Lucas Nogueira Padrão

Análise e Projeto de Sistemas

Como analisar, planejar, desenvolver e implementar sistemas de informação

editora
VIENA
1ª Edição
Bauru/SP
Editora Viena
2014

Sumário

Lista de Siglas e Abreviaturas

XP	*eXtreme Programming.*
DSDM	*Dynamic Systems Development Method.*
UML	*Unified Modeling Language.*
PMBOK	*Project Management Body of Knowledge.*
JAD	*Joint Application Development.*
CEP	*Código de Endereçamento Postal.*
MVC	*Model-View-Controller.*
SGBD	*Sistemas Gerenciadores de Banco de Dados.*
DML	*Data Manipulation Language.*
DDL	*Data Definition Language.*
CNPJ	*Cadastro Nacional de Pessoa Jurídica.*
CPF	*Cadastro Pessoal Física.*
CRM	*Customer Relationship Management.*
ANSI	*American National Standards Institute.*
ISO	*International Organization for Standardization.*
SQL	*Structured Query Language.*
PL	*Procedual Language.*
QBE	*Query By Example.*
ISBN	*International Standard Book Number.*
OLAP	*On-Line Analytical Processing.*
OLTP	*Online Transaction Processing.*
KB	*Kilobyte.*
DARPA	*Defense Advanced Research Projects Agency.*
ARPANET	*Advanced Research Projects Agency Network.*
IMP	*Interface Message Processor.*
NCP	*Network Control Protocol.*
DNS	*Domain Name System.*
LAN	*Local Area Network.*
MAN	*Metropolian Area Network.*
WAN	*Wide Area Network.*
TCP/IP	*Transmission Control Protocol/Internet Protocol.*
IHL	*Internet Header Length.*
UDP	*User Datagram Protocol.*
URL	*Uniform Resource Locator.*
HTML	*HyperText Markup Language.*
RMI	*Remote Method Invocation.*
TI	*Tecnologia da Informação.*
CRC	*Class-Responsability-Collaborator.*
EAP	*Estrutura Analítica do Projeto.*
ITIL	*Information Tecnology Infrastructure Library.*
ITSMF	*Information Technology Service Management Forum*
CCTA	*Central Computer and Telecommunications Agency.*

1
Análise de Sistemas

1. Análise de Sistemas

Em seu livro sobre Metas, o autor Brian Tracy afirma que é a clareza de objetivo que leva à realização da meta. E de fato, sem sabermos o que queremos, jamais vamos atingir algum objetivo. Se quero comprar uma bola, devo ter isso em mente e realizar todos os processos necessários para adquiri-la. Se quero produzir um software, devo saber que tipo de software quero produzir para iniciar seu processo de desenvolvimento.

Analisar consiste em enumerar ou descrever as principais características de um objeto. Para analisar a bola que queremos comprar, deveríamos dizer quais cores estão presentes na bola, o tamanho, o formato, a rigidez. Tudo isso tendo em vista o esporte que praticaremos com a bola: futebol, vôlei, tênis, rugby etc.

A Análise de Sistemas consiste em esclarecer as características do software que queremos. Assim poderemos ter a clareza para desenvolver um software que atenda exatamente às necessidades do cliente.

Vamos imaginar um software que realiza cálculos básicos. Devemos ter em conta que o software deva calcular as operações básicas (adição, subtração, multiplicação e divisão) retornando o resultado correto. Então, a análise de software deve responder à pergunta "Quais tarefas o software precisa realizar?". Perceba que uma resposta parcial ou incorreta levará a um software que não atenda à necessidade do cliente. Se um cliente quer somar números, não faz sentido oferecer uma calculadora que calcule a raiz cúbica dos números informados (uma operação mais complexa) ainda que funcione corretamente.

Em geral, o Analista de Sistemas é uma peça fundamental, que conhece um pouco de todos os elementos envolvidos no sistema, desde pessoas até tecnologias e práticas comuns. Sua função básica consiste em colher os desejos do cliente e transformá-los em algo realizável tecnologicamente.

Os sistemas de informação têm como consequência a mudança da organização. Por isso, o analista de sistemas é um personagem envolvido em um processo delicado, o processo de transformação. Assim, eles devem compreender o ambiente técnico da organização para que o sistema projetado seja condizente com as atividades envolvidas. Desse modo, o estudo de administração de empresas e produção se torna um quesito necessário ao analista de sistemas, afinal, este é visto como um funcionário para resolver problemas empresariais.

Não raramente, os analistas precisam se comunicar diretamente com os usuários, gerentes e programadores, para que possam ter a visão integrada do projeto a ser desenvolvido. Desse modo, habilidade de comunição é imprescindível ao analista de sistemas. Porém, há diversas especializações na área de análise de sistemas.

- **Analista de Negócios**: Analisa os aspectos empresariais do sistema.
- **Analista de Sistemas**: Descobre como resolver os problemas empresariais com o sistema.
- **Analista de Infraestrutura**: Assegura o suporte do novo sistema na empresa.
- **Gerente de Mudanças**: Planeja e executa as mudanças necessárias para evitar danos e perdas empresariais.
- **Gerente de Projetos**: Gerencia a equipe com o objetivo de atingir um objetivo concreto.

