

UNIDADE 2 – TÉCNICAS PARA MEDIÇÃO DE SOFTWARE**MÓDULO 1 – HORA****01****1 - ESTIMATIVA DO ESCOPO DO PROJETO**

O planejamento de projetos engloba uma série de atividades que constituem a base para a realização dos trabalhos da equipe de desenvolvimento. O **planejamento** engloba:

- A estimativa de atributos de produtos de trabalho e tarefas, determinando as necessidades de recursos.
- A definição das necessidades de recursos.
- A negociação de compromissos.
- A elaboração do cronograma.
- A identificação dos riscos do projeto.

O Plano do Projeto provê a base para a realização das atividades de execução e controle necessárias para atender aos compromissos firmados entre a equipe de desenvolvimento e o cliente. Este plano é revisado continuamente durante o andamento do projeto, com o objetivo de tratar mudanças em requisitos e acordos, estimativas inadequadas, ações corretivas e mudanças de processos. Pode ser considerado como o plano global que viabiliza o controle do projeto.

Os parâmetros do planejamento de um projeto de *software* incluem todas as informações necessárias para realizar o planejamento, a organização, a alocação dos recursos, o direcionamento, a coordenação, o reporte e o orçamento.

As estimativas dos parâmetros de planejamento devem possuir uma base sólida e confiável, que permite a elaboração de um plano capaz de viabilizar o alcance dos objetivos do projeto. Para a estimativa destes parâmetros é necessário considerar os requisitos do projeto, sejam eles requisitos do produto de *software*, requisitos impostos pela organização e/ou pelo cliente e quaisquer requisitos que afetam o projeto.

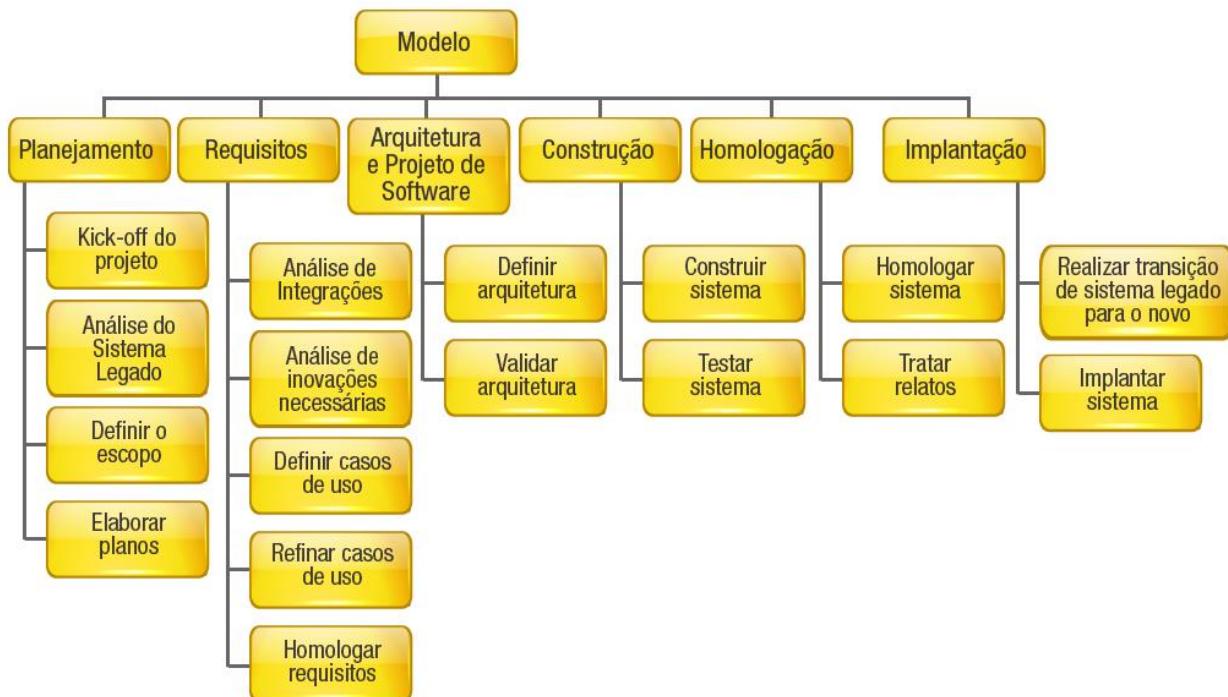
Elaborar as estimativas do projeto é uma atividade de grande importância que, para ser bem-sucedida, requer utilização de métodos, técnicas além da experiência em projetos anteriores. Estimativas bem feitas contribuem de forma significativa para o sucesso de um projeto que atenda as expectativas, conforme custo e prazo viáveis.

02

A **estimativa do escopo de um projeto** envolve o estabelecimento de uma WBS (*Work Breakdown Structure*), também conhecida como diagrama ou estrutura analítica do projeto.

A WBS evolui conforme o andamento do projeto. O estabelecimento da WBS permite a divisão do projeto em componentes gerenciáveis e conectados.

Tipicamente a WBS é uma estrutura orientada a produto, produto de trabalho ou tarefa. Provê um mecanismo organizacional e de referência para a definição do esforço, do cronograma e de responsabilidades. A imagem abaixo apresenta um exemplo de uma WBS para substituição de um sistema legado por um novo:



03

2 - ESTIMATIVAS DOS PRODUTOS DE TRABALHO E DOS ATRIBUTOS DAS TAREFAS

O tamanho é considerado a entrada principal de muitos modelos de mercado utilizados para as estimativas de esforço, custo e cronograma.

Outros **atributos** podem ser utilizados como base para elaboração de tais estimativas, como por exemplo:

- Número e complexidade de requisitos.
- Número e complexidade de interfaces.
- Volume de dados.
- Número de funções.
- Linhas de código-fonte.
- Número de classes e objetos.
- Número de tabelas de bancos de dados.

- Número de campos em tabelas.
- Elementos arquiteturais.
- Número de entradas e saídas.
- Proximidade dos clientes, usuários e fornecedores.

04

3 - ESTIMATIVAS DE ESFORÇO E CUSTO

As estimativas de esforço e custo são baseadas nos resultados de análises que fazem uso de modelos ou de dados históricos aplicáveis ao tamanho, às atividades e a outros parâmetros de planejamento.

O nível de confiabilidade destas estimativas é baseado nos motivos pelo qual determinado modelo é selecionado e pela natureza dos dados envolvidos. Podem ocorrer situações em que os dados históricos não são aplicáveis, por exemplo, quando a organização não possui experiência anterior em determinada atividade ou na elaboração de um produto de trabalho novo.

Exemplos de **insumos para as estimativas de esforço e custo**:

- Estimativas fornecidas por especialistas.
- Riscos do projeto.
- Competências críticas e papéis necessários para realizar determinada tarefa.
- WBS (*Work Breakdown Structure*).
- Tipo do modelo de ciclo de vida do projeto.
- Base histórica.

05

4 - HOMEM-HORA (HH)

Homem-hora é uma unidade que se refere ao trabalho realizado por uma pessoa durante uma hora.

É uma medida amplamente utilizada em diversas áreas para estimar a quantidade total de trabalho ininterrupto necessário para realizar determinada atividade. Alguns exemplos do conceito:

- Uma tarefa estimada em 10 HH será realizada por 1 recurso em 10 horas. A mesma tarefa se realizada por 2 recursos levará 5 horas para ser concluída.
- Uma tarefa estimada em 30 HH será realizada por 3 recursos. O tempo necessário para conclusão desta tarefa será de 10 horas.
- Um recurso trabalhou em uma tarefa 6 horas por dia, durante 3 dias. O trabalho realizado foi de 18 HH.

- Uma tarefa estimada em 160 HH, se for realizada por 5 recursos, cada um trabalhando 8 horas por dia, levará 4 dias para ser concluída. Caso a mesma tarefa seja realizada por 4 recursos, a duração aumentará para 5 dias.

A quantidade de homem-hora necessária para cada serviço serve de base para a definição da equipe. A partir do índice, define-se o número de trabalhadores necessários para realizar cada serviço, conforme explica o exemplo abaixo:

- Uma demanda de melhoria foi submetida e o tamanho em Pontos de Função da mesma é 20 PF.
- A produtividade média da equipe é de 8 horas/PF.
- Para a construção dos 20 PF serão necessários 2,5 HH.

O HH planejado é o total de horas estimadas (orçadas) para realizar o trabalho previsto para o projeto.

O HH real é o total de horas consumidas na execução do trabalho previsto para o projeto.

06

- **Procedimento exemplo para estimativa de HH para projetos de software**

As seguintes atividades podem ser realizadas para a estimativa da quantidade de homens-hora requerida para um projeto de desenvolvimento de *software*:

- 1  Dividir o projeto em componentes.
- 2  Definir os profissionais necessários.
- 3  Estimar o tempo necessário para realizar cada componente do projeto.
- 4  Definir o cronograma necessário para o projeto.
- 5  Ajustar as estimativas no decorrer do projeto.

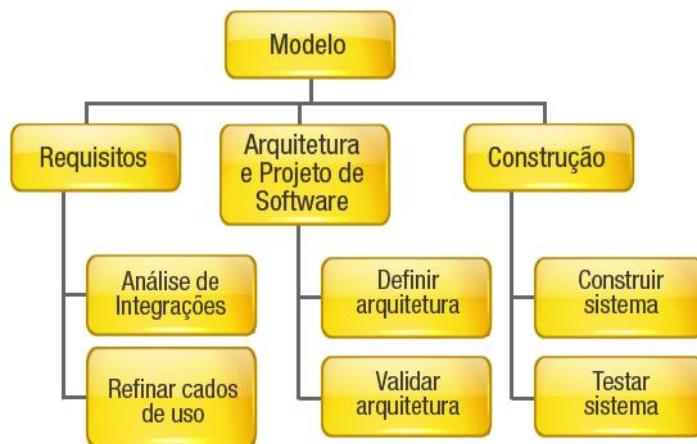
Veremos cada uma dessas atividades a seguir.

07

a) Dividir o projeto em componentes

Esta atividade tem por objetivo a divisão do projeto em partes menores, onde cada parte pode ser definida conforme o tipo de trabalho envolvido.

A WBS (*Work Breakdown Structure*) ou Estrutura Analítica do Projeto – produto da estimativa de escopo do projeto – permite a divisão do projeto em componentes gerenciáveis. Cada componente da WBS pode ser tratado como uma unidade que deve ser estimada individualmente, conforme ilustra a figura abaixo:



08

b) Definir os profissionais necessários

Esta atividade tem por objetivo avaliar as características do projeto e as eventuais restrições tecnológicas e de ambiente para definir o staff necessário para a execução do que está previsto na WBS (*Work Breakdown Structure*) do projeto.

Exemplos de **características** que podem influenciar a escolha dos profissionais:

- Nível de complexidade do negócio.
- Nível de conhecimento existente dos recursos humanos disponíveis com relação ao negócio.
- Nível de complexidade das tarefas e produtos previstos.
- Nível de conhecimento e experiência da equipe de desenvolvimento na tecnologia a ser utilizada para a construção do produto de *software*.
- Nível de experiência da equipe

Exemplos de **restrições** que podem influenciar a escolha dos profissionais:

- Quantidade de recursos humanos disponíveis para o projeto.
- Restrições orçamentárias.
- Restrições de prazos, por exemplo, a necessidade de atender a um prazo legal.

09

c) Estimar o tempo necessário para realizar cada componente do projeto

Esta atividade tem por objetivo a estimativa do total de homens-hora que são necessários para executar os componentes previstos na WBS (*Work Breakdown Structure*). A recomendação é levar em consideração o total de horas dedicadas ao trabalho necessário do início ao fim.

Para cada componente identificado é necessário:

- Avaliar as características e eventuais restrições.
- Definir o volume de homens-hora necessários.

Com base na WBS utilizada, os seguintes cenários são exemplos possíveis:

- Para as atividades de Requisitos;
- Para as atividades de Arquitetura e Projeto de *Software*;
- Para as atividades de Construção.

É importante estimar também o esforço necessário em tarefas adicionais e essenciais para a entrega de um produto de *software* com qualidade: tarefas de planejamento, monitoramento e controle do projeto, gestão de contrato, entre outras.

Atividades de Requisitos

Para as atividades de Requisitos, caso os recursos humanos disponíveis tenham pouco conhecimento do negócio envolvido, é esperado que um tempo maior seja necessário para entendimento do negócio e definição de que requisitos o *software* deve possuir para atender as necessidades do negócio.

Atividades de Arquitetura e Projeto de *Software*

Para as atividades de Arquitetura e Projeto de *Software*, caso os recursos humanos disponíveis tenham experiência na tecnologia a ser utilizada, bem como experiência em projetos similares, as estimativas de homem-hora podem ser obtidas a partir da experiência vivida em projetos anteriores. Conforme maturidade da organização, uma base histórica de estimativas pode ser utilizada como apoio à definição das estimativas para o novo projeto. Neste mesmo cenário, caso a experiência da equipe seja nenhuma ou pouca, é esperado que a quantidade de homens-horas necessária seja maior – uma equipe com pouca experiência não deverá ter o mesmo desempenho de uma equipe com experiência.

Atividades de Construção

Para as atividades de Construção, caso a organização já esteja habituada a trabalhar com o reaproveitamento de componentes, é esperado que o esforço em HH para a construção dos pontos de função estimados seja menor.

10

d) Definir o cronograma necessário do projeto

Esta atividade tem por objetivo definir uma “linha de tempo” para o projeto – o cronograma – a partir das estimativas elaboradas para cada componente da WBS (*Work Breakdown Structure*).

Nesta atividade é possível:

- A identificação de tarefas que podem ser realizadas em paralelo.
- A identificação de tarefas que possuem dependências entre si.
- A estimativa de tempo considerando uma quantidade restrita de recursos e/ou o compartilhamento de recursos entre projetos.
- A estimativa de tempo baseada na produtividade das equipes envolvidas.
- A estimativa de tempo considerando o apoio de equipes externas.
- A estimativa de tempo considerando uma restrição aplicável à entrega do produto (por exemplo, o atendimento de um prazo legal).

