

IFSP Instituto Federal de Educação Ciência e Tecnologia
Análise e Desenvolvimento de Sistemas

Igor Ferreira de Oliveira BP3033121
Luca Vianna Martins Silveira BP3032469
Mauro Sérgio Jircik A. M. R. L. Pereira BP3032116
Pedro Henrique da Silva Munarão BP3033325

“DYNAMIC SYSTEM DEVELOPMENT METHOD”

**IGOR FERREIRA DE OLIVEIRA
LUCA VIANNA MARTINS SILVEIRA
MAURO SÉRGIO JIRCIK ARRUDA MENDES RIBEIRO LEITE PEREIRA
PEDRO HENRIQUE DA SILVA MUNARÃO**

DYNAMIC SYSTEM DEVELOPMENT METHOD

Trabalho apresentado ao Curso de Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia de São Paulo – Campus Bragança Paulista, como requisito parcial da disciplina Engenharia de Software sob a orientação do Prof. Raphael Naves.

**BRAGANÇA PAULISTA
2023**

RESUMO

O presente trabalho tem como objetivo abordar o método para projetos denominado Dynamic System Development Method (Método de Desenvolvimento de Sistemas Dinâmicos) ou DSDM. Trata-se de um método para gerenciamento e entrega de projetos, ajudando a fornecer resultados de forma rápida e eficaz. O trabalho inicia com a contextualização do método e sua origem histórica. Desde seu lançamento em 1994, o DSDM tem estado presente nos projetos ágeis e nas entregas de soluções. Na sequência, os oito princípios ajudam a direcionar e moldar as atitudes no DSDM. Após, a compreensão e a avaliação dos fatores de sucesso nas fases iniciais são fundamentais para ajudar na abordagem. Aborda-se os processos do método que fornecem um desenvolvimento iterativo e incremental, com fases do ciclo de vida. Cada fase com uma finalidade específica e produtos bem definidos destinados a ajudar na evolução da solução e no bom andamento do projeto. São explanadas 7 técnicas para serem utilizadas pelas equipes para o desenvolvimento da solução. E, ao final, são apresentadas as vantagens e desvantagens do método, bem como sua comparação com outros métodos ágeis.

Palavras-chave: Agile, Agile Alliance, Filosofia, Princípios, Produtos, Processo Geral, Papéis e Responsabilidades, Técnicas, Fatores de Sucesso, Vantagens e Desvantagens, Comparação com outros métodos Ágeis

ABSTRACT

The present work aims to address the method for projects called Dynamic System Development Method (Dynamic Systems Development Method) or DSDM. It is a method for managing and delivering projects, helping to deliver results quickly and effectively. The work begins with the contextualization of the method and its historical origin. Since its launch in 1994, DSDM has been at the forefront of agile design and solution delivery. Next, the eight principles help guide and shape attitudes in DSDM. Afterwards, the understanding and evaluation of the success factors in the initial phases are fundamental to help in the approach. It addresses the method processes that provide an iterative and incremental development, with life cycle phases. Each phase has a specific purpose and well-defined products designed to help the evolution of the solution and the smooth running of the project. 7 techniques are explained to be used by teams to develop the solution. And, at the end, the advantages and disadvantages of the method are presented, as well as its comparison with other agile methods.

Key-words: Agile, Agile Alliance, Philosophy, Principles, Products, General Process, Roles and Responsibilities, Techniques, Success Factors, Advantages and Disadvantages, Comparison with other Agile methods

LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas e Técnicas
DSDM	Dynamic System Development Method
RAD	Rapid Application Development
TI	Tecnologia da Informação
MoSCoW	Must, Should, Could and Won't have
MVP	Produto Mínimo Viável
XP	Extreme Programming
FDD	Feataure driven development
PCB	Printed Circuit Board (Placa de Circuito Impresso)

SUMÁRIO

RESUMO	2
ABSTRACT	3
LISTA DE ABREVIATURAS E SIGLAS	4
1 INTRODUÇÃO	6
1.1 Contextualização	6
1.1.1 Histórico	6
1.1.2 DSDM, Agile e Agile Alliance	7
2 DESENVOLVIMENTO	8
2.1 Filosofia e princípios	8
2.1.1 Filosofia	8
2.1.2 Princípios	8
2.2 Fatores de sucesso	9
2.3 Produtos	10
2.3.1 Introdução	10
2.3.2 Produtos evolutivos	10
2.3.3 Produtos milestones	11
2.4 Processos	11
2.4.1 Introdução	11
2.4.2 Configurando o DSDM para escalabilidade e formalidade	12
2.4.3 Papéis e Responsabilidades	12
2.5 Técnicas	13
2.5.1 Timeboxing	13
2.5.2 MoSCoW - must, should, could and won't have	14
2.5.3 Prototipagem	16
2.5.4 Teste	16
2.5.5 Workshops	16
2.5.6 Modelagem	16
2.5.7 Planejamento e controle	17
2.6 Vantagens e desvantagens	18
2.6.1 Vantagens	18
2.6.2 Desvantagens	18
2.7 Comparação com outros métodos ágeis	19
2.7.1 Utilizar DSDM junto com outras metodologias	19
3 CONCLUSÃO	20
REFERÊNCIAS	21

1 INTRODUÇÃO

1.1 Contextualização

Segundo AGILE BUSINESS CONSORTIUM (2023), o Método de Desenvolvimento de Sistemas Dinâmicos (Dynamic System Development Method ou DSDM) é uma estrutura focada no desenvolvimento de projetos, com o estabelecimento de atribuições de responsabilidades e funções para as equipes.

