

# **Processo de Avaliação Não-Funcional de Software (SNAP)**

## **Manual de Práticas de Avaliação**

---

**Versão 2.0**



**Página deixada em branco intencionalmente**

---

**International Function Point Users Group (IFPUG)**

---

**Manual de Práticas de Avaliação SNAP**

**Versão 2.0**

---

Chairperson, Non-Functional Sizing Standards Committee (NFSSC)

Talmon Ben-Cnaan

[NFSSC@ifpug.org](mailto:NFSSC@ifpug.org)



O Manual de Práticas de Avaliação SNAP, criado pelo International Functional Point Users Group (IFPUG), é liberado sob a licença [Creative Commons Attribution-Non Commercial-Share Alike 3.0 Unported](https://creativecommons.org/licenses/by-nc-sa/3.0/).

Baseado em um trabalho do [www.ifpug.org](http://www.ifpug.org)

Permissões além do escopo desta licença podem estar disponíveis em [www.ifpug.org](http://www.ifpug.org).

ISBN 978-0-9830330-8-0

---

---

**Página deixada em branco intencionalmente**

---

---

## Versão 2.0, Janeiro 2013

---

Esta edição é a segunda versão.

Alterações nas informações contidas são realizadas periodicamente.

---

## Equipe de Documentação

---

### Comitê NFSS

---

Christine Green, IFPUG Director, Board Liaison

Talmon Ben-Cnaan, Amdocs (Chairperson)

Kathy Lamoureaux Aetna (Vice-Chairperson)

Abinash Sahoo, Amdocs; Roopali Anand IBM; Steve Chizar, Retired, Naval Systems Supply Business Solutions; Charley Tichenor, DCA; Joanna Soles, CSC; Jalaja Venkat, iGATE

---

### Equipe do Manual de Práticas de Avaliação SNAP 2011-2012

---

Steve Chizar, Retired, Naval Systems Supply Business Solutions, Chair

Talmon Ben-Cnaan, Amdocs

Abinash Sahoo, Amdocs

Charlene Zhao, BMO

Charley Tichenor, DCA

Jalaja Venkat, iGATE

Joanna Soles, CSC

Kathy Lamoureaux, AETNA

Peter R Hill, ISBSG

Roopali Thapar, IBM

---

### Comitê de Performance de TI

---

Dan Bradley, Chair

Christine Green, HP Enterprise Services, SNAP Project Manager

Wendy Bloomfield, Great-West Life Assurance Co.

Talmon Ben-Cnaan, Amdocs

Joanna Soles, CSC

Loredana Frallicciardi, CSC, IFPUG Director, SNAP Sponsor (2008-2010)

Janet Russac, Software Measurement Expertise, Inc., IFPUG Director, SNAP Sponsor (2010-2011)

---

### Equipe do Manual de Práticas de Avaliação SNAP 2009-2011

---

Patty Boyce, NASCO; Sergio Brigido, HP Enterprise Services; Steve Chizar, Retired, Naval Systems Supply Business Solutions; Sheila Dennis, The David Consulting Group; David Garmus, The David Consulting Group; Asha Goyal, Nucleus Software Exports Ltd; Claudia Hazan, Employee of a Brazilian Gov IT Organization; Peter Hill, ISBSG; Lori Holmes, Q/P Management Group, Inc.; Ashwin Krishnamurthy, IBM India Pvt Ltd; Monica Lelli, IT Consultant; Luca Santillo, Agile Metrics; Jalaja Venkat, iGATE; Charles Wesolowski, QinetiQ North America Systems Engineering Group

---

### Equipe do Framework SNAP 2008

---

Albert Meyn, W&W Informatik GmbH; Ana Miccolis, Petrobras; Ashwin Krishnamurthy, IBM; Carolyn Melvin, Northrop Grumman; Charles Wesolowski, Lockheed Martin Space Systems Company; David Seaver, PRICE Systems LLC; David Garmus, The David Consulting Group; Graeme Prescott, Australian Government; Jalaja, iGATE Lori Holmes, Q/P Management Group, Inc.; Marco Motta, Vodafone; Mick Burn-Murdoch, Lalcrest Ltd; Mike Pearl, EDS; Muhammet Oeztuerk, W&W Informatik GmbH; Peter Grüner, G4Q IT Services; Pranabendu Bhattacharyya, Tata Consultancy Services

---

---

## **Agradecimentos Especiais**

Agradecimentos especiais são oferecidos aos membros da equipe que trabalharam nos dois projetos de definição do Framework do SNAP, bem como na definição e criação do Manual de Práticas de Avaliação (APM), incluindo:

- As duas equipes;
- Todos os indivíduos e organizações participantes na revisão, teste beta e feedback;
- Bonnie Brown e Luc Vangrunderbeeck, por seu suporte com o layout do APM, com a ferramenta de teste beta e com a análise dos dados do teste beta.

---

### **Revisores do SNAP – Obrigado!**

Ajith PN, Alain Gontard, Albert Meyn, Alcione Jandir Candéas Ramos, Amith Kulkarni, Anil Joshi, Atluri Kumar, Bonnie Brown, Checng Cheng Zhao, Connie Smith, Dan Schultz, Elisa Gallo, Emilia Leboreiro, Ganeshan Guruswamy, Giselle Fróes, Gurpreetkaur Marve, Jay Fischer, Jaya Venkat, Jeff Lindskoog, Jim Watson, Joel Andre, Julian Gómez Bejarano, Jyoti Namjoshi, Kareem Quereshi, Kathy Lamoureaux, Krupanandra Babu, Krzysztof Klosowski, Lionel Perrot, Luigi Buglione, Luis Andasola, Luis Carlos Grande, Luís Geminiano, Luz Ibáñez, Manoj Ramchandra Sable, Marcelo Leme, Marcin Soboń, Mario Vacalebri, Michael Fuller, Muhammet Oezteurk, Namit Mishra, Neetika Jain, Nicolas Chantereau, Nitin S Dharrao, Pablo Soneira, Padmanabhan Kalyanasundaram, Parag Saha, Peter Grüner, Peter Thomas, Philippe Perrin, Piotr Popowski, Pranabendu Bhattacharyya, Radhika Srinivas, Rafał Wasilewski, Rajib Banerjee, Ranjana Jain, Ravindra Moghe, Renata Faray, Rocío Juárez, Sachin Thakur, Savitha G, Sherry Ferrel, Shreyas Bhargave, Sridhar Maheswaram, Srividhya Ganesan, Stanley Richardson, Susan Walls, Swapnil Dambe, Tomasz Gąsiorowski, Tomasz Marchel, Usha Yegnaseshan, Valkal Bairagi, Viral Shah, Wojciech Dratnal

---

### **Companhias do teste Beta e revisoras do SNAP – Obrigado!**

Accenture, ALLIGRA (groupe Astek), AMDOCS, Banco Bradesco S/A, Banco do Brasil, Banco de Crédito del Perú, BRQ IT Solutions, Business Integration Partners S.p.a., Centurylink, Cognizant Technology Solutions, CSI Piemonte, Engineering.IT SpA, FÓTON INFORMÁTICA S.A, GE Energy, GDF Suez, HP Enterprise Services, IBM, Kubota Systems Inc., Level 4 Ventures, Inc., MindTree, Mphasis an HP Company, Neoris, Orange Polska, SEMANTYS (Groupe Astek), SIDI srl, Sopra Group, Tata Consultancy Services Ltd, TELUS, TI METRICAS SERVICOS LTDA, T-Systems, Austria,UST global

Tradução brasileira: Daniel Messias, TI Métricas, 2013

Revisão da tradução brasileira: Mauricio Aguiar, TI Métricas, 2013

Para informações sobre cópias adicionais deste manual, contatar:

IFPUG

191 Clarksville Road

Princeton Junction, NJ 08550

U.S.A.

(609) 799-4900

E-mail: [ifpug@ifpug.org](mailto:ifpug@ifpug.org) - Web: <http://www.ifpug.org>

## Conteúdo

Prefácio	v
Introdução ao Manual de Práticas de Avaliação	ix
<b>Parte 1 O Método SNAP</b>	
Capítulo 1 Introdução ao SNAP	1-1
Capítulo 2 Visão Geral do SNAP	2-1
Capítulo 3 Preparação da Avaliação	3-1
Capítulo 4 Determinar o Propósito, Escopo, Fronteira e	4-1
Capítulo 5 Categorias e Subcategorias	5-1
Capítulo 6 Calcular o Tamanho Não-Funcional (Pontos SNAP)	6-1
<b>Parte 2 Exemplos</b>	
Capítulo 1 Exemplos	1-1
<b>Parte 3 Apêndices</b>	
Apêndice A Glossário	A-1
Apêndice B Ligação entre o APM do IFPUG e o CPM do IFPUG	B-1
Apêndice C Índice	C-1

---

**Página deixada em branco intencionalmente**



### Introdução

Ter disponíveis tanto o tamanho Funcional quanto o Não-Funcional fornece uma imagem mais completa do desenvolvimento de software. O tamanho Funcional é quantificável e representa uma boa medida da funcionalidade do projeto/aplicação. A disponibilidade de uma medida quantificável dos Requisitos Não-Funcionais (NFR) permitirá que as organizações construam repositórios de dados históricos que poderão ser referenciados para auxiliar na tomada de decisões sobre aspectos técnicos e/ou de qualidade das aplicações.

Uma avaliação Não-Funcional irá auxiliar as organizações de Tecnologia da Informação (TI) de diversas maneiras. Proverá informações sobre projetos e aplicações para ajudar na estimativa e na análise da qualidade e produtividade. Utilizada em conjunto com a Análise de Pontos de Função, a Avaliação Não-Funcional provê informações que podem identificar itens que impactam a qualidade e produtividade, de modo positivo ou negativo. A posse destas informações tornará possível para os profissionais de software:

- Planejar e estimar melhor os projetos;
- Identificar áreas de melhoria de processos;
- Auxiliar na determinação de estratégias técnicas futuras;
- Quantificar os impactos das estratégias técnicas atuais;
- Prover dados específicos sobre questões Não-Funcionais na comunicação com diversos públicos.

## **História e Base do SNAP**

Na conferência ISMA de 2007, o Comitê de Performance de TI (ITPC) recebeu aprovação da Diretoria do IFPUG para prosseguir com o Projeto de Framework de Tamanho Técnico (TSF). O objetivo do projeto era definir um framework que pudesse mensurar os requisitos técnicos do desenvolvimento de software. A questão atual com o Tamanho Funcional é que o mesmo não tem sido adequado para o dimensionamento dos requisitos técnicos associados a um projeto de desenvolvimento de software. O foco do projeto foi definir um framework de tamanho técnico separado da metodologia de Pontos de Função (PF) criada pelo IFPUG. Tornar isto independente da medida de tamanho funcional garantiria que os dados históricos de pontos de função pudessem continuar sendo utilizados. O framework definido teve de ser acordado e apoiado pela Diretoria e pelos filiados do IFPUG. O produto final definiria as diretrizes e regras para medir os requisitos não-funcionais do desenvolvimento de software e incluiria aspectos não-funcionais.

## **Versão Inicial**

O trabalho da equipe do Projeto do Processo de Avaliação Não-Funcional de Software (SNAP) resultou em uma primeira versão publicada em Outubro de 2009, com a finalidade de ser revisada pela equipe de revisores, assim como pela Diretoria do IFPUG, pelo Comitê de Novos Ambientes do IFPUG e pelo Comitê de Práticas de Contagem do IFPUG. Isto foi refletido na Versão 0.1 (outubro de 2009) do Manual de Práticas de Avaliação SNAP (APM) do International Function Users Group (IFPUG).

## **Versão Beta**

O trabalho da equipe do Projeto SNAP, juntamente com o retorno dos revisores, resultou na primeira versão Beta, publicada em Novembro de 2010, com a finalidade de obter uma revisão da Indústria de Software em geral. Isto resultou na Versão 1.0 Beta (novembro de 2010) do APM SNAP. O retorno da primeira revisão resultou em alterações em pequenas partes do APM.

Um teste Beta foi realizado em janeiro de 2011, sobre a segunda versão Beta, para confirmar o processo, bem como para disponibilizar dados para a consolidação do modelo. Pequenas modificações foram feitas no APM, assim como confirmações das Unidades de Tamanho SNAP usadas na maioria das subcategorias. Poucas alterações foram realizadas nas subcategorias, incluindo a adição de uma nova subcategoria.

