

Aula 2 – Engenharia de Sistemas

- ◆ Projetar, implementar, implantar e colocar em operação sistemas que incluem hardware, software e pessoas.

Aula 2 – Engenharia de Sistemas baseados em computadores

Apresentações correspondentes ao capítulo 2 do livro “Engenharia de Software”, de Ian Sommerville, 6.a Edição.

Baseado nos slides de:

©Ian Sommerville 2000 - Software Engineering, 6th edition

Objetivos

- ♦ Explicar porque sistemas de software são afetados amplamente por itens relacionados a engenharia de sistemas.
- ♦ Introduzir o conceito de propriedades emergentes de sistemas tais como confiabilidade e segurança.
- ♦ Explicar porque o ambiente do sistema deve ser considerado no processo de *design* do sistema.
- ♦ Explicar os processos de Engenharia de Sistemas e de Aquisição de Sistemas.

Motivação da Aula:

A Engenharia de Software precisa conhecer os conceitos de engenharia de sistemas, pois seus problemas normalmente são causados por decisões tomadas na engenharia de sistemas.



Assuntos abordados

- ◆ Propriedades emergentes de sistemas
- ◆ Os sistemas e seus ambientes
- ◆ Modelagem de sistemas
- ◆ O processo de engenharia de sistemas
- ◆ Procura e aquisição de sistemas

O que é um sistema?

- ♦ **Definição:** Uma determinada coleção de componentes interrelacionados trabalhando juntos de forma realizar um objetivo comum.
- ♦ Um sistema pode incluir software, hardware mecânico, elétrico e eletrônico e ainda ser operado por pessoas.
- ♦ Em geral, os componentes do sistema são dependentes dos outros componentes.
- ♦ As propriedades e o comportamento dos componentes do sistema estão intrinsecamente interligados.

A definição apesar de geral, abrange diversos tipos de sistemas, dos mais simples como um leitor óptico até os mais complexos como um Sistema de Controle de Tráfego Aéreo.

Num sistema, o funcionamento de um sistema depende do funcionamento dos componentes, e é comum um componente depender de outros componentes, no entanto, é possível que existam componentes que funcionem de forma isolada, por exemplo, um satélite meteorológico não depende de um barômetro para funcionar, mas sem ambos um sistema de previsão do tempo não funcionaria.

Problemas enfrentados pela Engenharia de Sistemas

- ◆ Grandes sistemas são usualmente projetados para resolver problemas do tipo '*wicked*', cuja tradução livre poderia ser *desagradáveis*.
- ◆ A Engenharia de Sistemas requer uma grande negociação de coordenação de diversas disciplinas.
 - As possibilidades quase infinitas para negociação do design dos componentes;
 - A falta de acordo mútua e de entendimento entre as áreas envolvidas.
- ◆ Os sistemas devem ser projetados para muitos anos, considerando as mudanças de ambiente.

Wicked Systems são sistemas tão complexos que a quantidade de entidades envolvidas é tão grande que não há nem sequer uma especificação clara do problema. Infelizmente a tradução do termo *wicked* não reflete o real significado do termo em inglês.

Nem sempre é possível um acordo entre as áreas envolvidas, um engenheiro civil tentará achar uma solução que seja mais correta do seu ponto de vista do que a solução de construção de uma obra que atenderia os interesses de um engenheiro elétrico por exemplo.

Em geral, um sistema deve resolver um problema por um tempo muito longo, no entanto há muitos fatores envolvidos, entre eles mudanças no ambiente organizacional, governamental ou mesmo ambiental. Imagine que um sistema de polimento de peças foi projetado usando um determinado componente químico e dois anos após a implantação de um sistema projetado para funcionar por dez anos, órgãos regulamentadores proibem o uso desse componente químico. Esse projeto deverá ser redesenhado ou simplesmente desativado.

