

Graduação de Tecnologia em Análise e Desenvolvimento de Sistemas

Métricas e auditoria de software

Prof. Reinaldo Freitas



Agenda

- **Introdução**
- **Métricas em Projetos Tradicionais**
- **Métricas em Projetos Ágeis**
- **Análise de Pontos de Função**
- **Melhoria de Processo de Software**
- **Normas de Qualidade**
- **Auditoria de Sistemas**
- **Referencias**



Introdução



Introdução



Por que Medir?

- Não se pode gerenciar o que não se pode medir;
- Se você não sabe para onde quer ir, pode seguir qualquer caminho, o que resultará em um resultado imprevisível.

Introdução



Por que medir software?

- Entender e aperfeiçoar o processo de desenvolvimento;
- Melhorar a gerência de projetos e o relacionamento com clientes;
- Reduzir frustrações e pressões de cronograma;
- Gerenciar contratos de software;
- Indicar a qualidade de um produto de software.

Introdução



Medida

- É a quantificação de dados em um padrão e qualidade aceitáveis (exatidão, completude, consistência, temporalidade).
- É a avaliação de uma grandeza por meio da comparação com outra grandeza da mesma espécie tomada como unidade. Por exemplo, avaliar o comprimento de um material, pode-se utilizar o metro como unidade de medida.

Introdução



Métrica

- É uma extrapolação de medidas, isto é, uma conclusão com base em dados finitos.
- Pode ser entendida como a relação entre duas medidas de grandezas iguais ou diferentes. Por exemplo, poderia ser a razão entre o número total (quantidade) de atendimentos de um Service Desk e o número de total de horas (tempo) consumidas pelos atendentes.

Introdução



Métrica

- Podem ser divididas quanto a sua aplicação em:
 - ❑ **Métricas de processo:** São métricas que dizem respeito à qualidade do processo. Elas são usadas para medir a eficiência e eficácia de processos.
 - ❑ **Métricas de projeto:** São métricas que dizem respeito principalmente à qualidade e desempenho do projeto. Elas são usadas para quantificar defeitos, custos, cronograma, produtividade e estimativa para vários recursos e entregas de projeto.

Introdução



Métrica

- Podem ser divididas quanto a sua orientação em:
 - ❑ **Métricas orientadas ao tamanho:** São medidas diretas do software e do processo por meio do qual ele é desenvolvido. Alguns exemplos são: **Linhas de código (*Lines of Code*), KLOC (*Thousand Lines of Code*) e custo da implementação.** Normalmente são dependentes da linguagem de programação utilizada na codificação do projeto, penalizando programas bem projetados, porém mais curtos. Também são difíceis de estimar no início do projeto.
 - ❑ **Métricas orientadas à função:** São medidas indiretas do software e do processo por meio do qual ele é desenvolvido. São medidas do ponto de vista do usuário, que determina de forma consistente o tamanho e a complexidade de um software. Concentram-se na funcionalidade ou utilidade do software. Alguns exemplos são: **PF (Pontos por Função) e SOD (Speed of Delivery).**

Introdução



Indicador

- É uma representação de forma simples ou intuitiva de uma métrica ou medida para facilitar sua interpretação quando comparada a uma referência ou alvo.
- Representam informações a partir das quais é possível avaliar uma situação e sua evolução histórica. Entretanto, caso os indicadores sejam mal definidos eles podem levar a conclusões erradas. Por exemplo, o aumento do número de defeitos em um mês pode não ser uma informação ruim caso o % de defeitos tenha reduzido no mesmo mês. O aumento da produção pode levar a um número absoluto maior de defeitos, sendo que o % de defeitos pode ter reduzido, o que é uma boa informação.

Introdução



Indicador

- Normalmente são divididos em:
 - ❑ **Indicadores de resultado (*lagging indicators*)**: são utilizados para medir resultados passados. Monitoram o efeito e não permitem mais alterar um dado resultado.
 - ❑ **Indicadores de desempenho (*leading indicators*)**: são utilizados para indicar tendências futuras. Monitoram a causa antes do efeito e caracterizam-se pela possibilidade de alterar o curso para o alcance de um resultado.
- Avaliar apenas indicadores de resultado não é uma boa estratégia.

Introdução



Indicador

- Normalmente são especificados através de um conjunto de informações, como por exemplo:
 - ☐ Título do indicador
 - ☐ Sigla
 - ☐ Unidade de medida
 - ☐ Fonte dos dados
 - ☐ Período de coleta
 - ☐ Regra de cálculo
 - ☐ Forma de coleta
 - ☐ Meta (valores desejados)
 - ☐ Forma de interpretação

Introdução



Exercícios:

- 1) Qual a importância de medir software?
- 2) O que é uma medida?
- 3) O que é uma métrica?
- 4) Qual a diferença entre métricas de processo e de projeto?
- 5) O que é um indicador?
- 6) Qual a diferença entre indicadores de desempenho e de resultado?
- 7) Especifique um indicador de software.

Métricas em Projetos Tradicionais



Métricas em Projetos Tradicionais



Modelo Tradicional

- Também chamado de ciclo de vida clássico ou Cascata (*Waterfall*). É utilizado quando os requisitos são razoavelmente bem compreendidos, ou seja, o trabalho flui da comunicação até a implantação de um modelo razoavelmente linear. É dividido nas seguintes etapas:
 - ☐ **Comunicação:** iniciação do projeto e levantamento de requisitos.
 - ☐ **Planejamento:** Estimativas e cronograma.
 - ☐ **Modelagem:** Análise e projeto do sistema.
 - ☐ **Construção:** Codificação e teste.
 - ☐ **Implantação:** Entrega, manutenção e feedback.

Métricas em Projetos Tradicionais



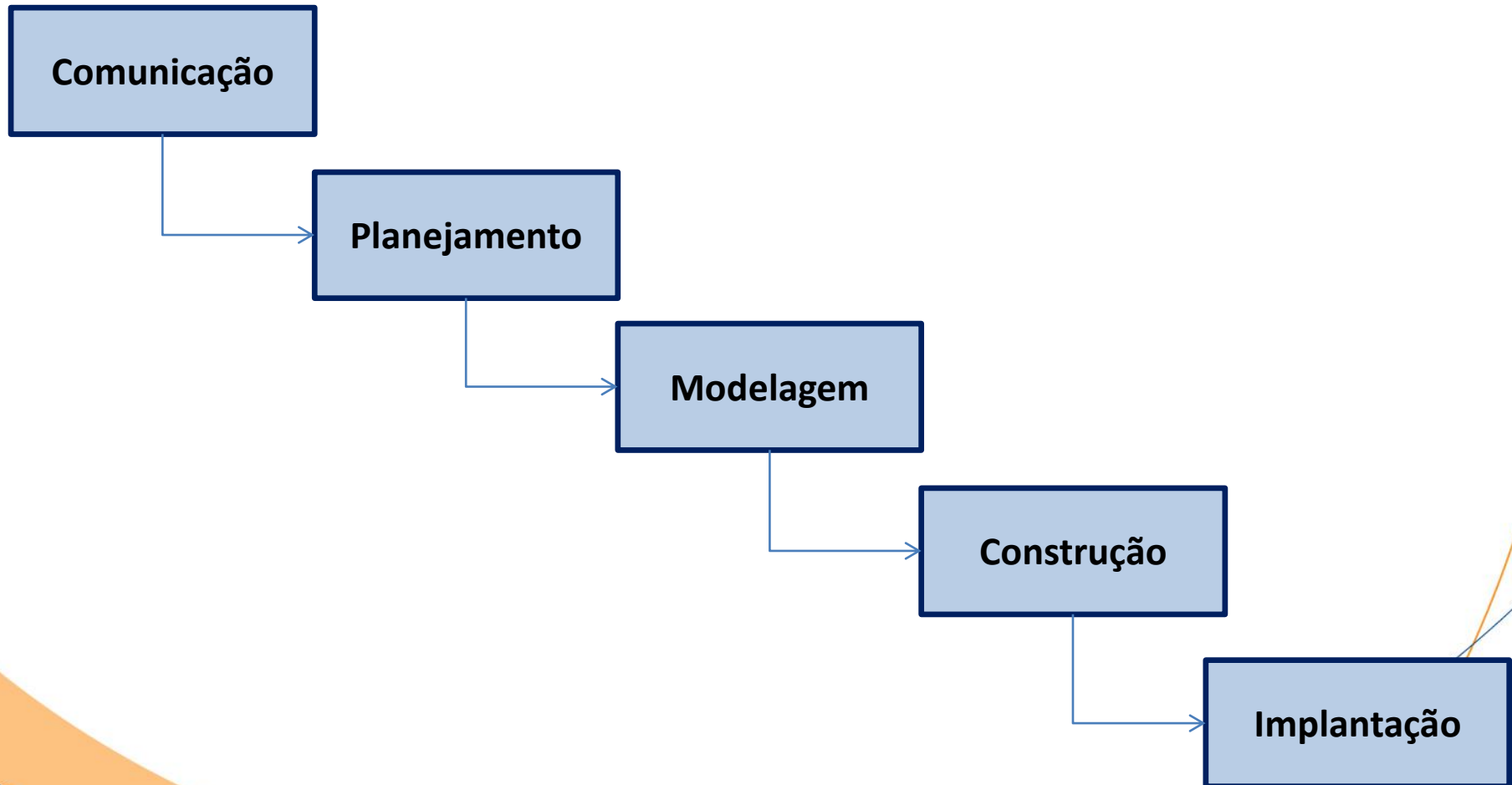
Modelo Tradicional

- Também chamado de ciclo de vida clássico ou Cascata (*Waterfall*). É utilizado quando os requisitos são razoavelmente bem compreendidos, ou seja, o trabalho flui da comunicação até a implantação de um modelo razoavelmente linear. É dividido nas seguintes etapas:
 - ☐ **Comunicação:** iniciação do projeto e levantamento de requisitos.
 - ☐ **Planejamento:** Estimativas e cronograma.
 - ☐ **Modelagem:** Análise e projeto do sistema.
 - ☐ **Construção:** Codificação e teste.
 - ☐ **Implantação:** Entrega, manutenção e feedback.

Métricas em Projetos Tradicionais



Modelo Tradicional



Métricas em Projetos Tradicionais



Tipos de métricas

- Um projeto pode ter vários tipos de métricas, mas principalmente:
 - ❑ **Métricas de qualidade** (Ex.: % de testes sem erros por mês)
 - ❑ **Métricas de desempenho** (Ex.: % de andamento do cronograma)
 - ❑ **Métricas orientadas à função** (Ex.: Pontos de função de um sistema)
- Vamos focar neste momento nas métricas de desempenho de projeto.

Métricas em Projetos Tradicionais



EVM (Earned Value Management).

- É o resultado da **Análise de Valor Agregado (*Earned Value Analysis* - EVA)** de um projeto.
- O EVA é um método que permite ao gerente de projeto medir a quantidade de trabalho realmente executada em um projeto e pode ser utilizado na revisão do custo e prazo do cronograma.
- O EVA fornece um método que permite que o projeto seja medido pelo progresso alcançado. O gerente de projeto pode então, usando o progresso medido, prever o custo total de um projeto e a data de conclusão. Esse método se baseia em uma medida chave conhecida como valor agregado do projeto.

Métricas em Projetos Tradicionais



Earned Value Analysis (EVA)

- O EVA é baseado em 3 medidas principais:
 - ❑ **Valor Planejado (VP):** É quanto do orçamento deve ter sido gasto em uma tarefa até a data atual, de acordo com o custo da linha de base da tarefa (orçamento). Também conhecido pela sigla **PV** (*Planned Value*).
 - ❑ **Valor Agregado (VA):** É o custo realizado do trabalho planejado. Também conhecido pela sigla **EV** (*Earned Value*).
 - ❑ **Custo Real (CR):** É o custo realizado até a data atual. Também conhecido pela sigla **AC** (*Actual Cost*).

Métricas em Projetos Tradicionais



Earned Value Analysis (EVA)

- As medidas PV, EV e AC combinadas geram as seguintes métricas:

❑ **Variação de Custo (Cost Variance - CV):** É a diferença entre o Valor Agregado (EV ou VA) e Valor Real (AC ou CR).

$$CV = EV (VA) - AC (CR)$$

$CV < 0$ - Custo do projeto está acima do orçamento

$CV > 0$ - Custo do projeto está abaixo do orçamento

❑ **Variação de Prazo (Schedule Variance - SV):** É a diferença entre o Valor Agregado (EV ou VA) e Valor Planejado (PV ou VP).

$$SV = EV (VA) - PV (VP)$$

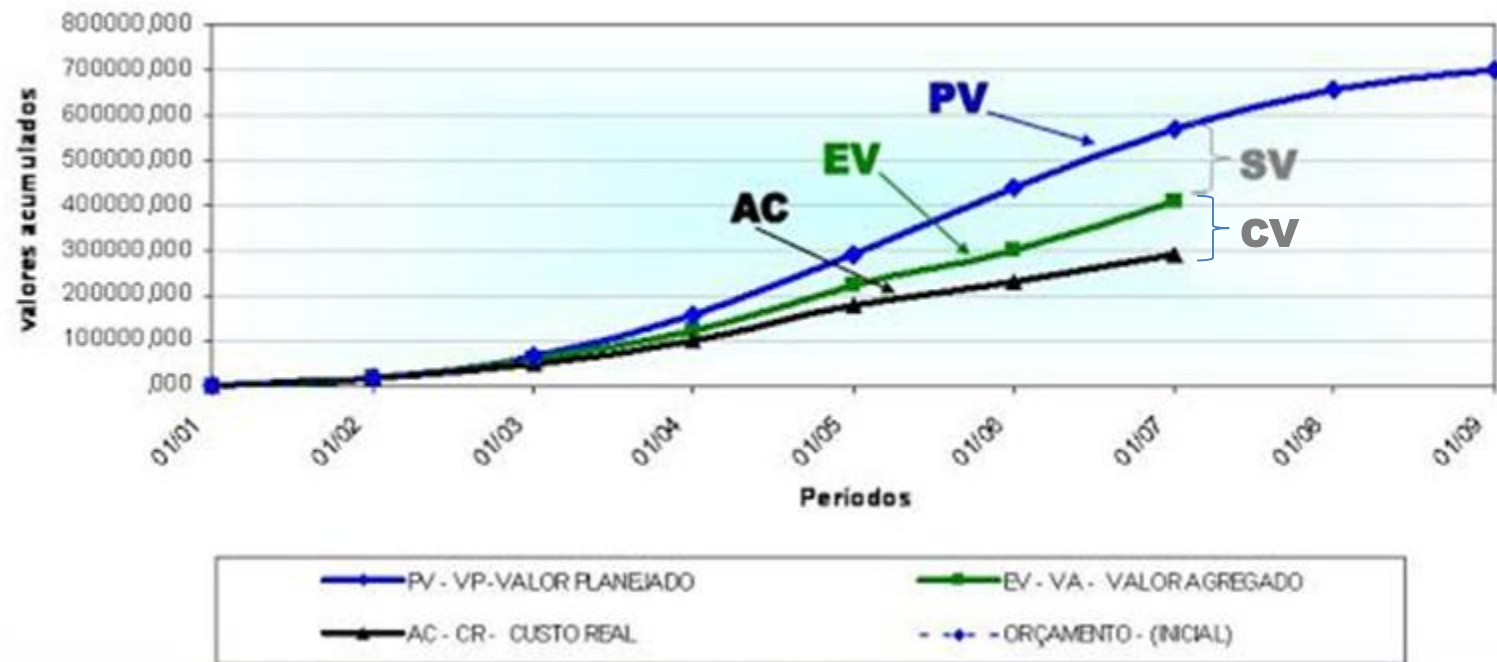
$SV < 0$ - Projeto atrasado

$SV > 0$ - Projeto adiantado

Métricas em Projetos Tradicionais



EVA - Earned Value Analysis



Métricas em Projetos Tradicionais



Earned Value Analysis (EVA)

- Seguem outras métricas que podem ser obtidas a partir das medidas PV, EV e AC:

- ☐ **Índice de Performance de Custo (*Cost Performance Index* - CPI):** É a variação do desempenho do custo planejado em relação ao custo real. Também conhecido como **IDC (Índice de Desempenho de Custo)**.

$CPI (IDC) = EV(VA) / AC(CR)$

$CPI (IDC) > 1$ - Custo está melhor do que o esperado

$CPI (IDC) < 1$ - Custo está pior do que o esperado

- ☐ **Índice de Performance de Prazo (*Schedule Performance Index* - SPI):** É a variação do desempenho do tempo do trabalho realizado em relação ao planejado. Também conhecido como **IDA (Índice de Desempenho de Agendamento)**.

$SPI (IDA) = EV(VA) / PV(VP)$

$SPI (IDA) > 1$ - Cronograma está melhor do que o esperado

$SPI (IDA) < 1$ - Cronograma está pior do que o esperado

Métricas em Projetos Tradicionais



Exercícios:

- 1) Quais são as etapas do modelo tradicional de desenvolvimento de sistemas?
- 2) Quais são os principais tipos de métricas associadas a projetos?
- 3) O que é EVM?
- 4) O que é EVA? Quais são as suas 3 medidas principais?
- 5) Considerando que em um determinada data um projeto tem $PV(VP)=\$100$, $EV(VA)=\$80$ e $AC(CR)=\$40$, responda:
 - a) Qual é o CV do projeto?
 - b) Qual é o SV do projeto?
 - c) Qual é o CPI (IDC) do projeto?
 - d) Qual é o SPI (IDA) do projeto?
 - e) O custo está melhor ou pior do que o valor orçado?
 - f) O cronograma está adiantado ou atrasado?

Métricas em Projetos Ágeis



Métricas em Projetos Ágeis



Modelo de Desenvolvimento Ágil

- Em 2001, Kent Beck e 16 outros notáveis desenvolvedores, produtores e consultores de software assinaram o “Manifesto para o Desenvolvimento Ágil de Software”. Eles declararam que passaram a valorizar:
 - ☐ **Indivíduos e interações** em vez de **processos e ferramentas**
 - ☐ **Software funcionando** em vez de **documentação abrangente**
 - ☐ **Colaboração do cliente** em vez de **negociação de contratos**
 - ☐ **Resposta a modificações** em vez de **seguir um plano**

Métricas em Projetos Ágeis



Modelo de Desenvolvimento Ágil

- Seguem abaixo alguns modelos de desenvolvimento ágil de software:
 - ☐ eXtreme Programming (XP)
 - ☐ Adaptative Software Development (ASD)
 - ☐ Dynamic Systems Development Method (DSDM)
 - ☐ Feature Driven Development (FDD)
 - ☐ Scrum

Métricas em Projetos Ágeis



Scrum

- É um framework leve que ajuda pessoas, times e organizações a gerar valor por meio de soluções adaptativas para problemas complexos.
- O framework Scrum é propositalmente incompleto, apenas definindo as partes necessárias para implementar a teoria Scrum. O Scrum é construído sobre a inteligência coletiva das pessoas que o utilizam. Em vez de fornecer às pessoas instruções detalhadas, as regras do Guia do Scrum orientam seus relacionamentos e interações.

