow* da figura 4. Este procedimento surge integrado numa das actividades DCU, Produção de Soluções de Projecto (protótipos), antecedendo a fase de avaliação destas soluções, com o utilizador final e/ou peritos. De suporte a estas actividades é utilizado como *software* colaborativo (*groupware*) a plataforma *Learning Management System* (LMS) Moodle. Apesar, desta plataforma não estar orientada para a gestão de processos desenvolvimento de *software*, a mesma foi essencial para o agilizar da comunicação entre os elementos da equipa multidisciplinar, para disponibilizar documentos, debater ideias, entre outros.

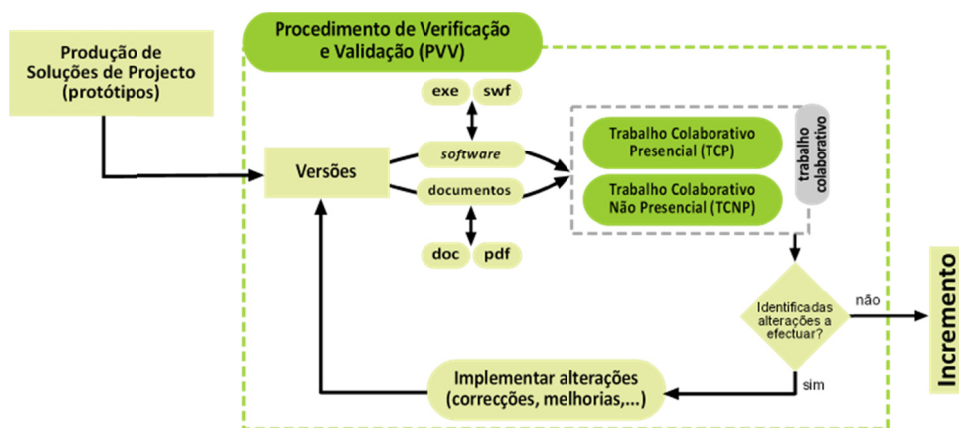


Figura 4 – Procedimento de Verificação e Validação

No PVV, compete aos elementos da equipa multidisciplinar efectuar a verificação e validação das versões do *software*, como das versões dos documentos (guiões do professor e do aluno, manual de utilizador, entre outros). Sendo identificadas alterações a efectuar, é disponibilizada no *moodle* uma nova versão, para verificação e validação. Estas iterações apenas terminam quando não se identificam alterações a efectuar.

No Trabalho Colaborativo Presencial (TCP), comumente, é o gestor de projecto que efectua um primeiro levantamento dos pontos a serem discutidos na reunião presencial. Estes pontos são ordenados por importância e/ou áreas de actuação, sendo enviados previamente para os elementos da equipa multidisciplinar. Para facilitar esta

tarefa é usada uma *mailling list* ou o *fórum* designado como “Notícias e Anúncios”. O TCP é registado através de gravação áudio (ver figura 5).

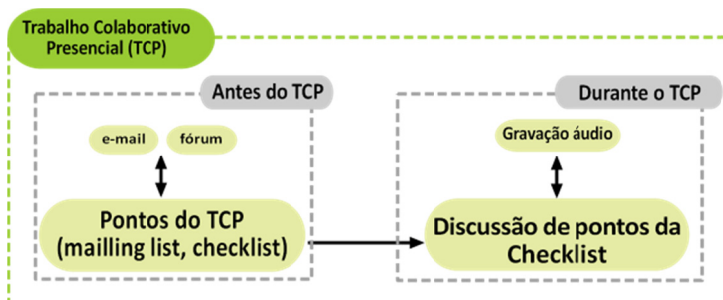


Figura 5 – Trabalho Colaborativo Presencial

Do TCP, ao serem identificadas alterações a efectuar, as mesmas são disponibilizadas na plataforma. Neste contexto, as ferramentas (recursos, módulos de actividades e blocos) utilizadas (ver figura 6) no Trabalho Colaborativo Não Presencial (TCNP), permitem promover e agilizar uma maior interacção entre os elementos da equipa multidisciplinar.

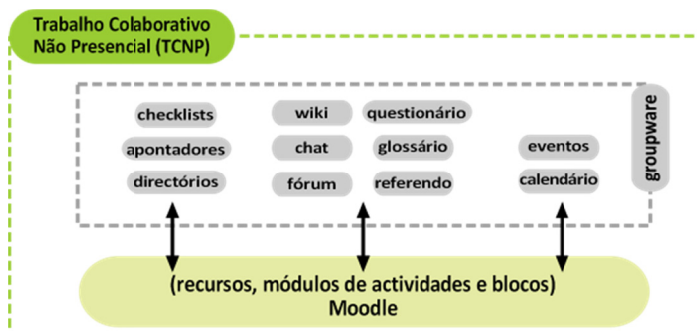


Figura 6 – Trabalho Colaborativo Não Presencial

Seguidamente, passamos a descrever alguns dos módulos de actividades, utilizados no TCNP e concretamente no desenvolvimento do *courseware*:

- referendos: a decisão sobre qual a versão do logótipo que se deveria optar, foi escolhida através de um referendo. Para a marcação de reuniões presenciais, também foi utilizado este módulo de actividades;
- fóruns: todas as versões do *software* e dos documentos (guiões de registo professor e aluno e manual de utilizador, entre outros), foram debatidos através de fóruns;
- glossários: definição de termos científicos e técnicos utilizados, com o intuito de facilitar a comunicação entre os elementos da equipa. O glossário, também serviu para disponibilizar as actas de reunião;

- *chats*: discussão síncrona de determinada questão do projecto, permitiu por vezes, que as tomadas de decisão fossem mais céleres;
- *wikis*: permitiu a construção colaborativa dos textos utilizados no *software*;
- inquéritos por questionário: este módulo foi utilizado durante os workshops de avaliação do recurso, em que os utilizadores no final do *workshop* respondiam a um inquérito por questionário de avaliação. Os resultados ficavam automaticamente disponibilizados para todos os elementos da equipa multidisciplinar.

As ferramentas apresentadas na figura 6, foram disponibilizadas evolutivamente, com o surgimento da necessidade de tornar o trabalho colaborativo mais fácil.

4.3. Reflexões em torno da metodologia explorada

Relativamente à MHDCU, foram detectadas algumas lacunas, que num próximo projecto necessitam de ser colmatadas. Será necessário na fase inicial definir o papel de cada elemento e dar formação na utilização das ferramentas de suporte à gestão de projecto. Além disso e tendo por base o estudo apresentado por Martin Maguire “*Methods to support human-centred design*” (2001), também concluímos que a fase do planeamento deve incorporar duas subfases: 1) contexto de uso e 2) requisitos. Na tabela 3 são identificados alguns métodos que podem ser aplicados em novos projectos.

Tabela 3 – Métodos DCU - Fase 2, Planeamento

Método	Descrição
<i>Identificação e análise de stakeholders</i>	Identificar todos os <u>grupos de utilizadores</u> (utilizadores finais, peritos, responsáveis pela instalação e manutenção) e outras <u>partes interessadas</u> (aqueles que podem ter influência ou são afectados pelo <i>software</i>) incluindo equipas de apoio, equipas comerciais e de marketing e diferentes clientes que poderão adquirir o recurso. Este levantamento pode ser complementado com uma análise de mercado. Nesta fase são atribuídos papéis e responsabilidades. Abordagem ao método: reunião realizada com o gestor de projecto e representantes dos utilizadores para discussão de papéis com maior detalhe.
<i>Análise do contexto de uso (Bevan & Macleod, 1994 ; Kirakowski & Cierlik, 1999 ; Maguire, 1998 ; Thomas & Bevan, 1995)</i>	Caracterização do <u>grupo de utilizadores</u> (competências e experiência, conhecimento da tarefa, formação, qualificações, competências linguísticas, capacidades físicas e cognitivas e atitudes e motivações), das <u>tarefas</u> (lista de tarefas, objectivos, output, passos, frequência, importância, duração e dependências) e do <u>ambiente técnico, físico e organizacional</u> (<i>hardware</i>, <i>software</i>, rede de internet, características das salas, objectivos organizacionais, políticas de utilização das TIC, apoio técnico). Abordagem ao método: Reunião com representantes de cada grupo de utilizadores e representantes da equipa de projecto.
<i>Análise de tarefas e mapeamento de tarefas/funções</i>	Estudo do que utilizador é obrigado a fazer em termos de acções e/ou processos cognitivos para realizar determinada tarefa. Clarificar quais as funcionalidades que são necessárias, de forma a excluir as menos importantes. Abordagem ao método: planear reuniões e eventualmente sessões de observação.
<i>Focus groups</i>	Grupo de discussão de potenciais utilizadores sobre o que pretendem que seja concebido no <i>software</i>. Para evidenciar aspectos dos utilizadores quando os mesmos não podem participar no projecto. Abordagem ao método: presencialmente ou através da plataforma LMS, os utilizadores discutem os requisitos.

5. Conclusões

Embora, a selecção do método dependa do ambiente em que se insere o projecto e de um conjunto de variáveis que, por vezes, são de difícil definição, os métodos existem para tentar auxiliar no desenvolvimento de *software*, minimizando a incerteza, de modo a permitir a obtenção do resultado esperado da forma mais eficiente possível. Tal como Toth (2005) e Sommerville (2007), consideramos que, a adopção do mesmo método para todos os projectos de desenvolvimento de *software*, dificilmente será uma boa escolha, se tivermos em conta a diversidade de utilizadores, objectivo da utilização do *software* e as alterações constantes da tecnologia.

Com base no trabalho desenvolvido, concordamos com Abbas, Gravell & Wills (2008), que designam o desenvolvimento de *software* como uma actividade imprevisível, sendo necessário um método adaptável para controlar esta imprevisibilidade. Quanto ao desenvolvimento de *software* educativo, os processos iterativos e incrementais associados a procedimentos de prototipagem, incluindo ferramentas de avaliação e monitorização nas diferentes fases, são uma forma eficiente de um processo se adaptar à mudança constante de requisitos e da tecnologia (Costa, Loureiro, & Reis, 2009b). Paralelamente, também revemos e estamos de acordo com Abras, Maloney-Krichmar &

Preece (2004), que duas das desvantagens do DCU, são a dos projectos necessitarem de mais tempo para serem desenvolvidos e tornarem-se mais dispendiosos. Porém e segundo Shneiderman & Plaisant, os métodos DCU permitem que o *software* gere menos problemas durante o desenvolvimento e que se reduza os custos na fase de manutenção (2005, p.118), fase esta, que é considerada a mais dispendiosa do ciclo de vida de um *software* (Duim, Andersson, & Sinnema, 2007).

A MHDCU constitui-se como uma solução possível para o desenvolvimento de *software* educativo e foi utilizada para o desenvolvimento do *Courseware* Ser_e, cuja qualidade tem sido reconhecida, nomeadamente num concurso nacional de produtos multimédia, dado ter sido um dos produtos finalistas. Porém, como defendem as normas ISO, a melhoria de um processo de desenvolvimento de *software* deve ser contínua. Estando previsto o desenvolvimento da 3ª fase do *Courseware* Ser_e, relacionado com as energias alternativas, perspectivamos, que a introdução de novos métodos DCU, na fase inicial do desenvolvimento, permitam auxiliar a tomada de decisões e contribuam para a melhoria do processo.

Referências

- Abbas, N. (2006). *Choosing the Appropriate Strategy for a Particular Software Development Project*. Unpublished MSc in Software Engineering, University of Southampton Shouthampton.
- Abbas, N., Gravell, A. M., & Wills, G. B. (2008). Historical Roots of Agile Methods: Where did “Agile Thinking” Come from? . In A. p. a. e. p. i. S. Engineering (Ed.). Limerick, Irlanda.
- Abras, C., Maloney-Krichmar, D., & Preece, J. (2004). User-Centered Design. In S. Publications (Ed.), *Encyclopedia of Human-Computer Interaction*: Thousand Oaks: Sage Publications.
- Bassani, P. S., Passerino, L. M., Pasqualotti, P. R., & Ritzel, M. I. (2006). Em busca de uma proposta metodológica para o desenvolvimento de software educativo colaborativo. *Novas Tecnologias na Educação*, 4(1), 1-10.
- Beck, K. (2000). *Extreme Programming Explained: Embrace Change*: Addison-Wesley.
- Benitti, F. B. V., Seara, E. F. R., & Schlindwein, L. M. (2005). Processo de Desenvolvimento de Software Educacional: proposta e experimentação *CINTED-UFRGS. Novas Tecnologias na Educação*.
- Bergin, J., Caristi, J., Dubinsky, Y., Hazzan, O., & Williams, L. (2004). Teaching Software Development Methods:The Case of Extreme Programming. In ACM (Ed.), *SIGCSE '04*. Norfolk, Virginia.
- Bevan, N., & Macleod, M. (1994). Usability measurement in context. 13, 132-145.
- Bicudo, S. F., Nogueira, T., Oliveira, G. S., Machuca, V. F., Romero, J. P. F., Montenegro, E., et al. (2007). Projecto e Desenvolvimento de Jogos Educativos em 3 Dimensões: a experiência da Univap Virtual.

- Boehm, B. (2002). Get Ready for Agile Methods, with Care. *IEEE Computer*, 64-69.
- Boehm, B., & Turner, R. (2003). *Observations on Balancing Discipline and Agility*: Addison Wesley.
- Boehm, B., & Turner, R. (s.d.). Rebalancing Your Organization's Agility and Discipline.
- Carvalho, C. V. (2003). *Conceitos básicos para o desenvolvimento de cursos multimédia - Manual do Formador (1.ª Edição)*. Porto: Sociedade Portuguesa de Inovação.
- Castro, G. C. M. d., & Aguiar, T. C. d. (1999). Engenharia de Software no Desenvolvimento de Software Educacional Hipermídia, *XXV Conferencia Latinoamericana de Informática*. Asunción-Paraguay.
- Costa, A. P., Loureiro, M. J., & Reis, L. P. (2009b). Development Methodologies for Educational Software: the practical case of Courseware Sere. In E. a. D. International Association of Technology (Ed.), *International Conference on Education and New Learning Technologies (EDULEARN09)* (pp. 5816-5825). Barcelona, Espanha: International Association of Technology, Education and Development (IATED).
- Costa, A. P., Loureiro, M. J., & Reis, L. P. (2010a). Metodologia Híbrida de Desenvolvimento Centrado no Utilizador: o caso prático do Courseware Sere. In A. I. d. S. e. T. d. Informação (Ed.), *5ª Conferência Ibérica de Sistemas e Tecnologias de Informação (CISTI2010)* (pp. 192-197). Santiago de Compostela, Espanha.
- Costa, A. P., Loureiro, M. J., & Reis, L. P. (2010b). Courseware Sere: Avaliação Técnica e Didáctica efectuada por Alunos. In A. I. d. S. e. T. d. Informação (Ed.), *5ª Conferência Ibérica de Sistemas e Tecnologias de Informação (CISTI2010)* (pp. 198-203). Santiago de Compostela, Espanha.
- Costa, A. P., Loureiro, M. J., Reis, L. P., Guerra, C., Sá, P., & Vieira, R. (2009a). Courseware Sere: Technical and Didactic Evaluation. In A. e. a. e. R. In Méndez-Vilas, *Reflections and Innovations in Integrating ICT in Education (Ed.)*, *V Conferência Internacional de Multimédia e TIC na Educação (m-ICTE2009)* (Vol. 1, pp. 502-506). Lisboa.
- Duim, L. v. d., Andersson, J., & Sinnema, M. (2007). Good Practices for Educational Software Engineering Projects, *Proceedings of the 29th international conference on Software Engineering*: IEEE Computer Society.
- e-Envoy, O. o. t. (2003). Quality Framework for UK Government Website Design: Usability issues for government websites
- Facer, K., & Williamson, B. (2004). *Designing educational technologies with users - A handbook from Futurelab*, em http://www.futurelab.org.uk/resources/documents/handbooks/designing_with_users.pdf
- Fowler, M. (2005). *The New Methodology* [online], em <http://www.martinfowler.com/articles/newMethodology.html>
- Guerra, C. (2007). *Avaliação do Storyboard e da Metodologia de Desenvolvimento do Courseware Sere*. Universidade de Aveiro, Aveiro.

- ISO13407. (1999). Human-centred design processes for interactive systems. Geneva: International Standards Organisation.
- ISO/TR18529. (2000). Ergonomics of Human-System Interaction. Geneva: International Standards Organisation.
- Keith, E. R. (2002). Agile Software Development Processes - A Different Approach to Software Design.
- Kirakowski, J., & Cierlik, B. (1999). *Context of Use: Introductory Notes*. Acedido a 10 de Março, 2010, em <http://hfrg.ucc.ie/baseline/filearchive.html#cou>
- Kreitzberg, C. (1996). Managing for usability, in Aber, Antone F. (Editor), *Multimedia: A Management Perspective*, Wadsworth, Belmont, CA, 65-88.
- Larman, C., & Basili, V. R. (2003). Iterative and Incremental Development: A Brief History. *Computer*, 36(6), 47-56.
- Maguire, M. (2001). Methods to support human-centred design. *Internacional Journal of Human-Computer Studies*, 55.4, 587-634.
- Maguire, M. C. (1998). Respect User-Centred Requirements Handbook, *Telematics Applications Project TE 2010: Requirements Engineering and Specification in Telematics* (3.3 ed.): HUSAT Research Institute.
- Miguel, A. (2003). *Gestão de Projectos de Software*: FCA - Editora de Informática.
- Paelke, V., & Nebe, K. (2008). Integrating Agile Methods for Mixed Reality Design Space Exploration, *Proceedings of the 7th ACM Conference on Designing Interactive Systems* (pp. 240-249). Cape Town, South Africa: ACM.
- Petersen, R. R. (2008). *ASAP: Agile Planning in Future Creative Room.*, University of Southern Denmark, Odense.
- Sá, P., Guerra, C., Martins, I. P., Loureiro, M. J., Vieira, R., Costa, A. P., Reis, L. P. (2010, Fevereiro). Desenvolvimento de Recursos Didáticos Informatizados no Âmbito da Educação para o Desenvolvimento Sustentável. O Exemplo do Courseware Sere. *Revista Eureka sobre Enseñanza y Divulgación de las Ciencias*, 7, pp. 330-345.
- Sá, P., Martins, I. P., Guerra, C., Loureiro, M. J., Vieira, R., Costa, A. P., Reis, L. P. (2009). Courseware Sere: Metodologia e finalidades de exploração, *XIII Encontro Nacional de Educação em Ciências - Educação e Formação: Ciência, Cultura e Cidadania (ENEC2009)* (pp. 899-908). Castelo Branco.
- Shneiderman, B., & Plaisant, C. (2005). *Designing the User Interface- Strategies for Effective Human-Computer Interaction* (Fourth ed.): Pearson Education.
- Sommerville. (2007). *Software Engineering* (Eighth Edition ed.): Addison Wesley.
- Svanaes, D., & Gulliksen, J. (2008). *Understanding the Context of Design - Towards Tactical User Centered Design*. Paper presented at the Nordic Conference on Human-Computer Interaction (NordiCHI2008), Lund, Sweden.

Thomas, C., & Bevan, N. (1995). *Usability Context Analysis: A Practical Guide* (4.04 ed.). Teddington, Middlesex, TW11 0LW, UK: National Physical Laboratory.

Toth, K. (2005). Which is the Right Software Process for Your Problem? .

Uma proposta de Arquitetura de Software para Construção e Integração de Ambientes Virtuais de Aprendizagem

Amanda Monteiro Sizo¹, Adriano Del Pino Lino², Eloi Luiz Favero¹

amandasizo@gmail.com, adrianod@ufpa.com, favero@ufpa.br

¹ Universidade Federal do Pará (UFPA) - Programa de Pós-graduação em Ciência da Computação, Caixa postal 479-66075-110, Belém-Pará, Brasil

² Universidade Federal do Oeste do Pará (UFOPA) – Instituto de Engenharia e Geociencias, CEP 68135-110, Santarém - Pará, Brasil

Resumo: Este artigo propõe uma arquitetura de software fundamentada no padrão arquitetural em camadas e demonstra sua aplicabilidade na construção e integração de ambientes virtuais de aprendizagem. Com o surgimento de ambientes cada vez mais modernos, sistemas legados são descontinuados pela dificuldade em mantê-los. A arquitetura proposta facilita a integração de novos módulos a estes sistemas de forma distribuída visando mantê-los competitivos. Esta proposta é implementada a partir da construção de um módulo avaliador e gerador de mapas conceituais que se integra via web service ao ambiente LabSQL. Ao fim são apresentados resultados obtidos com utilização da arquitetura e do módulo de serviço construído.

Palavras-chave: arquitetura de software; ambientes virtuais de aprendizagem; MVC; web service.

Abstract: This paper proposes a software architecture based on standard architectural and demonstrates its applicability in the construction and integration of virtual learning environments. With the emergence of more modern environments, legacy systems are discontinued by the difficulty in maintaining them. The proposed architecture facilitates the integration of new modules for these systems in a distributed way in order to keep them competitive. This proposal is implemented through the construction of a module generator and evaluator of concept maps that integrates environmental LabSQL via web service.

Keywords: software architecture; virtual learning environments, MVC, web service.

1. Introdução

Segundo a definição clássica de arquitetura apresentada por Shaw & Garlan (1996), uma arquitetura de software define o que é sistema em termos de componentes

computacionais e os relacionamentos entre estes componentes. Adicionalmente descreve a estrutura técnica, limitações e características dos componentes, bem como as interfaces entre eles. A arquitetura é o esqueleto do sistema e, por isso, torna-se o plano de mais alto-nível da construção de cada novo sistema (Krafzig, Banke & Slama, 2004).

Ao passo que aumenta a complexidade e abrangência de sistemas de informação, também aumenta a dificuldade em organizar e estruturar seus componentes (Sordi, Marinho & Nagy, 2006). Definir um padrão arquitetural é uma maneira de estimular a reutilização de componentes e padronizar a estrutura através do uso rotineiro de soluções existentes, por meio de um ponto de referência comum para as demais atividades que são executadas posteriormente a sua definição (Bass, Clements & Kazman, 2003). No caso da construção de sistemas educacionais, a arquitetura vem facilitar o desenvolvimento de sistemas que tem como características serem distribuídos e colaborativos, características estas essenciais e presente na maioria dos ambientes virtuais de aprendizagem (Costa et al, 2006).

Uma arquitetura se comporta como facilitador na construção de sistemas que derivam de uma mesma linha de produção (Queiroz & Braga, 2008), neste caso, sistemas educacionais. Para atender as peculiaridades de cada instituição de ensino, são desenvolvidas muitas ferramentas que auxiliam no ensino a distancia, como por exemplo, Teleduc, LabSQL, AulaNet, entre outros, e cada vez surgem ambiente educacionais com funcionalidade cada vez mais modernas. Nesse contexto, os sistemas legados tornam-se obsoletos rapidamente, muitas vezes desconsiderando todo o investimento, seja com pesquisa ou licença empregado na sua adoção.

Assim, este trabalho propõe uma arquitetura de *software* que facilita o desenvolvimento de ambientes de aprendizagem distribuídos que segue o padrão arquitetural em camadas, fundamentada em alguns princípios de GoF (Gamma et al., 1995), como *factory* e *abstract service*, além de padrões como business object, DAO, estabelecidos em (Alur et al., 2003). Também foram integrados nessa arquitetura os *frameworks* Jboss-Seam, EJB3 e Hibernate, que adicionalmente auxilia o trabalho do desenvolvedor e viabiliza o processo proposto neste trabalho.

A arquitetura proposta foi aplicada no desenvolvimento de um módulo que disponibiliza o serviço para geração e avaliação de mapas conceituais utilizando método desenvolvido por (Caldas & Favero, 2009) que compara via n-gramas o mapa conceitual (MC) do estudante com um modelo de resposta formada por vários mapas conceituais. Este módulo se integra ao ambiente LabSQL (Laboratório para ensino e aprendizagem de SQL), a partir de *web service* e está sendo utilizado por doze turmas de graduação e pós-graduação em diferentes áreas de atuação.

Além desta seção introdutória, este artigo está organizado como segue: na seção 2 são apresentados os trabalhos correlatos; na seção 3 é apresentada a arquitetura proposta; na seção 4 é apresentado a aplicação da arquitetura com um estudo de caso; na seção 5 é apresentado os resultados com a utilização dessa proposta e finalmente na seção 6 são realizadas as considerações finais.

2. Trabalhos correlatos

No campo educacional, tem-se o sistema AulaNet (Gerosa et al., 2004) que é um ambiente para a criação, aplicação e administração de cursos baseados na Web. Este sistema utiliza uma arquitetura implementada com técnicas de desenvolvimento baseado em componentes o que provê uma certa flexibilidade no acoplamento de funcionalidades porém precisa que as mesmas se adéquem às restrições da plataforma J2EE.

A versão 2.0 do AulaNet utiliza uma arquitetura cliente-servidor na *World Wide Web* baseada em *servlets*, sua limitação é acessar os componentes apenas desenvolvidos em Java.

A mesma limitação acontece no Moodle, que fornece uma API – *Application Programming Interface* que implementa uma infra-estrutura de *web services* que permiti abrir o núcleo do Moodle à sistemas externos, porem limita o desenvolvedor a construir módulos em linguagem php, de acordo com seu guia próprio.

Na literatura existem vários artigos sobre as arquiteturas orientadas à serviços, nomeadamente sobre *web services* e a sua utilização na integração de sistemas (Moura & Bernardino, 2010). Essas arquiteturas possuem um problema em comum, a falta de recursos para o desenvolvimento de novos serviços, que implica diretamente na escassez de novas funcionalidades em diferentes linguagens e plataformas.

Diante desse contexto, a motivação para o desenvolvimento do trabalho é propor uma arquitetura de software simples que desenvolve eficientemente módulos de serviços de maneira interoperável, isto é, independente de plataforma ou linguagem e os integram à ambientes virtuais de aprendizado, podendo os mesmos serem reutilizados por diferentes ambientes virtuais evitando reescrita de serviços previamente concebidos.

3. Arquitetura proposta

A solução proposta apresenta uma infra-estrutura no apoio à otimização de ambientes de educação à distância através do desenvolvimento e integração de novas funcionalidades, a fim de mantê-los competitivos e evitar que esses ambientes entrem em desuso por conta do surgimento de ambientes mais modernos. Acarretando assim perda no investimento financeiro aplicado na adoção desses ambientes que por não contemplarem certas funcionalidades são descartados periodicamente.

A arquitetura aqui definida para a construção de objetos educacionais utiliza o padrão arquitetural MVC (*Model-View-Controller*), que, por definição, desacopla as camadas provendo maior flexibilidade. Esta arquitetura de software tem como principal objetivo desenvolver sistemas e integrá-los de maneira distribuída a outros ambientes independe de plataforma ou linguagem.

Alguns *frameworks* abertos foram adotados como padrões na arquitetura, como o Jboss-Seam, EJB3 e Hibernate, a reutilização de seus componentes proporcionou aumento na produtividade. A seguir, são explicadas essas tecnologias.

1. EJB3: Permite que os objetos chamados de *enterprise beans* sejam expostos como serviços *web*, de modo que seus métodos possam ser invocados por

- outro aplicativo J2EE, e também por aplicativos escritos por outras linguagens de programação em diferentes plataformas (Burke & Moson-Haefel, 2007).
2. Jboss-Seam: unifica o modelo de componentes do JSF e do EJB3, além de eliminar o código de integração, o que permite que o desenvolvedor dispense mais atenção às regras de negócio (Seam, 2010).
 3. Hibernate: é responsável por mapear objetos Java em tabelas de um banco de dados relacional, estreitando o *gap* conceitual que existe entre os dois, deixando o desenvolvedor livre para se concentrar no negócio da aplicação (Bauer & King, 2007).

A arquitetura e a integração com *frameworks* citados são mostrados na figura 1, através de um diagrama de componentes.

