



COLÉGIO PROTÁSIO ALVES







CRONOGRAMA DAS AULAS :

20 SEMANAS





COLÉGIO PROTÁSIO ALVES

AULA 1: ANÁLISE DE PROJETOS

ANÁLISE:

É o mesmo que um **estudo detalhado sobre algo**, podendo ser **aplicada** em **diferentes áreas** do conhecimento como



forma de **observar minuciosamente**
determinado tema.



AULA 1: ANÁLISE DE PROJETOS

PROJETO: Segundo o PMBOK, é um **esforço temporário** empreendido para **criar um produto, serviço ou resultado exclusivo**. Ou seja, um projeto é **tudo aquilo que precisamos realizar para gerar algo novo**: seja uma **casa**, um **sistema informatizado**, um **estudo/pesquisa**, um **trabalho de conclusão de curso**, uma **contratação** ou uma **compra importante**.

O QUE É UM PROJETO DE SISTEMAS – SOFTWARES



Um projeto de sistemas traduz para o papel como serão atendidas as requisições dos clientes que precisam de uma solução tecnológica.

O objetivo é definir e detalhar um modelo de software que seja viável e capaz de solucionar as necessidades do cliente.

AULA 1: ANÁLISE DE PROJETOS

O QUE SÃO DADOS?

Dados são observações documentadas ou resultados da medição. A disponibilidade dos dados oferece oportunidades para a obtenção de informações. **Os dados** podem ser obtidos pela percepção através dos



sentidos (por exemplo observação) ou pela execução de um processo de medição.

O QUE É INFORMAÇÃO?

É um **conjunto de dados organizados** que fazem sentido e **referência** a um acontecimento, ***um fato ou um fenômeno, que no seu contexto tem um determinado significado*** para o receptor, cujo fim é reduzir a incerteza ou incrementar conhecimento sobre algo.

AULA 1: ANÁLISE DE PROJETOS

QUAL A **DIFERENÇA** ENTRE **DADOS** E **INFORMAÇÃO**?



Os **dados** são o idioma de **entrada** para um computador e para os analistas.

Já a **informação** seria uma espécie de **linguagem de saída**, **interpretada e transformada** em elementos relevantes para o negócio.

Enquanto os **dados** podem ser **meros números**, fatos não processados, a **informação** é o que **dá sentido ao que foi analisado**.



AULA 1: ANÁLISE DE PROJETOS

ANÁLISE DE UM PROJETO DE SISTEMAS DE



INFORMAÇÃO. É uma área da computação que visa **descrever as funções de um sistema**, verificar sua **funcionalidade**, **elaborar e implementar soluções** para diferentes problemas identificados durante a fase de análise.

AULA 2: ANÁLISE DE PROJETOS

INTRODUÇÃO:

Existe um encadeamento específico dessas fases, para a construção do sistema dá-se o nome de **Ciclo de vida**. O processo de software é o conjunto de atividades que constituem o desenvolvimento de um sistema

computacional. Estas atividades são agrupadas em fases, como:

- **Definição de requisitos,**
- **Análise, projeto,**
- **Desenvolvimento,**
- **Teste e implantação.**



AULA 2: CICLO DE VIDA DE

SISTEMA







AULA 2: CICLO DE VIDA DE SISTEMA

Levantamento das necessidades: Também chamado de **análise de requisitos** identifica as **necessidades** de informações da organização.

Análise de alternativas: Consiste na identificação e **avaliação de sistemas alternativos**.

Projeto: Trata da **construção das especificações detalhadas** para o projeto selecionado. Essas especificações incluem o projeto das **interfaces, banco de dados, características físicas do sistema**, tais como número, tipos e localizações das estações de trabalho, hardware de



processamento, o cabeamento e os dispositivos de rede. Deve especificar os procedimentos para **testar** o sistema completo antes da instalação.



AULA 2: CICLO DE VIDA DE SISTEMA

Desenvolvimento:

Inclui o desenvolvimento ou **aquisição do software**, a provável **aquisição do hardware** e o **teste** do novo sistema.

Implementação:

Ocorre após o sistema ter passado satisfatoriamente por testes de



aceitação.

O sistema é transferido do ambiente de desenvolvimento para o ambiente de produção. O sistema antigo (**se existir**) deve migrar para o novo.



