

O Software:
um produto e um processo



Aula 2

Sumário

O produto

- ▣ O software como produto
- ▣ Evolução
- ▣ Características
- ▣ Tipos
- ▣ Aplicações
- ▣ Atributos de um bom software

O processo

- ▣ Definição de processo, método e ferramenta
- ▣ Visão geral da Engenharia do Software
- ▣ O processo do software
- ▣ Modelos de processo
 - ▣ Linear sequencial
 - ▣ Prototipos
 - ▣ DRA
 - ▣ Modelos evolutivos

Produto e processo

20-09-2004

Aula 2 - SW: produto e processo

2

O produto

- # O software é simultaneamente um produto
 - software de gestão
 - pacotes de uso geral (folhas de cálculo, processadores de palavras)
- # e um veículo para o entregar
 - sistemas operativos,
 - software de comunicação
 - Ambientes de desenvolvimento
 - Ferramentas CASE

Evolução do software

Antes

- Hardware
 - Caro
 - Poder Computacional Limitado
- Programas simples

Preocupação:

- aproveitamento dos recursos de hardware i.e. Eficiência do produto

Depois

- Hardware
 - mais barato
 - e poderoso
- Software mais complexo

Preocupação:

- Satisfação dos requisitos do utilizador (produto eficaz)
- Baixar custos e tempo de desenvolvimento (processo eficiente)

Características do software

- # O software é um produto lógico, não físico e portanto,
 - não é fabricado num sentido clássico, senão desenvolvido
 - O software não se estraga
 - Embora cada vez o software é uma assemblagem de componentes, ainda se faz muito software à medida

Tipos de software

- # Genérico
- # Feito à medida
- # Problemas do desenvolvimento são os mesmos
- # A diferencia reside na especificação dos requisitos

Aplicações do software

- # Software de sistemas
- # Software de tempo real
- # Software de gestão
- # Software de engenharia e científico
- # Software embebido
- # Software de PC's
- # Software baseado na WEB
- # Software de IA

20-09-2004

Aula 2 - SW: produto e processo

7

Software de sistemas: servem outros programas. Exemplos: compiladores, editores y utilidades de gestão de ficheiros. Forte interacção com o hardware, múltiplos utilizadores, operação concorrente que requer planeamento e partilha de recursos.

Software de tempo real: coordena, analisa e controla factos do mundo real assim que ocorrem. Incluem componentes de aquisição de dados que recolhem e formatam a informação recebida, componentes de análise que transformam a informação segundo requerido, componente de controlo/saída que responde ao entorno externo e uma componente de monitorização que coordena as componentes restantes.

Software de gestão: suportam as operações de contabilidade, salários, recursos humanos, inventários. Transformam os dados recebidos para ajudar na tomada de decisões

Software de engenharia e científico: alta manipulação numérica. Aplicações vão desde a vulcanologia até a astronomia, os sistemas CAD são outro exemplo

Software embebido: em ROM, controlam produtos médicos, comerciais e industriais. Funções muito limitadas mas muito especializadas

Software de PC's: processadores de palavras, folhas de cálculo, pacotes gráficos, etc, etc.