Quando necessário, o DSDM pode ser adaptado para complementar outros métodos de gerenciamento de projetos, com o objetivo de evitar conflitos ou duplicação de esforços.

Teve sua origem em 1994 através da colaboração de um grande número de profissionais da área de projetos de várias empresas que procuravam buscar a qualidade nos processos RAD (Rapid Application Development) durante o desenvolvimento do projeto.

O DSDM é um método que está sendo reconhecido como o “pai” dos métodos ágeis e que foi pensado para o desenvolvimento de projetos com grande qualidade, mesmo possuindo suas limitações de prazos curtos e orçamentos fixos.

1.1.1 Histórico

Quando o DSDM foi criado em 1994, para AGILE BUSINESS CONSORTIUM (2023), a forma de entrega de soluções através de projetos era muito diferente, as empresas usavam principalmente a abordagem tradicional de Cascata. Muitos desses projetos falhavam, por várias razões, mas principalmente em função do tamanho pois eram muito grandes ou muito longos, com comunicação ruim entre as equipes e com progresso medido em porcentagens, em vez de entregas. As entregas das soluções eram geralmente feitas com atraso e de forma errada.

Em contrapartida ao modelo tradicional de Cascata que foca em um processo de desenvolvimento sequencial e pouco flexível, alguns projetos procuraram uma abordagem completamente diferente, a Rapid Application Development (Desenvolvimento Rápido de Aplicação) ou RAD que consiste em um método para desenvolver softwares que permite uma prototipagem mais rápida e uma entrega repetitiva do projeto, com os usuários da solução trabalhando em conjunto com os desenvolvedores, não com base em especificações iniciais formalizadas, mas em discussões, demonstrações e ciclos de feedback curtos. Entretanto, novos problemas surgiram, pois o RAD fornecia as soluções de forma rápida, mas na maioria das vezes afetava a qualidade da solução.

Assim, o DSDM foi criado para resolver os problemas do modelo Cascata que era muito lento, grande, com pouca transparência e envolvimento; bem como, os problemas da RAD que possuía velocidade e rapidez, mas pecava em qualidade. O DSDM reuniu as melhores partes das duas abordagens, reconhecendo que ambas tinham pontos fortes e fracos, tornando-se um método estruturado de gerenciamento e entrega de projetos, fornecendo resultados de forma rápida e eficaz.

Com o sucesso do DSDM, as empresas participantes formaram o DSDM Consortium, uma organização sem fins lucrativos para administrar o compartilhamento, exploração e evolução da propriedade intelectual da DSDM. No começo, as informações eram exclusivamente para os membros do Consórcio; mas, em 2007, o DSDM Consortium tornou o DSDM disponível de forma gratuita para visualização e uso gratuito.

1.1.2 DSDM, Agile e a Agile Alliance

O termo “Agile” foi usado pela primeira vez em 2001, depois que um grupo de pessoas com ideias semelhantes, incluindo Arie Van Bennekum, do DSDM Consortium, concordou em se encontrar por um fim de semana em Snowbird, Utah. Nessa reunião, eles reconheceram que todos compartilhavam valores e formas de trabalho comuns e concordaram com um conjunto formalizado desses valores e 12 princípios de apoio que definiram uma maneira Ágil de trabalhar. Essa nova maneira de trabalhar é fundamentalmente diferente em estilo da abordagem Cascata tradicional que dominava o mundo dos projetos de TI na época. Dessa reunião surgiu o Manifesto para Desenvolvimento Ágil de Software e um grupo chamado Agile Alliance, que existe até hoje. Nos últimos anos, o uso do Agile cresceu significativamente.

Conforme (CUNNINGHAM, 2001), o Manifesto para o Desenvolvimento Ágil de Software, apresenta os seguintes valores estabelecidos:

- Indivíduos e interações sobre processos e ferramentas: fornece orientação leve e apropriada para as funções e as responsabilidades dos envolvidos na busca do que precisa ser feito e da descoberta da melhor maneira de fazê-lo.
- Software que trabalha sobre uma documentação completa: utiliza produtos com envolvimento das partes para garantir que a solução certa seja entregue; bem como, uma documentação leve e de apoio
- Colaboração do cliente em vez de negociação de contratos: incentiva o trabalho de forma colaborativa o tempo todo entre as equipes de projeto e a empresa/usuário.
- Responder à mudança ao invés de seguir um plano: procura analisar e avaliar as mudanças durante o desenvolvimento do projeto, considerando a mudança como algo normal e não uma exceção.
- E mesmo havendo valor nos itens à direita, valorizamos mais os itens à esquerda: faz com que contratos, planos, processos, ferramentas e documentação sejam feitos na profundidade necessária para agregar valor.

