

ANÁLISE E DESENHO DE SISTEMAS

Processo de análise da situação de negócio, com o propósito de o melhorar através de procedimentos e métodos mais adequados.

ANÁLISE DE SISTEMAS

Processo de reunir e interpretar factos, diagnosticar problemas, utilizar estes factos para melhorar o sistema.

DESENHO DE SISTEMAS

Análise específica o que é que o sistema deve fazer. Desenho apresenta como é que o sistema deve ser desenvolvido.

PROFISSIONAIS DE SISTEMAS DE INFORMAÇÃO

Analista de Sistemas ou Analistas de Informação

Determinar as necessidades de informação e requisitos de processamento. Conduzir estudos relacionados com a actividade de negócio.

Analista e Desenhador de Sistemas ou Desenhadores de sistemas ou aplicações

Responsáveis pelo estudo exaustivo do negócio e pelo desenho do novo sistema.

Analistas, Desenhadores e Programadores de Sistemas ou Programadores Analistas

Conduzem o projecto desde a análise do sistema, desenvolvem as especificações e desenho, e programam o software que implementa o desenho especificado.

MUDANÇA ORGANIZACIONAL

O ambiente de instabilidade em que vivem hoje as organizações, leva a que estas sejam capazes de antecipar a mudança e de responder de forma inovadora

A MUDANÇA ORGANIZACIONAL PODE SER:

- ✓ **PLANEADA**, ou não, quando responde a um estímulo de uma forma praticamente instantânea;
- ✓ **REACTIVA**, porque se reconheceu internamente que a organização já não é viável;
- ✓ **PRÓ-ACTIVA**, quando a organização reconhece novas possibilidades para si mesma;
- ✓ **INCREMENTAL**, quando se trata de uma mudança planeada;
- ✓ **TRANSFORMACIONAL**, quando se procede a mudanças na cultura e na concepção do trabalho segundo um plano previamente elaborado;
- ✓ **EVOLUCIONÁRIA**, quando o crescimento da organização implica a implementação de alterações não planeadas;
- ✓ **REVOLUCIONÁRIA**, quando a organização se vê obrigada a fazer alterações profundas no seu funcionamento.

OS SI E A MUDANÇA ORGANIZACIONAL

As TI são um agente de mudança organizacional mas, no entanto, por si só, não provocam mudanças significativas para a organização, pelo que é necessária a sua organização com os SI

Várias organizações associam claramente aos SI o papel de permitir a mudança, sendo difícil exemplificar casos onde conduzem a mudança, apesar de lhe atribuírem esse papel

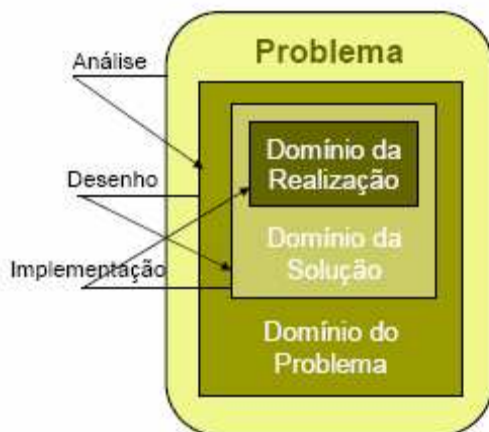
A vantagem conferida pelo SI na mudança surge de um balanço entre factores internos e externo e entre o domínio da organização e das TI

Os SI assumem um papel importante, acrescentando valor à mudança, uma vez que oferecem várias opções para reorganizar o trabalho nas organizações

DESENVOLVIMENTO ESTRUTURADO DE SISTEMAS

- ✓ Desenvolvimento sistemático.
- ✓ Entender o problema (O QUÊ).
- ✓ Descrever o problema.
- ✓ Desenhar a solução baseada na compreensão do problema.
- ✓ Construir modelos que assentem em regras bem definidas.
- ✓ Modelos revistos e comentados pelos utilizadores.
- ✓ Documentar todo o processo de desenvolvimento.
- ✓ Usar de preferência notações gráficas para representar o modelo do sistema.

PROCESSO DE DESENVOLVIMENTO DE S. I.



