

DSDM – Dynamic Systems Development Methodology

Daniel Dinis Teixeira, Fernando Jorge Afonso Pires, José Pedro Gaiolas de Sousa
Pinto, Tiago Alexandre Gonçalves Pereira Santos

Faculdade de Engenharia da Universidade do Porto

Resumo. Foi objectivo deste trabalho explorar a metodologia DSDM (*Dynamic Systems Development Methodology*). Depois de uma breve explicação sobre as metodologias ágeis, onde a DSDM se insere, foram apresentados os seus princípios e processos, bem como os prós e contras da sua utilização. Em anexo, decidimos apresentar um exemplo real dado pela consultora norte-americana Michelle Johnston para a importância da DSDM em soluções de comércio electrónico.

1. Introdução

A **DSDM (Dynamic Systems Development Methodology)** insere-se no ramo das metodologias ágeis de apoio ao desenvolvimento de software. Esta metodologia, visa desenvolver uma aplicação com a qualidade desejada sem ultrapassar limites de tempo e orçamento. Para o conseguir, a DSDM foca-se na interacção com o cliente e o utilizador final, entrega de protótipos frequentes, equipas de desenvolvimento autónomas, testes massificados durante todo o processo e na definição de prioridades entre a lista de requisitos dada pelo cliente. As três fases que constituem esta metodologia, assim como os seus processos, princípios e casos de utilização, são explicados neste artigo de síntese, realizado para a disciplina de Engenharia de Software, do 3º ano da Licenciatura de Engenharia Informática e Computação da Faculdade de Engenharia da Universidade do Porto.

2. Metodologias Ágeis

A maioria do desenvolvimento de software é feita de forma caótica, vulgarmente caracterizada pelo método *programar e corrigir*. O código é feito sem um planeamento aprofundado do problema, sendo muitas vezes agregado a porções de código distinto para a resolução de pequenos problemas que surgem conforme o software vai sendo desenvolvido.

Este procedimento é eficaz para pequenas aplicações mas, com o aumento do sistema construído, torna-se bastante mais difícil adicionar novas funcionalidades ao programa ou alterar outras já existentes. Além disso, os *bugs* tornam-se bastante mais frequentes e bem mais difíceis de resolver.

Uma longa fase de testes posterior à concepção do programa é uma característica deste tipo de programação, bem como um sinal de que foi usada. Dado ser muito difícil estabelecer prazos exactos para uma fase de testes e *debugging* tão longa, o tempo previsto para a execução do trabalho é também difícil de determinar.

Este estilo de desenvolvimento de projectos foi predominante durante muito tempo, tendo como mais forte concorrente o *Desenvolvimento Metódico*. O *Desenvolvimento Metódico* impõe um processo de desenvolvimento disciplinado, com o objectivo de tornar o desenvolvimento de software previsível e eficiente, conseguido através do desenvolvimento de documentação detalhada e dando um grande ênfase ao planeamento inspirado noutras áreas da engenharia de software.

O desenvolvimento metódico já existe há bastante tempo, mas nunca chegou a ser frequentemente utilizado devido à sua falta de eficácia – é uma metodologia burocrática sendo que, para a seguir, é necessário cumprir diversos itens que tornam o desenvolvimento do software muito mais demorado.



Fig. 1. Metodologias Ágeis

Derivada desta metodologia surgiu um novo grupo apelidado de **Metodologias Ágeis** (esta designação deriva da comparação, em termos burocráticos, com o desenvolvimento metódico). Estes novos métodos tentam encontrar um ponto de equilíbrio

óptimo entre o trabalho diminuto e o trabalho excessivo, rentabilizando o pagamento em relação ao trabalho feito.

Como resultado, as metodologias ágeis têm sido adoptadas em diversas aplicações e alterado significativamente as metodologias de engenharia de software. A diferença mais imediata pode ser observada pela falta de necessidade em gerar muita documentação para fazer as mesmas tarefas. Estas metodologias são mais orientadas ao código propriamente dito, planeando os passos para o desenvolvimento deste e documentando-o.