AULA 2: CICLO DE VIDA DE SISTEMA

Manutenção:

Refere-se a **todas as atividades** relacionadas a um sistema depois que ele é implementado.

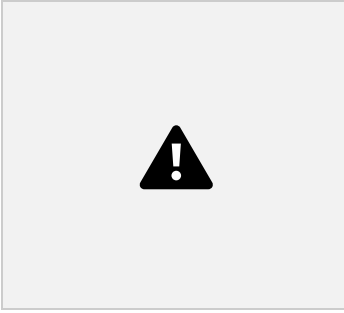
Deve **incluir atividades tais como a correção** de software que não

funcione corretamente, a **adição de novos recursos** aos sistemas em resposta às novas demandas dos usuários.

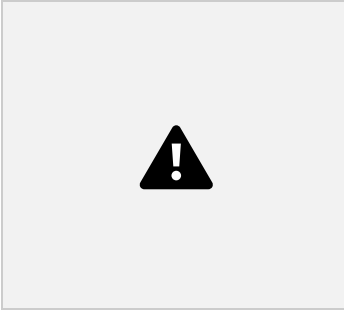
Alguns modelos combinam **desenvolvimento e implementação** em uma **única etapa**. Outros combinam o **levantamento e a análise** das necessidades também em uma única etapa.



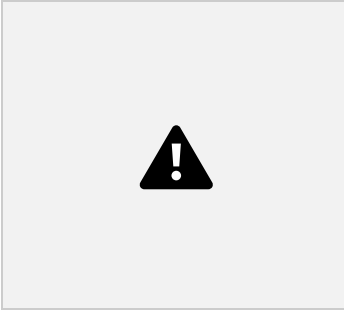




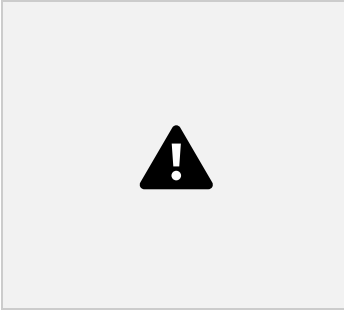




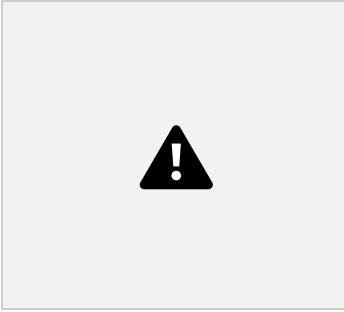
















AULA 3:







Modelos de Desenvolvimento de Software:

Definição:

São abordagens estruturadas para organizar e controlar o processo de desenvolvimento de software.

Objetivo: Guiar as atividades de desenvolvimento para produzir um software de qualidade dentro do prazo e orçamento estabelecidos.

Exemplos: Modelo Cascata, Modelo em V, Modelo Espiral,



Prototipagem, Modelos Ágeis (Scrum, Kanban, etc.).

Modelagem

Definição: Processo de **criar representações** abstratas (modelos) de aspectos do software para entender, especificar, projetar e documentar o sistema.

Objetivo: Facilitar a compreensão do sistema, detectar erros e omissões nos requisitos e design, e comunicar detalhes entre os membros da equipe.

Exemplos: Diagramas de Caso de Uso, Diagramas de

Classes, Diagramas de Sequência, Diagramas de Atividade (usando notações como UML - Unified Modeling Language).



Diferenças Principais:

FOCO:

Modelos de Desenvolvimento de Software:

Centram-se no **processo e nas fases de desenvolvimento** do software.



Modelagem de Software:

Centra-se na **representação e especificação do sistema**,
suas funcionalidades e arquitetura.

Diferenças Principais:

NÍVEL DE ABSTRAÇÃO:

Modelos de Desenvolvimento de Software: Alto nível,
abrangendo **todo o ciclo de vida** do software.



Modelagem de Software:

Pode variar, desde modelos de alto nível (**visão geral do sistema**) até modelos de baixo nível (**detalhes de implementação**).

Diferenças Principais:

Propósito:

Modelos de Desenvolvimento de Software: Definir como o desenvolvimento será conduzido, desde a concepção até a



entrega e manutenção.

