



Requisitos de SI's

António Palma dos Reis
Aristides Sousa Mendes
Filipa Pires da Silva
Winnie Picoto



“Um dos maiores desafios com que um gestor se pode deparar é a tarefa de definir as necessidades da empresa para o novo sistema de informação. Na maioria das empresas, esta tarefa aparece uma vez em cada 7-10 anos, o que significa que poucos gestores estão realmente qualificados para a executar. Adicionalmente, o resultado final desta tarefa – o novo sistema de informação – frequentemente determina o sucesso ou fracasso do negócio e não admira que os gestores que se deparam com esta tarefa se sintam aterrorizados.”

Kuiper, 1998



Falhas de Software

- É relativamente frequente depararmo-nos com problemas causados por falhas de software
- Exemplos:
 - Sistema de colocação de professores, no Ministério da Educação
 - Sistema de registo de propriedade, nos notários
 - Sistema de controlo de um equipamento médico – pelo menos dois doentes morreram entre 1985 e 1987 em consequência de terem recebido doses exageradas de radiação
 - Sistema de controlo da distribuição e encaminhamento de bagagem do aeroporto de Denver, nos Estados Unidos, provocou custos superiores a 1 milhão USD por dia



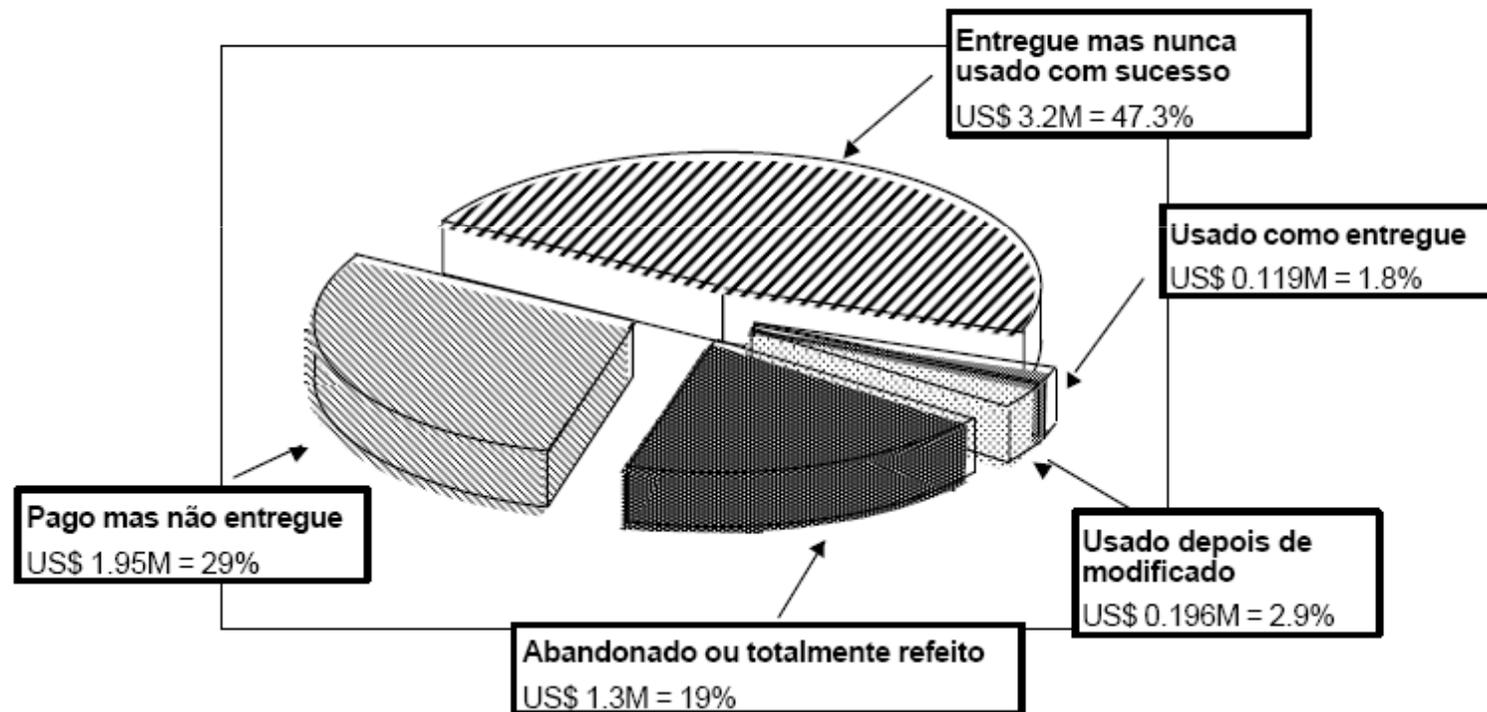
Impacto Económico das Falhas de SW

- Segundo o National Institute of Standards and Technology (NIST) dos EUA, os custos directos dos erros representavam 0,6 % do PIB deste país em 2002, num total de \$ 59,5 Biliões
- Segundo o Standish Group (2002) e Bender as consequências dos erros representavam \$ 293 Biliões.
- O custo da correcção do erro do Ano 2000 poderá ter custado mundialmente um total de \$ 750 Biliões, segundo Bender e Federal Computer Week (2000).
- Observação: vendas de software nos EUA em 2000: \$180 Biliões de dólares



Crise de Software

Resultado do estudo de nove contratos de software, realizado pelo Gabinete de Contabilidade do Governo dos EUA e apresentado ao Congresso, em 1979:

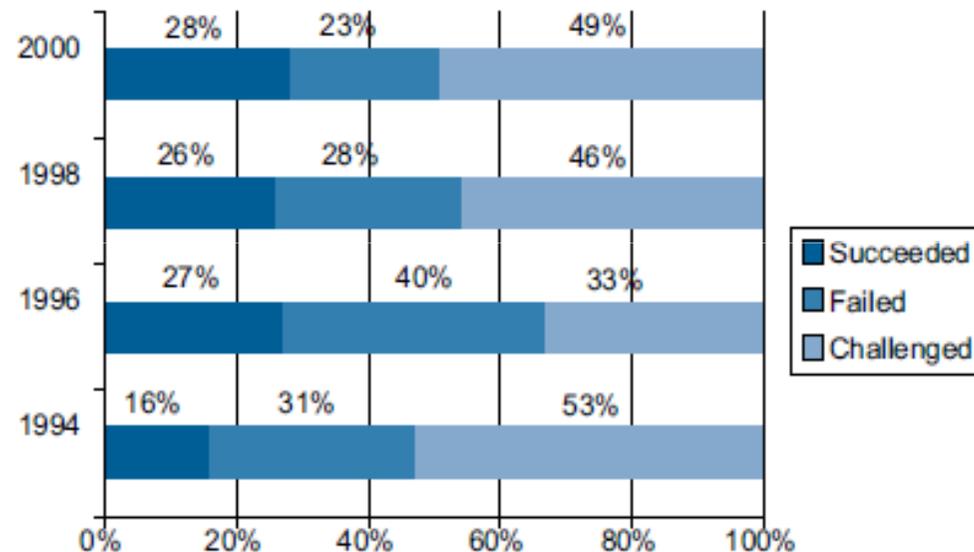




Crise de Software

- Segundo o CHAOS Report (2000):

Project Resolution History (1994–2000)



Apesar da melhoria observada, 137.000 projectos estavam atrasados e/ou acima dos custos estimados, enquanto que 65.000 falharam por completo.