2 DESENVOLVIMENTO

Para AGILE BUSINESS CONSORTIUM (2023), o método adota a filosofia de que qualquer projeto deve estar alinhado a objetivos estratégicos bem definidos e focar na entrega antecipada de benefícios reais para a solução. Apoiar essa filosofia permite que as equipes mantenham o foco e alcancem as metas do projeto.

2.1 Filosofia e Princípios

2.1.1 Filosofia

A filosofia DSDM é que o melhor valor comercial aparece quando os projetos estão focados a objetivos comerciais claros, são entregues com frequência e envolvem a colaboração de pessoas capacitadas e motivadas.

Para atingir este pensamento, as partes envolvidas devem:

- entender e internalizar a visão e os objetivos do negócio;
- possuir autonomia na tomada de decisões dentro de sua área de especialização;
- colaborar no fornecimento de uma solução de negócios adequada à finalidade;
- colaborar na entrega em prazos acordados de acordo com as prioridades de negócios;
- aceitar que a mudança é inevitável e que a compreensão cresce com o tempo;
- abranger todos de dentro ou de fora do projeto, envolvidos ou afetados por ele.

Variáveis e resultados esperados do projeto usando DSDM:

- Os projetos precisam equilibrar as quatro demandas mais comuns que são: tempo, custo, recursos e qualidade. Tentar considerar todas as quatro no início do projeto é impossível.
- Os Projetos com a filosofia DSDM que adotam os princípios e seguem as orientações relacionadas aos produtos, processos, práticas e pessoas, entregarão a solução certa, no momento certo, pelo custo certo e permite alcançar os benefícios que justificam o projeto.

Para aplicar a filosofia de gerar o melhor valor de solução através de projetos alinhados a objetivos claros, entrega frequente e colaboração de pessoas motivadas e capacitadas, o DSDM possui oito princípios, apoiados na definição e orientação de produtos, processos, pessoas e práticas.

2.1.2 Princípios

Segundo a Agile Business Consortium, existem 8 princípios da DSDM:

Os Oito Princípios do método DSDM são:

1. Foco na necessidade do negócio
2. Comunique-se de forma contínua e clara
3. Colaborar
4. Desenvolva de forma iterativa
5. Construa gradualmente a partir de fundações firmes
6. Entrega no prazo
7. Nunca comprometa a qualidade
8. Demonstrar controle

- Princípio 1 - Foco na Necessidade do Negócio: cada decisão tomada durante um projeto deve ser vista à luz do objetivo do projeto - entregar o que precisa, quando precisa ser entregue.
- Princípio 2 - Entrega no prazo: entrega pontual de produtos intermediários é a melhor maneira de demonstrar controle sobre a evolução da solução.
- Princípio 3 - Colaborar: trabalho em equipe com espírito de cooperação e comprometimento para superação dos desafios no desenvolvimento do projeto.
- Princípio 4 - Nunca comprometa a qualidade: o grau de qualidade a ser entregue é acordado no início. Todo o trabalho deve ter como objetivo atingir esse grau de qualidade.
- Princípio 5 - Construir gradualmente a partir de fundações firmes: estabelecimento de bases sólidas para a entrega incremental da solução para apresentar benefícios comerciais reais no início, incentivando a confiança das pessoas envolvidas.
- Princípio 6 - Desenvolver Iterativamente: repetição e revisão constante para garantir que o que está sendo desenvolvido seja a solução que realmente precisa.
- Princípio 7 - Comunique-se de forma contínua e clara: melhoraria na eficácia da comunicação das pessoas envolvidas, com ênfase na interação humana, papéis bem definidos e envolvimento comercial.
- Princípio 8 - Demonstrar Controle: estar no controle e demonstrar a transparência durante toda fase de desenvolvimento do projeto.

2.2 Fatores de sucesso

De acordo (REPM, 2019), os fatores de sucesso são vistos como instrumentos dos projetos na busca de um resultado bem-sucedido.