Um teste Beta adicional foi realizado em Maio de 2011, para obter os dados finais para a consolidação do cálculo do modelo em uma medida de tamanho consistente.

O teste Beta foi realizado por diversas companhias ao redor do mundo.

## **Versão 1.0**

A versão pública inicial do APM SNAP (setembro de 2011), definiu os conceitos, processos e regras para a avaliação dos requisitos não-funcionais de software. A Versão 1.0 do APM é o resultado de anos de trabalho da equipe SNAP e de seus experientes contribuidores. Esta versão incluiu adições e correções coletadas durante o período de teste beta e contribuições adicionais da equipe.

## Versão 2.0

A segunda versão pública do APM SNAP (janeiro de 2013), refina os conceitos, processos e regras para a avaliação dos requisitos não-funcionais do software.

Novas definições foram inseridas para esclarecer os termos utilizados neste manual.

Algumas subcategorias foram redefinidas baseando-se na experiência e comentários de usuários.

Foram acrescentadas diretrizes para refinar a ligação entre PF e SNAP, e detalhar como medir requisitos que envolvam aspectos funcionais e não-funcionais. Os usuários devem verificar que não haja dimensionamento duplicado, o SNAP é complementar à APF e não uma substituição.

## Versões Futuras

Este documento pretende ser vivo. Devemos reconhecer como realizar avaliações em novos ambientes conforme os mesmos forem aparecendo. Precisamos ser capazes de fazer isso no contexto da manutenção da validade das avaliações já feitas. Esta não é uma tarefa fácil, mas é essencial se quisermos ser capazes de medir o progresso que estamos fazendo na entrega de valor para os usuários e organizações que representam.

O Comitê de Padrões de Medição Não-Funcional (anteriormente Comitê de Desempenho de TI) deseja agradecer a todos que tenham nos ajudado em nossa pesquisa e na produção deste manual.

Talmon Ben-Cnaan

Chairperson, NFSSC.

**Página deixada em branco intencionalmente**

---

## Introdução ao Manual de Práticas de Avaliação

---

**Introdução** Esta introdução define os objetivos deste manual e o processo de revisão. Também são descritas as publicações relacionadas a este manual.

**Conteúdo** Este capítulo inclui as seguintes seções:

<b>Tópico</b>	<b>Página</b>
Objetivos do Manual de Práticas de Avaliação	x
Teste Beta do SNA	x
Documentos Utilizado	x
Público-Alvo	xi
Organização do Manual de Práticas de Avaliação	xi
Processo de Revisão do Manual	xii
Frequência de Alterações	xii
Processo de Alteração	xii
Documentação do IFPUG Relacionada	xiv
Requisitos de Treinamento	xv

## Objetivos do Manual de Práticas de Avaliação

Os objetivos primários do Manual de Práticas de Avaliação do IFPUG são:

- Prover uma descrição clara e detalhada da Avaliação Não-Funcional;
- Garantir que as avaliações sejam consistentes;
- Prover orientações para possibilitar que a Avaliação Não-Funcional seja aplicada às metodologias e técnicas populares;
- Prover um entendimento comum para possibilitar aos fornecedores de ferramentas proverem suporte automatizado para a Avaliação Não-Funcional.

## Teste Beta do SNAP

O propósito do teste beta do SNAP foi repetir o espírito do teste do Dr. Allan Albrecht na versão inicial da metodologia de pontos de função, conforme documentado em seu artigo de 1977, “Measuring Application Development Productivity”. Nosso teste beta, similar ao teste de pontos de função do Dr. Albrecht, encontrou uma correlação estatisticamente significativa entre o tamanho SNAP e o esforço, utilizando uma amostra estatisticamente grande, com 48 aplicações contendo mais de 500 entradas de dados. O  $r^2$  para a correlação entre a contagem SNAP e o esforço foi de .89, a correlação de ordem Spearman foi de .85, os ‘p-values’ correspondentes para ambos os testes foram inferiores a 0,0001 e o teste de aleatoriedade do modelo de regressão passou no ‘runs test’.

Tais estatísticas significam que, para este teste beta, o tamanho SNAP explicou 89% da variação no esforço gasto (os 11% restantes podem resultar das diferentes linguagens de programação, habilidade das equipes, erros de contagem, etc.).

## Documentos Utilizados

A documentação a seguir foi utilizada para elaborar esta versão:

- “International Function Point Users Group (IFPUG) Function Point Counting Practices Manual” (CPM), Versão 4.3.1, este manual do IFPUG foi utilizado para reutilização de definições e outros itens para alinhar o SNAP com os princípios atuais do Tamanho Funcional;
- “Framework for Functional Sizing”. Este artigo do IFPUG explica que o tamanho do produto possui três dimensões: tamanho funcional, tamanho técnico e o tamanho de qualidade. O método de APF do IFPUG fornece uma medida para o tamanho funcional;
- “Information technology – Software measurement – Functional size measurement – Part 1: Definition of concepts” da ISO/IEC 14143-1:2007 e “Systems and software engineering – Vocabulary” da ISO/IEC/IEEE 24765:2010. Utilizados como referências-padrão para definições dos Requisitos Funcionais e Não-Funcionais.

Questões não suficientemente cobertas pelas fontes acima foram decididas pelo Comitê de Performance de TI do IFPUG, baseadas em variantes das práticas de contagem atuais e validadas através de estudos de impacto.

Com esta versão, este manual deve ser considerado o padrão do IFPUG para a medição não-funcional. É imprescindível que cada filiado do IFPUG assuma um papel ativo para assegurar a consistência na contagem. A aderência dos filiados do IFPUG a este padrão contribuirá consideravelmente para a consistência da contagem.

## Público-Alvo

O padrão neste manual deve ser aplicado a qualquer avaliação de tamanho não-funcional de produto. O manual foi elaborado para uso tanto de iniciantes, quanto de profissionais de nível intermediário, ou experientes em avaliações.

## Organização do Manual de Práticas de Avaliação

Há três partes principais no Manual de Práticas de Avaliação (APM):

- Parte 1 O Método SNAP;
- Parte 2 Exemplos;
- Parte 3 Apêndices.

### Parte 1 - O Método SNAP

A Parte 1 é o método de medição de tamanho não-funcional do IFPUG que contém suas regras. Para falar um idioma como um nativo, aprender a gramática e palavras soltas não é suficiente. Isto fornece apenas um framework. É necessário experiência no idioma para entender como o idioma é falado na prática, como as regras de gramática deve ser aplicadas, quais expressões idiomáticas são comuns e assim por diante. É o mesmo com o SNAP. O conhecimento dos processos e regras, como refletidas na Parte 1, é necessário, mas, o conhecimento por si só não é suficiente para aplicar corretamente o SNAP. Por isso o APM contém as demais partes abaixo.

### Parte 2 - Exemplos

A Parte 2 apresenta exemplos detalhados para explicar os conceitos de práticas de contagem e regras. Cada exemplo deve ser considerado separadamente. Podem existir variações entre os exemplos, já que cada exemplo tem a intenção de ilustrar um cenário específico. Embora os exemplos contidos no Manual lidem com assuntos similares, não pretendem representar um único conjunto de requisitos.

### Parte 3 - Apêndices

A Parte 3 contém informações adicionais (Glossário, Ligação do APM com o CPM do IFPUG e o Índice).

Por princípio, cada parte é independente das demais.

## Processo de Revisão do Manual

Esta seção aborda a frequência de alterações no Manual de Práticas de Avaliação e define o processo de alteração.

## Frequência de Alterações

No mês de janeiro de cada ano poderá entrar em vigor uma nova versão do Manual de Práticas de Avaliação. A mesma incluirá quaisquer definições novas ou alteradas, bem como regras ou práticas de Avaliação que tenham sido finalizadas pelo Comitê de Padrões de Medição Não-Funcional (NFSSC) desde a versão anterior.

## Processo de Alteração

As atividades a seguir descrevem o processo para incluir ou alterar informações no Manual de Práticas de Avaliação. As explicações de cada atividade seguem no quadro abaixo.

Passo	Ação
1	A questão é submetida ao NFSSC.
2	A questão é atribuída para pesquisa.
3	O NFSSC revisa e discute a questão.
4	O NFSSC apresenta a solução proposta aos filiados do IFPUG.
5	Um estudo de impacto é iniciado caso a proposta de alteração tenha qualquer impacto em contagens existentes.
6	A decisão final é tomada.
7	Os filiados ao IFPUG são informados da decisão.
8	As alterações entram em vigor na próxima versão do Manual de Práticas de Avaliação.

- 1**  
**Questão Submetida**  
O leitor pode submeter ideias, alterações ou questões ao Comitê de Padrões de Medição Não-Funcional enviando um e-mail para [ifpug@ifpug.org](mailto:ifpug@ifpug.org) ou [nfssc@ifpug.org](mailto:nfssc@ifpug.org).
- 2**  
**Atribuída para Pesquisa**  
Um membro do ITPC recebe a responsabilidade de identificar todas as alternativas, a lógica e os impactos potenciais de cada alternativa, caso seja implementada. Uma análise aprofundada dos padrões de contagem existentes e de artigos históricos é realizada durante a compilação das alternativas. Além disso, procura-se determinar o que são consideradas práticas comuns.
- 3**  
**Revisão do NFCS**  
O NFSSC revisa e discute a base lógica para cada alternativa e seu impacto potencial. Esta revisão e discussão podem resultar em uma proposta de alteração ou a revisão pode levar o comitê a rejeitar o pedido de alteração.



- 4  
Solução  
Proposta**
- Uma solução é proposta aos filiados ao IFPUG e comentários por escrito são solicitados.
- Uma cópia das alterações propostas é enviada aos contatos do IFPUG em organizações filiadas. A proposta também pode ser anunciada e discutida durante uma conferência do IFPUG. Esta última opção depende do momento da reunião do comitê e não da programação da conferência.
- 5  
Estudo de  
Impacto  
Iniciado**
- O NFSSC adotou uma postura conservadora quanto à realização de estudos de impacto. Realiza-se um estudo de impacto caso seja possível que uma prática comum vá sofrer mudança, ou caso diversas organizações ou tipos de aplicações sejam afetados pela mudança.
- O sucesso do estudo de impacto é de responsabilidade de cada filiado do IFPUG. Caso o NFSSC receba retornos por escrito indicando que há um pequeno impacto ou que não há impacto, o estudo de impacto é cancelado.
- 6  
Decisão Final**
- O comitê toma a decisão final utilizando resultados de pesquisas, comentários escritos de filiados e o estudo de impacto.
- O comitê pode executar mais de uma iteração dos passos 2 ao 5 (pesquisa através do estudo de impacto) antes de tomar a decisão final. A decisão final pode resultar em uma mudança, ou o comitê pode decidir que a alteração não é devida.
- 7  
Decisão  
Comunicada**
- A decisão final é comunicada por escrito aos filiados ao IFPUG através dos contatos do IFPUG em várias organizações.
- Se quaisquer resultados dos estudos de impacto contribuírem para a tomada de decisão, os resultados e uma recomendação sobre como minimizar o impacto da alteração também serão comunicados.
- 8  
Data de  
Entrada em  
Vigor da  
Decisão**
- O Manual de Práticas de Avaliação será atualizado para refletir as decisões. A data de entrada em vigor das decisões será a data da próxima versão de janeiro do manual.

## Documentação do IFPUG Relacionada

Este Manual de Práticas de Avaliação é um módulo na documentação do IFPUG. Todos os documentos se complementam.

A seguinte lista descreve as outras publicações.