Estatísticas (1)

- Num universo de 34 analistas de sistemas, 70% consideraram que entre 20% e 50% dos projectos falham, porque não são satisfeitos os requisitos de negócio previstos (1987)
- Em 143 projectos, 25% não respondiam aos requisitos (1995)
- Em 365 empresas, 31% dos projectos foram cancelados antes do fim, 53% ultrapassaram os custos; só 12% de 3682 foram completados a tempo e nos custos previstos (1995)
- Em vários milhares de projectos analisados 28% foram bem sucedidos, 23% tiveram diversos tipos de problemas e 49% foram considerados sem sucesso (2005)

...e agora?



Estatísticas (2)

- “Os resultados deste ano mostram uma diminuição no mercado da taxa de sucesso dos projectos, com 32% de todos os projectos a obterem sucesso, o que significa terem sido entregues no tempo previsto, dentro do orçamento e com as funcionalidades requeridas.”
- “44% atrasaram-se, excederam o orçamento e/ou apresentaram menos funcionalidades e características do que as requeridas”
- “24% falharam, o que significa terem sido cancelados antes da sua finalização ou terem sido entregues mas nunca usados.”
- **“Este ano os resultados apresentam a maior taxa de insucesso da ultima década.”**

Standish Group CHAOS Report, 2009





Porquê? (1)

- Segundo o CHAOS Report (2000) as razões pelas quais a maioria dos projectos falha não estão relacionadas com falta de dinheiro ou com a tecnologia. A maioria falha por falta de gestores de projectos qualificados e de suporte executivo
- **“A má gestão pode aumentar os custos e diminuir a produtividade do software mais rapidamente do que qualquer outro factor” [Boehm81]**
- **“Os utilizadores não sabem o que querem até verem e interagirem com algo” [Anónimo]**



Porquê? (2)

- Lista de 10 factores de sucesso (ponderados)

Segundo o CHAOS Report (2000)

The CHAOS Ten

Executive Support	18
User Involvement	16
Experienced Project Manager	14
Clear Business Objectives	12
Minimized Scope	10
Standard Software Infrastructure	8
Firm Basic Requirements	6
Formal Methodology	6
Reliable Estimates	5
Other	5



Apesar dos projectos não necessitarem dos 10 factores para serem bem sucedidos, quanto mais factores estiverem presentes na estratégia do projecto, maior o nível de confiança

Each factor has been weighted according to its influence on a project's success. The more points, the lower the project risk.



Porquê? (3)

- São tipicamente relatadas como causas de falhas de projectos:
 - Requisitos mal definidos
 - Requisitos com rápida evolução
 - Processo mal definido
 - Planeamento não realista
 - Planificação não realista
 - Ferramentas inapropriadas
 - Falta de envolvimento dos utilizadores
 - Incapacidade de detectar problemas nas fases iniciais do desenvolvimento dos SIs
 - Rotação das equipas de desenvolvimento
 - Pessoal não qualificado
 - Falta de mecanismos de controlo quantitativos
 - Bateria de testes insuficiente

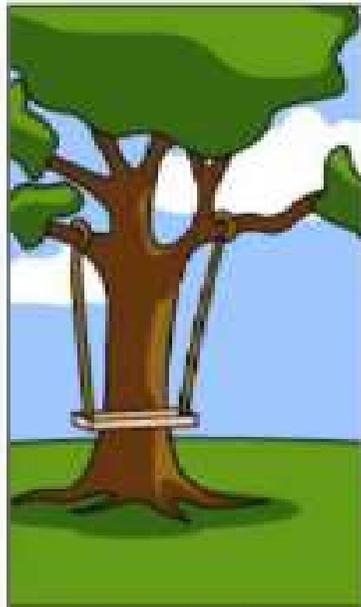


Porquê? (4)

- Muitos dos *stakeholders* não entendem completamente as possibilidades trazidas pelas últimas tecnologias e podem criar falsos pressupostos sobre como um problema pode ser resolvido
- Muitos dos *stakeholders* necessitam de orientação e formação para compreenderem completamente o que é expectável que façam durante o processo e o quais serão os resultados reais



