

O USO DOS ÍNDICES CP E CPK NO DESENVOLVIMENTO DE SOFTWARE: UMA CRÍTICA

Antonio Carlos Tonini (ESEG)

antonio.tonini@vanzolini.org.br

Mauro de Mesquita Spínola (EPUPS-PRO)

mauro.spinola@usp.br



Os índices de capacidade C_p e C_{pk} são recomendados para medir a capacidade de um processo em atender as suas especificações. Foram constatados outros usos desses índices, como servir de base de cálculo para o custo de oportunidade da realocação de recursos no desenvolvimento de software. O objetivo deste artigo é fazer uma crítica a respeito da superficialidade na aplicação desses conceitos, tomando por base a literatura especializada nos métodos estatísticos orientados para a qualidade dos processos produtivos. O rigor estatístico mostrou resultados diversos do que haviam sido originariamente encontrados, evidenciando falhas que poderiam comprometer a eficácia de sua aplicação.

Palavras-chaves: Índices C_p e C_{pk} ; Controle Estatístico do Processo; Desenvolvimento de Software.

1. Introdução

No desenvolvimento de software, o esforço humano é a medida da capacidade do processo. Para atender a demanda, a realocação dos recursos humanos entre os projetos é uma das práticas mais empregada, o que a torna uma das causas potenciais de variabilidade do processo.

Este assunto foi discutido em um artigo publicado pelo autor na 19ª edição do International Conference on Production Research (ICPR19, 2007) apresentando uma experiência do uso dos índices C_p e C_{pk} para o cálculo da capacidade do processo e para o cálculo do custo de oportunidade da realocação dos recursos humanos. Os resultados mostraram maior precisão no cálculo da capacidade do processo, tornando mais visível para os clientes as modificações feitas nos seus projetos.

O objetivo deste artigo é fazer uma crítica a respeito da superficialidade na aplicação dos conceitos estatísticos e, especialmente, do cálculo da capacidade do processo, tomando por base a literatura especializada nos métodos estatísticos orientados para a qualidade dos processos produtivos. Os principais pontos a serem analisados se referem à qualidade do conjunto de dados coletados, ao tipo de distribuição utilizado para interpretar o comportamento dos dados, à elaboração dos gráficos de controle, à construção dos gráficos de controle e ao cálculo dos índices de capacidade.

O rigor estatístico evidencia resultados diversos do que haviam sido originariamente encontrados, o que, no limite, obrigaria a organização a rever seus conceitos, bem como a estratégia de emprego das medidas do processo de desenvolvimento de software.

2. O esforço no desenvolvimento de software

O desenvolvimento de software é ao mesmo tempo uma atividade artesanal e uma atividade industrial passível de organização. Utiliza intensivamente a interação intelectual entre os desenvolvedores e os clientes-usuários e a criatividade humana para produzir soluções digitais num processo de iteração contínua e sinérgica (FENTON e PFLEEGER, 1997), enquanto segue procedimentos de desenvolvimento amplamente conhecidos (KAN, 2002). Para que possa ser mensurado duas medidas são necessárias: o tamanho ou volume de produção ou quantidade de software e a quantidade de recursos empregados.

Para se determinar o tamanho bruto do software, duas técnicas tem sido universalmente utilizadas: a quantidade de linhas de código (DeMARCO, 1982) (PARK, 1992) ou a quantidade de pontos de Função (DEKKERS e AGUIAR, 2001), sendo que esta última técnica se transformou na norma ISO/IEC 14143 (ISO/IEC 14143-1:6:2007). Para se obter o tamanho específico do software, considera-se ainda a possibilidade de reuso de artefatos já existentes na organização e o grau de complexidade tecnológica envolvido, entre outros fatores (JONES, 1996).

O principal recurso empregado é o esforço humano, medido pela quantidade de tempo empregado (KAN, 2002). Além disso, o total de esforço à disposição da organização define tanto a capacidade nominal como a capacidade disponível, as quais são informações essenciais para a definição do prazo de entrega de qualquer projeto de desenvolvimento.

Como o esforço efetivamente utilizado em um projeto de desenvolvimento de software só é conhecido ao término do trabalho, torna-se necessário formular uma estimativa do esforço

necessário para servir de base nas negociações de preço e de prazo de entrega. Para tanto, a Engenharia de Software aponta três tipos de técnicas:

- a) custos envolvidos (*Constructive Cost Model* – COCOMO de Boehm, 1981). Esta técnica calcula o esforço com base no tamanho do código e o ajusta com 15 fatores relacionados com o histórico da organização, com a tecnologia empregada e as características do projeto;
- b) recursos envolvidos (*Software Lifecycle Model* – SLIM de Putnam, 1978). Esta técnica calcula o esforço para cada uma das fases do ciclo de desenvolvimento, tendo como base a quantidade de linhas de código a ser produzida;
- c) composição da opinião de especialistas e histórico da organização (*Wideband Delphi Technique* de Putnam e Fitzsimmons, 1979). Esta técnica é recomendada para as estimativas iniciais do projeto e considera tanto o histórico da organização como a opinião de especialistas; na falta de um deles, recomenda o uso de tabelas de produtividade por tipo de linguagem de programação, tamanho esperado em linhas de código e complexidade. Para tanto, considera a incerteza na determinação do tempo necessário para cada atividade, a qual é uma variável aleatória com uma distribuição estatística de probabilidade do tipo Beta. A definição dos parâmetros estatísticos: média, a variância e os limites de controle, conforme mostra a figura 1.