A falta de documentação das metodologias ágeis resulta de algumas diferenças substanciais em relação ao desenvolvimento metódico:

1 – As Metodologias Ágeis são propositadamente flexíveis, ao invés de estáticas e previsíveis. O desenvolvimento metódico tende a tentar planejar uma grande parte da criação de software em grande detalhe, para um longo período de tempo, situação que só é satisfatória enquanto os requisitos não mudarem. Pelo contrário, os métodos ágeis estão preparados para as mudanças. Tendem a ser processos que se adaptam e melhoram com as alterações.

2 – As Metodologias Ágeis são orientadas às pessoas e não aos processos. O objectivo do desenvolvimento metódico é definir processos que funcionaram bem quaisquer que sejam os utilizadores. Estes métodos baseiam-se na suposição de que nenhum processo alguma vez irá ultrapassar as capacidades da equipa de desenvolvimento. Portanto, o seu papel neste processo é suportar a equipa de desenvolvimento no seu trabalho e dar-lhes a perceber o que realmente necessitam.

3. Dynamic Systems Development Methodology

3.1 Introdução à Metodologia

A DSDM fornece uma *framework* para uma abordagem interactiva e incremental de desenvolvimento de **Sistemas de Informação (SI)**. Desenvolveu-se nos anos 90 na Inglaterra e foi aplicado pela primeira vez em 1995. Nesta altura (Novembro de 2005), o **Manual DSDM** encontra-se na 4ª versão. Esta metodologia foi desenvolvida por um consórcio de vendedores e peritos no campo dos Sistemas de Informação, no qual partilharam e combinaram as suas melhores técnicas. Assim, a DSDM surge como uma extensão do **RAD (Rapid Application Development)**, focada em projectos de Sistemas de Informação caracterizados por prazos e orçamentos apertados.

A DSDM aborda os problemas que frequentemente ocorrem no desenvolvimento de informação que se prendem essencialmente com a falta de tempo, com orçamentos mais apertados ou com outro tipo de razões para que o projecto falhe, tal como a falta de envolvimento dos encarregados do projecto ou dos utilizadores finais.

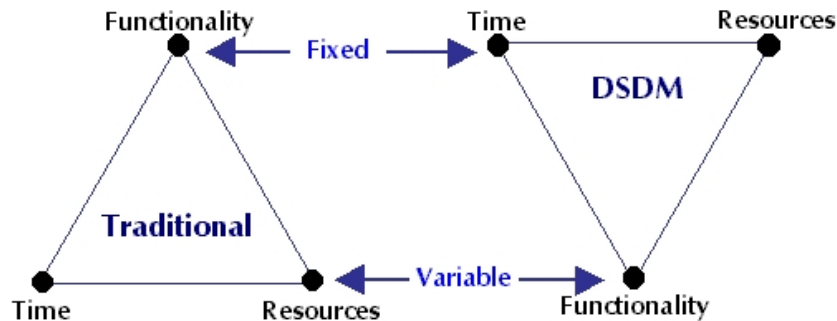


Fig. 2. Princípio fundamental da DSDM

3.2 Princípios da DSDM

A DSDM segue alguns princípios chave. Estes princípios delimitam as bases do desenvolvimento utilizando DSDM.

- O ponto fundamental desta metodologia prende-se com a entrega de um sistema que se aproxime das actuais necessidades de negócio. Não é uma metodologia tão directa que forneça todas as necessidades de negócio, mas centraliza todo o potencial na concretização final de todos os objectivos do projecto.
- Nenhum sistema é completamente construído na primeira tentativa. Num processo de desenvolvimento de um sistema informático 80% da solução pode ser desenvolvida em 20% do tempo necessário para encontrar a solução perfeita. Para aperfeiçoar a parte final poderá ser necessário que o projecto ultrapasse o seu tempo e orçamento estipulados. Uma vez que a DSDM é caracterizada por realizar exactamente o que a empresa necessita, é muitas vezes desnecessário chegar à solução perfeita.
- A entrega do projecto deve ser feita na data estipulada, dentro do orçamento previsto e com boa qualidade (Fig. 2).
- As exigências para o Sistema de Informação têm que ser flexíveis. Tal como falaremos mais tarde, exigências flexíveis são tópicos importantes da DSDM.
- Esta metodologia apenas requer que cada etapa do desenvolvimento seja completada até que seja possível iniciar o passo seguinte. Isto faz com que cada fase do projecto possa começar sem ter que esperar que as fases que começaram anteriormente sejam totalmente terminadas.
- A comunicação entre todas as partes envolvidas (*stakeholders*) é também um pré-requisito bastante importante para que o projecto corra com a eficiência desejada.
- O envolvimento dos utilizadores é a chave para esta eficiência.

