

## Medição e Análise no CMMI com Metodologia Seis Sigma e ISO/IEC/IEEE 15939

Charles S. F. Silva<sup>†1</sup>, Marcelo S. Oliveira<sup>2</sup>

<sup>1</sup> Professor de Informática, Escola Preparatória de Cadetes do Ar (EPICAR).

<sup>2</sup> Universidade Federal de Lavras (UFLA).

**Resumo:** Baseados na premissa de que a qualidade do produto de software é, em sua maior parte, fruto da qualidade do processo no qual é desenvolvido, modelos de melhoria de processos de software têm sido desenvolvidos. Nestes, a medição e análise assumem lugar de relevância, pois a avaliação da relação entre a capacidade (resultados estimados) e o desempenho (resultados observados) de um processo é importante para que este possa ser melhorado. O CMMI é um dos modelos de melhoria mais conhecidos, já contando com uma ampla experiência de uso bem sucedido pela indústria. O CMMI não diz que sistema de medição usar ou como implementá-lo. A metodologia Seis Sigma, conjugada com as recomendações da norma ISO/IEC/IEEE 15939, supre esta lacuna, conforme demonstrado por meio de um estudo de caso, que teve como resultados a indicação de ações para a melhoria do processo de software.

**Palavras-chave:** CMMI; Estatística; Seis Sigma; ISO/IEC/IEEE 15939; Processo de Software.

**Abstract:** Based on the premise that the quality of a software product is mostly the result of the quality of the process in which it is developed, models for the improvement of the process of software have been developed. The measurement and analysis for models assume great importance, because the evaluation of the relationship between the capacity (estimated results) and the performance (observed results) of a process is important so that it can be improved. CMMI is one of the more used models, having a great experience with success in the industry. CMMI do not indicate the measurement system to work and do not indicate its implementation. Six Sigma methodology conjugated to recommendations of ISO/IEC/IEEE 15939 aid CMMI in construction and implementation of that measurement system, as demonstrated by a case study that presented as results the indication of actions for the improvement of the software process.

**Keywords:** CMMI; Statistics; Six Sigma; ISO/IEC/IEEE 15939; Software Process.

### Introdução

A premissa de que a qualidade do produto de software é, em sua maior parte, fruto da qualidade do processo no qual é desenvolvido tem conduzido ao desenvolvimento de modelos de melhoria de processos de software. Nestes, a medição e a análise assumem lugar de relevância, tal como no modelo CMMI (*Capability Maturity Model Integration*, ou Integração de Modelos de Maturidade de Capacidade) que incorpora a medição no próprio conceito, visto que a definição de maturidade implica na relação capacidade (resultados estimados) *versus* desempenho (resultados observados) do processo, ao longo dos projetos.

O CMMI não diz que sistema de medição usar ou como implementá-lo, mas estabelece especificações a serem satisfeitas por esse sistema. De uma forma geral, medidas devem ser coletadas, de acordo com as metas da organização, para que indicadores de processo possam ser implementados. Estes, por sua vez, permitem à organização verificar a efetividade de um processo de desenvolvimento e comparar tal efetividade com as especificações estabelecidas pelo CMMI.

---

<sup>†</sup> Autor correspondente: [charlesfsilva@yahoo.com.br](mailto:charlesfsilva@yahoo.com.br).

O presente trabalho, então, se propõe a conjugar o “modo de gerenciar quantitativamente” da metodologia Seis Sigma com a estrutura conceitual de medição da norma ISO/IEC/IEEE 15939, de modo a atender às especificações demandadas pelo CMMI para o sistema de medição e galgar os degraus de melhoria indicados pelos níveis de maturidade ou de capacidade do CMMI com o uso de ferramentas e métodos estatísticos, de acordo com a metodologia Seis Sigma.

## O Modelo CMMI

Uma suposição básica da gestão de qualidade é que a qualidade do processo influencia diretamente a qualidade do produto (PAULK *et al.*, 1994). Há outros determinantes da qualidade do produto, como a capacidade do pessoal (equipe de desenvolvimento) e a tecnologia usada no processo. Não obstante, o investimento na qualidade do processo pode trazer retorno benéfico em prazo mais curto (PAULA FILHO, 2001).

Assim sendo, modelos de melhoria de processos de software têm sido desenvolvidos como parte essencial de um amplo esforço, envolvendo governos, universidades e indústrias de diversos países, no sentido de se evoluir de um estado de produção “artesanal” de software para um estado “industrial” de produção efetiva e profissional. A ideia é se obter processos de desenvolvimento de software definidos e alinhados com os objetivos da organização, padronizados, e que são previsíveis (pelo domínio da variabilidade, mediante rigoroso controle estatístico), condição essencial para se garantir o alcance do patamar de processo em estado de melhoria contínua, nível máximo na escala da gestão de qualidade de processo.

O CMMI é um dos modelos de melhoria mais difundidos, já contando com uma ampla experiência de uso bem sucedido pela indústria. Este modelo teve sua gênese histórica a partir da iniciativa do Departamento de Defesa Norte-Americano (DoD), visando melhorar as capacitações da indústria de software dos Estados Unidos. Para tanto, o DoD fundou o SEI, Instituto de Engenharia de Software, junto com a *Carnegie-Mellon University*. O resultado dessa iniciativa materializou-se no desenvolvimento do CMM – *Capability Maturity Model*, ou Modelo de Maturidade de Capacidade – modelo de melhoria de processo que descreve os elementos-chave de um processo de software efetivo (PAULK *et al.*, 1994). O CMMI surgiu a partir da necessidade de se integrar os modelos CMM desenvolvidos para várias atividades, como, por exemplo, engenharia de software, engenharia de sistemas, entre outras.