A partir do enunciado de um problema aplica-se um conjunto de actividades de análise para determinar o domínio do problema; a partir do domínio do problema aplica-se um conjunto de actividades de desenho para determinar o domínio da solução; a partir do domínio da solução aplica-se um conjunto de actividades de implementação para determinar o domínio da realização, que é o produto final de um projecto.

PROJECTO DE SISTEMAS

Barry Boehm identificou 7 questões que qualquer projecto de S. I. deverá responder:

1. Porque é que o sistema vai ser desenvolvido?
2. O que vai / deve ser feito?
3. Quando é que vai ser feito?
4. Quem é o responsável?
5. Onde é que as responsabilidades estão localizadas?

6. Como é que vai ser feito?
 7. Quanto vai custar em termos de recursos?
-

FASES DO PROCESSO DE DESENVOLVIMENTO DE S. I. – IMP.

1. CONCEPÇÃO

Identificar “o que é que o sistema deve fazer”, nomeadamente a informação a processar, as funcionalidades a implementar, as restrições existentes, os critérios que determinam o sucesso e a aceitação; devem ser consideradas e avaliadas diferentes alternativas e efectuada a respectiva selecção.

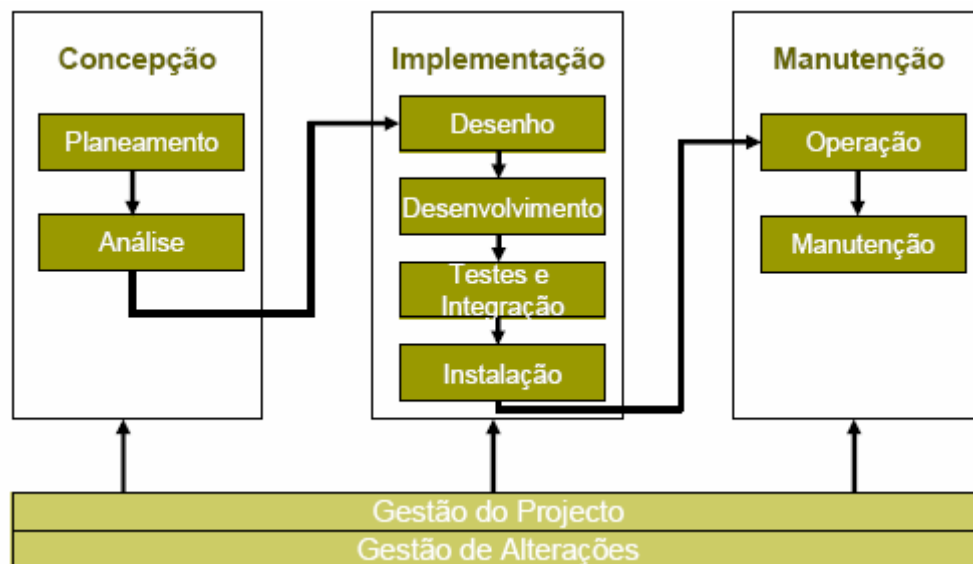
2. IMPLEMENTAÇÃO

Identificar “o como fazer o sistema”, e construí-lo de facto; nomeadamente, serão definidas e construídas as estruturas de dados, os programas, os módulos, as interfaces (internas e externas), os testes a realizar; no final desta fase deverá ser disponibilizado o sistema de forma funcional.

3. MANUTENÇÃO

Inclui todas as **alterações posteriores à aceitação do produto pelo cliente final**: correcção de erros, introdução de melhorias e/ou de novas funcionalidades.

FASES E TAREFAS DO PROCESSO DE DESENVOLVIMENTO DE S. I.



METODOLOGIAS ESTRUTURADAS

Aplicação de um conjunto de princípios com o objectivo de formalizar o processo de identificação de requisitos, de modo a reduzir as possibilidades da má interpretação dos mesmos e introduzir técnicas baseadas nas melhores práticas ao processo de análise e desenho.

Sequência de fases e actividades, com inputs e outputs, regras, intervenientes, técnicas, notações, ferramentas, documentação, técnicas de gestão, etc, com o objectivo de prestar mais atenção ao processo global e menos à programação.

