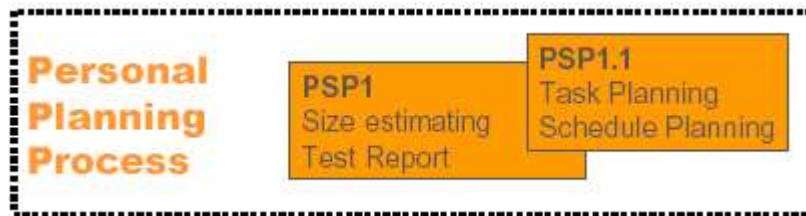


PERSONAL PLANNING PROCESS - PSP1 E PSP1.1



1 PSP1

O *Personal Planning Process*, representado pelos níveis PSP1 e PSP1.1, dedica-se a auxiliar o desenvolvedor a melhorar o seu perfil de planejamento pessoal. Para conseguir efetuar um planejamento adequado é necessário primeiro conhecer o tamanho da atividade, qualquer que seja a sua natureza. Portanto, o PSP1 dedica-se a estabelecer uma forma de estimar o tamanho da atividade, ou seja, uma forma para estimar o tamanho do programa a ser desenvolvido.

1.1 A importância do planejamento

A primeira grande importância do planejamento é facilitar o gerenciamento efetivo de projeto. Isto significa dizer que, uma vez estabelecido um plano e acompanhada a sua execução, pode ter uma idéia clara se o projeto está sendo desenvolvido dentro do prazo e do custo previstos.

Prazos cumpridos são equivalentes a acordos cumpridos. Para a indústria de *software* isto é muito importante, pois significa a conquista da confiança do cliente e, conseqüentemente, geração de lucros.

Mas afinal, o que é um plano?

“Um plano de projeto define o trabalho e como ele será feito. Ele provê uma definição das principais atividades, uma estimativa de tempos e recursos requeridos e um *framework* para revisão e controle gerenciais. O plano de projeto é também um poderoso instrumento de aprendizagem. Quando adequadamente documentado, funciona como um *benchmark* com o desempenho real. Esta comparação permite aos planejadores analisar seus erros de estimativa e melhorar a sua precisão de estimativas.” [HUM 95].

Um plano deve conter todas as informações necessárias para satisfazer os seus usuários. Os usuários de um plano de desenvolvimento de *software* são o próprio desenvolvedor e o cliente. Para o desenvolvedor, é importante conhecer o tamanho, a estrutura e o status da atividade. Para ele o plano serve também como um importante instrumento de auto-avaliação.

Já o cliente deseja saber o que ele receberá, quando e a que custo. É normal que o cliente queira ter meios de acompanhar o progresso do projeto, identificando se ele está em dia ou atrasado.

Para fazer um plano, o desenvolvedor deve primeiramente definir claramente os objetivos do projeto e os requisitos do usuário. Em seguida, deve dividir o trabalho em tarefas e estimar cada uma delas separadamente de acordo com o histórico de projetos anteriores. O resultado deverá ser um plano claro e bem documentado.

Para checar um plano o desenvolvedor deve verificar se ele:

- está completo: todas as considerações importantes para o desenvolvedor e o cliente estão previstas e documentadas (produtos, prazos, etc.) ?
- é acessível: é fácil de ser encontrado, validado e acompanhado ?
- é claro: é de fácil compreensão pelos seus usuários ?
- é específico: estabelece claramente o quê, quem, quando ?
- é adequado: tem a granularidade adequada ?
- é preciso: o planejado é próximo ao real ?

1.2 O método de estimativas PROBE

O PSP possui uma forma própria de estimar tamanho de programa que é o método PROBE – *PROxy Based Estimating*. A concepção do PROBE foi baseada em outros métodos de estimativas como: o método da lógica *Fuzzy* (Lawrence Putnam), o método do componente padrão (Lawrence Putnam) e o Wideband-Delphi (Barry Boehm). Além disto, também foi baseado nos princípios estatísticos (regressão linear, intervalo de previsão, distribuição normal etc.).

O objetivo do PROBE é estabelecer um procedimento ordenado e repetível para estimar o tamanho de *software*. Para auxiliar o desenvolvedor na sua utilização, o PSP oferece o formulário *C39 – Size Estimating Template*.

Conforme ilustra a Figura 1, o PROBE usa objetos como *proxies*. Primeiramente o desenvolvedor produz um projeto conceitual do programa, identificando os possíveis objetos que o comporão. A intenção é que este seja mesmo um projeto conceitual. Ou seja, deve servir apenas para o propósito de estimar, não havendo uma obrigação em realmente implementá-lo.