Documento	Descrição
Folheto sobre o IFPUG (Disponível)	Esta publicação é uma introdução ao International Function Point Users Group. Inclui uma breve história da organização, introduz a análise de pontos de função e define o propósito do IFPUG. O folheto também inclui um formulário de filiação.  Público: Esta publicação é destinada a todos que desejem ter uma visão geral do IFPUG ou um formulário de filiação.
IFPUG: Estrutura Organizacional e Serviços (Disponível)	Esta publicação descreve os serviços do IFPUG e a lista a diretoria, comitês e membros afiliados ao redor do mundo.  Público: Esta publicação é destinada a todos que queiram informações internas sobre o IFPUG.
Manual de Práticas de Contagem (Data da Versão: Janeiro de 2010)	International Function Point Users Group (IFPUG) Function Point Counting Practices Manual (CPM), Versão 4.3.1.
Guia para Medição de Software (Publicado em: Agosto de 2004)	Este manual provê uma visão geral de métricas de software para as organizações trabalharem em criar ou aperfeiçoar programas de medição de software. O manual aborda ambos o gerenciamento de sistemas e o de clientes, fornece justificativas de alto nível para a medição de software e examina os componentes de programas ativos de medição.  Público: Este manual é destinado a filiados do IFPUG, Coordenadores de Pontos de Função, pessoas que preparam relatórios de gerenciamento e outros com conhecimento e que trabalhem diretamente com pontos de função.
Guia de Referência Rápida de Contagem v2.0 (Publicado em: 2009)	Este guia de referência rápida é um resumo das regras e procedimentos da contagem de pontos de função.  Público: Esta informação é destinada a qualquer pessoa que aplique a análise de pontos de função.
Guia de Referência Rápida de Tamanho Funcional Ajustado (Publicado em: 2010)	Este guia de referência rápida é um sumário das Características Gerais do Sistema.  Público: Esta informação destina-se a qualquer pessoa que utilize as Características Gerais do Sistema, cuja utilização é opcional.
Glossário do IFPUG (Disponível com o CPM e com o Guia de Medição de Software)	Este é um extenso glossário que define os termos usados nas publicações do IFPUG.  Público: O glossário é recomendado a todos que recebam qualquer documento do IFPUG, ou qualquer pessoa que precise de definições dos termos do IFPUG.
A Framework for Functional Sizing, IFPUG, Setembro de 2003	Este artigo explica que o tamanho do projeto possui três dimensões: tamanho funcional, tamanho técnico e tamanho de qualidade. O método de APF do IFPUG provê uma medição para o tamanho funcional.
IT Measurement: Practical Advice from the Experts, Addison-Wesley, Abril de 2002	Este livro é uma excelente compilação de artigos escritos por especialistas do ramo da Tecnologia da Informação. Foi compilado pelo IFPUG para incluir ideias recentes sobre a aplicação de métricas de software na prática.

O Guia do IFPUG para Medição de TI e de Software

(Publicado em: 2012)

O Guia do IFPUG para Medição de TI e de Software reuniu 52 líderes especialistas em medição de software, de 13 países diferentes, que compartilharam suas ideias e experiências. Abordando programas de medição, pontos de função na medição, novas tecnologias e análise de métricas, estes volume:

- Ilustra o papel da medição de software em tecnologias novas e emergentes;
- Aborda o impacto do desenvolvimento ágil sobre a medição de software;
- Apresenta a medição como uma poderosa ferramenta para auditoria e prestação de contas;

Inclui métricas para o CIO.

Estudos de Caso de Análise de Pontos de Função

(Publicados em:

Estudo de Caso 1 Versão 3.0, Setembro de 2005 (CPM 4.2)

Estudo de Caso 2 Versão 3.0, Março de 2006 (CPM 4.2)

Estudo de Caso 3 Versão 2.0, Setembro de 2001 (CPM 4.1)

Estudo de Caso 4 Versão 2.0, Setembro de 2005 (CPM 4.2)

Os estudos de caso ilustram as principais técnicas de contagem que compreendem o Manual de Práticas de Contagem de Pontos de Função. Os casos ilustram contagens de pontos de função para uma aplicação exemplo. Incluem contagens que ocorrem ao final da fase de análise do desenvolvimento de software e após a construção do sistema.

Público: Os estudos de caso são destinados a pessoas iniciantes em análise de pontos de função, bem como àquelas com experiência intermediária ou avançada.

## Requisitos de Treinamento

As validações de usabilidade desta publicação indicaram que apenas a leitura do Manual de Práticas de Avaliação não constitui treinamento suficiente para aplicar o processo em seu melhor nível. É recomendado treinamento, especialmente para aqueles que são novos no SNAP.

Um workshop com duração de dois dias foi desenvolvido pelo IFPUG para prover treinamento sobre o SNAP.

Até o momento, o IFPUG não desenvolveu um treinamento oficial de certificação ou linha de treinamento para o SNAP.

**Página deixada em branco intencionalmente**

---

## Parte 1 – O Método SNAP

---

---

**Página deixada em branco intencionalmente**

---

---

## Introdução ao SNAP

---

**Introdução** Este capítulo apresenta o Framework de Avaliação para Medição Não-Funcional e uma descrição dos objetivos e benefícios do SNAP.

**Conteúdo** Este capítulo inclui as seguintes seções:

<b>Tópico</b>	<b>Página</b>
Um Framework para a Medição Não-Funciona	1-2
Relação entre o APM e Outros Padrões	1-2
ISO/IEC/IEEE – Definições	1-3
Esforço do Projet	1-8
O Framework SNAP	1-9
Objetivos e Benefícios do SNA	1-10

## Um Framework para a Medição Não-Funcional

O objetivo principal do projeto do Framework para Medição Não-Funcional do IFPUG (2008), foi garantir que um framework não-funcional possa ser utilizado para estabelecer uma ligação entre o tamanho não-funcional e o esforço para prover os requisitos não-funcionais.

O framework resultante possui as seguintes características:

- O framework como um todo é uma avaliação do tamanho dos requisitos não-funcionais;
- O framework é composto de categorias e subcategorias de avaliação;
- As subcategorias são avaliadas utilizando critérios especificados;
- A avaliação utiliza ambos os critérios de avaliação e/ou de medição.

Os resultados da avaliação não-funcional possuem as seguintes características:

- Podem ser utilizados em conjunto com o tamanho funcional e irão auxiliar na explicação da variação do esforço de desenvolvimento e da produtividade;
- Juntamente com o tamanho funcional, podem ser utilizados como entrada para modelos de estimativas;
- São determinados a partir de uma visão não-funcional do usuário, porém entendidos e acordados entre o usuário e as equipes de desenvolvimento.

## Relação entre o APM e Outros Padrões

O Manual de Práticas de Avaliação (APM) do IFPUG para o Processo de Avaliação Não-Funcional de Software (SNAP) utiliza definições e terminologias de importantes organizações internacionais de padronização tais como ISO, IEC e IEEE, sempre que possível.

Uma série de definições base é fornecida na seção seguinte, a respeito da classificação dos requisitos.



## ISO/IEC/IEEE – Definições

### Requisitos do Usuário

Em 1998 foi publicada a primeira norma ISO/IEC de Medição de Tamanho Funcional (ISO/IEC 14143-1:1998 “Software and Systems Engineering – Software measurement – Functional size measurement – Definition of concepts”). Esta norma define o Tamanho Funcional como “um tamanho do software entregue através da quantificação dos Requisitos Funcionais do Usuário” (FUR). (Esta norma foi atualizada em 2007 e é publicada atualmente como ISO/IEC 14143-1:2007.)

A ISO/IEC 14143-1 distingue dois subconjuntos de requisitos do usuário (RU):

- Requisitos Funcionais do Usuário (RFU);
- Requisitos Não-Funcionais (RNF).

**ISO/IEC 9126-1:2001** fornece a definição de características e do processo de avaliação da qualidade associado, a ser utilizado ao especificar os requisitos para avaliação da qualidade dos produtos de software ao longo de seu ciclo de vida.

A **ISO/IEC 25010:2011** substituiu a ISO/IEC 9196-1.

A definição da norma ISO/IEC 14143-1 para os Requisitos Funcionais do Usuário é a seguinte:

### Requisitos Funcionais do Usuário

“Um subconjunto dos Requisitos do Usuário (RU). Requisitos que descrevem o que o software deve fazer, em termos de tarefas e serviços.”

Nota: Requisitos Funcionais do Usuário incluem, mas não estão limitados a:

- Transferência de dados (por exemplo, Inserir dados de clientes, Enviar sinal de controle);
- Transformação de dados (por exemplo, Calcular juros bancários, Derivar a temperatura média);
- Armazenamento de dados (por exemplo, Armazenar pedidos de clientes, Gravar a temperatura ambiente ao longo do tempo);
- Recuperação de dados (por exemplo, Listar os empregados atuais, Recuperar a posição da aeronave).

Exemplos de Requisitos do Usuário que não são Requisitos Funcionais do Usuário incluem, mas não estão limitados à:

- Restrições de qualidade (por exemplo, usabilidade, confiabilidade, eficiência e portabilidade);
- Restrições organizacionais (por exemplo, locais para operação, hardware-alvo, aderência a normas);
- Restrições ambientais (por exemplo, interoperabilidade, segurança, privacidade e proteção);
- Restrições de implementação (por exemplo, linguagem de desenvolvimento e prazo de entrega).

A **ISO/IEC 9126-1:2001** define funcionalidade como segue:

Funcionalidade é o conjunto de atributos relevantes para a existência de um conjunto de funções e suas propriedades específicas. As funções são aquelas que satisfazem as necessidades explícitas e implícitas.

A ISO/IEC 14143-1 não provê uma definição para requisitos não-funcionais, mas fornece alguns exemplos em uma nota após a definição de RFU. Em 2009, uma iniciativa separada sobre desenvolvimento (ref. ISO/IEC/IEEE 24765:2010 Systems and software engineering – Vocabulary) propôs uma definição formal de um Requisito Não-Funcional do Usuário, como segue:

### **Requisitos Não-Funcionais do Usuário**

- “Um requisito de software que descreve não o que o software irá fazer, mas como o software irá fazer.” [ISO/IEC 24765, Systems and Software Engineering Vocabulary]. Sinônimos: restrições de projeto, requisitos não-funcionais. Veja também: requisito funcional.

Exemplos: requisitos de desempenho de software, requisitos de interface externa de software, restrições de projeto de software e atributos de qualidade de software.

Nota: Requisitos Não-funcionais são, algumas vezes, difíceis de testar, então são, usualmente, avaliados subjetivamente.

Novamente e analogamente, a ISO/IEC/IEEE 24765 define um Requisito Funcional como:

- Uma afirmação que identifica o que um produto ou processo deve realizar para produzir o comportamento e/ou resultados requeridos. [ISO/IEC 26710:2007 (IEEE std.1220-2005) IEEE Standard for the Application and Management of the Systems Engineering Process. § 3.1.1.6];
- Um requisito que especifica uma função que um sistema ou componente do sistema deve ser capaz de executar [ISO/IEC 24765, Systems and Software Engineering Vocabulary].

### **ISO/IEC 25010:2011**

O modelo de qualidade de produto categoriza as propriedades da qualidade de produto segundo oito características (adequação funcional, confiabilidade, eficiência de desempenho, usabilidade, segurança, compatibilidade, manutenibilidade e portabilidade).

Cada característica é composta de um conjunto de subcategorias relacionadas.

- **Adequação Funcional**

Grau em que um produto ou sistema disponibiliza funções que atendam necessidades explícitas e implícitas, quando usadas sob condições especificadas.

- Completude Funcional;
- Correção Funcional;
- Adequação Funcional.

- **Eficiência de Desempenho**

Desempenho em relação à quantidade de recursos utilizados sob condições especificadas.

- Tempo de Resposta;
- Utilização de Recursos;
- Capacidade.

- **Compatibilidade**

Grau em que um produto, sistema ou componente pode trocar informações com outros produtos, sistemas ou componentes e/ou desempenha suas funções requeridas enquanto compartilha o mesmo hardware ou ambiente de software.

- Coexistência;
- Interoperabilidade.

- **Usabilidade**

Grau em que um produto ou sistema pode ser usado por usuários especificados para alcançar objetivos especificados em um contexto de uso especificado, com eficácia, eficiência e satisfação.

- Reconhecimento da Adequabilidade;
- Aprendizabilidade;
- Operabilidade;
- Proteção a erros do usuário;
- Estética da interface com o usuário;
- Acessibilidade.

- **Confiabilidade**

Grau em que um sistema, produto ou componente desempenha funções específicas em condições definidas por um período de tempo especificado.

- Maturidade;
- Disponibilidade;
- Tolerância a falhas;
- Recuperabilidade.

- **Segurança**

Grau em que um produto ou sistema protege informações e dados, de forma que pessoas ou outros produtos ou sistemas tenham o nível de acesso a dados apropriado para os seus tipos e níveis de autorização.

- Confidencialidade;
- Integridade;
- Não repúdio;
- Responsabilidade ('accountability');
- Autenticidade.

