



O Método COSMIC para Medição de Tamanho Funcional
Versão 3.0

Visão Geral
do Método

Setembro de 2007

AGRADECIMENTOS

Autores do Método¹ 'COSMIC-FFP' VERSÃO 2.0 (em ordem alfabética)

Alain Abran, École de technologie supérieure – Université du Québec,
Jean-Marc Desharnais, Software Engineering Laboratory in Applied Metrics - SELAM,
Serge Oigny, Bell Canada,
Denis St-Pierre, DSA Consulting,
Charles Symons, Software Measurement Services, UK

Revisores da Versão 2.0 1998/1999 (em ordem alfabética)		
Moritsugu Araki, JECS Systems Research, Japan	Thomas Fetcke, Germany	Patrice Nolin, Hydro Québec, Canada
Fred Bootsma, Nortel, Canada	Eric Foltin, University of Magdeburg, Germany	Marie O'Neill, Software Management Methods, Ireland
Denis Bourdeau, Bell Canada, Canada	Anna Franco, CRSSM, Canada	Jolijn Onvlee, The Netherlands *
Pierre Bourque, , École de Technologie supérieure, Canada	Paul Goodman, Software Measurement Services, United Kingdom	Laura Primera, UQAM, Canada
Gunter Guerhen, Bürhen & Partner, Germany	Nihal Kececi, University of Maryland, United States	Paul Radford, Charismatek, Australia
Sylvain Clermont, Hydro Québec, Canada	Robyn Lawrie, Australia	Eberhard Rudolph, Germany
David Déry, CGI, Canada	Ghislain Lévesque, UQAM, Canada	Grant Rule, Software Measurement Services, United Kingdom*
Gilles Desoblins, France	Roberto Meli, Data Processing Organization, Italy	Richard Stutzke, Science Applications Int'l Corporation, United States
Martin D'Souza, Total Metrics, Australia	Pam Morris, Total Metrics, Australia*	Ilionar Sylva, UQAM, Canada
Reiner Dumke, University of Magdeburg, Germany	Risto Nevalainen, Software Technology Transfer Finland, Finland *	Vinh T. Ho, UQAM, Vietnam
Peter Fagg, United Kingdom	Jin Ng, Hmaster, Australia	

* Membros fundadores da Equipe Principal COSMIC, juntamente com os autores do Método COSMIC-FFP v2.0

¹ A Versão 2.0 foi a primeira versão do método COSMIC-FFP disponível para o público, como o método foi inicialmente conhecido.

Revisores da Versão 3.0 2006/07 (em ordem alfabética)		
Alain Abran, École de Technologie Supérieure, Université du Québec, Canada	Jean-Marc Desharnais, Software Engineering Lab in Applied Metrics – SELAM, Canada	Arlan Lesterhuis*, Sogeti, The Netherlands
Bernard Londeix, Telmaco, United Kingdom	Roberto Meli, Data Processing Organization, Italy	Pam Morris, Total Metrics, Australia
Serge Oigny, Bell Canada	Marie O'Neill, Software Management Methods, Ireland	Tony Rollo, Software Measurement Services, United Kingdom
Grant Rule, Software Measurement Services, United Kingdom	Luca Santillo, Agile Metrics, Italy	Charles Symons*, United Kingdom
Hannu Toivonen, Nokia Siemens Networks, Finland	Frank Vogelezang, Sogeti, The Netherlands	

* Editores da versão 3.0 do método COSMIC

Tradução Brasileira

- Maria Cecilia Techy, TI Métricas, São Paulo, Brasil (2012) - tradutor.
- Mauricio Aguiar, TI Métricas, Rio de Janeiro, Brasil (2012) – tradutor e revisor. Comentários sobre a tradução brasileira devem ser enviados ao revisor via mauricio@metricas.com.br.

Copyright 2007. Todos os Direitos Reservados. The Common Software Measurement International Consortium (COSMIC). A permissão para copiar este material no todo ou em parte é concedida desde que as cópias não sejam feitas ou distribuídas para a obtenção de vantagem comercial e que o título da publicação, o respectivo número de versão e data sejam citados, assim como seja citado que a cópia foi efetuada com permissão do Common Software Measurement International Consortium (COSMIC). Cópias que não se enquadrarem no critério acima requerem permissão específica.

Uma versão de domínio público da documentação do Método COSMIC, inclusive traduções para outros idiomas, pode ser encontrada na Web em www.cosmicon.com.

CONTROLE DE VERSÃO

A tabela abaixo resume as mudanças neste documento 'Visão Geral'

DATA	REVISOR(ES)	Mudanças/Acréscimos
Setembro de 2007	Comitê de Práticas de Medição COSMIC	Primeira versão para divulgação ao público. O conteúdo é parcialmente baseado no capítulo 2 do Manual de Medição v2.2.

NOTAS INTRODUTÓRIAS

O método COSMIC é um método padrão para a medição do tamanho funcional de software correspondente aos domínios funcionais normalmente denominados software para “aplicações de negócio” (ou ‘MIS’), software ‘real-time’ e combinações dessas duas categorias.

O método COSMIC foi aceito pelo ISO/IEC JTC1 SC7 em dezembro de 2002 como o Padrão Internacional ISO/IEC 19761 ‘Engenharia de Software – COSMIC-FFP – Um método para a medição funcional de tamanho’ (daqui em diante referenciado como ‘ISO/IEC 19761’).

Propósito deste documento

O propósito desta ‘Visão Geral do Método’ é fornecer um resumo do Método COSMIC para Medição de Tamanho Funcional, versão 3.0. Consideramos que este documento é de interesse para leitores que

- Precisam de uma visão geral do método sem precisar conhecer todos os seus detalhes
- São novos no conceito de medição de tamanho funcional de software, ou que precisem de uma introdução ao tema
- Estão familiarizados com algum método de medição de tamanho funcional de ‘1ª. Geração’ (tais como os métodos ‘IFPUG’, ‘MkII’ ou ‘NESMA’) e/ou que estão querendo aprofundar-se no método COSMIC.

Documentação do Método COSMIC

Para uma completa documentação do método COSMIC, consulte o documento ‘Visão Geral da Documentação e Glossário de Termos’. O Glossário define todos os termos comuns a todos os documentos COSMIC. Este documento e todos os outros documentos mencionados abaixo podem ser baixados gratuitamente do site www.cosmicon.com

Para maior clareza, outros documentos de interesse são os seguintes.

- A norma ISO/IEC 19761, que contém as definições normativas e regras fundamentais do método.
- O ‘Método COSMIC versão 3.0: Manual de Medição’, que apresenta tais regras e definições, além de explicações adicionais e exemplos com o objetivo de ajudar os medidores a compreender e aplicar o método. Deverá ser o principal ‘documento de trabalho’ que os medidores irão precisar na prática.
- O ‘Método COSMIC versão 3.0: Tópicos Avançados e Relacionados’, contém informações além do método básico, tais como medições aproximadas e na fase inicial do projeto, bem como a convertibilidade entre outros métodos de medição e o COSMIC.

Outras documentações disponíveis sobre COSMIC incluem Guias para aplicação do método em domínios específicos de software, vários Estudos de Caso ilustrando o uso do método, artigos de pesquisa, dados de benchmark, etc. Traduções do Manual de Medição para outros idiomas também estão disponíveis. Todos os documentos podem ser encontrados em www.cosmicon.com.

Informações de apoio sobre a medição de tamanho funcional e sua utilização, vantagens do método COSMIC, a organização COSMIC e suas atividades, fornecedores de serviços relacionados ao COSMIC, Newsletters COSMIC, etc., podem ser encontradas em <http://www.cosmicon.com>.

Comitê de Práticas de Medição COSMIC

CONTEÚDO

INTRODUÇÃO	7
VISÃO GERAL DO MÉTODO DE MEDIÇÃO COSMIC	9
2.1 Aplicabilidade do método COSMIC.....	10
2.1.1 Domínios aplicáveis.....	10
2.1.2 Domínio não-aplicável.....	10
2.2 Os modelos de software COSMIC	10
2.2.1 Requisitos Funcionais do Usuário	10
2.2.2 O Modelo de Contexto de Software COSMIC.....	11
2.2.3 O Modelo Genérico de Software COSMIC.....	18
2.3 Visão Geral do Processo de Medição COSMIC.....	22
2.3.1 A Fase Estratégia de Medição	22
2.3.2 A Fase de Mapeamento.....	23
2.3.3 A Fase de Medição	24
APÊNDICE A - PROCEDIMENTO COSMIC PARA SOLICITAÇÃO DE MUDANÇA E COMENTÁRIOS.....	25

INTRODUÇÃO

O software é um componente importante em muitos orçamentos corporativos. As organizações reconhecem a importância de controlar as despesas com software e analisar o desempenho dos montantes alocados no desenvolvimento e manutenção de software, com o objetivo de se avaliarem em relação aos melhores da área. Para tanto, são necessárias medidas para analisar tanto a produtividade quanto a qualidade associados ao desenvolvimento e manutenção de software. Por um lado, desenvolvedores necessitam de medidas técnicas para quantificar o desempenho técnico de produtos ou serviços. Medidas técnicas podem ser utilizadas, por exemplo, para a análise de eficiência ou para melhorar o desempenho do design, etc.