1.1. Ciclo de Vida

Da mesma forma que na engenharia civil, o processo de construção de um sistema é composto por estágios. Em geral, a literatura técnica comporta quatro tipos de estágios ou fases, que podem ser imaginadas como passos para atingir o estágio final, que consiste no sistema pronto para o uso.

As quatro fases são as seguintes:

- **Planejamento**: Estágio em que as oportunidades são identificadas, o projeto é definido e o plano do projeto é designado.
- **Análise**: Fase para determinar a estratégia de desenvolvimento, recolher os requisitos e modelar os processos que serão automatizados pelo sistema.
- **Projeto**: Etapa em que todos os componentes do sistema são planejados, projetados.
- **Implementação**: Fase em que o sistema é construído, instalado e manutenido.

1.2. Planejamento

Talvez a questão mais difícil para um administrador de empresas é saber se há necessidade de construir um novo sistema para a empresa. Certamente a resposta envolve os custos da produção e o lucro ou prejuízo que poderia ser obtido. Também é necessário decidir como a equipe de desenvolvimento irá construir o determinado sistema. Para que isso seja realizado seguramente, seriam necessários dois passos.

O primeiro passo consiste em determinar e avaliar a viabilidade do sistema. Quando um departamento empresarial faz uma requisição de um sistema, em geral, suas necessidades vêm resumidas na forma de um problema a resolver. Contudo a implementação dessa solução pode ser inviável econômica, técnica ou fisicamente. Ainda há a possibilidade de o sistema não ser usado efetivamente e não produzir lucros. Desse modo, a viabilidade deve ser analisada com cuidado.

Já o segundo passo consiste em iniciar a gerência do projeto. Nessa etapa, um gerente de projeto é requisitado para controlar e dirigir o projeto, oferecendo para isso um planejamento do projeto, que descreve como os funcionários irão interagir durante o projeto e como este irá caminhar durante sua construção.

Na etapa de planejamento, o analista de sistemas e os administradores da empresa devem trabalhar juntos para conseguir delinear as necessidades e características do sistema. Embora o objetivo de todo negócio seja o lucro, é necessário investimento para que o negócio se concretize, e a área de tecnologia da informação é uma das áreas que mais carecem de investimento, atualmente. Desse modo, é preciso que o investimento retorne lucro para o proprietário do negócio, de forma que ele precisará acompanhar o sistema durante sua construção a fim de verificar se ele está sendo construído na direção certa, a de gerar lucro.

Certamente o tamanho do projeto irá interferir diretamente no investimento realizado e, de acordo com o tamanho do investimento, mais atenção requererá dos proprietários e investidores.

Como visto anteriormente, os requisitos são as funcionalidades que o sistema deverá prover. Assim, é preciso analisar quais funcionalidades são imprescindíveis para o negócio em questão. Porém, como os proprietários e investidores não detêm

conhecimento técnico, é preciso que as funcionalidades sejam explicadas em linguagem de alto nível, isto é, de forma compreensiva para os interlocutores. Assim, os investidores poderão ter uma noção concreta do valor que o desenvolvimento de um sistema poderá gerar.

Há, pelo menos, duas maneiras de classificar o valor gerado pelos sistemas de informação. Os valores tangíveis são aqueles que podem ser numericamente mensuráveis, como o aumento de 2% dos clientes. Porém, há outros valores que são intangíveis, por serem difíceis de medir numericamente, como a satisfação do cliente. Uma vez identificado o valor gerado pelo sistema e a sua importância para a empresa, o projeto pode iniciar normalmente. Na maioria das grandes empresas, a iniciação do projeto ocorre por meio da requisição de sistema.

A requisição de sistema consiste em um documento que descreve as razões para construção de um sistema específico e a previsão de geração de valor. Por meio de tal documento, o projeto é submetido à aprovação por um comitê, que pode ser formado por executivos da empresa. O comitê investigará as razões da construção do sistema e poderá aprová-la ou rejeitá-la.

1.2.1. Viabilidade

Uma vez aprovado, o projeto deve ser submetido à análise de viabilidade. O comitê requisita uma descrição mais detalhada do projeto e tenta estabelecer os limites e as oportunidades geradas por meio dele. Essa etapa, não menos importante, é chamada de estudo de viabilidade e compõe-se de três partes: análise da viabilidade técnica, análise da viabilidade financeira e análise da viabilidade organizacional.

Resumidamente, a análise de viabilidade técnica consiste em saber se é possível construir o sistema. É bem famoso o caso do diretor de cinema James Cameron, que, para realizar a filmagem da ficção "Avatar", teve de esperar mais de dez anos até que a tecnologia pudesse ser produzida. Assim, há projetos inviáveis tecnicamente, podendo-se afirmar que a análise de viabilidade técnica consiste em uma espécie de análise de risco, já que, caso não haja técnica suficiente para realizar o projeto, ele correrá sério risco de não ser finalizado.