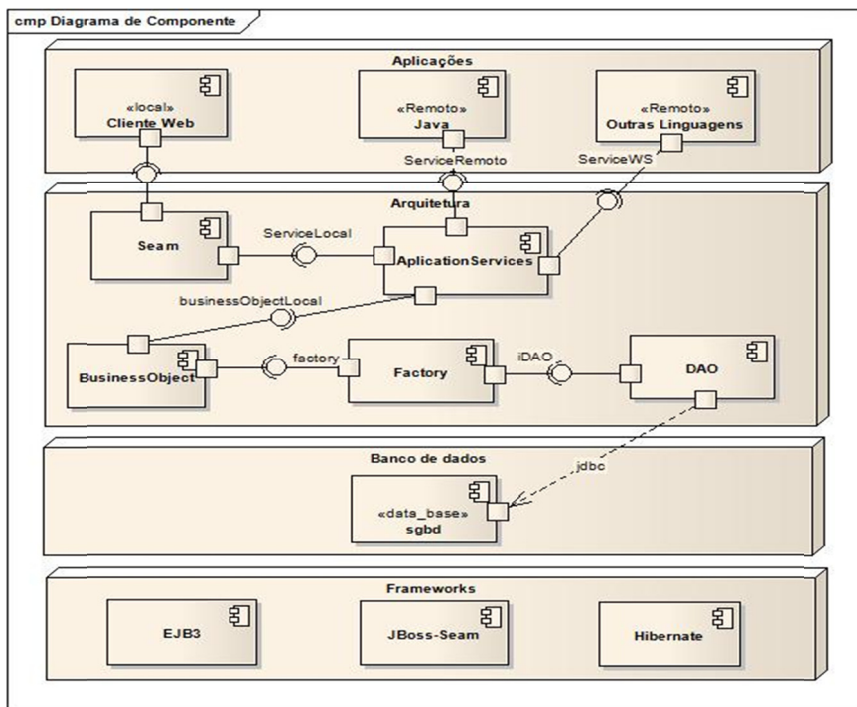


Figura 1 - Arquitetura integrada aos frameworks

3.1. Application Service

A camada *Application service* (Alur et al., 2003) é um padrão J2EE que centraliza a lógica de negócios de diversos componentes da camada em diversos grupos ou pontos de acesso. É a porta de entrada para efetuar uma transação. Funciona como ouvinte da camada de apresentação, ou seja, toda ação efetuada pelo usuário do sistema será “ouvida” pelo *Application service* e repassada para o *Business Object* executar.

A descentralização de acesso do *Application service*, permite que o cliente tenha acesso apenas no domínio de seu interesse, isolando assim o acesso ao restante da aplicação,

resultando em segurança e alta performance, uma vez que só é compilado o grupo de métodos referente ao serviço acessado. Diferentemente do padrão utilizado na maioria das arquiteturas, o *Facade* que sobrecarrega o processamento através de um único ponto de acesso.

3.2. Business Object

A camada *business object* é responsável por implementar a camada de lógica de negócios da aplicação e fornece os serviços para o objeto de negócio específico (Alur et al., 2003). As classes encontradas neste pacote são responsáveis por executar as transações que a aplicação cliente necessita e instanciar os componentes de persistência.

Esta camada utiliza tanto seus próprios métodos quanto os métodos disponibilizados por outros *business objects* permitindo o reuso dos serviços implementados.

3.3. Factory

Esta camada é responsável por criar objetos que devem obedecer a certos critérios dentro de uma lógica de criação complexa (Larman, 2007). Este padrão permite que a criação dos objetos DAO sejam feitos de forma transparente. Uma das vantagens de utilizar esse padrão é que ela indica qual implementação de persistência (DAO) será utilizada pela aplicação a partir de um único ponto de acesso, o que diminui o retrabalho no caso de manutenção da aplicação.

3.4. DAO

O *Data Access Object* (DAO) é um padrão conhecido na literatura como responsável por intermediar o acesso aos dados armazenados em um banco de dados (Fowler, 2006).

Cada DAO deve possuir uma interface, que especifica os métodos de manipulação de dados. Os serviços acessam apenas as interfaces dos DAOs, desconhecendo a implementação utilizada.

De uma maneira geral, objetos educacionais desenvolvidos sobre a arquitetura proposta apresentam alto desempenho, interoperabilidade além de reusabilidade, isso se deve principalmente aos seguintes fatores: descentralização no acesso, agrupamento de serviços por domínio da aplicação, e reutilização de implementação dos DAOs.

4. Aplicação da Arquitetura

Com a difusão dos ambientes virtuais de aprendizagem é cada dia mais relevante o estudo e desenvolvimento de ferramentas que possam avaliar os aprendizes à medida que diferentes formas de ensino são difundidas. Entende-se ser necessário saber tirar proveito das possibilidades tecnológicas que emergem com esses novos espaços pedagógicos a fim de otimizar os ambientes através da inclusão de novas abordagens (Behar & Leite, 2005).

Mapas Conceituais têm a sua origem no movimento da teoria construtivista da aprendizagem, de Ausubel (1980) e são utilizados como ferramenta de apoio à

representação de conhecimento, com um vasto relato de aplicações em educação, notadamente no apoio à avaliação da aprendizagem (Araújo, Menezes & Cury, 2002).

O LabSQL é uma ferramenta de ensino e aprendizagem da linguagem SQL e sua proposta inicial está centrada na avaliação automática de consultas SQL (Lino et al., 2007). Atualmente, o LabSQL agrega funcionalidades que são resultados de várias pesquisas.

Dando continuidade às pesquisas que tem como fim maximizar a aprendizagem dos estudantes, pretende-se agregar os benefícios pedagógicos alcançados por meio do ensino a partir da utilização de mapas conceituais, a fim de detectar através do mapeamento dos conceitos chaves a deficiência dos estudantes na área de banco de dados.

4.1. Estudo de Caso

Para compor o estudo de caso desta pesquisa, foi desenvolvido um módulo de serviço para geração e avaliação de mapas conceituais, que são implementações de *webservices* disponibilizadas para o LabSQL conforme mostra a figura 2. As telas aqui mostradas fazem parte do LabSQL e acessam o serviço disponibilizado pelo módulo proposto.

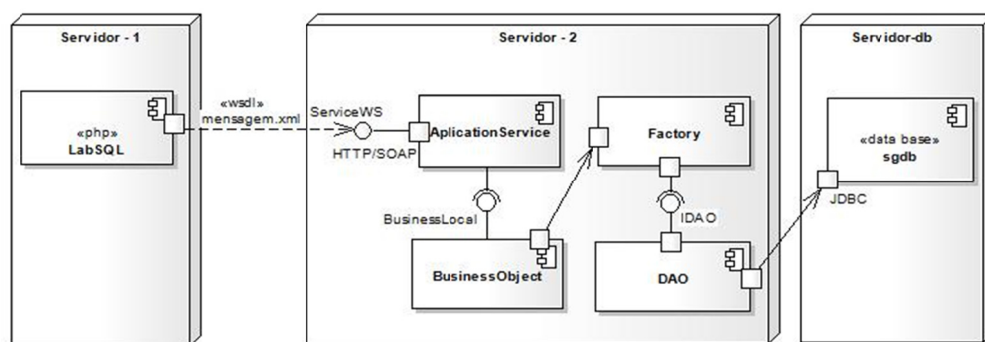


Figura 2 - Diagrama de implantação - Integração entre as aplicações via web service

A primeira fase para construir um projeto baseado na arquitetura, é definir as entidades decorrentes do modelo de negócio. Essas entidades compõem a camada *entity* da arquitetura, isto é, entidades do domínio da aplicação também conhecidas como POJO (*Plain Old Java Objects*) Para o projeto de estudo de caso foi definido o modelo do domínio do negócio como mostrado na figura 3 a seguir:

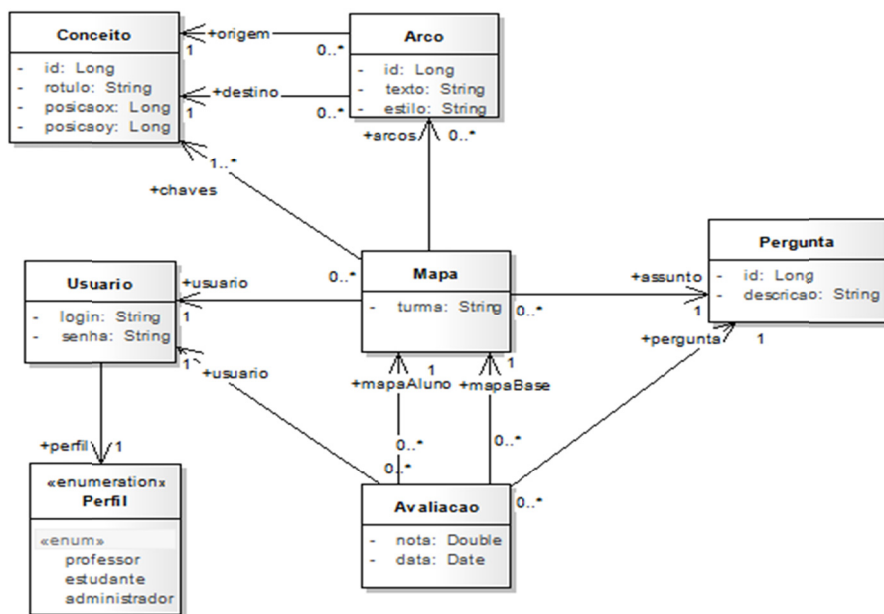


Figura 3 - Diagrama de classes – Modelo de Negócios

O módulo para geração e avaliação de mapas conceituais possui como requisitos principais: a) manter mapa conceitual: Esta funcionalidade permite que usuários com perfil de professor ou estudante gerem mapas conceituais, porém só o professor pode cadastrar seu mapa como modelo, isto é, o mapa que serve de referência na correção dos mapas dos estudantes e b) corrigir mapa conceitual: Esta funcionalidade compara o MC do estudante com o MC base previamente cadastrado pelo professor, e compara com os mapas conceituais dos demais estudantes gerando uma nota de acordo com seu número de acertos. Caso o aluno acrescente um conceito inválido o mesmo é diferenciado.

O componente desenvolvido tem como propósito principal a avaliação automática de MCs a partir da análise de similaridade dos MCs dos estudantes contra um modelo de resposta do professor, retornando um escore da avaliação quantitativa. Este resultado é obtido através da análise via bigrama da formação conceito-link ou link-conceito e análise via trigramma da formação dos arcos conceito-link-conceito ou link-conceito-link (Caldas & Favero, 2009).

A figura 4 mostra a tela de resultado da avaliação de mapas conceituais produzida pelo estudante, como pode ser observado, foi gerada uma nota para questão 217 que define o conceito de banco de dados relacional, sendo que além de apresentar o histórico de notas para a questão o sistema também aponta o conceito que não faz parte da definição.

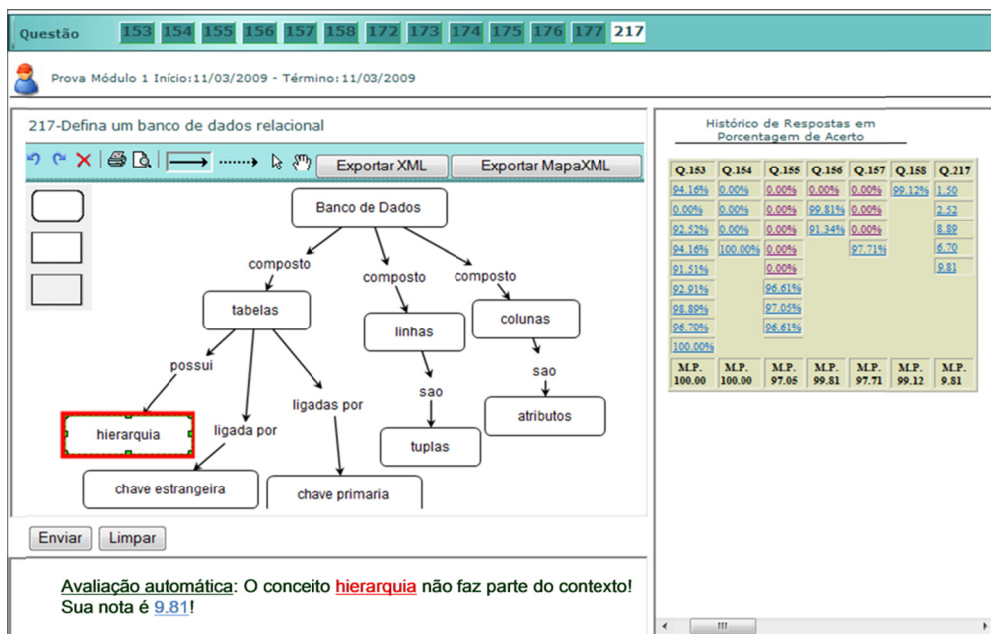


Figura 4 - Resultado da avaliação do mapa conceitual do aluno

Os serviços como, cadastrar mapas conceituais, avaliar mapas conceituais entre outros do módulo citado, são disponibilizados por uma interface webservice (ServiceWS), essa interface é acessada pelo LabSQL que consegue assim fornecer esses serviços a partir de sua aplicação a seus usuários finais.

A Figura 5 ilustra o fluxo de execução para a funcionalidade cadastrar mapas, observe que cada camada possui uma interface (ServiceWS, BusinessMapaLocal e IDAOMapa) que define quais serviços estão disponibilizados e sua respectiva implementação (ApplicationService, BusinessMapaImpl, IDAOMapaImpl) com exceção da camada *factory* que já disponibiliza seus métodos públicos na própria classe.

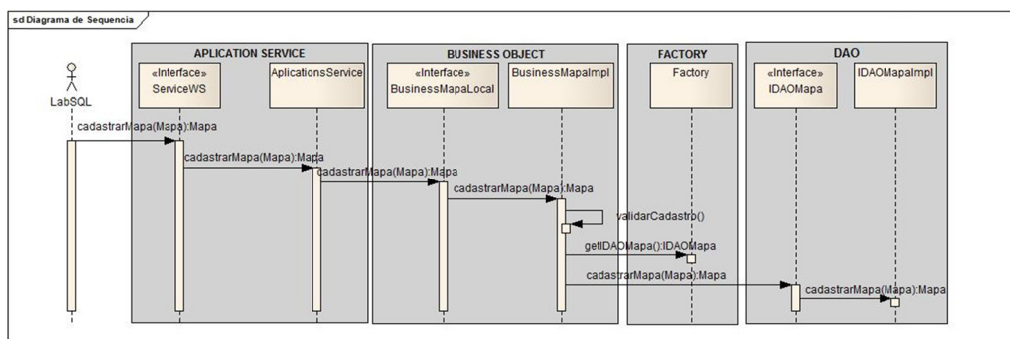


Figura 5 - Diagrama de sequência - Método cadastrar mapa

Esta arquitetura além de mais simples adiciona performance no tratamento de seus objetos, isso se deve aos seguintes fatores:

1. para cada funcionalidade disponibilizada existe um *Application Service*, evitando a centralização em um único ponto de acesso e em consequência sua sobrecarga, resultando na diminuição no tempo de resposta no processamento das operações;
2. inclusão da camada de *business object* que além de ser responsáveis pelas regras de negocio, permite que os serviços implementados em um BO sejam reutilizados em outros BO permitindo um relacionamento cíclico entre os mesmos, evitando assim reescrita de código de uma mesma função em BOs distintos;
3. inclusão da camada *factory* que permite a partir de um único ponto indicar qual DAO será utilizado na aplicação, isso aumenta a produtividade na fase de desenvolvimento e manutenção. Em outras arquiteturas se houver mudanças na implementação no DAO a ser utilizado, o desenvolvedor teria que apontar em cada BO que utiliza o DAO a nova implementação, ocasionando aumento de retrabalho. Esta camada também possui como vantagem a reutilização da instância do DAO que esta na memória auxiliando na melhoria do desempenho.

5. Resultados Obtidos

5.1. Quanto à arquitetura

A definição de padrões de qualidade é importante na elaboração de produtos de software, notadamente, na construção de componentes (Simão & Belchior, 2002). Os padrões são usados como parâmetros, para dar mais atenção aos aspectos que levem os componentes a níveis de qualidade esperados, com maior rapidez e menor custo. Alguns trabalhos abordam mais diretamente as características de qualidade para componentes de software (Preiss et al., 2001; Bertoa, 2002).

Entretanto, ainda se faz necessário um conjunto de características específicas, que possa ser usado para avaliar vários aspectos que definem uma arquitetura com qualidade. As características aqui avaliadas foram organizadas segundo a norma ISO/IEC 9126 (ISO, 2001), que define requisitos de qualidade para componentes de software, além de ser um padrão estabelecido internacionalmente.

A Tabela 1 apresenta o conjunto das características de qualidade organizado segundo o modelo de qualidade da ISO/IEC 9126, e os principais fatores que evidenciam o atendimento desse requisito na arquitetura proposta.

Tabela 1 - Requisitos de qualidade atendidos

Característica	Definição	Atendimento
Funcionalidade	Característica do componente prover serviços (funções) que satisfaçam as necessidades especificadas, quando o componente for usado sob condições específicas (ISO, 2001)	A arquitetura desenvolveu o módulo de serviço satisfazendo as especificações determinadas pelo domínio do negocio de mapas conceituais.
Confiabilidade	Característica do componente que o faz manter em um determinado nível de desempenho, quando usado sob condições especificadas (ISO, 2001).	A camada de application Service descentralizou o acesso, diminuindo o tempo de resposta para o usuário.
Usabilidade	Característica do componente ser entendido, aprendido, usado e atrativo para o usuário, quando usado sob determinadas condições (ISO, 2001).	As camadas são fracamente acopladas e fornecem uma maneira eficiente e padronizado de desenvolvimento ao arquiteto de software.
Eficiência	Característica do componente realizar suas funções sem desperdício de recursos, sob condições especificadas (ISO, 2001).	A camada Business utiliza tanto seus próprios métodos quanto os métodos disponibilizados por outros business objects permitindo o reuso dos serviços implementados.
Manutenibilidade	Característica do componente ser modificado. Modificações podem incluir correção, melhorias ou adaptação do componente para mudanças em ambiente, e em requisitos e especificações funcionais (ISO, 2001).	A camada Factory indica qual implementação de persistência (DAO) será utilizada pela aplicação a partir de um único ponto de acesso, o que diminui o retrabalho no caso de manutenção da aplicação.
Portabilidade	Característica do componente ser transferido de um ambiente para outro. O ambiente pode ser uma organização, um hardware, um ambiente de software ou uma aplicação. (ISO, 2001)	As tecnologias e frameworks especificados proporcionam a construção de serviços independentes de plataforma ou linguagem.

Importante salientar que, na análise apresentada não pretende-se contemplar os requisitos da ISO/IEC 9126 em sua totalidade, pois uma característica pode ser observada em vários aspectos da arquitetura e por uma questão de organização, uma determinada característica é definida somente uma única vez, onde se observa que a sua maior influência é exercida.

5.2. Quanto à aprendizagem

A versão do LabSQL que contém a funcionalidade dos Mapas Conceituais foi utilizada por várias turmas diferentes e os resultados obtidos até o momento são satisfatórios. Essa versão foi testada por 12 (doze) turmas, nos níveis de graduação e pós-graduação da UFPA, durante o período de 4 (quatro) semestres letivos. A partir dos dados obtidos destacamos as seguintes informações: melhora no desempenho, aumento na execução de exemplos e aumento de submissões de respostas em exercícios.

Confrontando o histórico de notas das turmas sem MC no período de 2004 até 2007 contra o histórico de 2008 até 2009, no nível de graduação, observou-se que, nas turmas tradicionais sem a utilização do MC, o desempenho é menor do que nas turmas com a utilização do MC. O aspecto positivo da utilização de MC quanto aos resultados

pode ser observado por meio do acréscimo de 13% para 27% no percentual de alunos com conceito de bom a excelente (Figura 6 - A).

Quanto à reutilização do sistema ocorre quando um aprendiz utiliza o ambiente mais de uma vez a fim de treinar ou exercitar exemplos práticos que estão no LabSQL, essa medida é extraída automaticamente do ambiente pode ser utilizada como complemento na avaliação do aprendizado do estudante (Lobato et al., 2008). Em Silva et al. (2008) o LabSQL não possuía o MC, e em sua pesquisa foi relatada que o número de exemplos realizados no sistema era de 1.706 submissões, contra um total de 2.527 submissões na versão atual que contém o MC, conforme gráfico (Figura 6 - B).

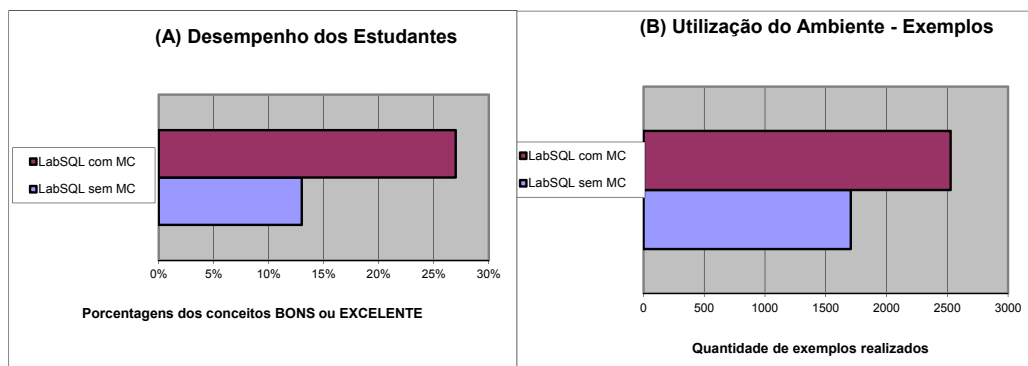


Figura 6 - Gráficos de Desempenho e Reutilização do sistema

Além disso, uma consulta no banco de dados do histórico do perfil dos aprendizes revelou que ocorreu um acréscimo de utilização dos módulos de treinamentos e execução de exercícios. Acreditamos que o MC trouxe o benefício da reflexão do aprendiz e na melhora da qualidade das respostas submetido pelo aprendiz, isso pode ser constatado por meio de uma consulta que confronta os conceitos finais dos aprendizes nos períodos informados. Em média foram acrescidos 12% no conceito final dos estudantes e uma redução de 6% no índice de reprovação. A Figura 7, ilustra que ocorreu uma redução nesses índices de reprovação, que comprovam que o LabSQL com MC aumentou sua usabilidade.

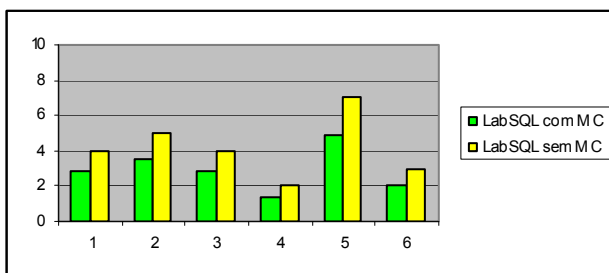


Figura 7- Gráfico de índice de redução de reprovação

6. Conclusão

Em termos gerais, este artigo apresentou uma arquitetura que integra aplicações distribuídas de maneira eficiente além de potencializar o processamento durante a execução da aplicação. Também contemplou requisitos, que de acordo com (Sordi , Marinho & Nagy, 2006) fazem parte de uma boa arquitetura, como: desempenho, reusabilidade, manutenibilidade, entre outros.

O grande diferencial desta arquitetura é o aumento de desempenho e reusabilidade proporcionada por meio do acesso descentralizado aos serviços na *camada business object*. Também prove a interoperabilidade que desempenha um papel essencial em qualquer infra-estrutura onde os atores e suas aplicações são distribuídos e heterogêneos.

Como principal contribuição dessa arquitetura tem-se o acoplamento eficientemente de serviços independente de plataforma ou linguagem em qualquer ambiente virtual de aprendizagem, otimizando os mesmos. Além disso, a possibilidade de inserção de novas funcionalidades aos ambientes trouxe flexibilidade, ou seja, pode-se criar se desejável, módulos dentro do próprio ambiente que se integram a outros ambientes virtuais ou sistemas administrativos, requisito este muito desejável em qualquer ambiente educacional.

A proposta foi viabilizada por meio do desenvolvimento de um componente que gera e avalia mapas conceituais que se integra ao ambiente LabSQL através de webservices.

Como perspectiva futura, pretende-se estender o experimento à integração de novas funcionalidades a outros ambientes virtuais de aprendizagem, também se pode agregar novos recursos a fim de contemplar funcionalidades administrativas ao ambiente como administrar notas, faltas, cadastro de usuários etc.

Referências

- Alur, D.; Malks, D.; & Crupli, J.(2003). Core J2EE Patterns: Best Practices and Design Strategies, *Second Edition*. Pearson.
- Araújo, A. M.T., Menezes, C. S. & Cury, D. (2002). Um Ambiente Integrado para Apoiar a Avaliação da Aprendizagem Baseado em Mapas Conceituais. *Anais do XIII SBIE*, São Leopoldo, RS.
- Ausubel, D. (1980). Psicologia Educacional. - 2. Ed. Interamericana.
- Bass, Len; Clements, P., Kazman, R. (2003). Software Architecture in Practice. Addison-Wesley, 2. ed.
- Bauer, C. & King G. (2007). Java Persistence com Hibernate. *Ciência Moderna*.
- Behar, P.A. & Leite, S.M. (2005). Criando novos espaços pedagógicos na internet: O ambiente ROODA. *Revista de Gestão da Tecnologia e Sistemas de Informação*.
- Bertoa, M. F., Troya, J.M. & Vallecillo, A. (2002). Aspectos de la Calidad en el Desarrollo de Software Basado em Componentes, *Capítulo do livro: Calidad en el desarrollo y mantenimiento del software*.

- Burke, B. & Moson-Haefel, R. (2007). Enterprise javaBeans 3.0. 5th. *Peason*.
- Caldas, V.M. & Favero, E.L. (2009). Uma Proposta de Avaliação Automática de Mapas Conceituais para Ambientes de Ensino a Distância”, *XXXV Conferencia Latinoamericana de Informática – Pelotas – RS*.
- Costa, K. C. F., Harb, M. D. P. A., Brito, S. R. & Favero, E. L.(2006). Um sistema de recomendação para ambientes virtuais de aprendizagem baseado em agentes e componentes de software. *32º Conferencia LatinoAmericana de Informática- Santiago – Chile*.
- Fowler, M. (2006). Padrões de Arquitetura de Aplicações Corporativas. *Bookman*.
- Gamma, E., Helm, R., Johnson, R. & Vlissides, J.(1995). Design patterns: Elements of reusable object oriented software. *Reading*,
- Gerosa, M. A., Raposo, A.B., Fulks, H. & Lucena, C.J.P. (2004).Uma arquitetura para o Desenvolvimento de Ferramentas Colaborativas para o Ambiente de Aprendizagem AulaNet, XV Simpósio Brasileiro de Informática na Educação
- ISO/IEC 9126-1.(2001). Information Technology . Software Product Quality. part 1.
- Krafzig, D., Banke, K. & Slama, D. (2004).Enterprise SOA: Service-Oriented Architecture Best Practices. Indianapolis: Prentice Hall.
- Larman, C. (2007). Utilizando UML e Padrões: uma introdução à análise e ao projeto orientado a objetos. 3ª edição, Porto Alegre: Bookman.
- Lino, A. D. P., Silva, A. S., Favero, E. L. & Harb, M. D. P. A. (2007). Avaliação automática de consultas SQL em ambiente virtual de ensino-aprendizagem. *2ª Conferência Ibérica de Sistemas e Tecnologias de Informação*, v.1. p.89 – 100.
- Lobato, A. S., Harb, M. D. P. A., Lino, A. D. P., Elói L. F., Silva, H. A. N & Santos, T. L. T. (2008). Aplicando Rubrica para Avaliar Qualitativamente o Estudante no LabSQL. *Conferencia LatinoAmericana de Informática* , Santa Fe.
- Moura, R. & Bernadino J. (2010). Um modelo para a integração de serviços – Moodle e Sistemas de Gestão Acadêmica. *Revista Ibérica de Sistemas de Tecnologia da Informação*, N° 5, pag. 31.
- Preiss, O., Wegmann, A. & Wong, J. (2001). On Quality Attribute Based Software Engineering. *27th Euromicro Conference*. Warsaw, Poland, September.
- Queiroz, P.G. & Braga, R.T.V. (2008). Desenvolvimento de Linhas de Produtos de Software com uma Arquitetura Orientada a Serviços. *13º Workshop de Teses e Dissertações em Engenharia de Software*.
- Seam. (2010). Sitio oficial do framework seam. Disponível em: <http://seamframework.org/Documentation/SeamDocumentation#HSeamReferenceDocumentation>, Acesso: 21 out 2010.
- Shaw, M. & Garlan, D.(1996). Software Architecture. Perspectives on an Emerging Discipline. *Prentice Hall*.

- Silva, H. A. N., Lino, A. D. P., Silveira, A. M., Favero, E. L., Santos, T. L. T. & Moraes, R. F. (2008). Um sistema baseado na lógica difusa para decidir os conceitos finais dos estudantes críticos. *Workshop sobre Educação em Computação* | SBC - Belém. p.1 – 10
- Simão, R.P.S. & Belchior, A.D.(2002). Um Padrão de Qualidade para Componentes de Software. *I Simpósio Brasileiro em Qualidade de Software*. Gramado, RS.
- Sordi, J.O., Marinho, B.L. & Nagy, M.(2006). Benefícios da Arquitetura de Software Orientada a Serviços para as Empresas: Análise da Experiência do ABN AMRO Brasil. *Revista de Gestão da Tecnologia e Sistemas de Informação*.