A figura exemplifica o cronograma parcial de um projeto, elaborado a partir da WBS exemplo:

Nome	Duração	Trabalho	Predecessoras	Nome do Recurso
Projeto SISRH	101 dias	1.168 horas		
Requisitos	30 dias	360 horas		
Análise de integrações	15 dias	120 horas		João Pedro
Refinar casos de uso	15 dias	240 horas	3	João Pedro; Maria Alice
Arquitetura e Projeto de Software	21 dias	168 horas	2	
Definir arquitetura	14 dias	112 horas		Ana Cláudia
Validar arquitetura	7 dias	56 horas	6	Mário Sérgio
Construção	50 dias	640 horas	5	
Construir sistema	30 dias	480 horas		Ana Cristina; Júlio César
Testar sistema	20 dias	160 horas	9	Maria Alice

11

O cronograma é o roteiro que define as tarefas e os pontos de controle a serem acompanhados e controlados no decorrer do progresso do projeto.

O esforço estimado é distribuído nas tarefas específicas. De acordo com PRESMANN (2011, 7ª Edição) alguns **princípios básicos** devem ser seguidos e aplicados à medida que o projeto evolui:

- Divisão do trabalho;
- Interdependência;
- Alocação do tempo;
- Validação do esforço;
- Definição de responsabilidades;
- Definição dos resultados;
- Definição de pontos de controle.

Divisão do trabalho

O projeto deve ser dividido em uma série de atividades e tarefas gerenciáveis.

Interdependência

Deve ser definida a interdependência de cada tarefa. Algumas tarefas podem acontecer de forma sequencial, outras podem ocorrer em paralelo. Algumas tarefas podem ser iniciadas sem que o resultado de outra tarefa esteja disponível. Outras atividades podem ocorrer de forma independente.

Alocação do tempo

Para cada tarefa definida, deve ser alocado o trabalho previsto. Devem ser definidas as datas de início e término para cada tarefa, bem como se o trabalho será realizado em tempo integral ou parcial.

Validação do esforço

Cada projeto possui um número definido de pessoas na equipe. É necessário assegurar que não seja alocado um número de profissionais superior ao realmente disponível para o projeto.

Definição dos resultados

Cada tarefa deve possuir um resultado definido, seja ele um produto de trabalho (artefato) ou parte de um produto de trabalho.

Definição de responsabilidades

Cada tarefa disposta no cronograma deverá ser atribuída a um membro específico da equipe.

Definição de pontos de controle

Cada tarefa ou grupo de tarefas deve estar associado a um ponto de controle no projeto.

12

e) Ajustar as estimativas no decorrer do projeto

Esta atividade tem por objetivo a revisão e o ajuste das estimativas no decorrer do projeto.

Durante as atividades de monitoramento do projeto é possível obter visibilidade do progresso do projeto. O monitoramento tipicamente envolve a medição dos parâmetros atuais do projeto e a comparação dos valores atuais com as estimativas realizadas em tempo de planejamento.

Caso desvios do planejamento inicial sejam identificados é possível a tomada de ações corretivas. Tipicamente, as necessidades de ajustes das estimativas são identificadas durante o monitoramento e controle do projeto.

13

5- ESTIMATIVA BASEADA EM HORA – RECOMENDAÇÃO DO SISP

O *Guia de Boas Práticas para Contratação de Soluções de Tecnologia da Informação – TI* descreve os processos, atividades e artefatos do MCTI - Modelo de Contratação de Soluções de TI, com o objetivo de apoiar os profissionais na realização de contratações de Soluções de TI. Foi elaborado pelo Departamento de Serviços de Redes, através da Coordenação de Normas e Contratações de Tecnologia da Informação – CONTI.

O mesmo é resultado do processo de revisão da Instrução Normativa N° 04 de 12 de novembro de 2010, publicada pela Secretaria de Logística e Tecnologia da Informação – SLTI do Ministério do Planejamento, Orçamento e Gestão – MP.



De acordo com o respectivo guia, é vedado adotar a métrica homem-hora ou equivalente para aferição de esforço, salvo mediante justificativa e sempre vinculada à entrega de produtos de acordo com prazos e qualidade previamente definidos.

14

6- ESTIMATIVA WIDEBAND DELPHI

Wideband Delphi é uma técnica que tem por objetivo prover uma estimativa de esforço baseada no consenso obtido a partir da opinião de especialistas.

Este método foi popularizado por Barry Boehm no livro *Software Engineering Economics* (1981).

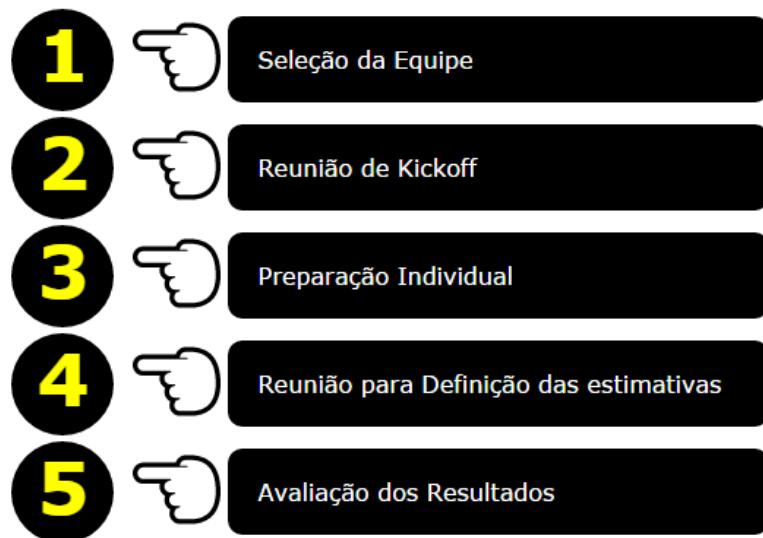
- **Pré-requisito para a aplicação do método**

A entrada necessária ao processo de estimativa é a descrição do escopo do produto de *software*, que deve estar previamente acordado entre todas as partes envolvidas: *stakeholders*, usuários, gerentes e equipe de desenvolvimento.

A WBS (*Work Breakdown Structure*) é um instrumento que pode ser utilizado como insumo para a aplicação da técnica.

- **Procedimento de estimativa**

O procedimento de estimativa é ilustrado conforme a figura abaixo:



Veremos cada uma dessas etapas a seguir.

15

a) Seleção da equipe

O Gerente de Projetos seleciona o moderador e a equipe que vai participar da elaboração das estimativas:

- O Moderador deve atuar como um facilitador na aplicação da técnica e deve ser capacitado nas estimativas, apesar de não as fazer durante a execução da técnica.
- Recomenda-se selecionar especialistas no processo que será estimado.
- O número de participantes recomendado é de 2 a 4 estimadores.

b) Reunião de KICKOFF

Esta é a primeira reunião do grupo de pessoas envolvidas nas estimativas:

- O Moderador deve apresentar o objeto da estimativa, que será base para o trabalho dos demais envolvidos.
- A partir da WBS do projeto o grupo alinha o entendimento sobre o escopo do produto.
- A unidade a ser utilizada para a estimativa é definida (por exemplo, homem-hora).
- O moderador é o responsável pela condução da reunião.

16

c) Preparação individual

A partir das definições existentes, cada estimador membro da equipe faz a estimativa do esforço das tarefas definidas na reunião de *kickoff*. O estimador pode realizar seu trabalho tendo como base:

- A sua experiência pessoal.
- Base histórica de estimativas.

Algumas orientações básicas devem ser seguidas para elaboração das estimativas.

Tarefas adicionais podem ser identificadas e podem ser anotadas para posterior apresentação aos membros da equipe.

Um formulário padrão para a estimativa do esforço das atividades deve ser preenchido. A figura abaixo apresenta um exemplo simples para registro das estimativas:

Tarefa	Estativa 1	Mudança 1	Mudança 2	Mudança 3	Final
Mudança	---				---
Total					

orientações básicas

Orientações básicas a serem seguidas para elaboração das estimativas:

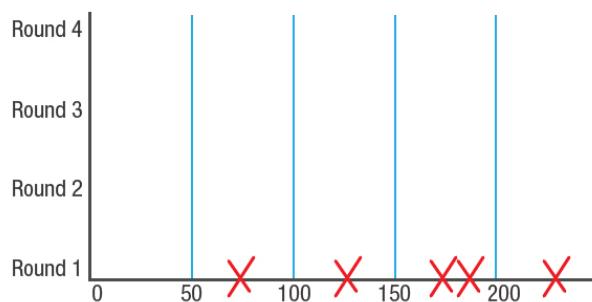
- Assumir que uma única pessoa irá realizar a tarefa.
- Assumir que todas as tarefas serão realizadas de forma sequencial. Não há necessidade de preocupação com qual sequência será definida ou com a eventual dependência que as tarefas possam possuir.
- Assumir que um esforço ininterrupto pode ser dedicado a cada tarefa.

d) Reunião para definição das estimativas

Esta é a segunda reunião do grupo de pessoas envolvido nas estimativas:

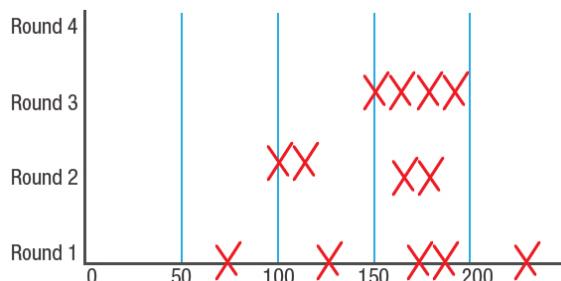
- O moderador deve coletar e consolidar as estimativas de esforço propostas.
- Durante esta reunião as estimativas são revistas e discutidas no grupo.

A figura abaixo ilustra um exemplo de como pode ser representado o gráfico com as estimativas iniciais:



Cada “x” no gráfico representa a estimativa indicada por um membro da equipe. Cada “Round” apresentado no gráfico corresponde a cada reunião do grupo.

Esta reunião pode ser repetida até que o grupo chegue a um consenso. A figura abaixo ilustra um exemplo de como pode ser representado o gráfico com as estimativas ao longo das reuniões:



e) Avaliação dos resultados

Após cada reunião para definição das estimativas, o Gerente de Projetos e o Moderador trabalham juntos com o objetivo de consolidar os resultados obtidos. Um sumário de resultados deve ser elaborado.

f) Revisão de resultados

Os resultados são avaliados por todos os envolvidos. As estimativas finais são divulgadas.

19

- **Benefícios e desafios da técnica Widebrand Delphi**

São considerados **benefícios** da técnica:

- A simplicidade.
- As estimativas obtidas em consenso são mais precisas do que as estimativas individuais.
- Questões são discutidas, acordos definidos e documentados.

São considerados **desafios** para a aplicação do método Wideband Delphi:

- Obter a cooperação de todos os envolvidos.
- Obter o consenso de todos os envolvidos.
- A condução adequada dos trabalhos, com foco no objetivo final.

20

7- ESTIMATIVA PARA O DESENVOLVIMENTO ÁGIL

A engenharia de *software* “ágil” combina uma filosofia e um conjunto de diretrizes de desenvolvimento de *software*.

A *Agile Alliance* define 12 princípios de agilidade. São eles:

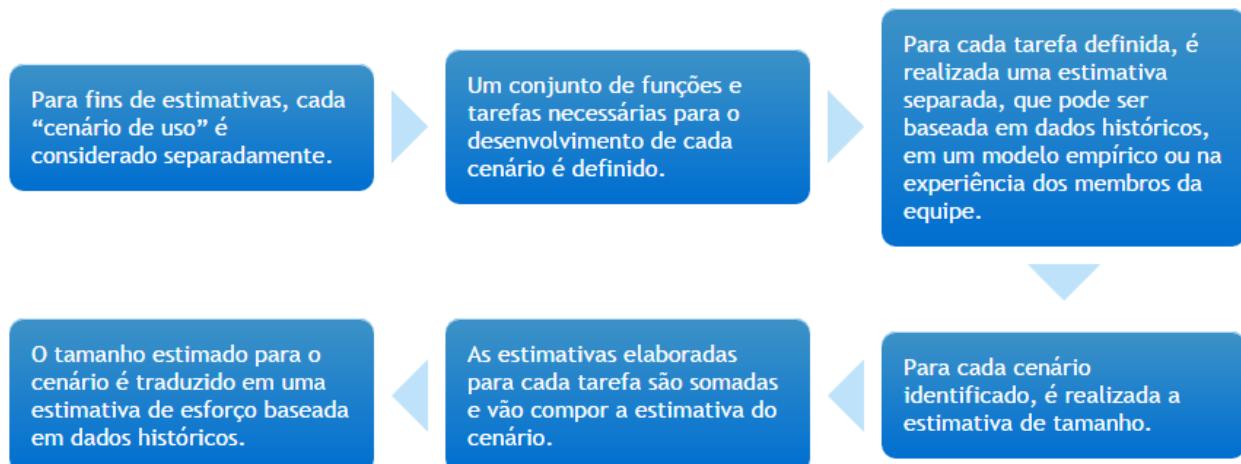
1. Nossa maior prioridade é satisfazer o cliente através da entrega contínua de um *software* de “valor”.
2. As mudanças em requisitos são bem-vindas, mesmo em tempo de desenvolvimento. Os processos ágeis apoiam a mudança como uma vantagem competitiva para o cliente.
3. Entregue *software* funcionando com uma frequência de duas semanas a dois meses, escolhendo sempre a menor escala de tempo possível.
4. O pessoal de negócios e os desenvolvedores devem trabalhar juntos no projeto diariamente.
5. Construa os projetos com pessoas motivadas. Forneça o ambiente, os equipamentos e as ferramentas de que elas precisam e confie que elas farão o trabalho.
6. Uma conversa “cara a cara” é a melhor forma de transmitir e receber informação do time de desenvolvimento.
7. *Software* funcionando é a principal medida de progresso.
8. Processos ágeis promovem um desenvolvimento sustentado. Gerência, desenvolvedores e usuários devem ser capazes de manter um ritmo constante indefinidamente.
9. A atenção contínua, a excelência técnica e um bom design aumentam a agilidade.
10. Simplicidade – a arte de maximizar a quantidade de trabalho que não precisou ser feito.
11. As melhores arquiteturas, designs e requisitos surgem de times autogerenciados.
12. Os intervalos regulares, o time reflete sobre como se tornar mais eficaz, e então ajusta seu

comportamento de acordo com as reflexões.

21

Veja alguns exemplos de modelos ágeis de processo de desenvolvimento (PRESSMAN, 2010).

