

Processos de Desenvolvimento de Software

Sistemas de software têm desempenhado um papel cada vez mais preponderante no dia-a-dia das pessoas, e em muitas situações o funcionamento correto ou incorreto desses sistemas pode ser a diferença entre a vida e a morte. Entretanto, a construção de sistemas é complexa, pois deve lidar com requisitos intransigentes, restrições de integridade e a necessidade de um vasto conhecimento sobre a aplicação para que as interações esperadas entre o software e o ambiente possam ser adequadamente descritos. Quando os requisitos não são totalmente compreendidos, registrados e comunicados para a equipe de desenvolvimento, muito provavelmente, haverá discrepância entre o que o sistema construído faz e o que ele deveria fazer.

Hoje em dia o software assume um duplo papel. Ele é o produto e ao mesmo tempo o veículo para entrega do produto. Como produto ele disponibiliza o potencial de computação presente em computador, ou mais amplamente numa rede de computadores acessível pelo hardware local. Quer resida em um telefone celular, quer opere em um computador de grande porte (Mainframes) o software é transformador de informações – produzindo, gerando, adquirindo, modificando, exibindo, ou transmitindo informação, que pode ser tão simples como um bit ou tão complexa como uma apresentação multimídia. Como veículo usado para a entrega do produto, o software age como base para controle do computador – sistemas operacionais – para a comunicação da informação e para a criação e o controle de outros programas.

Atualmente, o desenvolvimento de software não ocorre como no passado, o programador solitário foi substituído por uma equipe de especialistas com cada um se concentrando numa parte da tecnologia necessária para produzir uma aplicação. No entanto, os mesmos questionamentos feitos ao programador solitário estão sendo feitos nos dias atuais:

Por que leva tanto tempo para concluir o software?

Por que os custos de desenvolvimentos são tão altos?

Por que não podemos achar todos os erros antes da entrega do software aos clientes?

Por que continuamos a ter dificuldades em avaliar o progresso enquanto o software é desenvolvido?

Há alguns anos tem-se discutido maneiras de contornar a complexidade do software, visto que a cada dia novas áreas de aplicação têm surgido e exigido mais confiabilidade e precisão dos softwares já existentes e dos que ainda virão a ser construídos. Não há consenso sobre qual a melhor prática para o desenvolvimento de sistemas de software, mais existe um esforço em encontrar soluções para reduzir as dificuldades oriundas da própria natureza do software, da sua complexidade, de necessidades de cumprir seus objetivos e da rapidez com que sofre alterações. A Engenharia de software propõe a adoção da disciplina para lidar com essas dificuldades, tentando reduzir ao máximo a influência delas no processo de desenvolvimento de software.

Engenharia de software

A Engenharia de software é uma disciplina que reúne metodologias, métodos e ferramentas a ser utilizadas, desde a percepção do problema até o momento em que o sistema desenvolvido deixa de ser operacional, visando resolver problemas inerentes ao processo de desenvolvimento e ao produto de software.

O objetivo da Engenharia de software é auxiliar no processo de produção de software, de forma que o processo de origem a produtos de alta qualidade, produzidos mais rapidamente e a um custo cada vez menor. A Engenharia de software segue o conceito de disciplina na produção de software, fundamentado nas metodologias, que por sua vez seguem métodos que utilizam de ferramentas automáticas para englobar as principais atividades do processo de produção.

Metodologia de desenvolvimento de software

Já faz alguns anos que o desenvolvimento de software deixou de ser sinônimo apenas de código. Hoje em dia, sabe-se que é necessária a utilização de uma metodologia de trabalho.

Mas o que é necessariamente uma metodologia de software? Entende-se por metodologia, como a maneira – forma – de se utilizar um conjunto coerente e coordenado de métodos para atingir um objetivo, de modo que se evite, tanto quanto possível, a subjetividade na execução do trabalho. Fornecendo um roteiro, um processo dinâmico e interativo para desenvolvimento estruturado de projetos, sistemas ou software, visando à qualidade e produtividade dos projetos.