Por outro lado, medidas funcionais são necessárias, por exemplo, aos desenvolvedores para estimar ou medir o tamanho do software a partir de requisitos no início do ciclo de vida de um projeto, como principal entrada para a estimativa de esforço do mesmo, ou análise da produtividade de produtos ou serviços da perspectiva do usuário ou proprietário. As medidas funcionais devem ser independentes do desenvolvimento técnico e decisões de implementação. Desta forma podem ser utilizadas para comparar as produtividades obtidas através de diferentes técnicas e tecnologias.

A Análise de Pontos de Função (APF)² é um exemplo de um método de medição do tamanho funcional. Está disponível para o domínio de aplicações de negócio (MIS), no qual tem sido amplamente utilizada na análise de produtividade e estimativas (Abran, 1996; Desharnais, 1988; Jones, 1996; Kemerer, 1987). Consegue capturar com sucesso as características funcionais específicas de software do tipo MIS.

No entanto, a APF tem sido criticada por não poder ser aplicada a todos os tipos de software [Conte, 1986; Galea, 1995; Grady, 1992; Hetzel, 1993; Ince, 1991; Jones, 1988; Jones, 1991; Kan, 1993; Whitmire, 1992]. Em particular, a APF não foi bem aceita pela comunidade de aplicações do tipo 'real-time'.

O método 'Full Function Point' (versão 1.0) foi proposto em 1997³ com o objetivo de ser uma extensão da APF para capturar o tamanho funcional de aplicações em tempo real e de aplicações técnicas e de sistema. Os testes de campo mostraram que o FFP é também adequado para medir o tamanho funcional de aplicações do tipo MIS, levando a resultados semelhantes⁴.

Em 1998, o grupo de FFP somou seus esforços ao trabalho do grupo COSMIC⁵, propondo os princípios de uma segunda geração de métodos para a medição de tamanho funcional. Estes esforços resultaram no primeiro 'teste de campo' disponível ao público do método de medição COSMIC-FFP versão 2.0, publicado em outubro de 1999.

² Albrecht A.J., Gaffney Jr. J.E., "Software function, source lines of code and development effort prediction: a software science validation", IEEE Transactions on Software Engineering, Vol. SE-9, pp. 639-648, Novembro 1983. A 'APF' é atualmente conhecida como o método do 'IFPUG'.

³ St-Pierre D., Maya M., Abran A., Desharnais J.-M., Bourque P., "Full Function Points: Counting Practices Manual", Technical Report 1997-04, Université du Québec à Montréal, Montréal, Canada. Disponível na Web em URL: www.lrgl.uqam.ca/ffp.html

⁴ Oligny, S.; Abran, A.; Desharnais, J.-M.; Morris, P., *Functional Size of Real-Time Software: Overview of Field Tests*, in Proceedings of the 13th International Forum on COCOMO and Software Cost Modeling, Los Angeles, CA, Outubro 1998.

⁵ Para informações adicionais consulte www.cosmicon.com.

Da versão 2.0 em diante, o método de medição COSMIC também foi desenhado para garantir a sua total conformidade com a norma ISO / IEC 14143-1: 1998 (e, posteriormente, ISO / IEC 14143-1: 2007), bem como com os princípios COSMIC.

A partir da versão 3.0, o nome do método foi simplificado de 'COSMIC-FFP' para 'COSMIC'.

A respeito da iniciativa COSMIC

Em função do crescimento explosivo e diversidade de contratação e outsourcing de software, fornecedores e clientes necessitam de métodos mais precisos de estimativa e de medição de desempenho. Esses métodos devem funcionar de forma igualmente confiável para todos os tipos de software. Os métodos de 'primeira geração' para medição de tamanho de software não são suficientemente fortes para atender as necessidades do mercado, ou funcionam apenas para alguns tipos de software. A indústria precisa urgentemente de medidas de tamanho de software comprovadamente mais acuradas e mais abrangentes.

O grupo COSMIC visa atender a essas necessidades. Primeiro, de fornecedores de software diante da tarefa de traduzir os requisitos do usuário em tamanho do software a ser produzido, como uma etapa chave na estimativa de custos do projeto e, segundo, de clientes que querem saber o tamanho funcional do software que será entregue, como um componente importante da medição do desempenho do fornecedor.

COSMIC, "COmmon Software Measurement International Consortium", é uma iniciativa voluntária de um grupo realmente internacional de especialistas em medição de software, praticantes e acadêmicos, da Ásia/Pacífico, Europa e América do Norte. Os objetivos originais do projeto COSMIC eram desenvolver, testar, trazer e buscar a aceitação do mercado para um novo método de medição de tamanho com o objetivo de apoiar a estimativa e medições de desempenho. Tais objetivos foram agora atingidos e o método é aceito em um crescente número de organizações dos setores públicos e privados em todo o mundo.

Depois que os princípios do método COSMIC foram estabelecidos em 1999, testes de campo foram conduzidos com sucesso, em 200/01, envolvendo diversas organizações e instituições acadêmicas internacionais. Os artigos descrevendo os resultados destes testes e muitas outras descobertas de pesquisas estão listados no site www.cosmicon.com. Em 2001 foi iniciado o processo de desenvolvimento de um Padrão Internacional para o método COSMIC. O padrão foi aprovado em dezembro de 2002 e publicado pela ISO no início de 2003 como ISO/IEC 19761.

COSMIC continua refinando a definição e explicação do método à luz de experiências práticas, porém **deve ser enfatizado que o Modelo Genérico de Software, o qual é a base da medição de tamanho, não foi alterado desde a sua primeira publicação em 1999**. A versão 3.0 do Manual de Medição é o mais recente passo no processo de refinamento, o qual continua sempre compatível com o padrão ISO/IEC 19761. A denominação de 'versão 3.0' comparada com 'versão 2.2' indica que v3.0 representa um importante passo em direção ao refinamento do método. Para uma lista completa das mudanças feitas na evolução da v2.2 para v3.0, veja 'O Método COSMIC v3.0: Manual de Medição'

O Common Software Measurement International Consortium (COSMIC) planeja que tais acréscimos e refinamentos sejam submetidos à ISO para inclusão na ISO/IEC 19761 na época de sua revisão em 2007/08.

Em 2006, COSMIC introduziu o primeiro exame de certificação 'Entry-level' para praticantes do método. Usuários do método COSMIC são encorajados a submeter informações de seus projetos para a base de dados do International Software Benchmarking Standards Group ('ISBSG'), para melhorar os dados de referência existentes relacionados a medidas utilizando o método COSMIC.

Para informações adicionais sobre COSMIC, suas publicações, atividades e exames, visite www.cosmicon.com. Para informações adicionais sobre o ISBSG, visite www.isbsg.org.

VISÃO GERAL DO MÉTODO DE MEDIÇÃO COSMIC

O método de medição COSMIC define uma medida padronizada de tamanho funcional de software. Este capítulo apresenta e discute:

- Os tipos de software para os quais o método foi desenhado (também conhecido como 'domínio de aplicabilidade' do método), na seção 2.1
- Os modelos de software utilizados para medição, na seção 2.2. Estes modelos introduzem todos os conceitos básicos do método COSMIC. O entendimento destes conceitos é importante porque para medir o tamanho funcional de um pedaço de software o medidor deve mapear, por exemplo, a definição de requisitos ou a implementação física a partir dos artefatos do pedaço de software para os conceitos dos modelos COSMIC
- A visão geral do processo de medição COSMIC, o qual consiste de três fases:
 - a Estratégia de Medição, executada antes de iniciar a medição (subseção 2.3.1)
 - a Fase de Mapeamento (subseção 2.3.2)
 - a Fase de Medição (subseção 2.3.3)

O resultado do processo de medição é uma medida de tamanho expressa em 'Pontos de Função COSMIC' (ou 'PFC'). Estas fases são apresentadas na figura 2.0 abaixo.

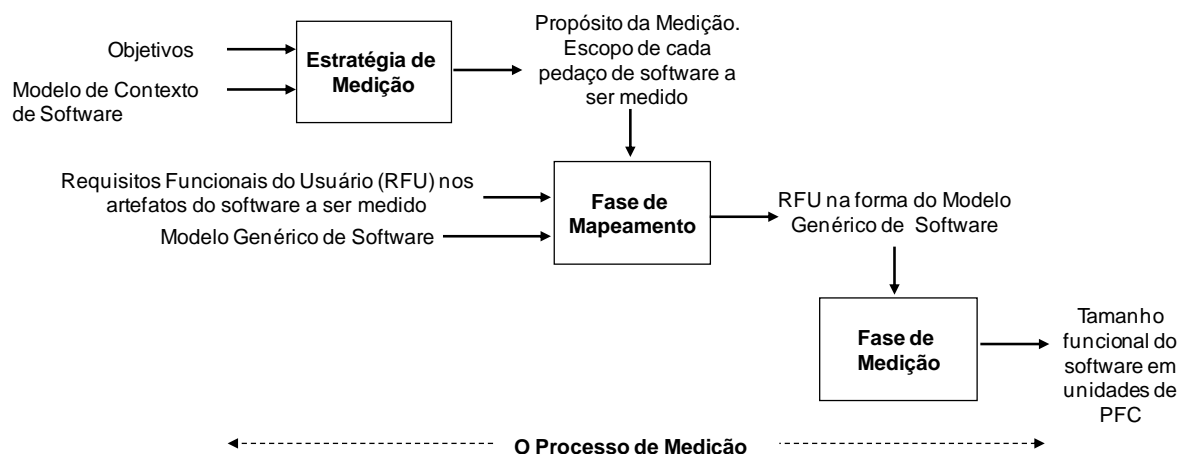


Figura 2.0 - Estrutura do Método COSMIC

As visões gerais contidas nas subseções 2.3.1, 2.3.2 e 2.3.3 são individualmente ampliadas em um capítulo do Manual de Medição, onde as definições completas e detalhadas, bem como os princípios e regras do método são apresentados com exemplos ilustrativos.

