


História do Sistemas



Roteiro da aula

- ◆ Introdução aos Sistemas de Informação
 - ◆ Modelagem de sistemas: evolução histórica
 - ◆ O processo de desenvolvimento de software
 - ◆ Atividades típicas de um processo de desenvolvimento
 - ◆ Participantes do processo de desenvolvimento
 - ◆ Classificação dos Sistemas
- 

Introdução

- ◆ As grandes transformações ocorridas nos últimos anos, impulsionadas pelo avanço da tecnologia provocaram a passagem da antiga sociedade industrial para uma nova sociedade baseada na informação e no conhecimento.
- ◆ Nos dias de hoje, a empresa que dispõe de mais informações sobre seu processo está em vantagem em relação a suas competidoras



Introdução

- ◆ Em consequência do crescimento da importância da informação, surgiu a necessidade de gerenciar informações de uma forma adequada e eficiente e, dessa necessidade, surgiram os denominados sistemas de informação.
- ◆ Mas o que são Sistemas?



Sistema de Informação

- ◆ Um sistema de informações é uma combinação de pessoas, dados, processos, interface, redes de comunicação e tecnologia que interagem com o objetivo de dar suporte e melhorar o processo de negócio de uma organização empresarial com relação as informações que nela fluem. O desenvolvimento de um sistema de informações é uma tarefa complexa e um dos seus componentes é denominado sistema de software.