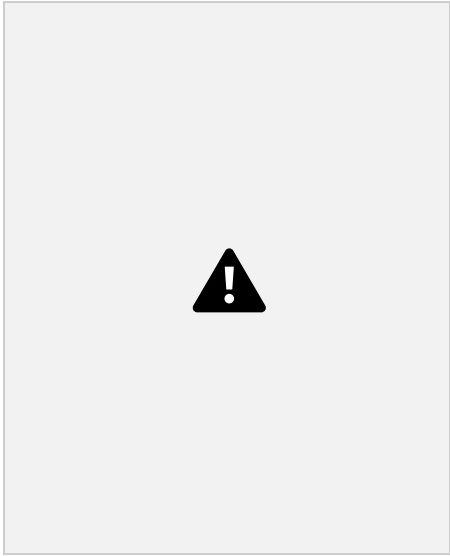
Modelagem de Software: Prover uma ferramenta visual e descritiva para análise, design e comunicação do sistema.



AULA 3: MODELO EM CASCATA



AULA 7: MODELO EM CASCATA







AULA 7: MODELO EM CASCATA

Se, por

exemplo, as **necessidades dos usuários mudarem durante o projeto**, não existe nenhum mecanismo formal para ajustar o processo de desenvolvimento.

O **uso deste modelo significa**, que nenhum componente do sistema será entregue até a proximidade final do projeto.

Frequentemente esta demora na entrega conduz a tensões entre usuários e desenvolvedores, especialmente se os



prazos finais são ultrapassados.







AULA 4: MODELO EM CASCATA V



Existe uma variação do Modelo em Cascata que é chamado de **modelo V**. Nesse modelo, procura-se **ênfatisar a estreita relação entre as atividades de teste** (**teste de unidade, teste de integração, teste de sistema e teste de aceitação**) e as demais fases do processo. Pressman (2011) explica que à medida que a **equipe de software desce** em direção ao **lado esquerdo do V**, os **requisitos básicos do problema são refinados** em representações progressivamente cada vez mais detalhadas e técnicas do problema e de sua solução. Como mostra a Figura



O Modelo em Cascata é o modelo mais antigo na engenharia de software, e nos últimos anos recebeu várias críticas, conforme relatadas por Pressman (2011, p. 61): **Projetos reais raramente seguem o fluxo sequencial que o modelo propõe.**

Embora o **modelo linear** possa conter interações, ele o faz indiretamente. Como **consequência, mudanças podem provocar confusão** à medida que a equipe de projeto prossegue.

- **Frequentemente**, é difícil para o cliente estabelecer explicitamente todas as necessidades. O **modelo em cascata** requer isso e **tem dificuldade** para adequar a incerteza natural que existe no início de muitos projetos.