- As equipas responsáveis têm que ser dotadas de um sentido de decisão, sendo este também um ponto fulcral na progressão do projecto.
- Tal como as equipas de desenvolvimento também as equipas de gestão do projecto estão incorporadas na DSDM.
- Após o desenvolvimento do Sistema de Informação, a DSDM pode também ser usado para expandir o Sistema obtido.

3.3 Pré-requisitos para a Utilização da DSDM

Para que a DSDM seja um sucesso, um número definido de pré-requisitos devem ser implementados. Em primeiro lugar, é necessária a existência de uma interactividade entre a equipa de desenvolvimento, os utilizadores finais e os gestores de projecto. A falta de motivação da equipa de gestão ou a falta de envolvimento dos utilizadores-finais é uma das mais frequentes causas de falha do desenvolvimento de projectos de Sistemas de Informação. Um segundo, e importante, pré-requisito é a facilidade de decomposição em módulos. Esta possibilidade de decompor o projecto em partes mais pequenas, possibilita as abordagens iterativa e incremental da DSDM. Esta funcionalidade possibilita a criação de diversos projectos mais pequenos, também eles a serem desenvolvidos segundo os princípios da DSDM. A terceira, e não menos importante, propriedade relaciona-se com a possibilidade de definir claramente os requisitos do SI, introduzindo-lhes uma ordem segundo a sua prioridade de implementação.

Os projectos que não preencham estes requisitos ajustam-se menos ao desenvolvimento utilizando a DSDM. Por exemplo, projectos complexos que são difíceis ou impossíveis de decompor não poderão ser desenvolvidos usando uma abordagem DSDM. De uma forma semelhante, requisitos pouco claros ou que não possam ser escalonados por uma ordem de preferência podem fazer com que os projectos ultrapassem o tempo e orçamento estipulados, efeitos que a DSDM foi feita para evitar.

Outro grupo de projectos para o qual a DSDM não é apropriada é aquele cujo produto final necessitará de um estado de segurança-crítica. A intensiva fase de testes e validação necessária neste tipo de projectos colide com os objectivos da DSDM em estar pronta a tempo e sem ultrapassar o orçamento estipulado. Por fim, os projectos que apontam para componentes reutilizáveis podem não ser apropriados para um desenvolvimento utilizando a DSDM já que as necessidades de perfeição neste tipo de projectos colidem com o princípio 80%/20% descrito anteriormente.

3.4 As fases da DSDM

O *framework* DSDM consiste em três fases sequenciais: **Pré-Projecto**, **Projecto** e **Pós-Projecto**. A fase de Projecto do DSDM é a mais elaborada das três fases. Ela consiste em 5 níveis formadas por uma abordagem passo-a-passo e iterativa no desenvolvimento de um SI. As três fases e correspondentes níveis são explicadas exaustivamente nas secções seguintes.

Fase 1: O Pré-Projecto

Na fase do pré-projecto, o projecto candidato é identificado, tratando-se depois do seu plano de financiamento e sendo assegurado um compromisso de realização. Tratar destas questões numa fase inicial evita problemas futuros em fases mais avançadas do projecto.