2.1 Aplicabilidade do método COSMIC

2.1.1 Domínios aplicáveis

O método COSMIC para medição de tamanho funcional foi projetado para ser aplicável à funcionalidade de software dos seguintes domínios:

- Software aplicativo de negócio normalmente necessário para apoiar a administração de negócios tais como: bancos, seguros, contabilidade, pessoal, compras, distribuição ou manufatura. Tal tipo de software é muitas vezes caracterizado como 'rico em dados', pois é amplamente dominado pela necessidade de gerenciar uma grande quantidade de dados sobre eventos do mundo real.
- Software de tempo-real, com a finalidade de acompanhar ou controlar eventos acontecendo no mundo real. Exemplos seriam: software para ligações telefônicas e troca de mensagens, software embarcado em dispositivos para controlar máquinas tais como eletrodomésticos, elevadores, motores de automóveis e aeronaves, controle de processos e aquisição automática de dados, assim como software contido no sistema operacional de computadores.
- Híbridos dos tipos de software acima, tais como sistemas em tempo-real para reservas de companhias aéreas e hotéis, por exemplo.

2.1.2 Domínio não-aplicável

O método de medição COSMIC ainda não foi projetado para levar em conta a funcionalidade de software intensivo em matemática, isto é, software caracterizado por algoritmos matemáticos complexos, ou outras regras especializadas e complexas, tais como sistemas especialistas, software para simulação, software para autoaprendizagem, sistemas para predição do tempo, etc., ou que processe variáveis contínuas, tais como sons de áudio ou imagens de vídeo, por exemplo: software para jogos de computador, instrumentos musicais, etc. Tampouco o método COSMIC tenta medir aspectos de funcionalidade, como 'complexidade' (seja lá como for definida) que poderiam ser considerados com contribuintes do 'tamanho' do software⁶.

No caso de software com tal funcionalidade é possível, no entanto, definir extensões locais ao método de medição COSMIC. O Manual de Medição explica em quais contextos tais extensões locais deveriam ser usadas e fornece exemplos de uma extensão local.

2.2 Os modelos de software COSMIC

Esta seção fornece uma visão geral do método COSMIC e de todos os seus conceitos básicos. As definições de todos estes conceitos são fornecidas no documento 'Método COSMIC v3.0: Visão Geral da Documentação e Glossário de Termos'. Nesta seção, na primeira vez em que um destes conceitos for utilizado o mesmo será apresentado em **negrito**.

É essencial que os medidores compreendam TODOS os modelos e conceitos básicos do COSMIC descritos abaixo e que sejam capazes de aplicar TODOS os princípios e regras descritas no Manual de Medição no momento da contagem. Somente com disciplina o medidor poderá ter certeza de que a medição é significativa e poderá ser repetida por outros medidores no mesmo software e/ou comparada com medições realizadas por outros medidores em ambientes de software diferentes.

2.2.1 Requisitos Funcionais do Usuário

O método de medição COSMIC envolve a aplicação de um conjunto de modelos, princípios, regras e processos aos **Requisitos Funcionais do Usuário** (ou '**RFU**') de um dado pedaço de software. O resultado é um número, o 'valor de uma quantidade'⁶ representando o tamanho funcional do pedaço

⁶ Como definido pela ISO, ver 'O Método COSMIC v3.0: Visão Geral da Documentação e Glossário de Termos'

de software de acordo com o método COSMIC em unidades de 'Pontos de Função COSMIC' (ou 'PFC')

O tamanho funcional produzido pelo método COSMIC foi projetado para ser independente de quaisquer decisões de implementação contidas nos artefatos operacionais do software a ser medido. A 'Funcionalidade' refere-se ao 'processamento da informação que o software deve realizar para seus usuários'.

Mais especificamente, uma declaração de RFU descreve 'o quê' o software deve fazer para os **usuários funcionais**. Estes são a fonte e o destino pretendido para os dados, de e para a funcionalidade requerida. Uma declaração de RFU exclui quaisquer requisitos técnicos ou de qualidade que dizem 'como' o software deve executar. Somente os RFU são levados em conta na medição do tamanho funcional.

Extraindo os requisitos funcionais do usuário dos artefatos de software, na prática

No mundo real de desenvolvimento de software é raro encontrar artefatos de software nos quais os RFU estejam claramente separados dos outros tipos de requisitos e expressos de uma forma adequada para a medição direta sem qualquer tipo de interpretação. Isto significa que geralmente o medidor terá que extrair os RFU conforme fornecidos, ou subentendidos nos artefatos reais do software, antes de mapeá-los para os conceitos dos 'modelos de software' COSMIC.

Os Requisitos Funcionais do Usuário podem ser derivados dos artefatos de engenharia de software produzidos antes que o software exista, tais como documentos de 'Definição de Requisitos', dados e funcionalidades resultantes da análise de requisitos, etc. Por isso o tamanho funcional do software pode ser medido antes de sua implementação em um sistema de computador.

Em algumas situações poderá ser necessário medir algum pedaço de software existente sem que estejam disponíveis artefatos de design ou arquitetura em quantidade suficiente, e os RFU podem não estar documentados (por exemplo, no caso de software legado). Em tais situações, ainda é possível derivar os RFU dos artefatos instalados no sistema de computador, tais como telas ou relatórios físicos, ou examinando o fluxo de dados, após a implementação.

Extraindo ou derivando os requisitos funcionais do usuário dos artefatos de software

O processo para extrair os RFU dos diferentes tipos de artefatos de engenharia de software, ou derivá-los a partir de software instalado e expressá-los na forma dos modelos de software COSMIC obviamente irá variar com os tipos de artefatos. Tais processos dependem do domínio e variam tanto que não podem ser tratados no método COSMIC. Este último assume que os requisitos funcionais do usuário do software a ser medido existam, ou que possam ser extraídos ou derivados de seus artefatos. No entanto, COSMIC também publica 'Guias' dependentes do domínio, que descrevem alguns aspectos da derivação de RFU.⁷

O método COSMIC, contudo, limita-se a descrever e definir os conceitos dos modelos de software COSMIC, isto é, os objetivos para a extração ou derivação. Tais conceitos encontram-se em um conjunto de princípios estabelecidos em dois modelos de software COSMIC – o 'Modelo de Contexto de Software' e o 'Modelo Genérico de Software'.

2.2.2 O Modelo de Contexto de Software COSMIC

Um pedaço de software a ser medido com o método COSMIC deve ser cuidadosamente definido (no **escopo** da medição) e tal definição deve levar em conta, em seu contexto, qualquer outro software

⁷ O 'Guia para Dimensionamento de Software Aplicativo de Negócios utilizando COSMIC' fornece orientações para o mapeamento de vários métodos de análise de dados e determinação de requisitos utilizados no domínio da aplicação de negócio para os conceitos COSMIC.

e/ou hardware com o qual o mesmo interaja. Este Modelo de Contexto de Software apresenta os princípios e conceitos necessários a esta definição.

PRINCÍPIOS – O Modelo de Contexto de Software COSMIC
<p>a) O Software é delimitado pelo hardware</p> <p>b) O software é normalmente estruturado em camadas</p> <p>c) Uma camada pode conter um ou mais 'pares' de pedaços de software distintos e qualquer pedaço de software pode ser composto por componentes pares distintos</p> <p>d) Qualquer pedaço de software a ser medido deverá ser definido por seu escopo de medição, o qual deve estar integralmente contido em uma única camada</p> <p>e) O escopo de um pedaço de software a ser medido depende do propósito da medição</p> <p>f) Os usuários funcionais de um pedaço de software devem ser identificados a partir dos requisitos funcionais do usuário do pedaço de software a ser medido, como fontes e/ou destinos pretendidos para os dados</p> <p>g) Um pedaço de software interage com os seus usuários funcionais por meio de movimentações de dados através de uma fronteira, e o pedaço de software pode mover dados de e para o armazenamento persistente dentro da fronteira</p> <p>h) Os RFU do software podem ser expressos em diferentes níveis de granularidade</p> <p>i) O nível de granularidade no qual as medições devem ser normalmente efetuadas é o dos processos funcionais (ver seção 2.2.3)</p> <p>j) Se não for possível medir no nível de granularidade dos processos funcionais, nesse caso os RFU do software devem ser medidos através de uma abordagem de aproximação e escalonados para o nível de granularidade dos processos funcionais⁸</p>

Estes princípios e conceitos serão agora detalhados e ilustrados com alguns exemplos simples.

Para isto, precisamos distinguir duas visões de um sistema computacional, isto é, do contexto do pedaço de software a ser medido, a saber

- A visão 'física', que mostra como, na prática, o software é normalmente estruturado em uma hierarquia de camadas, cada uma com sua especialidade própria. Esta visão mostra que, na realidade, toda comunicação com qualquer pedaço de software acontece via dispositivos de hardware e (talvez) outras camadas intermediárias de software.
- A visão 'lógica', a qual é uma abstração da visão física utilizada para fins de medição do tamanho funcional. Esta visão mostra que os usuários funcionais de um pedaço de software a ser medido (fontes e/ou destinos pretendidos para os dados) interagem com o software através de uma fronteira e que o software movimenta dados de e para o armazenamento persistente. Neste nível de abstração todo hardware e software que permitem tais interações são ignorados.