Software baseado na WEB: paginas HTML com cgi, perl, ou java applets

Sistemas de inteligência artificial: algoritmos baseados na lógica

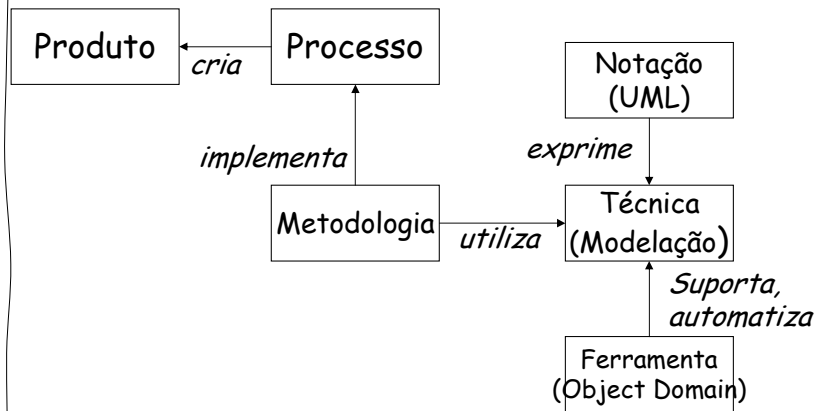
Atributos de um bom software

<i>Atributo</i>	<i>Descrição</i>
<i>Maintainability</i>	O software deve ser capaz de evoluir perante as mudanças nos requisitos
<i>Dependability</i>	Características como Confiabilidade e segurança
<i>Eficiência</i>	Aproveitamento dos recursos de hardware
<i>Usabilidade</i>	Apropriadas interface do utilizador e documentação

Estrutura de conceitos (I)

- # A criação de um *produto* de software implica um *processo*
- # Processo: Conjunto de actividades e resultados associados que produzem um produto de software
- # Metodologia: descreve um processo incluindo a utilização de um conjunto de ferramentas, técnicas e notações específicas (Unified Software Development Process)
- # Os resultados associados são especificados em termos de *modelos*
- # Modelo: é uma interpretação da realidade exprimido através de uma linguagem textual ou gráfica.
- # Técnicas, Notações e Ferramentas: A modelação é uma técnica, o UML é uma notação e o Object Domain, uma ferramenta

Estrutura de conceitos (II)

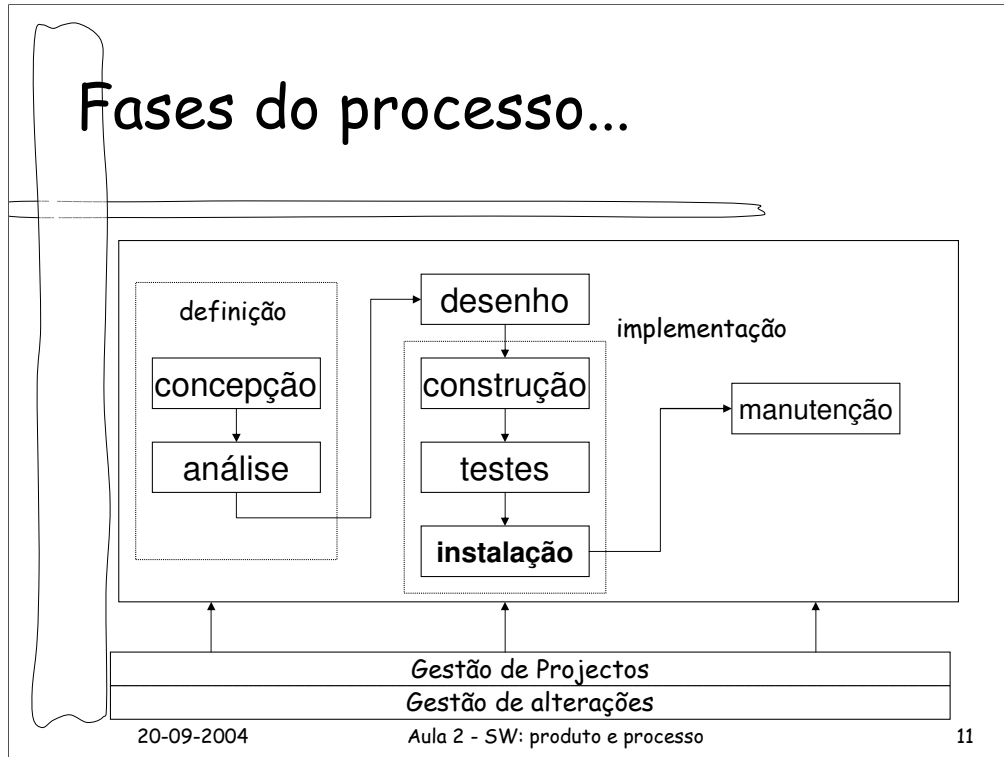


20-09-2004

Aula 2 - SW: produto e processo

10

Fases do processo...