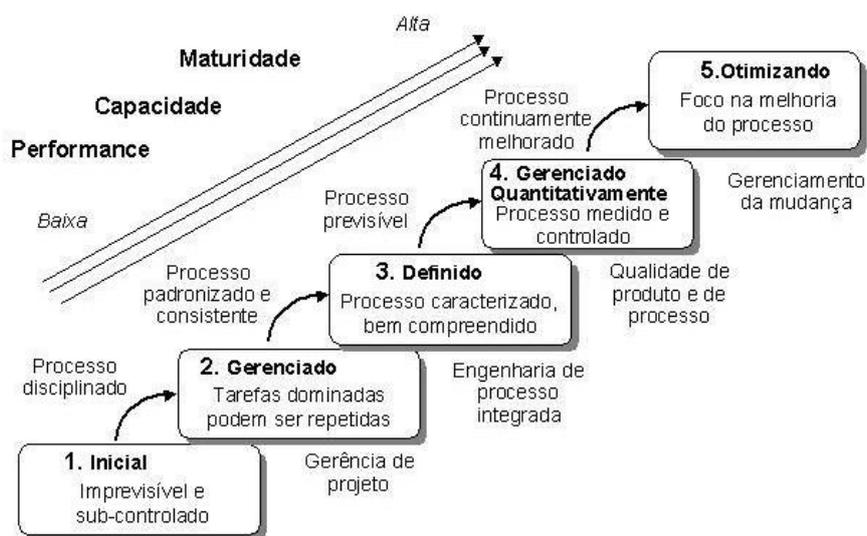
### Arquitetura do CMMI

A estrutura do CMMI é constituída pela definição de 22 áreas de processo – conjunto de atividades relacionadas que, se realizadas adequadamente, atendem um conjunto de metas consideradas importantes para aumentar a capacidade desse processo – organizadas em duas representações: uma representação por estágios e uma representação contínua (CMMI PRODUCT TEAM, 2014).

### Níveis de maturidade

Pertencem à representação por estágios e são aplicados para atingir melhorias de processos organizacionais por meio de múltiplas áreas de processo. Existem cinco níveis de maturidade, numerados de 1 a 5 (Figura 1), os quais indicam uma ordem crescente de melhoria.

Figura 1. CMMI – Níveis de maturidade na representação por estágios.



Fonte: Dos autores.

### Níveis de capacidade

Pertencem à representação contínua e são aplicados para atingir melhorias de processos organizacionais nas áreas de processos individuais. Existem quatro níveis de capacidade, numerados de 0 a 3 (os níveis 4, Gerenciado quantitativamente, e 5, Otimizando, foram removidos da representação contínua do CMMI, na versão 1.3, publicada em 2010):

- Nível 0 – Incompleto, que satisfaz apenas alguns objetivos específicos da área de processo;
- Nível 1 – Executado, que satisfaz todos os objetivos específicos da área de processo;
- Nível 2 – Gerenciado (processo planejado);
- Nível 3 – Definido (processo padronizado).

Os níveis 1 a 3 de capacidade da área de processo são similares aos níveis 1 a 3 de maturidade da organização da representação por estágios, com a ressalva de que, nesta última, cada nível diz respeito a um conjunto de áreas de processo pré-definidas (e não a uma área de processo em particular).

Na representação contínua, as áreas de processo estão agrupadas em quatro categorias: gestão de processos, gestão de projetos, engenharia e suporte (Figura 2).

Figura 2. Categorias de processo no CMMI – representação contínua.



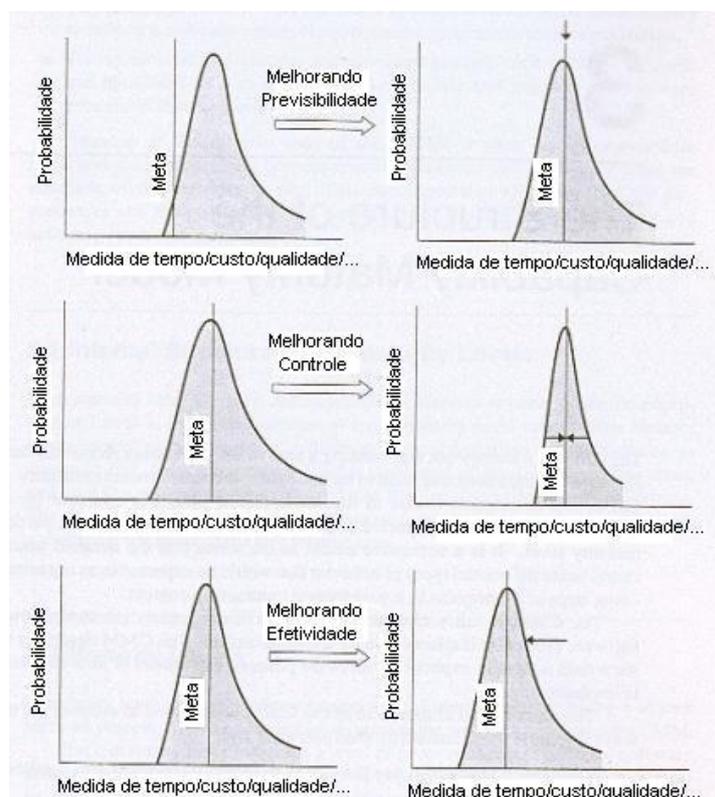
Fonte: Dos autores.

## Benefícios na melhoria de processos pelo uso do CMMI

A maturidade do processo de software da organização ajuda a prever a capacidade de um projeto de atingir seus objetivos (PAULK *et al.*, 1994). Os projetos de software, antes da melhoria, têm grandes variações na execução das metas de custo, prazo e qualidade. Segundo Paulk *et al.* (1994), com a implantação do CMMI, as seguintes melhorias, as quais são ilustradas na Figura 3, são esperadas:

- Previsibilidade: a primeira melhoria esperada quando a organização amadurece é na previsibilidade – a diferença entre os resultados estimados e os resultados observados diminui (não-tendenciosidade), e um exemplo seria um prazo real de desenvolvimento de software mais próximo do prazo estimado.
- Controle: a segunda melhoria é no controle – a variabilidade dos resultados observados ao redor dos resultados estimados diminui (variância mínima), por exemplo: prazos de entrega de projetos similares variam pouco.
- Efetividade: a terceira melhoria, na efetividade, indica que os resultados estimados, como um todo, melhoram; custo e tempo de desenvolvimento diminuem, e a produtividade e qualidade aumentam, por causa da redução do retrabalho na correção de erros.

