

Engenharia de Software III

Introdução ao Processo Unificado de Gerenciamento de Software

O **processo unificado** (*Unified Process - UP*) de desenvolvimento de software é o conjunto de atividades necessárias para transformar requisitos do usuário em um sistema de software. O UP de desenvolvimento de sistemas combina os ciclos [iterativo e incremental](#) para a construção de [softwares](#). É fundamental na visão de que o avanço de um projeto deve estar baseado na construção de artefatos de [software](#), e não apenas em documentação.

O Processo Unificado (PU) surgiu para realizar o desenvolvimento de software visando a construção de sistemas orientados a objetos. Este modelo de desenvolvimento de software é iterativo e adaptativo, desta forma consegue produzir um sistema de grande porte como se fossem vários pequenos sistemas, o que diminui o risco do projeto.

Características

Iterativo e Incremental

Fases do Processo Unificado(PU)

O Processo Unificado é um processo [Iterativo e Incremental](#). As fases de Elaboração, Construção e Transição são divididas em uma série de interações. (A fase de Iniciação também pode ser dividida em iterações para grandes projetos). Cada iteração resulta em um *incremento*, que é uma versão do sistema que contém funcionalidades adicionais ou melhoradas em comparação com a versão anterior.

Dirigido por Casos de Uso

No Processo Unificado, [Casos de Uso](#) são usados para capturar requisitos funcionais e refinar o conteúdo das iterações. Cada iteração tem um conjunto de casos de uso ou cenários de requisitos durante todo o tempo de implementação, teste e desenvolvimento.

Centrado na Arquitetura

O Processo Unificado insiste que a [Arquitetura](#) deve estar no centro dos esforços da equipe do projeto, para dar forma ao sistema. Uma vez que não existe um modelo único suficiente para cobrir todos os aspectos do sistema, o Processo Unificado suporta múltiplas visões e modelos arquiteturais.

Uma das entregas mais importantes do processo é a arquitetura executável, que é criada durante a fase de Elaboração. Esta implementação parcial do sistema serve para validar a arquitetura e atuar como uma base para o desenvolvimento do restante.

Focado no Risco

O Processo Unificado requer que a equipe do projeto concentre-se em enfrentar os [Riscos](#) mais críticos no início do ciclo de vida do projeto. As entregas de cada iteração, especialmente na fase de Elaboração, devem ser selecionadas de forma a garantir que os maiores riscos sejam tratados em primeiro lugar.

Funcionamento

Ele é baseado em [componentes](#), o que significa o [sistema](#) ser construído a partir de componentes de [software](#) interconectados via interfaces muito bem definidas. O processo unificado utiliza a *Linguagem de Modelagem Unificada (Unified Modeling Language – UML)* no preparo de todos os artefatos do sistema.

Contexto

- Necessidade de software cada vez mais complexo:
Cliente sempre quer mais, melhor e mais rápido.
- Não é suficiente apenas a presença de desenvolvedores altamente treinados :
Precisamos de um guia organizacional : Um processo !
- Os métodos não evoluíram a contento
É necessário um processo que integre as muitas facetas do desenvolvimento.
- Solução apresentada: UP (*Unified Process*)

Histórico

- O PU tem suas raízes no trabalho feito por Ivar Jacobson na década de 60 na Ericsson;
- 1987 Jacobson deixou a empresa Ericsson, em que trabalhava, iniciando uma companhia chamada Objectory AB; Desenvolveram o Objectory, que era semelhante em estrutura com (Processo e Protudo) ao que hoje é o RUP; Seu livro Object-Oriented Software Engineering foi um marco na comunidade OO; Alguns anos apos a Rational comprou a Objectory AB;
- Em 94 foi construido o Processo Objectory da Rational (ROP) em paralelo com o Método Unificado, que depois foi chamado de UML;
- 1995 – a Rational comprou a Objectory AB, aperfeiçoou o Objectory e foi criado o Processo Objectory da Rational (ROP) (Jacobson, Rumbaugh e Booch)
- Em 98 a Rational mudou o nome do produto-processo para RUP. Apesar do PU utilizar a UML para atividades de modelagem, eles são intrinsecamente separados. A UML é independente de processo. Qualquer que seja o processo usado no desenvolvimento de um projeto, a UML poderá ser utilizada para registrar as decisões de análise e de projeto resultantes

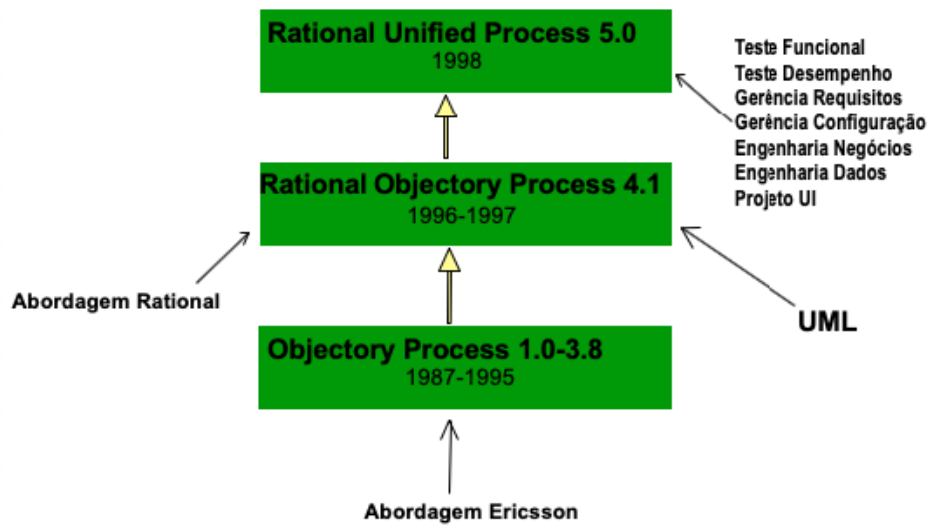
Definição Conceitual do Processo Unificado

- O processo unificado encaixa-se na definição de processo: Um conjunto de atividades executadas para transformar um conjunto de requisitos do cliente em um sistema de software.
- O PU também é uma estrutura genérica de processo que pode ser customizado adicionando-se ou removendo-se atividades com base nas necessidades específicas e nos recursos disponíveis para o projeto.

P4 = Pessoa, Projeto, Produto, Processo

- UP é um *framework* genérico de um processo de desenvolvimento
- UP é baseado em componentes
- UP utiliza toda a definição da UML
- UP é dirigido pelos *use cases*, centrado na arquitetura, iterativo e incremental (conceitos-chave)

Histórico UP

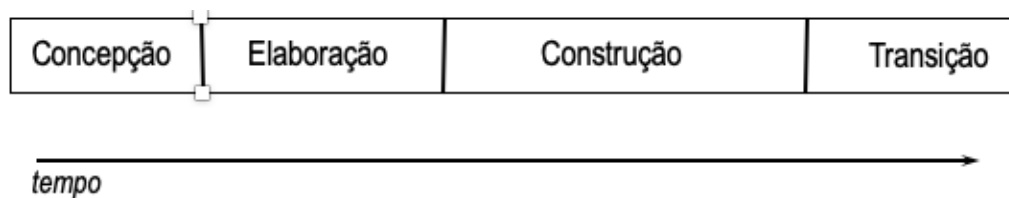


RUP

- O RUP, abreviação de Rational Unified Process (ou Processo Unificado da Rational), é um processo proprietário de Engenharia de software criado pela Rational Software Corporation, adquirida pela IBM, ganhando um novo nome IRUP que agora é uma abreviação de IBM Rational Unified Processes tornando-se uma brand na área de Software, fornecendo técnicas a serem seguidas pelos membros da equipe de desenvolvimento de software com o objetivo de aumentar a sua produtividade no processo de desenvolvimento.
- O RUP usa a abordagem da orientação a objetos em sua concepção e é projetado e documentado utilizando a notação UML (Unified Modeling Language) para ilustrar os processos em ação. Utiliza técnicas e práticas aprovadas comercialmente.
- É um processo considerado pesado e preferencialmente aplicável a grandes equipes de desenvolvimento e a grandes projetos, porém o fato de ser amplamente customizável torna possível que seja adaptado para projetos de qualquer escala. Para a gerência do projeto, o RUP provê uma solução disciplinada de como assinalar tarefas e responsabilidades dentro de uma organização de desenvolvimento de software.
- O RUP é, por si só, um produto de software. É modular e automatizado, e toda a sua metodologia é apoiada por diversas ferramentas de desenvolvimento integradas e vendidas pela IBM através de seus "Rational Suites".
- Métodos concorrentes no campo da engenharia de software incluem o "Cleanroom" (considerado pesado) e os Métodos Ágeis (leves) como a Programação Extrema (XP-Extreme Programming), Scrum, FDD e outros.

Ciclo de Vida do RUP

UP repete vários ciclos até a aposentadoria do sistema
cada ciclo gera um produto liberado para uso.
Cada ciclo possui 4 fases:



O RUP é descrito a partir de três perspectivas. Na perspectiva dinâmica, o **RUP identifica o ciclo de desenvolvimento do projeto em quatro fases sequenciais sendo**, cada fase, finalizada por um marco principal. As fases do RUP são iniciação, elaboração, construção e transição.

A Figura 1 apresenta as fases do RUP.

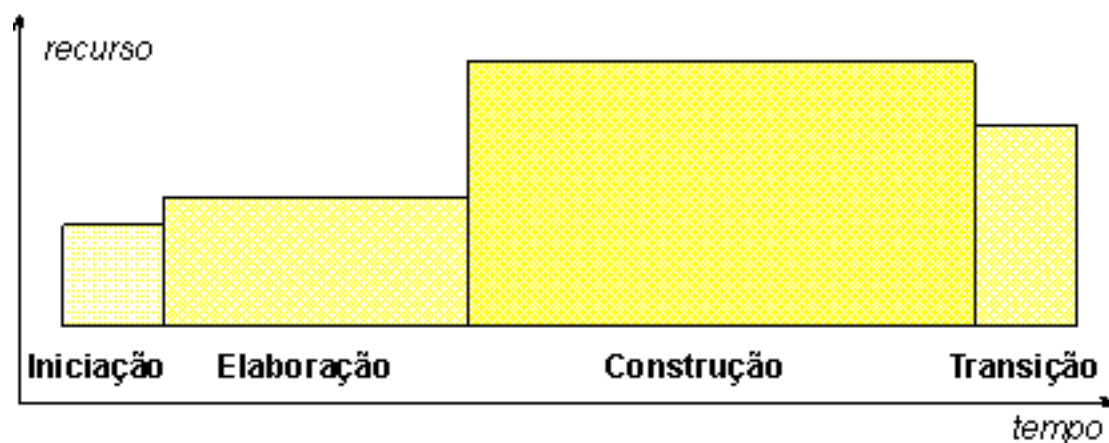
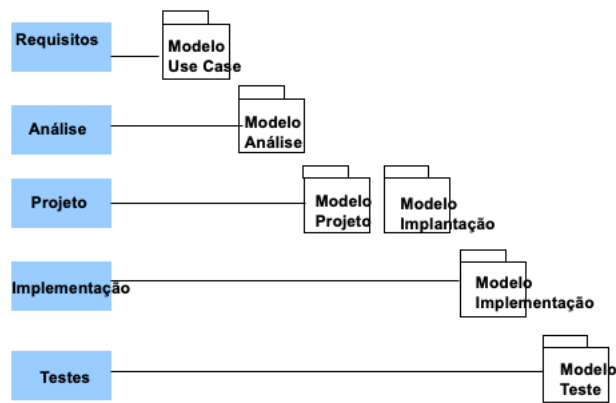


Figura 1. Fases do RUP



Fase de Concepção

A fase de iniciação ou concepção contém os workflows necessários à concordância dos stakeholders - as partes interessadas - com os objetivos, a arquitetura e o planejamento do projeto. Se essas partes interessadas tiverem bons conhecimentos, pouca análise será requerida. Caso contrário, será exigida uma análise mais elaborada. Nesta fase, os requisitos essenciais do sistema são transformados em casos de uso. O objetivo não é fechá-los em sua totalidade, mas apenas aqueles necessários à formação de opinião. A etapa é geralmente curta e serve para definir se é viável continuar com o projeto e definir os riscos e o custo deste último. Um protótipo pode ser feito para que o cliente possa aprovar. Como cita o RUP, o ideal é que sejam feitas iterações, mas estas devem ser bem definidas quanto à sua quantidade e aos objetivos.

1. Concepção: o objetivo desta fase é estabelecer um *business case*[2] para o sistema. Devem ser identificadas todas as entidades externas (pessoas e sistemas) que irão interagir com o sistema em desenvolvimento e definir essas interações. Essas informações são utilizadas para avaliar a contribuição do novo sistema para o negócio.

Os principais artefatos produzidos durante fase de Concepção estão descritos a seguir. No entanto, estes artefatos podem ser detalhados e atualizados durante as fases e iterações seguintes do projeto:

- Documento de visão contendo requisitos principais do projeto, funcionalidades chave e principais restrições;
- Um “Business Case” inicial que descreve o contexto de negócio no qual o projeto está inserido, os critérios de sucesso do projeto e previsões de investimento e retorno de investimento;
- Lista de riscos inicial do projeto;
- Plano de desenvolvimento de software;

5. Iterações na Fase de Concepção Tradicionalmente os projetos possuem apenas uma iteração na fase de Concepção. No entanto, em casos onde o projeto envolve o desenvolvimento de um novo produto ou a criação de uma nova tecnologia, mais de uma iteração pode ser necessária nesta fase para melhor definir o escopo do projeto, seus riscos e benefícios. A extensão da fase de concepção também pode ser prudente nos casos onde os riscos e os investimentos são altos ou onde não haja domínio do problema ou ainda se o time é inexperiente. Os planos desenvolvidos nas primeiras fases do projeto são refinados e tornam-se mais precisos a cada iteração. A escolha do subconjunto do sistema a ser tratado a cada iteração e qual a profundidade a ser considerada é crítica para a redução dos riscos nas iterações subseqüentes. Há duas estratégias básicas a serem consideradas:

5.1.Estratégia Wide/Shallow Nesta estratégia, o domínio de problema inteiro é analisado, mas apenas os detalhes superficiais são considerados. Todos os casos de uso são definidos e os principais são detalhados de forma a obter um entendimento claro do problema a ser tratado. A arquitetura também é superficialmente definida, bem como as interfaces do sistema. Detalhes internos são definidos apenas quando há grande risco ou incerteza. Este tipo de estratégia é apropriado quando:

- O time é inexperiente ou não há domínio do problema a ser resolvido ou da tecnologia a ser usada, incluindo a metodologia ou processo;
- A arquitetura é um ponto crítico e não existem precedentes. No entanto a equipe deve estar atenta para não cair na armadilha da paralisia da análise, procurando detalhar cada vez mais o escopo e não conseguindo finalizar a iteração, além de não conseguir mostrar resultados cedo, o que pode trazer descrédito para a equipe e o projeto;

5.2.Estratégia Narrow/Deep Na estratégia Narrow/Deep um pedaço do domínio do problema é analisado detalhadamente. Os casos de uso relacionados a este “pedaço” são definidos e detalhados profundamente. A arquitetura requerida para suportar o comportamento desejado é definida, o sistema é desenhado e implementado. Iterações subseqüentes focam na análise, desenho e implementação de novas partes do problema. Este tipo de estratégia é apropriado quando:

- Resultados precisam aparecer cedo no projeto para mitigar riscos, obter suporte ou provar viabilidade;
- Os requisitos estão evoluindo continuamente, tornando difícil definir completamente todos os requisitos antes de iniciar o trabalho de projeto e implementação;
- Um alto grau de re-uso é possível, permitindo um maior grau de entrega incremental. No entanto, deve-se ter cuidado para que os “pedaços” construídos a cada iteração sejam compatíveis e fáceis de integrar e que não haja um alto índice de retrabalho.