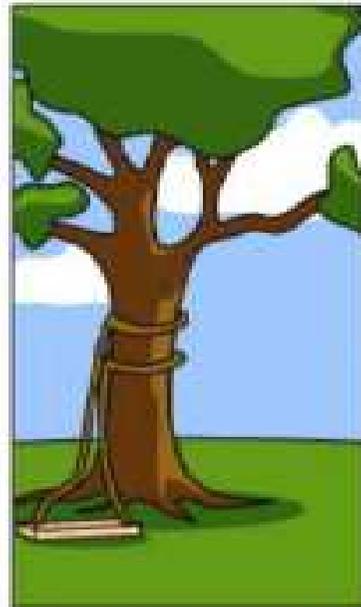
How the customer explained it



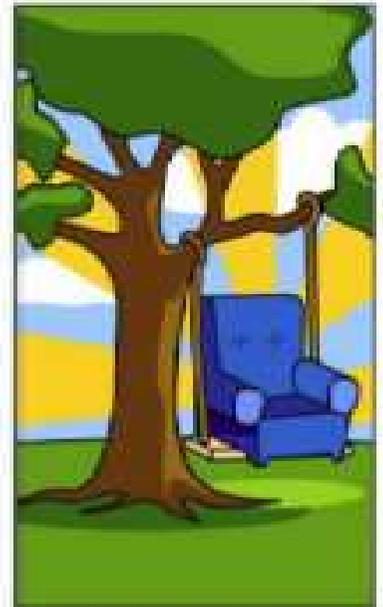
How the Project Leader understood it



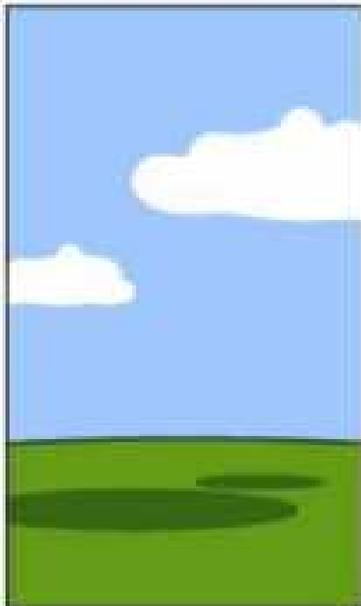
How the Analyst designed it



How the Programmer wrote it



How the Business Consultant described it



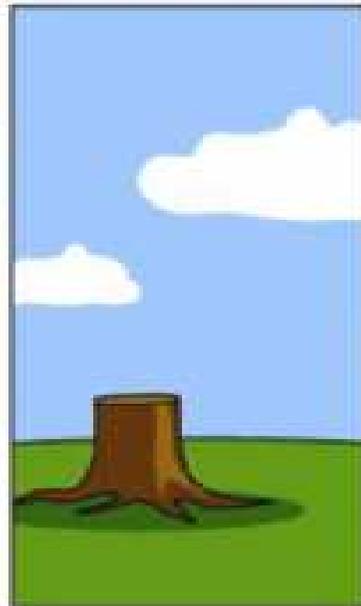
How the project was documented



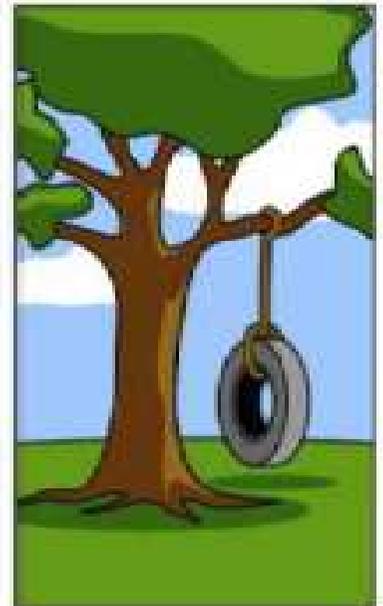
What operations installed



How the customer was billed



How it was supported



What the customer really needed



Os Quatro “P’s” de *Roger Pressman*

- Só **peessoas** (informáticos, gestores e utilizadores) motivadas e comprometidas com o projecto garantem o respectivo sucesso;
- Só um **processo** com técnicas e regras bem definidas permite atingir os objectivos propostos;
- Só se pode produzir um **produto** de qualidade compreendendo as necessidades reais dos utilizadores;
- Só com um **projecto** credível e controlado é possível cumprir prazos e custos propostos.

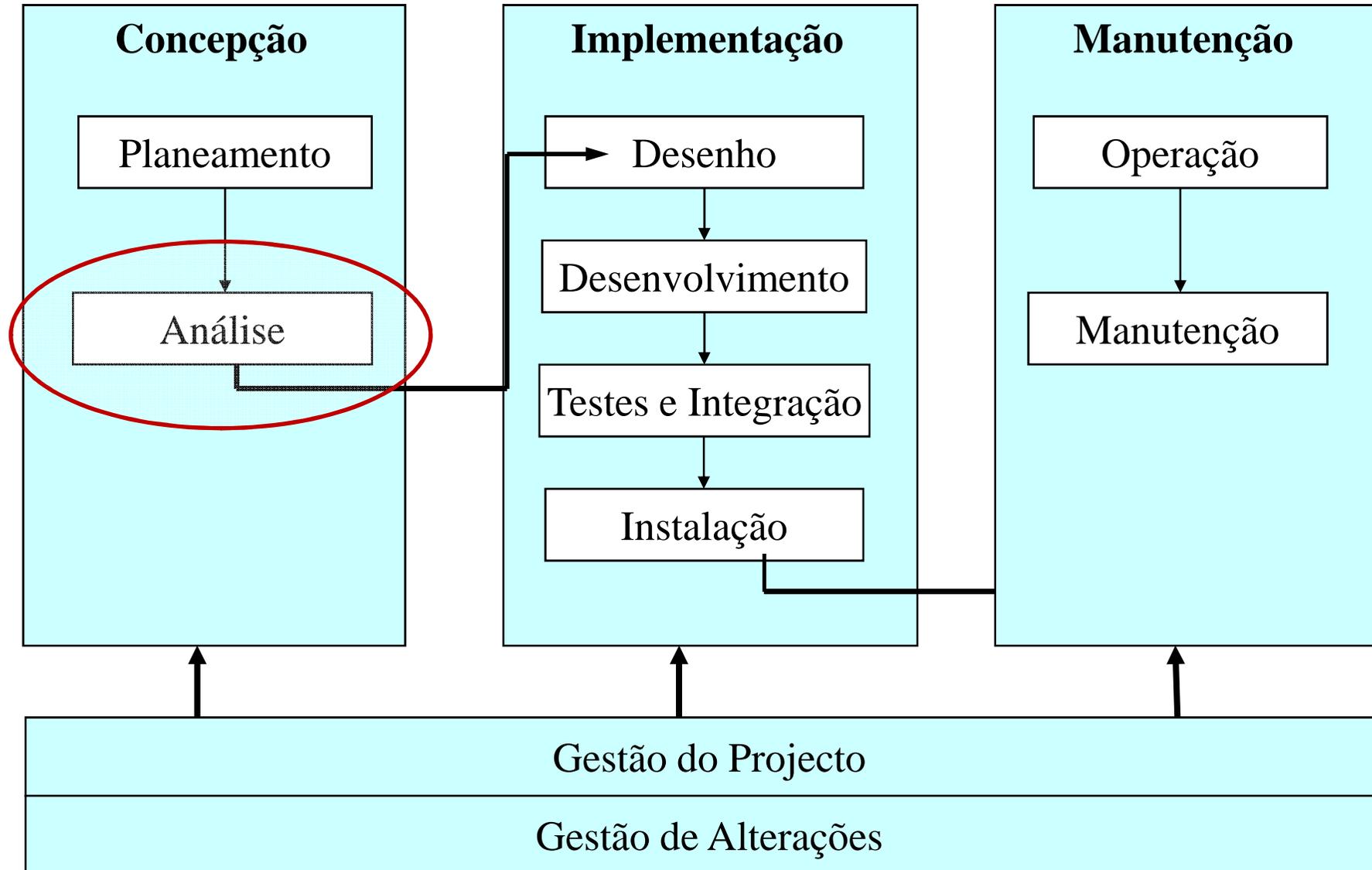


Projectos de SIs e restantes projectos

- Todos os projectos estão sujeitos a restrições de orçamento e tempo
- No entanto, os projectos de SIs apresentam elevadas taxas de insucesso:
 - Entrega atrasada,
 - Não confiáveis,
 - Performance pobre,
 - Custos altos,
 - Nem sempre satisfazem as necessidades para as quais foram criados
- Diferença entre SIs e outras áreas
 - O produto é intangível
 - Não há um processo padrão
 - Projectos de software de grande porte são frequentemente *one-off*
 - Área é relativamente nova e tem mudado o seu papel de destaque



