

CÍNTIA CUNHA, LUANA GONÇALVES, MILENA ALVES, RUANNE GODOI

Adaptive Software Development

O que é ASD?

Adaptive Software Development (ASD) ou Desenvolvimento de Software Adaptativo é uma das técnicas de métodos ágeis que visa melhorar o processo de desenvolvimento de softwares complexos.

De acordo com Jim Highsmith e Sam Bayer, criadores da metodologia, o caminho para o sucesso de um software é através da aprendizagem contínua ao longo do projeto, sem se prender a planos muito rígidos. O objetivo da metodologia é que a equipe esteja sempre pronta para ajustar seus planos e estratégias em resposta a novas informações e/ou mudanças no software.



Fundamentos do ASD

1

Especulação: nessa fase, o foco é a equipe fazer um brainstorming sobre o que precisa ser feito, ou seja, é onde realizam o resumo das informações coletadas com os stakeholders e fazem um esboço do protótipo.

2

Colaboração: durante essa fase, a comunicação é contínua entre a equipe e os stakeholders do projeto. É onde iniciam as reuniões diárias, o ciclo de desenvolvimento e o feedback constante do cliente. A equipe deve ser capaz de suprir as necessidades mais urgentes do cliente.

3

Aprendizado: na fase do aprendizado, a equipe já conhece os desejos do cliente e deve refletir sobre todo o trabalho realizado até o momento e identificar quais questões devem ser melhoradas e/ou aprimoradas de acordo com a necessidade dos usuários. Nesse momento são realizadas revisões técnicas, retrospectivas e feedback dos usuários.

Princípios

Entrega incremental

Desenvolvimento dividido em partes funcionais para feedback contínuo e entrega de valor ao cliente ao longo do tempo.

Pensado para o cliente

Foco na satisfação do cliente através de comunicação e colaboração para refinamento contínuo de requisitos e construção de confiança.

Foco na qualidade

Foco contínuo na qualidade com práticas rigorosas desde o início do desenvolvimento, garantindo funcionalidade, usabilidade e segurança do produto entregue.

Adaptabilidade

Adaptabilidade é essencial, ajustes ágeis às mudanças nos requisitos e no ambiente, com entrega incremental e gestão adaptativa de riscos.

Como funciona?

01

Abordagem baseada em componentes

O Adaptive Software Development (ASD) adota uma abordagem baseada em componentes para o desenvolvimento de software. Nesta abordagem, os projetos são divididos em componentes, que são grupos de funcionalidades planejadas, implementadas e entregues juntas. Componentes esses que incluem todos os aspectos relacionados.

Existem três tipos principais de componentes no ASD:

- **Componentes principais:** Refere-se às funcionalidades de negócios essenciais do software.
- **Componentes tecnológicos:** Incluem infraestrutura e tecnologias necessárias para a implementação dos componentes principais.
- **Componentes de suporte:** Englobam outros aspectos do projeto, como documentação externa e materiais de treinamento.

Como funciona?

02

Ciclos adaptativos e sua estrutura

- Limitados no tempo: Cada ciclo tem uma duração específica, garantindo que o trabalho seja concluído dentro de prazos predefinidos.
- Orientados por riscos: Os ciclos são impulsionados pelo gerenciamento de riscos, com itens de alto risco sendo priorizados nas iterações iniciais.
- Tolerantes a mudanças: Os ciclos são projetados para lidar com mudanças, permitindo que a equipe se adapte rapidamente a novos requisitos ou circunstâncias.

A estrutura típica de um ciclo adaptativo no ASD inclui:

- Definição do tempo: Estabelecimento do período de tempo para o ciclo, geralmente de acordo com a complexidade e escopo do projeto.
- Definição de objetivos: Identificação dos objetivos específicos a serem alcançados durante o ciclo.
- Atribuição de componentes: Alocação de componentes a serem desenvolvidos ou aprimorados durante o ciclo.
- Implementação e Entrega: Desenvolvimento e entrega dos componentes definidos dentro do período de tempo especificado.

Como funciona?

03

Estratégias de gestão de riscos e tolerância a mudanças

Uma das características distintivas do ASD é sua abordagem proativa para gerenciar riscos e tolerar mudanças. Isso é alcançado através de várias estratégias:

- **Priorização de riscos:** Itens de alto risco são identificados e priorizados para as iterações iniciais, permitindo que a equipe lide com potenciais desafios desde o início.
- **Falha rápida:** Se um determinado caminho ou decisão é identificado como de alto risco, o ASD advoga por uma abordagem de "falhar rápido", acelerando a descoberta de falhas para que possam ser corrigidas prontamente.
- **Adaptação contínua:** O ASD promove uma mentalidade de adaptação contínua, onde a equipe está sempre pronta para ajustar seus planos e estratégias em resposta a novas informações ou mudanças nas circunstâncias.

Papéis e responsabilidades em projetos ASD

Executivo patrocinador

O executivo patrocinador atua como o ponto de contato único com a alta administração da organização. Suas responsabilidades principais incluem definir os objetivos de negócios do projeto e aprovar os recursos necessários para sua execução.