$\text{Estimativa média} = \frac{\text{menor tempo (otimista)} + 4 (\text{tempo mais provável}) + \text{maior tempo (pessimista)}}{6}$	
$\text{Variância} = \frac{\text{maior tempo (pessimista)} - \text{menor tempo (otimista)}}{6}$	$\text{Desvio padrão} = \sqrt{\text{Variância}}$
$\text{Limite superior} = \text{Estimativa média} + 3 \sqrt{\text{Variância}}$	$\text{Limite inferior} = \text{Estimativa média} - 3 \sqrt{\text{Variância}}$

FIGURA 1 – Distribuição da estimativa de esforço no desenvolvimento de software.
 Fonte: Putnam e Fitzsimmons (1979)

onde:

- a) menor tempo (otimista): menor tempo possível para realização da atividade, desde que haja recursos suficientes e disponíveis ininterruptamente;
- b) maior tempo (pessimista): maior tempo admissível para realização da atividade, considerando situações adversas para realização do trabalho;
- c) tempo mais provável: tempo médio efetivamente praticado pela organização (ou resultado de benchmarking) para atividades semelhantes em complexidade, tamanho e recursos empregados.

Por outro lado, a ocupação efetiva do tempo pode ser desmembrada de forma a prover maior precisão de controle. Assim, o tempo total pode ser dividido entre a aplicação efetiva no trabalho e tempo sem atividade alguma; a aplicação efetiva pode conter uma parte de tempo utilizado de forma não produtiva; o tempo produtivo pode significar avanço do trabalho físico ou qualquer tipo de retrabalho necessário para corrigir erros e modificações (NAKAJIMA, 1989).

Dvir (1998) reforça que existe uma forte sinergia entre os fatores de qualidade do processo (adequação aos procedimentos de desenvolvimento, cumprimento de prazos, conclusão

completa do projeto, ausência de problemas pós-encerramento, identificação e resolução de problemas durante a execução do projeto) e os fatores de qualidade do produto (atendimento às especificações de acordo com o prazo e custo). Shenhar (2001) mostra a mudança da importância relativa dos fatores ao longo do tempo (figura 2-a) e de acordo com a incerteza tecnológica do projeto (figura 2-b).

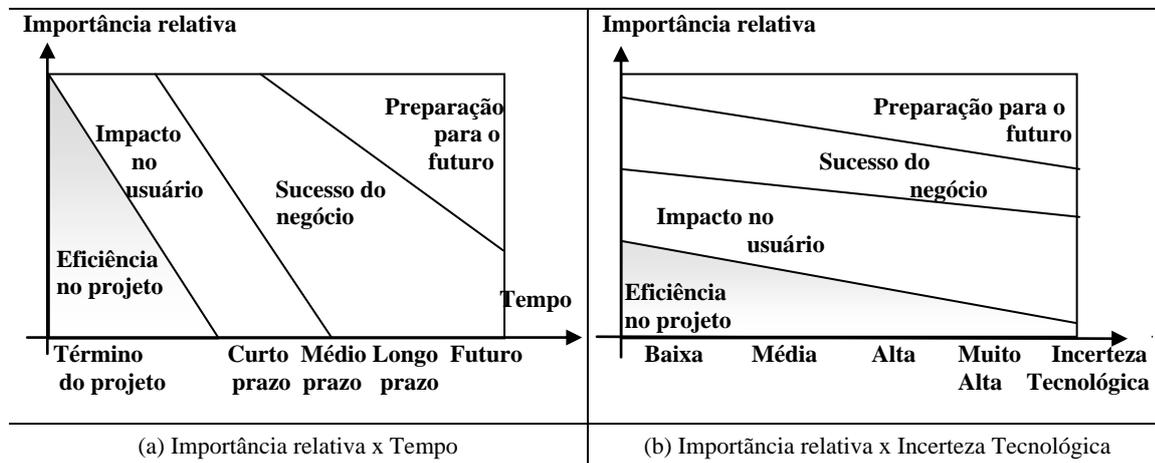


FIGURA 2 – Importância relativa dos fatores x Tempo x Incerteza tecnológica. Fonte: Shenhar (2001)

No curto prazo, os aspectos de eficiência do projeto são os mais importantes e passíveis de medição. Em projetos de baixa incerteza tecnológica, as expectativas em relação ao projeto estão muito mais ligadas a contribuições marginais em que a eficiência do desenvolvimento é fator determinante. Com o passar do tempo, é esperado um aumento de maturidade da equipe de desenvolvimento, traduzido no aumento da taxa de produtividade, na estabilidade e previsibilidade do trabalho.

Diversos contratemplos contribuem para a perda de eficiência e previsibilidade do desenvolvimento de software, tais como mudanças nos requisitos do projeto, mudanças tecnológicas, aumento no volume de projetos sem o correspondente aumento dos recursos etc. Uma das práticas mais comuns para cobrir estes contratemplos é a realocação de recursos entre os projetos (FENTON e PFLEEGER, 1997) (KAN, 2002) (CARD, 2002). Ela consiste em redistribuir os recursos entre os projetos que estão sendo executados de forma a garantir a continuidade de todos eles, mas, em contrapartida provoca o surgimento do custo de oportunidade pela troca dos recursos, conforme mostra a figura 3.