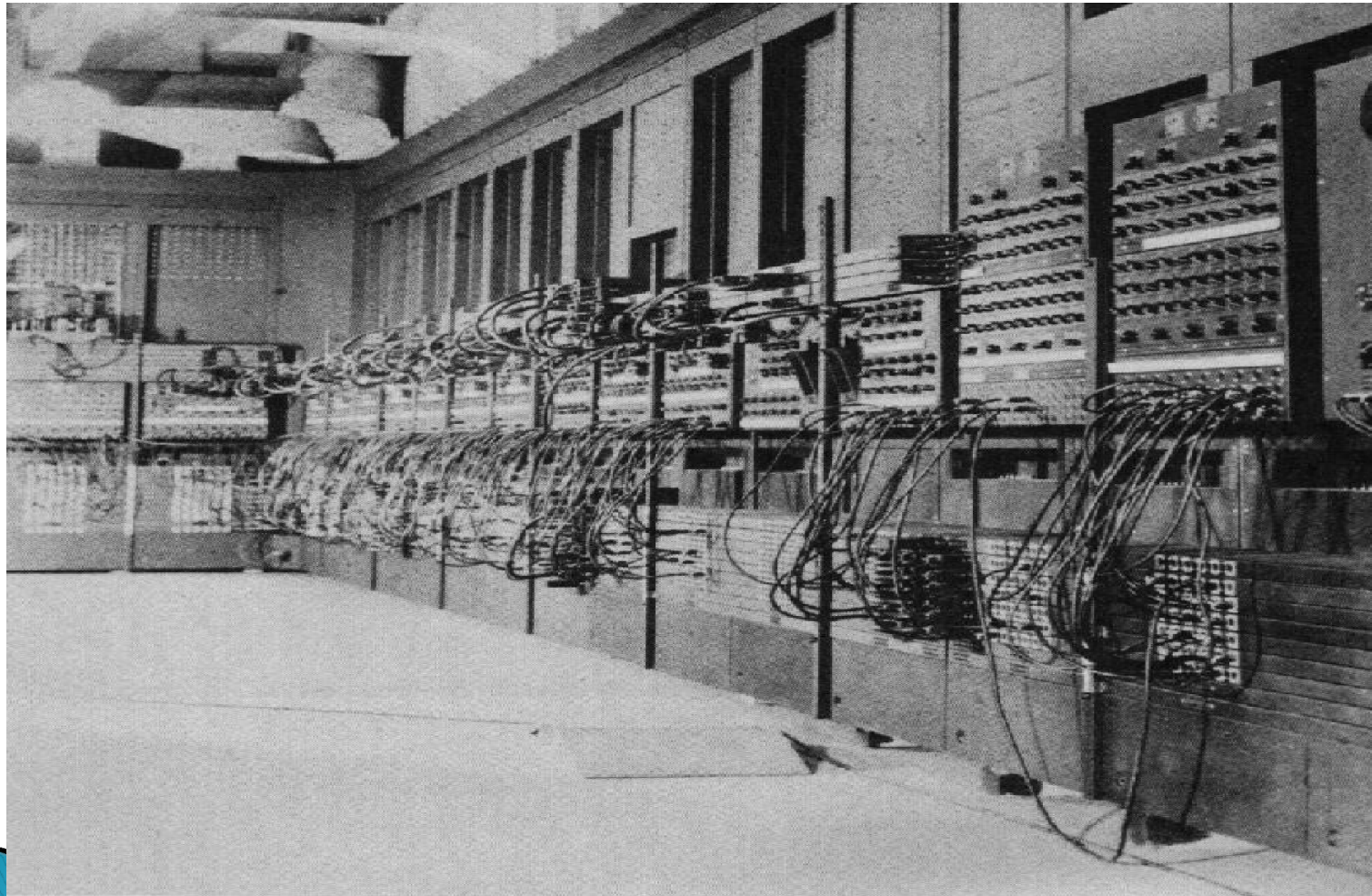


Sistema de Software

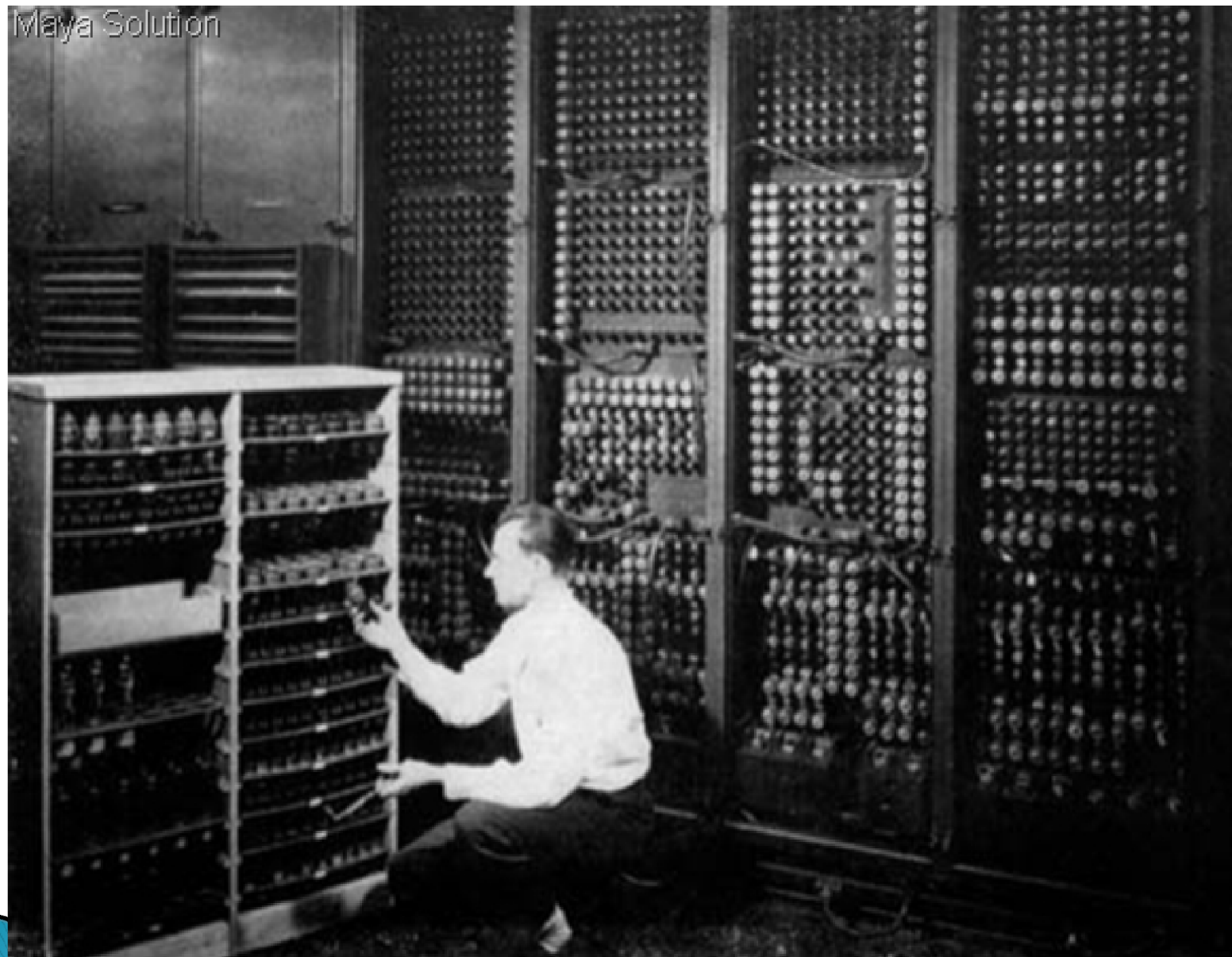
- ◆ O objetivo principal de um sistema de software é adicionar valor a organização na qual esse sistema será utilizado, ou seja, ele implicará na produtividade dos processos da empresa na qual o sistema será utilizado.
- ◆ Um sistema de software é composto de módulos funcionais computadorizados que interagem entre si para proporcionar aos usuários do sistema a automatização de diversas tarefas.