TIPOS DE ANÁLISE

- ✓ **FUNCIONAL** – orientada para a decomposição funcional do sistema e a identificação dos respectivos processos.
- ✓ **ORGÂNICA** – centra-se nos dados, colocando os conceitos e as entidades manipuladas no negócio como os elementos mais importantes do desenvolvimento do sistema.

PROCESSO DE DESENVOLVIMENTO DE S. I.

ACTIVIDADES:

- ✓ Compreender a missão e organização da organização.
- ✓ Identificar e envolver todos os interessados e afectados pela introdução do sistema.
- ✓ Obter uma visão de alto nível do funcionamento do sistema actual, caso ele exista.
- ✓ Definir o âmbito do sistema.
- ✓ Elaborar uma descrição de alto nível do problema.
- ✓ Identificar restrições, problemas e riscos do projecto.
- ✓ Identificar alternativas de implementação, proceder à sua avaliação e selecção.
- ✓ Apresentar resultados e recomendações, com justificação técnica, funcional e financeira.
- ✓ Elaborar e obter aprovação de um plano de projecto.
- ✓ Definir o processo de controlo do projecto.

TÉCNICAS:

- ✓ Análise financeira de custos e/ou benefícios.
- ✓ Elaboração de estimativas.
- ✓ Elaboração de planos do projecto (identificação de tarefas, elaboração de
- ✓ Diagramas de Pert e/ou Gantt).
- ✓ Identificação de riscos e medidas de os controlar.
- ✓ Elaboração de árvores ou tabelas de decisão.
- ✓ Aplicação de técnicas de modelação de processos.

PROBLEMAS NO DESENVOLVIMENTO DE S. I.

Falta de qualidade, traduzida no incumprimento dos requisitos e nos problemas que se verificam após a instalação do produto.

Incumprimentos dos prazos estabelecidos para o desenvolvimento do software.

Ineficiente controlo de custos previamente definidos para o desenvolvimento do software.

PARADIGMAS PARA O DESENVOLVIMENTO DE SI

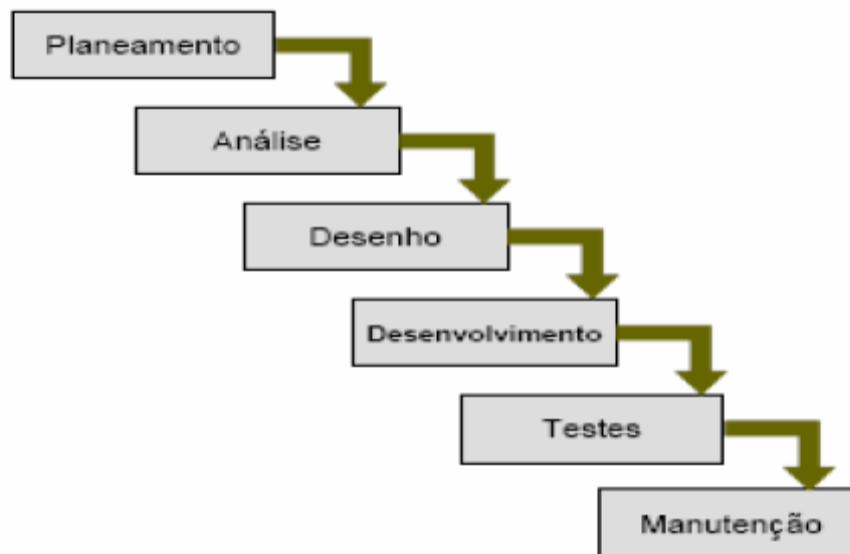
Existem vários paradigmas para o desenvolvimento de SI:

Ciclo convencional de desenvolvimento de sistemas de informação (Waterfall Model):

- ✓ Prototipagem
- ✓ Spiral Model
- ✓ RAD (Rapid Application Development)

O paradigma deve ser escolhido baseado na natureza do sistema de informação a desenvolver.

MODELO EM CASCATA (WATERFALL MODEL) – IMP.



