



ESTIMAÇÃO DE ESFORÇO II

COCOMO II

Prof. Raul Sidnei Wazlawick
UFSC-CTC-INE

CONTEÚDO

SLOC e
KSLOC

COCOMO
81

COCOMO II



SLOC E KSLOC



- Como se faz a estimaco?



KSLOC USADO NAS ESTIMATIVAS

- Consenso ou:
- $KSLOC = (4 * KSLOC_{esperado} + KSLOC_{otimista} + KSLOC_{pessimista}) / 6$



PADRÃO PARA CONTAGEM DE SLOC EM JAVA, C, C++ E C#

Precedência	Estrutura	Regra de contagem
1	Comandos de seleção: if, else if, else, operador "?", try, catch, switch	Conta uma vez cada ocorrência. Comandos aninhados são contados de forma similar.
2	Comandos de iteração: for, while, do..while	Conta uma vez cada ocorrência. A inicialização, a condição e o incremento da instrução for não contam, bem como quaisquer outras expressões opcionais do for.
3	Comandos de desvio: return, break, goto, exit, continue, throw	Conta uma vez cada ocorrência. Rótulos usados com comandos goto não contam.
4	Expressões: Chamada de função, atribuição, comando vazio	Conta uma vez cada ocorrência.
5	Outras expressões: Expressões que terminam com ";"	Conta uma vez cada ocorrência. Expressões consistindo unicamente de ";" dentro de comandos de iteração não contam.
6	Delimitadores de bloco {...}	Contam uma vez por par exceto se o fechamento for sucedido por ";", ou seja, "};". Chaves usadas com comandos de seleção e iteração não contam. Definição de função que necessariamente inclui delimitadores de bloco conta uma única vez.
7	Diretivas de compilação	Conta uma vez cada ocorrência.
8	Declaração de dados	Conta uma vez cada ocorrência. Isso inclui protótipos de função, declaração de variáveis e declarações typedef. Palavras chave como struct e class não contam.

COMO CONTAR SLOCs?

	Contagem	Regra (precedência)
public class Fibonacci {	-	8
static long fibo(int n) {	-	
if (n < 2) {	1	6
return n;	1	1
} else {	1	3
return fibo(n - 1) + fibo(n - 2);	1	1
}	1	3
}	-	
public static void main(String[] args) {	-	
// teste do programa. Imprime os 30 primeiros termos	1	6
for (int i = 0; i < 30; i++) {	-	
System.out.print("(" + i + "):" + Fibonacci.fibo(i) + "\t");	1	2
}	1	5
}	-	
}	-	
}	-	
TOTAL	8	



BACKFIRE TABLE

Linguagem	Média	Mediana	Melhor caso	Pior Caso
Assembly	119	98	25	320
C	97	99	39	333
C++	50	53	25	80
C#	54	59	29	70
Java	53	53	14	134
Perl	24	15	15	60
SQL	21	21	13	37
Visual Basic	42	44	20	60



COCOMO

- *Constructive Cost Model*
(também conhecido como COCOMO 81).
- Boehm (1981)

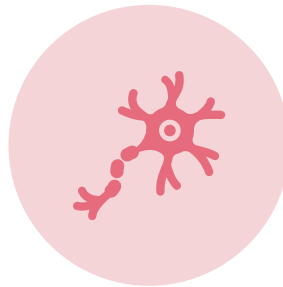


Barry Boehm
Born in 1935
In America
Software engineer,
Distinguished
Professor at the
University of
Southern California

TIPOS DE PROJETO



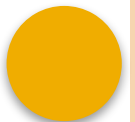
***MOD
ORGÂNICO***



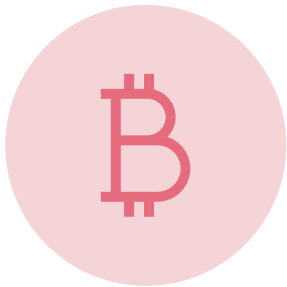
***MOD
SEMIDESTACA
DO***



***MOD
EMBTIDO***



AS TRÊS IMPLEMENTAÇÕES DO MODELO COCOMO PERMITEM DETERMINAR 3 INFORMAÇÕES BÁSICAS:



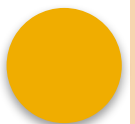
*E – ESFORÇO
TOTAL*



*T – TEMPO
LINEAR IDEAL*



*P – TAMANHO
MÉDIO DA
EQUIPE*

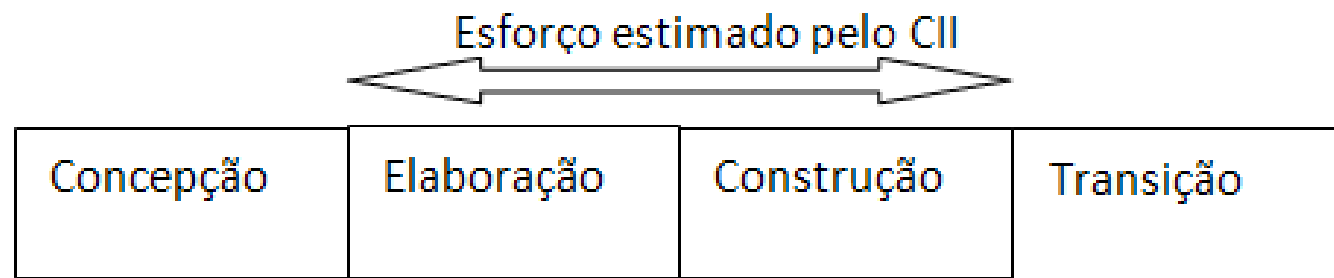


ESFORÇO, TEMPO LINEAR E TAMANHO DE EQUIPE

- $E = ab * KSLOC^{bb}$
- $T = cb * E^{db}$
- $P = E/T$

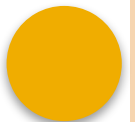
Tipo de projeto	<i>ab</i>	<i>bb</i>	<i>cb</i>	<i>db</i>
Orgânico	2,4	1,05	2,5	0,38
semidestacado	3,0	1,12	2,5	0,35
Embutido	3,6	1,2	2,5	0,32