Engenharias de Software e de Sistemas

- ♦ A proporção de software nos sistemas tem crescido muito nos últimos anos. Equipamentos eletrônicos tem sido desenvolvidos utilizando soluções genéricas baseadas em software do que de uso específico.
- ♦ Os problemas de engenharia de software são similares ao encontrados pela engenharia de sistemas.
- ♦ O software é (infelizmente) visto como um problema na disciplina de engenharia de sistemas. Uma variedade muito grande de projetos tem sofrido atrasos devido a problemas de software.

Propriedades Emergentes

- ◆ São as propriedades de um sistema com um todo, em vez de propriedades que possam ser derivadas das propriedades individuais dos componentes do sistema.
- ◆ Essas propriedades emergentes são consequência das relações entre os componentes do sistema.
- ◆ Elas podem portanto apenas serem avaliadas e medidas uma vez que todos os componentes tenham sido integrados no sistema.

As propriedades emergentes de um sistema só são vistas como um todo, não são específicas, nem se aplicam a um componente isolado.

Exemplos de propriedades emergentes

- ◆ *O valor efetivo do sistema*
 - Este é um exemplo simples de uma propriedade que pode ser computada a partir das propriedades individuais dos componentes.
- ◆ *A confiabilidade do sistema*
 - Esta depende da confiabilidade dos componentes do sistema e das relações entre os componentes.
- ◆ *A usabilidade do sistema*
 - Esta é uma propriedade complexa que não é simplesmente dependente dos sistemas de hardware e software, mas também depende dos operadores do sistema e dos fatores de ambiente de configuração onde este é usado.

Tipos de propriedades emergentes

- ◆ Propriedades funcionais
 - Estas propriedades aparecem quando das as partes do sistema trabalham juntas para atingir o objetivo proposto. Por exemplo, uma bicicleta tem uma propriedade funcional de ser um dispositivo de transporte, uma vez que foi construída a partir de seus componentes.
- ◆ Propriedades emergentes não funcionais
 - Exemplos são confiabilidade, performance e segurança. Estas propriedades referem-se ao comportamento no seu ambiente operacional. Elas são sempre críticas para sistemas baseados em computadores. Falhas para atingir um nível mínimo nestas propriedades podem tornar o sistema não utilizável.

Sobre os defeitos:

- Defeitos funcionais não percebidos, em geral não tem impacto no uso do sistema
- Defeitos não funcionais normalmente são rapidamente detectados e podem tornar o sistema não utilizável, tais como “lentidão” ou “não confiabilidade”.

Confiabilidade nos Sistemas de Engenharia

- ◆ Devido a interdependência dos componentes, as falhas podem ser propagadas através do sistema.
- ◆ Falhas de sistemas geralmente ocorrem por imprevistos nas inter-relações entre os componentes.
- ◆ É praticamente impossível antecipar todas as relações entre os componentes.
- ◆ As medidas de confiabilidade do software pode dar uma falsa idéia da confiabilidade do sistema.

Influências na confiabilidade

- ◆ *Confiabilidade do Hardware*
 - Qual é a probabilidade de um componente de hardware falhar e quanto tempo leva para consertá-lo?
- ◆ *Confiabilidade do Software*
 - Qual a probabilidade de um componente do software produzir um rendimento incorreto. Falha de Software normalmente é diferente de falha de Hardware na qual software não é afetado.
- ◆ *Confiabilidade do Operador*
 - Qual a probabilidade do operador de um sistema cometer um erro?

Relações de confiabilidade

- ◆ Falhas de Hardware podem gerar sinais espúrios (falsos indícios) que estão fora do limite de entradas esperadas pelo software.
- ◆ Erros de software podem causar alarmes a serem ativados, que podem causar *stress* no operador e então levar um erro do operador.
- ◆ O ambiente no qual o sistema está instalado pode afetar sua confiabilidade.

Um exemplo típico de falha causada por ambiente:

Uma falha no sistema de ar condicionado de precisão (que mantém equipamentos a uma determinada temperatura) pode acarretar problemas no funcionamento e na confiabilidade de um sistema como um todo. Computadores podem ter o comportamento do processador e da memória afetados por temperaturas altas.

