



# Engenharia para a Web

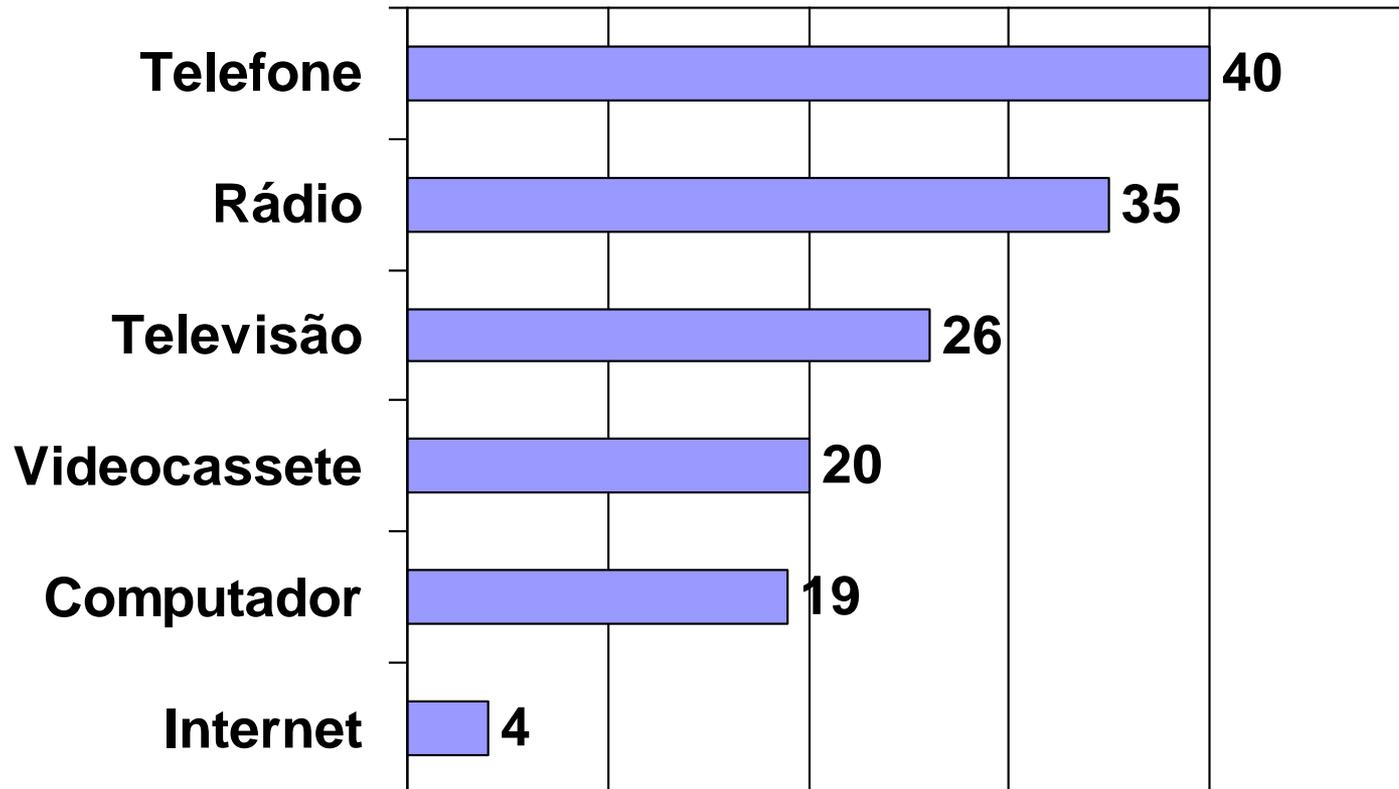
Fabricio Breve  
Universidade Federal de São Carlos – UFSCar  
São Carlos – Junho de 2002  
Projeto e Gerência de Sistemas de Software  
Professoras: Sandra / Rosângela / Júnia

# Tópicos

- A Evolução da Web
- Engenharia para a Web
- Características de Aplicativos para a Web
- Modelo de Processo
- Formulação
- Análise
- Projeto
  - Projeto de Arquitetura
  - Projeto de Navegação
  - Projeto de Interface
- Testes
- Gerenciamento
  - A equipe de desenvolvimento
  - Gerenciamento de Projeto
  - Gerenciamento de Configuração
- Conclusão

# A Evolução da Web

Tempo levado para chegar a 30% da população americana (em anos)





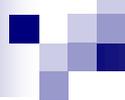
# No início:

- Conteúdo estático e textual
- Interação limitada
- Navegabilidade fácil
- Desempenho não era essencial
- Sites desenvolvidos apenas por uma pessoa ou por um pequeno grupo



# Atualmente:

- Escopo e complexidade vêm aumentando
- Sons, imagens, animações, etc.
- Navegabilidade complexa
- Desempenho essencial



# Alguns dados...

- 84% dos sistemas entregues não atendem as necessidades do cliente
- 79% dos projetos sofrem com atrasos
- 63% tem custo maior que o previsto
- 50% dos sistemas prontos tem problemas: são de baixa qualidade e faltam funcionalidades necessárias

# O cenário atual

- Sites feitos com pouca disciplina, sem preocupação com técnicas e métodos padronizados ou maneiras de controlar a qualidade
- Muitos desenvolvedores acham que o desenvolvimento para a Web é apenas criação de páginas usando HTML, *Frontpage* ou *Dreamweaver*
- Muitos livros e revistas também tratam o desenvolvimento para a Web desta maneira