O dicionário define metodologia como um conjunto de métodos, regras e postulados empregados por uma disciplina: um procedimento particular ou conjuntos de procedimentos.

É objetivo de uma metodologia definir de forma clara “quem” faz “o que”, “quando”, “como”, e até mesmo “onde”, para todos os que estejam envolvidos diretamente ou não com o desenvolvimento de software. Deve definir também qual o papel dos técnicos, dos usuários, e o da administração da empresa no processo de desenvolvimento. Com isso, evita-se a situação a qual o conhecimento sobre o sistema é de poucos, comumente apelidados, de “os donos do sistema”. Além disso, deve instruir um conjunto de padrões preestabelecidos, de modo a ser evitar a subjetividade na abordagem, a fim de garantir fácil integração entre os sistemas desenvolvidos. Com isso, o uso de uma metodologia possibilita:

Ao gerente: controlar o projeto de desenvolvimento de software mantendo o rumo do projeto sobre controle para que não haja desvios de planejamentos de custos e prazos, que, se negligenciados ou mal conduzidos, podem por em risco o sucesso do projeto.

Ao desenvolvedor: obter a base para produzir de maneira eficiente, software de qualidade que satisfaça os requisitos estabelecidos.

Muitas vezes, o uso de uma metodologia é encarado como cerceamento da criatividade dos técnicos, ou como, acréscimo de burocracia, leia-se documentações, por muitos tidos como desnecessário a construção de software. Uma metodologia não deve limitar a criatividade profissional, mas deve ser um instrumento que determine um planejamento sistemático, que harmonize e coordena as áreas envolvidas. O que limita a criatividade não é a metodologia, mas os requisitos de qualidade e produtividade de um projeto.

Como uma metodologia é um conjunto de métodos, convém definir o que é um método e qual o seu objetivo.

Um método é abordagem técnica passo a passo para realizar uma ou mais tarefas indicadas na metodologia. Ou seja, é (são) o(s) procedimento(s) necessário(s) a ser (em) adotado(s) para atingir um objetivo. Já uma técnica, pode ser compreendida como sendo um modo apropriado de se investigar sistematicamente um universo de interesse ou domínio do problema. Para tanto, utiliza-se de uma notação. Como exemplo de técnica, temos: Análise estruturada, Análise Essencial, Projeto Estruturado, Análise Orientada a Objetos.

A escolha de uma metodologia a ser utilizada no desenvolvimento, deve ser realizada com base na natureza do projeto e do produto a ser desenvolvido, dos métodos e ferramentas a serem utilizadas e dos controles e produtos intermediários desejados.

O uso de metodologia, mesmo que ainda não fortemente sedimentada, no desenvolvimento de software é de extrema importância, para que o sistema construído atenda as necessidades dos interessados, com um mínimo de qualidade.

Saiu na DevMedia!

Dê o próximo passo após o HTML/CSS!:

HTTP (Hypertext Transfer Protocol) é o protocolo base para as aplicações web e clientes (como aplicativos mobile, entre outros) que se conectam a web APIs. Ele permite a comunicação entre clientes e servidores através da internet.

Você sabe o que é requisição e resposta?:

Descubra o que acontece quando o navegador dispara uma requisição e, do outro lado, como um servidor web responde a isso. Essa é a hora de falar sobre um dos fundamentos da internet, o protocolo HTTP.

API REST + Cliente web React + Mobile:

É hora de por em prática o que você sabe criando um projeto completo! Na série de hoje vamos analisar requisitos, codificar uma API RESTful, um cliente mobile multiplataforma, um site e uma aplicação web de moderação.

Tem como objetivo esclarecer aos desenvolvedores o funcionamento das metodologias no desenvolvimento de software e o modo como elas evoluíram no decorrer dos anos.

Em meados da primeira guerra mundial tivemos uma evolução significativa no segmento corporativo.