Tipo de Custo	Implicação no projeto que:	
	Perde o recurso	Ganha o recurso
Custo do recurso	Reduz	Aumenta
Custo do atraso	Aumenta	Reduz
Custo com a acomodação dos outros recursos	Aumenta	Aumenta
Custo de Gerenciamento	Aumenta	Aumenta

FIGURA 3 – Implicação da realocação de recursos nos custos dos projetos (PARK *et al*, 1996)

3. Uma experiência da gestão de realocação de recursos

O caso em análise se refere a uma organização desenvolvedora de software, com expressiva presença no mercado brasileiro que conta com uma carteira de 200 projetos/ano com diferentes escopos, complexidade e esforço. Para ela, a constante realocação de mão-de-obra entre projetos é questão de sobrevivência, pois ela passa para os clientes a impressão de que

sempre dispõe de mão-de-obra suficiente para atendê-los dentro do prazo.

Para realizar a realocação com parcimônia e mitigar o custo de oportunidade, a organização incorporou os índices Cp e Cpk no cálculo do custo de oportunidade da realocação dos recursos. Para tanto, adotou (a) os valores históricos extremos como sendo seus limites de controle; (b) o tempo médio histórico da organização com uma redução de 5% como sendo o tempo otimista; (c) o tempo médio histórico da organização acrescido em 30% como sendo o tempo pessimista. Com estes valores, passou a negociar com os clientes que solicitavam mudanças nos seus prazos.

Os resultados apresentados no trabalho (TONINI *et al.*, 2007) se referiram a dois momentos: “antes”, quando a organização passou a adotar os índices Cp e Cpk e o outro, com resultados de cinco anos “depois”. Foram utilizados dados de dez projetos similares em tamanho e complexidade relativos ao tempo utilizado em cada um dos quatro processos fundamentais de desenvolvimento de software (levantamento de requisitos, elaboração do projeto, construção do software e implantação junto ao cliente-usuário). Admitiu-se que a distribuição dos dados era normal. Os valores dos índices de capacidade mostram que houve redução da variabilidade nos processos, um aumento significativo da capacidade nominal (índice Cp) e ainda, uma tímida evolução da capacidade dos processos frente às metas (índice Cpk).

TABELA 1 – INDICES Cp e Cpk

Índices	Antes				Depois			
	Proc. 1	Proc. 2	Proc. 3	Proc. 4	Proc. 1	Proc. 2	Proc. 3	Proc. 4
Cp	1,07	0,79	0,37	0,87	2,01	1,35	1,8	1,44
Cpk	0,73	0,30	0,15	0,41	1,01	1,34	1,34	0,86

Fonte: Tonini *et al.* (2007).

4. Críticas aos fundamentos estatísticos do trabalho

Não obstante os benefícios proporcionados, algumas críticas devem ser feitas quanto à falta de rigor estatístico. Tomando-se por base uma série de precauções levantadas na literatura específica, foram identificados alguns pontos que, se tratados convenientemente, poderiam agregar mais valor, consistência e maior confiabilidade nos resultados. São eles:

a) falta de consistência dos objetivos e a coleta das variáveis (BALESTRACCI, 1998).

Para conhecer a capacidade do processo em atender às especificações dos clientes, deveria ter sido utilizado somente o tempo total dos projetos e não os tempos de cada atividade, uma vez que os clientes adquirem produtos prontos, cuja produção envolve todas as atividades. Além disso, como as equipes eram especializadas na atividade, não faz sentido realocar recursos de equipes diferentes mas apenas entre projetos, o que não ficou claro no trabalho;

b) falta de clareza nos critérios de estratificação dos dez projetos e nas evidências sobre suas similaridades (HUFF, 1993);

c) coleta dos dados confusa e omissão da unidade de tempo adotada (HUFF, 1993);

d) ausência da análise de tendência dos dados (4ª armadilha, BALESTRACCI, 1998) (RAMOS, 2000);

e) ausência dos testes relativos ao tipo de distribuição de dados; ao invés, foi admitido que a distribuição deles era normal sem comprovar (3ª armadilha, BALESTRACCI, 1998)

f) ausência de consideração sobre pontos fora dos limites, bem como não exibição de sequer um gráfico de controle. Como os limites de controle adotados foram os valores extremos

(menor e maior valor coletados) perdeu-se a oportunidade de se identificar causas especiais (5ª armadilha, BALESTRACCI, 1998);

f) implementação arbitrária de melhoria no processo atribuída ao uso dos valores dos índices de capacidade sem uma análise mais profunda (6ª armadilha, BALESTRACCI, 1998).

4. Revisão do problema

Para se reavaliar o estudo, levantou-se novamente a questão com a organização. A realocação de recursos ainda é uma estratégia praticada para garantir o cumprimento dos prazos de entrega. Os desenvolvedores foram distribuídos em equipes de desenvolvimento e os projetos em carteira garantem o pleno emprego das equipes. O critério de escolha da equipe que cede o recurso nas realocações continua sendo aquele da equipe com o maior índice de capacidade Cp. Os limites de especificação passaram a ser: Inferior – menor tempo obtido acrescido de 10%; Superior – maior tempo praticado pela equipe reduzido em 15 %.

