

Instituto INFNET

ESTI – Escola Superior da Tecnologia da Informação



Projeto de Bloco: Desenvolvimento Java

Jennifer Omena

Professor: Armênio Torres Santiago Cardoso

Rio de Janeiro – RJ

Junho de 2022

Modelo Cascata

O modelo Sequencial Linear, mais conhecido como Modelo Cascata, aborda o desenvolvimento dos projetos em etapas, e essas etapas são: Comunicação; Planejamento; Modelagem; Construção; Implantação.

Para iniciarmos temos que entender primeiramente que qualquer projeto apresentado, nada mais é do que a demanda de alguém ou um grupo, seja interna, quando a própria empresa/equipe solicita o desenvolvimento/solução de algo; seja uma demanda externa, quando alguém, normalmente denominado “Cliente”, apresenta sua demanda a equipe para que ela execute/desenvolva o projeto desejado.

Pegando essa premissa, e entendendo que na maioria dos casos os projetos tem origens externas, ou seja, são solicitados por “Clientes”, podemos entender que quem solicita na maioria das vezes são pessoas que não tem o conhecimento técnico para tal, e por isso muitas das vezes não tem a expertise de apresentar a solicitação do projeto do modo que ela realmente deseja, logo normalmente a solicitação que é feita normalmente é genérica, mas a verdadeira finalidade do mesmo é muitas das vezes específica, e o cliente não consegue passar isso da forma mais precisa e correta para a equipe de desenvolvedores. Por esse motivo, o modelo cascata é muito bem aplicado quando as diretrizes do projeto são muito bem definidas, como por exemplo projetos específicos para órgãos públicos e licitações.

Pensando nisso, nós já podemos entender de cara que a primeira etapa, que é a Fase de Comunicação, onde temos a Inclusão do Projeto, que é a contratação do serviço, e o Levantamento de Requisitos, é uma das se não a mais sensível do projeto, podendo ela por si só definir se o projeto irá ser aprovado pelo cliente final ou não. Pois essa etapa necessita, diretamente que o cliente saiba passar com detalhes e objetivamente qual a finalidade, destino e modo como deve ser apresentado o projeto, e por sua vez a equipe de desenvolvimento tem que ter a expertise de captar com fidedignidade as solicitações e diretrizes do cliente para que possa desenvolver/realizar o projeto como ele deve ser e conseguir entender quais são as necessidades do usuário final e se elas são compatíveis com as solicitações feitas pelo cliente.

A partir dessas informações se pode iniciar a fase de Planejamento, onde será estimado o cronograma de trabalho; as atividades técnicas que serão realizadas; os recursos necessários, sendo eles tanto financeiros, quanto de equipamentos e de pessoal; a análise e gestão dos riscos envolvidos no projeto; e por fim o(s) produto(s) a ser(em) produzido(s);

Em seguida se inicia a etapa de Modelagem, onde a partir da análise das informações obtidas, na primeira etapa onde foram feitos o Levantamento de Requisitos, com o cliente e com os usuários finais do projeto, se constroem modelos que irão permitir ao desenvolvedores e ao cliente, um melhor entendimento de como poderá ficar o projeto final, e não só isso, poderá apresentar se o projeto é viável, tanto financeiramente, quanto tecnologicamente,

e se o projeto irá funcionar conforme o solicitado para poder atender aos requisitos demandados pelo solicitante.

Caso o ou os modelos sejam aceitos, se avança para a etapa de Construção. Nela é desenvolvida/codificada/gerada o código do software, através da tradução da descrição computacional realizada na etapa anterior do projeto. Essa geração pode ser automática ou manual. É nessa etapa, também, que são realizados os testes necessários para encontrar possíveis falhas/ erros no(s) código(s).

Por fim, se chega à etapa de Fase de Implantação, onde o projeto é entregue ao cliente. Essa entrega envolve, normalmente, a sua instalação e configuração no ambiente do(s) usuário(s), e o treinamento dele(s). Além é claro, caso seja contratado, haverá o serviço de manutenção e aperfeiçoamento do projeto a partir dos feedbacks apresentados pelo cliente e pelos usuários finais do projeto.