Métricas em Projetos Ágeis



Scrum

- Emprega uma abordagem **iterativa** e **incremental** para otimizar a previsibilidade e controlar o risco.
- Vários processos, técnicas e métodos podem ser empregados com o framework. Scrum se acopla as práticas existentes ou as torna desnecessárias.
- Scrum é baseado no **empirismo** e **lean thinking**. O empirismo por aprender com a experiência e por tomar decisões com base na observação. O *lean thinking* por reduzir o desperdício e se concentrar no essencial.

Métricas em Projetos Ágeis



Scrum

- O Scrum se baseia em 3 pilares:
 - ❑ **Transparência:** O processo emergente e o trabalho devem ser visíveis tanto para quem executa o trabalho quanto para quem recebe o trabalho.
 - ❑ **Inspeção:** Os artefatos do Scrum e o progresso em direção às metas acordadas devem ser inspecionados com frequência e diligência para detectar variações ou problemas potencialmente indesejáveis.
 - ❑ **Adaptação:** Se algum aspecto de um processo se desviar fora dos limites aceitáveis ou se o produto resultante for inaceitável, o processo que está sendo aplicado ou os materiais que estão sendo produzidos devem ser ajustados. O ajuste deve ser feito o mais rápido possível para minimizar novos desvios.

Métricas em Projetos Ágeis



Scrum

- O time Scrum (*Scrum Team*) é dividido em:
 - ❑ **Scrum Master:** É responsável por estabelecer o Scrum conforme definido no Guia do Scrum. Eles fazem isso ajudando todos a entender a teoria e a prática do Scrum, tanto no Scrum Team quanto na organização.
 - ❑ **Dono do produto (Product Owner):** É responsável por maximizar o valor do produto resultante do trabalho do Scrum Team. É o responsável pelo *Product Backlog*.
 - ❑ **Desenvolvedores (Developers):** São as pessoas do *Scrum Team* que estão comprometidas em criar qualquer aspecto de um Incremento utilizável a cada Sprint.

Métricas em Projetos Ágeis



Scrum

- Combina quatro eventos formais para inspeção e adaptação, contidos dentro de um evento, a Sprint. Os eventos são:
 - ❑ **Sprint:** São eventos de duração fixa de um mês ou menos para criar consistência. Uma nova Sprint começa imediatamente após a conclusão da Sprint anterior. Todo o trabalho necessário para atingir a meta do Produto acontece dentro de Sprints. Inclui *Sprint Planning*, *Daily Scrums*, *Sprint Review* e *Sprint Retrospective*.
 - ❑ **Planejamento da Sprint (*Sprint Planning*):** Inicia a Sprint ao definir o trabalho a ser realizado na Sprint. Este plano resultante é criado pelo trabalho colaborativo de todo o *Scrum Team*. O Scrum Team também pode convidar outras pessoas para participar da Sprint Planning para fornecer conselhos.

Métricas em Projetos Ágeis



Scrum

- ❑ **Reunião diária (Daily Scrum):** Tem o propósito de inspecionar o progresso em direção a Meta da Sprint e adaptar o Sprint Backlog conforme necessário, ajustando o próximo trabalho planejado. É um evento de 15 minutos para os *Developers* do *Scrum Team*. O *Product Owner* e o *Scrum Master* podem participar se estiverem atuando como *developers na Sprint*. Os *developers* podem se reunir ao longo do dia para discutir mais detalhadamente sobre a Sprint.
- ❑ **Revisão da Sprint (*Sprint Review*):** Tem o propósito de inspecionar o resultado da Sprint e determinar as adaptações futuras. O Scrum Team apresenta os resultados de seu trabalho para os principais stakeholders e o progresso em direção a Meta do Produto é discutido.

Métricas em Projetos Ágeis



Scrum

- ❑ **Retrospectiva da Sprint (Sprint Retrospective):** Tem o propósito de planejar maneiras de aumentar a qualidade e a eficácia. O *Scrum team* inspeciona como foi a última Sprint em relação a indivíduos, interações, processos, ferramentas e sua Definição de Pronto.

Métricas em Projetos Ágeis



Scrum

- O Scrum trabalha com alguns artefatos que representam trabalho e valor. Seguem os artefatos:
 - ❑ **Backlog do Produto (*Product Backlog*)**: É uma lista ordenada e emergente do que é necessário para melhorar o produto. É a única fonte de trabalho realizado pelo Scrum Team. Gera como compromisso a **Meta do Produto**. Normalmente dividido em histórias de usuário (*User Stories*).
 - ❑ **Backlog da Sprint (*Sprint Backlog*)**: É composto pela **Meta da Sprint** (por que), o conjunto de itens do *Product Backlog* selecionados para a Sprint (o que), bem como um plano de ação para entregar o Incremento (como).

Métricas em Projetos Ágeis



Scrum

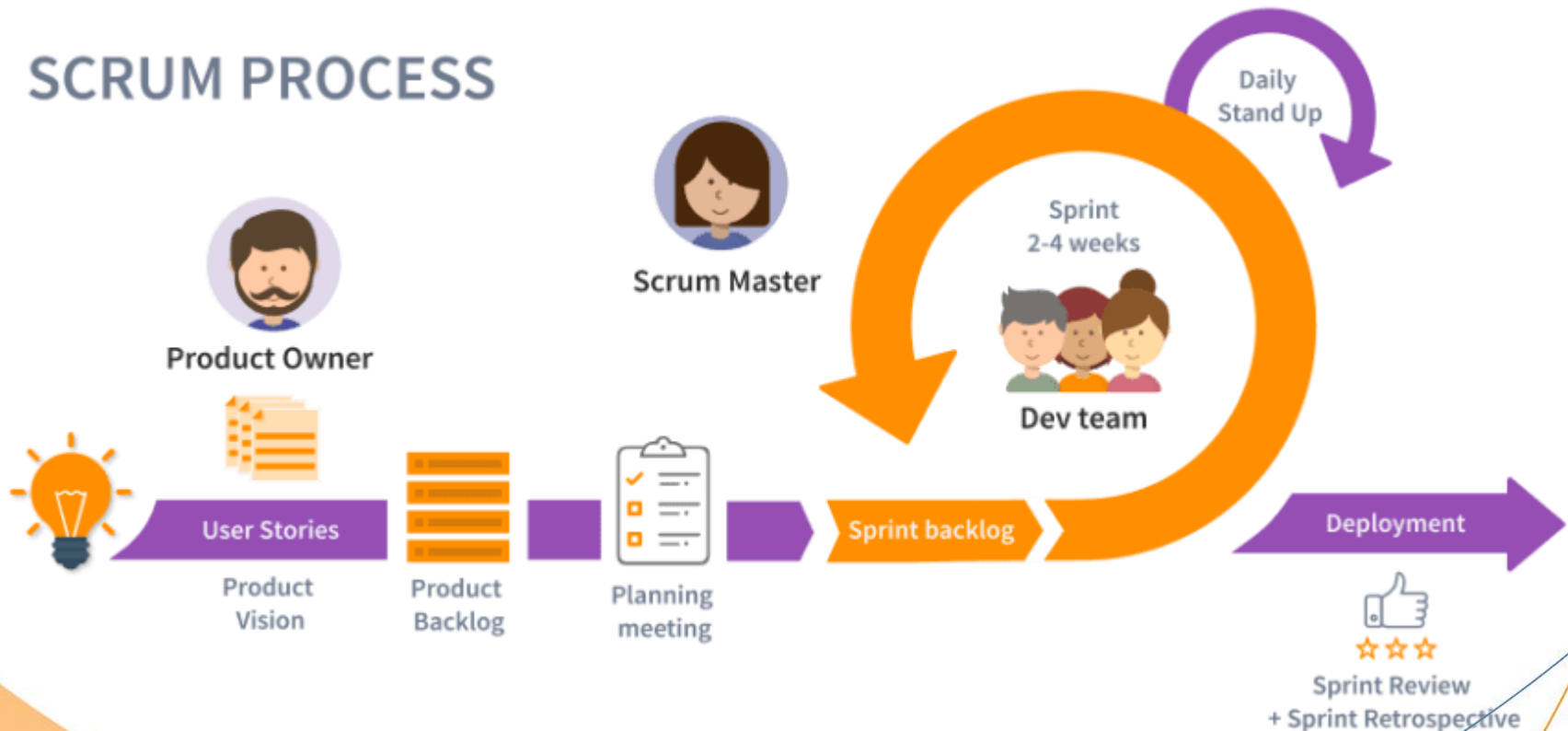
- ❑ **Incremento:** É um trampolim concreto em direção a Meta do produto. Cada incremento é adicionado a todos os incrementos anteriores e completamente verificado, garantindo que todos os incrementos funcionem juntos. A fim de fornecer valor, o incremento deve ser utilizável. O trabalho não pode ser considerado parte de um incremento a menos que atenda a Definição de Pronto. A **Definição de Pronto** é uma descrição formal do estado do Incremento quando ela atende às medidas de qualidade exigidas para o produto.

Métricas em Projetos Ágeis

Scrum

- Segue um resumo do Scrum:

SCRUM PROCESS



Métricas em Projetos Ágeis



Scrum

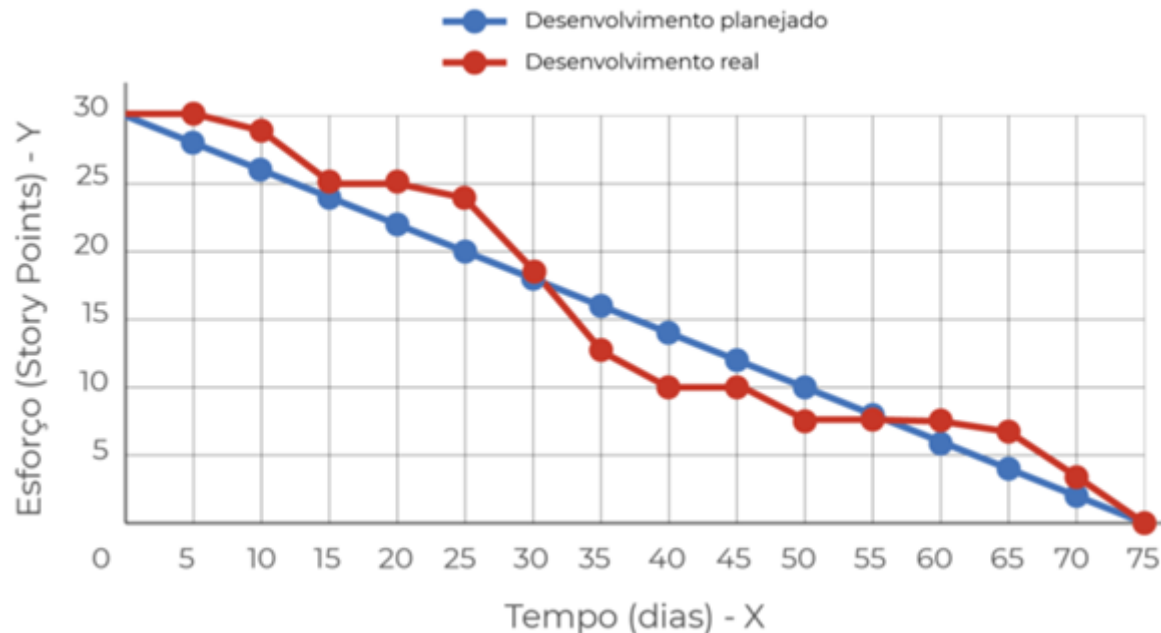
- Seguem algumas ferramentas para medição do progresso ou tempo das entregas:
 - ☐ Burndown
 - ☐ Burnup
 - ☐ Velocity Chart
 - ☐ Cumulative flows (CDF)
 - ☐ Lead time
 - ☐ Lead time breakdown
 - ☐ Throughput

Métricas em Projetos Ágeis



Scrum

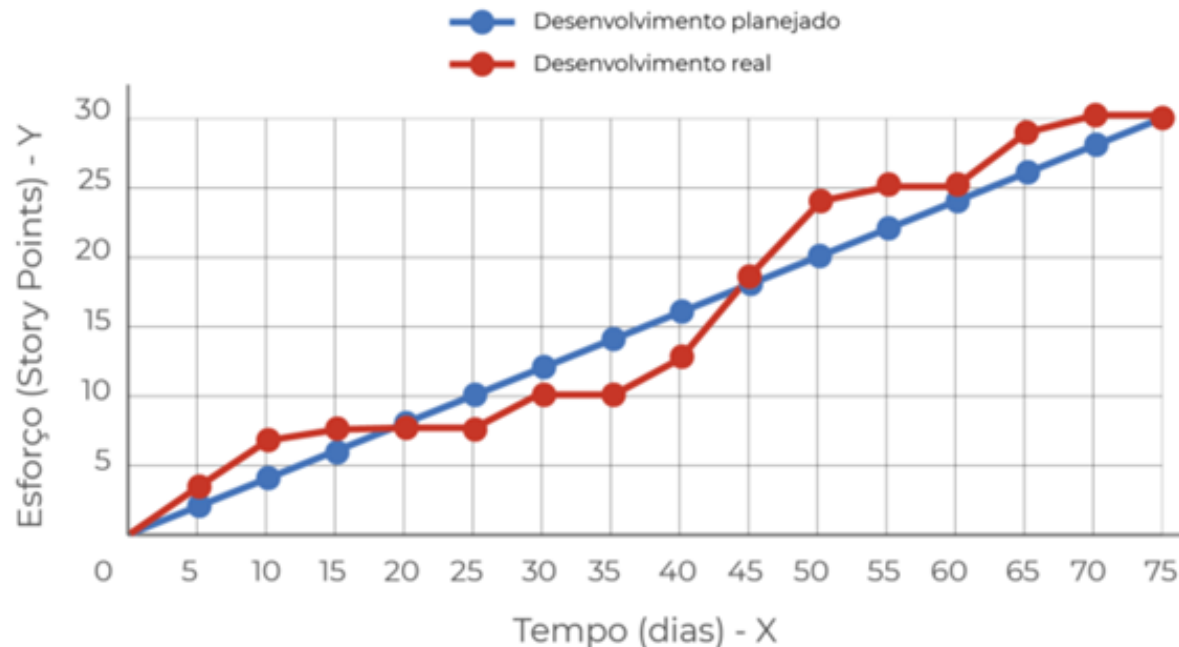
- **Burndown:** Apresenta dois eixos, onde o eixo horizontal X representa o tempo de uma sprint, e o eixo vertical Y representa o esforço em pontos de história (*Story Points*) necessário para completar a sprint. Parte da estimativa total em *Story Points* e apresenta a redução delas ao longo do tempo, conforme as histórias de usuário (*User Stories*) estão sendo concluídas.



Métricas em Projetos Ágeis

Scrum

- **Burnup:** Também apresenta dois eixos, onde o eixo horizontal X representa o tempo de uma sprint, e o eixo vertical Y representa o esforço em pontos de história (Story Points) necessário para completar a sprint. A diferença é que o gráfico apresenta o acumulado de pontos de história (Story Points) das histórias de usuário (*User Stories*) concluídas ao longo do tempo.

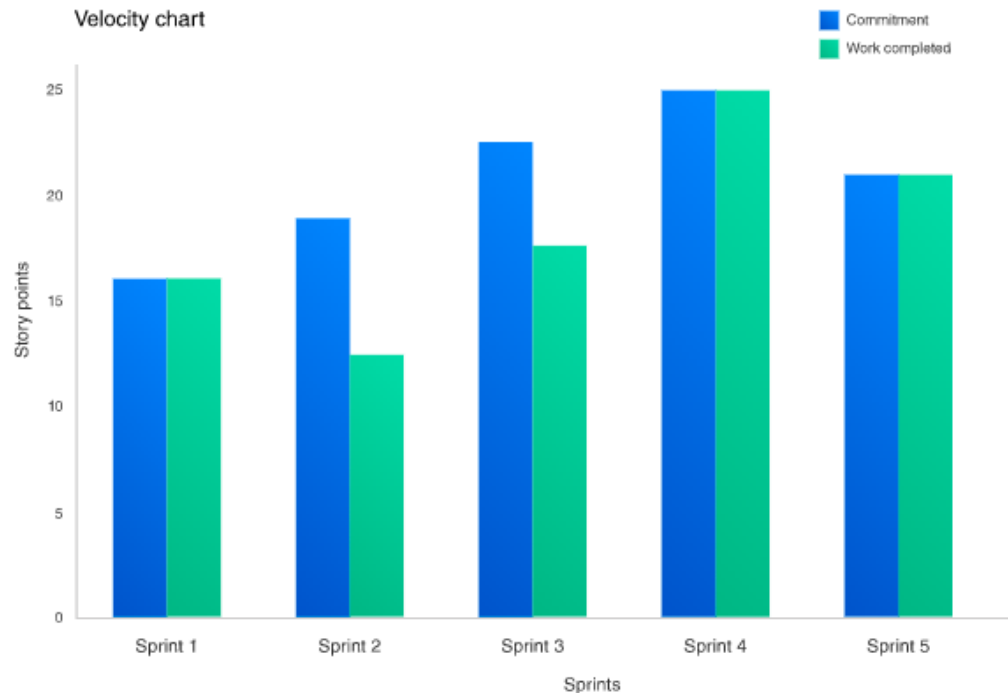


Métricas em Projetos Ágeis



Scrum

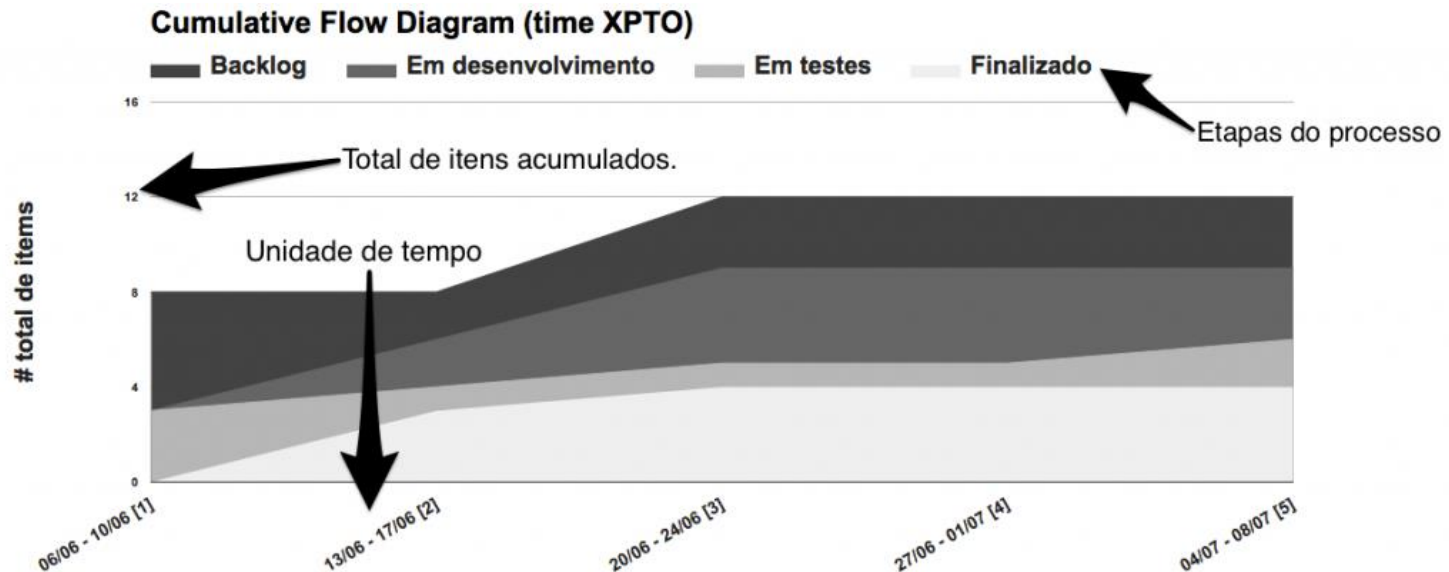
- **Velocity Chart:** Apresenta a quantidade de pontos de história (*Story Points*) concluídos em cada Sprint. A velocidade da equipe é calculada pela média das *Story Points* concluídas em um determinado período, e pode ser utilizada para prever a quantidade de *Story Points* que poderão ser concluídas em uma nova Sprint.