As propriedades “Não deve”

- ◆ Propriedades como performance e confiabilidade podem ser medidas.
- ◆ Entretanto, algumas dessas propriedades são propriedades que o sistema não deve exibir.
 - Proteção/Segurança (Safety) – O sistema não deve se comportar de uma forma insegura, ou que possa causar danos.
 - Acesso/Segurança (Security) – O sistema não deve permitir o uso não autorizado de seus recursos.
- ◆ A medição ou avaliação destas propriedades, em geral, não são triviais.

A palavra segurança em português tem várias conotações e seu uso é empregado em várias formas. Em inglês existem palavras específicas para cada situação.

Os sistemas e seus ambientes

- ◆ Os sistemas não são independentes, mas sim estão associados a um ambiente.
- ◆ As funções do sistema podem alterar seu ambiente.
- ◆ O oposto também é válido. O ambiente pode afetar o funcionamento e as funcionalidades de um sistema. Por exemplo, um sistema pode precisar de uma fonte de energia do seu ambiente.
- ◆ O ambiente organizacional, assim como o ambiente físico podem ser importantes. Outros fatores ambientais como governo e agência regulamentadoras podem ter impacto no sistema.

Hierarquias nos Sistemas



Esse é um exemplo simples, mas mostra como um sistema pode depender de outros, como por exemplo, o sistema de água de um prédio pode depender do sistema de abastecimento de água de uma determinada rua, bairro ou mesmo cidade.

Ou mesmo, o sistema de segurança pode ser comprometido por um corte de energia, por essa razão foi considerado no exemplo um sistema de energia próprio (talvez baseado em geradores).

Fatores humanos e organizacionais

- ◆ *Mudanças de processos*
 - O sistema requer mudanças no processo de trabalho dentro do ambiente da organização? (automatização, redução de pessoal, reengenharia, etc...)
- ◆ *Mudanças nos empregos*
 - O sistema envolverá novos perfis de trabalhadores no ambiente ou causará mudanças na forma de trabalhar?
- ◆ *Mudanças organizacionais*
 - O sistema alterará as estruturas de poder político dentro da organização?

Durante o projeto de um sistema esses fatores são relevados ou mesmo ignorados. No entanto, são responsáveis por falhas irremediáveis nos sistemas ou mesmo a morte precoce do projeto.

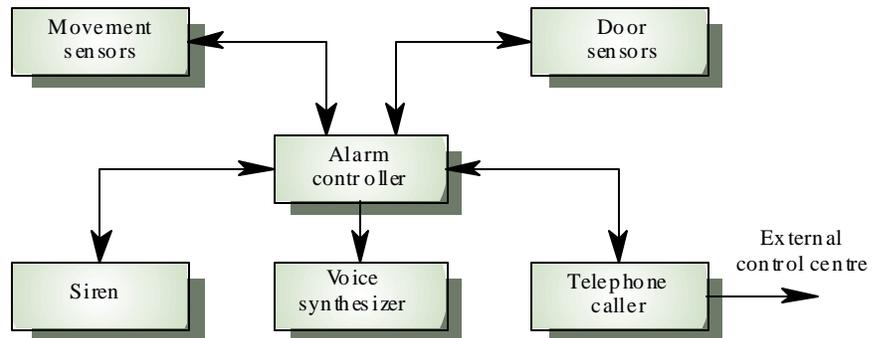
Alguns exemplos:

- Se o sistema significar “perda de emprego”, os usuários não contribuem para a análise do problema.
- Os gerentes não sofrem impacto com a ativação de sistemas, mas se isso acarretar possíveis impactos nas carreiras, os mesmos criarão dificuldades no desenvolvimento e futuramente na operação.
- Uma possível modificação na organização é o novo status que um operador passa a ter dentro da organização depois da implantação de um novo sistema. Isso pode ser benéfico ou maléfico para a organização.