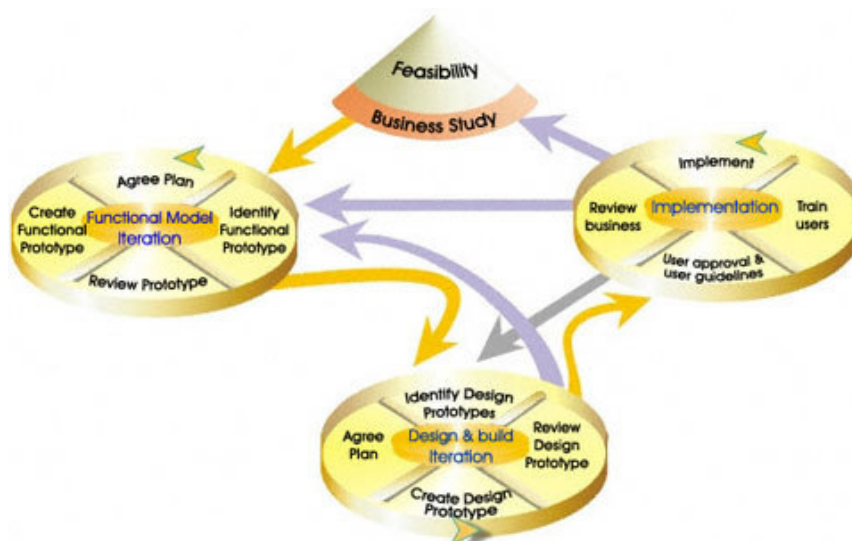


Fig. 3. Ciclo de Vida de um Projecto em DSDM

Fase 2: O Ciclo de Vida do Projecto

A visão geral de um processo DSDM, presente na Fig. 3, representa o **Ciclo de Vida** do Projecto nesta segunda fase da metodologia. Ela mostra os 5 níveis que a equipa de desenvolvimento terá de percorrer para criar um SI. Os dois primeiros níveis, o **Estudo de Viabilidade** e o **Estudo de Negócio**, são fases sequenciais que se complementam. Depois destas fases estarem concluídas, o sistema é desenvolvido iterativamente e de forma incremental nos níveis de Análise Funcional, Desenho e Implementação.

Nível 1: O Estudo de Viabilidade

Durante este nível do projecto, a viabilidade de utilização da DSDM para este projecto é examinada. Os pré-requisitos para o uso da DSDM são encontrados respondendo a questões como: “Pode este projecto ir de encontro às necessidades de negócio apontadas?”, “É, este projecto, adequado ao uso da DSDM?” e “Quais são os riscos mais importantes envolvidos?”. As técnicas mais importantes utilizadas nesta fase são os *workshops*.

Para entrega ao cliente, são preparados neste nível o **Relatório** e o **Protótipo de Viabilidade** que dizem respeito à viabilidade do projecto em mãos. A estes, adicionam-se um esboço global do plano para o resto do projecto e um **Registo de Risco** que identifica os riscos mais importantes no projecto.

Nível 2: O Estudo do Negócio

O Estudo do Negócio incrementa todo o trabalho realizado no Estudo de Viabilidade. Depois do projecto ser declarado fiável para o uso da DSDM, este nível examina o processo de financiamento, os utilizadores envolvidos e as suas necessidades e desejos respectivos. Uma vez mais, os *workshops* são uma das mais valiosas técnicas. *Workshops* nos quais os diferentes *stakeholders* se reúnam e discutam o sistema proposto. A informação retirada destas sessões é combinada numa lista de requisitos. Uma importante propriedade desta lista de requisitos é a possibilidade de se definir prioridades. Estas prioridades são definidas utilizando uma perspectiva *MoSCoW*¹. Baseado neste escalonamento, um plano de desenvolvimento é construído como uma linha mestra para o resto do projecto. Uma importante técnica utilizada no desenvolvimento do plano é a técnica de *Timeboxing*². Esta técnica é essencial para serem atingidos os objectivos da DSDM, nomeadamente a imposição de tempo e orçamento fixos, garantindo no entanto a qualidade desejada. Uma arquitectura de sistema é outro meio para guiar o desenvolvimento do Sistema de Informação.