# Engenharia para a Web

- Processo utilizado para criar aplicativos *Web* de alta qualidade
- Não é igual a Engenharia de Software tradicional, mas compartilham muitos conceitos e princípios fundamentais
- Incorpora novas abordagens, metodologias, ferramentas, técnicas e normas para atender os requisitos exclusivos dos sistemas para a *Web*
- Desenvolvimento para a *Web* é uma mistura de publicações impressas e desenvolvimento de software, entre marketing e computação, entre comunicações internas e relações externas, e entre arte e tecnologia

# Áreas envolvidas:

- análise de sistemas e projetos
- engenharia de software
- engenharia de hipermídia e hipertexto
- engenharia de requisitos
- interação humano-computador
- desenvolvimento de interface de usuário
- engenharia de informação
- indexação e recuperação de informações
- teste
- modelagem e simulação
- gerenciamento de projetos
- projeto gráfico e apresentação



# Características de Aplicativos Web

- Uso intenso de recursos de rede
- Dirigido a conteúdo
- Evolução contínua

# Analogia da ponte



# Analogia do jardim



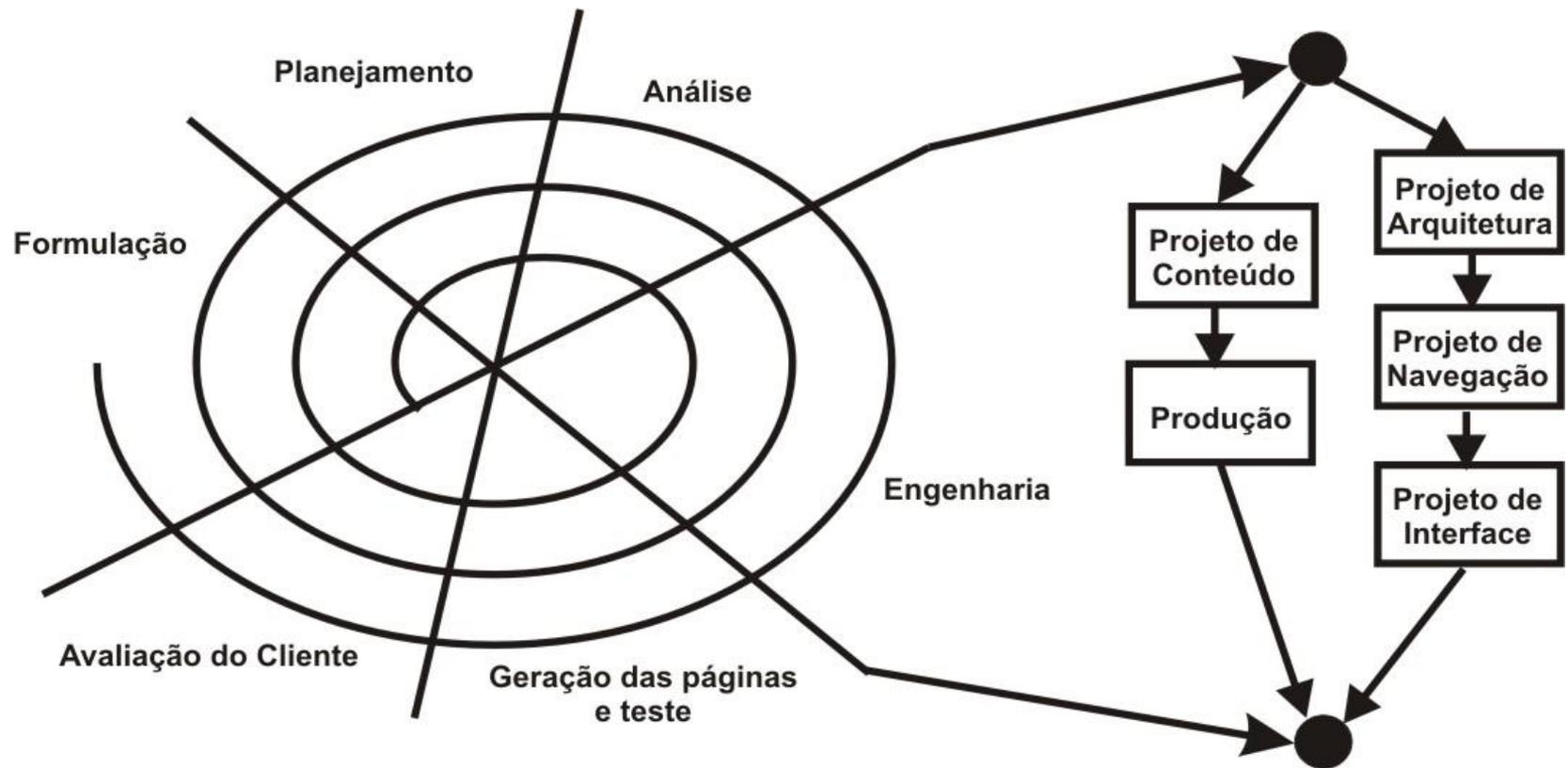
# Desenvolvimento para a Web X Desenvolvimento tradicional

- ***Imediatismo:*** o tempo em que um *site* completo precisa ficar pronto pode ser apenas alguns poucos dias ou semanas
- ***Segurança:*** aplicativos para a *Web* estão disponíveis via rede, é difícil ou até mesmo impossível limitar a população de usuários que irão acessar o aplicativo
- ***Estética:*** é inegável que boa parte do apelo dos aplicativos para a *Web* é o seu visual