Nesta época o mundo passava por intensas transformações e isto provocou drásticas mudanças no ciclo produtivo das empresas e percebeu-se a necessidade de controlar o seu processo de trabalho.

Baseado nestas transformações houve a necessidade de se aplicar o conceito de dinamização de processos e daí surgiu a necessidade de se administrar grandes volumes de dados em organizações de todas as esferas.

Com a criação dos computadores comerciais após a segunda guerra mundial tivemos um aumento significativo na dinamização da indústria de computadores e, conseqüentemente, o processo de construção de softwares, para que os mesmos automatizassem processos manuais e pudessem avaliar situações complexas que são parte integrante do cotidiano das organizações.

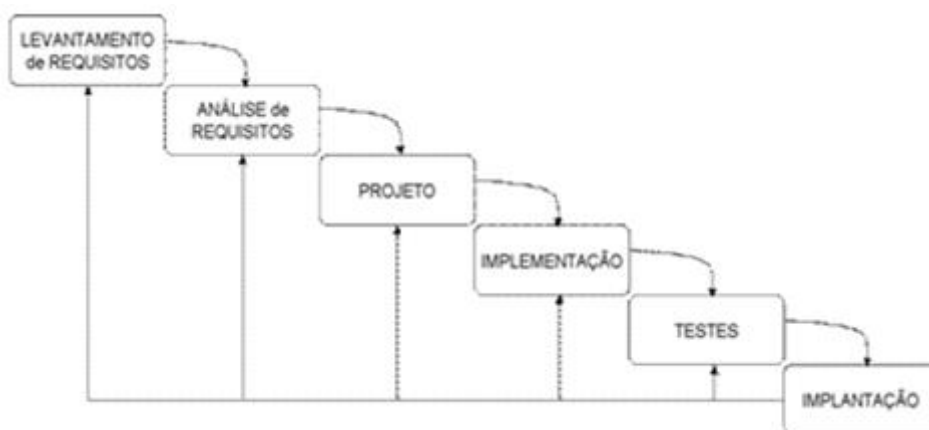
E a partir desse cenário, criou-se modelos de desenvolvimento de softwares que atendessem a determinadas necessidades específicas e ao mesmo tempo pudessem ser utilizados na elaboração softwares sem grandes complexidades.

A seguir são apresentados os modelos de desenvolvimento de softwares.

Modelo Cascata

O Modelo Cascata, também chamado de Clássico ou Linear, caracteriza-se por possuir uma tendência na progressão sequencial entre uma fase e a seguinte. Eventualmente, pode haver uma retroalimentação de uma fase para a fase anterior, mas de um ponto de vista macro, as fases seguem fundamentalmente de forma sequencial.

A figura abaixo nos dá uma idéia visual do conceito apresentado acima.