Geralmente, nas iterações iniciais há uma tendência para a adoção da estratégia Wide/Shallow enquanto que nas iterações posteriores adota-se, mais comumente, a estratégia Narrow/Deep. Na fase de Concepção a estratégia Narrow/Deep deve ser usada quando há a necessidade de exploração de uma nova tecnologia que é fundamental para a viabilidade do projeto. Neste caso, um protótipo do tipo “proof-of-concept” é desenvolvido para pesquisa da tecnologia e não será aproveitado no futuro. Já a estratégia Wide/Shallow deve ser usada quando se deseja obter um entendimento do escopo do sistema, e para exemplificar as funcionalidades do sistema de forma a garantir que a arquitetura seja capaz de fornecer as habilidades necessárias.

Com a descrição geral feita, precisamos identificar os requisitos do sistema, tanto funcionais como não-funcionais. A diferença básica entre esses dois grandes grupos de requisitos é que nos funcionais, definimos O QUE o sistema DEVE fazer e nos não-funcionais, COMO o sistema DEVE fazer.

Alguns artefatos opcionais que podem ser fundamentais, dependendo do sistema, são:

Glossário: Definição de termos utilizados para evitar conflitos entre aquilo que o analista quis dizer e o que o cliente realmente entendeu;

Análise de riscos: Esse é um artefato muito importante para identificar possíveis riscos que possam acontecer durante o projeto. Devem ser levados em consideração, a possível causa, a probabilidade e o impacto. O PMBoK possui uma seção que trata somente sobre esse assunto, ele divide o gerenciamento dos riscos em seis processos: o planejamento da gestão de riscos, a identificação, a análise quantitativa e qualitativa, o planejamento de resposta aos riscos identificados e o monitoramento e controle;

Protótipo e provas: Pode haver a necessidade de elaborar alguns protótipos de telas para esclarecer requisitos ainda pendentes. Lembrando que esse procedimento deve ser realizado se tiver tempo disponível e se for realmente necessário.

O levantamento de requisitos nesse momento deve ser focado apenas com os requisitos que compõem os principais processos da empresa. Os requisitos funcionais podem ser classificados em evidentes, aqueles que o usuário interage diretamente, e ocultos, em que não existe o conhecimento explícito do usuário.

Provavelmente os requisitos funcionais terão alguma particularidade que deve ser identificada nos requisitos não-funcionais, que são restrições do sistema. Esses requisitos podem ser classificados por obrigatoriedade, sendo obrigatório ou desejado, e por permanência, sendo permanente ou transitório.

Os requisitos não-funcionais ainda podem ser classificados por atributos:

Usabilidade: Consistência na interface, ajudas, estéticas, materiais para treinamento, acessibilidade, fatores humanos, etc;

Confiabilidade: Tratamento de falhas, possibilidade de previsão, não erros de programação;

Desempenho: Velocidade, eficiência, precisão, tempo de recuperação, tempo de resposta, uso de recurso, etc;

Configurabilidade: O que pode ser configurado pelos usuários do sistema;

Suportabilidade: Extensibilidade, adaptabilidade, compatibilidade, instalação, regionalização;

Interface: Arquitetura da informação;

Segurança: Permissões de usuários do sistema;

Implementação: Linguagem, banco de dados, componentes, sistemas operacionais, etc;

Empacotamento: Forma de entrega do sistema;

Físicos: Material, peso, tamanho;

Legais: Assessoria jurídica, cumprimento de leis;

Com os principais requisitos identificados, precisamos agora organizá-los. É nesse momento que os casos de uso de alto nível são descobertos, aqui devemos identificá-los atribuindo a eles os atores envolvidos e suas referências com os requisitos.

As referências cruzadas são usadas para garantir a rastreabilidade dos requisitos, tornando muito importante quando nos deparamos com a mudança dos requisitos, uma situação muito comum na maioria dos projetos. Através desse procedimento temos como saber o impacto que aquela mudança vai ocasionar, e criar um plano para tratar essa situação.

A organização dos requisitos também deve ser feita através das funções de conceitos, nesse caso pode ser necessário elaborar um modelo conceitual preliminar, para associar os conceitos.

Outra organização que deve ser feita é a das consultas e relatórios, que, na maioria dos casos, são as mais importantes, mas também as mais simples. Aqui devemos listar as consultas mais complexas e suas referências cruzadas.

É bom lembrar que o processo unificado trabalha com ciclos iterativos, a partir da fase de elaboração, por isso devemos planejar esses ciclos já na fase de concepção. Esse planejamento contém alguns itens importantes como, a estimativa de esforço, que pode ser feita utilizando técnicas de análise por pontos de função (APF) ou análise por casos de uso (UCP).

No planejamento também devemos ter em mente os casos de uso mais importantes, para priorizar seu desenvolvimento no ciclo iterativo. Podemos levar em consideração

dois pontos importantes, o grau de complexidade e o grau de importância dos casos de uso.

Craig Larman, o pai da AOO (Análise Orientada a Objetos), criou algumas perguntas para ajudar no escalonamento dos casos de uso, são elas:

Representa impacto direto na receita?

Processo primário na linha de negócio?

Requer suporte à persistência e provoca impacto no projeto de arquitetura?

Oferecem rapidamente quantidade significativa nas informações?

Funções de alto risco, críticas com relação ao tempo ou complexas?

Conforme as respostas das perguntas, temos como escalonar os casos de uso e priorizar o desenvolvimento.

Além disso, não podemos deixar de analisar os recursos disponíveis para o projeto, por isso é necessário a criação de um cronograma de execução e custos, que deve conter a duração dos ciclos iterativos. É recomendado que nesse momento sejam utilizadas técnicas de gerenciamento de tempo previstas no PMBoK.

Disciplinas do RUP

Há 9 disciplinas seguidas pelo RUP:

- Disciplinas de processo
 - Modelagem de Negócios;
 - Requisitos;
 - Análise e Design;
 - Implementação;
 - Teste;
 - Implantação;
- Disciplinas de suporte
 - Gerenciamento de Configuração e Mudanças;
 - Gerenciamento de Projetos;
 - Ambiente.

As atividades que ocorrem durante o processo de desenvolvimento são chamadas de *workflows*. Existem seis *workflows* principais, exibidos na Tabela 1.

Workflow	Descrição
Modelagem de Negócios	Os processos de negócio são modelados usando casos de uso de negócios.
Requisitos	Os agentes que interagem com o sistema são identificados e os casos de uso são desenvolvidos para modelar os requisitos do sistema.

Análise e Projeto	Um modelo de projeto é criado e documentado usando modelos de arquitetura, modelos de componente, modelos de objetos e modelos de sequencia.
Implementação	Os componentes de sistema são implementados e estruturados em subsistemas de implementação. A geração automática de código com base os modelos de projeto ajuda a acelerar esse processo.
Teste	O teste é um processo iterativo realizado em conjunto com a implementação. O teste de sistema segue o término da implementação.
Implantação	Uma versão do produto é criada, distribuída aos usuários e instalada no local de trabalho.
Gerenciamento de Configuração e Mudança	Este workflow de apoio gerencia mudanças no sistema.
Gerenciamento de Projetos	Este workflow de apoio gerencia o desenvolvimento do sistema.
Ambiente	Este workflow está relacionado à disponibilização de ferramentas apropriadas de software para a equipe de desenvolvimento.

Tabela 1 : Workflows no Rational Unified Process (Sommerville 2007, pág. 55)

Embora o RUP não seja um processo adequado a todos os tipos de [desenvolvimento de software](#), ele representa uma nova geração de processos genéricos. A mais importante inovação é a separação de fases e workflows, e o reconhecimento de que a implantação de software no ambiente do usuário é parte do processo. As fases são dinâmicas e tem objetivos. Os [workflows](#) são estáticos e constituem atividades técnicas que não estão associadas a uma única fase, mas podem ser utilizados ao longo do desenvolvimento para atingir os objetivos de cada fase (Sommerville 2007, pág. 56).

A iniciação, também conhecida como concepção, **é a fase em que se estabelece o escopo do projeto de software**, levantando-se uma visão geral do produto final. A iniciação tem como objetivo estabelecer o escopo, detalhar os casos de uso crítico do *software*, estimar o custo, os riscos e preparar o ambiente para o projeto. Se o projeto for de pouca importância ou inviável, ele pode ser cancelado após essa fase.

Na elaboração, identificam-se os casos de uso principais e elaboram-se o sistema em iterações. A elaboração tem como objetivo garantir que a arquitetura, os requisitos e os planos estejam estáveis, que os riscos identificados sejam reduzidos, permitindo a criação de protótipos e estabelecer um ambiente de suporte.

A construção é a etapa do desenvolvimento do *software*. Nela, **produzem-se o código fonte do produto na linguagem de programação escolhida pela equipe do projeto**. Os objetivos principais da construção compreendem reduzir os custos da implementação, obter a qualidade, concluir a análise, o design, a implementação e os testes das

funcionalidades necessárias, desenvolver o produto de *software*, bem como, verificar se as funcionalidades foram finalizadas e se os usuários estão prontos para receber o sistema em ambiente de produção.

A transição é a fase final do RUP. Nela, ocorre a validação e a entrega definitiva do *software* para o cliente. Essa etapa normalmente inclui a entrega do sistema, teste beta para validação das funcionalidades, conversão de bancos de dados operacionais, treinamento com os usuários, ajustes no sistema e a avaliação do produto.

Como Adotar

Com base nestes recursos a adoção do RUP pode ser feita de mais de uma maneira. Um extremo seria usar o RUP à risca, ou seja, aplicar todos os métodos e processos exatamente como são propostos. A vantagem desta abordagem é que nada deve ser alterado, pois o RUP é bem completo e detalhado. Porém existe um preço a ser pago, pois o RUP na íntegra é complexo. Esta abordagem implicaria em treinamentos, projetos piloto, etc. Propostas de projetos de adoção do RUP são descritos no próprio produto.

O extremo oposto seria adotar outro modelo de processo mais simples ou conhecido (o atual, se existir) e utilizar o material do RUP como fonte de referência complementar para assuntos não abordados em outro modelo como, por exemplo, os modelos de documentos.

A primeira abordagem é interessante para empresas que precisam de uma grande formalização do processo de desenvolvimento de software e cujo método atual seja totalmente inadequado ou inexistente. A segunda abordagem seria interessante para quem já tem alguma metodologia que considera adequada, mas que tem deficiência em alguma área como, por exemplo, suporte a UML. Soluções intermediárias também são possíveis.

Linhas mestras

O RUP define as seguintes linhas-mestras e esqueletos (templates) para os membros da equipe de um ciclo de produção: parte do cliente, e uma avaliação do progresso do projeto pela sua gerência. Além disso, ajuda os programadores a manterem-se concentrados no projeto.

Gestão de requisitos

Uma documentação apropriada é essencial para qualquer grande projeto; note que o RUP descreve como documentar a funcionalidade, restrições de sistema, restrições de projeto e requisitos de negócio.

Os casos de uso (em inglês Use Cases) e os cenários são exemplos de artefatos dependentes do processo, que têm sido considerados muito mais eficazes na captura de requisitos funcionais.

Uso de arquitetura baseada em componentes

A arquitetura baseada em componentes cria um sistema que pode ser facilmente extensível, promovendo a reutilização de software e um entendimento intuitivo. Um componente normalmente se relaciona com um objeto na programação orientada a objetos.

O RUP oferece uma forma sistemática para construir este tipo de sistema, focando-se em produzir uma arquitetura executável nas fases iniciais do projeto, ou seja, antes de comprometer recursos em larga escala.

Estes componentes são normalmente incluídos em infraestruturas existentes como o CORBA e o COM (Modelo de Componentes de Objetos).

Uso de software de modelos visuais

Ao abstrair a programação do seu código e representá-la utilizando blocos de construção gráfica, o RUP consegue uma maneira efetiva de se ter uma visão geral de uma solução.

O uso de modelos visuais também pode permitir que indivíduos de perfil menos técnico (como clientes) tenham um melhor entendimento de um dado problema, e assim se envolvam mais no projeto como um todo.

A linguagem de modelagem UML tornou-se um padrão industrial para representar projetos, e é amplamente utilizada pelo RUP.

Verificação da qualidade do software

Não assegurar a qualidade do software é a falha mais comum em todos os projetos de sistemas computacionais. Normalmente pensa-se em qualidade de software após o término dos projetos, ou a qualidade é responsabilidade de uma equipe diferente da equipe de desenvolvimento.

O RUP visa auxiliar no controle do planejamento da qualidade, verificando-a na construção de todo o processo e envolvendo todos os membros da equipe de desenvolvimento.

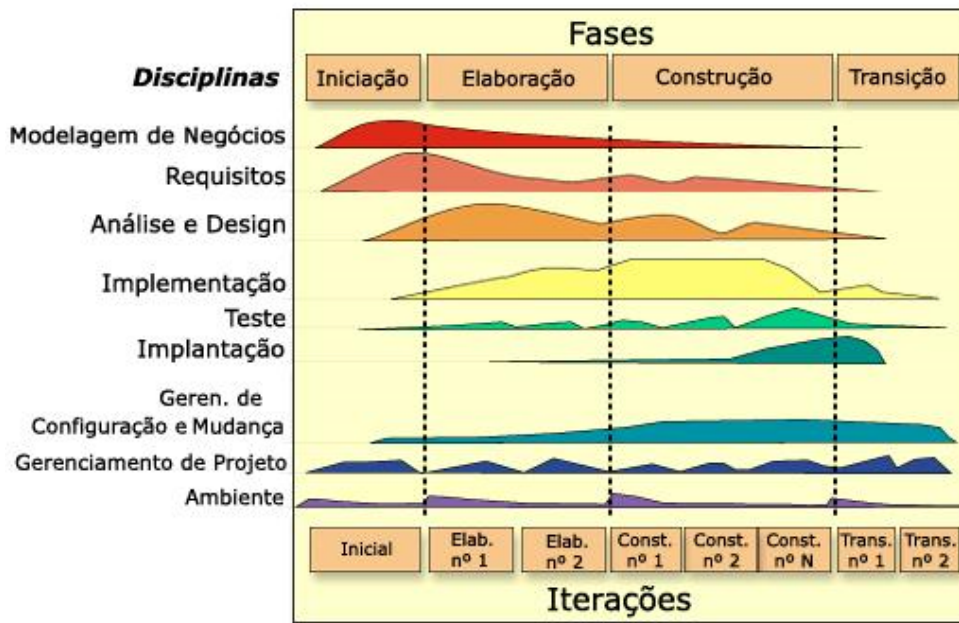
Gestão e Controle de Mudanças do Software

Em todos os projetos de software a existência de mudanças é inevitável. O RUP define métodos para controlar e monitorar mudanças. Como uma pequena mudança pode afetar aplicações de formas inteiramente imprevisíveis, o controle de mudanças é essencial para o sucesso de um projeto.

O RUP também define áreas de trabalho seguras, garantindo a um programador que as mudanças efetuadas em outro sistema não afetarão o seu sistema.

A perspectiva estática do RUP enfoca as atividades que acontecem no processo de desenvolvimento. Elas são denominadas *workflows* na descrição do *framework*. O RUP

oferece seis *workflows* de processos principais e três *workflows* de apoio principais. Os *workflows* estão representados na Figura 2.



Fases

Até agora estas linhas de guia são gerais, a serem aderidas ao percorrer do ciclo de vida de um projeto. As fases (vide figura abaixo) indicam a ênfase que é dada no projeto em um dado instante. Para capturar a dimensão do tempo de um projeto, o RUP divide o projeto em quatro fases diferentes:

1. **Iniciação ou Concepção:** ênfase no escopo do sistema;
2. **Elaboração:** ênfase na arquitetura;
3. **Construção:** ênfase no desenvolvimento;
4. **Transição:** ênfase na implantação.

O RUP também se baseia nos 4 Ps:

1. Pessoas
2. Projeto
3. Produto
4. Processo

As fases são compostas de iterações. As iterações são janelas de tempo; as iterações possuem prazo definido enquanto as fases são objetivas.

Todas as fases geram artefatos. Estes serão utilizados nas próximas fases e documentam o projeto, além de permitir melhor acompanhamento.