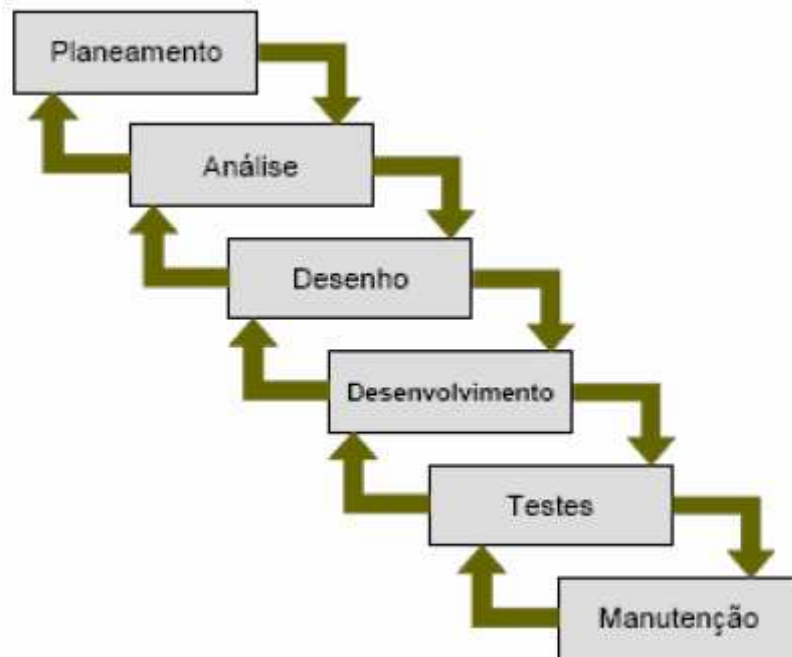
VANTAGENS

1. Este modelo adapta-se às necessidades das organizações.
2. As fases do modelo correspondem exactamente aos níveis de abstracção da hierarquia da organização.
3. Só se avança para a tarefa seguinte quando o cliente valida e aceita os produtos finais.
4. Pressupõe que o cliente participa activamente no projecto e sabe bem o que quer.

DESVANTAGENS

1. Dependência da sequência linear do processo de desenvolvimento, onde uma fase deve ser completada antes de começar outra.
2. Promove a separação dos esforços ao longo das diferentes tarefas, desencorajando a comunicação e partilha de visões entre todos os intervenientes.
3. Minimiza o impacto da compreensão adquirida no decurso de um projecto, pois se um processo não pode voltar atrás de modo a alterar os modelos e as conclusões das tarefas anteriores, é normal que as novas ideias sobre o sistema não sejam aproveitadas.