Figura 3. CMMI e a melhoria de previsibilidade, controle e efetividade.

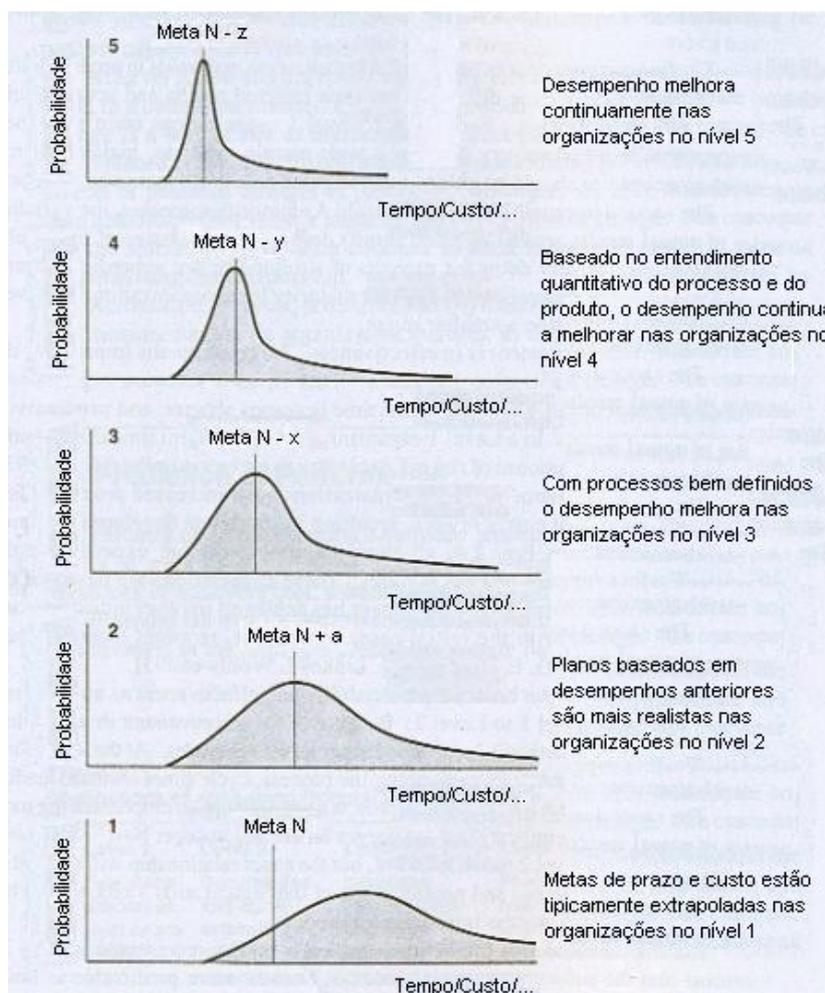


Fonte: Adaptado de Paulk et al.(1994).

Por causa da melhoria na estimaco com a implantao do CMMI, os planos se tornam mais realistas, permitindo planejamentos de prazos maiores. E, por causa de melhorias na execuo do processo, ciclos de vida mais curtos levam a prazos reais de desenvolvimento mais curtos.

As melhorias na predico dos resultados dos projetos de software, ilustrados na Figura 4, pressupem que as saidas desses projetos se tornam mais previsveis quando o retrabalho  removido do processo de software.

Figura 4. Capacidade de processo por nível de maturidade.



Fonte: Adaptado de Paulk et al. (1994).

É importante frisar que sistemas sem precedentes não apresentam uma sequência de melhorias semelhante a da Figura 4, desde que novas tecnologias e aplicações diminuem a capacidade de processo pelo aumento da variabilidade. Porém, mesmo no caso de tais sistemas, o gerenciamento e as práticas de engenharia de organizações mais maduras ajudam a identificar problemas mais cedo no ciclo de desenvolvimento, minimizando o risco de investimento numa causa perdida (PAULK et al., 1994).

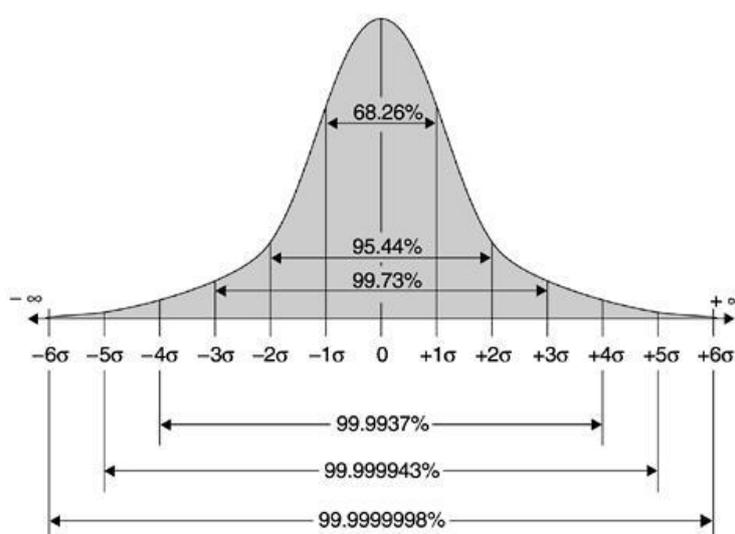
### A metodologia Seis Sigma

O uso dessa metodologia tem por objetivo o oferecimento de serviços e produtos de primeira qualidade, ao mesmo tempo em que, praticamente, elimina todas as ineficiências gerais do processo. O termo Seis Sigma representa um alto nível de qualidade, significando uma redução da variação no processo de desenvolvimento de maneira a se alcançar uma taxa de erros específica: 3,4 erros por milhão de oportunidades ou 99,999998% de perfeição (KAN, 1995).

A metodologia Seis Sigma é, portanto, uma medida de qualidade e eficiência, além de ser uma estratégia gerencial que busca a excelência na melhoria de processos. Cabe ressaltar que, ao aplicar essa metodologia, a organização passa a operar seus processos sob controle estatístico, controlando as variáveis de entrada do processo ao invés de, como de costume, as variáveis de saída, maximizando o uso dos equipamentos e otimizando o tempo de ciclo de produção.