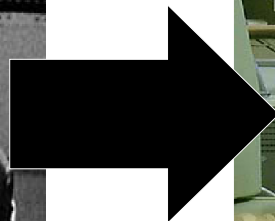
Surgiu o computador



Maya Solution



E a tecnologia **deu um salto** na história





Modelagem de Software

- ◆ Uma característica de sistemas de software é a complexidade de seu desenvolvimento, que aumenta à medida que cresce o tamanho do sistema.
 - Comparação: Construir uma casa de cachorro, uma casa para família e um edifício.
- ◆ Assim como na construção civil, existe uma gradação de complexidade na construção de software. Quanto mais complexo um sistema de software, mais ele deverá ser planejado.
 - Comparação: Sistema de estoque, sistema de controle de tráfego aéreo.



Modelagem de Software

- ◆ A necessidade de planejamento no desenvolvimento de sistemas de informação leva ao conceito de modelagem de software, ou seja, antes do software ser concebido deve-se criar um modelo para o mesmo.
- ◆ Um modelo pode ser visto como uma representação idealizada de um sistema a ser construído.
- ◆ Exemplos de modelos: maquetes de edifício, plantas de casa, fluxogramas, etc.



Modelagem de Software

- ◆ A modelagem de sistemas de software consiste na utilização de notações gráficas e textuais com o objetivo de construir modelos que representam as partes essenciais de um sistema.
- ◆ São várias as razões para se utilizar modelos na construção de sistemas.



Modelagem de Software

- ◆ Razões para se utilizar modelos:
 - Gerenciamento de complexidade
 - Comunicação entre as pessoas envolvidas
 - Redução dos custos no desenvolvimento
 - Previsão do comportamento futuro de sistema

ETC...



Modelagem de Software

- ◆ Apesar de um diagrama conseguir expressar diversas informações de forma gráfica, em diversos momentos há a necessidade de adicionar informações na forma de texto, com o objetivo de explicar ou definir certas partes desse diagrama.
- ◆ A modelagem de um sistemas em forma de diagrama, juntamente com a informação textual associada, formam a documentação de um sistema de software.



Evolução da Modelagem de Sistemas

- ◆ Década de 1950/60: os sistemas de software eram bastante simples e dessa forma as técnicas de modelagem também. Era a época dos fluxogramas e diagramas de módulos
- Década de 1970: nessa época houve uma grande expansão do mercado computacional. Sistemas complexos começavam a surgir e por consequência, modelos mais robustos foram propostos. Nesse período surge a programação estruturada e no final da década a análise e o projeto estruturado.



Evolução da Modelagem de Sistemas

- ◆ Década de 1980: surge a necessidade por interfaces homem-máquina mais sofisticadas, o que originou a produção de sistemas de software mais complexos. A análise estruturada se consolidou na primeira metade dessa década e em 1989 Edward Yourdon lança o livro *Análise Estruturada Moderna*, tornando-o uma referência no assunto



Evolução da Modelagem de Sistemas

- ◆ Década de 1990: nesse período surge um novo paradigma de modelagem, a Análise Orientada a Objetos, como resposta a dificuldades encontradas na aplicação da Análise Estruturada a certos domínios de aplicação.



Evolução da Modelagem de Sistemas

- ◆ Final da década de 90 e momento atual: o paradigma da orientação a objetos atinge a sua maturidade. Os conceitos de padrões de projetos (design patterns), frameworks de desenvolvimento, componentes e padrões de qualidade começam a ganhar espaço. Nesse período surge a Linguagem de Modelagem Unificada (UML), que é a ferramenta de modelagem utilizada no desenvolvimento atual de sistemas.



