HAROLD MARCEL ILLERT JUNIOR

APS 1 – Ciclos iterativos e incrementais, modelos cascata, RUP e espiral, AGILE.

UniFG – Jaboatão dos Guararapes

1. CICLOS DE VIDA DO SOFTWARE

O ciclo de vida é a estrutura contendo processos, atividades e tarefas envolvidas no desenvolvimento, operação e manutenção de um produto de software, abrangendo a vida do sistema, desde a definição de seus requisitos até o término de seu uso. O modelo de ciclo de vida é a primeira escolha a ser feita no processo de software. A partir desta escolha definir-se-á desde a maneira mais adequada de obter as necessidades do cliente, até quando e como o cliente receberá sua primeira versão operacional do sistema. Processo de software é o conjunto de atividades que constituem o desenvolvimento de um sistema computacional. Estas atividades são agrupadas em fases, como: definição de requisitos, análise, projeto, desenvolvimento, teste e implantação. Em cada fase são definidas, além das suas atividades, as funções e responsabilidades de cada membro da equipe, e como produto resultante, os artefatos. O que diferencia um processo de software do outro é a ordem em que as fases vão ocorrer, o tempo e a ênfase dados a cada fase, as atividades presentes, e os produtos entregues. O modelo de ciclo de vida é a primeira escolha a ser feita no processo de software. A partir desta escolha definir-se-á desde a maneira mais adequada de obter as necessidades do cliente, até quando e como o cliente receberá sua primeira versão operacional do sistema. Não existe um modelo ideal. O perfil e complexidade do negócio do cliente, o tempo disponível, o custo, a equipe, o ambiente operacional são fatores que influenciarão diretamente na escolha do ciclo de vida de software a ser adotado.

1. MODELO CASCATA:

O modelo em cascata tem o grande mérito de ser o primeiro a impor o planejamento e o gerenciamento ao processo de software, que antes era casual. O nome "cascata" foi atribuído em razão da sequência das fases, onde cada fase só começa quando a anterior termina; e da transmissão do resultado da fase anterior como entrada para a fase atual Nesse modelo, portanto, é dada muita ênfase às fases de análise e projeto antes de partir para a programação, a fim de que o objetivo do software esteja bem definido e que seja evitado refazer o trabalho. Devido à sua simplicidade, o modelo em cascata é fácil de ser entendido pelo cliente. É um modelo que supõe um início e fim claro e determinado, assim como uma estimativa precisa de custo logo no início, fatores importantes na conquista do cliente. O problema se dá depois, quando o cliente, após esperar até o fim do processo para receber a primeira versão do sistema, pode não concordar com ela. Apesar de cada fase terminar com uma documentação aprovada, certamente haverá lacunas devido a requisitos mal descritos pelo cliente, mal entendido pelo analista ou por mudança de cenário na organização que exija adaptação de requisitos. O modelo em cascata não prevê revisão de fases.

2. MODELO ESPIRAL:

O modelo proposto por Boehm em 1988 trata de uma abordagem cíclica das fases do processo, onde a cada "volta" ou iteração tem versões evolucionárias do sistema. Este é um modelo guiado por risco, suporta sistemas complexos e de grande porte, onde falhas não são toleráveis. Para isso, a cada iteração há uma

atividade dedicada à análise de riscos e apoiada através de geração de protótipos, não necessariamente operacionais para que haja um envolvimento constante do cliente nas decisões. Cada iteração ou volta é dedicada a uma fase do processo de vida de um software. Ao mesmo tempo, cada volta é seccionada em 4 setores, os quatro setores são explicados da seguinte forma. Na Definição de Objetivos, desempenhos, funcionalidade, entre outros objetivos, são levantados. Visando alcançar esses objetivos são listadas alternativas e restrições, e cria-se um plano gerencial detalhado. Na Análise de Riscos, as alternativas, restrições e riscos anteriormente levantados são avaliados. Neste setor protótipos são utilizados para ajudar na análise de riscos. No Desenvolvimento e Validação um modelo apropriado para o desenvolvimento do sistema é escolhido, de acordo com o risco analisado no setor anterior. No Planejamento da Próxima fase ocorre a revisão do projeto e a decisão de partir para a próxima fase.