1. Comprometimento com a Abordagem: direcionamento aos objetivos do negócio, entrega com frequência e colaboração da equipe;
2. Equipe Eficaz de Desenvolvimento de Soluções: uma equipe eficaz se faz através de quatro elementos:
 - Fortalecimento: desenvolvimento em cima de decisões seguras e estruturadas;
 - Estabilidade: análise do impacto frente mudanças;
 - Habilidades: diversidade de conhecimento e experiências;
 - Tamanho: pequenos tamanhos facilitam a comunicação entre os envolvidos.
3. Engajamento Comercial - Ativo e Contínuo: garantia de envolvimento comercial ativo e contínuo através do compromisso e da colaboração diária envolvendo uma relação comercial de suporte na busca da solução;

4. Desenvolvimento Iterativo, Teste Integrado e Entrega Incremental: garantia que os elementos da solução sejam parte da abordagem de desenvolvimento iterativo com testes integrados e de entrega incremental, proporcionando redução do risco e maior grau de confiança ao projeto.;
5. Transparência: construção da confiança reduz os riscos. Apresentação do produto Solução Evolutiva fornece prova física e objetiva do progresso do desenvolvimento do projeto;
6. Questionário de abordagem de projeto - avaliação de opções e riscos: definir de projetos com as melhores chances de sucesso, é importante para se ter uma visão real do ambiente de trabalho, das pessoas e dos relacionamentos e avaliar os riscos.

2.3 Produtos

2.3.1 Introdução

Os produtos são orientações para levantar as informações necessárias durante o desenvolvimento do projeto da solução. Eles não são obrigatórios e nem sempre podem ser apresentados como documentos, mas auxiliam no gerenciamento.

O DSDM descreve um conjunto de produtos a serem considerados durante o desenvolvimento do projeto. Esses produtos auxiliam na elaboração da própria solução (a principal entrega do projeto).

A utilização dos produtos e a formalidade envolvida depende de cada desenvolvimento de projeto, influenciado por fatores como relações contratuais, padrões corporativos.

O DSDM identifica dois tipos de produtos:

- produtos evolutivos; e,
- produtos milestone.

2.3.2 Produtos evolutivos

São produtos que evoluem com o decorrer do desenvolvimento do projeto. Eles abrangem várias fases do projeto. Podem ser:

- Caso de Negócio: é a visão e a justificativa para o projeto;
- Lista de Requisitos Priorizados (PRL): descreve os requisitos a serem atendidos e suas prioridades;
- Definição da Arquitetura da Solução: fornece a estrutura de design da solução, para atender os aspectos comerciais e técnicos da solução;
- Definição da Abordagem de Desenvolvimento: define as ferramentas, as técnicas, as práticas e os padrões a serem aplicados;
- Plano de Entrega: fornece o cronograma dos incrementos do projeto;
- Definição da Abordagem de Gestão: é a organização do projeto, o engajamento das partes envolvidas e a forma de apresentação do progresso; e,
- Solução Evolutiva: é a reunião dos modelos, dos materiais de apoio, de teste e de revisão.

2.3.3 Produtos milestone

São os produtos criados para evidenciar uma fase e atendem a um propósito específico como um ponto de verificação ou para facilitar o gerenciamento do desenvolvimento do projeto.

Podem ser:

- Relatório de Revisão do Projeto: é documento final de cada Incremento do Projeto, dos pontos de aprendizado, das práticas, das funções e responsabilidades dos envolvidos.
- Avaliação de Benefícios: é feita periodicamente no cronograma para justificar o investimento.

2.4 PROCESSOS

2.4.1 Introdução

Segundo (BECKER, 2020), os projetos desenvolvidos em DSDM apresentam três etapas diferentes: pré-projeto, ciclo de vida e pós-projeto.

2.4.1.1 Pré-Projeto

Nesta fase, são identificados os projetos candidatos. Além disso, realizam-se orçamentos e contratos. É a etapa anterior ao desenvolvimento do projeto. Os principais objetivos são:

- Definir a visão e os objetivos do projeto;
- Analisar as necessidades do negócio e definir o escopo do projeto;
- Avaliar a viabilidade do projeto em termos de recursos, prazos e orçamentos.

2.4.1.2 Ciclo de vida

Essa etapa está subdividida em 5 partes que representam o decorrer do desenvolvimento do projeto. As partes são: estudo de viabilidade, estudo do negócio, iteração de modelos funcionais, iteração de projeto e desenvolvimento, implementação. Os principais objetivos dessa fase são:

- Entregar um produto de software funcional e de alta qualidade;
- Entregar o produto dentro do prazo e do orçamento definidos;
- Garantir que o produto atenda aos requisitos e necessidades dos usuários.

2.4.1.3 Pós-Projeto

Esta fase é dedicada à implantação do produto e à manutenção contínua. Nesta fase, a equipe do projeto trabalha em estreita colaboração com os usuários para garantir que o produto funcione de forma eficaz e eficiente. Os principais objetivos desta fase são:

- Garantir que o produto atenda às necessidades dos usuários;
- Manter e atualizar o produto para garantir sua funcionalidade contínua;
- Identificar oportunidades de melhoria e evolução do produto.