O Processo do Desenvolvimento de Software

- ◆ O processo de desenvolvimento de software é uma atividade bastante complexa. Isso se reflete no alto número de projetos de software que não chegam ao fim, ou que extrapolam recursos de tempo e de dinheiro alocados.
- ◆ Em um estudo clássico sobre projetos de desenvolvimento de software realizado em 1994 foi constatado que:



O Processo do Desenvolvimento de Software

- ◆ Porcentagem de projetos que terminam dentro do prazo estimado: 10%
- ◆ Porcentagem de projetos que são descontinuados antes de chegarem ao fim: 25%
- ◆ Porcentagem de projetos acima do custo esperado: 60%
- ◆ Atraso médio nos projetos: um ano



O Processo do Desenvolvimento de Software

- ◆ Para lidar com essa complexidade e minimizar os problemas envolvidos no desenvolvimento de software, temos a definição de processos de desenvolvimento de software.
- ◆ Um processo de desenvolvimento de software compreende todas as atividades necessárias para definir, desenvolver, testar e manter um produto de software.



O Processo do Desenvolvimento de Software

- ◆ Entre os objetivos de um processo de desenvolvimento de software, temos:
 - Definir quais as atividades a serem executadas ao longo do projeto
 - Quando, como e por quem tais atividades serão executadas
 - Prover pontos de controle para verificar o andamento do desenvolvimento
 - Padronizar a forma de desenvolver software numa organização
- ◆ Exemplos de processos: RUP, XP, SCRUM, etc



Atividades Típicas de um Processo de Desenvolvimento

- ◆ Cada processo tem suas particularidades em relação as atividades de desenvolvimento, porém em geral temos algumas atividades comuns à maioria dos processos existentes.

São elas:

- Levantamento de requisitos
- Análise
- Projeto
- Implementação
- Testes
- Implantação



Levantamento de Requisitos

- ◆ A atividade de levantamento de requisitos corresponde à etapa de compreensão do problema aplicada ao desenvolvimento de software. O principal objetivo é que usuários e desenvolvedores tenham a mesma visão do problema a ser resolvido.
- ◆ Nessa etapa, os desenvolvedores juntamente com os clientes, tentam levantar e definir as necessidades dos futuros usuários do sistema a ser desenvolvido. Essas necessidades são denominadas requisitos.



Levantamento de Requisitos

- ◆ Durante o levantamento de requisitos, a equipe de desenvolvimento tentar entender o domínio que deve ser automatizado pelo sistema de software. Essa etapa compreende também um estudo exploratório das necessidades dos usuários e da situação do sistema atual (caso exista). Existem várias técnicas utilizadas para isso, como, exemplo: leitura de obras de referência e livros-texto, observação do ambiente do usuário, realização de entrevistas com usuários e especialista do domínio, reutilização de análises anteriores, comparação com sistemas preexistentes.



Levantamento de Requisitos

- ◆ Durante o levantamento de requisitos, a equipe de desenvolvimento tentar entender o domínio que deve ser automatizado pelo sistema de software. Essa etapa compreende também um estudo exploratório das necessidades dos usuários e da situação do sistema atual (caso exista). Existem várias técnicas utilizadas para isso, como, exemplo: leitura de obras de referência e livros-texto, observação do ambiente do usuário, realização de entrevistas com usuários e especialista do domínio, reutilização de análises anteriores, comparação com sistemas preexistentes.



Levantamento de Requisitos

- ◆ O produto do levantamento de requisitos é o documento de requisitos, que declara os diversos tipos de requisitos do sistema. As principais seções de um documento de requisitos são:
 1. Requisitos funcionais: definem as funcionalidades do sistema. Veja alguns exemplos a seguir:
 - a) O sistema deve permitir que cada professor realize o lançamento de notas das turmas nas quais lecionou.
 - b) O sistema deve permitir que um aluno realize a sua matrícula nas disciplinas oferecidas em um semestre letivo.



Levantamento de Requisitos

- ◆ 1.Requisitos não-funcionais: declaram as características de qualidade que o sistema deve possuir e que estão relacionadas às suas funcionalidades. Alguns tipos de requisitos não funcionais são:
 - a)Desempenho: requisitos que definem tempos de resposta esperados para as funcionalidades do sistema
 - b)Segurança: limitações sobre a segurança do sistema em relação a acessos não autorizados
 - c)Portabilidade: restrições sobre a plataforma de hardware e software nas quais o sistema será implantado



Levantamento de Requisitos

- ◆ 1. Restrições normativas: declaração de restrições impostas sobre o desenvolvimento do sistema. Essas restrições definem, por exemplo, a adequação a custos e prazos, a plataforma tecnológica, aspectos legais, limitações sobre a interface com o usuário, eventuais necessidades de comunicação do novo sistema com sistemas legados, componentes de hardware e software a serem adquiridos, etc.



