

DISCIPLINA:

QUALIDADE E GOVERNANÇA EM PROJETOS DE SOFTWARE

AULA:

13 – ESTIMATIVAS COM USE CASE / OBJECT POINTS PARA PROJETOS O.O.

PROFESSOR:

RENATO JARDIM PARDUCCI

PROFRENATO.PARDUCCI@FIAP.COM.BR

ESTIMATIVAS EM PROJETOS O.O.

Exclusivamente para projetos de software Orientados a Objetos, podemos empregar uma outra técnica de estimativas que tem obtido cada vez mais espaço no mercado:

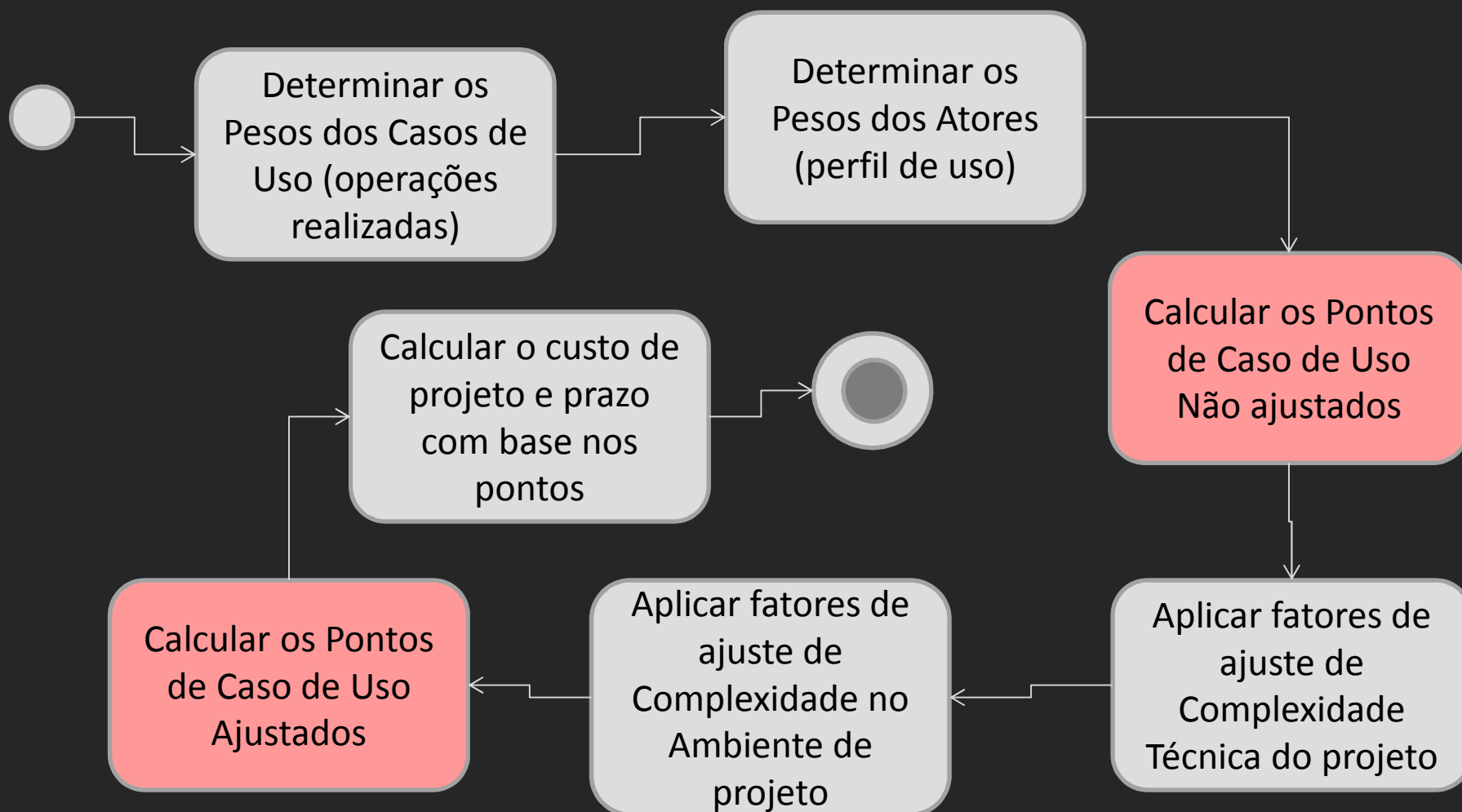
Use Case Points Estimation

Diferente da APF (FPA), a UCP (Use Case Points) não pode ser aplicada antes que se tenha o modelo de Casos de Uso completo do sistema com seus Diagramas e Documentações, onde é possível confirmar Atores*, Classes candidatas* e utilizações das Classes pelos Atores.

* Termos da Engenharia de Software Orientada a Objetos com UML.

A proposta da Estimativa por Use Case Points é **embasar as projeções de esforço no modelo de software** proposto, levando em consideração a sua complexidade funcional, realizando ajustes não funcionais ao final (como ocorre na APF).

Passos do UCP Estimation:

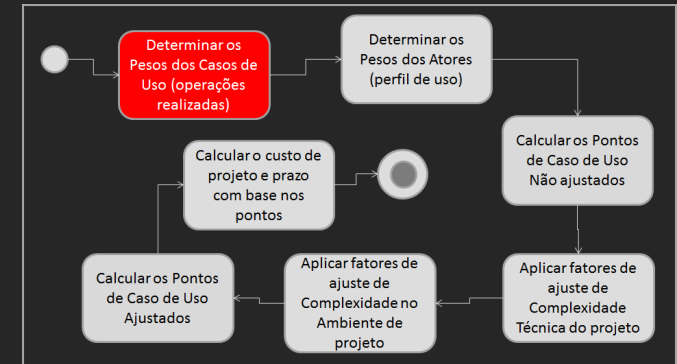


Etapa 1

Determinação de Pesos de Casos de Uso

Calcule o número de transações de cada Use Case e o número de Entidades operadas (Classes Candidatas de Entidade*)

O número de transações de um Caso de Uso corresponde ao número de Cenários de Uso* identificados no Documento de Caso de Uso, incluindo os passos dos Fluxos Principal, Alternativos e de Exceção*

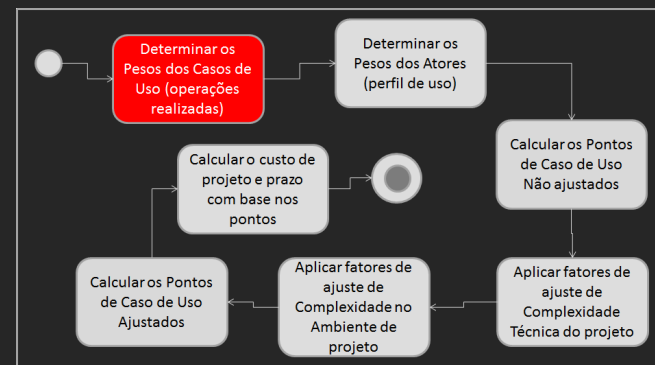


* Termos da Engenharia de Software Orientada a Objetos com UML.