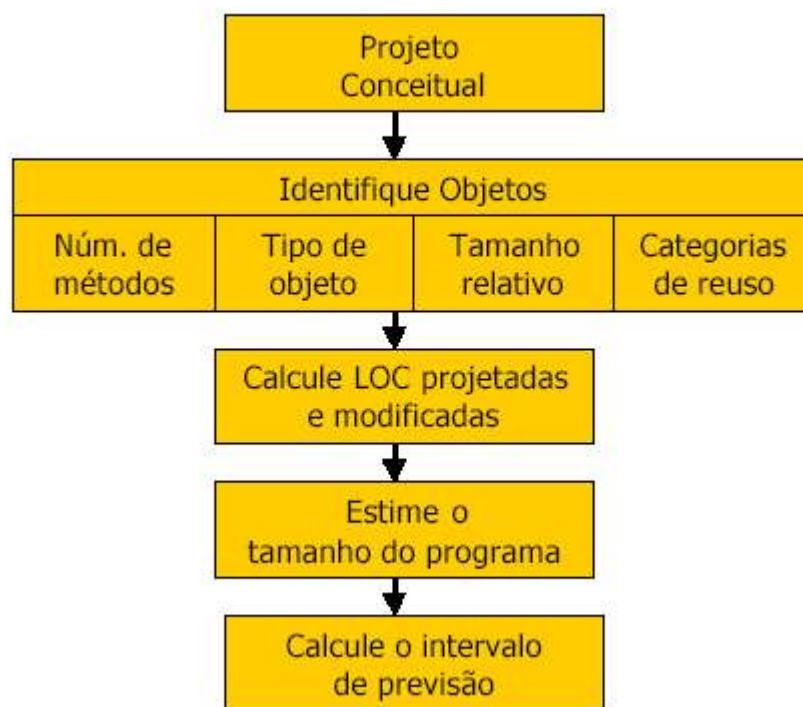


Figura 1: Estrutura do PROBE

Em seguida, para cada um dos objetos identificados no projeto conceitual, o desenvolvedor deverá identificar o tipo do objeto, a categoria de tamanho em que ele se encaixa (muito pequena, pequena, média, grande ou muito grande) e de quantos métodos será composto. Em seguida, deverá consultar o tamanho médio de cada método em uma tabela que deve ser proveniente do seu histórico de desenvolvimento. A Tabela 1 ilustra o tamanho dos métodos em LOC para objetos de diferentes tipos desenvolvidos em C++. O equivalente para Object Pascal é mostrado na Tabela 2.

É importante observar que estes números devem ser provenientes do histórico do próprio desenvolvedor para que possa refletir suas práticas de programação. No caso de linguagens procedurais, se utilizaria o tamanho de funções ao invés de métodos. Em ambos os casos, pressupõem-se um desenvolvimento modularizado. Para montar a sua própria tabela, refletindo suas práticas de programação, consulte [HUM 95].

	Muito Pequeno	Pequeno	Médio	Grande	Muito Grande
Cálculo	2.34	5.13	11.25	24.66	54.04
Dados	2.60	4.79	8.84	16.31	30.09
I/O	9.01	12.06	16.15	21.62	28.93
Lógica	7.55	10.98	15.98	23.25	33.83
Set-up	3.88	5.04	6.56	8.53	11.09
Texto	3.75	8.00	17.07	36.41	77.66

Tabela 1: C++

	Muito Pequeno	Pequeno	Médio	Grande	Muito Grande
Controle	4.24	8.68	17.79	36.46	74.71
Display	4.72	6.35	8.55	11.50	15.46
Arquivo	3.30	6.23	11.74	22.14	41.74
Lógica	6.41	12.42	24.06	46.60	90.27
Impressão	3.38	5.86	10.15	17.59	30.49
Texto	4.63	8.73	16.48	31.09	58.62

Tabela 2: Object Pascal

Ao somar todos os tamanhos obtidos para os objetos, o desenvolvedor terá o tamanho total dos objetos que comporão o programa. Se ele estiver reutilizando um programa pronto, então deverá somar ao total dos objetos, o total do programa a ser utilizado, bem como a quantidade de linhas que se pretende modificar. Neste momento terá então obtido o tamanho total do programa em termos de seus objetos.

Para fazer o ajuste no tamanho total de objetos y e obter o tamanho estimado do programa x , se o índice de correlação permitir, o desenvolvedor poderá aplicar a técnica estatística de regressão linear representada na Equação 1.

$$x = \beta_0 + \beta_1 y$$

Equação 1: Regressão Linear

Os parâmetros da regressão linear β_1 e β_0 são calculados com base no histórico de estimativas de tamanho de programa (x_i) e no histórico dos dados reais de tamanho de programa (y_i), através da Equação 2 e da Equação 3, respectivamente.