Em geral, três fatores podem ser considerados na análise de viabilidade técnica. O primeiro deles consiste na familiaridade da equipe técnica com a tecnologia desenvolvida. Caso um sistema seja produzido em uma linguagem de programação desconhecida pela maioria dos participantes, é possível que haja desconforto no desenvolvimento do projeto, ocasionando inviabilidade técnica. O segundo consiste no tamanho do projeto. Obviamente, um projeto demasiado grande para uma organização comercial pode ser inviável. Também é preciso considerar, em terceiro lugar, a compatibilidade da tecnologia utilizada no novo projeto com as tecnologias empregadas na empresa. Um sistema de cadastro que exija plataforma Unix em uma empresa onde todos os computadores utilizam plataforma Windows é totalmente inviável, por exemplo.

A avaliação da viabilidade técnica, muitas vezes, pode se valer da experiência do analista ou das avaliações de projetos anteriores, que descrevem detalhadamente os problemas encontrados e as soluções propostas para resolvê-los.

Há também de se considerar a questão da viabilidade econômica, que pode ser brevemente descrita em termos de benefícios e custos. Existem custos de desenvolvimento e custos operacionais, além dos benefícios tangíveis e intangíveis. Os custos de desenvolvimento envolvem aqueles custos concernentes à criação do sistema, como salários e despesas de hardware. Já os custos operacionais são aqueles destinados a manter o projeto funcionando, como licensas e gastos com comunicação. Como dito acima, os benefícios tangíveis podem, ao contrário dos intangíveis, ser mensurados.

Sabemos que o homem é um ser cultural e que por causa disso adere fortemente a certos hábitos e comportamentos. Por isso, a mudança da cultura organizacional é uma das mudanças mais difíceis de realizar no mundo corporativo. Assim, é possível que a implantação de um novo sistema de informação possa afetar a cultura organizacional, isto é, os hábitos dos empregados da empresa. Isso quer dizer que muitas vezes o dinheiro gasto no desenvolvimento de um sistema pode ser desperdiçado caso os usuários se recusem a utilizá-lo. Para aferir essa possibilidade, há o que se chama de viabilidade organizacional.

A análise de viabilidade organizacional pretende avaliar se os benefícios virão realmente, caso o sistema seja implantado. É possível encontrar a viabilidade organizacional descobrindo se os benefícios do sistema estão alinhados com os objetivos empresariais naquele momento. Também é possível consultar um grupo de beneficiados ou afetados pelo sistema, bem como de seus possíveis usuários.

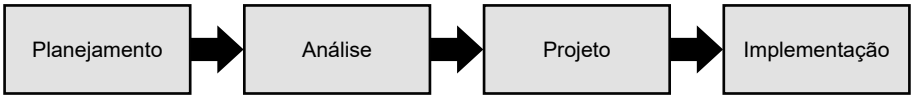
O estudo da viabilidade do projeto é de suma importância para o desenvolvimento dele. Por meio dele, é possível fazer investimentos mais seguros e proteger os profissionais de tecnologia da informação de críticas desnecessárias. É preciso enfatizar a necessidade de revisar diversas vezes o documento da análise de viabilidade.

Modernamente, a gerência de projetos considera a gerência de portfólio uma área especial. Um portfólio consiste na reunião de todos os projetos de uma organização. A gerência de portfólio considera questões como risco, importância, custo e propósito do projeto. Assim, é possível manejar os projetos, de forma que os mais urgentes sejam apressados, e os mais caros, financiados. Obviamente, a menor parte desses projetos consiste em projetos de tecnologia da informação, apesar de essa quantidade vir crescendo significativamente.

1.2.2. Metodologia de Desenvolvimento

Além da prioridade, a fase de planejamento discute qual metodologia utilizar para a construção do sistema.

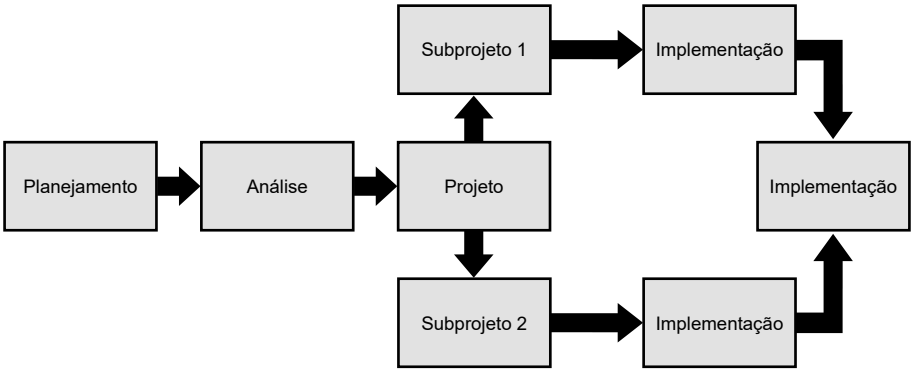
Durante muito tempo, nos primórdios da engenharia de software, era empregada a metodologia em cascata. Isso significa que as partes que constituem o projeto, planejamento, análise, projeto e implementação, são realizadas sucessivamente e uma fase só poderá ser iniciada após o término de outra.