Embora apenas a segunda visão seja necessária para fins de medição de tamanho funcional, é útil explicar ambas as visões, uma vez que os medidores devem ser capazes de distinguir as duas visões. Além disso, o método COSMIC usa termos como 'camada' e 'componente par' de uma

⁸ O assunto escalonamento entre diferentes níveis de granularidade é tratado no documento do método COSMIC 'Tópicos Avançados e Relacionados'.

maneira bem específica que precisa ser compreendida. (Estes termos são utilizados com muitos significados diferentes na indústria de software.)

A fig. 2.2.2.1 apresenta a visão física para um pedaço de software típico de uma aplicação de negócios no contexto de arquitetura de camada de software, compreendendo o sistema operacional, 'drivers' de dispositivos, etc, e hardware. A fig. 2.2.2.2 apresenta a mesma visão física para um exemplo simples de um software 'real-time' embarcado.

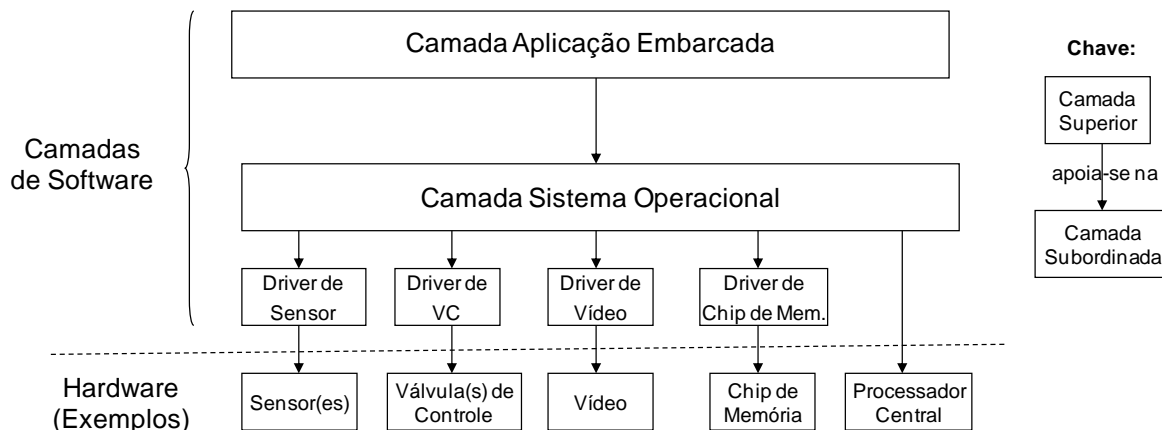


Figura 2.2.2.1 – Arquitetura de software típica em camadas, para um Sistema de Negócios/MIS

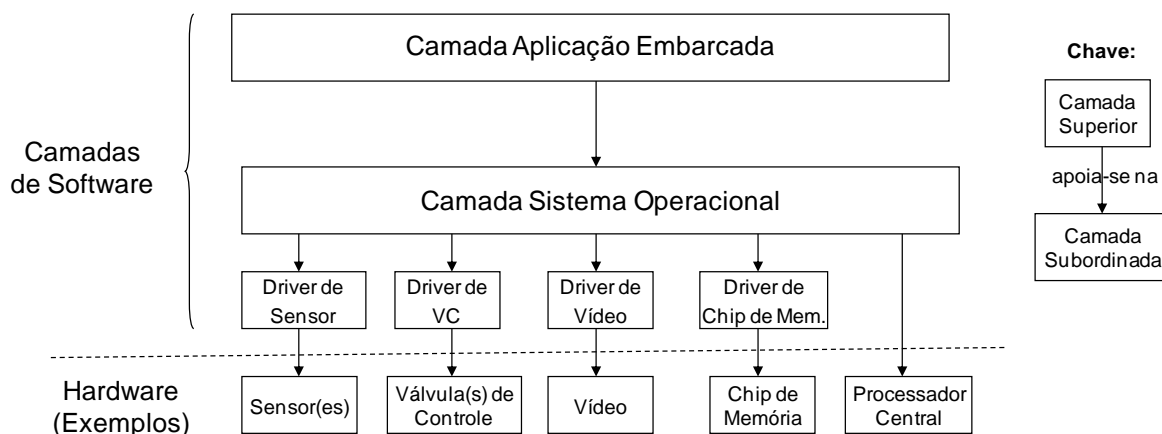


Figura 2.2.2.2 – Arquitetura típica em camadas para um sistema de software 'real-time' embarcado

Princípio (a)

O software usado por seres humanos é delimitado por dispositivos de hardware de entrada/saída tais como: mouse, teclado, impressora ou monitor; o software 'real-time' geralmente é delimitado por dispositivos tais como sensores ou relés. O software também é delimitado por hardware de armazenamento de dados tal como disco rígido, ou outros tipos de memória utilizados para guardar dados.

Princípio (b)

Se o pedaço de software a ser medido é parte de uma arquitetura desenhada em camadas, deve ser fácil decidir à qual camada o pedaço pertence. Entretanto, se um ambiente de software cresceu e evoluiu ao longo do tempo, as camadas (se houver) não poderão ser claramente identificáveis. Para estas situações, o método COSMIC inclui algumas regras para identificar camadas.

Princípio (c)

Por exemplo, os principais componentes distintos de uma aplicação de negócios (por exemplo, uma arquitetura 'três-camadas' com um componente de 'interface com o usuário / apresentação', um componente de 'regras de negócios' e um componente 'serviços de dados') são componentes pares. Tais componentes principais podem ser medidos separadamente e o método COSMIC fornece as regras para tal dimensionamento. Esta habilidade de medir separadamente os tamanhos dos principais componentes de um pedaço de software quando os mesmos executam em plataformas técnicas diferentes é muito importante, na prática, para os propósitos de medição de performance e estimativas.

Qualquer pedaço de software em qualquer camada pode ser decomposto em seus componentes em vários níveis (por exemplo, descer a módulos individuais ou objetos de classes) e o método COSMIC pode ser utilizado para medir o respectivo tamanho funcional em qualquer um dos níveis. Abaixo do nível dos componentes pares principais em qualquer camada, os medidores terão que definir seus padrões locais de níveis de decomposição (em conjunto com o arquiteto de software ou sistemas), se desejarem manter a compatibilidade entre medições provenientes de diferentes fontes.

Princípio (d)

Este princípio requer que o escopo de qualquer pedaço de software medido deva estar inteiramente contido em uma única camada. A razão disto é que cada camada tem uma função especializada e pode ser desenvolvida com tecnologia diferente daquela utilizada nas outras camadas. Dessa forma, pode ou não fazer sentido medir o tamanho de alguns pedaços de software residentes em duas ou mais camadas e, depois, somar tais tamanhos como se o resultado representasse o tamanho de uma simples entidade. A medida de tamanho resultante poderia, assim como a soma dos tamanhos de maçãs e laranjas, ser muito difícil de interpretar e/ou comparar com outras medidas de tamanho funcional. Para as regras de agregação do tamanho de pedaços de software em diferentes camadas, ver a seção "Agregando os resultados da medição", no Manual de Medição.

Princípio (e)

Como exemplo, supondo que os pedaços da aplicação de software nas Figs. 2.2.2.1 e 2.2.2.2 sejam cada um pedaços distintos de software, desenvolvidos por equipes de projeto distintas, então em cada caso, para a maioria dos propósitos de medição mais comuns, faria sentido definir o escopo da medição como a 'aplicação como um todo'. Entretanto, se uma aplicação for desenvolvida como três componentes pares principais (conforme mencionado no princípio (c) acima) cada um utilizando tecnologias diferentes e/ou desenvolvidos por diferentes equipes de projeto, então para o propósito de estimar o esforço do projeto fará sentido definir três escopos de medição separados, um para cada componente par. O tamanho medido de cada um dos três componentes servirá como entrada para uma fórmula de estimativa capaz de explicar diferentes tecnologias e/ou características das equipes de projeto de cada componente.

Princípio (f)

Para ilustrar este e o próximo princípio (g), precisamos examinar a visão lógica do software a ser medido. As Figuras 2.2.2.3 e 2.2.2.4 ilustram esta visão lógica, mostrando a interação dos 'usuários funcionais', respectivamente, com um pedaço de software de aplicação de negócio e com um pedaço de sistema 'real-time' embarcado.