$\beta_1 = \frac{\sum_{i=1}^n x_i y_i - n x_{méd} y_{méd}}{\sum_{i=1}^n x_i^2 - n (x_{méd})^2}$	<p>Onde:</p> <p>x_i : dados históricos de tamanho estimado de programa</p> <p>y_i : dados históricos de tamanho real de programa</p> <p>n : qtde de elementos da base histórica</p> <p>$x_{méd}$: média dos dados históricos de tamanho estimado de programa</p> <p>$y_{méd}$: média dos dados históricos de tamanho real de programa</p>
-------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Equação 2: Cálculo do parâmetro β_1

$\beta_0 = y_{méd} - \beta_1 x_{méd}$	<p>Onde:</p> <p>$x_{méd}$: média dos dados históricos de tamanho estimado de programa</p> <p>$y_{méd}$: média dos dados históricos de tamanho real de programa</p>
---------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Equação 3: Cálculo do parâmetro β_0 .

Uma vez calculado o tamanho estimado total para o programa, é importante saber qual o erro provável desta estimativa. Isto é feito através do cálculo do intervalo de previsão, conforme a Equação 4. Note que o valor de t é obtido através da tabela da distribuição normal t para a probabilidade $\alpha/2$ e $n-2$ graus de liberdade. O intervalo de previsão poderá ser obtido para 70 ou 90% e significa, na realidade, a quantidade de LOC a maior ou a menor, que pode variar a estimativa de tamanho calculada.

$Int = t(\alpha/2, n-2) \delta \sqrt{1 + \frac{1}{n} + \frac{(x_k - x_{méd})^2}{\sum_{i=1}^n (x_i - x_{méd})^2}}$	<p>Onde:</p> <p>x_k : novo valor estimado</p> <p>x_i : dados históricos de tamanho estimado de programa</p> <p>n : qtde de elementos da base histórica</p> <p>$x_{méd}$: média dos dados históricos de tamanho estimado de programa</p> <p>$t(\alpha/2, n-2)$: valor da distribuição t para 70 ou 90%</p>
-------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Equação 4: Cálculo do intervalo de previsão.

Exemplo: Um desenvolvedor efetuou o projeto conceitual de um programa em C++ e identificou que, além do programa base que seria usado sem modificações (222 LOC), seriam necessários mais 2 novos objetos:

- uma matriz de médio porte, que ele classificou como do tipo “dado” e que ele imagina que conterá 13 métodos;
- uma lista encadeada de grande porte, que ele também classificou como tipo “dado” e que ele imagina que conterá 3 métodos.

Como ele estimou o tamanho do novo programa se sua base histórica de estimativas é a seguinte:

Num Pgm.	LOC estimada	LOC real
1	130	186
2	650	699
3	99	132
4	150	272
5	128	291
6	302	331
7	95	199
8	945	1890
9	368	788
10	961	1601

Tabela 3:LOC estimadas x real.

Passo 1: estimar o tamanho dos novos objetos:

- ao consultar a Tabela 1, ele obteve que um objeto médio do tipo “dado” tem 8.84 LOC/método. Logo, a matriz terá $13 \times 8.84 = 114.92$ (115) LOC;
- também consultando a Tabela 1 ele chegou à conclusão que um objeto grande do tipo “dado” tem 16.31 LOC/método. Logo, a lista encadeada terá $3 \times 16.31 = 48.93$ (49) LOC.

Passo 2: obter o total de LOC projetadas:

- o tamanho total dos objetos seria então: $222 + 115 + 49 = 386$ LOC.

Passo 3: calcular os parâmetros da regressão linear β_1 e β_0 :

- utilizando os dados da Tabela 3, que representam as LOC estimadas e as LOC reais, e aplicando a Equação 2, o desenvolvedor obteve o parâmetro $\beta_1 = 1.7279$;
- da mesma forma, utilizando os dados da Tabela 3 e aplicando a Equação 3, ele obteve $\beta_0 = -22.55$.

Passo 4: calcular o tamanho estimado do programa:

- utilizando os parâmetros β_1 e β_0 calculados e aplicando a Equação 1, o desenvolvedor obteve o tamanho de 644 LOC para o novo programa.

Passo 5: calcular o intervalo de previsão:

- pesquisando a tabela da distribuição t para $(n-2)$ graus de liberdade, ou seja, $10 - 2 = 8$ graus de liberdade, e para um intervalo de previsão de 90% (para $\alpha/2$), o desenvolvedor obteve o valor 1.86;
- aplicando a Equação 4, o desenvolvedor obteve o intervalo de 386.05. Combinando este resultado com o tamanho obtido de 644 LOC, teríamos um intervalo de 258 a 1030 LOC.

1.3 O registro dos testes

O objetivo de se registrar os dados de teste é, primeiramente, manter o registro dos testes efetuados e dos resultados obtidos. Isto é, não apenas uma forma de documentação da fase e teste, como também uma maneira de realizar testes de regressão mais seguros.

O registro dos testes no PSP é feito através do formulário *C37 – Test Report Template*, ilustrado na Figura 2. Nele, é registrado o objetivo do teste, a descrição do teste, as condições do teste, os resultados esperados e os resultados reais.