Para elaboração dos cálculos de capacidade, a empresa disponibilizou dados relativos ao tempo de desenvolvimento praticado por quatro de suas equipes, referentes a 20 projetos similares (projetos com 1200 PF – pontos de função) de cada uma das equipes (tabela 2). Os apontamentos equivalem a cada “¼ da jornada de trabalho” (cerca de 2 horas) ajustados para base semanal (1 semana corresponde a quatro jornadas diárias). ‘

TABELA 2 – TEMPOS TOTAIS DOS PROJETOS por EQUIPE

Equipe 1		Equipe 2		Equipe 3		Equipe 4	
4,1	2,8	2,6	2,5	4,6	2,9	1,2	1,3
1,3	3,1	1,5	1,3	6,2	1,4	1,3	3,1
3,1	4,2	3,4	3,8	3,7	4,2	1,3	2,5
2,5	3,1	2,7	4,1	0,7	2,7	1,4	3,4
3,4	1,2	2,9	2,0	3,9	1,8	2,1	1,2
1,2	2,9	1,4	2,7	4,7	3,5	1,3	4,2
1,9	2,6	4,2	3,1	0,7	2,7	2,3	2,7
1,1	1,8	2,7	1,6	4,9	3,1	5,2	1,8
3,6	2,8	1,8	3,0	4,5	4,1	2,1	3,5
4,6	2,7	3,5	2,1	4,1	1,9	1,8	2,5

Fonte: Dados do projeto

As questões de pesquisa, no entanto, são agora mais amplas:

a) os dados coletados representam valores referentes a projetos já concluídos. No entanto, a cada nova realocação de recursos é necessário refazer as estimativas para as atividades remanescentes de cada projeto. Assim, com os dados coletados é possível fazer o cálculo da capacidade?

b) os índices de capacidade Cp e Cpk são os mais recomendados neste caso?

5. Avaliação estatística da situação

A construção dos gráficos foi feita pelo software Minitab - versão 15 (MINITAB, 2007) e a análise realizada se caracteriza como observação de dados históricos, sem possibilidade de interferência no processo em andamento. O tipo de variação mais apropriado para este estudo é a “variação cíclica”, pois se pretende buscar variações entre as equipes (RAMOS, 2007).

As figuras 4, 5, 6 e 7 apresentam os histogramas de cada equipe, as medidas de concentração (media, mediana) e de dispersão (desvio padrão amostral) o teste de normalidade Anderson-Darling referentes às equipes 1, 2, 3 e 4, respectivamente (MONTGOMERY, 2004).

Apenas a Equipe 4 apresenta uma distribuição de dados que pode ser considerada Não-

Normal, decorrente do resultado do p-value no teste de Anderson-Darling menor que 0,05.

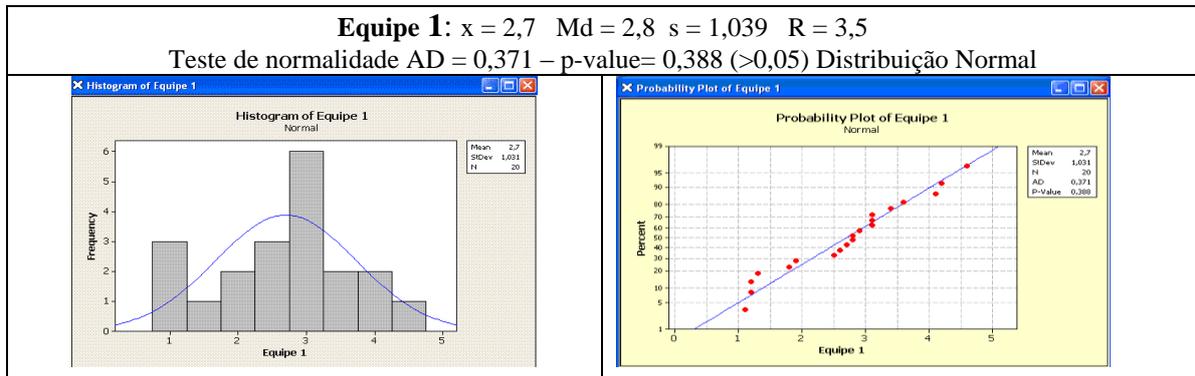


FIGURA 4 – Histograma, Teste de Normalidade e medidas de concentração e dispersão da Equipe 1

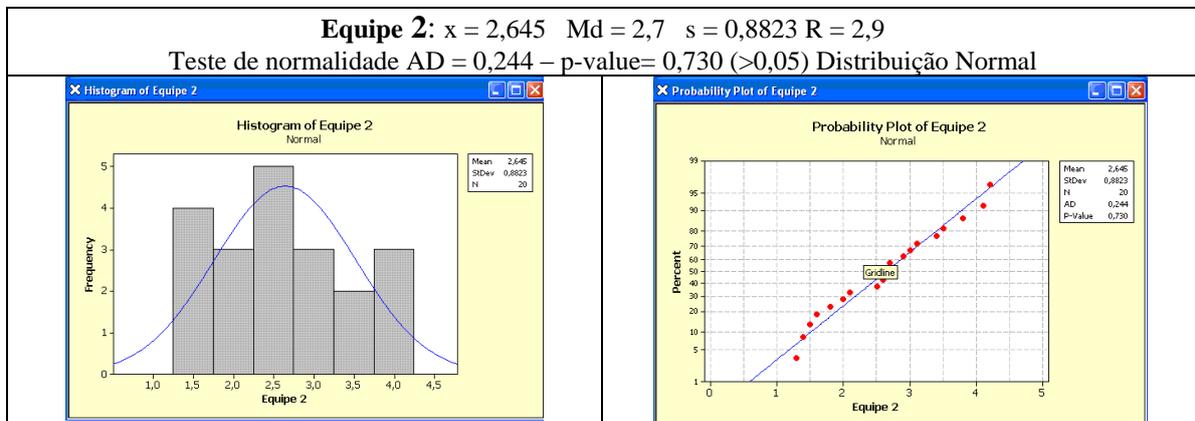


FIGURA 5 – Histograma, Teste de Normalidade e medidas de concentração e dispersão da Equipe 2