COCOMO II - CII



EQUAÇÃO GERAL CII

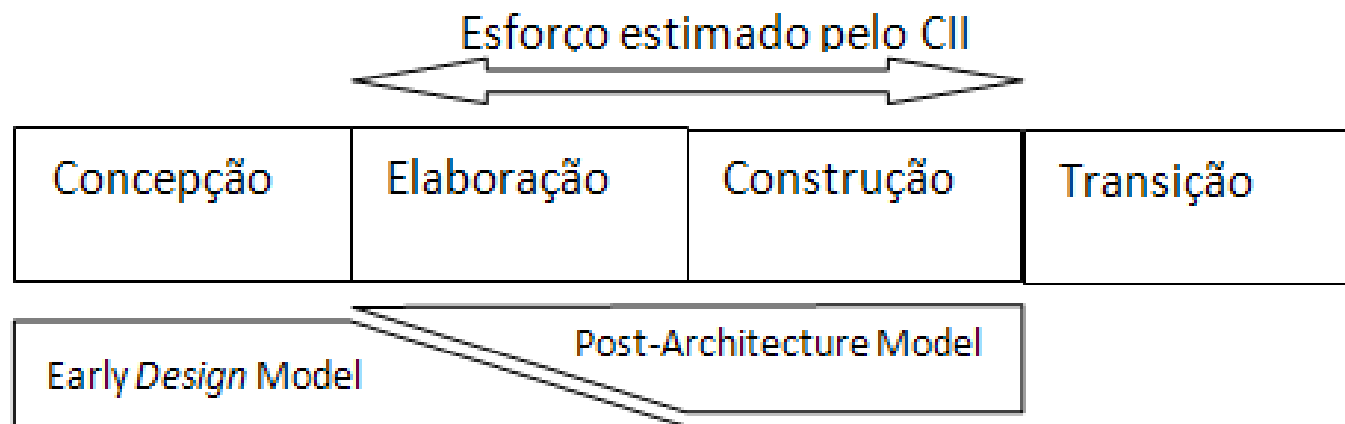
$$E = A * KSLOC^S * \prod_{i=1}^n M_i$$



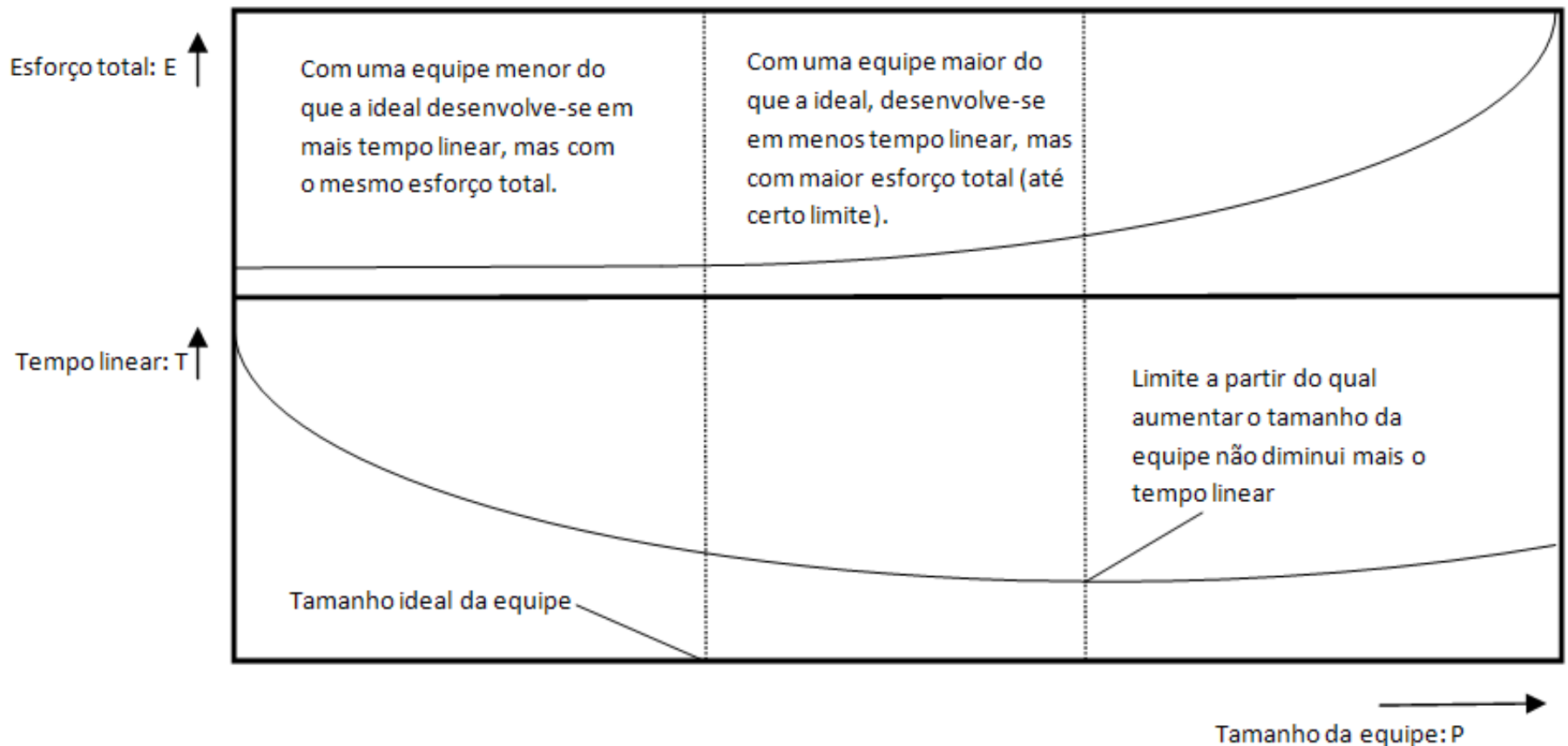
COEFICIENTE DE ESFORÇO

$$S = B + 0,001 * \sum_{j=1}^5 E_j$$





RELAÇÃO ENTRE TAMANHO DE EQUIPE, ESFORÇO E TEMPO LINEAR



TEMPO LINEAR

$$T = C * (E)^{D+0,2*(S-B)}$$

Onde:

- a) T é o tempo linear ideal de desenvolvimento.
- b) B , C e D são constantes que devem ser calibradas a partir de dados históricos (ver abaixo).
- c) E é o esforço total para o projeto, conforme calculado anteriormente.
- d) S é o expoente de esforço que já foi mencionado.

a) $A = 2,94$

b) $B = 0,91$

c) $C = 3,67$

d) $D = 0,28$



FATORES DE ESCALA

$$E = A * KSLOC^S * \prod_{i=1}^n M_i$$

$$S = B + 0,001 * \sum_{j=1}^5 F_j \leftarrow$$

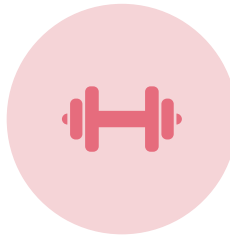
- “muito baixo” -> “extremamente alto”.
- Impacto exponencial



FATORES DE ESCALA



PRECEDENTES (PREC)



*FLEXIBILIDADE NO
DESENVOLVIMENTO
(FLEX)*



*ARQUITETURA/
RESOLUÇÃO DE
RISCOS (RESL)*



*COESÃO DA EQUIPE
(TEAM)*



*MATURIDADE DE
PROCESSO (PMAT)*



PRECEDENTES

Tabela 7-7: Forma de obtenção do equivalente numérico para PREC.

Característica	Muito baixo Baixo	Nominal Alto	Muito alto Extra-alto
Compreensão organizacional dos objetivos do produto	Geral	Considerável	Total
Experiência no trabalho com sistemas de software relacionados	Moderada	Considerável	Extensiva
Desenvolvimento concorrente de novo hardware e procedimentos operacionais associados	Extensivo	Moderado	Algum
Necessidade de arquiteturas e algoritmos de processamento de dados inovadores	Considerável	Algum	Mínimo

Nota média	Muito baixo	Baixo	Nominal	Alto	Muito alto	Extra-alto
Interpretação	Totalmente sem precedentes	Largamente sem precedentes	Um tanto sem precedentes	Genericamente familiar	Altamente familiar	Totalmente familiar
Fator numérico	6,20	4,96	3,72	2,48	1,24	0,00



FLEXIBILIDADE NO DESENVOLVIMENTO

Tabela 7-8: Forma de obtenção do equivalente numérico para FLEX.