Levantamento de Requisitos

- ◆ Os requisitos devem ser expressos de uma maneira tal que eles possam ser verificados e comunicados a leitores técnicos e não-técnicos. Um ponto importante sobre o documento de requisitos é que ele não deve conter informações sobre as soluções técnicas que serão adotadas para desenvolver o sistema. O enfoque prioritário do levantamento de requisitos é: “o que o usuário necessita do novo sistema?”.



Levantamento de Requisitos

- ◆ O documento de requisitos serve como um termo de consenso entre a equipe técnica (desenvolvedores) e o cliente. Esse documento constitui a base para as atividades subsequentes do desenvolvimento do sistema e fornece um ponto de referência para qualquer validação futura do software construído. Além disso, o documento de requisitos estabelece o escopo do sistema (isto é, o que faz parte e o que não faz parte do sistema). O escopo de um sistema muda muitas vezes durante o desenvolvimento, contudo o planejamento inicial do projeto deve se basear no escopo inicial. O documento de requisitos serve como um termo de consenso entre a equipe técnica (desenvolvedores) e o cliente.



Análise

- ◆ Nessa etapa os analistas realizam um estudo detalhado dos requisitos levantadas na atividade anterior. A partir desse estudo, são construídos modelos para representar o sistema a ser construído. Assim como na etapa anterior (levantamento de requisitos, a etapa de análise não leva em conta a tecnologia a ser utilizada. Nesta atividade o objetivo é obter a melhor solução para o problema sem se preocupar com os detalhes da tecnologia usada.



Análise

- ◆ Os modelos construídos na fase de análise devem ser cuidadosamente validados e verificados.
- ◆ O objetivo da validação é assegurar que as necessidades do cliente estão sendo atendidas. Nessa atividade, os analistas apresentam os modelos criados para representar o sistema aos futuros usuários para que esses modelos sejam validados



Projeto

- ◆ O foco principal da análise são os aspectos lógicos e independentes de implementação de um sistema. Na fase de projeto, determina-se “como” o sistema funcionará para atender aos requisitos, de acordo com os recursos tecnológicos existentes. Aos modelos construídos na fase de análise são adicionados alguns aspectos como: arquitetura do sistema, padrão de interface gráfica, a linguagem de programação, o gerenciador de banco de dados, etc.



Implementação

- ◆ Na fase de implementação, o sistema é codificado, ou seja, ocorre a tradução da descrição computacional obtida na fase de projeto em código executável mediante o uso de uma ou mais linguagens de programação.
- ◆ A implementação envolve a criação do código fonte correspondente às classes de objetos do sistema utilizando linguagens de programação como C#, C++, Java, etc.
- ◆ Além da codificação desde o início, a implementação pode também reutilizar componentes de software, bibliotecas de classes e frameworks para agilizar a atividade.



Testes

- ◆ Diversas atividades de teste são realizadas para verificação do sistema construído, levando-se em conta a especificação feita na fase de projeto. O principal produto dessa fase é o relatório de testes, com informações sobre erros detectados no software.
- ◆ Após a atividade de testes, os diversos módulos do sistema são integrados, resultando finalmente no produto de software.



Implantação

- ◆ O sistema é empacotado, distribuído e instalado no ambiente do usuário. Os manuais do sistema são escritos, os arquivos são carregados, os dados são incorporados para o sistema, e os usuários treinados para utilizar o sistema corretamente.
- ◆ Em alguns casos, ocorre a migração de sistemas de software e de dados preexistentes.



Participantes do processo

- ◆ Uma equipe de desenvolvimento de sistemas de software pode envolver vários especialistas como, por exemplo, profissionais de informática para fornecer o conhecimento técnico necessário ao desenvolvimento do sistema e especialistas do domínio para o qual o sistema de software será desenvolvido. Uma equipe típica consiste de: gerentes de projetos, analistas, projetistas, programadores, clientes, especialistas de domínio e grupos de avaliação de qualidade. Essa divisão tem mais um fim didático, na prática, a mesma pessoa desempenha diferentes funções.