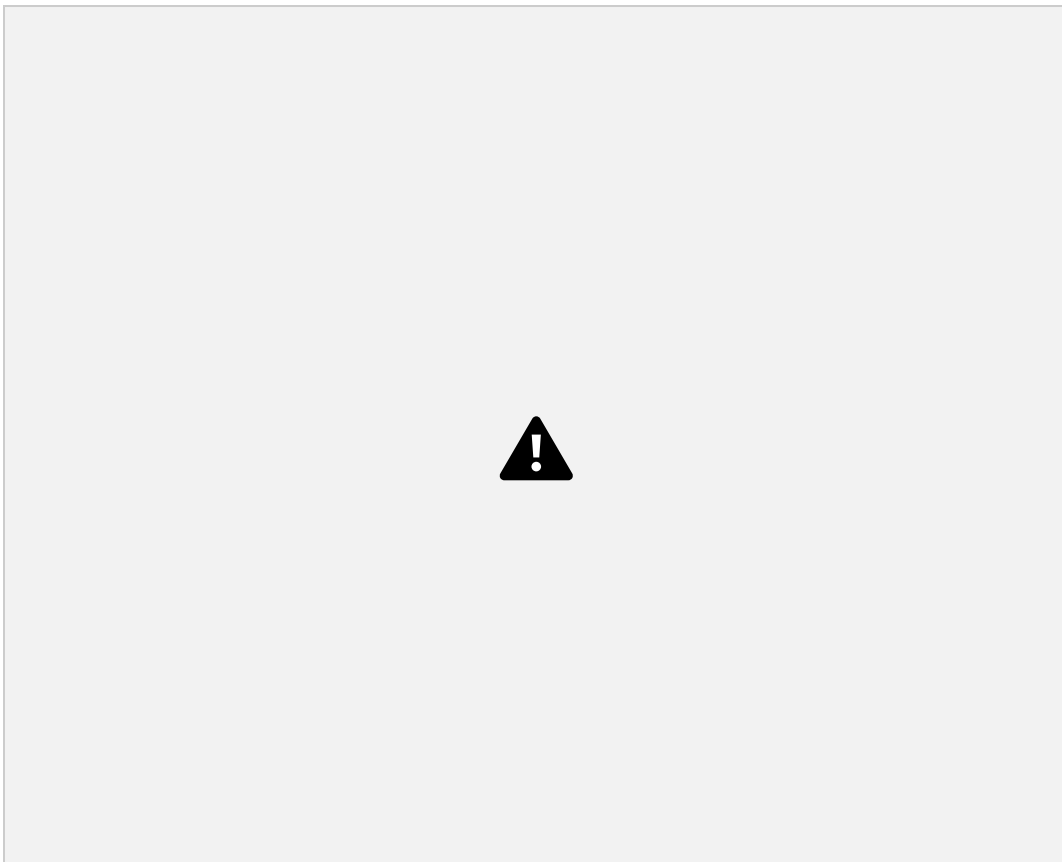
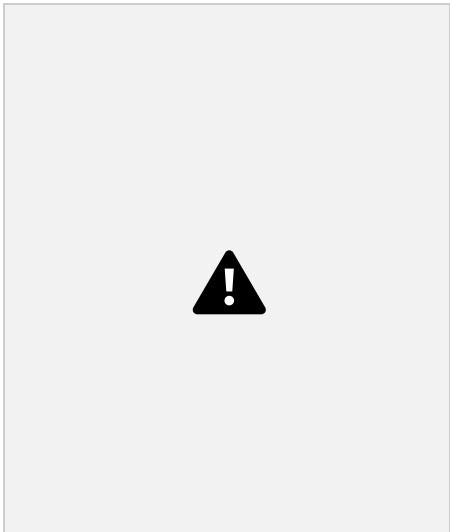
- O cliente deve ter paciência. **Uma versão operacional do(s) programa(s) não estará disponível** antes de estarmos próximo do final do

projeto. Um erro grave, se não detectado até o programa operacional ser visto, pode ser desastroso.



AULA 4: MODELO EM CASCATA

V





AULA 4: MODELO INCREMENTAL

Nem todos os projetos conseguem ter todos os requisitos **são razoavelmente** bem definidos, e devido ao tamanho do sistema **a ser desenvolvido**, isso torna impossível a **adoção de um modelo sequencial**, especialmente pela necessidade de disponibilizar uma **versão** para o usuário rapidamente.

Para esses casos o **Modelo Incremental** é o mais indicado. Pressman (2011) diz que o Modelo Incremental

combina elementos dos fluxos de processos lineares e paralelos.



AULA 4 : MODELO INCREMENTAL

O princípio fundamental desse modelo é que, seja iteração ou ciclo, **uma versão operacional do sistema é gerada e entregue para avaliação e uso detalhado do cliente.**

Para que isso aconteça, **os requisitos** têm que ser **levantados detalhadamente** e há de se constatar que o **sistema é modular**, de modo que se possa planejar o desenvolvimento em incrementos.

O primeiro incremento, tipicamente, contém **funcionalidades centrais**,



tratando dos requisitos básicos.

A cada ciclo, novas funcionalidades são adicionadas e as funcionalidades providas anteriormente podem ser modificadas para melhor satisfazer às necessidades dos clientes/ usuários.



AULA 4 : MODELO

INCREMENTAL







AULA 4 : MODELO INCREMENTAL

O princípio

fundamental desse modelo é que, seja iteração ou ciclo,

uma versão operacional do sistema é gerada e entregue para avaliação e uso detalhado do cliente. Para que isso aconteça, os requisitos têm que ser levantados detalhadamente e há de se constatar que o sistema é modular, de modo que se possa planejar o desenvolvimento em incrementos. O primeiro incremento, tipicamente, contém funcionalidades centrais, tratando dos requisitos básicos. A cada ciclo, novas funcionalidades são adicionadas e as funcionalidades providas anteriormente podem ser modificadas para melhor satisfazer às necessidades dos clientes/ usuários. Outras características são tratadas



em ciclos subsequentes. Dependendo do tempo estabelecido para a liberação dos incrementos, algumas atividades podem ser feitas para o sistema como um todo ou não.