O Modelo Cascata, como pode ter ficado claro durante a sua apresentação, pode e normalmente apresenta algumas deficiências, que podem dificultar ou atrapalhar o desenvolvimento do projeto.

Algumas dessas desvantagens se dão por conta do modelo ser muito rígido, o que acaba dificultando mudanças durante o desenvolvimento do projeto. Por conta disso, parte da equipe fica ociosa durante as etapas do projeto, o que acaba gerando a demanda de mais tempo para que o cliente ou solicitante do projeto possa ver alguma funcionalidade do programa. Isso se deve, primeiramente ao fato da equipe perder muito tempo para que se entenda realmente qual a finalidade e necessidades do projeto, e só então a partir das informações apresentadas pelo cliente e obtidas com o futuro público alvo do projeto, o projeto poder iniciar a etapa de modelagem, e só depois dessa etapa estiver concluída se poder passar para a fase de construção e assim por diante. O que acaba gerando uma trava no projeto, por ele não poder andar sem antes a etapa anterior estiver plenamente concluída.

Modelo Iterativo e Incremental

Felizmente não temos somente esse modelo para trabalharmos, visto que ele foi concebido por volta dos anos 70. Com o passar dos anos novas metodologias foram surgindo, e uma delas é o Modelo Iterativo & Incremental, que por natureza vem para tentar sanar as limitações e travas do modelo anteriormente apresentado (Modelo Cascata).

A principal característica deste modelo é dividir e desenvolver, separadamente, em diversos “Blocos” o projeto. Assim se pode fazer o desenvolvimento de várias partes do projeto ao mesmo tempo, não ficando preso a etapas anteriores, como acontece com o Modelo Cascata e caso algum bloco

dê problema, ou precise passar por uma revisão, basta refazer o ciclo no próprio bloco, sem assim afetar os outros blocos que estão ou já foram desenvolvidos.

Cada Bloco, pode seguir as mesmas etapas do Modelo Cascata, buscando desenvolver uma parte do projeto, como se fosse uma peça de quebra-cabeça, onde de tempos em tempos blocos que são concluídos são anexados aos que já estão prontos, e assim ao se colocar a última peça do quebra-cabeça o projeto estará concluído, mas caso haja a necessidade de se fazer uma correção, basta retirar somente a peça necessária (Bloco) e refazer o ciclo ou trabalhar somente nela, não afetando as outras peças (Bloco).

Assim é possível, ao cliente ter em mão partes do projeto não só para testar, mas também para entender se o projeto está atendendo a sua demanda, e caso necessário já ir implantando e solucionando partes das demandas que originaram a necessidade do desenvolvimento do projeto. Com isso não só a equipe de projetista ganha tempo, como também o cliente, pois ele já pode ir implantando e entendendo o funcionamento do sistema, para quando ele estiver totalmente pronto, a fase de implantação final ser mais tranquila e eficiente.

Por ironia do destino, a sua principal característica é sua principal desvantagem, pois como cada Bloco é desenvolvido separadamente, pode haver o risco de os blocos não serem compatíveis, podendo trazer uma demanda de tempo maior para se solucionar a incompatibilidade, ou dependendo podendo até acarretar na inviabilidade ou cancelamento do projeto, através não só dos problemas apresentados, como também do surgimento de novas demandas pelo cliente com o passar do tempo, podendo assim deixar o projeto muito longo e acabando por deixá-lo inviável, seja pelo cliente ter ficado sem recursos financeiros para continuar com o projeto, seja pelo prazo necessário para sua implementação ou lançamento ter se esgotado/ultrapassado.

Relação entre OO, UML e RUP

Com o passar dos anos sempre novas metodologias vão surgindo para sanar os problemas evidenciados pelas metodologias já existentes, conforme foi apresentado anteriormente. Já nos anos 90, foi desenvolvido o modelo RUP (Ration Unifed Process), comumente chamado de Processo Unificado, que foi desenvolvido pela Rational Software Corporation.