Metodologia de desenvolvimento em cascata.

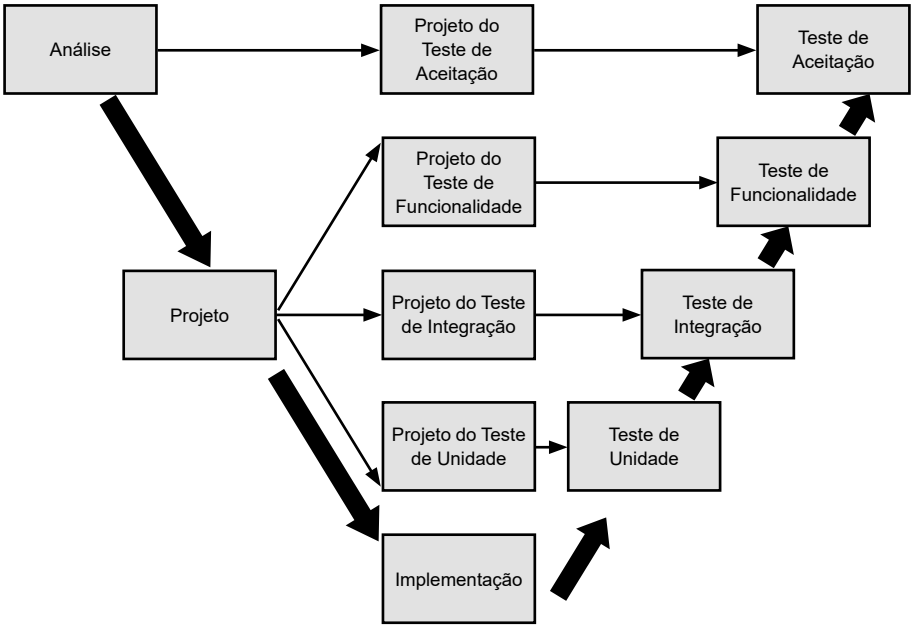
Em geral, cada entrega só era realizada após a finalização da etapa e apresentada ao comitê para aprovação. O grande problema consiste no fato de que a fase de planejamento poderia demorar meses e, caso fosse rejeitada pelo comitê de aprovação, deveria ser reconstruída desde o início. Além disso, como a fase de implementação demorava para ocorrer, muitas vezes os requisitos mudavam ou perdiam o valor que tinham inicialmente. Como solução, algumas variantes do modelo em cascata foram criadas para minimizar tais problemas.

A primeira tentativa foi a criação do modelo de desenvolvimento paralelo em que vários subprojetos eram criados a partir do planejamento e análise iniciais, permitindo terminar o projeto mais rapidamente. Contudo, como os subprojetos não eram independentes por completo, muitas decisões acarretavam consequências em outros subprojetos. Além disso, persistia o problema de que as fases iniciais eram muito longas e entregavam produtos muito volumosos de uma única vez.



Desenvolvimento paralelo.

Também há a metodologia de desenvolvimento em V. O nome advém do fato de o fluxograma aparentar o desenho de um V. O desenvolvimento em V é fundamentado no desenvolvimento em cascata, porém, com modificações. Nesse tipo de metodologia, os testes começam a ser planejados logo após a análise e não há a fase de planejamento. O fato de os testes serem projetados precocemente permite a melhoria significativa da qualidade do produto final, contudo ele ainda padece da rigidez imposta pelo desenvolvimento em cascata, do qual deriva.



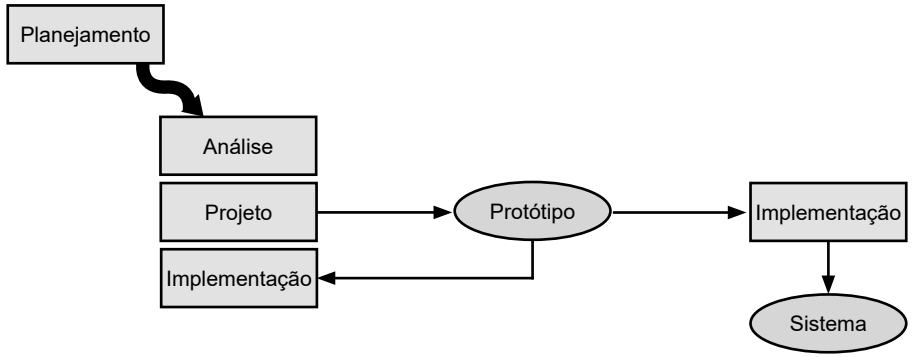
Desenvolvimento em V.