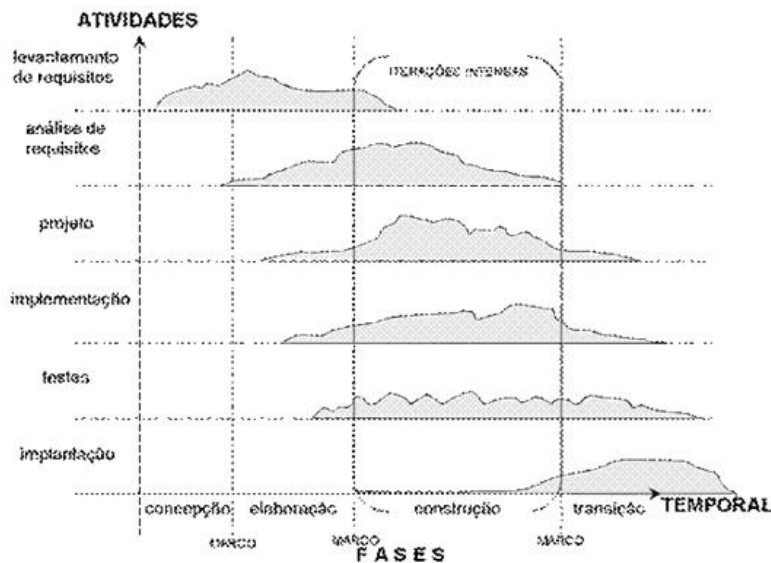


Modelo Iterativo e Incremental

O Modelo de ciclo de vida Iterativo e Incremental foi proposto justamente para ser a resposta aos problemas encontrados no Modelo em Cascata. Um processo de desenvolvimento, segundo essa abordagem, divide o desenvolvimento de um produto de software em ciclos. Em cada ciclo de desenvolvimento, podem ser identificadas as fases de análise, projeto, implementação e testes.

Essa característica contrasta com a abordagem clássica, na qual as fases de análise, projeto, implementação e testes são realizados uma única vez.

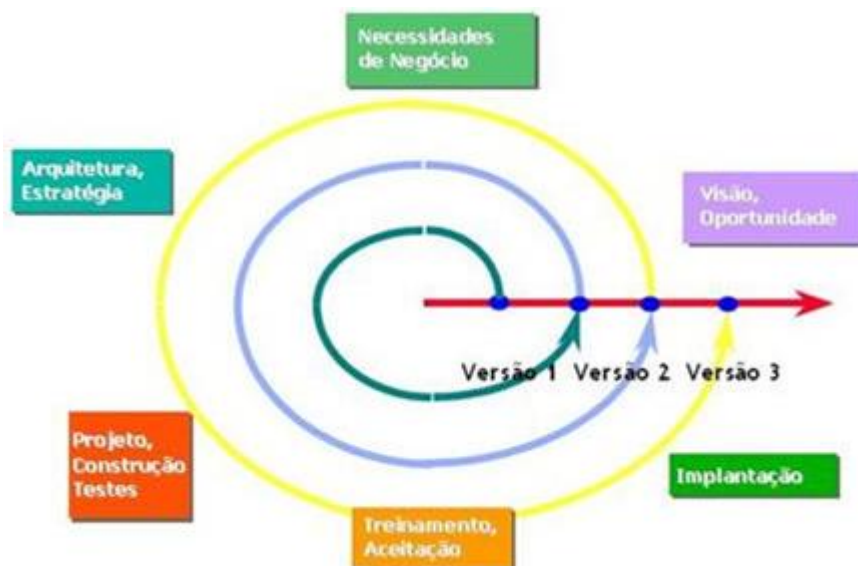
No Modelo de ciclo de vida iterativo e incremental, um sistema de software é desenvolvido em vários passos similares (iterativo). Em cada passo, o sistema é estendido com mais funcionalidades (incremental).



Existe um processo de desenvolvimento de software que é o principal representante da abordagem de desenvolvimento incremental e iterativo. Conhecido como RUP - Rational Unified Process (Processo Unificado Racional). E foi patenteado pela empresa Rational, onde trabalham os famosos três amigos (Jacobson, Booch e Rumbaugh).

Agora vamos explicar um pouco da metodologia em espiral que consiste em ser desenvolvido em uma sequência de iterações e nisso cada iteração corresponde a uma volta na espiral e cada fase ou atividade é um setor, um "ângulo" da volta.

A figura abaixo nos dá uma idéia visual do conceito apresentado acima.



E com o surgimento desse modelo podemos realizar a construção de versões básicas dos produtos em prazos curtos e deixar novos requisitos para implementação futura, no momento em que se tornam essenciais.