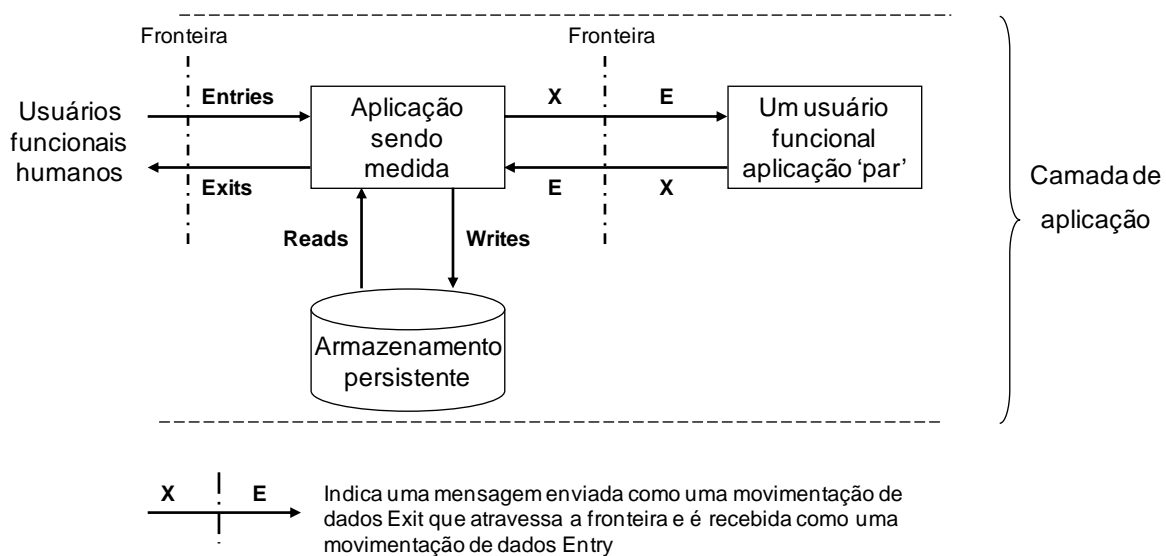


Figure 2.2.2.3 - Uma aplicação de negócio com seres humanos e outra aplicação 'par' como seus usuários funcionais (visão lógica)

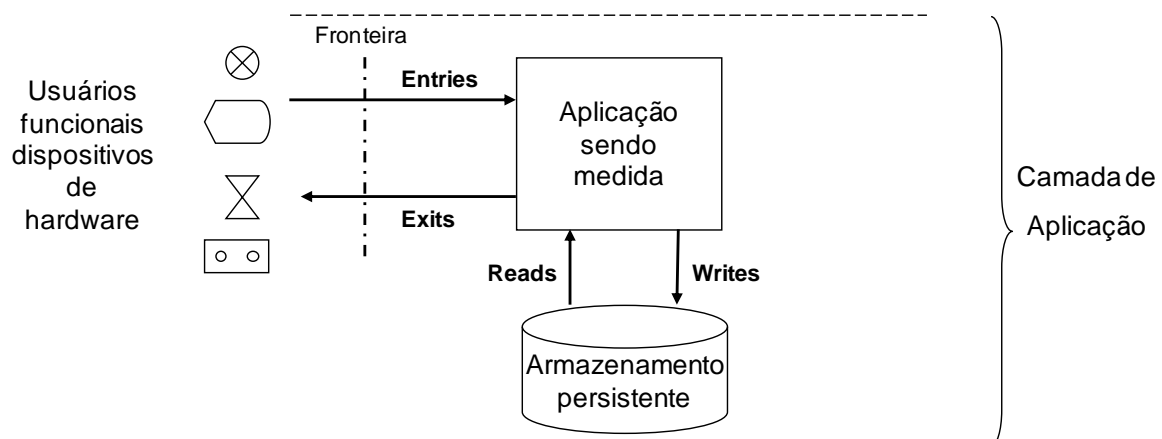


Figura 2.2.2.4 - Uma aplicação de software 'real-time' embarcado com vários dispositivos de hardware como seus usuários funcionais (visão lógica)

Considere, como exemplo, um pedaço de software de aplicação de negócio a ser medido. A Fig. 2.2.2.1 mostra que poderiam ser considerados 'usuários' da aplicação o sistema operacional, qualquer um dos dispositivos de hardware (por exemplo, o teclado, a impressora, etc.), e os usuários humanos, uma vez que todos eles 'interagem' direta ou indiretamente com a aplicação. Porém nem todos estes (tipos de) usuários serão especificados nos RFU como remetentes e destinatários para os dados, de/para o software. O sistema operacional e os dispositivos de hardware são 'facilitadores' destas trocas de dados, ao invés de remetentes ou destinatários.

Para um pedaço de software de uma aplicação de negócio, em geral os RFU descrevem somente a funcionalidade requerida do ponto de vista dos usuários humanos da aplicação e, talvez, outras aplicações pares que enviem ou recebam dados de/para a aplicação. Tais seres humanos e

aplicações pares serão, portanto, os usuários funcionais da aplicação, conforme a visão lógica da Fig. 2.2.2.3.

Devido à separação rígida da funcionalidade em camadas, conforme apresentada na Fig. 2.2.2.1, os RFU da aplicação de negócio podem ignorar quaisquer camadas de software e dispositivos de hardware, tais como o sistema operacional ou telas que possibilitem a interação dos usuários funcionais com a aplicação. Normalmente não devem existir dúvidas sobre a identidade dos usuários funcionais.

Para o exemplo do software embarcado de tempo real, seus RFU geralmente descreverão a funcionalidade requerida do ponto de vista dos dispositivos de hardware (sensores, válvulas, etc.) que o software deve suportar. Estes dispositivos, entretanto, serão os usuários 'funcionais' do software embarcado, conforme mostrado na Fig. 2.2.2.4. Porém, como veremos no Manual de Medição, embora incomum, os RFU de alguns tipos de software são escritos para situações onde há mais de um tipo de usuário funcional, dando assim origem a diferentes funcionalidades e tamanho funcional.⁹

Princípio (g)

As Figs. 2.2.2.3 e 2.2.2.4 demonstram também que os usuários funcionais interagem com o software através da fronteira via dois tipos de movimentação de dados (**Entries** e **Exits**). O software também troca dados com o dispositivo de armazenamento persistente via dois tipos de movimentação de dados (**Reads** e **Writes**). Estas movimentações de dados são definidas adiante na seção 2.2.3.

A 'fronteira' é definida como 'uma interface conceitual entre o software e seus usuários funcionais'. Esta fronteira não deve ser confundida com qualquer linha desenhada em um diagrama para delimitar o escopo de um pedaço de software a ser medido, ou ao redor de uma camada de software.

A fronteira permite fazer uma distinção clara entre qualquer coisa que seja parte do pedaço de software sendo medido (isto é, o que está dentro da fronteira do software) e qualquer coisa que seja parte do ambiente dos usuários funcionais (isto é, o que está fora da fronteira do software). O dispositivo de armazenamento não é considerado como um usuário do software e portanto está dentro da fronteira do software.

A Fig. 2.2.2.5 apresenta a visão lógica de alguma aplicação de negócio que foi desenvolvida em três componentes pares principais, conforme descrito nos princípios (c) e (e) acima. Supondo que os componentes pares tenham sido desenvolvidos utilizando tecnologias diferentes, é provável que a finalidade da medição determinasse a definição de um escopo de medição separado para cada componente par.

⁹ Uma exceção na qual o sistema operacional pode ser um usuário funcional de uma aplicação acontece quando os RFU da aplicação solicitam que o sistema operacional forneça, por exemplo, um 'tique de relógio' para iniciar um processo funcional na aplicação.

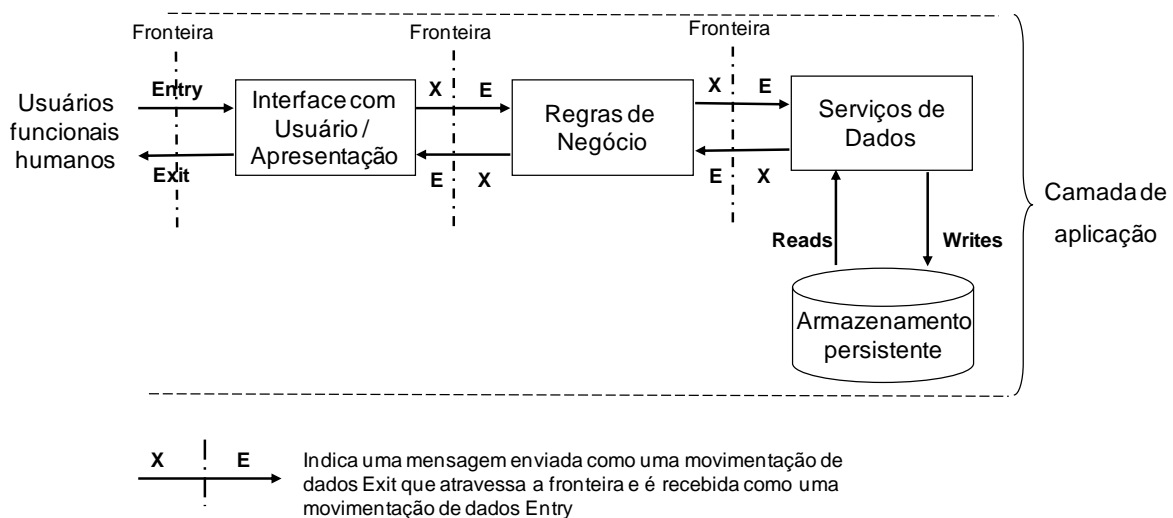


Figura 2.2.2.5 – Uma aplicação de negócio onde seus principais componentes’ devem ser medidos separadamente (visão lógica)

Na visão lógica da Fig. 2.2.2.5, vemos que o componente ‘apresentação/interface do usuário’ tem como usuários funcionais seres humanos e o componente ‘regras de negócio’, cada um interagindo com o componente via Entries e Exits através da fronteira. Esta Figura também mostra que somente o componente ‘serviço de dados’ interage com o armazenamento persistente e que o usuário funcional é o componente ‘regras de negócio’. Com tais visões lógicas, os RFU de cada componente poderão ser medidos separadamente.

