

Uma Abordagem ao Processo de Desenvolvimento dos Sistemas de Informação: cuidados a ter ao longo do processo no caso dos SI's tradicionais e SI's distribuídos na Web

Leonor Teixeira

Departamento de Economia, Gestão e Engenharia Industrial, Universidade de Aveiro
3810-193 Aveiro, Portugal

e

Carlos Ferreira

Departamento de Economia, Gestão e Engenharia Industrial, Universidade de Aveiro,
3810-193 Aveiro, Portugal

CIO, Centro de Investigação Operacional, Universidade de Lisboa,
Lisboa, Portugal

e

Beatriz Sousa Santos

IEETA / Departamento de Electrónica e Telecomunicações, Universidade de Aveiro
3810-193 Aveiro, Portugal

RESUMO

Hoje em dia, cada vez mais os casos de insucesso verificados em projectos de Sistemas de Informação (SI) se atribuem a falhas cometidas ao longo do processo de desenvolvimento, quer pela falta de planeamento, quer pelo uso inadequado de técnicas e/ou metodologias que conduzem ao projecto da solução desejada. O propósito deste artigo, ainda que de exposição, é precisamente identificar, ao longo do Processo de Desenvolvimento de um Sistema de Informação (PDSI), fases onde pequenas falhas podem conduzir a resultados inesperados e desagradáveis. Para tal, começa-se por uma caracterização estrutural dos SI's, nomeadamente, a identificação e caracterização das suas componentes estruturais: dados, processo e interfaces; identificação dos principais *stakeholders* envolvidos no projecto; e por fim, o caminho de desenvolvimento a seguir desde a fase em que se tem conhecimento da existência do problema até à instalação da melhor solução, passando por quatro etapas fundamentais: (i) análise do problema; (ii) análise de requisitos; (iii) modelização da solução; e (iv) implementação. Particularmente para as três primeiras etapas serão apresentadas as técnicas consideradas mais adequadas para a obtenção dos dados e as diferentes alternativas de modelização. Toda esta discussão será desenvolvida em torno de dois tipos de SI's posicionados em pontos extremos a nível de complexidade de desenvolvimento: SI's tradicionais (sistemas locais) e SI's distribuídos na *Web* (aplicações *Web*).

Palavras-Chave: Desenvolvimento de Sistemas, Processo de desenvolvimento, SI's distribuídos, Engenharia de *Software*, Modelização, OOHDM, UML.

1. INTRODUÇÃO

Os Sistemas de Informação (SI's) são criados com o objectivo de, directa ou indirectamente, servirem a estratégia de negócio da organização da qual fazem parte, transmitindo aos seus

colaboradores informação adequada, em tempo útil e de forma eficiente. Para tal, é necessário que os SI's estejam devidamente enquadrados na organização, alinhados com os objectivos de negócio e sejam adequados ao tipo de utilizadores que potencialmente irão servir. De acordo com Varajão [1], não é possível compreender um SI sem compreender a organização de que aquele é parte integrante, afirmando mesmo o autor que as organizações e os seus SI's são indissociáveis. Como consequência, um SI terá necessariamente reflexos a nível organizacional e o fracasso de um projecto deste tipo pode conduzir a custos elevados e até ao próprio fracasso de desempenho da organização portadora do mesmo. É muito comum, nomeadamente em empresas mais tradicionais e menos viradas para as novas tecnologias, verificar que os SI's têm um efeito contrário àquele que inicialmente se propunham, chegando mesmo a transformar, de forma negativa, práticas sociais ao nível organizacional. Outras vezes, após o forte investimento em projectos deste tipo, quando chega a altura de rentabilizar o mesmo, os utilizadores simplesmente optam por continuar a trabalhar a informação usando os métodos tradicionais face à utilização do novo sistema [2]. Estes são exemplos de cenários de SI's cujo resultado fracassou pelo facto de não estarem devidamente contextualizados na cultura e práticas sociais da organização, assim como no propósito da sua existência. Na maior parte das vezes, a causa destes resultados inesperados e desagradáveis deve-se à falta de planeamento ou ao uso de processos, técnicas e metodologias inadequadas à situação [3]. O mesmo é dizer, falhas cometidas ao longo do Processo de Desenvolvimento do Sistema de Informação (PDSI), tratando-se do tipo de erros mais fáceis de cometer e, conseqüentemente, mais difíceis de recuperar, uma vez serem detectados numa fase onde o custo de recuperação de erro tem o seu valor máximo (produto final). É claro que a introdução de um SI numa organização é um processo que requer muito cuidado e cujo sucesso de implementação é, em muitos casos, resultado do sucesso obtido nas várias etapas que compõem o PDSI. O esquecimento, ou simplesmente um tratamento inadequado de uma das fases, poderá conduzir a um resultado diferente daquele que se desejaria. Neste sentido, é importante

considerar todas as componentes que compõem a estrutura de um SI, auscultar todos os *stakeholders* não técnicos que, directa ou indirectamente, irão ser influenciados com a integração do novo sistema e, finalmente, adoptar uma metodologia que garanta uma correcta passagem da informação entre as diferentes fases que compõem o processo de desenvolvimento e consequentemente entre os diferentes elementos (*stakeholders* técnicos) que fazem parte da equipa de projecto. Por último, e não menos importante, é necessário considerar o potencial tecnológico existente no mercado e enquadrá-lo razoavelmente na solução pretendida, uma vez que o seu contributo é também importante para o sucesso do resultado final [4].