Métricas em Projetos Ágeis

Scrum

- **Cumulative flows (CDF):** Apresenta dois eixos, onde o eixo horizontal representa a unidade de tempo (ex: dias, semanas, sprints, etc.), e o eixo vertical representa o número de itens (ex: total de histórias de usuário) a serem completados. As áreas pintadas representam a evolução das etapas do processo ao longo do tempo.

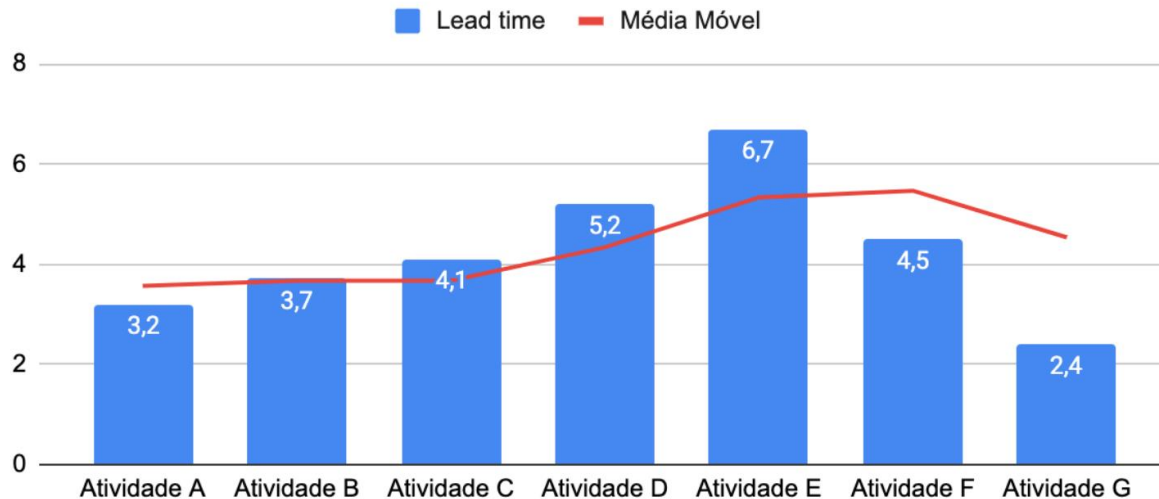


Métricas em Projetos Ágeis



Scrum

- Lead time:** O Lead Time é o Tempo que um item demora para percorrer todo o fluxo até chegar no estado de concluído. O gráfico apresenta no eixo horizontal as atividades concluídas (Ex: User Stories) e o eixo vertical apresenta a quantidade de tempo (ex: dias) que levou para conclui-las. O gráfico demonstra o ritmo de trabalho da equipe.

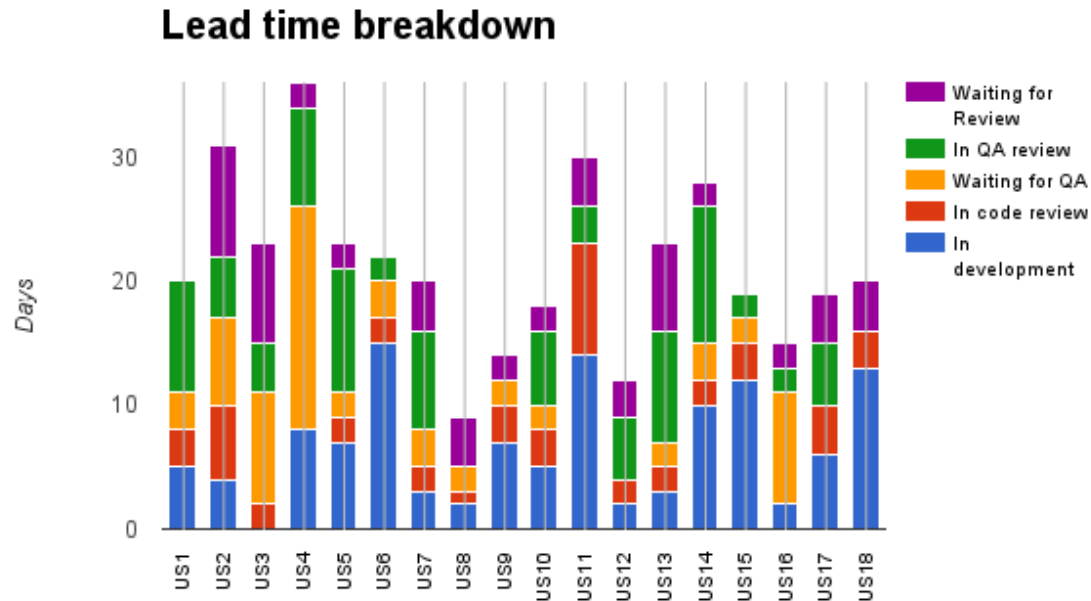


Métricas em Projetos Ágeis



Scrum

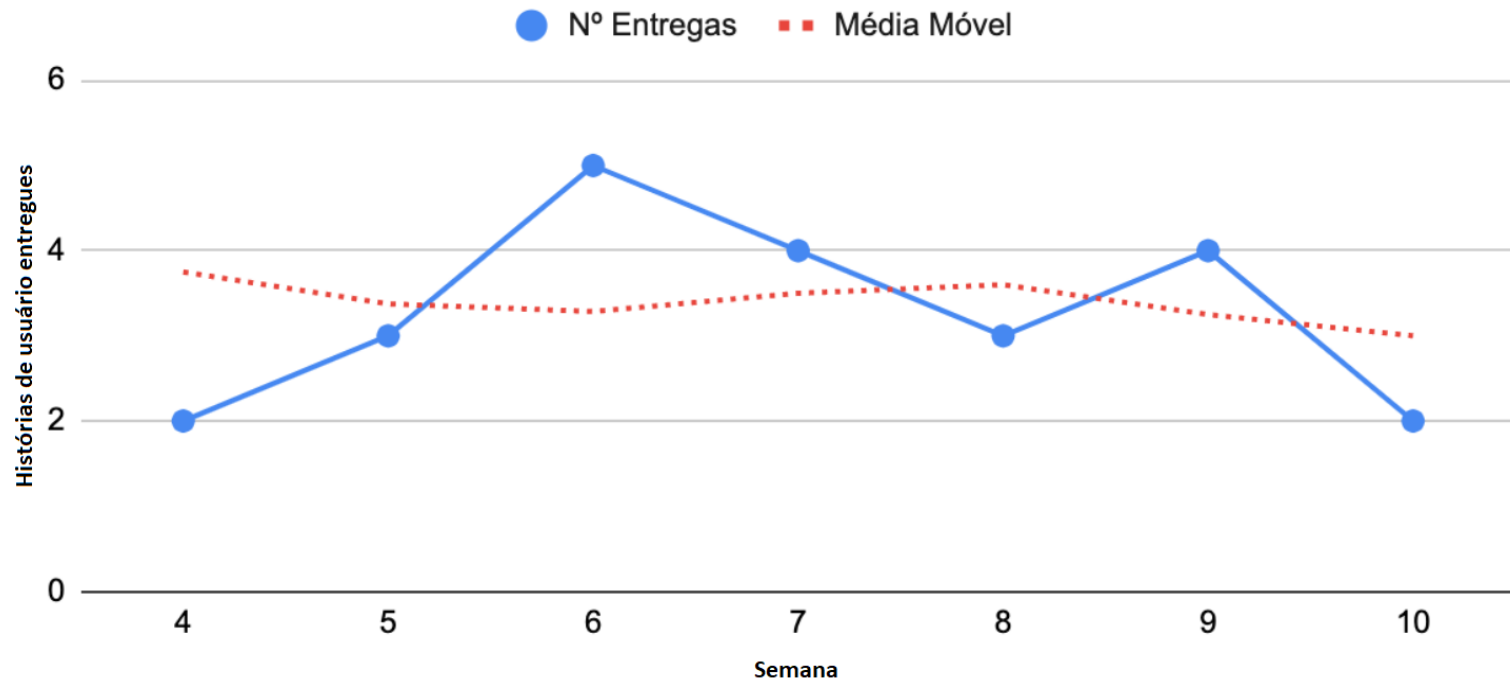
- Lead time breakdown:** No eixo horizontal apresenta as atividades concluídas (Ex: User Stories) e o eixo vertical apresenta a quantidade de tempo (ex: dias) que levou para concluí-las. O gráfico também detalha para cada atividade a proporção de tempo gasto em cada uma das etapas de construção. É útil para identificar quais são as principais razões para determinada atividade estar demorando para ser entregue.



Métricas em Projetos Ágeis

Scrum

- **Throughput:** Apresenta no eixo horizontal a unidade de tempo (Ex: Semanas) e o eixo vertical apresenta a quantidade de itens (Ex: User Stories) concluídos na unidade de tempo.



Métricas em Projetos Ágeis



Exercícios:

- 1) O que é o Scrum?
- 2) Quais os papéis de um time Scrum?
- 3) Quais os pilares do Scrum?
- 4) Cite os eventos do Scrum.
- 5) Cite 3 ferramentas utilizadas para medição do progresso ou tempo das entregas em um projeto ágil?
- 6) O que é a velocidade em um projeto ágil?
- 7) O que é *lead time* em um projeto ágil?
- 8) O que é o *Throughput* em um projeto ágil?

Análise de Pontos de Função



Análise de Pontos de Função



Objetivos

- Medir a funcionalidade que o usuário solicita e recebe.
- Medir o desenvolvimento e a manutenção de software de forma independente da tecnologia utilizada para sua implementação.

Análise de Pontos de Função



Padronização

- O processo de medição de pontos de função é padronizado pela IFPUG (*International Function Point Users Group*). A documentação pode ser adquirida através do link abaixo:
https://ifpug.memberclicks.net/index.php?option=com_mcform&view=ngforms&id=2131805#!/
- O IFPUG é representado no Brasil pelo BFPUG (*Brazilian Function Point Users Group*).
- Outro padrão de medição de pontos de função é o NESMA (Netherlands Software Metrics Association).

Análise de Pontos de Função



Processo de medição

- Deve ser simples o suficiente para minimizar o trabalho adicional envolvido no processo de medição. O esforço de medição deve ficar em um patamar que seja uma fração pequena do esforço total do projeto.
- Deve ser uma medida consistente entre vários projetos e organizações, ou seja, pessoas diferentes medindo o mesmo projeto devem encontrar resultados similares. Como a medição atua nos requisitos do software que podem não estar documentados adequadamente, podem existir interpretações distintas sobre um mesmo requisito. Desta forma, medições consistentes dependem de requisitos de qualidade e de pessoal bem treinado.

Análise de Pontos de Função



Usuário

- Para a APF (Análise de Pontos de Função), o conceito de usuário não está restrito apenas à pessoa física que usa o software. Usuário é qualquer pessoa ou coisa que interaja (envia ou recebe dados) com a aplicação. Exemplos de usuário: pessoa física, outra aplicação, um hardware um ator em um caso de uso, agentes governamentais, gestores do negócio que interajam com o software.
- Se a APF fosse baseada apenas no conceito de usuário como sendo uma pessoa, não seria possível medir sistemas que não têm interface com o usuário final. Assim, a APF pode ser usada para medir qualquer tipo de software.

Análise de Pontos de Função



Benefícios

- Uma ferramenta para determinar o tamanho de um pacote adquirido pela contagem de todas as funções incluídas.
- Provê auxílio aos usuários na determinação dos benefícios de um pacote para sua organização, através da contagem das funções que especificamente correspondem aos seus requisitos. As avaliar o custo do pacote, o tamanho das funções que serão efetivamente utilizadas, a produtividade e o custo da própria equipe, é possível realizar uma análise do tipo *make or buy*.

Análise de Pontos de Função



Benefícios

- Suporta a análise de produtividade e qualidade, seja diretamente ou em conjunto com outras métricas como esforço, defeitos e custo. Indicadores de produtividade (Horas/PF) e qualidade (Defeitos/PF) exercem um papel fundamental em muitas iniciativas de melhoria de processo de software.
- Apoiar os gerentes de projeto no gerenciamento de escopo de projetos. Ao realizar estimativas e medições dos pontos de função do projeto em cada fase do seu ciclo de vida, é possível determinar se os requisitos funcionais cresceram ou diminuíram, seja pela criação de novos requisitos ou pelo detalhamento de requisitos já existentes.

Análise de Pontos de Função



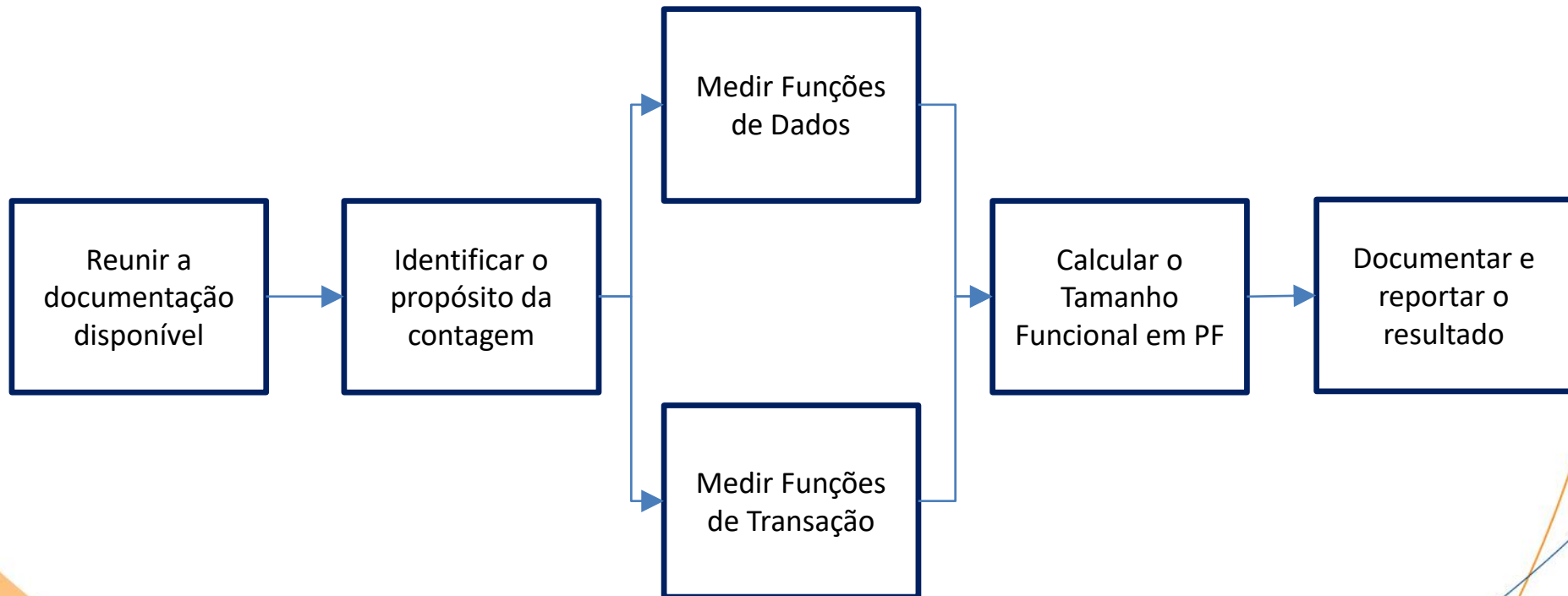
Benefícios

- Complementa o gerenciamento dos requisitos ao auxiliar na verificação da solidez e completeza dos requisitos especificados. A medição ajuda a expor falhas na documentação dos requisitos. Quando não se consegue medir algum aspecto do projeto, em geral isso está relacionado às deficiências nos requisitos.
- Um meio de estimar custo e recursos para o desenvolvimento e manutenção de software.
- Uma ferramenta para fundamentar a negociação de contratos. É possível utilizar pontos de função para gerar diversos acordos de níveis de serviço (SLA – *Service Level Agreement*).

Análise de Pontos de Função



Processo da contagem de pontos de função



Análise de Pontos de Função



Reunir a documentação disponível

- A documentação ideal deve:
 - ☐ Descrever a funcionalidade entregue pelo software; ou
 - ☐ Descrever a funcionalidade que é impactada pelo projeto de software medido.
- Exemplos de documentos que podem ser utilizados na medição:
 - ☐ Modelos de dados/objetos;
 - ☐ Diagramas de classe;
 - ☐ Diagramas de fluxo de dados;
 - ☐ Casos de uso;
 - ☐ Descrições procedurais;
 - ☐ Layout de relatório e telas;
 - ☐ Manuais de usuário.