Em resumo, o DSDM segue um ciclo de vida iterativo e incremental, que permite uma entrega ágil de software de alta qualidade. O ciclo de vida é dividido em três fases principais (Pré-Projeto,

Projeto e Pós-Projeto), cada uma com seus próprios objetivos, atividades e entregas. A colaboração e a comunicação constantes entre os membros da equipe são fundamentais para o sucesso do projeto.

2.4.2 Configurando o DSDM para escalabilidade e formalidade

Para configurar o DSDM para escalabilidade e formalidade, é necessário adaptar as práticas do DSDM para atender às necessidades específicas do projeto. Isso pode incluir adotar uma abordagem em cascata, definir papéis e responsabilidades claramente, estabelecer um processo formal de revisão de design e código, definir um plano de gerenciamento de riscos e adotar ferramentas e técnicas de gestão de projetos mais avançadas.

1. Adotar uma abordagem em cascata: O DSDM é um modelo iterativo e incremental, mas para projetos maiores e mais complexos, pode ser necessário adotar uma abordagem mais sequencial e em cascata. Nesse caso, é importante ter fases distintas e bem definidas, com entregas formais e aprovações do cliente.
2. Definir papéis e responsabilidades claramente: Para projetos maiores e mais complexos, é fundamental ter uma estrutura organizacional clara, com papéis e responsabilidades bem definidos para cada membro da equipe. Além disso, é importante ter um processo formal de gerenciamento de mudanças e controle de versão.
3. Estabelecer um processo de revisão de design e código: Para projetos maiores e mais complexos, pode ser necessário estabelecer um processo formal de revisão de design e código, para garantir a qualidade e a conformidade com os padrões estabelecidos.
4. Definir um plano de gerenciamento de riscos: Projetos maiores e mais complexos também apresentam mais riscos e incertezas. Por isso, é importante ter um plano de gerenciamento de riscos formal, que identifique, avalie e priorize os riscos, e que tenha ações definidas para mitigar ou responder a esses riscos.
5. Adotar ferramentas e técnicas de gestão de projetos: Para projetos maiores e mais complexos, pode ser necessário adotar ferramentas e técnicas mais avançadas de gestão de projetos, como diagramas de Gantt, planilhas de acompanhamento de projetos e ferramentas de gestão de tempo e recursos.

2.4.3 Papéis e Responsabilidades

O DSDM (Dynamic Systems Development Method) é um framework ágil que define um conjunto de papéis e responsabilidades para os membros da equipe envolvida no projeto. Aqui estão os principais papéis e responsabilidades no DSDM:

- Patrocinador: O patrocinador é a pessoa que tem a autoridade e o poder de tomar decisões importantes relacionadas ao projeto. Ele é responsável por fornecer recursos, orientações e apoio para a equipe do projeto e garantir que o projeto esteja alinhado com os objetivos estratégicos da organização.
- Visionário: O visionário é a pessoa que tem uma visão clara do que o projeto deve alcançar. Ele é responsável por definir e comunicar a visão do projeto, estabelecer os objetivos do projeto e definir os critérios de sucesso.

- **Gerente de Projeto:** O gerente de projeto é responsável por gerenciar o projeto em todos os aspectos, incluindo planejamento, orçamento, controle de qualidade, riscos, recursos e comunicação. Ele trabalha em estreita colaboração com o patrocinador e o visionário para garantir que o projeto esteja sendo entregue dentro do prazo, do orçamento e dos padrões de qualidade.
- **Facilitador:** O facilitador é responsável por liderar as reuniões e sessões de trabalho com a equipe do projeto e com os stakeholders externos. Ele garante que a equipe esteja alinhada e trabalhando em conjunto para alcançar os objetivos do projeto.
- **Desenvolvedor:** O desenvolvedor é responsável por desenvolver o produto ou solução do projeto. Ele trabalha em estreita colaboração com o visionário e o gerente de projeto para garantir que o produto esteja sendo desenvolvido dentro das especificações e requisitos definidos.
- **Testador:** O testador é responsável por testar o produto ou solução desenvolvido pelo desenvolvedor para garantir que ele esteja funcionando conforme o esperado e atenda aos requisitos definidos pelo visionário.
- **Embaixador do Usuário:** O embaixador do usuário é responsável por representar as necessidades e interesses dos usuários finais do produto ou solução. Ele trabalha em estreita colaboração com o visionário e o gerente de projeto para garantir que o produto esteja sendo desenvolvido de acordo com as necessidades e expectativas dos usuários finais.
- **Consultor:** O consultor é um especialista externo que fornece conselhos e orientações para a equipe do projeto. Ele pode ter uma ampla gama de habilidades e experiências que ajudam a equipe do projeto a superar desafios específicos ou atingir objetivos específicos.

Esses são os principais papéis e responsabilidades no DSDM. É importante lembrar que, como o DSDM é um framework flexível, a equipe do projeto pode adaptar esses papéis e responsabilidades para atender às necessidades específicas do projeto.