Test Name/Number	Teste 001
Test Objective	Testar a entrada de dados – valores negativos
Test Description	Entrar com valor negativo para idade
Test Conditions	Sem condições especiais
Expected Results	Exibir mensagem : “Idade não pode ser negativa.”
Actual Results	Exibiu a mensagem: “Idade não pode ser zero.”

Figura 2: Registro dos Testes.

1.4 O Sumário de Projeto PSP1

Em relação ao nível anterior, o PSP1 introduz como mudanças no *Project Plan Summary*, a coluna referente ao planejamento do tamanho do programa, originária do método de estimativas PROBE, conforme a Figura 3. Além disto, acrescenta uma seção de sumário ue inicia com os dados da produtividade em termos de LOC/hora (planejada, real e acumulada).

<i>Summary LOC/Hour</i>	<i>Plan</i>	<i>Actual</i>	<i>To Date</i>
Program Size (LOC):	<i>Plan</i>	<i>Actual</i>	<i>To Date</i>
Base(B)	(Measured)	(Measured)	
Deleted (D)	(Estimated)	(Counted)	
Modified (M)	(Estimated)	(Counted)	
Added (A)	(N-M)	(T-B+D-R)	
Reused (R)	(Estimated)	(Counted)	
Total New & Changed (N)	(Estimated)	(A+M)	
Total LOC (T)	(N+B-M-D+R)	(Measured)	
Total New Reused			

Figura 3: *C34 - PSP1 Project Plan Summary – modificações.*

2 Exercício 2

Objetivo	Utilizar o processo PSP1 para desenvolver um programa simples, permitindo ao desenvolvedor aprender a estimar utilizando o método PROBE.
Descrição	Escreva um programa para calcular a variância de uma série de n números reais. Utilize como base o programa do exercício anterior.
Formulários	<i>C16 – Time Recording Log</i> <i>C18 – Defect Recording Log</i> <i>C27 – PIP – Process Improvement Proposal</i> <i>C34 – PSP1 Project Plan Summary</i> <i>C37 – Test Report Template</i> <i>C39 – Size Estimating Template</i>
Passo a Passo	<p>Planejamento:</p> <ul style="list-style-type: none"> ▪ Entre com a data e a hora no formulário <i>C16 – Time Recording Log</i> ▪ Efetue a estimativa das LOC novas e modificadas utilizando o formulário <i>C39 – Size Estimating Template</i> ▪ Efetue a estimativa das LOC base, modificadas, adicionadas e reutilizadas e registre na coluna <i>Plan</i> do formulário <i>C34 – PSP1 Project Plan Summary</i> ▪ Efetue a estimativa de tempo com base na quantidade de LOC/h do exercício anterior e registre na coluna <i>Plan</i> do formulário <i>C34 – PSP1 Project Plan Summary</i> (utilize o <i>% To Date</i> para basear a distribuição de tempo nas fases) ▪ Registre o encerramento da fase de planejamento no formulário <i>C16 – Time Recording Log</i> <p>Desenvolvimento:</p> <ul style="list-style-type: none"> ▪ Desenvolva o programa registrando os tempos e as fases no formulário <i>C16 – Time Recording Log</i> e os defeitos inseridos e removidos no formulário <i>C18 – Defect Recording Log</i> ▪ Efetue os testes do programa e registre os dados de teste no formulário <i>C37 – Test Report Template</i> ▪ Registre o encerramento da fase de teste no formulário <i>C16 – Time Recording Log</i> <p>Autópsia:</p> <ul style="list-style-type: none"> ▪ Some os tempos e os defeitos coletados e registre os totais nos campos apropriados do formulário <i>C34 – PSP1 Project Plan Summary</i> ▪ Conte o total de LOC do programa completo, determine as LOC deletadas, modificadas, adicionadas, reusadas e desenvolvidas para reuso e registre no formulário <i>C34 – Project Plan Summary</i>

Dicas	<p>Variância σ^2 de uma distribuição é a medida de quanto os valores se afastam da média. Uma variância pequena significa que a distribuição está concentrada em torno da média. Uma variância grande significa que a distribuição está mais esparsa em relação à média. A variância é calculada pela equação:</p> $\sigma^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - x_{med})^2$
Dados de Teste	Testar o programa com os dados da Tabela 8.

Núm Prog	LOC do Objeto	LOC novas e modif.	Horas de Desenv.
1	160	186	15.0
2	591	699	69.9
3	114	132	6.5
4	229	272	22.4
5	230	291	28.4
6	270	331	65.9
7	128	199	19.4
8	1657	1890	198.7
9	624	788	38.8
10	1503	1601	138.2
Variância	327218,32	656,90	3875,06

Tabela 8. Dados para teste do Exercício 2.