Fase de Concepção

A fase de iniciação ou concepção contém os *workflows* necessários à concordância dos *stakeholders* - as partes interessadas - com os objetivos, a arquitetura e o planejamento do projeto. Se essas partes

interessadas tiverem bons conhecimentos, pouca análise será requerida. Caso contrário, será exigida uma análise mais elaborada. Nesta fase, os [requisitos essenciais do sistema](#) são transformados em [casos de uso](#). O objetivo não é fechá-los em sua totalidade, mas apenas aqueles necessários à formação de opinião. A etapa é geralmente curta e serve para definir se é viável continuar com o projeto e definir os riscos e o custo deste último. Um [protótipo](#) pode ser feito para que o cliente possa aprovar. Como cita o RUP, o ideal é que sejam feitas [iterações](#), mas estas devem ser bem definidas quanto à sua quantidade e aos objetivos.

Fase de Elaboração

A fase de elaboração será apenas para o projeto do sistema, buscando complementar o levantamento / documentação dos casos de uso, voltado para a arquitetura do sistema, revisa a modelagem do negócio para os projetos e inicia a versão do manual do usuário. Deve-se aceitar: Visão geral do produto (incremento + integração) está estável?; O plano do projeto é confiável?; Custos são admissíveis?

Fase de Construção

Na fase de construção, começa o desenvolvimento físico do software, produção de códigos, testes alfa. Os testes beta são realizados no início da fase de Transição.

Deve-se aceitar testes, e processos de testes estáveis, e se os códigos do sistema constituem "baseline".

Fase de Transição

Nesta fase ocorre a entrega ("deployment") do software, é realizado o plano de implantação e entrega, acompanhamento e qualidade do software. Produtos (releases, versões) devem ser entregues, e ocorrer a satisfação do cliente. Nesta fase também é realizada a capacitação dos usuários.

Disciplinas

Seis Disciplinas de Engenharia de Software

Disciplina de Modelagem de Negócios

As organizações estão cada vez mais dependentes de sistemas de [TI](#), tornando-se imperativo que os engenheiros de [sistemas de informação](#) saibam como as aplicações em desenvolvimento se inserem na organização. As empresas investem em TI, quando entendem a vantagem competitiva do valor acrescentado pela tecnologia. O objetivo de [modelagem de negócios](#) é, primeiramente, estabelecer uma melhor compreensão e canal de comunicação entre [engenharia de negócios](#) e [engenharia de software](#). Compreender o negócio significa que os engenheiros de software devem compreender a estrutura e a dinâmica da empresa alvo (o cliente), os atuais problemas na organização e possíveis melhorias. Eles também devem garantir um entendimento comum da organização-alvo entre os clientes, usuários finais e desenvolvedores.

Modelagem de negócios, explica como descrever uma visão da organização na qual o sistema será implantado e como usar esta visão como uma base para descrever o processo, papéis e responsabilidades.

Disciplina de Requisitos

Esta disciplina explica como levantar pedidos das partes interessadas ("stakeholders") e transformá-los em um conjunto de [requisitos](#) que os produtos funcionam no âmbito do sistema a ser construído e fornecem requisitos detalhados para o que deve fazer o sistema.

Disciplina de Análise e Projeto ("Design")

O objetivo da análise e projeto é mostrar como o sistema vai ser realizado. O objetivo é construir um sistema que:

- Execute, em um ambiente de execução específico, as tarefas e funções especificadas nas descrições de [casos de uso](#)
- Cumpra todas as suas necessidades
- Seja fácil de manter quando ocorrerem mudanças de [requisitos funcionais](#)

Resultados de projeto em um modelo de análise e projeto tem, opcionalmente, um modelo de análise. O modelo de design serve como uma abstração do código-fonte, isto é, o projeto atua como um modelo de "gabarito" de como o código-fonte é estruturado e escrito. O modelo de projeto consiste em classes de design estruturado em pacotes e subsistemas com interfaces bem definidas, representando o que irá se tornar componentes da aplicação. Ele também contém descrições de como os objetos dessas classes colaboram para desempenhar casos de uso do projeto.

Disciplina de Implementação

Os efeitos da implementação são:

- Para definir a organização do código, em termos de subsistemas de implementação organizadas em camadas
- Para implementar classes e objetos em termos de componentes (arquivos-fonte, binários, executáveis e outros)
- Para testar os componentes desenvolvidos como unidades
- Integrar os resultados produzidos por implementadores individuais (ou equipes), em um sistema executável

Sistemas são realizados através da aplicação de componentes. O processo descreve como reutilizar componentes existentes ou implementar novos componentes com responsabilidades bem definidas, tornando o sistema mais fácil de manter e aumentar as possibilidades de reutilização.

Disciplina de Teste

As finalidades da disciplina de teste são:

- Verificar a interação entre objetos
- Verificar a integração adequada de todos os componentes do software
- Verificar se todos os requisitos foram corretamente implementados
- Identificar e garantir que os defeitos são abordados antes da implantação do software
- Garantir que todos os defeitos são corrigidos, reanalisados e fechados

O Rational Unified Process propõe uma abordagem iterativa, o que significa que deve-se testar todo o projeto. Isto permite encontrar defeitos tão cedo quanto possível, o que reduz radicalmente o custo de reparar o defeito. Os testes são realizados ao longo de quatro dimensões da qualidade: *confiabilidade, funcionalidade, desempenho da aplicação, e o desempenho do sistema*. Para cada uma destas dimensões da qualidade, o processo descreve como passar pelo teste do ciclo de planejamento, projeto, implementação, execução e avaliação.

Disciplina de Implantação

O objetivo da implantação é o de produzir com sucesso lançamentos de produtos e entregar o software para seus usuários finais. Ele cobre uma vasta gama de atividades, incluindo a produção de releases externos do software, a embalagem do software e aplicativos de negócios, distribuição do software, instalação do software e prestação de ajuda e assistência aos usuários. Embora as atividades de implantação estejam principalmente centradas em torno da fase de transição, muitas das atividades devem ser incluídas nas fases anteriores para se preparar para a implantação, no final da fase de construção. Os processos ("workflows") de "Implantação e Ambiente" do RUP contêm menos detalhes do que outros workflows.

Três Disciplinas de Apoio/Suporte

Disciplina de Ambiente

O ambiente enfoca as atividades necessárias para configurar o processo para um projeto. Ele descreve as atividades necessárias para desenvolver as diretrizes de apoio a um projeto. A proposta das atividades de ambiente é prover à organização de desenvolvimento de software os processos e as ferramentas que darão suporte à equipe de desenvolvimento. Se os usuários do RUP não entendem que o RUP é um framework de processo, eles podem percebê-lo como um processo pesado e caro. No entanto, um conceito-chave dentro do RUP foi que o processo RUP pode e, muitas vezes, deve ser refinado. Este foi inicialmente feito manualmente, ou seja, por escrito, um documento de "caso de desenvolvimento" que especificou o processo refinado para ser utilizado. Posteriormente, o produto IBM Rational Method Composer foi criado para ajudar a tornar esta etapa mais simples, [engenheiros de processos](#) e [gerentes de projeto](#) poderiam mais facilmente personalizar o RUP para suas necessidades de projeto. Muitas das variações posteriores do RUP, incluindo OpenUP/Basic, a versão open-source e leve do RUP, são agora apresentados como processos distintos, por direito próprio, e atendem a diferentes tipos e tamanhos de projetos, tendências e tecnologias de desenvolvimento de software. Historicamente, como o RUP, muitas vezes é personalizado para cada projeto por um perito do processo RUP, o sucesso total do projeto pode ser um pouco dependente da capacidade desta pessoa.

Disciplina de Configuração e Gerência de Mudança

A disciplina de [Gestão de Mudança](#) em negócios com RUP abrange três gerenciamentos específicos: de configuração, de solicitações de mudança, e de status e medição.

- Gerenciamento de configuração: A gestão de configuração é responsável pela estruturação sistemática dos produtos. Artefatos, como documentos e modelos, precisam estar sob controle de versão e essas alterações devem ser visíveis. Ele também mantém o controle de dependências entre artefatos para que todos os artigos relacionados sejam atualizados quando são feitas alterações
- Gerenciamento de solicitações de mudança: Durante o processo de desenvolvimento de sistemas com muitos artefatos existem diversas versões. O [CRM](#) mantém o controle das propostas de mudança
- Gerenciamento de status e medição: Os pedidos de mudança têm os estados: *novo*, *conectado*, *aprovado*, *cedido* e *completo*. A solicitação de mudança também tem atributos como a causa raiz, ou a natureza (como o defeito e valorização), prioridade, etc. Esses estados e atributos são armazenados no banco de dados para produzir relatórios úteis sobre o andamento do projeto. A Rational também tem um produto para manter as solicitações de mudança chamado [ClearQuest](#). Esta atividade tem procedimentos a serem seguidos

Disciplina de Gerência de Projeto

O [planejamento de projeto](#) no RUP ocorre em dois níveis. Há uma baixa granularidade ou planos de **Fase** que descreve todo o projeto, e uma série de alta granularidade ou planos de **Iteração** que descrevem os passos iterativos. Esta disciplina concentra-se principalmente sobre os aspectos importantes de um processo de desenvolvimento iterativo: Gestão de riscos; Planejamento de um projeto iterativo através do ciclo de vida e para uma iteração particular; E o processo de acompanhamento de um projeto iterativo, métricas. No entanto, esta disciplina do RUP não tenta cobrir todos os aspectos do gerenciamento de projetos.

Por exemplo, não abrange questões como:

- Gestão de Pessoas: contratação, treinamento, etc
- Orçamento Geral: definição, alocação, etc
- Gestão de Contratos: com fornecedores, clientes, etc

Workflows

Os workflows são definidos da seguinte maneira:

- **Modelagem de negócios:** os processos de negócios são modelados com a utilização de casos de uso de negócio;
- **Requisitos:** os casos de usos são criados para modelar os requisitos do *software*;
- **Análise e projeto:** cria-se um modelo de projeto com base em modelos de arquitetura, de componente, de objeto e de sequência;
- **Implementação:** os componentes do software são implementados;
- **Teste:** o teste é um processo iterativo e é efetuado em conjunto com a implementação do sistema;
- **Implantação:** cria-se uma versão do produto e instala-a no local de trabalho dos usuários;
- **Gerenciamento de configuração e mudanças:** esse *workflow* serve como apoio à gerência do projeto em relação às mudanças no sistema;
- **Gerenciamento de projetos:** esse *workflow* de apoio gerencia o processo de desenvolvimento do *software*;
- **Ambiente:** este *workflow* relaciona-se à disponibilização de ferramentas de *software* adequadas para a equipe de desenvolvimento.

A perspectiva prática do RUP enfoca as **boas práticas de engenharia de software** recomendadas para o desenvolvimento de sistemas. São recomendadas seis melhores práticas:

1. Desenvolver o sistema iterativamente;
2. Gerenciar requisitos;
3. Arquiteturas baseadas em componentes;
4. Modelar o *software* visualmente;
5. Verificar a qualidade do *software*;
6. Controlar as mudanças do *software*.

Conceitos Relacionados

Pessoas (Users):

Worker: papel representado por uma pessoa ou grupo no processo de software.
Cada *worker* é responsável por um conjunto de atividades.

- Projeto:
Possui uma sequência de mudanças / várias iterações / padrão organizacional
- Produto:
Não é apenas código.
Artefato: qualquer tipo de informação criada.
Artefatos são criados pelos *workers* em cada uma de suas atividades.
- Processo:
Direciona o projeto.
Template para criação de instâncias (projetos).

Processo Dirigido pelos Use Cases

- **Benefícios:** *use cases* associam todos os *workflows* de forma conjunta.
- Dirigem várias atividades de desenvolvimento:
 - Criação e validação da arquitetura do sistema
 - Criação de casos de teste
 - Planejamento das iterações
 - Criação de documentação do usuário
 - Implantação do sistema
 - Sincronizam conteúdo dos modelos criados em cada *workflow*.

Processo Centrado na Arquitetura

- **Benefícios:**
 - Fornece uma base sólida para a construção do software
 - Melhor compreensão do sistema e organização do desenvolvimento.
- Descrição arquitetura envolve elementos de modelo mais importantes - coleção de visões dos modelos do sistema.
- UP prescreve um refinamento sucessivo à arquitetura.
- A arquitetura representa a forma, enquanto que os *use cases* representam funcionalidade.
- Arquitetura e *use cases* devem ser balanceados

Processo Iterativo e Incremental

- **Benefícios:**
 - Identificação de riscos é adiantada
 - Preparação do Sistema para requisitos que mudam
 - Integração contínua (facilita testes) e aprendizado facilitado
- Iteração : mini-projeto - transversal pelos *workflows*
Modelos evoluem nas iterações.
- Resultado de uma iteração : incremento.

Existem outros processos além do PU?

- IBM/Rational (Baseado no PU) • RUP Rational Unified Process
- Scott W. Ambler (Baseados no PU) • AUP Agile Unified Process
- EUP Enterprise Unified Process

Modelos de Processo de Software

Desenvolver software é geralmente uma tarefa complexa e sujeita a erros

Sucesso ou fracasso dependem de inúmeros fatores que ocorrem durante todo o processo

Necessidade de estabelecer processos sistemáticos para desenvolvimento → Modelos de processo de Software

- quem (papel) está fazendo
- o quê (artefato),
- como (atividade) e
- quando (disciplina).

Artefatos

Porção significativa de informação interna ou a ser fornecida a interessados externos que desempenhe um papel no desenvolvimento do sistema.

Um artefato é algum documento, relatório, modelo ou código que é produzido, manipulado ou consumido.

Exemplos: modelo de caso de uso, modelo do projeto, um caso de uso, um subsistema, um caso de negócio, um documento de arquitetura de software, código fonte, executáveis, etc.

Descreve as sequências das atividades que produzem algum resultado significativo e mostra as interações entre os participantes

São realizadas a qualquer momento durante o ciclo de desenvolvimento (Fases do PU)

Requisitos, Análise, Projeto, Implementação e Teste

Principais artefatos da fase de concepção:

1. Reunião de KickOff

<https://escritoriodeprojetos.com.br/kick-off-meeting>

2. Termo de abertura do projeto

<https://escritoriodeprojetos.com.br/termo-de-abertura-do-projeto>

3. Briefing

<https://desenvolvimentoparaweb.com/miscelanea/briefing-para-desenvolvimento-de-web-sites-consideracoes-dicas-e-modelos/>

4. Especificação de trabalho do projeto

<https://escritoriodeprojetos.com.br/especificacao-do-trabalho-do-projeto>

5. Business Case

<https://escritoriodeprojetos.com.br/business-case>

Técnicas de entrevistas e de coleta de dados para projetos de desenvolvimento de software

Diretrizes para as entrevistas que você fará durante a fase de análise do projeto de desenvolvimento de um sistema. Provavelmente entrevistará usuários, gerentes, auditores, programadores e auxiliares que utilizam sistemas já existentes (informatizados ou manuais) e também várias outras pessoas envolvidas.

Por que fazemos entrevistas durante a análise do projeto?

- Precisamos coletar informações sobre o comportamento de um sistema atual ou sobre os requisitos de um novo sistema de pessoas que têm essas informações armazenadas em algum lugar em suas cabeças.
- Precisamos verificar nossa própria compreensão, como analistas de sistemas, do comportamento de um sistema atual ou dos requisitos de um novo sistema. Essa compreensão deve ser adquirida através de entrevistas em combinação com informações coletadas de modo independente.
- Precisamos coletar informações sobre o(s) sistema(s) atual(is) para a execução do estudo de custo/benefício para o novo sistema.

A aplicação da entrevista no levantamento de requisitos de software deve seguir as seguintes etapas:

1. Selecione os usuários chaves.
2. Agende uma entrevista com cada um deles.
3. Defina o assunto da entrevista, lembre-se que o entrevistado é quem deve expor suas ideias, a entrevista deve ser focada e assuntos dispersivos devem ser evitados.
4. Planeje a entrevista focando sempre o assunto delineado. Questione sobre o ambiente de trabalho do entrevistado. Esse ambiente irá definir uma fração do sistema. Formas de interlocução em uma entrevista: explique o relacionamento entre o que está em discussão e as demais áreas dos sistemas; você consegue descrever o ponto de vista dos outros usuários a respeito do que está sendo discutido? o item em discussão depende da existência de alguma outra coisa?
5. Utilize ferramentas para automatizar a entrevista e a coleta de dados.
6. Documente as respostas dos usuários.
7. Valide a documentação, apresente-a aos entrevistados e colha sugestões de melhoria.
8. Evite termos técnicos da área de informática.