- **Manutenibilidade**

Grau de eficácia e eficiência com que um produto ou sistema pode ser modificado por seus mantenedores.

- Modularidade;
- Reusabilidade;
- Analisabilidade;
- Modificabilidade;
- Testabilidade.

- **Portabilidade**

Grau de eficácia e eficiência com que um sistema, produto ou componente pode ser transferido de um ambiente de hardware, software

ou outro ambiente operacional ou de utilização, para outro.

- Adaptabilidade;
- Instalabilidade;
- Substituibilidade.

### **Relação entre o APM e a ISO/IEC 9126-1, ISO/IEC 25010**

As subcategorias do SNAP não definem ou descrevem requisitos não-funcionais, apenas classificam como estes requisitos são encontrados em um produto de software.

Um requisito não-funcional definido pela ISO/IEC 9126-1 ou pela ISO/IEC 25010 pode ser implementado no produto utilizando-se mais de uma subcategoria. Consequentemente, uma subcategoria pode satisfazer diversas características de qualidade, como definidas pela ISO/IEC 9126-1 ou ISO/IEC 25010.

Uma visão desta relação é apresentada no exemplo e na tabela 1-1 abaixo.

### **Exemplo**

Este exemplo analisa um requisito para aperfeiçoar o modo segundo o qual o sistema se recupera de um erro:

Usando a ISO/IEC 25010, este requisito recai sobre a área de Confiabilidade e o atributo é Recuperabilidade.

O SNAP mede este requisito de acordo com a solução:

- Um algoritmo é incluído para identificar dados corrompidos em campos específicos;
- Marcadores de tempo ('time stamps') são incluídos nos registros da base de dados;
- Um algoritmo é escrito para reconstruir os dados corrompidos usando registros não corrompidos.

A solução envolve as seguintes subcategorias do SNAP:

- Tecnologia de Banco de Dados (incluir o marcador de tempo);
- Operações Lógicas e Matemáticas.

Neste exemplo, o requisito do tipo "recuperabilidade" é mapeado em duas subcategorias. "Tecnologia de Banco de Dados" e "Operações Lógicas e Matemáticas".

Mais exemplos de tal mapeamento são ilustrados na tabela 1-1.

### **Nota**

Todos os exemplos na tabela 1-1 assumem que tais requisitos não são cobertos por Pontos de Função.

Para aprender mais sobre a ligação entre APF e SNAP, por gentileza consulte o Apêndice B e as tabelas 3-1 e 3-2.

Subcaracterísticas da ISO 25010 (Uma lista parcial para demonstração)		Categorias e Subcategorias do SNAP													
		Operações de Dados					Design de Interface				Ambiente Técnico			Arquitetura	
		Validações na Entrada de Dados	Operações Lógicas e Matemáticas	Formatação de Dados	Movimentação de Dados Internos	Funcionalidade de Entrega por Configuração de Dados	Alterações de IU	Métodos de Ajuda	Múltiplos Métodos de Entrada	Múltiplos Métodos de Saída	Múltiplas Plataformas	Tecnologia do Banco de Dados	Processos Batch	Software Baseado em Componentes	Múltiplas Interfaces de Entrada/Saída
Portabilidade	Substituibilidade														
	Instalabilidade														
	Adaptabilidade														
Segurança	Alterabilidade														
	Autenticidade	Ex. 2													
Confiabilidade	Tolerância a falhas														
	Recuperabilidade														
Usabilidade	Acessibilidade														
	Proteção a erros do usuário														
	Operabilidade							Ex. 4							
	Estética de interface do usuário						Ex. 1								
	Aprendizibilidade														
	Reconhecimento Adequada														
Eficiência de Desempenho	Capacidade														
	Utilização de recursos														
	Tempo de Resposta										Ex. 3				
Adequação Funcional	Compleitude Funcional														
	Adequação Funcional														
	Correção Funcional														

Tabela 1-1 Mapeamento das características da ISO/IEC 25010 para as subcategorias do SNAP

Os números dos exemplos na Tabela 1-1 referem-se a:

- Ex. 1. Aperfeiçoar a facilidade de entendimento e aprendizagem através da inclusão de menus pop-ups de ajuda e reorganização das telas.
- Ex. 2. Melhorar a Segurança, através da inclusão de mais validações no processo de autenticação, usando “validações na entrada de dados” e “operações Lógicas e Matemáticas”.
- Ex. 3. Melhorar o desempenho, incluindo índices nas bases de dados e melhorando as consultas.
- Ex. 4. Incluir a leitura de código de barras como um método de entrada adicional.

## Esforço do Projeto

### Projeto de Software

Um empreendimento colaborativo, que é planejado e executado a fim de alcançar um determinado objetivo, principalmente através de desenvolvimento de software.

IEEE Std-1058-1998: O conjunto de atividades técnicas e gerenciais requeridas para satisfazer os termos e condições de um contrato de projeto.

O esforço do projeto é investido para prover os seguintes itens:

- Características que são incluídas no produto para satisfazer os requisitos funcionais (veja a definição em capítulo anterior);
- Características que são incluídas no produto para satisfazer os requisitos não-funcionais (veja a definição em capítulo anterior);
- Tarefas de projeto para garantir que:
  - O projeto seja gerenciado;
  - O projeto satisfaça as restrições de qualidade, cronograma e orçamento;
  - Os riscos sejam gerenciados.

#### **Satisfazer os requisitos funcionais – o tamanho funcional do Produto**

O tamanho funcional pode ser mensurado/avaliado utilizando o método de medição de tamanho funcional do IFPUG, baseando-se nos requisitos funcionais do usuário (RFU).

#### **Satisfazer os requisitos não-funcionais – o tamanho não-funcional do Produto**

O framework do SNAP fornece a base para a medição dos requisitos não-funcionais.

#### **Executando as tarefas do projeto**

As tarefas do projeto não afetam o tamanho do Produto. Embora estas tarefas afetem o esforço requerido para entregar o produto, influenciam a produtividade e não o tamanho do software.

Exemplos de tarefas do projeto:

- Treinamento da equipe;
- Treinamento do usuário;
- Documentação do projeto (manuais, planos, status reports, roteiros, instruções de trabalho, padrões de qualidade).

Os Pontos de Função e o SNAP resultam em diferentes medidas de tamanho, representando diferentes dimensões do tamanho do produto. Embora estes tamanhos não possam ser somados porque representam dimensões diferentes (como volume e temperatura de uma sala), podem ser utilizados em conjunto para estimar o esforço de desenvolvimento de uma aplicação ou de um sistema.

Adicionalmente, as tarefas do projeto também são utilizadas para estimar o esforço de desenvolvimento da aplicação ou sistema.

## O Framework SNAP

O framework SNAP e os pontos de função podem ser vistos como três dimensões de um bloco:

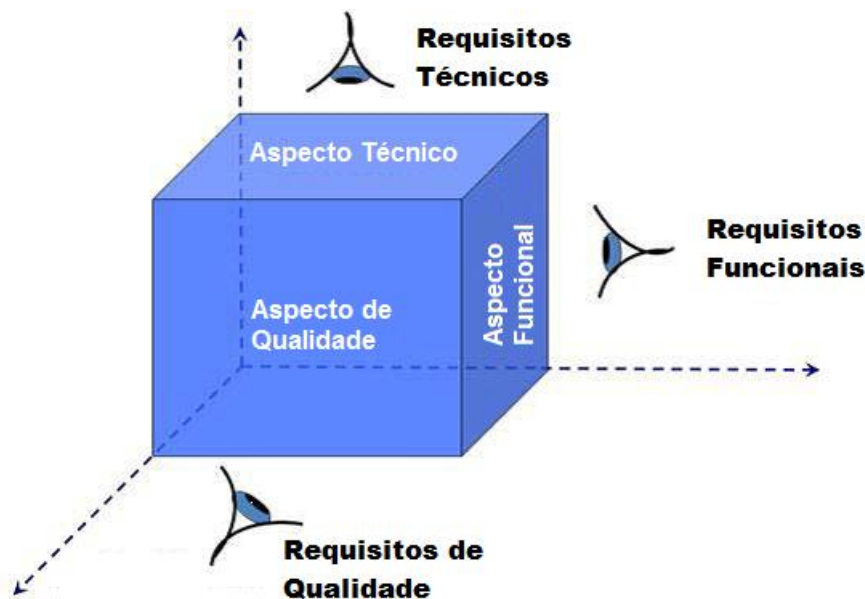


Figura 1-1 O Framework e Requisitos

A dimensão dos requisitos funcionais e a perspectiva funcional do projeto de desenvolvimento de software são atualmente abrangidas pela medida de tamanho funcional - pontos de função do IFPUG, tal como definido no Manual de Práticas de Contagem do IFPUG. A dimensão dos requisitos não-funcionais (tanto técnicos como de qualidade) e a perspectiva não-funcional do projeto de desenvolvimento de software são definidos neste manual.

A seguir, apresentamos as definições de requisitos técnicos produzidas pelas organizações de padronização:

- ISO – Requisitos relacionados à tecnologia e ao ambiente para o desenvolvimento, manutenção, suporte e execução do software;
- IEEE – Combinação de requisitos de projeto, implementação, interface, desempenho e físicos.

Os Requisitos de Qualidade são requisitos que não são definidos como funcionais ou técnicos e se relacionam à qualidade do sistema ou componentes. As seguintes definições estão disponíveis:

- ISO – As seguintes características fazem parte do modelo de qualidade: Funcionalidade, Confiabilidade, Usabilidade, Eficiência, Manutenibilidade e Portabilidade.
- IFPUG – A qualidade inclui: conformidade às expectativas do usuário, requisitos do usuário, satisfação do cliente, confiabilidade e nível de defeitos presentes.



## Objetivos e Benefícios do SNAP

### Objetivos

O SNAP mede o software pela quantificação do tamanho dos requisitos não-funcionais. Com isto em mente, os objetivos do SNAP são:

- Medir o tamanho não-funcional do software que o usuário solicita e recebe;
- Medir o desenvolvimento e manutenção de software baseado nos requisitos não-funcionais (por exemplo, a tecnologia utilizada na implementação).

Além de atender os objetivos acima, o processo de avaliação de requisitos não-funcionais deve ser:

- Simples o suficiente para minimizar a sobrecarga do processo de medição;
- Uma medida consistente entre vários projetos e organizações.

### Benefícios

A avaliação não-funcional irá auxiliar as organização de TI de diversas formas. Proverá informações sobre a entrega de projetos e manutenção de aplicações, para ajudar na estimativa e na análise da qualidade e produtividade. Utilizada em conjunto com as medições em Pontos de Função, a avaliação não-funcional fornecerá informações que podem identificar itens que impactem a qualidade e a produtividade, de forma positiva ou negativa.

A posse destas informações tornará possível aos profissionais de software:

- Planejar e estimar melhor os projetos;
- Identificar áreas de melhoria de processos;
- Auxiliar na determinação futura de estratégias não-funcionais
- Quantificar os impactos das estratégias não-funcionais atuais;
- Prover dados específicos na comunicação de questões não-funcionais a vários públicos.

As organizações podem aplicar o SNAP como:

- Uma metodologia para medir o tamanho não-funcional de um produto de software, para dar suporte a análises de qualidade e produtividade;
- Uma metodologia para estimar custos e recursos necessários ao desenvolvimento e manutenção de software;
- Um fator de normalização para a comparação de softwares;
- Uma metodologia para determinar o tamanho não-funcional de um pacote de aplicativos adquirido, através da avaliação de todas as partes e categorias incluídas no pacote;
- Uma metodologia para ajudar os usuários a determinar os benefícios de um pacote de aplicativos para sua organização, através da avaliação de partes ou categorias que satisfaçam seus requisitos específicos.



**Nota:**

'PF + Pontos SNAP' **não** representam o tamanho total do produto.

Na data desta publicação, o tamanho de uma aplicação de software é considerado como tendo duas partes distintas: o tamanho dos requisitos funcionais e o tamanho dos requisitos não-funcionais. Por exemplo, caso o tamanho funcional da aplicação seja 700 pontos de função e o tamanho não-funcional seja 200 pontos SNAP, então, o tamanho total poderia ser declarado como “700 pontos de função e 200 pontos SNAP”. Não é correto considerar o tamanho da aplicação como sendo “900 pontos”.