No final deste nível, deverão estar prontos para entrega ao cliente: uma definição de área de negócio que descreve o contexto do projecto dentro da companhia, a definição da arquitectura do sistema que fornece uma arquitectura global inicial do SI em desenvolvimento juntamente com o plano de desenvolvimento que realça os passos mais importantes no processo de desenvolvimento. Na base destes dois últimos documentos está a lista de prioridades dos requisitos. A lista define todos os requisitos do sistema, organizados de acordo com o princípio do *MoSCoW*. Por fim, o Registo de Risco é actualizado com os factos que foram identificados durante esta fase da DSDM.

¹ Método de definição de prioridades, definido na secção 3.5

² Encapsulamento de tempo, definido na secção 3.5

Nível 3: Análise Funcional

Os requisitos que foram identificados nos níveis anteriores são convertidos para um **Modelo Funcional**. A **Prototipagem** é uma das técnicas chave dentro deste nível, que ajuda no bom envolvimento do utilizador final com o projecto. O protótipo desenvolvido é revisto pelos diferentes grupos de utilizadores. Para assegurar a qualidade do projecto, os testes são implementados em cada iteração da DSDM. Uma importante parte dos testes são realizados na Análise Funcional. O Modelo Funcional pode ser subdividido em quatro sub níveis:

Identificar Protótipo Funcional: determinar as funcionalidades a ser implementadas no protótipo que resulta desta iteração.

Acordar Calendário de Tarefas: definir como e quando desenvolver estas funcionalidades.

Criar Protótipo Funcional: desenvolver o protótipo.

Rever o Protótipo: procurar correcções possíveis no protótipo desenvolvido. Isto pode ser feito através de testes com utilizadores finais e revendo a documentação.

Neste nível, é necessário entregar ao cliente o Modelo Funcional e o Protótipo Funcional que, juntos, representam as funcionalidades que podem ser realizadas nesta iteração, prontas para serem testadas pelos utilizadores. Além destes dois documentos, a Lista de Requisições é actualizada, sendo apagados os itens que foram implementados e repensando as prioridades dos requisitos restantes.

Nível 4: Desenho

O ponto central desta iteração da DSDM é a integração das componentes funcionais do nível anterior num sistema que satisfaça as necessidades do utilizador. Mais uma vez, os testes são uma das actividades mais importantes. O Desenho pode ser dividido em quatro sub níveis:

Identificar Protótipo de Desenho: identificar requisições funcionais e não-funcionais que são necessários no sistema testado.

Acordar Calendário de Tarefas: definir como e quando desenvolver estes requisitos.

Criar Protótipo de Desenho: criar um sistema que possa, com segurança, ser fornecido aos utilizadores finais para um uso diário.

Rever Protótipo de Desenho: verificar a exactidão do sistema desenhado. Mais uma vez, os testes e revisões são peças fundamentais.

Ao utilizador, serão entregues o Protótipo de Desenho para que estes efectuem testes ao produto-protótipo.

Nível 5: Implementação

No nível de Implementação, o sistema testado e a sua documentação são entregues aos utilizadores finais que deverão começar a ser treinados para a futura utilização do novo SI. O sistema a ser entregue foi revisto para incluir todos os requisitos que foram definidos nos primeiros níveis do projecto. O nível de implementação pode ser dividido em quatro sub níveis:

Aprovação do utilizador: os utilizadores finais aprovam o sistema testado para implementação e as linhas mestras para a implementação e uso do sistema são criadas.

Treinar os utilizadores: treinar os futuros utilizadores no uso do sistema.

Implementação: implementar o sistema testado no local de trabalho dos utilizadores finais.

Rever Negócio: rever o impacto do sistema implementado no negócio, um problema central será tentar compreender se o sistema vai de encontro aos objectivos definidos no início do projecto. Dependendo disto, o projecto passará para a fase seguinte, o Pós-Projecto ou voltará a uma das fases anteriores para desenvolvimento posterior.

No final deste nível, o sistema deverá ser entregue e instalado, pronto para o uso de todos os utilizadores finais e a **Documentação de Utilização do Sistema** deverá ser detalhada.

Fase 3: Pós-Projecto

A fase de pós-projecto assegura um sistema de actuação eficiente. Isto é implementado através da manutenção e melhoramentos de acordo com os princípios da DSDM. Até mesmo a iniciação de novos projectos, para actualizar o sistema existente ou desenvolver um novo sistema, é possível.