Análise de Pontos de Função



Identificar o propósito da contagem

- Pode ser dividida em:
 - ☐ Determinar o tipo de contagem
 - ☐ Determinar o escopo da contagem
 - ☐ Determinar a fronteira da aplicação

Análise de Pontos de Função



Identificar o propósito da contagem

- Os três tipos de contagem são:
 - ☐ **Contagem de um projeto de desenvolvimento:** Mede a funcionalidade fornecida aos usuários finais do software quando da sua primeira instalação, além de possíveis migrações de dados.
 - ☐ **Contagem de um projeto de melhoria:** Mede as funções adicionadas, modificadas ou excluídas do sistema pelo projeto de melhoria, além de possíveis migrações de dados.
 - ☐ **Contagem de uma aplicação:** Mede a funcionalidade fornecida aos usuários por uma aplicação instalada.

Análise de Pontos de Função



Determinar o escopo da contagem

- O escopo define quais funções serão incluídas na contagem, se ela abrangerá um ou mais sistemas ou apenas parte de um sistema. Assim, o escopo pode abranger:
 - ☐ Todas as funcionalidades disponíveis;
 - ☐ Apenas as funcionalidades efetivamente utilizadas pelo usuário;
 - ☐ Apenas algumas funcionalidades específicas (relatório, transações, etc.).

Análise de Pontos de Função



Determinar a fronteira da aplicação

- A fronteira da aplicação é a interface conceitual que delimita o software que será medido e o mundo exterior (seus usuários).
- Se a definição da fronteira não estiver muito clara, há um risco muito grande de que todo trabalho de contagem posterior seja invalidado, pois várias funções serão indevidamente medidas ou deixadas de fora da medição.
- A fronteira deve ser determinada com base no ponto de vista do usuário.
- A fronteira deve ser baseada na separação das funções conforme estabelecido pelos processos de negócio, não em considerações tecnológicas.
- Em projetos de melhoria, a fronteira estabelecida no início do projeto deve estar de acordo com a fronteira já estabelecida para a aplicação sendo modificada.

Análise de Pontos de Função



Medir funções de dados

- As funções do tipo dado representam as funcionalidades fornecidas pelo sistema ao usuário para atender a suas necessidades de armazenamento de dados. São classificadas em:
 - ☐ **Arquivo Lógico Interno (ALI)**
 - ☐ **Arquivo de Interface Externa (AIE)**
- O termo arquivo não significa um arquivo do sistema operacional, tabela de banco de dados ou entidade no modelo de dados, refere-se a um grupo de dados logicamente relacionados e reconhecido pelo usuário. Eventualmente um arquivo pode estar mapeado em um ou mais desses itens. O foco deve estar em como o negócio manipula o dado (conceitual) e não como a aplicação o implementa (técnico). Nesse caso, cada entidade do modelo conceitual de dados seria um ALI ou AIE.

Análise de Pontos de Função



Medir funções de dados

- **Arquivo Lógico Interno (ALI):**
 - ☐ É um grupo logicamente relacionado de dados, reconhecido pelo usuário, mantido **dentro da fronteira da aplicação** sendo contada. Sua principal intenção é armazenar dados mantidos através de uma ou mais transações da aplicação sendo contada. Ex: dados de entrada e saída de um trabalhador.
 - ☐ Em geral encontramos os ALIs implementados fisicamente como tabelas de banco de dados. Porém, não há regra que estabeleça o mapeamento de tabelas do banco de dados para arquivos lógicos.
 - ☐ Agrupar entidades dependentes às suas respectivas entidades primárias.
 - ☐ Desconsiderar entidades associativas que contenham apenas atributos não requeridos pelo usuário ou que contenham apenas chaves estrangeiras.

Análise de Pontos de Função



Medir funções de dados

- **Não são exemplos de Arquivos Lógicos Interno (ALI):**
 - ☐ Arquivos de movimento recebidos de outra aplicação para manter um ALI (ex: arquivos de remessa e de retorno), no entanto os processos de carga e de geração desses arquivos podem ser funções do tipo transação. O arquivo de movimento é simplesmente um relatório gerado em formato de arquivo do sistema operacional.
 - ☐ Dados estáticos.
 - ☐ Dados temporários (cujo tempo de vida é o processamento de uma transação).
 - ☐ Arquivos introduzidos exclusivamente em função da tecnologia utilizada ou por decisão de projeto de software, também conhecidas como “dados de código”. Ex: Entidades com valores default ou com intervalos válidos para campos.
 - ☐ Arquivo de backup.

Análise de Pontos de Função



Medir funções de dados

- **Arquivo de Interface Externa (AIE):**
 - ❑ É um grupo logicamente relacionado de dados, reconhecido pelo usuário, mantidos **fora da fronteira da aplicação** sendo contada. Sua principal intenção é armazenar dados referenciados através de uma ou mais transações da aplicação sendo contada. Ex: dados do trabalhador contido no sistema de controle de ponto.

Análise de Pontos de Função



Medir funções de dados

- **Não são exemplos de Arquivos Interface Externa (AIE):**
 - ☐ Dados que estão armazenados na aplicação sendo contada e que são utilizados por uma aplicação externa. Neste caso, a sua aplicação possui um ALI e a outra aplicação reconhece estes dados vindos de um AIE.

Análise de Pontos de Função



Medir funções de Transação

- As funções do tipo transação representam os requisitos de processamento fornecidos pelo sistema ao usuário. São classificadas em:
 - ☐ **Entrada Externa (EE)**
 - ☐ **Saída Externa (SE)**
 - ☐ **Consulta Externa (CE)**

Análise de Pontos de Função



Medir funções de transações

- **Entrada Externa (EE):**
 - ☐ É uma transação que processa dados ou informações de controle originados **de fora da fronteira da aplicação**. Sua principal intenção é manter um ou mais arquivos lógicos internos e/ou alterar o comportamento do sistema. Ex: incluir cliente, alterar cliente, excluir cliente.
- **Não são exemplos de Entrada Externa (EE):**
 - ☐ Telas de login.
 - ☐ Telas de filtro.
 - ☐ Preenchimento de campos de dados.
 - ☐ Gerar relatórios

Análise de Pontos de Função



Medir funções de transações

- **Saída Externa (SE):**

- ☐ É uma transação que apresenta dado ao usuário ou envia dados ou informações de controle **para fora da fronteira da aplicação**. Sua principal intenção é apresentar informação ao usuário através de lógica de processamento que não seja apenas uma simples recuperação de dados ou informações de controle. Seu processamento deve conter cálculo ou criar dados derivados, manter um arquivo lógico interno ou alterar o comportamento do sistema. Ex: relatório de total de faturamento por cliente, tela de login (com criptografia), consultas complexas com processamento de dados a partir de cálculos, apresentação de gráficos com dados processados a partir de cálculos.

- **Não são exemplos de Saída Externa (SE):**

- ☐ Telas de filtro.
- ☐ Consultas simples, sem processamento de dados utilizando cálculos.

Análise de Pontos de Função



Medir funções de transações

- **Consulta Externa (CE):**
 - ☐ É uma transação que envia dados ou informações de controle **para fora da fronteira da aplicação**. Sua principal intenção é apresentar informações ao usuário pela simples recuperação de dados ou informações de controle de ALIs ou AIEs. Ex: consulta ao cadastro de cliente, apresentação de gráfico a partir de recuperação simples (sem cálculos).
- **Não são exemplos de Consulta Externa (EE):**
 - ☐ Relatórios financeiros, gerados a partir de cálculos.
 - ☐ Telas de filtro.

Análise de Pontos de Função



Calcular o tamanho funcional

- Cada função do tipo dado e transação possui um peso em PF, que é determinado pela complexidade da função que pode ser baixa, média ou alta.
- A complexidade das funções do tipo dados (ALI ou AIE) é determinada pela quantidade de dois parâmetros:
 - ❑ **Tipos de Dados (TD):** campo único, reconhecido pelo usuário e não repetido. Em uma visão limitada, seria cada atributo de uma tabela. Também é chamado de **Dado Elementar Referenciado (DER)**.
 - ❑ **Tipos de Registros (TR):** subgrupos de dados, reconhecido pelo usuário, componente de um ALI ou AIE. Pode ser opcional, quando não é obrigatório o preenchimento na transação que cria ou adiciona os dados, ou obrigatório, quando o usuário precisa sempre preencher os dados na transação. Em uma visão limitada, quando um agrupamento de tabelas são caracterizadas como um único arquivo lógico, ALI ou AIE, a tabela reconhecida pelo usuário e as demais se tornam tipos de registro. Também chamado de **Registro Lógico Referenciado (RLR)**.

Análise de Pontos de Função



Calcular o tamanho funcional

- Algumas observações para identificação de Tipos de Dados (TD):
 - ☐ Contar apenas dados reconhecidos pelo usuário;
 - ☐ Contar dados repetidos apenas uma vez;
 - ☐ Não contar atributos não referenciados pela aplicação medida;
 - ☐ Quando um dado for mantido ou referenciado por mais de uma aplicação, contar apenas os dados utilizados pela aplicação medida;
 - ☐ Atributos chaves em entidades associativas não devem ser contados como tipos de dados, apenas seus atributos não chave, pois estes atributos já serão contados nas suas entidades primárias;
 - ☐ Atributos compostos podem ser contados com base em cada um dos seus elementos ou como um único tipo de dados dependendo se são utilizados juntos ou separados pela aplicação medida. Por exemplo, o atributo composto Nome pode estar dividido nos atributos Primeiro nome e Sobrenome, sendo contado como dois tipos de dados se as partes forem utilizadas separadamente.

Análise de Pontos de Função



Calcular o tamanho funcional

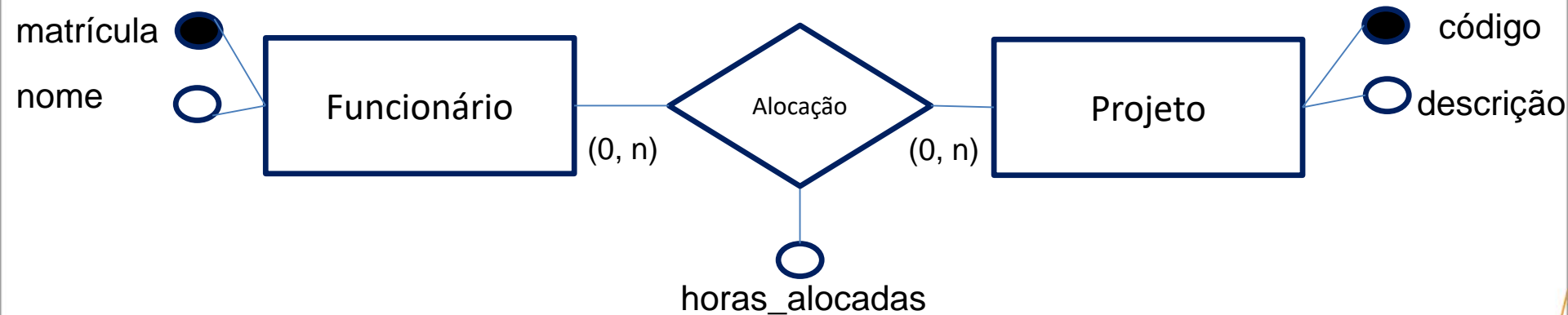
- Algumas observações para identificação de Tipos de Registros (TR):
 - ☐ Contar um tipo de registro para cada função de dados (ALI ou AIE);
 - ☐ Contar um tipo de registro para cada entidade associativa com atributos não chave;
 - ☐ Contar um tipo de registro adicional para cada subtipo de uma classe;
 - ☐ Se não houver um modelo de dados, procurar grupos de dados repetidos para identificar tipos de registros.

Análise de Pontos de Função



Calcular o tamanho funcional

- No exemplo abaixo, Funcionário, Alocação e Projeto seriam um ALI ou AIE, onde cada um deles é considerado um Tipo de Registro (TR) porque o usuário reconhece o atributo horas_alocadas.



- Então o mapeamento ficaria da seguinte forma:

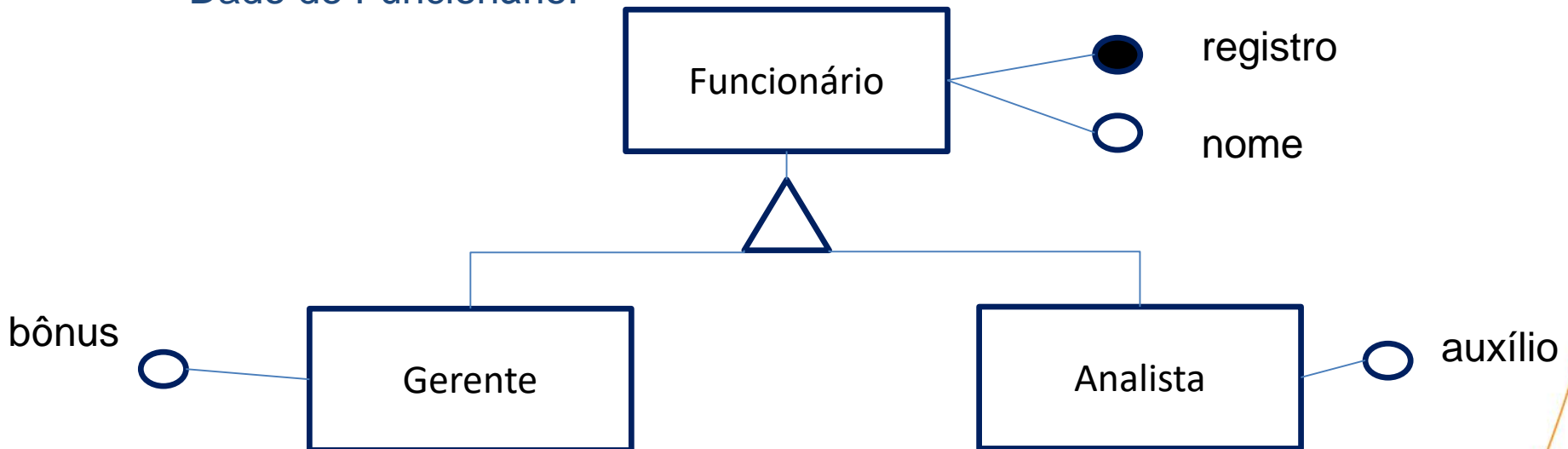
Descrição	Tipo	TD	TR
Funcionário	ALI ou AIE	2	1
Projeto	ALI ou AIE	2	1
Alocação	ALI ou AIE	1	1

Análise de Pontos de Função



Calcular o tamanho funcional

- No exemplo abaixo, Funcionário seria um ALI ou AIE, porque o usuário reconhece um gerente e uma analista como um funcionário, cada tabela seria um Tipo de Registro (TR) e cada campo seria um Tipo de Dado de Funcionário.



- Então o mapeamento ficaria da seguinte forma:

Descrição	Tipo	TD	TR
Funcionário	ALI ou AIE	4	3

Análise de Pontos de Função



Calcular o tamanho funcional

- Segue a tabela para definição da complexidade funcional de um ALI ou AIE:

	Tipos de dados (TD)			
Tipos de registros (TR)		<20	20-50	>50
	1	Baixa	Baixa	Média
	2-5	Baixa	Média	Alta
	>5	Média	Alta	Alta

Análise de Pontos de Função



Calcular o tamanho funcional

- Segue a contribuição em pontos de função das funções do tipo dado:

Tipo de função do tipo dado	Baixa	Média	Alta
Arquivo Lógico Interno	7 PF	10 PF	15 PF
Arquivo de Interface Externa	5 PF	7 PF	10 PF

Análise de Pontos de Função



Calcular o tamanho funcional

- Então, com base no último modelo apresentado, considerando que Funcionário seja um ALI, a contribuição em pontos de função dele seria:

Descrição	Tipo	TD	TR	Complexidade	Contribuição (PF)
Funcionário	ALI	4	3	Baixa	7
Total da contribuição em PF					7

Análise de Pontos de Função



Calcular o tamanho funcional

- A complexidade das funções do tipo transação (EE, SE e CE) é determinada pela quantidade de dois parâmetros:
 - ❑ **Tipos de Dados (TD):** campo único, reconhecido pelo usuário e não repetido. Em uma visão limitada, seria cada campo da tela. Por exemplo, em um formulário os campos nome, cpf, endereço e o botão de confirmação seriam tipos de dados,
 - ❑ **Arquivos Referenciados (AR):** um arquivo lógico interno (ALI) lido ou mantido por uma função do tipo transação.

Análise de Pontos de Função



Calcular o tamanho funcional

- Algumas regras para contagem de **Tipo de Dado (TD)**:
 - ☐ Conte um único tipo de dado para cada atributo que atravessa a fronteira da aplicação (entrando e/ou saindo), reconhecido pelo usuário, único, não repetido. Por exemplo, um gráfico do tipo pizza por conter uma legenda e um número equivalente à representação gráfica. Neste caso, considere dois tipos de dados, um para a legenda e outro para o valor numérico.
 - ☐ Conte um único tipo de dado para as mensagens de erro que são apresentadas pelo sistema, ou seja, um tipo de dado para a capacidade do sistema de apresentar mensagens de erro.
 - ☐ Conte um único tipo de dado para a capacidade de especificar uma ação a ser tomada. Por exemplo, se a tela tiver botões de confirmação ou cancelamento da operação, considere apenas um tipo de dado.

Análise de Pontos de Função



Calcular o tamanho funcional

- ☐ Se a tela apresentar uma tabela contendo vários registros, conte um tipo de dado para cada coluna da tabela.
- ☐ Não conte como tipo de dados literais apresentados na tela (ex: títulos de campos), ou seja, considere apenas os campos que o usuário informa e aqueles campos apresentados ao usuário.