Característica	Muito baixo Baixo	Nominal Alto	Muito alto Extra-alto
Necessidade de conformação do software a requisitos pré-estabelecidos	Total	Considerável	Básica
Necessidade de conformação do software a especificações de interfaces com sistemas externos	Total	Considerável	Básica
Combinação das inflexibilidades acima com prêmio por término antecipado do projeto	Alto	Médio	Baixo

Nota média	Muito baixo	Baixo	Nominal	Alto	Muito alto	Extra-alto
Interpretação	Rigoroso	Relaxamento ocasional	Algum relaxamento	Conformidade geral	Alguma conformidade	Metas gerais
Fator numérico	5,07	4,05	3,04	2,03	1,01	0,00



RESOLUÇÃO DE RISCOS

Tabela 7-9: Forma de obtenção do equivalente numérico para RESL.

Característica	Muito baixo	Baixo	Nominal	Alto	Muito alto	Extra-alto
O plano de gerenciamento de risco identifica todos os itens de risco críticos e estabelece marcos para resolvê-los	Nada	Um pouco	Alguma coisa	Geralmente	Largamente	Totalmente
Cronograma, orçamento e marcos internos são compatíveis com o plano de gerenciamento de risco	Nada	Um pouco	Alguma coisa	Geralmente	Largamente	Totalmente
Percentual do cronograma de desenvolvimento devotado a estabelecer a arquitetura, uma vez definidos os objetivos gerais do produto	5	10	17	25	33	40
Percentual de arquitetos de software experientes (top) disponíveis para o projeto em relação ao considerado necessário	20	40	60	80	100	120
Suporte de ferramentas disponível para resolver itens de risco, desenvolver e verificar especificações arquiteturais	Nenhum	Pouco	Algum	Bom	Forte	Total
Nível de incerteza nos determinantes-chave da arquitetura: missão, interface com usuário, COTS, hardware, tecnologia, desempenho	Extremo	Significativo	Considerável	Algum	Pouco	Muito pouco
Número de itens de risco e sua importância	Mais de 10 críticos	5 a 10 críticos	2 a 4 críticos	1 crítico	Mais de 5 não críticos	Menos de 5 não críticos

RESOLUÇÃO DE RISCOS

Nota média	Muito baixo	Baixo	Nominal	Alto	Muito alto	Extra-alto
Interpretação	Pouco (20%)	Algum (40%)	Frequente (60%)	Geralmente (75%)	Largamente (90%)	Totalmente (100%)
Fator numérico	7,07	5,65	4,24	2,83	1,41	0,00



COESÃO DA EQUIPE DE DESENVOLVIMENTO

Tabela 7-10: Forma de obtenção do equivalente numérico para TEAM.

Característica	Muito baixo	Baixo	Nominal	Alto	Muito alto	Extra-alto
Consistência dos objetivos e cultura dos interessados	Pouca	Alguma	Básica	Considerável	Forte	Total
Habilidade e vontade dos interessados em acomodar os objetivos de outros interessados	Pouca	Alguma	Básica	Considerável	Forte	Total
Experiência dos interessados em trabalhar como uma equipe	Nenhuma	Pouca	Pouca	Básica	Considerável	Extensiva
Construção de equipes com os interessados para obter visão compartilhada e compromissos	Nenhuma	Pouca	Pouca	Básica	Considerável	Extensiva

Nota média	Muito baixo	Baixo	Nominal	Alto	Muito alto	Extra-alto
Interpretação	Interações muito difíceis	Algumas interações difíceis	Interações basicamente cooperativas	Predominantemente cooperativas	Altamente cooperativas	Interações perfeitas
Fator numérico	5,48	4,38	3,29	2,19	1,10	0,00

MATURIDADE DO PROCESSO

Tabela 7-11: Forma de obtenção do equivalente numérico para PMAT.

Característica	Muito baixo	Baixo	Nominal	Alto	Muito alto	Extra-alto
Nível CMM (ou CMMI)	1 – inferior	1 – superior	2	3	4	5
Nível EPML (ou SPICE)	0	1	2	3	4	5

Nota média	Muito baixo	Baixo	Nominal	Alto	Muito alto	Extra-alto
Interpretação	Sem processo definido	Processo incipiente	Processo definido	Processo gerenciado	Processo padronizado gerenciado quantitativamente	Processo em otimização constante
Fator numérico	7,80	6,24	4,68	3,12	1,56	0,00