2. ESTRUTURA DE CONSTRUÇÃO E EQUIPA DE PROJECTO DE UM SISTEMA DE INFORMAÇÃO

Na maior parte dos casos, o desenvolvimento de SI's corresponde a projectos de grande dimensão, envolve a participação de vários grupos de *stakeholders* (técnicos e não técnicos), agrega várias componentes e tecnologias na sua estrutura (dados, processo e interfaces) e faz uso de várias técnicas e metodologias para apoiar o desenvolvimento das etapas que compõem o processo. De acordo com Whitten [5],

hoje em dia grande parte dos SI's suportam uma estrutura em torno de três componentes diferentes (Fig. 1):

- (i) **Dados** - componente que corresponde à camada mais profunda de um SI e é nela que se armazena todo o material potencialmente informativo (em bruto);
- (ii) **Processo** - componente que corresponde à camada intermédia onde é colocada toda a lógica aplicacional de modo a permitir uma correcta passagem de dados/informação com a máxima garantia de segurança;
- (iii) **Interface** - componente que corresponde à camada onde são estabelecidos os principais pontos de contacto entre o sistema e o exterior, quer para *inputs* de dados, quer para *outputs* de informação.

Ainda que estas componentes se encontrem em todos os tipos de SI's, a sua identificação vai sendo cada vez mais fácil à medida que passamos de uma filosofia de SI's tradicionais (ex: aplicações locais) para SI's distribuídos (ex: aplicações *Web*), uma vez que nestes, as três componentes encontram-se logicamente e fisicamente separadas, necessitando de uma quarta (rede) para integrar as três anteriores. Por este motivo (dispersão das componentes), o tratamento que cada componente em particular requer na construção do SI torna-se mais especial no caso dos SI's distribuídos.

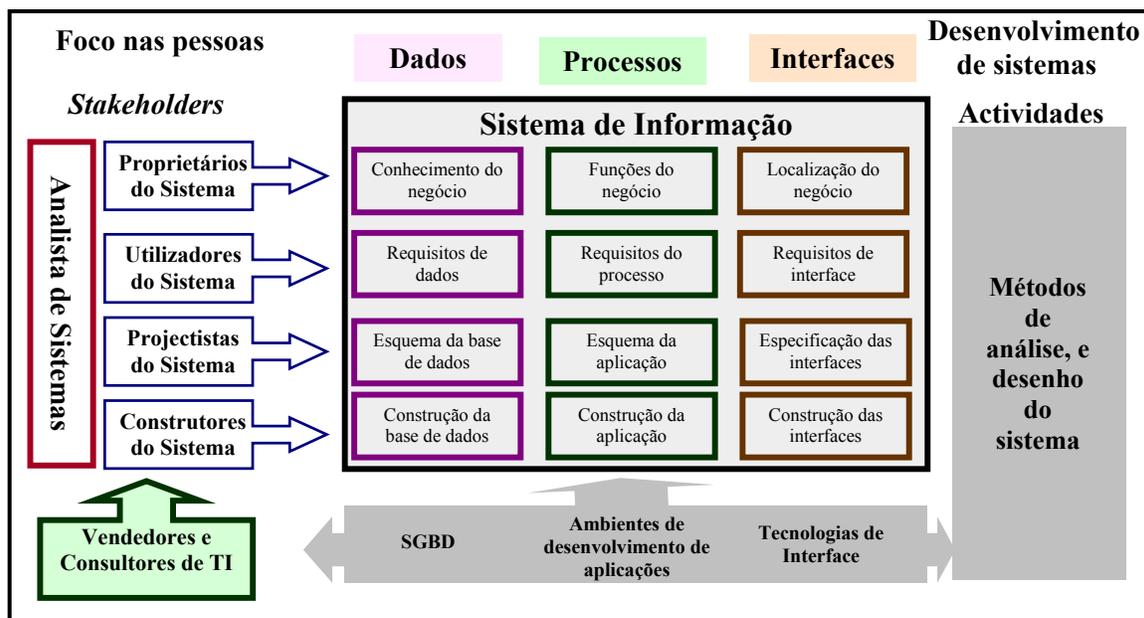


Fig. 1 – Estrutura de construção de um SI – adaptado de [Whitten et al. 2001].

Hoje em dia a maior parte dos projectos de SI's desenvolvidos têm características que os permitem classificar como SI's distribuídos. A descrição feita ao longo deste artigo irá considerar essa tendência, fazendo uma abordagem ao PDSI para o caso dos SI's distribuídos na *Web* (aplicações *Web*) e, confrontando essa abordagem com o caso dos SI's tradicionais.

O desenvolvimento de SI's é efectuado ao longo de várias etapas, contando com a participação de vários *stakeholders*. Os *stakeholders* técnicos que colaboram directamente no desenvolvimento da solução, tais como:

- (i) analista de sistemas - tem como função principal coordenar os esforços dos restantes elementos participantes,

desempenhando o papel de facilitador ao longo do processo. Geralmente está presente em todas as fases de desenvolvimento com importância especial nas fases iniciais. Nestas, estuda o problema conjuntamente com o proprietário do sistema e determina as necessidades conjuntamente com os potenciais utilizadores, elaborando um relatório (denominado habitualmente como 'especificação') para posteriormente ser trabalhado pelos projectistas;

- (ii) projectistas - geralmente especialistas tecnológicos que traduzem os requisitos descritos na especificação para soluções técnicas utilizando, para tal, uma linguagem de modelização adequada. Neste grupo podemos ainda

encontrar os especialistas de usabilidade responsáveis pelo projecto de interfaces.