Concepção: define-se de forma geral o que deve fazer o sistema (funcionalidades e informação a manusear), também inclui o desenvolvimento do business case para o sistema de informação. Os clientes são o grupo de interesse de maior peso na determinação dos parâmetros para o business case. Em tal business case, o IS é avaliado estrategicamente em termos da contribuição ao rendimento organizacional. Uma avaliação da sua factibilidade também é feita em função das restrições e recursos organizacionais, e uma estimativa do grau de risco associado ao projecto.

Análise: documentação do funcionamento de sistemas existentes e determinação dos requisitos do novo sistema. Exprime o que deve fazer o novo sistema. A análise envolve 2 actividades interligadas; extracção de requisitos e especificação de requisitos e deve envolver tanto pessoal de informática como representantes dos utilizadores

Desenho: planeamento da forma do sistema de informação novo considerando a tecnologia a ser usada (sw, hw, redes, bd). Exprime como deve funcionar o novo sistema. Esta fase também devia envolver participação de representantes dos utilizadores

Construção: implica quer actividades de programação –subcontratadas ou com pessoal da casa-, quer a compra e configuração de pacotes e alguns testes (de unidade)

Testes: actividades verificação de testes de integração e aceitação

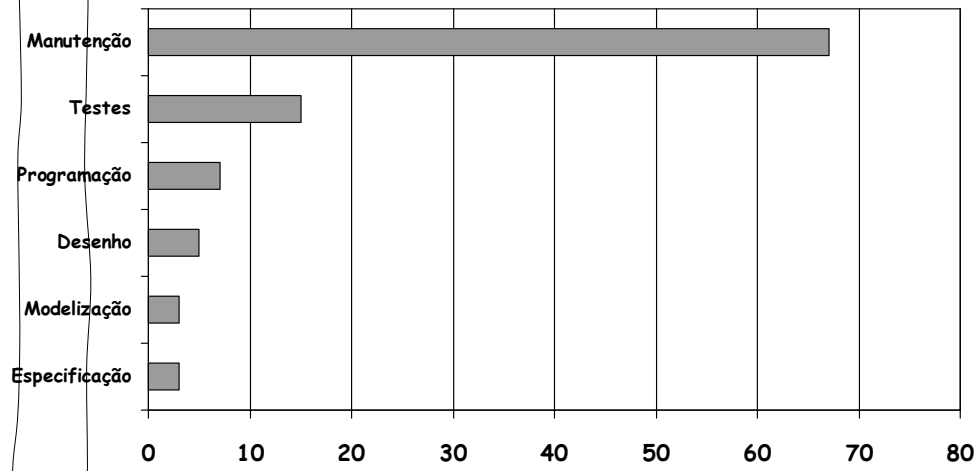
Instalação: disponibilização do sistema no seu contexto de uso.

Manutenção: processo de feedback que envolve mudanças ao sistemas. Os sistemas de informação raramente permanecem na sua versão original. Os SI sofrem mudanças pelas seguintes razões: erros na construção, mudanças na organização, erros na especificação, de requisitos, mudanças tecnológicas

Gestão de Projectos: Conjunto de actividades relacionadas com a gestão dos recursos humanos, financeiros e o controlo dos prazos de execução. Implica actividades de planeamento, controlo e monitorização e medidas de correcção. Funciona como a interface oficial da equipa de projecto.

Gestão de Alterações: O desenvolvimento de software implica a elaboração de produtos interligados entre si (modelos). Portanto, é que estejam previstos mecanismos de controlo das alterações feitas nos diversos produtos do projecto ao longo do processo de desenvolvimento.

Custos relativos do processo..



20-09-2004

Aula 2 - SW: produto e processo

12

Uma visão diferente... do problema à solução

<i>Actividade</i>	<i>Determina...</i>
<i>Concepção</i>	O enunciado do Problema
<i>Análise</i>	O domínio do problema
<i>Desenho</i>	O domínio da Solução
<i>Construção, Testes, Instalação</i>	O domínio da realização

20-09-2004

Aula 2 - SW: produto e processo

13

Modelos de processo

Definição:

- Estratégia de desenvolvimento, especialmente a ordem de execução das fases do processo

Tipos:

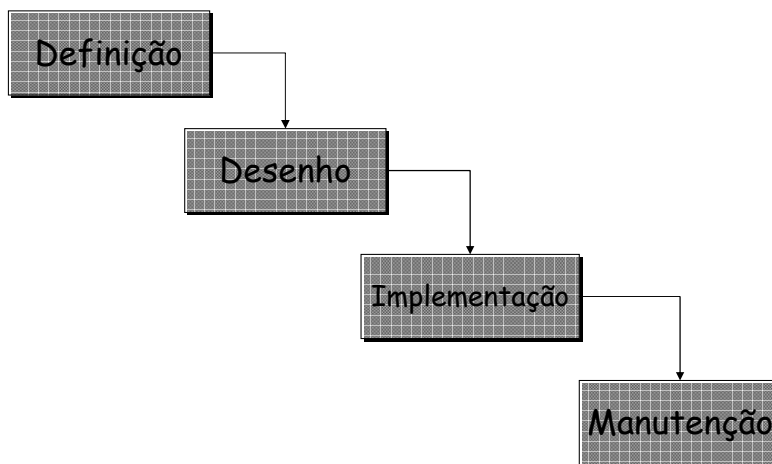
- Modelos em cascata
 - Sequencial
 - Revisto
- Construção de Protótipos
- Desenvolvimento Rápido de Aplicações
- Modelos evolutivos
 - Incrementais
 - Iterativos
 - Iterativos e incrementais
 - Espiral

20-09-2004

Aula 2 - SW: produto e processo

14

Modelo Sequencial



20-09-2004

Aula 2 - SW: produto e processo

15

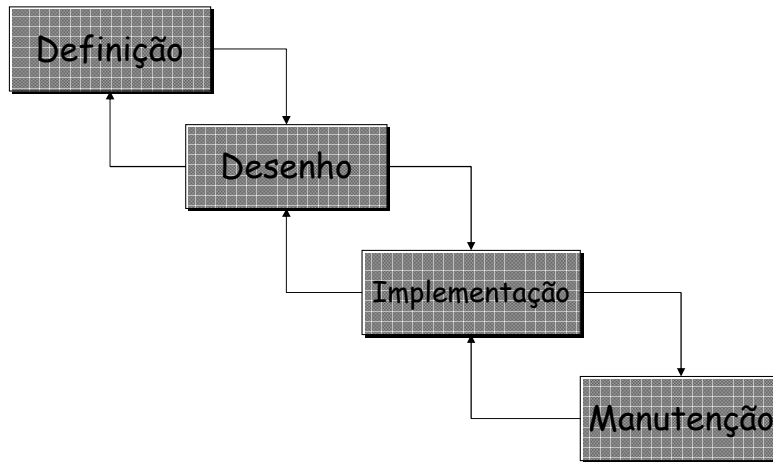
Também denominado por ciclo de vida clássico, neste modelo as actividades são executadas sequencialmente e uma actividade é iniciada só quando a anterior tiver terminado. Este modelo é muito simples de gerir mas baseia-se no suposto que o cliente sabe muito bem o que quer o qual é normalmente difícil. No caso de ser um desenvolvimento contratado a uma empresa, esta tem prova da aceitação do cliente das entregas intermédios. Esta salvaguarda não evita que o projecto não obtenha os resultados pretendidos e que o cliente fique desapontado.

Vantagens:

- Serve para sistemas sistemas
- Fornece um template do desenvolvimento
- Muito utilizado => Muita experiência
- É melhor que o método NIKE (just do it!)

O grande problema deste modelo é que a realidade raras vezes acompanha o processo. Os problemas são detectados tardiamente. Outras limitações são a compartimentação de esforços que promove uma alta especialização e desencoraja a comunicação e partilha de visões entre todos os participantes do projecto, por ex. analistas, desenhadores, programadores e utilizadores. Minimiza portanto o impacto da compreensão adquirida no decurso do projecto.. Os câmbios que igual surgem para fases já realizadas ocasionam confusão.