A letra grega sigma ( $\sigma$ ) é usada na estatística para representar o desvio padrão populacional de uma distribuição. O valor de sigma pode ser estimado pelo desvio padrão amostral. Então, sigma ou, para fins práticos, desvio padrão é uma estatística que quantifica a variabilidade em um processo, resposta ou característica. A Figura 5 mostra que as áreas sob a curva de uma distribuição normal de dados centrada na média do processo (que é a distribuição que se espera ter por indicar, estatisticamente, que se tem um processo estável e, portanto, passível de sofrer melhoria) apresentam probabilidades relacionadas aos desvios padrões. A área definida pela média mais ou menos seis desvios padrões (seis sigma) é de 99,999998%, que representa a porcentagem de componentes livres de erros. A área fora da região abrangida pela área de seis sigma é igual à área total da distribuição, a qual tem o valor de 100%, subtraída de 99,999998%. Isto resulta no valor de 0,000002%, que indica a porcentagem de componentes com erros. Estes últimos, portanto, correspondem a dois erros por bilhão ou 0,002 erros por milhão.

Figura 5: Áreas sob a curva da distribuição normal.



Fonte: Dos autores.

É importante ressaltar que, na prática, existem mudanças na execução de um processo devido a variações. Estas mudanças, no entanto, não excedem 1,5 sigma (KAN, 1995). Assim, ao se considerar esta variação de mais ou menos 1,5 sigma, chega-se ao valor de, praticamente, zero erro por um lado e 3,4 erros por milhão por outro (o primeiro valor diz respeito ao limite de 7,5 sigma e o segundo ao limite de 4,5 sigma). Ou seja, na pior das hipóteses a metodologia garante o resultado de 3,4 erros por milhão, tendo sido este o valor proposto pela Motorola Incorporation, fabricante de equipamentos eletrônicos, para o nível de qualidade seis sigma.

De acordo com Rotondaro et al. (2002), a metodologia seis sigma usa ferramentas e métodos estatísticos para identificar os problemas que necessitam ser resolvidos, medir com o fim de obter informação, analisar a informação mensurada, incorporar e implementar melhorias nos processos, e controlar os processos ou produtos. Busca-se, com isso, gerar um ciclo de melhoria contínua.

### A norma ISO/IEC/IEEE 15939

Muito esforço tem sido feito no sentido de se estabelecer um conjunto de práticas, ferramentas e serviços, para auxiliar os gerentes de projetos de software a obter informações objetivas sobre os projetos em andamento, a fim de que estes atinjam suas metas de prazo, custo e qualidade. A ideia é, com a participação de órgãos governamentais, universidades e empresas privadas, reunir as práticas de maior sucesso utilizando mensuração das comunidades de software e engenharia, e compor um conjunto de diretrizes e recomendações para a implantação de uma política de medição na

organização. O PSM – *Practical Software Measurement* – representa uma importante iniciativa nesse sentido.

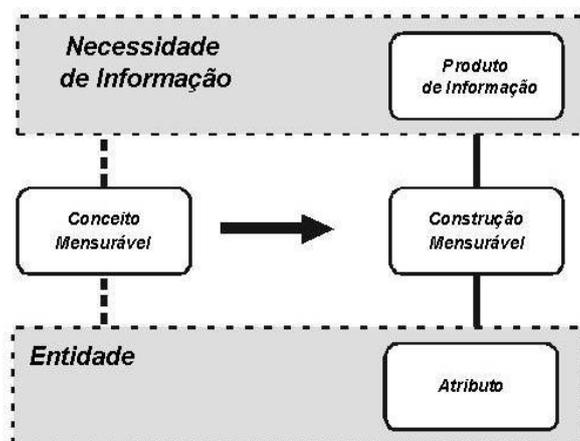
O PSM consiste em um processo objetivo de mensuração e análise de instâncias<sup>a</sup> de projeto, riscos e custos, para projetos de desenvolvimento de software (PSM Support Center, 2014). Os seus princípios foram formalizados em um padrão, o ISO/IEC/IEEE 15939 (IEEE, 2009), que estabeleceu algumas convenções terminológicas. Assim, a norma ISO/IEC/IEEE 15939 se apresenta estruturada em dois modelos, referentes às atividades de mensuração:

- Modelo de informação de medição;
- Processo de medição.

### O modelo de informação de medição da norma ISO/IEC/IEEE 15939

O modelo de informação fornece uma estrutura que relaciona as necessidades de informação a um conjunto de medidas. Salviano (2006) destaca que “O modelo de medição define o Produto de Informação para satisfazer uma Necessidade de Informação. Este Produto de Informação é produzido por uma Construção Mensurável a partir de Atributos que representam Entidades”. E ainda: “Um Conceito Mensurável é uma ideia sobre como uma Necessidade de Informação pode ser satisfeita a partir de Entidades. Um Conceito Mensurável pode ser implementado por diferentes Construções Mensuráveis”. Isso é ilustrado na Figura 6, a seguir.

Figura 6. Visão geral do modelo ISO/IEC/IEEE 15939.



Fonte: Adaptado de Salviano (2006).

Uma construção mensurável é composta por medidas em três níveis diferentes:

- Medida básica;
- Medida derivada;
- Indicador.

A medida básica inclui um atributo mensurável de uma entidade do processo ou produto, um método de medição para a quantificação de um atributo, e um valor resultante da aplicação do método. Pode-se dizer que, relacionados a essa medida, são os conceitos de escala de medição, unidade de medição, observação (ato de designar um valor) e unidade de observação (SALVIANO, 2006). Exemplos de medidas básicas são: custo (R\$), tamanho (pode ser mensurado em linhas de código – LOC), número de erros, esforço (pode ser mensurado em pessoas.mês), etc.

A medida derivada incorpora informações sobre dois ou mais atributos, ou várias observações de um mesmo atributo. Ela inclui uma função matemática que combina dois ou mais valores de

<sup>a</sup> Instância: fator crítico que descreve situações que exigem atenção para a satisfação das necessidades de informação do negócio da organização.

medidas básicas e/ou derivadas, e um valor resultante da aplicação da função. Exemplo de medida derivada é a produtividade ( $\text{produtividade} = \text{tamanho} / \text{esforço}$ ).