Ciclo de Vida do Software



A fase de captura e análise de requisitos é muitas vezes considerada a mais crítica para a obtenção de sucesso no desenvolvimento de projectos de SIs

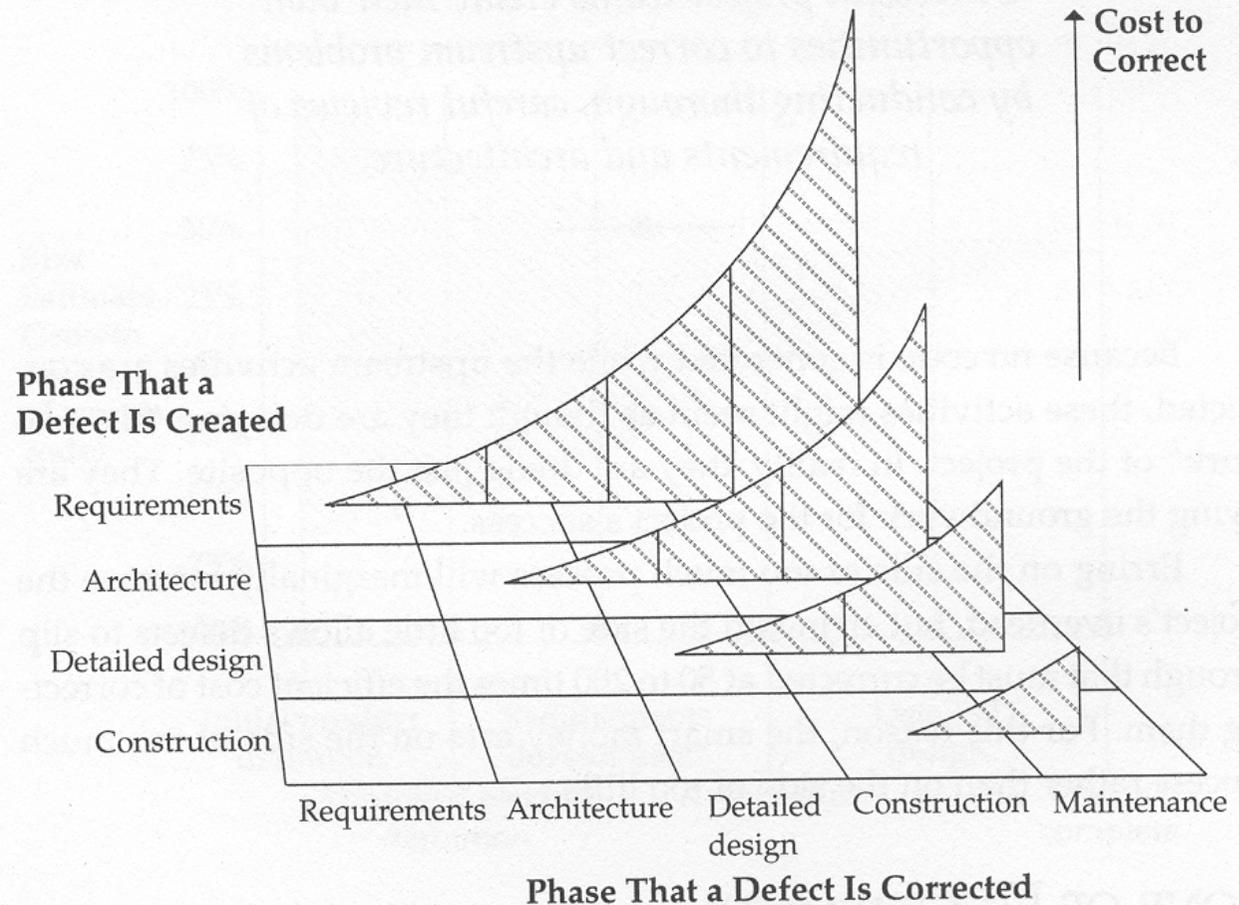


FIGURE 3-5 Increase in defect cost as time between defect creation and defect correction increases. Effective projects practice “phase containment”—the detection and correction of defects in the same phase in which they are created.



O que é um Requisito?

- De forma geral, um requisito é uma propriedade que deve ser exibida de forma a resolver um problema do mundo real
- Os requisitos são afirmações que descrevem os serviços e as restrições de um sistema
- Os requisitos podem ser descritos de forma mais abstracta ou ir até uma especificação matemática detalhada
- Exemplo de descrição abstracta de nível mais alto dos requisitos
 - ”O software deve prover meios de representar e aceder a ficheiros externos criados por outras ferramentas”
 - 1.1 O utilizador deve definir o tipo dos ficheiros a usar.
 - 1.2 Cada ficheiro deve ser representado por um ícone específico, etc.



Características dos Requisitos

- **Identificador:** Cada requisito deve ter um identificador único para permitir a sua gestão e controlo ao longo de todo o ciclo de vida do software
- **Tipo:** Cada requisito deve ter um tipo, o que permite agrupar os requisitos e organizá-los
- **Stakeholder:** Quem é o principal responsável pelo requisito
- **Descrição:** Cada requisito deve ser acompanhado da respectiva descrição
- **Prioridade:** A cada requisito deve ser dada uma prioridade para efeitos de negociação e progressão do sistema
- **Estado:** Cada requisito deve ter um estado, para permitir monitorizar o progresso do projecto
- **Âmbito:** Cada requisito deve ter a indicação da forma como pode afetar a arquitectura do sistema
- **Volatilidade:** Indicação da taxa de alteração esperada durante o ciclo de vida



Problemas com Requisitos

O que é que acontece quando os requisitos estão errados?