Etapa 1

Determinação de Pesos de Casos de Uso

Classifique então, os Casos de Uso conforme a regra de complexidade a seguir:

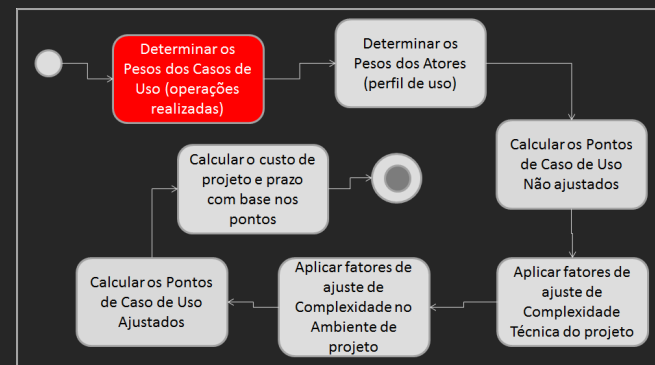


Classificação	Número de Transações	Número de Entidades	Peso Atribuído
SIMPLES	até 3	até 4	5
MÉDIO	de 4 a 7	de 5 a 10	10
COMPLEXO	mais que 7	mais que 10	15

Etapa 1

Determinação de Pesos de Casos de Uso

Exemplo do Cálculo de UUCW



Tipo	Peso	Nº de Casos de Uso	Resultado
Simples	5	7	35
Médio	10	13	130
Complexo	15	3	45
		Total UUCW	210

UUCW

U-Unadjusted

U-Use

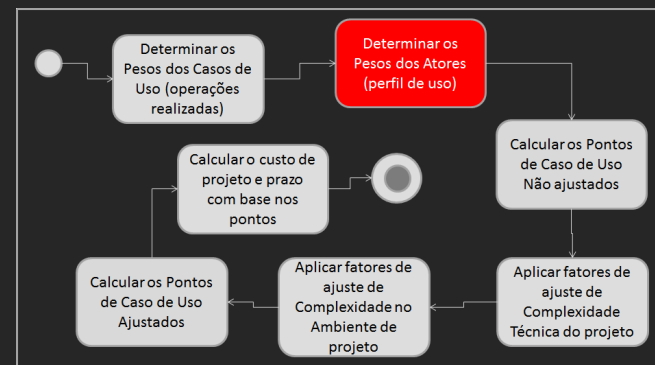
C-Case

W-Weight

Etapa2

Determinação de Pesos dos Atores

Calcule o grau de dificuldade em tratar os perfis e interações de cada Ator* do sistema.



Utilize a regra de atribuição de complexidade a seguir:

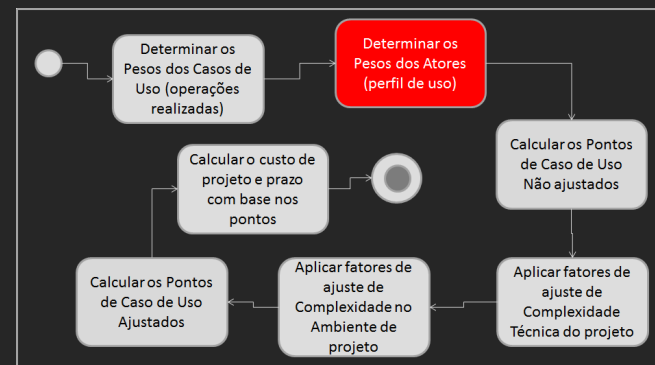
Tipo de Ator	Peso	Descrição
Ator Simples	1	Outro sistema acessado através de uma API de programação
Ator Médio	2	Outro sistema acessado interagindo através da rede por protocolo de comunicação
Ator Complexo	3	Um usuário interagindo através de uma interface gráfica

* Termos da Engenharia de Software Orientada a Objetos com UML.

Etapa2

Determinação de Pesos dos Atores

Exemplo do cálculo do UAW



Tipo de Ator	Peso	Nº de atores	Resultado
Ator Simples	1	0	0
Ator Médio	2	0	0
Ator Complexo	3	4	12
		Total UAW	12

UAW

U-Unadjusted
A-Actor
W-Weight

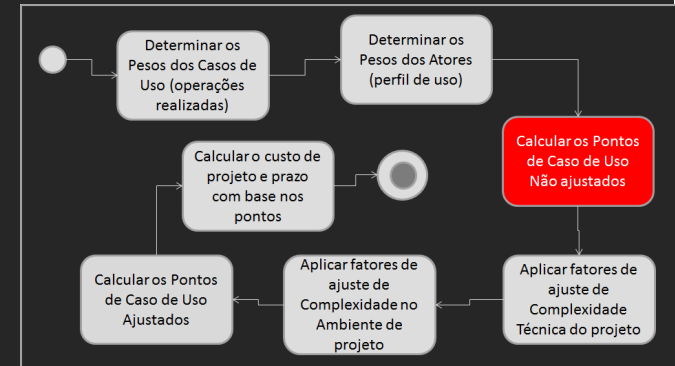
* Termos da Engenharia de Software Orientada a Objetos com UML.

Etapa3

Calcular UCP não ajustados

Cálculo do UUCP (Unadjusted Use Case Points)

- $UUCP = UAW + UUCW$



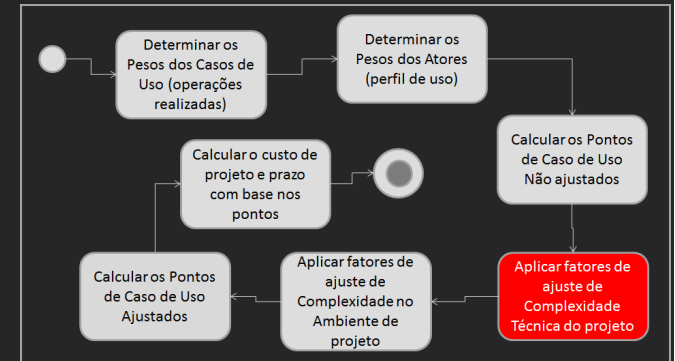
No caso do exemplo: $UUCP = 12 + 210 = 222$

Esses pontos definem a complexidade funcional do software, assim como ocorria com os pontos de função não ajustados no método de Análise de Pontos de Função (APF)

Etapa4

Ajustar complexidade técnica

O ajuste de complexidade técnica envolve a identificação de um fator (TCF – Technical Complexity Factor) que aponta o quão difícil é atender alguns requisitos não funcionais específicos do projeto do software (requisitos não funcionais intimamente relacionados com os recursos funcionais).