Princípios (h) e (i)

O RFU do software pode ser expresso em diferentes ‘níveis de granularidade’. (Perceba que o conceito ‘nível de granularidade’ está relacionado com o nível de detalhamento da especificação do pedaço de software. Deve ser diferenciado do ‘nível de decomposição’, o qual está relacionado com a quebra do software em suas partes componentes). O nível de granularidade no qual a medição deve ser realizada é o do processo funcional (ver seção 2.2.3). Ao iniciar um novo desenvolvimento de software, o processo de determinação dos requisitos funcionais do usuário (‘RFU’) do software normalmente começa com a definição e concordância a respeito de uma declaração de requisitos de ‘alto nível’, os quais são refinados e trabalhados em maior detalhe. Os RFU de um pedaço de software de um dado escopo podem dessa forma existir em diferentes níveis de granularidade. Um exemplo típico de utilização da técnica de ‘análise funcional’ para determinar os RFU do pedaço de software poderia resultar na seguinte hierarquia de níveis de RFU.

Ao ser analisada em maior detalhe, uma ‘função principal nível 1’ demonstra ser constituída por um certo número de ‘funções nível 2’; cada uma destas funções, por sua vez, é composta por um ‘subfunções nível 3’, cada uma composta por ‘sub-sub-funções nível 4’, etc. Em algum ponto dessa hierarquia a análise revelará processos funcionais individuais (veja a seção 2.2.3 para mais informações; por enquanto precisamos pensar nesses processos apenas como partes de funcionalidades que podemos medir.)

Como a análise ‘amplia’ mais e mais os detalhes, o tamanho funcional medido parece aumentar porque mais detalhes são levados em consideração. (Nota: este fenômeno é diferente do ‘scope creep’, no qual o tamanho aumenta porque o escopo do software aumenta.)

Consequentemente, a fim de poder comparar as medições de forma sensata a partir de fontes diferentes, ou usar medições em algum outro processo, todas as medições devem ser feitas (ou escalonadas) em um nível padrão de granularidade, o qual denominamos ‘nível de granularidade de processo funcional’. Na maioria dos casos, quando o objetivo é medir o tamanho funcional de um

software totalmente especificado ou de um pedaço de software, o nível de granularidade de processo funcional ao qual se deve medir é óbvio.

Todo o mundo conhece a idéia de medir distâncias em mapas utilizando diferentes escalas, por exemplo, quando 1 km é representado em um mapa por 1 cm ou 1 mm, bem como a conversão de uma escala para outra. Porém, ao medir o tamanho funcional de um pedaço de software usando COSMIC, temos apenas o padrão 'nível de granularidade de processo funcional' e apenas uma unidade de medida. Portanto, se precisarmos comparar medições feitas em diferentes níveis de granularidade, teremos de criar os nossos próprios fatores de escala locais para converter os tamanhos para as unidades correspondentes ao nível de granularidade padrão de processo funcional. Estes conceitos serão mais explorados na seção sobre o nível de granularidade padrão no Manual de Medição.

Princípio (j)

O problema de se precisar medir um pedaço de software em um nível de granularidade mais alto que o de processo funcional geralmente aparece apenas nas primeiras fases do desenvolvimento de um novo software, enquanto os requisitos ainda estão em evolução. Em tais circunstâncias, quando não é possível medir no nível de granularidade dos processos funcionais, uma abordagem de medição por aproximação deve ser utilizada e o resultado escalonado para o nível de granularidade dos processos funcionais (ver o capítulo sobre estimativas em fases iniciais do projeto, no documento 'Método COSMIC v3.0: Tópicos Avançados e Relacionados').

Em resumo, o Modelo de Contexto de Software do método de medição COSMIC fornece um conjunto de conceitos e princípios, a saber, camadas de software e componentes pares, o escopo de um pedaço de software a ser medido, seus usuários funcionais, movimentações de dados e fronteira para auxiliar a medir os requisitos funcionais do usuário, os quais podem ser representados em diferentes níveis de granularidade. Durante o processo de Estratégia de Medição (como descrito na seção 2.2.4) aplicamos estes conceitos e princípios nos RFU do software a ser medido, para responder questões tais como 'qual medição é requerida?' ou 'como interpretar esta medição?'

2.2.3 O Modelo Genérico de Software COSMIC

Uma vez interpretados os RFU do software a ser medido em termos do Modelo de Contexto de Software, aplicamos o Modelo Genérico de Software aos RFU, para identificar os componentes da funcionalidade que serão medidos. Este Modelo Genérico de Software assume que os seguintes princípios gerais são verdadeiros para qualquer software que possa ser medido com o método. Ver o Glossário para a obter a definição de todos os termos¹⁰.

PRINCÍPIOS – O Modelo Genérico de Software COSMIC

- a) O software recebe dados de **entrada** dos seus usuários funcionais, gerando **saídas** e/ou outros resultados para os usuários funcionais
- b) Os requisitos dos usuários funcionais de um pedaço de software a ser medido podem ser mapeados para processos funcionais distintos
- c) Cada processo funcional consiste de **subprocessos**
- d) Um subprocesso pode ser uma movimentação de dados ou uma **manipulação de dados**
- e) Cada processo funcional é disparado por um movimento de dados do tipo **Entry** proveniente de um usuário funcional, o qual informa ao processo funcional que o usuário funcional identificou um **evento**

¹⁰ Como observado no glossário, qualquer método de medição de tamanho funcional busca identificar 'tipos' e não 'ocorrências' de dados ou funções. No texto abaixo, deixaremos de utilizar o sufixo '(tipo)' quando mencionados os conceitos básicos do COSMIC, a menos que seja essencial distinguir 'tipos' e não 'ocorrências' de dados ou funções.

- f) Uma movimentação de dados movimenta um único **grupo de dados**
- g) Um grupo de dados consiste de um único conjunto de **atributos de dados** que descreve um único objeto de interesse
- h) Há quatro tipos de movimentação de dados. Uma **Entry** movimenta um grupo de dados para dentro do software, a partir de um usuário funcional. Uma **Exit** movimenta um grupo de dados para fora do software, em direção a um usuário funcional. Um **Write** movimenta um grupo de dados do software para o armazenamento persistente. Um **Read** movimenta um grupo de dados do armazenamento persistente para o software
- i) Um processo funcional deve incluir pelo menos uma movimentação de dados Entry e uma movimentação de dados Write ou Exit, ou seja, deve incluir no mínimo duas movimentações de dados
- j) Como uma aproximação para fins de medição, os subprocessos de manipulação de dados não são medidos separadamente; assume-se que a funcionalidade de qualquer manipulação de dados será considerada na movimentação de dados com a qual a mesma esteja associada

Estes princípios surgem a partir de um entendimento comum de software como segue.

Princípios (a) a (e)

A tarefa do software é responder a eventos que ocorrem do lado de fora da sua fronteira, isto é, no mundo dos usuários funcionais. Um usuário funcional notifica o software da ocorrência de um evento e pode enviar dados relacionados ao evento. O software deve fazer alguma coisa útil para o usuário funcional em resposta a aquele evento. Chamamos esta 'alguma coisa útil' de 'processo funcional'. Todos os RFU de software portanto podem ser expressos como uma lista de tipos de eventos e processos funcionais correspondentes que executam a resposta do software para cada evento.

Os princípios (c) e (d) dizem que os processos funcionais podem ser compostos de dois tipos de subprocessos, denominados movimentação de dados e manipulação de dados. O software somente pode movimentar e/ou manipular dados.

A Fig. 2.2.3.1, abaixo, apresenta os princípios (b) a (d) do Modelo Genérico de Software.

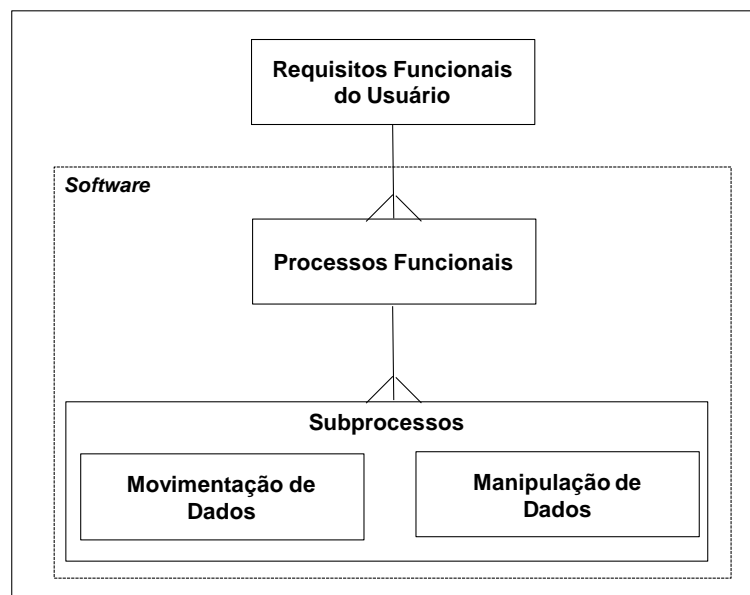


Figura 2.2.3.1 – A estrutura dos Requisitos Funcionais do Usuário

Princípios f) e g)

Cada movimentação de dados transporta apenas um grupo de dados, isto é, dados sobre um simples objeto de interesse, ou seja, algo 'de interesse' para o usuário funcional. Por exemplo, no domínio de software de aplicação de negócio, um processo funcional relativamente simples para registrar um pedido tipicamente envolve a seguinte movimentação de dados (objetos de interesse são apresentados entre aspas):

- Duas Entries de grupos de dados relacionados ao 'pedido' e ao 'item do pedido' (assumindo um pedido com vários itens). A primeira destas Entries de dados descrevendo o objeto de interesse 'pedido' é uma das que dispara (ou inicia) o processo funcional
- Dois Reads de grupos de dados relacionados a 'cliente' e 'produto', para validar que o cliente está autorizado a fazer o pedido e que o produto solicitado é válido e está disponível
- Writes de grupos de dados relacionados ao 'pedido' e ao 'item do pedido', para movimentar os dados informados para o armazenamento persistente
- Uma ou mais Exits de grupos de dados contendo, por exemplo, uma 'confirmação do pedido', uma mensagem incluindo o valor total do pedido, uma orientação ao almoxarifado para separar cada 'item do pedido', etc.