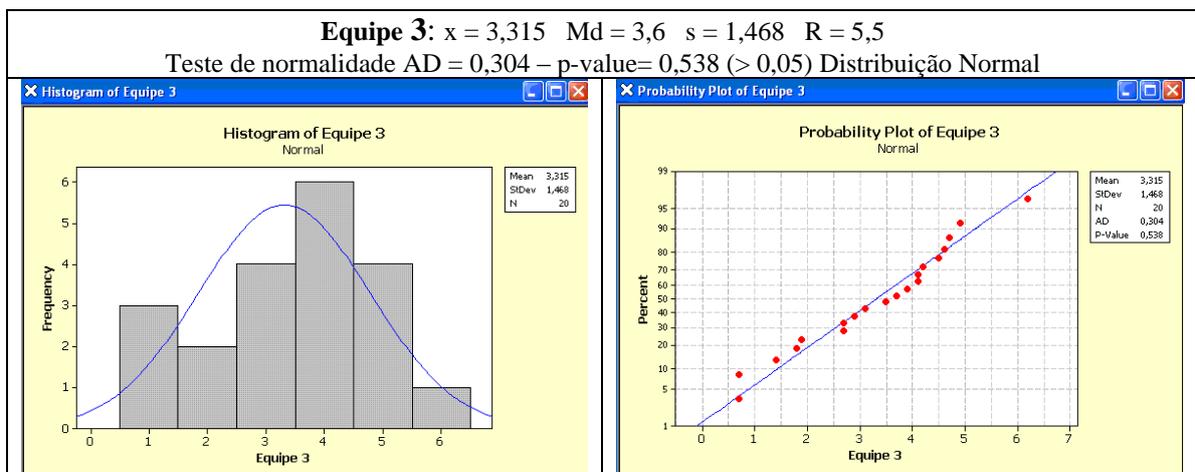


FIGURA 6 – Histograma, Teste de Normalidade e medidas de concentração e dispersão da Equipe 3

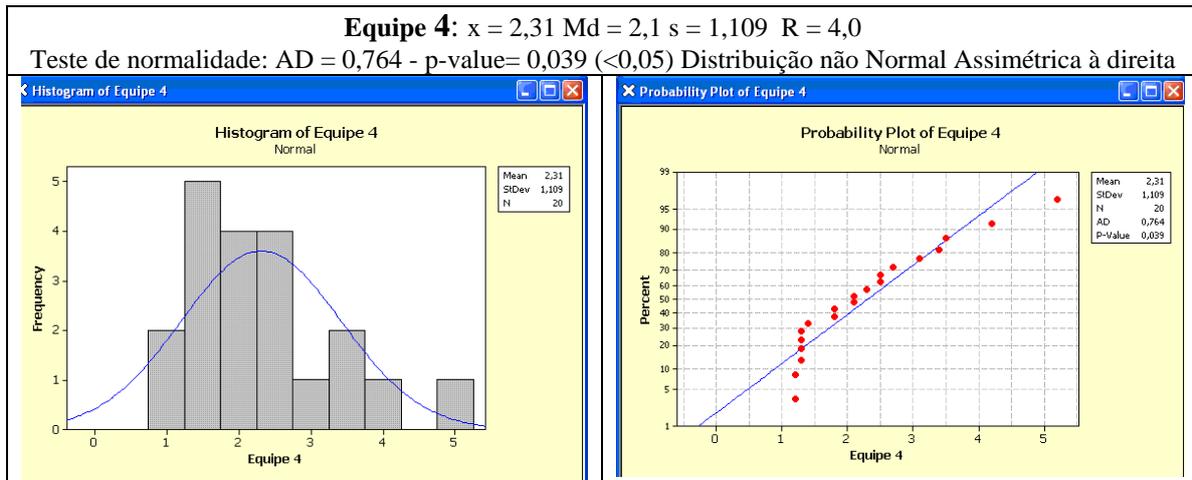


FIGURA 7 – Histograma, Teste de Normalidade e medidas de concentração e dispersão da Equipe 4

Como a variável analisada é o tempo e cada uma das equipes apresenta um valor único para o tempo de desenvolvimento, o gráfico de controle mais apropriado é o para valor individual e amplitude móvel (\bar{x} e \bar{R}_m).

A amplitude móvel é a diferença (em módulo) entre dois valores consecutivos obtidos ao longo do tempo (RAMOS, 2000).

Os parâmetros e suas respectivas formulações estão na tabela 3.