A rigidez da metodologia em cascata não é compatível com o ambiente organizacional moderno, em que as mudanças ocorrem abruptamente. Por isso, tentou-se desenvolver outras metodologias que pudessem acompanhar a evolução rápida dos negócios. Assim, foi criada a metodologia de desenvolvimento rápido de aplicações, que consiste basicamente em utilizar ferramentas computacionais para auxiliar todas as fases da produção do sistema. Essa abordagem permitiu o aumento da velocidade de produção bem como um melhor entendimento do produto pelo usuário durante a fase de produção. A melhoria obtida pela proposta foi tamanha, que hoje não se é mais possível projetar um sistema sem o auxílio dessas ferramentas.

O desenvolvimento iterativo segue a essa proposta. Ele consiste em dividir o sistema em pequenas versões que são entregues paulatinamente aos usuários. Contudo, cada versão é produzida mediante metodologia em cascata, o que ainda dificulta a flexibilidade. Apesar disso, o desenvolvimento iterativo pode oferecer uma versão do software ao cliente rapidamente, ainda que incompleta. As versões posteriores se encarregarão de concluir o restante do projeto.

Ainda entre as metodologias de desenvolvimento rápido, além do desenvolvimento iterativo, há o que se chama de prototipação, o ato de oferecer ao usuário um protótipo simplificado do sistema, o qual oferece algumas funcionalidades básicas. Por meio da reação do usuário, os desenvolvedores reanalisam, reprojetam e reimplementam o sistema, de forma a solidificar as características aceitáveis e modificar as inaceitáveis.

Essas etapas persistem até a aceitação total do protótipo, que por fim é instalado no ambiente. A grande vantagem da prototipação consiste no fato de facilitar o entendimento dos requisitos pelo analista de sistemas. Muitas vezes, o usuário não consegue esclarecer com precisão suas necessidades, então os desenvolvedores oferecem um sistema provisório para verificar se ele atende às necessidades do usuário.



Prototipação.

Apesar dos esforços e avanços reconhecidos à metodologia de desenvolvimento rápido de aplicações, ainda havia resquícios de rigidez em tais metodologias. Esse fato justificou o aparecimento de novas metodologias de desenvolvimento, como as metodologias ágeis.

Juntamente com Fowler, diversos desenvolvedores e professores de computação escreveram o manifesto ágil, que consiste numa série de prerrogativas concernentes ao desenvolvimento e engenharia de software. Essas prerrogativas se baseavam na observação de que as metodologias de desenvolvimento eram muito burocráticas e fundamentadas excessivamente nos documentos, oferecendo, muitas vezes, um produto insatisfatório e demorado.

Todas as metodologias ágeis se fundamentam essencialmente na programação. Todos os membros da equipe devem ser de fato bons programadores. Na realidade, as metodologias ágeis exigem um conhecimento diversificado na análise, pois supõem que o mesmo programador que irá criar o código deverá compreender bem as necessidades do cliente. Além disso, as metodologias ágeis se concentram em entregar pequenas unidades do sistema, às vezes em tempo hábil inferior a uma semana, de forma que o sistema possa gerar valor para o cliente o mais rapidamente possível.

Por outro lado, essa maneira de agir só se tornou possível com o avanço tecnológico, já que entregar pequenas unidades do sistema sem que isso interferisse nas futuras entregas só foi possível mediante novas linguagens e técnicas de programação. Ademais, por estar fundamentada na produção de código, as metodologias ágeis recomendam fortemente a testagem exaustiva do sistema, que, obviamente, se trata de testes automatizados.

Como cada iteração corresponde a uma entrega, boa parte da documentação é dispensada, deixando a fase de planejamento em nível mínimo. Por isso, toda comunicação deve ocorrer preferencialmente de forma presencial. Isso possibilita o aumento do entendimento do sistema pelas partes envolvidas e agiliza a comunicação.

A comunidade de desenvolvimento suscitou, pelo menos, a criação de três metodologias ágeis distintas, mas que se fundamentam sob os mesmos princípios: **XP** (extreme programming), **DSDM** (dynamic systems development method) e **Scrum**. Aparentemente, o **XP** é o mais utilizado, seguido pelo **Scrum**.

A metodologia **XP** concentra seus esforços na satisfação do cliente, por isso o feedback é uma das ferramentas mais importantes desse método. Também prevê a mudança rápida dos requisitos, isto é, um requisito que pode ser útil hoje talvez não seja útil amanhã. Isso forçaria o desenvolvimento ágil da aplicação para que ela pudesse resolver o problema com rapidez. Para isso, é necessário que o projeto seja concebido da forma mais simples possível e que atenda estritamente às necessidades do cliente.

Da mesma forma que o projeto deve ser simples, a equipe deve ser composta por um pequeno número de pessoas, o que facilita a comunicação e a integração de pessoal. Por outro lado, os funcionários devem ser bem treinados e altamente capazes.

Um projeto desenvolvido em XP começa com a descrição de estórias. Estórias nada mais são do que a descrição de acontecimentos comerciais que serão transformados em sistemas automatizados. O ato de calcular o total de uma compra pode ser considerado uma estória, que servirá de base para codificação do sistema.