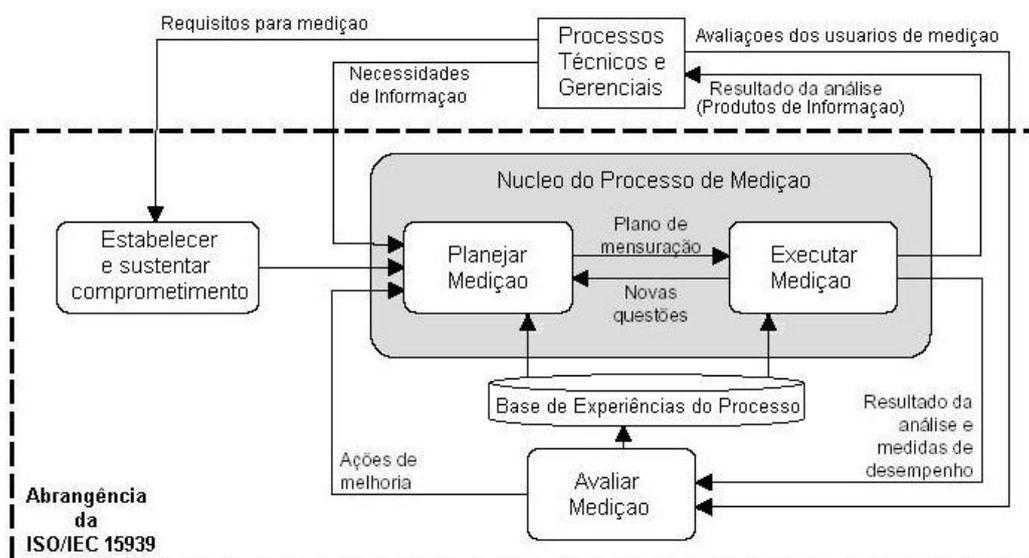
O indicador é uma medida que fornece uma estimativa ou avaliação de atributos específicos, com respeito a uma necessidade de informação. Ele inclui um ou mais valores de medidas básicas e/ou derivadas, um modelo de análise que combina os diversos valores, um valor resultante da aplicação deste modelo, e um critério de decisão para a avaliação do valor do indicador.

### O processo de medição da norma ISO/IEC/IEEE 15939

O processo de medição (Figura 7) consiste numa estrutura para a implementação da mensuração em projetos. Ele envolve as atividades de:

- Estabelecer e sustentar o comprometimento para a medição;
- Planejar a medição;
- Realizar a medição;
- Avaliar a medição.

Figura 7. Modelo do processo de medição na norma ISO/IEC/IEEE 15939.



Fonte: Adaptado de Salviano (2006).

### Medição e análise – aplicação numa empresa

Uma experiência real de aplicação das metodologias conjugadas foi realizada, por meio de estudo de caso, numa empresa desenvolvedora de software, de pequeno porte, a XYZ (o nome é fictício, com vistas a resguardar a privacidade da empresa). Os dados e as estimativas usadas aqui se referem a um projeto de software concluído pela empresa, a qual adota o ciclo de vida iterativo incremental no processo de desenvolvimento de software.

### Metodologia de aplicação

Tendo em vista sair do nível 1 de maturidade na representação por estágios do CMMI em direção ao nível 2 a empresa, em questão, tem buscado angariar o apoio de todos os envolvidos, no sentido de estabelecer e sustentar o comprometimento para a medição (1ª atividade do processo de medição da ISO/IEC/IEEE 15939). Assim, tem-se caminho aberto para as demais atividades relacionadas à medição.

Os dados e as estimativas utilizadas para análise dizem respeito, de acordo com as necessidades de informação, a três instâncias (entidades para o modelo de informação de medição): cronograma (prazo), custo e qualidade. Com relação a esta última, cabe ressaltar que falar em gestão da qualidade é sinônimo de empenhar-se pela redução de erros no processo de desenvolvimento e no produto (software, neste trabalho). Assim, a instância qualidade diz respeito aqui à detecção e remoção de erros, de modo que sejam atendidas as especificações do cliente.

No planejamento da medição (2ª atividade do processo de medição da ISO/IEC/IEEE 15939), procurou-se relacionar às necessidades de informação identificadas as instâncias mencionadas anteriormente e estabelecer medidas para satisfazer a tais necessidades de informação:

i) Necessidade de informação: Qual o desvio (diferença) entre o prazo real observado e o prazo estimado para cada fase do projeto?

- Entidade: cronograma.
- Atributo: hora.
- Método de medição: contar total de horas observadas e estimadas para cada fase do projeto.
- Medidas básicas: horas observadas e horas estimadas.
- Indicador: estudo da correlação entre os valores estimados e observados, mediante gráfico de dispersão, sendo que quanto mais próximos os pontos representativos dos valores pareados de uma reta bissetriz (cabe aqui a ressalva de que as escalas dos eixos cartesianos são as mesmas), mais próximo de 1 o valor para o coeficiente de correlação linear e menor o desvio entre o prazo observado e o estimado (maior a precisão<sup>b</sup> das estimativas).

ii) Necessidade de informação: Qual o desvio (diferença) entre o custo real observado e o custo estimado para cada fase do projeto?

- Entidade: custo.
- Atributo: unidade monetária (UM).
- Método de medição: contar total de UM's observadas e estimadas para cada fase do projeto.
- Medidas básicas: UM's observadas e UM's estimadas.
- Indicador: estudo da correlação entre os valores estimados e observados, mediante gráfico de dispersão.

iii) Necessidade de informação: o desvio (diferença) entre o prazo real observado e o prazo estimado para cada atividade (cada fase do projeto é composta de várias atividades) do projeto segue uma distribuição de dados normal (há indícios de um processo de estimativas de prazo estável)?