Os sistemas tendem a ficar atrasados, não confiáveis e não satisfazem as necessidades dos utilizadores

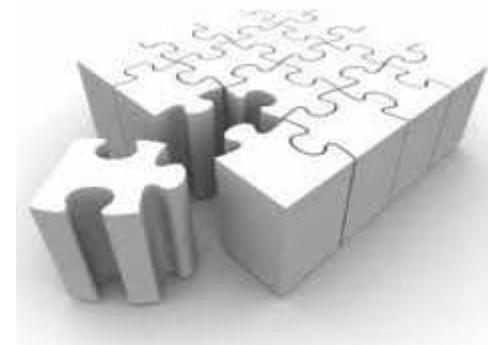
- Os problemas surgem quando os requisitos não são definidos precisamente:

- 
- Requisitos ambíguos
 - Requisitos incompletos e /ou inconsistentes
 - Requisitos não reflectem as necessidades reais do cliente
 - É demasiado caro fazer alterações aos requisitos após terem sido acordados
 - Existem discordâncias entre os clientes, quem desenvolve o sistema, e quem desenvolve os requisitos do sistema



Tipos de Requisitos

- Os requisitos podem ser categorizados em:
 - Requisitos de domínio
 - Requisitos do utilizador
 - Requisitos do sistema
- Dentro desta categorização ainda é possível dividir os requisitos em:
 - Requisitos funcionais
 - Requisitos não funcionais





Requisitos do Domínio

- São requisitos que vêm do domínio da aplicação do sistema e que reflectem as características do domínio
- Podem ser requisitos funcionais ou não funcionais
- Exemplo: requisitos do domínio de um sistema de biblioteca
 - “A interface do utilizador para aceder às bases de dados deve seguir o padrão Z39.50”
- Existem dois problemas principais na identificação destes requisitos:
 - Compreensão - os requisitos são expressos na linguagem do domínio da aplicação, o que não é sempre bem compreendido pelos técnicos
 - Requisitos implícitos - os especialistas do domínio compreendem a área tão bem que não se preocupam em tornar os requisitos do domínio explícitos



Requisitos do utilizador

- Os requisitos do utilizador devem descrever requisitos funcionais e não funcionais de tal forma que sejam compreendidos pelos utilizadores que não tenham conhecimento técnico detalhado
- São normalmente definidos através de linguagem natural, tabelas e diagramas, sempre direccionados aos clientes
- Problemas:
 - Podem estar descritos de forma pouco clara e ambígua
 - Requisitos funcionais e não funcionais são frequentemente confundidos
- Exemplos:
 - O sistema deve ter uma apresentação amigável e intuitiva, com as cores da organização
 - O sistema deve permitir aos utilizadores procurar livros por título e autor



Requisitos do sistema

- Definem o que o sistema deve fazer e com que restrições deve operar
- A sua especificação é mais detalhada do que os requisitos do utilizador
- Servem como base para a fase do desenho do sistema
- São apresentados num documento estruturado com descrições detalhadas dos serviços do sistema expressos através de modelos, descrições em linguagem natural, etc.
- Exemplos
 - O sistema deve manter os registos de todos os materiais existentes na biblioteca, tais como livros, jornais, revistas, etc.
 - O sistema deve permitir aos utilizadores pesquisar um item por título, autor ou ISBN, através da introdução do campo em *text boxes*
 - A interface do utilizador do sistema deve ser implementada usando um browser WWW
 - O sistema deve suportar pelo menos 20 transacções por segundo



Requisitos Funcionais

- Descrevem os serviços que o sistema deve oferecer, como o sistema deve reagir a certas entradas e como o sistema se deve comportar em determinadas situações
- Requisitos funcionais do utilizador podem ser descrições mais abstractas do que o sistema deve fazer, mas os requisitos funcionais do sistema devem descrevê-lo em detalhe
- Exemplos:
 - “O software deve verificar que o estudante satisfaz todos os pré-requisitos antes de se inscrever no curso”
 - “Cada pedido deve ter um identificador único”



Requisitos Não Funcionais

- Definem as propriedades e as restrições do sistema, tais como restrições de tempo de resposta, restrições de segurança, fiabilidade, disponibilidade, etc.
- Exemplos
 - Requisitos de Performance
 - Requisitos de Segurança
 - Requisitos de Fiabilidade
 - ...



Requisitos Não Funcionais–Exemplos

- Requisitos de produto: especificam que o produto se deve comportar de uma determinada forma (e.g. rapidez, fiabilidade)
- Requisitos organizacionais: são consequência das políticas e dos procedimentos organizacionais (e.g. padrões de processo usados)
- Requisitos externos: surgem a partir de factores externos ao sistema (e.g. requisitos legislativos, éticos)



Requisitos - Exercício

Indicar quais dos seguintes requisitos são “funcionais” e quais são “não funcionais”:

O sistema deve...

- Ser acessível a utilizadores Web
- Incluir o logo e esquema de cores da empresa
- Ter acesso restrito a informação sobre lucros
- Tratar informação dos custos actuais e orçamentados
- Fornecer relatórios de gestão
- Imprimir relatório automaticamente com vendas actualizadas pelo menos diariamente



Requisitos – Exercício (continuação)

- Tratar informação sobre todas as empresas subsidiárias
- O sistema deve permitir até N utilizadores trabalhando simultaneamente
- O sistema deve aplicar um desconto aos clientes cujos recibos nos últimos X meses totalizem mais de Y euros
- O sistema devem manter online as Encomendas dos últimos X anos
- O sistema deve ser retirado de serviço periodicamente para efectuação de backups globais



Stakeholders

O que são *stakeholders* do sistema?

É qualquer pessoa que de alguma forma está afectada ao sistema



- Utilizadores:
 - Aqueles que vão utilizar o sistema
 - Normalmente são um grupo bastante heterogéneo composto por pessoas com diferentes funções e requisitos
- Clientes:
 - Aqueles que encomendaram o software ou aqueles que representam o seu mercado alvo
- Analistas de mercado:
 - Na ausência do cliente, os profissionais de marketing são muitas vezes necessários para estabelecer quais são as necessidades do mercado



Stakeholders

- Fornecedores
- Reguladores:
 - Alguns dos domínios das aplicações necessitam de ser regulamentadas por forma a que o software cumpra com as exigências das autoridades reguladoras (e.g. Instituições bancárias)
- Engenheiros de Software:
 - Estão preocupados com aspectos técnicos
- Gestores do Projecto

Não é possível satisfazer perfeitamente todos os requisitos de todos os *stakeholders*

- É necessários haver uma negociação:
 - Considerar os aceites pelos principais *stakeholders*
 - Considerar restrições de orçamento, técnicos, regulamentais e outros
 - Todas as decisões tomadas devem ser formalmente aceites pelo cliente



O processo de Engenharia de Requisitos

O que é o processo de Engenharia de Requisitos?