Modelo Revisto



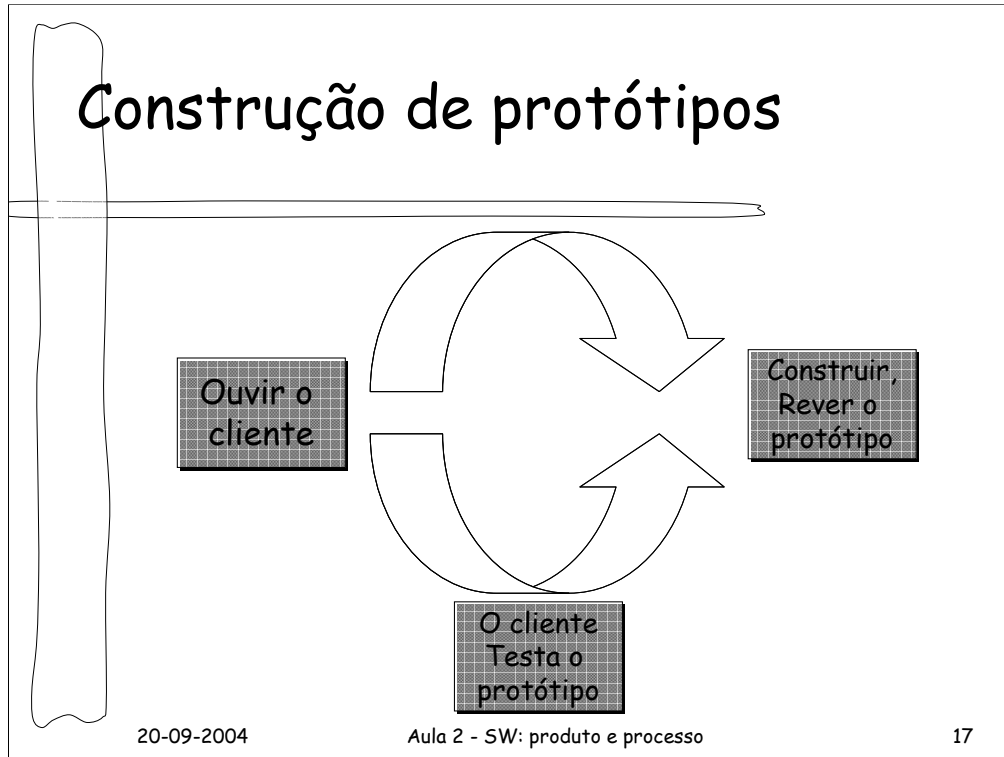
20-09-2004

Aula 2 - SW: produto e processo

16

O modelo revisto pretende resolver os problemas do modelo sequencial, prevendo a hipótese de poder regressar à fases anteriores desde qualquer fase. Isto reduz a confusão causada pelos inevitáveis câmbios surgidos. Este modelo é muito mais difícil de gerir já que em ausência de um estrito controlo das alterações, corre-se o risco de passarmos o tempo num ciclo sem fim, sem nunca chegar a um produto final.

Ambos modelos em cascata apresentam o problema da extensão do período de tempo que decorre entre o início do projecto e a entrega do produto final.



Em situações de incerteza sobre os requisitos, o paradigma de construção de protótipos pode ser a melhor alternativa. Este paradigma começa com a recolha de requisitos. O analista e o cliente falam sobre os objectivos globais do software e os requisitos conhecidos. Então aparece um desenho rápido focado na representação dos aspectos do software visíveis ao utilizador. O desenho rápido passa-se à construção do protótipo, que é avaliado pelo utilizador e é utilizado para a refinação de requisitos.

Na maioria dos casos, o protótipo não serve como produto operacional. Pode ser muito lento, pesado de usar ou grande

Vantagens:

- O tempo entre entregas é curto
- Serve como 1er sistema
- Facilita a aceitação do sistema
- Ideal quando os requisitos não estão claros, é de facto um método de identificação de requisitos
- Os requisitos são verificados atempadamente

Desvantagens:

Falsas expectativas geradas pelo protótipo quanto a sua funcionalidade e ao tempo requerido para as incorporar que propícios para o estabelecimento de compromissos para por funcionalidades no protótipo muitas vezes desenvolvido em sistemas operativos e linguagens inadequadas para o sistema final. Para solucionar isto, deve falar-se muito claro com o cliente.