- Entidade: cronograma.
- Atributo: hora.
- Método de medição: contar total de horas observadas e estimadas para cada atividade do projeto em todas as suas fases.
- Medidas básicas: horas observadas e horas estimadas.
- Função de medição: subtrair total de horas estimadas do total de horas observadas para cada atividade do projeto em todas as suas fases.
- Medida derivada: diferença entre horas observadas e horas estimadas.
- Indicador: análise de distribuição dos valores oriundos da diferença entre os valores observados e estimados, e da média e desvio padrão desta distribuição, mediante

---

<sup>b</sup> Precisão na medição diz respeito ao grau de concordância entre valores medidos, obtidos por medições repetidas no mesmo objeto (Vocabulário Internacional de Metrologia: Conceitos fundamentais e gerais e termos associados (VIM 2012). 1ª ed. Duque de Caxias: INMETRO, 2012. 94p. Disponível em: <[http://www.inmetro.gov.br/inovacao/publicacoes/vim\\_2012.pdf](http://www.inmetro.gov.br/inovacao/publicacoes/vim_2012.pdf)>. Acesso em: 20 dez. 2014).

histograma. Interpreta-se que o perfil desejado para o histograma é o de uma distribuição normal, pois isso garante que apenas erros aleatórios estão presentes no processo de estimativas, de maneira que esse processo apresenta estabilidade e pode sofrer melhoria com a redução do desvio padrão.

iv) Necessidade de informação: o desvio (diferença) entre o custo real observado e o custo estimado para cada atividade do projeto segue uma distribuição de dados normal (há indícios de um processo de estimativas de custo estável)?

- Entidade: custo.
- Atributo: unidade monetária (UM).
- Método de medição: contar total de UM's observadas e estimadas para cada atividade do projeto em todas as suas fases.
- Medidas básicas: UM's observadas e UM's estimadas.
- Função de medição: subtrair total de UM's estimadas do total de UM's observadas para cada atividade do projeto em todas as suas fases.
- Medida derivada: diferença entre UM's observadas e UM's estimadas.
- Indicador: análise de distribuição dos valores oriundos da diferença entre os valores observados e estimados, e da média e desvio padrão desta distribuição, mediante histograma.

v) Necessidade de informação: Quais os tipos de erros mais comuns no processo de desenvolvimento?

- Entidade: qualidade.
- Atributo: erro.
- Método de medição: contar total de erros descobertos por tipo no projeto.
- Medidas básicas: erros descobertos.
- Indicador: análise gráfica dos valores de erros descobertos por tipo, mediante gráfico de Pareto. Interpreta-se que as barras verticais maiores (e mais à esquerda) indicam os tipos de erros com maior ocorrência.

A 3ª atividade do processo de medição da ISO/IEC/IEEE 15939, que consiste na realização da medição, foi feita por meio do software R versão 3.4.0 (R CORE TEAM, 2014), usado na implementação dos histogramas, e do software LibreOffice Calc (LibreOffice, 2014), usado na análise de correlação e de Pareto.

A avaliação da medição realizada (4ª atividade do processo de medição da ISO/IEC/IEEE 15939) foi feita tendo sempre em vista a correta interpretação dos indicadores de cronograma, custo e qualidade, delineados no plano da medição.

## Resultados e discussão

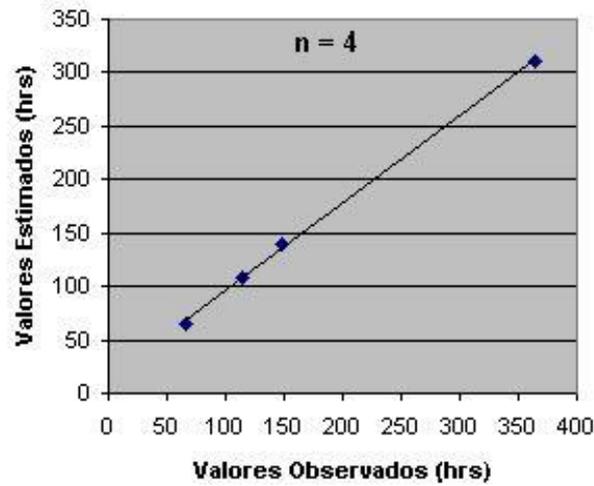
Em relação às necessidades de informação (i) e (ii), o estudo da correlação dos valores de cronograma (Tabela 1) e de custo (Tabela 2) apresentou coeficientes de correlação linear muito próximos de 1 (0,9994 para cronograma e 0,9993 para custo), indicando precisão muito alta para as estimativas, algo inusitado na indústria de software. Isso pode ser visto pela distribuição dos pontos muito próxima à reta bissetriz nas Figuras 8 e 9.

Tabela 1. Valores do cronograma das fases do projeto de software.

Fase	Cronograma observado (h)	Cronograma estimado (h)
Prospecção	66	64
Planejamento	147,67	140
Desenvolvimento (iterações 1 e 2)	363,68	311,32
Fechamento	115	109

Fonte: Dos autores.

Figura 8. Gráfico de dispersão para o cronograma das fases do projeto de software – valores estimados *versus* valores observados.



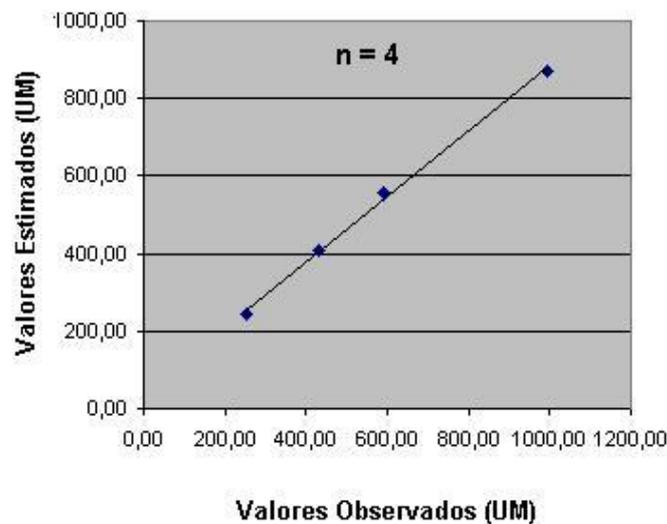
Fonte: Dos autores.

Tabela 2. Valores do custo das fases do projeto de software.