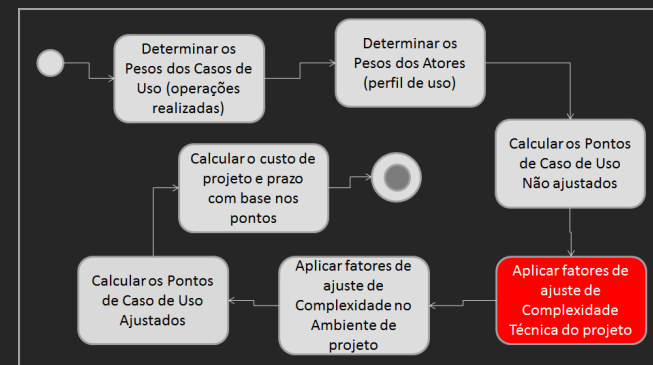


Etapa4

Ajustar complexidade técnica

O TCF é calculado pela aplicação da seguinte tabela de definição com 13 fatores:

Fator	Requisito	Peso
T1	Sistema distribuído	2
T2	Tempo de resposta	2
T3	Eficiência	1
T4	Processamento complexo	1
T5	Código reusável	1
T6	Facilidade de instalação	0.5
T7	Facilidade de uso	0.5
T8	Portabilidade	2
T9	Facilidade de mudança	1
T10	Concorrência	1
T11	Recursos de segurança	1
T12	Acessível por terceiros	1
T13	Requer treinamento especial	1



O quadro ao lado traz os pesos padrão, determinados pelo modelo de cálculo de Use Case Points, indicando a importância de cada requisito não funcional dentro de um projeto de software.

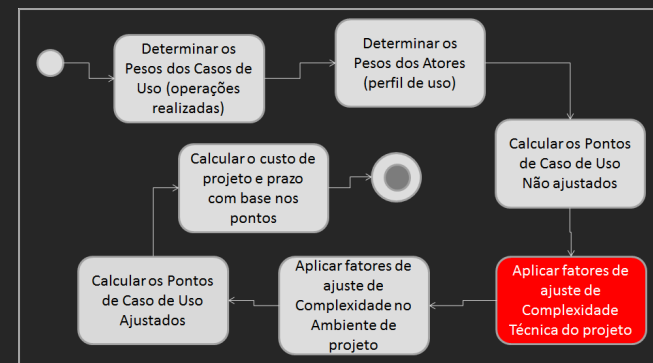
Esses valores serão usados como base para o cálculo do fator de ajuste final do projeto, por motivos de requisitos não funcionais.

Etapa4

Ajustar complexidade técnica

O TCF é calculado pela aplicação da seguinte tabela de definição com 13 fatores:

Fator	Requisito	Peso
T1	Sistema distribuído	2
T2	Tempo de resposta	2
T3	Eficiência	1
T4	Processamento complexo	1
T5	Código reusável	1
T6	Facilidade de instalação	0.5
T7	Facilidade de uso	0.5
T8	Portabilidade	2
T9	Facilidade de mudança	1
T10	Concorrência	1
T11	Recursos de segurança	1
T12	Acessível por terceiros	1
T13	Requer treinamento especial	1



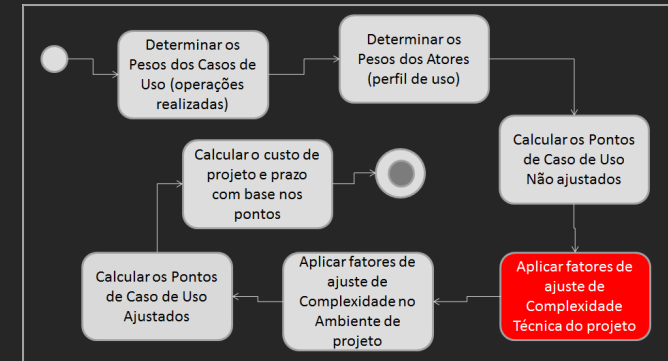
Note que **quanto maior o peso, mais difícil** se torna o projeto

Etapa4

Ajustar complexidade técnica

O TCF é calculado pela aplicação da seguinte tabela de definição com 13 fatores:

Fator	Requisito	Peso
T1	Sistema distribuído	2
T2	Tempo de resposta	2
T3	Eficiência	1
T4	Processamento complexo	1
T5	Código reusável	1
T6	Facilidade de instalação	0.5
T7	Facilidade de uso	0.5
T8	Portabilidade	2
T9	Facilidade de mudança	1
T10	Concorrência	1
T11	Recursos de segurança	1
T12	Acessível por terceiros	1
T13	Requer treinamento especial	1



Cada fator de ponderação deve receber um peso de 0 a 5 que determina a influência do requisito no sistema, sendo a escolha dessa pontuação, a critério de quem faz a estimativa:

- 0 – não influencia
- 1-influencia uma função / operação específica do Sw
- 2-influencia um conjunto pequeno de funções
- 3-influencia as funções mais importantes (core)
- 4-influencia quase todo o sistema
- 5-influencia todo o sistema

Etapa4

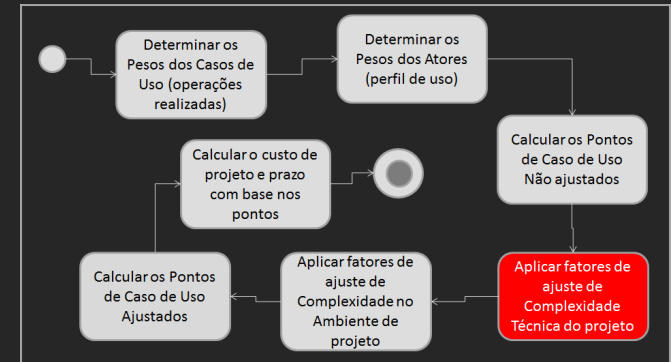
Ajustar complexidade técnica

Para calcular o TCF, atribui-se as notas de 0 a 5 para cada fator e multiplica-se a nota pelo peso do fator.

Depois, soma-se o total de notas ponderadas, gerando um Tfactor (Fator Técnico).