- (iii) programadores - os especialistas tecnológicos, que trabalham no último elo da cadeia de desenvolvimento, com a responsabilidade de converter todo o trabalho resultante da etapa anterior para uma linguagem interpretável e executável pelo computador, assim como instalar e acompanhar a fase de arranque do novo sistema. Neste grupo podemos ainda encontrar os especialistas de usabilidade responsáveis pelo projecto de interfaces.

Os *stakeholders* não técnicos, embora não integrando directamente a equipa de projecto, contribuem (de forma muito importante) para o sucesso do resultado final, através das suas propostas e sugestões a nível de requisitos funcionais e não-funcionais; sendo assim:

- (iv) utilizadores – são todas as pessoas que irão interagir com o sistema, tendo um papel fundamental na identificação dos requisitos (funcionais e alguns requisitos não-funcionais);
- (v) proprietários do sistema - um dos principais beneficiários com a integração do novo sistema, é também o principal responsável pelo financiamento do projecto, podendo tratar-se do proprietário da organização do qual o sistema irá fazer parte ou, simplesmente, de sócios ou administradores da mesma. O seu contributo para o desenvolvimento do sistema é muito importante, uma vez que se trata do principal detentor do conhecimento do negócio, missão da organização e consequente alinhamento dos objectivos pretendidos com o sistema.

Para além destes, é necessário auscultar os consultores e vendedores das tecnologias de informação, uma vez que também estes são detentores do conhecimento tecnológico que irá constituir a base de suporte do sistema a desenvolver.

3. PROCESSO DE DESENVOLVIMENTO DE UM SISTEMA DE INFORMAÇÃO

O Processo de Desenvolvimento de um SI (PDSI) define as etapas e actividades necessárias para percorrer o caminho de DSI, desde a fase em que se tem conhecimento da existência do problema, até à instalação da solução desejada. As actividades determinam a maneira como as etapas são levadas a cabo, fazendo uso de um conjunto de técnicas e métodos que, por sua vez, podem ser auxiliadas por diferentes ferramentas tecnológicas.

Existem diferentes abordagens em relação às etapas que compõem o PDSI [5, 6, 7, 8, 9]. De uma maneira geral todas elas referem as quatro principais: (1) Análise do Problema; (2) Análise de Requisitos; (3) Modelização do Sistema; e, (4) Implementação; no entanto, alguns métodos sugeridos por aqueles autores trabalham apenas a etapa da modelização. Em relação às técnicas utilizadas para a obtenção dos dados em cada etapa, nomeadamente, inquéritos, entrevistas, reuniões, análise de documentação existente ou técnicas de observação directa, existe alguma flexibilidade na escolha, uma vez que qualquer das alternativas, desde que bem aplicada e adequada à situação, pode conduzir a bons resultados. O mesmo já não se pode dizer sobre os métodos usados, já que estes devem ser escolhidos de acordo com o tipo de SI que se pretende desenvolver, nomeadamente se se trata de um SI tradicional ou de um SI distribuído na Web (Fig. 2)

De acordo com a perspectiva de Koch [3, 10], as aplicações *Web* diferem dos SI's tradicionais em várias dimensões: (i) diversidade de pessoas envolvidas no processo (para além dos *stakeholders* já identificados anteriormente e comuns a qualquer sistema, podemos ter ainda a participação de *designers*, especialistas em multimédia, ou mesmo, especialistas em *marketing*); (ii) o tamanho e dispersão do universo de potenciais utilizadores, o que torna mais difícil o processo de captura de requisitos; (iii) a não linearidade dos documentos hipertexto, bem como a possibilidade de ligação fácil a outras aplicações que aumenta o risco de desorientação no espaço informativo por parte dos utilizadores.

Estas dimensões permitem classificar os SI's distribuídos como sistemas complexos arrastando, consequentemente, essa complexidade para o processo de desenvolvimento. Por esta mesma razão, a partir de uma determinada fase do PDSI, é necessário considerar o tipo de SI que se pretende instalar, de forma a chegar à melhor solução seguindo, naturalmente, o trajecto adequado (Fig. 2).



Fig. 2 - Trajecto do PDSI no caso dos tradicionais e dos SI's distribuídos.

De uma maneira geral, um SI numa organização surge com o objectivo de solucionar um problema. Note-se que o conceito aqui utilizado como problema se restringe apenas ao cenário que impulsiona a necessidade de construção do SI e não, propriamente, ao significado de um problema no seu verdadeiro sentido da palavra. De acordo com Whitten [5] o cenário pode ser:

- (i) uma situação de verdadeiro problema, real ou antecipado, que requer acções correctivas;
- (ii) oportunidades de melhorar uma situação, ainda que não fosse identificado um verdadeiro problema;
- (iii) directivas governamentais de mudanças de situação que requerem um ajustamento dos sistemas existentes.

No fundo, o problema (Fig. 2 e Fig. 3) constitui o principal motivo que conduz à necessidade de um novo sistema, sendo por isso muito importante a sua correcta identificação. A identificação do problema certo é o ponto de partida para a obtenção de uma boa especificação que irá constituir a base de todo o desenvolvimento da solução pretendida. Por outro lado, uma falha na identificação do problema, pode conduzir à elaboração de uma especificação diferente daquela que se pretende e todo o restante trabalho irá assentar numa solução, à partida, errada. Estando o problema devidamente identificado procede-se, então, à sua análise.

3.1 Análise do problema

Nesta etapa, primeira do PDSI, o analista reúne com o proprietário (ou outro responsável da organização) para clarificar o problema e discutir propostas de resolução. As técnicas utilizadas para a recolha de dados são, basicamente, reuniões, técnicas de observação directa e análise da documentação existente.

Nesta fase é necessário, ainda, proceder à análise dos processos de negócios (PN), de maneira a contextualizar a solução proposta no sistema organizacional em causa.