# Tipos de aplicativos *Web*

- **Informacional.** Conteúdo apenas para leitura é fornecido com navegação simples e *links*.
- **Download.** Um usuário faz o *download* de informações dos servidores apropriados
- **Personalizável.** O usuário personaliza o conteúdo para suas necessidades específicas
- **Interação.** Comunicação entre uma comunidade de usuários ocorre em salas de bate-papo, fóruns ou mensagens instantâneas.
- **Entrada de Usuário.** Entradas baseadas em formulários são os mecanismos primários para a comunicação necessária.
- **Orientado a transações.** O usuário faz um pedido que é atendido pelo aplicativo
- **Orientado a serviços.** O aplicativo fornece um serviço para o usuário.
- **Portal.** O aplicativo direciona o usuário para outros conteúdos ou serviços fora do domínio do portal do aplicativo.
- **Acesso a Banco de Dados.** O usuário faz uma consulta em um grande banco de dados e extrai informações.
- **Data warehousing.** O usuário consulta uma coleção de grandes bancos de dados e extrai informações.

# Modelo de Processo



# Atividades do Modelo de Processo

- **Formulação:** atividade que identifica as metas e objetivos do aplicativo e determina um escopo para o primeiro incremento
- **Planejamento:** estima os custos do projeto, avalia riscos associados com o desenvolvimento, e define um cronograma bem definido para o primeiro incremento, e um menos definido para os demais
- **Análise:** são estabelecidos os requisitos técnicos e identificados os itens de conteúdo que serão incorporados. Requisitos para projeto gráfico (estética) também são definidos.

# Atividades do Modelo de Processo

## ■ ***Engenharia:***

- ***Projeto de Conteúdo e Produção:*** projetar, produzir e/ou obter todos os textos, gráficos, conteúdo de áudio e vídeo que serão integrados ao aplicativo.

# Atividades do Modelo de Processo

- ***Geração de Páginas:*** conteúdo definido na atividade de engenharia é fundido com os projetos de arquitetura, navegação e interface para produzir páginas
- ***Teste:*** tenta descobrir erros em *applets*, *scripts* e formulários; e ajuda a garantir que o aplicativo irá funcionar corretamente em diferentes ambientes
- ***Avaliação do Cliente:*** mudanças são pedidas e serão integradas ao sistema no próximo ciclo do processo incremental.

# Problemas que normalmente ocorrem:

- Inconsistência das informações;
- Manutenção normalmente utiliza muitos recursos e tempo;
- Falta de escalabilidade;
- Necessidade de um “visual” comum;
- No início, requisitos são muito vagos;
- Os requisitos irão mudar consideravelmente durante o desenvolvimento e especialmente quando o aplicativo já estiver em uso;
- A tecnologia também vem mudando muito rapidamente.

# Cenário atual

- Grande parte dos desenvolvedores para a Web não utilizam um processo de desenvolvimento bem definido e documentado
- Dos poucos que estão utilizando processos bem definidos e documentados, apenas alguns estão utilizando modelos padronizados pela indústria de software
- Na maioria dos casos o processo se foca na Implementação. Análise de Requisitos e Projeto normalmente estão combinadas no começo do projeto, e as atividades de Teste são conduzidas em conjunto com a implementação.



# Formulação

As seguintes perguntas devem ser feitas:

- Qual o principal motivo para desenvolvermos este aplicativo?
- Por que este aplicativo é necessário?
- Quem vai usar este aplicativo?

# Formulação

As respostas destas perguntas servirão para definir:

- ***Metas de informação:*** Indicam a intenção de fornecer conteúdo específico e/ou informação para o usuário
- ***Metas de aplicativo:*** Indicam a habilidade de executar tarefas do aplicativo

# Formulação

- Quando todas as metas de ambos os tipos forem identificadas, um perfil de usuário é desenvolvido
- Quando todas as metas e perfis de usuários estiverem desenvolvidos, a atividade de formulação irá focar a declaração de escopo do aplicativo para a *Web*

# Análise

- **Análise de Conteúdo:** todo o conteúdo a ser fornecido pelo aplicativo é identificado
- **Análise de Interação:** a maneira pela qual o usuário interage com o aplicativo é descrita em detalhes
- **Análise Funcional:** os cenários de uso criados na análise de interação irão definir operações que irão ser utilizadas no aplicativo, que implicam outras funções de processamento
- **Análise de Configuração:** O ambiente e a infraestrutura na qual o aplicativo reside são descritos em detalhe

# Análise: cenário atual

- especificação detalhada de requisitos raramente é feita
- argumentação mais comum é que a evolução contínua dos aplicativos para a *Web* torna os documentos de requisitos obsoletos antes mesmo de completar o desenvolvimento



# Projeto

O dilema do projetista:

- O projeto deve resolver os problemas imediatos
- É preciso criar uma arquitetura que comporte uma evolução rápida