O TCF é calculado então pela equação:

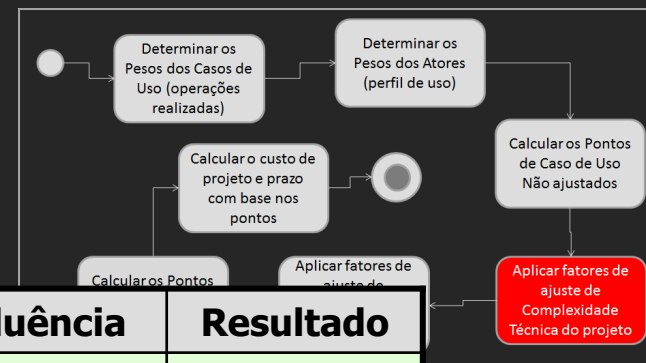
- $TCF = 0.6 + (0.01 \times Tfactor)$



Etapa4

Ajustar complexidade técnica

Exemplo de apuração do TCF:

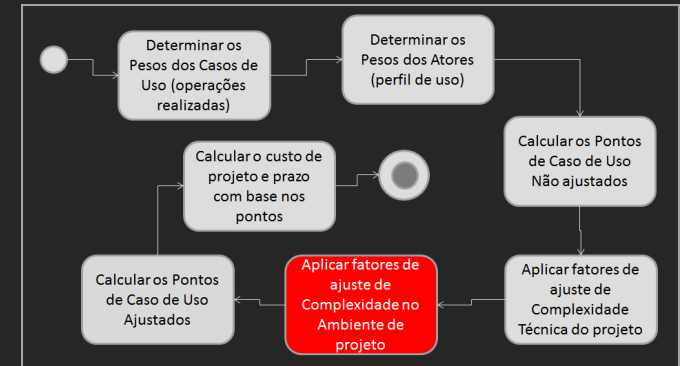


Fator	Requisito	Peso	Influência	Resultado
T1	Sistema distribuído	2	1	2
T2	Tempo de resposta	2	3	6
T3	Eficiência	1	3	3
T4	Processamento complexo	1	3	3
T5	Código reusável	1	0	0
T6	Facilidade de instalação	0.5	0	0
T7	Facilidade de uso	0.5	5	2.5
T8	Portabilidade	2	0	0
T9	Facilidade de mudança	1	3	3
T10	Concorrência	1	0	0
T11	Recursos de segurança	1	0	0
T12	Acessível por terceiros	1	0	0
T13	Requer treinamento especial	1	0	0
TCF = 0.6 + (0.01 × 19.5) = 0.795			Tfactor	19,5

Etapa5

Ajustar complexidade ambiental

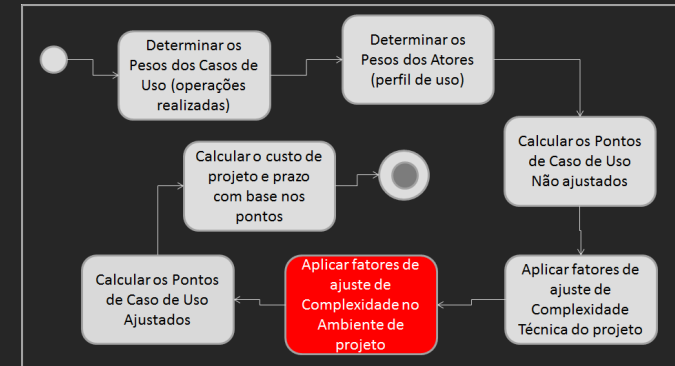
O ajuste de complexidade ambiental envolve a identificação de um fator (ECF – Environmental Complexity Factor) que aponta o quão difícil é atender alguns requisitos não funcionais relacionados ao processo de produção do software.



Etapa5

Ajustar complexidade ambiental

O ECF é calculado pela aplicação da seguinte tabela de definição:



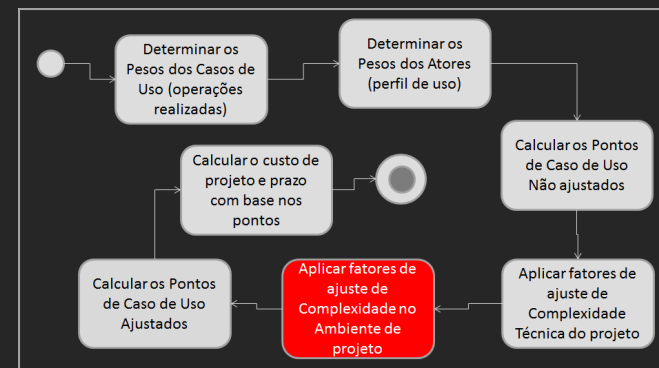
O quadro ao lado traz os pesos padrão, determinados pelo modelo de cálculo de Use Case Points, indicando a importância de cada requisito não funcional dentro de um projeto de software. Esses valores serão usados como base para o cálculo do fator de ajuste final do projeto, por motivos de processo de software.

Fator	Descrição	Peso
E1	Familiaridade com RUP ou outro processo formal	1.5
E2	Experiência com a aplicação em desenvolvimento	0.5
E3	Experiência em Orientação a Objetos	1
E4	Presença de analista experiente	0.5
E5	Motivação	1
E6	Requisitos estáveis	2
E7	Desenvolvedores em meio-expediente	-1
E8	Linguagem de programação difícil	-1

Etapa5

Ajustar complexidade ambiental

O ECF é calculado pela aplicação da seguinte tabela de definição:



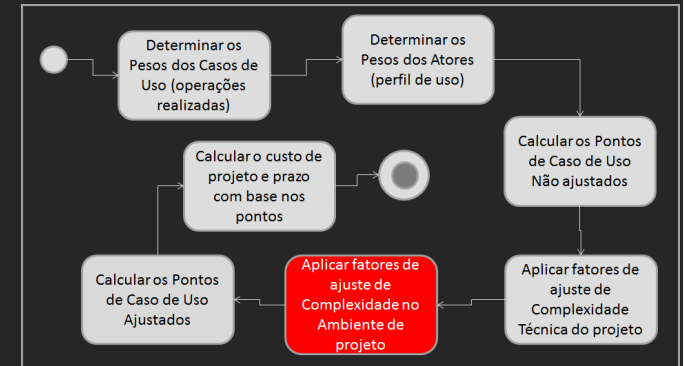
Fator	Descrição	Peso
E1	Familiaridade com RUP ou outro processo formal	1.5
E2	Experiência com a aplicação em desenvolvimento	0.5
E3	Experiência em Orientação a Objetos	1
E4	Presença de analista experiente	0.5
E5	Motivação	1
E6	Requisitos estáveis	2
E7	Desenvolvedores em meio-expediente	-1
E8	Linguagem de programação difícil	-1

Perceba que alguns fatores **influenciam positivamente na redução de complexidade** (quanto maior a redução, maior a pontuação)

Etapa5

Ajustar complexidade ambiental

O ECF é calculado pela aplicação da seguinte tabela de definição:



Fator	Descrição	Peso
E1	Familiaridade com RUP ou outro processo formal	1.5
E2	Experiência com a aplicação em desenvolvimento	0.5
E3	Experiência em Orientação a Objetos	1
E4	Presença de analista experiente	0.5
E5	Motivação	1
E6	Requisitos estáveis	1
E7	Desenvolvedores em meio-expediente	-1
E8	Linguagem de programação difícil	-1

Enquanto alguns fatores influenciam negativamente, aumentando a complexidade

Etapa4

Ajustar complexidade técnica

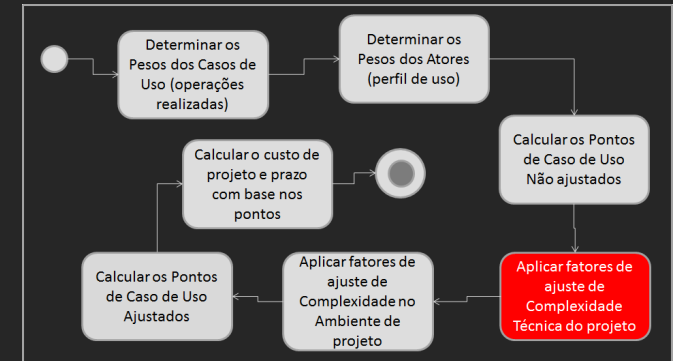
Para calcular o ECF, atribui-se as notas de 0 a 5 para cada fator e multiplica-se a nota pelo Peso do fator.