MODELOS DE PROCESSOS EVOLUCIONÁRIOS

Pressman (2011) afirma *que o software, assim como todos os sistemas complexos, evolui ao longo do tempo e, conforme o desenvolvimento deste software avança, temos mudanças nas necessidades de negócio e de produtos, uma vez que mudam frequentemente.*

Torna inadequado seguirmos um **planejamento em linha reta** de determinado produto. Muitas vezes, os **prazos**



reduzidos, a necessidade de atender à pressão comercial e até mesmo da concorrência e em outras situações como essas ou similares, faz-se necessário um **modelo de processo que tenha sido **projetado especificamente** para **desenvolver um produto que evolua ao longo do tempo.****



MODELOS DE PROCESSOS EVOLUCIONÁRIOS

MODELOS EVOLUCIONÁRIOS: São caracterizados por **serem interativos** e apresentarem **características que possibilitem desenvolvermos versões cada vez mais completas** do software. Os Processos Evolucionários se **caracterizam** por dois modelos comuns:

Prototipação e Espiral.

MODELO ESPIRAL: Foi originalmente proposto por Boehm em 1988. Para definir esse modelo de uma maneira mais simples, basta analisá-lo como um **Modelo em Cascata**, onde *cada fase é precedida por uma análise de risco e sua execução é feita evolucionaria mente (ou incrementalmente)*.



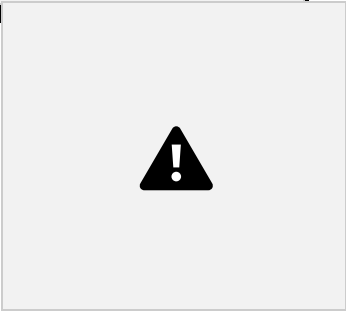
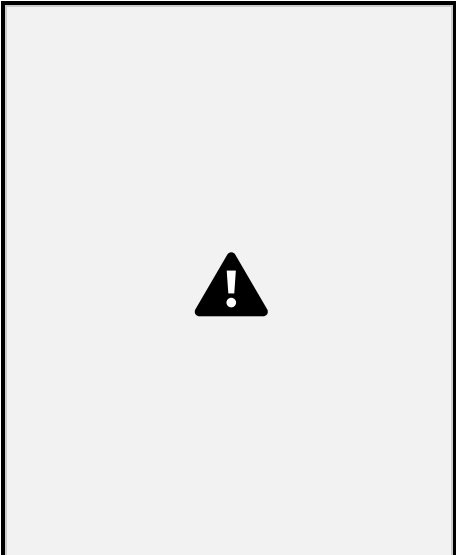
MODELO ESPIRAL:

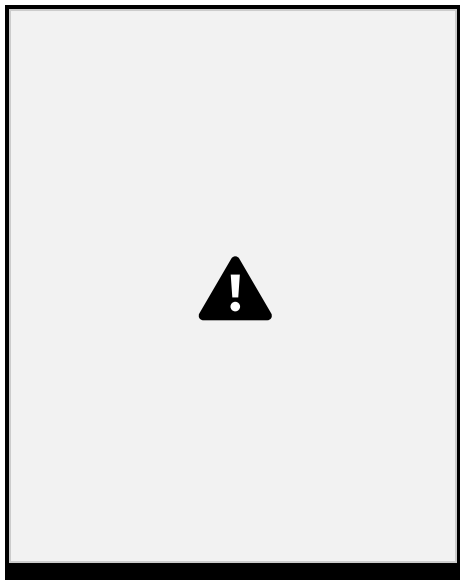
Neste modelo de desenvolvimento é **possível avaliar riscos de projeto, tomando-se decisões baseadas na experimentação de diferentes soluções**. A **dimensão radial** representa o **custo** acumulado

atualizado e a **dimensão angular** representa o **progresso** através da espiral. **Cada setor da espiral** corresponde a uma tarefa (fase) do desenvolvimento.