Modelagem da arquitetura do sistema

- ◆ Um modelo arquitetônico apresenta uma visão abstrata dos subsistemas que compõem o sistema.
- ◆ Podem incluir as principais fluxos de informações entre os sub-sistemas.
- ◆ Usualmente é apresentado como um diagrama de blocos.
- ◆ Além dos elementos gráficos traz também uma descrição textual.
- ◆ Pode identificar diferentes tipos de funcionalidades dos componentes no modelo.

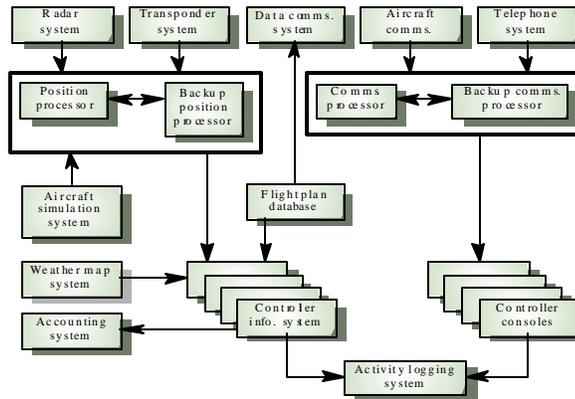
Exemplo: Sistema de alarme de intrusos



Tipos de componentes num sistema de alarme

- ◆ Sensores
 - Sensores de movimento, sensores de portas e janelas
- ◆ Atuantes
 - Sirene
- ◆ Comunicação
 - Discador de telefone
- ◆ Coordenação
 - Controle do alarme
- ◆ Interface
 - Sintetizador de voz

Exemplo: Sistema de Controle de Tráfego Aéreo



Componentes funcionais do sistema

- ◆ Componentes sensores
- ◆ Componentes atuantes
- ◆ Componentes computacionais
- ◆ Componentes de comunicação
- ◆ Componentes de coordenação
- ◆ Componentes de interface

Componentes do sistema

- ◆ Componentes sensores
 - Obter informações do ambiente do sistema, como radares no sistema de controle de tráfego aéreo.
- ◆ Componentes atuantes
 - Causam alguma alteração no ambiente do sistema, como válvulas no sistema de controle de pressão, que aumentam ou diminuem a quantidade de material num duto.
- ◆ Componentes de computação
 - Responsáveis por calcular ou computar dados de entrada e fornecer resultados, como um processador de ponto flutuante num sistema.

Componentes do sistema

- ◆ Componentes de comunicação
 - Permitem aos componentes comunicarem-se entre si, por exemplo, computadores distribuídos numa rede.
- ◆ Componentes de coordenação
 - Coordenar as interações de outros componentes de sistema. Exemplo: Um escalonador num sistema de tempo real.
- ◆ Componentes de interface
 - Facilitar as interações com outros componentes, como por exemplo, uma interface para o operador usar o sistema.
- ◆ Em geral, a maioria dos componentes são controlados por software.

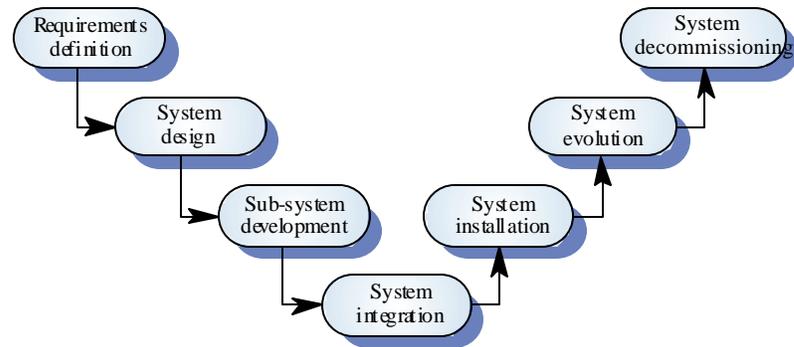
Exercício Prático

1. Detalhe e classifique o sistema de controle de tráfego aéreo, mostrado na página 20.
2. Faça um modelo gráfico e sua descrição e classificação de sub-sistemas para um sistema de prevenção de enchentes num grande centro urbano.