Depois, soma-se o total de notas ponderadas, gerando um Efator (Fator Ambiental).

O ECF é calculado então pela equação:

- $ECF = 1.4 + (-0.03 \times \text{Efator})$

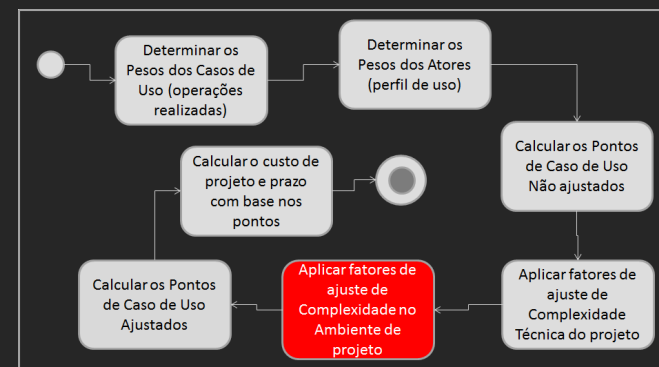
*Observe que esse fator é redutor de complexidade, uma vez que notas mais altas são atribuídas à maior desenvoltura de equipe no projeto.



Etapa5

Ajustar complexidade ambiental

O ECF é calculado pela aplicação da seguinte tabela de definição:



Fator	Descrição	Peso
E1	Familiaridade com RUP ou outro processo formal	1.5
E2	Experiência com a aplicação em desenvolvimento	0.5
E3	Experiência em Orientação a Objetos	1
E4	Presença de analista experiente	0.5
E5	Motivação	1
E6	Requisitos estáveis	2
E7	Desenvolvedores em meio-expediente	-1
E8	Linguagem de programação difícil	-1

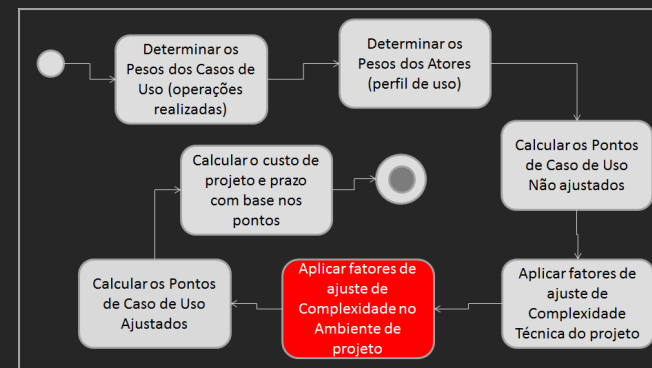
Cada fator de ponderação deve receber um peso de 0 a 5 que determina a influência do requisito na condução do projeto

0 – não requerido/
presente durante o projeto
1-não requerido/ presente
2-razoavelmente não
requerido/ presente
3-bastante não requerido/
presente
4-extremamente
requerido/ presente
5-obrigatório

Etapa5

Ajustar complexidade ambiental

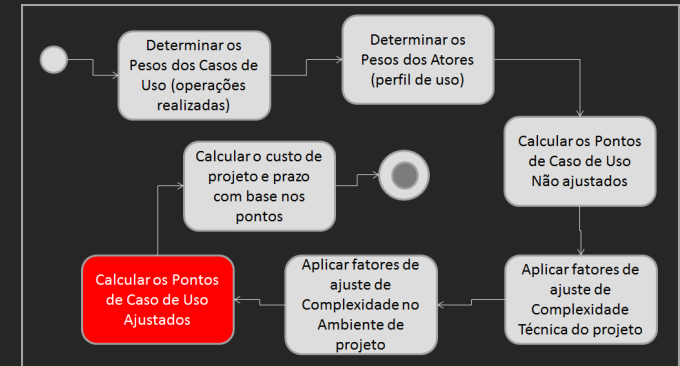
Exemplo de cálculo do ECF:



Fator	Descrição	Peso	Influência	Resultado
E1	Familiaridade com RUP ou outro processo formal	1.5	5	7.5
E2	Experiência com a aplicação em desenvolvimento	0.5	0	0
E3	Experiência em Orientação a Objetos	1	5	5
E4	Presença de analista experiente	0.5	5	2.5
E5	Motivação	1	5	5
E6	Requisitos estáveis	2	3	6
E7	Desenvolvedores em meio-expediente	-1	0	0
E8	Linguagem de programação difícil	-1	0	0
ECF = 1.4 + (-0.03 × 26) = 0.62			Efactor	26

Etapa 6

Calcular UCP ajustados



Cálculo dos UCP (Use Case Points)

- $UCP = UUCP \times TCF \times ECF$

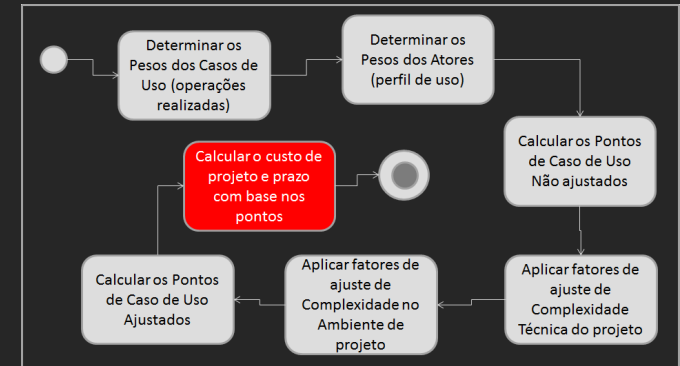
No nosso exemplo:

$$UCP = 222 \times 0.795 \times 0.62 = 109.42 \text{ ou } 109 \text{ Use Case Points}$$

Etapa 7

Calcular Esforço em Horas, Custo e Prazo

Para calcular o esforço, tempo e o custo, usamos raciocínio equivalente ao usado na APF.



Se a empresa tiver uma tabela de conversão e Pontos de Caso de Uso para Horas de Trabalho, basta fazer a multiplicação.