3. MODELO RUP:

Derivado da UML e do Processo Unificado de Desenvolvimento de Software, o RUP, Rational Unified Process, é um modelo de processo iterativo e incremental, dividido em fases, orientado a casos de uso. Possui framework de processo e manuais que guiam na utilização das melhores práticas de especificação de projeto. O objetivo do RUP é produzir software com qualidade que satisfaça as necessidades dos clientes dentro de um prazo e orçamento estabelecidos. Este modelo foi desenvolvido pela Rational Software Corporation e adquirido pela IBM, que o define da seguinte maneira: IBM Rational Unified Process, ou RUP, é uma plataforma de processo de desenvolvimento de software configurável que oferece melhores práticas comprovadas e uma arquitetura configurável. O RUP usa templates que descrevem o que é esperado no resultado de cada fase ou cada iteração, identificando as competências e responsabilidades, as atividades e os artefatos. Para descrever as atividades RUP faz o uso de manuais que descrevem técnicas e heurísticas; e de "Mentores de Ferramentas", que explicam o uso da ferramenta para executar a atividade. Os artefatos de cada fase são criados, juntamente com templates e exemplos, para melhor entendimento da equipe e do cliente. Os templates também ajudam no gerenciamento, pois definem o que precisa ser executado. Servem também como guia para que as boas práticas de especificação de projeto não sejam esquecidas no processo de desenvolvimento daquele software. Assim, toda a preocupação dada pelo RUP em disciplinar o processo através de frameworks, guias, templates, faz com que haja uma melhor alocação de pessoas na equipe, padronização do sistema, visão concreta do andamento do projeto.

2. AGILE SOFTWARE DEVELOPMENT

Agile software development é um grupo de metodologias de desenvolvimento de software baseado em desenvolvimento iterativo, onde requisitos e soluções evoluem através de colaboração entre times organizados e multifuncionais. Métodos Agile ou processos Agile geralmente promovem uma gerencia de projetos disciplinada que encoraja inspeção frequente e adaptação, uma filosofia de liderança que encoraja trabalho em equipe, auto-organização e competência, um arsenal das melhores praticas

de engenharia que permita uma entrega rápida de um software de alta qualidade, e um negocio que aproxima o que o cliente necessita com as metas da empresa. O Agile tem vários benefícios são eles: benefícios para o cliente que acredita que os vendedores são mais ágeis nos pedidos de desenvolvimento. Produtos de alta qualidade são desenvolvidos e entregues mais rapidamente com pequenos ciclos, menores processos que no ciclo catarata por exemplo. Benefícios a os vendedores que reduzem o desgaste focando seu desenvolvimento em produtos de qualidade, e reduzindo o tempo de entrega ao mercado, e tem a satisfação do cliente garantida que traduz a ter um melhor relacionamento com o cliente e mais referencias positivas. Benefícios para as equipes de desenvolvimento que reduz o trabalho não produtivo, e mais tempo para trabalhar com o que cada um membro gosta mais de fazer, os membros da equipe também são reconhecidos pois tem requisitos escolhidos par maximizar seu valor para os clientes.

3. DESENVOLVIMENTO ITERATIVO E INCREMENTAL:

Um processo iterativo é aquele que faz progresso através de tentativas sucessivas de refinamento. Por exemplo, uma equipe de desenvolvimento faz sua primeira tentativa para construção de um software, porém, existem pontos de informação falhos ou incompletos em algumas partes. A equipe de forma iterativa refina essas partes até que o produto atinja um nivel satisfatório. Com cada iteração, o software é melhorado através da adição de mais e mais detalhes. Vamos a outro exemplo, em uma primeira iteração, temos que codificar uma tela de relatórios que suporte apenas o tipo mais simples para exibição. Na segunda iteração, podemos adicionar critérios de pesquisa para ampliar o leque de opções no relatório. Finalmente, em uma terceira iteração, poderíamos adicionar gráficos para explanar visualmente o relatório. Um processo incremental é aquele em que o software/produto é construído e entregue por pedaços. Cada pedaço ou incremento representa um subconjunto de funcionalidades completas. O incremento pode ser pequeno ou grande, por exemplo, ele pode variar apenas de uma tela de relatórios simples, para um conjunto altamente flexível de telas de gerenciamento de dados. Cada incremento é totalmente codificado e testado, e a expectativa geral é que o trabalho tenha a conclusão mais completa possível. Métodos ágeis são tanto iterativos, quanto incrementais. São iterativos por que o trabalho realizado é sempre melhorado em ciclos subsequentes. São também incrementais, por que o trabalho planejado é entregue em partes que são adicionadas ao todo do projeto.

REFERÊNCIAS

PRESSMAN, Roger, Engenharia Software, McGraw-Hill. 6ª ed

SPINOLA, Rodrigo, Boas Práticas de Engenharia de Software, 2011

PAULA Filho, Wilson, Engenharia de Software: fundamentos, métodos e padrões, LTC, 3ª Ed

www.cprime.com/resources/what-is-agile-what-is-scrum/ 11/2018