Mais pesquisas são necessárias para determinar se os pontos de função e os pontos SNAP possam ser combinados de alguma forma em uma “única métrica”.

**Página deixada em branco intencionalmente**

---

## Visão Geral do SNAP

---

**Introdução** Este capítulo apresenta uma visão geral do SNAP. Apresenta um resumo dos conceitos relacionados e das etapas do processo.

**Conteúdo** Este capítulo inclui os seguintes tópicos:

<b>Tópico</b>	<b>Página</b>
Descrição do SNAP	2-2
O Processo de Avaliação Não-Funcional	2-2
Seção 1: Determinar o Propósito, Escopo, Fronteira e Partição da Avaliação	2-3
Seção 2: Associar os Requisitos Não-Funcionais às Categorias e Subcategorias	2-5
Seção 3: Identificar as Unidades de Contagem SNAP (UCSs)	2-7
Seção 4: Determinar a Complexidade de cada Unidade de Contagem SNAP (UCS)	2-7
Seção 5: Determinar os Pontos SNAP (PS) de cada UCS	2-8
Seção 6: Calcular o Tamanho Não-Funcional	2-8

## Descrição do SNAP

### Introdução

O propósito deste capítulo é descrever como os requisitos não-funcionais podem ser medidos utilizando o método SNAP.

Analisando vários cenários em uma aplicação sob os aspectos não-funcionais, diversas subcategorias são identificadas e agrupadas sob Categorias lógicas.

As categorias e subcategorias não substituem ou explicam os padrões que descrevem e classificam os requisitos não-funcionais (tais como a ISO/IEC 25010:2011). As Categorias e subcategorias descrevem como o projeto ou produto avaliado irá atender estes requisitos não-funcionais.

## O Processo de Avaliação Não-Funcional

A avaliação não-funcional utilizará uma série de questões agrupadas por categorias, para medir o tamanho dos requisitos não-funcionais para o desenvolvimento e a entrega do produto de software.

- As categorias focarão nos requisitos não-funcionais que afetam o tamanho do produto;
- O processo possibilitará a medição dos requisitos não-funcionais utilizando uma série de questões e medidas;
- O processo pode ser usado para Projetos de Desenvolvimento, Projetos de Melhoria, Atividades de Manutenção e Aplicações.

### Procedimento por Seção

A seguinte tabela apresenta o procedimento SNAP, como explicado nos demais capítulos da Parte 1.

Procedimento	Seção
Determinar o Propósito e o Escopo da Avaliação	1
Associar os requisitos não-funcionais às categorias e subcategorias	2
Identificar as UCSs	3
Determinar a complexidade da UCS	4
Calcular os Pontos SNAP da UCS	5
Calcular o Tamanho Não-Funcional (Pontos SNAP)	6

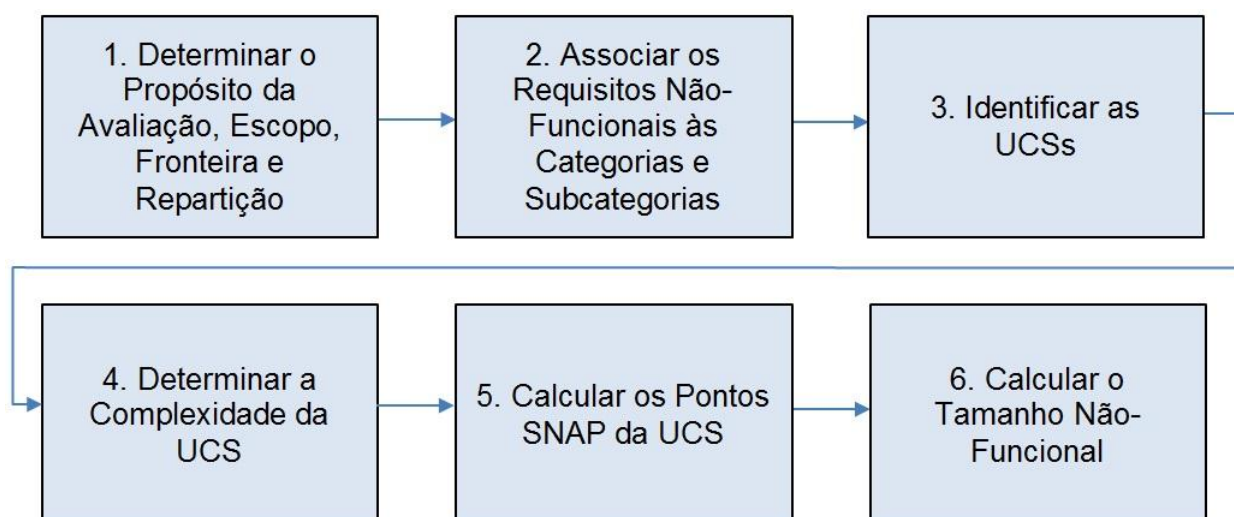


Figura 1-2 Diagrama SNAP

## Seção 1: Determinar o Propósito, Escopo, Fronteira e Partição da Avaliação

Os seguintes passos devem ser executados ao identificar o Propósito, Escopo, Fronteira e Partição da Avaliação:

Passo	Ação
1	Identificar o propósito da avaliação
2	Identificar o tipo de avaliação
3	Identificar o escopo da avaliação
4	Identificar a(s) fronteira(s) da(s) aplicação(ões)
5	Identificar as partições, caso aplicável
6	Documentar o propósito, tipo, escopo, fronteira, partição e suposições

Veja o Capítulo 4 para mais detalhes sobre o propósito, escopo e fronteira da avaliação.

Esta seção define o tipo de avaliação, escopo da avaliação, fronteiras e partições.

### Tipo da Avaliação

O tamanho funcional e o tamanho não-funcional podem ser medidos tanto para projetos quanto para aplicações. O tipo da avaliação é determinado baseando-se no propósito, como um dos seguintes:

- Avaliação de projeto de Desenvolvimento;
- Avaliação de projeto de melhoria;
- Avaliação de aplicação.

O capítulo 4 da Parte 1 define cada tipo de avaliação.

### Fronteira

A fronteira é uma interface conceitual entre o software em estudo e seus usuários.

A fronteira (também conhecida como fronteira da aplicação):

- Define o que é externo à aplicação;

- Indica a fronteira entre o software que está sendo medido e o usuário;
- Atua como uma “membrana” através da qual os dados processados pelas transações passam para dentro e para fora da aplicação;
- Depende da visão externa do negócio do usuário da aplicação. É independente de considerações não-funcionais e/ou de implementação.

## Partição

Uma partição é um conjunto de funções de software dentro da fronteira da aplicação que compartilham critérios e valores de avaliação homogêneos. Uma partição requer esforço de desenvolvimento, que pode não ser refletido ao medir o aspecto funcional do projeto/produto utilizando a APF.

O posicionamento da partição pode ser subjetivo. É comum haver dificuldade para estabelecer onde uma partição termina e a outra começa. Tente posicionar a partição a partir de uma perspectiva não-funcional dos usuários, tal como manutenibilidade, portabilidade ou instalabilidade, ao invés de se basear em considerações técnicas ou físicas. É importante que a partição seja posicionada cuidadosamente, pois todos os dados cruzando a partição impactam no tamanho SNAP.

- A partição é determinada com base na visão do usuário. O foco é no que o usuário pode entender e descrever.

Dentro de uma fronteira, as partições:

- Contém todas as funcionalidades de software que constituem a aplicação sendo avaliada;
- Podem cooperar com o objetivo de fornecer funcionalidades de software completas ao usuário da aplicação;
- Não devem se sobrepor;
- Devem ser consistentes ao longo do tempo.

Uma partição:

- Pode ser usada para satisfazer requisitos não-funcionais;
- Pode ser medida utilizando as categorias e subcategorias do SNAP;
- Pode coincidir com a aplicação inteira (como: cliente e servidor residem em um único sistema).

## Exemplos de partições

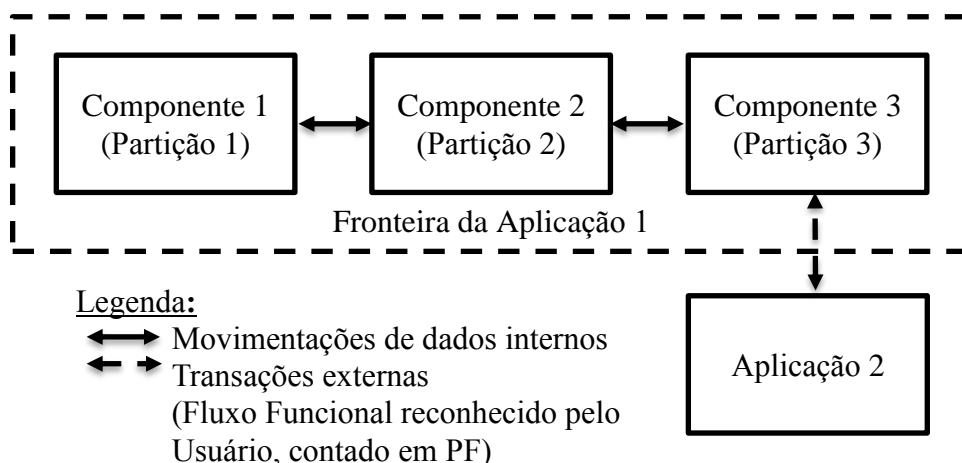
Exemplos de partições são:

- A aplicação (uma única partição);
- As funções cliente em uma aplicação cliente-servidor;
- As funções servidor em uma aplicação cliente-servidor;
- As funções batch em uma aplicação de software mista online/batch;
- As funções do “usuário A”, (a serem) projetadas e/ou implementadas independentemente das funções do “usuário B”, dentro de uma mesma aplicação;
- As funções (a serem) implantadas em uma plataforma não-funcional “X”, identificadas separadamente das funções (a serem) implementadas em uma plataforma não-funcional “Y”, dentro de uma mesma aplicação;
- Aplicação - SOA dentro da fronteira;
- Componente par dentro da fronteira.

## Nota Geral

A contagem de pontos SNAP é realizada no nível de fronteira. As partições

incrementam o tamanho SNAP no que se refere a movimentações de dados internos (Ver Figura 1-3)



*Figura 1-3 Relações entre partições e aplicações*

## Seção 2: Associar os Requisitos Não-Funcionais às Categorias e Subcategorias

Esta seção define as categorias e subcategorias e descreve a associação de requisitos não-funcionais às subcategorias. As categorias e subcategorias são padronizadas para qualquer avaliação SNAP.

- Definição de Categoria** Uma categoria é um grupo de componentes, processos ou atividades que são usados com a finalidade de satisfazer o requisito não-funcional.
- As categorias classificam os requisitos não-funcionais;
  - As categorias são suficientemente genéricas para acomodar tecnologias futuras;
  - As categorias são divididas em subcategorias. Cada subcategoria possui características comuns (dentro da subcategoria). Isto simplifica a avaliação.

Cada categoria SNAP agrupa as subcategorias com base em um mesmo nível de operações e/ou tipos similares de atividades executadas pelo processo de avaliação não-funcional.

**Definição de Subcategoria**

Uma subcategoria é definida como um componente, um processo ou uma atividade executada no projeto para satisfazer o requisito não-funcional.

Nota: Um processo não-funcional pode ter que executar mais que uma subcategoria para satisfazer o requisito não-funcional.

**Categoria e Subcategoria**

A associação dos requisitos não-funcionais com categorias e subcategorias é realizada como segue:

- Identificar o requisito não-funcional contido no escopo (por exemplo, requisitos de segurança de dados, requisitos de aumento de desempenho);
- Analisar a solução e identificar as subcategorias que são usadas para satisfazer o requisito.