Table C16 - Time Recording Log

Student _____ Date _____
Instructor _____ Program # _____

[illegible]

Table C16 – Time Recording Log (instruções)

Cabeçalho:

- *Student*: nome do aluno // *Date*: data do início do registro
- *Instructor*: nome do professor // *Program #*: número do programa

Tabela:

- *Date*: data da atividade.
- *Start*: hora de início da atividade.
- *Stop*: hora de fim da atividade.
- *Interruption Time*: tempo total de interrupção da atividade.
- *Delta Time*: diferença entre o início e o fim da atividade, deduzido o tempo de interrupção.
- *Phase*: nome da fase (planejamento, código, teste, etc.).
- *Comments*: motivo das interrupções ou qualquer comentário que pareça importante

para análises futuras.

Table C18 – Defect Recording Log (instruções)

Cabeçalho:

- *Student*: nome do aluno // *Date*: data do início do registro
- *Instructor*: nome do professor // *Program #*: número do programa

Tabela:

- *Date*: data onde o defeito foi encontrado.
- *Number*: número seqüencial do defeito, iniciando em 001.
- *Type*: tipo do defeito de acordo com a tabela padrão de defeitos.
- *Inject*: fase na qual o defeito foi inserido.
- *Removed*: fase na qual o defeito foi removido.
- *Fix Time*: tempo, em minutos, gasto na correção do defeito.
- *Fix Defect*: se o defeito foi inserido enquanto removia outro defeito, entre com o número do defeito que estava sendo consertado.
- *Description*: descrição breve do defeito.

Table C18 - Defect Recording Log

Defect Types	
10 Documentation	60 Checking
20 Syntax	70 Data
30 Build, Package	80 Function
40 Assignment	90 System
50 Interface	100 Environment

Student _____ Date _____
 Instructor _____ Program # _____

Date	Number	Type	Inject	Remove	Fix Time	Fix Defect
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Description: _____

Date	Number	Type	Inject	Remove	Fix Time	Fix Defect
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Description: _____

Date	Number	Type	Inject	Remove	Fix Time	Fix Defect
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Description: _____

Date	Number	Type	Inject	Remove	Fix Time	Fix Defect
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Description: _____

Date	Number	Type	Inject	Remove	Fix Time	Fix Defect
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Description: _____

Date	Number	Type	Inject	Remove	Fix Time	Fix Defect
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Description: _____

Date	Number	Type	Inject	Remove	Fix Time	Fix Defect
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Description: _____

Date	Number	Type	Inject	Remove	Fix Time	Fix Defect
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Description: _____

Table C27 - Process Improvement Proposal (PIP)

Student	_____	Date	_____
Instructor	_____	Program #	_____
Process	_____	Elements	_____

PROBLEM	
PIP Number	Problem Description:
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____

PROPOSAL	
PIP Number	Proposal Description:
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____

Notes and Comments:

Table C27 – Process Improvement Proposal (PIP) – (instruções)**Cabeçalho:**

- *Student*: nome do aluno // *Date*: data do início do registro
- *Program #*: número do programa // *Instructor*: nome do professor
- *Process*: nome do processo que está sendo utilizado (Ex.: PSP0.1)
- *Element*: Se a proposta disser respeito a um elemento específico do processo que está sendo utilizado (Ex.: PSP0.1 Project Plan Summary)

Problema:

- *PIP Number*: número sequencial do problema encontrado.
- *Problem Description*: Descrição, mais clara possível, do problema encontrado, bem como os impactos no processo, no produto e em você.

Proposta:

- *PIP Number*: número sequencial correspondente ao problema para o qual está sendo proposta a solução.
- *Proposal Description*: Descrição, mais clara possível da melhoria de processo proposta. Se for possível, identifique o elemento do processo e as palavras ou entradas a serem modificadas. Se achar que a melhoria proposta é particularmente importante ou prioritária, indique porque.

Notas e Comentários:

- *Notes and Comments*: Para cada projeto complete pelo menos uma PIP com as anotações e comentários sobre o processo: lições aprendidas e anotações que podem auxiliar a lembrar porque o processo funcionou bem ou mal.

Table C34 - PSP1 Project Plan Summary

Student			Date	
Program			Program #	
Instructor			Language	

Summary	Plan	Actual	To Date	
LOC/Hour				
Program Size (LOC):	Plan	Actual	To Date	
Base(B)	(Measured)	(Measured)		
Deleted (D)	(Estimated)	(Counted)		
Modified (M)	(Estimated)	(Counted)		
Added (A)	(N-M)	(T-B+D-R)		
Reused (R)	(Estimated)	(Counted)		
Total New & Changed (N)	(Estimated)	(A+M)		
Total LOC (T)	(N+B-M-D+R)	(Measured)		
Total New Reused				
Time in Phase (min.)	Plan	Actual	To Date	To Date %
Planning				
Design				
Code				
Compile				
Test				
Postmortem				
Total				
Defects Injected		Actual	To Date	To Date %
Planning				
Design				
Code				
Compile				
Test				
Total Development				
Defects Removed		Actual	To Date	To Date %
Planning				
Design				
Code				
Compile				
Test				
Total Development				
After Development				

Table C34 - PSP1 Project Plan Summary (instruções)

Observações: Seguir as mesmas instruções de preenchimento do formulário *C25 – PSP0.1 Project Plan Summary*, acrescidas das informações relativas ao planejamento do tamanho do

programa, introduzidas pelo *PSP1*, conforme descrito abaixo.