Análise de Pontos de Função



Calcular o tamanho funcional

- Segue a tabela para definição da complexidade funcional para Entradas Externas (EEs):

	Tipos de dados (TD)			
Arquivos referenciados (ARs)		<5	5-15	>15
	<2	Baixa	Baixa	Média
	2	Baixa	Média	Alta
	>2	Média	Alta	Alta

Análise de Pontos de Função



Calcular o tamanho funcional

- Segue a tabela para definição da complexidade funcional para Saídas Externas (SEs) e Consultas Externas (CEs):

	Tipos de dados (TD)			
Arquivos referenciados (ARs)		<6	6-19	>19
	<2	Baixa	Baixa	Média
	2-3	Baixa	Média	Alta
	>3	Média	Alta	Alta

Análise de Pontos de Função



Calcular o tamanho funcional

- Segue a contribuição em pontos de função das funções do tipo transação:

Tipo de função do tipo transação	Baixa	Média	Alta
Entrada Externa (EE)	3 PF	4 PF	6 PF
Saída Externa (SE)	4 PF	5 PF	7 PF
Consulta Externa (CE)	3 PF	4 PF	6 PF

Análise de Pontos de Função



Calcular o tamanho funcional

- Então, com base no último modelo apresentado, considerando que Funcionário possua três processos elementares (transações) e que as telas possuam os mesmos campos, a contribuição em pontos de função das transações seria:

Transação	Tipo	TD *	AR	Complexidade	Contribuição (PF)
Incluir funcionário	EE	7	1	Baixa	3
Alterar funcionário	EE	7	1	Baixa	3
Excluir funcionário	EE	7	1	Baixa	3
Total da contribuição em PF					9

* Considerando 1 TD para um campo de seleção do tipo do funcionário, 1 TD para botões de confirmação / cancelamento da operação e 1 TD para as mensagens a serem apresentadas.

Análise de Pontos de Função



Calcular o tamanho funcional

- Então, o total de pontos de função da aplicação é determinado pela contribuição dos tipos dados somados a contribuição dos tipos transação.

Descrição	Contribuição (PF)
Funções do tipo dado	7
Funções do tipo transação	9
Total de pontos de função	16

Análise de Pontos de Função



Calcular o tamanho funcional

- Por último, deve-se calcular o tamanho funcional considerando o tipo de contagem. Vamos identificar os componentes para o cálculo do número de pontos de função para cada tipo de contagem:

☐ **Pontos de Função de Projeto de Desenvolvimento (DFP) é igual a:**

$$\text{DFP} = (\text{ADD} + \text{CFP})$$

Onde:

- ✓ ADD: tamanho das funções entregues ao usuário.
- ✓ CFP: tamanho das funções de conversão.

Análise de Pontos de Função



Calcular o tamanho funcional

- ❑ Pontos de Função de Projeto de Melhoria (EFP) é igual a:

$$\text{EFP} = (\text{ADD} + \text{CHGA} + \text{CFP} + \text{DEL})$$

Onde:

- ✓ ADD: tamanho das funções incluídas.
- ✓ CHGA: tamanho das funções alteradas.
- ✓ CFP: tamanho das funções de conversão.
- ✓ DEL: tamanho das funções excluídas.

Análise de Pontos de Função



Calcular o tamanho funcional

- ❑ Pontos de Função de Aplicação (AFP) é igual a:

$$AFP = ADD$$

Onde:

- ✓ ADD: tamanho das funções entregues.

Análise de Pontos de Função



Calcular o tamanho funcional

- O total de pontos de função calculados até o momento também podem ser chamados de **Pontos de Função Brutos ou Pontos de Função Não Ajustados**. A estes pontos de função ainda poderiam ser aplicados um fator de ajuste levando em consideração os requisitos não funcionais da aplicação, o que resultaria em **Pontos de Função Ajustados**, mas como esta parte atualmente é considerada opcional pela IFPUG, não faz parte do escopo deste material.

Análise de Pontos de Função



Documentar e reportar o resultado

- O resultado final da contagem deve ser documentado, incluindo:
 - ☐ A documentação utilizada na contagem
 - ☐ O propósito da contagem
 - ☐ O tipo da contagem
 - ☐ O escopo da contagem
 - ☐ A fronteira da aplicação
 - ☐ A data e os participantes da contagem
 - ☐ Identificação das funções de dados e transações com suas respectivas complexidades
 - ☐ O total de pontos de função identificados para cada função de dados e transações

Análise de Pontos de Função



Outros tipos de contagens

- O tipo de contagem apresentado até o momento é o que chamamos de **contagem detalhada**, onde é feita a contagem de todos os tipos de dados e transações. Entretanto, a **NESMA** ainda define outros dois tipos de contagem a serem utilizadas no início do ciclo de vida da aplicação:
 - ❑ **Contagem por estimativa:** neste tipo de contagem deve-se:
 - ✓ Determinar todas as funções de todos os tipos (ALI, AIE, EE, SE, CE).
 - ✓ Para cada função do tipo dado (ALI, AIE) ter a sua complexidade funcional avaliada como Baixa, e cada função transação (EE, SE, CE) deve ser avaliada como complexidade Média.

Análise de Pontos de Função



Outros tipos de contagens

- ❑ **Contagem indicativa:** neste tipo de contagem deve-se:
 - ✓ Determinar a quantidade das funções do tipo dado (ALI e AIE);
 - ✓ Calcular o total de pontos de função da aplicação da seguinte forma:

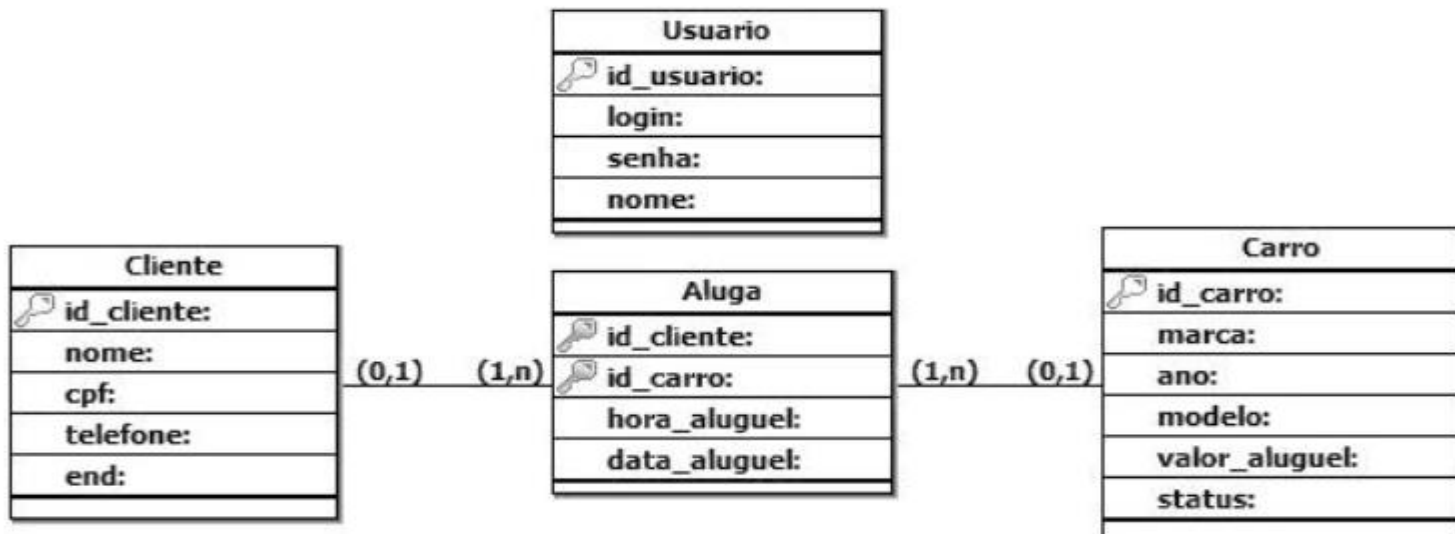
$$\text{Contagem Indicativa} = 35 \times \text{Qtde de ALI} + 15 \times \text{Qtde de AIE}$$

Análise de Pontos de Função



Exercícios:

- 1) Considerando o modelo lógico abaixo, e considerando 3 transações para cada entidade (inclusão, alteração e exclusão), calcule com base na contagem detalhada o total de pontos de função da aplicação. Considere que cada transação terá botões para confirmação / cancelamento da operação e que serão exibidas mensagens de acordo com a transação executada.



Análise de Pontos de Função



Exercícios:

- 2) Calcule o total de pontos de função da aplicação considerando as contagens por estimativa e indicativa.

Melhoria de Processo de Software



Melhoria de Processo de Software



Modelo de Maturidade da Capabilidade de Software

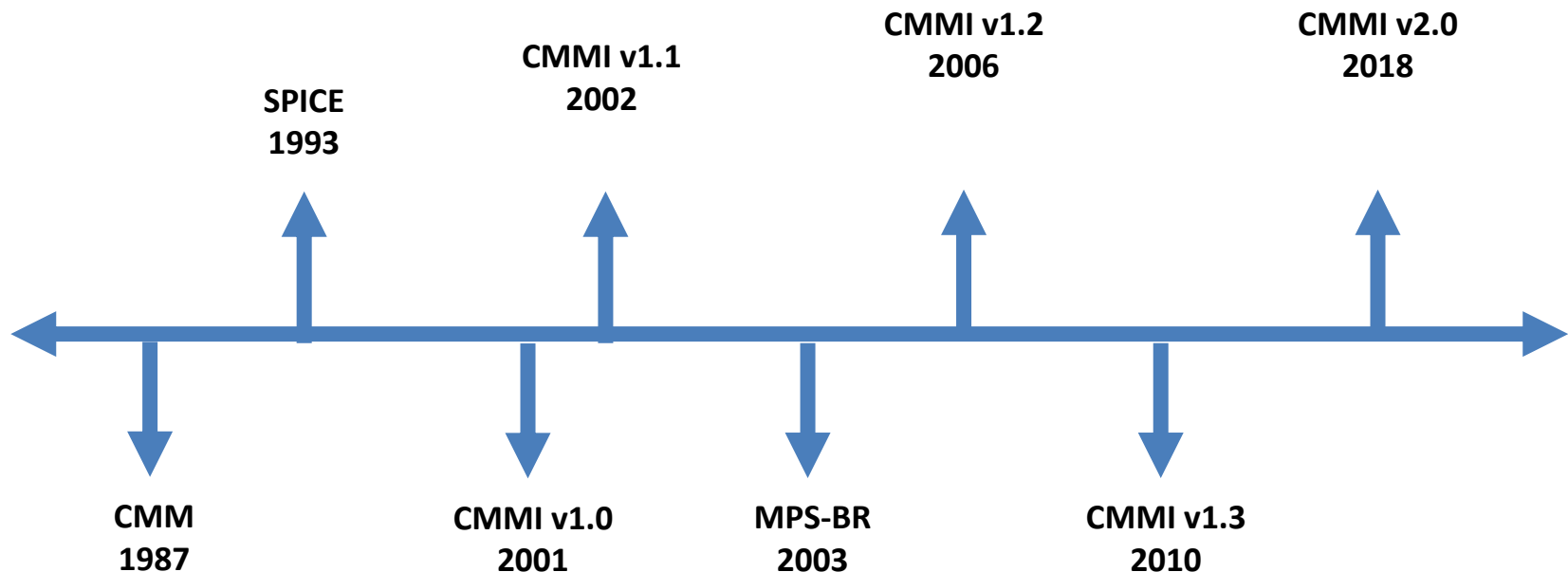
- Fornece às organizações de software um guia de como obter controle em seus processos para desenvolver e manter software e como evoluir em direção a uma cultura de engenharia de software e excelência de gestão.

Melhoria de Processo de Software



Histórico

- **Evolução dos modelos de melhoria de software:**



Melhoria de Processo de Software



CMM (Capability Maturity Model)

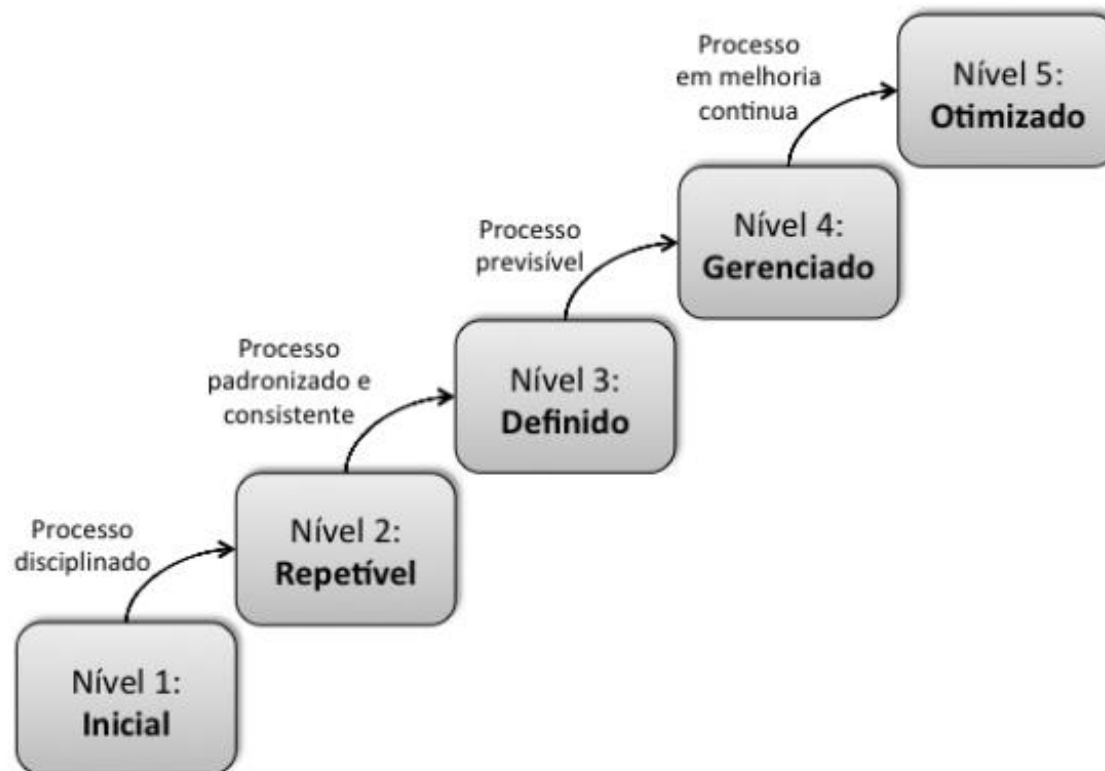
- Criado pelo SEI (Software Engineering Institute) em 1997.
- Foi projetado para guiar as organizações de software no processo de seleção das estratégias de melhoria, determinando a maturidade atual do processo e identificando as questões mais críticas para a qualidade e melhoria do processo de software. Focando em um conjunto limitado de atividades e trabalhando agressivamente para concluí-las com êxito, a organização pode melhorar o processo em toda a sua estrutura, possibilitando ganhos contínuos e duradouros na capacidade do processo de software.

Melhoria de Processo de Software



CMM (Capability Maturity Model)

Níveis de maturidade



Melhoria de Processo de Software



CMM (Capability Maturity Model)

- Possui 5 níveis de maturidade:
 - ❑ **Inicial:** O processo de software é caracterizado como “ad hoc” e até mesmo ocasionalmente caótico. Poucos processos são definidos e o sucesso depende de esforço individual.
 - ❑ **Repetível:** Os processos básicos de gestão de projeto são estabelecidos para acompanhar custo, cronograma e funcionalidade. A necessária disciplina do processo existe para repetir sucessos anteriores em projetos com aplicações similares.
 - ❑ **Definido:** O processo de software para as atividades de gestão e engenharia é documentado, padronizado e integrado em um processo de software padrão para a organização. Todos os projetos utilizam uma versão aprovada do processo de software padrão para desenvolver e manter software.

Melhoria de Processo de Software



CMM (Capability Maturity Model)

- Possui 5 níveis de maturidade:
 - ❑ **Gerenciado:** Medidas detalhadas do processo de software e da qualidade do produto são realizadas. O processo e os produtos de software são quantitativamente compreendidos e controlados.
 - ❑ **Em Otimização:** Medidas detalhadas do processo de software e da qualidade do produto são realizadas. O processo e os produtos de software são quantitativamente compreendidos e controlados.

Melhoria de Processo de Software



CMM (Capability Maturity Model)

- Não é prescritivo, ou seja, ele não diz à organização como melhorar. O CMM descreve a organização em cada nível de maturidade sem prescrever os meios específicos para alcançá-lo. A mudança do nível 1 para o nível 2 pode levar anos. A movimentação entre os níveis geralmente leva cerca de dois anos.
- Cada nível de maturidade constrói a base para os níveis seguintes alavancarem a implementação de processos de forma efetiva e eficiente.
- A melhoria de processo de software ocorre dentro do contexto dos planos estratégicos e dos objetivos de negócio da organização, da sua estrutura organizacional, das tecnologias em uso, da sua cultura social e sistema de gestão.

Melhoria de Processo de Software



CMM (Capability Maturity Model)

- **Áreas-chave de Processo:** Com exceção do Nível 1, cada nível de maturidade é decomposto em várias áreas-chave de processo, que indicam as áreas nas quais uma organização deveria focar seus esforços para a melhoria de seu processo de software. As áreas-chave de processo identificam os assuntos que devem ser tratados para se obter um determinado nível de maturidade. Cada área-chave de processo identifica um grupo de atividades relacionadas, as quais, quando executadas coletivamente, atingem um conjunto de metas consideradas importantes para aumentar a capacidade do processo.

Melhoria de Processo de Software



CMM (Capability Maturity Model)

Nível	Áreas-chave de processo
Nível 2 (Repetível)	Gerência de requisitos Planejamento do projeto de software Acompanhamento e supervisão do projeto de software Gerência de subcontratos do software Garantia da qualidade do software Gerência de configuração do software
Nível 3 (Definido)	Foco nos processos da organização Definição do processo da organização Programa de treinamento Gerência do software integrado Engenharia de produto de software Coordenação inter-grupos Revisões
Nível 4 (Gerenciado)	Gerência quantitativa do processo Gerência de qualidade do software
Nível 5 (Em Otimização)	Prevenção de defeitos Gerência de mudança de tecnologia Gerência de mudança de processo

Melhoria de Processo de Software



Exercícios:

- 1) O que é o CMM?
- 2) Quais são seus níveis de maturidade?
- 3) O que são áreas-chave de processo?

Melhoria de Processo de Software



CMMI

O CMMI (*Capability Maturity Model Integration*) foi criado pelo SEI (*Software Engineering Institute*) em 2002 como um modelo evolutivo em relação aos vários CMMs, com o objetivo de combinar as suas várias disciplinas em uma estrutura única, flexível e componentizada que pudesse ser utilizada de forma integrada por organizações que demandavam processos de melhoria em âmbito corporativo.

CMMI está atualmente na versão 2.0 lançada em 2018.

Para ter uma visão geral de CMMI, clique no link abaixo:

<https://www.youtube.com/watch?v=IFrsy6sPVic>

Melhoria de Processo de Software



CMMI – Objetivos do modelo

O principal propósito do CMMI (Capability Maturity Model Integration) é fornecer diretrizes baseadas em melhores práticas para a melhoria dos processos e habilidades organizacionais, cobrindo o ciclo de vida de produtos e serviços completos, nas fases de concepção, desenvolvimento, aquisição, entrega e manutenção. Nesse sentido, suas abordagens envolvem a avaliação da maturidade da organização ou a capacitação das suas áreas de processo, o estabelecimento de prioridades e a implementação de ações de melhoria.