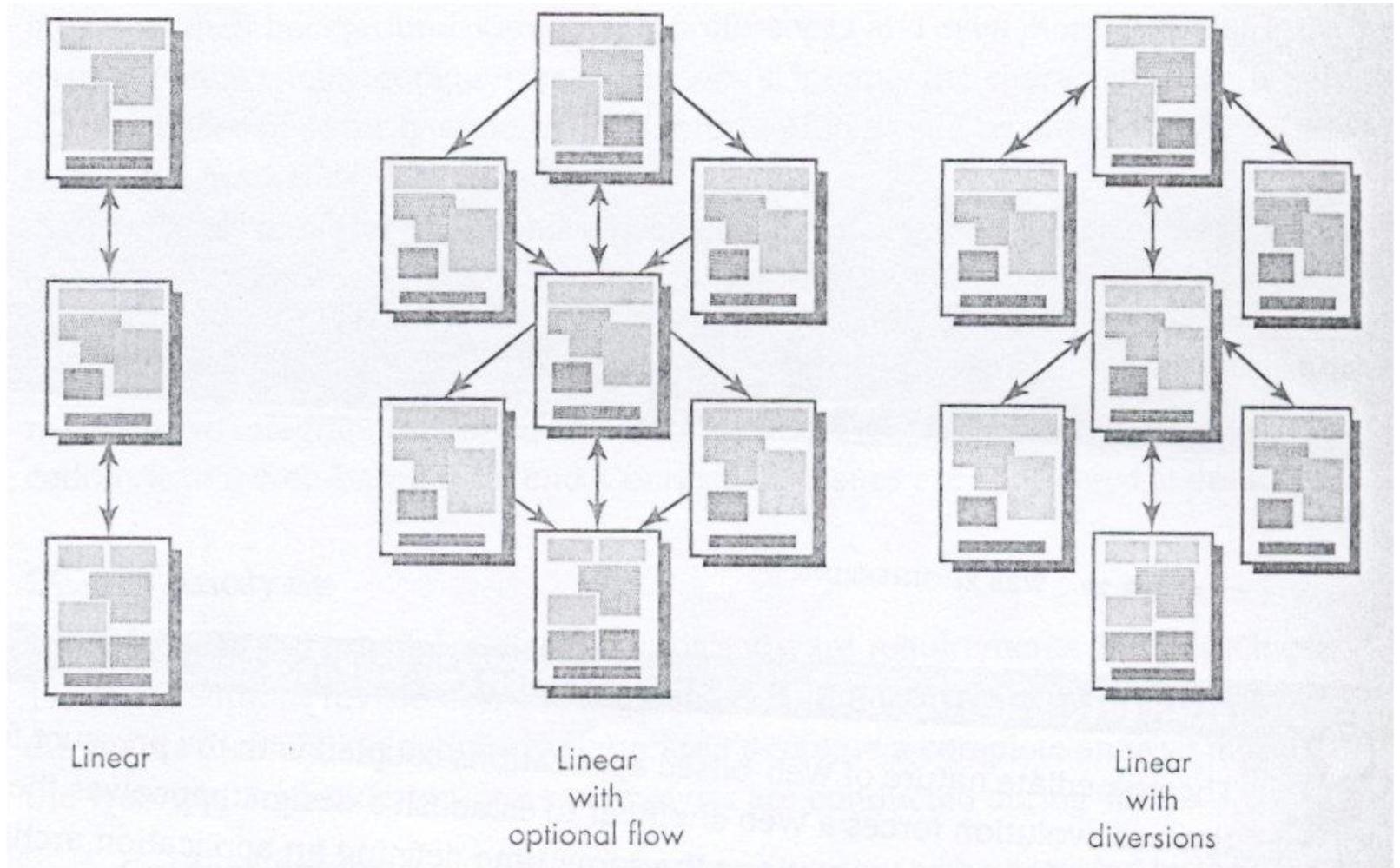
# Elementos técnicos para um projeto efetivo:

- **Métodos e Princípios de Projeto:** Modularidade eficiente (alta coesão e baixo acoplamento) e outras heurísticas da construção de software devem ser utilizadas também para a *Web*. Pode-se utilizar inclusive os métodos de projetos para sistemas orientados a objetos, pois a hipermídia define “objetos” que interagem através de um protocolo de comunicação que é bem parecido com o utilizado na orientação a objetos.
- **Regras de Ouro (*Golden Rules*):** Sistemas para a *Web* já vêm sendo construídos há uma década. Neste tempo, os projetistas desenvolveram um conjunto de heurísticas que podem ser reaplicadas durante o projeto de novos aplicativos.
- **Padrões de Projetos (*Design Patterns*):** São abordagens genéricas utilizadas para resolver problemas genéricos que podem ser adaptadas para resolverem uma grande variedade de problemas mais específicos.
- **Modelos (*Templates*):** um modelo pode ser utilizado para fornecer um esqueleto para qualquer tipo de padrão de projeto que será utilizado no aplicativo.