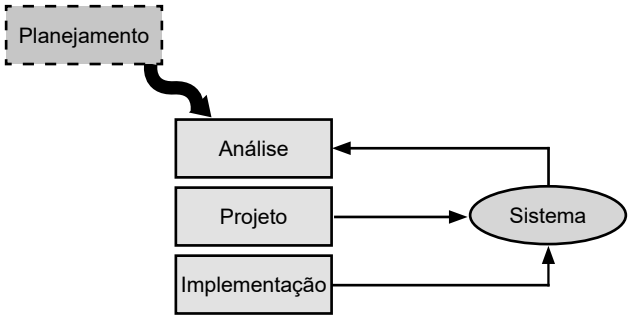
Após o recolhimento de diversas estórias, é necessário que o usuário decida qual daquelas estórias é a mais significativa, isto é, qual geraria mais valor caso fosse codificada imediatamente. Assim, criando uma hierarquia de estórias, a equipe pode codificá-las da mais importante para a menos importante e até mesmo adicionar novas estórias no decorrer do projeto, em caso de sugestão do usuário.

Como a satisfação do cliente é a meta geral em XP, isso implica que a codificação deve ser realizada lado a lado com o cliente. Isso não quer dizer que o cliente seja obrigado a permanecer disponível durante a codificação, mas preferencialmente deve estar à disposição para esclarecimento de dúvidas e avaliação de problemas.

Um dos princípios mais bem estabelecidos nesta metodologia consiste na necessidade de reduzir a documentação simbólica e a **UML**, tornando o código o mais claro e descritivo possível. O uso de comentários no código não deve ser economizado e os nomes de classes, variáveis e módulos devem ser sugestivos o suficiente para descrever suas respectivas funções.

Certamente, só é possível a realização de uma autêntica metodologia ágil com o uso de padrões de projeto, além dos padrões internos que cada corporação pode possuir. Porém, tal princípio não se trata de uma proibição. É, antes, uma sugestão de melhores práticas.

Caso haja necessidade de documentar o sistema por meio de diagramas, isso não deve ser evitado. Ademais, é possível observar que o mesmo Martin Fowler responsável pelo manifesto ágil é também responsável pelo desenvolvimento e aplicação da **UML**. Desse modo, a simplicidade é o lema, mas não uma regra rígida a ser seguida.



Metodologia de desenvolvimento ágil XP.

1.2.3. Estimativas

Uma outra questão de suma importância está relacionada à estimativa de tempo. Certamente, é impossível determinar com exatidão e precisão a duração do projeto, devido exatamente aos acontecimentos imprevisíveis, como acidentes e contratempos. Todavia, é possível estimar uma margem de duração, que servirá como base para medir o andamento do projeto. As estimativas podem ser realizadas manualmente ou por meio de softwares especializados.

O fator decisivo para o gerente de projetos consiste na sua experiência particular, que indicará, de acordo com a equipe de que dispõe, a duração do projeto. É muito comum a utilização de projetos semelhantes já realizados para estimar a duração do projeto. A construção de um prédio de doze andares com vinte e quatro apartamentos pode se basear na construção de um outro prédio construído nas mesmas condições para estimar a duração da obra, apesar dos imprevistos que possam ocorrer e que são específicos de cada projeto.

É possível, também, calcular a duração do projeto de acordo com a duração da fase de planejamento. Estima-se que o tamanho da fase de planejamento seja proporcional ao tamanho do projeto e que o tempo gasto para planejar é em torno de 15% do total. Já a fase de análise, tipicamente, gasta cerca de 20% da duração total do projeto, enquanto a fase de projeto gasta aproximadamente 35%. A fase de implementação costuma demorar cerca de 30% da duração total de um projeto. Com esses dados em mãos, é possível estimar que, caso a fase de planejamento dure em torno de quize dias, o projeto total deverá durar aproximadamente cem dias.

Também há a estimativa por pontos de função. A primeira etapa nesse modo de estimar consiste em determinar valores para cada função do negócio, de acordo com a complexidade. Pensando em metodologia XP, cada estória pode ter um valor atribuído de acordo com a sua complexidade. Os valores atribuídos servirão como fundamento para equações que definirão a duração do projeto. Ademais, pode-se medir o andamento do projeto de acordo com pontos realizados a cada iteração ou outra fração de tempo.

Apesar de sua importância, a estimativa é seguida por outra importantíssima etapa do planejamento, a escrita do plano de trabalho, e este consiste em uma espécie de calendário que identifica tarefas e estabelece as datas para a realização dessas tarefas, acompanhando sua realização. Atualmente, existem inúmeros softwares tanto gratuitos quanto pagos que facilitam a gerência do projeto.