M2DAT-HYMO: una herramienta basada en MDA para la generación automática de aplicaciones Web a partir del modelo del hipertexto

Feliu Trias Nicolau, Iván Santiago Viñambres, Juan Manuel Vara, Valeria de Castro

feliu.trias@urjc.es, ivan.santiago@urjc.es, juanmanuel.vara@urjc.es,
valeria.decastro@urjc.es

Grupo de Investigación Kybele, Departamento de Lenguajes y Sistemas Informáticos II, Universidad Rey Juan Carlos, Avda. Tulipán s/n, 28933, Móstoles (Madrid)

Resumen: Una premisa básica exigible a cualquier propuesta metodológica para el desarrollo de software es la de ofrecer algún tipo de herramienta de apoyo. En el caso de propuestas basadas en los principios de la Ingeniería Dirigida por Modelos (MDE) esta premisa se convierte en un requisito imprescindible. Este trabajo presenta la herramienta que soporta HM³, el método para el modelado del hipertexto del marco metodológico MIDAS. Así, el artículo presenta la definición y codificación de las transformaciones automáticas entre los modelos que componen HM³, y las transformaciones modelo a texto que soportan la generación automática de código. En este sentido, uno de los aspectos más innovadores de este trabajo es la utilización de modelos weaving para personalizar algunas de las transformaciones entre modelos que propone HM³.

Palabras clave: MDA, MDE, Transformaciones de Modelos, Modelos Weaving, Generación de Código

Abstract: Nowadays, any software development methodological proposal must be accompanied by the corresponding technological support. Such premise becomes even more relevant when the proposal is based on Model Driven Engineering principles. Therefore, this proposal presents the tool support for HM³, the Hypertext Modelling Method of MIDAS. In particular, it presents the specification and coding of the model-to-model transformations that bridge the different models that compose HM³, as well as the model-to-text transformations that support code generation. One of the most outstanding aspects of this proposal is the use of weaving models to personalize the model-to-model transformations proposed by HM³.

Keywords: MDA, MDE, Model Transformations, Weaving Models, Code Generation

1. Introducción

Los Sistemas de Información Web (SIW) se han convertido en los últimos años en una herramienta clave, no sólo para el funcionamiento diario, sino también para la expansión y crecimiento de cualquier organización: la mayor parte de su actividad, comercial y de gestión se realiza a través de Internet y mediante estos sistemas (Lowe, 2003). Por este motivo, en la actualidad adquieren especial relevancia las técnicas y metodologías que facilitan la construcción de este tipo de sistemas.

Por otra parte, uno de los paradigmas de desarrollo software más importante en la actualidad es la Ingeniería Dirigida por Modelos (MDE, *Model Driven Engineering*) (Selic, 2001; Schmidt, 2006) cuyo principio fundamental es trasladar el foco de atención de las actividades de codificación a las actividades de modelado. De este modo, los modelos pasan a ser los artefactos principales en el proceso de desarrollo.

Dentro de este nuevo paradigma surge MIDAS (Cáceres et al., 2006; Marcos, et al., 2006; De Castro, 2007), una metodología dirigida por modelos para el desarrollo de aplicaciones orientadas a servicios (De Castro et al., 2009) que considera los tres aspectos fundamentales asociados con la clásica arquitectura de tres capas: contenido, hipertexto y comportamiento. Como parte de la propuesta de MIDAS se ha definido un método específico para el desarrollo de cada capa. En particular, para la capa del hipertexto MIDAS propone un Método para el Modelado del Hipertexto (HM³, *Hypertext Modelling Method of MIDAS*). Este método propone un conjunto de modelos a distintos niveles de abstracción y las reglas de transformación entre ellos. En particular se adopta la separación en niveles de abstracción propuesta por la Arquitectura Dirigida por Modelos (MDA, *Model Driven Architecture*) (Selic, 2001; Miller & Mukerji, 2003) que distingue entre Modelos Independientes de Computación (CIMs, *Computer Independent Models*), Modelos Independientes de Plataforma (PIMs, *Platform Independent Models*) y Modelos Específicos de Plataforma (PSMs, *Platform Specific Models*).

El soporte tecnológico para la propuesta de MIDAS es M2DAT (MIDAS MDA Tool) un entorno extensible y modular, de forma que las herramientas de soporte para cada propuesta metodológica comprendida en MIDAS se desarrollan como un nuevo módulo que se integra con el resto de la herramienta (Vara, 2009). Este artículo se centra en M2DAT-HYMO (M2DAT for *HY*per*text* *MO*delling), el módulo de M2DAT que da soporte a HM³. En particular se presentan los editores gráficos para los diferentes modelos propuestos en HM³; las transformaciones modelo-a-modelo (M2M) para conectar dichos modelos y las transformaciones modelo-a-texto (M2T) para generar el código que implementa el sistema a partir de dichos modelos.

En este punto conviene mencionar que un problema de las herramientas que dan soporte al desarrollo dirigido por modelos de SIWs, como ArgoUWE (Knapp et al., 2003), OOWS Suite (Valverde et al., 2007) o HyperDE (Hypermedia Developing Environment) (Nunes & Schwabe, 2006) gira en torno el objetivo de soportar una automatización completa del proceso de desarrollo propuesto. No siempre los modelos son capaces de recoger toda la información necesaria para generar el código que implementa el sistema. En otras palabras, no siempre se pueden recoger todas las decisiones de diseño en las transformaciones que conectan los diferentes modelos. Por ello, frecuentemente es necesario llevar refinar los modelos generados por una

denomina ruta principal (*main route*) y es una de las principales aportaciones de HM³. En particular, mejora la navegabilidad del sistema porque permite guiar al usuario en la ejecución de cualquier servicio ofrecido por el SIW. En el contexto de la Ingeniería Web, la usabilidad se asocia tradicionalmente con la navegación (Palmer, 2002). Por lo tanto, un sistema de navegación que permita a los usuarios encontrar la información buscada de forma rápida y eficiente contribuye a mejorar la usabilidad del SIW, aumentando sus probabilidades de éxito (Nielsen & Loranger, 2006).

3. Soporte Tecnológico para HM³: M2DAT-HYMO

En las siguientes secciones se presentan las distintas actividades relacionadas con la construcción de M2DAT-HYMO utilizando un caso de estudio: un SIW para una comunidad científica compuesta de varios grupos de investigación que permite consultar la información relativa a los proyectos de investigación que gestiona.

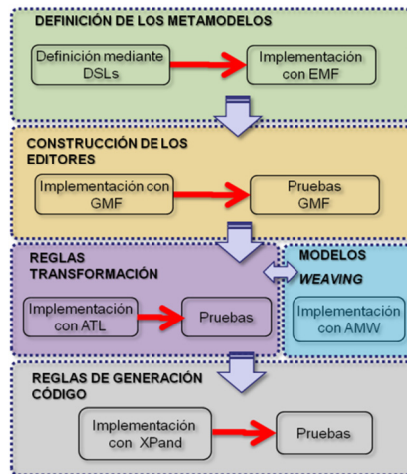


Figura 2– Proceso de desarrollo de M2DAT-HYMO

La Figura 2 representa un esquema de las tareas realizadas y el orden en el que se han llevado a cabo. Desde la implementación y definición de los metamodelos –sección 3.1–, pasando por el desarrollo de editores gráficos –sección 3.2–, la implementación de las transformaciones de modelos –sección 3.3– y los modelos de weaving –sección 3.4– hasta llegar a la generación de código –sección 3.5–.

3.1 Definición e implementación de los metamodelos

El soporte tecnológico para cualquier propuesta metodológica en el marco de la MDE comprende las herramientas necesarias para trabajar con un conjunto de lenguajes específicos de dominio (DSLs, Domain Specific Languages) (Mernik et al., 2005). El primer paso hacia la construcción de cada uno de estos DSLs pasa por definir e implementar el metamodelo subyacente.

Para la definición e implementación de los metamodelos en M2DAT-HYMO se ha utilizado EMF (*Eclipse Modelling Framework*) (Budinsky et al., 2008), un framework de Eclipse que permite definir, editar y manejar metamodelos. Posibilita la implementación en Ecore –el lenguaje de metamodelado de EMF– de los metamodelos de los DSLs de HM³. A partir del metamodelo se pueden generar herramientas para la edición y gestión de modelos conformes a dicho metamodelo.

En los últimos años EMF se ha convertido en el marco de metamodelado más utilizado en el contexto de la MDE (Vara, 2009). Por lo tanto, utilizar este framework para implementar los DSLs de M2DAT-HYMO aumenta el nivel de interoperabilidad de los modelos de HM³ ya que podrán ser importados y/o exportados desde/hacia otras herramientas MDE basadas en EMF (Vara, 2009). Otra ventaja es que EMF es un framework abierto que incorpora periódicamente nuevas tecnologías que surgen en el contexto de la MDE, lo que aumenta la extensibilidad de cualquier herramienta construida sobre EMF.

A partir de la implementación de los metamodelos en EMF se generó de forma semi-automática un editor gráfico para cada modelo de HM³. El principal inconveniente de estos editores fue la compleja estructura jerárquica con la que representaban los modelos. Por este motivo se optó por la construcción de editores gráficos de tipo nodos y arcos (o diagramadores) que simplificasen la representación gráfica de los modelos. El desarrollo de éstos se explica en la sección 3.2.

3.2 Desarrollo de editores gráficos

Tras la implementación de los metamodelos en EMF de los distintos DSLs se pasó a construir los diagramadores para éstos. Para esta tarea se contemplaron dos alternativas: utilizar un GPL (*Graphical Programming Language*) o utilizar una herramienta generativa, que proporcionase parte del código que implementaba dichos editores, como GMF (*Graphical Modelling Framework*) el framework de modelado gráfico de Eclipse (Eclipse Foundation, 2008).

El uso de un GPL permite construir editores gráficos más optimizados y potentes desde un punto de vista gráfico, pero supone una mayor inversión de recursos porque obliga a codificar prácticamente desde cero cada funcionalidad de la que se desea dotar al editor. Por otra parte, GMF permite generar de forma semi-automática el código que implementa los editores gráficos para cada uno de los DSLs, a partir de los metamodelos implementados en EMF. Además, este framework permite especificar el aspecto (forma, tamaño, color, etc.) de cada uno de los elementos del modelo que van a ser representados con el editor y definir cómo será la paleta de herramientas del editor gráfico.

Por todo esto se eligió GMF como herramienta para la construcción de los editores gráficos. La Figura 3 muestra, a modo de ejemplo, el editor gráfico que permite representar el modelo de casos de uso extendido. Por motivos de espacio, en este trabajo no presentamos el resto de editores, pero el proceso de desarrollo para ellos ha sido el mismo.

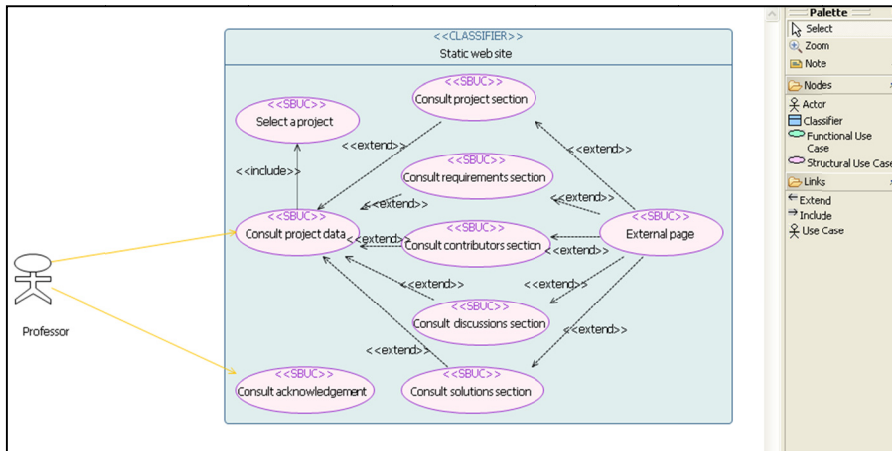


Figura 3– Editor gráfico implementado en GMF para el modelo de casos de uso extendido

3.3 Implementación de las transformaciones de modelos

Uno de los principios básicos de la MDE es la automatización del proceso de desarrollo. Por lo tanto, el siguiente paso en el desarrollo de M2DAT-HYMO fue implementar las reglas de transformación que conectan los diferentes modelos que componen en proceso de desarrollo propuesto.

Las alternativas evaluadas para el desarrollo de estas transformaciones fueron las siguientes: propuestas basadas en el uso de gramáticas de grafos (como MOLA o VIATRA); lenguajes de transformación que implementan el estándar Query/View/Transformation (QVT), tanto declarativos (MediniQVT o ModelMof) como imperativos (SmartQvt o Borland Qvto); el lenguaje de transformación ATL (Jouault & Piers, 2009).

Se optó por utilizar ATL por ser un lenguaje estable y maduro, aunque en constante fase de mejora y del cual se puede encontrar documentación variada –manuales, escenarios– y herramientas de comunicación, como listas de noticias o foros muy activos que ayudan a resolver cualquier duda no cubierta por la documentación (Vara, 2009). Otro motivo importante a la hora de escoger ATL fue su buen acoplamiento con la herramienta AMW (*ATLAS Model Weaver*) (Didonet Del Fabro et al., 2006), lo que permite utilizar modelos de weaving para establecer relaciones entre los diferentes modelos utilizados a lo largo del proceso, así como para implementar modelos de anotación. Estos modelos permiten representar la información adicional que permite llevar introducir decisiones de diseño en las transformaciones sin perjudicar el nivel de automatización de la propuesta.

En este trabajo se han definido e implementado dos transformaciones. La que permite pasar del modelo de casos de uso extendido al modelo de fragmentos extendido (EUCM2ESM) y la que, a partir de este último, genera el modelo de navegación extendido (ESM2ENM). Por motivos de espacio, nos centraremos en la transformación EUCM2ESM. Esta transformación EUCM2ESM toma como entrada el modelo de casos

de uso extendido, el modelo de weaving que implementa el modelo de anotación y el diagrama de clases UML que representa el modelo conceptual de datos.

Las reglas de transformación se definen siempre a nivel de metamodelo. De esta forma, cada regla especifica un patrón origen y un patrón destino en base a los metamodelo origen y destino. Cada vez que se encuentre una coincidencia entre el patrón origen y el modelo origen, la regla producirá en el modelo destino el patrón destino. A modo de ejemplo, la Figura 4 muestra la implementación en ATL de una de las reglas de transformación definidas en el modelo EUCM2ESM. Esta regla convierte un elemento *Functional Use Case* del modelo de casos de uso extendido en un elemento *Functional Slice* del modelo de fragmentos extendido.

```
rule FunctionalUseCase2FunctionalSlice
{
    from
        u : EUCM!FunctionalBasicUseCase
    to
        fs : ESM!FunctionalSlice (model <-thisModule.model,name <-u.name,
        sliceProperties<-u.getSliceProperties ()
        ->collect(p|thisModule.UmlProperty2SliceProperty(p)).debug ('SliceProperties'))
}
```

Figura 4 – Regla de transformación en ATL

3.4 Uso de modelos de weaving

Como se mencionaba en la introducción, uno de los aspectos tecnológicos de esta propuesta es la utilización de modelos de weaving para poder dirigir la ejecución de las diferentes transformaciones de modelos.

La Figura 5 muestra de forma esquemática el uso habitual de un modelo de weaving: establecer o identificar relaciones entre los elementos de dos o más modelos. Por ejemplo, el objeto R_1 del modelo de weaving relaciona el elemento A del modelo M_A con los elementos Y y Z del modelo M_B .

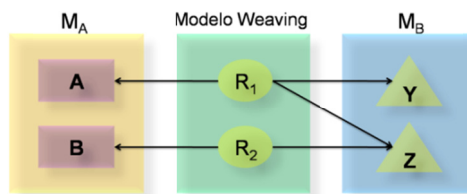


Figura 5 – Utilización habitual de un modelo de weaving

Los modelos de weaving también se definen conforme a un metamodelo implementado en Ecore. Esta es otra de las ventajas de haber usado EMF como marco de metamodelado: tanto los metamodelos de los DSLs de M2DAT-HYMO, como los metamodelos para los modelos de weaving e incluso el metamodelo de ATL o el de UML están definidos usando un mismo lenguaje de metamodelado, y esto aumenta la interoperabilidad entre los modelos. La Figura 6 describe esta idea.

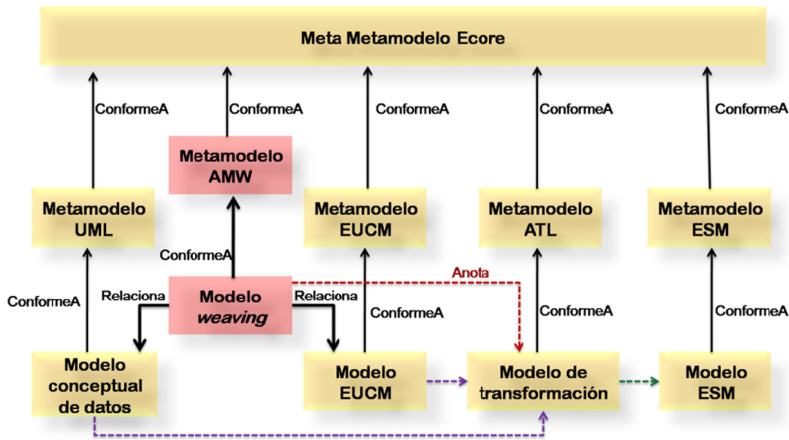


Figura 6– Relación entre modelos y metamodelos en M2DAT-HYMO

El modelo de weaving relaciona los elementos del modelo de casos de uso extendido (EUCM, *Extended Use Case Model*) con los elementos del modelo conceptual de datos. También se puede observar que el modelo de casos de uso extendido, el modelo conceptual de datos y el modelo de weaving son las entradas de la transformación. Los dos últimos proporcionan información adicional para guiar la transformación del modelo de casos de uso extendido en el modelo de fragmentos extendido (ESM, *Extended Slice Model*).

Para adaptar los modelos de weaving a las necesidades de cada transformación utilizamos el mecanismo de extensión soportado por AMW. Así, definimos un nuevo metamodelo, de forma que en los modelos definidos conformes a dichos metamodelos se puedan incluir los objetos que representan la información adicional que dirige la ejecución de la transformación. La Figura 7 recoge dicho metamodelo, donde se resaltan los elementos añadidos. Éstos son: los elementos *Containment* y *Association* que derivan del elemento *WLink*, y los elementos *Container*, *Property*, *Source* y *Target* que heredan del elemento *Associated Model Element*, que a su vez proviene del elemento *WLinkEnd*.

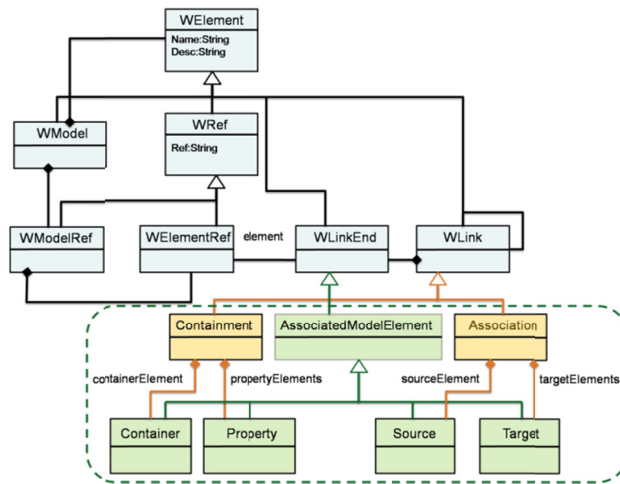


Figura 7– Extensión del metamodelo del modelo weaving

En particular, para el caso de estudio que se incluye en este artículo se han implementado dos modelos de weaving. El primero, llamado *AnnotatingCM*, relaciona los elementos *property* del modelo conceptual de datos con los elementos *basic use case* del modelo de casos de uso extendido. De esta forma, el modelo *AnnotatingCM* proporciona información para la ejecución de la transformación EUCM2ESM. Permite asociar un conjunto de propiedades a cada fragmento o *slice* del modelo de fragmentos extendido que genera. La Figura 8 muestra un sencillo ejemplo de cómo se utiliza este primer modelo de weaving.

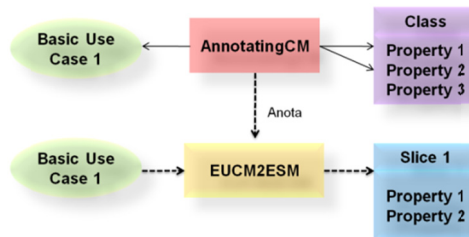


Figura 8– Utilización del modelo de weaving *AnnotatingCM*

El segundo modelo, *AnnotatingUCM*, asocia los elementos *business service* del modelo de casos de uso con los elementos *basic use case* del modelo de casos de uso extendido. A partir de los elementos *business service* se obtienen los elementos *main route* del modelo de fragmentos extendido. Una *main route* tiene asociada un conjunto de elementos *slice* involucrados en la ejecución de un servicio ofrecido por el SIW. De esta forma, este modelo de weaving aporta información para la ejecución de la transformación EUCM2ESM. Permite agrupar los elementos *slice* en las diferentes *main routes* existentes. De nuevo, la Figura 9 muestra un pequeño ejemplo de cómo se usa este modelo.

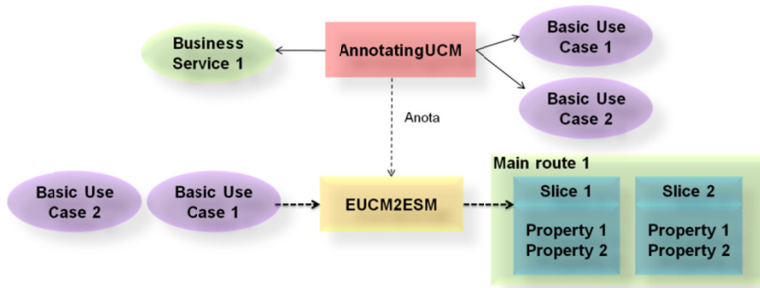


Figura 9– Utilización del modelo de anotación *AnnotatingUCM*

3.5 Generación de código

La última tarea en el desarrollo de M2DAT-HYMO fue la implementación de las reglas para la generación automática de código. Existen varias herramientas en el marco de Eclipse dirigidas a dar soporte a esta actividad, como las Java Emitter Templates (JET) (Powell, 2004) o lenguajes orientados directamente a utilizar modelos como entradas para la generación de código, como MofScript (Oldvik et al., 2005) o Xpand de openArchitectureWare (Klatt, 2007).

En particular, Xpand es un DSL basado en plantillas para la generación de código. Posee una sintaxis básica que, en combinación con Xtend (Efftinge, 2007), le permite incrementar su alcance para transformaciones modelo a modelo.

Un proyecto de generación de código basado en Xpand necesita incluir el metamodelo concreto como parte del mismo. Además, contiene el fichero de *workflow*, el fichero de propiedades del *workflow* y el fichero de configuración. El paquete *templates* contiene todos los ficheros de plantillas para la generación de código. El fichero de *workflow* representa el núcleo central de un proyecto de generación de código y permite indicar el modo deseado de ejecución de la generación de código. Tiene una estructura de componentes en el que cada uno será el encargado de una labor diferente. Se pueden distinguir los siguientes componentes: (1) el componente *xmiParser* que contiene la estructura del modelo y es donde se identifica el paquete principal del metamodelo. (2) El componente *dirCleaner* que elimina el directorio donde se deposita el código generado. (3) El componente *generator* indica el paquete del metamodelo, el nodo raíz a partir del cual vamos a empezar a generar el código y el directorio de salida para la generación de código.

Para que M2DAT-HYMO pueda generar código a partir del modelo de navegación extendido es necesario definir reglas que permitan transformar los elementos del modelo a código y así obtener fragmentos HTML, hipervínculos entre ellos y las rutas de navegación que proporcionará el sistema. Dado que HM³ no incorpora un modelo de presentación, es necesario que nuestras reglas de transformación generen otros dos tipos de contenidos. Por una parte, una hoja de estilos que permita dar formato a la salida del código HTML generado; y por otra parte un fichero de funciones Javascript que proporcione la funcionalidad necesaria para manipular la información de las rutas principales. Con ello dotamos al SIW de una interfaz gráfica aceptable y cierto comportamiento.

A modo de ejemplo, la Figura 10(a) muestra el mapa de navegación generado por M2DAT-HYMO para el caso de estudio, referido a la ruta principal *consult a project*. La Figura 10(b) muestra cada uno de los ficheros generados.

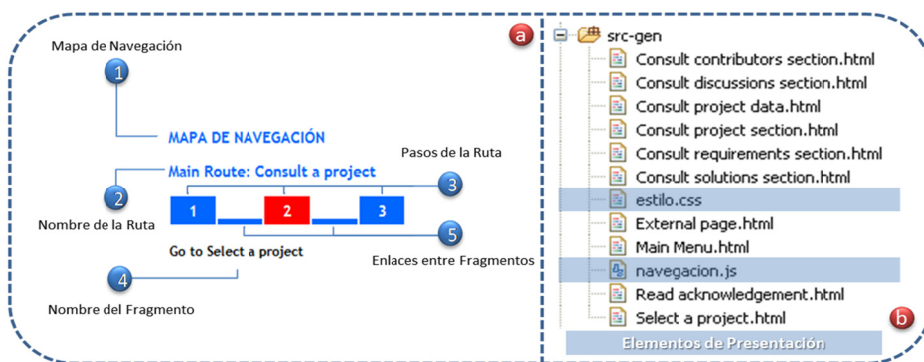


Figura 10 – Caso de Estudio: (a) Mapa de navegación (b) Ficheros generados

4. Conclusiones y Trabajos Futuros

En este artículo hemos presentado M2DAT-HYMO, el módulo de M2DAT que da soporte al Método del Modelado del Hipertexto de MIDAS (HM³). En esencia, M2DAT-HYMO implementa varios DSLs utilizando EMF como marco de metamodelado y GMF para la construcción de diagramadores, mientras que la automatización del proceso de desarrollo descansa en la implementación de varias transformaciones modelo a modelo y modelo a texto. Para la implementación de las primeras se ha utilizado el lenguaje ATL y las segundas se han desarrollado con Xpand.

A diferencia de otras herramientas, como ArgoUWE o HyperDE, en las que los modelos generados por algunas transformaciones deben ser refinados manualmente, M2DAT-HYMO permite añadir información adicional utilizando modelos de weaving para controlar la ejecución de la transformación. Además, los SIWs generados por M2DAT-HYMO incluyen rutas de navegación que ayudan a mejorar la usabilidad y navegabilidad del sistema.

Por otro lado, la Ingeniería Web está en constante avance y desarrollo y esto ha hecho que sus principios básicos (navegabilidad, usabilidad, etc.) hayan evolucionado de igual manera. Así, durante los últimos años ha surgido un nuevo tipo de aplicaciones denominadas RIAs (*Rich Internet Application*) (Fraternali et al., 2010) que suponen un salto cualitativo para los SIWs. Aquellos SIWs que hasta ahora sólo mostraban datos, se convierten en lugares que incluyen objetos multimedia y donde aumenta el nivel de interactividad con los usuarios, lo que posibilita que éstos puedan llevar a cabo tareas mucho más complejas. Dado que la demanda de este tipo de aplicaciones crece cada día, las metodologías de desarrollo Web se encuentran inmersas en un proceso de adaptación a estas nuevas necesidades. Así, una de las líneas futuras de este trabajo pasa por adaptar HM³ para soportar el desarrollo de este nuevo tipo de aplicaciones. Otras metodologías como UWE (Koch et al., 2009), OOHDM (Rossi et al., 2008) o

WebML (Comai & Toffetti Carughi, 2007) ya han empezado con este proceso de adaptación añadiendo nuevos modelos o combinando los suyos con otros ya existentes.

Por otro lado, en la sección 3.5 se mencionaba que HM³ no incluye por ahora un modelo de presentación. Por lo tanto, otra línea futura pasa por añadir un modelo de presentación que soporte la especificación de la interfaz gráfica del SIW. De hecho, la adaptación de HM³ para soportar la generación de RIAs hace imprescindible la inclusión de un modelo de presentación, ya que una de las ideas principales de las RIAs es ofrecer una interfaz muy compleja y estructurada en una sola pantalla, siguiendo el paradigma de la ventana única (*single window paradigm*). En esta misma dirección viene trabajando otras propuestas para la Ingeniería Web. En particular, las metodologías OOHDM y WebML, que ya disponían de modelo de presentación, lo han adaptado a los nuevos escenarios de las RIAs (Comai & Toffetti Carughi, 2007; Rossi et al., 2008) o han adoptado los modelos de presentación de otras metodologías; como RUX que clasifica la construcción de interfaces en interfaz abstracta, concreta y final (Preciado et al., 2007).

Agradecimientos

Este trabajo se ha llevado a cabo en el marco de los proyectos MODEL CAOS (Ref. TIN2008-03582) y Agreement Technologies (CONSOLIDER CSD2007-0022) financiado por el Ministerio de Ciencia y Tecnología de España.