Tipos de Entrevistas

A forma mais comum de entrevista é uma reunião pessoal e direta entre você (talvez acompanhado por até dois analistas auxiliares do projeto) e um ou mais interlocutores (entrevistados). Normalmente são efetuadas anotações por um dos entrevistadores; menos costumeiramente, a entrevista pode ser gravada ou um(a) secretário(a) tomará notas durante a entrevista.

Pode-se perceber que as informações obtidas em uma entrevista também podem ser obtidas por outros meios, por exemplo, solicitando-se que os entrevistados respondam por escrito a

um questionário formal, ou solicitando que descrevam por escrito os requisitos do novo sistema. Também podemos aumentar as entrevistas pela presença de vários especialistas (que podem até conduzir a entrevista enquanto o analista de sistemas participa como assistente), como peritos em indústria/comércio, psicólogos comportamentais e negociadores. E, finalizando, deve-se lembrar que outros meios de obtenção de dados (ex.: videocassete, gravadores, etc...) podem ser utilizados para registrar uma entrevista.

Durante a década de 80, uma forma especializada de entrevista tornou-se popular em algumas empresas; conhecida como JAD (Joint Application Development) ou projeto acelerado, ou análise em equipe, e por vários outros nomes. Ela consiste em uma rápida entrevista e um processo acelerado de coleta de dados em que todos os principais usuários e o pessoal da análise de sistemas agrupam-se em uma única e intensiva reunião (que pode prolongar-se de um dia a uma semana) para documentar os requisitos do usuário. A reunião costuma ser supervisionada por um profissional experiente e bem treinado que atua como mediador para encorajar melhores comunicações entre os analistas de sistemas e os usuários. Frequentemente, este supervisor é apoiado por algumas pessoas que se dedicam integralmente ao processo.

Embora todas essas variações tenham de fato sido utilizadas, elas são relativamente raras e não serão discutidas em maiores detalhes neste texto. A entrevista mais utilizada ainda é a reunião pessoal entre um analista de sistemas e um usuário final.

Problemas Fundamentais

Inicialmente pode parecer que o processo de entrevistar um usuário seja uma questão simples e direta. Afinal, o analista e o usuário são pessoas inteligentes e capazes de expressarem. Os dois são pessoas racionais e ambos têm o mesmo objetivo: passar informações relativas a um novo sistema proposto da mente do usuário para a do analista. Qual é o problema?

Na realidade, existem muitos problemas que podem ocorrer. Em muitos projetos de alta tecnologia, tem-se observado que a maioria dos problemas difíceis não envolvem hardware nem software, mas sim o *peopleware*. Os problemas de *peopleware* na análise de sistemas são muitas vezes encontrados nas entrevistas. Os problemas mais comuns a que você deve estar atento são:

- Entrevistar a pessoa errada no momento errado. É muito fácil, por causa dos problemas e interesses organizacionais, falar com a pessoa que é o perito oficial na orientação do usuário, que demonstra nada saber a respeito dos verdadeiros requisitos do sistema; também é possível perder a oportunidade de falar com o usuário desconhecido que realmente sabe quais são os requisitos. Mesmo que encontre a pessoa certa, talvez o analista tente entrevistá-la durante um período em que o usuário não está disponível ou esteja mergulhado sob outras pressões e emergências.
- Fazer perguntas erradas e obter respostas erradas. A análise de sistemas é, como Tom DeMarco gosta de dizer, uma forma de comunicação entre estranhos. Os usuários e os analistas de sistemas têm vocabulários diferentes, experiências básicas diferentes e muitas vezes um diferente conjunto de pressuposições, percepções, valores e prioridades. Desse modo, é fácil fazer ao usuário uma pergunta racional sobre os requisitos do sistema e o usuário não entender absolutamente a pergunta, sem que nenhum dos dois perceba o fato. É fácil para o usuário prestar-lhe algumas informações sobre os requisitos e o analista não entender essas informações, novamente sem que nenhum dos dois perceba o fato. As ferramentas de modelagem são uma tentativa de fornecer uma linguagem comum e sem ambigüidades para diminuir esses mal entendidos. Porém as entrevistas costumam ser realizadas em uma língua comum (inglês, espanhol, francês, português etc.), portanto o problema é real. Isso explica

porque é tão importante marcar entrevistas subseqüentes para confirmar que ambas as partes entenderam as perguntas e as respostas.

- Criar ressentimentos recíprocos. existem algumas razões pelas quais o usuário pode não se sentir à vontade ou mesmo colocar-se em posição antagônica na entrevista com um analista de sistema (ex.: porque ele percebe que o verdadeiro objetivo do novo sistema que o analista está especificando é tomar-lhe o emprego). E o analista pode ressentir-se do modo como o usuário está respondendo as perguntas. Em qualquer caso, o ressentimento pode surgir entre as duas partes, tornando a comunicação muito mais difícil. Não existe um modo mágico de garantir que esses problemas não ocorrerão; eles são a consequência das interações pessoa-a-pessoa, e cada uma dessas interações é única. Contudo, as sugestões dadas a seguir podem ajudar a reduzir a probabilidade desses problemas; fora isso, dependerá de prática para melhorar cada vez mais em cada entrevista subseqüente.

Diretrizes Para a Realização de Entrevistas

As seguintes diretrizes podem ser de grande auxílio na direção de entrevistas bem-sucedidas com o usuário.

Desenvolva um Plano Geral de Entrevistas

Antes de tudo, é extremamente importante que o analista descubra quem deve ser entrevistado. Caso contrário, desperdiçará o tempo de todos e criará um enorme tumulto, por falar com pessoas erradas sobre coisas erradas.

Isso requer que obtenha um organograma da empresa que mostre as pessoas da organização usuária, bem como a hierarquia entre elas. Se não existir um organograma formal, encontrar alguém que saiba como a organização funciona e pedir ajuda. Se o organograma existir, certificar-se de que esteja correto e atualizado; as empresas muitas vezes modificam-se com muito mais frequência do que o ciclo editorial anual em que os diagramas são produzidos!

O fato de conhecer o esquema organizacional não diz necessariamente com quem o analista deve entender-se; às vezes, a pessoa que mais sabe a respeito de algum aspecto de um sistema é um funcionário administrativo ou burocrata que sequer aparece no organograma. Muitas vezes existem três níveis de usuários em uma organização grande e complexa (o usuário verdadeiro, o usuário supervisor operativo e o usuário supervisor executivo) e é muitas vezes de grande importância falar com todos os três níveis.

Em muitos casos também é importante conversar com os usuários na seqüência adequada e na combinação certa. Por vezes o analista poderá saber em que seqüência os usuários deverão ser entrevistados face a seu conhecimento geral da organização e por vezes os próprios usuários lhe dirão uma vez que saibam que serão entrevistados.

Certifique-se de que tem Autorização para Falar com os Usuários

Em algumas organizações informais não haverá restrições na escolha dos usuários com quem falar ou sobre como as entrevistas serão marcadas. Porém isso não é comum em empresas grandes; é politicamente perigoso vagar pela organização usuária realizando entrevistas sem prévia autorização.

Na maioria dos casos, a autorização virá ou do encarregado de um setor usuário (um departamento ou divisão) ou do representante da empresa que estará auxiliando o projeto de desenvolvimento do sistema. Em qualquer caso, os usuários têm legítimas razões em querer aprovar, antecipadamente, quem será entrevistado:

- Eles podem saber que alguns usuários não serão capazes de compreender ou exprimir bem os requisitos do sistema.
- Eles podem desconfiar de que alguns dos usuários do nível operativo sejam rebeldes que apresentarão requisitos falsos (ou requisitos que a direção não aprova).
- Eles podem recear que as entrevistas interfiram com as atribuições normais de trabalho que os usuários tenham de executar. Por causa disso, desejaram marcar as entrevistas para os momentos apropriados.
- Eles podem recear que as entrevistas sejam vistas como o início de um esforço de substituição dos usuários humanos por um sistema computadorizado, provocando demissões e coisas desse gênero.
- Eles podem considerar que eles próprios (os diretores) sabem mais a respeito dos requisitos do sistema do que qualquer outro e por isso não desejam que você entreviste qualquer usuário de nível operativo.
- Pode estar havendo uma batalha política em andamento em um nível de chefia muito mais elevado, entre o setor usuário e a organização de desenvolvimento de sistemas. Desse modo, o gerente usuário pode não ter reais objeções a suas entrevistas, porém, impedindo que elas sejam feitas, ele estará enviando uma mensagem política para o chefe do chefe do seu chefe.

Por todos esses motivos, é uma boa medida obter uma prévia autorização. Na maior parte dos casos, basta uma autorização verbal; se a organização for terrivelmente burocrática ou paranóica, pode-se precisar de uma autorização escrita. Isso, a propósito, também significa que o analista deve estar atento à política organizacional se tiver certeza da necessidade de falar com um usuário (normalmente um usuário do nível operativo) com quem tenha sido avisado para não conversar.

Planeje a Entrevista para Fazer Uso Eficiente do Tempo

O principal aspecto desta sugestão é que deve-se compreender que está tomando o tempo do usuário e que ele (ou o chefe dele) pode achar até que o analista esteja desperdiçando o tempo dele. Assim sendo, é importante o planejamento e preparação tão antecipadamente quanto possível para poder fazer uso eficiente da entrevista.

A primeira coisa a fazer é certificar-se de que o usuário conhece o assunto da entrevista. Em alguns casos isso pode ser feito por telefone; em outros, pode ser adequado preparar uma lista das perguntas que serão feitas, ou dos tópicos que serão abordados, ou dos DFD que necessita ser revisados, e remetê-la ao usuário com um dia ou dois de antecipação. Se não puder fazer isso, é um indício de que de fato o analista não está preparado para a entrevista. E se o usuário não tiver lido o material remetido, é sinal de que:

- está muito ocupado,

- está desinteressado,
- opõe-se a toda a idéia da entrevista ou
- é incapaz de entender as perguntas apresentadas.

Um aspecto relacionado: coletar, antes da entrevista, tantos dados pertinentes quanto possível. Se houver formulários ou relatórios que sejam pertinentes à discussão, geralmente poderá obtê-los antecipadamente. Se existirem outros documentos escritos do usuário descrevendo o novo ou o antigo sistema consiga-os e estude-os antes da entrevista.

Se tiver preparado as perguntas antecipadamente, o analista deve ser capaz de realizar a entrevista em uma hora ou menos. Isso é importante; não só o usuário é geralmente incapaz de reservar mais do que uma hora de cada vez, mas também as pessoas normalmente não conseguem se concentrar intencionalmente (principalmente se estiverem examinando diagramas um tanto estranhos) por mais do que uma hora. Isso naturalmente significa que deve-se organizar a entrevista para abranger um escopo relativamente limitado, focalizando normalmente uma parte do sistema. Isso também pode significar que tenha de marcar algumas entrevistas com o mesmo usuário para abranger inteiramente a área em que ele está envolvido.

Finalizando, marcar uma reunião subsequente para rever o material que foi coletado. Normalmente, após a entrevista, o analista irá para sua mesa com todas as informações colhidas na entrevista, e executará bastante trabalho com os dados brutos. Pode haver DFD a serem desenhados ou itens a serem criados no dicionário de dados; cálculos de custo/benefício podem precisar ser feitos; as informações provenientes da entrevista podem precisar ser correlacionados com dados de outras entrevistas, e assim por diante. Em qualquer caso, os dados dessa entrevista serão manipulados, documentados, analisados e convertidos em uma forma que o usuário pode nunca ter visto antes. Desse modo, pode ser necessário marcar uma entrevista posterior para verificar:

- que o analista não cometeu nenhum engano em seu entendimento do que o usuário lhe disse,
- que o usuário não mudou de opinião nesse ínterim,
- que o usuário entende a notação ou a representação gráfica dessas informações.

Utilize Ferramentas Automatizadas que Sejam Adequadas, Mas Não Abuse

Durante a entrevista, pode-se achar conveniente utilizar ferramentas de prototipação, principalmente se o objetivo da entrevista for discutir a visão que o usuário tem da interface pessoa-máquina. De modo semelhante, se estiver revisando um diagrama de fluxo de dados e discutindo possíveis modificações, poderá achar prático usar uma ferramentas CASE.

Lembre-se, que o objetivo dessas ferramentas é facilitar as discussões e não complicá-las; elas devem permitir que o analista e o usuário examinem alternativas e modificações com rapidez e facilidade; elas podem ajudar a registrar o conhecimento de um requisito do usuário e corrigir imediatamente quaisquer erros que tenha sido cometido.

Se, a tecnologia introduzir-se no assunto, deve-se deixar fora da entrevista. Se o usuário tiver de aventurar-se além de seu ambiente normal de atividade (para outro prédio, na sala do computador) poderá encarar a ferramenta como um aborrecimento. Se o usuário não estiver familiarizado com a tecnologia de computadores e for solicitado a utilizar a ferramenta, poderá

rejeitá-la. E se o analista não estiver familiarizado com a ferramenta (ou se a ferramenta for lenta, apresentar erros ou de emprego limitado) isso interferirá sensivelmente na entrevista. Em qualquer desses casos, talvez seja preferível usar a ferramenta off-line depois da entrevista; então, poderá mostrar ao usuário a saída da ferramenta sem causar quaisquer problemas desnecessários.

Tente Descobrir quais Informações o Usuário tem mais Interesse

Se o analista tiver de desenvolver um modelo completo de sistema para alguma parte de um sistema, possivelmente necessitará determinar entradas, saídas, funções, características tempo-dependentes e a memória armazenada do sistema. Porém a ordem em que se obtém essas informações costumam não ter muita importância. Mas pode significar muito para o usuário, e deve deixá-lo começar a entrevista por onde ele preferir. Alguns usuários desejam começar pelas saídas, isto é, pelos relatórios e valores de dados que eles querem que o sistema produza (na realidade, eles talvez nem saibam que entradas serão necessárias para produzir as saídas desejadas). Outros usuários poderão estar mais interessados nas entradas ou nos detalhes de uma transformação funcional. Outros ainda preferirão falar sobre os detalhes dos dados de um depósito de dados. Deve-se esforçar para enxergar os requisitos do sistema da perspectiva desses usuários, e conservar esta perspectiva em mente quando fizer as perguntas necessárias sua entrevista.

Use um Estilo Adequado de Entrevistar

Como diz William Davis [Davis, 1983]: Sua atitude em relação à entrevista é importante na determinação de seu sucesso ou fracasso. Uma entrevista não é uma competição. Evite ataques; evite o uso excessivo do jargão técnico; conduza uma entrevista, não uma tentativa de persuasão. Fale com as pessoas, não fale muito alto, nem muito baixo, nem indiretamente. Uma entrevista não é um julgamento. Faça perguntas detalhadas, mas não faça perguntas para confirmar outras respostas. Lembre-se que o entrevistado é o perito e que é você que precisa de respostas. Para concluir, de modo algum critique a credibilidade de outras pessoas.