Para gerenciar o projeto podem ser utilizadas as práticas mais comuns. Para tal, é muito conhecido o **PMBOK**, um conjunto de práticas mais aceitas entre os gerentes de projeto americanos. O PMBOK divide o projeto em nove áreas: gerência de integração, gerência de escopo, gerência de tempo, gerência de custos, gerência de qualidade, gerência de pessoal, gerência de comunicação, gerência de risco e gerência de aquisição. A quinta edição do **PMBOK** adicionou a gerência de stakeholders. A gerência de projeto será cuidadosamente analisada mais adiante neste livro. Por enquanto, basta saber da sua existência e necessidade para controlar o projeto e obter bons resultados.

1.3. Análise

A questão da análise envolve um assunto de suma importância: requisitos de software. Os requisitos são asserções do que o sistema deve realizar ou que características deve possuir. Existem os requisitos de negócio e os requisitos de sistema. Os requisitos de negócio são aquelas funcionalidades necessárias ao usuário. Já os requisitos de sistema são aqueles necessários à construção do sistema em si. Calcular o total da compra é um requisito de negócio, um hardware é um requisito do sistema. Apesar disso, há uma linha tênue entre requisitos de negócio e requisitos de sistema, afinal cadastrar usuário é um requisito de sistema ou requisito de negócio?

Os requisitos podem também ser classificados em funcionais e não funcionais. Os requisitos funcionais são aqueles que definem as informações que o sistema deve possuir ou os procedimentos que deve realizar, como salvar informações em banco de dados. Ao contrário, os requisitos não funcionais são requisitos genéricos que definem as características que um sistema deve possuir, como um determinado custo e confiabilidade.

Recolher os requisitos não é tarefa fácil. O analista deve ser minucioso e detalhista e, com muita paciência, compreender as necessidades do cliente. Os requisitos são enumerados em uma espécie de documento, uma lista comumente chamada definição de requisitos. A definição de requisitos deve conter uma seção para cada tipo de requisito, como requisitos funcionais e requisitos de negócio e pode ser utilizada legalmente, a medida que for aprovada ou rejeitada pele comitê avaliador.

Para determinar os requisitos, é preciso que o profissional de tecnologia trabalhe conjuntamente com o profissional da área de negócios. O grande problema resultado dessa união é que nem o analista de sistemas entende perfeitamente de administração empresarial e nem o usuário entende bem a tecnologia que poderá ser empregada. É exatamente como o momento em que uma distribuidora deve decidir qual caminhão comprar.

Embora não entendam sobre a tecnologia, sabem que precisam transportar uma certa quantidade de carga. Isso ficaria ainda mais difícil se o caminhão fosse construído conforme as especificações do cliente, como ocorre com o software. Questões como qual tipo de engrenagem utilizar e qual tamanho do motor são de total desconhecimento do usuário final da mesma maneira que as partes do sistema o seriam para o usuário. Apesar disso, com o aumento da informática na vida pessoal, muitas questões foram realmente facilitadas para o analista.

Quando se fala em sistemas de informação, na realidade se quer dizer sistemas automatizados de informação. Os sistemas de informação, de certa forma, já existem nas empresas operando manualmente. O que se deseja é a automação desses sistemas. Passar um banco de dados de um arquivo em papel para um banco de dados informatizado é um tipo comum de evento empresarial. Portanto, o que o analista de sistemas vem propor é uma mudança.

Tais mudanças podem ser radicais ou suaves e por isso são classificadas em três grupos. Automação de processos de negócios são aqueles procedimentos que envolvem pequenas mudanças no sistema vigente. Já a melhoria dos processos de negócio pretende uma modificação maior. Ao contrário, a reengenharia de processos fomenta uma modificação substancial nos sistemas vigentes. A questão de classificar a análise depende do estado em que se encontra o sistema no momento da análise.

Idealmente, o processo de análise percorre três etapas. A primeira delas consiste em avaliar o estado atual do sistema. Logo após, o analista deve identificar as melhorias necessárias e possíveis para aquele sistema no momento. E, finalmente, é preciso definir os requisitos do sistema. Contudo, a primeira etapa pode ser evitada, caso não haja nenhum sistema a ser avaliado. Além do mais, as metodologias ágeis tendem a gastar muito pouco tempo avaliando o estado do sistema, dando maior ênfase às outras etapas. Já a definição dos requisitos deverá se valer de alguma das técnicas mencionadas há pouco (automação de processos de negócio, melhoria dos processos de negócio ou reengenharia dos processos de negócio).

A automação de processos de negócio ocasiona os menores impactos e tenta conservar a maneira como os processos são realizados na organização. Por esse motivo, embora haja um aumento da eficiência nos processos, ela tende a gerar um menor valor para o usuário. Para determinar as modificações, a automação de processos vale-se de duas atividades distintas: análise do problema e a análise da causa raiz.

A análise de problemas consiste basicamente em perguntar aos usuários e gerentes os problemas que eles pretendem resolver com o sistema. Já a análise da causa raiz consiste em focar no entendimento da natureza do problema e assim prover soluções mais adequadas. É mais indicada quando o usuário oferece soluções imprecisas aos problemas encontrados na análise de problemas.