Referencias

- Budinsky, F., Merks, E., & Steinber, D. (2008). *Eclipse Modelling Framework 2.0* (Segunda ed.). Addison-Wesley Professional.
- Cáceres, P., De Castro, V., Vara, J., & Marcos, E. (2006). Model transformations for hypertext modelling on web information systems. *SAC*, 1232-1239.
- Comai, S., & Toffetti Carughi, G. (2007). A Behavioral Model for Rich Internet Applications. *Proceedings of the 7th International Conference Web Engineering (ICWE'07, LNCS 4607)*, (págs. 364-369).
- De Castro, V. (2007). *Aproximación MDA para el Desarrollo Orientado a Servicios de sistemas de Información Web: del Modelo de Negocio al Modelo de Composición de Servicios Web*. Madrid: Universidad Rey Juan Carlos.
- De Castro, V., Marcos, E., & Wieringa, R. (Junio de 2009). Towards a Service-oriented MDA-Based Approach to the Alignment of Business Processes with it Systems: From the Business Model to a Web Service Composition Model. *International Journal on Cooperative Systems*, 18(2), 225-260.
- Didonet Del Fabro, M., Bézivin, J., & P., V. (2006). Weaving Models with the Eclipse AMW plugin. Esslingen, Alemania: Eclipse Modeling Symposium.
- Efftinge, S. (22 de Julio de 2007). *Xtend Language Reference*, 4.1. Obtenido de http://www.eclipse.org/gmt/oaw/doc/4.1/r25_extendReference.pdf

- Fraternali, P., Comai, S., Bozzon, A., & Toffetti, G. (2010). Engineering Rich Internet Applications with a Model-Driven Approach. *ACM Transactions on the Web*, Vol. 4, No. 2, Article 7.
- Jouault, F., & Piers, W. (2009). *ATL User Guide*. Obtenido de http://wiki.eclipse.org/ATL/User_Guide
- Klatt, B. (2007). A Closer Look at the model2text Transformation Language. *Chair for Software Design and Quality (SDQ)*.
- Knapp, A., Koch, N., Moser, F., & Zhang, G. (2003). ArgoUWE: A Case Tool for Web Applications. *First International Workshop on Engineering Methods to Support Information Systems Evolution (EMSISE'03)*. Geneva, Switzerland.
- Koch, N., Pigerl, M., Zhang, G., & Morozova, T. (2009). Patterns for the Model-based Development of RIAs. *Proceedings of the 9th International Conference Web Engineering (ICWE'09), LNCS 5648*, (págs. 283-291).
- Lowe, D. (2003). Web System Requirements: An Overview. *Requirements Engineering Journal*, 8, 102-113.
- Marcos, E., Acuña, C., & Cuesta, C. (2006). Integrating Software Architecture into a MDA Framework. *Proc. of 3rd European Workshop on Software Architecture* (págs. 27-143). France, Nantes: Springer Verlag, LNCS 4344.
- Mernik, M., Heering, J., & Sloane, A. (2005). When and How to Develop Domain-Specific Languages. *ACM Computing Surveys*, 37(4), 316-344.
- Miller, J., & Mukerji, J. (1 de Junio de 2003). *MDA Guide*, 1.0.1. Obtenido de <http://www.omg.com/mda>
- Nielsen, J., & Loranger, H. (2006). Prioritizing Web Usability. *New Riders Publishing*.
- Nunes, D., & Schwabe, D. (2006). Rapid Prototyping of Web Applications Combining Domain Specific Languages and Model Driven Design. *6th International Conference on Web Engineering (ICWE'06)*. Palo Alto, California, USA.
- Oldvik, J., Neple, T., Gronmo, R., Aagedal, J., & Berre, A.-J. (2005). Toward Standardized Model to Text Transformations In Model Driven Architecture. *Foundations and Applications*, 239-253.
- OMG. (2002). *MOF 2.0 Query/View/Transformations RFP*. Obtenido de OMG document ad/2002-04-10
- Palmer, J. (2002). Web Site Usability, Design and Performance Metrics. *Information Systems Research*, XIII(2), 151-167.
- Powell, A. (2004). *Generate Code with Eclipse's Java Emitter Templates*. IBM Developer Works.
- Preciado, J.C., Linaje, M., Comai, S., & Sánchez-Figueroa, F. (2007). Designing Rich Internet Applications with Web Engineering Methodologies. *International Symposium on Web Site Evolution*, 23-30.

- Rossi, G., Urbietta, M., Ginzburg, J., Distante, D., & Garrido, A. (2008). Refactoring to Rich Internet Applications. A Model-Driven Approach. *Proceedings of the 8th International Conference Web Engineering (ICWE'08)*, (págs. 1-12).
- Schmidt, D. (2006). Model-Driven Engineering. *IEEE Computer*(39).
- Selic, B. (2001). MDA Manifestations. *Upgrade. The European Journal for the Informatics Professional*, IX(2), 12-16.
- Valverde, F., Valderas, P., Fons, J., & Pastor, O. (2007). A MDA-Based Environment for Web Applications Development: From Conceptual Models to Code. *6th International Workshop on Web-Oriented Software Technologies (IWWOST)*. Como, Italy.
- Vara, J.M. (2009). M2DAT: a Technical Solution for Model-Driven Development of Web Information Systems. Madrid: Universidad Rey Juan Carlos.

SAM - Sistema Automatizado del Método MECAP para Especificar Casos de Prueba

Edumilis Méndez¹, María A. Pérez¹, Kenyer Domínguez¹, Luis E. Mendoza¹.

{emendez, movalles, kdoming, lmendoza}@usb.ve

¹ Laboratorio de Investigación en Sistemas de Información (LISI), Departamento de Procesos y Sistemas, Universidad Simón Bolívar, Apartado Postal 89000, Caracas 1080-A, Venezuela.

Resumen: Existen cuatro elementos que son relevantes al momento de definir las pruebas: Confiabilidad, Costo, Tiempo y Calidad. El tiempo de desarrollo y el costo del producto se incrementan cuando se desean pruebas confiables y un software de calidad. Pero ¿qué se puede hacer para que los involucrados comprendan que las pruebas deben ser vistas como una red de seguridad? Si la calidad no se contempla antes de comenzar las pruebas, entonces ella no estará cuando se éstas terminen. El objetivo de este artículo es presentar la herramienta, SAM – Sistema Automatizado del Método MECAP que permite especificar Casos de Prueba a partir de Casos de Uso incorporando elementos que promueven la verificación y validación de la trazabilidad entre la Gestión de Requerimientos, el Análisis y Diseño y las Pruebas. SAM soporta el proceso de pruebas de forma automatizada, mejorando la confiabilidad de las mismas.

Palabras clave: Pruebas de Software; Calidad del Software; Casos de Prueba; Herramienta de Pruebas.

Abstract: There are four elements that are relevant when test are defined, Reliability, Cost, Time, and Quality. Development time and product cost increases when you want reliable tests and quality software. But, what can you do to make that stakeholders understand that the tests must be viewed as a security net? If the quality is not considered before starting the tests, then it will not be when they finish. The goal of this paper is to present the tool, SAM - Automated Systems of MECAP Method, which allow to specify test cases from Use Cases incorporating elements that promote the verification and validation of the traceability among Requirement Management, Analysis and Design, and Test. SAM automatically support the testing process, improving the test reliability.

Keywords: Software Test; Software Quality; Test Cases; Test Tool.

1. Introducción

La Disciplina de Pruebas se aborda desde las primeras fases, esto quiere decir que las pruebas se deben comenzar a planificar y además se debe establecer cuál es la estrategia de pruebas a seguir una vez que han aprobado todos los Casos de Uso (CU). Las pruebas del software contraponen dos aspectos importantes: La Verificación y la Validación (V&V). Durante la Verificación se responde a si ¿Se está construyendo el producto correctamente? Así mismo, la Validación establece si ¿Se está construyendo el producto correcto?

El objetivo de la Disciplina de Pruebas es evaluar la calidad del producto a lo largo de todo el ciclo de vida apoyándose en un conjunto de buenas prácticas, entre las que destacan según Leffingwell & Widrig (2006): (a) Encontrar y documentar las fallas en el producto de software: defectos y problemas. (b) Evaluar las suposiciones hechas en las especificaciones de requerimientos y diseño a través de demostraciones concretas. (c) Verificar que el producto de software trabaja según el diseño. (d) Validar que los requerimientos son implementados apropiadamente.

Kruchten (2000), Pressman (2002), Pfleeger (1998) y Sommerville (2000) afirman que el proceso de ejecución de Pruebas debe ser considerado durante todo el ciclo de vida de un proyecto, para así obtener un producto de alta calidad. Su éxito dependerá del seguimiento de una Estrategia de Prueba adecuada. La Estrategia de Prueba de Software integra un conjunto de actividades que describen los pasos que hay que llevar a cabo en un proceso de prueba, tomando en consideración cuánto esfuerzo y recursos se van a requerir, con el fin de obtener como resultado una correcta construcción del software (Pressman, 2002).

Las empresas desarrolladoras de software invierten en las pruebas entre el 30% y 50% del costo total del software (Pressman, 2002). Esto representa un esfuerzo considerable que indica que es una disciplina importante y que los resultados que ella arroje así como su confiabilidad pueden impactar sobre la percepción del cliente o usuario en cuanto a la calidad del software que se le está entregando.

Las pruebas de software consisten en la dinámica verificación del comportamiento de un programa en un conjunto finito de casos de prueba, convenientemente seleccionados de la ejecución de un dominio contra el comportamiento esperado. ¿cómo se evidencia que se efectuaron las pruebas de un software? ¿cuáles fueron los resultados? ¿quién las llevó a cabo? ¿cuándo? ¿se alcanzó el porcentaje de satisfacción esperado? Las respuestas se obtienen a través de la documentación.

El estándar IEEE para Documentación de las pruebas de software (IEEE829-98) proporciona una buena descripción de los documentos de prueba y de su relación entre sí e incluso con el proceso de prueba. SWEBOK (2004) señala que los documentos de pruebas pueden incluir, entre otros, la especificación de los Casos de Prueba (CP). Es por ello que al hacer las pruebas funcionales, hay tres cuestiones claves (Utting y Legeard, 2007): El diseño del caso de prueba, realizar los ensayos y análisis de los resultados, y la verificación de cómo la prueba cubre las necesidades. Cada CP se define por el contexto de prueba, una situación, y algunos criterios de falla o de aprobación. Perry (2006), presenta una guía completa de cómo llevar a cabo un proceso de prueba efectivo. Inclusive, habla de los CP y propone una plantilla para CP, la cual incluye

algunos aspectos que nosotros hemos considerado para nuestro método. Por ejemplo, usar ID (Identificador) para el CP y para el UC. Lewis (2000) propone una plantilla para CPs, donde incorpora condiciones, ambiente, versión y sistema.

Algunos autores como Pinkster et al., 2006 indican que una mejora subsecuente en la calidad de las pruebas es posible si los requerimientos son tomados como la base de las pruebas, denominándola Pruebas basadas en Requerimientos. El probador puede centrarse en los requerimientos de mayor prioridad, y a los problemas se les da la misma prioridad que al requisito relacionado con el CP y el problema que se ha encontrado. De esta forma, los desarrolladores pueden ver los aspectos que necesitan resolverse primero.

Así mismo, el Profile del Unified Modeling Language (UML) para Pruebas (2005) indica que un contexto de la prueba es sólo un CP de alto nivel. Un CP siempre devuelve un veredicto. El veredicto puede ser arbitrado —calculado por el árbitro— o no arbitrado (es decir, siempre provisto por el comportamiento de prueba). Algunos de los beneficios de la matriz de prueba/requerimientos son (Lewis, 2000): correlación de las pruebas y los scripts con los requisitos; facilita el estado de las revisiones y actúa como un mecanismo de trazabilidad a lo largo del ciclo de desarrollo; incluye el diseño y ejecución de prueba. El Plan de Calidad de Planificar-Hacer-Verificar-Actuar (PHVA) se aplica al proceso de pruebas de software. El método propuesto ayuda a aplicar PHVA.

A diferencia de las iniciativas anteriores, el método MECAP – Método para Especificar Casos de Prueba (Méndez, Pérez & Mendoza, 2007 y 2008) incorpora todas las ideas citadas anteriormente con la finalidad de obtener un método que soporte todos los elementos que conforman una estrategia de pruebas. Ejecutar MECAP de forma manual representa un esfuerzo considerable, así lo evidencian resultados preliminares en varios proyectos por lo que surgió la necesidad de automatizarlo a través del uso de una herramienta que garantizara todos los aspectos provistos con el método.

En este sentido, el objetivo de este artículo es presentar la herramienta SAM (Sistema Automatizado del Método MECAP) que facilita la implementación del método MECAP y contribuye con la trazabilidad entre los requerimientos, el análisis y diseño y las pruebas del software.

La metodología con la que se desarrolló SAM fue UP (Unified Process). El tiempo de ejecución del proyecto de desarrollo de SAM fue de 5 meses, a través 5 iteraciones.

Este trabajo consta de 4 secciones, incluyendo la presente introducción. En la sección 2 se presenta el método MECAP. En la sección 3, se muestra la herramienta SAM. Por último, en la sección 4 se presentan las conclusiones y el trabajo futuro.

3. Método para Especificar Casos de Prueba (MECAP)

El método propuesto consiste en construir CP a partir de CU ya que se parte del supuesto que se debe probar el comportamiento del software en base a las solicitudes o requerimientos.

Un CP es una especificación, usualmente formal, de un conjunto de entradas de prueba, condiciones de ejecución y resultados esperados, identificados con el propósito

de hacer una evaluación de aspectos particulares de un elemento objeto de prueba (Kruchten, 2000):

- Los CP reflejan trazabilidad con los CU (Funcionalidad), ya que éstos muestran una secuencia ordenada de eventos, al describir flujos básicos, flujos alternos, precondiciones y postcondiciones.
- Las especificaciones suplementarias de requerimientos ya que existen otras características de calidad a evaluar, además de la funcionalidad, como Usabilidad, Confiabilidad, Eficiencia, Mantenibilidad y Portabilidad.
- Las especificaciones de diseño del Sistema, ya que se debe verificar que el software fue implementado según el diseño y que los elementos arquitectónicos garantizan la calidad del software.

Esto garantiza que los procedimientos de pruebas sean compatibles con las necesidades y requisitos de los usuarios/clientes. En la práctica se tiende a asumir que un CU en sí mismo es un CP y que el equipo del proyecto trabaje correctamente sobre los CU sin planificar los CP. Cuando se sienta a probar los CU, intuitivamente asume datos y procedimientos de pruebas sin que éstas queden documentadas. Esto es un error ya que el CP extiende o amplía la información contenida en un CU, por ejemplo, en los CU no indicamos valores ni condiciones para la pruebas.

Los CP son esenciales para todas las actividades de pruebas (Kruchten, 2000): (a) Son la base para diseñar y ejecutar los procedimientos de pruebas. (b) La profundidad de las pruebas es proporcional al número de casos de pruebas. (c) El diseño y desarrollo, y los recursos necesarios son gobernados por los casos de pruebas requeridos. (d) Si los CP no son correctos, la Calidad del Sistema se pone en duda y las pruebas dejan de ser confiables.

El moverse de un CU a un CP es un proceso razonablemente amplio y no trivial. Leffingwell & Widrig (2006) señalan cuatro (4) pasos para lograrlo: (a) Identificar los escenarios del CU. (b) Por cada escenario, identificar uno o más CP. (c) Por cada CP, identificar las condiciones que originaran su ejecución. (d) Completar el CP incorporando valores de datos.

Estos pasos indican qué debe hacerse pero no se enseña el cómo de una forma detallada. Tomando como base los pasos que establecen Leffingwell & Widrig (2006) se propone un método para especificar CP a partir de CU, constituido por cuatro fases principales: (1) Identificar Escenarios, (2) Identificar CP, (3) Especificar los CP, (4) Ejecución y Aprobación del CP.

El aporte de este método es que incorpora elementos de trazabilidad de las pruebas con respecto a todo el proceso de desarrollo y a su vez mejora la estrategia de las pruebas al disciplinar este proceso. Así mismo, ayuda a documentar las ideas previas a las pruebas, los CP y cómo estos fueron generados. Este método mantiene los 4 roles que se proponen dentro de la Disciplina de Pruebas: Gerente de Pruebas, Diseñador de las Pruebas, Analista de Pruebas y Probador.

Fase 1: Identificar Escenarios.

El analista de pruebas activa esta fase una vez que se hayan verificado las narrativas de los CU, por lo menos se tengan claras definidas las funcionalidades del sistema, para:

1. Identificar los escenarios tomando como base las narrativas de los CU y considerando cada uno de los escenarios específicos que ocurren para cada CU. El flujo normal, cada flujo alternativo o la combinación de ellos es un escenario, que puede ser ejecutado y probado. Esto deriva que siempre el primer escenario sea el que evoca todo el flujo normal de ese CU en particular y que la relación entre CU y escenarios sea de uno a muchos.
2. Presentar gráficamente la secuencia de eventos que se plantea en cada CU: esto permite, como lo muestra la Figura 1 abstraer los eventos que ocurren en un CU: el flujo normal o básico así como, los flujos alternos lo cual sirve de apoyo para visualizar fácilmente las posibles combinaciones que representarían un escenario ya que establece en qué punto del flujo básico ocurre y además qué sucede después que se activa ese flujo alternativo: finaliza el CU o retorna al flujo básico.
3. Chequear que se hayan representado gráficamente todos los Flujos alternos con su acción de finalización o retorno.
4. Establecer a través de una tabla (como lo refleja la Figura 2), todos los escenarios asociados a un CU de la Figura 1.
5. Identificar cada escenario del CU indicando el flujo normal y/o el o los flujo(s) alternativo(s) que lo componen. La Figura 3 constituye el primero de los 3 artefactos que se generan en MECAP: Tabla de Escenarios por CU. En ésta se puede observar que se incorpora el ID del Escenario con el propósito de establecer un elemento de trazabilidad de las pruebas lo que facilita, a su vez, las actividades de verificación y aprobación de las pruebas. Como se puede observar en la Figura 3, el ID puede estar conformado por el Nro. CU y Nro. Escenario, esto nos indicaría por ejemplo, que si se tiene 02-02, se traduce como el escenario 02 del CU 02. Esta nomenclatura debe ser establecida por la organización. Además, es importante que se agregue un nombre corto durante la identificación de los flujos alternos que representan o conforman un escenario. La información de los escenarios está asociada a la narrativa de un CU (Validar Usuario del Sistema TAI – Tarjeta Académica Inteligente), el cual sirve de ejemplo para explicar el método propuesto.
6. Verificar que se hayan identificado y descrito todos los escenarios posibles para cada CU. En conclusión, cada escenario representa el número de posibilidades o formas a través de las cuales se puede recorrer el CU. Esto evita que solo se prueben algunas de las combinaciones posibles.

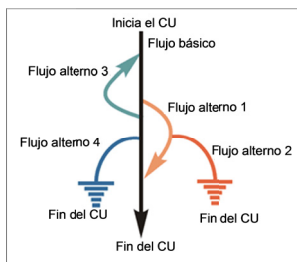


Figura 1. Visualización de los Flujos en un CU (Leffingwell & Widrig, 2006)

Nro. Escenario	Flujo Originario	Flujo alterno	Próximo alterno	Próximo alterno
Escenario 1	Flujo Básico			
Escenario 2	Flujo Básico	Flujo alterno 1		
Escenario 3	Flujo Básico	Flujo alterno 1	Flujo alterno 2	
Escenario 4	Flujo Básico	Flujo alterno 3		
Escenario 5	Flujo Básico	Flujo alterno 3	Flujo alterno 1	
Escenario 6	Flujo Básico	Flujo alterno 3	Flujo alterno 1	Flujo alterno 2
Escenario 7	Flujo Básico	Flujo alterno 4		
Escenario 8	Flujo Básico	Flujo alterno 3	Flujo alterno 4	

Figura 2. Escenarios por CU (Leffingwell & Widrig, 2006)

ID Escenario	Flujo Originario	Flujo alterno	Próximo alterno	Próximo alterno
02-01	Flujo Básico			
02-02	Flujo Básico	Flujo alterno 1: Error Login		
02-03	Flujo Básico	Flujo alterno 1: Error Password	Flujo alterno 3: Presionar Cancelar	
02-04	Flujo Básico	Flujo alterno 2: Error Password		
02-05	Flujo Básico	Flujo alterno 2: Error Password	Flujo alterno 3: Presionar Cancelar	
02-06	Flujo Básico	Flujo alterno 1: Error Login	Flujo alterno 2: Error Password	
02-07	Flujo Básico	Flujo alterno 1: Error Login	Flujo alterno 2: Error Password	Flujo alterno 3: Presionar Cancelar
02-08	Flujo Básico	Flujo alterno 3: Presionar Cancelar		

Figura 3. Tabla de Escenarios para CU002

Fase 2: Identificar los CP

El proceso de pruebas varía de empresa a empresa y de proyecto a proyecto, pero en cada situación un CP documenta un número de elementos comunes. De un escenario de un CU pueden derivarse varios CP. Después que se han identificado todos los escenarios de un CU, se analizan cada uno de ellos en la búsqueda de los posibles CP. Esto se traduce en las siguientes actividades:

1. Se organizan las ideas de pruebas en base los elementos que se desean probar: Funcionalidad (CU), atributos de Calidad, Validaciones de entradas y salidas, Base de datos, interfaces, etc. Esto va a depender del tipo de aplicación, de las restricciones tecnológicas y que maneja el proyecto y de propósito y motivación de las pruebas y experiencia del equipo de pruebas (sobre todo, del probador).
2. Se comienza a llenar la Tabla que se muestra a través de la Figura 4, la cual representa el segundo artefacto del método propuesto y cuyos datos están asociados a considerar los CP para el Escenario 02-02 (Error en Login). Con la información proveniente de las ideas de pruebas se puede decir que para este caso se podría tener un CP para validar “error en login” cuando se introducen caracteres inválidos, un CP cuando el login está en minúsculas, un CP cuando el login es mayor a 10 caracteres, un CP cuando el login está en blanco. Se completa la información asociada a cada CP identificado: ID Caso de Prueba, Nombre del CP, Resultados esperados (valores, mensajes de error, etc.), Nivel de Prueba y Tipo de Prueba. Con respecto al ID del CP se propone incluir en la nomenclatura estándar que determine la organización una estructura que refleje CU-Escenario-CP para así saber por ejemplo que 02-02-01 significa el CP 01 del Escenario 02 del CU 02.
3. Se verifica que se hayan identificado todos los CP para cada escenario. Después de esto, se procede a la siguiente fase.

ID Caso de Prueba	Nombre del Caso de Prueba	Resultados esperados	Nivel de Prueba	Tipo de Prueba
02-02-01	Login con caracteres inválidos	<i>Mensaje Error 20: Login Inválido</i>	Sistema/ Aceptación	Función/ Control de acceso
02-02-02	Login con minúsculas	<i>Mensaje Error 20: Login inválido</i>	Sistema/ Aceptación	Función/ Control de acceso
02-02-03	Login con longitud mayor a 10 caracteres	<i>Mensaje Error 20: Login inválido</i>	Sistema/ Aceptación	Función/ Control de acceso
02-02-04	Login en Blanco	<i>Mensaje Error 20: Login inválido</i>	Sistema/ Aceptación	Función/ Control de acceso

Figura 4. Tabla de Casos de Prueba para el Escenario 02-02

Fase 3: Especificaciones de los CP

Uno de los aportes más importantes de esta investigación es el tercer artefacto que se utiliza para especificar con detalle el CP y que se muestra a través de la Figura 5: Tabla de Especificaciones de CP (ECP).

Para cada CP, se deben llevar a cabo las siguientes actividades:

1. Identificar el nombre del Sistema, ID CU, ID de Requerimiento, ID Escenario, ID del CP y versión del CP. Esto permite hacer una traza bidireccional entre estos elementos: por ejemplo, se puede conocer si se especificaron todos los CP para todos los CU y se probaron todos los CU (cobertura de las pruebas).
2. Identificar el nivel y tipo de prueba asociada al CU y cuya información proviene de la Tabla CP por Escenario.
3. Indicar el ambiente de las pruebas. Se puede señalar el nombre de la empresa, si es el ambiente de desarrollo o producción, o si tengo varios ambientes en la empresa se indica en cuál de ellos se ejecutará ese CP en particular.
4. Identificar el nombre del autor del CP y del Probador. Se recomienda que sean personas diferentes las que ocupen estos roles a fin de darle objetividad y confiabilidad al proceso de las pruebas.
5. Señalar la fecha de creación de la versión de ese CP y la fecha en la que fue ejecutado.
6. Identificar las condiciones que deben estar presentes para ejecutar el CP. ¿Cuáles son las condiciones que originan o causan que un usuario ejecute un evento específico o una secuencia de eventos? En la Figura 5, se tiene el CP 02-02-02 en cual se propone el CP para Login con minúsculas. En esta figura se observa que se deben haber implementado todas las funcionalidades asociadas con Validar Usuario, así mismo, los Datos que se utilizaran en este CP hayan sido validados y aprobados por la instancia correspondiente, etc.
7. Describir el procedimiento del CP. Este procedimiento está compuesto de los pasos que se deben hacer para probar el Escenario del CU a través de lo que plantea el CP, de las condiciones particulares que pudieran aplicar para un determinado paso, los valores utilizados, los resultados esperados y los resultados obtenidos. Cabe resaltar, que estos últimos se incluyen en la Tabla ECP una vez que el CP es ejecutado.

ID/Nombre Sistema/Proyecto: SIS.PROY	Nivel de Prueba: Sistema/Aceptación			
ID Caso de Uso: CU.02 Validar Usuario	Tipo(s) de Prueba(s): Función/Control de acceso			
ID Requerimiento: (solo para Caso de Uso No Funcional)	Ambiente de Prueba: AMBIENTE1			
ID/Nombre Escenario: 02.02 Error en Login	Autor del Caso de Prueba: LISI			
ID/Nombre Caso de Prueba: 02.02.02 Login con minúsculas	Nombre del Probador: Probador 1			
Versión del Caso de Prueba: v.1.	Fecha de Creación: 10.01.07 Fecha de Ejecución: 15.03.07			
Condición(es) para que se ejecute el Caso de Prueba:				
El usuario desea acceder al sistema. Se hayan Implementado todas las funciones que se relacionan con validar usuario. Datos a utilizar para las pruebas hayan sido validados y aprobados. Deben existir usuarios registrados como válidos.				
Para la Ejecución del ID Caso de Prueba:				
Paso	Condición	Valor(es)	Resultado Esperado	Resultado Obtenido
Introduce login presiona Ok	intento de acceso : 3	aDM22	Mensaje Error 20 Login inválido	✓
Introduce login presiona Ok	intento de acceso : 3	administrador	Mensaje Error 20 Login inválido	✓
Introduce login presiona Ok	intento de acceso : 3	AdminisTRADOR	Mensaje Error 20 Login inválido	✓
Criterios de Aprobación del Caso de Prueba: Si se cumplen en un 100% los resultados esperados				
Decisión de Aprobación del Caso de Prueba: Aprobó: ☒ Fallo: ☐ (marque con una X el resultado obtenido)				
Fecha de Aprobación del Caso de Prueba: 15.03.07				

Figura 5. Tabla de Especificación del CP 02-02-02

Sin la incorporación de datos no es posible ejecutar las pruebas y determinar los resultados. Se deben observar las especificaciones suplementarias para encontrar medidas de desempeño (mínimo y máximo), rangos válidos de entrada, protocolos de interfaces, entre otros.

- Se indica cuál es el Criterio de Aprobación del CP. Como se observa en el ejemplo de la Figura 5. El Criterio es que se deben cumplir en un 100% todos los resultados esperados. También puede indicarse a nivel de pasos cuando el CP involucra pasos diferentes y alguno(s) de ellos son más importante que otros: Si se cumplen en un 100% los resultados esperados del Paso 2, 4 y 5.
- El analista y el diseñador de las pruebas verifican que todos los CP se hayan especificados correctamente.

Fase 4: Ejecución y Aprobación del CP

Esta fase consiste en poner en práctica el procedimiento de los CP que se encuentran en la Tabla ECP. A continuación se indican las actividades de esta fase:

- Debe comprobarse que está dado el ambiente y las condiciones para ejecutar los CP. Todos los involucrados en las pruebas deben cooperar en esto.
- El Gerente y el Diseñador de las pruebas deben autorizar que se active el ciclo de prueba.
- El Probador ejecuta todos los CP e incorpora los datos de los resultados obtenidos en cada Tabla ECP.
- El Gerente de Pruebas toma la decisión de si aprobó o falló el CP en base al criterio de aprobación y además, indica la fecha de la aprobación y en algunos casos, coloca su firma.
- El equipo de pruebas verifica si se cumplió el criterio de salida del ciclo de prueba para decidir si se aprueba el ciclo de prueba o hay que hacer pruebas de regresión y pruebas adicionales a ciertos CP en un próximo ciclo de prueba hasta que se obtengan los criterios de aceptación.
- El equipo de pruebas guarda todos estos entregables y publica los resultados en

base a los resultados de los ciclos de prueba, cambios, etc. que se dieron hasta completar el proceso de las pruebas.

Puntos de Control

Para garantizar la correcta aplicación del método y cumplir con la estrategia, a continuación se enumeran los puntos de control que se realizan en cada fase.

Fase 1:

- ¿Existe una matriz de Escenarios por cada CU del sistema?
- Revise que todos los escenarios del CU correspondiente se hayan indicado en la matriz de escenarios por CU.
- Revise que los ID del CU y CP estén completos y se correspondan con la nomenclatura propuesta.