Fazer perguntas detalhadas nem sempre é fácil; dependendo da personalidade do entrevistado e do tema da entrevista, pode-se precisar de um conjunto de estilos para extrair as informações necessárias. Alguns estilos que podem mostrar-se úteis:

- **Relacionamentos.** Pedir ao usuário para explicar o relacionamento entre o que está em discussão e as demais partes do sistema. Se o usuário estiver falando sobre um assunto (p.ex.: um cliente), pedir-lhe que explique seu relacionamento sob outros aspectos; se ele estiver descrevendo uma função (ex.: uma bolha de um DFD), pedir-lhe que explique seu relacionamento com outras funções. Isso não só o auxiliará a descobrir mais detalhes sobre o item em pauta, mas também o ajudará a descobrir interfaces (ex.: fluxos de dados de uma bolha para outra no DFD) e relacionamentos formais.
- **Pontos de vista alternativos.** Solicitar ao usuário que descreva o ponto de vista de outros usuários em relação ao item que esteja sendo discutido. Perguntar ao usuário, por exemplo, o que seu chefe pensa sobre uma bolha do DFD, ou um tipo de objeto do DER; ou pergunte o que pensa seu subordinado.
- **Detalhamento.** Solicitar ao usuário uma informal descrição narrativa do item em que estiver interessado. Fale-me sobre o modo como você calcula o valor das remessas. Ou, se estiver falando com o usuário sobre um tipo de objeto no DER, poderia dizer-lhe: Fale-me a respeito de um cliente. O que você sabe (ou precisa saber) sobre um cliente?
- **Dependências.** Perguntar ao usuário se o item em discussão depende, para sua existência, de alguma outra coisa. Isso é especialmente útil quando se discutem possíveis tipos de objetos e

relacionamentos no DER. Em um sistema de controle de pedidos, por exemplo, pode-se perguntar ao usuário se seria possível haver um pedido sem que haja um cliente.

- Confirmação. Dizer ao usuário o que acha que ouviu ele dizer; usar suas próprias palavras em lugar das dele e peça confirmação. Pode-se dizer: Deixe-me ver se entendi o que você disse...

Possíveis Formas de Resistência na Entrevista

Deve-se estar preparado para o fato de que alguns usuários serão contrários à idéia de uma entrevista; essa é uma das razões para garantir que o chefe ou alguém com autoridade no setor esteja ciente e tenha permitido a entrevista. Algumas das objeções mais comuns e algumas possíveis respostas a essas objeções:

- Você está tomando tempo demais de mim. A resposta a isso é explicar que compreende, e desculpar-se pelo tempo que precisará tomar, mas que já preparou tudo e fará a entrevista no tempo mais curto possível. Isso naturalmente exige que o analista chegue pontualmente na hora marcada para o início da entrevista, mantenha a discussão no rumo previsto, e encerre-a no momento em que tenha dito que o faria.
- Você está ameaçando meu emprego. Isso muitas vezes é uma reação emocional que pode ou não ter fundamento. Embora possa pensar em várias maneiras de responder a esse comentário, lembre-se de que o analista não é o patrão dessa pessoa e que não está em posição de garantir que o emprego dela não esteja em perigo, ou de informá-la do contrário. Ele vai considerar o analista como o perito em eficiência cuja tarefa é orientar a direção em como o emprego dele pode ser eliminado pela informatização. A solução para esse problema, caso ele ocorra, é fazer com que seja levado ao conhecimento dos níveis superiores dos usuários e obter o pronunciamento oficial, se possível, pessoalmente ou por escrito.
- Você não conhece nossa empresa, como você quer dizer-nos como deve ser o novo sistema? A resposta a essa pergunta é: Você tem razão! E por isso que o estou entrevistando para saber o que você pensa sobre quais devam ser os requisitos!. Por outro lado, deverá sugerir várias maneiras de melhorar as coisas (especialmente se parte ou todo o serviço realizado atualmente pelo usuário for em um antigo e ineficiente sistema); assim, esse tipo de comentário pode ser inevitável. Entretanto, o verdadeiro truque é continuar sendo tão respeitoso quanto possível e reconhecer constantemente a experiência do usuário na sua área, embora continuando a perguntar se ele poderia explicar-lhe porque sua idéia não funcionaria.
- Você está tentando mudar o modo como as coisas são feitas aqui. O artifício neste caso é mostrar ao usuário que embora possa estar propondo algumas (radicais) mudanças na implementação do sistema atual, não é pensamento modificar as características essenciais desse sistema, exceto nas áreas onde eles mesmos tenham solicitado uma alteração. Devemos lembrar, que algumas das características da implementação do sistema atual podem ser preservadas, por causa das interfaces do sistema atual com outros sistemas externos que exijam que as entradas ou saídas apresentem formatos prescritos.
- Não queremos esse sistema. Esta é uma variação da queixa você está querendo tirar meu emprego. A verdadeira resposta é que o analista está ali, conduzindo a entrevista, porque a direção quer o novo sistema. Não é da sua competência convencer os usuários operativos que eles devem querer o novo sistema; fazer isso é colocar o peso da responsabilidade sobre seus ombros, onde ela não deve ficar.
- Por que você está desperdiçando nosso tempo com esta entrevista? Sabemos o que queremos, e se você fosse competente, você saberia imediatamente o que queremos. Por que você não vai em frente e constrói o sistema? Esta é uma reclamação difícil de se lidar, porque relaciona-se com o fato fundamental de que os usuários e os analistas de sistemas estão falando línguas diferentes e estranhas. Lembre-se, que com a disponibilidade das linguagens de quarta geração e dos computadores pessoais, o usuário pode achar que pode construir ele próprio o sistema; sucessos fáceis com projetos simples (planilhas eletrônicas, por exemplo) podem ter-lhe dado a impressão de que todos os sistemas são fáceis de implementar. Isso pode explicar a impaciência em relação ao analista.

Outros Problemas

As diretrizes acima alertaram sobre os inúmeros problemas políticos com que o analista pode se defrontar em uma entrevista e os muitos motivos pelos quais o usuário pode mostrar-se hostil. Mas ainda existem alguns problemas para os quais deve-se estar atento:

- Uma discussão que focalize mais os problemas de implementação do que os problemas dos requisitos. Isso muitas vezes ocorre quando o usuário diz: Eis como eu gostaria que você construísse o sistema... . Isso acontece quase sempre quando o usuário está raciocinando em termos da implementação do sistema atual; e pode acontecer se o usuário conhecer alguma coisa da tecnologia de computadores (ex.: quando ele possui um PC particular ou quando ele é um ex-programador). Lembrar que não é obrigação em uma entrevista de análise descrever características de implementação do sistema a não ser que sejam tão importantes que realmente pertençam ao modelo de implementação do usuário.
- Confusão entre sintomas e problemas. Isso ocorre em muitas áreas, não apenas na área do processamento de dados. O que pode acontecer nas entrevistas de análise de sistemas. Entretanto, boa parte dele depende de onde a fronteira foi estabelecida no diagrama de contexto: se a queixa do usuário é um sintoma ou um problema depende de ela estar associada a alguma coisa dentro dos limites do sistema ou fora deles. Desse modo, deve-se dar especial atenção ao desenvolvimento do modelo ambiental.
- O usuário pode ser incapaz de explicar o que ele quer que o sistema faça ou pode mudar de opinião. Esse é um problema comum e o analista de sistemas deve estar preparado para ele. Quanto maior for esse problema mais importante torna-se a prototipação.
- Desentendimento entre usuários de mesmo nível, subordinados e chefes. Infelizmente, isso coloca o analista de sistemas no papel de mediador entre as partes em desentendimento. Como analista, não pode abdicar desse papel; não pode dizer: Quando vocês decidirem o que querem e entrarem em um acordo, procurem-me. Em vez disso, deve-se agir como um negociador conduzindo todos os interessados a uma sala e trabalhando com eles na tentativa de chegar a um consenso. Isso, infelizmente, envolve habilidades e procedimentos fora do escopo deste texto.
- Descobrir disparidades entre o padrão oficial e a prática do trabalho. Durante as entrevistas, o analista descobre que podem existir algumas disparidades entre a versão oficial de como o sistema funciona e como realmente ocorre no nível operacional. Neste caso, o analista deve utilizar a diplomacia quando relatar estas discrepâncias para a gerência, pois o pessoal de nível operacional pode relutar em admitir que as operações nem sempre seguem os padrões oficiais de trabalho.

Formas Alternativas de Coleta de Dados

As entrevistas não são o único modo de coleta de informações sobre os requisitos de um sistema. Na realidade, quanto mais informações puder colher de outras fontes, mais produtivas poderão ser as entrevistas pessoais. Alternativas para as entrevistas:

- Questionários. Pode remeter questionários escritos para os usuários dentro da organização, para as pessoas (ou setores) que interagem com o sistema, para os diretores que aprovaram o projeto e para outros.
- Demonstrações feitas pelos fornecedores. Os fornecedores de hardware e os fornecedores de software podem já haver desenvolvido sistemas prontos para a aplicação em que você esteja interessado. Solicitando-lhes uma demonstração dos recursos desses sistemas pode não somente auxiliá-lo a decidir se o produto é uma boa solução, mas também revelar funções e dados armazenados que você pode não ter percebido.
- Visitas a outras instalações. Procure outras empresas que estejam no mesmo ramo de atividades ou que tenham sistemas semelhantes aquele em que você esteja trabalhando.

Combine uma visita à instalação para obter informações diretas sobre as características e aptidões do sistema.

- Coleta de dados. Procure formulários, relatórios, manuais, procedimentos escritos, registros, imagens de tela de terminais e listagens de programas que já existam na organização usuária. Lembre-se, todavia, que esses recursos normalmente estão relacionados com a implementação atual do sistema; devemos lembrar que isto costuma incluir informações redundantes e/ou contraditórias e/ou obsoletas. Não obstante, isto muitas vezes é um bom ponto de partida para você familiarizar-se com o terreno antes de iniciar as entrevistas pessoais com o usuário.
- Pesquisa externa. Se você estiver construindo um sistema para uma nova aplicação, para a qual o usuário não dispõe de qualquer experiência para descrever os requisitos, talvez seja necessário tentar obter informações em sociedades profissionais, ou em periódicos e livros técnicos e em relatórios de pesquisas.

Questionário de Pesquisa

Essa ferramenta é muito conveniente quando há um grande número de usuários ou vários grupos de usuários em locais diferentes. Nestas situações não é prático entrevistar todas as pessoas, mas tão logo as informações obtidas através dos questionários tenham sido analisadas, podem ser realizadas entrevistas com usuários selecionados. Podem ser usados diversos formatos para o questionário de pesquisa, tais como: de múltipla escolha, lista de verificação (checklist), questões abertas (descritivas) e combinações das anteriores. Os passos para a utilização do questionário de pesquisa são:

a) Preparação do questionário:

- Identifique o tipo de informação que deseja obter, como problemas experimentados ou oportunidades a explorar;
- Definidos os requisitos, escolha um formato apropriado para o questionário;
- Monte questões de forma simples, clara e concisa;
- Se incluir questões abertas, observar o espaço necessário para a resposta;
- Agrupar as questões por tópicos ou áreas afins;
- Se possível, enviar junto com o questionário uma carta da direção da empresa

explicando os motivos e a importância da pesquisa para a organização.

b) Identificação dos respondentes

- O questionário deve ser personalizado com o nome, função e localização do respondente;
- Elaborar um controle de identificação das pessoas que receberão os questionários. Utilizar o controle para monitorar a situação dos questionários.

c) Distribuição do questionário

- Distribua o questionário junto com as instruções de preenchimento;
- Indique claramente o prazo para a devolução do questionário e a forma de devolução.

d) Análise das respostas

- Consolide e analise as informações fornecidas pelos questionários devolvidos;
- Documente as principais descobertas;
- Envie uma cópia do relatório com as principais descobertas para cada respondente

como uma cortesia pelo tempo dedicado à pesquisa.

Observações no ambiente

A análise de observação é uma técnica de observação de fatos muito efetiva. Ela pode ser usada para diversas finalidades, como processamento e confirmação de resultados de uma entrevista, identificação de documentos que devem ser coletados para análise, esclarecimento do que está sendo feito no ambiente atual.

Esta técnica é simples. Durante algum tempo, o analista fica observando os usuários em seu ambiente enquanto eles executam suas tarefas diárias. Frequentemente o analista fará perguntas para conhecer o funcionamento e as atividades desenvolvidas. É importante explicar para as pessoas que serão observadas o que será observado e porque. Os passos para a observação são:

a) Antes:

- Identifique as áreas a serem observadas;
- Obter a aprovação da gerencia apropriada para executar a observação;
- Obter os nomes e funções das pessoas-chaves que serão envolvidas no estudo de observação;
- Explicar a finalidade do estudo;

b) Durante:

- Familiarizar-se com o local de trabalho a ser observado;
- Observar os agrupamentos organizacionais atuais;
- Observar as facilidades manuais e automatizadas em uso;
- Coletar amostras de documentos e procedimentos escritos que são utilizados nos processos observados;
- Acumular informações estatísticas das tarefas observadas: frequência que ocorrem, estimativas de volumes, tempo de duração, etc...
- Enquanto interage com os usuários, tente ser objetivo e não comentar as formas de trabalho de maneira não construtiva;
- Observar também as exceções;
- Terminando as observações, agradeça às pessoas o apoio e a colaboração dispensada ao seu trabalho.

c) Após:

- Documente as descobertas resultantes das observações;
- Consolide os resultados;
- Reveja os resultados consolidados com as pessoas observadas e/ou com seus superiores.

Técnicas para levantamento de Requisitos

O início para toda a atividade de desenvolvimento de software é o **levantamento de requisitos**, sendo esta atividade repetida em todas as demais etapas da **engenharia de requisitos**.

Sommerville (2003) propõe um processo genérico de levantamento e análise que contém as seguintes atividades:

- Compreensão do domínio: Os analistas devem desenvolver sua compreensão do domínio da aplicação;
- Coleta de requisitos: É o processo de interagir com os stakeholders do sistema para descobrir seus requisitos. A compreensão do domínio se desenvolve mais durante essa atividade;
- Classificação: Essa atividade considera o conjunto não estruturado dos requisitos e os organiza em grupos coerentes;
- Resolução de conflitos: Quando múltiplos stakeholders estão envolvidos, os requisitos apresentarão conflitos. Essa atividade tem por objetivo solucionar esses conflitos;
- Definição das prioridades: Em qualquer conjunto de requisitos, alguns serão mais importantes do que outros. Esse estágio envolve interação com os stakeholders para a definição dos requisitos mais importantes;
- Verificação de requisitos: Os requisitos são verificados para descobrir se estão completos e consistentes e se estão em concordância com o que os stakeholders desejam do sistema.

O **levantamento e análise de requisitos** é um processo iterativo, com uma contínua validação de uma atividade para outra, conforme exemplificado pela **Figura 1**.

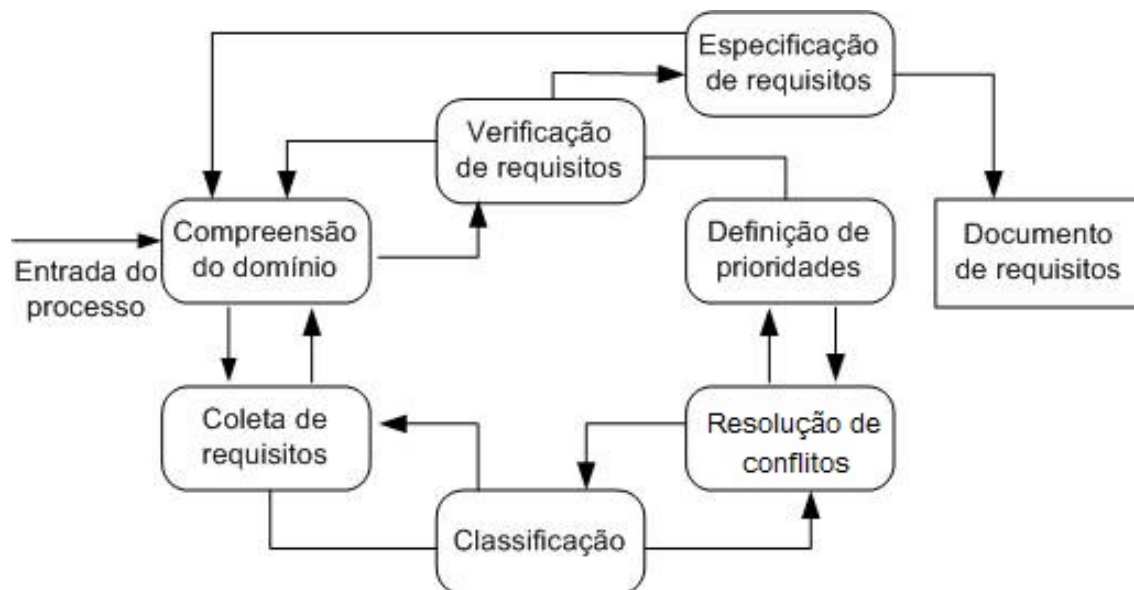


Figura 1. Processo de levantamento e análise de requisitos (SOMMERVILLE, 2003).