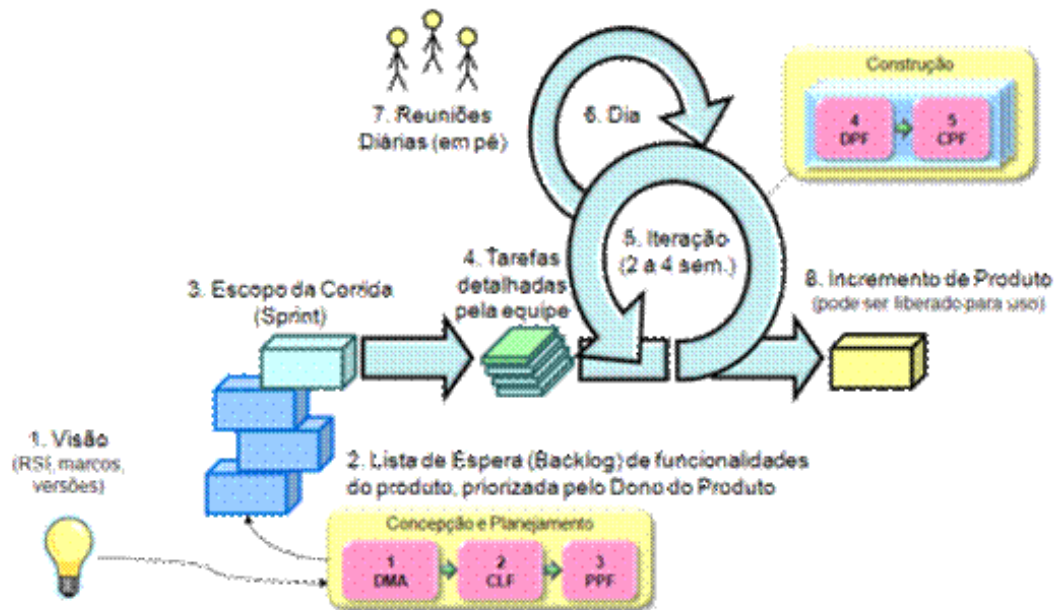
Conforme o avanço tecnológico foi ocorrendo à mudança de paradigma no desenvolvimento de sistemas aconteceu e atualmente trabalhamos com o conceito de metodologia ágil.

Muitos ainda se perguntam o que é e qual é a função desta metodologia e em que ela irá melhorar os processos existentes na elaboração de um software.

Vamos entender o que seria primeiramente esse Desenvolvimento ágil de software (do inglês Agile software development) ou Método ágil é um conjunto de metodologias de desenvolvimento de software.

O desenvolvimento ágil, tal como qualquer metodologia de software, providencia uma estrutura conceitual para reger projetos de engenharia de software.

A figura abaixo nos dá uma idéia visual do conceito apresentado acima.



Porém existem vários modelos desta metodologia que são:

Cleanroom

XP

Scrum

Estarei detalhando cada um deles abaixo.

Cleanroom é uma metodologia muito utilizada no desenvolvimento de software. É considerada "pesada" pelos padrões da Engenharia de Software, mas muito difundida no desenvolvimento de grandes projetos corporativos.

O processo é baseado no projeto apurado das funções, que são analisadas pelo método de revisão-par com o objetivo de verificar se fazem realmente o que foram especificadas a fazer. Por analogia, podemos comparar esta metodologia com as salas limpas na fabricação de semicondutores, que eliminam a necessidade de se limpar wafers de silício pelo fato de que eles nunca começam sujos. O desenvolvimento Cleanroom remove a necessidade de depuração do programa, assegurando que os erros nunca começam introduzidos no sistema.

Programação Extrema (do inglês eXtreme Programming), ou simplesmente XP, é uma metodologia ágil para equipes pequenas e médias e que irão desenvolver software com requisitos vagos e em constante mudança. Para isso, adota a estratégia de constante acompanhamento e realização de vários pequenos ajustes durante o desenvolvimento de software.

Scrum é um método ágil que foi concebido como um estilo de gerenciamento de projetos em empresas de fabricação de automóveis e produtos de consumo, por Takeuchi e Nonaka no artigo "The New New Product Development Game". Eles notaram que projetos usando equipes pequenas e multidisciplinares