2.5 TÉCNICAS

Segundo a AGILE BUSINESS CONSORTIUM (2023), as principais técnicas do Método de Desenvolvimento de Sistemas Dinâmicos (Dynamic System Development Method ou DSDM) são:

- Timeboxing
- MoSCoW - must, should, could and won't have
- Prototipagem
- Teste
- Workshops
- Modelagem

2.5.1 Timeboxing

Definição: Uma técnica para estimar o tempo que determinada tarefa levará, comprometendo-se a concluí-la neste prazo.

Introdução: Uma das principais práticas do DSDM, o Timeboxing trabalha em conjunto com a técnica MoSCoW. Em sua duração, geralmente são discutidos Must e Should Haves, ao longo de um período que pode variar entre dias e semanas. Ao concluir um timebox, caso os requisitos estejam funcionando bem, de forma eficaz, validados e com riscos gerenciados, o nível de confiança do projeto aumenta.

- Período fixo de tempo.
- Uma ou mais entregas.
- Eficiência medida pela conclusão dos produtos.
- Timeboxes curtos para melhor foco / Timeboxes extensos para alcançar algo útil.

No caso de timeboxes prolongados, dar preferência para série de timeboxes curtos! Para um processo mais iterativo e controlado, revisão frequente e prazo atendido!

Estilos: Existem dois estilos de timeboxing (kick-off e encerramento):

- Timebox estruturado DSDM.
 1. Padrão e repetível.
 2. Planejar antecipadamente os horários do embaixador de negócios.
 3. Sessões de planejamento, feedback e revisões.
- Timebox de formato livre.

Etapas: São 3 etapas que compõem a estrutura dos timeboxes:

- Investigação (10%-20%) - Fornece base sólida para o trabalho a ser realizado no refinamento.
- Refinamento (60%-80%) - Concluir o máximo possível do trabalho de desenvolvimento (teste de produto - atender os critérios de aceitação).
- Consolidação (10%-20%) - Revisão do refinamento, teste final (atender os padrões com qualidade - revisar a eficácia do timebox).

Atividade de desenvolvimento iterativo para convergir em uma solução de negócios precisa.

2.5.2 MoSCoW - must, should, could and won't have

Introdução/Definição: para (PRODUCTPLAN, 2023) foi criado por Dai Clegg em 1994, a prática MoSCoW parte do princípio de que o tempo é fixado, então a priorização para entrega no prazo é essencial. MoSCoW é uma técnica de priorização de tarefas em 4 partes:

- M - Must - essencial, deve ter.
- S - Should - deveria ter.
- C - Could - poderia ter.
- W - Won't - não terá desta vez.

Na construção do MoSCoW, 60% é composto por Must (s) e Should (s) e 20% por Could (s). Além disso, User Stories (histórias de usuário) são uma maneira muito eficaz de definir requisitos em um estilo ágil.

Outros Modelos de Classificação de Requisitos:

- Uso de alta, média e baixa (errado) - definições das prioridades não definidas, única opção média, o que esperar do projeto?
- Uso de 1, 2, 3, 4 (errado) - Menos eficiência e importâncias semelhantes, discussões prolongadas sobre as posições dos itens.
- MoSCoW (correta) - Fornece indicação clara e expectativa para sua conclusão.

Partes do MoSCoW:

- MUST (DEVE) - Subconjunto mínimo usável:
 1. Não adianta entregar sem isso.
 2. Não é legal sem isso.
 3. Inseguro sem isso.
 4. Não é possível uma solução sem isso.

Must have (s) não é negociável e deve garantir cerca de 60% do esforço da equipe/time. Porém, mais de 60% do esforço dedicado a Must have (s) podem acarretar em riscos de falha. Além disso, um Must have pode ser dividido em requisitos menores (decomposição).

O que acontece se esse requisito não for atendido? - Caso a resposta seja "não faz sentido sem este requisito", o requisito é obrigatório. Porém, caso a resposta seja diferente disso, o requisito provavelmente faz parte do grupo Should ou Could.

- SHOULD (DEVERIA):
 1. Importante, mas não vital.
 2. Doloroso, mas a solução ainda é viável.
 3. Solução alternativa.

Dor causada ou pessoas afetadas são medidas classificatórias de requisitos entre DEVERIA E PODERIA.

- COULD (PODERIA):
 1. Desejado, mas menos importante.
 2. Menos impacto se deixado de fora.

Só poderiam ser entregues no melhor cenário possível. Em caso de risco na entrega, COULD HAVE (s) são o primeiro descarte.

- WON'T (NÃO TERÁ):
 1. Não serão entregues.
 2. Esclarece o escopo.

Desta vez, não serão atendidos estes requisitos (Won't have this time).

Modelo Tradicional – Todos os requisitos são tratados como Must have (s), tempo adiado caso encontrem-se problemas.

DSDM – Fixação de tempo, custo, recursos e qualidade. Cria-se requisitos priorizados mediante a técnica MoSCoW.