Dificuldades encontradas

O problema de não saber especificar corretamente o que o sistema deverá fazer é muito antigo. Pompilho (1995) cita um exemplo do relatório produzido por McKinsey, em 1968, e mencionado por B. Langefords e B. Sundgren onde se afirmava que dois terços das empresas ali estudadas estavam desapontadas com o atendimento recebido em sistemas de informação.

Após mais de 30 anos da elaboração do relatório a situação não é muito diferente. Algumas das razões para o baixo grau de satisfação dos usuários para os sistemas destacam-se:

- **Na fase de levantamento de requisitos do projeto**, onde não é utilizada uma técnica adequada para extrair os requisitos do sistema;
- A falha do analista em não descrever os requisitos do sistema de modo claro, sem ambigüidades, conciso e consistente com todos os aspectos significativos do sistema proposto.

Entre as dificuldades encontradas na fase de levantamento de requisitos estão: o usuário principal do sistema não sabe o que quer que o sistema faça ou sabe e não consegue transmitir para o analista; requisitos identificados, mas que não são realistas e não identificam os requisitos similares informados por pessoas diferentes. Um stakeholder errado afetará em perda de tempo e dinheiro para ambas as partes envolvidas no desenvolvimento do sistema.

Identifica-se um levantamento de requisitos adequado através da boa definição do projeto, da efetividade do projeto, de informações necessárias a um perfeito diagnóstico e de soluções inteligentes. **Quanto ao levantamento de requisitos inadequado, o resultado é um diagnóstico pobre com conclusões comprometidas**, não identificação das causas dos problemas, custos elevados, prazos vencidos ou comprometedores, omissão de processos fundamentais e descréditos.

Técnicas de Levantamento de Requisitos

As técnicas de levantamento de requisitos têm por objetivo superar as dificuldades relativas a esta fase. Todas as técnicas possuem um conceito próprio e suas respectivas vantagens e desvantagens, que podem ser utilizadas em conjunto pelo analista.

Serão apresentadas de forma resumida nesse artigo algumas técnicas de levantamento de requisitos.

Levantamento orientado a pontos de vista

Para qualquer sistema, de tamanho médio ou grande, normalmente há diferentes tipos de usuário final. Muitos *stakeholders* têm algum tipo de interesse nos requisitos do sistema. Por esse motivo, mesmo para um sistema relativamente simples, existem muitos pontos de vista diferentes que devem ser considerados. Os diferentes pontos de vista a respeito de um problema ‘vêm’ o problema de modos diferentes. Contudo, suas perspectivas não são inteiramente independentes, mas em geral apresentam alguma duplicidade, de modo que apresentam requisitos comuns.

As abordagens orientadas a ponto de vista, na engenharia de requisitos, reconhecem esses diferentes pontos de vista e os utilizam para estruturar e organizar o processo de levantamento e os próprios requisitos. Uma importante capacidade da análise orientada a pontos de vista é que ela reconhece a existência de várias perspectivas e oferece um *framework* para descobrir conflitos nos requisitos propostos por diferentes *stakeholders*.

O método VORD (*viewpoint-oriented requirements definition – definição de requisitos orientada a ponto de vista*) foi projetado como um framework orientado a serviço para o levantamento e análise de requisitos.

A primeira etapa da análise de ponto de vista é identificar os possíveis pontos de vista. Nessa etapa os analistas se reúnem com os *stakeholders* e utilizam a abordagem de *brainstorming* para identificar os serviços em potencial e as entidades que interagem com o sistema.

A segunda etapa é a estruturação de pontos de vista, que envolve agrupar pontos de vista relacionados, segundo uma hierarquia. Serviços comuns estão localizados nos níveis mais altos da hierarquia e herdados por pontos de vista de nível inferior.

A etapa de documentação do ponto de vista tem por objetivo refinar a descrição dos pontos de vista e serviços identificados.

O mapeamento de sistema conforme ponto de vista envolve identificar objetos em um projeto orientado a objetos, utilizando as informações de serviço que estão encapsuladas nos pontos de vista.

A **Figura 2** exemplifica a técnica de levantamento orientado a ponto de vista.

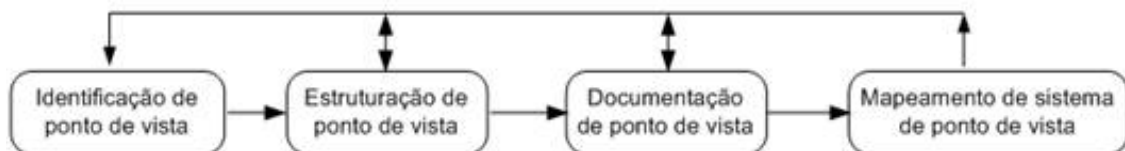


Figura 2. Método VORD, SOMMERVILLE – 2003.

Etnografia

A etnografia é uma técnica de observação que pode ser utilizada para compreender os requisitos sociais e organizacionais, ou seja, entender a política organizacional bem como a cultura de trabalho com objetivo de familiarizar-se com o sistema e sua história. Os cientistas sociais e antropólogos usam técnicas de observação para desenvolver um entendimento completo e detalhado de culturas particulares.

Nesta técnica, o analista se insere no ambiente de trabalho em que o sistema será utilizado. O trabalho diário é observado e são anotadas as tarefas reais em que o sistema será utilizado. O principal objetivo da etnografia é que ela ajuda a descobrir requisitos de sistema implícitos, que refletem os processos reais, em vez de os processos formais, onde as pessoas estão envolvidas.

Etnografia é particularmente eficaz na descoberta de dois tipos de requisitos:

- Os requisitos derivados da maneira como as pessoas realmente trabalham, em vez da maneira pelas quais as definições de processo dizem como elas deveriam trabalhar;

- Os requisitos derivados da cooperação e conscientização das atividades de outras pessoas.

Alguns itens importantes que devem ser executados antes, durante e depois do estudo de observação:

- Antes, é necessário identificar as áreas de usuário a serem observadas; obter a aprovação das gerências apropriadas para executar as observações; obter os nomes e funções das pessoas chave que estão envolvidas no estudo de observação; e explicar a finalidade do estudo;
- Durante, é necessário familiarizar-se com o local de trabalho que está sendo observado. Para isso é preciso observar os agrupamentos organizacionais atuais; as facilidades manuais e automatizadas; coletar amostras de documentos e procedimentos escritos que são usados em cada processo específico que está sendo observado; e acumular informações estatísticas a respeito das tarefas, como: frequência que ocorrem, estimativas de volumes, tempo de duração para cada pessoa que está sendo observada. Além de observar as operações normais de negócios acima é importante observar as exceções;
- Depois, é necessário documentar as descobertas resultantes das observações feitas. Para consolidar o resultado é preciso rever os resultados com as pessoas observadas e/ou com seus superiores.

A análise de observação tem algumas desvantagens como, consumir bastante tempo e o analista ser induzido a erros em suas observações. Mas em geral a técnica de observação é muito útil e frequentemente usada para complementar descobertas obtidas por outras técnicas.

Workshops

Trata-se de uma técnica de elicitação em grupo usada em uma reunião estruturada. Devem fazer parte do grupo uma equipe de analistas e uma seleção dos *stakeholders* que melhor representam a organização e o contexto em que o sistema será usado, obtendo assim um conjunto de requisitos bem definidos.

Ao contrário das reuniões, onde existe pouca interação entre todos os elementos presentes, o *workshop* tem o objetivo de acionar o trabalho em equipe. Há um facilitador neutro cujo papel é conduzir a *workshop* e promover a discussão entre os vários mediadores. As tomadas de decisão são baseadas em processos bem definidos e com o objetivo de obter um processo de negociação, mediado pelo facilitador.

Uma técnica utilizada em *workshops* é o *brainstorming*. Após os *workshops* serão produzidas documentações que refletem os requisitos e decisões tomadas sobre o sistema a ser desenvolvido.

Alguns aspectos importantes a serem considerados: a postura do condutor do seminário deve ser de mediador e observador; a convocação deve possuir dia, hora, local, horário de início e de término; assunto a ser discutido e a documentação do seminário.

Prototipagem

Protótipo tem por objetivo explorar aspectos críticos dos requisitos de um produto, implementando de forma rápida um pequeno subconjunto de funcionalidades deste produto. O protótipo é indicado para estudar as alternativas de interface do usuário; problemas de comunicação com outros produtos; e a viabilidade de atendimento dos requisitos de desempenho. As técnicas utilizadas na elaboração do protótipo são várias: interface de usuário, relatórios textuais, relatórios gráficos, entre outras.

Alguns dos benefícios do protótipo são as reduções dos riscos na construção do sistema, pois o usuário chave já verificou o que o analista captou nos requisitos do produto. Para ter sucesso na elaboração dos protótipos é necessária a escolha do ambiente de prototipagem, o entendimento dos objetivos do protótipo por parte de todos os interessados no projeto, a focalização em áreas menos compreendidas e a rapidez na construção.

Entrevistas

A entrevista é uma das técnicas tradicionais mais simples de utilizar e que produz bons resultados na fase inicial de obtenção de dados. Convém que o entrevistador dê margem ao entrevistado para expor as suas idéias. É necessário ter um plano de entrevista para que não haja dispersão do assunto principal e a entrevista fique longa, deixando o entrevistado cansado e não produzindo bons resultados.

As seguintes diretrizes podem ser de grande auxílio na direção de entrevistas bem sucedidas com o usuário: desenvolver um plano geral de entrevistas, certificar-se da autorização para falar com os usuários, planejar a entrevista para fazer uso eficiente do tempo, utilizar ferramentas automatizadas que sejam adequadas, tentar descobrir que informação o usuário está mais interessado e usar um estilo adequado ao entrevistar.

Para planejar a entrevista é necessário que antes dela sejam coletados e estudados todos os dados pertinentes à discussão, como formulários, relatórios, documentos e outros. Dessa forma, o analista estará bem contextualizado e terá mais produtividade nos assuntos a serem discutidos na entrevista.

É importante determinar um escopo relativamente limitado, focalizando uma pequena parte do sistema para que a reunião não se estenda por mais de uma hora. O usuário tem dificuldade de concentração em reuniões muito longas, por isso é importante focalizar a reunião no escopo definido.

Após a entrevista é necessário validar se o que foi documentado pelo analista está de acordo com a necessidade do usuário, que o usuário não mudou de opinião e que o usuário entende a notação ou representação gráfica de suas informações.

A atitude do analista em relação à entrevista é determinar seu fracasso ou sucesso. Uma entrevista não é uma competição, deve-se evitar o uso excessivo de termos técnicos e não conduzir a entrevista em uma tentativa de persuasão. O modo como o analista fala não deve ser muito alto, nem muito baixo, tampouco indiretamente, ou seja, utilizar os termos: ele disse isso ou aquilo na reunião para o outro entrevistado. O modo melhor para agir seria, por exemplo, dizer: O João vê a solução para o projeto dessa forma. E o senhor André, qual é a sua opinião? Em uma entrevista o analista nunca deve criticar a credibilidade do entrevistado. O analista deve ter em mente que o entrevistado é o perito no assunto e fornecerá as informações necessárias ao sistema.

Para elaborar perguntas detalhadas é necessário solicitar que o usuário:

- Explique o relacionamento entre o que está em discussão e as demais partes do sistema;
- Descreva o ponto de vista de outros usuários em relação ao item que esteja sendo discutido;
- Descreva informalmente a narrativa do item em que o analista deseja obter informações;
- Perguntar ao usuário se o item em discussão depende para a sua existência de alguma outra coisa, para assim poder juntar os requisitos comuns do sistema, formando assim um escopo conciso.

Pode-se utilizar a confirmação, para tanto o analista deve dizer ao usuário o que acha que ouviu ele dizer. Neste caso, o analista deve utilizar as suas próprias palavras em lugar das do entrevistado e solicitar ao entrevistado confirmação do que foi dito.

Saiba mais sobre [entrevistas para coleta de requisitos](#)

Questionários

O uso de questionário é indicado, por exemplo, quando há diversos grupos de usuários que podem estar em diversos locais diferentes do país. Neste caso, elaboram-se pesquisas específicas de acompanhamento com usuários selecionados, que a contribuição em potencial pareça mais importante, pois não seria prático entrevistar todas as pessoas em todos os locais.

Existem vários tipos de questionários que podem ser utilizados. Entre estes podemos listar: múltipla escolha, lista de verificação e questões com espaços em branco. O questionário deve ser desenvolvido de forma a minimizar o tempo gasto em sua resposta.

Na fase de preparação do questionário deve ser indicado o tipo de informação que se deseja obter. Assim que os requisitos forem definidos o analista deve elaborar o questionário com questões de forma simples, clara e concisa, deixar espaço suficiente para as repostas que forem descritivas e agrupar as questões de tópicos específicos em um conjunto com um título especial. O questionário deve ser acompanhado por uma carta explicativa, redigida por um alto executivo, para enfatizar a importância dessa pesquisa para a organização.

Deve ser desenvolvido um controle que identifique todas as pessoas que receberão os questionários. A distribuição deve ocorrer junto com instruções detalhadas sobre como preenchê-lo e ser indicado claramente o prazo para devolução do questionário. Ao analisar as respostas dos participantes é feito uma consolidação das informações fornecidas no questionário, documentando as principais descobertas e enviando uma cópia com estas informações para o participante como forma de consideração pelo tempo dedicado a pesquisa.

Brainstorming

Brainstorming é uma técnica para geração de idéias. Ela consiste em uma ou várias reuniões que permitem que as pessoas sugiram e explorem idéias.

As principais etapas necessárias para conduzir uma sessão de *brainstorming* são:

- Seleção dos participantes: Os participantes devem ser selecionados em função das contribuições diretas que possam dar durante a sessão. A presença de pessoas bem informadas, vindas de diferentes grupos garantirá uma boa representação;
- Explicar a técnica e as regras a serem seguidas: O líder da sessão explica os conceitos básicos de brainstorming e as regras a serem seguidas durante a sessão;
- Produzir uma boa quantidade de idéias: Os participantes geram tantas idéias quantas forem exigidas pelos tópicos que estão sendo o objeto do brainstorming. Os participantes são convidados, um por vez, a dar uma única idéia. Se alguém tiver problema, passa a vez e espera a próxima rodada.

No *brainstorming* as idéias que a princípio pareçam não convencionais, são encorajadas, pois elas frequentemente estimulam os participantes, o que pode levar a soluções criativas para o problema. O número de idéias geradas deve ser bem grande, pois quanto mais idéias forem propostas, maior será a chance de aparecerem boas idéias. Os participantes também devem ser encorajados a combinar ou enriquecer as idéias de outros e, para isso, é necessário que todas as idéias permaneçam visíveis a todos os participantes.

Nesta técnica é designada uma pessoa para registrar todas as idéias em uma lousa branca ou em papel. À medida que cada folha de papel é preenchida, ela é colocada de forma que todos os participantes possam vê-la.

Analisar as idéias é a fase final do *brainstorming*. Nessa fase é realizada uma revisão das idéias, uma de cada vez. As consideradas valiosas pelo grupo são mantidas e classificadas em ordem de prioridade.

JAD

JAD (*Joint Application Design*) é uma técnica para promover cooperação, entendimento e trabalho em grupo entre os usuários desenvolvedores.

O JAD facilita a criação de uma visão compartilhada do que o produto de software deve ser. Através da sua utilização os desenvolvedores ajudam os usuários a formular problemas e explorar soluções. Dessa forma, os usuários ganham um sentimento de envolvimento, posse e responsabilidade com o sucesso do produto.