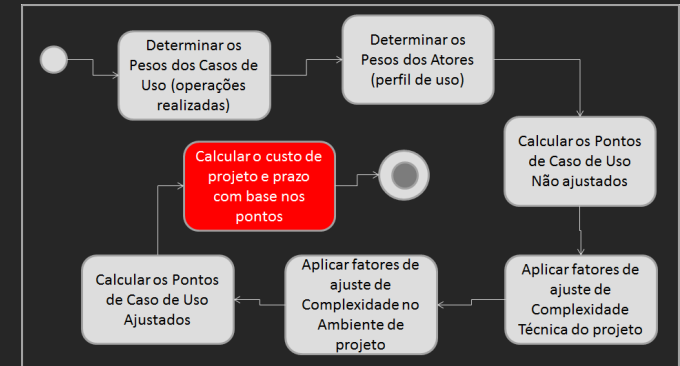
Tendo as horas de trabalho, basta descobrir o custo médio da equipe de TI por hora e multiplicar o valor hora pelo tempo total, gerando a estimativa de custo total de mão-de-obra do projeto.

O prazo dependerá o grau de paralelismo das frentes de trabalho e do número de horas efetivas de trabalho (produtividade diária por analista).

Etapa 7

Calcular Esforço em Horas, Custo e Prazo

Exemplo dos cálculos de custo e tempo, considerando que a média histórica de tempo por ponto do UCP é de 20 horas.



No caso do exemplo:

$$\text{Esforço estimado} = 109 * 20 = \mathbf{2180 \text{ horas de trabalho}}$$

E considerando que o custo médio dos profissionais de desenvolvimento de software é de R\$ 50,00, temos o custo da mão-de-obra para o projeto pela multiplicação do tempo estimado

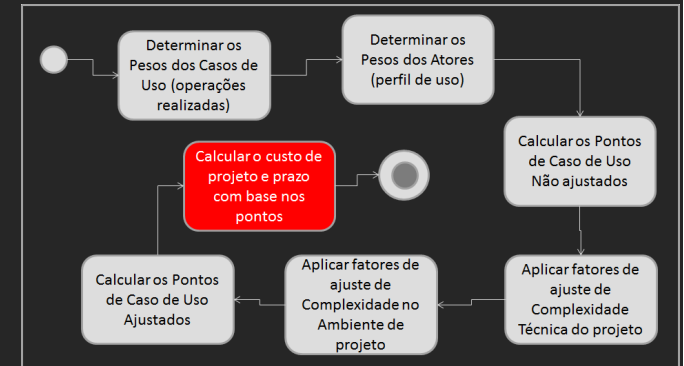
No caso do exemplo:

$$\text{Custo estimado} = 2180 * 50 = \mathbf{R\$ 109.000}$$

Etapa 7

Calcular Custo e Prazo

IMPORTANTE: quando a empresa não tem histórico de projetos suficiente para confirma a sua produtividade, recomenda-se a conversão 1 pra 1 de Pontos de Caso de Uso para Horas de trabalho!



Para fechar o custo total do projeto é necessário somar:

- Custo de treinamento
- Custo compra ou locação de hardware
- Custo de compra ou locação para software de apoio (licenças de BD, Ferramenta CASE, etc.)
- Custo de serviços de terceiros
- Aluguéis de espaços físicos;
- Despesas em geral com transporte e alimentação;
- Outros itens específicos do projeto.

FERRAMENTAS DE CONTROLE DE VERSÃO

ATIVIDADE PRÁTICA



Utilize o arquivo ESTUDO-DE-CASO-UCP.doc e estime o projeto usando Use Case Points.

Considere que 1 UCP equivale a 20 horas de trabalho e que o custo médio de um profissional da equipe de desenvolvimento é de R\$ 30,00 por hora para calcular o esforço em horas e o custo estimado do projeto.

FERRAMENTAS DE CONTROLE DE VERSÃO

ATIVIDADE PRÁTICA



Estude os seguintes arquivos referentes ao projeto de sistema de informação para Service Desk que estamos usando como projeto piloto (documentos publicados na sua área de apostilas):

- Diag-UC-Refinado-SD.asta
- Descrição-UC-SD.doc

Estime o esforço de projeto por UCP.

Considere que 1 UCP equivale a 20 horas de trabalho e que o custo médio de um profissional da equipe de desenvolvimento é de R\$ 30,00 por hora para calcular o esforço em horas e o custo estimado do projeto.

FERRAMENTAS DE CONTROLE DE VERSÃO



PROJETO

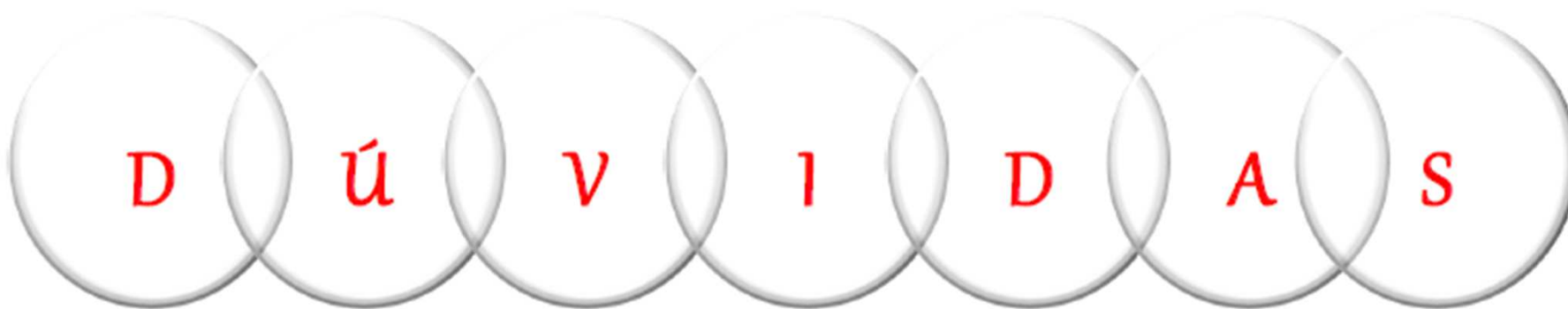
Estime o seu projeto AM usando UCP.

Estabeleça um tempo em horas equivalente a 1 Ponto de Caso de Uso, com base na produtividade da equipe.

Calcule o esforço total do seu projeto e compare com os cálculos que fez por APF.

ESSA ENTREGA NÃO SERÁ COBRADA PARA NOTA MAS É IMPORTANTE DE SER REALIZADA!

VALIDE COM O PROFESSOR!



EXERCÍCIOS extras



Resolvidos

1. Estimativa por UCP

Considere os materiais que seguem nos próximos Slides com fundo branco para calcular o esforço do projeto em UCP.