A estimativa para projetos ágeis usa uma **abordagem de decomposição**. Os seguintes **passos** estão previstos:



É realizada a soma das estimativas de esforço para todos os cenários, que devem ser construídos em um incremento específico.

exemplos

- *Extreme Programming (XP)*.
- DAS – Desenvolvimento Adaptativo de Software (*Adaptative Software Development – ASD*).
- DSDM (*Dynamic Systems Development Method* – Método de Desenvolvimento Dinâmico de Sistemas).
- *Scrum*.
- *Crystal*.
- FDD (*Feature Driven Development* – Desenvolvimento Guiado por Características).
- LSD (*Lean Software Development*).
- Modelagem Ágil (*Agile Modeling – AM*).
- *Agile Unified Process (AUP)*.

22

8- ESTIMATIVA PARA PROJETOS DE APLICAÇÕES BASEADAS NA WEB (WEBAPP)

Os projetos de WebApp (aplicações baseadas na web) muitas vezes adotam o modelo de processo de desenvolvimento ágil. A medição de pontos de função pode ser adaptada para prover o tamanho de um *software* desenvolvido como um WebApp.

PRESSMAN (2011, 7^a Edição) cita que de acordo com Roetzheim a seguinte **abordagem** pode ser utilizada para adaptar os pontos de função para a estimativa de WebApp:

- **Entradas (inputs)**

Cada tela de entrada ou formulário [form], cada tela de manutenção e cada conjunto de guias [tab].

- **Saídas (outputs)**

Cada página web estática, cada script de página web dinâmica e cada relatório.

- **Tabelas (tables)**

Cada tabela lógica na base de dados e, se estiver fazendo uso de XML para armazenar dados em um arquivo, cada objeto XML (ou coleção de atributos XML).

- **Interfaces (interfaces)**

Retêm sua definição como arquivos lógicos (por exemplo, formatos únicos de registros) em nossas fronteiras de saída do *software*.

- **Consultas (queries)**

Publicadas externamente ou usam uma interface orientada por mensagem. Um exemplo típico são referências externas DCOM ou COM.

23

Ainda de acordo com PRESSMAN, Mendes e seus colegas sugerem que o volume em tamanho de uma WebApp é bem mais determinado coletando-se:

- **Medidas associadas à aplicação**, por exemplo, número de páginas, contagem de mídias, número de funções.
- As **características das páginas Web**, por exemplo, complexidade de vínculo, complexidade gráfica.
- As **características de mídia**, por exemplo, duração da mídia.
- As **características funcionais**, por exemplo, tamanho do código, tamanho do código reutilizado.

Tais medidas podem ser utilizadas para o desenvolvimento de modelos empíricos de estimativa de esforço total do projeto, do esforço para a criação da página, do esforço de autorização de mídia e do esforço de *scripting*.

Considera-se que tais modelos necessitam de amadurecimento antes da utilização dos mesmos com mais segurança.

24

RESUMO

A elaboração das estimativas de um projeto de desenvolvimento de *software* é uma das principais atividades previstas no Planejamento de Projetos.

O nível de precisão das estimativas dependerá de uma série de fatores, entre eles:

- A utilização de insumos adequados e que refletem a realidade. O planejamento do escopo do produto a ser desenvolvido é atividade essencial e seu principal produto – o escopo do produto a ser construído ou do projeto em si, demonstrado através da WBS – é insumo básico para a elaboração das estimativas.
- A utilização de técnicas de estimativas adequadas às necessidades e características do projeto.
- O envolvimento dos profissionais certos na elaboração das estimativas.
- O monitoramento adequado do projeto, com vistas ao alcance dos objetivos definidos e a devida adequação das estimativas à realidade.

A estimativa em homem-hora (HH) é amplamente utilizada no mercado para a definição do esforço requerido para a execução das tarefas do projeto, bem como para a elaboração dos produtos de trabalho previstos. Quando se trata de análise de produtividade, sua utilização pode ser combinada com a Análise de Pontos de Função. Recomendações legais em nível de Governo Brasileiro definem recomendações quanto à utilização deste tipo de estimativa na Administração Pública Federal.

A estimativa *Wideband Delphi* é uma técnica que tem por objetivo prover uma estimativa de esforço a partir da avaliação de especialistas. É realizada conforme um processo básico.

Com o advento dos processos ágeis de desenvolvimento de *software*, alguns métodos para estimativas de esforço têm sido revistos e devidamente ajustados para atender a uma nova realidade na entrega de produtos de *software*: a utilização de processos executados por equipes autogerenciáveis, que visam a entrega contínua de um produto de valor para o cliente.

UNIDADE 2 – TÉCNICAS PARA MEDIÇÃO DE SOFTWARE

MÓDULO 2 – SERVIÇOS

01

1 - CONCEITO E CARACTERÍSTICAS DOS SERVIÇOS

Um **serviço** pode ser conceituado como toda atividade realizada em alguma área de atuação que tem por objetivo produzir um resultado – seja este resultado a execução de uma tarefa ou a criação de algo de valor a quem se destina o serviço prestado.

Os bancos prestam **serviços às pessoas** jurídicas e pessoas físicas, tais como:

- Fornecimento de cartão com função débito.
- Realização de até quatro saques, por mês, em guichê de caixa, inclusive por meio de cheque ou de cheque avulso, ou em terminal de autoatendimento.
- Realização de consultas mediante utilização da internet.
- O aditamento de contratos (exemplo de serviço diferenciado).

Os estabelecimentos assistenciais de saúde prestam **serviços hospitalares**, tais como:

- Enfermagem.
- Atendimento terapêutico.
- Internação de pacientes.
- Laboratório e radiologia.

02

O ITIL (*Information Technology Infrastructure Library*) V3 define **serviço** como:

“Meio de entregar algo de valor aos clientes, facilitando o alcance dos resultados que os clientes esperam, tirando deles a propriedade dos cursos e riscos específicos”.

Do ponto de vista do cliente, a criação do valor de um serviço é uma função de duas variáveis:

UTILIDADE

Possui o desempenho desejado ou redução das restrições de desempenho.

GARANTIA

Disponibilidade, capacidade, continuidade e segurança suficientes para o uso.

Os serviços podem ser prestados tanto para clientes externos à organização, quanto para clientes internos.

ITIL

Information Technology Infrastructure Library - ITIL é um conjunto de boas práticas a serem aplicadas na infraestrutura, operação e gerenciamento de serviços de tecnologia da informação (ITSM).

03

De acordo com LAS CASAS (2004) os serviços possuem as seguintes **características**:

- **Intangibilidade**

A característica de intangibilidade significa que os serviços são abstratos. Esta característica torna difícil a inspeção antes do consumo do serviço, o que torna difícil avaliar seu resultado.

- **Inseparabilidade**

O aspecto da inseparabilidade dos serviços refere-se a outro importante determinante mercadológico de comercialização. Não se pode produzir ou estocar serviços como se fossem bens. Geralmente, os serviços são prestados quando vendedor e comprador estão frente a frente. Por isso é necessária uma capacidade de prestação de serviço antecipada.

- **Heterogeneidade**

Refere-se à impossibilidade de se manter a qualidade do serviço constante. Os serviços são produzidos pelo ser humano, que é por natureza instável, a qualidade da produção será também instável.

- **Simultaneidade**

A simultaneidade dos serviços nos diz que produção e consumo ocorrem ao mesmo tempo e, sendo assim, será sempre necessário considerar o momento de contato com a clientela como fator principal de qualquer esforço mercadológico.

04

2 - SERVIÇOS DE TI

Diversos tipos de serviços de tecnologia da informação podem ser prestados. São exemplos típicos de serviços de TI:

- Armazenamento de dados.
- Bancos de dados.
- Acesso à Internet.
- Antivírus.
- Backup de dados.
- Servidor de rede.
- Acesso à rede.
- Correio eletrônico.
- Impressão corporativa.
- *Storage*.
- Infraestrutura de rede.
- Desenvolvimento de sistemas.
- Suporte a hardware.

De acordo com LAUDON & LAUDON (2007), os serviços de TI podem ser organizados em grupos:

- Desenvolvimento de padrões de TI;
- Desenvolvimento e suporte de aplicações;
- Gestão das instalações físicas;
- Gestão de TI;
- Educação em TI;
- Manutenção das plataformas computacionais;
- Pesquisa e desenvolvimento em TI;
- Telecomunicações.

Desenvolvimento de padrões de TI

São os serviços que se referem às atividades que definem as políticas que determinam como a TI será empregada na organização.

Desenvolvimento e suporte de aplicações

São os que se referem às atividades de construção e manutenção das aplicações de negócios (sistemas gerenciais, sistemas CRM, ERP).

Gestão das instalações físicas

São aqueles responsáveis por administrar e desenvolver as instalações físicas necessárias aos serviços de informática, telecomunicações e administração de dados.

Gestão de TI

São os serviços que se referem ao planejamento e organização da infraestrutura, à coordenação das atividades de TI, administração dos gastos em TI, à gestão de projetos, etc.

Educação em TI

São os serviços que oferecem capacitação e treinamento aos usuários no uso dos sistemas corporativos e nos demais aplicativos.

Manutenção das plataformas computacionais

São os serviços que existem para garantir que os vários dispositivos computacionais de propriedade da organização, desde computadores de grande porte a laptops e notebooks, sejam mantidos em pleno funcionamento.

Pesquisa e desenvolvimento em TI

São aqueles serviços que se referem às atividades de pesquisa em sistemas e tecnologias com vistas à inovação em TI na organização.

Telecomunicações

São os que fornecem conectividade de dados, voz e vídeo a funcionários, clientes e fornecedores os quais fazem uso desses recursos.

05

O **Gerenciamento de Serviços** pode ser definido como:

“Um conjunto de capacitações organizacionais especializadas para fornecer valor aos clientes na forma de serviço”, ou seja, de transformar recursos em serviços valiosos.

As capacitações citadas nesta definição podem ser interpretadas como processos e funções para gerenciar serviços ao longo do seu ciclo de vida.

Em Gerenciamento de Serviços de TI, os conceitos de “**cliente**” e “**usuário**” possuem uma diferenciação:

- **Cliente**

É o elemento que paga pelos serviços de TI. Quando a área de tecnologia é um departamento dentro uma organização, os clientes da TI serão unidades de negócio da organização. Em se tratando de uma organização que presta serviços em TI, os clientes são as entidades por ela atendidas.

- **Usuário**

É a pessoa que faz uso dos serviços de TI.

06

3 - UST – UNIDADE DE SERVIÇO TÉCNICO

A UST – Unidade de Serviço Técnico – é uma medida utilizada para a mensuração de serviços. Na Engenharia de Software a UST é utilizada no contexto da prestação de serviços em Tecnologia da Informação.

Como métrica para quantificação de unidades de serviço técnico, a UST é consideravelmente utilizada em contratos de prestação de serviços de variados tipos e complexidade.

3.1- Atribuição de valores à UST

No momento da definição da quantidade de UST aplicável a determinado serviço:

- A quantidade de UST pode variar conforme tecnologia envolvida na prestação do serviço.
- A quantidade de UST a ser atribuída a tarefa realizada por prestador de serviço pode variar conforme nível de conhecimento e especialização do profissional.
- A quantidade de UST pode ser definida de acordo com a natureza, a complexidade e criticidade do serviço.

Os valores atribuídos podem ser revisados e reajustados, conforme acordo entre as partes (provedor de serviços e cliente).

O valor a ser cobrado pela UST nos contratos deve ser definido conforme necessidades do provedor de serviços.

As informações de custos e a relação das mesmas com a UST devem ser estabelecidas conforme necessidades do provedor de serviços e/ou requisitos do cliente expressos em contrato.

07

3.2- Utilização da UST em contratos de governo

No Governo Federal, a métrica UST tem sido utilizada em contratos de prestação de serviços.

As **regras e recomendações de uso** são definidas pelos órgãos conforme:

- Critérios predefinidos e estabelecidos internamente.

- Características específicas do órgão.
- Recomendações gerais e práticas em uso em outros órgãos da Administração Pública.

08

3.3- Abordagens de utilização da UST

Sobre as formas de utilização da UST, observam-se as seguintes abordagens:

- Escolha da UST por permitir a execução especializada e a aferição do serviço prestado em termos de qualidade, custo, prazos com observação de acordos de níveis de serviço bem definidos.
- A quantidade de UST pode ser definida pela relação entre força de trabalho e dias úteis estimados de realização do serviço.
- O número de UST estimadas para um serviço pode ser dado pela relação entre a quantidade de tarefas previstas para o serviço e a produtividade da equipe envolvida.
- Uso de metodologia própria para medição dos serviços, elaborada pelo órgão para uso interno, desenvolvida a partir do conceito básico da UST.
- Utilização da Ordem de Serviço como instrumento básico para formalização do serviço a ser prestado, bem como do número de UST requerido para a execução do serviço.
- Definição da demanda por serviço com escopo predefinido.
- Custeamento da UST.
- Definição de tarefas mensuráveis através UST onde existem resultados esperados e padrões de qualidade a serem atendidos. A complexidade definida para as tarefas pode considerar diversas variáveis, como por exemplo, a prioridade do serviço, as características dos profissionais do mercado e sua capacidade em cumprir as atividades.
- Previsão de um volume global de UST a ser consumido no decorrer de um contrato (total e conforme periodicidade mensal). Em certos casos há previsão de um consumo mínimo.
- Uso de fator de ponderação da UST conforme níveis de complexidade das atividades previstas.
- Quantidade total previstas de UST divididas conforme categoria dos serviços.
- Previsão em contrato do valor global e unitário (em reais) da UST a ser praticada.
- Uso de indicadores que permitam mensurar o nível de conformidade do resultado entregue com o resultado esperado.

09

3.4- Uma visão sobre as abordagens para utilização da UST

Em Relatório de Auditoria Operacional realizada pelo TCU – Tribunal de Contas da União [TC 002.116/2015-4] foi informado que:

- Foi evidenciada a utilização da UST – Unidade de Serviços Técnicos quando a técnica de Análise de Pontos de Função não é a mais adequada.
- “A técnica consiste em listar uma série de serviços na forma, por exemplo, de um catálogo e valorá-los a fim de pagar mediante a conclusão.” (conforme item 4.1.2 do relatório).
- As atividades em que mais se identificaram pagamentos por UST foram levantamentos de

requisitos e sustentação de sistemas.

3.5- Escopo não recomendado para uso da UST

Há entendimento de que o uso da UST é **inadequado** para:

- Serviços que não podem ser monitorados, devido às suas características.
- Serviços que não geram um produto ou resultado verificável.

10

4 - BOAS PRÁTICAS QUE SUSTENTAM A UTILIZAÇÃO DA UST

4.1- Definição dos serviços da TI

Esta prática consiste na concepção e criação de serviços de TI que visam realizar uma estratégia organizacional definida previamente.