A técnica JAD tem quatro princípios básicos:

- Dinâmica de grupo: são realizadas reuniões com um líder experiente, analista, usuários e gerentes, para despertar a força e criatividade dos participantes. O resultado final será a determinação dos objetivos e requisitos do sistema;
- Uso de técnicas visuais: para aumentar a comunicação e o entendimento;
- Manutenção do processo organizado e racional: o JAD emprega a análise top down e atividades bem definidas. Possibilita assim, a garantia de uma análise completa reduzindo as chances de falhas ou lacunas no projeto e cada nível de detalhe recebe a devida atenção;
- Utilização de documentação padrão: preenchida e assinada por todos os participantes. Este documento garante a qualidade esperada do projeto e promove a confiança dos participantes.

A técnica JAD é composta de duas etapas principais: planejamento, que tem por objetivo elicitar e especificar os requisitos; e projeto, em que se lida com o projeto de software.

Cada etapa consiste em três fases: adaptação, sessão e finalização. A fase de adaptação consiste na preparação para a sessão, ou seja, organizar a equipe, adaptar o processo JAD ao produto a ser construído e preparar o material. Na fase de sessão é realizado um ou mais encontros estruturados, envolvendo desenvolvedores e usuários onde os requisitos são desenvolvidos e documentados. A fase de finalização tem por objetivo converter a informação da fase de sessão em sua forma final (um documento de especificação de requisitos).

Há seis tipos de participantes, embora nem todos participem de todas as fases:

- Líder da sessão: é responsável pelo sucesso do esforço, sendo o facilitador dos encontros. Deve ser competente, com bom relacionamento pessoal e qualidades gerenciais de liderança;
- Engenheiro de requisitos: é o participante diretamente responsável pela produção dos documentos de saída das sessões JAD. Deve ser um desenvolvedor experiente para entender as questões técnicas e detalhes que são discutidos durante as sessões e ter habilidade de organizar idéias e expressá-las com clareza;
- Executor: é o responsável pelo produto sendo construído. Tem que fornecer aos participantes uma visão geral dos pontos estratégicos do produto de software a ser construído e tomar as decisões executivas, tais como alocação de recursos, que podem afetar os requisitos e o projeto do novo produto;
- Representantes dos usuários: são as pessoas na empresa que irão utilizar o produto de software. Durante a extração de requisitos, os representantes são frequentemente gerentes ou pessoas-chave dentro da empresa que tem uma visão melhor do todo e de como ele será usado;
- Representantes de produtos de software: são pessoas que estão bastante familiarizadas com as capacidades dos produtos de software. Seu papel é ajudar os usuários a entender o que é razoável ou possível que o novo produto faça;
- Especialista: é a pessoa que pode fornecer informações detalhadas sobre um tópico específico.

O conceito do JAD de abordagem e dinâmica de grupo poderá ser utilizado para diversas finalidades, como: planejamento de atividades técnicas para um grande projeto, discussão do escopo e objetivos de um projeto e estimativa da quantidade de horas necessárias para desenvolver sistemas grandes e complexos.

A maioria das técnicas JAD funciona melhor em projetos pequenos ou médios. Para um sistema grande e complexo podem ser usadas múltiplas sessões JAD para acelerar a definição dos requisitos do sistema.

Saiba mais sobre [as competências para se tornar um Engenheiro de Requisitos](#)

Saiba mais sobre [JAD](#)

Reunião de Kick-off

No início de um projeto, é muito importante reunir todos os participantes para definir vários assuntos referentes ao desenvolvimento e implantação do projeto. Logo, pode-se dizer que este é um momento de kick-off do projeto. Se você já jogou futebol nos videogames, deve se lembrar que a partida começa quando a palavra kick-off aparece na tela, não é? Pois bem, é nesse sentido que o conceito de kick-off será tratado no contato com um novo projeto ou possibilidade de negócio.

O que é uma reunião de Kick-off?

A reunião de passagem, Kick-off, em inglês, significa o sinal de início de uma atividade ou evento, e é um termo bastante utilizado em projetos de marketing. Essa reunião normalmente tem duração de duas a três horas, dependendo da dimensão do projeto e do compartilhamento de informações para esclarecer as possíveis dúvidas dos participantes. Ao final desta reunião, finalmente dá-se o início ao desenvolvimento do projeto.

A reunião de Kick-off sempre conta com um responsável que irá conduzir o andamento de cada tópico da reunião. Para isso, é necessário que exista um documento principal composto de todas as características do projeto, de forma que todos possam acompanhar a discussão da reunião. Este documento recebe o nome de Project Charter.

O mesmo é uma forma documental da passagem de bastão entre o setor de vendas ao setor de acompanhamento e/ou gerencia do projeto.

Em algumas empresas o setor de acompanhamento do sucesso é o ponto de comunicação direto com o cliente e dará continuidade a todo o projeto estando abaixo de uma pessoa que dará o direcionamento do atendimento.

O Project Charter na Reunião de Kick-off

O Termo de Abertura do Projeto (TAP), ou em inglês Project Charter, é o documento que autoriza formalmente o projeto. Ele concede ao Gerente do Projeto a autoridade para utilizar os recursos da organização na execução das atividades do projeto. Um projeto não deve ser iniciado sem que o Termo de Abertura do Projeto seja criado e assinado pela alta gerência e/ou pelo patrocinador do projeto. Além de formalizar a iniciação de um projeto, o TAP traz outros benefícios:

Garante um início de projeto com limites definidos;

Criação de um registro formal do projeto;

Uma maneira de se obter aceitação e comprometimentos formais da direção executiva com o projeto;

Serve como um ponto central para futuras referências durante o andamento do projeto.

Especificação do Trabalho do Projeto (ETP)

A ETP é uma descrição dos produtos, serviços ou resultados a serem entregues pelo projeto. Para projetos internos, o responsável inicial ou patrocinador fornece a especificação do trabalho com base nos requisitos e nas necessidades do negócio. Para projetos externos, os detalhes de um documento de licitação ou de um contato podem compor ou ser interpretados como a Especificação do Trabalho do Projeto.

A ETP informa, entre outros:

1. Necessidades de negócio

Por exemplo, uma necessidade da organização para cortar custos, aumentar produtividade, se adequar a uma lei ou regulamentação governamental ou diminuir impacto ambiental. Geralmente necessidades de negócio e análises de custo benefício são expressas através de um business case como justificativa para o projeto.

2. Descrição do escopo do produto

Documenta características, requisito e detalhes funcionais e técnicos do produto, serviço ou resultado esperado.

3- Planejamento estratégico

Documenta o cenário no qual a organização está inserida e seus objetivos e metas estratégicos. Por exemplo, posicionamento no mercado, liderança no setor, aumento de participação, investimentos no exterior, etc. O documento esclarece como o projeto está alinhado com o planejamento estratégico.

Acordos

Acordos são usados para definir as intenções de um projeto, podem tomar a forma de contratos, acordos de nível de serviço, cartas de intenção, e-mails ou outros tipos de acordo por escrito. Acordos podem acontecer entre departamentos ou áreas funcionais ou na forma de contrato quando o projeto está sendo realizado para um cliente ou por fornecedor externo.

Fatores Ambientais e Ativos Específicos da Organização

Como exemplo, pode-se mencionar:

Padrões de mercado;

Normas governamentais para o segmento no qual a organização atua;

Processos corporativos;

Políticas internas de aprovações e projetos;

Modelos usados em projetos anteriores ou similares;

Informações de projetos similares (riscos, custos, cronograma, etc.);

Base de lições aprendidas;

EAP's;

Recursos usados;

Riscos de mercado.

Outras Informações de Alto Nível, Sumarizadas ou Preliminares

Premissas e restrições (de custos, cronograma, qualidade, escopo, etc.);

Limites do Projeto (de custos, cronograma, qualidade, escopo, etc.);

Riscos de Alto Nível ou Preliminares;

Resumo dos principais marcos do cronograma;

Resumo do orçamento;

Lista das partes interessadas conhecidas;

Critérios de Sucesso do Projeto;

Gerente do Projeto designado, seu nível de autoridade e suas responsabilidades;

Nome e assinatura do patrocinador ou responsável pela autorização do Termo de Abertura do Projeto.

Customer Success na reunião de Kick-off

O conceito de CS (customer success) ou acompanhamento do sucesso tem que estar exposto desde a reunião de passagem pois dela surgiu todo o alinhamento que dará o encanto necessário para não só seu cliente achar seu serviço eficaz como também trazer o encantamento.

Identificando pontos em comum entre o atendedor e o atendido, também buscando as formas como o cliente gosta e gostaria de ser tratado, como espera que o projeto seja levado.

É importante estabelecer uma relação humana, que muitas vezes pode ser direcionada a forma como sua empresa acha melhor através de gatilhos mentais.

3 Gatilhos mentais para reuniões

Gatilho de autoridade

Transmita a ideia de que você é uma autoridade em seu nicho. Assim irá gerar menos complicações na hora de colocar um ponto de discordância com o cliente.

E mais importante que isso: seja uma autoridade! Castelos construídos sobre areia não duram muito tempo.

Gatilho de reciprocidade

O gatilho da reciprocidade é também a base do inbound marketing. Isso porque temos uma tendência natural a querer retribuir àquilo ou àquele que nos gera valor de alguma forma.

Mas é claro que precisamos perceber esse ato como algo espontâneo, feito de coração. Por isso suas ações devem, primeiramente, objetivar tornar a vida das pessoas mais simples e, depois, gerar algum lucro para você. Não o contrário. As pessoas sentem isso.

Gatilho de prova social

O ser humano tem a necessidade de pertencer a grupos que nos identifiquem como indivíduos, dessa forma, a prova social é um gatilho muito poderoso.

Como a fila de um restaurante que a princípio pode parecer uma coisa ruim ao pensar que você irá ter que esperar mais para comer porém se parar para pensar que se está cheio quer dizer que muita gente quer comer ali e portanto ele deve ser melhor do que os que estão vazios.

Por que fazer uma reunião de kick-off?

Muitos projetos fracassam justamente por não contarem com o preparo adequado. Enquanto a equipe tem uma ideia do que será feito, o cliente pode ter outra, completamente diferente, mesmo que todos os pontos já tenham sido discutidos.

A reunião de kick off serve para alinhar as expectativas de todos, rever os pontos mais importantes do planejamento e eliminar toda e qualquer dúvida ainda existente.

É por isso que todos os envolvidos devem estar presentes. Trata-se do momento certo para você motivar equipe e cliente, mostrando que, com um bom trabalho e um fluxo correto de comunicação, todos os desafios serão superados e o sucesso estará muito próximo.

Se bem realizada, a reunião de kick off pode garantir a execução de todo o projeto. Muitos problemas são evitados com ela, uma vez que as expectativas de todos são alinhadas e cada etapa é revisada de maneira abrangente.

Assim, atingir os resultados esperados torna-se uma tarefa fácil e simples.

Um bom resumo de uma reunião é dado pelo Professor Robson Camargo que leciona nos cursos de MBA e Pós Graduação pela FGV, FIA USP e Fundação Dom Cabral em parceria com a especialista em comunicação Marília Queiroz no vídeo abaixo

11 dicas para sua reunião de Kick-off?

Uma reunião de kick-off bem conduzida pode trazer grandes benefícios para o desenvolvimento do projeto, pois o cliente terá todas as informações de que precisa para acompanhar a evolução dos trabalhos, criando uma relação de confiança com a equipe.

Sendo assim, na hora de planejá-la e conduzi-la, seja cuidadoso e se atente aos seguintes passos que são a base ou um fio guia para que tenha sucesso nesse que será o primeiro ponto de contato do cliente com sua organização interna:

1. Alinhe as expectativas

O alinhamento de expectativas é o primeiro passo para uma reunião de kick off bem-sucedida. Esse é um ponto crítico. Se estiverem mal alinhadas, cada profissional terá uma perspectiva diferente sobre os resultados do projeto.

Assim, na hora de verificar os resultados, é muito provável que alguns não enxerguem que aquela foi uma iniciativa de sucesso.

Para que o alinhamento de expectativas seja eficaz, liste os pontos que serão abordados durante o encontro. Busque, também, determinar como a reunião será conduzida e até mesmo sua duração.

Isso ajudará a manter todos os envolvidos focados nos objetivos do encontro, evitando a dispersão.

O processo de alinhamento de expectativas envolve uma completa revisão dos objetivos de cada etapa do projeto. Mostre o ponto de partida delas e verifique sempre se todos entenderam cada fator envolvido.

Isso garantirá que os colaboradores atuem de acordo com os objetivos do negócio e evitará que, ao final de uma etapa, o time tenha que refazer algo.

2. Introduza a equipe

Conquistar a confiança do cliente logo no primeiro momento o transformará em um grande aliado durante todo o projeto.

E uma forma de começar com o pé direito é simplesmente apresentar a equipe que desenvolverá a solução, citando brevemente o nível de experiência profissional e de conhecimentos que agregarão valor ao produto final.

Fornecer e-mails e telefones de pessoas-chave para que o cliente possa manter um contato direto também ajuda a estabelecer uma relação baseada na confiança e na cooperação.

Os canais de comunicação devem ser abertos e acessíveis. É importante que todos consigam trocar ideias e atuar de maneira colaborativa sempre que for necessário.

A boa comunicação torna os times mais unidos e eficientes. Todos terão uma rotina de trabalho em que dados importantes podem ser trocados a qualquer momento.

Com isso, a resolução de problemas torna-se ágil, o número de erros cai e as entregas de resultados ficam mais rápidas.

3. Retome o escopo

O primeiro ponto prático a ser abordado é a definição do escopo. Em poucas palavras, trata-se de um resumo de todos os limites do projeto.

Nele, são listados o que será feito, o que não deve ser executado e até que ponto há uma flexibilidade para mudanças ao longo da iniciativa.

Relembre a todos os objetivos e as metas do projeto, assim como as premissas, as restrições, os riscos e outros detalhes que sejam importantes para dimensionar o que será feito.

Um alinhamento de escopos eficaz reduz as chances de o projeto ter uma série de problemas, como o estouro do orçamento causado por um grande aumento das atividades previstas.

Além disso, as chances de o gestor perder o controle sobre o projeto são elevadas. Sem que ele saiba até que ponto cada time pode atuar, desvios ocorreram com maior frequência, o que atrasará a finalização de etapas e tornará a rotina de todos menos eficaz.

Portanto, sempre inicie um projeto a partir do momento em que os escopos estão bem definidos.

4. Apresente o cronograma

O cronograma de execução também deve ser discutido na reunião de kick off, pois consiste na garantia de que o cliente sabe exatamente como e quando serão realizadas as entregas do projeto.

Dessa forma, não haverá motivos para discordâncias posteriores. Se houver qualquer tipo de questionamento em relação ao cronograma, anote para uma verificação futura.

Revise todas as etapas do projeto. Repasse custos, metas e datas. Verifique também se todos os que serão afetados por uma atividade ou rotina já estão preparados.

A apresentação do cronograma evita que a companhia tenha de fazer modificações durante a execução do projeto nas datas e nos horários definidos previamente.

Como todos estarão conscientes dos dias em que os processos críticos serão executados, será mais fácil para cada time preparar-se para as mudanças planejadas.

5. Mostre os custos

Outro ponto bastante importante a ser abordado na reunião de kick off está relacionado ao levantamento de custos do projeto.

Esse dado é fundamental para que o cliente saiba quanto está investindo e qual será o retorno obtido após a implementação total da solução.

Lembre-se de que, em um projeto (especialmente os de grande porte), as chances de os custos operacionais elevarem-se rapidamente são altas.

Os gastos com a iniciativa podem aumentar a qualquer momento e por uma grande quantidade de motivos.

Erros, atrasos, falta de materiais e mudanças no escopo estão entre os principais.

Saber como controlá-los e ter um time de profissionais capazes de atuar de maneira econômica auxilia a empresa a executar tarefas sempre da melhor maneira possível, buscando maximizar o retorno sobre o investimento realizado.

6. Liste os stakeholders

Por mais que o projeto seja uma solução personalizada para o cliente, podem existir outros stakeholders envolvidos que impactem significativamente em sua condução.