Uma grande parte dos casos de insucesso dos SI's deve-se, precisamente, a falhas nesta etapa. O facto do analista apresentar uma proposta sem ter conhecimento do PN da organização conduz, muitas vezes, a soluções técnicas baseadas em artificios puramente tecnológicos e que pode não corresponder à solução pretendida. Deste modo, deve começar-se sempre pela compreensão dos processos de negócio, assim como, a sua modelização (MPN). O objectivo aqui é perceber como o negócio é levado a cabo e em que medida o novo sistema poderá contribuir para o seu melhoramento, no que respeita ao fluxo de informação [11]. Desta forma evita-se o constrangimento de chegar a uma solução que, embora inclua os requisitos dos utilizadores, estes não tenham uma justificação sólida e não estejam alinhados com os objectivos organizacionais.

Existem vários métodos que permitem fazer este tipo de modelização, sendo o IDEF (*Integrated Computer Aided*

Manufacturing DEfinition) um dos mais utilizados. Dentro deste, existem vários modelos, estando o IDEF0 e o IDEF3 direccionados para a MPN [12]. Enquanto o IDEF0 permite fazer a modelização funcional através de uma notação muito simples (caixas e setas) especificando os *Inputs*, *Outputs* e mecanismos de controlo em cada uma das áreas funcionais, o IDEF3 corresponde à descrição dos processos, através da captura de precedências e relacionamentos entre situações e eventos, representando a sequência das tarefas.

Também é possível utilizar a linguagem UML, nomeadamente o seu digrama de actividades, para fazer a modelização de processo de negócio, embora não permita a especificação com o detalhe que o IDEF permite.

Como resultado desta etapa, tem-se um relatório (Especificação 1 – Fig. 3) com a descrição da resolução do problema devidamente contextualizada no sistema organizacional e alinhada com o modelo de negócios da organização. Passa-se, então, à etapa seguinte, ou seja, à validação da proposta junto dos potenciais utilizadores.

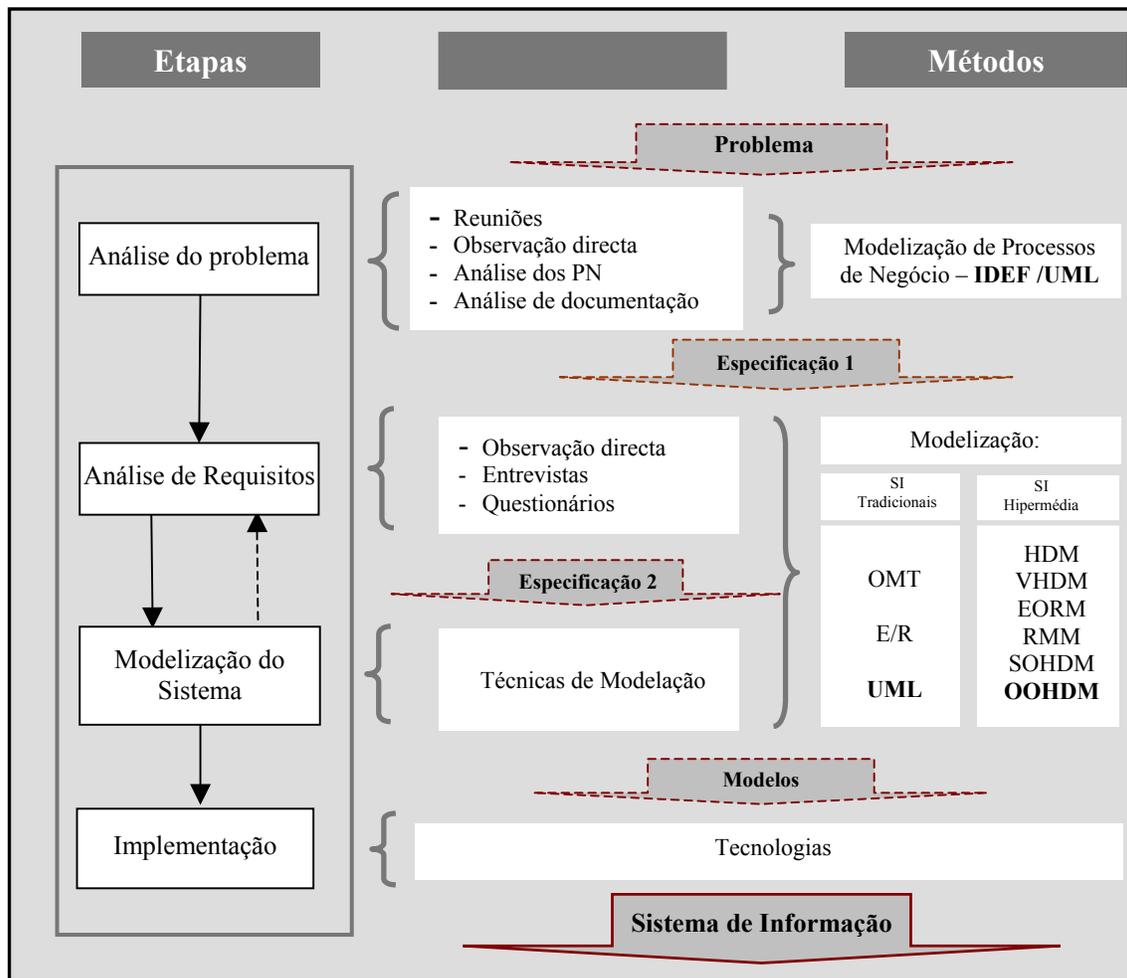


Fig. 3 - Etapas, técnicas e métodos no PDSI.