A ITIL v3 define o **Desenho de Serviço** como:

“O desenho de serviços de TI apropriados, inovadores, incluindo suas arquiteturas, processos, políticas e documentação, para atender os requisitos do negócio atuais e futuros”.

Para que os serviços sejam definidos de forma adequada, cinco aspectos devem ser levados em consideração:

- Serviço = solução completa;
- Sistemas e ferramentas de gerenciamento de serviços;
- Arquiteturas tecnológicas e sistemas de gestão;
- Processos;
- Métricas.

Serviço = solução completa

O desenho de um novo serviço ou a alteração de um serviço existente deve ser tratado como um projeto de uma solução completa, aderente aos requisitos estabelecidos pelo negócio. Os requisitos, os recursos e capacitações necessárias para o serviço devem estar de acordo com a estratégia organizacional.

Sistemas e ferramentas de gerenciamento de serviços

Desenhar sistemas e ferramentas que sejam capazes de apoiar os serviços em todos os momentos do ciclo de vida.

Arquiteturas tecnológicas e sistemas de gestão

Serviços, aplicações, dados/informação, infraestrutura e ambiente que tenham as capacidades necessárias para a operação consistente dos serviços.

Processos

Desenhar processos de TI e de Gerenciamento de Serviços, papéis, responsabilidades e habilidades relacionadas para que os serviços possam ser operados, apoiados e mantidos.

Métricas

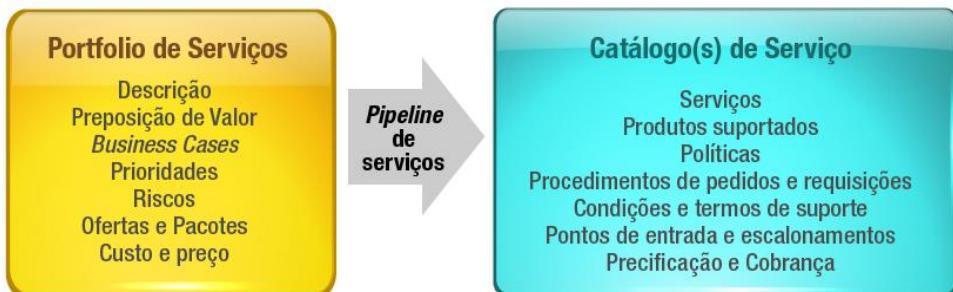
Desenhar métricas e métodos para medição da qualidade do processo de desenho do serviço, em termos do seu progresso, conformidade com requisitos, eficácia e eficiência.

11

4.2- Definição do portfólio de serviços

O Portfólio de Serviços representa os compromissos e investimentos feitos por um provedor de serviços para todos os clientes e espaços de mercado.

A figura abaixo apresenta os elementos típicos de um portfólio e de um catálogo de serviços:



De acordo com ARAGON (2008), o Gerenciamento de Portfólio de Serviços visa governar os investimentos em gerenciamento de serviços através da empresa, e gerenciá-los para que adicionem valor ao negócio. As duas categorias para os serviços são:

- Serviços de Negócio;
- Serviços de TI.

Serviços de Negócio

Serviços definidos pelo próprio negócio da organização.

Serviços de TI

Serviços fornecidos pela TI ao negócio, nos quais são isolados do cliente a complexidade estrutural, detalhes técnicos e operações de baixo nível. Para os clientes fica exposta somente a interface de utilização do serviço.

12

4.3- Estabelecimento e uso do catálogo de serviços

Esta prática consiste na criação do Catálogo de Serviços de TI.

O Catálogo de Serviços é o instrumento que contém todos os serviços fornecidos pela TI.

O uso deste instrumento promove e institucionaliza uma fonte única de informações consistentes e atualizadas sobre todos os serviços que estão operacionais e sobre aqueles que estão sendo preparados para entrar em operação.

De acordo com ARAGON (2008), o Catálogo de Serviços tem duas subdivisões:

Catálogo de Serviços de Negócio

Contém a visão do cliente sobre os serviços de TI e os seus relacionamentos com os processos e estruturas organizacionais do negócio.

Catálogo de Serviços Técnicos

Contém detalhes técnicos de todos os serviços entregues ao cliente e os seus relacionamentos com os serviços de suporte, itens de configuração, componentes e serviços compartilhados necessários à entrega do serviço ao cliente.

13

4.3.1- A estrutura do catálogo de serviços

O Catálogo de Serviços deve conter um conjunto consistente de informações sobre os serviços da TI, por exemplo:

- Histórico de revisões (versão, data de criação/modificação).
- Índice.
- Prefácio do Diretor de TI.
- Descrição do perfil da área provedora da TI (ou da organização de TI).
- Disponibilidade (horários, meios de comunicação) do provedor de serviços de TI.

- Visão geral de serviços e produtos.
- Descrição dos serviços ou produtos.
- Especificações dos serviços.
- Entregáveis.
- Horários de serviços.
- Horários de manutenção, suporte e entrega.
- Metas de qualidade (confiabilidade, usabilidade disponibilidade, prioridade).
- Procedimento de solicitação de mudanças.

14

4.3.2- A especificação dos serviços

Abaixo são apresentados alguns exemplos para especificação de serviços:

NOME DO SERVIÇO	Atualização do ambiente de TI
DESCRÍÇÃO	Atualização de <i>Softwares</i> nas categorias sistemas operacionais (Windows e Linux), antivírus e suíte de aplicativos para escritório (Microsoft Office e OpenOffice).
ATIVIDADES	UST
Atualizar servidores de aplicação	5
Atualizar estações de trabalho e notebook dos servidores	2

NOME DO SERVIÇO	Correio Eletrônico
DESCRÍÇÃO	Serviço que permite trocar mensagens através de sistemas de comunicação eletrônicos.
ATIVIDADES	UST
Criar, alterar e excluir conta de e-mail	1
Configurar e-mail no celular	1
Criar, alterar e excluir catálogo de endereços	1

Nos exemplos anteriores, para cada atividade necessária à execução do serviço é apresentada a quantidade de UST – Unidade de Serviço Técnico necessária para a atividade, bem como o Acordo de Nível de Serviço – ANS a ser atendido.

ATIVIDADE	DESCRÍÇÃO	ENTREGA	COMPLEXIDADE	UNIDADE DE MEDIDA	UST
Criar usuário	Criar e habilitar o usuário para acesso à rede.	Usuário criado e habilitado	N/A	Por usuário	1

Instalar antivírus	Instalar antivírus na estação de trabalho do usuário e realizar verificação básica.	Antivírus instalado. Estação de trabalho verificada.	1	Por usuário	1
--------------------	---	--	---	-------------	---

No exemplo acima, o parâmetro de complexidade foi estabelecido por atividade relacionada ao serviço disponível.

15

4.4- Definição de acordos de nível de serviço

Esta prática consiste na definição de níveis de serviço em relação aos clientes, para cada serviço ou grupo de serviços oferecidos pela TI.

O **Acordo de Nível de Serviço (ANS)**, também conhecido como *SLA – Service Level Agreement* pode ser visto como um contrato ou parte de um contrato que especifica os níveis de serviços a serem prestados pela área de tecnologia.

Representa um compromisso formal estabelecido entre a TI e os consumidores de seus serviços – seus clientes.

O ANS é utilizado em vários contratos de prestação de serviços de TI e permitem:

- A identificação e a definição das necessidades dos clientes.
- A simplificação de problemas complexos.
- A promoção do diálogo quando as partes envolvidas – fornecedor e clientes – entram em conflito.
- A eliminação de expectativas irreais.
- A transformação de uma obrigação de direito em uma obrigação de resultado.

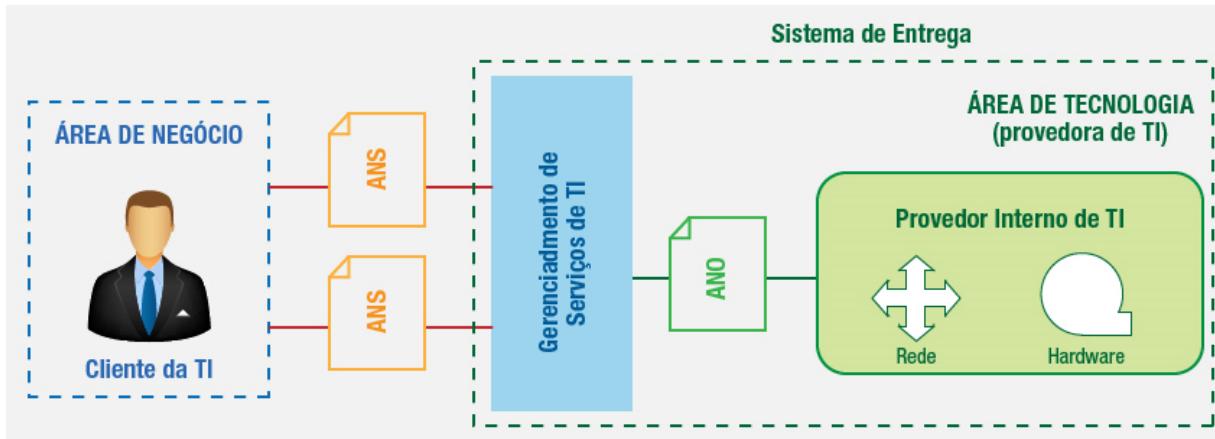


Os Acordos de Níveis de Serviços devem ser **mensuráveis** de forma a permitir o adequado monitoramento e avaliação do atendimento dos serviços da TI por ambas as partes – fornecedor e clientes.

16

Existem também os **Acordos de Níveis Operacionais (ANO)**, que são voltados para os serviços internos a TI que servem de apoio a outros serviços fornecidos para área de TI.

A figura abaixo ilustra o relacionamento entre clientes e provedores de serviços:



O cliente da área de TI consome os serviços disponíveis. Um serviço disponibilizado por um sistema de informação que, para que esteja disponível e opere conforme os ANS estabelecidos, dependente de um serviço interno de rede – sem a disponibilidade da rede, haverá comprometimento no fornecimento do primeiro serviço. A estes serviços internos serão atribuídos os Acordos de Níveis Operacionais.

O gerenciamento de níveis de serviço tem por objetivo manter e melhorar a qualidade dos serviços de TI, através de um ciclo contínuo de atividades que envolvem o planejamento, a coordenação, elaboração e o estabelecimento de acordo com metas de desempenho e responsabilidades mútuas, monitoramento e divulgação de níveis de serviços, níveis operacionais e de contratos de apoio com os fornecedores de serviços.

17

As tabelas abaixo exemplificam serviços e os ANS definidos para os mesmos:

NOME DO SERVIÇO	Atualização do ambiente de TI		
DESCRÍÇÃO	ATIVIDADES	UST	ANS
	Atualização de <i>Softwares</i> nas categorias sistemas operacionais (Windows e Linux), antivírus e suíte de aplicativos para escritório (Microsoft Office e OpenOffice).		
ATIVIDADES			
Atualizar servidores de aplicação		5	4 horas
Atualizar estações de trabalho e notebook dos servidores		2	2 horas

NOME DO SERVIÇO	Correio Eletrônico		
DESCRÍÇÃO	ATIVIDADES	UST	ANS
	Serviço que permite trocar mensagens através de sistemas de comunicação eletrônicos.		

Criar, alterar e excluir conta de e-mail	1	2 horas
Configurar e-mail no celular	1	2 horas
Criar, alterar e excluir catálogo de endereços	1	2 horas

O valor em horas definido para os ANS citados nos exemplos informa que, para que a prestação do serviço seja realizada conforme o esperado, o tempo para realização das atividades deve ser no tempo máximo apresentado pelo ANS. Por exemplo, para o serviço de Correio Eletrônico, a configuração de um e-mail no celular não pode ultrapassar mais de 2 horas. Ultrapassando este valor, haverá comprometimento do nível de atendimento do serviço.

Indicadores são tipicamente utilizados para a aferição do alcance dos níveis de serviços definidos. Formas objetivas de cálculo e meta são definidas por indicador. São exemplos destes indicadores:

- Disponibilidade dos Serviços Críticos;
- Incidentes que paralisem serviços críticos.

Disponibilidade dos Serviços Críticos

Indicador que estabelece que os serviços considerados críticos e quaisquer outros que deem suporte a eles deverão permanecer disponíveis para uso em determinado período (por exemplo, em regime 24x7).

Incidentes que paralisem serviços críticos

Indicador que permite medir a quantidade de ocorrências de incidentes causados pela contratada (fornecedor dos serviços) em equipamentos e serviços que tenham, por exemplo, algum impacto nos serviços críticos.

18

5 - HST – HORA DE SERVIÇO TÉCNICO

A **HST – Hora de Serviço Técnico** é uma unidade de medida adotada para a mensuração do esforço necessário para a execução de atividades ou elaboração de artefatos e “pacotes de trabalho” que podem ser representados através da EAP – Estrutura Análise do Projeto.

As regras para utilização da HST são estabelecidas conforme necessidade da organização que a adota.

Em linhas gerais, as seguintes abordagens de uso são identificadas:

- A quantidade de HST equivalente a cada tarefa a ser executada pode ser ajustada conforme entendimentos entre a organização contratante e a contratada, observando o valor limite de HST previsto em contrato.

- Tipicamente os contratos que preveem a utilização desta métrica estabelecem um volume de HST a ser consumido durante o contrato.
- Alguns exemplos de serviços mensurados em HST incluem serviços de ETL (*Extract Transform Load*, que trabalha com a extração de dados de fontes externas e transformação para atender necessidades que envolvam carga de dados em DW) e serviços de suporte técnico.

19

- São exemplos de **atividades** mensuradas em HST:

- Administração de bancos de dados.
- Administração de ferramentas de integração.
- Administração de ferramentas de apoio à análise de indicadores.
- Treinamento.
- Montagem de ambiente.

- São exemplos de tipos de **serviços** que podem ser mensurados em HST:

- Análise de processos de *Software*.
- Análise e garantia da qualidade.
- Gerência de configuração e mudanças.
- Apoio à gestão e qualidade de processos.
- Apoio à gestão e melhoria de processos de negócio.
- Apoio à gestão e fiscalização de contratos.