Summary:

- **LOC/Hour - Plan:** produtividade planejada, calculada usando os valores planejados de (new + changed LOC)/hora (antes do desenvolvimento).
- **LOC/Hour - Actual:** produtividade real, calculada usando os valores reais de (new + changed LOC)/hora (após desenvolvimento).
- **LOC/Hour - To Date:** produtividade acumulada, calculada usando (soma do LOC/hour To Date do programa mais recente com o LOC/Hour Actual deste programa).

Program Size (antes do desenvolvimento):

- **Base(B) – Plan:** se estiver modificando ou melhorando um programa existente, entrar com o número total de LOC do programa base
- **Estimated Object LOC (E) - Plan:** entre com o tamanho estimado para o objeto LOC(E) obtido a partir do *Size Estimating Template*
- **Total New and Changed (N) - Plan:** entre com o valor LOC (N) obtido a partir do *Size Estimating Template*
- **Added (A) – Plan:** estime o valor de LOC a serem adicionadas de modo que $N=A+M$
- **Modified (M) – Plan:** estime o valor de LOC a serem modificadas de modo que $N=A+M$
- **Deleted (D) – Plan:** estime o valor de LOC a serem deletadas e combine com o valor de LOC do programa base (B) de modo que $T=N+B-M-D+R$
- **Reused (R) – Plan:** estime o valor de LOC a serem deletadas e combine com o valor de LOC do programa base (B) de modo que $T=N+B-M-D+R$

Program Size (depois do desenvolvimento):

- **Base(B):** se a quantidade de LOC mudou, entre com o novo valor.
- **Total LOC (T):** quantidade total de LOC após o desenvolvimento.
- **Deleted (D), Modified (M) e Reused (R):** quantidade de LOC deletadas (D), modificadas (M) e reutilizadas (R) do programa após o desenvolvimento – podem ser obtidos utilizando o programa 3A.
- **Added (A):** Calcule o total de linhas adicionadas ao programa $A=T-B+D-R$
- **Total New and Changed (N) – Actual:** Calcule o total de LOC novas e modificadas $N=A+M$

Table C36 – PROBE Estimating Script

	Propósito	Guiar o processo de estimativas PROBE
	Critério de entrada	Declaração de requisitos <i>Size Estimating Template</i> e instruções Dados de LOC por método para os tipos de objetos Time Recording Log Dados históricos de tamanho e tempo
	Projeto Conceitual	Reveja os requisitos e produza um projeto conceitual
	Estime objetos	Siga as instruções contidas no <i>Size Estimating Template</i> para estimar as LOC dos objetos novos e as LOC reusadas novas.
	Outras LOC de programa	Siga as instruções contidas em <i>Size Estimating Template</i> para estimar as LOC base, deletadas, modificadas, adicionadas e reusadas.
	Procedimentos para estimativa de tamanho	A – se você possuir dados suficientes de LOC de objetos estimadas e dados reais de LOC novas e modificadas (três ou mais pontos correlacionados), use o procedimento 4A B – se você não possuir dados suficientes de LOC de objetos estimadas, mas tem dados suficientes de LOC novas e modificadas estimadas (três ou mais pontos correlacionados), use o procedimento 4B C – se você não tem dados suficientes ou se eles não estão correlacionados, use o procedimento 4C D – se você não tem dados históricos, use o procedimento 4D
4A	Procedimento de Estimativa de Tamanho 4A	Usando os procedimentos da tabela A27 (procedimentos para a regressão linear), calcule os parâmetros de regressão β_0 e β_1 a partir das LOC de objeto estimadas e LOC novas e modificadas reais. Se o parâmetro β_0 não estiver próximo de 0 (substancialmente menor do que o tamanho esperado do novo programa) ou o parâmetro β_1 não estiver razoavelmente próximo de 1 (entre 0.5 e 2.0), utilize o procedimento 4B.
4B	Procedimento de Estimativa de Tamanho 4B	Usando os procedimentos da tabela A27 (procedimentos para a regressão linear), calcule os parâmetros da regressão β_0 e β_1 a partir das LOC novas e modificadas estimadas e LOC novas e modificadas reais. Se o parâmetro β_0 não for próximo a 0 (substancialmente menor do que o tamanho esperado do novo programa) ou o parâmetro β_1 não estiver razoavelmente próximo de 1 (entre 0.5 e 2.0), utilize o procedimento 4C.