Fase 2:

- ¿Existe una matriz de CPs por escenario?
- Para cada matriz de CP por escenario, revisar que están completos y correctos los ID, nombres, resultados esperados, nivel y tipo de prueba.

Fase 3:

- ¿Existe una tabla de Especificación de CP para cada CP identificado en la fase anterior?
- Revisar la trazabilidad entre los ID de los CP, CU, requerimiento y escenario.
- Chequear que todos los puntos indicados en la tabla de especificación de CP estén completos y correctos a excepción de los campos a llenar para la próxima fase (ejecución).
- ¿Se validaron los criterios de aprobación para cada especificación de CP?

Fase 4:

- ¿Se documentaron los resultados de las ejecuciones de los CP a través del llenado de los campos: resultados obtenidos, aprobó/falló, fecha de aprobación, fecha de ejecución.
- ¿Se indicó en el documento Plan de Pruebas, el criterio de cumplimiento del ciclo de pruebas?
- ¿Los resultados de las pruebas y los cambios durante el proceso de las pruebas fueron entregados y publicados?

En la siguiente sección se presenta la herramienta SAM que automatiza el método propuesto y presentado en esta sección.

4. Herramienta SAM – Sistema Automatizado del Método MECAP

A fin de integrar las disciplinas de Requerimientos con las de Pruebas, se desarrolló una herramienta denominada Workbench UML-UCM que permite convertir un archivo .uml (generado a través de StarUML) en un archivo transformado .ucm (en la herramienta Use Case Maker) para especificar las narrativas de los CU. Una vez finalizadas las especificaciones, se procede a exportar/generar un archivo (.xml) el cual será cargado en la herramienta SAM para luego proceder con la ejecución del método MECAP. Los detalles de Workbench UML-UCM escapan del alcance de este artículo.

SAM fue desarrollado en Software Libre y para su implantación se requiere:

Hardware: El Sistema no requiere de hardware con especificaciones determinadas. Sin embargo se recomienda no utilizar un computador con menos de 1GB de RAM, menos de 1.6GHz de velocidad de procesador, sin puerto de internet o sin conectividad WIFI.

Software: Para la instalación del servidor web: WAMP (Windows Apache 2.2.11, php 5.3 y MySQL 5.1.36) o LAMP ((Similar a WAMP pero con Linux como Sistema Operativo. Software de Soporte: Navegador web: Firefox (desde la versión 3, no menor) o Internet Explorer (IE7, no menor).

A continuación, se muestran algunas de las interfaces de SAM al momento de generar los escenarios y especificar los CP.

En la Figura 6, se muestra la pantalla de inicio de sesión. Después, se puede seleccionar un proyecto.



Figura 6. Pantalla de Inicio de Sesión.

Después de haber seleccionado el proyecto, se debe generar los escenarios a partir de los CU. En la Figura 7, se muestra cómo el Sistema permite importar CU (la narrativa de los CU) en formato XML desde archivos guardados en la computadora. Para utilizar esta opción se presiona el botón de *Cargar Casos de Uso* del menú desplegable de *Casos de Uso*.

Cuando ya se han importado CU al sistema, éste permite consultar los datos o narrativa de cada CU. Para ello, se debe seleccionar el ciclo de prueba asociado a los CU. Para esto se presiona el botón de *Ver casos de uso* del menú desplegable de Casos de Uso o la opción de *Ver casos de uso* del menú interno ubicado del lado izquierdo, como se muestra en la Figura 8.

Cuando ya se han importado los CU, se pueden generar los escenarios para cada CU como se puede apreciar en la Figura 9. Primero se presiona el botón de *Generar Escenarios*, ubicado en la parte inferior de la página de consultar los datos de un CU. Una vez que se selecciona la opción de *Generar Escenarios* se presenta una lista de los escenarios para dicho CU (se pueden mantener todos los escenarios o descartar algunos, dependiendo de los acuerdos a los que llegue el equipo de pruebas) y también se puede ver el grafo asociado al CU, similar a lo que se presenta en la Figura 1.

Para generar los CP, se deben seleccionar los Escenarios como se muestra en la Figura 10. Finalmente en la Figura 11, se muestra la especificación del CP.



Figura 7. Pantalla de Importar Casos de Uso.



Figura 8. Pantalla de Ver Casos de Uso.

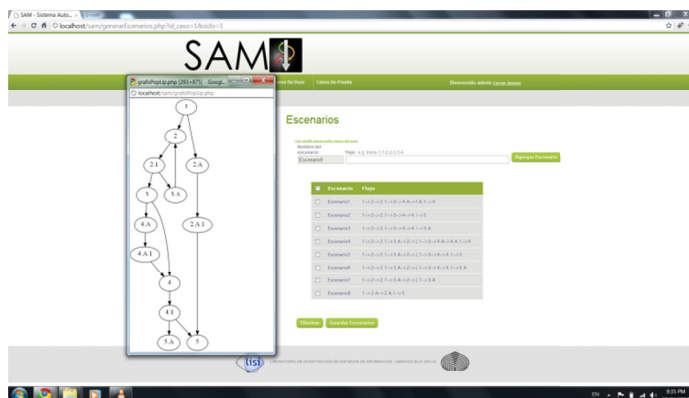


Figura 9. Pantalla de Generar Escenarios.

ID caso de prueba	Nombre del caso de prueba	Resultados esperados	Nivel de prueba	Tipo de prueba	Especificación de caso de prueba
SAM-01-00-01	Test	No se	1	12	Ver
SAM-01-00-02	PPrueba1	100% aprobados	Inicial	Básico	Ver

Figura 10. Pantalla de Generar Casos de Prueba.

Paso	Condición	Valor(es)	Resultado Esperado	Resultado Obtenido
1	hola	23	ajshdjkashdjkahnd	dasdadssdas
1	asasd	222222	asdc333	212edawcwax
1	asasd	3456	2345678	23456789

Figura 11. Pantalla de Especificación de Casos de Prueba.

5. Conclusiones y trabajo futuro

Este artículo presenta la herramienta SAM en una primera versión, la cual permite apoyar la ejecución del método MECAP de forma automatizada, facilitando una mejor gestión del proceso de las pruebas así como la trazabilidad entre diferentes disciplinas de Ingeniería de Software. Así mismo SAM permite apoyar la documentación de las pruebas y mejorar su efectividad.

En estos momentos se están probando el funcionamiento de SAM en el desarrollo de las pruebas de sistemas relacionados con el sector bancario para evaluar la calidad y completitud de los casos de prueba que son elaborados con su soporte. Se espera incorporar a corto plazo algunas funcionalidades que mejoren el Flujo de Trabajo de la Disciplinas de Pruebas como es el caso de los reportes de los ciclos de pruebas así como presentar el Workbench detallado.

6. Referencias bibliográficas

- Leffingwell D. & Widrig D. (2006). Managing Software Requirements, a Use Case Approach. Second Edition. Addison-Wesley, Pearson Education.
- Kruchten, P. (2000). The Rational Unified Process as Introduction. 2nd Edition. Addison – Wesley.
- Lewis W. 2000. Software testing and continuous quality improvement 000 by CRC Press LLC Auerbach is an imprint of CRC Press LLC.
- Méndez E., Pérez, M. & Mendoza, L. E. (2008). Improving Software Test Strategy with a Method to Specify Test Cases (MSTC). 10th International Conference on Enterprise Information Systems (ICEIS 2008). Barcelona, España. Junio 2008.
- Méndez, E. Pérez, M., & Mendoza, L. E. (2007). Aplicación de un Método para Especificar Casos de Prueba de Software en la Administración Pública. PRIS 2007: Taller sobre Pruebas en Ingeniería del Software. Libro: "Actas de Talleres de Ingeniería del Software y Bases de Datos (JISBD 2007)", Zaragoza, España.
- Perry W. 2006. Effective Methods for Software Testing. Wiley. Third Edition.
- Pinkster I., Burgt B., Janssen D. and Veenendaal E. 2006. Successful Test Management. Springer and Logicacmg.
- Pfleeger, S. (1998). Software Engineering.
- Pressman, R. (2002). Ingeniería del Software: Un enfoque Práctico. McGraw Hill.
- Sommerville, I. (2000). Software Engineering. Pearson Education.
- SWEBOK. 2004. Guide to the Software Engineering Body of Knowledge 2004 Version. IEEE Computer Society.
- UML Testing Profile Version 1.0 formal/05-07-07. This is a testing profile for UML 2.0.
- Utting M. and Legeard B. 2007. Practical Model-based Testing. Morgan Kaufmann and Elsevier Editorial.

Um Algoritmo Genético para Programação de Projectos em Redes de Actividades com Complementaridade de Recursos

Helder C. Silva ¹, José A. Oliveira ², Anabela P. Tereso ²

helder@ifam.edu.br, zan@dps.uminho.pt, anabelat@dps.uminho.pt

¹ IFAM – Instituto Federal de Educação Tecnológica do Amazonas, Manaus, Brazil

² University of Minho, Guimarães, Portugal

Resumo: Neste artigo abordamos a questão da alocação óptima dos recursos, mais especificamente, a análise da complementaridade dos recursos (recurso principal ou recurso-*P* e do recurso de suporte ou recurso-*S*) às actividades de um projecto. Neste artigo apresenta-se um algoritmo genético, baseado num alfabeto de chaves aleatórias, e analisa-se o procedimento para criar soluções a partir de um cromossoma.

Palavras-chave: Gestão de Projectos, Programação de Projectos, Complementaridade de Recursos.

Abstract: We address the issue of optimal resource allocation, and more specifically, the analysis of complementarity of resources (primary resource or *P*-resource and supportive resource or *S*-resource) to activities in a project. In this paper we present a Genetic Algorithm, based in a random keys alphabet, and we analyse the procedure to build solutions from a chromosome.

Keywords: Project Management, Scheduling, Complementarity of resources.

1. Introdução

A Gestão de Projectos é actualmente uma actividade importante em vários sectores da sociedade pós-industrial que inclui agora também a produção de informação. Torna-se fulcral ter-se uma clara percepção das diferentes fases do ciclo de vida de um projecto, dos processos, técnicas e ferramentas, adequados à gestão de um projecto, tendo em atenção a especificidade dos ambientes onde o projecto decorre e a dimensão e complexidade do mesmo.

Para o gestor de projectos é essencial a compreensão dos objectivos de cada projecto em relação a resultados expectáveis, âmbito, custos, tempo e níveis de qualidade e as interligações e dependências entre eles. Não menos importante é também o

entendimento da importância da vertente humana (liderança, equipas, conflitos, negociações...) no sucesso de um projecto.

No âmbito das tecnologias de informação e comunicação (TIC) a engenharia de software profissional está sujeita a restrições de orçamento e planeamento como as demais engenharias. As falhas ocorridas em vários sistemas estavam relacionadas com problemas de gestão da produção de software, nomeadamente a entrega atrasada, produtos não confiáveis, desempenho fraco e custos elevados.

A diferença entre a engenharia de software e outras engenharias destaca-se ao nível do produto, que se considera ser intangível, e ao nível do processo, dada a inexistência de um processo padrão. Normalmente os projectos de software de grande dimensão são frequentemente projectos que não se repetem, e com características únicas.

As principais actividades de gestão na engenharia de software estão relacionadas com a preparação de proposta do produto, a definição de objectivos e o modo como o projecto vai ser conduzido. O cálculo do orçamento do projecto, com a definição dos recursos necessários, bem como o planeamento e escalonamento do projecto, nomeadamente ao nível do estabelecimento de “*milestones*” (final de uma actividade) e “*deliverables*” (um resultado entregue ao cliente no fim de uma fase do projecto) são também actividades fundamentais na gestão de projectos informáticos.

Na gestão de recursos, a selecção e avaliação de pessoal terão em conta o orçamento estabelecido, a disponibilidade dos recursos e ainda o treino e competências específicas. Esta actividade é particularmente crítica, pois pode não ser possível seleccionar as pessoas ideais para trabalhar no projecto, porque o orçamento estabelecido pode não permitir a utilização de profissionais altamente especializados; concomitantemente os profissionais com a experiência apropriada podem não estar disponíveis. Os gestores têm que decidir dentro destas restrições de recursos, resultando que o planeamento de um projecto de software é uma actividade que consome considerável tempo de gestão.

A partição do projecto em tarefas e a estimação do tempo e recursos necessários para completar cada tarefa; a organização das tarefas num formato concorrente para fazer o melhor uso da força de trabalho; e a minimização das dependências entre as tarefas para evitar atrasos, são as funções do gestor de projectos que dependem fortemente da intuição e experiência. Deste modo, a gestão de projectos de software representa um desafio constante nas empresas.

O adequado controlo das várias variáveis de gestão tais como o âmbito, o custo, o prazo, a qualidade e o risco necessitam de uma base de técnicas e competências que não são fáceis de gerir. Uma perspectiva prática da gestão de projectos de software, pode ser adquirida nos modelos PMBOK (*Project Management Body of Knowledge* - <http://www.pmi.org/>) e PRINCE (*PRojects IN Controlled Environments* - <http://www.prince-officialsite.com/home/home.asp>).

A Figura 1 esquematiza um projecto composto por 12 tarefas, com as suas durações, dependências e respectivos *milestones*.

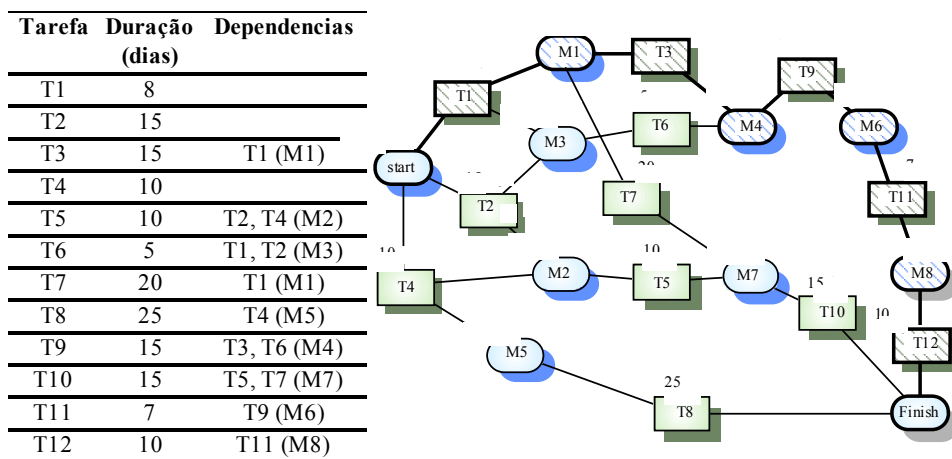


Figura 1: Projecto de Software

Este artigo apresenta uma abordagem para a alocação óptima de recursos, em redes de actividades, em condições de complementaridade de recursos, que ocorrem frequentemente na indústria do software, em projectos de engenharia de software.

O conceito de complementaridade, que tem sido discutido sob um ponto de vista económico (Kemer, 1993) pode ser incorporado no domínio da engenharia como um aumento da eficácia de um recurso principal (recurso-*P*) através da adição de um recurso de suporte (recurso-*S*). Aspectos relacionados com a melhoria do desempenho, redução da duração e aumento da qualidade, bem como o efeito do recurso de suporte no custo do projecto, têm sido apresentados por Silva, Tereso e Oliveira (2010).

A questão pode ser formulada como: Que quantidade de recursos principais (recursos-*P*) e de suporte (recursos-*S*) devem ser alocados às actividades do projecto, para atingirmos os melhores resultados, de forma mais económica?

2. Descrição do Problema

Considere uma rede de actividades nos vértices (nodos) (AoN) representada por $G=(N,A)$ com o conjunto de vértices $|N|=n$ (representando as “actividades”) e o conjunto de arcos $|A|=m$ (representando a relação de precedência entre as actividades). Em geral, cada actividade requer o uso simultâneo de vários recursos (Tereso, Araújo, & Elmaghraby, 2008), (Tereso, Araujo, Moutinho, & Elmaghraby, 2009), (Tereso, Araújo, Moutinho, & Elmaghraby, 2009b).

Existe um conjunto de recursos “principais”, denotados por P , com $|P|=\rho$. Tipicamente, um recurso principal tem várias unidades disponíveis (p.e. trabalhadores, máquinas, processadores, etc.) (Mulcahy, 2005). Adicionalmente, existe um conjunto de recursos de suporte, representados pela letra S , com $|S|=\sigma$ (tais como trabalhadores menos qualificados, ou computadores e dispositivos electrónicos, etc.)

que devem ser utilizados em conjunto com os recursos principais para aumentar o desempenho destes últimos. O número de recursos de suporte utilizados numa actividade varia em função dos recursos principais necessários para a execução dessa actividade. O impacto sobre o recurso- P é representado utilizando a variável $0 < v(r_p, s_q) \leq 1$ que indica a fracção pela qual o recurso- S (s_q) aumenta a performance do recurso- P (r_p). Tipicamente, $v(r_p, s_q) \in [0, 1 ; 0, 5]$.

Assume-se que o impacto dos recursos- S é aditivo: se um subconjunto $\{s_q\}_{q=1}^v$ de recursos- S é usado como suporte ao recurso- P r_p na actividade a , e somente uma unidade de cada recurso- S é utilizada, então o desempenho do recurso- P é aumentado para,

$$x_a(r_p, \{s_q\}_{q=1}^\sigma) = x_a(r_p) + \sum_{q=1}^\sigma v(r_p, s_q) \quad (1)$$

Com $w_a(r_p)$ representando o conteúdo de trabalho do recurso- P r_p na actividade a , o recurso principal $r_p \in P$ completaria a actividade a no tempo $y_a(r_p)$ (eq. 2). Se este recurso principal é melhorado pela adição do recurso de suporte, então o seu tempo de processamento é reduzido para $y_a(r_p, s_q)$ (eq. 3).

$$y_a(r_p) = \frac{w_a(r_p)}{x_a(r_p)} \quad (2)$$

$$y_a(r_p, s_q) = \frac{w_a(r_p)}{x_a(r_p, s_q)} \quad (3)$$

Uma actividade normalmente requer uma utilização simultânea de mais de um recurso- P para a sua execução. O problema coloca-se então do seguinte modo.

A que nível deve ser utilizado cada recurso principal e que recursos de suporte lhe devem ser adicionados para otimizar um dado objectivo?

O tempo de processamento de uma actividade é dado pelo valor máximo das durações que podem resultar de uma afectação específica a cada recurso (veja a discussão prévia sobre o cálculo da duração considerando múltiplos recursos em (Tereso, Araújo, & Elmaghraby, 2008) (Tereso, Araujo, Moutinho, & Elmaghraby, 2009), (Tereso, Araújo, Moutinho, & Elmaghraby, 2009b)).

$$y(a) = \max_{all\ r_p} \{y_a(r_p)\} \quad (4)$$

Para melhor se entender esta representação, considere-se um simples projecto em modo de representação AoN (Figura 2). Este projecto é constituído por três actividades, 1, 2 e 3. Para cada uma iremos assumir que é necessário a utilização de

quatro recursos-*P*; nem todos os recursos são requeridos por todas as actividades (veja Tabela 1).

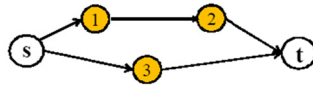


Figura 2: Projecto AoN com 3 actividades

Tabela 1: Conteúdo de trabalho (em homens-dia) das actividades do projecto 1

Recurso- <i>P</i> →	1	2	3	4
↓ Actividade/Disponibilidade →	2	1	3	2
A1	16	0	12	12
A2	0	7	10	8
A3	20	0	22	0

A relevância e o impacto dos recursos de suporte são representados na Tabela 2.

Tabela 2: A matriz P-S: Impacto dos recursos-*S* nos recursos-*P*

Recurso- <i>P</i> →		1	2	3	4
↓ Recurso- <i>S</i>	↓ Disponibilidade				
1	1	0,25	ϕ	0,25	ϕ
2	2	0,15	0,35	ϕ	ϕ

Neste problema considera-se o custo da utilização dos recursos, uma bonificação por se terminar mais cedo do que o previsto e uma penalização por se terminar mais tarde do que uma data prevista de término para o projecto. Foi já desenvolvido um modelo para minimizar o custo total do projecto, considerando que as actividades podem ser iniciadas logo que sejam sequenciáveis, se houver recursos principais suficientes para as iniciar (Silva, Tereso, & Oliveira, 2010). Alguns resultados foram também apresentados utilizando um procedimento baseado na análise da rede de actividades e no conceito estado (State Space) (Silva, Tereso, & Oliveira, 2010).

3. Algoritmo Genético

Uma vez que o Problema Job Shop (JSP) pode ser visto como um caso particular do problema da programação de projectos com restrições de recursos, do inglês “Resource Constrained Project Scheduling Problem” (RCPSP), iremos estender o Algoritmo Genético (AG) desenvolvido para o JSP (Oliveira, Dias, & Pereira, 2010) ao RCPSP, e particularmente para a programação de projectos em redes de actividades sob condições de complementaridade de recursos. O AG é baseado numa representação por

chaves aleatórias, a qual permite uma fácil reconfiguração para ser aplicada noutros problemas.

A simplicidade dos AG para modelar problemas complexos e a sua fácil integração com outros métodos de optimização foram os factores considerados para esta escolha. Inicialmente, o algoritmo proposto foi concebido para solucionar a versão clássica do JSP (Oliveira, 2007), mas é possível usar o mesmo método para solucionar outras variantes do JSP (Oliveira, 2006), ou neste caso, para solucionar também uma generalização do JSP que é o caso do RCPSP com complementaridade de recursos.

Uma das características que diferenciam os algoritmos genéticos convencionais é o facto de o algoritmo não tratar directamente com as soluções do problema, mas sim com uma representação da solução, o cromossoma. As manipulações do algoritmo são feitas sobre a representação e não directamente sobre a solução (Goldberg, 1989).

Representamos o problema da programação de projectos num gráfico AoN, porque é semelhante ao gráfico disjuntivo que é usado para representar o JSP (Roy & Sussmann, 1964). Uma actividade só pode ser iniciada se as precedentes estiverem concluídas e se o conjunto de todos os recursos principais necessários estiverem disponíveis. Um projecto tem restrições tecnológicas que determinam uma ordem específica para processar algumas das actividades e é necessário garantir que não haja sobreposição temporal, no processamento de tais actividades em termos de recursos comuns. Atendendo a esta característica, usamos o conceito de actividade sequenciável (uma actividade que pode ser iniciada) e em cada instante de decisão só é necessário escolher uma actividade do conjunto das actividades sequenciáveis. A escolha da actividade é dirigida pelo algoritmo genético respeitando os alelos existentes nos cromossomas que definem a prioridade de cada actividade.

Tradicionalmente, os algoritmos genéticos utilizavam cadeias binárias que formavam os cromossomas, que consistiam apenas em '0's' e '1's'. Alguns dos algoritmos genéticos mais recentes apresentam cromossomas mais específicos, para um dado problema, que resultam de um balanço do ganho de eficiência e da perda de alguma flexibilidade. Uma análise aos resultados mais recentes demonstra que o uso de cromossomas com valores reais frequentemente supera o uso de cromossomas binários. Outra alternativa de alfabeto ao código binário convencional é o código de Gray, que é também um sistema binário, onde dois valores sucessivos diferem em apenas um dígito (Goldberg, 1989).

O código de permutação era adequado para problemas de permutação. Neste tipo de representação, o cromossoma é um literal da sequência de operações nas máquinas. No caso JSP clássico, o cromossoma é composto por m subcromossomas, um para cada máquina, cada um composto por n genes, um para cada operação (Oliveira, 2007). O gene i do subcromossoma corresponde à operação processada na i -ésima posição na máquina correspondente. O alelo identifica o índice da operação no gráfico disjuntivo (Oliveira, 2007). Para as redes de actividades sob complementaridade de recursos, definimos um cromossoma com $n(\rho + \sigma + 1)$ genes. Para cada actividade o cromossoma estabelece a quantidade de cada recurso- P e a quantidade de cada recurso- S complementar. Além disso, o cromossoma indica a prioridade de cada actividade.

Neste trabalho é utilizado o alfabeto de chaves aleatórias apresentado por Bean (1994). A característica mais importante das chaves aleatórias é que todos os descendentes gerados pelo cruzamento são soluções admissíveis, quando é usada em conjunto com um procedimento construtivo da solução baseado em todas as operações disponíveis para sequenciamento, e a prioridade é dada pelo alelo da chave aleatória. Através da dinâmica do algoritmo genético, o sistema aprende a relação entre vectores aleatórios e as soluções com bons valores da função objectivo. Outra vantagem da representação de chave aleatória é a possibilidade de se utilizar os operadores genéticos convencionais. Esta característica permite o uso do algoritmo genético noutros problemas de optimização, apenas adaptando algumas rotinas relacionadas com o problema.

Um cromossoma representa uma solução para o problema e é codificado como um vector de chaves aleatórias (números aleatórios). Neste trabalho, de acordo com Cheng, Gen, e Tsujimura (1996), a representação do problema pode ser entendida como uma mistura de uma representação baseada em regras de prioridade e na representação das chaves aleatórias.

O algoritmo genético tem uma estrutura muito simples e pode ser representado no algoritmo da Figura 3. Ela começa com a geração da população e a sua avaliação. Atendendo à aptidão dos cromossomas, os indivíduos são seleccionados para serem progenitores. O cruzamento é aplicado e é gerada uma nova população temporária, que também é avaliada. Comparando-se a aptidão dos novos elementos e dos seus progenitores, a população anterior é actualizada.

```
begin  
P ← GerarPopulaçãoInicial()  
Avaliar(P)  
while condição de paragem não for alcançada do  
    P' ← Recombinar(P) // UX  
    Avaliar(P')  
    P ← Seleccionar(P ∪ P')  
end while
```

Figura 3: Algoritmo Genético

O Cruzamento Uniforme (UX) é usado neste trabalho. Este operador genético usa uma terceira sequência de números aleatórios e permuta os alelos de ambos progenitores se a chave aleatória é superior a um valor prefixado. A Figura 4 ilustra a aplicação de UX sobre dois progenitores (prnt1, prnt2) e troca os alelos, se a chave aleatória é maior ou igual a 0,75. Os genes 3, 4 e 16 são alterados o que dá origem a dois descendentes (dscndt1, dscndt2). O descendente 1 é semelhante ao progenitor prnt1, porque tem cerca de 75% dos seus genes.

<i>i</i>	<i>J1</i>				<i>J2</i>				<i>J3</i>				<i>J4</i>			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
prnt1	0,89	0,49	0,24	0,03	0,41	0,11	0,24	0,12	0,33	0,30	0,27	0,24	0,80	0,53	0,28	0,18
prnt2	0,83	0,41	0,40	0,04	0,29	0,35	0,38	0,01	0,42	0,32	0,28	0,28	0,72	0,61	0,44	0,13
randkey	0,64	0,72	0,75	0,83	0,26	0,56	0,28	0,31	0,09	0,11	0,37	0,25	0,23	0,05	0,40	0,76
dscndt1	0,89	0,49	0,40	0,04	0,41	0,11	0,24	0,12	0,33	0,30	0,27	0,24	0,80	0,53	0,28	0,13
dscndt2	0,83	0,41	0,24	0,03	0,29	0,35	0,38	0,01	0,42	0,32	0,28	0,28	0,72	0,61	0,44	0,18

Figura 4: O Cruzamento UX

4. Algoritmo Construtivo

As soluções são obtidas por um algoritmo, que é baseado no algoritmo de Giffler e Thompson (1960). Enquanto o algoritmo de Giffler e Thompson (1960) pode gerar todos os planos activos para o JSP, o algoritmo construtivo apenas gera o plano de acordo com o cromossoma. Como vantagens desta estratégia, salienta-se a menor dimensão do espaço de soluções, que no entanto inclui a solução óptima e o facto de o procedimento não produzir soluções impossíveis ou desinteressantes do ponto de vista da optimização. Por outro lado, uma vez que as dimensões do espaço de representação e do espaço de soluções são muito diferentes, esta opção pode representar um problema, porque muitos cromossomas podem representar a mesma solução.

O algoritmo construtivo tem n estágios e em cada estágio uma actividade é sequenciada. Para auxiliar na apresentação do algoritmo, considere a seguinte notação para o estágio t :

P_t - Plano parcial formado pelas $(t-1)$ actividades sequenciadas;

S_t - Conjunto de actividades sequenciáveis no estágio t , i.e. todas as actividades que devem preceder as actividades no conjunto S_t estão em P_t ;

δ_k - A data mais cedo na qual a actividade a_k em S_t poderia ser iniciada. Este tempo representa a conclusão de todos os precedentes de a_k e a disponibilidade de todos os recursos que a_k irá usar (recursos principais e recursos de suporte);

φ_k - A data mais cedo que a actividade a_k em S_t pode ser finalizada, que é $\varphi_k = \delta_k + y_k$;

A^* - A actividade onde $\varphi^* = \min_{a_k \in S_t} \{\varphi_k\}$;

S_t^* - O conjunto de conflito formado por $a_k \in S_t$ que tem $\delta_k < \varphi^*$.

a_k^* - A actividade seleccionada para ser sequenciada no estágio t .