Fase	Custo Observado (UM)	Custo Estimado (UM)
Prospecção	252,00	245,50
Planejamento	592,94	555,00
Desenvolvimento (iterações 1 e 2)	992,79	871,43
Fechamento	430,63	408,13

Fonte: Dos autores.

Figura 9. Gráfico de dispersão para o custo das fases do projeto de software – valores estimados *versus* valores observados.

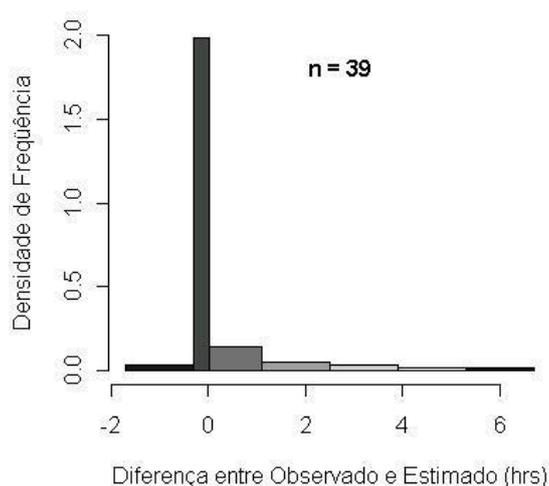


Fonte: Dos autores.

Sugeriu-se, então, que medições sejam feitas pela empresa em atributos dos produtos de software, tais como usabilidade<sup>c</sup>, manutenibilidade<sup>d</sup>, etc. Estes podem estar sendo preteridos em decorrência do atendimento de alvos de cronograma e custo. Além disso, sugeriu-se que, com o auxílio de cartas de controle de processo para variáveis, e com base em valores adicionais, oriundos da medição dos futuros projetos de categoria similar<sup>e</sup>, um acompanhamento dos valores de cronograma e custo seja feito, a fim de se verificar se correspondem a processos estáveis e podem ser melhorados ainda mais, mediante investigação de causas de variação.

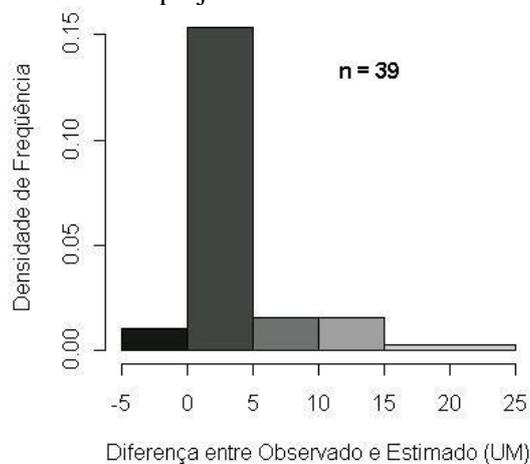
Em relação às necessidades de informação (iii) e (iv), histogramas para o cronograma e para o custo detalhados, abrangendo as várias atividades que compõem cada uma das quatro fases do projeto, num total de trinta e nove atividades ( $n = 39$ ), podem ser vistos nas Figuras 10 e 11. O histograma para o cronograma detalhado apresentou média  $\bar{x} = 0,570$  horas e desvio padrão  $s_x = 1,366$  horas. O histograma para o custo detalhado apresentou média  $\bar{x} = 2,118$  UM e desvio padrão  $s_x = 5,104$  UM.

Figura 10. Histograma da diferença entre valores observados e estimados do cronograma detalhado do projeto de software.



Fonte: Dos autores.

Figura 11. Histograma da diferença entre valores observados e estimados do custo detalhado do projeto de software.



Fonte: Do autor (2014).

<sup>c</sup> Usabilidade: grau em que o software é fácil de entender e usar.

<sup>d</sup> Manutenibilidade: facilidade com a qual reparos podem ser feitos no software.

<sup>e</sup> Projetos de categoria similar: não apresentam diferença significativa quanto a fatores como processo de desenvolvimento, nível de tecnologia, linguagem de programação, tamanho do programa, etc.

Os histogramas são assimétricos à direita indicando a presença de causa de variação especial “puxando” a distribuição das atividades para valores maiores (indesejáveis), o qual foi identificado como tendo sua origem nas fases de planejamento e fechamento do projeto. No decorrer das medições dos futuros projetos de mesma categoria, deve-se continuar fazendo um estudo das distribuições das diferenças entre os valores observados e estimados, a fim de se verificar os comportamentos destas distribuições, e dos valores de suas médias e desvios padrões; isto é importante, pois a visualização de uma tendência das distribuições em direção ao modelo normal, e de reduções nos valores de média e variação, é sinal da ocorrência de melhoria no processo de desenvolvimento.

No que tange à necessidade de informação (v), convém ressaltar que a empresa desenvolvedora de software em questão trabalha com dois tipos de erro: por gravidade e por categoria.

Conforme significado dado pela referida empresa, os erros por gravidade seguem uma escala ordinal de mensuração, sendo o de maior prioridade o de gravidade grande, seguido pelo de gravidade pequena, obstáculo e texto. Os erros por categoria seguem uma escala nominal de mensuração: erros de implementação de código, erros na construção do banco de dados, erros de padronização de interface, erros de incompatibilidade da funcionalidade implementada com a especificação dos requisitos funcionais (RF), *bugs* ou erros por interpretação dúbia da funcionalidade requerida.

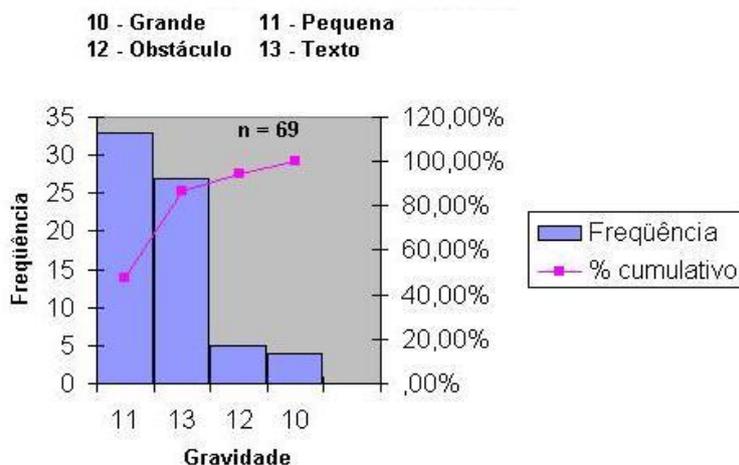
A análise de Pareto mostrou que a maior parte dos erros descobertos era de gravidade pequena (Tabela 3 e Figura 12) e relacionada à incompatibilidade da funcionalidade implementada com a especificação dos requisitos funcionais (Tabela 4 e Figura 13).