MULTIPLICADORES DE ESFORÇO

$$E = A * KSLOC^S * \prod_{i=1}^n M_i \leftarrow$$

$$S = B + 0,001 * \sum_{j=1}^5 F_j$$

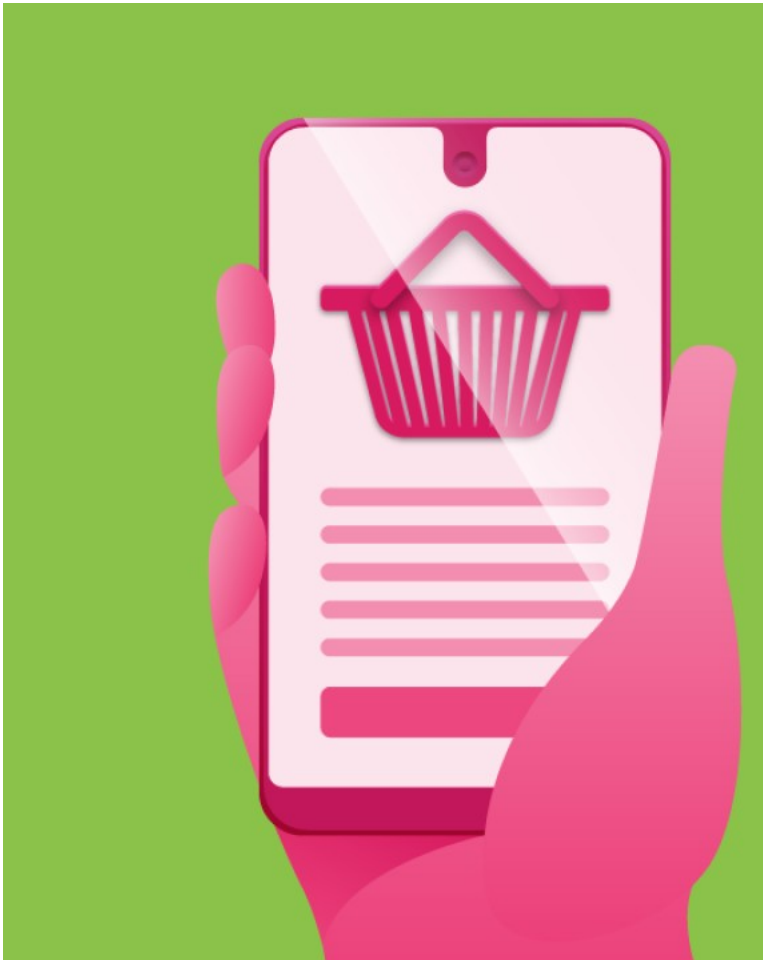


MULTIPLICADORES DE ESFORÇO DO POST-ARCHITECTURE MODEL

- Fatores do Produto.
- Fatores da Plataforma.
- Fatores Humanos.
- Fatores de Projeto.



FATORES DO PRODUTO



- *Software com Confiabilidade Requerida (RELY).*
- *Tamanho da Base de Dados (DATA).*
- *Complexidade do Produto (CPLX).*
- *Desenvolvimento Visando Reusabilidade (RUSE).*
- *Documentação Necessária para o Ciclo de Desenvolvimento (DOCU).*



CONFIABILIDADE REQUERIDA

Tabela 7-12: Forma de obtenção do equivalente numérico para RELY.

Descritor	Pequena inconveniência	Perdas pequenas, facilmente recuperáveis	Perdas moderadas, facilmente recuperáveis	Alta perda financeira	Risco a vida humana	
Avaliação	Muito baixo	Baixo	Nominal	Alto	Muito Alto	Extra-alto
Equivalente numérico	0,82	0,92	1,00	1,10	1,26	n/a

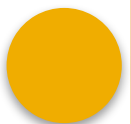


TAMANHO DA BASE DE DADOS

O multiplicador de esforço *DATA* (*Tamanho da Base de Dados*) avalia o tamanho relativo da base de dados usada *para testes* do programa (não a base de dados final). A razão D/P é o número de *Kbytes* na base de dados de teste (D) dividido pelo número de milhares de linhas (P) estimado do programa (em KSLOC). A Tabela 7-13 apresenta os parâmetros de cálculo para *DATA*.

Tabela 7-13: Forma de obtenção do equivalente numérico para *DATA*.

Descritor		$D/P < 10$	$10 \leq D/P \leq 100$	$100 \leq D/P \leq 1000$	$DP \geq 1000$	
Avaliação	Muito baixo	Baixo	Nominal	Alto	Muito Alto	Extra-alto
Equivalente numérico	n/a	0,90	1,00	1,14	1,28	n/a



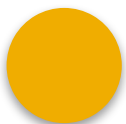
COMPLEXIDADE DO PRODUTO 1/2

Tabela 7-14: Forma de obtenção do equivalente numérico para CPLX.

Operações de controle	Código sequencial com poucas estruturas não aninhadas. Composição simples de módulos via chamada de procedimentos ou <i>scripts</i>	Aninhamento simples de estruturas de controle. Basicamente predicados simples.	Basicamente aninhamento simples. Algum controle intermódulos. Tabelas de decisão. Chamadas ou passagem de mensagens, incluindo processamento distribuído suportado por middleware.	Estruturas altamente aninhadas com vários predicados compostos. Controle de fila e pilha. Processamento distribuído homogêneo. Controle de tempo real simples em processador único.	Código reentrante e recursivo. Gerenciamento de interrupção com prioridade fixa. Sincronização de tarefas. Chamadas complexas. Processamento distribuído heterogêneo. Controle de tempo real complexo em processador único.	Escalonamento de múltiplos recursos com mudança dinâmica de prioridades. Controle em nível de micro código. Controle complexo de tempo real distribuído.
Operações computacionais	Avaliação de expressões simples como $A := B + C * (D - E)$	Avaliação de expressões de nível moderado como $D := \text{SQRT}(B ** 2 - 4 * A * C)$	Uso de rotinas matemáticas e estatísticas padrão. Operações básicas sobre matrizes e vetores.	Análise numérica básica: interpolação multivariada e equações diferenciais ordinárias. Arredondamento e truncamento básicos.	Análise numérica complexa, mas estruturada: equações de matrizes, equações diferenciais parciais. Paralelização simples.	Análise numérica complexa e não estruturada: análise de ruído altamente precisa, dados estocásticos. Paralelização complexa.