3.2 Análise de Requisitos

A análise de requisitos é a fase que tem como objectivo validar o resultado da etapa anterior (Especificação 1 – Fig. 3) junto dos potenciais utilizadores do sistema, através da especificação dos requisitos. Nesta fase, o analista terá que trabalhar com

outro grupo de *stakeholders* (utilizadores do sistema) de maneira a ajustar a proposta às necessidades dos mesmos. Como técnicas de recolha de dados, geralmente utilizam-se mecanismos de observação directa, entrevistas ou questionários. Ainda que qualquer uma delas possa conduzir a bons resultados,

é necessário analisar o tipo de clientes do sistema e a dispersão dos mesmos, de maneira a aplicar a mais adequada. Note-se que em muitas situações, nomeadamente quando se está perante aplicações distribuídas na *Web*, existe uma certa tendência para “aligeirar” esta etapa, utilizando técnicas de recolha de dados que incluem apenas os utilizadores internos. Ao contrário dos SI’s tradicionais, que basicamente contam com os utilizadores internos, os SI’s distribuídos podem ser acedidos externamente, tendo por isso um leque de potenciais utilizadores externos à organização que deverão ser considerados [4]. Muitas vezes, questões de facilitismo e conveniência propiciam erros nesta fase, que depois se irão repercutir ao longo do restante processo e instalar no produto final. Portanto, para o sucesso de qualquer sistema é muito importante considerar as necessidades de todos os potenciais utilizadores, independentemente de se tratarem de utilizadores internos ou externos. O questionário é a técnica mais indicada para auscultar as necessidades de uma população dispersa, podendo mesmo tirar-se partido das potencialidades da *Web*, fazendo-o chegar aos inquiridos através desta plataforma.

Enquanto nos SI’s tradicionais existe uma predominância de requisitos funcionais (o ênfase dos não-funcionais reside basicamente na especificação de interfaces), no caso dos SI’s distribuídos na *Web*, para além de um aumento na quota dos requisitos não-funcionais, torna-se muito importante incluir os requisitos de navegação, para além dos de interface [13].

3.3 Modelização do Sistema

A modelização do sistema representa a etapa onde toda a informação textual descrita na especificação é convertida para uma linguagem gráfica, de maneira a facilitar a comunicação entre os diferentes membros da equipa de projecto, nomeadamente os programadores que irão trabalhar sobre os modelos. Ainda que o analista esteja envolvido nesta etapa, a sua participação é menos activa, uma vez que esta tarefa é da responsabilidade directa do projectista. É nesta fase que se faz a ‘ponte’ entre a solução descrita na especificação e a implementação dessa solução, fazendo uso de uma determinada linguagem gráfica (metodologia). O trabalho efectuado nesta fase não acrescenta informação, apenas converte para modelos, existindo alguma propensão para se perder informação, quando não se utilizam os métodos adequados. Por esta razão, é precisamente neste ponto que se deve ter cuidado, para que nenhum trabalho efectuado anteriormente tenha aqui o seu fim. Desta forma, é muito importante utilizar um método adequado ao tipo de SI, de forma a garantir uma correcta passagem de informação entre a fase anterior e a fase da implementação.

Em relação ao grupo de SI’s classificados como SI’s tradicionais, existem vários métodos Orientados a Objectos (OO) e já normalizados, sendo a *Entity-Relationship* (E/R) [9], a *Object Modeling Technique* (OMT) [8] ou a *Unified Modeling Language* (UML) [7], os mais utilizados. Dentro destes, o último responde de forma muito satisfatória à modelização deste tipo de SI’s, uma vez ter resultado do que melhor existia nas práticas anteriores, da qual a OMT fazia parte. A linguagem UML, ao longo das suas cinco visões e nove diagramas, apresenta soluções de modelização OO para qualquer sistema (e não apenas para *software*) fazendo uso de uma notação simples e fácil de usar, com a particularidade de ser independente de qualquer ferramenta CASE [7] e [14]. A OMT, uma das linguagens antecessoras da UML, apresenta uma notação muito semelhante, mas não cobre todas as visões do sistema como a UML o faz. A E/R foi desenvolvida para modelizar Bases de Dados, apresentando-se pouco prática para a modelização conceptual em outros domínios.

Se passarmos para o grupo dos SI’s hipermédia, será um risco apresentar qualquer uma das soluções anteriores como a solução de modelização ideal, uma vez que há aspectos importantes nestes sistemas que esses métodos não cobrem. Até mesmo a UML, com a sua capacidade de modelizar qualquer tipo de sistema, carece de notação para modelizar aspectos ‘chave’ dos sistemas hipermédia, como os requisitos não-funcionais que predominam neste tipo de sistemas, nomeadamente a nível da estrutura de navegação [15]. Note-se que são aspectos que, quando não considerados, podem conduzir à implementação de funções diferentes daquelas que eram desejadas, precisamente pela predominância e relevância que têm neste tipo de SI’s.

Foi precisamente este assunto, conjuntamente com a necessidade de estender a modelização OO a aspectos navegacionais nos sistemas hipermédia, que despertou a atenção de alguns investigadores que trabalham nesta área de conhecimento e que conduziu ao aparecimento de algumas propostas de modelização. Ainda que estas propostas tenham surgido para colmatar a lacuna das propostas tradicionais face a este tipo de SI’s, elas fazem-no de maneiras distintas. Uma dão mais importância à notação, deixando para trás o processo, outras focam mais aspectos de *design*, enquanto outras se preocupam com o processo, mesmo que para tal tenham que reaproveitar a notação já existente nas metodologias tradicionais [3]. Por exemplo, o trabalho de Conallen [16, 17] foi uma das tentativas que resultou da extensão da UML, definindo um conjunto de estereótipos para a *Web*, mas que deixou para trás o método para projectar as aplicações ao longo das suas diferentes fases de desenvolvimento. Estes estereótipos são apropriados para definir aspectos de implementação de aplicações *Web* (como por exemplo, *server page* e *client page*), mas perdem pelo facto de não contemplarem as características mais importantes deste tipo de aplicações como, por exemplo, a estrutura de navegação (a navegabilidade).