Gerentes do projeto

- ◆ É o profissional responsável pela gerência ou coordenação das atividades necessárias à construção do sistema. Esse profissional é o responsável por fazer o orçamento do projeto de desenvolvimento, estimar o tempo necessário de desenvolvimento do sistema, definir qual o processo de desenvolvimento, o cronograma de execução das atividades, a mão-de-obra especializada, os recursos de hardware e software, etc.



Analistas

- ◆ O analista de sistemas deve entender os problemas do domínio do negócio para que possa definir os requisitos do sistema a ser desenvolvido.
- ◆ Uma característica do analista é ser o profissional responsável por entender as necessidades dos clientes em relação ao sistema a ser desenvolvido e repassar esse entendimento aos demais desenvolvedores do sistema. O analista representa uma ponte de comunicação entre duas “facções”: a dos profissionais de computação e a dos profissionais de negócio.



Analistas

- ◆ Em alguns casos há profissionais em uma equipe de desenvolvimento para desempenhar dois papéis: analista de negócios e o analista de sistema.
- ◆ O analista de negócios é responsável por entender o que o cliente faz, por que ele o faz e determinar se as práticas atuais da organização realmente fazem sentido. O analista de sistemas é especializado em traduzir as necessidades do usuário em características de um produto de software.



Analistas

- ◆ Uma característica importante que um analista deve ter é a capacidade de comunicação tanto escrita quanto falada, pois ele é um agente facilitador entre o cliente e a equipe técnica.
- ◆ Outra característica é a ética profissional pois muitas vezes esse profissional está em contato com informações sigilosas e estratégicas dentro da organização na qual está trabalhando.



Projetistas

- ◆ As funções de um projetista são:
 1. Avaliar as alternativas de solução do problema resultante da análise
 2. Gerar a especificação de uma solução computacional detalhada.Existem diversos tipos de projetistas:
 - Especializados nos padrões de uma interface gráfica como Windows, MacOs, Kde, etc.
 - Especializados em redes de comunicação
 - Especializados em projeto de banco de dados
 - Etc.



Arquiteto de software

- ◆ Profissional encontrado principalmente em grandes equipes reunidas para desenvolver sistemas complexos. O objetivo desse profissional é elaborar a arquitetura do sistema como um todo. É ele quem toma decisões sobre quais são os subsistemas que compõem o sistema como um todo e quais são as interfaces entre esses subsistemas.



Programadores

- ◆ É o responsável pela implementação do sistema.
- ◆ Na maioria das equipes de desenvolvimento possui analistas que realizam alguma programação e programadores que realizam alguma análise.



Especialista de domínio

- ◆ Também conhecido como especialista do negócio. Possui conhecimento acerca da área ou do negócio em que o sistema em desenvolvimento estará inserido.
- ◆ Normalmente o especialista de domínio é o próprio cliente. Podem-se distinguir dois tipos de clientes: o cliente usuário e o cliente contratante.



Especialista de domínio

- ◆ Cliente usuário: é o indivíduo que efetivamente utilizará o sistema. É com esse tipo de cliente que o analista de sistemas interage para levantar os requisitos do sistema.
- ◆ Cliente contratante: é o indivíduo que solicita o desenvolvimento do sistema.



Avaliadores de qualidade

- ◆ São responsáveis por assegurar a adequação do processo de desenvolvimento e do produto de software para que sejam desenvolvidos aos padrões de qualidade estabelecidos pela organização. O desempenho e a confiabilidade são exemplos de características que devem ser encontradas em um sistema de software de boa qualidade



Classificação dos sistemas

- ◆ Os Sistemas de informação podem ser classificados de acordo com o nível organizacional atendido, ou seja, o tipo de informação gerado pelo sistema e o perfil do usuário que utiliza ou necessita daquelas informações.
- ◆ Quanto mais alto o nível do usuário, mais consolidadas devem ser as informações.
- ◆ Quanto mais baixo ele se encontrar na pirâmide organizacional, mais detalhada devem ser as informações.
- ◆ Dessa forma podemos classificar os sistemas em 3 categorias (operacional, tático, estratégico)



Classificação dos sistemas

- ◆ Estratégico
- ◆ Tático
- ◆ Operacional
- ◆ Sistemas de Informações Executivas
- ◆ Sistemas de Informações Gerenciais
- ◆ Sistemas de Processamento de Transações