COMPLEXIDADE DO PRODUTO 2/2

Operações dependentes de dispositivo	Comandos simples de leitura e escrita com formatação simples.	Sem necessidade de conhecimento de características particulares de processador ou dispositivo de E/S. E/S feita por Get e Put.	Processamento de E/S inclui seleção de dispositivo, checagem de <i>status</i> e processamento de erros.	Operações de E/S em nível físico (traduções de endereços de armazenamento físicos; buscas e leituras, etc.). Overlap de E/S otimizado.	Rotinas para diagnóstico de interrupção. Gerenciamento de linha de comunicação. Sistemas embarcados com consideração intensiva de performance.	Codificação de dispositivos dependentes de tempo. Operações microprogramadas. Sistemas embutidos com performance crítica.
Operações de gerenciamento de dados	Arrays simples em memória. Simples consultas e atualizações em COTS ou banco de dados.	Arquivos simples sem edição nem buffers. Consultas e atualizações em bancos de dados ou COTS moderadamente complexas.	Entrada de múltiplos arquivos e saída em arquivo único. Mudanças estruturais simples. Edição simples. Consultas e atualizações complexas em COTS ou banco de dados.	Gatilhos simples ativados pelo conteúdo de sequências de dados. Reestruturação de dados complexa.	Coordenação de bancos de dados distribuídos. Gatilhos complexos. Otimização.	Estruturas relacionais e de objetos dinâmicas e altamente acopladas. Gerenciamento de dados em linguagem natural.
Operações de gerenciamento de interface com usuário	Formulários de entrada simples e geradores de relatórios.	Uso de construtores de interface com usuário (GUI) simples	Simples uso de um conjunto de <i>widgets</i> .	Desenvolvimento e extensão de conjunto de <i>widgets</i> . E/S por voz. Multimídia.	Gráficos dinâmicos 2D e 3D moderadamente complexos. Multimídia.	Multimídia complexa. Realidade virtual. Interface em linguagem natural.
Avaliação	Muito baixo	Baixo	Nominal	Alto	Muito Alto	Extra-alto
Equivalente numérico	0,73	0,87	1,00	1,17	1,34	1,74



DESENVOLVIMENTO VISANDO REUSABILIDADE

Tabela 7-15: Forma de obtenção do equivalente numérico para RUSE.

Descritor		Nenhum reuso	Dentro do projeto	Dentro de um programa	Dentro de uma <i>SPL</i>	Entre múltiplas <i>SPLs</i>
Avaliação	Muito baixo	Baixo	Nominal	Alto	Muito Alto	Extra-alto
Equivalente numérico	n/a	0,95	1,00	1,07	1,15	1,24



DOCUMENTAÇÃO NECESSÁRIA PARA O CICLO DE DESENVOLVIMENTO

Tabela 7-16: Forma de obtenção do equivalente numérico para DOCU.



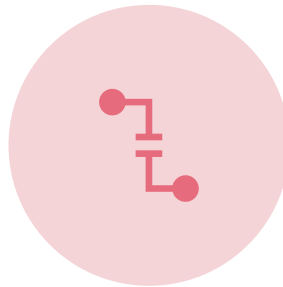
Descritor	Muitas necessidades de ciclo de vida não cobertas	Algumas necessidades de ciclo de vida não cobertas	Exatamente dimensionada para as necessidades do ciclo de vida	Excessiva para as necessidades do ciclo de vida	Muito excessiva para as necessidades do ciclo de vida	
Avaliação	Muito baixo	Baixo	Nominal	Alto	Muito Alto	Extra-alto
Equivalente numérico	0,81	0,91	1,00	1,11	1,23	n/a



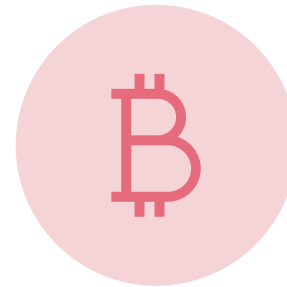
MULTIPLICADORES DE ESFORÇO REFERENTES À PLATAFORMA



*RESTRIÇÃO DE
TEMPO DE
EXECUÇÃO (TIME).*



*RESTRIÇÃO DE
MEMÓRIA
PRINCIPAL (STOR).*



*VOLATILIDADE DA
PLATAFORMA
(PVOL).*



RESTRIÇÃO DE TEMPO DE EXECUÇÃO

Tabela 7-17: Forma de obtenção do equivalente numérico para TIME.