No modelo original foram **propostas quatro tarefas** (**fases ou quadrantes**) onde, por exemplo, **na primeira volta** pode **gerar** como **resultado a especificação do produto**, na **espiral subsequente** o **resultado** pode ser o **protótipo**, e nas voltas seguintes podem ser *relativos a versões aprimoradas do sistema* (PRESSMAN, 2008, p.38).











MODELO ESPIRAL:

Divide o projeto em **subprojetos menores**, cada um administrado mais facilmente do que o projeto principal e cada um fornecendo um produto em funcionamento e pronto para a entrega. Ao final de cada projeto, os subprojetos restantes **são redefinidos considerando o *feedback* do usuário**.

A abordagem em espiral entrega o produto rapidamente. Não há documentação trabalhosa das especificações, porque os usuários podem revisar o produto em versões posteriores.

Os usuários podem ver o progresso e julgar quanto tempo passará até



que o sistema em desenvolvimento satisfaça suficientemente suas necessidades para que possa substituir o sistema existente.



AULA 6: PROTOTIPAÇÃO

Segundo **Pressman** : Explica o funcionamento deste paradigma de maneira reduzida. Na fase de **comunicação**, faz-se uma reunião com os envolvidos para **definir** os objetivos gerais do software, **identificar** quais requisitos já são conhecidos e esquematizar quais áreas necessitam,



obrigatoriamente, de uma definição mais ampla. Uma interação de prototipação é **planejada rapidamente** e ocorre a modelagem (na forma de um “projeto rápido”).



AULA 6 : PROTOTIPAÇÃO

Durante a fase de modelagem do projeto rápido: se concentra em uma representação daqueles aspectos do software que serão visíveis aos usuários finais, como, por exemplo, **o layout da interface com o usuário ou formatos de exibição na tela**. E, finalmente, a **construção de um protótipo**: Nesta fase é feita a construção do protótipo, que é

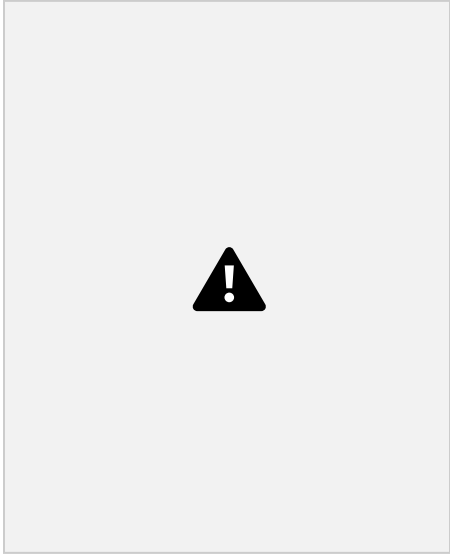


empregado e avaliado, conforme vemos na Figura abaixo

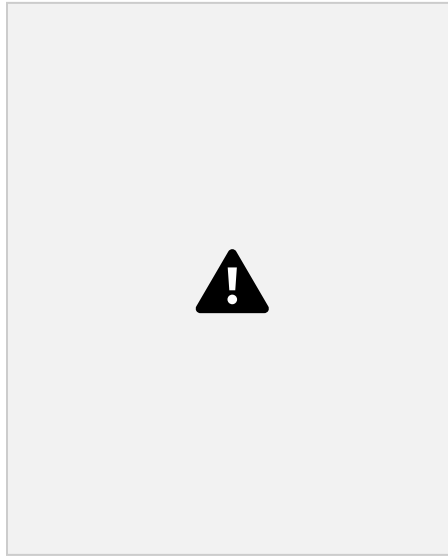
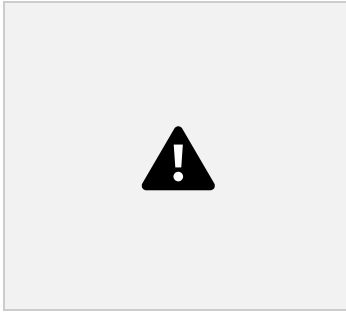
Sommerville (2011, p. 31) alerta que, às vezes, os **desenvolvedores são pressionados pelos gerentes** para entregar protótipos **descartáveis**, especialmente quando há atrasos na entrega da versão final do software. **No entanto, isso costuma ser desaconselhável:**