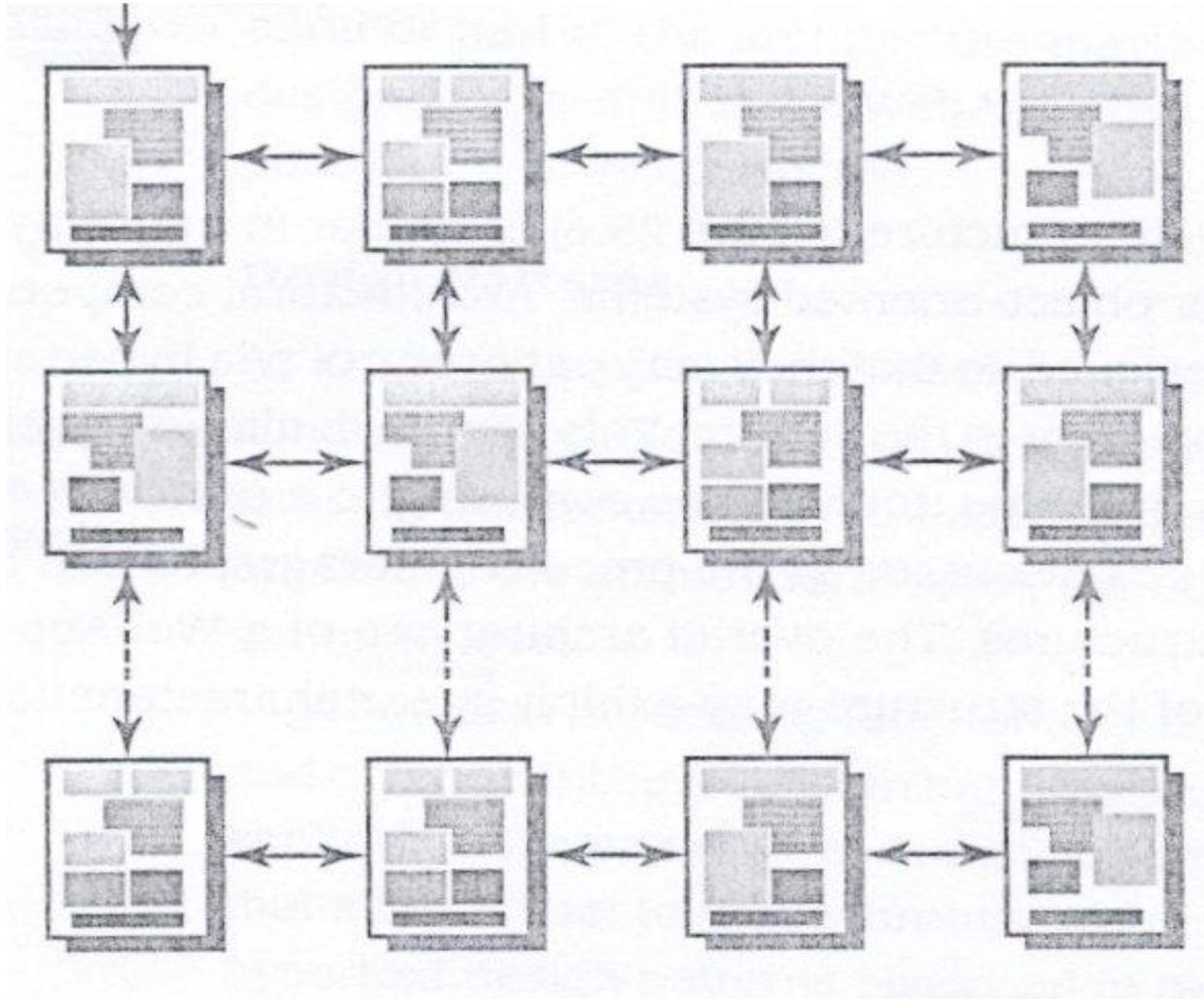
# Projeto de Arquitetura

- Tem foco na definição da estrutura hipermídia do aplicativo, na aplicação de padrões e na construção de modelos (*templates*) para montar a estrutura e permitir reuso
- A atividade paralela chamada de *Projeto de Conteúdo*, deriva a estrutura geral e o esboço detalhado do conteúdo que será apresentado no aplicativo
- Nesta etapa é definida a estrutura que será utilizada