Descritor			Menos de 50% de uso do tempo de execução disponível	70% de uso do tempo de execução disponível	85% de uso do tempo de execução disponível	95% de uso do tempo de execução disponível.
Avaliação	Muito baixo	Baixo	Nominal	Alto	Muito Alto	Extra-alto
Equivalente numérico	n/a	n/a	1,00	1,11	1,29	1,63



RESTRIÇÃO DE MEMÓRIA PRINCIPAL

Tabela 7-18: Forma de obtenção do equivalente numérico para STOR.



Descritor			Menos de 50% de uso da memória principal	70% de uso da memória principal	85% de uso da memória principal	95% de uso da memória principal
Avaliação	Muito baixo	Baixo	Nominal	Alto	Muito Alto	Extra-alto
Equivalente numérico	n/a	n/a	1,00	1,05	1,17	1,46



VOLATILIDADE DA PLATAFORMA

Tabela 7-19: Forma de obtenção do equivalente numérico para PVOL.

Descritor		Mudanças grandes a cada 12 meses, pequenas a cada mês	Grandes: 6 meses; pequenas: 2 semanas	Grandes: 2 meses; pequenas: 1 semana	Grandes: 2 semanas; pequenas: 2 dias	
Avaliação	Muito baixo	Baixo	Nominal	Alto	Muito Alto	Extra-alto
Equivalente numérico	n/a	0,87	1,00	1,15	1,30	n/a



FATORES HUMANOS



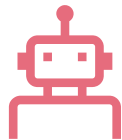
*Capacidade dos
Analistas (ACAP).*



*Capacidade dos
Programadores
(PCAP).*



*Continuidade de
Pessoal (PCON).*



*Experiência em
Aplicações
Semelhantes
(APEX).*



*Experiência na
Plataforma (PLEX).*



*Experiência na
Linguagem e
Ferramentas
(LTEX).*



CAPACIDADE DOS ANALISTAS

Tabela 7-20: Forma de obtenção do equivalente numérico para ACAP.

Descritor	Percentil 15	35	55	75	90	
Avaliação	Muito baixo	Baixo	Nominal	Alto	Muito Alto	Extra-alto
Equivalente numérico	1,42	1,19	1,00	0,85	0,71	n/a



CAPACIDADE DOS PROGRAMADORES

Tabela 7-21: Forma de obtenção do equivalente numérico para PCAP.

Descritor	Percentil 15	35	55	75	90	
Avaliação	Muito baixo	Baixo	Nominal	Alto	Muito Alto	Extra-alto
Equivalente numérico	1,34	1,15	1,00	0,88	0,76	n/a



CONTINUIDADE DE PESSOAL

Tabela 7-22: Forma de obtenção do equivalente numérico para PCON.

Descritor	48%/ano	24%/ano	12%/ano	6%/ano	3%/ano	
Avaliação	Muito baixo	Baixo	Nominal	Alto	Muito Alto	Extra-alto
Equivalente numérico	1,29	1,12	1,00	0,90	0,81	n/a



EXPERIÊNCIA EM APLICAÇÕES SEMELHANTES

Tabela 7-23: Forma de obtenção do equivalente numérico para APEX.

Descritor	Menos de 2 meses	6 meses	1 ano	3 anos	6 anos	
Avaliação	Muito baixo	Baixo	Nominal	Alto	Muito Alto	Extra-alto
Equivalente numérico	1,22	1,10	1,00	0,88	0,81	n/a



EXPERIÊNCIA NA PLATAFORMA

Tabela 7-24: Forma de obtenção do equivalente numérico para PLEX.



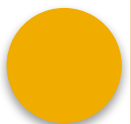
Descritor	Menos de 2 meses	6 meses	1 ano	3 anos	6 anos	
Avaliação	Muito baixo	Baixo	Nominal	Alto	Muito Alto	Extra-alto
Equivalente numérico	1,19	1,09	1,00	0,91	0,85	n/a



EXPERIÊNCIA NA LINGUAGEM E FERRAMENTAS

Tabela 7-25: Forma de obtenção do equivalente numérico para LTEX.

Descritor	Menos de 2 meses	6 meses	1 ano	3 anos	6 anos	
Avaliação	Muito baixo	Baixo	Nominal	Alto	Muito Alto	Extra-alto
Equivalente numérico	1,20	1,09	1,00	0,91	0,84	n/a



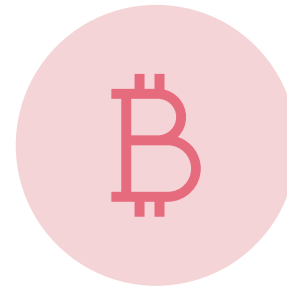
FATORES DE PROJETO



*USO DE
FERRAMENTAS DE
SOFTWARE (TOOL).*



*EQUIPE DE
DESENVOLVIMENTO
DISTRIBUÍDA (SITE).*



*CRONOGRAMA DE
DESENVOLVIMENTO
REQUERIDO (SCED).*



USO DE FERRAMENTAS DE SOFTWARE

Tabela 7-26: Forma de obtenção do equivalente numérico para TOOL.