4C	Procedimento de Estimativa de Tamanho 4C	Se você tiver qualquer dado sobre LOC de objeto estimadas e LOC novas e modificadas reais, faça $\beta_0=0$ e $\beta_1 = (\text{total de LOC novas e modificadas acumulado}) / (\text{total de LOC de objeto estimadas acumulado})$
4D	Procedimento de Estimativa de Tamanho 4D	Se você não tiver dados históricos, use seu julgamento para estimar as LOC novas e modificadas a partir das LOC de objeto estimadas.
5	Procedimento para Estimativa de Tempo	A – se você possuir dados suficientes de LOC de objetos estimadas e tempo real de desenvolvimento (três ou mais pontos correlacionados), use o procedimento 5A B – se você não possuir dados suficientes de LOC de objetos estimadas, mas tem dados suficientes de LOC novas e modificadas estimadas (três ou mais pontos correlacionados), use o procedimento 5B C1 – se você tiver alguns dados de LOC de objeto estimadas e eles forem insuficientes ou não se correlacionarem, use o procedimento 5C1 C2 – se você tiver alguns dados de LOC novas e modificadas estimadas e eles forem insuficientes ou não se correlacionarem, use o procedimento 5C2 C3 – se você tiver apenas dados reais de LOC e modificadas, use o procedimento 5C3 D – se você não tiver dados históricos, use o procedimento 5D
5A	Procedimento para Estimativa de Tempo 5A	Usando os procedimentos da tabela A27 (procedimentos para a regressão linear), calcule os parâmetros de regressão β_0 e β_1 a partir das LOC de objeto estimadas e o tempo total real de desenvolvimento Se o parâmetro β_0 não for substancialmente menor do que o tempo de desenvolvimento esperado para o novo programa, ou o parâmetro β_1 não estiver razoavelmente próximo de $60/(\text{LOC acumuladas/hora})$ para o programa mais recente, use o procedimento 5C1
5B	Procedimento para Estimativa de Tempo 5B	Usando os procedimentos da tabela A27 (procedimentos para a regressão linear), calcule os parâmetros da regressão β_0 e β_1 a partir das LOC novas e modificadas estimadas e o tempo total real de desenvolvimento Se o parâmetro β_0 não for substancialmente menor do que o tempo de desenvolvimento esperado para o novo programa, ou o parâmetro β_1 não estiver razoavelmente próximo de $60/(\text{LOC acumuladas/hora})$ para o programa mais recente, use o procedimento 5C1
5C1	Procedimento para	Se você tiver qualquer dado sobre LOC de objeto estimadas

Table C37 - Test Report Template

Student	_____	Date	_____
Instructor	_____	Program #	_____

Test Name/Number	_____
Test Objective	_____
Test Description	_____

Test Conditions	_____

Expected Results	_____

Actual Results	_____

Test Name/Number	_____
Test Objective	_____
Test Description	_____

Test Conditions	_____

Expected Results	_____

Actual Results	_____

Table C37 – Test Report Template (instruções)

Cabeçalho:

- *Student*: nome do aluno // *Date*: data do início do registro
- *Program #*: número do programa // *Instructor*: nome do professor

Dados de Teste:

- **Test Name/Number**: identificar cada teste de forma única, prevendo: mesmos testes com dados diferentes, mesmos dados com testes diferentes, reproprocessamento de teste com correção de defeitos.
- **Test Objective**: descreva de forma sucinta o objetivo do teste.
- **Test Description**: descreva os dados e procedimentos de forma que o teste possa ser posteriormente reproprocessado.
- **Test Conditions**: liste qualquer configuração especial, momento, conserto ou outras

condições do teste. Se vários testes estão sendo executados para várias correções, liste cada um separadamente.

- **Expected results:** liste os resultados que deveriam ser produzidos pelo teste se ele processasse normalmente.
- **Actual results:** liste os resultados efetivamente produzidos. Quando o mesmo teste é rodado múltiplas vezes para correção de múltiplos defeitos, anote cada resultado separadamente.

Table C39 - Size Estimating Template

BASE PROGRAM				LOC
BASE SIZE (B)	=>	=>	=>	_____
LOC DELETED (D)	=>	=>	=>	_____
LOC MODIFIED (M)	=>	=>	=>	_____
PROJECTED LOC				
BASE ADDITIONS:	TYPE	METHODS	REL. SIZE	LOC
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
TOTAL BASE ADDITIONS (BA)		=>	=>	_____
NEW OBJECTS:	TYPE ¹	METHODS	REL. SIZE	LOC (NewReuse*)
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
TOTAL NEW OBJECTS (NO)		=>	=>	_____
REUSED PROGRAMS				LOC
_____				_____
_____				_____
REUSED TOTAL (R)		=>	=>	_____
Estimated Object LOC (E)		E = BA + NO + M		
Regression Parameter:		β_0 (size and time)		
Regression Parameter:		β_1 (size and time)		
Estimated New and Changed LOC (N)		N = $\beta_0 + \beta_1 * E$		
Estimated Total LOC:		T = N + B - D - M + R		
Estimated Total New Reuse (sum of * LOC):		_____		
Prediction Range:		Range		
Upper Prediction Interval:		UPI = N + Range		
Lower Prediction Interval:		LPI = N - Range		
Prediction Interval Percent:		_____		