Equipe principal

Há uma equipe principal para cada conjunto principal de funcionalidades. Cada equipe deve ter membros dedicados exclusivamente ao projeto. Além disso, membros-chave da organização do cliente também devem participar em tempo integral, quando possível.

Gerente de projeto

A liderança eficaz é essencial para o sucesso do projeto. No ASD, o gerente de projeto é um papel dedicado para cada projeto, necessário devido à natureza intensa do papel.

Critérios	ASD (Adaptive Software Development)	XP (Programação Extrema)	RUP (Rational Unified Process)	RAD (Rapid Application Development)	Família Crystal	Scrum
Abordagem Principal	Iterativa e adaptativa	Ágil e Iterativa	Iterativa e Incremental	Prototipagem rápida e iteração	Flexibilidade e adaptação às necessidades únicas de cada projeto	Desenvolvimento Iterativo e Incremental
Enfoque Temporal	Tempo fixo, escopo variável	Tempo fixo, escopo variável	Tempo e Recursos fixos	Entrega rápida de software	Variável, adaptando-se ao tamanho e complexidade do projeto	Iterações de tempo fixo
Flexibilidade	Alta	Muito flexível	Flexível, mas estruturado	Alta	Alta	Flexível e Adaptação Rápida
Cerimônias/Reuniões	- Especulação, Colaboração, Aprendizado- Revisão Técnica, Retrospectiva	Reuniões Rápidas e Diárias	Reuniões Formais e Informais	Menos estruturadas	- Reuniões de planejamento (Sprint Planning) - Reuniões diárias (Daily Standup) - Revisão da Sprint (Sprint Review) - Retrospectiva da Sprint (Sprint Retrospective)	Scrum Cerimonies
Artefatos Principais	- Declaração de Missão do Projeto, Planilha de Dados do Projeto, Especificação de Produto	Código, Testes, Documentação	Documentos e Modelos	Protótipos, Documentos de Requisitos	- Lista de Tarefas (Backlog) - Incrementos do produto - Relatórios de Progresso	Backlog, Burndown Chart
Metas de Alto Nível	Sim	Alta Qualidade e Entrega Rápida	Satisfação do Cliente	Sim	Sim	Entrega Contínua de Valor
Agilidade e Adaptação	Alta	Muito Ágil e Adaptável	Adaptável a Mudanças	Alta	Alto	Altamente Adaptável
Indústrias Populares	Tecnologia da Informação, Desenvolvimento de software	Software e TI	Software e Desenvolvimento	Tecnologia da Informação, Startups	Software, Tecnologia	Várias Indústrias

Artefatos

1

1. Declaração de missão do projeto

2

2. Planilha de dados do projeto

3

3. Especificação de produto

Vantagens

- Promove o trabalho em equipe buscando engajamento e colaboração.
- Projetado para o desenvolvimento de softwares complexos.
- O modelo de ciclos curtos de ajuda a evitar e aprender com os erros.
- Incentiva a transparência entre o time de desenvolvimento e o cliente.
- É orientado a resultados, durante todos os ciclos os usuários finais aplicam feedbacks sobre as entregas.

Desvantagens

- Não é tão simples ter um alto envolvimento do usuário final em todos os ciclos.
- A alta quantidade de testes aumentam os custos do produto.
- As diversas mudanças no projeto resultam em um baixo nível de documentação.
- A missão de uma iteração nem sempre irá entregar valor como planejado.
- Este modelo de desenvolvimento exige uma equipe experiente e com uma mentalidade adaptativa.

Exemplo Práticos

Spotify

- **Desafio:** Adaptar-se rapidamente às preferências dos usuários em um mercado de streaming musical altamente competitivo.
- **Abordagem ASD:** Utilização de squads autônomos e sprints curtas para desenvolvimento iterativo.
- **Resultado:** Lançamento rápido de novos recursos e personalização contínua da experiência do usuário, mantendo alta fidelidade e engajamento.

Amazon

- **Desafio:** Atender à demanda crescente e às mudanças rápidas no mercado de comércio eletrônico.
- **Abordagem ASD:** Arquiteturas escaláveis e sistemas modulares permitem adaptação ágil e integração contínua de novas funcionalidades.
- **Resultado:** Melhoria contínua da experiência do usuário, personalização de recomendações e otimização da eficiência operacional.

Conclusão

O Adaptive Software Development (ASD) é uma metodologia ágil que enfatiza a entrega incremental e adaptação contínua em projetos de desenvolvimento de software complexos. Prioriza a aprendizagem ao longo do projeto, promove ciclos adaptativos orientados por riscos e colaboração constante com os clientes para garantir que cada incremento do software atenda às necessidades reais e expectativas do usuário final, melhorando assim a qualidade e a satisfação geral do produto entregue.

Referências

- <https://www.linkedin.com/pulse/agile-methods-12-adaptive-software-development-asd-rodrigo-coimbra-wpexf/>
- <https://www.treinaweb.com.br/blog/o-que-e-asd-adaptative-software-development>
- https://pt.wikipedia.org/wiki/Adaptive_Software_Development
- <https://www.dtidigital.com.br/blog/metodos-ageis-vantagens-e-desvantagens>

OBRIGADA!