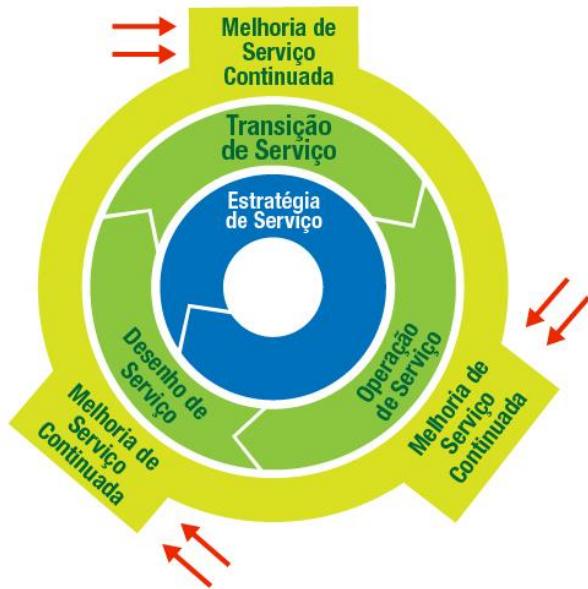
20

6 - ITIL – UM FRAMEWORK DE REFERÊNCIA PARA O GERENCIAMENTO DE SERVIÇOS

O **ITIL – Information Technology Infrastructure Library** é um framework que provê um conjunto de práticas de gerenciamento de serviços de TI testadas e comprovadas no mercado. As práticas estão organizadas segundo uma lógica de ciclo de vida de serviços.

A adoção das práticas do ITIL por uma organização promove a maturidade e a qualidade que permitem a utilização eficaz e eficiente dos seus ativos estratégicos de TI (incluindo sistemas de informação e infraestrutura de TI), sempre com foco no alinhamento e na integração com as necessidades dos clientes e dos usuários.

De acordo com ARAGON (2008), o núcleo do ITIL é composto por cinco publicações, cada uma delas relacionada a um estágio do ciclo de vida do serviço, contendo orientações para uma abordagem integrada do gerenciamento de serviços, conforme ilustra a figura abaixo:



Estratégia de Serviço

Provê orientação sobre como as políticas e processos de gerenciamento de serviços podem ser desenhadas, desenvolvidas e implementadas como ativos estratégicos ao longo do ciclo de vida de serviço. Esta publicação trata os ativos de serviço, o catálogo de serviços, o gerenciamento financeiro, o gerenciamento do portfólio de serviços, o desenvolvimento organizacional, entre outros.

Desenho do Serviço

Provê orientação para o desenho e desenvolvimento dos serviços e dos processos de gerenciamento de serviços, detalhando aspectos do gerenciamento do catálogo de serviços, do nível de serviço, da capacidade, da disponibilidade, da continuidade, da segurança da informação e dos fornecedores, além de mudanças e melhorias necessárias para manter ou agregar valor aos clientes ao longo do ciclo de vida de serviço.

Transição de Serviço

Provê orientação sobre como efetivar a transição dos serviços novos e modificados para operações implementadas, detalhando os processos de planejamento e suporte à transição, gerenciamento de mudanças, gerenciamento da configuração e dos ativos de serviços, gerenciamento de liberação e da distribuição, teste e validação de serviço, avaliação e gerenciamento do conhecimento.

Operação de Serviço

Provê orientação sobre o ciclo de vida do gerenciamento de serviços que é responsável pelas atividades do dia a dia. As orientações focam em como garantir a entrega e o suporte a serviços de forma eficiente e eficaz, e detalhando os processos de gerenciamento de eventos, incidentes, problemas, acesso e de execução de requisições.

Melhoria Continuada dos Serviços

Provê orientações, através de princípios, práticas e métodos de gerenciamento da qualidade, sobre como fazer sistematicamente melhorias incrementais e de larga escala na qualidade dos serviços, nas metas de eficiência operacional, na continuidade dos serviços, etc., tendo por base o modelo PDCA (*Plan-Do-Check-Act*), preconizado pela ISO/IEC 20000.

21

RESUMO

Serviço pode ser definido como um instrumento que tem por objetivo entregar ao importante ao cliente – onde há um resultado esperado e padrões de qualidade a serem seguidos. Organizações contratam serviços com vistas a apoiar seus negócios e obter melhores resultados.

Considerando a Tecnologia da Informação como grande aliada das organizações no alcance de bons resultados, o consumo de serviços de TI torna-se imprescindível. Neste cenário, saber definir, estruturar, operacionalizar e medir os serviços torna-se um desafio para fornecedores e clientes.

Na Engenharia de *Software*, a UST – Unidade de Serviço Técnico – é um instrumento utilizado para viabilizar a mensuração de serviços.

Os detalhes da utilização da UST são tipicamente definidos no contexto da organização, levando-se em consideração sua maturidade, suas necessidades, seus processos e ferramentas disponíveis. Diversas abordagens de uso da UST são evidenciadas, porém, é possível observar alguns alicerces que promovem a boa aplicação da técnica:

- A definição clara e objetiva de que serviços serão fornecidos pela TI.
- Estabelecimento e utilização de um Catálogo de Serviços.
- Definição do Portfólio de Serviços da TI.
- A definição de acordos de níveis de serviços.

UNIDADE 2 – TÉCNICAS PARA MEDIÇÃO DE SOFTWARE**MÓDULO 3 – COCOMO**

01

1 - O PRIMEIRO MODELO COCOMO

O COCOMO – *CO*nstructive *C*ost *M*odel – é um dos modelos de custos de *software* mais populares e bem documentados que existem em uso. Foi elaborado para permitir as estimativas de esforço, prazo, custo e tamanho da equipe para um projeto de desenvolvimento de *software*.

O modelo original COCOMO foi desenvolvido por Barry Boehm e apresentado à indústria de *software* em 1981. Este modelo teve como base 56 projetos de diferentes domínios.

O *software* poderia ser classificado de acordo com a forma de desenvolvimento praticada.

São três as formas definidas, conforme a visão de Barry Boehm:

- Modo Orgânico;
- Modo Embutido;
- Modo Semidestacado.

Vejamos cada um deles a seguir.

02

1.1- MODO ORGÂNICO

Os projetos possuem as seguintes características:

- Uso de processos flexíveis.
- Mudanças em funcionalidades, requisitos de qualidade, custos e cronogramas podem ocorrer livremente com o mínimo de *overhead*.
- Equipes de desenvolvimento pequenas.
- Equipes com experiência no desenvolvimento de sistemas com poucos requisitos.

A fórmula original para o cálculo de esforço e tempo é definida da seguinte forma:

Esforço = 3.2 EAF (Tamanho) ^{1.05}

Tempo (em meses) = 2.5 (Esforço) 0.38

Onde:

Esforço = número de pessoas-mês.

EAF = produto do cálculo dos 15 fatores de ajuste de esforço.

Tamanho = número de instruções entregues (em unidades de milhares de linhas de código – KLOC).

03

1.2- MODO EMBUTIDO

Os projetos possuíam as seguintes características:

- Uso de processos rígidos.
- Existência de restrições relacionadas à complexidade, confiabilidade e execução *realtime*.
- Elevado nível de controle das funcionalidades, requisitos de qualidade, custos e cronogramas.
- As mudanças devem ser aprovadas por vários *stakeholders*.
- Projetos grandes.

A fórmula original para o cálculo de esforço e tempo é definida da seguinte forma:

$$\text{Esforço} = 3.0 \text{ EAF (Tamanho)}^{1.12}$$

$$\text{Tempo (em meses)} = 2.5 \text{ (Esforço)}^{0.35}$$

Onde:

Esforço = número de pessoas-mês.

EAF = produto do cálculo dos 15 fatores de ajuste de esforço.

Tamanho = número de instruções entregues (em unidades de milhares de linhas de código – KLOC).

04

1.3- MODO SEMIDESTACADO

Os projetos possuíam as seguintes características:

- Considerados um “meio termo” entre os modos “orgânico” e “embutido”.
- Equipes de desenvolvimento mistas (pessoas experientes e inexperientes).
- Requisitos funcionais e não funcionais de dois tipos (rígidos e flexíveis).

A fórmula original para o cálculo de esforço e tempo é definida da seguinte forma:

$$\text{Esforço} = 2.8 \text{ EAF (Tamanho)}^{1.2}$$

$$\text{Tempo (em meses)} = 2.5 \text{ (Esforço)}^{0.32}$$

Onde:

Esforço = número de pessoas-mês.

EAF = produto do cálculo dos 15 fatores de ajuste de esforço.

Tamanho = número de instruções entregues (em unidades de milhares de linhas de código – KLOC).

05

1.4- EAF (EFFORT ADJUSTMENT FACTOR)

O EAF (*Effort Adjustment Factor*) – Fator de Ajuste de Esforço – representa a combinação de múltiplos parâmetros, que permitem que o projeto seja caracterizado e normalizado com relação aos projetos da base de dados COCOMO.

Cada parâmetro é avaliado conforme os seguintes **critérios**:

- Muito baixo
- Baixo
- Nominal
- Alto
- Muito alto

O efeito da configuração de cada parâmetro é um multiplicador entre os valores 0.5 e 1.5. O produto dos 15 efeitos é utilizado como o coeficiente EAF da equação. A tabela abaixo apresenta os parâmetros:

IDENTIFICADOR	EAF (Fator de ajuste do esforço)	RANGE DO PARÂMETRO	IMPACTO POTENCIAL
RELY	<i>Confiabilidade requerida</i>	<i>0,75 – 1,40</i>	<i>1,87</i>
DATA	<i>Tamanho da base de dados</i>	<i>0,94 – 1,16</i>	<i>1,23</i>
CPLX	<i>Complexidade do produto</i>	<i>0,70 – 1,65</i>	<i>2,36</i>
TIME	<i>Restrições de tempo de execução</i>	<i>1,0 – 1,66</i>	<i>1,66</i>
STOR	<i>Restrição de uso de memória</i>	<i>1,0 – 1,56</i>	<i>1,56</i>
VIRT	<i>Mudanças do ambiente de Software</i>	<i>0,87 – 1,30</i>	<i>1,49</i>
TURN	<i>Tempo de resposta</i>	<i>0,87 – 1,15</i>	<i>1,32</i>
ACAP	<i>Capacidade dos analistas</i>	<i>1,46 – 0,71</i>	<i>2,06</i>
AEXP	<i>Experiência na aplicação</i>	<i>1,29 – 0,82</i>	<i>1,57</i>

PCAP	<i>Capacidade dos programadores</i>	<i>1,42 – 0,70</i>	<i>2,03</i>
VEXP	<i>Experiência no ambiente de hardware</i>	<i>1,21 – 0,90</i>	<i>1,34</i>
LEXP	<i>Experiência na linguagem de programação</i>	<i>1,14 – 0,95</i>	<i>1,20</i>
MODP	<i>Uso de práticas modernas de programação</i>	<i>1,24 – 0,82</i>	<i>1,51</i>
TOOL	<i>Uso de ferramentas de Software</i>	<i>1,24 – 0,83</i>	<i>1,49</i>
SCED	<i>Prazo requerido para o desenvolvimento</i>	<i>1,23 – 1,10</i>	<i>1,23</i>

Fonte: [ROYCE] Software Project Management – A Unified Framework

06

1.5- O CICLO DE VIDA

O ciclo de vida do COCOMO prevê cinco fases básicas:



A distribuição de esforço e tempo (em percentuais) no ciclo de vida é recomendada pelo COCOMO conforme demonstra a tabela abaixo:

ATIVIDADE	ESFORÇO (%)	TEMPO (%)
Planejamento e requisitos	(+8%)	(+36)
Design de produto	18	36
Design detalhado	25	18
Codificação e testes unitários	26	18
Integração e testes	31	28

Fonte: [ROYCE] Software Project Management – A Unified Framework

07

2 - COCOMO II

O “projeto COCOMO II” é uma ação inicialmente realizada pelo *USC Center for Software Engineering*, com os suportes financeiro e técnico de uma série de empresas, entre elas: AT&T, Bell Labs, Hewlett-Packard, Motorola, Rational, SEI.

Os **objetivos** deste projeto foram:

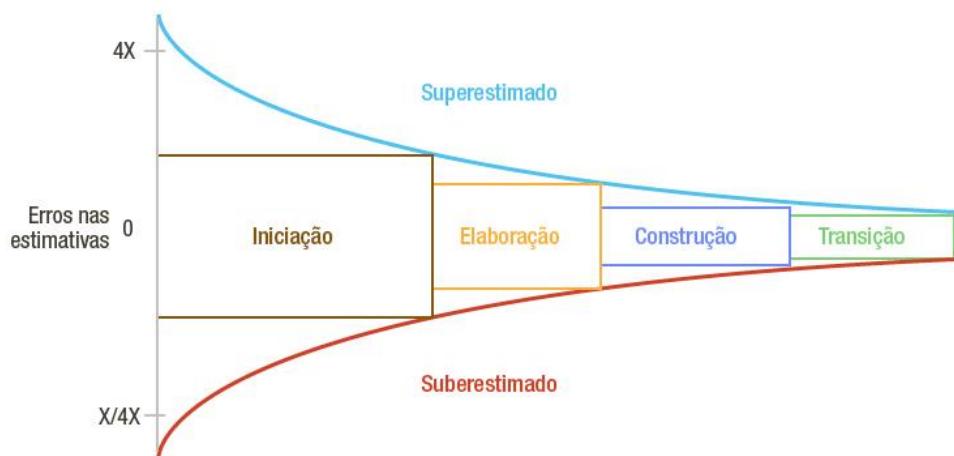
- Desenvolver um modelo de estimativa de custo e cronograma para as práticas de ciclo de vida de projetos dos anos de 1990 e 2000.
- Desenvolver uma base de dados de custos em desenvolvimento de *Software* e uma ferramenta de suporte à melhoria do modelo de custos.
- Prover um *framework* analítico e quantitativo para avaliação de tecnologias e seu impacto econômico.

O modelo COCOMO II pode ser utilizado como instrumento de apoio em situações que envolvam o processo de tomada de decisão, como por exemplo:

- Na realização de investimentos ou outras decisões de ordem financeira que envolvam desenvolvimento de *Software*.
- Na definição do orçamento e do cronograma de um projeto.
- Em negociações que envolvam custo, cronograma e fatores de qualidade.
- Em decisões que envolvam riscos relacionados a cronograma e custo.
- Na definição de estratégias de investimentos em melhorias organizacionais, como melhoria da capacidade da organização via reuso, ferramentas, entre outros.