MODELO EM CASCATA REVISTO



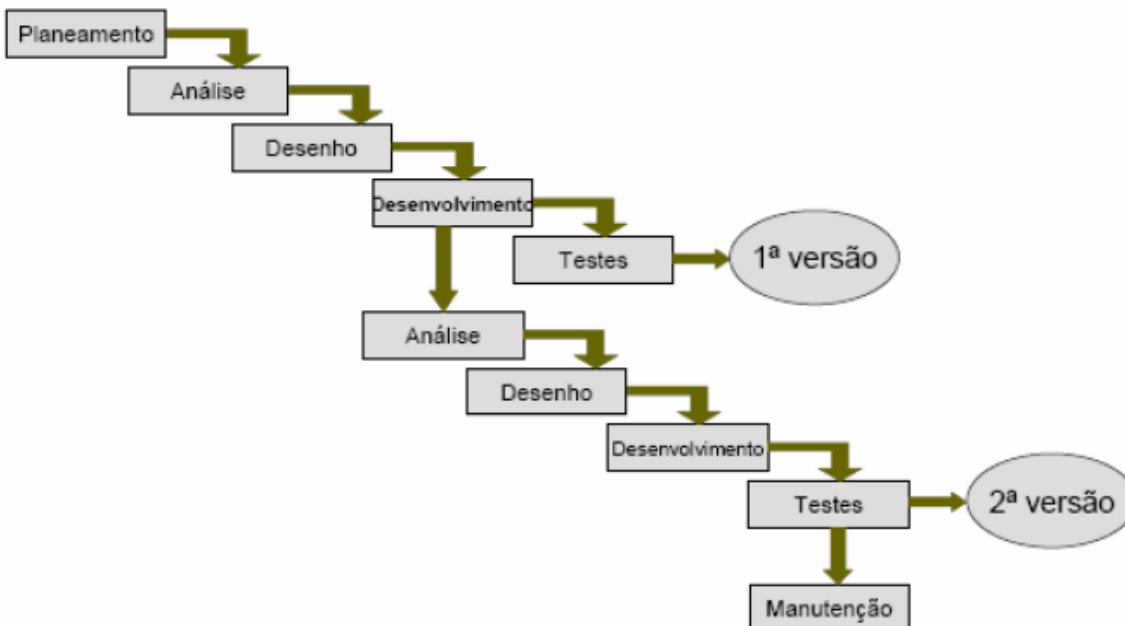
VANTAGENS

Prevê a possibilidade de a partir de qualquer tarefa do ciclo se poder regressado a uma tarefa anterior de forma a contemplar alterações funcionais e/ou técnicas que entretanto tenham surgido.

DESVANTAGENS

Risco desta aproximação é que, na ausência de um processo de gestão do projecto e de controlo das alterações bem definido, pode-se passar o tempo num ciclo sem fim, sem nunca se atingir o objectivo final que é disponibilizar um sistema a funcionar.

MODELO ITERATIVO E INCREMENTAL

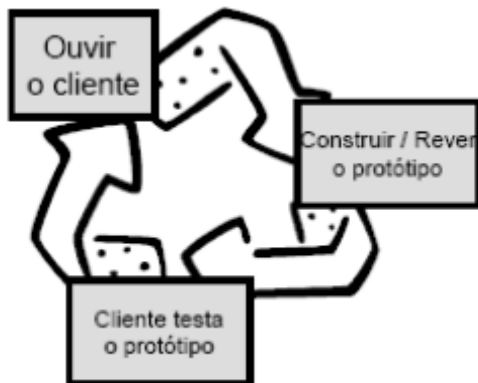


Baseia-se no princípio que a equipa envolvida possa refinar ou alargar pouco-a-pouco a qualidade, detalhe e âmbito do sistema envolvido.

A principal consequência da aproximação iterativa é que os produtos finais de todo o processo vão sendo amadurecidos e completados ao longo do tempo, mas cada iteração produz sempre um conjunto de produtos finais.

Requer processos ágeis!

MODELO DE PROTOTIPAGEM



Apoiar a definição de requisitos.

Desenho rápido do sistema a desenvolver.

Produtos executáveis.

É importante definir as regras do jogo no início – cliente e programador devem estar de acordo que o protótipo seja apenas construído como um mecanismo de definição de requisitos.

MODELO ESPIRAL



Variante do modelo iterativo e incremental.

Foi proposto por Barry Boehm como respostas às críticas que os processos existentes não favoreciam a utilização de prototipagem e reutilização de software.

Além das actividades previstas pelos outros processos propõe logo de seguida à tarefa de planeamento a realização de uma tarefa de prototipagem e de análise de risco, como forma de eliminar os principais problemas e identificar os requisitos do sistema.

XP (Extreme Programming) pode ser considerado um exemplo de um processo de desenvolvimento ágil em espiral.

CRITÉRIOS PARA SELECIONAR O MODELO:

1. CLAREZA DE REQUISITOS

As metodologias RAD de prototipagem são mais apropriadas quando os requisitos não estão claramente definidos pelos utilizadores, assim há a interacção e validação mais cedo com o sistema.

2. FAMILIARIDADE COM A TECNOLOGIA

Se o sistema é desenhado sem alguma familiaridade com a tecnologia base que será utilizada, os riscos aumentam pois as ferramentas podem não ser capazes de fazer o necessário.

3. COMPLEXIDADE DO SISTEMA

Sistemas complexos requerem uma análise e desenho mais detalhada.

4. SEGURANÇA DO SISTEMA

A segurança do sistema é um factor importante no desenvolvimento do sistema.

Metodologias de prototipagem na análise são mais apropriadas quando a segurança do sistema tem uma prioridade elevada.

Metodologias de prototipagem não costumam ser uma boa opção porque não é dada muita importância às fases de análise e desenho.