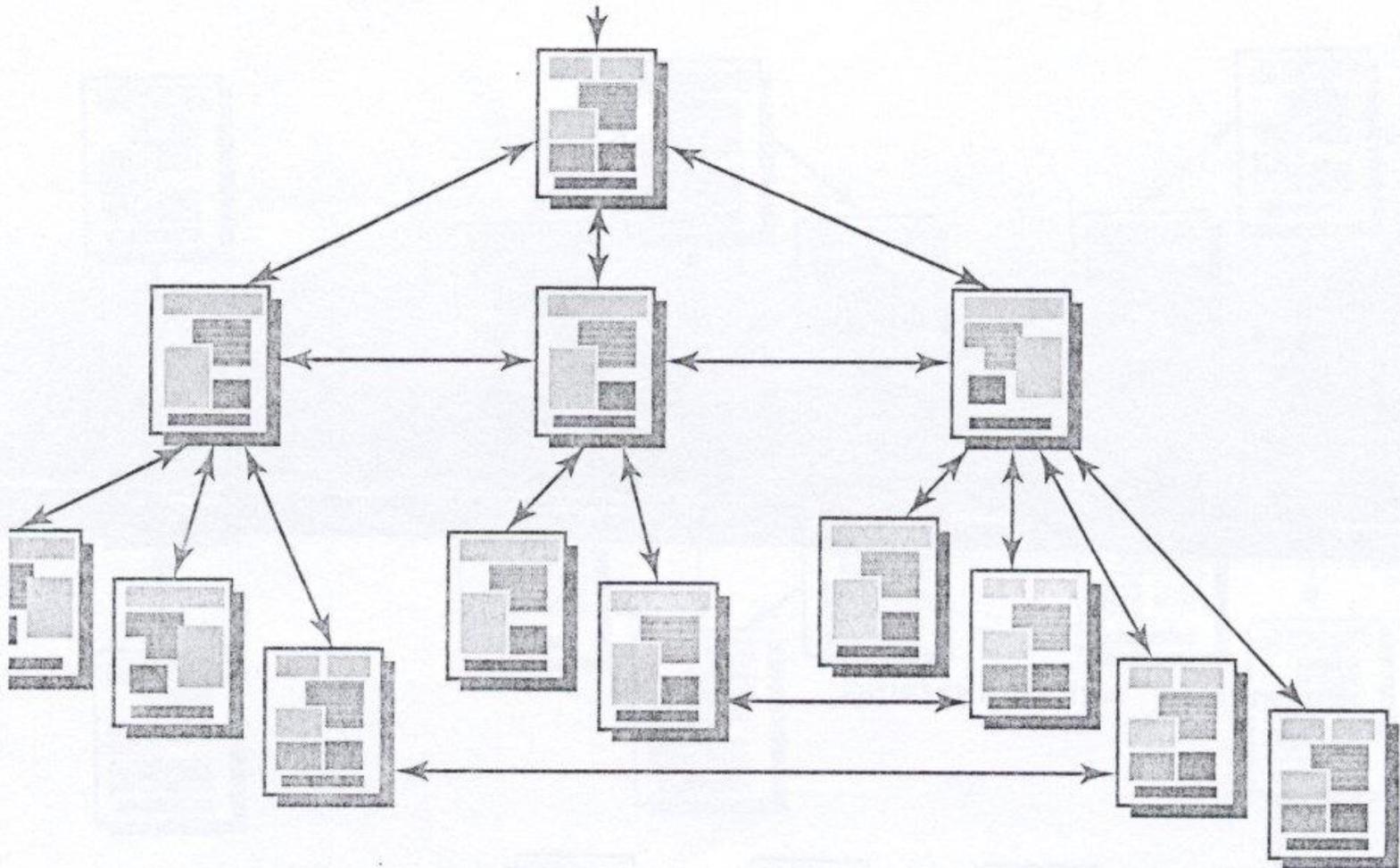
# Estrutura Linear



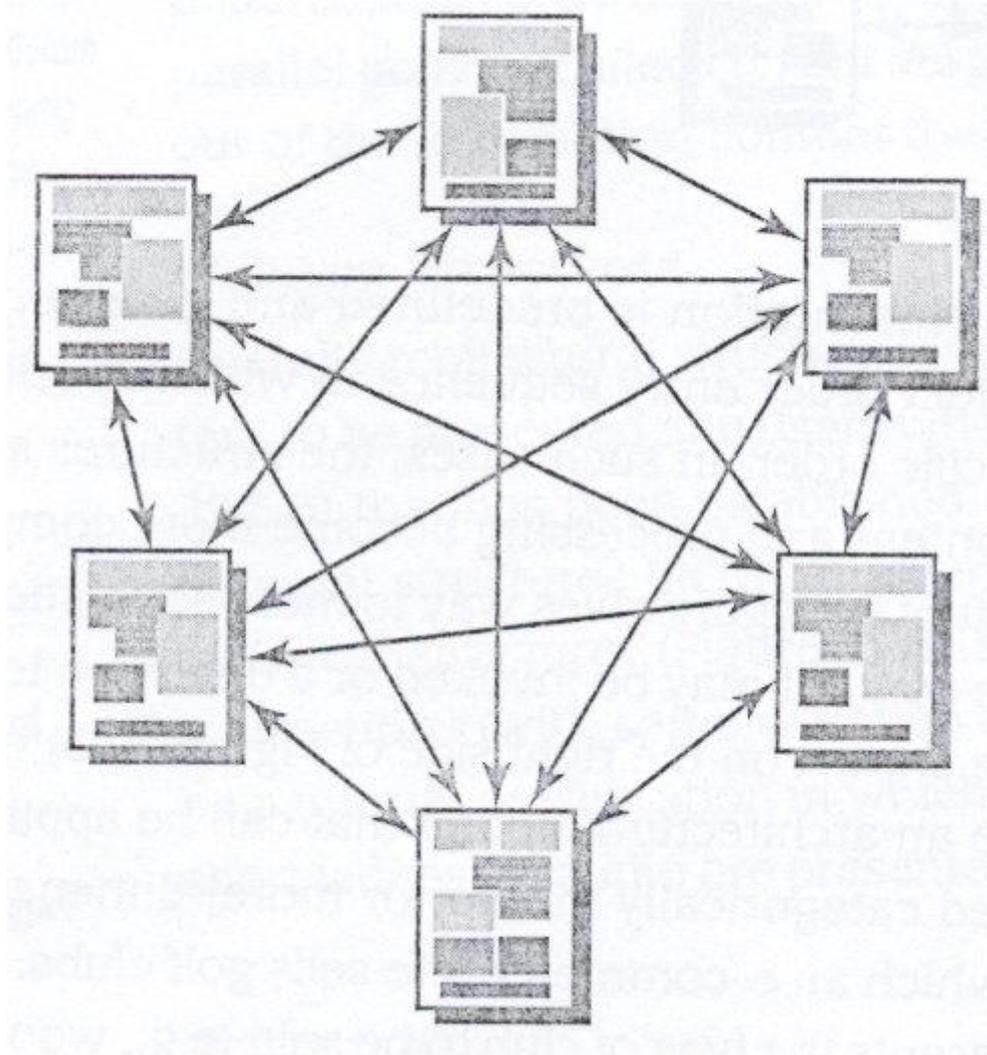
# Estrutura de Grade



# Estrutura Hierárquica



# Estrutura de Rede (*Pure Web*)



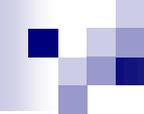


# Projeto de Navegação

- Objetivo: definir caminhos que permitam ao usuário acessar conteúdos e serviços
- identificar as semânticas de navegação para diferentes usuários e definir os mecanismos para realizar a navegação