Relembrar quem são eles, seu nível de importância e como serão tratados servirá para que a equipe e o cliente estejam cientes de seus esforços.

Durante um projeto, o conhecimento desse fator evita problemas e conflitos internos. Gerentes e stakeholders saberão seus respectivos papéis, a quem responder e como as suas atividades devem ser executadas.

Como consequência, será mais prático gerenciar as rotinas e o fluxo de trabalho.

7. Anuncie os indicadores

Você também pode apresentar os KPIs de sucesso do projeto, demonstrando como o desempenho da equipe será avaliado.

Lembre-se, afinal, de que pessoas cientes do que é esperado delas são mais engajadas e trabalham por resultados concretos.

Cada indicador auxiliará todos a mensurar fatores como o resultado de uma etapa, o nível de produtividade e a qualidade do trabalho do time.

Além disso, as métricas dão uma visão abrangente sobre o funcionamento de todos os processos do projeto.

Não se esqueça de definir, de maneira eficaz, as métricas que serão utilizadas durante o projeto.

Para que isso seja possível, você deve avaliar o perfil de cada time, suas capacidades e o que é esperado para a equipe.

Indicadores muito elevados podem dificultar a identificação de um trabalho bem executado.

Ao mesmo tempo, uma métrica abaixo do que se espera dos profissionais pode dificultar o rastreamento de falhas e pontos a serem melhorados.

8. Estabeleça os canais

Já que o cliente precisa ser informado de tudo o que ocorre no projeto, estipule como, quando e porque ele será informado. Definir a frequência das comunicações, quem será o responsável por manter o contato e qual tipo de comunicação será estabelecido o deixará mais tranquilo e fortalecerá os laços com a equipe do projeto.

Só não vale falhar nesse compromisso, deixando de informá-lo sobre o andamento dos trabalhos, ok? Um bom fluxo de comunicação contribui para que todos tenham maior segurança diante do trabalho que está sendo executado.

Faça relatórios regulares, apontando o progresso do projeto, os resultados obtidos e o que foi mensurado pelas métricas.

Também inclua, durante as comunicações, os problemas e desafios encontrados.

Mencione o que foi executado para eliminar falhas e quais são as medidas adotadas para evitar a recorrência do problema.

Ao mesmo tempo, busque por feedbacks, garantindo que a iniciativa está no caminho certo diante do que é esperado por todos.

9. Escolha a metodologia

Gerenciamento em cascata ou metodologias ágeis de projetos? Uma única metodologia ou a combinação de duas ou mais? Definir a forma com que o time trabalhará é de suma importância para que todos atuem em sinergia, contribuindo uns com os outros para maximizar os resultados do projeto.

A escolha da metodologia deve ser feita de acordo com o perfil do projeto. Identifique os desafios, as metas e os requisitos de cada etapa.

Além disso, avalie como cada sistemática pode ser integrada aos fluxos de trabalho dos colaboradores envolvidos.

Uma má escolha neste contexto impactará diretamente nos resultados obtidos. E é justamente daí que parte a importância de uma decisão acertada: ela direcionará as rotinas de trabalho, os padrões e uma série de fatores existentes em cada etapa.

Se você preferir uma metodologia ágil, por exemplo, o cliente deverá estar ciente de seu funcionamento, de quais são suas vantagens e do que esperar em cada entrega.

Ao mesmo tempo, se o negócio optar por um método com viés tradicional, garanta que o cliente saiba dos desafios dessa escolha e de como ela afetará os resultados.

10. Capriche na apresentação

Todas as informações que você preparar para a reunião de kick off devem ser apresentadas de maneira clara e assertiva, visando manter o encontro produtivo e eficiente. Evite slides com muita informação — prefira sempre inserir um dado em cada lâmina, a fim de facilitar o entendimento.

É importante ser objetivo e claro na hora de passar informações importantes. Bons slides são aqueles que complementam a apresentação (e não os que são o “principal personagem”).

Em outras palavras, utilize-os como uma ferramenta, e não a única forma de exibir o que será feito.

Lembre-se também de que a apresentação é apenas um ponto de apoio! Todos os convocados para a reunião devem ter recebido a pauta com antecedência — o que possibilita que listem seus questionamentos antes do encontro.

Abra espaço para críticas e para solucionar dúvidas. Assim, todos sairão da reunião de kick off prontos para lidar com os desafios do projeto.

11. Aposte na negociação

Como é bastante comum que o cliente queira fazer ajustes de última hora no projeto, você deve estar preparado para a negociação. Pequenas alterações que não afetem a condução do trabalho podem ser facilmente aceitas.

Em contrapartida, aquelas que colocam a equipe em xeque (como alterações de prazos) devem ser avaliadas com cuidado, de forma a não arriscar toda a iniciativa.

Negocie as mudanças focando nos resultados que elas trarão para o projeto. Avalie o impacto das medidas tomadas e tenha a certeza de que todos estão de acordo com os novos rumos tomados.

Assim, o projeto manterá o seu sucesso ao término de todas as etapas e a companhia poderá ter um trabalho de ponta.

Especificação do Trabalho do Projeto (ETP)

A ETP é uma descrição dos produtos, serviços ou resultados a serem entregues pelo projeto. Para projetos internos, o responsável inicial ou patrocinador fornece a especificação do trabalho com base nos requisitos e nas necessidades do negócio. Para projetos externos, os detalhes de um documento de licitação ou de um contato podem compor ou ser interpretados como a Especificação do Trabalho do Projeto. (por exemplo, Solicitação de Proposta, Solicitação de Informações ou Solicitação de Licitação).

A ETP informa, entre outros:

1. Necessidades de negócio

Por exemplo, uma necessidade da organização para cortar custos, aumentar produtividade, se adequar a uma lei ou regulamentação governamental ou diminuir impacto ambiental. Geralmente necessidades de negócio e análises de custo benefício são expressas através de um business case como justificativa para o projeto.

2. Descrição do escopo do produto

Documenta características, requisito e detalhes funcionais e técnicos do produto, serviço ou resultado esperado.

3- Planejamento estratégico

Documenta o cenário no qual a organização está inserida e seus objetivos e metas estratégicos. Por exemplo, posicionamento no mercado, liderança no setor, aumento de participação, investimentos no exterior, etc. O documento esclarece como o projeto está alinhado com o planejamento estratégico.

Business Case

Business Case ou documento de propósito e conteúdo semelhante descreve informações e detalhes do ponto de vista de negócios para determinar como o projeto justifica o seu investimento. O business case é normalmente usado pela alta gerência no processo decisório e de comparação entre vários possíveis projetos. O Business Case diz a todos porque a organização deve fazer determinado projeto. O Projeto precisa estar alinhado com o Business case por isso o business case, na íntegra ou parcialmente, faz parte do Termo de Abertura do Projeto

Acordos

Acordos são usados para definir as intenções de um projeto, podem tomar a forma de contratos, acordos de nível de serviço, cartas de intenção, e-mails ou outros tipos de acordo por escrito. Acordos podem acontecer entre departamentos ou áreas funcionais ou na forma de contrato quando o projeto está sendo realizado para um cliente ou por fornecedor externo.

Fatores Ambientais e Ativos Específicos da Organização

Como exemplo, pode-se mencionar:

- Padrões de mercado;
- Normas governamentais para o segmento no qual a organização atua;
- Processos corporativos;
- Políticas internas de aprovações e projetos;
- Modelos usados em projetos anteriores ou similares;
- Informações de projetos similares (riscos, custos, cronograma, etc);
- Base de lições aprendidas;
- EAP's;
- Recursos usados;
- Riscos de mercado.

Outras Informações de Alto Nível, Sumarizadas ou Preliminares

- Premissas e restrições (de custos, cronograma, qualidade, escopo, etc.);
- Limites do Projeto (de custos, cronograma, qualidade, escopo, etc.);
- Riscos de Alto Nível ou Preliminares;
- Resumo dos principais marcos do cronograma;
- Resumo do orçamento;
- Lista das partes interessadas conhecidas;
- Critérios de Sucesso do Projeto;
- Gerente do Projeto designado, seu nível de autoridade e suas responsabilidades;
- Nome a assinatura do patrocinador ou responsável pela autorização do Termo de Abertura do Projeto.

No gerenciamento de projetos, quando você quer apresentar uma argumentação bem elaborada a respeito de uma nova ideia, de um projeto a ser desenvolvido, uma boa saída é elaborar um **business case**.

Mas o que é um business case?

O **business case** ou caso de negócios, é um documento ou até mesmo uma proposta verbal bem fundamentada para incentivar a tomada de decisão ou ação a respeito de um negócio, desenvolvido durante os estágios iniciais de um projeto e descreve o porquê, o que, como e quem é necessário para decidir se vale a pena ou não tocar adiante.

Se você é gerente de projetos e consultar: “**business case, o que é?**”, vai obter a resposta do PMI® (Project Management Institute) que esse documento vai servir para justificar ou não o investimento em um projeto. E para quem ainda não sabe, o PMI é o instituto mais renomado mundialmente nas boas práticas do gerenciamento de projetos.

Mas em sua dimensão mais simples, o **business case** pode também ser apresentado apenas oralmente, sem necessidade de papéis, para ações de curto prazo e que trarão benefícios imediatos.

Mas nem sempre os projetos podem ser defendidos de forma tão simplificada. Para isso, é fundamental a construção de um **business case** bem eficaz e explicativo, cuidadosamente construído, com as partes interessadas, expectativas a serem atendidas, indicativos de sucesso, benefícios, previsão de entregas e outros dados.

Como fazer business case?

Para quem trabalha em um Project Management Office (**PMO**), **business case** pode até ter modelos a serem seguidos, especialmente em grandes corporações que trabalham com demandas muito grandes de investimentos e múltiplos projetos.

Ou você pensa que **business case para Ambev** ou outras grandes empresas não é um procedimento até mesmo comum em qualquer projeto a ser executado?

Ter um **business case exemplo** pode adiantar alguns passos na montagem dessa argumentação, porque o gerente de projetos vai que saber exatamente que respostas buscar a questões como: qual o custo de um projeto, retorno do investimento, fluxo de caixa e diversas outras perguntas que certamente o Sponsor (patrocinador) fará, antes de dar o seu aval a qualquer empreitada.

Basicamente em um **business case template**, 4 tópicos são essenciais para um documento bem elaborado, que vai fornecer informações sobre riscos e recompensas envolvidos na tomada de ação, com abordagens viáveis para a implantação do projeto:

Números: quais são as receitas para tocar esse projeto e as despesas e resultados esperados financeiramente após implementação deste novo produto, serviço ou solução para um problema?

Pessoas: quais são os profissionais que farão parte do projeto? Há profissionais especializados em cada setor necessário do projeto? Quem vai dar o respaldo?

Mercado: qual é o momento e comportamento do mercado para esse projeto? Tem demanda? Vai inovar? Vai gerar interesse? Baseado em quê? Estudos e pesquisas?

Produtos ou serviço: o que esse novo produto ou serviço vai oferecer de inovação ou benefício para os usuários? O que vai solucionar?

O business case precisa reunir os benefícios, as desvantagens, custos e riscos da situação atual e projeção do futuro.

Por isso, com uma boa base em dados de análise de mercado, fatores econômicos, viabilidade financeira, ficará muito mais fácil justificar um projeto.

Por isso, o **business case** vai servir como uma excelente ferramenta para transmitir a ideia do projeto e até para obter financiamento.

Como começar o business case?

Mas mesmo quando falamos em **business case template**, estamos dando uma ideia geral de como pode ser feito o documento, porque cada projeto vai exigir sua pesquisa, de acordo com seu segmento de mercado.

Você deve pensar em seguir essa linha condutora no seu documento:

- Objetivo do documento
- Objetivo do projeto
- Necessidades a serem atendidas com a realização do projeto
- Governança
- Análise competitiva
- Stakeholders e custos
- Recursos humanos

Fique atento especialmente ao item de projeções financeiras, especialmente no caso de um financiamento, que deve estar orientados para um período de 3 a 5 anos.

Mas lembre-se que o documento não deve ser tão extenso, até 3 páginas é o suficiente, mas nele devem ser transmitidos o essencial para demonstrar valor e benefícios.

Embora fique focado mais na figura do sponsor e do gerente de projetos, o business case deve contar com a colaboração também da equipe que será envolvida e de especialistas de outros setores que ficam em sua órbita, como TI, RH e financeiro.

Depois de seu projeto aprovado após a elaboração do business case, agora é partir para outra etapa: planejar e se preparar a fase de elaboração onde serão envolvidos todos os demais membros de elaboração da sua equipe.

Como fazer um business case

Quanto vai custar o projeto? Em quanto tempo vamos ter o retorno sobre o investimento? Qual será o fluxo de caixa do projeto? Estas são algumas perguntas que seu business case, ou plano de negócio, precisa responder para justificar o investimento do seu patrocinador em seu projeto. Seja gerenciando projetos internos ou externos, mais cedo ou mais tarde, você precisará justificar o investimento necessário para um projeto.

Vendo que alguns gerentes de projetos possuem dificuldade quando o assunto é atestar a viabilidade financeira de um projeto, preparamos este artigo com o objetivo de te ajudar a construir seu primeiro business case ou otimizar a criação de planos de negócio dentro da sua organização.

O Plano de Negócio

Um plano de negócio (ou business case) é um guia para orientar a criação de um novo negócio ou empreendimento. Esse documento descreve como estará estruturado o negócio com base em diferentes perspectivas, como: análise do mercado; fatores econômicos; viabilidade financeira, dentre outros levantamentos necessários que servirão de base para justificar a implementação do novo projeto. Em resumo, o plano de negócios é um instrumento que permite comunicar uma ideia de negócio para vender (internamente ou externamente) ou para obter um financiamento. Trata-se também de uma ferramenta de uso interno para a organização, pois lhe permite validar a viabilidade das suas iniciativas e serve também como métrica de avaliação pós implementação.

Como elaborar o Business Case

Não existe uma forma exata, seu formato será mais granulado ou mais detalhista conforme a sua aplicação. Para solicitar um recurso no BNDES, por exemplo, existe uma necessidade grande de detalhamento, para seguir um processo interno de apresentação. No caso de um comitê de projeto pode ser mais resumido ou até mesmo um Pitch (documento de uma folha que descreve um novo negócio). O sumário abaixo pode ser utilizado como um guia simples e eficaz:

- Sumário executivo
- História da empresa/projeto
- O produto/serviço
- O mercado
- A concorrência
- Marketing
- Manufatura e operações
- Equipe de gestão
- Resultados históricos (quando disponíveis) e projeções financeiras

Cada um dos itens listados possuem sua importância, no entanto, um ponto que necessita uma atenção especial é a seção *Resultados históricos e projeções financeiras*. Neste item, todos os detalhes e os dados quantitativos são apresentados em outras partes e submetidos a um teste numérico. Em resumo, as projeções financeiras são construídas com todas as estimativas de vendas da empresa, enquanto que, todos os custos de produção, distribuição, logística, despesas comerciais e administrativas são construídas em um resumo financeiro.

A elaboração das projeções financeiras deve estar sempre orientada a um período de três a cinco anos para que estas sirvam como base para a equipe de gestão, em relação aos objetivos da organização nesse período, e também para comunicá-las à instituições financeiras ou à possíveis investidores. O plano financeiro deve ser construído em padrões conhecidos, podendo ser mais ou menos simplificado, de acordo com seu segmento de mercado e a qualidade das informações obtidas em toda a sua pesquisa, base histórica ou levantamento, mas em resumo conter como resultados:

- Demonstrações financeiras (balanço patrimonial, demonstrativo de resultados/ lucros e perdas e fluxo de caixa) dos últimos anos (se disponível);
- Projeções financeiras e fluxo de caixa para o período de três anos;
- Análises econômicas: ponto de equilíbrio, lucratividade, retorno sobre investimento etc.

Quando houver a necessidade de captação de capital externo, por meio de financiamento, é importante incluir em suas projeções um resumo, desde a utilização até a liquidação dos recursos. Potenciais investidores e financiadores querem ver quanto de recurso financeiro seu negócio irá precisar, assim como a sua capacidade de pagamento e geração de valor.