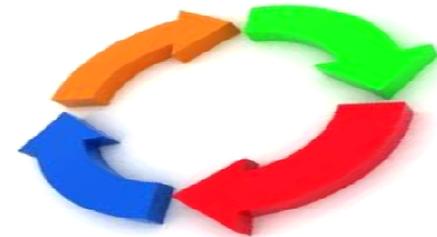
É o conjunto estruturado de actividades que envolvem o desenvolvimento de requisitos do sistema

Não existe um processo ideal

- Variam muito e dependem do domínio da aplicação, das pessoas envolvidas, das necessidades do projecto e da organização

Entretanto, há actividades genéricas comuns a todos os processos:

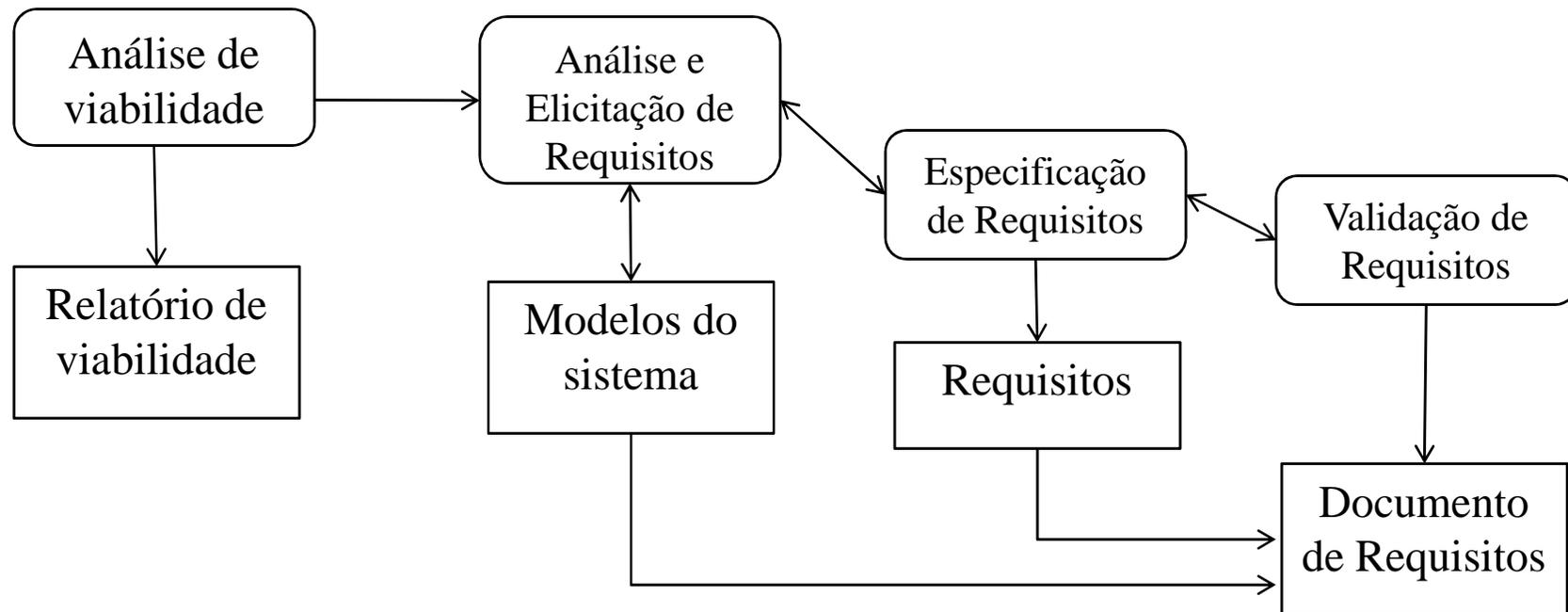
- **Análise de viabilidade**
- **Elicitação de requisitos**
- **Análise de requisitos**
- **Negociação de requisitos**
- **Especificação de requisitos**
- **Validação de requisitos**
- **Gestão de requisitos**





O processo de Engenharia de Requisitos

- Processo genérico de ER





Iteração no processo

- Os requisitos de um sistema evoluem e modificam-se durante o decorrer do projecto
- O processo de ER pode ser considerado como um processo iterativo
 - Cada iteração passa pelas fases descritas até à obtenção de uma especificação final
- Também o desenvolvimento do sistema pode ser feito de forma incremental
 - Em vez de se entregar o sistema de uma só vez, o desenvolvimento e a entrega é feito por incrementos de acordo com as prioridades dos requisitos que foram definidas



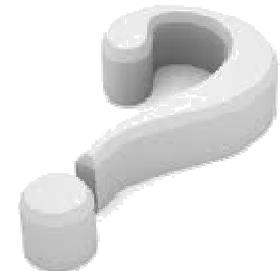
Análise de Viabilidade

- A análise de viabilidade decide se o sistema proposto:
 - contribui para os objetivos da organização
 - pode ser construído utilizando a tecnologia corrente dentro do orçamento
 - pode ser integrado com outros sistemas
- Esta análise pretende estabelecer:
 - Um entendimento base do problema
 - Quem deseja o sistema
 - A natureza da solução que é desejada
 - Uma comunicação e colaboração prévia entre o cliente e quem desenvolve



Análise de Viabilidade - Questões

- Durante análise devem ser feito um conjunto de questões a várias pessoas da organização:
 - E se o sistema não fosse implementado?
 - Quais são os problemas dos processos correntes?
 - Como é que o sistema proposto ajudará?
 - Quais serão os problemas de integração?
 - É precisa nova tecnologia?
 - Quais as facilidades que devem ser suportadas pelo sistema proposto?





W5H2 de Barry Boehm (1996)

- Porque é que o sistema vai ser construído (*Why*)?
- O que vai / deve ser feito (*What*)?
- Quando é que vai ser feito (*When*)?
- Quem é o responsável (*Who*)?
- Onde é que as responsabilidades estão localizadas (*Where*)?
- Como é que vai ser feito (*How*)?
- Quanto vai custar em termos de recursos (*How much*)?



Elicitação de Requisitos

- Pretende capturar os requisitos do sistema
- Envolve pessoal técnico que trabalha com os clientes a fim de investigar o domínio da aplicação, os serviços que o sistema deveria oferecer e as restrições operacionais do sistema
- Envolve clientes, utilizadores finais, gestores, especialistas do domínio, técnicos, etc.