Existem outros trabalhos (ver Quadro 1) que apresentam técnicas de modelação com notações baseadas nas tradicionais técnicas *Object-Oriented* -OO- ou *Entity-Relationship* -E/R- e que já se preocupam mais com o aspecto navegacional, apresentando uma notação (própria ou não) para modelizar os três níveis típicos de uma aplicação *Web*: nível conceptual, nível estrutural e nível de apresentação [3, 18]:

- (i) o nível conceptual é responsável pelo armazenamento e estruturação dos dados;
- (ii) o nível estrutural é responsável pela especificação do caminho para chegar até aos dados da camada anterior e, vice-versa;
- (iii) o nível da apresentação é a camada responsável pela interacção com os diferentes actores que se encontram no exterior das fronteiras do sistema.

Métodos	Notação
OOHDM (Object-Oriented Hypermedia Design Method) [20]	UML (ou OMT) + própria + ADV’s
WSDM (Web Site Design Method) [21]	E/R (ou UML) + própria
HDM (Hypermedia Design Method) [22]	E/R
EORM (Enhanced-Object Relationship Model) [23]	OMT
RMM (Relationship Management Methodology) [24]	E/R + própria
SOHDM (Scenario-based Object Oriented Hypermedia Design Methodology) [25]	Própria

Quadro 1 - Algumas métodos de desenvolvimento de SI’s Hipermédia.

Do ponto de vista da arquitectura da modelização SI's hipermédia é o mesmo que falar em modelo conceptual, modelo navegacional, e modelo de interfaces [19], correspondendo simultaneamente à componente dos dados, processos e interfaces na nomenclatura de Whitten [5].

Em suma, todos estes métodos contribuem com uma excelente ideia - reaproveitar o paradigma da OO e/ou E/R, e aplicá-lo aos conceitos que os métodos tradicionais não cobrem. Alguns fazem uso da notação existente, sem qualquer notação nova (HDM, EORM); outros acrescentam à já existente uma notação própria (OOHDM, WSDM, RMM), existindo ainda o caso do SOHDM que propõe uma simbologia completamente nova, sem qualquer reaproveitamento da já existente.

Será ainda importante referir que estes foram seleccionadas de entre muitos outros propostos, pelo facto de terem algo em comum - o tratamento da informação a nível conceptual, navegacional e de interfaces, embora a maneira como o fazem seja diferente. A título de exemplo deve referir-se a excelente visão de Schwabe e Rossi [20] com a metodologia OOHDM, ao tentarem 'apanhar' aspectos cognitivos da navegação, com o conceito de contexto navegacional, já que uma das maiores dificuldades na modelização de aplicações *Web* reside, precisamente, na dificuldade em traduzir para modelos a rede navegacional que é apresentada ao utilizador para chegar a uma determinada informação no domínio conceptual. A metodologia *Object-Oriented Hypermedia Design Method* (OOHDM) apresenta uma proposta de modelização evidenciando aspectos importantes a nível da navegação. Esta particularidade reflecte uma das suas maiores inovações, pressupondo que os objectos navegáveis pelo utilizador não são exactamente os objectos representados no modelo conceptual [20, 26]. Mostra-se bastante poderosa na construção de aplicações *Web*, tendo sido nos últimos anos usada de uma forma intensa para desenhar *Web* sites, quiosques interactivos, aplicações de comércio electrónico, entre outras [27]. É uma metodologia que permite o desenvolvimento incremental da aplicação, com a obtenção de progressos em pequenos passos, através da subdivisão do problema em subproblemas e a posterior combinação das soluções [28, 29]. Para tal, subdivide as tarefas desta etapa em quatro diferentes actividades: desenho conceptual, desenho navegacional, desenho de interfaces abstractas e a implementação; as três primeiras desenvolvidas interactivamente e a última após o término daquelas.

- Durante a fase do desenho conceptual é construído o modelo do problema que descreve o domínio da aplicação, englobando todo o universo de informações relevantes para o sistema a desenvolver. Nesta fase não se consideram de forma particular os diferentes tipos de utilizadores e tarefas por eles desempenhadas, trabalhando-se apenas a nível do domínio da aplicação.
- O desenho navegacional é a fase onde se esquematiza o conceito de navegabilidade, através da determinação de uma estrutura a partir da qual a informação vai ser apresentada aos utilizadores. Diferentes modelos navegacionais podem ser construídos para um mesmo esquema conceptual, expressando desta forma diferentes visões da aplicação num mesmo domínio do problema [15]. Deriva do modelo conceptual através de um conjunto de mecanismos de definição de visões, tendo em conta os diferentes perfis de utilizadores (actores). Para tal utiliza dois tipos diferentes de esquemas: (1) esquema de classes navegacionais, responsável pela especificação das classes e atributos vistos pelo diferentes perfis de utilizadores, definindo as visões do

modelo conceptual; (2) esquema de contextos navegacionais, que define a estrutura geral da aplicação através da especificação de um conjunto de contextos navegacionais. Os contextos navegacionais surgem com o intento de tornar os elementos de informação de um documento de hipertexto mais facilmente entendidos quando apresentados num determinado contexto.