Melhoria de Processo de Software



CMMI – Modelos

É dividido nos seguintes modelos:

- **CMMI para Desenvolvimento (CMMI-DEV):** provê diretrizes para monitorar, mensurar e gerenciar processos de desenvolvimento.
- **CMMI para Serviços (CMMI-SVC):** provê diretrizes para entrega de serviços dentro das organizações e para clientes externos.
- **CMMI para Aquisições (CMMI-ACQ):** provê diretrizes para suporte às decisões relacionadas à aquisição de produtos e serviços.
- Maiores informações podem ser obtidas no link abaixo:
<https://cmmiinstitute.com/products/cmmi/cmmi-v2-products>

Melhoria de Processo de Software



CMMI – Abordagens de implementação

Oferece duas abordagens distintas para a sua implementação:

- Abordagem por Estágios;
- Abordagem Contínua.

CMMI – Abordagens por Estágios

- Um nível de maturidade pode ser considerado um degrau evolucionário para a melhoria do processo organizacional como um todo e consiste em práticas específicas e genéricas que integram um conjunto predefinido de áreas de processo. O cumprimento das metas específicas e genéricas correspondentes a essas áreas de processo é um pré-requisito para o atingimento do nível de maturidade correspondente. Esta abordagem pode ser considerada uma evolução direta do CMM.
- Esta abordagem é baseada em cinco níveis de maturidade:
 - 1) Inicial
 - 2) Gerenciado
 - 3) Definido
 - 4) Gerenciado Quantitativamente
 - 5) Otimizado

Melhoria de Processo de Software



CMMI – Abordagens por Estágios

1) Inicial

- Não possui processos.

2) Gerenciado

- O foco é direcionado para práticas de gestão de projetos, indicando que, em uma organização ainda imatura, é mais prioritário aprender a planejar, controlar e gerenciar os projetos do que investir em técnicas e metodologias de desenvolvimento de produtos.
- Gerenciar projetos envolve gerenciar, durante o seu andamento, os requisitos estabelecidos junto aos grupos interessados, a qualidade e a integridade dos produtos gerados, a aderência aos processos existentes e os acordos formalizados com os fornecedores envolvidos.
- Há uma preocupação explícita em relação à criação de uma infraestrutura para medição e análise dos processos, para viabilizar o seu controle e gerenciamento efetivo.

Melhoria de Processo de Software



CMMI – Abordagens por Estágios

3) Definido

- O foco está no processo de engenharia de produtos, que espelha as fases de um ciclo de vida padrão: Concepção (“Desenvolvimento de Requisitos”), Análise e Desenho (“Solução Técnica”), Testes e Implantação (“Integração do Produto”, “Verificação” e “Validação”).
- O modelo fomenta a criação de um ambiente organizacional orientado à integração entre equipes de trabalho e ao compartilhamento de conhecimentos e habilidades.
- O estímulo a práticas de gestão de riscos e à tomada de decisão baseada em análises formais fortalece as responsabilidades e os compromissos assumidos.

Melhoria de Processo de Software



CMMI – Abordagens por Estágios

4) Gerenciado Quantitativamente

- A gestão quantitativa baseada em medições e indicadores cobre, de forma integrada, todo o conjunto de processos organizacionais, assim como os projetos e respectivos produtos, como instrumento de suporte para o atendimento dos objetivos de desempenho de processo e de qualidade.
- Os projetos e seus produtos, assim como o processo organizacional, são controlados estatisticamente.

5) Otimizado

- O conceito de inovação organizacional integra os processos de gestão de mudanças tanto em processos como na tecnologia.
- A importância da análise e da resolução das causas dos desvios é explicitamente enfatizada.

Melhoria de Processo de Software



CMMI – Abordagem Contínua

- Através desta abordagem, o CMMI permite que cada uma de suas áreas de processo seja implementada de forma independente e evolutiva.
- Esta abordagem é baseada em quatro níveis de capacidade:
 - 0) Incompleto
 - 1) Executado
 - 2) Gerenciado
 - 3) Definido

Melhoria de Processo de Software



CMMI – Abordagem Contínua

0) Incompleto

O processo não é executado ou é parcialmente executado, ou seja, uma (ou mais) das metas específicas de sua área de processo não é satisfeita.

1) Executado

O processo satisfaz todas as metas específicas de sua área de processo e realiza o trabalho necessário para gerar os seus produtos.

Melhoria de Processo de Software



CMMI – Abordagem Contínua

2) Gerenciado

O processo é planejado e executado de acordo com políticas organizacionais, utiliza pessoal habilitado e recursos adequados para gerar saídas de forma controlada e envolve os grupos interessados adequados, além de ser monitorado, controlado, revisado, avaliado quanto à conformidade com sua descrição e ao desempenho previsto nos seus planos.

3) Definido

O processo é gerenciado e adaptado a partir de um conjunto de processos padronizados da organização, que, por sua vez, também evoluem continuamente.

Melhoria de Processo de Software



CMMI – Aplicação do modelo

- O CMMI pode ser implementado em quaisquer organizações cujo foco seja o desenvolvimento de produtos (sistemas em geral, software, hardware etc.) para o atendimento de necessidades de clientes externos ou internos, utilizando ou não recursos e/ou serviços terceirizados.
- A abordagem por estágios é mais recomendada para organizações que já estão familiarizadas com a incorporação de melhorias nos seus processos organizacionais através de grandes saltos de qualidade, tais como aquelas que já possuem um nível de maturidade do CMM/CMMI ou que possuem modelos de qualidade baseados na melhoria simultânea e integrada de vários processos.
- A abordagem contínua é mais recomendada para organizações que preferem uma evolução gradual na sua capacidade, processo a processo, possibilitando uma maior diluição do investimento a ser feito no programa de melhoria ao longo do tempo (organizações de menor porte também podem ter mais facilidade para utilizar o modelo nesta abordagem).

Melhoria de Processo de Software



CMMI – Aplicação do modelo

- 1) O que é o CMMI?
- 2) Quais são os objetivos do modelo?
- 3) É dividido em quais modelos?
- 4) Quais são os tipos de abordagens utilizadas?
- 5) Qual a diferença principal entre os tipos de abordagens?

Melhoria de Processo de Software



MPS-BR (Melhoria do Processo de Software Brasileiro)

- É um programa da Softex (Associação para Promoção da Excelência do Software Brasileiro) com apoio do Ministério da Ciência, Tecnologia, Inovações e Comunicações (MCTIC). Com início em dezembro de 2003, o programa tem como objetivo melhorar a capacidade de desenvolvimento de software, serviços e as práticas de gestão de RH na indústria de TIC.
- A Softex é uma Organização Social Civil de Interesse Público (OSCIP) que desenvolve ações para promover a melhoria da competitividade da Indústria Brasileira de Software e Serviços de TI (IBSS), bem como a disponibilidade de recursos humanos qualificados, tanto em tecnologias como em negócios.

Melhoria de Processo de Software



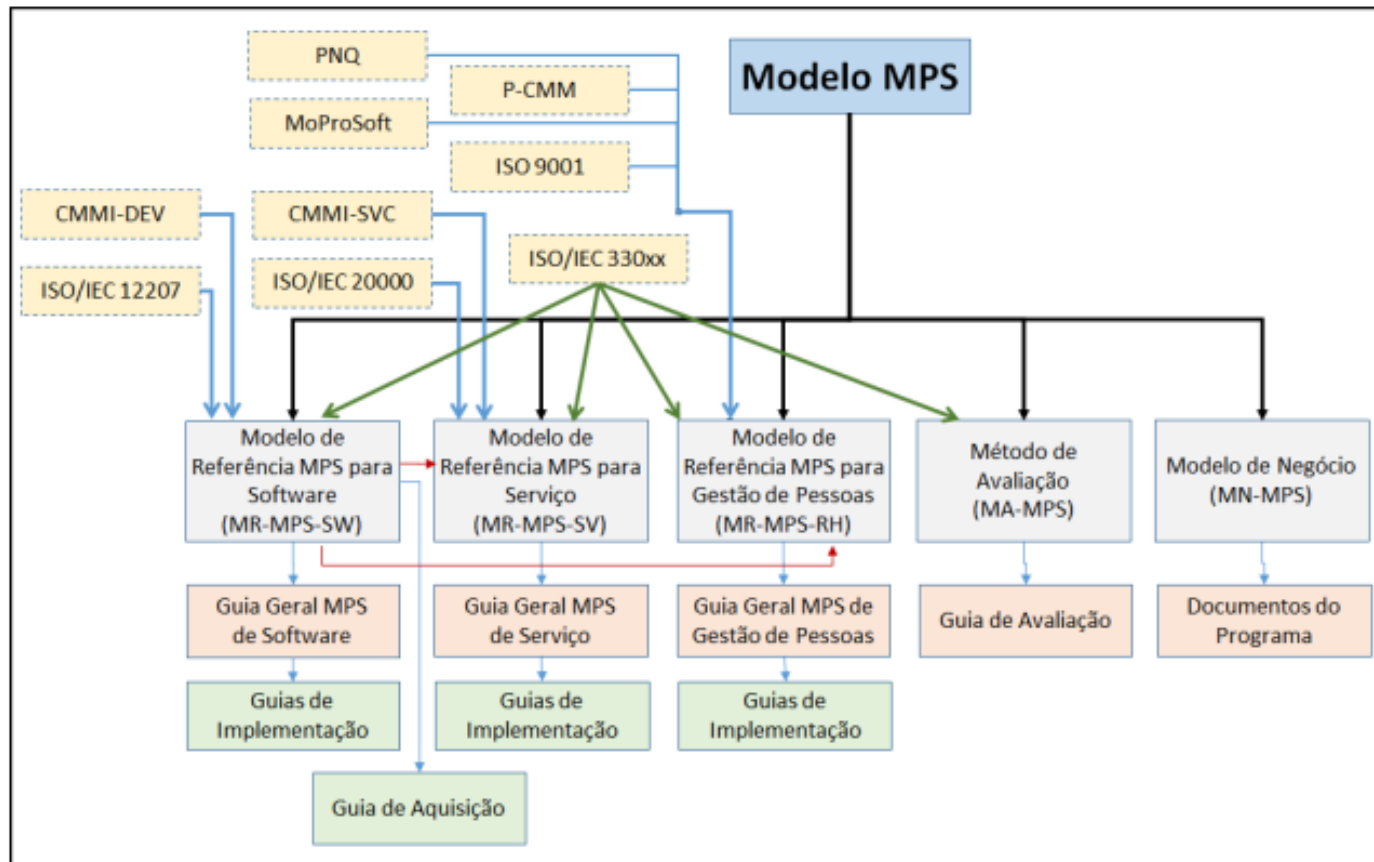
MPS-BR (Melhoria do Processo de Software Brasileiro)

- **Possui os seguintes modelos de referência:**
 - ❑ **MPS para software (MPS-SW):** Tem como base os requisitos de processos definidos nos modelos de melhoria de processo e atende a necessidade de implantar os princípios de engenharia de software de forma adequada ao contexto das empresas, estando em conformidade com as principais abordagens internacionais para definição, avaliação e melhoria de processos de software.
 - ❑ **MPS para Serviços (MPS-SV):** Foi desenvolvido para complementar ao modelo MPS para Software (MPS-SW), que vem ao encontro das necessidades desse mercado, tanto para apoiar a melhoria de processos de serviços como para oferecer um processo de avaliação que atesta a aderência das práticas da organização em relação às melhores práticas do setor.
 - ❑ **MPS para gestão de pessoas (MPS-RH):** Tem como base os requisitos de gestão de pessoas dos modelos e normas que focam a definição, a avaliação e a melhoria desta gestão, oferecendo às Empresas orientações para a implementação gradativa de práticas de gestão de RH na indústria de TIC, de forma a selecionar, desenvolver e reter talentos humanos necessários ao atendimento dos objetivos organizacionais.

Melhoria de Processo de Software



MPS-BR (Melhoria do Processo de Software Brasileiro)



Melhoria de Processo de Software



MPS-BR (Melhoria do Processo de Software Brasileiro)

- Níveis de maturidade x Projetos de Projeto:



Melhoria de Processo de Software



MPS-BR (Melhoria do Processo de Software Brasileiro)

- Cada um dos níveis de maturidade é composto por um conjunto de processos.
 - ❑ **Processos de projeto:** são aqueles que são executados para os projetos de software. Esses projetos podem ser de desenvolvimento de um novo produto, manutenção ou evolução de produto.
 - ❑ **Processos organizacionais:** são os processos concebidos para fornecer os recursos necessários para que o projeto/serviço atenda às expectativas e necessidades das partes interessadas.
- **Capacidade do processo:** caracteriza o quanto o processo é capaz de alcançar os objetivos de negócio atuais e futuros. Está relacionada à execução dos processos e aos resultados esperados de cada nível de capacidade. Expressa o grau de refinamento e institucionalização com que o processo é executado na organização/unidade organizacional.

Melhoria de Processo de Software



MPS-BR (Melhoria do Processo de Software Brasileiro)

- Cada um dos níveis de maturidade é composto por um conjunto de processos.

Processos de Projeto e Processos Organizacionais



Melhoria de Processo de Software



MPS-BR (Melhoria do Processo de Software Brasileiro)

- Níveis de maturidade x Processos de Serviço:



Melhoria de Processo de Software



MPS-BR (Melhoria do Processo de Software Brasileiro)

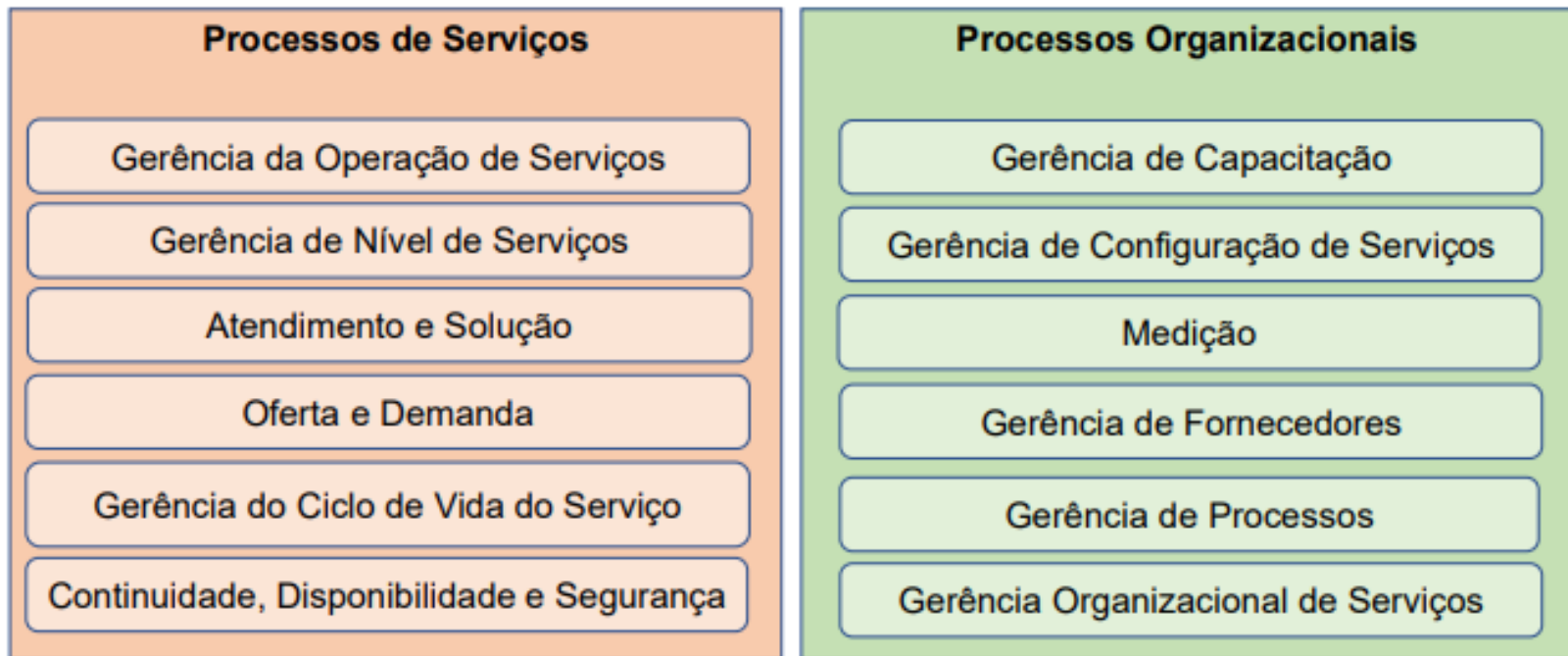
- Cada um dos níveis de maturidade é composto por um conjunto de processos.
 - ❑ **Processos de serviço:** são aqueles que são inerentes aos serviços e que podem ser utilizados para criar, operar, manter e evoluir um serviço.
 - ❑ **Processos organizacionais:** são os processos concebidos para fornecer os recursos necessários para que o serviço atenda às expectativas e necessidades das partes interessadas.
- **Capacidade do processo:** caracteriza o quanto o processo é capaz de alcançar os objetivos de negócio atuais e futuros. Está relacionada à execução dos processos e aos resultados esperados de cada nível de capacidade. Expressa o grau de refinamento e institucionalização com que o processo é executado na organização/unidade organizacional.