Todos estes objetos de interesse são coisas reais ou conceituais no mundo real dos usuários funcionais (humanos, neste caso), sobre as quais o pedaço de software é solicitado a processar dados. Devem ser identificados e distinguidos de forma a identificar as movimentações de dados.

Aplicam-se exatamente os mesmos princípios na medição de software de tempo real ou embarcado, embora o 'usuário funcional' e o 'objeto de interesse' sejam frequentemente indistinguíveis na prática. Por exemplo, suponha que um processo funcional precisa obter a temperatura atual de um sensor, supondo que o sensor é equipado de tal forma que pode comunicar-se diretamente com o software; o sensor é desta forma um usuário funcional do pedaço de software. Neste caso, o sensor envia uma movimentação de dados Entry que provavelmente tem dois atributos de dados (a ID do sensor e a temperatura). Estes dois atributos transmitem informações sobre o sensor (como objeto de interesse) – ou ainda, poderia ser igualmente argumentado que o objeto de interesse é a "coisa" cuja temperatura é medida pelo sensor.

Suponha um processo funcional em tempo real bem simples, para medir a temperatura através do sensor e controlá-lo em relação a uma temperatura alvo. Este processo funcional é acionado a intervalos regulares por um sinal recebido de um relógio, e poderia consistir das seguintes movimentações de dados.

- Uma Entry recebida do relógio que dispara (inicia) o processo funcional
- Uma Entry recebida do sensor de temperatura contendo o ID do sensor e a temperatura atual
- Um Read no armazenamento persistente para obter a temperatura alvo (supondo que a temperatura alvo pode ser definida e alterada por outro processo funcional)
- Uma Exit para o aquecedor contendo um sinal para ligar ou desligar, se o estado do aquecedor necessita ser alterado

Observe que neste exemplo bem simples todos menos um dos grupos de dados é composto por somente um atributo.

Princípio (h)

Este princípio afirma que os quatro tipos de movimentação de dados são distinguidos por sua origem e destino. As movimentações de dados cruzam a fronteira entre o software sendo medido e seus usuários funcionais (Entries e Exits), ou movimentam-se entre o software e o armazenamento persistente (Reads e Writes). Tais relacionamentos são mostrados na fig. 2.2.3.2 abaixo.

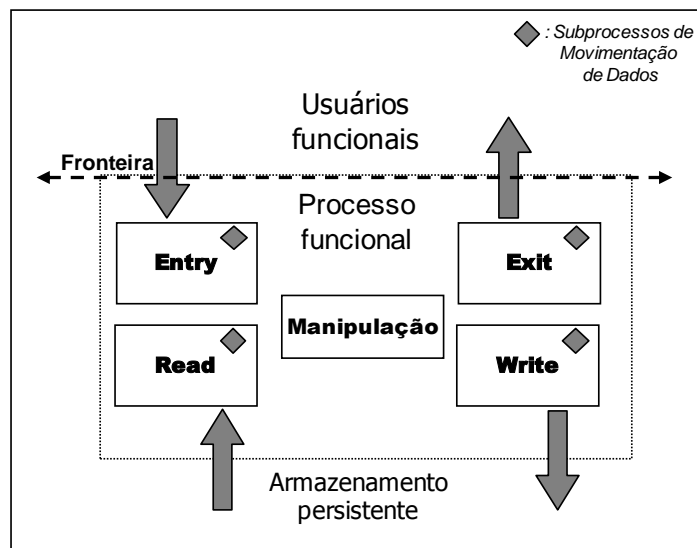


Figura 2.2.3.2 – Os componentes de um processo funcional e alguns de seus relacionamentos

Princípio (i)

Este princípio afirma que o processo funcional deve ter pelo menos duas movimentações de dados. Isto decorre dos princípios anteriores. Um processo funcional que apenas recebesse uma movimentação de dados e nada fizesse seria praticamente inútil. Portanto, todos os processos funcionais devem ter pelo menos uma movimentação de dados informando sobre a ocorrência de um evento (uma Entry), e pelo menos uma outra movimentação de dados como resposta (ou resultado útil), para um usuário funcional (uma Exit) ou para o armazenamento persistente (um Write).

Princípio (j)

Para fins de medição, e dado o domínio de software para o qual o método foi desenhado, o método COSMIC assume uma simplificação do Modelo Genérico de Software.

Como uma primeira aproximação nesta versão do método de medição, subprocessos do tipo manipulação de dados, ilustrados na figura 2.2.3.2, não são reconhecidos separadamente, e sim considerados associados com ou parte de um subprocesso de movimentação de dados. (De agora em diante, para abreviar, a expressão ‘movimentação de dados’ será utilizada, ao invés de ‘subprocesso de movimentação de dados’). A razão desta aproximação que os conceitos e definições necessários, requeridos para medir a manipulação de dados, são ainda tema de muito debate e discussão entre os especialistas em engenharia de software. O tipo específico de funcionalidade de manipulação de dados que deve ser incluído dentro de cada tipo de movimentação de dados é descrito na subseção de manipulação de dados associada à movimentação de dados, no Manual de Medição.

Dada esta aproximação, podemos ver por que o método padrão COSMIC é adequado para medir tipos de software ‘ricos em movimentações’, tais como a maioria das aplicações de negócio e muitos softwares de tempo real, porém inadequado para medir software ‘rico em manipulação’ (ou ‘rico em algoritmos’). O Manual de Medição salienta também a necessidade de cuidado ao medir pedaços de software muito pequenos, especialmente pequenas mudanças de software, onde a hipótese do princípio (j) pode deixar de ser válida.

O Manual de Medição também fornece um mecanismo para definir uma extensão local do método COSMIC que permita a uma organização levar em conta funcionalidades de manipulação de dados, se assim desejar. Tais casos devem ser reportados conforme as convenções especiais definidas.

Utilizando os conceitos, definições, princípios e regras do método de medição COSMIC, os requisitos funcionais do usuário extraídos dos artefatos do pedaço de software podem ser mapeados para o Modelo Genérico de Software, criando assim uma instância do mesmo. Esta instância do modelo conterá todos os elementos necessários para a medição do tamanho funcional, ao mesmo tempo ocultando informações não relevantes a essa atividade.

As regras e os processos de medição serão então aplicados a esta instância do Modelo Genérico de Software, com o objetivo de produzir o valor de uma quantidade representando o tamanho funcional do pedaço de software – veja 2.3.3 abaixo para as regras de medição.

2.3 Visão Geral do Processo de Medição COSMIC

São necessárias três fases distintas e relacionadas para medir o tamanho funcional de um pedaço de software:

1. Estabelecer a estratégia de medição utilizando os princípios do Modelo de Contexto de Software
2. Mapeamento dos artefatos do software a ser medido para o Modelo Genérico de Software
3. Medição dos elementos específicos deste modelo.

2.3.1 A Fase Estratégia de Medição

Antes de começar uma medição o medidor deve, de comum acordo com os patrocinadores da medição, documentar: (a) o propósito da medição, (b) o escopo de cada pedaço de software a ser medido, (c) os usuários funcionais e a fronteira de cada pedaço de software, e (d) o nível de granularidade que a medição requer. Estabelecer uma declaração clara de propósito da medição (a) é criticamente importante porque o propósito determina os outros três parâmetros (b), (c), e (d). A determinação destes outros três parâmetros será com frequência um processo iterativo.

(a) O propósito da medição

O propósito da medição estabelece por quê a medição está sendo realizada e para quê os resultados serão utilizados. Isto por sua vez ajudará a determinar não apenas os outros três parâmetros da Estratégia de Medição, mas também, por exemplo, a exatidão requerida pela medição. (Como será visto no capítulo de medição inicial no documento 'Tópicos Avançados e Relacionados', é possível estimar o tamanho funcional com uma aproximação do método básico COSMIC. Esta variação do método pode ser aplicada na fase inicial do ciclo de vida do projeto, antes de todos os requisitos serem descritos em detalhes suficientes para medir o tamanho seguindo exatamente as regras do método.

(b) O escopo do software a ser medido

O escopo total do software a ser medido vem do propósito. O escopo total estabelece qual funcionalidade do software será incluída na medição (e o que será excluído). Dependendo do propósito, o escopo total talvez precise ser subdividido em um número de pedaços de software separados, cada um medido separadamente com seu escopo próprio.