# Tarefas do Projeto de Navegação

- Definir tipos de usuários
- Criar uma unidade semântica de navegação (*semantic navigation unit – SNU*) para cada objetivo associado a cada tipo de usuário
  - *SNU* é composta de um conjunto de sub-estruturas navegáveis que podemos chamar de caminhos (*ways of navigating – WoN*)
  - Cada um desses caminhos representará a melhor maneira de navegar para que um determinado usuário atinja sua meta ou sub-meta.
- Escolher como os *links* serão identificados. Dentre as opções estão: textos, ícones, botões, etc



# Projeto de Interface

- Métodos utilizados na Engenharia de Software podem ser utilizados, porém são necessárias algumas considerações adicionais
- Na Web a interface tem um papel fundamental

# Recomendações para construção de uma boa interface:

- Erros no servidor, mesmo os menores, podem fazer com que um usuário deixe o *site* e procure a informação ou serviço que deseja em outro lugar;
- Não se deve forçar o usuário a ler grandes quantidades de texto, principalmente se for texto explicando como operar o aplicativo ou navegar por ele;
- Avisos de “Em Construção” devem ser evitados, são *links* desnecessários que causam uma expectativa do usuário que com certeza irá se desapontar;
- Usuários não gostam de rolar a tela, informações importantes devem ser colocadas no topo, de forma que apareça logo que a página é carregada;
- Menus e barras de navegação devem ser projetados de forma consistente, e devem estar disponíveis em todas as páginas que o usuário irá navegar. Não se deve contar com as funcionalidades do *browser*;
- Opções de navegação devem ser óbvias, mesmo para o usuário casual. O usuário não pode ficar procurando pela tela até encontrar o que deseja.

# Considerações

- Pessoas gastam horas clicando em um *site* para encontrar uma simples informação.
- “Surfar” na *Web* é muito mais difícil do que parece.
- Os usuários (potenciais clientes) estão conseguindo navegar no seu site?
- Por que os humanos têm que se adaptar a tecnologia? Por que a tecnologia não se adapta aos humanos?
- Sua empresa não é o público-alvo.
- Você deve escrever seu *site* em uma linguagem que seus usuários compreendam.
- Grandes erros continuam acontecendo na *Web*: *links* quebrados, cores não padronizadas, *URLs* que mudam e fazem com que *links* externos para seu *site* fiquem quebrados.
- Na Internet, sobrevive o mais fácil de usar.
- Se o usuário não consegue achar o produto, ele não irá comprá-lo.
- Atenção com a usabilidade aumenta a porcentagem de visitantes que se tornam clientes.

# Testes

- Assim como na Engenharia de Software, na Engenharia para a *Web* as atividades de teste também visam encontrar erros
- Na *Web* o desafio é ainda maior, pois os aplicativos podem ser acessados utilizando diferentes *browsers*, sistemas operacionais, plataformas de hardware, etc

# Testes

- O modelo de conteúdo é revisto para descobrir erros
- O modelo de projeto é revisto para descobrir erros de navegação
- Componentes selecionados passam por um processo de teste de unidade
- A arquitetura é construída e testes de integração são conduzidos
- O aplicativo já integrado é testado em sua funcionalidade geral e conteúdo fornecido
- O aplicativo é implementado em diferentes configurações de ambientes e testado em sua compatibilidade com cada configuração
- O aplicativo é testado por uma população controlada de usuários

# Gerenciamento

- Muitos desenvolvedores acabam optando por pouco ou nenhum gerenciamento por causa do curto período de desenvolvimento e da constante evolução de aplicativos para a *Web*
- A combinação de tarefas técnicas e não técnicas que ocorrem são um desafio para qualquer grupo de profissionais
- Para evitar confusões, frustrações e falhas, um planejamento precisa ocorrer, riscos precisam ser considerados, um cronograma precisa ser estabelecido e acompanhado, e mecanismos de controle precisam ser definidos

# Equipe de Desenvolvimento

- **Desenvolvedores e Provedores de Conteúdo:** coleta e geração de conteúdo, vem de diversas áreas não relacionadas com software
- ***Web Publisher:*** organiza conteúdo, faz a conexão entre o pessoal técnico que constrói o aplicativo e o pessoal não-técnico que desenvolve o conteúdo
- ***Web Engineer:*** se envolve em várias atividades durante o desenvolvimento (levantamento de requisitos; análise; projeto arquitetural, de navegação e de interface; implementação; e teste)

# Equipe de Desenvolvimento

- **Especialista de Suporte:** serão responsáveis por dar continuidade no suporte ao aplicativo para a *Web*, responsável pelas correções, adaptações e melhorias no site
- **Administrador (WebMaster):** responsável pela operação do *site* no dia-a-dia, incluindo desenvolvimento e implementação de regras para operação do aplicativo; fixação de procedimentos de suporte; implementação de procedimentos de segurança e direitos de acesso; medição e análise de tráfego; coordenação de mudanças de procedimentos; coordenação de especialistas de suporte



# Equipe de Desenvolvimento

- são normalmente menores que as equipes de desenvolvimento de software tradicional
- no desenvolvimento de software tradicional as equipes são divididas em unidades menores para resolver diferentes problemas e executar diferentes tarefas
- no desenvolvimento para a *Web*, as equipes são divididas em grupos multidisciplinares, que construirão diferentes seções do aplicativo para a *Web*, mas em geral irão trabalhar em problemas similares