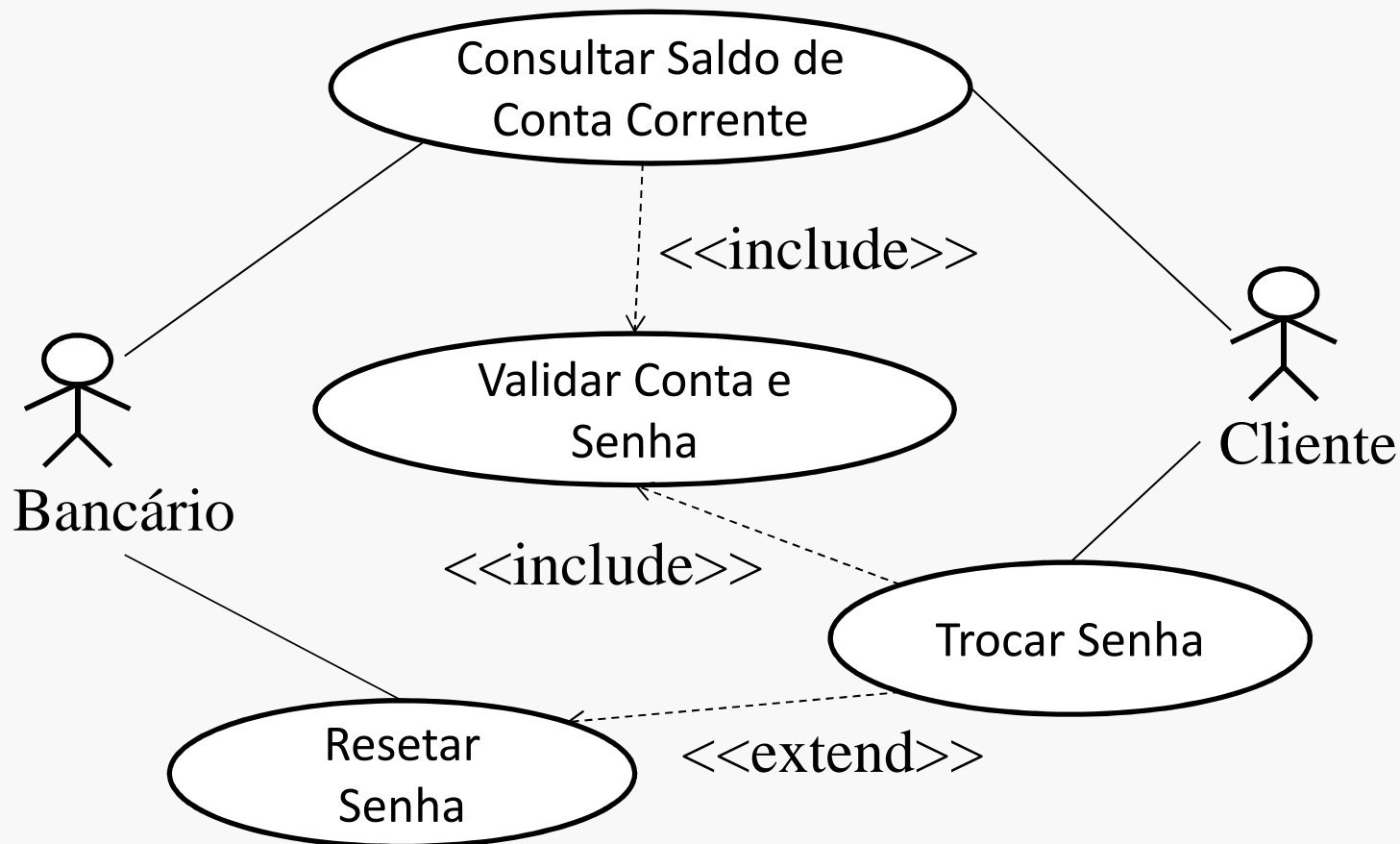
Leve em conta que: a Facilidade de Uso e o Tempo de Resposta são críticos para a aplicação. O Treinamento de usuários será um desafio razoável, em função da quantidade de pessoas envolvidas, porém, serão utilizados replicadores de conhecimento entre os próprios usuários para reduzir o esforço da equipe de projeto. Como o projeto é aparentemente simples, foi destacada uma equipe bastante inexperiente e iniciante na metodologia de gestão de projeto e na execução de projetos O.O. Os desenvolvedores atuarão em tempo parcial.

Considere que cada UCP equivale a 1 hora de trabalho.

Considere que apenas um desenvolvedor atuará por vez no projeto, sem paralelismos e calcule o prazo em dias. A produtividade média diária por desenvolvedor é de 6 horas por dia.

Leve em conta o valor médio por hora de desenvolvedor igual a R\$ 30,00 e apure o custo do projeto.

Diagrama de Casos de Uso de Consulta de Saldo Bancário



Nome do Caso de Uso: Consultar Saldo de Conta Corrente

Resumo: possibilitar que um bancário consulte o saldo para um cliente da agência, cliente esse que informará dados de conta para o bancário e digitará a sua senha no sistema

Atores participantes: Bancário; Cliente

Casos de Uso relacionados: Validar Conta e Senha

Pré-condição: Bancário deve estar logado no sistema

Fluxo principal

Ações do ator

1. Informar Número da Conta do Cliente
2. Informar a senha
5. Visualizar seu Saldo
6. Encerrar a operação/UC

Ações do Sistema

3. **Executar UC Validar Conta e Senha**
4. Exibir Saldo da Conta [E1]

Fluxo alternativo

-

Fluxo de exceção E1 – Tratar Inexistência de Conta ou Senha inválida

1. Sistema exibe a mensagem “Dados de Conta e/ou Senha incorretos!” e encerra

Pós condição: Exibição do saldo da conta do cliente ou Mensagem de erro de identificação

Nome do Caso de Uso: Validar Conta e Senha

Resumo: confirma a existência da combinação Conta e Senha no banco de dados de clientes da instituição financeira

Atores participantes: -

Casos de Uso relacionados: Consultar Saldo de Conta Corrente

Pré-condição: caso de uso Consultar Saldo de Conta Corrente ou Trocar Senha em execução

Fluxo principal

Ações do ator

Ações do Sistema

1. Receber dados de Conta e Senha
2. Pesquisas dados de Conta e Senha nos registros de clientes do banco
3. Retornar confirmação de dados corretos [E1]

Fluxo alternativo

-

Fluxo de exceção E1 – Tratar Inexistência da combinação Conta e Senha

1. Sistema retorna a negação dos dados informados

Pós condição: Retorno de indicador de existência ou não de Conta e Senha

Nome do Caso de Uso: Resetar Senha

Resumo: coloca a senha padrão como senha do cliente bancário para que possa ocorrer a substituição em seguida por uma nova senha criada pelo cliente

Atores participantes: Bancário

Casos de Uso relacionados: Trocar Senha

Pré-condição: -

Fluxo principal

Ações do ator

1. Informar Número da Conta do Cliente
2. Solicitar reset de senha
4. Encerrar a operação/UC [A1]

Ações do Sistema

3. Atribuir a senha default para a Conta do Cliente

Fluxo alternativo A1 – Trocar a Senha por novo código definido pelo Cliente

1. Executar UC Trocar Senha

Fluxo de exceção

-

Pós condição: Senha default atribuída à Conta do Cliente bancário

Nome do Caso de Uso: Trocar Senha

Resumo: Substituição da senha atual associada à Conta de um cliente bancário por uma nova senha criada pelo próprio cliente