A gestão é mais demorada

Desenvolvimento rápido de aplicações



Desenvolvimento com o ciclo clássico de vida muito curto através da limitação do âmbito do sistema e utilizando uma construção baseada em componentes. Abrange as fases seguintes:

Modelado da gestão: Qual informação conduz o processo? Qual informação é gerada? Aonde vai a informação? Quem a processa?

Modelado dos dados: conjunto de objectos de dados necessários para o sistema. Se definem as suas características e ligações

Modelado do processo: descrição das funções para inserir, modificar, apagar ou recuperar dados

Geração de aplicações: assume-se a utilização de técnicas e ferramentas de 4ª geração baseadas na reutilização de componentes ou a criação de componentes reutilizáveis.

Testes e entrega: Testes dos componentes novos e as interfaces e testes de integração

Vantagens:

Tempo de entrega

Redução dos testes

Apropriado para software de gestão ou com funcionalidades vulgares

Desvantagens:

Requer muitos recursos humanos e muito comprometidos com o projecto

Clientes e programadores comprometidos com os rápidos tempos de entrega

Não todos os tipos de aplicações são apropriado. O sistema deve poder ser decomposto em módulos

Não apropriado aquando os riscos técnicos são altos

Modelo iterativo

Ideia:

- Desenvolvimento através de iterações sucessivas do ciclo de vida clássico
- Refinação progressiva do sistema
- O âmbito é definido no início e o não é alterado

Vantagens:

- Identificação atempada dos maiores riscos e dúvidas
- Encoraja a participação activa dos utilizadores
- Identificação atempada de inconsistência entre a análise, o desenho e a implementação
- Distribuição da participação dos envolvidos ao longo do projecto
- Aprendizagem e melhora continua do processo
- Provas inequívocas do avanço do projecto

20-09-2004

Aula 2 - SW: produto e processo

19

Ideia:

refinar o sistema progressivamente aumentando o seu nível de detalhe e sofisticação

o âmbito do sistema não é alterado

Vantagens:

Os maiores riscos e dúvidas são identificados nas primeiras iterações

Encoraja a participação activa dos utilizadores

As inconsistências entre a análise, o desenho e a implementação são identificadas atempadamente

O esforço dos diversos participantes é distribuído equitativamente ao longo do projecto

A equipa pode aprender das iterações anteriores e melhorar continuamente o processo

Pode ser mostrado inequivocamente a todos os envolvidos e interessados o respectivo avanço

Desvantagens

Gestão mais demorada

Modelo incremental

Ideia:

- Desenvolvimento através de incrementos sucessivas do âmbito do sistema
- O sistema é alargado progressivamente

Vantagens:

- Esta abordagem é útil para
 - Problemas complexos
 - Recursos humanos insuficientes
 - Datas de entrega inflexíveis

20-09-2004

Aula 2 - SW: produto e processo

20

Idea:

Alargamento progressivo do âmbito do sistema

Vantagens:

Quando os recursos são escasos

Quando as datas de entrega são inflexíveis

Modelo iterativo e incremental

Ideia:

- Combina os 2 métodos anteriores
- Envolve uma refinação e alargamento sucessivo do sistema da sua qualidade, detalhe e âmbito

Vantagens:

- A maturação e completação progressiva dos produtos
- Cada iteração-incremento produz um conjunto de produtos utilizáveis

20-09-2004

Aula 2 - SW: produto e processo

21

Vantagens adicionais:

Avaliação atempada de riscos e pontos críticos

Identificação atempada de medidas para os controlar ou até eliminar

Definição de uma arquitectura que melhor possa orientar todo o desenvolvimento

Disponibilização natural de um conjunto de regras para controlar os inevitáveis pedidos de alterações futuras

Trabalho em grupo mais eficaz pela interacção e partilha de comunicação resultante

A diferencia entre o modelo incremental e a construção de prototipos reside na entrega de produtos operacionais com cada incremento, enquanto o prototipo é uma casca que pode ser deitada fora. Os primeiros incrementos são versões incompletas mas que fornecem funcionalidades requeridas.