# Gerenciamento de Projeto

- O desenvolvimento de aplicativos para a *Web* é uma área relativamente nova e há poucos dados históricos que podem ser utilizados para fazer estimativa
- Até agora, nenhum tipo de métrica foi publicado e ainda há pouca discussão de como devem ser estas métricas
- Estimativas são baseadas apenas em experiências com projetos similares
- Quase todo aplicativo para a *Web* quer inovar em alguma coisa, isto acaba fazendo com que estimativas baseadas em experiência com outros projetos, apesar de úteis, estejam sujeitas a uma alta margem de erro.
- Grande parte dos aplicativos na *Web* é construída por terceiros, especializados neste tipo de desenvolvimento

# Tarefas da empresa contratante:

- Muitas das atividades de análise devem ser feitas internamente, incluindo a definição do público-alvo; dos objetivos; das informações e serviços a serem fornecidos; e das medidas quantitativas e qualitativas que serão utilizadas para medir o sucesso.
- Um esboço do projeto deve ser criado, pois economizará tempo e custo para o desenvolvedor, que terá uma idéia melhor de como deverá ser o aplicativo.
- Um esboço do cronograma deve ser definido e acompanhado.
- Os níveis de interação entre o contratante e o contratado devem ser identificados, incluindo as responsabilidades de cada um.



# Gerenciamento de Configuração

- estratégias utilizadas na Engenharia de Software são aplicáveis, porém táticas e ferramentas devem ser adaptadas para as características dos aplicativos para a *Web*

# Limitações a serem consideradas:

- **Conteúdo:** um aplicativo para a *Web* típico possui bastante conteúdo. O desafio é organizá-lo em um conjunto racional de objetos e então estabelecer mecanismos de controle de configuração apropriados.
- **Pessoas:** como o desenvolvimento do aplicativo para a *Web* é contínuo, qualquer pessoa pode criar conteúdo, muitas delas não têm conhecimentos em engenharia de software e desconhecem as necessidades de gerenciamento de configuração.
- **Escalabilidade:** as técnicas e controles aplicados a aplicativos para a *Web* pequenos não são bem escaláveis, é comum ver aplicativos crescendo em tamanho e complexidade, e com isso pequenas mudanças acabam tendo efeitos inesperados e problemáticos.
- **Política.** Quem é o dono do aplicativo? Esta questão é bastante discutida em grandes e pequenas companhias, e a resposta tem impacto significativo nas atividades de gerenciamento e controle.

# Conclusão

- A maneira desordenada com que a maior parte dos aplicativos para a *Web* são construídos é preocupante, principalmente quando analisamos as pesquisas que mostram que a grande maioria dos *sites* tem problemas de funcionalidade e/ou usabilidade
- As técnicas criadas e já consolidadas na Engenharia de Software podem ser utilizadas e isto é uma vantagem. Porém, é preciso fazer algumas adaptações, tendo muito cuidado com todas as características específicas de sistemas para a *Web*

# Conclusão

- Infelizmente poucos desenvolvedores utilizam as técnicas de engenharia, e os poucos que utilizam o fazem de forma restrita ou errônea. Este é um problema que também ocorre na Engenharia de Software, mas na Engenharia para a *Web* a incidência é ainda maior
- Os problemas ocorrem principalmente nas áreas de Análise, Requisitos, Testes, Validação e Manutenção, que são áreas importantes que ainda não recebem a atenção necessária, pois a maior parte dos desenvolvedores concentram-se apenas na fase de Implementação

# Conclusão

- Os estudos mostram que o uso de técnicas de engenharia é eficaz e eficiente, portanto os esforços devem ser concentrados nesta adaptação das técnicas utilizadas na Engenharia de Software, incluindo o desenvolvimento de métricas e técnicas para o gerenciamento de projeto.
- Paralelamente precisa haver um trabalho para conscientizar desenvolvedores da importância do uso destas técnicas, mostrando como as mesmas poderiam tornar seu trabalho mais eficiente, trazendo também melhores resultados

# Referências Bibliográficas

- **[BIE98]** BIEBER, Michael. “Web Engineering”. New Jersey Institute of Technology, 1998. <<http://www-ec.njit.edu/~bieber/web-engineering.html>>
- **[BRE01]** BREVE, Fabricio e WELLER, Daniel. “Métodos de Avaliação para Sites de Entretenimento”. Universidade Metodista de Piracicaba, 2001.
- **[GIN01]** GINIGE, Athula e MURUGESAN, San. “Web Engineering: An Introduction”. IEEE Multimedia. Janeiro-Março 2001.
- **[GIN01b]** GINIGE, Athula. “Engineering A Better Website”. University Of Western Sydney. Austrália, Outubro de 2000. <<http://aeims.uws.edu.au/talks/eng-web-sites.pdf>>
- **[GIN01c]** GINIGE, Athula. “Web Engineering in Action”. University Of Western Sydney. Austrália, Outubro de 2000 <<http://aeims.uws.edu.au/talks/Webe-in-action.pdf>>
- **[MCD01]** MCDONALD, Andrew e WELLAND, Ray. The University, Glasgow G12 8QQ, Scotland. 2001. <<http://www.dcs.gla.ac.uk/~andrew/webe2001.pdf>>
- **[MUR00]** MURUGESAN, San. “Web Engineering For Successful Web Application Development”. University Of Western Sydney. Austrália, Outubro de 2000. <<http://aeims.uws.edu.au/Talks/Web-ApWeb2000.PDF>>
- **[NIE00]** NIELSEN, Jakob. “Designing Web Usability”. New Riders Publishing, 2000.
- **[PRE01]** PRESSMAN, Roger S. “Software Engineering: A Practitioner’s Approach”. McGraw-Hill. 5ª edição. 2001.