5. ESCALONAMENTO DO TEMPO CURTO

Metodologias RAD adequam-se bem a projectos com tempos curtos pois introduzem celeridade ao projecto.

Metodologias Waterfall são a pior escolha quando o tempo é essencial, pois este não permite fáceis alterações no escalonamento.

6. VISIBILIDADE DO PLANO

Metodologias RAD antecipam algumas das decisões críticas de desenho; consequentemente, isto vai ajudar os gestores de projecto a reconhecer e endereçar os factores de risco e manter as expectativas altas.

REQUISITOS – IMP.

CONCEITOS GERAIS

- ✓ Requisito
- ✓ Requisitos funcionais e não-funcionais
- ✓ Documento de requisitos

LEVANTAMENTO DE REQUISITOS

- ✓ Questionários, entrevistas, observação directa
- ✓ Análise de requisitos
- ✓ Lista de verificações
- ✓ Matrizes de interacção
- ✓ Negociação dos requisitos

ANÁLISE DE REQUISITOS

A análise de requisitos é fundamental para o desenvolvimento de sistemas, pois trata de descobrir o que o cliente quer com o sistema.

A análise de requisitos está associada ao processo de descobrir quais são as operações que o sistema deve realizar e quais são as restrições que existem sobre estas operações.

OBJECTIVOS

Encontrar problemas, falhas, inconsistências e necessidades.

A análise é intercalada com o levantamento de requisitos.

A análise é suportada por uma lista de verificação de problemas.

O QUE É UM REQUISITO?

- ✓ Funcionalidades pretendidas no sistema
- ✓ Descrição de propriedades do sistema.
- ✓ Descrição de restrições ou condicionantes do sistema.
- ✓ O foco está nas necessidades de negócio
- ✓ Requisitos mudarão ao longo do desenvolvimento do projecto.

TIPOS DE REQUISITOS

FUNCIONAIS – o que o sistema deve fazer

Um processo que o sistema tem de efectuar Informação que o sistema deve conter.

NÃO FUNCIONAIS

Propriedades comportamentais que o sistema deve ter – restrições sobre como o sistema deve desempenhar suas funções:

- ✓ Operacionais
- ✓ Desempenho
- ✓ Segurança
- ✓ Culturais e políticas

LISTA DE VERIFICAÇÃO

1. DESENHO PREMATURO DO SISTEMA

Verificar se o requisito inclui informação prematura sobre o desenho ou implementação.

2. COMBINAÇÃO DE REQUISITOS

Verificar se a descrição do requisito descreve um único requisito ou se pode ser dividida em diferentes requisitos.

3. REQUISITOS DESNECESSÁRIOS

Verificar se o requisito é apenas uma adição “cosmética” ao sistema.

4. UTILIZAÇÃO DE HW NÃO-STANDARD

Verificar se o requisito obriga ao uso de hardware que não é standard.

5. CONFORMIDADE COM OS OBJECTIVOS DE NEGÓCIO

Verificar se o requisito é consistente com os objectivos do negócio definidos na introdução do documento de requisitos.

6. REQUISITOS AMBÍGUOS

Verificar se o requisito pode ter leituras diferentes por pessoas diferentes.

Quais as interpretações possíveis?

7. REQUISITOS REALISTAS

Verificar se o requisito é realista tendo em conta a tecnologia a ser usada para implementar o sistema.

8. REQUISITOS “TESTÁVEIS”

Verificar se os engenheiros de teste podem derivar um teste a partir da descrição do requisito que mostre que o sistema satisfaz esse requisito.

DOCUMENTAÇÃO

RELATÓRIO DE DEFINIÇÃO DE REQUISITOS

Documento formal usado para registar e comunicar os requisitos dos/aos stakeholders.

Podem ser incluídas prioridades.

Importante definir o âmbito do projecto: o que será e não será incluído no sistema.

CONTEÚDO

Serviços e funções que o sistema deve efectuar;

Restrições nas quais o sistema deve funcionar;

Todas as propriedades ou características do sistema;

Identificação de outros sistemas com o qual o sistema deve comunicar ou integrar-se;

Informação sobre o domínio de aplicação do sistema;

Restrições sobre os processos usados para desenvolver o sistema;

Descrição das plataformas computacionais (hardware, redes, ...) sobre as quais o sistema deve correr.