O processo de Engenharia de Sistemas

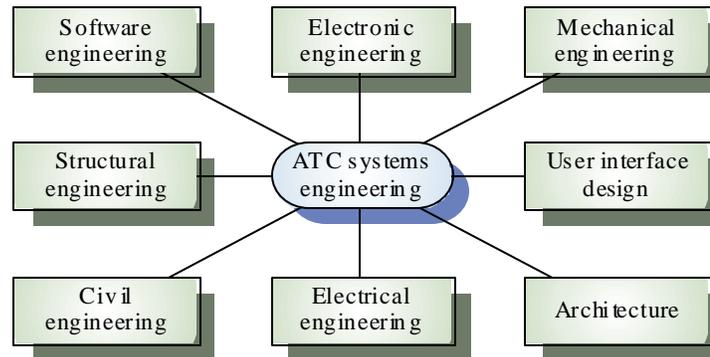
- ◆ Geralmente, é utilizado um modelo “*cascata*” (“*waterfall*”) devido a impossibilidade de desenvolvimento paralelo das partes diferentes do sistema.
 - Um pequeno escopo para interações entre as fases devido as custos elevados se houve alterações de hardware. O Software pode ter que compensar problemas encontrados no hardware.
- ◆ Inevitavelmente envolve engenheiros de diferentes áreas que devem trabalhar juntos.
 - Há muito escopo para conseguir um entendimento. Diferentes áreas envolvem diferentes vocabulários e muita negociação é necessária. Além dos engenheiros terem que ajustar suas agendas para atender o mesmo objetivo.

O processo de Engenharia de Sistemas



Modelo cascata (*waterfall*) para a Engenharia de Sistemas.

Envolvimento interdisciplinar no projeto de um Sistema de Controle de Tráfego Aéreo (ATC)



Note que a Engenharia de Software é apenas uma disciplina dentro da Engenharia de Sistemas. O Engenheiro de Software para produzir seus produtos terá que interagir com outros tipos de profissionais.

Definição de Requisitos de Sistema

- ◆ Três tipos de requisitos são definidos neste estágio:
 - Requisitos Funcionais Abstratos. As funções do sistema são definidas de uma forma abstrata.
 - Propriedades do Sistema. Requisitos não funcionais são definidos.
 - Características indesejáveis. Os comportamentos não aceitáveis do sistema são especificados.
- ◆ É necessário também definir os objetivos globais da organização para o sistema.

Objetivos do Sistema

Exemplo: Sistema de Alarme

◆ Objetivos Funcionais

- Prover um sistema de alarme para detectar incêndios e intrusos para o prédio, o qual fornecerá um aviso interno e externo de fogo ou entrada não autorizada de pessoas.

◆ Objetivos Organizacionais

- Garantir que o funcionamento normal do trabalho desempenhado no prédio não seja seriamente prejudicado por eventos como fogo ou invasão de elementos não autorizados.

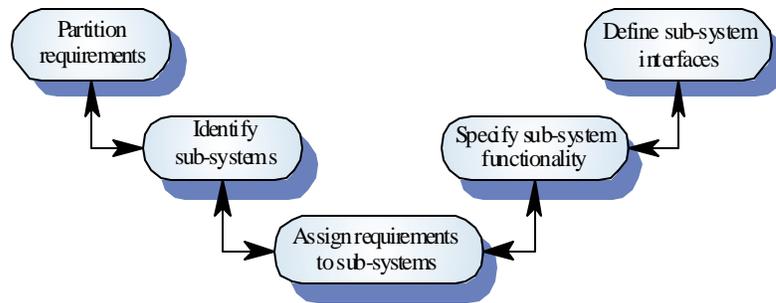
Problemas nos Requisitos de Sistema

- ◆ Mudanças no sistema enquanto ele ainda estiver sendo especificado.
- ◆ Deve antecipar elementos de hardware/comunicação para funcionamento maior que a vida útil do sistema. Prever uma sobrevida dos componentes.
- ◆ Dificuldade de definir requisitos não funcionais, especialmente, sem uma impressão da estrutura de componentes do sistema.