08

À medida que o projeto avança no tempo, mais assertivas se tornam as estimativas. A figura abaixo apresenta um exemplo que ilustra este comportamento considerando o ciclo de vida iterativo de desenvolvimento de *Software*:



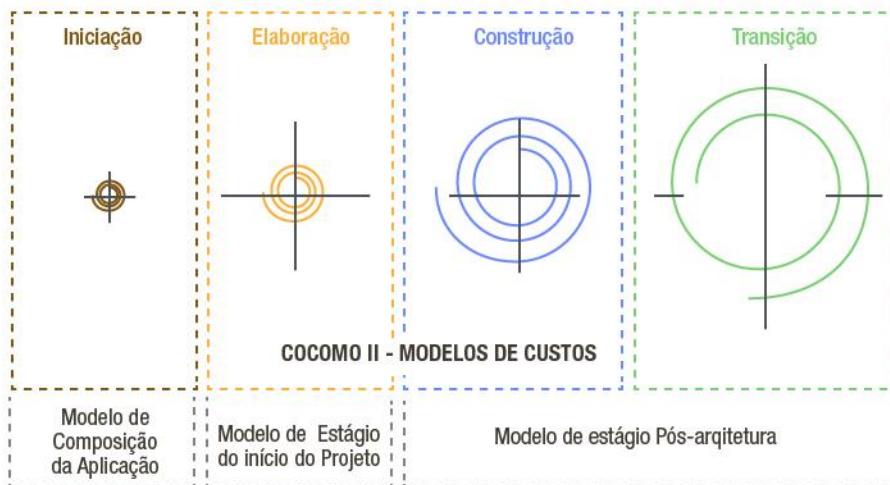
No início de um projeto de desenvolvimento de *Software* existem informações pouco detalhadas e precisas; com base nestas informações as estimativas são elaboradas – e são afetadas pelo nível de incertezas no projeto neste ponto do ciclo de vida. À medida que o projeto avança, riscos são detalhados, requisitos se tornam mais precisos, bem como necessidades e restrições técnicas. Por consequência, as estimativas se tornam mais próximas da realidade.

09

3 - MODELOS PARA ESTIMATIVAS DE CUSTOS COCOMO II

O COCOMO II define três modelos diferentes para a estimativa de custos, onde cada modelo corresponde ao **nível de fidelidade e incerteza** apropriado para determinada fase do ciclo de vida.

A figura exemplifica esta definição, tomando por base o ciclo iterativo de desenvolvimento de *Software*:



Os modelos são:

- **Modelo de composição da aplicação;**

- **Modelo de estágio de início do projeto;**
- **Modelo de estágio pós-arquitetura.**

Veja a descrição de cada um deles a seguir.

10

3.1- MODELO DE COMPOSIÇÃO DA APLICAÇÃO

O Modelo de Composição da Aplicação é utilizado durante os primeiros estágios da engenharia de *Software*.

Saiba+ sobre este modelo.

3.2- MODELO DE ESTÁGIO DE INÍCIO DO PROJETO

O Modelo de Estágio de Início de Projeto é utilizado após os requisitos alcançarem um bom nível de estabilidade e quando a arquitetura básica do *Software* estiver definida. **Saiba+**

3.3- MODELO DE ESTÁGIO PÓS-ARQUITETURA

O Modelo de Estágio Pós-arquitetura é utilizado durante a construção do *Software*. **Saiba+**

Os modelos de estimativa COCOMO II utilizam a informação do **tamanho do Software**. São três as opções disponíveis:

- Pontos de função.
- Linhas de código-fonte.
- Pontos de objeto.

Vejamos a seguir.

Saiba+ (modelo de composição da aplicação)

No momento do uso deste modelo:

- Os requisitos são definidos em alto nível.
- A arquitetura existe em nível conceitual.
- As estimativas são definidas em alto nível.

Saiba+ (modelo de estágio de início do projeto)

No momento do uso deste modelo:

- O projeto é considerado “bem entendido” pelos envolvidos.
- As estimativas possuem um nível moderado de fidelidade.

Saiba+ (modelo de estágio pós-arquitetura)

No momento do uso deste modelo:

- O projeto é bem caracterizado.
- As estimativas possuem um elevado nível de fidelidade.
- Há uma *baseline* estável dos requisitos.
- Há uma *baseline* arquitetural estável.

11

4 - PONTO DE OBJETO

A métrica de Pontos de Objeto é uma medida de tamanho funcional de um *Software* baseada na contagem dos seguintes elementos:

- Telas que compõem a interface do usuário.
- Relatórios.
- Componentes necessários à construção do *Software*.

O procedimento de contagem dos pontos de objeto é composto pelas atividades ilustradas pela figura abaixo:

**Realizar a contagem dos objetos**

1. REALIZAR A CONTAGEM DOS OBJETOS

O objetivo desta atividade é contagem do número de telas, relatórios e componentes.

Classificar as instâncias de objetos**2. CLASSIFICAR AS INSTÂNCIAS DE OBJETOS**

O objetivo desta atividade é classificar cada elemento identificado (instância de objetos), de acordo com a complexidade.

Realizar a ponderação dos tipos de objetos**3. REALIZAR A PONDERAÇÃO DOS TIPOS DE OBJETOS**

O objetivo desta atividade é ponderar os objetos identificados, com base na tabela abaixo:

TIPO DE OBJETO	PESO PARA A COMPLEXIDADE		
	SIMPLES	MÉDIO	DIFÍCIL
Tela	1	2	3
Relatório	2	5	8
Componente			10

Calcular os Pontos de Objeto**4. CALCULAR OS PONTOS DE OBJETO**

O objetivo desta atividade é realizar o cálculo dos Pontos de Objeto, que é realizada da seguinte forma:

- Para cada instância de objeto identificada, multiplicar a quantidade pelo peso definido.
- Somar os valores obtidos.

Quando há reuso de código, a contagem de pontos de objeto deve ser ajustada conforme fórmula abaixo:

$$NOP = (\text{pontos de objeto}) * [(100 - \% \text{reuso})/100]$$

12

4.1- ESTIMATIVA DE ESFORÇO BASEADA NOS PONTOS DE OBJETO

A partir dos NOP – Número de Pontos de Objeto é possível obter a taxa de produtividade. A fórmula abaixo exemplifica a forma de cálculo da produtividade:

$$\text{PRODUTIVIDADE} = NOP / \text{pessoa-mês}$$

A estimativa de esforço é obtida a partir da seguinte fórmula:

$$\text{ESFORÇO ESTIMADO} = NOP / \text{PRODUTIVIDADE}$$

De acordo com PRESSMAN (2011), modelos COCOMO II mais avançados fazem uso de fatores de escala, custos e procedimentos de ajustes. Um exemplo da utilização de tais fatores será apresentado a seguir.

5 - EXEMPLO DE FERRAMENTA PARA UTILIZAÇÃO DO COCOMO II

Como exemplo de ferramenta para utilização do modelo COCOMO II será utilizada uma ferramenta baseada na web, disponibilizada pelo Centro de Engenharia de *Software* da Universidade da Califórnia (*USC Center for Software Engineering*). Veja aqui o site para acesso.

Existem várias ferramentas de livre acesso para utilização dos modelos COCOMO. Para fins de exemplo, será utilizada a ferramenta baseada na web COCOMO® II, acessível através [deste link](#).

A figura abaixo apresenta a tela da ferramenta. Em seguida é dado o passo a passo para cálculo das estimativas com o uso da ferramenta:

OBS.: Para melhor visualização, ver material online.

Veja aqui

http://csse.usc.edu/csse/research/COCOMOII/cocomo_main.html

deste link

<http://csse.usc.edu/tools/COCOMOII.php>

PASSO 1: INFORMAR O TAMANHO DO SOFTWARE

Para fins de exemplo, considerar o tamanho de 250 Pontos de Função e linguagem de programação JAVA, conforme ilustra a figura abaixo:

Software Size Sizing Method **Function Points** ↗
 Unadjusted Function Points **250** Language **Java** ↗
 ↗ **Selecionar "Function Points" (Pontos de Função)**
 ↗ **Selecionar "JAVA"**
 ↗ **Preencher com o valor "250"**

15

PASSO 2: INFORMAR OS VALORES PARA OS FATORES DIRECIONADORES DA ESCALA DO SOFTWARE

A utilização dos fatores de escala representa uma evolução com relação ao modelo original COCOMO. São cinco os fatores de escala definidos:

1. **PREC** - Precedência (*Precedendedness*);
2. **FLEX** - Flexibilidade no desenvolvimento (*Development Flexibility - FLEX*);
3. **RESL** - Arquitetura / Resolução de Riscos (*Architecture / Risk Resolution*);
4. **TEAM** – Fator de Escala de Coesão do Time (*Team Cohesion*);
5. **PMAT** – Fator de Escala de Maturidade do Processo (*Process Maturity*).

A classificação dos direcionadores da escala de *software* deve ser realizada conforme exemplifica a figura abaixo:

Software Scale Drivers	Architecture / Risk Resolution	Process Maturity
Precedendedness	Low	High
Development Flexibility	Nominal	Team Cohesion
		Low

PREC - Precedência (*Precedendedness*)

Refere-se ao nível de similaridade do produto a ser construído com produtos construídos em projetos anteriores.

FLEX - Flexibilidade no desenvolvimento (*Development Flexibility - FLEX*)

Refere-se ao nível de conformidade requerido do *Software* com requisitos.

RESL - Arquitetura / Resolução de Riscos (*Architecture / Risk Resolution*)

Representa a combinação de dois fatores utilizados no modelo Ada COCOMO que envolvem características relacionadas ao estabelecimento da arquitetura e ao gerenciamento de riscos.

TEAM – Fator de Escala de Coesão do Time (Team Cohesion)

Refere-se aos fatores de “turbulência” do projeto e entropia devido a dificuldades na harmonização das partes interessadas (usuários, clientes, desenvolvedores, entre outros).

PMAT – Fator de Escala de Maturidade do Processo (Process Maturity)

O procedimento para definição de PMAT é organizado conforme o CMM – *Software Engineering Institute's Capability Maturity Model*.

16

PASSO 3: INFORMAR OS VALORES PARA OS DIRECIONADORES DE CUSTO

Os direcionadores de custo são utilizados por ambos os modelos COCOMO de Estágio de Início de Projeto e de Estágio Pós-arquitetura.

A diferença está na quantidade de direcionadores utilizados no escopo de cada modelo:

- no modelo de Estágio de Início de Projeto são utilizados **sete** direcionadores de custo,
- no modelo Pós-arquitetura são utilizados **dezessete**, conforme apresenta a tabela abaixo:

DIRECIONADORES DE CUSTO	
Estágio de Início de Projeto	Pós-arquitetura
PERS	ACAP, PCAP, PCON
RCPX	RELY, DATA, CPLX, DOCU
RUSE	RUSE
PDIF	TIME, STOR, PVOL
PREX	APEX, PLEX, LTEX
FCIL	TOOL, SITE
SCED	SCED

17

Os direcionadores de custos são agrupados em quatro categorias:

Categoria PRODUTO	Categoria PLATAFORMA	Categoria PESSOAL	Categoria PROJETO
<ul style="list-style-type: none"> •RELY •DATA •CPLX •RUSE •DOCU 	<ul style="list-style-type: none"> •TIME •WTOR •PVOL 	<ul style="list-style-type: none"> •ACAP •PCAP •PCON •AEXP •PEXP •LTEX 	<ul style="list-style-type: none"> •TOOL •SITE •SCED

A categoria **PRODUTO** é composta pelos seguintes direcionadores de custo:

1. RELY – Confiabilidade requerida do *Software*.
2. DATA – Tamanho da base de dados.
3. CPLX – Complexidade do produto.
4. RUSE – Reusabilidade requerida.
5. DOCU – Documentação adequada às necessidades do projeto

18

A categoria **PLATAFORMA** é composta pelos seguintes direcionadores de custo:

1. TIME – Restrição de tempo de execução.
2. STOR – Restrição de armazenamento principal.
3. PVOL – Volatilidade da plataforma.

A categoria **PESSOAL** é composta pelos seguintes direcionadores de custo:

1. ACAP – Capacidade do analista.
2. PCAP – Capacidade dos programadores.
3. PCON – Rotatividade de pessoal.
4. AEXP – Experiência com a aplicação.
5. PEXP – Experiência com a plataforma de desenvolvimento.
6. LTEX – Experiência com linguagens e ferramentas de desenvolvimento.

A categoria **PROJETO** é composta pelos seguintes direcionadores de custo:

1. TOOL – Uso de ferramenta de *Software*.
2. SITE – Desenvolvimento distribuído ou multisite.
3. SCED – Cronograma de desenvolvimento requerido.

19

A classificação dos direcionadores de custos deve ser realizada conforme exemplifica a figura abaixo:

Software Cost Drivers		Personnel		Platform	
Product		Analyst Capability	Nominal	Time Constraint	Nominal
Required Software Reliability	Nominal	Programmer Capability	High	Storage Constraint	Nominal
Data Base Size	Nominal	Personnel Continuity	Nominal	Platform Volatility	Nominal
Product Complexity	Nominal	Application Experience	High	Project	
Developed for Reusability	High	Platform Experience	High	Use of Software Tools	High
Documentation Match to Lifecycle Needs	High	Language and Toolset Experience	High	Multisite Development	Nominal
				Required Development Schedule	Nominal

20

PASSO 4: CALCULAR E VERIFICAR O RESULTADO

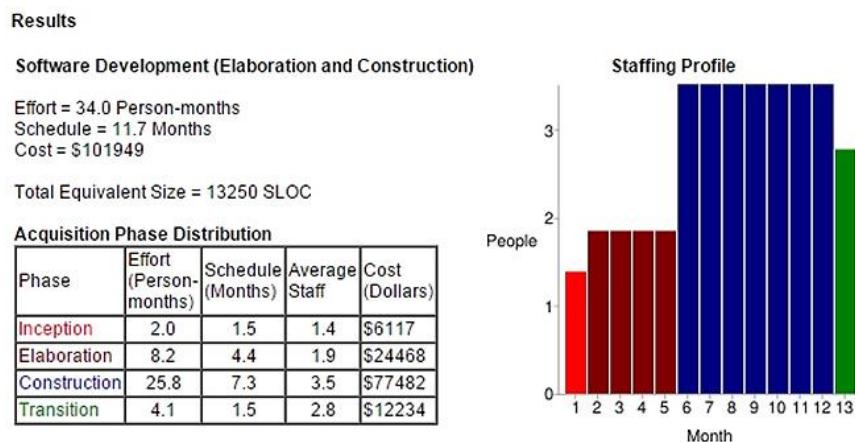
Esta ferramenta permite que a informação de custo seja fornecida (no caso, em dólares). Informar o valor \$ 3.000 e clicar em “Calculate”, conforme figura abaixo:

Software Labor Rates

Cost per Person-Month (Dollars)

Calculate

A figura abaixo apresenta o primeiro grupo dos resultados:



O esforço estimado para o projeto é de 34 pessoas-mês.