Tal subdivisão do escopo total será necessária primeiro, se o escopo total incluir software de mais de uma camada, porque cada pedaço de software a ser medido deve residir completamente em uma única camada. Segundo, uma subdivisão pode ser necessária, por exemplo, se o propósito for relacionado a estimativa, e o software total a ser medido compreenda componentes separados que serão desenvolvidos com diferentes técnicas e/ou tecnologias e/ou executados em diferentes plataformas técnicas e/ou desenvolvidos por equipes diferentes. Cada componente, então, deve ter seu escopo próprio de medição.

(c) Os usuários funcionais e a fronteira de cada pedaço de software a ser medido

Os usuários funcionais de cada pedaço de software podem ser identificados examinando o fluxo de dados de entrada e saída do software como declarado ou implícito nos requisitos funcionais do

usuário e considerando o propósito da medição. Os usuários funcionais serão os remetentes ou destinatários dos dados.

Na maioria das situações, é óbvia a identificação dos usuários funcionais a partir do propósito da medição e dos RFU. Excepcionalmente, os usuários funcionais podem variar dependendo do propósito da medição. Um exemplo será apresentado na subseção de usuários funcionais do Manual de Medição, o qual mostra um caso onde poderá haver uma escolha.

Quando os usuários funcionais são conhecidos, a fronteira – interface conceitual entre os usuários funcionais e o pedaço de software sendo medido – pode ser facilmente estabelecida.

(d) O nível de granularidade das medições

A medição dos RFU de um pedaço de software geralmente é feita no nível de granularidade no qual os processos funcionais são identificados e suas quebras em movimentações de dados são definidas.

Quando o propósito é medir os RFU de software de algum pedaço de software totalmente especificado ou existente, o nível de granularidade fica normalmente evidente quando os processos funcionais são identificados.

Por outro lado, nos estágios iniciais do desenvolvimento de software, onde o propósito é medir os RFU de algum(ns) pedaço(s) de software conforme este(s) evolui(em), os RFU devem ser medidos antes do ponto em que algum(ns) ou todos os processos funcionais individuais foram identificados e suas movimentações de dados definidas. Para complicar mais, os RFU dos diferentes pedaços de software podem existir em diferentes níveis de granularidade no momento em que a medição é necessária. Nestas circunstâncias, é necessário definir alguma ‘unidade funcional’ local que possa ser identificada e contada nos artefatos de software disponíveis, assim como será necessário um método para escalonar a partir do nível de granularidade que o tamanho foi medido através da contagem das unidades funcionais para o nível de granularidade no qual os processos funcionais podem ser identificados e medidos (isto é para Pontos de Função COSMIC). Tais métodos de escalonamento são discutidos na seção sobre a identificação de um nível padrão de granularidade, no Manual de Medição e no documento ‘Tópicos Avançados e Relacionados’.

Uma vez completados os passos de (a) até (d), alguma iteração ainda pode ser necessária. Por exemplo, conforme alguns requisitos são detalhados, isto pode levar à necessidade de refinar o escopo do(s) pedaço(s) de software a ser(em) medido(s).

Uma descrição completa com as definições e mais exemplos de propósito, usuários funcionais e nível de granularidade é apresentada no capítulo referente ‘a Estratégia de Medição do Manual de Medição.

2.3.2 A Fase de Mapeamento

A Fase de Mapeamento tem como entrada os requisitos funcionais do usuário de cada pedaço de software a ser medido, extraídos de seus artefatos (como encontrados ou documentados dentro da organização, ou inferidos a partir do software físico existente), considerando o Modelo de Contexto de Software. A saída da Fase de Mapeamento é uma instância do Modelo Genérico de Software.

Os passos para instanciar o Modelo Genérico de Software na Fase de Mapeamento são:

- identificar os eventos no mundo dos usuários funcionais aos quais o software deve responder e daí identificar os processos funcionais
- identificar as movimentações de dados (Entries, Exits, Reads e Writes) de cada processo funcional, os quais por sua vez dependem da identificação dos grupos de dados movimentados.

Uma descrição completa com as definições e exemplos dos conceitos e passos da Fase de Mapeamento é apresentada no capítulo correspondente do Manual de Medição.

2.3.3 A Fase de Medição

A Fase de Medição tem como entrada uma instância do Modelo Genérico de Software e, utilizando um conjunto de regras e processos definidos, esta fase produz um valor numérico cuja magnitude é diretamente proporcional ao tamanho funcional do modelo, com base no seguinte princípio:

PRINCÍPIO – O princípio de medição COSMIC
--

O tamanho funcional de um pedaço de software é diretamente proporcional ao número de movimentações de dados.
--

As características do conjunto de regras e processos que regulam a produção deste valor numérico são as seguintes:

Característica 1 – Unidade de medida

O padrão de medição, nominalmente 1 PFC (Ponto de Função COSMIC) é definido, por convenção, como o tamanho de uma movimentação de dados.

Característica 2 – Somatório dos tamanhos dentro de um dado escopo de medição

O tamanho funcional de um processo funcional é definido como a soma aritmética do número de suas movimentações de dados. Por consequência, o tamanho funcional de qualquer pedaço de software com um dado escopo de medição¹¹ em qualquer camada no modelo de software é a soma aritmética dos tamanhos funcionais dos processos funcionais desse pedaço de software.

Característica 3 – Tamanho de mudança(s) de um pedaço de software

O tamanho funcional de qualquer mudança solicitada em um pedaço de software é por convenção a soma aritmética do número de movimentações de dados que devem ser incluídas, alteradas ou excluídas como consequência da(s) mudança(s) solicitada(s).

Característica 4 – O tamanho mínimo e máximo de um processo funcional

De acordo com estas características, como já estabelecido no princípio (g) do Modelo Genérico de Software, o tamanho funcional mínimo de um processo funcional simples é 2 PFC, porque o menor processo funcional deve ter pelo menos uma Entry (como entrada), e uma Exit (como saída) ou um Write (como um resultado útil alternativo).

Como uma mudança pode afetar somente uma movimentação de dados, segue que o tamanho mínimo de uma mudança em um processo funcional é 1 PFC.

Além disso, segundo estas características, não há limite máximo para o tamanho funcional de qualquer processo funcional e consequentemente não há limite para o tamanho funcional de um pedaço de software.

Princípios adicionais e regras e processos detalhados da Fase de Medição para determinar o tamanho funcional dos RFU de um pedaço de software expresso no Modelo Genérico de Software são apresentados nos capítulos relacionados do Manual de Medição e resumidos nos seus apêndices B e C.

¹¹ Para as regras de agregação dos tamanhos dos pedaços de software em diferentes escopos de medição, veja o capítulo do Manual de Medição sobre a fase de medição.

APÊNDICE A - PROCEDIMENTO COSMIC PARA SOLICITAÇÃO DE MUDANÇA E COMENTÁRIOS

O Comitê de Práticas de Medição COSMIC ('COSMIC Measurement Practices Committee – MPC') está bastante disposto a receber 'feedback', comentários e, se necessário, solicitações de mudança para o método COSMIC. Este apêndice estabelece a forma de comunicação com o COSMIC MPC.

Todas as comunicações com o COSMIC MPC devem ser enviadas via e-mail para o seguinte endereço:

mpc-chair@cosmicon.com

Comentários e 'feedback' gerais e informais

Comentários informais e/ou 'feedback' sobre a documentação COSMIC, tais como quaisquer dificuldades no entendimento ou na aplicação do método COSMIC, sugestões para melhoria geral, etc., devem ser enviadas via e-mail para o endereço acima. As mensagens serão registradas e normalmente reconhecidas dentro de duas semanas do recebimento. O MPC não garante que iniciará qualquer ação em função desses comentários gerais.

Solicitações formais de mudança

Quando o leitor da documentação COSMIC acredita que há um erro no texto, necessidade de esclarecimento, ou que alguma parte do texto precisa ser melhorada, uma solicitação formal de mudança ('change request' – CR) pode ser submetida. As CRs formais serão registradas e reconhecidas dentro de duas semanas do respectivo recebimento. Cada CR receberá um número de série e circulará entre os membros do COSMIC MPC, um grupo mundial de especialistas no método COSMIC. Seu ciclo normal de revisão é de no mínimo um mês e pode levar mais se a CR for de difícil resolução. O resultado da revisão poderá indicar que a CR será aceita, rejeitada, ou 'suspensa aguardando discussão adicional' (neste último caso, por exemplo, se houver uma dependência em relação a outra CR), e o resultado será comunicado ao autor da submissão assim que possível.

Uma CR formal será aceita somente se vier com todas as informações relacionadas a seguir.

- Nome, posição e organização da pessoa que submeteu a CR
- Informações de contato da pessoa que submeteu a CR
- Data da submissão
- Declaração geral do propósito da CR (por exemplo: 'necessidade de melhorar o texto...')
- Texto que necessita de mudança, substituição ou exclusão (ou uma referência clara ao mesmo)
- Texto proposto adicional ou substituto
- Explicação completa sobre por que a mudança é necessária

Um formulário para submissão de CR está disponível no sítio www.cosmicon.com.

A decisão do COSMIC MPC sobre o resultado da revisão de uma CR e, se aceita, em qual versão da documentação COSMIC a CR será aplicada, é final.

Questões sobre a aplicação do método COSMIC

O COSMIC MPC lamenta ser incapaz de responder questões relativas ao uso e aplicação do método COSMIC. Existem organizações comerciais que podem oferecer treinamento e consultoria, ou

suporte de ferramentas para o método. Por gentileza, para maiores detalhes, consulte o sítio www.cosmicon.com.