O processo de Design do Sistema

- ◆ Divisão dos Requisitos
 - Organizar os requisitos em grupos relacionados.
- ◆ Identificar os sub-sistemas
 - Identificar um conjunto de sub-sistemas que conjuntamente podem atender os requisitos do sistema.
- ◆ Atribuir os requisitos aos sub-sistemas
 - Causa problemas quando COTS (componentes de prateleira / Components Of The Shelf) são integrados.
- ◆ Especificar funcionalidades dos sub-sistemas
- ◆ Definir as interfaces dos sub-sistemas
 - Atividades críticas para paralelizar o desenvolvimento dos sub-sistemas.

The system design process



As setas em sentido duplo indicam:

- Interação
- Feedback

Problemas no Design de Sistemas

- ◆ A divisão dos requisitos para componentes de hardware, de software e humanos podem envolver uma série de negociações.
- ◆ Problemas associados ao design são assumidos que serão resolvidos pelo software.
- ◆ Plataformas de hardware podem ser inapropriadas para os requisitos de software, desta forma o Software deve compensar esse item.

Desenvolvimento de sub-sistemas e seus problemas

- ◆ Tipicamente paraleliza os projetos de desenvolvimento de hardware, software e comunicações.
- ◆ Pode envolver procura e aquisição de COTS.
- ◆ Falta de comunicação entre os times de implementação.
- ◆ Mecanismos burocráticos e lentos para propor mudanças no sistema significam que o prazo pode ser estendido devido a necessidade de re-trabalho (rework).

Se os módulos são dependentes pode haver mudanças de requisitos (“change requirements”) no software e muitos ajustes de contorno (“work-arounds”) para satisfazer as necessidades de mudanças.

Integração de Sistemas

- ◆ Consiste no processo de colocar hardware, software e pessoas juntas para completar o sistema.
- ◆ Deve ser conduzido em incrementos, de forma que os sub-sistemas são integrados um por vez, evitando maiores problemas.
- ◆ Problemas de interface entre os sub-sistemas são usualmente encontrados neste estágio.
- ◆ Pode envolver problemas com entregas não coordenadas de componentes do sistema.

É comum ocorrer a dificuldade de agendar/planejar o desenvolvimento individual dos componentes.

Instalação do Sistema

- ◆ Considerações sobre o ambiente podem estar incorretas.
- ◆ Pode haver resistência humana na introdução de um novo sistema.
- ◆ O sistema pode ter que coexistir com sistemas alternativos por um certo tempo.
- ◆ Pode haver problemas de instalação física (exemplo: falta de cabeamento de rede/elétrico).
- ◆ Necessidade de treinamento de operadores deve ser indentificada.

Problemas comuns:

- Ambiente diferente do considerado pelos desenvolvedores.
- Boicote de informações.
- Problemas de compatibilidade. Típico: Rede elétrica considerada: 127V; encontrada no ambiente: 220V

Operação do Sistema

- ◆ Trará requisitos não previstos à tona.
- ◆ Usuários podem usar o sistema de uma forma não prevista pelos engenheiros.
- ◆ Pode revelar problemas na interação com outros sistemas:
 - Problemas físicos de compatibilidade;
 - Problemas de conversão de dados;
 - Taxa crescente de erros do operador causadas por inconsistência nas interfaces.

Evolução dos Sistemas

- ◆ Grandes sistemas tem uma longa vida. Eles devem envolver um mecanismo para atender mudanças de requisitos.
- ◆ Evolução/Manutenção é inerentemente cara:
 - Mudanças deve ser analisadas pelas perspectivas técnicas e de negócio;
 - Os sub-sistemas interagem, e problemas não-antecipados aparecem;
 - Raramente há razões para decisões no design original;
 - A estrutura do sistema pode ser corrompida quando mudanças são feitas.
- ◆ Sistemas existentes que devem ser mantidos são usualmente chamados de **Sistemas Legados**.