O tempo estimado para o projeto é de 11,7 meses.

O custo estimado é de \$ 101.949.

O tamanho equivalente em **linhas de código-fonte** é de 13.250 SLOC.

O gráfico apresentado demonstra que a alocação de pessoal cresce no decorrer dos meses e tem o seu ápice no período que compreende do 6º ao 12º mês (período que compreende a fase de construção do projeto). A partir da fase de transição do projeto, o nível de alocação de pessoal é reduzido.

A tabela apresentada (*Acquisition Phase Distribution*) apresenta o esforço, o tempo, a taxa de alocação de pessoal e o custo distribuídos por fase do projeto.

21

A figura abaixo apresenta o segundo grupo dos resultados:

Software Effort Distribution for RUP/MBASE (Person-Months)				
Phase/Activity	Inception	Elaboration	Construction	Transition
Management	0.3	1.0	2.6	0.6
Environment/CM	0.2	0.7	1.3	0.2
Requirements	0.8	1.5	2.1	0.2
Design	0.4	2.9	4.1	0.2
Implementation	0.2	1.1	8.8	0.8
Assessment	0.2	0.8	6.2	1.0
Deployment	0.1	0.2	0.8	1.2

A tabela apresenta a distribuição de esforço (em pessoas-mês) por área de conhecimento da Engenharia de *Software*, bem como das áreas de suporte à mesma, por fase do ciclo de vida de desenvolvimento do produto. Algumas análises podem ser feitas, por exemplo:

- No início do projeto (Fase de Iniciação ou *Inception*) a distribuição de esforço é menor se comparada à Fase de Elaboração (*Elaboration*). Este comportamento é normal diante do cenário de que é na primeira fase que ocorrem os primeiros entendimentos sobre o produto a ser desenvolvido, sobre uma possível arquitetura a ser utilizada.
- Na Fase de Construção (*Construction*) o esforço nas atividades de construção (*implementation*) é naturalmente maior, uma vez que a arquitetura está definida e os requisitos em um nível de detalhamento maior.
- Na Fase de Transição (*Transition*) o esforço nas atividades de construção é consideravelmente reduzido – tipicamente espera-se que durante esta fase a equipe de codificação esteja concentrada na correção de eventuais relatos de homologação.

22

RESUMO

O modelo COCOMO – Constructive Cost Model – surgiu com o objetivo de viabilizar a elaboração de estimativas de esforço, prazo, custo e tamanho de equipes em projetos de desenvolvimento de

sistemas. Sua primeira versão foi apresentada em 1981 por seu criador Barry Boehm. Desde então, conforme as mudanças na indústria de *Software*, variações deste modelo surgiram e foram melhoradas no decorrer do tempo.

O modelo COCOMO II surgiu em 1994 e, assim como aconteceu com o modelo anterior, passou por um processo de evolução que deu origem a três modelos:

- Modelo de Composição da Aplicação.
- Modelo de Estágio de Início de Projeto.
- Modelo de Estágio Pós-arquitetura.

Cada modelo utiliza uma série de elementos, critérios e fórmulas de cálculo definidos no escopo dos modelos apresentados.

Atualmente existem diversas ferramentas que auxiliam a aplicação do conjunto de modelos COCOMO para as estimativas de esforço, prazo, custo e tamanho de equipes.

UNIDADE 2 – TÉCNICAS PARA MEDIÇÃO DE SOFTWARE

MÓDULO 4 – TÉCNICAS DE DECOMPOSIÇÃO

01

1 - ESTIMATIVA BASEADA NO PROBLEMA

A máxima “Dividir para conquistar” refletida nas estratégias de alguns dos grandes imperadores que fizeram parte da história da humanidade também se aplica à Engenharia de *Software* quando se trata da decomposição do problema a ser entendido no contexto da análise de requisitos de *software*. A decomposição se faz em duas áreas significativas:

- Na funcionalidade a ser entregue.
- No processo a ser utilizado para a construção do *software*.

As funções do *software* refletidas e definidas no escopo do produto são a principal entrada para a elaboração das estimativas.



De acordo com PRESMANN (2011), o dimensionamento é o primeiro grande desafio do responsável pelo planejamento do projeto. Por meio da decomposição de um projeto a estimativa de custo e esforço pode ser realizada passo a passo.

As atividades previstas para realizar as **estimativas baseadas no problema** são:

- **A divisão do software em partes menores (funções).**
- **A estimativa das funções utilizando as técnicas: número de linhas de código (LOC) e número de pontos de função (PF).**

Linhas de Código (LOC) é uma medida de tamanho de *software* baseada na contagem do número de linhas código-fonte de um programa de *software*.

Pontos de Função (PF) é uma medida de tamanho funcional para projetos de desenvolvimento e de melhoria de *software* baseada em características do *software* (entradas, saídas, arquivos de dados, consultas, interfaces externas) e fatores de complexidade.

02

As técnicas de estimativas em LOC e PF possuem as seguintes **características comuns**:

- Iniciam a partir do escopo delimitado do *software* a ser construído.
- O escopo é decomposto em funções que podem ser estimadas individualmente.
- A estimativa é realizada para cada função obtida na decomposição.
- Para obtenção do custo ou esforço para a função, métricas de produtividade podem ser aplicadas.
- As estimativas individuais das funções são combinadas para que se obtenha a estimativa geral de todo o projeto.

As técnicas de estimativas em LOC e PF **diferem** nos seguintes aspectos:

- No nível de detalhe requerido para a decomposição: quando LOC é utilizada, a decomposição é essencial e bem detalhada.
- No alvo do particionamento.

Para a estimativa em LOC, a decomposição é essencial e mais detalhada. Quanto maior o nível de particionamento, mais precisão pode ser obtida neste tipo de estimativa.

Para a estimativa em PF, o foco se concentra em características como entradas, saídas, arquivos de dados, consultas e interfaces externas.

De acordo com PRESSMAN (2011), o “valor esperado” para o tamanho pode ser calculado como uma média ponderada das estimativas otimista (S_{opt}), mais provável (S_m) e pessimista (S_{spess}):

$$\text{LOC estimado} = S_{\text{opt}} + 4 \cdot S_{\text{m}} + S_{\text{spess}} / 6$$

03

1.1. Exemplo de estimativa baseada em LOC

Dado um sistema hipotético, seguem as funções identificadas:

- **Recursos de controle e interface.**
- **Análise geográfica bidimensional.**
- **Análise geográfica tridimensional.**
- **Gestão de base de dados.**

Para cada função deve ser desenvolvido um intervalo de estimativas LOC. Para fins de exemplo, utilizaremos a função “Análise geográfica tridimensional”. Para esta função foi desenvolvido o seguinte intervalo de estimativas LOC:

- Otimista (S_{opt}) = 4.600 LOC
- Mais provável (S_m) = 6.900 LOC
- Pessimista (S_{spess}) = 8.600 LOC

Aplicando a equação, temos:

$$S = 4.600 + (4 * 6.900) + 8.600 / 6$$

$$S = 6.800 \text{ LOC.}$$

FUNÇÃO	S _{opt}	S _m	S _{spess}	ESTIMATIVA EM LOC

Análise geográfica tridimensional	4600	6900	8600	6800
--	------	------	------	-------------

Considerando uma produtividade média de 620 LOC/ pm (pessoa-mês), a estimativa de esforço é de 54 pessoas/mês.

Considerando um custo de R\$ 13 por linha de código, a estimativa de custo da função “Análise geográfica tridimensional” é R\$ 88.400,00.

A estimativa deve ser repetida para cada função prevista. O valor total de LOC estimado será dado pela soma das estimativas em LOC de todas as funções envolvidas. O custo total do projeto será dado pela equação custo por linha de código * valor LOC estimado total.

04

1.2. Exemplo de estimativa baseada em PF

Dado um novo sistema hipotético solicitado pelo cliente, a contagem de pontos de função realizada apresenta como resultado a tabela abaixo:

TIPO DE FUNÇÃO	COMPLEXIDADE FUNCIONAL	TOTAL COMPLEXIDADE	TOTAL TIPO FUNÇÃO
ALI	3 BAIXA x 7 = 21	21	
	0 MÉDIA x 10 = 0	0	
	0 ALTA x 15 = 0	0	
AIE	2 BAIXA x 5 = 10	10	
	0 MÉDIA x 7 = 0	0	
	0 ALTA x 10 = 0	0	
EE	4 BAIXA x 3 = 12	12	26
	2 MÉDIA x 4 = 8	8	
	1 ALTA x 6 = 6	6	
SE	2 BAIXA x 4 = 8	8	
	0 MÉDIA x 5 = 0	0	

0	ALTA	x 7 =	0
CE	2	BAIXA	x 3 = 6
2	MÉDIA	x 4 =	8
0	ALTA	X 6 =	0
TOTAL DE PONTOS DE FUNÇÃO BRUTOS			79

05

Os fatores de complexidade foram analisados e pontuados:

FATOR DE AJUSTE	VALOR
Comunicação de Dados	4
Processamento Distribuído	2
Performance	0
Utilização do Equipamento	4
Volume de Transações	3
Entrada de dados on-line	4
Usabilidade	5
Atualização on-line	3
Processamento complexo	5
Reutilização de código	5
Facilidade de implantação	4
Facilidade operacional	3
Múltiplos locais	5
Facilidade de mudanças	5
TOTAL	52

O fator de ajuste é calculado conforme a fórmula abaixo:

$$\mathbf{FA = (NI * 0,01) + 0,65}$$

$$\mathbf{FA = (52 * 0,01) + 0,65}$$

$$\mathbf{FA = 1,17}$$

A fórmula para cálculo dos pontos de função para um projeto novo é:

$$\mathbf{PF_DESENV = PF_NA * FA}$$

$$\mathbf{PF_DESENV = 79 * 1,17}$$

$$\mathbf{PF_DESENV = 92,43}$$

Considerando uma produtividade média de 7 PF/ pm (pessoas-mês), a estimativa de esforço é de 13,5 pessoas/mês.

Considerando um custo dos recursos alocados de R\$ 8 mil por mês, o custo por PF é R\$ 1.142,85. O custo total estimado do projeto é de R\$ 105.633,62.

06

2 - ESTIMATIVA BASEADA NO PROCESSO

A técnica de estimativa baseada no processo tem por base o processo de desenvolvimento a ser utilizado na construção do sistema.

As seguintes atividades devem ser realizadas:

- **Identificação do processo de desenvolvimento a ser utilizado e das atividades estruturais a serem executadas no projeto.**
- **A identificação das funções a partir do escopo do produto de *software*.**
- **Para cada função de *software* identificada, estimar o esforço necessário para execução, por atividade estrutural.**

A tabela abaixo apresenta uma estrutura exemplo para inclusão dos dados das estimativas:

ATIVIDADES DO PROCESSO									
FUNÇÕES	CC	PLANEJAMENTO	ANÁLISE DE RISCO	ENGENHARIA		LIBERAÇÃO DA VERSÃO		AC	TOTais
				ANÁLISE	DESIGN	CONSTRUÇÃO	TESTES		
<i>Função 1</i>									
<i>Função 2</i>									
<i>Função n</i>									

CC = comunicação com o cliente e AC = avaliação pelo cliente

07

2.1 Exemplo de estimativa baseada em PF

Dado um novo sistema hipotético solicitado pelo cliente, a contagem baseada em processo resultou na estimativa exemplificada no quadro abaixo:

Atividade	CC	Planejamento	Análise de risco	Engenharia		Liberação da versão		AC	Totais
Tarefa→				Análises	Design	Code	Test		
Função ↓									
UICF				0,50	2,50	0,40	5,00	n/a	8,40
2DGA				0,75	4,00	0,60	2,00	n/a	7,35
3DGA				0,50	4,00	1,00	3,00	n/a	8,50
CGDF				0,50	3,00	1,00	1,50	n/a	6,00
DBM				0,50	3,00	0,75	1,50	n/a	5,75
PCF				0,25	2,00	0,50	1,50	n/a	5,00
Totais	0,25	0,25	0,25	3,50	20,50	4,50	16,50		46,00
% da mão de obra	1%	1%	1%	8%	45%	10%	36%		

CC = comunicação com o cliente AC = avaliação pelo cliente

Para cada função identificada, foi realizada a estimativa de esforço. Os valores estimados são totalizados logo abaixo das estimativas individuais.

Considerando o custo médio da mão de obra por mês como R\$ 8 mil, o custo total estimado para o projeto é de **R\$ 368.000,00**.

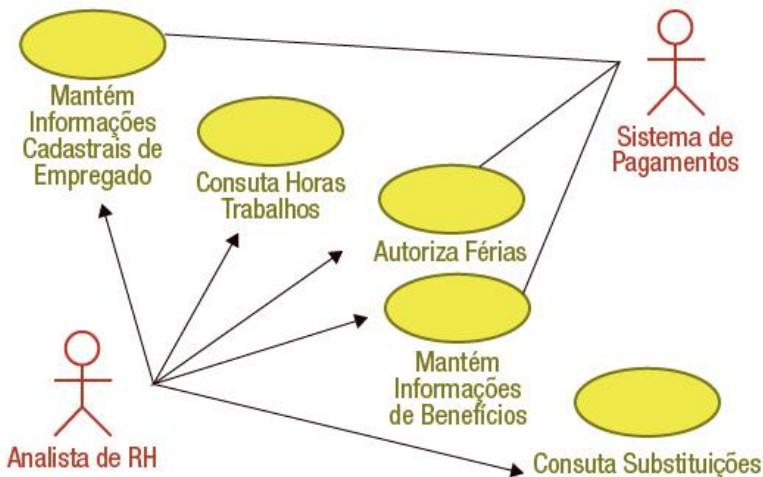
08

3 - ESTIMATIVA BASEADA EM CASO DE USO

A UML (*Unified Modeling Language*) permite que requisitos funcionais sejam organizados em unidades chamadas de casos de uso. Os casos de uso definem uma série de instâncias, onde cada uma estabelece uma sequência de ações a serem realizadas pelo sistema. Tais ações provêm um resultado observável para um ator.