UTILIZADORES DO RELATÓRIO DE REQUISITOS

CLIENTES DO SISTEMA

Especificam requisitos e validam a sua adequação às necessidades.

GESTORES DE PROJECTO

Planear os custos, prazos e tempo de desenvolvimento adequado.

ENG. DO SISTEMA

Entender o sistema a desenvolver.

ENG. DE TESTES DO SISTEMA

Desenvolver testes de validação.

ENG. DE MANUTENÇÃO DO SISTEMA

Compreender o sistema.

TÉCNICAS DE LEVANTAMENTO DE REQUISITOS

- ✓ Entrevistas
- ✓ Questionários
- ✓ Observação directa
- ✓ Análise de documentos
- ✓ JAD – Joint Application Design
- ✓ Prototipagem

ANÁLISE DE DOCUMENTOS

Documentos que contém informação do sistema “as-is”.

DOCUMENTOS TÍPICOS:

- ✓ Regulamentos;
- ✓ Relatórios internos (vendas, stocks, produção);
- ✓ Registos periódicos (registo de pagamento a fornecedores, registo de encomendas);
- ✓ Formulários (facturas, recibos, ficha de cliente, ficha de funcionário);
- ✓ Manuais de procedimentos.

Procurar elementos adicionados pelos utilizadores aos formulários.

Procurar elementos não utilizados.

Deve ser dada particular atenção aos documentos:

Que descrevem a organização;

Que descrevem os conteúdos funcionais dos vários cargos;

Que relatam as actividades da organização;

Que constituem material publicitário e promocional da organização.

QUESTIONÁRIOS

Conjunto de questões escritas, usualmente enviadas para um grande nº de pessoas.

Podem ser em formato de papel ou electrónico.

Seleccionar participantes representativos

Desenvolver questões claras e de fácil análise.

Definir estratégias para obter um bom nº de respostas.

ENTREVISTAS

- ✓ Seleccionar os entrevistados.
- ✓ Desenvolver questões da entrevista.
- ✓ Estabelecer objectivos da entrevista.
- ✓ Conduzir a entrevista.

SELECÇÃO DOS ENTREVISTADOS

Baseada nas necessidades de informação

Obter diferentes perspectivas:

- ✓ Gestores;
- ✓ Utilizadores.

PREPARAÇÃO DA ENTREVISTA

Plano geral da entrevista

Lista de questões;

Previsão de respostas e Follow-Ups.

Confirmar áreas de conhecimento.

Estabelecer prioridades no caso de ter pouco tempo.

Preparação do entrevistado

Marcação da entrevista;

Informar os motivos da entrevista;

Informar as áreas de discussão.

CONDUZIR A ENTREVISTA

Aparência profissional e imparcial

Registar toda a informação

Verificar a política de gravação áudio da organização

Garantir que entende todas as questões e termos.

Separar factos de opiniões.

Dar tempo ao entrevistado para fazer questões.

Agradecer ao entrevistado!

FOLLOW-UP DA ENTREVISTA

- ✓ Preparação das notas da entrevista.
- ✓ Preparação do relatório da entrevista.
- ✓ Verificar falhas e novas questões.

ENTREVISTA VS QUESTIONÁRIO

A entrevista é preenchida por um entrevistador.

O questionário é preenchido pelo questionado.

A Entrevista pode ser Exploratória ou Uniformizada.

O objectivo da Entrevista Exploratória é essencialmente desenvolver ideias e pesquisar hipóteses.

Entrevistas exploratórias individuais são caras, demoradas e não são fáceis de arranjar. Algumas pessoas preferem, conduzir sessões com grupos de entrevistado. (Neste caso é necessário considerar influências e contaminação própria do grupo).

O objectivo da Entrevista Uniformizada é essencialmente recolha de dados (obtenção de factos e de estatísticas).

OBSERVAÇÃO DIRECTA

- ✓ Observação dos processos a serem executados.
- ✓ Utilizadores/gestores não se lembram com exactidão de tudo o que fazem.
- ✓ Valida a informação recolhida com outros métodos.
- ✓ Ter em atenção que o comportamento das pessoas muda quando estão a ser observadas.
- ✓ Tentar ser discreto.
- ✓ Identificar períodos mortos e picos.