¹ L-Logic, I-I/O, C-Calculation, T-Text, D-Data, S-Set-up

Tabela C39 – Size Estimating Template (instruções)

Cabeçalho:

- *Student:* nome do aluno // *Date:* data do início do registro
- *Program #:* número do programa (Ex.: 4A) // *Instructor:* nome do professor

Base Program LOC – estimate (antes do desenvolvimento):

- **Base Size (B) - estimate:** total de LOC do programa base (contado).
- **LOC deleted (D) - estimate:** total de LOC que serão deletadas do programa base.
- **LOC modified (M) - estimate:** total de LOC que serão modificadas no programa base.

Base Program LOC – actual (após o desenvolvimento):

- **Base Size (B) - actual:** total de LOC do programa base (contado).
- **LOC deleted (D) - actual:** total de LOC que foram efetivamente deletadas do programa base.
- **LOC modified (M) - actual:** total de LOC que foram efetivamente modificadas no programa base.

Object LOC – new objects (antes do desenvolvimento):

- **Base Additions:** nome dos objetos que se pretende adicionar ao programa já existente. Se estes objetos puderem ser tratados como novos objetos, usar os mesmos procedimentos de *new objects*.
- **New objects:** nome do objeto novo que se espera desenvolver de acordo com o projeto conceitual.
- **Type:** categoria do novo objeto (ver tipo na tabela 5.7: cálculo, dado, I/O, lógica, setup, texto).
- **Methods:** quantidade de métodos por objeto.
- **Rel. Size:** tamanho relativo do objeto (muito pequeno, pequeno, médio, grande, muito grande).
- **LOC - estimate:** total de LOC do objeto obtido multiplicando-se a quantidade de métodos x tamanho médio do método. Tamanho médio de cada método é obtido na tabela 5.7, levando-se em consideração o tipo do objeto e o tamanho relativo do objeto.
- **(*):** objetos que se pretende incorporar à biblioteca de reuso devem ter seu tamanho estimado marcado com um (*).
- **Total New Objects (NO):** soma do total de LOC de todos os objetos novos.

Reused Objects:

- Identificar o nome e o tamanho dos objetos a serem reusados sem alteração (Ex: lista encadeada, entrada de dados).
- **Reused Total (R):** total de LOC dos objetos reusados sem alteração.

Informações da regressão:

- **Estimated object (E):** Somar BA (base additions), NO (new objects) e M (LOC que se prevê modificar).
- **Regression parameters β_0 (size and time):** calcular o parâmetro de regressão β_0 (para tamanho e tempo) com base no histórico de desenvolvimento, conforme tabela A27 (procedimentos para regressão linear) e o script do PROBE.
- **Regression parameters β_1 (size and time):** calcular o parâmetro de regressão β_1 (para tamanho e tempo) com base no histórico de desenvolvimento, conforme tabela A27 (procedimentos para regressão linear) e o script do PROBE.
- **Estimated new and changed LOC (N):** $N = \beta_0 + \beta_1 * E$ (utilizar β_0 e β_1 calculados para tamanho).
- **Estimated Total LOC (T):** $T = N + B - D - M + R$.
- **Estimated Total New Reuse:** total de LOC estimadas a serem desenvolvidas para reuso posterior (inclusão na biblioteca de reuso). Obtida a partir da soma das LOC marcadas com (*) na seção de *New Objects*.

- *Estimated Total Development Time*: $\text{Time} = \beta_0 + \beta_1 * E$ (utilizar β_0 e β_1 calculados para tempo).
- *PredictionRange*: calcule o intervalo de previsão usando a tabela A29 (procedimento para intervalo de previsão).
- *Upper Prediction Interval (UPI) - size*: $\text{UPI} = N + \text{Range}$ (utilizar “Range” calculado para tamanho).
- *Upper Prediction Interval (UPI) - time*: $\text{UPI} = N + \text{Range}$ (utilizar “Range” calculado para tempo).
- *Lower Prediction Interval (LPI) - size*: $\text{LPI} = N - \text{Range}$ (utilizar “Range” calculado para tamanho).
- *Lower Prediction Interval (LPI) - time*: $\text{LPI} = N - \text{Range}$ (utilizar “Range” calculado para tempo).
- *Prediction Interval Percent*: registrar o percentual de probabilidade utilizado para calcular os intervalos de previsão (70% ou 90%).