Descomissionamento do Sistema

- ◆ Retirar o Sistema de serviço após seu ciclo de vida útil.
- ◆ Pode envolver remoção de materiais (como materiais químicos perigosos) que podem poluir o ambiente.
 - Deve ser feito um planejamento no design do sistema para seu encapsulamento. (Uma atividade como essa poderia ter evitado o acidente do Césio 137 em Goiânia na década de 1990).
- ◆ Pode necessitar uma reestruturação de dados e convertê-los para um outro sistema.
- ◆ Pode envolver também a destruição de informações que não serão utilizadas, mas que podem significar dados confidenciais de uma organização.

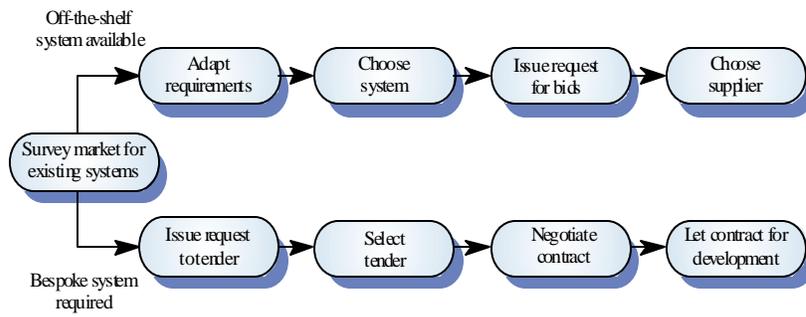
Sistema de Procura e Aquisição de Sistemas

- ◆ Consiste em adquirir um sistema para uma organização para atender alguma necessidade.
- ◆ Alguma especificação e projeto arquitetural do sistema é necessário antes da procura:
 - É preciso uma especificação para levar um contrato ao desenvolvimento do sistema;
 - A especificação pode permitir a aquisição de um sistema comercial pronto (COTS). Quase sempre mais barato que o desenvolvimento de um sistema a partir de um esboço.
 - No caso de uso de COTS, às vezes é necessário um software para interligá-los (*glueware*).

Esse processo vem da palavra “procure” ou “procurement”, em inglês. No entanto sua tradução livre não leva diretamente a “procura”.

Em inglês essa palavra significa “busca por algo importante ou valioso”.

O processo de Procura/Aquisição de Sistemas



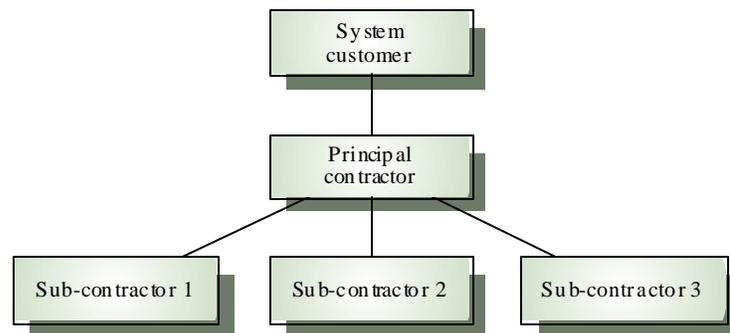
Questões associadas a procura/aquisição

- ◆ Requisitos pode ter que ser modificados para casar com as capacidades dos componentes de prateleira (COTS).
- ◆ A especificação de requisitos podem ser parte de um contrato para o desenvolvimento do sistema.
- ◆ É comum haver um período de negociação de contrato para acordar mudanças após o contratado para construir o sistema ter sido escolhido.

Contratados e sub-contratados

- ♦ A aquisição de grandes sistemas de Software/Hardware é normalmente baseada em torno de um contratado principal.
- ♦ Sub-contratados são apontados para outros fornecedores para suprir determinadas partes do sistema.
- ♦ O cliente negocia com o contratado principal e não se envolve diretamente com sub-contratados.
- ♦ A sub-contratação é um item essencial para a certificação CMM, e tem requisitos específicos a serem atendidos.