Problemas na fase de Elicitação de Requisitos

- Problemas de Entendimento:
 - *Stakeholders* não sabem o que querem
 - *Stakeholders* omitem informação que acreditam ser “obvia”
 - *Stakeholders* expressam os requisitos nos seus próprios termos
 - *Stakeholders* diferentes podem ter requisitos que entram em conflito
- Problemas de Abrangência:
 - Factores políticos e organizacionais podem influenciar os requisitos do sistema
 - Fronteiras do sistema não estão bem definidas
- Problemas de Volatilidade:
 - Os requisitos mudam durante o processo de análise.
 - Novos *stakeholders* podem surgir e o ambiente do negócio pode mudar



Análise de Requisitos

- Expande e corrige a informação obtida nas fases anteriores
 - Faz a ponte entre a descrição do sistema e os modelos de desenho
- Foca-se no desenvolvimento de modelos técnicos, explorando funcionalidades, características e restrições do sistema futuro
- O seu objectivo é ajudar a entender o problema (*What?*) e não iniciar o desenho da solução (*how?*)
- Nesta fase são desenvolvidos vários modelos, podendo ser usadas várias notações de modelação



Negociação de Requisitos

- É frequente que diferentes *stakeholders* proponham requisitos que entram em conflito
- É necessário reconciliar esses conflitos através de um processo de negociação
- Clientes, utilizadores e outros *stakeholders* devem ordenar os requisitos consoante a prioridade que lhes atribuem
- Segue-se um processo iterativo que pretende eliminar, combinar e/ou modificar os requisitos de forma a conseguir um resultado “*win-win*”



Especificação de Requisitos

- É resultante da fase de análise de requisitos
- Consiste numa combinação complexa de requisitos a partir de:
 - Diferentes pessoas de diferentes níveis da organização
 - Ambiente no qual o sistema vai operar
- Dá origem a um documento que deve ser sistematicamente revisto, avaliado e aprovado



Documento de Requisitos

O que é um documento de requisitos?

É a declaração formal dos requisitos do sistema, sob a forma de um documento, que deve ser usado para comunicar os requisitos aos clientes, gestores e técnicos

- Deve incluir a definição e descrição dos requisitos
- Deve dizer “O QUÊ” que o sistema deve fazer e não “COMO” é que vai ser feito



Documento de Requisitos

- O documento de requisitos deve:
 - Descrever os serviços e funções que o sistema deve fornecer especificar o comportamento externo do sistema
 - Descrever as restrições sob as quais o sistema deve operar
 - Descrever o conjunto global de propriedades que o sistema deve apresentar
 - Fornecer definições de outros sistemas com os quais o sistema deverá ser integrado
 - Ser fácil de pesquisar e modificar
 - Servir de referência para as restantes fases do processo de desenvolvimento
 - Prever alterações



Escrever Requisitos

- Os requisitos são mais frequentemente lidos do que escritos
 - Deve ser investido tempo para escrever requisitos que sejam perceptíveis e de fácil leitura
- Normalmente os requisitos são escritos em parágrafos de texto em linguagem natural (LN) anexando diagramas e equações
 - Existem alternativas à LN, tais como, linguagens de descrição de desenho, notações gráficas, especificações matemáticas,...
- Não se deve assumir que todos os leitores têm os mesmos conhecimentos e usam a mesma terminologia
- Deve ser disponibilizado tempo para revisão e reformulação do documento



Escrever Requisitos

- Deve ser definido um formato padrão para descrever todos os requisitos
- A linguagem deve ser usada de forma consistente, clara e o menos ambígua possível
- As partes mais importantes do texto devem ser destacadas
- Devem ser evitados termos demasiado técnicos
- Os requisitos devem ser descritos de forma a que possam ser:
 - Validados (por todos os *stakeholders*)
 - Verificados no desenvolvimento do sistema
 - Estimados os custos (pelos gestores do projecto)
- Exemplo de requisitos:
 - “O sistema deve ser confiável” – Descrito de forma vaga e subjectiva
 - “O sistema de um call center tem de aumentar o rendimento do centro em 20%” – Descrito de forma quantificada



Fontes de Informação

- Conhecimento do domínio
- Ambiente organizacional
 - Deve ter-se especial atenção às estruturas culturais e/ou políticas da organização cliente
- Ambiente Operacional
 - Os requisitos também têm origem no ambiente no qual o sistema vai operar
 - Estes requisitos afectam a viabilidade do sistema, bem como os custos e as escolhas de desenho
 - e.g. restrições de tempo de sistemas em tempo real ou restrições de interoperacionalidade com os sistemas envolventes



Técnicas

- Entrevistas
- Reuniões acompanhadas
- Cenários
- Observação (Etnografia)
- Prototipagem
- Objectivos ou *Goals*
- *ViewPoints*
- ...



Entrevistas

- Requerem empatia de ambas as partes
- O entrevistado pode esconder aspectos problemáticos
- É uma boa prática anotar e assinar os resultados da entrevista





Reuniões Acompanhadas

- A ideia subjacente é que um grupo de pessoas pode trazer mais conhecimento sobre os requisitos do sistema do que as mesmas a trabalhar individualmente
- Podem ser debatidas e aperfeiçoadas ideias que poderiam ser difíceis de encontrar por meio de entrevistas
- Permitem reconhecer quando há um conflito
- Normalmente requerem um facilitador





Cenários

- O que é um cenário?
- Cenários são descrições de como o sistema é, ou deve vir a ser, usado na prática.
- Descrevem interacções concretas entre objectos do sistema → bom para os estágios iniciais do desenvolvimento, e para a comunicação com os *stakeholders*
- Permitem responder a questões como “e se?” e “como é que isto se faz?”



Descrição de cenários

- O tipo de cenário mais comum é o Caso de Utilização (UML)
- Descrição de cenários deve conter:
 - O estado do sistema no início do cenário
 - O fluxo normal de eventos no cenário
 - O que pode falhar e como é tratado
 - Outras actividades concorrentes
 - O estado do sistema no fim do cenário



Descrição Estruturada

Efectuar Encomenda Internet (Cenário Principal)

Pré-condição:	O cliente é um utilizador válido no sistema
Descrição	<ol style="list-style-type: none">1. O use case começa quando o cliente selecciona a opção de Encomendar2. Em simultâneo com a sua encomenda é mostrado o catálogo de produtos3. O cliente adiciona produtos à encomenda através da introdução do código do produto4. Automaticamente, o sistema mostra o nome, descrição e preço do produto5. De cada vez que é adicionado um produto, o valor total da encomenda é calculado6. O cliente confirma a sua encomenda através da opção Confirmar7. O sistema pede os detalhes do cartão de crédito8. O sistema confirma os dados do pagamento e atribui um número de identificação à encomenda
Pós-condição:	A encomenda será entregue na morada do cliente



Descrição Estruturada

Efectuar Encomenda Internet (Cenários Secundários)