Modelo espiral de Boehm

Vantagens:

- Enfoque realista
- Reduz os riscos
 - Construção de protótipos
 - Consideração directa
- Mantém o enfoque sistemático do ciclo clássico

Desvantagens

- Custa crer que é controlável
- Requer boa capacidade de estimação de riscos
- Pouco utilizado

20-09-2004

Aula 2 - SW: produto e processo

23

Vantagens:

Abordagem realista do desenvolvimento de sistemas e do software de grande escala. Prevê a evolução do software e a compreensão e reacção progressiva dos informáticos e clientes perante os riscos ao longo do projecto.

Redução de riscos através da construção de protótipos em qualquer etapa do processo

Exige a consideração directa dos riscos técnicos em todas as etapas do projecto e se aplicado apropriadamente, reduz os riscos antes de se converterem em problemas

Mantém o enfoque sistemático do ciclo de vida clássico mas incorpora um esquema de trabalho iterativo que reflecte melhor a realidade

Desvantagens

Difícil de convencer os clientes que o método é controlável

Requer de uma considerável capacidade para a estimação de riscos e depende dela para o seu sucesso

Tem sido pouco utilizado, há pouca experiência acumulada

Outros modelos

- ‡ Win-Win
 - ‡ Negociação com o cliente
- ‡ Desenvolvimento concorrente
 - ‡ Actividades concorrentes
 - ‡ Diagramas de estado
- ‡ Métodos formais
 - ‡ Especificação matemática
 - ‡ Exacta e rigorosa
 - ‡ Detecta e corrige requisitos incompletos, ambíguos e inconsistentes
- ‡ Baseado em componentes
 - ‡ Unified Development Process
 - ‡ Derivado da orientação a objectos
 - ‡ Utiliza UML
- ‡ Ferramentas de 4ª Geração
 - ‡ Suporte automatizado à especificação de requisitos

20-09-2004

Aula 2 - SW: produto e processo

24

WIN-WIN: Modelo em espiral. Comunicação com o cliente implica um processo de negociação onde idealmente o cliente fica com um sistema que satisfaz a maior parte dos seus requisitos e quem desenvolve o sistema obtém orçamentos rentáveis e prazos realistas

Desenvolvimento concorrente: Processo é muito complexo, não é representado no modelo linear. Existência de muitas actividades concorrentes. Representado através de diagramas de estado.

Desenvolvimento baseado em componentes: a OO fornece o enquadramento tecnológico para o desenvolvimento baseado em componentes. OO contempla a criação de classes que encapsulam dados e algoritmos. Se desenhadas e implementadas apropriadamente permitem a reutilização. Características do modelo em espiral, é evolutivo por natureza. Configura aplicações a partir de componentes preparados e armazenados em bibliotecas. Conduz à reutilização. O processo unificado de desenvolvimento enquadra-se neste tipo de modelo. Utilizando UML no processo defini-se os componentes que se utilizarão no sistema e as interfaces que ligarão esses componentes. Utilizando uma combinação de desenvolvimento iterativo e incremental, o processo unificado define a função do sistema baseado em cenários na perspectiva do utilizador (use-cases). Depois acopla a função com um enquadramento arquitectónico que define a forma do software.

Métodos formais: actividades que conduzem à especificação formal do sw. Notação rigorosa e matemática. Notação Z. Variação ES de “sala limpa”. Permitem detectar e corrigir requisitos incompletos, ambíguos e inconsistentes, através de análises matemáticas. É um método difícil, caro e demorado. Requer estudo detalhado, difícil de comunicar a clientes sem bases técnicas. Adequado para software que requer muita segurança ou quando problemas na especificação de requisitos impliquem perdas grandes de dinheiro.

Ferramentas de 4ª Geração: Suporte automatizado à especificação de requisitos, geração automática de código, construção rápida de programas. Tipicamente incluem linguagens não procedimentais de consulta de bd, geração de informes, gestão de dados, interacção e definição de ecrãs, capacidades gráficas e de cálculo.

Produto vs. Processo

- # Se o processo é fraco, o produto vai sofrer
- # A dependência excessiva no processo também é má
- # A comunidade alterna a sua preocupação pelo produto e processo cada 5 ou 10 anos aprox.
- # Esta dualidade vai se manter por uns tempos