3.5 Lista de Conceitos Utilizados

Timeboxing

Timeboxing, ou encapsulamento do tempo, é uma das técnicas utilizadas nos projectos baseados na metodologia DSDM. Esta técnica é utilizada para suportar um dos objectivos principais da DSDM: realizar o desenvolvimento de um Sistema de Informação no tempo previsto, dentro do orçamento e com a qualidade desejada. A principal ideia por de trás do *Timeboxing* é dividir o projecto em porções, cada uma com um orçamento fixo e uma data de entrega estipulada. Para cada porção, é seleccionado um número de requisitos que são escalonados de acordo com o princípio de *MoS-CoW*. Uma vez que o tempo e o orçamento são fixos, a única variável restante é aquela que representa os requisitos. Portanto, se um projecto está a ficar sem tempo ou dinheiro, os requisitos com menor prioridade são omitidos. Isto significa, efectivamente, que um produto incompleto é entregue, devido aos princípios subjacentes à DSDM de que 80% do projecto pode ser realizado em 20% do tempo que leva a cons-

truir o produto completo e de que nenhum sistema é construído na perfeição à primeira tentativa.

MoSCoW

A técnica *MoSCoW* representa um método de definição de prioridades nas tarefas efectuadas. Neste contexto, a técnica *MoSCoW* da DSDM é usada para dar prioridade aos requisitos enunciados. É um acrónimo que representa:

MUST have this.

SHOULD have this if at all possible.

COULD have this if it does not affect anything else.

WON'T have this time but WOULD like in the future.

TEM de ter isto.

DEVE ter isto se for possível de todo.

PODE ter isto se não afectar o resto.

NÃO VAI ter isto agora mas SERIA bom ter no futuro.

Prototipagem

Esta técnica refere-se à criação de protótipos do sistema em desenvolvimento numa fase inicial do projecto. Ela possibilita a descoberta antecipada de possíveis problemas no sistema e o teste por parte dos futuros utilizadores do sistema. Deste modo, é conseguido um bom envolvimento do utilizador final com o sistema, sendo este um dos principais factores de sucesso da DSDM.

Workshop

Esta é uma das técnicas de um projecto DSDM que tem como alvo juntar os diferentes *stakeholders* fazendo com que estes discutam as requisições estipuladas e as funcionalidades do produto, levando, no fim, a um entendimento mútuo. Num *workshop*, os *stakeholders* reúnem-se e discutem o projecto.

Testes

Outro aspecto de relevo dos objectivos da DSDM é a criação de um SI com boa qualidade. Para obter esta solução, a DSDM obriga a efectuar testes em todas as iterações. Uma vez que a DSDM é uma metodologia independente de ferramentas e técnicas, a equipa do projecto é livre de escolher o seu próprio método de realização de testes.

3.6 Natureza Iterativa e Incremental

Seguindo a técnica de *Timeboxing* e dando prioridade a todos os requisitos, a DSDM fornece também uma aproximação iterativa ao desenvolvimento de Sistemas de Informação, aproximação que está ilustrada no esquema apresentado anteriormente

(Fig. 3). Os passos de Análise Funcional, Desenho e Implementação estão totalmente descritos de maneira iterativa o que demonstra que cada um dos níveis pode regressar aos sub-níveis, antes de entrar no passo seguinte. Cada iteração aborda um novo conjunto de funcionalidades. Graças à aproximação de desenvolvimento incremental, todas as iterações são construídas a partir de um antecessor que já se encontra a funcionar. Cada um destes incrementos pode então voltar atrás quando/se necessário. A Fig. 3 representa situações em que tal pode acontecer, ou seja, as setas regressam às etapas anteriores. Como exemplo temos uma seta que regressa ao Estudo do Negócio proveniente da Implementação. Se uma grande funcionalidade for descoberta durante o decorrer do projecto, e esta não possa ser implementada, existe a possibilidade de recomeçar tendo em conta as novas exigências no Estudo do Negócio.