Este tem como característica ser iterativo e incremental. Ele não tem somente semelhança com o nome da metodologia explicada anteriormente, mas também a forma de dividir em partes do projeto em blocos, e ir desenvolvendo eles todos praticamente ao mesmo tempo, e no final juntar tudo e finalizar o projeto, mas mais do que isso, agora se busca que cada parte tenha mais complexidade, ou seja tenha mais participação referente ao mesmo projeto e não só isso se busca e se permite que cada bloco possa e deva ser revisitado, caso necessário para atualizações, correções e/ou incrementações, sendo assim se diminui a chance de incompatibilidade em áreas sensíveis ao projeto,

diminuindo o tempo demandado para a conclusão do mesmo e aumentando a sua confiabilidade e funcionalidade, vale ressaltar que esse modelo normalmente é utilizado em projetos que necessitam de uma demanda de prazo de conclusão maior, devido à alta complexidade do sistema a ser desenvolvido, havendo a necessidade de uma equipe maior e por consequência um controle de maior de todo o projeto.

Esse aumento na qualidade e complexidade do projeto, faz com que cada “bloco” desenvolvido, também seja mais complexo, e isso se deve principalmente pelo fato do Modelo RUP utilizar uma abordagem de OO (Orientação a Objetos) em sua concepção e ser projetado e documentado utilizando o UML (Unified Modeling Language) para ilustrar os processos. Assim, através de pequenos ciclos do projeto, o sistema/projeto/software, é melhorado através da adição de mais detalhes, e a correção de possíveis falhas, o que resulta em um incremento geral ao projeto demandado.

O Modelo RUP organiza o desenvolvimento em 4 Fases (Concepção, Elaboração, Construção e Transição), e cada uma delas tem no mínimo uma interação, que é normalmente chamado de “Ciclo de Vida”. São nessas interações que são apresentados ao cliente o andamento do serviço, para que ele possa validar e assim se poder continuar o desenvolvimento.

Na Fase de Concepção se define o escopo do projeto, ou seja, é a fase preliminar, onde se faz os devidos levantamentos de requisitos, análise de riscos, definição de valores e do prazo de entrega.

Já na Fase de Elaboração, se há a definição do plano do projeto, especificações das características e da arquitetura que será utilizada, e também de uma análise mais profunda e detalhada de riscos e dos custos.

Posteriormente se chega na Fase de Construção, onde ocorre a codificação do sistema contratado.

E por fim, há a Fase de Transição, onde ocorre após os testes finais a implantação do sistema, bem como o treinamento dos usuários do novo sistema.

Outro ponto fundamental do modelo RUP, é que ele possui atividades lógicas, chamadas comumente de disciplinas, sendo elas muito bem trabalhadas e desenvolvidas, o que acaba gerando uma organização melhor e mais eficiente do projeto, com em decorrência disso se obtém um controle maior não só do andamento do projeto, mas também do seu desenvolvimento. Possibilitando o desenvolvimento de projetos mais complexos, de alta qualidade e que demandem um prazo de conclusão mais longo.

E essas atividades lógicas são: Modelagem de Negócio; Requisitos; Análise e Design; Implementação; Testes; Implantação; Gerenciamento de Configuração e Mudança; Gerenciamento de Projeto; e Ambiente.

As quatro fases descritas anteriormente interagem com as disciplinas apresentadas acima, assim se consegue uma organização e controle maior e melhor do produto final.

Toda essa complexidade envolvendo o modelo RUP, se deve principalmente ao tipo de projeto que normalmente ele é utilizado, como já salientado anteriormente, esse modelo é muito utilizado, e funciona muito bem em projetos de maior complexidade, que necessitem de uma equipe maior, que requerem uma documentação mais detalhada, que necessitem de aferição dos custos e prazos, e que não necessitem de uma entrega tão imediata.

Bibliografia

Materiais disponibilizados nos Roteiros de Aprendizagem do Moodle.

<https://www.treinaweb.com.br/blog/o-que-e-rup-rational-unified-process>