Já a melhoria de processos de negócio provê não só mais eficiência como mais eficácia aos processos organizacionais. Isso quer dizer que a geração de valor é maior nesse tipo de abordagem, apesar de as mudanças serem um pouco mais severas que as mudanças propostas pela automação de processos de negócio. Ao contrário da automação de processos, a melhoria dos processos de negócio tende a dar menos ênfase na compreensão do sistema atual e mais ênfase nas melhorias que podem ser realizadas. Para isso, fundamenta-se em três atividades básicas: análise de duração, custeio de atividades e aferição informal ou benchmarking informal.

A análise de duração consiste simplesmente em descobrir o tempo utilizado para realizar certo processo de negócio. Com essa informação, é possível prever quanto tempo poderá ser economizado com a automação do processo. Em contrapartida, o custeio de atividades visa descobrir o quanto se gasta financeiramente para realizar certa atividade do negócio.

Isso permite avaliar o quanto se economizará após a implementação do sistema, avaliando o quão vantajosa pode ser essa implementação. Já a aferição informal tenta avaliar os processos de negócio de forma a entender como eles poderiam ser melhorados. Chama-se **informal**, pois muitas vezes o analista visita o empreendimento como consumidor e verifica se há algum processo a ser melhorado. Essa melhoria poderá ser por automação.

Por fim, há a reengenharia de processos, que gera o maior valor por meio das modificações mais drásticas nos processos organizacionais. Ela se fundamenta em três atividades: análise de resultados, análise de tecnologia e eliminação de atividades.

A verificação dos valores oferecidos ao cliente chama-se de análise de resultados. Muitas vezes, o valor oferecido ao cliente é insuficiente para a sua satisfação e, assim, enxerga-se a possibilidade de melhorias. Já a análise de tecnologia trata-se de listar as tecnologias disponíveis para então escolher aquela que mais se adapta ao empreendimento e que geraria mais valor. Também, os gerentes e analistas podem

trabalhar em conjunto procurando atividades desnecessárias em algum processo empresarial. Após enumerá-las, elas são eliminadas dos processos, por isso tal procedimento recebe o nome de eliminação de atividades.

Além de classificar o tipo de mudança que ocorrerá, o analista deve se valer de técnicas adequadas para colher as informações necessárias e formar a definição de requisitos. O analista de sistemas pode realizar entrevistas, enviar questionários, analisar documentos e até mesmo observar as atividades na empresa. Todas as técnicas em conjunto oferecerão uma boa visão das necessidades do usuário. O analista de sistemas também pode se valer da técnica de **JAD**, joint aplication development, que é similar à prototipagem de software e pretende reunir os analistas com os usuários por vários dias até gerar um protótipo aceitável.

É preciso lembrar que a coleta de requisitos ocorrerá durante toda a fase de análise, podendo surgir requisitos imprevistos. Por outro lado, necessita-se ter em mente o escopo do sistema para que requisitos sem conformidade com o projeto não sejam submetidos à definição de requisitos. Além do mais, a definição de requisitos é uma lista direta e objetiva contendo os requisitos, conforme definição anterior, ao contrário de uma subjetiva lista de desejos do cliente que pode retratrar as mais diversas necessidades.

1.3.1. Análise dos Requisitos do Sistema

Já foi definido o que é análise. Definiu-se, inclusive, que a análise tem um resultado. Esse resultado chamamos de requisitos do sistema, isto é, o conjunto de características que o software deverá possuir após seu desenvolvimento. Em outras palavras, o requerimento é uma proposição do que o sistema deve realizar.

Há diversas formas de classificar os requisitos do sistema. Uma delas consiste em dividir os **requisitos de negócio** e **requisitos de sistema**. Os primeiros são as características que o sistema deve possuir para atender às expectativas do cliente. Já os segundos são as características que o sistema deve possuir para funcionar corretamente, do ponto de vista da tecnologia empregada.

Também podemos classificar os requisitos em **requisitos funcionais** e **requisitos não funcionais**. Os requisitos funcionais são os requisitos que o sistema exige para seu correto funcionamento. Os requisitos não funcionais são aqueles que se referem ao comportamento do sistema. Por exemplo, dados pessoais dos usuários são os requisitos funcionais que todo sistema bancário deve possuir. Velocidade na transação e facilidade de manuseio consistem em requisitos não funcionais.

Em geral, os requisitos funcionais são processos que o sistema deve executar ou informações que ele deve conter para funcionar adequadamente. Já os requisitos não funcionais podem ser requisitos operacionais, de performance, segurança ou culturais.

Para calcular o salário final de um colaborador, um sistema precisa conter o total de horas trabalhadas, que são requisitos funcionais. O próprio cálculo é um requisito funcional, um processo que o sistema deve executar. O sistema operacional e o hardware em que opera o sistema são o que chamamos de requisitos operacionais, e o imposto que deve ser descontado do salário bruto é um requisito cultural, pois pode variar de país para país. Importante também ter em conta a velocidade do cálculo, que é um requisito de performance.