Outro exemplo de retrocesso encontra-se ilustrado pela seta que liga a Implementação à Análise Funcional. Isto pode acontecer quando a funcionalidade é omitida no processo prévio de Análise Funcional, essencialmente devido à falta de tempo ou de restrições orçamentais. Apenas quando todas as exigências necessárias à realização do ajuste do projecto e aos objectivos do trabalho estiverem concluídas, o projecto avança para as fases posteriores.

Graças a esta natureza iterativa, é necessário manter um bom grau de exigência da equipa gestora durante todo o desenvolvimento. Isto assegura que o projecto implementa todos os requisitos pedidos essenciais, na forma desejada definida nas fases anteriores do projecto.

3.7 Factores de Sucesso da DSDM

Dentro da DSDM existe um grande número de factores identificados como de extrema importância para assegurar o sucesso do projecto. Em primeira instância é necessária uma boa receptividade do DSDM por parte da equipa sénior de gestão tal como da restante equipa. Isto assegura que todas as partes envolvidas no projecto ganhem motivação desde o início e que permaneçam determinadas durante o decorrer do projecto.

O segundo factor provém directamente do tópico anterior e relaciona-se com o compromisso de gestão que assegura o envolvimento do utilizador. A Prototipagem necessita de um envolvimento forte e dedicado por parte do utilizador a fim de testar e julgar o funcionamento do protótipo.

Temos também a equipa responsável pelo projecto. Esta equipa deve ser composta por uma equipa hábil com o propósito de formar uma união estável. Uma questão relevante prende-se com o facto de ser necessário dotar a equipa de desenvolvimento de algum poder. Isto significa que a equipa deve ter a possibilidade de tomar decisões importantes sem necessitar de enviar propostas formais aos elementos mais influentes ou ao cliente, o que permite uma grande poupança de tempo.

Para que a equipa seja capaz de obter sucesso, é também necessário oferecer todas as condições e tecnologias requeridas para a condução correcta do projecto.

3.8 Comparação com outras Metodologias de Desenvolvimento

Ao longo dos anos, um grande número de métodos de desenvolvimento de sistemas de informação vêm a ser desenvolvidos e aplicados, dividindo-se em **Métodos Estruturados**, **Métodos RAD** e **Métodos Orientados-a-Objectos**. Muitos destes métodos têm grandes semelhanças entre eles, o mesmo acontecendo com a DSDM. Como exemplo poderemos citar a *extreme Programming (XP)*, que se baseia numa aproximação iterativa ao desenvolvimento de sistemas de informação com uma grande influência do utilizador.

O *Rational Unified Process* será talvez o método que mais tem em comum com a DSDM já que se apresenta como uma Metodologia Dinâmica de desenvolvimento de Sistemas de Informação. Mais uma vez é utilizada uma aproximação iterativa neste método de desenvolvimento.

Tal como a XP e a RUP, existem muitos outros métodos de desenvolvimento que partilham características comuns com a DSDM. Apesar disso, esta metodologia distingue-se de todas as outras em diversas vertentes. Em primeiro lugar está o facto de fornecer um *framework* independente de ferramentas e de técnicas, o que possibilita que a equipa de desenvolvimento utilize as ferramentas às quais se encontra mais adaptada. Outra particularidade desta metodologia prende-se com o facto do desenvolvimento não girar em torno do tempo/recursos, mas sim das exigências. Isto assegura o principal objectivo da DSDM, nomeadamente manter todo o projecto dentro dos limites orçamentais e temporais.

Por fim, é dada maior importância à comunicação entre todos os elementos da *stakeholder* do sistema. Embora este ponto seja comum a outras metodologias, a DSDM acredita verdadeiramente num compromisso que assegura o sucesso final do projecto.

4. Adequação da DSDM

4.1 Projectos Adequados à Utilização da DSDM

Projectos de Mudança – Estes projectos necessitam constantemente de sugestões dos clientes, conhecer os seus pontos de vista para poder encontrar a melhor maneira de alcançar os objectivos.