- O modelo de Interfaces Abstractas especifica os objectos de interface vistos pelo utilizador e, particularmente, a forma que tomarão os objectos navegacionais quando activados pelas interfaces. O OOHDM faz uso de ADV's (*Abstract Data View*) para ajudar na especificação.
- Por último, a etapa da implementação, responsável pela tradução dos modelos anteriormente descritos para um ambiente de programação. Esta etapa tem em conta o contributo do potencial tecnológico actualmente existente, fazendo uso das tecnologias mais apropriadas, no sentido de melhorar as funcionalidades do sistema, as suas potencialidades, assim como facilitar futuras intervenções.

Não sendo objectivo deste trabalho descrever de forma detalhada as diferentes metodologias, mas antes sugerir uma alternativa de modelização adequada a cada tipo de SI's aqui abordado, defende-se que: no caso dos SI's tradicionais, não será difícil justificar a Linguagem UML; em relação aos sistemas distribuídos na *Web*, a metodologia OOHDM mostra-se bastante eficiente uma vez que contempla os modelos adequados, nomeadamente os navegacionais, que permitem modelizar os requisitos não-funcionais (de navegabilidade) predominantes neste tipo de sistemas. Esta hipótese foi validada, aquando da utilização da mesma na modelização de uma aplicação *Web* para a gestão da informação académica [4, 13].

3.4 Implementação

A fase da implementação é da responsabilidade dos construtores ou programadores, tendo estes que interpretar a informação contida nos modelos e convertê-la para uma linguagem executável pelo computador. É nesta fase que se faz a escolha da tecnologia mais apropriada, podendo-se recorrer à ajuda dos vendedores ou consultores tecnológicos. Qualquer SI, independentemente do tipo, necessita de futuras intervenções a nível de manutenção de conteúdos; por esta razão, no momento da implementação esse facto deve ser compreendido em toda a sua extensão e por forma a que o problema possa ser devidamente enfrentado. Infelizmente, no caso dos SI distribuídos na *Web*, o problema aumenta de complexidade, e muitas vezes alterações simples a nível de interfaces podem envolver muito trabalho. Se imaginarmos aplicações de grande dimensão, na ordem das centenas de instâncias por classe, rapidamente se conclui que a manutenção a nível das instâncias seria impensável. O ideal seria a escolha de tecnologias, tais como, ASP, ASP.NET, PHP, JSP, que permitam actualizações ou outro tipo de manutenção a nível das classes, o que, por sua, permitirá simplificar bastante intervenções futuras no SI.

4. CONCLUSÃO

O Processo de Desenvolvimento de um Sistema de Informação (PDSI) descreve o caminho a percorrer desde o conhecimento da existência do problema, até à instalação da solução desejada. Ao longo deste artigo foi caracterizado esse caminho para dois tipos de SI's diferente (SI's tradicionais e SI's distribuídos na

Web) subdividindo as actividades do processo em quatro etapas básicas: análise do problema, análise de requisitos, modelização do sistema e implementação. Estas etapas, particularmente as três primeiras, contam com um conjunto de técnicas e métodos para poderem ser percorridas, que, por sua vez, devem ser escolhidas de acordo com o tipo de sistema que se pretende desenvolver. Em relação à etapa de análise do problema, devem ser utilizadas técnicas de recolha de informação, no sentido de construir a especificação do problema, tais como: reuniões, técnicas de observação directa ou análise de documentação existente. Nesta etapa, ainda não se encontram divergências no que respeita às técnicas de obtenção de dados para os diferentes SI's. Em paralelo com as técnicas de recolha de dados, é necessário fazer o estudo dos processos organizacionais no sentido de contextualizar a solução de acordo com os propósitos da organização. Para tal recorre-se a um processo de modelização, tendo a IDEF dois modelos adequados e voltados para esse fim (IDEF0 e IDEF3). Na fase de análise de requisitos, o analista terá que recolher informação, junto aos potenciais utilizadores no sentido de completar e validar a especificação vinda da etapa anterior. Existem várias técnicas para apoiar o processo de recolha de dados, sendo as técnicas de observação directa, questionários e entrevistas, as mais utilizadas. Qualquer uma das delas, desde que bem aplicada e adequada à situação, pode conduzir a bons resultados, independentemente do tipo de SI a tratar. Desta etapa resulta uma especificação devidamente contextualizada na organização e validada pelos potenciais utilizadores, que passará a ser trabalhada pelos projectistas. A etapa que se segue é a modelização do sistema, da responsabilidade dos projectistas, onde a especificação resultante da etapa anterior é convertida para uma linguagem gráfica. Ao contrário do que acontece nas duas anteriores, o tipo de SI's que se pretende desenvolver influencia a escolha da metodologia. Se em relação aos SI's tradicionais, não será difícil de justificar a UML como a metodologia adequada, uma vez que contém os principais modelos que cobrem os requisitos predominantes neste tipo de sistemas. Em relação aos SI's distribuídos na *Web*, a metodologia OOADM revela-se bastante satisfatória pela capacidade que apresenta, através da captação dos aspectos cognitivos da navegação com o conceito de contexto navegacional, em traduzir para modelos os requisitos não-funcionais (de navegabilidade) predominantes neste tipo de sistemas.