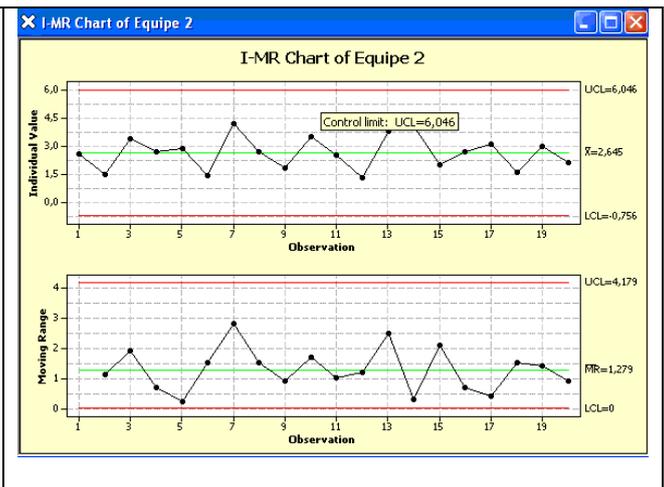
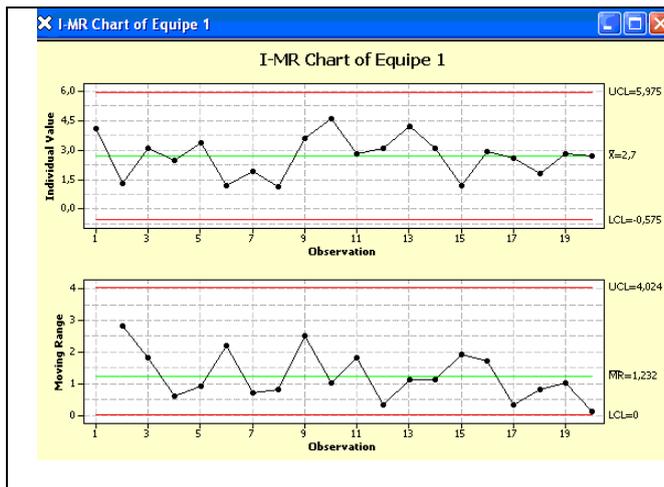
A tabela 4 apresenta os valores calculados para elaboração dos os gráficos de controle (\bar{x} e \bar{R}_m) de cada uma das equipes de desenvolvimento e a figura 8 apresenta os referidos gráficos.

TABELA 3 – PARÂMETROS PARA O CÁLCULO DOS GRÁFICOS DE CONTROLE

Parâmetro		Fórmula
Média da amostra		$\bar{x} = \frac{\sum x}{k}$
Amplitude móvel		$\bar{R}_m = \frac{\sum R_m}{k-1}$
Fator de correção pela introdução do vício (viej) por substituir a dispersão natural pela amplitude móvel		$E_2 = \frac{3}{d_2} = 3,267$
Gráfico de controle dos valores individuais	Limite superior de controle	$LSC_x = \bar{x} + E_2 \times \bar{R}_m = \bar{x} + \frac{3}{d_2} \times \bar{R}_m$
	Média	$LM_x = \bar{x}$
	Limite inferior de controle	$LIC_x = \bar{x} - E_2 \times \bar{R}_m = \bar{x} - \frac{3}{d_2} \times \bar{R}_m$
Gráfico de controle da amplitude móvel	Limite superior de controle	$LSC_{R_m} = D_4 \times \bar{R}_m$
	Média	$LM_{R_m} = \bar{R}_m$
	Limite inferior de controle	$LIC_{R_m} = D_3 \times \bar{R}_m$

TABELA 4 – VALORES PARA A ELABORAÇÃO DO GRÁFICO DE CONTROLE por EQUIPE

Amostra	Equipe 1		Equipe 2		Equipe 3		Equipe 4	
	Xi	Rm	Xi	Rm	Xi	Rm	Xi	Rm
1	4,1	-	2,6	-	4,6	-	1,2	-
2	1,3	2,8	1,5	1,1	6,2	1,6	1,3	0,1
3	3,1	1,8	3,4	1,9	3,7	2,5	1,3	0,0
4	2,5	0,6	2,7	0,7	0,7	3,0	1,4	0,1
5	3,4	0,9	2,9	0,2	3,9	3,2	2,1	0,7
6	1,2	2,2	1,4	1,5	4,7	0,8	1,3	0,8
7	1,9	0,7	4,2	2,8	0,7	4,0	2,3	1,0
8	1,1	0,8	2,7	1,5	4,9	4,2	5,2	2,9
9	3,6	2,5	1,8	0,9	4,5	0,4	2,1	3,1
10	4,6	-1,0	3,5	1,7	4,1	0,4	1,8	0,3
11	2,8	1,8	2,5	1,0	2,9	1,2	1,3	0,5
12	3,1	0,3	1,3	1,2	1,4	1,5	3,1	1,8
13	4,2	1,1	3,8	2,5	4,2	2,8	2,5	0,6
14	3,1	1,1	4,1	-0,3	2,7	1,5	3,4	0,9
15	1,2	1,9	2,0	2,1	1,8	0,9	1,2	2,2
16	2,9	1,7	2,7	0,7	3,5	1,7	4,2	3,0
17	2,6	0,3	3,1	0,4	2,7	0,8	2,7	1,5
18	1,8	0,8	1,6	1,5	3,1	0,4	1,8	0,9
19	2,8	1,0	3,0	1,4	4,1	1,0	3,5	1,7
20	2,7	0,1	2,1	0,9	1,9	2,2	2,5	1,0
Total	54,0	21,4	52,9	23,7	66,3	34,1	46,2	23,1
\bar{x} e \bar{R}_m	2,700	1,232	2,645	1,274	3,320	1,795	2,231	1,216
LSC _x e LSC _{Rm}	5,975	4,024	6,046	4,179	8,090	5,864	5,543	3,972
LM _x e LM _{Rm}	2,700	1,232	2,645	1,279	3,320	1,795	2,231	1,216
LIC _x e LIC _{Rm}	-0,575	0	-0,756	0	-1,460	0	-0,923	0