Atores participantes: Cliente

Casos de Uso relacionados: Resetar Senha

Pré-condição: -

Fluxo principal

Ações do ator

1. Informar Número da Conta do Cliente
2. Informar Senha atual
4. Informar Nova Senha [E1]
6. Encerrar a operação/UC

Ações do Sistema

3. **Executar UC Validar Conta e Senha**
5. Associar Nova Senha à Conta bancária

Fluxo alternativo

Fluxo de exceção E1 – Tratar Inexistência de Conta ou Senha inválida

1. Sistema exibe a mensagem “Dados de Conta e/ou Senha incorretos!” e encerra

Pós condição: Senha definida pelo Cliente é atribuída à Conta do Cliente bancário

1. SOLUÇÃO

Análises sobre a modelagem do projeto de software para calcular UCCW

a. Número de Transações identificadas (Cenários de Uso):

a.1. UC Consultar Saldo de Conta Corrente = 2 (Principal + 1 Exceção)

a.2. UC Validar Conta e Senha = 2 (Principal + 1 Exceção)

a.3. UC Resetar Senha = 2 (Principal + 1 Alternativo)

a.4. UC Trocar Senha = 2 (Principal + 1 Exceção)

b. Número de Entidades identificadas:

b.1. UC Consultar Saldo de Conta Corrente = 2 (Cadastro de Conta + Controle de Saldo)

b.2. UC Validar Conta e Senha = 1 (Cadastro de Conta)

b.3. UC Resetar Senha = 1 (Cadastro de Conta)

b.4. UC Trocar Senha = 1 (Cadastro de Conta)

Análises sobre a modelagem do projeto de software para calcular UAW

c. Ator Bancário = Um usuário interagindo através de uma interface gráfica

d. Ator Cliente = Um usuário interagindo através de uma interface gráfica

Aplicação da Matriz de Complexidade para calcular UCCW

Classificação	Número de Transações	Número de Entidades	Peso Atribuído
SIMPLES	até 3	até 4	5
MÉDIO	de 4 a 7	de 5 a 10	10
COMPLEXO	mais que 7	mais que 10	15

UC	Nro Transações	Nro Entidades	Peso atribuído
Consultar Saldo de Conta Corrente	2	2	5
Validar Conta e Senha	2	1	5
Resetar Senha	2	1	5
Trocar Senha	2	1	5

$$\text{UCCW} = 5+5+5+5 = 20$$

Aplicação da Matriz de Complexidade para calcular UAW

Tipo de Ator	Peso	Descrição
Ator Simples	1	Outro sistema acessado através de uma API de programação
Ator Médio	2	Outro sistema acessado interagindo através da rede por protocolo de comunicação
Ator Complexo	3	Um usuário interagindo através de uma interface gráfica

Ator	Outro sistema acessado através de uma API de programação	Outro sistema acessado interagindo através da rede por protocolo de comunicação	Um usuário interagindo através de uma interface gráfica	Peso atribuído
Bancário	-	-	X	3
Cliente	-	-	X	3

UAW = 3+3 = 6

Cálculo do UUCP (Pontos de Caso de Uso não ajustados):

- $UUCP = UAW + UUCW = 20 + 6 = 26$

Apurando a influência dos ajustes por fatores técnicos do projeto:

Fator	Requisito	Peso	Influência	Resultado
T1	Sistema distribuído	2	0	0
T2	Tempo de resposta	2	5	5
T3	Eficiência	1	1	1
T4	Processamento complexo	1	0	0
T5	Código reusável	1	0	0
T6	Facilidade de instalação	0.5	0	0
T7	Facilidade de uso	0.5	5	2.5
T8	Portabilidade	2	0	0
T9	Facilidade de mudança	1	0	0
T10	Concorrência	1	0	0
T11	Recursos de segurança	1	1	1
T12	Acessível por terceiros	1	0	0
T13	Requer treinamento especial	1	0	0
			Tfactor	Soma = 8.5

- $TCF = 0.6 + (0.01 \times Tfactor) = 0.6 + (0.01 \times 8.5)$
 $= 0.685$

Apurando a influência dos ajustes por fatores ambientais do projeto:

Fator	Descrição	Peso	Influência	Resultado
E1	Familiaridade com RUP ou outro processo formal	1.5	1	1.5
E2	Experiência com a aplicação em desenvolvimento	0.5	1	0.5
E3	Experiência em Orientação a Objetos	1	1	1
E4	Presença de analista experiente	0.5	0	0
E5	Motivação	1	0	0
E6	Requisitos estáveis	2	0	0
E7	Desenvolvedores em meio-expediente	-1	5	-5
E8	Linguagem de programação difícil	-1	0	0
			Efactor	Soma = -2

- $ECF = 1.4 + (-0.03 \times Efactor) = 1.4 + (-0.03 \times (-2)) = 1.46$

Cálculo do UCP (Pontos de Caso de Uso ajustados):

- $UCP = UUCP \times TCF \times ECF$

$$= 26 \times 0.685 \times 1.46 = \mathbf{26.0026}$$

Cálculo do Esforço em Horas, considerando que cada Ponto de Caso de Uso equivale a 1 hora de trabalho.

- Horas de Esforço de Trabalho = UCP x 20

$$= 26.0026 \times 1 = 26 \text{ horas de trabalho!}$$

Cálculo do Prazo em dias, considerando que não existe paralelismo e que a produtividade média diária é de 6 horas de trabalho.

- Prazo de projeto em dias =
= Horas de esforço / Produtividade média / Número de frentes paralelas

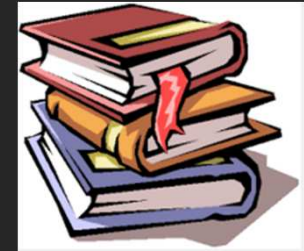
= 26 / 6 / 1 = 4.5 dias (4 dias e meio = 1 semana útil)

Cálculo do Custo, considerando um valor médio de profissional de desenvolvimento de software de R\$ 30,00.

- **Custo da Mão-de-obra do projeto =
= Horas de esforço x Custo médio por hora de profissional**

$$= 26 \times 30 = \text{R\$ } 780.00$$

Referência bibliográficas



BIBLIOGRAFIA :

- PRESSMAN, Roger S.. Engenharia de software. - Uma abordagem profissional, 7ª edição. São Paulo, AMGH, 2011.

ESTIMATIVA DE SOFTWARE

FIM

PROFESSOR:
RENATO JARDIM PARDUCCI

PROFRENATO.PARDUCCI@FIAP.COM.BR