As categorias e subcategorias são:

1. Operações de Dados
  - 1.1. Validações na Entrada de Dados
  - 1.2. Operações Lógicas e Matemáticas
  - 1.3. Formatação de Dados
  - 1.4. Movimentações de Dados Internos
  - 1.5. Entregar valor agregado aos usuários por configuração de dados
2. Projeto de Interface
  - 2.1. Interfaces do Usuário
  - 2.2. Métodos de Ajuda
  - 2.3. Múltiplos Métodos de Entrada
  - 2.4. Múltiplos Métodos de Saída
3. Ambiente Técnico
  - 3.1. Múltiplas Plataformas
  - 3.2. Tecnologia de Banco de Dados
  - 3.3. Processos Batch
4. Arquitetura
  - 4.1. Software baseado em componentes
  - 4.2. Múltiplas Interfaces de Entrada / Saída

As subcategorias abordam todos os requisitos não-funcionais, incluindo requisitos Técnicos e de Qualidade. (Requisitos de qualidade tais como usabilidade ou confiabilidade, conforme definido pela ISO/IEC 9126-1 ou pela ISO/IEC 25010, podem ser abordadas pelas seguintes subcategorias: validações na entrada de dados, formatação de dados e mudanças na Interface do Usuário (IU). Por exemplo, a inclusão de Ajuda On-line é um requisito para melhorar a facilidade de aprendizado e pode ser medida através da subcategoria “Métodos de Ajuda”).

Ver o Capítulo 5 para mais detalhes sobre as categorias e subcategorias.



## Seção 3: Identificar as Unidades de Contagem SNAP (UCSs)

Esta seção define as regras e procedimentos que se aplicam ao identificar as Unidades de Contagem SNAP (UCSs).

As UCSs são únicas para cada subcategoria; são determinadas pela natureza da subcategoria.

A UCS é parte da definição de subcategoria.

A medição é feita separadamente para cada UCS.

Nota: Um requisito pode conter tanto aspectos funcionais quanto não-funcionais. Neste caso, o requisito terá um tamanho funcional, medido em Pontos de Função, e um tamanho SNAP, medido em pontos SNAP.

Utilize a Parte 3, Apêndice B (“Contando Pontos de Função e Pontos SNAP”) para requisitos que envolvam requisitos Funcionais e Não-Funcionais.

Os exemplos na Parte 2 demonstram como as UCSs são utilizadas.

**Definição de Unidade de Contagem SNAP (UCS)** A UCS é um componente ou atividade em que a complexidade e tamanho é avaliada.

A UCS pode ser um componente, um processo ou uma atividade identificada de acordo com a natureza da(s) subcategoria(s).

Uma UCS pode conter características funcionais e não-funcionais. Nestes casos, a medição do processo elementar será realizada para o seu tamanho funcional utilizando a análise de pontos de função, e para o seu tamanho não-funcional utilizando o SNAP.

Ver o Capítulo 5 para mais detalhes sobre a UCS.

## Seção 4: Determinar a Complexidade de cada Unidade de Contagem SNAP (UCS)

Esta seção define como determinar a complexidade e o tamanho de cada UCS dentro da subcategoria.

Responder as questões de avaliação para cada subcategoria.

As questões de avaliação são relacionadas aos parâmetros que afetam a complexidade de uma dada subcategoria.

O grau da avaliação é a resposta para as questões de avaliação.

O nível de complexidade de um grau de avaliação ou o valor dos parâmetros dentro de cada UCS é mapeado para um tamanho.

Ver o Capítulo 5 para mais detalhes sobre a determinação da complexidade.

## Seção 5: Determinar os Pontos SNAP (PS) de cada UCS

Esta seção define como determinar o tamanho de cada subcategoria.

**Definição de Pontos SNAP** Os pontos SNAP (PS) são a soma aritmética do tamanho de todas as UCS identificadas em cada subcategoria.

Uma vez que todos os parâmetros de complexidade tenham sido avaliados, o tamanho de cada UCS é calculado e os PS de todas as UCSs são somados para obter os PS calculados para a subcategoria.

Ver o Capítulo 5 para mais detalhes sobre como calcular os PS de cada UCS.

## Seção 6: Calcular o Tamanho Não-Funcional

Esta seção define como calcular o tamanho do aspecto não-funcional do projeto/produto no escopo.

Os PSs são o tamanho não-funcional final obtido pela combinação de todos os valores das categorias.

Quando mais de uma categoria é identificada, o tamanho não-funcional geral (PS) deverá ser obtido pela combinação dos tamanhos de cada subcategoria contida na fronteira do produto de software que está sendo avaliado.

**Nota:** Quando muitas (mais que uma) subcategorias estão utilizando a mesma definição de UCS, tais como o processo elementar, então devemos responder as questões de avaliação de todas as subcategorias relevantes para este processo elementar.

A contagem será por UCS por subcategoria. Portanto, uma UCS pode gerar PS em mais de uma subcategoria. O Requisito No.1 no estudo de caso é apresentado como exemplo.

Ver o Capítulo 6 para mais detalhes sobre os cálculos e definições das fórmulas.

## Preparação da Avaliação

---

**Introdução** Este capítulo apresenta o conceito do papel do usuário na definição dos requisitos não-funcionais para um projeto ou aplicação. Este capítulo também inclui uma descrição da documentação útil durante o ciclo de vida de uma aplicação.

**Conteúdo** Este capítulo inclui os seguintes tópicos:

Tópico	Página
Quando Realizar a Quando Realizar a Avaliação Não-Funcional	3-2
Documentação Útil do Projeto/Aplicação	3-2
Avaliação Não-Funcional Estimada e Final	3-4

## Quando Realizar a Avaliação Não-Funcional

As avaliações não-funcionais podem ser concluídas em qualquer momento do ciclo de vida de desenvolvimento, para auxiliar na estimativa do projeto, monitorar a alteração de escopo e avaliar os requisitos não-funcionais entregues.

Antes de iniciar uma avaliação não-funcional, determine se está realizando uma aproximação ou medindo o tamanho, e documente quaisquer suposições.

As aproximações permitem realizar suposições sobre categorias não-funcionais desconhecidas e/ou suas complexidades, a fim de determinar um tamanho não-funcional aproximado.

As medições incluem a identificação de todas as subcategorias não-funcionais aplicáveis e suas complexidades, para efetuar uma análise do tamanho não-funcional.

Em um estágio inicial, os requisitos não-funcionais podem não estar totalmente definidos. Apesar das desvantagens, esta avaliação pode ser muito útil para produzir uma estimativa inicial. Abaixo são apresentadas utilizações da avaliação não-funcional para aproximar ou medir o tamanho não-funcional nas várias fases do ciclo de vida de desenvolvimento:

Fase do Ciclo de Vida	PS podem ser aproximados	PS podem ser medidos
<b>Proposta:</b> usuários expressam necessidades e intenções	Sim	Não
<b>Requisitos:</b> desenvolvedores e usuários revisam e concordam quanto às necessidades e intenções do usuário	Sim	Não
<b>Projeto:</b> desenvolvedores podem incluir elementos para implementação	Sim	Sim
<b>Construção</b>	Sim	Sim
<b>Entrega</b>	Sim	Sim
<b>Manutenção</b> ( <u>Adaptativa</u> – modificar o sistema para lidar com alterações no ambiente do software. <u>Perfectiva</u> – implementar requisitos do usuário novos ou alterados, que dizem respeito a melhorias funcionais no software. <u>Preventiva</u> – aperfeiçoar a manutenibilidade ou a confiabilidade do software para prevenir problemas no futuro.)	Sim	Sim
<b>Nota:</b> Não foi assumido nenhum ciclo de vida específico. Se utilizar uma abordagem iterativa, você poderá esperar aproximar/medir os PS diversas vezes durante o ciclo de vida do projeto.		

*Tabela 1-2 Quando Realizar a Avaliação SNAP*

## Documentação Útil do Projeto/Aplicação

Normalmente as atividades do ciclo de vida do desenvolvimento de software incluem o desenvolvimento de um documento de Requisitos Técnicos e /ou um documento de Requisitos Não-Funcionais. Os Requisitos Técnicos podem incluir elementos necessários à implementação, mas não considerados na medição do tamanho funcional (ex.: arquivos temporários, índice, etc.).

Este documento pode possuir uma ou mais das seguintes características:

- Dependência Tecnológica;  
Por exemplo, arquivos físicos variam de acordo com o ambiente de banco de dados.
- Terminologia não familiar aos usuários;  
Por exemplo, desenvolvedores de software podem se referir a arquivos físicos ao invés de grupos lógicos de dados.
- Restrições técnicas;  
Por exemplo, a capacidade computacional (suporte de infraestrutura para apoiar o processamento) atualmente disponível na organização.
- Fronteiras físicas;  
Por exemplo, podem existir requisitos técnicos separados para cliente e para servidor.

Em geral, os itens seguintes são úteis quando conduzida uma avaliação não-funcional:

- Documentos de requisitos;
- Diagramas de entidade-relacionamento;
- Documentos de requisitos técnicos;
- Modelos de objetos;
- Modelos físicos de dados;
- Layouts de arquivos físicos e de banco de dados;
- Acordos de interface com descrições de cargas batch/ arquivos transacionais e interfaces de/para outras aplicações;
- Exemplos de relatórios, telas ou outras interfaces do usuário;
- Demonstração da operação da aplicação;
- Padrões de IU;
- Disponibilidade de um ou mais técnicos especialistas serão muito úteis (para a aplicação sendo avaliada);
- Documentação do projeto do sistema;
- Documento do projeto técnico;
- Diagramas de arquitetura;
- Casos de uso/documentos de requisito funcional.

Nota: A lista acima não é pretende ser completa.

## Avaliação Não-Funcional Estimada e Final

É importante entender que as avaliações não-funcionais iniciais são estimativas dos requisitos não-funcionais a serem entregues. Além disso, à medida que o escopo é esclarecido e os requisitos são desenvolvidos/evoluídos, é normal identificar características não-funcionais adicionais não especificadas nos requisitos originais. Este fenômeno é, algumas vezes, chamado de “scope creep” (crescimento descontrolado dos requisitos). É essencial atualizar o tamanho da aplicação após a conclusão do projeto. Caso o tamanho mude durante o desenvolvimento, o tamanho não-funcional ao final do ciclo de vida deverá refletir com exatidão todas as características não-funcionais entregues ao usuário.

---

## Determinar o Propósito, Escopo, Fronteira e Partição

---

**Introdução** Este capítulo descreve em detalhe os passos na determinação do Escopo e da Fronteira. Explica como são influenciados pelo propósito da avaliação e pelo tipo da avaliação.

**Conteúdo** Este capítulo inclui as seguintes seções:

Tópico	Página
Passos para a Determinação do Escopo e Fronteir	4-2
Identificar o Propósito da Avaliação	4-2
Identificar o Tipo da Avaliação	4-3
O Diagrama dos Tipos de Avaliação	4-4
Identificar o Escopo da Avaliação	4-4
	4-5
Determinar a Fronteira	
Determinar a Partição	4-6
Regras e Proced	4-6

## Passos para a Determinação do Escopo e Fronteira

Ao identificar Escopo e Fronteira, os seguintes passos devem ser executados:

- 1 Identificar o propósito da avaliação
- 2 Identificar o tipo da avaliação, baseado no propósito
- 3 Determinar o escopo da avaliação, baseado no propósito e no tipo de avaliação.
- 4 Determinar a fronteira da(s) aplicação(ões)
- 5 Determinar as partições, se aplicável
- 6 Documentar os seguintes itens:
  - O propósito e tipo da avaliação;
  - O escopo da avaliação;
  - A fronteira da(s) aplicação(ões);
  - A(s) partição(ões) dentro da(s) fronteira(s);
  - Quaisquer suposições relacionadas aos itens acima.

**Consistência com a APF** O propósito, escopo e fronteiras lógicas da aplicação precisam ser consistentes entre os processos de APF e SNAP.

Veja o Apêndice B para a ligação entre a APF e o SNAP.

## Identificar o Propósito da Avaliação

Uma medição do tamanho não-funcional é conduzida para medir o tamanho dos requisitos não-funcionais para o desenvolvimento e entrega de um produto de software. A fim de prover respostas relevantes ao propósito da avaliação, o escopo da avaliação, fronteiras e partições devem ser identificados separadamente.

O propósito:

- Determina o tipo da avaliação não-funcional e o escopo da avaliação requerida para obter o tamanho não-funcional;
- Determina o tamanho do aspecto não-funcional dos produtos avaliados;

Exemplos de propósito são:

- Fornecer o tamanho não-funcional de um projeto de desenvolvimento como uma entrada para o processo de estimativa, a fim de determinar o esforço de desenvolvimento para a primeira versão de uma aplicação;
- Fornecer o tamanho não-funcional de uma base instalada de aplicações, para determinar o custo de suporte por ponto de função;
- Fornecer o tamanho não-funcional entregue por uma melhoria;
- Fornecer o tamanho não-funcional para atividades de manutenção.