Melhoria de Processo de Software



MPS-BR (Melhoria do Processo de Software Brasileiro)

Processos de Serviços e de Processos Organizacionais:



Melhoria de Processo de Software



MPS-BR (Melhoria do Processo de Software Brasileiro)

Níveis de maturidade x Processos de Gestão de Pessoas

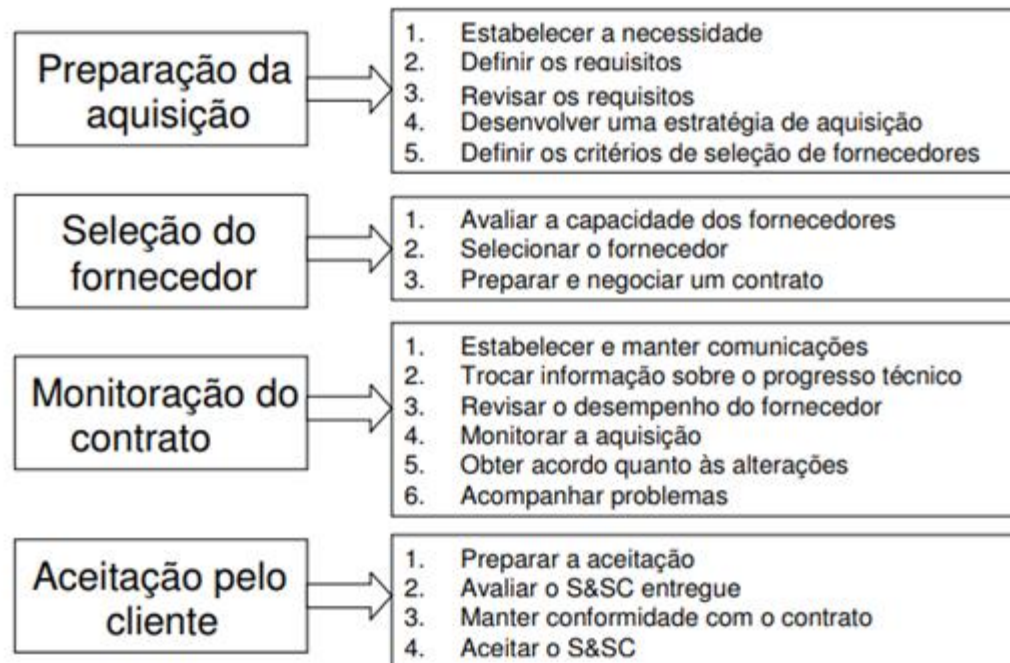
Nível	Processos
A	Melhoria Contínua da Capacidade – MCC
	Alinhamento do Desempenho Organizacional – ADO
	Inovação Contínua da Força de Trabalho – ICF
B	Integração de Competências e Grupos de Trabalho Habilitados – ICG
	Gerência Quantitativa de Desempenho – QGD
	Gerência da Capacidade Organizacional – GCP
C	Desenvolvimento de Carreira – DCA
	<i>Mentoring</i> – MEN
D	Práticas Baseadas em Competências – PBC
	Desenvolvimento de Competências – DVC
E	Desenvolvimento de Grupos de Trabalho – DGT
	Planejamento da Força de Trabalho – PFT
	Análise de Competências – ACP
F	Reconhecimento – REC
	Gerência de Pessoas – GPE (evolução)
	Capacitação e Desenvolvimento – CDV
G	Gerência de Pessoas – GPE
	Gerência de Recursos - GRC

Melhoria de Processo de Software



MPS-BR - Processo de aquisição de Software e Serviços Correlatos

- Fases do Processo:



* S&SC (Software e Serviço Correlato)

Melhoria de Processo de Software



Exercícios:

- 1) O que é o MPS-BR?
- 2) Quais são seus modelos de referência?
- 3) Quais são seus níveis de maturidade?
- 4) Quais são as fases do Processo de aquisição de Software e Serviços Correlatos?

Melhoria de Processo de Software



SPICE (Software Improvement and Capacibility Determination)

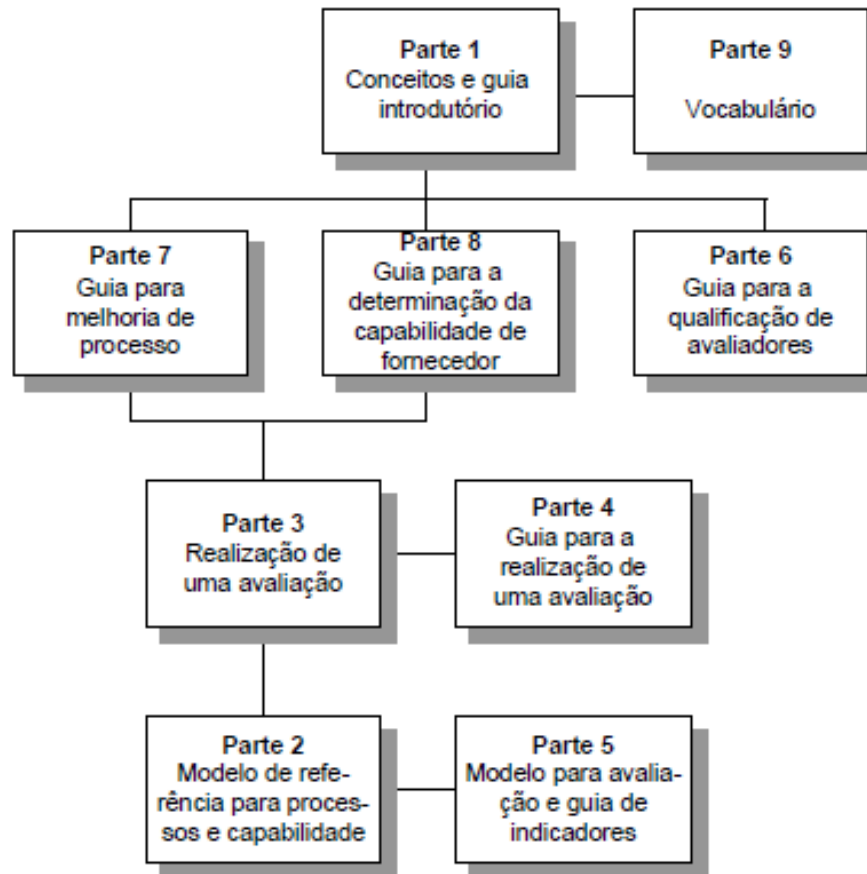
- Desenvolvido em 1993 pela “International Organization for Standardization and the International Electrotechnical Commission” (ISO/IEC).
- Avalia o software com foco na melhoria de seus processos (identifica pontos fracos e fortes, que serão utilizados para a criação de um plano de melhorias) e determina a capacidade dos processos, viabilizando a avaliação de um fornecedor em potencial.
- É um framework de avaliação que busca:
 - ☐ Facilitar o autojulgamento
 - ☐ Despertar consciência do contexto
 - ☐ Produzir um perfil do processo
 - ☐ Direcionar a adequação das atividades
 - ☐ Ser apropriado para organizações de diversos tamanhos.

Melhoria de Processo de Software



SPICE (Software Improvement and Capacibility Determination)

- É composto pelas seguintes partes:



Melhoria de Processo de Software



SPICE (Software Improvement and Capacibility Determination)

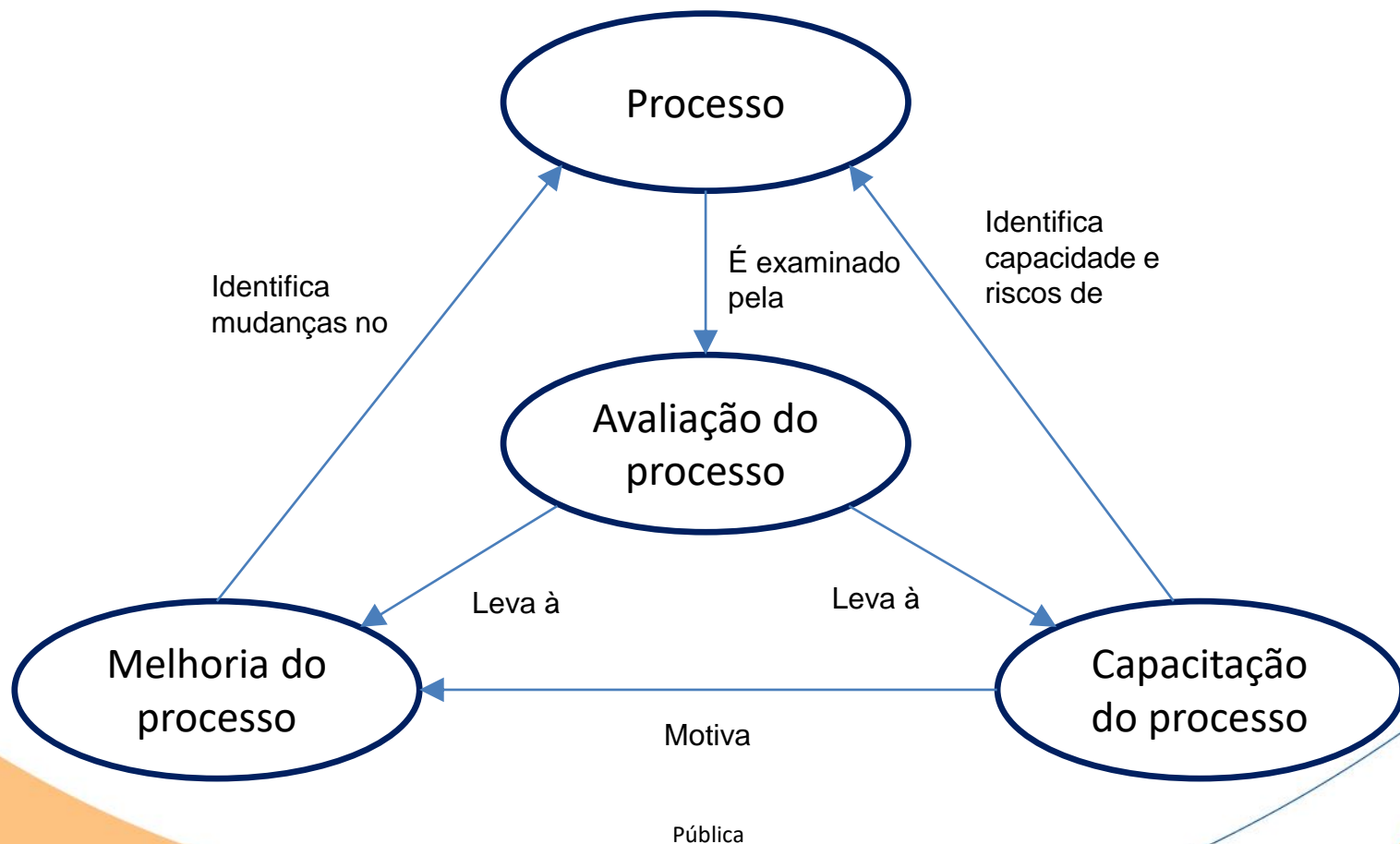
- A avaliação de processos de software tem como propósito:
 - ❑ Entender o estado dos processos de uma organização para a melhoria destes processos;
 - ❑ Determinar a adequação dos processos de uma organização para um requisito particular ou uma classe de requisitos;
 - ❑ Determinar a adequação dos processos de uma outra organização para um determinado contrato ou para uma classe de contratos.

Melhoria de Processo de Software



SPICE (Software Improvement and Capability Determination)

Avaliação de Processo de Software



Melhoria de Processo de Software



SPICE (Software Improvement and Capacibility Determination)

- O SPICE possui um modelo de referência de processos que serve como base para o processo de avaliação. Este modelo é dividido em cinco grandes categorias de processo:
 - ❑ **Cliente-Fornecedor:** consiste de processos que impactam diretamente com o cliente, tais como desenvolvimento de suporte, transição do software para o cliente e fornecimento de assistência/consultoria com relação à operação e uso do produto de software e/ou serviço.
 - ❑ **Engenharia:** consiste de processos que especificam, implementam ou mantém o produto de software com relação ao sistema e documentação ao cliente.
 - ❑ **Suporte:** consiste de processos que podem ser empregados por qualquer um dos outros processos.

Melhoria de Processo de Software



SPICE (Software Improvement and Capacibility Determination)

- ❑ **Gerência:** consiste de processos que contém práticas de natureza genérica que podem ser usadas por quem gerencia projetos ou processos dentro de um ciclo de vida de software..
- ❑ **Organização:** consiste de processos que estabelecem os objetivos de negócios da organização.

Melhoria de Processo de Software



SPICE (Software Improvement and Capability Determination)

Cliente-Fornecedor

Processo	Descrição
CUS.1	Processo de Aquisição: Obtenção do produto de software e/ou serviço que satisfaça as necessidades expressas pelo cliente.
CUS.1.1	Preparação do Processo de Aquisição: Obtenção do produto de software e/ou serviço que satisfaça as necessidades expressas pelo cliente.
CUS.1.2	Processo de Seleção do Fornecedor: Identifica a organização que irá se responsabilizar pelo projeto definido em CUS.1.1
CUS.1.3	Monitoramento do Processo de Fornecimento: Monitora o fornecedor durante o processo de desenvolvimento do produto de software e/ou serviço.
CUS.1.4	Processo de Aceitação do Cliente: Aprovação do fornecedor que satisfaz todas as condições impostas pelo cliente.
CUS.2	Processo de Fornecimento: Fornece o software ao cliente conforme os requisitos concordados.
CUS.3	Processo de Elicitação de Requisitos : Localiza problemas que venham interferir na vida do produto de software e/ou serviço.
CUS.4	Processo de Operação : Define o processo de operação e entendimento do produto de software e fornecimento de suporte ao cliente.
CUS.4.1	Processo de Uso Operacional : Assegura a correta e eficiente operação do produto de software pela duração de entendimento de seu uso e seu ambiente.
CUS.4.2	Processo de Suporte ao Cliente : Estabelece e mantém um nível de suporte, assistência e consultoria aceitável ao cliente.

Melhoria de Processo de Software



SPICE (Software Improvement and Capability Determination)

Engenharia

Processo	Descrição
ENG.1	Processo de Desenvolvimento : Transforma os requisitos de software anteriormente acordado em um produto de software funcional.
ENG.1.1	Processo de Análise dos Requisitos e Projeto do Sistema : Estabelece os requisitos e arquitetura do sistema, identifica quais requisitos deverão ser alocados para quais elementos do sistema.
ENG.1.2	Processo de Análise dos Requisitos : Estabelece os requisitos dos componentes do sistema.
ENG.1.3	Processo de Projeto de Software : Define um modelo que acomode os requisitos do software.
ENG.1.4	Processo de Construção do Software : Produz módulos executáveis de software e verifica se os mesmos refletem o projeto do sistema.
ENG.1.5	Processo de Integração de Software : Junta os módulos executáveis de forma que integrem os itens conforme o projeto do sistema.
ENG.1.6	Processo de Teste de Software : Testa a integridade do sistema conforme seus requisitos.
ENG.1.7	Processo de Teste e Integração do Sistema : Integra os vários componentes do software, produzindo um sistema completo que satisfaça as expectativas expressas nos requisitos do sistema, incluindo manuais e hardware.
ENG.2	Processo de Manutenção de Software e Sistema : Modificações, migrações e exclusões de componentes do sistema, são realizadas neste processo.

Melhoria de Processo de Software



SPICE (Software Improvement and Capability Determination)

Suporte

Processo	Descrição
SUP.1	Processo de Documentação : Desenvolve e mantém documentos que registram informações produzidas por processos ou atividades.
SUP.2	Processo de Gerenciamento de Configurações : Estabelece e mantém a integridade de todos os produtos ou projetos.
SUP.3	Processo de Garantia da Qualidade : Assegura que os produtos e atividades de um processo ou projeto esta conforme todos os padrões aplicáveis.
SUP.4	Processo de Verificação : Confirma que cada produto de software e/ou serviço de um processo ou projeto estão de acordo com os requisitos.
SUP.5	Processo de Validação : Confirma que os requisitos especificados estão de acordo com o software que foi trabalhado.
SUP.6	Processo de Revisão Conjunta : Mantém um entendimento comum com o cliente do progresso do projeto contra os objetivos do contrato e que deve ser continuado para assegurar o desenvolvimento de um produto que satisfaça o cliente.
SUP.7	Processo de Auditoria : Confirma independentemente que os produtos e processos empregados estão conforme os requisitos acordados.
SUP.8	Processo de Resolução de Problemas : Assegura que todos os problemas descobertos são analisados e resolvidos.
SUP.9	Processo de Medição : Coleta e analisa dados relativos aos produtos desenvolvidos e processos implementados dentro da unidade organizacional e demonstra objetivamente a qualidade dos produtos.
SUP.10	Processo de Reuso : Promove e facilita o reuso de software.

Melhoria de Processo de Software



SPICE (Software Improvement and Capability Determination)

Gerência

Processo	Descrição
MAN.1	Gerenciar o projeto: Define os processos necessários para estabelecer, coordenar e gerenciar um projeto e seus recursos necessários para a produção do produto.
MAN.2	Gerenciar a qualidade: Gerencia a qualidade dos serviços e dos produtos do projeto e assegura que os mesmos satisfaçam as necessidades do cliente.
MAN.3	Gerenciar riscos: Identifica continuamente e alivia os riscos do projeto, desde de seu início e durante o ciclo de vida do mesmo.
Man.4	Gerenciar subcontratantes: Seleciona os subcontratados qualificados e gerencia a eficiência dos mesmos.

Melhoria de Processo de Software



SPICE (Software Improvement and Capability Determination)

Organização

Processo	Descrição
ORG.1	Construir o negócio: Suprir as pessoas na organização e projetos com uma visão e cultura que capacitem os mesmos para uma função eficiente.
ORG.2	Definir o processo: Construir e reutilizar bibliotecas de definições de processos (incluindo padrões, procedimentos e modelos).
ORG.3	Melhorar o processo: Melhoria contínua, efetiva e eficaz dos processos usados pela organização que estão em linha com as necessidades do negócio da mesma.
ORG.4	Prover recursos de treinamento: Suprir a organização e os projetos com pessoas capazes e eficientes, fornecendo a elas treinamento.
ORG.5	Prover infra-estrutura organizacional: Sustentar um ambiente estável e confiante, integrando métodos de desenvolvimento com ferramentas que auxiliem nos processos da organização.

Melhoria de Processo de Software



SPICE (Software Improvement and Capacibility Determination)

- Possui 6 níveis de capacitação para avaliação dos processos:

- ☐ **Nível 0 (Incompleto):** Há uma falha geral em realizar o objetivo do processo. Não existem produtos de trabalho nem saídas do processo facilmente identificáveis.

Atributos de processos:

Nenhum

- ☐ **Nível 1 (Realizado):** O objetivo do processo em geral é atingido, embora não necessariamente de forma planejada e controlada. Há um consenso na organização de que as ações devem ser realizadas e quando são necessárias. Existem produtos de trabalho para o processo e eles são utilizados para atestar o atendimento dos objetivos

Atributos de processos:

PA 1.1: Atributo de Execução de Processo

Melhoria de Processo de Software



SPICE (Software Improvement and Capacibility Determination)

- ❑ **Nível 2 (Gerenciado):** O processo produz os produtos de trabalho com qualidade aceitável e dentro do prazo. Isto é feito de forma planejada e controlada. Os produtos de trabalho estão de acordo com padrões e requisitos.

Atributos de processos:

PA 2.1: Atributo da Gerência de Execução

PA 2.2: Atributo de Gerência de Produto de Trabalho

- ❑ **Nível 3 (Estabelecido):** O processo é realizado e gerenciado usando um processo definido, baseado em princípios de Engenharia de Software. As pessoas que implementam o processo usam processos aprovados, que são versões adaptadas do processo padrão documentado.