- MoSCoW para projetos.
- MoSCoW para incremento do projeto.
- MoSCoW para este timebox.

Princípios:

- Equilibrar risco e previsibilidade para cada projeto.
- Prioridades estabelecidas antes do início, mas revisadas ao final dos timebox.
- Discutir os requisitos obrigatórios.
- Mais do que apenas o mínimo
- Aplicar pressão de entrega (de Could have (s) por exemplo) a equipe é contraproducente.
- Importante > interessante.

2.5.3 Prototipagem

Definição: Construção de um modelo, essencial em desenvolvimento ágil baseado em recursos (FDD).

- Experimentar ideias e obter feedback.
- MVP – Produto Mínimo Viável.

2.5.4 Teste

Definição: Verificação de que o sistema está funcionando como o esperado.

Tipos de Teste:

- Teste antes do desenvolvimento.
- Teste rápido à medida que se desenvolve.
- Teste de integração.
- Teste automatizado.

2.5.5 Workshops

Definição: Sessões de trabalho interativas, entre usuários e desenvolvedores para trabalhar conceitos em tempo real.

- Feedback contínuo e colaboração.
- Face to face.
- Abraça vários pontos de vista.
- Tomada de decisão rápida e de alta qualidade.
- Maior aceitação de todas as partes interessadas.
- Sinergia (esforço simultâneo).
- Esclarecimento de questões (minimizar ambiguidades).

Introdução: Os workshops são uma das principais técnicas do DSDM, para garantir a entrega de objetivos claros. Normalmente compostos por um facilitador de workshop (independente) que busca a interação entre o pessoal envolvido, sendo essa o medidor de sucesso da prática.

Esta técnica permite maior adesão às decisões tomadas.

Facilitador:

- Neutro, não focado no resultado/ resolução do problema.
- Incita a comunicação e trabalho em equipe, em curto período de tempo (visual e verbal).

- Orienta o grupo.
- Faz com que os indivíduos assumam responsabilidades e honrem com seus compromissos.
- Resolvem desentendimentos.
- Faz com que os indivíduos se sintam envolvidos e comprometidos com o resultado final.
- Constrói um plano reunindo requisitos.

2.5.6 Modelagem

Definição: Semelhante à prototipagem, fornece um conceito HOLÍSTICO do que deve ser produzido. Deve ser utilizado como ajuda e não ser considerado apenas uma sobrecarga burocrática.

A modelagem aumenta a eficácia da comunicação para todos os membros e níveis do processo e o público-alvo é quem dita o estilo do modelo. Sua formalidade depende da relevância do projeto.

- Abordagem iterativa, incremental e colaborativa.
- Divisão do trabalho.
- Backlog de tarefas ou histórias de usuários.
- Comunicação clara e continua.

Princípios:

- Faça o que funciona para o projeto e para a organização.
- Use diagramas para estabelecer uma linguagem comum entre as equipes.
- Faça modelagem adequada o suficiente, nada mais.
- A modelagem ajuda a visualizar elementos da solução o mais rápido possível.
- Modelos ajudam na divisão do projeto de forma compreensível.
- Modelos ajudam no planejamento de timeboxes.
- Solução funcional ou parcial o mais rápido possível.
- EDUF – enough design up front.

Exemplos: A modelagem melhora a compreensão através de representações visuais, transparência e abstração de informações relevantes.

- StoryBoard – anúncio de televisão.
- Planta Arquitetura – conjunto habitacional.
- Modelo em escala – carro.
- Diagramas de processo – funcionalidade de software.

Modelo: Uma descrição usada para ajudar a visualizar algo que não pode ser observado diretamente.

- Cópia pequena, mas exata de algo
- Padrão a ser seguido.

MODELAGEM não é necessariamente um protótipo, mas PROTOTIPAGEM sempre é uma espécie de modelo.

2.5.7 Planejamento e controle

Introdução: Método disciplinado para rastrear alterações e versões do software.

- Agile and Control + Agile and Planning
- Resposta à mudança ao invés de seguir um plano.

- Entrega incremental da solução (timeboxes).
- Garantia de qualidade na solução (revisão e testes integrados ao desenvolvimento).
- Fornecer o que o negócio precisa, quando precisa.
- Planejamento baseado em resultados.
- Planejamento para horizontes sensíveis no nível certo de detalhe.

Tipos de Planejamento:

- Deliveryplan – duração maior (entrega) / finalização do projeto.
- Timeboxplan – duração menor (fim de um timebox)/ mais detalhado.

Quanto maior o prazo, mais o resultado será incerto.

O melhor valor comercial surge quando os projetos estão alinhados e objetivos comerciais claros, são entregues com frequência e envolvem a colaboração de pessoas motivadas e capacitadas.

2.6 Vantagens e desvantagens

O foco desse subtópico é fornecer informações sobre as principais vantagens e desvantagens que podem influenciar a escolha do DSDM como metodologia ágil principal.