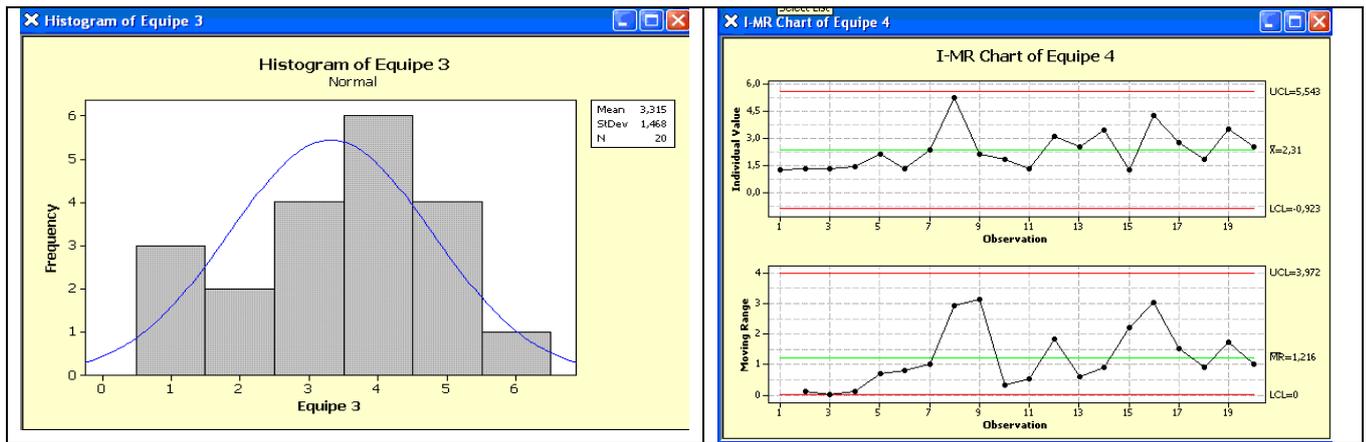


FIGURA 8 – Gráficos de controle de todas as equipes

Analisando-se os gráficos de controle dos valores individuais e amplitudes móveis, verifica-se que não existem pontos fora dos limites de controle (superior e inferior). Como nenhum dos oito testes sugeridos por Nelson (RAMOS, 1995) apresentou falha, pode-se admitir que não existem causas especiais de variação atuantes no processo em todas as equipes de desenvolvimento e que, portanto, elas apresentam estabilidade estatística. A amplitude móvel média (valor de \bar{R}_m) representa efetivamente a variabilidade desses processos.

Considerando que apenas a equipe 4 apresenta uma distribuição de dados Não-Normal, foram calculados os índices de capacidade C_p e C_{pk} e também os índices P_p e P_{pk} para as equipes 1, 2 e 3 (tabela 5). Os limites de especificação considerados foram: Inferior – menor tempo obtido pela equipe acrescido de 10%; Superior – maior tempo praticado pela equipe reduzido em 15%. As fórmulas dos índices de capacidade são as seguintes:

$C_p = \frac{LSE - LIE}{6 \times \frac{\bar{s}}{c_4}}$	$C_{pk} = \text{Min}[C_{pi}, C_{ps}], \quad C_{pi} = \frac{\bar{x} - LIE}{3 \times \frac{\bar{s}}{c_4}}, \quad C_{ps} = \frac{LSE - \bar{x}}{3 \times \frac{\bar{s}}{c_4}}$
$P_p = \frac{LSE - LIE}{6 \times s}$	$P_{pk} = \text{Min}[P_{pi}, P_{ps}], \quad P_{pi} = \frac{\bar{x} - LIE}{3 \times s}, \quad P_{ps} = \frac{LSE - \bar{x}}{3 \times s}$

TABELA 5 – VALORES DOS ÍNDICES DE CAPACIDADE por EQUIPE

	Equipe 1	Equipe 2	Equipe 3
LIE – Limite inferior de especificação	1,210	1,430	0,770
LSE – Limite superior de especificação	3,910	3,570	5,270
Variação natural do processo	2,700	2,140	4,500
Média (\bar{x})	2,700	2,645	3,320
Desvio padrão (S)	1,039	0,882	1,468
C4	0,987	0,987	0,987

$6 \times S$	6,234	5,292	8,808
$6 \times S / C_4$	6,317	5,362	8,925
Cp	0,427	0,399	0,504
Cpi	0,472	0,453	0,571
Cps	0,383	0,345	0,437
Cpk	0,383	0,345	0,437
Pp	0,433	0,404	0,511
Ppi	0,516	0,357	1,247
Pps	0,419	0,272	0,954
Ppk	0,419	0,272	0,954

5. Continuidade do estudo

Considerando tanto o índice Cp quanto o índice Pp, nenhuma das equipes apresenta capacidade para atender as especificações formuladas.