O algoritmo construtivo das soluções é apresentado na Figura 5.

<i>Passo 1</i>	Seja $t = 1$ com P_1 sendo nulo. S_1 será o conjunto formado por todas as actividades que não têm predecessoras, ou seja pelas actividades ligadas ao vértice início.
<i>Passo 2</i>	Identificar $\varphi^* = \min_{a_k \in S_k} \{\varphi_k\}$ e identificar A^* . Formar S_t^* .
<i>Passo 3</i>	Seleccionar a actividade a_k^* em S_t^* com o maior valor de alelo de prioridade.
<i>Passo 4</i>	Passar ao próximo estágio por: (1) adição de a_k^* a P_t, criando P_{t+1}; (2) remoção de a_k^* de S_t, criando S_{t+1} por adição (se existirem) a S_t das actividades directamente sucessoras de a_k^* e que tenham todos os seus predecessores em P_{t+1}; (3) aumentar t em 1.
<i>Passo 5</i>	Se existir alguma actividade ainda por sequenciar ($t < N$), voltar ao <i>Passo 2</i> . Senão, Parar.

Figura 5: Algoritmo Construtivo

O formato utilizado é similar ao utilizado por Cheng, Gen e Tsujimura (1996), para apresentar o algoritmo de Giffler e Thompson (1960). No *Passo 3*, em vez de se usar uma prioridade estabelecida por uma regra de despacho, é usada a informação dada pelo cromossoma. Se o maior valor dos alelos for igual para duas ou mais actividades, então é escolhida aleatoriamente uma dessas actividades.

Considere o exemplo apresentado na Figura 2, Tabela 1 e Tabela 2, com três actividades (A_1, A_2, A_3). Nesta instância existem $\rho = 4$ recursos principais e $\sigma = 2$ recursos de suporte. Um cromossoma para representar uma solução para esta instância tem 21 genes, porque existem seis genes para cada actividade para estabelecer o número de elementos de cada recurso que serão usados, mais o gene A_k que define a prioridade da actividade. A Figura 5 apresenta o cromossoma de valores de chaves aleatórias desta instância. Os valores são gerados aleatoriamente entre 0 e 99.

A1	P1	P2	P3	P4	S1	S2	A2	P1	P2	P3	P4	S1	S2	A3	P1	P2	P3	P4	S1	S2
94	51	88	76	52	23	68	36	73	60	61	53	75	35	47	7	15	42	86	16	16

Figura 6: Um cromossoma

A actividade A_1 tem uma prioridade de 94, a maior, enquanto A_2 tem a menor prioridade (36). Para definir o número de elementos de cada recurso, utilizamos a Tabela 1. A disponibilidade do recurso- P_3 é de 3 unidades. Definimos três intervalos iguais entre 0 e 99. Para este recurso, se o alelo é um valor entre 0 e 32 é alocada 1 unidade, para valores entre 33 e 66, são alocadas 2 unidades e para valores de alelos entre 67 e 99, serão alocadas 3 unidades. Para estabelecer o número de unidades para os recursos de suporte, o procedimento é similar, mas também é incluída a possibilidade de se alocar o unidades de recursos de suporte, porque não é obrigatório

o uso destes recursos. Respeitando esta regra, o cromossoma apresentado na Figura 6 define as seguintes quantidades de recursos a serem utilizadas, que são apresentadas na Figura 7.

A1	P1	P2	P3	P4	S1	S2	A2	P1	P2	P3	P4	S1	S2	A3	P1	P2	P3	P4	S1	S2
Units	2	0	3	2	0	2	Units	0	1	2	2	1	1	Units	1	0	2	0	0	0

Figura 7: Quantidade de unidades de recursos a ser utilizada

São alocadas zero unidades de um recurso- P a uma actividade, se a actividade não utiliza esse recurso principal, que é o caso de P_2 na actividade A_1 , de acordo com a Tabela 1.

A alocação dos recursos de suporte para os recursos principais é feita respeitando a quantidade de trabalho existente depois da alocação dos recursos principais. A primeira unidade é alocada ao recurso- P com maior quantidade de trabalho. Após a alocação, a quantidade de trabalho é recalculada e então é feita a próxima alocação e assim sucessivamente. A actividade A_1 apresenta o conteúdo de trabalho, indicado na Tabela 3.

Tabela 3: Conteúdo de trabalho da Actividade A1 (em homens-dia)

P-resource →	1	2	3	4
A1	16	0	12	12

Alocando as unidades de recursos principais definidas pelo cromossoma a duração da actividade A_1 é a indicada na Tabela 4.

Tabela 4: Duração da Actividade A1 após afectação dos recursos principais

P-resource →	1	2	3	4
A1	8	0	4	6

A primeira unidade de recurso de suporte S_2 é alocada a P_1 . Recalculando a duração da actividade A_1 usando (3), tem-se os valores apresentados na Tabela 5.

Tabela 5: Duração da Actividade A1 após afectação de uma unidade de recurso de suporte

P-resource →	1	2	3	4
A1	7,442	0	4	6

A segunda unidade de recursos de suporte S_2 é alocada a P_1 . Recalculando a duração da actividade A_1 usando (3), tem-se os valores apresentados na Tabela 6.

Tabela 6: Duração da Actividade A1 após afectação de duas unidades de recurso de suporte

P-resource →	1	2	3	4
A1	6,957	0	4	6

Aplicando o mesmo procedimento, as durações de todas as actividades são apresentadas na Tabela 7.

Tabela 7: Duração das Actividades

P-resource →	1	2	3	4
A1	6,957	0	4	6
A2	0	5,185	4,444	4
A3	20	0	11	0

5. Exemplo Numérico

A Figura 8 ilustra os resultados da aplicação do algoritmo construtivo. Apresenta-se a evolução da programação do conjunto S_i e o correspondente tempo de início e o tempo de conclusão para cada actividade durante a execução do algoritmo construtivo. O gráfico de Gantt final do projecto também é apresentado e mostra a ocupação de todas as unidades de recursos.

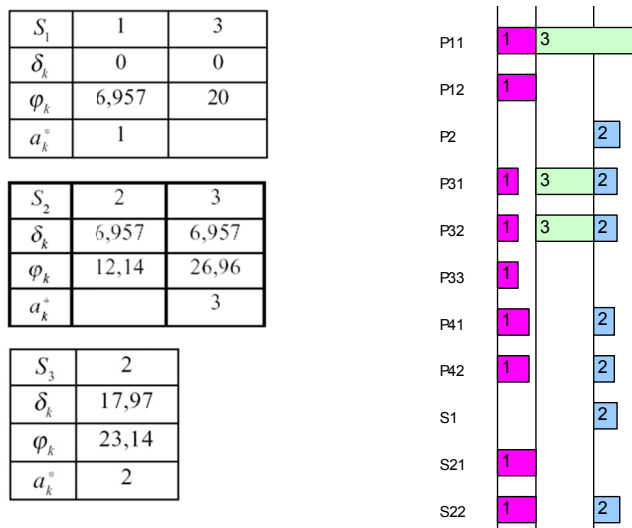


Figura 8: Execução do algoritmo construtivo

Considere a data final do projecto igual a 24 unidades de tempo, e os seguintes parâmetros:

- Cada unidade de recurso principal custa 4 unidades monetárias (UM) por unidade de conteúdo de trabalho;
- Cada unidade de recurso de suporte custa 1 UM por unidade de conteúdo de trabalho;
- O custo do atraso é de 60 UM por unidade de tempo;
- O custo (bónus) pelo término mais cedo é -40 UM por unidade de tempo.

O projecto termina no instante de tempo 26,96, com 2,96 unidade de atraso. O custo dos recursos é igual a 845 UM e o custo do atraso é de 177,39 UM. O custo total da solução é 1022,39 UM.

6. Conclusão

O objectivo deste artigo foi fornecer um modelo formal para alguns problemas não resolvidos na gestão de projectos, especialmente no que diz respeito à utilização de recursos de suporte, e apresentar a sua implementação num algoritmo genético. A importância do problema reside na oportunidade de se desenvolver um sistema que permita não só melhorar a afectação de recursos frequentemente escassos, mas também resultar numa redução das incertezas dentro dos projectos, combinada com o aumento da performance e com a redução do custo do projecto. O modelo foi inicialmente apresentado em Silva, Tereso e Oliveira (2010) mas faltava a sua implementação e aplicação a algumas redes de actividades, para demonstrar a sua validade. Depois foi apresentado o procedimento desenvolvido para solucionar o problema descrito pelo modelo matemático tendo sido aplicado a duas redes de actividades simples, obtendo assim os resultados desejados, através de uma implementação inicial em C (Silva, Tereso, & Oliveira, 2010).

Considerando a viabilidade do modelo proposto, acreditamos que ele pode fornecer ao utilizador uma nova opção de planeamento para determinar a melhor combinação de recursos e o menor custo do projecto, melhorando a capacidade de planeamento das empresas.

Este artigo apresenta uma estrutura para implementar um algoritmo genético para solucionar o problema do planeamento de projectos em redes de actividades em condições de complementaridade de recursos. O plano é construído usando informações fornecidas através do algoritmo genético para ordenar as actividades. Apresentamos um exemplo da aplicação do modelo e obtivemos os resultados preliminares para um pequeno projecto.

Na sequência deste estudo pretendemos testar o procedimento apresentado num conjunto de instâncias disponíveis na literatura de RCPSP às quais serão adicionados os recursos de suporte.

8. Referências

- Bean, J. (1994). Genetics and random keys for sequencing and optimization. *ORSA Journal on Computing*, pp. 154–160.
- Cheng, R., Gen, M., & Tsujimura, Y. (1996). A tutorial survey of job-shop scheduling problems using genetic algorithms - I. *Representation, Computers & Industrial Engineering*, (pp. 983-997).
- Giffler, B., & Thompson, G. L. (1960). Algorithms for solving production scheduling problems. *Operations Research*, 8, pp. 487-503.
- Goldberg, D. E. (1989). Genetic Algorithms in Search. *Optimization and Machine Learning*. Addison-Wesley, Reading, MA.
- Kemer, M. (1993). The O-Ring Theory of Economic Development. *The Quarterly Journal of Economics*, 108, 551-575.
- Mulcahy, R. (2005). *PMP, Exam Prep – Rita’s Course in a Book for Passing the PMP Exam Fifth Edition – For PMBOK Guide – Third Edition*. (5a Ed. ed.). USA: RMC Publications.
- Oliveira, J. (2006). A genetic algorithm with a quasi-local search for the job shop problem with recirculation. *Applied Soft Computing Technologies: The Challenge of Complexity*, (pp. 221-234). Springer, Berlin / Heidelberg.
- Oliveira, J. (2007). Scheduling the truckload operations in automatic warehouses. *European Journal of Operational Research*.
- Oliveira, J., Dias, L., & Pereira, G. (2010). Solving the Job Shop Problem with a random keys genetic algorithm with instance parameters. *2nd International Conference on Engineering Optimization*. Lisbon – Portugal.
- Roy, B., & Sussmann, B. (1964). Les problèmes d’ordonnancement avec contraintes disjonctives. Paris.
- Silva, H., Tereso, A., & Oliveira, J. (2010). On Resource Complementarity in Activity Networks - Further Results. *2nd International Conference on Engineering Optimization*. Lisbon - Portugal.
- Silva, H., Tereso, A., & Oliveira, J. (2010). On Resource Complementarity. *3rd International Conference on Information System, Logistics and Supply Chain*. Casablanca - Morocco.
- Tereso, A. P., Araujo, M., Moutinho, R., & Elmaghraby, S. (2009). Duration Oriented Resource Allocation Strategy on Multiple Resources Projects under Stochastic Conditions. *International Conference on Industrial Engineering and Systems Management*. Montreal - Canada.
- Tereso, A., Araújo, M., & Elmaghraby, S. (2008). Project management: multiple resources allocation. *International Conference on Engineering Optimization*. Rio de Janeiro - Brazil.
- Tereso, A., Araújo, M., Moutinho, R., & Elmaghraby, S. (2009b). Quantity Oriented Resource Allocation Strategy on Multiple Resources Projects under Stochastic

Conditions. *International Conference on Industrial Engineering and Systems Management*. Montreal - Canada.

Lecciones Aprendidas en la Implantación de MoProSoft en una empresa escolar: caso AvanTI

Angélica Astorga¹, Brenda Flores², Gloria Chavez¹, Mónica Lam¹, Araceli Justo¹

angelicaastorga@uabc.edu.mx, brenda.flores@uabc.edu.mx, gloria_chavez@uabc.edu.mx, monica.lam@uabc.edu.mx, araceli.justo@uabc.edu.mx

¹ Facultad de Ingeniería Campus Mexicali, Universidad Autónoma de Baja California, C.P. 21290, Mexicali, Baja California, México.

² Instituto de Ingeniería, Universidad Autónoma de Baja California, C.P. 21290, Mexicali, Baja California, México.

Resumen: Este artículo presenta las lecciones aprendidas del caso AvanTI, una empresa escolar incubada en el programa de estudios Licenciado en Sistemas Computacionales (LSC) de la Universidad Autónoma de Baja California (UABC), que durante tres años ha implantado el Modelo de Procesos para la Industria del Software en México (MoProSoft). AvanTI surgió, en el año 2007, con el propósito de responder a las demandas de la industria del software de Baja California, México. Los alumnos que han participado en AvanTI han aprendido sobre la Ingeniería de Software al poner en práctica la implantación MoProSoft, con el propósito de que al egresar posean el conocimiento y la habilidad de interpretación del modelo como una estrategia para apoyar a las MIPyMEs de la región².

Palabras clave: MoProSoft; Ingeniería de Software; Empresa escolar.

Abstract: This paper presents the lessons learned from the case AvanTI, an enterprise in incubation within the Computer Systems program of the Universidad Autonoma of Baja California (UABC), which for three years has implemented the Process Model for Software Industry in Mexico (MoProSoft). AvanTI emerged, in 2007, in order to respond the demands of the software industry in Baja California, Mexico. Students who have participated in AvanTI have learned about software engineering through the practical implementation of MoProSoft, such that upon graduation possessed both the knowledge and ability to interpret the model as a strategy to support Small and Medium Enterprises (SMEs) in the region.

Keywords: MoProSoft; Software Engineering; Enterprise in incubator.

² Versión extendida del artículo mejor evaluado y presentado en las Jornadas Iberoamericanas de Ingeniería del Software e Ingeniería del Conocimiento (JIISIC 2010)

1. Introducción

En el 2002, México a través de la Secretaría de Economía (SE) desarrolló el Programa de Fortalecimiento de la Industria del Software denominado ProSoft, con el fin de impulsar a este sector representado en un 90% por Micro, Pequeñas y Medianas Empresas (MIPyMEs) con niveles de capacidad de procesos menores a uno. Una de las estrategias para cambiar esta situación ha sido “Alcanzar niveles internacionales de capacidad de procesos”. Para ello, se decide crear un modelo propio para la industria mexicana conocido como Modelo de Procesos para la Industria del Software (MoProSoft) y que bajo su adopción les permita a éstas ser competitivas en el mercado internacional (PROSOFT, 2005). En el 2005, MoProSoft fue dado a conocer tanto a la industria del software como a las instituciones educativas. Durante la fase de presentación del modelo se llevaron a cabo una serie de capacitaciones a empresarios y docentes sobre la estructura del modelo y las estrategias para su implantación. Como resultado de estas capacitaciones, desde el año 2007, la carrera de Licenciado en Sistemas Computacionales (LSC), de la Facultad de Ingeniería Mexicali, de la Universidad Autónoma de Baja California (UABC), integró en el curso de Ingeniería de Software la enseñanza de MoProSoft con la finalidad de que los alumnos al egresar posean el conocimiento necesario para apoyar a las empresas a implantar el modelo. Para que el aprendizaje fuera significativo, se creó la empresa escolar AvanTI, acrónimo de “*Avanzando las Tecnologías de la Información*” en la que se ha establecido el enfoque de procesos propuesto por MoProSoft. Por otro lado, en paralelo a la conformación de la empresa se participó en la Convocatoria 2007 del programa de Servicio Social Profesional (SSP) de la UABC, con el propósito de brindar un servicio a la comunidad a través del desarrollo de un software en beneficio a una institución pública. Los resultados fueron positivos y de esta manera se asignó un proyecto real para la empresa AvanTI. En los años de 2008 y 2009 también se obtuvo el apoyo para dar continuidad a este proyecto ya que por su alcance ha requerido más de una liberación/*release*. Este documento está estructurado en 3 secciones. En la segunda sección se describe cómo se implantó MoProSoft dentro de AvanTI. En la tercera sección se presentan las conclusiones y lecciones aprendidas del caso de estudio.

2. Implantación de MoProSoft en AvanTI

La estructura de procesos de MoProSoft comprende 3 categorías y 9 procesos. Cada categoría contiene al menos un proceso (Tabla 1). Además, el modelo establece 5 niveles de capacidad de procesos: 1 “Realizado”, 2 “Administrado”, 3 “Establecido”, 4 “Predecible” y 5 “Optimizado”. Cada uno de ellos responde a un conjunto de Prácticas Base, Productos de Trabajo y atributos de proceso (Oktaba *et al*, 2005). Cabe mencionar que a partir del nivel 2 es necesario calificar el logro de atributos de proceso tomando como referencia la ISO 15504 contenida en EvalProSoft (Oktaba *et al*, 2004).

Tabla 1. Estructura de Procesos de MoProSoft

Categoría	Procesos
Alta Dirección(DIR)	Gestión de Negocio (GN)
Gerencia (GES)	Gestión de Procesos (GPR)
	Gestión de Proyectos (GPY)
	Gestión de Recursos (GR)
	Recursos Humanos y Ambiente de Trabajo (RHAT)
	Bienes, Servicios e Infraestructura (BSI)
Operación (OPE)	Conocimiento de la Organización (CO)
	Administración de Proyectos Específicos (APE)
	Desarrollo y Mantenimiento de Software (DMS)

2.1 Definición de la estructura organizacional

La empresa AvanTI fue implementada de tal manera que se pudieran cubrir los nueve procesos requeridos por MoProSoft (Oktaba *et al*, 2005). En la figura 1 se puede observar la estructura organizacional de AvanTI definida a partir del 2008 y debajo del nombre del departamento se relaciona el proceso implantado en el mismo. Regularmente en un curso de Ingeniería de Software el enfoque de los equipos de trabajo es cubrir las actividades básicas y de soporte del ciclo de vida del software, pero en este caso también había que incorporar las actividades de Dirección y Gerencia como en cualquier organización real. Para cubrir la implantación de todo el modelo se asociaron además de la unidad de aprendizaje de Ingeniería de Software la de Reingeniería de Procesos, Administración, Recursos Humanos, Base de Datos y Redes. Se implantaron las tres categorías de Alta Dirección, Gerencia y Operación. Cada categoría comprende uno o más departamentos. Inicialmente en el año 2007, existía sólo el departamento de Recursos Humanos y Materiales en donde se ejecutaban a la vez los procesos de RHAT y BSI.

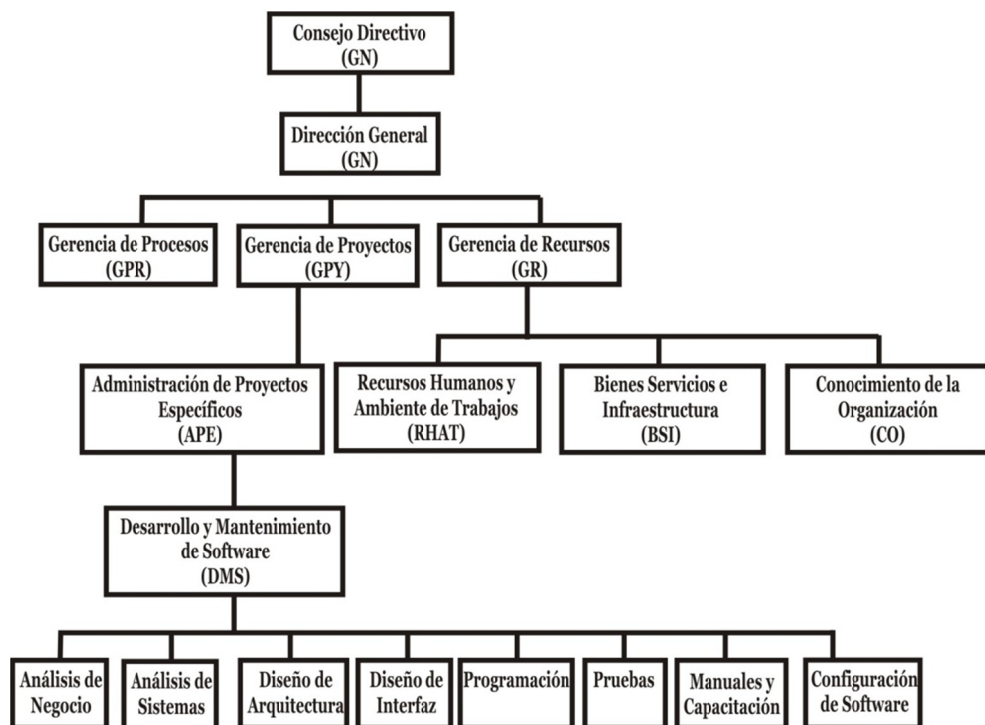


Figura 1. Estructura organizacional AvanTI en el 2008.

Asimismo, se contaba sólo con el departamento de Proyectos Específicos pues la persona Responsable de APE era la misma que desempeñaba el rol de Responsable de DMS. La decisión de separar los departamentos fue de no asignar a una sola persona la responsabilidad de más de un proceso ya que implicaba más actividades y tiempo que no podía cubrir.

Para la implantación se utilizó como guía la versión 1.3 MoProSoft por niveles de capacidad de procesos (Oktaba *et al*, 2005). Al tratarse de un primer acercamiento con la implantación de MoProSoft y los requisitos establecidos por los niveles de capacidad de procesos, el alcance fue el nivel 1 de capacidad de procesos “Realizado” sin excluir aquellas actividades de nivel 2 “Administrado” que benefician al logro de los objetivos de la empresa, pero sin someter a los procesos a los requisitos de los atributos de proceso de nivel 2. La estrategia de AvanTI fue implantar primero los procesos de la categoría de Alta Dirección, posteriormente los de Gestión y finalmente los de Operación. Para ello se tomaron en cuenta las lecciones aprendidas obtenidas durante el programa de Pruebas controladas (Alquicira & Su, 2005).

2.2 Categoría de Alta Dirección

La Alta Dirección está dirigida por el Consejo Directivo conformado actualmente por un grupo de 5 docentes y por el Responsable de GN, representado por un alumno. Es

importante destacar en esta experiencia que no sólo es un reto para los alumnos el participar creando un proyecto de software real, sino que también tienen que aplicar sus conocimientos en las áreas de negocio. Esta fue la decisión de considerar las unidades de aprendizaje de Administración y Recursos Humanos.

Gestión de Negocios (GN)

GN se ubicó como el primer proceso a implantar, pues a partir de éste se define el Plan Estratégico en el que se declara la misión, visión, valores, objetivos, procesos requeridos entre otros elementos necesarios para dirigir los esfuerzos de la empresa y motivar al grupo a continuar con la creación de AvanTI como un proyecto formal y no una unidad de aprendizaje más en la que sólo se persiguiera su aprobación o acreditación. El Plan Estratégico fue elaborado por la Dirección General y las tres Gerencias contando con la asesoría del Consejo Directivo. Una vez terminado este plan se les presentó a todos los miembros y se distribuyó a las Gerencias para su ejecución. En los años 2008 y 2009, al asignarse al nuevo Director (Responsable de GN) este recibe de su antecesor un informe del cumplimiento del Plan Estratégico para que se actualice de acuerdo a la situación actual de AvanTI.

2.3 Categoría de Gerencia

Gestión de Procesos (GPR)

Con base en los procesos requeridos por AvanTI y determinados en el Plan Estratégico, el departamento de GPR se dio a la tarea de elaborar y seguir el Plan de Procesos para llevar a cabo la capacitación tanto de los Responsables de Procesos como a su personal. Esta capacitación apoyó a los alumnos a identificar el (los) rol(es) que deseaban desempeñar y el perfil requerido para ese rol en el modelo; además de concientizar a los responsables e involucrados de los procesos no sólo buscar obtener los productos, sino considerar que éstos son el resultado de las prácticas base de los procesos (Figura. 2).

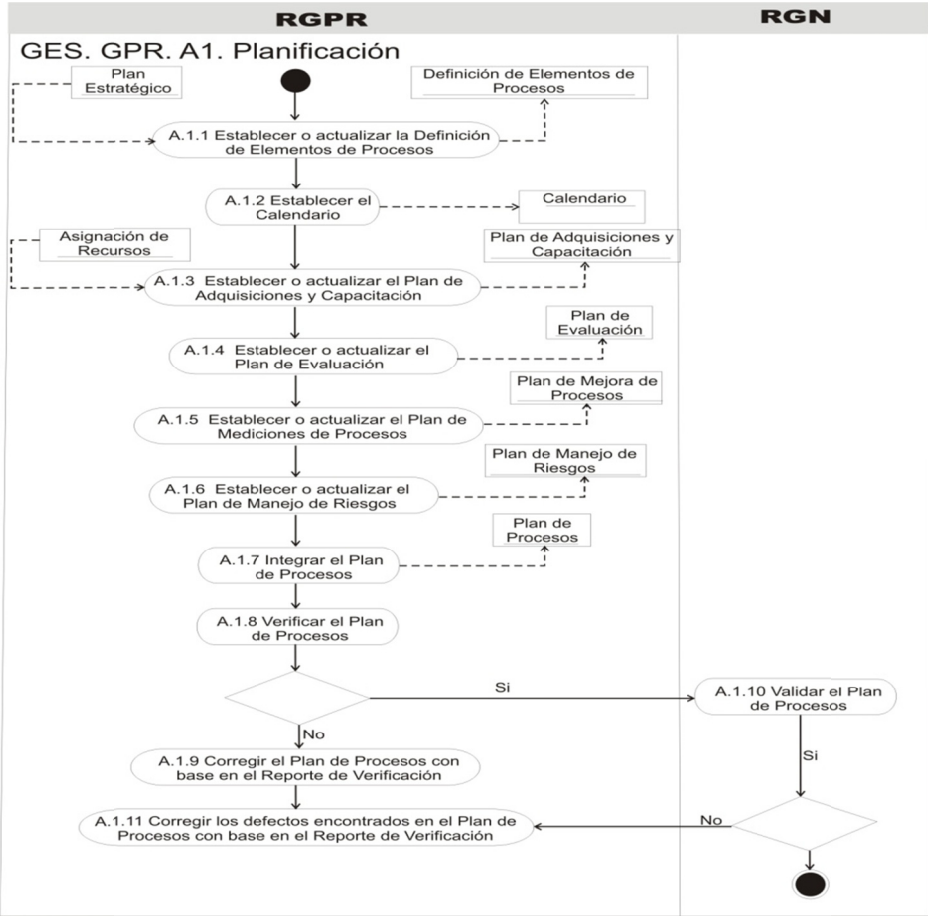


Figura 2. Diagrama de actividades correspondiente a GPR en la práctica base A1. Planificación.

Una vez capacitados se inició la implantación con la documentación de cada proceso. En este caso no hubo ajustes significativos a los procesos por lo que no se requirió de una nueva capacitación. Durante la definición e implantación se observó que este proceso es crítico para la implantación de los demás procesos (Astorga, 2007). El primer año los Responsables de Procesos iniciaron atendiendo las prácticas base correspondientes al nivel 1 y algunas de nivel 2. Por cada práctica base se identificaron los productos de trabajo que se debían obtener (Tabla 2).

Tabla 2. Prácticas base y productos por proceso correspondiente al nivel 1

Proceso.Práctica Base	Producto
GN.Planificación	Plan Estratégico
	Plan de Adquisiciones y Capacitación

Proceso.Práctica Base	Producto
GPR.Planificación	Plan de Procesos
GPR.Preparación	Documentación del Proceso
GPY.Planificación	Plan de Adquisiciones y Capacitación
	Plan de Gestión de Proyectos y Ventas
	Plan de Proyectos
	Plan de Ventas
	Alternativas de Realización de Proyectos Internos
GPY.Realización	Contrato
	Registro de Proyecto
	Descripción del Proyecto
	Responsable de Administración del Proyecto Específico
GR.Planificación	Plan de Adquisiciones y Capacitación
	Plan Operativo de RHAT
	Plan Operativo de BSI
	Plan Operativo de CO
	Plan de Capacitaciones
GR.Instrumentación	Registro de Recursos Humanos
BSI.Preparación	Plan de Mantenimiento
	Solicitud de Bienes y Servicios
BSI.Instrumentación	Registro de Bienes y Servicios
CO.Planificación	Plan de Administración de la Base de Conocimiento
CO.Realización	Diseño de la Base de Conocimiento
APE.Planificación	Plan de Proyecto
	Plan de Desarrollo
APE.Realización	No Aplica
APE.Cierre	Aceptación del Cliente
DMS.Inicio	No Aplica
DMS.Requisitos	Especificación de Requisitos
DMS.Análisis y Diseño	Análisis y Diseño
DMS.Construcción	Componente
DMS.Construcción	Software
DMS.Integración y pruebas	Manual de Usuario
DMS.Integración y pruebas	Manual de Operación
DMS.Cierre	No Aplica

Asimismo se identificaron los elementos requeridos en cada uno de los productos. La identificación de estos elementos también permitió generar las plantillas necesarias para su documentación. En la tabla 3 se muestra un ejemplo de los elementos del producto de trabajo Plan de Procesos.