Atributos de processos:

PA 3.1: Atributo de Definição de Processo

PA 3.2: Atributo de Implementação de Processo

Melhoria de Processo de Software



SPICE (Software Improvement and Capacibility Determination)

- ❑ **Nível 4 (Previsível):** O processo é realizado de forma consistente, dentro dos limites de controle, para atingir os objetivos. Medidas da realização do processo são coletadas e analisadas. Isto leva a um entendimento quantitativo da capacitação do processo a uma habilidade de prever a realização.

Atributos de processos:

PA 4.1: Atributo de Medição de Processo

PA 4.2: Atributo de Controle de Processo

- ❑ **Nível 5 (Otimizado):** A realização do processo é otimizada para atender às necessidades atuais e futuras do negócio. O processo atinge seus objetivos de negócio e consegue ser repetido.

❑ **Atributos de processos:**

PA 5.1: Atributo de Inovação de Processo

PA 5.2: Atributo de Otimização de Processo

Melhoria de Processo de Software



SPICE (Software Improvement and Capacibility Determination)

- Cada atributo é avaliado com base no nível de atingimento dos requisitos solicitados:
 - ☐ **N:** o atributo não foi atingido pelo processo;
 - ☐ **P:** o atributo foi atingindo apenas parcialmente pelo processo;
 - ☐ **L:** o atributo foi atingido largamente pelo processo; e
 - ☐ **F:** o atributo foi atingido completamente pelo processo.
- Para atingir um nível de capacidade, um processo tem que ter avaliação L ou F nos atributos do nível e F em todos os atributos dos níveis anteriores.

Melhoria de Processo de Software



SPICE (Software Improvement and Capacibility Determination)

Atributo de Processo	Níveis				
	Nível 1 – Realizado	Nível 2 - Gerenciado	Nível 3 – Estabelecido	Nível 4 – Previsível	Nível 5 - Otimizado
PA 1.1: Atributo de Execução de Processo	L ou F	F	F	F	F
PA 2.1: Atributo da Gerência de Execução		L ou F	F	F	F
PA 2.2: Atributo de Gerência de Produto de Trabalho		L ou F	F	F	F
PA 3.1: Atributo de Definição de Processo			L ou F	F	F
PA 3.2: Atributo de Implementação de Processo			L ou F	F	F
PA 4.1: Atributo de Medição de Processo				L ou F	F
PA 4.2: Atributo de Controle de Processo				L ou F	F
PA 5.1: Atributo de Inovação de Processo					L ou F
PA 5.2: Atributo de Otimização de Processo					L ou F

Exercícios:

- 1) O que é o SPICE?
- 2) Como é composto o SPICE?
- 3) Quais são as suas categorias de processos?
- 4) Quais são os níveis de capacidade utilizados na avaliação dos processo?
- 5) O que são atributos de processos e como eles são avaliados?
- 6) Qual o critério para que um processo alcance um nível de capacidade?

Normas de Qualidade



Normas de Qualidade



Normas

- Seguem algumas normas voltadas para a qualidade:
 - ❑ **NBR ISO/IEC 12119:** Pacotes de software (Cancelada)
 - ❑ **NBR ISO/IEC 9241-11:2021:** Orientações sobre Usabilidade
 - ❑ **NBR ISO/IEC 14598:** Avaliação de produto de software (Cancelada)
 - ❑ **NBR ISO/IEC 9000-3:** Normas de Gestão da Qualidade e Garantia da Qualidade
 - ❑ **NBR ISO/IEC 12207:** Processos do ciclo de vida de software (Cancelada)
 - ❑ **NBR ISO/IEC 14001:** Sistemas da gestão ambiental
- Cada norma pode ser adquirida no link abaixo:
<https://www.abntcatalogo.com.br/>

Normas de Qualidade



NBR ISO/IEC 9241-11 - Orientações sobre Usabilidade

- Define usabilidade e explica como identificar a informação necessária a ser considerada na especificação ou avaliação de usabilidade de um computador em termos de medidas de desempenho e satisfação do usuário. É dada orientação sobre como descrever explicitamente o contexto de uso do produto (hardware, software ou serviços) e as medidas relevantes de usabilidade. A orientação é dada na forma de princípios e técnicas gerais, em vez de requisitos para usar métodos específicos.
- As orientações podem ser usadas na aquisição, projeto, desenvolvimento, avaliação, e comunicação da informação sobre usabilidade. A ISO 9241-11 inclui orientações sobre como a usabilidade de um produto pode ser especificada e avaliada. Ela se aplica tanto a produtos de uso geral quanto a produtos sendo adquiridos ou sendo desenvolvidos dentro de uma organização específica.

Normas de Qualidade



NBR ISO/IEC 9241-11 - Orientações sobre Usabilidade

- Também explica como medidas de desempenho e satisfação do usuário podem ser usadas para medir como qualquer componente de um sistema afeta todo o sistema de trabalho em uso.
- A orientação inclui procedimentos para medir usabilidade, mas, não detalha todas as atividades a serem realizadas. A especificação de métodos de medidas detalhados baseados no usuário está além do objetivo da ISO 9241-11.
- A ISO 9241-11 aplica-se ao trabalho de escritório com computadores. Ela também pode ser aplicada em outras situações
- onde o usuário está interagindo com um produto para alcançar seus objetivos. A ISO 9241, partes 12 a 17, fornece NBR 9241-11:2002 3 recomendações condicionais que são aplicadas em contextos de uso específico. As orientações nesta parte da ISO 9241 podem ser usadas em conjunto com a ISO 9241, partes 12 a 17, para ajudar a identificar a aplicabilidade de recomendações individuais.

Normas de Qualidade



NBR ISO/IEC 9241-11 - Orientações sobre Usabilidade

- A ISO 9241-11 enfoca a usabilidade e não fornece uma ampla cobertura de todos os objetivos de projeto ergonômico citados na ISO 6385. Entretanto, o fato de projetar para a usabilidade irá contribuir positivamente para os objetivos ergonômicos, tal como a redução de possíveis efeitos de uso adversos do sistema de computadores sobre a saúde, segurança e desempenho humano.
- A ISO 9241-11 não cobre os processos de desenvolvimento de sistema. Os processos de projeto centrados no ser humano para sistemas interativos são descritos na ISO 13407.

Normas de Qualidade



NBR ISO/IEC 9000-3 - Normas de Gestão da Qualidade e Garantia da Qualidade

- Esta norma define diretrizes para facilitar a aplicação da norma ISO 9001 a organizações que desenvolvem, fornecem e mantêm software. Destina-se a fornecer orientação quando um contrato entre duas partes exigir a demonstração da capacidade do fornecedor em desenvolver, fornecer e manter produtos de software.
- Para cada item da ISO 9001 existe um correspondente na ISO 9000-3 que o detalha e o adequa ao software. De fato, a ISO 9000-3 é um guia para a aplicação da ISO 9001 para o desenvolvimento, fornecimento e manutenção de software. As diretrizes propostas na ISO 9000-3 cobrem questões como:
 - ☐ Entendimento dos requisitos funcionais entre contratante e contratado;
 - ☐ Uso de metodologias consistentes para o desenvolvimento de software;
 - ☐ Gerenciamento de projeto desde a concepção até a manutenção.

Normas de Qualidade



NBR ISO/IEC 14001 - Sistemas da gestão ambiental

- Esta Norma especifica os requisitos relativos a um sistema da gestão ambiental, permitindo a uma organização desenvolver e implementar uma política e objetivos que levem em conta os requisitos legais e outros requisitos por ela subscritos e informações referentes aos aspectos ambientais significativos. Aplica-se aos aspectos ambientais que a organização identifica como aqueles que possa controlar e aqueles que possa influenciar. Em si, esta Norma não estabelece critérios específicos de desempenho ambiental!.
- Esta Norma se aplica a qualquer organização que deseje :
 - a) estabelecer, implementar, manter e aprimorar um sistema da gestão ambiental,
 - b) assegurar-se da conformidade com sua política ambiental definida,
 - c) demonstrar conformidade com esta Norma ao

Normas de Qualidade



NBR ISO/IEC 14001

- 1) fazer uma autoavaliação ou autodeclaração, ou
- 2) buscar confirmação de sua conformidade por partes que tenham interesse na organização, tais como clientes, ou
- 3) buscar confirmação de sua autodeclaração por meio de uma organização externa, ou
- 4) buscar certificação/registro de seu sistema da gestão ambiental por uma organização externa.

Normas de Qualidade



Exercícios:

- 1) Cite algumas características da norma 9241 (usabilidade).
- 2) Cite algumas características da norma 9000-3 (gestão da qualidade)
- 3) Cite algumas características da norma 14001 (gestão ambiental).

Auditoria de Sistemas



Auditoria de Sistemas



Conceito

- A filosofia de auditoria em tecnologia de informação está calcada em confiança e em controles internos. Estes visam confirmar se os controles internos foram implementados e se existem; caso afirmativo, se são efetivos.
- As atividades de auditoria de tecnologia de informação, além de tentar utilizar os recursos de informática para auditar o próprio computador, também visa automatizar todos os processos de auditoria.

Conceito

- Como em qualquer outra atividade, as empresas de auditoria também buscam um diferencial competitivo. Entre outros objetivos, consideram-se:
 - a) Melhorar a eficiência e reduzir os custos;
 - b) Melhorar a qualidade do trabalho de auditoria, reduzindo, assim, os níveis de risco de auditoria;
 - c) Atender às expectativas dos clientes, que esperam de seus auditores o mesmo grau de automatização que utilizam em seu negócio;
 - d) Preparar-se para a globalização dos negócios, que vem exigindo uma globalização dos auditores;
 - e) Manter-se entre as maiores e mais reconhecidas pelo mercado;
 - f) Criar valor agregado para seus clientes, ajudando-os a reduzir os riscos nos processos operacionais.

Competências e perfil do auditor

- Seguem algumas competências e perfil do auditor de sistemas de informação:
 - ☐ Planejar a auditoria de sistemas documentando nível de risco aparente do ambiente.
 - ☐ Compreender os negócios, o setor, as unidades, os comitês, os executivos, a gerência, o organograma e as partes interessadas, além do plano de valor agregado.
 - ☐ Compreender e certificar-se dos processos-chave e dos procedimentos operacionais para mitigar os riscos.
 - ☐ Compreender e certificar-se da integridade da comunicação de dados interinstituição e intrainstituição e certificar-se do controle.
 - ☐ Compreender e certificar-se da classificação de informações, de pessoas, dados conforme aplicações essenciais para negócios.

Competências e perfil do auditor

- ❑ Verificar a proteção de dados, informações, pessoas e ativos em geral em TI e certificar-se do controle, de SLAs, da disponibilidade e das políticas.
- ❑ Compreender e verificar o suporte para usuários, para atender às funções críticas aos negócios.
- ❑ Verificar a consistência e a confiabilidade da evolução de sistemas aplicativos em pró-tendências dos negócios e seus funcionamentos.
- ❑ Certificar-se da continuidade em caso de contingências nas operações gerais dos negócios.
- ❑ Certificar-se dos contratos de disponibilidades de operações com outsourcing e cloud computing.

Controles Internos

- O conceito de controle interno em um sistema de informação, conforme declaração do Instituto Americano de Contadores Públicos, significa “planos organizacionais e coordenação de um conjunto de métodos e medidas adotado numa empresa, a fim de salvaguardar o ativo, verificar a exatidão e veracidade de registros contábeis, promover a efetividade de sistema de informação contábil e eficiência operacional, assim como fomentar uma grande adesão às políticas da organização”.
- A responsabilidade total sobre o sistema de controle interno de uma organização pertence à gerência e, mais particularmente, àqueles cuja autoridade foi delegada no que diz respeito a responsabilidades funcionais.

Princípios de controles internos

- Alguns princípios seguidos são:
 - ❑ **Supervisão:** A gerência, por objetivos, procedimentos e tomada de decisões, deve manter um controle que a capacite a uma supervisão efetiva dentro do ambiente de tecnologia de informação.
 - ❑ **Registro e comunicação:** A gerência da empresa deve estabelecer critérios para criação, processamento e disseminação de informação de dados, através da autorização e registro de responsabilidades.
 - ❑ **Segregação de funções:** As responsabilidades e ocupações incompatíveis devem estar segregadas de maneira a minimizar as possibilidades de perpetuação de fraudes e até suprimir erro e irregularidade na operação normal.

Princípios de controles internos

- ❑ **Classificação de informação:** A gerência deve estabelecer um plano para classificação de informação que melhor sirva às necessidades da organização, em conformidade com os princípios de contabilidade geralmente aceitos e também padrões de auditoria geralmente aceitos.
- ❑ **Tempestividade:** A gerência deve delinear procedimentos, monitorar os registros corretos das transações econômicas, financeiras e contábeis das empresas, processando-as e comunicando os resultados às pessoas necessárias em tempo hábil.
- ❑ **Auditoriabilidade:** Os procedimentos operacionais devem permitir a programação e verificação periódica no que concerne à precisão do processo de processamento de dados e de geração de relatório, de acordo com as políticas.

Princípios de controles internos

- ❑ **Controle independente:** Os sistemas em funcionamento devem ter procedimentos adequados para identificação e correções de erros no fluxo de processamento, inclusive nos processos executados concomitantemente.
- ❑ **Monitoramento:** A gerência deve possuir acesso master ao sistema e controle de uso, que lhe permita fazer o acompanhamento das transações.
- ❑ **Implantação:** A gerência deve planejar a aquisição, o desenvolvimento, a manutenção e a documentação de sistema, de forma a coincidir com as metas empresariais.
- ❑ **Contingência:** A gerência deve implementar um plano adequado e procedimentos para prevenir-se contra as falhas de controles que podem surgir durante especificações de sistema, desenho, programação, teste e documentação de sistemas e nas fases pós-implantações.

Auditoria de Sistemas



Princípios de controles internos

- ❑ **Custo efetivo:** Investimentos em tecnologia de informação devem ser propriamente planejados, a fim de coincidirem com o custo efetivo.

Avaliação dos procedimentos de controles internos

- O processo de avaliação implica naquele utilizado pelo auditor para confirmar as assertivas em reação à transação ou aos processos contábeis de uma entidade, seja ele de controles internos sistêmicos ou de transações sobre os documentos, com a finalidade de apoiar a formação de uma opinião e proporcionar uma informação sobre o grau em que as assertivas estão em conformidade com um conjunto de padrões identificados. Isso pode ser físico (exame de inventário, ativos fixos, papéis com valores monetários e registros dos documentos contábeis, etc.); a lógica (customizações para atender certas leis pela solução ERP; exame da configuração de sistemas de segurança de informação para atender às políticas) e do processo de compliance que fornece o mais alto nível de garantia sobre a afirmação de que um auditor pode proporcionar.

Técnicas de auditoria

- Seguem algumas técnicas utilizadas:
 - a) **Produtividade:** com a melhoria no processo de planejamento, auxilia na redução do ciclo operacional de auditoria, focalizando o exercício nas funções mais importantes.
 - b) **Custo:** reduz custos relacionados com auditoria, pois não necessita de geração de relatórios e listagens para análise. Evita também o gasto referente ao desenvolvimento de programas pelas empresas de auditoria com a disponibilidade de software generalistas.
 - c) **Qualidade assegurada:** com o uso de softwares que têm padrões devidamente testados, o auditor aproveita para adequar seus trabalhos aos padrões internacionais geralmente aceitos obrigatoriamente, aumentando a qualidade dos serviços prestados.

Técnicas de auditoria

- d) **Valor agregado:** disponibiliza tempestivamente resultados para a tomada de decisões que necessitam de mudanças de rumos mais urgentes, facilitando a correção também dos desvios ou irregularidades em tempo hábil.
- e) **Benefícios corporativos:** proporcionam às empresas de auditoria os seguintes benefícios – eficiência nos trabalhos, eficácia, em termos de execução, de somente aqueles passos que atenuam riscos aparentes, otimização dos recursos disponíveis, principalmente a respeito de compartilhamento de ambientes entre vários auditores em múltiplas localidades; melhoria na imagem do auditor, por estar utilizando tecnologia mais apropriada.

Auditoria de Sistemas



Técnicas de auditoria

f) **Benefícios para o auditor:** proporcionam aos auditores os seguintes benefícios: independência, pois não dependerá do auditado ou do seu funcionário de processamento de dados para gerar relatórios; renovação do foco de auditoria, visando atender às expectativas e tendência do mercado; eliminação das tarefas mais repetitivas, que geralmente podem ser automatizadas, dentre outras.

Lei Sarbanes Oxley (SOX)

- Lei americana assinada em 2002 pelo senador Paul Sarbanes e pelo deputado Michael Oxley com o objetivo de garantir a criação de mecanismos de auditoria e segurança nas empresas, reduzindo assim os riscos de fraudes. Afeta as empresas que mantêm ADRs (American Depositary Receipts) negociadas na bolsa de valores de Nova York. Por conta da lei, algumas regras de conformidade precisam ser atendidas, o que implicam em algumas revisões de auditoria:
 - ❑ **Project Management:** Revisão de gerenciamento de projetos de sistemas.
 - ❑ **Backup and Recovery:** Revisão de procedimentos de backup e recuperação de dados.
 - ❑ **Data Interface:** Revisão de gerenciamento de interface de dados.
 - ❑ **Emergency Changes:** Revisão de mudanças emergenciais.
 - ❑ **IT Security:** Revisão de políticas e procedimentos de segurança física.
 - ❑ **User Authentication:** Revisão de segurança de informação, autenticação do usuário.
 - ❑ **Job Scheduling:** Revisão de procedimentos das operações.

Auditoria de Sistemas



Exercícios:

- 1) Cite alguns objetivos da auditoria de sistemas.
- 2) Cite algumas competências de um auditor.
- 3) O que são controles internos?
- 4) Cite alguns princípios de controles internos.
- 5) Cite algumas técnicas utilizadas por auditores.
- 6) O que é a SOX?
- 7) Cite algumas regras de conformidade que precisam ser atendidas segundo a SOX.

Referências



Referências

- ALBERTIN, Marcos; GUERTZENSTEIN, Viviane. Planejamento avançado da qualidade: sistemas de gestão, técnicas e ferramentas. Rio de Janeiro: Alta Books, 2018.
- IMONIANA, Joshua Onome. Auditoria de sistemas de informação. 3. ed. São Paulo: Atlas, 2016.
- VAZQUEZ, Carlos Eduardo. Análise de pontos de função: medição, estimativas e gerenciamento de projetos de software. 13. ed. São Paulo: Érica, 2013.