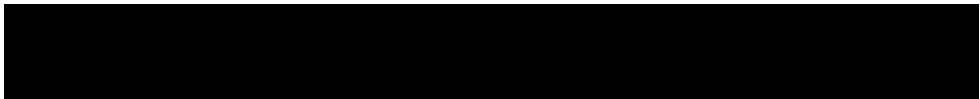
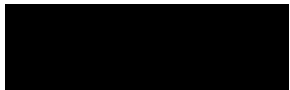
AULA 6: PROTOTIPAÇÃO











Vantagens sobre a abordagem em cascata:

Diminui o tempo entre a análise e a implementação;

Assegura que o novo sistema **satisfaça as necessidades** do usuário;

Mostra os benefícios de um novo sistema antes que o



esforço e os custos se tornem excessivos;

Explora as potencialidades que os usuários têm em articular mais facilmente aquilo de que não gostam em um sistema do que aquilo que apreciam nele.

AULA 6: PROTOTIPAÇÃO

Desvantagens em relação ao modelo em cascata:

Elevar as expectativas dos usuários a níveis que os desenvolvedores não podem atender dentro de seu orçamento.

Os programas de software que permitem aos **desenvolvedores desenvolver rapidamente a interface de um novo sistema e de**



customizá-la rapidamente em atendimento às solicitações do usuário, atualmente, custam caro.

Atrasa a demonstração da funcionalidade do sistema. Metade da funcionalidade pode não aparecer até que se atinjam os 10% finais do cronograma de desenvolvimento . **Levantamento das Necessidades, Análise de Alternativas Projeto ,Desenvolvimento , Implementação, Manutenção.**



AULA 6 : ANLISE DE PROJETOS

Pode ser **impossível ajustar** o protótipo para atender aos **requisitos não funcionais**, como requisitos de desempenho, proteção, robustez e confiabilidade, que foram ignorados durante



o desenvolvimento do protótipo. **# Mudanças rápidas** durante o desenvolvimento inevitavelmente significam que o protótipo não está documentado. **A única especificação do projeto** é o código do protótipo. Para a manutenção a longo prazo, isso não é bom o suficiente.

As mudanças durante o desenvolvimento do protótipo provavelmente terão degradado a estrutura do sistema. O sistema será difícil e custoso de ser mantido.

Padrões de qualidade organizacional geralmente são relaxados para o desenvolvimento do protótipo

AULA 7: SELEÇÃO DE UM CAMINHO A





melhor abordagem para um determinado projeto depende, em grande parte, da natureza do projeto e da natureza da organização.

A **ABORDAGEM EM CASCATA** funciona melhor com **projetos de grande porte**, complexos, que têm numerosos interessados, afetam a empresa toda e não podem ser facilmente divididos em subprojetos. Ela também funciona bem com organizações que têm uma cultura formal e uma estrutura hierárquica.



AULA 7: SELEÇÃO DE UM CAMINHO A

ABORDAGEM EM ESPIRAL e a **PROGRAMAÇÃO ÁGIL** funcionam bem nas **organizações dinâmicas**, que podem tolerar a ambiguidade e necessitam obter resultados rapidamente.



O **CAMINHO EM ESPIRAL** pode apresentar melhores resultados quando adotado para projetos que **se dividem** facilmente em **subprojetos** e para **projetos mais simples**, em especial o desenvolvimento de sistemas de **usuário único**

onde as *necessidades do usuário* são *difíceis de especificar* ou *mudam rapidamente*.

AULA 7: SELEÇÃO DE UM CAMINHO

A **PROTOTIPAGEM** funciona melhor para projetos de pequeno e médio portes.

Ela funciona bem onde a cultura suporta equipes funcionalmente mistas.



A prototipagem pode ser combinada com a abordagem em espiral e ser usada para um ou mais dos subprojetos em um desenvolvimento em espiral.

ATIVIDADE 2:

A) CRIAR UM MAPA MENTAL PARA CADA TIPO DE MODELO:

- MODELO EM CASCATA;
- MODELO EM CASCATA V
- INCREMENTAL;



- ESPIRAL;
- PROTOTIPAGEM;
- METODOLOGIAS ÁGEIS(SCRUM, LEAN, KANBAN) ;

**PRA QUE SERVE;
VANTAGENS E DESVANTAGENS;
ONDE E COMO UTILIZAR.**