## Identificar o Tipo da Avaliação

O tamanho funcional e o tamanho não-funcional podem ser medidos para projetos ou aplicações. O tipo da avaliação é determinado como um dos seguintes, com base no propósito:

- Avaliação de Projeto de Desenvolvimento;
- Avaliação de Projeto de Melhoria;
- Avaliação de Aplicação.

Os parágrafos seguintes definem cada tipo de avaliação.

### **Projeto de Desenvolvimento**

Um **projeto de desenvolvimento** é um projeto para desenvolver e entregar a primeira versão de uma aplicação de software.

DSP – Pontos SNAP de Projeto de Desenvolvimento (“Development Project SNAP Points”)

O **tamanho não-funcional do projeto de desenvolvimento** é uma avaliação dos requisitos não-funcionais fornecidos aos usuários através da primeira versão do software, conforme medidos pela avaliação SNAP de projeto de desenvolvimento por meio da aplicação do Método SNAP.

### **Projeto de Melhoria**

Um **projeto de melhoria** é um projeto para desenvolver e entregar uma melhoria corretiva, preventiva, adaptativa ou perfectiva.

ESP – Pontos SNAP de Projeto de Melhoria (“Enhancement Project SNAP Points”)

O **tamanho não-funcional do projeto de melhoria** é a uma medida das características não-funcionais incluídas, alteradas ou excluídas na conclusão de um projeto de melhoria, conforme medidas pela avaliação SNAP de projeto de melhoria.

### **Aplicação**

Uma **aplicação** é um conjunto coeso de procedimentos automatizados e dados que suportam um objetivo de negócio; consiste de um ou mais componentes, módulos ou subsistemas.

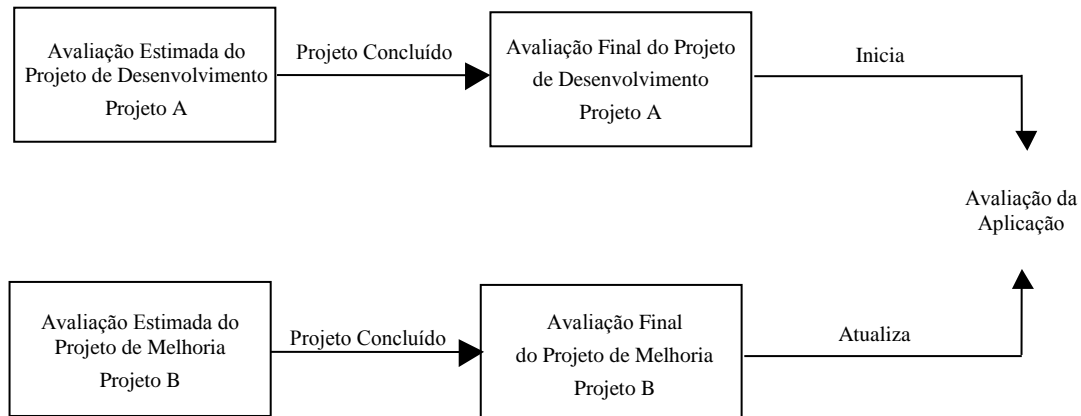
ASPA – Os PS da aplicação após o projeto de melhoria

O **tamanho não-funcional da aplicação** é uma medição das características não-funcionais que uma aplicação provê ao usuário, determinada pela condução da avaliação SNAP de aplicação.

Também é conhecida como baseline ou tamanho não-funcional instalado. Este tamanho fornece uma medida das características não-funcionais que a aplicação atualmente fornece ao usuário. Este número é inicializado quando a avaliação SNAP do projeto de desenvolvimento é concluída. É atualizado a cada conclusão de projeto de melhoria que altere o tamanho não-funcional da aplicação.

## O Diagrama dos Tipos de Avaliação

O diagrama seguinte ilustra os tipos de avaliação SNAP e seus relacionamentos (o Projeto A é concluído primeiro, seguido pelo Projeto B).



O exemplo anterior demonstra o conceito da avaliação não-funcional, mas não reflete o exemplo completo do processo.

## Identificar o Escopo da Avaliação

O escopo da avaliação define o conjunto de requisitos não-funcionais do usuário a serem incluídos na avaliação. O escopo:

- É determinado pelo propósito da realização da avaliação não-funcional;
- Define um conjunto de partições;
- Identifica quais categorias e subcategorias da avaliação não-funcional serão incluídas na medição do tamanho não-funcional, para medir o tamanho dos requisitos não-funcionais para o desenvolvimento e entrega do produto de software;
- Pode conter mais de uma aplicação.

O escopo de:

- Uma avaliação não-funcional de projeto de desenvolvimento inclui todos os requisitos não-funcionais para o desenvolvimento e entrega do produto de software;
- Uma avaliação de uma base instalada de aplicações inclui todos os requisitos não-funcionais para o suporte das aplicações instaladas;
- Uma avaliação não-funcional de melhoria inclui todos os requisitos não-funcionais para o desenvolvimento e entrega do projeto de melhoria; a fronteira da(s) aplicação(ões) impactadas permanece a mesma.
- Uma avaliação de manutenção inclui todos os requisitos não-funcionais para um escopo selecionado.

## Determinar a Fronteira

Determinar a fronteira para cada aplicação contida no escopo da avaliação, com base na visão do usuário.

### Definição de Visão do Usuário

A visão do usuário deve ser definida no intuito de estabelecer a fronteira. As seguintes sugestões podem ajudar a identificar a visão do usuário:

- Um usuário é qualquer pessoa ou “coisa” (aplicação, dispositivo, etc.) que se comunica ou interage com o sistema a qualquer momento;
- A visão do usuário consiste dos requisitos funcionais e não-funcionais conforme percebidos pelo usuário;
- Uma partição pode agir como um “usuário interno” para outra partição dentro da mesma fronteira da aplicação, em termos de troca de dados ou compartilhamento de dados. Consequentemente, diferentes avaliações não-funcionais podem ser criadas para cada partição.

A visão do usuário:

- É uma descrição das funções de negócio e dos requisitos não-funcionais;
- Representa uma descrição formal das necessidades do usuário na linguagem do usuário;
- Podem ser declarações verbais feitas pelo usuário sobre a sua visão;
- É aprovada pelo usuário;
- Pode ser utilizada para medir o tamanho funcional e não-funcional;
- Pode variar na forma física (ex.: catálogo de transações, propostas, documento de requisitos, especificações externas, especificações detalhadas, manual do usuário, especificações não-funcionais ou de qualidade).

### Definição de Fronteira

A fronteira é uma interface conceitual entre o software em estudo e seus usuários.

A fronteira (também conhecida como fronteira da aplicação):

- Define o que é externo à aplicação;
- Indica a fronteira entre o software que está sendo medido e o usuário;
- Atua como uma “membrana” através da qual dados processados por transações passam para dentro e para fora da aplicação;
- Depende da visão externa do negócio do usuário da aplicação. É independente de considerações não-funcionais e/ou implementação.

O posicionamento da fronteira entre o software em análise e outras aplicações de software pode ser subjetivo. É comum haver dificuldade para estabelecer onde uma aplicação termina e a outra se inicia. Tente colocar a fronteira de uma perspectiva de negócio ao invés de se basear em considerações técnicas ou físicas.

Por exemplo, o diagrama a seguir mostra as fronteiras entre a aplicação de Recursos Humanos e a aplicação externa, Sistema Monetário (Câmbio). O exemplo mostra, ainda, a fronteira entre o

usuário humano (Usuário 1) e a aplicação de Recursos Humanos. A aplicação de Recursos Humanos, por sua vez, pode satisfazer internamente os requisitos funcionais, técnicos e de qualidade especificados pelo usuário.

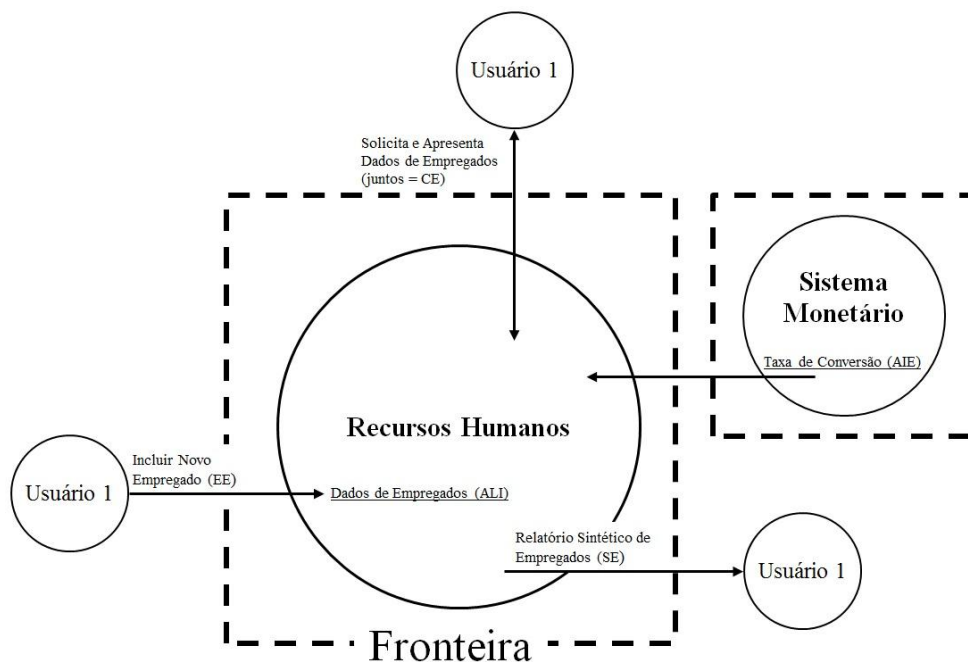


Figura 1-4 – Exemplo de Fronteira

## Determinar a Partição

**Partição** Quando identificadas, as partições podem incrementar o tamanho não-funcional. A subcategoria 1.4 (Movimentações de Dados Internos) é utilizada para fornecer o tamanho não-funcional adicional à aplicação em avaliação.

## Regras e Procedimentos

Esta seção define as regras e procedimentos aplicados ao determinar o escopo da avaliação e a fronteira da(s) aplicação(ões).

### Dicas de escopo

As seguintes dicas podem ajudar a identificação do escopo da avaliação:

- Revisar o propósito da avaliação não-funcional para auxiliar a determinação do escopo da avaliação;
- Ao identificar o escopo para a avaliação do tamanho não-funcional da base instalada de aplicações, incluir todas as categorias não-funcionais suportadas pela equipe de manutenção, eventualmente distinguidas por partição dentro de cada fronteira de aplicação.

### Regras de Fronteiras

As seguintes regras devem ser aplicadas para fronteiras:

- As Fronteiras Lógicas das Aplicações precisam ser consistentes entre os processos da APF e o do SNAP;
- A fronteira é determinada com base na visão do usuário. O foco está no que o usuário consegue entender e descrever;
- A fronteira inicial já estabelecida para a aplicação ou aplicações que estejam sendo modificadas não é influenciada pelo escopo da avaliação.

**Nota:** Pode haver mais de uma aplicação incluída no escopo da avaliação. Caso isso ocorra, múltiplas fronteiras de aplicações deverão ser identificadas. Quando a fronteira não estiver bem definida (como em análises iniciais), a mesma deverá ser localizada com a maior exatidão possível.

**Dicas de Fronteiras e Partições** As seguintes dicas podem ajudar na identificação de fronteiras e partições das aplicações:

- Use as especificações externas ou um fluxograma de sistema e desenhe uma fronteira ao redor da aplicação, para destacar quais partes são internas e quais são externas à aplicação;
- Verifique como os grupos de dados e partições de software estão sendo mantidos;
- Identifique áreas funcionais atribuindo a posse de certos tipos de objetos de análise (como entidades ou processos elementares) a uma área funcional; categorias não-funcionais são determinadas pela identificação das fronteiras funcionais e, eventualmente, pelas partições nelas contidas;
- Observe os dados de medição associados, tais como esforço, custo e defeitos. As fronteiras para a medição devem ser as mesmas ou, eventualmente, estes dados de medição podem ser separados por partições dentro de uma única fronteira;
- Entreviste especialistas no assunto para obter ajuda na identificação da fronteira;
- Entreviste analistas de sistemas para obter ajuda na identificação de partições, se existirem.