Tabla 3. Elementos del producto Plan de Procesos

Proceso.Práctica base	Producto de Trabajo	Elementos
GPR.Planificación	Plan de Procesos	Definición de elementos Calendario Plan de Adquisiciones Capacitación Plan de evaluación Plan de manejo de riesgos

La creación de estas plantillas facilitó la comprensión de los elementos solicitados en cada producto de trabajo, eliminando dudas en el propósito de cada elemento. En los siguientes cursos cada vez que se incorpora el nuevo Responsable de GPR actualiza y da continuidad al Plan de Procesos para lo cual considera los resultados de las evaluaciones anteriores con el fin de dar seguimiento a los hallazgos encontrados. Estas evaluaciones se han realizado utilizando los criterios de calificación de EvalProSoft (Oktaba *et al*, 2004).

Gestión de Recursos (GR) y sus subprocesos

Los siguientes procesos implantados fueron GR y sus subprocesos Recursos Humanos y Ambiente de Trabajo (RHAT), Bienes, Servicios e Infraestructura (BSI) y Conocimiento de la Organización (CO). Estos procesos han sido importantes ya que a través de ellos se proveen los Recursos Humanos, materiales y la base de conocimiento (BC) a toda la empresa. Para ello, ha sido clave la elaboración de los Planes Operativos ya que han permitido definir los criterios para que los responsables de estos procesos puedan llevar a cabo entre otras actividades la selección, aceptación y asignación del personal, la evaluación del ambiente de trabajo, la evaluación de desempeño, la adquisición y asignación de BSI, así como el diseño y administración de la BC para garantizar la disponibilidad de los productos de trabajo.

Además de los productos de nivel 1 se añadieron la Encuesta de Ambiente de Trabajo y Evaluación de Desempeño, ambos productos de nivel 2 correspondientes al proceso de RHAT. La encuesta apoyó a medir el grado de satisfacción que los alumnos tenían al pertenecer a la empresa considerando los criterios de comunicación en el área, relación con los compañeros, relación con el jefe inmediato, capacitación brindada, horario de trabajo, importancia del equipo de trabajo, entre otros. La evaluación de desempeño es equivalente a un 80% de la calificación asignada a las unidades de aprendizaje y que de acuerdo al Plan Operativo de RHAT son los jefes inmediatos quienes evalúan a su personal retroalimentando con ellos los resultados en cuanto a conocimiento del

puesto, productividad, habilidades de comunicación, equipo de trabajo y habilidades de dirección en aquellos puestos que tienen personal a cargo. Los resultados de la evaluación han mostrado que la mayoría de los alumnos asumen con responsabilidad y compromiso el formar parte de una empresa aun cuando ésta sea un ejercicio escolar.

AvanTI no cuenta con una infraestructura propia pero se consideran como parte de ésta los salones y laboratorios de cómputo en donde se imparten las clases y talleres. Para la BC se otorgó un espacio en uno de los servidores de la carrera.

Gestión de Proyectos (GPY)

La implantación de este proceso permitió establecer una comunicación formal con el cliente a quien se le iba a desarrollar el proyecto de software. Cabe mencionar que los alumnos antes de ingresar a la unidad de aprendizaje de Ingeniería de Software ya han llevado el curso de Análisis y Diseño de Sistemas de Información, por lo que tienen las bases, más no la suficiente experiencia en el desarrollo de software para un cliente real.

A partir de la Descripción del Proyecto se identificó que éste era bastante extenso y con cierto grado de complejidad. La carrera trabaja por semestres con un total de 14 semanas. Por lo que las fechas de inicio y fin del calendario están sujetas a estos periodos. Esto determinó que se tendría que desarrollar más de una liberación/release durante más de un semestre para cubrir el alcance total de proyecto.

2.4 Categoría de Operación

Para esta categoría se implementó como modelo específico el Proceso Unificado de Rational (RUP por sus siglas en inglés). Este modelo comprende un conjunto de flujos básicos (ciclo de vida) y de soporte (administración). En cada flujo de trabajo se determinan los productos, roles y actividades a realizar (Jacobson, Booch & Rumbaugh, 2000). Mientras que MoProSoft responde al ¿Qué al definir los procesos en las organizaciones?, RUP responde al ¿Cómo al establecer una guía para la ejecución de los proyectos?. En la figura 3 se muestra la correspondencia que se estableció entre ambos modelos.

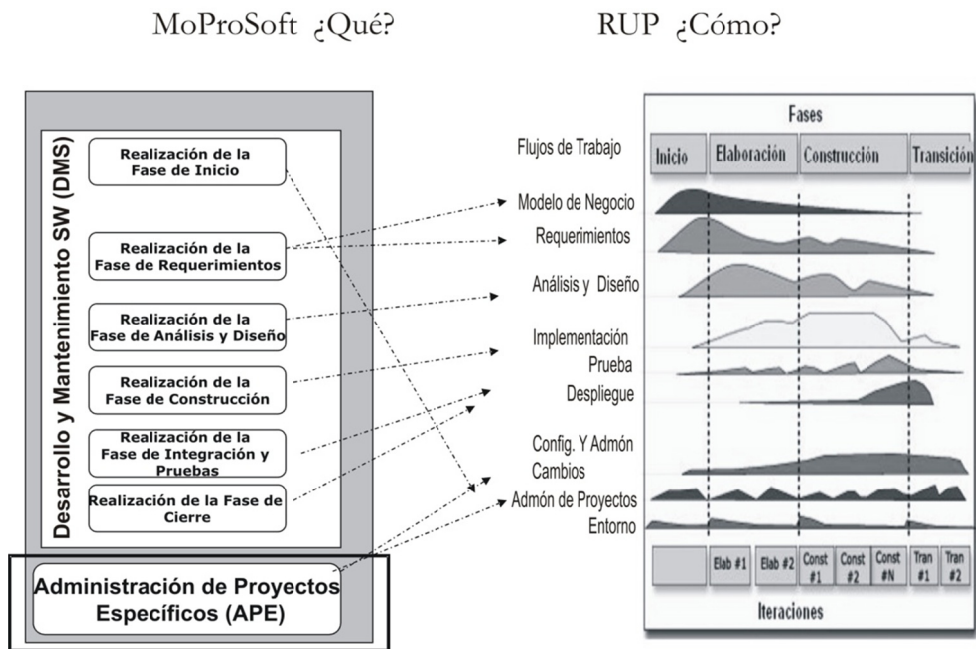


Figura 3. Correspondencia entre MoProSoft y RUP.

Las unidades de aprendizaje asociadas a toda la categoría son Ingeniería de Software I, Reingeniería de Procesos, Base de Datos y Aplicaciones de Ingeniería de Software.

Administración de Proyectos Específicos (APE)

En este proceso se estableció el Plan de Proyecto definiendo el ciclo de vida de desarrollo basado en el proceso de DMS de MoProSoft y en el RUP. APE es equivalente a Administración de Proyectos (Project Management). En el 2007, considerando que en la guía de ajuste de este proceso, se establece que cuando el RAPE y el RDMS son desempeñados por la misma persona, entonces se considera el Plan de Proyecto como el documento guía. El RUP utiliza el Plan de Desarrollo de Software por lo que se aplicó esta plantilla en la que se documentaron los entregables, equipo de trabajo, ciclo de vida de desarrollo definiendo las iteraciones y *releases* que fueron calendarizados con base en los periodos de tiempo estipulados en el semestre. A partir del 2008, al asignarse un Responsable a APE y uno a DMS se han elaborado ambos documentos.

Desarrollo y Mantenimiento de Software (DMS)

Los ciclos y actividades se han resuelto apoyados con el RUP estableciendo una relación con las prácticas base (fases) de MoProSoft.

1. Fase de Inicio

La realización de la fase de inicio se relacionó con la Administración de proyectos pues

su fin es revisar el Plan de Desarrollo con el equipo de trabajo para iniciar con el proyecto. Así como llevar un seguimiento de las actividades definidas con base en el ciclo de actividades asignadas a los roles correspondientes, quienes tienen que registrar el avance de sus tareas en un Reporte de Actividades que es entregado semanalmente al Responsable de DMS.

2. Fase de Requerimientos

Esta fase corresponde a la actividad de Modelo de negocios y de Requerimientos del RUP. Para identificar los procesos involucrados con el área de negocio en estudio, se asoció la unidad de aprendizaje Reingeniería de Procesos, a partir de la cual se analizaron con el cliente y usuarios las actividades realizadas, las áreas involucradas, los documentos y sistemas utilizados. De esta manera se validó el flujo de trabajo actual y se determinó el alcance del proyecto. Los roles involucrados en esta actividad fueron los Analistas de Negocio, rol que no se sugiere en MoProSoft pero que se integró siguiendo la metodología del RUP. En cuanto a los alumnos se pudo observar como se fueron familiarizando con los conceptos de la organización en estudio. Una vez que se analizó y aprobó el modelo de negocio entonces los Analistas de Sistemas empezaron a recabar la información para obtener la especificación de requerimientos a través de la elaboración del documento de visión, modelo de casos de uso, especificación de casos de uso, glosario y especificaciones suplementarias. También se trabajó un prototipo de interfaz con la participación de los Diseñadores de Interfaz siguiendo una guía de diseño enfocada principalmente en darle el control al usuario sobre la aplicación, reducir la carga de memoria y mantener una interfaz consistente (Pressman, 2006). La elaboración del manual de usuario se pospuso a la fase de construcción hasta tener en los *releases* una interfaz completamente aprobada por el cliente y los usuarios. Su elaboración está a cargo del Responsable de Manuales.

3. Fase de Análisis y Diseño

Esta fase corresponde a la actividad de Análisis y Diseño del RUP, en la que se elaboraron las realizaciones de casos de uso a nivel de diseño planteando la arquitectura en subsistemas y estos a su vez en componentes (Oktaba *et al*, 2005). Se eligió el lenguaje de programación Java el cual se imparte en las asignaturas de Programación Orientada a Objetos I y Programación Orientada a Objetos II, lo que significaba una familiaridad de los alumnos para la construcción, así como por ser un software de uso libre. También se describió el detalle de la arquitectura con el fin de que las decisiones tomadas queden documentadas para los siguientes *releases* y para que el nuevo personal que se incorpore en próximos semestres al proyecto obtenga la información necesaria para cualquier mantenimiento. Para el modelo de datos se utilizó el administrador de base de datos MySQL, que al igual que Java, es un software de uso libre que se imparte en las unidades de aprendizaje de Base de Datos, mismas que son cursadas por los alumnos, previo al proyecto. Los roles involucrados son los Diseñadores de Arquitectura, quienes por la tecnología anteriormente mencionada deben haber cursado estas unidades de aprendizaje.

4. Fase de Construcción

Esta fase corresponde a la Implementación y Pruebas del RUP. Los roles responsables son los Programadores y Probadores. En ella se codifican los componentes de acuerdo al plan de iteraciones y *releases*. La codificación se ajustó a la guía de programación de Java. Por otro lado, los alumnos han reutilizado utilerías y componentes de terceros para optimizar su codificación y desempeño. Aun cuando el nivel de capacidad de proceso era 1 “Realizado”, se han adoptado actividades de nivel 2, como es el caso de las pruebas ya que se tiene que asegurar que los componentes estén libres de fallos, así que una vez que se van liberando los componentes los Responsables de Prueba van haciendo las pruebas de unidad utilizando para ello casos de prueba y/o reportes de verificación.

5. Fase de Integración y Pruebas

Esta fase tiene correspondencia a la actividad de Pruebas y Despliegue del RUP, en la que se van integrando los componentes que previamente han concluido con éxito sus pruebas unitarias para conformar los subsistemas de acuerdo al plan de iteraciones. En esta fase los roles involucrados son el Programador, Responsable de pruebas y Responsable de manuales que se encargan de confirmar que los componentes trabajen exitosamente de manera conjunta. La participación del Responsable de manuales es porque en esta fase el Manual de Usuario puede actualizarse con más certeza ya con los últimos cambios realizados. La correspondencia con el despliegue es principalmente por el empaquetamiento e instalación del producto.

6. Fase de Cierre

Esta fase corresponde a la actividad de Despliegue del RUP, y los productos requeridos son de nivel 2. Para el proyecto se elaboraron el Manual de Usuario, Manual de Operación pero no el Manual de Mantenimiento. En el Reporte de Actividades quedaron especificadas aquellas actividades que principalmente se tenían que realizar a partir de los fallos encontrados en los componentes. En esta fase se concentra la configuración del software y los productos obtenidos son enviados al Responsable de APE para su revisión e instalación, y obtener posteriormente la aceptación del cliente.

3. Conclusiones

La implantación de la empresa escolar AvantI ha significado un reto tanto para los docentes como para los alumnos de la UABC, sobre todo durante el primer año en el que se trabajó tanto en la implantación de MoProSoft, como en el proyecto de desarrollo de software. Algunas de las lecciones aprendidas recabadas en los 3 años de implantación son: en el proceso de GN los alumnos realizan un plan estratégico que ellos mismos deben seguir para cumplir con sus objetivos, no obstante se ha identificado que en los procesos de GR, RHAT y BSI hay que reforzar en el alumnado la importancia de estas actividades en el mismo sentido que tienen las tecnológicas. Se detectó que cuando un alumno es responsable de más de un proceso o se le asigna más de rol implica responsabilidad y tiempo que no pueden cubrir ya que entre otros

factores están cursando otras unidades aprendizaje que no están relacionadas con el proyecto. También hace falta explorar mecanismos que permitan hacer más fácil la transferencia de conocimientos al hacerse el cambio de responsables para que no se presente pérdida o fuga de conocimiento. CO, tratándose de un nivel 1 de capacidad de procesos, ha cubierto la necesidad de un repositorio que permite alimentar y obtener los productos, pero carece de la suficiente administración al ser un servidor que se comparte con otros cursos. GPR ha alcanzado la implantación de los 9 procesos; pero tal y como ocurre en las empresas reales para que se observen resultados se requiere dedicación, factor que no ha podido ser controlado tampoco en este ejercicio. Todavía se requieren de ciertos productos genéricos para mostrar mejor evidencia de la implantación de los procesos. GPY, APE y DMS son los procesos que por su naturaleza se les presta más atención, pues de ellos depende la satisfacción del cliente. Otra lección en estos procesos ha sido la falta de experiencia de los alumnos en el desarrollo de proyectos de software y más siendo un proyecto real, por lo que, en algunos semestres ha sido más lento el avance para la liberación de los releases. El reto ha sido tener alineados los objetivos de estos procesos con los de negocio. Asimismo, durante este tiempo se ha observado la madurez y capacitación que han adquirido aproximadamente 60 alumnos. Esto les ha brindado seguridad para integrarse en el mercado laboral local con el convencimiento de que el éxito de una empresa depende en gran medida de la adopción de modelos como MoProSoft.

Además de que se da respuesta al Diagnóstico Externo realizado, durante el 2008, a empresarios del Comité de Vinculación Escuela-Empresa de la ciudad de Mexicali, Baja California para la modificación de la carrera (LSC) (UABC, 2008). AvantI, continuará integrando nuevas generaciones de alumnos, por lo pronto, se tiene estimado que la terminación del proyecto de software actual sea en el ciclo escolar 2010-1 y que en ese mismo ciclo se alcance completamente el nivel 1.

Referencias bibliográficas

- Alquicira, C. y Su Ramos, A. (2005). Pruebas controladas de MoProSoft, *Revista Software Guru*, Conocimiento en Práctica, Año 1, No. 1, México, pp. 28-30.
- Astorga Vargas, M. A (2007). Caracterización de los factores de cambio para que las PYMES dedicadas al desarrollo de software implanten MoProSoft. Caso de estudio. *Tesis de maestría*. Universidad Autónoma de Baja California. México.
- Jacobson, I., Booch, G. y Rumbaugh, J. (2000). *El Proceso Unificado de Desarrollo de Software*. Madrid: Editora Pearson.
- Oktaba, H., Alquicira, C., Su Ramos, A., Palacios, J., Pérez, C. y López, F. (2004). Método de Evaluación de procesos para la industria del software EvalProSoft, Versión 1.1, México.
- Oktaba, H., Alquicira, C., Su Ramos, A., et al. (2005). Modelo de Procesos para la Industria de Software, MoProSoft: por niveles de capacidad de procesos. Versión 1.3. México.

Pressman, R. (2006). Ingeniería de Software: un enfoque práctico. 6ta Edición, México: Editora McGrawHill.

PROSOFT (2005). Programa para el Desarrollo de la Industria de Software. Secretaría de Economía. 3er. Trimestre. México

UABC (2008). *Diagnóstico externo del plan de estudios de la carrera de Licenciado en Sistemas Computacionales*. Facultad de Ingeniería Mexicali. Universidad Autónoma de Baja California. México.

Experiencia en la Mejora de Procesos de Gestión de Proyectos Utilizando un Entorno de Referencia Multimodelo*

Cuevas Gonzalo¹, Mejia Jezreel¹, Muñoz Mirna¹, San Feliu Tomás¹

{gonzalo.cuevas, ma.munoz, tomas.sanfeliu}@upm.es, jmejia@mpsei.fi.upm.es

¹ Departamento Lenguajes y Sistemas Informáticos e Ingeniería del Software Universidad Politécnica de Madrid, Facultad de Informática 28660 Boadilla del Monte, Madrid, España.

* Los autores están ordenados alfabéticamente.

Resumen: Aun cuando muchas organizaciones están motivadas para realizar mejoras en sus procesos, pocas son las que saben cuál es la mejor manera de llevarla a cabo. El resultado de esta falta de conocimiento, se refleja principalmente en los fracasos que muy a menudo se tienen en los esfuerzos de mejora, dejando a los implicados en la mejora agotados, frustrados y más convencidos que nunca, en seguir haciendo su trabajo como lo venían haciendo hasta ahora. Como consecuencia crece cada vez más la resistencia al cambio que supone la implementación de mejoras en los procesos de una organización. Este artículo analiza los resultados de la experiencia de la mejora de procesos de los procesos de gestión de proyectos de una organización mediante una metodología que permite el uso de un entorno de referencia multimodelo.

Palabras clave: entornos multimodelo, mejora de procesos, gestión de proyectos.

Abstract: Although many organizations are motivated to improve their software processes, very few know how best to do so. The result of this lack of knowledge is mainly reflected in the failures that often have the improvements efforts. These failures leave stakeholders exhausted, frustrated and more convinced than ever that they must continue doing their work as so far. As consequence the resistance to change in software process improvement increases. This paper analyses the results of the implementation of a project management software process improvement in an organization by using a methodology that allows to use a multi-model environment as reference.

Key words: multi-model environment, software process improvement, project management.

1. Introducción

Actualmente, para una organización es crucial aumentar su ventaja competitiva con respecto a sus competidores para garantizar su supervivencia en el mercado. Por ello mejorar la forma de desarrollar software a través de sus procesos software es un desafío crítico para ellas, ya que, el software es el núcleo de cualquier servicio o producto moderno.

Con el fin de aumentar la capacidad de sus procesos para el desarrollo de software, el proyecto de mejoras de los procesos software es uno de los esfuerzos más críticos e importantes que cualquier organización de software debe perseguir.

Sin embargo, para la mayoría de las organizaciones, la mejora de procesos es indisciplinada, caótica y completamente impredecible, debido principalmente a que las iniciativas de mejora de proceso no son implementadas con éxito (Goldenson. 2007) o tienen éxito limitado (Conradi, et al. 2002), siendo el principal problema las dificultades a las que se enfrenta la organización al adaptar el modelo de mejora de procesos elegido a su situación real (Morgan. 2007).

El estudio de Goldenson y Herbsleb (Goldenson. 2007) indica que el 26% de las organizaciones que invirtieron en la mejora de procesos, no obtuvieron un cambio significativo después de la mejora, y que el 49% de las organizaciones se declararon desilusionadas por la falta de visibilidad en la mejora. Para vencer estas dificultades con el cambio, es necesario conocer las resistencias existentes e introducir los elementos necesarios para su disminución.

Aún cuando existen muchos modelos y estándares disponibles en el mercado para apoyar a las organizaciones en las mejoras de sus procesos, éstos son incapaces de abordar los desafíos críticos de una compañía de desarrollo de software, generando la falta de interés, de compromiso y de orientación hacia el cumplimiento de sus objetivos (Conradi, et al. 2002).

Esto confirma que las organizaciones han tratado de utilizar los modelos y estándares existentes como herramientas mágicas para el diseño de sus procesos, dando como resultado que en las organizaciones realmente no se utilizan los nuevos procesos resultantes, ya que son percibidos por los usuarios de los procesos como los procesos de alguien, que en absoluto reflejan su forma de trabajo (Pries-Heje, et al. 2010).

En este contexto, Niazi et al. (Niazi, et al. 2006) y Mahmood et al. (Mahmood, et al. 2008), mencionan que la investigación enfocada en la aplicación efectiva de los modelos y normas es mínima, lo que da como resultado que los esfuerzos de mejora en las organizaciones tengan éxitos limitados.

En este artículo se muestra como a través de la incorporación de mejores prácticas externas basadas en un entorno de referencia multimodelo acorde a la forma de trabajo de la organización, permite la adopción de mejores prácticas de forma justificada conforme a la capacidad de adopción del cambio de la organización, disminuyendo la resistencia al cambio que pueda surgir en las organizaciones por parte de los implicados con la mejora.

Este artículo está estructurado como sigue: la sección 2, revisa el estado de la cuestión de entornos multimodelo; la sección 3, presenta la experiencia en la mejora de los

procesos de gestión de proyectos de una organización; la sección 4, analiza los resultados obtenidos; y la sección 5 muestra las conclusiones.

2. Estado de la Cuestión de Entornos Multimodelo

El objetivo de esta sección es en primer lugar introducir al tema de entorno de referencia multimodelo.

2.1. Entorno de Referencia multimodelo.

Hoy en días muchas organizaciones se están enfrentando a las presiones del mercado para responder a más de un conjunto de modelos y técnicas de mejora (SuZ, G., 2007). De acuerdo a Lisa Marino y John Morley (Marino, L., John, M., 2009), basándose en la experiencia profesional de un total de 80 gestores, profesionales de la mejora de procesos, profesionales técnicos e investigadores de 40 organizaciones representando al Departamento de Defensa de Estados Unidos (DoD), agencias gubernamentales, consultoras, negocios comerciales, universidades y centros de investigación, tres de cada cinco grandes organizaciones ya están enfrentando los retos de la utilización de múltiples tecnologías para lograr la satisfacción del cliente, la rentabilidad del negocio, la cuota de mercado, la calidad del producto y del servicio y, la reducción de costos.

Por esta razón surge como resultado de un esfuerzo común de las organizaciones de todo el mundo la integración de las normas internacionales y modelos, en un esfuerzo por lograr un proceso de mejora de software exitoso (Urs, et al. 2009), (Mogilensky. 2009).

2.2. Mejora de procesos y entorno de referencia multimodelo

La mejora de procesos multimodelo se caracteriza por un enfoque armonizado y unificado para la mejora de procesos a través de la implementación de múltiples modelos (Siviy, J. M., et al. 2008), (Srivastava, et al. 2009).

De acuerdo a Lawrence y Becker (Lawrence, J., Becker, N. 2009) los principales objetivos son reducir la redundancia, integrar la mejora, crear sinergia, promover las buenas prácticas y crear un entorno de referencia transparente.

Al utilizar un entorno de referencia multimodelo en una organización se pueden observar beneficios como: la unificación de un solo programa de mejora, enfoque de la mejora en la organización en lugar del modelo utilizado, reducción del coste total en relación con la implementación del modelo, entre otros.

Sin embargo, de acuerdo a (Kirwan, et al. 2008) la integración práctica del conjunto de tecnologías seleccionadas en un entorno multimodelo, por lo tanto, se necesita entender tres cosas:

1. La relación entre las tecnologías seleccionadas.
2. Los tipos de elementos que contienen las tecnologías seleccionadas.
3. La trazabilidad entre las tecnologías y el proceso de implementación.

2.3. Barreras para el uso de entornos de referencia multimodelo

Las barreras al uso de entornos de referencia multimodelo tienen su origen en que las organizaciones intentan aplicar las normas y modelos que han elegido para cumplir sus objetivos de negocio, además de aquellos que están obligados a elegir (Urs, et al. 2009). Entre las principales barreras en el uso de entornos de referencia multimodelos se pueden mencionar:

- La proliferación de modelos que la organización debe considerar.
- Las diferencias en la estructura y terminología entre los estándares y modelos.
- Las dificultades para reconocer similitudes entre los estándares y modelos.
- Los conflictos entre los diferentes programas de mejora en la organización, debido a que cada uno defiende y fomenta la mejora basada en el modelo o estándar que utiliza.
- La falta de soporte para mantener e institucionalizar los logros realizados por programas individuales de mejora.
- La proliferación del número de auditorías, evaluaciones y benchmarking que la organización debe someterse.

3. Experiencia en la mejora de los procesos de gestión de proyectos en everis

En esta sección se presenta la experiencia de la mejora de los procesos de gestión de proyectos en la empresa everis, utilizando un entorno de referencia multimodelo.

3.1. Descripción del entorno de aplicación de la metodología

everis es una consultora multinacional con presencia en Europa y Latinoamérica que persigue convertir conocimiento y tecnología en herramientas para mejorar sustancialmente el desempeño de las organizaciones. Su oferta de servicios, que ofrece soluciones a grandes compañías de cualquier sector de actividad, se fundamenta en tres pilares básicos: innovación, eficiencia, mediante la utilización de especialistas y metodologías, que hacen uso del conocimiento específico para cada proyecto, y productividad, al optimizar los resultados en tiempo y coste.

everis, desde su creación en 1996, ha ido creciendo tanto en facturación como en número de empleados de manera constante y siempre de manera orgánica. Su cierre del ejercicio 2009 ha reportado unos ingresos superiores a los 404 M€ con una plantilla de más de 7000 empleados, siendo superior a 1000 el número medio de proyectos abiertos mensualmente.

3.2. Alcance de aplicación de la mejora

La gestión de proyectos en everis tiene un amplio impacto en su negocio, y esta medida de acuerdo a una serie de indicadores relacionados con esos objetivos. Por lo tanto el establecimiento del alcance de la aplicación de la mejora se basa en dos aspectos: procesos (*los procesos de gestión de proyectos tienen un alto impacto en los objetivos del negocio*) y personas (*la gestión de proyectos llevada a cabo por los*

gerentes de cuenta, quienes realizan la gestión de los proyectos (uno o más proyectos) y son los responsables últimos de un proyecto).

3.3. Aplicación de la mejora

La implementación de las mejoras se ha llevado a cabo a través del desarrollo de un método de gestión de proyectos, que se incluirá en la metodología propia de everis: Corporate Methods (COM).

El método ha sido el resultado de la aplicación de la metodología MIGME-RRC. La metodología permite la implementación gradual y continua de mejoras en los procesos de una organización enfocando en la reducción de la resistencia al cambio. En este artículo no se incluye la descripción de la metodología MIGME-RRC, sin embargo se considera importante describir brevemente cada etapa de la metodología:

- *Identificar mejores prácticas internas:* el objetivo de esta fase es entender cómo trabaja la organización
- *Evaluar el rendimiento de la organización:* el objetivo de esta fase es establecer el rendimiento de los procesos de la organización a través del análisis de las mejores prácticas identificadas
- *Analizar las prácticas externas:* el objetivo de esta fase es establecer un entorno multimodelo que se utilizará como referencia para la selección de mejores prácticas externas que se ajusten a la forma de trabajo de la organización y hagan los procesos más eficientes
- *Implementar mejoras en los procesos:* el objetivo de esta fase es el diseño de los nuevos procesos a través de la integración de las mejores prácticas internas y mejores prácticas externas

Para más información de la metodología referirse a (Calvo-Manzano et al. 2009), (Calvo-Manzano et al. 2010). El artículo se centra en relatar la experiencia de la implementación de la mejora de procesos de gestión de proyectos en everis.

En esta sección se describen las actividades realizadas en everis siguiendo la metodología MIGME-RRC.

3.3.1 Identificar mejores prácticas internas

La identificación de las mejores prácticas en everis se realizó como se describe a continuación:

- Se condujeron entrevistas con casos de éxito de la organización. Como paso previo a la realización de las entrevistas, se solicitó a la alta dirección de la organización la selección de los gerentes de cuenta con más éxito.
- Se analizó la información recopilada en las entrevistas, y se realizaron diagramas por cada entrevista. Los diagramas fueron validados por los gerentes entrevistados.
Este paso se considera crítico para la identificación de las actividades, por lo que se ha repetido hasta que los diagramas de actividades fueron aprobados.
- Se realizó una la primer trazabilidad de actividades con los diferentes diagramas de actividades aprobados en entrevistas, para obtener “actividades

genéricas” de la organización. La validación de estas actividades fue realizada a tres niveles gerentes de cuenta, gerentes y jefes de proyectos.