5. REFERÊNCIAS

- [1] J. Varajão, **Sistemas de Informação - A Arquitectura da Gestão de Sistemas de Informação**, Lisboa: FCA – Editora de Informática, 1998.
- [2] J. O'Brien, **Management Information Systems - Managing Information Technology in the Internet worked Enterprise** (4th ed.), McGraw - Hill, International Edition, 1999.
- [3] N. Koch, "A Comparative Study of Methods for Hypermedia Development", **Technical Report 9905**, pps: 19, Ludwig-Maximilians-Universität München, 1999.
- [4] L. Teixeira, **Gestão de Informação Académica com Base na Web - um sistema de apoio a programas de pós-graduação**, MSc. Thesis, Universidade de Aveiro, 2002.
- [5] J. L. Whitten; L. D. Bentley; K. C. Dittman, **Systems Analysis and Design Methods** (5th ed.), McGraw-Hill – Irwin, 2001.
- [6] K. Laudon; J. Laudon, **Management Information Systems - Managing the Digital Firm** (7th ed.), Upper Saddle River: Prentice Hall, Inc., 2002.
- [7] G. Booch; J. Rumbaugh; I. Jacobson, **The Unified Modeling Language Guide, Reading** (MA): Addison-Wesley, 1999.
- [8] J. Rumbaugh; M. Blaha; W. Premerlani; F. Eddy; W. Lorensen, **Object Oriented Modeling and Design**, New Jersey: Prentice Hall, Inc., 1991.
- [9] B. Richard, **CASE Method: entity relationship modelling**, Wokingham: Addison-Wesley, 1990.
- [10] N. Koch; A. Kraus; R. Hennicker, "The Authoring Process of the UML-based Web Engineering Approach", **First International Workshop on Web-Oriented Software Technology - IWWOST'2001**, Valencia, Espanha, 2001.
- [11] O. Noran, "Business Modelling: UML vs. IDEF", Griffith University, <http://www.cit.gu.edu.au/~noran/Docs/UMLvsIDEF.pdf> (Maio de 2004), 2000.
- [12] IDEF, "IDEF Family of Methods - A structured approach to enterprise modelling and analysis", **Knowledge Based Systems, Inc.** (KBSI), <http://www.idef.com>, (Maio de 2004), 2000.
- [13] L. Teixeira; C. Ferreira; R. Santiago, "Desenvolvimento de aplicações Hipermedia utilizando a metodologia OOADM: a possibilidade de incluir os requisitos funcionais e não funcionais de um sistema", **Actas do congresso da 3ª Conferência da Associação Portuguesa de Sistemas de Informação – CAPSI'02** (Em CD-ROM), Coimbra. Portugal, 2002 {ISBN 972-97548-7-X}
- [14] M. Fowler; K. Scott, **UML Distilled - Applying the Standard Object Modeling Language**, Reading (MA): Addison-Wesley Longman, Inc., 1997.
- [15] A. Diaz; S. Gordillo; G. Rossi, "Specifying navigational structures by querying hypermedia design models", **Proceedings of the Third Basque International Workshop on Information Technology - BIWIT'97**, 1997, pp.125 - 130.
- [16] J. Conallen, "Modeling web application architectures with UML", **Communications of the ACM**, Vol. 42, No.1, 1999, pp. 63-70.
- [17] J. Conallen, **Building Web application with UML**, Reading (MA): Addison-Wesley, 2000.
- [18] H. Baumeister; N. Koch; L. Mandel, "Towards a UML extension for hypermedia design", <http://www.fast.de/Projekte/forsoft/uml99>, (Maio, 2004), 1999.
- [19] K. Spencer, "An overview of the Windows Distributed Internet Applications Architecture", <http://www.webtechniques.com/archives/1999/05/spencer>, (Maio, 2004), 2003.
- [20] D. Schwabe; G. Rossi, "The object-oriented hypermedia design model", **Communications of the ACM**, Vol. 38, No.8, 1995, pp. 45 - 46.
- [21] O. F. De Troyer; C. J. Leune, "WSDM: A User-Centered Design Method for Web Sites", **Proceedings of the 7th International World Wide Web Conference – WWW7**, Brisbane, Australia, 1998.
- [22] F. Garzotto; P. Paolini; D. Schwabe, "HDM - A model-based approach to hypertext application design", **ACM Transactions on Information Systems**, Vol. 11, No. 1, 1993, pp. 1-26.

- [23] D. Lange, "Object-oriented hyper-modeling of hypertext supported information systems", **Proceedings of the Twenty-Sixth Annual Hawaii International Conference on System Sciences**, Vol. 3, 1993, pp. 380-389.
- [24] T. Isakowitz; E. Stohr; P. Balasubramanian, "RMM – A methodology for structured hypermedia design", **Communications of the ACM**, Vol. 38, No. 8, 1995, pp. 34-44.
- [25] H. Lee; C. Lee; C. Yoo, "A scenario-based object-oriented hypermedia design methodology", **Information & Management**, Vol 36, 1999, pp. 121-138.
- [26] G. Rossi, **OOHDM – Object Oriented Hypermedia Design Method-** (in Portuguese), PhD, Thesis, PUC-Rio, 1996.
- [27] G. Rossi; D. Schwabe; F. Lyardet, A. Silva, "Web application models are more than conceptual models", **Proceedings of the World Wide Web and Conceptual Modeling'99 Workshop - ER'99**, Paris, France, 1999, pp. 239-252.
- [28] D. Schwabe; G. Rossi; L. Esmeraldo; F. Lyardet, "Web design frameworks: an approach to improve reuse in web application", **Proceedings of the Second International Workshop on Web Engineering**, 2000, pp. 1-12.
- [29] D. Schwabe; G. Rossi; L. Esmeraldo; F. Lyardet, "Engineering web applications for reuse", **IEEE Multimédia**, 2001, pp. 2-12.