2.6.1 Vantagens

- Entrega rápida: o DSDM consegue fazer uma entrega rápida, porque, ele consegue obter feedbacks frequentes pelo seu modelo de processo iterativo e incremental. Com isso acaba reduzindo o número de atrasos nos projetos, feedbacks negativos ao final do projeto e assim, conseguindo entregar mais valor para o cliente e usuário final.
- Feedback frequente: o DSDM utiliza de técnicas como workshops frequentes, para o recebimento de feedback dos projetos, para que assim, possa realizar ajustes, solicitar melhorias e corrigir erros, para que no final, os desenvolvedores possam entregar projeto de alto valor e que atenda as expectativas solicitadas.
- Flexibilidade: essa vantagem corresponde ao que já foi citado em outras vantagens, mas de forma resumida o DSDM consegue ser um método flexível e ágil pelo modelo de processo iterativo e incremental.
- Controle de qualidade: consegue entregar o projeto atendendo as expectativas do cliente, pois consegue fazer análises frequentes e revisões regulares para realizar a melhoria e ajustes no projeto.

2.6.2 Desvantagens

- Complexidade: é uma das maiores desvantagens do DSDM, porque é necessário um alto nível de entendimento da metodologia para conseguir usufruir todas as vantagens do DSDM.

- Investimento alto em treinamento: devido ao alto nível de complexidade em comparação com outras metodologias, é necessário um investimento alto em treinamento para a familiarização com o método.
- Colaboração frequente com os clientes: um dos grandes pontos levantados no tópico de vantagens foi o “feedback frequente”, mas de fato essa vantagem traz consigo uma desvantagem, pois é necessário a colaboração frequentes dos clientes em workshops e em reuniões regulares para o andamento do projeto.

2.7 Comparações com outra metodologia ágeis

O foco desse subtópico é comparar o DSDM com outras metodologias ágeis e trazer como conhecimento as qualidades, defeitos, abordagens e semelhanças da própria metodologia.

As metodologias citadas vão ser: Scrum, XP (Extreme Programming) e FDD (Feature driven development)

Levantando os principais pontos chaves como:

- O que difere uma metodologia e outra;
- Quais são as semelhanças

2.7.1 Utilizar DSDM junto com outras metodologias

Vale salientar que o DSDM de fato, consegue trabalhar de forma independente, mas também pode trabalhar de forma complementar com outras metodologias, como Scrum, Kanban, Lean, entre outras para alcançar uma melhor entrega do projeto final. A utilização do DSDM com outras metodologias pode ter inúmeras vantagens e desvantagens, por isso vou citar algumas delas

Vantagens:

Gerar qualidade: o complemento de outra metodologia, pode suprir na carência

Desvantagens:

Complexidade: o DSDM por si só já é uma metodologia muito complexa, tem que ter um bom nível de experiência com metodologias para conseguir trabalhar, mas pode ser ainda mais complexo ao utilizar outras metodologias em conjunto.

Interferência: o conjunto de técnicas entre um conjunto de metodologias, pode causar uma confusão de técnicas e isso pode tornar o processo confuso.

3 CONCLUSÃO

O DSDM revolucionou mundialmente a forma de se trabalhar com projetos em engenharia de software. O método surgiu ao tentar trazer uma junção das vantagens dos melhores métodos que se tinha na época, trazendo consigo uma filosofia única alinhada com oito princípios fundamentais para o funcionamento, conjunto de processos estruturado e diversas técnicas, como o MoSCoW, que consegue entregar requisitos nos projetos por questão de prioridades, o Timeboxing para gerenciamento de tempo, Workshops para ajustes e melhorias no projeto e diversas outras técnicas.

Após um longo período de estudo do grupo, a conclusão a que se chegou é que, de fato, o DSDM é uma boa metodologia, viável e bem estruturada e consegue cumprir todos os objetivos para os quais foi criado, entregando como projeto final algo eficiente, de alto valor e num período menor, atendendo a todas as expectativas dos clientes e usuários finais e assim reduzindo o número de feedbacks negativos ao final.

REFERÊNCIAS

- [1] AGILE BUSINESS CONSORTIUM. **What is DSDM?**. Disponível em: <https://www.agilebusiness.org/business-agility/what-is-dsdm.html>. Acesso em: 18 fev. 2023
- [2] CUNNINGHAM, WARD. **Manifesto for Agile Software Development**. Disponível em: <https://agilemanifesto.org/>. Acesso em: 18 fev. 2023
- [3] BECKER, ALICE. **Conheça o DSDM, uma alternativa ágil que proporciona grandes resultados**. Disponível em: <https://www.voitto.com.br/blog/artigo/dsdm>. Acesso em: 19 fev. 2023
- [4] REPM. **Agile Project Management**. Disponível em: <https://repmt.com/learning-paths/agile-project-management/>. Acesso em: 19 fev. 2023