- Se analizó la documentación del Sistema de Gestión de Calidad (SGC). El SGC contiene la documentación formal de los procesos de gestión de proyectos de la organización.
- Finalmente se realizó la segunda trazabilidad entre las “*actividades genéricas*” y el Sistema de Gestión de Calidad., con lo que se obtuvieron las actividades reales llevadas a cabo en la organización, y se realizaron validaciones de estas actividades a tres niveles gerentes de cuenta, gerentes y jefes de proyectos.

Las lecciones aprendidas de la realización de esta fase se listan a continuación:

1. Con la implicación de un representante de la alta dirección en la elección de los gerentes de cuenta que han llevado a su cargo proyectos de éxito se obtienen compromiso de la alta dirección antes del inicio del proyecto de mejora.
2. Utilizar cuestionarios con preguntas abiertas como guía para las entrevistas ha permitido que cada gerente entrevistado tenga la libertad de definir las actividades que realiza, y por consiguiente ha permitido hacer una mejor extracción del conocimiento tácito.
3. Permitir la validación de las actividades genéricas a tres niveles permite obtener el compromiso de las partes interesadas con la mejora, y asegura que se esté hablando el mismo lenguaje en el contexto de procesos en toda la organización.
4. Involucrar e integrar tanto a las fuentes de información (gerentes de cuenta) como al grupo de validación (gerentes de proyecto, jefes de proyecto y líderes de proyectos) asegura que el establecimiento de cómo son los procesos realmente refleje la forma en que la organización trabaja.

3.3.2 *Evaluar el rendimiento de la organización con las prácticas reales*

Para la realización de esta fase, se ha tomado como información de entrada la documentación correspondiente a las auditorías de control internas. El análisis de los resultados de estas auditorías ha permitido establecer una visión global del cumplimiento de los indicadores clave de la organización obtenido con las prácticas que están siendo realizadas actualmente en la organización.

La evaluación del rendimiento de la organización en everis se realizó como se describe a continuación:

- Se recogieron datos de las auditorías de everis que contenían información sobre los valores planificados para los indicadores del negocio relacionados con los objetivos del negocio para el periodo 2007- 2008.
- Se recogieron datos de las auditorías de everis que contenían información sobre valores reales de los indicadores del negocio relacionados con los objetivos del negocio medidos durante el periodo 2007-2008.
- Se realizó la trazabilidad de mejores prácticas-indicadores del negocio y objetivos del negocio mediante diagramas de objetivos del negocio.
- Se realizó la matriz de objetivos del negocio-indicadores relacionados con los objetivos del negocio en base a la trazabilidad realizada. Posteriormente, se registraron los valores planificados y reales de los indicadores.

- Con los resultados se realizó una priorización de los indicadores del negocio a mejorar. Los indicadores a mejorar elegidos fueron (1) normas del negocio, (2) documentación del plan de proyecto, y (3) actas de arranque. Aún cuando los tres indicadores seleccionados no se desviaban mucho entre el valor planificado y el valor real, todos debían ser mejorados debido a su alto impacto en el logro de los objetivos del negocio.

Las lecciones aprendidas de la realización de esta fase se listan a continuación:

1. Utilizar las auditorías realizadas normalmente por la organización han permitido establecer una visión general del rendimiento de los procesos de gestión de proyectos reales.
2. Realizar una trazabilidad entre las mejores prácticas reales y los objetivos del negocio han permitido la detección de oportunidad de mejora sabiendo las fortalezas de la organización.
3. Presentar a las partes interesadas con la mejora un informe de la visión del estado actual y deseado de los procesos ha permitido obtener compromiso de estas partes, ya que se ha recalcado en todo momento la importancia de su participación para el logro del objetivo.

3.3.3 Analizar las prácticas externas

Esta fase permite el establecimiento del entorno de referencia multimodelo para la selección de aquellas prácticas que mejor se adapten a la forma de trabajo de la organización. El análisis de las prácticas externas en everis se realizó como se describe a continuación:

- Se seleccionaron los modelos y estándares de mejora de procesos que se correspondían con la forma de trabajo de la organización. Los modelos seleccionados fueron: CMMI-DEV v1.2, PMBOOK, PRINCE2, TSP, COBIT, ISO9001, and ISO/IEC 15504
- Se seleccionó el modelo CMMI-DEV v1.2 como modelo de referencia.
- Se seleccionaron los procesos relacionados con la gestión de proyectos.
- Se estableció el nivel de detalle en prácticas específicas.
- Se realizó una plantilla para registrar las prácticas en base a: información de entrada, subprácticas, herramientas y técnicas, productos de trabajo y componentes informativos.
- Se identificaron las similitudes entre los modelos y estándares.
- Se estableció el entorno de referencia multimodelo registrando las prácticas en la plantilla de tal manera que se enriqueciera la información para llevar a cabo cada práctica específica.

Las lecciones aprendidas de la realización de esta fase se listan a continuación:

1. Seleccionar los modelos y estándares en base a la forma de trabajo de la organización ha permitido establecer un entorno de referencia multimodelo adecuado a ella.
2. Realizar un entorno de referencia multimodelo a nivel de prácticas específicas ha permitido la visualización por parte de las partes interesadas de un conjunto de mejores prácticas externas acordes a la forma de trabajo de la organización.

3. La selección de un modelo o estándar de referencia permite hacer el análisis más estructurado para reconocer elementos comunes que a pesar de las diferentes estructuras y terminologías y los diferentes niveles de abstracción tienen los estándares y modelos utilizados en las organizaciones y que es un gran paso hacia el establecimiento de un entorno de referencia multimodelo.

3.3.4 Implementar mejoras en los procesos

La implementación de mejoras en everis se realizó como se describe a continuación:

- Se analizaron los factores de resistencia al cambio y de riesgos asociados con la mejora de los procesos. Los principales riesgos de resistencia fueron: sentimiento de sobrecarga de trabajo, falta de compromiso de los mandos intermedios y la falta de algún tipo de recompensa relacionada con la mejora. Por su parte los principales riesgos asociados con la mejora fueron: material de entrenamiento en el uso de los procesos inadecuado, falta de una adecuada comunicación, falta de soporte, falta de seguimiento de la mejora y falta de implicación de las partes interesadas.
- Se analizaron las dependencias entre prácticas internas y prácticas externas, valorando su impacto en el logro de sus objetivos y su facilidad de adopción.
- Se seleccionaron las prácticas externas del entorno de referencia multimodelo en base al análisis del impacto y facilidad de adopción.
- Se realizaron los procesos mejorados que contenían todo el conocimiento de everis reflejado en sus mejores prácticas identificadas en la primera fase y las prácticas externas seleccionadas.
- Se integraron los procesos para la creación del método de gestión de proyecto para su metodología corporativa Corporate Methods (COM).
- El método fue validado y aprobado por el grupo de calidad y metodología de everis.
- Se realizaron proyectos piloto para recopilar la experiencia de la utilización el nuevo método y se habilitaron canales para recibir realimentación para el método.
- Finalmente, una vez realimentado, el método fue puesto en la intranet de everis para su uso como parte de la metodología corporativa.

Las lecciones aprendidas de la realización de esta fase se listan a continuación:

1. Analizar los factores de resistencia al cambio y riesgos asociados a la mejora antes de la implementación de la mejora ha permitido tomar las acciones preventivas para su disminución.
2. Realizar el análisis de dependencias entre prácticas internas y externas y el análisis de facilidad de adopción a permitido la selección de aquellas prácticas que hacen más eficientes a los procesos y que al mismo tiempo son percibidas por los usuarios como prácticas que ayudan a la evolución de su trabajo.
3. Mantener un soporte y seguimiento constante durante todo el proceso de implementación de mejoras, así como una estrategia de comunicación adecuada ha permitido reducir la resistencia al cambio por parte de las partes interesadas.

4. Seleccionar los proyectos pilotos teniendo en cuenta por un lado que no representen un riesgo para la organización y por otro seleccionar aquellos proyectos en los que participe personal con perfil de adaptadores primarios permite una mejor realimentación al método y un mejor análisis del comportamiento del mismo.

4. Resultados Obtenidos

El análisis de los resultados obtenidos se ha realizado teniendo en cuenta tres aspectos. Los aspectos establecidos se muestran en la Tabla 1.

Tabla 1 – Aspectos seguidos para realizar el análisis de los resultados

✓ Aspecto	✓ Definición
✓ Métricas	✓ Utilizar las metricas definidas por la metodología MIGME-RRR para realizar el análisis: Utilización del proceso, Rendimiento del proceso y Aceptación del proceso
✓ Recogida de datos	✓ Utilizar un enfoque de recogida de datos en tres tiempos: antes de la mejora, durante la mejora y después de la mejora
✓ Técnica o Herramienta utilizar	✓ Utilizar gráficas de control para analizar el comportamiento de los procesos

Por lo tanto el análisis de los resultados se mostrará en tres apartados de acuerdo a las métricas establecidas por la metodología MIGME-RRR. Además para el análisis se han agrupado los valores en dos grupos: por un lado grupo I (antes(FY'07) y durante (FY'08)) y por el otro grupo II (después (FY'09)).

4.1. Utilización del proceso

El objetivo de esta métrica es analizar el grado de rendimiento de las mejores prácticas contenidas en los nuevos procesos. Los datos utilizados para este análisis se han tomado de las auditorías internas de everis. En estas auditorías se analizan las no conformidades de indicadores del proceso relacionados con los objetivos del negocio.

Las no conformidades son problemas identificados en las evaluaciones que reflejan la no adherencia a los estándares, descripciones de proceso o procedimientos aplicables.

El análisis de las no conformidades de los indicadores de los procesos permitirá obtener información sobre la utilización de los procesos.

Los indicadores analizados son normas de gestión, planificación del proyecto y actas de arranque ya que son en los que está enfocada la mejora de los procesos de gestión de

proyectos. A continuación se muestran los resultados obtenidos de cada indicador como sigue: la Figura 1 muestra el gráfico de control para el indicador normas de gestión, la Figura 2 muestra el gráfico de control para el indicador actas de arranque y la Figura 3 muestra el gráfico de control para el indicador planificación del proyecto.

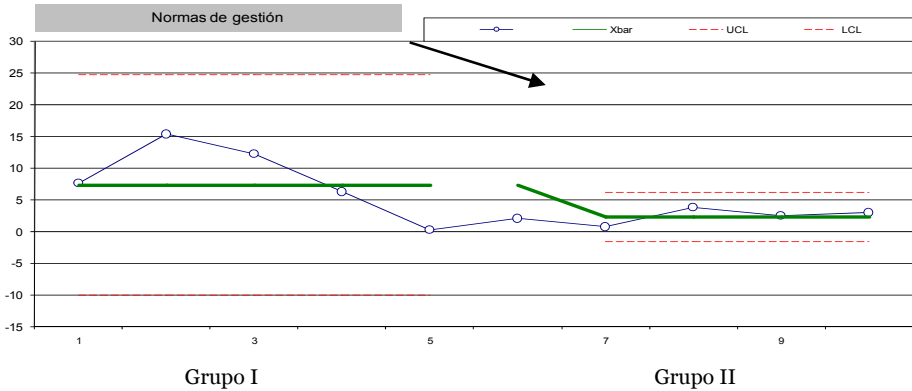


Figura 1 – Gráfico de control para el análisis del indicador normas de gestión

Como se observa en la Figura 1 el gráfico de control del indicador normas de gestión muestra una mejora al observarse una disminución en la media (Xbar) de 7,25 en el grupo I a 2,25 en el grupo II. Además, se observa un desplazamiento de los valores en los límites de control superior (UCL) e inferior (LCL). El límite superior de 24,62 (grupo I) a 6,09 (grupo II) y el límite de control inferior de -10,12 (grupo I) a -1,59 (grupo II). De igual manera se observa que se tiene un mejor control ya que los valores tienden a estar más cerca de la media.

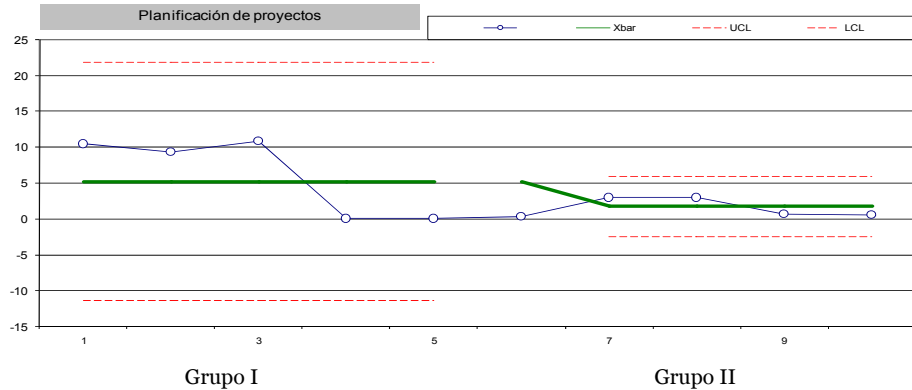


Figura 2 – Gráfico de control para el análisis del indicador planificación del proyecto

Como se observa en la Figura 2 el gráfico de control del indicador planificación del proyecto muestra una mejora al observarse una disminución en la media (Xbar) de 5,13 en el grupo I a 1,67 en el grupo II. Además, se observa un desplazamiento de los valores en los límites de control superior (UCL) e inferior (LCL). El límite superior de 21,74 (grupo I) a 5,93 (grupo II) y el límite de control inferior de -11,47 (grupo I) a -2,58 (grupo II).

(grupo II). De igual manera se observa que se tiene un mejor control ya que los valores tienden a estar más cerca de la media.

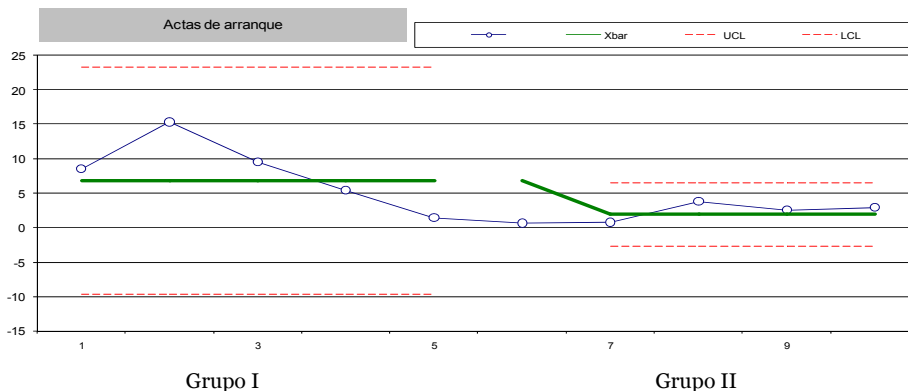


Figura 3 – Gráfico de control para el análisis del indicador actas de arranque

Como se observa en la Figura 3 el gráfico de control del indicador actas de arranque muestra una mejora al observarse una disminución en la media (Xbar) de 6,78 en el grupo I a 1,9 en el grupo II. Además, se observa un desplazamiento de los valores en los límites de control superior (UCL) e inferior (LCL). El límite superior de 23,32 (grupo I) a 6,51 (grupo II) y el límite de control inferior de -9,75 (grupo I) a -2,71 (grupo II). De igual manera se observa que se tiene un mejor control ya que los valores tienden a estar más cerca de la media.

4.2. Rendimiento del proceso

El objetivo de esta métrica es analizar el rendimiento de los nuevos procesos. En el caso de estudio realizado en everis los datos para esta métrica se han obtenido analizado aquellos proyectos que han tenido algún tipo de desviación en sus costes internos, ya sea en horas incurridas, gastos externos o subcontratación. No obstante, estas desviaciones no afectaron a los plazos de tiempo acordado con los clientes.

La Figura 4 muestra la gráfica de control para analizar el rendimiento de los procesos a través del análisis de los proyectos con algún tipo de desviación.

Como se muestra en la Figura 4 el gráfico de control de los proyectos con desviación muestra una mejora ya que se observa una disminución en la media (Xbar) de 12,79 en el grupo I a 11,34 en el grupo II. Además, se observa un desplazamiento de los valores en los límites de control superior (UCL) e inferior (LCL). El límite superior de 15,67 (grupo I) a 13,96 (grupo II) y el límite de control inferior de -9,92 (grupo I) a -8,71 (grupo II). De igual manera se observa que se tiene un mejor control ya que los valores tienden a ser igual a la media. Esto demuestra que se tiene una mejora gradual y continúa a través de los tres periodos analizados, por lo tanto, los nuevos procesos son más eficientes.

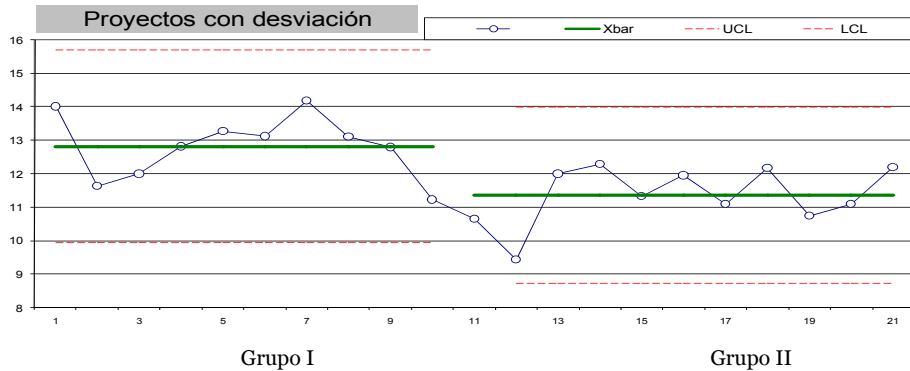


Figura 4 – Gráfico de control para el análisis de proyectos con algún tipo de desviación

4.3. Aceptación del proceso

El objetivo de esta métrica es analizar la aceptación de los nuevos procesos por las partes implicadas con la gestión de proyectos y, por consiguiente, permite analizar si ha existido disminución de la resistencia al cambio en la aceptación de los nuevos procesos.

Los datos para esta métrica fueron obtenidos mediante la realización de encuestas a 355 personas implicadas en la gestión de proyectos de las diferentes oficinas de everis en todo el mundo.

La Figura 5 muestra el gráfico de control para el análisis del uso de método de gestión de proyectos COM.

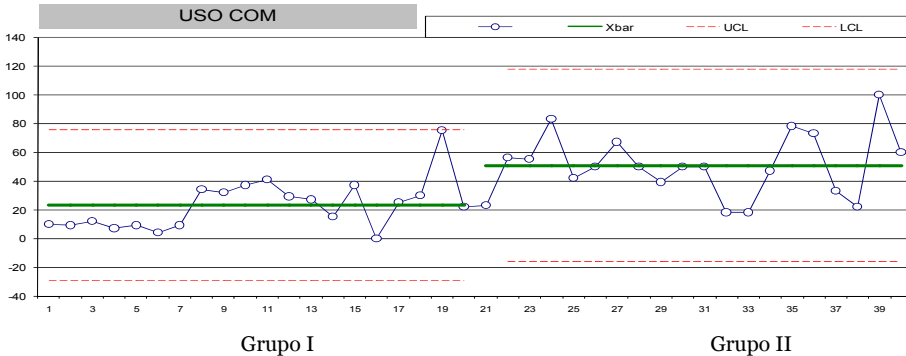


Figura 5 – Gráfico de control de uso del método de gestión de proyectos COM

Como se observa en la Figura 5 el gráfico de control del uso de COM muestra una mejora ya que se observa un aumento gradual y continuo en la media (Xbar) de managers que usan el método COM para la gestión de sus proyectos de 23,32 en el grupo I a 50,7 en el grupo II.

5. Conclusiones

Implementar mejoras en los procesos de una organización estableciendo un entorno de referencia multimodelo ha permitido la adopción de mejores prácticas externas acordes a la forma de trabajo de la organización, y por lo tanto la obtención de procesos más eficientes que reflejan la forma de trabajo de la organización.

Como resultado el personal disminuye la resistencia al cambio cuando se implementan las mejoras, ya que ellos perciben la adopción de las prácticas contenidas en los nuevos procesos como una evolución de su trabajo.

Finalmente, la implementación de MIGME-RCC para la mejora de proceso gradual y continuo confirmó que el personal solo acepta cambios asimilados con beneficios identificados. De esta manera, el personal percibe el cambio como una evolución a su forma de trabajo.

Cabe resaltar que everis es una empresa que busca dar siempre más valor a sus clientes. Esta situación se ve reflejada en la satisfacción de sus clientes y en el crecimiento año tras año de sus proyectos. Por lo tanto, ha permitido la implementación de mejora en sus procesos a través MIGME-RCC. Los datos analizados en este artículo son datos verídicos que muestran que ha habido en general una mejora en el desarrollo del delivery en everis en todas las oficinas analizadas, lo que refleja su compromiso con la mejora continua.

Agradecimientos

Este trabajo ha sido patrocinado por la empresa Fundación everis a través de la “Cátedra de Mejora del Proceso Software en el Espacio Iberoamericano”. Mención especial merece el apoyo prestado por everis facilitando datos y posibilitando la realización de los pilotos, sin cuya ayuda no hubiera sido posible la realización de esta investigación.

Referencias bibliográficas

- Goldenson, D. (2007). Performance Outcomes from Process Improvement -Tech Views. *Software Tech*, Volumen 10
- Conradi, H., Fuggetta, A. (2002). Improving software process improvement, *IEEE Software*, Volume 19, Issue 4, pp.92–99, doi: 10.1109/MS.2002.1020295
- Pries-Heje, J., Johansen, J. (2010). SPI Manifesto. *eurospi.net*
- Niazi, M., Wilson, D., Zowghi, D. (2006). Critical success factors for software process improvement implementation: An empirical study. *Software process improvement and practices*
- Mahmood, N., Muhammad, A.B., Nolin, M.K. (2008). Demotivators of software process improvement: An empirical investigation. *Software process improvement and practice*
- SuZ, G. (2007). "Process Improvement "At the Edges"", Software Engineering Institute / Carnegie Mellon University

- Marino, L., John, M. (2009). Process Improvement in a Multimodel Environment Builds Resilient Organizations
- Urs, A., Andre, H., Patrick, K. (2009). A Unified Process Improvement Approach for Multi-Model Improvement Environments
- Mogilensky, J. (2009). Pathological Box-Checking: The Dark Side of Process Improvement", SEPG 2009 North America Conference
- Siviy, J., Patrick, K., Lisa, M., John, M. (2008). The Value of Harmonization Multiple Improvement Technologies: A process improvement professional's view. Software Engineering Institute (SEI), Carnegie Mellon University. First part
- Srivastava, N., Singh, S., Dokken, T. (2009). Assorted Chocolates & Cookies in a Multimodel Box, SEPG 2009 North America Conference
- Lawrence, J., Becker, N. (2009). Implementing Insights and Lesson Learned Using ISF for Excellence", SEPG 2009 North America Conference
- Kirwan, P., Jeannie S., Lisa, M., John, M. (2008). Improvement Technology Classification and Composition in Multimodel Environments. *Software Engineering Institute (SEI), Carnegie Mellon University*
- Calvo-Manzano Jose A., Cuevas Gonzalo, Muñoz Mirna, San Feliu Tomás. (2009). Metodología para la Implementación Gradual de Mejoras, con Enfoque en la Reducción de la Resistencia al Cambio. Proceedings 4ª Conferencia Ibérica de Sistemas y Tecnologías de la Información (CISTI 2009)
- Calvo-Manzano Jose A., Cuevas Gonzalo, Gómez Gerzón, Mejia Jezreel, Muñoz Mirna, Tomás San Feliu. (2010). Methodology for process improvement through basic components and focusing on the resistance to change. *Journal of Software Maintenance and Evolution: Research and Practice*

Crítérios Editoriais

A RISTI (Revista Ibérica de Sistemas e Tecnologias de Informação) é um periódico científico, propriedade da AISTI (Associação Ibérica de Sistemas e Tecnologias de Informação), que foca a investigação e a aplicação prática inovadora no domínio dos sistemas e tecnologias de informação.

O Conselho Editorial da RISTI incentiva potenciais autores a submeterem artigos originais e inovadores para avaliação pelo Conselho Científico.

A submissão de artigos para publicação na RISTI deve realizar-se de acordo com as chamadas de artigos e as instruções e normas disponibilizadas no sítio Web da revista (<http://www.aisti.eu>).

Todos os artigos submetidos são avaliados por um conjunto de membros do Conselho Científico, não inferior a três elementos.

Em cada número da revista são publicados entre seis a oito dos melhores artigos submetidos.

Criterios Editoriales

La RISTI (Revista Ibérica de Sistemas y Tecnologías de la Información) es un periódico científico, propiedad de la AISTI (Asociación Ibérica de Sistemas y Tecnologías de la Información), centrado en la investigación y en la aplicación práctica innovadora en el dominio de los sistemas y tecnologías de la información.

El Consejo Editorial de la RISTI incentiva autores potenciales a someter sus artículos originales e innovadores para evaluación por el Consejo Científico.

La sumisión de artículos para publicación en la RISTI debe hacerse de conformidad con las llamadas de los artículos y las instrucciones y normas establecidas en el sitio Web de la revista (<http://www.aisti.eu>).

Todos los trabajos sometidos son evaluados por un número de miembros del Consejo Científico de no menos de tres elementos.

En cada número de la revista se publican seis a ocho de los mejores artículos sometidos.

Chamada de Artigos

Encontra-se aberto até 17 de Abril de 2011 o período de envio de artigos para o sétimo número da RISTI (Revista Ibérica de Sistemas e Tecnologias de Informação), o qual será publicado durante o próximo mês de Junho de 2011.

Este número é dedicado à temática do Negócio Digital (e-Business e e-Commerce) e pretende integrar contribuições originais e relevantes nas suas diferentes dimensões (tecnológica, organizacional, social, etc.).

Os tópicos recomendados incluem os listados abaixo, no entanto também serão bem-vindos outros tópicos relacionados com a temática mas aqui não incluídos:

- Modelos e Estratégias de Negócio.
- Modelos de Maturidade.
- Aplicações, Sistemas e Tecnologias.
- Métodos, Ferramentas e Ambientes.
- Cadeias de Fornecimento.
- Integração de Sistemas.
- Web Marketing.
- Sistemas Móveis.
- Sistemas Colaborativos.
- Acessibilidade e Usabilidade.
- Web Social e Web Semântica.
- Confiança, Segurança, Privacidade e Regulamentação.

Os artigos devem ser escritos em Português ou Espanhol. Para informações sobre dimensão, normas de formatação e processo de submissão, agradecemos a consulta do Portal da RISTI: <http://www.aisti.eu>

Llamada de Artículos

Se encuentra abierto hasta al día 17 de Abril de 2011 el período de envío de artículos para el séptimo número de la RISTI (Revista Ibérica de Sistemas y Tecnologías de la Información), el cual será publicado durante el próximo mes de Junio de 2011.

Este número se dedica a la temática del Negocio Digital (e-Business e e-Commerce) y pretende integrar contribuciones originales y relevantes en sus diferentes dimensiones (tecnológica, organizacional, social, etc.).

Los tópicos recomendados incluyen los abajo listados, pero también serán bienvenidos otros tópicos relacionados con la temática y aquí no incluidos:

- Modelos y Estrategias de Negocio.
- Modelos de Madurez.
- Aplicaciones, Sistemas y Tecnologías.
- Métodos, Herramientas y Entornos.
- Cadenas de Suministro.
- Integración de Sistemas.
- Web Marketing.
- Sistemas Móviles.
- Sistemas Colaborativos.
- Accesibilidad y Usabilidad.
- Redes Sociales y Web Semántica.
- Confianza, Seguridad, Privacidad, y Regulación.

Los artículos deben ser escritos en portugués o español. Para obtener información sobre longitud, reglas de formato y proceso de envío, por favor consulte el Portal de la RISTI: <http://www.aisti.eu>

Os associados da AISTI recebem a RISTI gratuitamente, por correio postal. Torne-se associado da AISTI. Preencha o formulário abaixo e envie-o para o e-mail aisti@aisti.eu

Los asociados de la AISTI reciben la RISTI por correo, sin costo alguno. Hazte miembro de la AISTI. Rellena el siguiente formulario y remítelo al e-mail aisti@aisti.eu



Formulário de Associado / Formulario de Asociado

Nome/Nombre: _____

Instituição/Institución: _____

Departamento: _____

Morada/Dirección: _____

Código Postal: _____ Localidade/Localidad: _____

País: _____

Telefone/Teléfono: _____

E-mail: _____ Web: _____

Tipo de Associado e valor da anuidade:

☐ Individual - 25€

☐ Instituição de Ensino ou I&D/Institución de Educación o I&D - 250€

☐ Outro (Empresa, etc.) - 500€

NIF/CIF: _____

Data/Fecha: ____/____/____ Assinatura/Firma: _____