Modelo de Contratação/Sub-Contratação



Pontos-Chave

- ♦ A Engenharia de Sistemas envolve entradas de uma série de áreas.
- ♦ Propriedades Emergentes são as propriedades que são características de um sistema como um todo, não apenas de seus componentes.
- ♦ Modelos de arquitetura de sistema mostram os principais sub-sistemas e suas interligações. São usualmente descritas através de diagramas de blocos.

Pontos-Chave

- ◆ Os tipos de componentes de um sistema podem ser classificados em: sensores, atuantes, calculadores/computacionais, coordenação, comunicação e interface.
- ◆ O processo de engenharia de sistemas adotado geralmente é o modelo *cascata*.
- ◆ A Engenharia de Sistemas é uma área muito complexa, e portanto é difícil estabelecer um processo padrão para desenvolver um sistema.
- ◆ A procura/aquisição de sistemas é envolvida com a decisão de que sistema comprar ou escolher quem irá desenvolvê-lo.

Conclusões

- ♦ Engenharia de Sistemas é difícil e trabalhosa! Nunca haverá uma resposta simples para os problemas complexos do desenvolvimento de sistemas.
- ♦ Os Engenheiros de Software não tem todas as respostas, mas podem fazer o melhor possível se usarem o ponto de vista dos engenheiros de sistemas.
- ♦ Uma área precisa reconhecer os esforços das outras e atuar ativamente em vez de relutar a usar soluções propostas. As diferentes áreas devem cooperar no processo de engenharia de desenvolvimento de sistemas.

Leitura Recomendada

- ♦ Capítulo 2 (Texto base)
Ian Sommerville; Engenharia de Software, 6.a Edição; Makron Books, **2002**
- ♦ Capítulo 10
Roger S. Pressman; Engenharia de Software, 5.a Edição; McGraw-Hill, **2002**
- ♦ B. Thomé; System Engineering: Principles and Praticce of Computer-based Systems Engineering; John Wiley and Sons, **1993**

Questões

1. Explique porque outros sistemas dentro do ambiente do sistema pode ter efeitos não previstos no funcionamento de um sistema.
2. Modifique o processo de design (pg 32) para incorporar explicitamente as atividades do processo de procura/aquisição de software. Mostre no seu diagrama, o *feedback* que os resultados da incorporação dess atividade.
3. Explique porque a especificação de um sistema usado por serviços de emergência para gerenciamento de catastrofes é inerentemente um sistema '*wicked*'.
4. Sugira como um sistema de software usado num carro pode ajudar com o descomissionamento do sistema como um todo.
5. Explique porque é importante produzir uma descrição geral da arquitetura do sistema num estágio inicial do processo de especificação do sistema.

Questões

6. Foi dado um exemplo no texto de um sistema de alarme. Extenda esse sistema para detectar incêndios. Desenhe um diagrama de blocos para um possível design desse sistema.
7. Assumindo que um sistema que foi encomendado por uma determinada empresa atende suas especificação. Descreva usando exemplos associados, três problemas que podem surgir quando o sistema estiver sendo instalado na organização.
8. Quais os argumentos pró e contra ao considerar a engenharia de sistemas com uma profissão como engenharia elétrica, mecânica ou mesmo engenharia de computação?
9. Imagine que você é um engenheiro envolvido no desenvolvimento de um sistema financeiro. Durante a instalação você descobre que este sistema causará a possível demissão de muitas pessoas. As pessoas no ambiente negam a você o acesso a informações essenciais para você completar a instalação do sistema. Como você atuaria, como um engenheiro de sistema, para resolver esse problema? É sua responsabilidade completar a instalação como contratado? Ou você deve simplesmente abandonar o trabalho antes que a organização tenha encontrado uma saída para o problema?