Um ator define uma série de papéis que os usuários do sistema podem assumir quando interagem com o sistema. Um ator também pode representar um *hardware* ou um sistema externo.

O diagrama de casos de uso é a representação visual de todas as funções de um sistema. A figura abaixo ilustra esta representação:



09

3.1 Estimativa que relaciona casos de uso e LOC

De acordo com SMITH (1999), em seu artigo de título “*A Estimativa de Esforço Baseada em Casos de Uso*” existem **características** na Análise de Casos de Uso que dificultam o processo de contagem. São elas:

- Casos de uso são descritos por meio de muitos estilos e formatos diferentes (não há uma padronização).
- Casos de uso representam uma visão externa do software (visão sob o ponto de vista do usuário) e podem, portanto, ser escritos em diversos níveis de abstração.
- Casos de uso não tratam da complexidade das funções e características que são descritas.
- Casos de uso podem descrever comportamentos complexos que envolvem muitas funções e características.

Os casos de uso podem ser empregados para estimativa, mas somente se forem considerados no contexto da “hierarquia estrutural” em que são usados para escrever.

PRESSMAN (2011) apresenta, para fins de exemplo, a fórmula do cálculo de uma estimativa aproximada em número de LOC (*Lines of Code*) com base nos casos de uso identificados:

$$\text{LOC estimado} = N * \text{LOC média} + [(Sa/Sh - 1) + (Pa/Ph - 1)] * \text{LOC ajustado}$$

Onde:

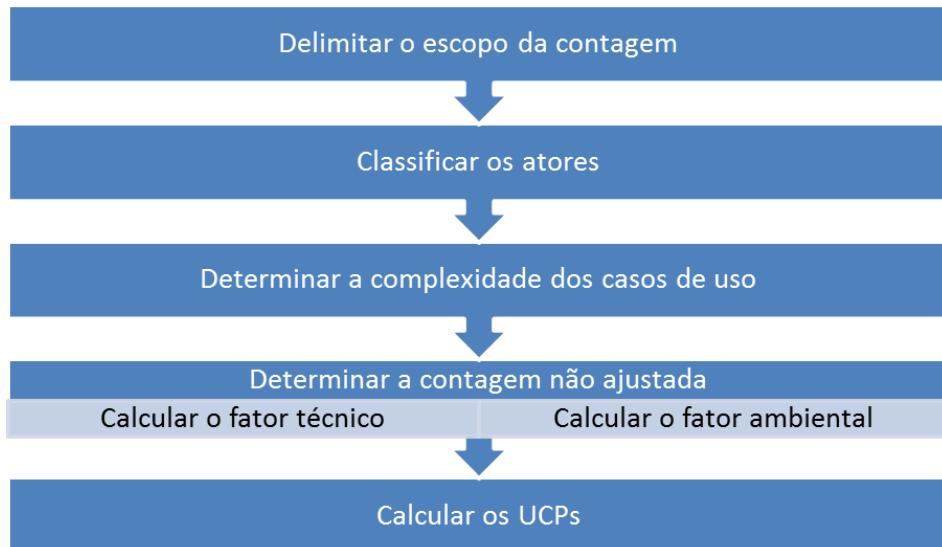
- N = quantidade de casos de uso
- LOC média = média histórica de LOC por caso de uso para este tipo de sistema
- LOC ajustado = representa um ajuste baseado em *n* por cento de LOC média em que *n* é definido localmente e representa a diferença entre esse projeto e a “média” de outros projetos
- Sa = quantidade de cenários por caso de uso
- Sh = quantidade média de cenários por caso de uso para este tipo de sistema
- Pa = número de páginas por caso de uso
- Ph = quantidade média de páginas por caso de uso para este tipo de sistema

10

3.2 Estimativa de pontos de caso de uso

O método Pontos de Caso de Uso – *Use-Case Points* – é uma abordagem para estimativa das atividades de desenvolvimento de *software*, que tem por base o Modelo de Casos de Uso. O tamanho do sistema é calculado a partir dos pontos de casos de uso não ajustados e do fator de complexidade técnica, obtido a partir das características técnicas do sistema.

A figura abaixo apresenta o procedimento de contagem em pontos de caso de uso:



11

3.2.1 Delimitar o escopo da contagem

Esta atividade tem como objetivos:

- **Identificar que funcionalidades deverão fazer parte do escopo da contagem.**
- **Identificar que casos de uso deverão fazer parte do escopo da contagem.**
- **Identificar se existem fronteiras com outros sistemas que devem fazer parte do escopo da contagem.**

3.2.2 Classificar os atores

Esta atividade tem como objetivo a classificação dos atores conforme a complexidade:

COMPLEXIDADE DO ATOR	DESCRIÇÃO	PESO
Simples	Outro sistema acessa através de API (<i>application programming interface</i>) bem definida.	1
Médio	Outro sistema que interage através de um protocolo de comunicação (exemplo: TCP/IP e FTP).	2
Complexo	Usuário que interage com uma interface gráfica (<i>stand-alone</i> ou <i>web</i>)	3

O peso total dos atores (UAW - *Unadjusted Actor Weight*) é calculado pela fórmula:

$$\text{UAW} = \text{Quantidade atores "Simples"} * 1 + \text{Quantidade atores "Médios"} * 2 + \text{Quantidade atores "Complexos"} * 3$$

12

3.2.3 Determinar a complexidade dos casos de uso

Esta atividade tem como objetivo determinar a complexidade dos casos de uso, conforme o número de transações que fazem parte do caso de uso a ser contado.

Transação pode ser conceituada como:

- Um conjunto de atividades atômicas, as quais são executadas completamente ou não.
- Um evento que ocorre entre um ator e o sistema.
- São passos dos fluxos de eventos de casos de uso que devem ser executados por completo ou a realização de um processamento complexo.

A tabela abaixo apresenta as regras para determinação da complexidade dos casos de uso:

CASO DE USO	DESCRIÇÃO	PESO
Simples	Possui até 3 transações (incluindo os passos alternativos) OU deve ser realizado com pelo menos 5 classes de análise.	5

Médio	Possui de 4 a 7 transações (incluindo os passos alternativos) OU deve ser realizado com 5 a 10 classes de análise.	10
Complexo	Possui mais de 7 transações (incluindo os passos alternativos) OU deve ser realizado com pelo menos 10 classes de análise.	15

O peso bruto dos casos de uso (UUCW - *Unadjusted Use Case Weight*) é calculado pela fórmula:

UUCW = Quantidade de casos de uso “Simples” * 5 + Quantidade de casos de uso “Médios” * 10 + Quantidade de casos de uso “Complexos” * 15

13

3.2.4 Determinar a contagem não ajustada

Esta atividade tem como objetivo a quantificação de outros fatores que afetam o tamanho a ser medido em Pontos de Caso de Uso.

O peso total não ajustado (*Unadjusted Use Case Points*) é calculado pelo somatório entre os pesos de atores (UAW) e casos de uso (UUCW), de acordo com a fórmula abaixo:

UUCP = UAW + UUCW

O cálculo do fator de ajuste é similar ao adotado na Contagem de Pontos de Função.

3.2.4.1 Cálculo do Fator Técnico

Para o cálculo do fator de complexidade técnica (TCF – *Technical Complexity Factor*), deve ser utilizada a tabela abaixo:

FATOR	DESCRIÇÃO	PESO
T1	Sistema distribuído	2
T2	Objetivos de performance	1
T3	Eficiência on-line	1

T4	Complexidade de processamento	1
T5	Código reusável em outras aplicações	1
T6	Facilidade de instalação	0,5
T7	Facilidade de uso	0,5
T8	Portabilidade	2
T9	Facilidade de alterações (<i>changeability</i>)	1
T10	Concorrência	1
T11	Segurança	1
T12	Acesso direto a terceiros	1
T13	Necessidade de facilidades especiais de treinamento para usuários	1

Para cada fator (T1 a T13) o nível de influência pode ser definido conforme os seguintes critérios:

0 – Não

1 – Pouco

2 – Razoável

3 – Médio

4 – Muito

5 – Elevado

O cálculo do TCF é realizado conforme fórmula abaixo:

$$\mathbf{TCF = 0.6 + (0.01 \times TFactor)}$$

Onde **TFactor** é obtido pelo somatório dos níveis de influência atribuídos a cada fator (T1 a T13) multiplicados pelo seu peso correspondente.

14

3.2.4.2 Cálculo do Fator Ambiental

Os fatores de ambiente (EF – *Environment Factor*) são aqueles relacionados aos requisitos associados ao processo de desenvolvimento.

A tabela abaixo apresenta os fatores de ambiente e o respectivo peso associado aos mesmos:

FATOR	DESCRÍÇÃO	PESO
A1	Familiaridade com o processo de desenvolvimento	1,5
A2	Experiência na aplicação	0,5
A3	Experiência em orientação à objetos	1,0
A4	Experiência do Líder de Projetos	0,5
A5	Motivação	1,0
A6	Estabilidade dos requisitos	2,0
A7	Membros da equipe com dedicação parcial	-1,0
A8	Dificuldade da linguagem de programação	-1,0

O nível de influência aponta o nível de disponibilidade de cada recurso no decorrer do projeto. Para cada fator (A1 a A8) o nível de influência pode ser definido conforme os seguintes critérios:

0 – Não

1 – Pouco

2 – Razoável

3 – Médio

4 – Muito

5 – Elevado

O cálculo do EF é realizado conforme fórmula abaixo:

$$EF = 1,4 + (-0,03 \times EFactor)$$

Onde o valor de **EFactor** é dado pela soma dos produtos entre o peso de cada fator (A1 a A8) e seu grau de influência atribuído, da mesma forma como no cálculo da variável **TFactor**.

15

3.2.5 Calcular os UCP (Use Case Points)

Esta atividade tem como objetivo realizar o cálculo dos Pontos de Caso de Uso (UCPs).

O cálculo do UCP é realizado conforme fórmula abaixo:

$$\text{UCP} = \text{UUCP} * \text{TCF} * \text{EF}$$

Onde:

- UUCP = Pontos de Caso de Uso não ajustados (*Unadjusted Use Case Points*)
- TCF = Fator de Complexidade Técnica (*Technical Complexity Factor*)
- EF = Fator Ambiental (*Environment Factor*)

16

3.3 Exemplo de estimativa baseada em pontos de casos de uso

Dado um novo sistema hipotético solicitado pelo cliente, a contagem baseada em pontos de caso de uso resultou nos seguintes dados:

Complexidade dos atores:

ATOR	PESO	Quantidade Identificada	Valor	UAW = 7
Simples	1	1	1	
Médio	2	0	0	
Complexo	3	2	6	

Complexidade dos casos de uso:

ATOR	PESO	Quantidade Identificada	Valor	UUCW = 55
Simples	5	8	40	

Médio	10	0	0	
Complexo	15	1	15	

Pontos de casos de uso não ajustados:

$$\text{UUCP} = \text{UAW} + \text{UUCW}$$

$$\text{UUCP} = 7 + 55$$

$$\text{UUCP} = 62$$

17

Cálculo do Fator de Complexidade Técnica:

FATOR	DESCRIÇÃO	PESO	VALOR ATRIBUÍDO	VALOR CALCULADO
T1	Sistema distribuído	2	2	4
T2	Objetivos de performance	1	5	5
T3	Eficiência on-line	1	1	1
T4	Complexidade de processamento	1	1	1
T5	Código reusável em outras aplicações	1	4	4
T6	Facilidade de instalação	0,5	2	1
T7	Facilidade de uso	0,5	2	1
T8	Portabilidade	2	1	2
T9	Facilidade de alterações (<i>changeability</i>)	1	1	1
T10	Concorrência	1	2	2
T11	Segurança	1	3	3
T12	Acesso direto a terceiros	1	0	0
T13	Necessidade de facilidades especiais de treinamento para usuários	1	0	0

			TFATOR =	25
--	--	--	-----------------	----

$$TCF = 0.6 + (0.01 \times TFactor)$$

$$TCF = 0,85$$

18

Cálculo do Fator Ambiental:

FATOR	DESCRIÇÃO	PESO	VALOR ATRIBUÍDO	VALOR CALCULADO
A1	Familiaridade com o processo de desenvolvimento	1,5	2	3
A2	Experiência na aplicação	0,5	3	1,5
A3	Experiência em orientação à objetos	1,0	4	4
A4	Experiência do Líder de Projetos	0,5	2	1
A5	Motivação	1,0	3	3
A6	Estabilidade dos requisitos	2,0	5	10
A7	Membros da equipe com dedicação parcial	-1,0	0	0
A8	Dificuldade da linguagem de programação	-1,0	0	0
			EFACTOR =	22,5

$$EF = 1.4 + (-0.03 \times EFactor)$$

$$EF = 0,725$$

Cálculo dos Use Case Points (UCP):

$$UCP = UUCP * TCF * EF$$

$$UCP = 62 * 0,85 * 0,725$$

$$UCP = 38,20$$

Uma das formas de derivação da estimativa de esforço é relacionar o número de UCP – *Use Case Points* – com a produtividade observada na organização, conforme tecnologia utilizada para a construção do *software*.

4- HARMONIZAÇÃO DAS ESTIMATIVAS

Cada técnica utilizada para geração das estimativas de esforço, duração ou custo produz um resultado específico. Como cada técnica possui suas características, regras e critérios, é natural que os resultados não sejam iguais, porém, quando as diferenças de resultados são muito grandes, recomenda-se:

- **Reavaliar os insumos utilizados para geração das estimativas.**
- **Harmonizar as estimativas.**

RESUMO

As técnicas de decomposição usam a abordagem “dividir e conquistar” para a estimativa de projetos de desenvolvimento de *software*: decompor o *software* em partes menores e administráveis, que podem ser analisadas pontualmente.

Na Estimativa Baseada no Problema estão previstas as estimativas baseadas na contagem do número de linhas de código (LOC) e de pontos de função (PF).

Na Estimativa Baseada no Processo, a estimativa do esforço necessário para cada função definida deve levar em consideração as atividades previstas no processo de desenvolvimento escolhido para a construção do *software*. O momento no ciclo de vida do projeto também influencia o percentual de esforço previsto para as atividades de engenharia de *software*.

Na Estimativa Baseada em Casos de Uso, a literatura apresenta formas de derivar a estimativa em LOC – linhas de código – a partir dos casos de uso identificados e suas características (quantidade, número de cenários e quantidade de páginas).

O método Pontos de Caso de Uso é um modelo para estimativa de tamanho de *software* que provê insumo para a derivação da estimativa de esforço necessário para desenvolver o *software*.