Descritor	Editar, codificar, debugar.	CASE simples. Pouca integração.	Ferramentas básicas de ciclo de vida moderadamente integradas.	Ferramentas de ciclo de vida fortes e maduras, moderadamente integradas	Ferramentas de ciclo de vida fortes, maduras e bem integradas com processos, métodos e reuso	
Avaliação	Muito baixo	Baixo	Nominal	Alto	Muito Alto	Extra- alto
Equivalente numérico	1,17	1,09	1,00	0,90	0,78	n/a



EQUIPE DE DESENVOLVIMENTO DISTRIBUÍDA

Tabela 7-27: Forma de obtenção do equivalente numérico para SITE.

Descritor de co-locação	Internacional	Multi-cidade e multi-empresa	Multi-cidade ou multi-empresa	Mesma cidade ou área metropolitana	Mesmo edifício ou complexo	Totalmente co-locada
Descritor de comunicação	Alguns telefones, correio	Telefones individuais, FAX	Email	Comunicação eletrônica de banda larga	Videoconferência	Multimídia interativa
Avaliação	Muito baixo	Baixo	Nominal	Alto	Muito Alto	Extra-alto
Equivalente numérico	1,22	1,09	1,00	0,93	0,86	0,80



CRONOGRAMA DE DESENVOLVIMENTO REQUERIDO

Tabela 7-28: Forma de obtenção do equivalente numérico para SCED.

Descritor	75% do tempo nominal	85%	100%	130%	160%	
Avaliação	Muito baixo	Baixo	Nominal	Alto	Muito Alto	Extra-alto
Equivalente numérico	1,43	1,14	1,00	1,00	1,00	n/a

FATORES DO EARLY DESIGN MODEL



*Capacidade de
Pessoal (PERS).*



*Confiabilidade e
Complexidade do
Produto (RCPX).*



*Desenvolvimento
para Reuso
(RUSE).*



*Dificuldade com a
Plataforma
(PDIF).*



*Experiência do
Pessoal (PREX).*



*Instalações
(FCIL).*



*Cronograma de
Desenvolvimento
Requerido
(SCED).*

ESFORÇO E TEMPO POR FASE RUP

Fase	Esforço nominal	Intervalo	Tempo linear nominal	Intervalo
Concepção	0,06 <i>E</i>	0,02 a 0,15	0,125 <i>T</i>	0,02 a 0,3
Elaboração	0,24 <i>E</i>	0,20 a 0,28	0,375 <i>T</i>	0,33 a 0,42
Construção	0,76 <i>E</i>	0,72 a 0,80	0,625 <i>T</i>	0,58 a 0,67
Transição	0,12 <i>E</i>	0,00 a 0,20	0,125 <i>T</i>	0,00 a 0,20
Totais	1,18<i>E</i>		1,25<i>T</i>	



EXEMPLO

Assim, por exemplo, um projeto com $E = 56$ desenvolvedor-mês e $T = 11,5$ meses terá, em média os valores de esforço (em desenvolvedor-mês) e duração por fase definidos como na Tabela 7-35.

Tabela 7-35: Exemplo de cálculo de tempo e esforço para as fases do UP de um projeto com $E = 56$ e $T = 11,5$.

Fase	Esforço (desenvolvedor-mês)	Tempo (meses)
Concepção	3,4	1,4
Elaboração	13,4	4,3
Construção	42,6	7,2
Transição	6,7	1,4
Totais	66,1	14,3



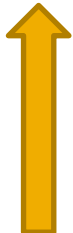
ESFORÇO POR DISCIPLINA

Tabela 7-36: Resumo do esforço relativo às disciplinas UP nas diferentes fases.

Disciplina	Concepção	Elaboração	Construção	Transição
Gerenciamento	14%	12%	10%	14%
Ambiente/Configuração	10%	8%	5%	5%
Requisitos	38%	18%	8%	4%
Design	19%	36%	16%	4%
Implementação	8%	13%	34%	19%
Avaliação/Teste	8%	10%	24%	24%
Implantação	3%	3%	3%	30%
Total	100%	100%	100%	100%



CALIBRAGEM DO MODELO

$$E = A * KSLOC^S * \prod_{i=1}^n M_i$$




TENDO REALIZADO PELO MENOS 5 PROJETOS...

Tabela 7-37: Exemplo de calibragem para a constante A.

<i>Real</i>	$KSLOC^S * \prod_{i=1}^n M_i$	$\ln(Real)$	$\ln(KSLOC^S * \prod_{i=1}^n M_i)$	$\ln(Real) - \ln(KSLOC^S * \prod_{i=1}^n M_i)$
1854,6	686,7	7,53	6,53	0,99
258,5	94,3	5,55	4,55	1,01
201,0	77,7	5,30	4,35	0,95
58,9	20,3	4,08	3,01	1,07
9661,0	3338,8	9,18	8,11	1,06
7021,3	2753,5	8,86	7,92	0,94
91,7	38,9	4,52	3,66	0,86
689,7	301,1	6,54	5,71	0,83
				X = 0,96
				A = 2,62

$$A = e^X$$