Uma das limitações do estudo foi não ter analisado uma distribuição que considere os tempos otimista e o tempo pessimista (distribuição Beta) e também medir o grau de assertividade das estimativas, que pode ser traduzido pela fórmula:

$$\text{Erro de estimativa} = \frac{\text{Tempo previsto} - \text{Tempo utilizado}}{\text{Tempo previsto}}$$

onde:

- se o tempo utilizado for exatamente o tempo previsto, então não há erro de estimativa;
- se o tempo utilizado for menor que o tempo previsto, então há erro de estimativa que varia de $0 < \text{erro} < 1$
- se o tempo utilizado for maior que o tempo previsto, há erro de estimativa que varia $-\infty < \text{erro} < 0$

6. Considerações finais

O uso mais preciso do ferramental estatístico traz para a organização uma confiança nos resultados históricos praticados e mais clareza ao estipular sua política comercial baseada nos tempos desempenhados pelas equipes. A realocação de recursos entre os projetos passa a ser uma prática sistematizada e não ad-hoc fundamentada no pragmatismo e experiência gerencial.

Por outro lado, as organizações desenvolvedoras de software devem continuamente atentar para as questões de eficiência de seus recursos e políticas salutares de redução racional de custos como diferencial competitivo.

A importância do cálculo do custo de oportunidade baseado no cálculo da capacidade do processo reside no fato de que ela expressa mais fielmente a relação custo-benefício de qualquer modificação que tenha que ser feita em um projeto de desenvolvimento de software. Torna-se uma medida justa, que pode ser utilizada como moeda de troca nas negociações com os clientes.

Referências

BALESTRACCI, D. *Data "Sanity": statistical thinking applied to everyday data.* Milwaukee, ASQ Statistics Division, 1998.

- CARD, D.** *Managing "Software Quality with defects"*. 26th COMPSAC: Computer Software and Applications Conference. Proceedings of IEEE Computer Society. p.472-474. aug, 2002.
- DeMARCO, T.** *Controlling Software Projects*. Yourdon Press, New York, 1982.
- DEKKERS, C.; AGUIAR, M.** *Applying Function Point Analysis to Requirements Completeness*. Crosstalk: The Journal of Defense Software Engineering, v.14, n.2, p.8-11, feb, 2001.
- DVIR, D.** *In Search of Project Classification: a Non-universal Approach to Project Success Factors*. Research Policy n.27, p.915-935, 1998.
- FENTON, N. E.; PFLEEGER, S. L.** *Software metrics: a rigorous approach* 2nd Edition. International Thomson Computer Press, 1997.
- HUFF, D.** *How to lie with statistics*. New York: Norton & Company, 1993.
- ISO/IEC 14143-1:6:2007.** *Information technology: Software measurement: Functional size measurement*. International Standardization Organization, 2007.
- KAN, S.** *Metrics and models in software quality engineering*, 2nd ed. Addison-Wesley, 2002.
- JONES, C.** *Applied Software Measurement, Assuring Productivity and Quality*. 2nd ed., NY: McGraw, 1996.
- MINITAB.** *Software Minitab* ® 15.1.0.0.. Minitab Inc. 2007.
- MONTGOMERY, D. C.** *Introduction to statistical quality control*, 3rd Edition. John Wiley & Sons, 1997.
- NAKAJIMA, S.** *Introdução ao Total Productive Maintenance (TPM)*. São Paulo: IMC, 1989.
- PARK, R.** *Software size measurement: a framework for counting source statements*, CMU/SEI-92-TR-20, Software Engineering Institute Technical Report, Pittsburg, PA, 1992.
- PARK, R. E.; GOETHERT, W. B.; FLORAC, W. A.** *Goal-Driven Software Measurement: a guidebook*. Pittsburgh, PA: *Software Engineering Institute*, Carnegie Mellon University, august 1996. HANDBOOK. CMU/SEI-96-HB-002. Disponível em: <<http://www.sei.cmu.edu/pub/documents/96.reports/pdf/hb002.96.pdf> >
- PUTMAN, L.H.** *A general empirical solution to the macro software sizing and estimating problem*. IEEE Trans. on Software Engineering, v. 4, n.4, p.345-61, april 1978.
- PUTMAN, L.H.; FITZSIMMONS, A.** *Estimating Software Costs. Datamation*. sep/1979, p.189-198, oct/1979, p.171-198 e nov/1979, p.137-140.
- RAMOS, A.W.** *Controle estatístico de processo para pequenos lotes*. São Paulo: Edgard Blücher, 1995.
- RAMOS, A.W.** *Controle estatístico de processo para processos contínuos e em bateladas*. São Paulo: Edgard Blücher, 2000.
- RAMOS, A.W.** *Análise Estatística da Qualidade*. Disciplina PRO 5777, Análise Estatística da Qualidade. Curso de Pós-Graduação. Departamento de Engenharia de Produção. Escola Politécnica da Universidade de São Paulo. set-dez, 2007. 93 f. Notas de Aula. Impresso.
- SHENHAR, A.** *Project Success: a Multidimensional Strategic Concept*. Long Range Planning n.34, p.699-725, 2001.
- TONINI, A.C; PESSOA, M.S.P.; SPINOLA, M.M.** *Assessment of resource relocation with Cp and Cpk indexes: a Six Sigma contribution for software development process*. Proceedings of the 19th International Conference on Production Research (ICPR10, 2007). Advanced Methods for Production System. Disponível em CD. \mswd\Papers\151.pdf., Valparaiso, Chile, jul 2007.