Tabela 3. Erros descobertos por gravidade no projeto de software.

Gravidade:	Nº	Percentual
Grande	4	5,80%
Pequena	33	47,83%
Obstáculo	5	7,25%
Texto	27	39,13%

Fonte: Dos autores.

Figura 12: Gráfico de Pareto para os erros descobertos por gravidade no projeto de software.



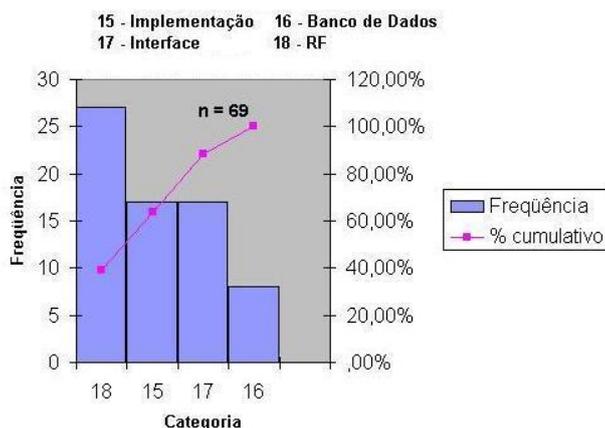
Fonte: Dos autores.

Tabela 4: Erros descobertos por categoria no projeto de software.

Categoria:	Nº	Percentual
Implementação	17	24,64%
Banco de dados	8	11,59%
Interface	17	24,64%
RF	27	39,13%

Fonte: Dos autores.

Figura 13. Gráfico de Pareto para os erros descobertos por categoria no projeto de software.



Fonte: Dos autores.

## Conclusão

A análise dos dados, no estudo de caso realizado, serviu para demonstrar a contribuição da metodologia proposta na gestão quantitativa, controle e melhoria de processos de software por meio dos seguintes pontos:

- Avaliação da precisão das estimativas de prazo e custo do projeto de software, a qual pode ser utilizada como um indicador da qualidade das estimativas;
- Estudo da distribuição dos dados oriundos da diferença entre valores observados e estimados do cronograma do projeto de software, e dos dados oriundos da diferença entre valores observados e estimados do custo, a fim de se averiguar qual perfil seguem na medição do projeto, bem como dos valores de suas médias e desvios padrões, para ver como se comportam diante das atividades de controle e melhoria;
- Identificação das fontes causadoras da maior parte dos erros no processo de desenvolvimento, contribuindo para a sua remoção e conseqüente redução de retrabalho (e, por tabela, de custo e prazo), mediante investigação de causas;
- Preparação para o acompanhamento dos futuros projetos, mediante cartas de controle de processo, as quais permitirão comparar medições futuras com as atuais, de maneira a se verificar a estabilidade dos processos de desenvolvimento, alcançar previsibilidade e identificar oportunidades de melhoria;
- Auxílio para a implantação de uma sistemática, calcada no pensamento científico e objetivo, com vistas a compor um banco de dados históricos adequado, fruto de medições efetivas, alinhadas às necessidades de informação e metas da organização (coerente com o preconizado nas metas específicas da área de medição e análise do modelo CMMI);
- Diagnóstico dos processos com o uso da estatística, propiciando o vislumbre e decisão de ações gerenciais mais lúcidas e “certeiras” para melhoria de processos de software.

## Referências bibliográficas

CMMI PRODUCT TEAM. *CMMI for Development, Version 1.3*. Pittsburgh: Software Engineering Institute, Carnegie Mellon University, 2010. Disponível em: <http://www.sei.cmu.edu/>. Acesso em: 04 jul. 2014.

IEEE. *IEEE Standard Adoption of ISO/IEC 15939:2007: Systems and software engineering – Measurement process*. New York: IEEE, 2009. 38 p.

KAN, S.H. *Metrics and models in software quality engineering*. Reading: Addison-Wesley, 1995. 344 p.

LibreOffice: The Document Foundation. Disponível em: <http://www.libreoffice.org/>. Acesso em: 08 de dezembro de 2014.

PAULA FILHO, W.P. *Engenharia de software: fundamentos, métodos e padrões*. Rio de Janeiro: LTC, 2001. 584 p.

PAULK, M.C.; WEBER, C.V.; CURTIS, B.; CHRISSIS, M.B. *The Capability Maturity Model: guidelines for improving the software process*. Reading: Addison-Wesley, 1994. 441 p.

PSM Support Center. *PSM – Practical Software e Systems Measurement*. Disponível em: <http://www.psmc.com>. Acesso em: 10 jul. 2014.

R CORE TEAM. *R: A language and environment for statistical computing*. Vienna: R foundation for Statistical Computing, 2017. Disponível em: <http://www.r-project.org>. Acesso em: 13 jun. 2014.

ROTONDARO, R.G.; RAMOS, A.W.; RIBEIRO, C.O.; MIYAKE, D.I.; NAKANO, D.; LAURINDO, F.J.B.; HO, L.L.; CARVALHO, M.M. DE.; BRAZ, M.A.; BALESTRASSI, P.P. *Seis Sigma: estratégia gerencial para a melhoria de processos, produtos e serviços*. São Paulo: Atlas, 2002. 375 p.

SALVIANO, C.F. *Uma Proposta Orientada a Perfis de Capacidade de Processo para Evolução da Melhoria de Processo de Software*. 2006. 246 p. Tese (Doutorado em Engenharia Elétrica e de Computação) – Universidade Estadual de Campinas, Campinas, 2006.