Pré-condição:	O cliente é um utilizador válido no sistema
Descrição	<ol style="list-style-type: none">1. O use case começa quando o cliente selecciona a opção de Encomendar2. Em simultâneo com a sua encomenda é mostrado o catálogo de produtos3. O cliente adiciona produtos à encomenda através da introdução do código do produto<ol style="list-style-type: none">a) Se um código é inválido o sistema avisa o cliente com uma mensagem4. Automaticamente, o sistema mostra o nome, descrição e preço do produto5. De cada vez que é adicionado um produto, o valor total da encomenda é calculado6. O cliente confirma a sua encomenda através da opção Confirmar7. O sistema pede os detalhes do cartão de crédito8. O sistema confirma os dados do pagamento e atribui um número de identificação à encomenda<ol style="list-style-type: none">a) Se o cartão for inválido, o sistema avisa o cliente através de uma mensagem, voltando em seguida ao passo 7
Caminhos Alternativos	A qualquer momento o cliente pode cancelar a sua encomenda, pressionando no botão Cancelar
Pós-condição:	A encomenda será entregue na morada do cliente



Prototipagem



- É o rápido desenvolvimento de um sistema para validar requisitos
- Na elicitação de requisitos os utilizadores podem experimentar o protótipo para verem de que forma o sistema suporta o seu trabalho
- Na validação de requisitos o protótipo pode revelar erros e omissões nos requisitos
- A prototipagem pode reduzir o risco de erros dos requisitos
- Existem dois tipos de Prototipagem:
 - *Evolutionary*
 - O protótipo vai sendo refinado através de um conjunto de iterações até ao sistema final
 - *Throw-away*
 - O protótipo consiste na produção de uma implementação do sistema para ajudar a descobrir e validar os requisitos que depois é descartado



Outras

- Etnografia
 - Técnica onde o analista gasta um tempo considerável no ambiente de trabalho a observar e a anotar como os participantes envolvidos trabalham
- Abordagem orientada a objetivos (*Goals*)
 - *Goals* são as propriedades desejadas do sistema que são expressas pelos intervenientes no mesmo, podendo ser formulados a diferentes níveis de abstracção (alto nível - preocupações de estratégia; baixo nível - preocupações técnicas)
- Elicitação orientada a Viewpoints
 - Viewpoints são uma forma natural para estruturar a elicitação de requisitos. Nesta abordagem, os *stakeholders* descrevem diferentes formas de olhar para o problema (*Pontos de vista*)



Validação de Requisitos



- Examina a especificação para garantir que :
 - os requisitos definem realmente o que o cliente realmente quer
 - todos os requisitos foram descritos de forma não ambígua,
 - que foram detectadas e corrigidas inconsistências, omissões e erros
 - está de acordo com os standards estabelecidos para o processo, o projecto e o produto
- Erros nos requisitos custam caro, portanto a sua validação é importante
 - corrigir um erro de requisitos = 100 x corrigir erro de implementação



Como validar requisitos?

- Para validar requisitos devemos verificar:
 - Validade - O sistema fornece as funções que satisfazem as necessidades do cliente?
 - Consistência – Existem incompatibilidades e/ou conflitos de requisitos?
 - Completude - Todas as funcionalidades requeridas pelo cliente estão incluídas?
 - Viabilidade - Os requisitos podem ser implementados com o orçamento e tecnologia disponíveis?
 - Avaliação - Os requisitos podem ser verificados/avaliados?



Técnicas de validação

- Revisões de requisitos
 - Análise sistemática dos requisitos, que consiste em responder a questões sobre os requisitos especificados
- Prototipagem
 - Utilização de um modelo executável do sistema
- Geração de testes de aceitação
 - Desenvolver testes para verificar os requisitos
 - Pode ser difícil para os requisitos não funcionais
- Validação do modelo
 - Podem ser realizadas análises estáticas para validar o modelo
- Análise automatizada da consistência
 - Verificar a consistência de uma descrição de requisitos estruturada



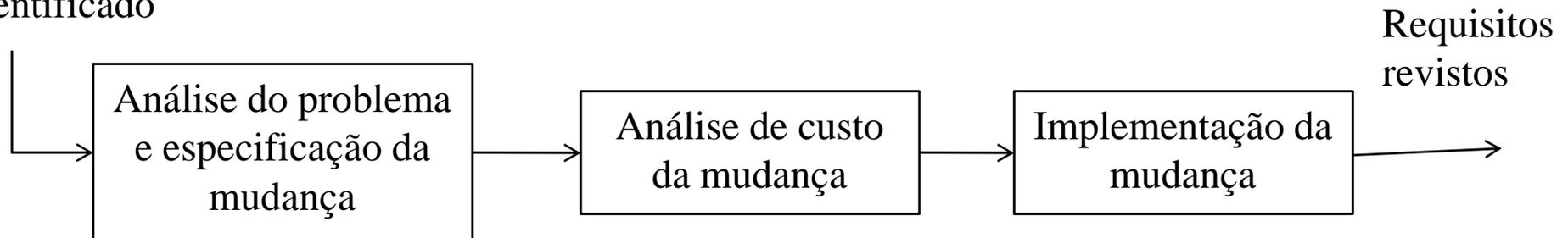
Gestão de Requisitos

O que é a gestão de requisitos?

É o processo que envolve a gestão de alterações dos requisitos

- É o conjunto de actividades que ajudam a equipa do projecto a identificar, controlar, e acompanhar os requisitos e alterações aos requisitos durante todo o projecto

Problema
identificado





Volatilidade

- Os requisitos **vão mudar!**
- Requisitos são inevitavelmente incompletos e inconsistentes
 - Novos requisitos surgem durante o processo originados pela alteração das necessidades do negócio e pela alteração da compreensão dos requisitos
- Mudança de requisitos
 - A prioridade dos requisitos pode mudar durante o processo
 - As regras do negócio podem sofrer alterações durante o desenvolvimento
- O contrato deve identificar claramente como lidar com a volatilidade dos requisitos
- O Acompanhamento (*Traceability*) dos requisitos é fundamental



Requisitos Permanentes e Voláteis

- Requisitos Permanentes
 - Estáveis. Derivam da actividade principal da organização
 - Ex: Requisitos de um sistema hospitalar: médicos, pacientes, etc.
- Requisitos Voláteis
 - Mudança provável durante ou depois do desenvolvimento
 - Ex: Requisitos dos serviços bancários
- Tipos de requisitos voláteis:
 - Mutáveis: mudanças no ambiente
 - Emergentes: durante o desenvolvimento
 - Consequentes: resulta da introdução do sistema de computador
 - Compatibilidade: depende de sistemas ou processos particulares dentro da organização





Ferramentas de Apoio

- Alguns exemplos:
 - CaliberRM da Borland
 - <http://www.borland.com/us/products/caliber/index.html>
 - Enterprise Architect da Spark Systems
 - <http://www.sparxsystems.com.au/>
 - Rational RequisitePro da IBM
 - <http://www-01.ibm.com/software/awdtools/reqpro/>
 - Micro Focus Optimal Trace
 - <http://www.microfocus.com/products/OptimalTrace/index.asp>