O QUE É A UML?

UML é uma linguagem (notação com semântica associada) para Visualizar, Especificar, Construir e Documentar os artefactos de um sistema com uma componente intensiva de software (software intensive system)

UML não é uma metodologia porque não diz quem deve fazer o quê, quando e como.

UML pode ser usado segundo diferentes metodologias, tais como RUP (Rational Unified Process), ICONIX, FDD (Feature Driven Development), etc.

UML não é uma linguagem de programação.

A linguagem é considerada abrangente e genérica o suficiente para permitir a modelagem de sistemas diversos.

A UML É COMPOSTA PELAS SEGUINTE PARTES:

- ✓ **Visões**
- ✓ **Diagramas**
- ✓ **Elementos de Modelo**
- ✓ **Mecanismos Gerais**

Pode ser aplicada em diferentes fases do desenvolvimento de um sistema, desde a especificação da análise de requisitos até à finalização com a fase de testes

A UML suporta modelos estáticos, dinâmicos e funcionais.

A MODELAÇÃO PODE SER:

- ✓ **ESTÁTICA** – suportada pelos diagramas de classes e de objectos.
- ✓ **DINÂMICA** – suportada pelos diagramas de estado, sequência, colaboração e actividade.
- ✓ **FUNCIONAL** – suportada pelos diagramas de componentes e instalação.

Um sistema é descrito por visões, cada uma representa uma projecção da descrição completa e mostra características particulares do sistema:

VISÕES DA UML

1. VISÃO USE CASE (CASO DE UTILIZAÇÃO)

- ✓ Visão externa do sistema e suas interacções com o mundo exterior
- ✓ Visão de alto nível de funcionalidade
- ✓ É uma visão central, pois o seu conteúdo é base do desenvolvimento das outras visões do sistema
- ✓ Usa diagramas de Use Case e de Actividade

2. VISÃO LÓGICA

- ✓ Descreve como a funcionalidade do sistema será implementada
- ✓ Em contraste com a visão use case, a visão lógica observa e estuda o sistema internamente
- ✓ Descreve e especifica a estrutura estática do sistema e as colaborações dinâmicas
- ✓ A estrutura estática utiliza diagramas de Classes e de Objectos
- ✓ A modelação dinâmica é efectuada com os diagramas de Estado, Sequência, Colaboração e Actividade

3. VISÃO DE COMPONENTES

- ✓ Descrição da implementação dos módulos e das suas dependência

4. VISÃO DE CONCORRÊNCIA

- ✓ Trata a divisão do sistema em processos e processadores
- ✓ Permite uma melhor utilização do ambiente onde o sistema será implementado, a nível de execuções paralelas e de gestão de eventos assíncronos
- ✓ Suportada por diagramas dinâmicos: Estados, Sequência, Colaboração e Actividade; e por diagramas de implementação: Componentes e Execução

5. VISÃO DE ORGANIZAÇÃO

- ✓ Descreve a organização física do sistema, os computadores, os periféricos e como estes se interligam
- ✓ Suportada pelo diagrama de execução
- ✓ Actividade executada pelos responsáveis pelo desenvolvimento, integração e testes

MODELOS E DIAGRAMAS

Um modelo é uma representação em pequena escala, numa perspectiva particular, de um sistema existente ou a criar.

Atitude de abstracção (omissão de detalhes) fundamental na construção de um modelo.

Modelos são a linguagem por excelência do projectista (designer).

Modelos são veículos para comunicação com vários interessados

Modelos permitem raciocinar acerca do sistema real, sem o chegar a construir.

Um modelo é constituído por um conjunto de diagramas (desenhos) consistentes entre si, acompanhados de descrições textuais dos elementos que aparecem nos vários diagramas.

Um diagrama é uma vista sobre um modelo.

O mesmo elemento (exemplo: classe) pode aparecer em vários diagramas de um modelo.

Ao longo do ciclo de vida de um sistema são construídos vários modelos, sucessivamente refinados e enriquecidos.

DIAGRAMAS DA UML