Projectos Criativos – Os projectos criativos apelam a sessões de *brainstorming*. Estes projectos necessitam de uma forte participação dos clientes e privilegia a modelação e visualização conforme a motivação dos utilizadores.

Projectos Urgentes – Projectos *time to market* são dos mais conceituados para se poderem adaptar DSDM. O método das *Timeboxes* e prioridades de funções levam à entrega do produto à hora com as funcionalidades essenciais.

Projectos Ergonómicos – Este tipo de projectos exigem uma interface bastante evoluída e adaptada ao homem tornando o papel dos utilizadores e de peritos em factores humanos, estéticos e ergonómicos cruciais.

Projectos Multidisciplinares – Os princípios de responsabilidade e cooperação da DSDM reflectem o seu poder neste tipo de projectos. É necessário efectuar constantemente testes durante todo o projecto com de uma forma colectiva.

Projectos de Novas Tecnologias – Estes projectos necessitam de um espaço para a experimentação e privilegiam o retorno das experiências antes de tomar decisões definitivas. É necessário por vezes alterar a estrutura do projecto e com os princípios da reutilização e reconfiguração que DSDM oferece, isto torna-se possível.

Projectos de Desenvolvimento de Produtos – A DSDM é adaptada para projectos de desenvolvimento de produtos. Estes projectos têm frequentemente *timings* apertados. DSDM insiste sobre validações frequentes adaptadas às necessidades do consumidor.

4.2 Projectos Não Adequados à Utilização da DSDM

- Clientes não disponíveis para efectuar testes.
- Culturas muito hierárquicas.
- Algoritmos complexos de validar e visualizar.
- Sistemas críticos à segurança humana.
- Arquitectura pouco modular.
- Tecnologias que não permitem a Prototipagem
- Reversibilidades e gestão de configuração difíceis de praticar.
- Cargo e responsabilidade dos clientes muito falsa.
- Falta de confiança dos clientes.

5. Conclusões

Apesar da utilização de uma metodologia ágil não ser aconselhada para todos os tipos de projecto, a DSDM é uma ótima e segura base para equipas de desenvolvimento que têm em mãos projectos com necessidade de execução rápida e com requisitos flexíveis. A partir desta metodologia, é possível construir um trabalho que agrade ao cliente - graças ao cumprimento dos prazos e orçamentos - à equipa de desenvolvimento - que terá em mãos um projecto organizado e com obrigação de cumprir apenas os requisitos necessários à execução do sistema - e aos utilizadores finais que terão a oportunidade de interagir com todo o desenvolvimento do sistema de informação, através das frequentes fases de testes.

6. Bibliografia

[DSDMC, 1997] DSDM Consortium. *Delivering Agile Business Solution on Time*. [Online]. Disponível em <http://www.dsdm.org/> [Consultado em 30/11/05]

[Fowler, 2003] Fowler, Martin. *The New Methodology*. [Online]. Disponível em <http://www.martinfowler.com/articles/newMethodology.html> [Consultado em 30/11/05]

[HI5, 2005] HI5, *Who's in?* [Online]. Disponível em <http://www.hi5.com/default.html> [Consultado (pelo Tiago) em 30/11/05]

[Clifton et al., 2003] Clifton, Marc. Dunlap, J., *What is DSDM?* [Online]. Disponível em <http://www.codeproject.com/gen/design/dsdm.asp> [Consultado em 30/11/05]

[Sollicito, 2003] Sollicito, *To DSDM or not to DSDM?* [Online]. Disponível em <http://www.informit.com/articles/article.asp?p=20696&redir=1&r1=1> [Consultado em 30/11/05]

[Wikipedia, 2003] Wikipedia.org, *Dynamic Systems Development Methodology* [Online]. Disponível em <http://en.wikipedia.org/wiki/DSDM> [Consultado em 30/11/05]

[Morais, 1998] Morais, A., *Dicionário Inglês-Portugues 3ª Edição*

[Dynamic Ventures, 2004] Dynamyc Ventures Inc., *Costum Software Development?* [Online]. Disponível em www.dyve.com/dv/methodology.htm [Consultado em 30/11/05]