**Página deixada em branco intencionalmente**

## Categorias e Subcategorias

---

**Introdução** Este capítulo apresenta os detalhes por trás das categorias e subcategorias, incluindo questões, classificações e UCSs.

**Conteúdo** Este capítulo inclui os seguintes tópicos:

<b>Tópico</b>	<b>Página</b>
Categorias e subcategorias	5-2
Complexidade da Subcatego	5-2
<b>Categoria 1: Operações de Dados</b>	5-5
1.1 Validações na Entrada de Dados	5-5
1.2 Operações Lógicas e Matemáticas	5-6
1.3 Formatação de Dados	5-7
1.4 Movimentações de Dados Internos	5-9
1.5 Entregar Valor Agregado aos Usuários por Configuração de Dados	5-10
<b>Categoria 2: Projeto de Interface</b>	5-12
2.1 Interfaces do Usuário	5-12
2.2 Métodos de Ajuda	5-14
2.3 Múltiplos Métodos de Entrada	5-18
2.4 Múltiplos Métodos de Saída	5-19
<b>Categoria 3: Ambiente Técnico</b>	5-20
3.1 Múltiplas Plataformas	5-20
3.2 Tecnologia de	5-23
3.3 Processos Batch	5-25
<b>Categoria 4: Arquitetura</b>	5-26
4.1 Software Baseado em Componentes	5-26
4.2 Múltiplas Interfaces de Entrada / Saída	5-27
Exemplo do Cálculo dos PS	5-30

## Categorias e subcategorias

**Definição** Veja o [Capítulo 2 Seção 2](#) para a definição de categorias e subcategorias.

## Complexidade da Subcategoria

**Definição de Complexidade** Cada subcategoria é de uma natureza diferente, por isso podem ter um conjunto diferente de parâmetros que definem sua complexidade. A complexidade foi definida pelas seguintes perguntas:

1. Quais são os principais fatores que afetam a complexidade do item, conforme considerado pelos estimadores de projeto de software?
2. Assumindo que uma pequena equipe tenha o seu próprio conjunto de valores de produtividade (conjunto de habilidades, metodologias, ambientes de trabalho, etc.), tal equipe irá estimar como necessário mais trabalho para prover um item complexo do que um item médio.

**Parâmetros de Complexidade** São os Parâmetros contados ou avaliados com o objetivo de avaliar a Complexidade.

No exemplo abaixo (veja a tabela), os parâmetros de complexidade são:

- Número de Dados Elementares;
- Número de níveis de aninhamento.

**Exemplo de Complexidade** A complexidade das Validações na Entrada de Dados pode ser definida pelo número de níveis de aninhamento e também pela quantidade de dados elementares usados no processo. Neste caso, podemos decidir que o Parâmetro No. 1 será baseado no número de níveis de aninhamento e o Parâmetro No. 2 será o número da dados elementares.

**Grade de Complexidade** Utilize as seguintes tabelas para ilustrar o cálculo dos PS.

	Complexidade do Nível de Aninhamento		
	Baixa	Média	Alta
	1-5	6-14	15+
<b>PS=</b>	2* #DERs	3*#DERs	4*#DERs

*Tabela 1-3 Exemplo 1 do cálculo de PS*

**Outra opção para ilustrar os cálculos dos PS baseia-se no tipo de item avaliado, por exemplo:**

Tipo de Ajuda	PS =
Manual do Usuário	1*(#itens de ajuda)
Texto On-line	2*(#itens de ajuda)
Ajuda de Contexto	2*(#itens de ajuda)
Contexto + On-line	3*(#itens de ajuda)

*Tabela 1-4 Exemplo 2 do cálculo de PS*



## Definições de termos adicionais usados neste capítulo

<b>Processo Elementar (PE)</b>	Um processo elementar é a menor unidade de atividade que é significativa para o usuário. O processo elementar deve ser autocontido e deixar o negócio da aplicação sendo contada em um estado consistente.		
<b>Tipo de Dado Elementar (DER)</b>	Atributo único, reconhecido pelo usuário e não repetido.		
<b># DERs</b>	A Soma de todos os dados elementares de negócio distintos que são parte da entrada + saída do processo elementar, mais os dados elementares que são lidos ou atualizados dentro da fronteira.		
<b>Tipo de Dado Elementar Impactado</b>	Um dado elementar que é impactado por um requisito não-funcional.		
<b>Registro Lógico Referenciado (RLR)</b>	Subgrupo de dados elementares referenciados reconhecido pelo usuário dentro de uma função de dados.		
<b>Arquivo Lógico</b>	<p>Uma função de dados representa a funcionalidade fornecida ao usuário para satisfazer seus requisitos de armazenamento de dados internos e externos. Uma função de dados é um arquivo lógico interno ou um arquivo de interface externa.</p> <p>Um arquivo lógico é um grupo lógico de dados como visto pelo usuário. Um arquivo lógico é composto de uma ou mais entidades de dados.</p> <p>O agrupamento de dados dentro de arquivos lógicos é o resultado do efeito combinado de dois métodos de agrupamento:</p> <ul style="list-style-type: none"> <li>• O método a) é dirigido por processos, baseado nas transações do usuário na aplicação;</li> <li>• O método b) é dirigido por dados, baseado nas regras de negócio.</li> </ul>		
<b>Arquivo Lógico Referenciado (ALR)</b>	<p>Um arquivo lógico referenciado é uma função de dados lida e/ou mantida por uma função de transação. Um arquivo lógico referenciado inclui:</p> <ul style="list-style-type: none"> <li>• Um arquivo lógico interno lido ou mantido por uma função de transação ou;</li> <li>• Um arquivo de interface externa lido por uma função de transação.</li> </ul>		
<b>Fator Constante</b>	<p>Um Multiplicador utilizado para calcular o número de pontos SNAP. O tamanho em pontos SNAP é um resultado do [fator constante] vezes [um parâmetro de complexidade].</p> <p>Exemplo:</p> <table border="1" data-bbox="539 1854 1010 1910"> <tr> <td>PS=</td> <td>2*#DERs</td> </tr> </table> <p>O 2 é o fator constante. O #DERs é o parâmetro de complexidade.</p>	PS=	2*#DERs
PS=	2*#DERs		

**Dados de Código**

O usuário nem sempre especifica diretamente os Dados de Código, algumas vezes referenciados como Lista de Dados ou Dados de Tradução. Em outros casos, são identificados pelo desenvolvedor em resposta a um ou mais requisitos não-funcionais do usuário.

Os Dados de Código fornecem uma lista de valores válidos que um atributo descritivo pode ter. Normalmente os atributos de Dados de Código são Código, Descrição e/ou outros atributos “padrão” que descrevem o código; ex.: abreviação padrão, datas de início e término de vigência, dados de trilhas de auditoria, etc.

Ao utilizar códigos nos Dados de Negócio, é necessário que haja um meio de converter o código em algo mais facilmente reconhecido pelo usuário. Os desenvolvedores criam, muitas vezes, uma ou mais tabelas contendo os Dados de Código, com o objetivo de satisfazer os requisitos não-funcionais do usuário. Logicamente, o código e sua respectiva descrição possuem o mesmo significado. Sem uma descrição, o código nem sempre poderá ser claramente entendido.

**Abordagem de Única / Múltiplas Instâncias**

Diferentes organizações podem utilizar diferentes abordagens para medir funcionalidades semelhantes sendo entregues em diferentes meios. As mesmas utilizam a abordagem de uma única instância ou a de múltiplas instâncias na medição. Na abordagem de única instância, a mesma funcionalidade é entregue em diferentes meios (entrada ou saída), mas é contada uma única vez.

Na abordagem de múltiplas instâncias, cada método de entrega da mesma funcionalidade é contado separadamente.

As organizações que utilizam a abordagem de única instância para o tamanho em PF podem dimensionar os outros métodos de entrega utilizando o SNAP.

**Abordagem de Única Instância**

A abordagem de única instância não reconhece o meio de entrega de uma Função de Transação como uma característica diferenciadora na identificação de Funções de Transação distintas. Se duas funções entregarem a mesma funcionalidade utilizando diferentes meios, estas serão consideradas como uma única função para o propósito de medição funcional.

**Abordagem de Múltiplas Instâncias**

A abordagem de múltiplas instâncias especifica que o tamanho funcional da Instância é levado em conta no contexto do objetivo da contagem, possibilitando que uma funcionalidade de negócio possa ser reconhecida no contexto do meio no qual deve operar.

A abordagem de múltiplas instâncias reconhece o meio de entrega de uma Função de Transação como uma característica diferenciadora na identificação de Funções de Transação distintas.

## Categoria 1: Operações de Dados

**Operações de Dados** A Categoria Operações de Dados relaciona-se a como os dados são processados dentro da UCS para satisfazer os requisitos não-funcionais da Aplicação.

### 1.1 Validações na Entrada de Dados

**Definição** Operações que são realizadas para permitir somente dados autorizados (predefinidos) ou para prevenir a aceitação de dados não autorizados.

**UCS** O processo elementar.

**Termos** **Nível de Aninhamento:**  
O número de validações condicionais (conjuntos IF-Else /While loop/For loop ou quaisquer outros blocos de validação) na cadeia de validação mais longa.

**Parâmetros de Complexidade:**

1. Complexidade de nível de Aninhamento:
  - a. Complexidade baixa: 2 níveis de aninhamento ou menos;
  - b. Complexidade média: mais de 3 níveis de aninhamento e menos de ou igual a 5;
  - c. Complexidade alta: Mais de 6 níveis de aninhamento.
2. Número de Dados Elementares Referenciados (DER) utilizados para a validação. (O número máximo de DERs e não a soma de DERs passando pelos diferentes níveis de aninhamento)

**Cálculo do PS** Identifique a complexidade baseada no nível de aninhamento. Calcule os PS com base no fator constante e no número de DERs (#DERs).

	Complexidade do Nível de Aninhamento		
	Baixa	Média	Alta
	1-2	3-5	6+
<b>PS=</b>	$2 * \#DERs$	$3 * \#DERs$	$4 * \#DERs$

*Tabela 1-5 Medição SNAP para Validações na Entrada de Dados*

**Notas:**

- Esta subcategoria pode incluir requisitos para tratamento de erros ou tratamento de exceções;
- Dados Elementares referem-se a dados de código ou dados na base de dados.

**Exemplos** **Incluindo um empregado em uma organização**

No processo lógico de incluir um empregado na organização, o número de validações executadas dentro do processo elementar e o nível de complexidade das validações não são contados no tamanho em PF, pois são considerados requisitos técnicos. Contudo, devem ser contados utilizando o

SNAP.

Conforme o Manual de Práticas de Contagem de Pontos de Função, algumas operações realizadas no processo elementar para garantir dados válidos são incluídas nas informações sobre dados. Os dados elementares nestas operações não são visíveis para o usuário (embora acordados com o usuário) e não são contados como DERs nos PFs. Os dados elementares envolvidos nestas operações ocultas podem ser contados utilizando o modelo SNAP.

## 1.2 Operações Lógicas e Matemáticas

**Definição** Decisões lógicas extensivas, operações booleanas e operações matemáticas extensivas aplicadas ao processo.

**UCS** O processo elementar.

**Parâmetros de Complexidade:**

- 1) Complexidade de ALR dos arquivos lógicos acessados para realizar o processamento lógico de negócio - tabela 1-6.
- 2) Tipo de processamento lógico do Processo Elementar (lógico / matemático) – tabela 1-7.
- 3) Número de dados elementares referenciados (#DERs) – tabela 1-8.

a) O Fator de Complexidade de ALR é medido como segue:

	Nível de Complexidade de ALR		
	0-3 ALR	4-9 ALR	10+ ALR
	Baixa	Média	Alta

*Tabela 1-6 Complexidade de ALR, operações Lógicas e Matemáticas*

b) Tipo do Processo Elementar (lógico / matemático)

Identificar o tipo do Processo Elementar.

Tipo do PE	Intenção Primária do PE
Lógico	Tomar uma decisão ou validar uma condição utilizando dados que existem em um ou mais arquivos lógicos (internos e / ou externos). Exemplo: Processamento de Exceção
Matemático	Transformação de dados e / ou uso de informação de controle que existe em um ou mais arquivos lógicos (internos e / ou externos) utilizado para uma operação matemática extensiva. Exemplo: Cálculo complexo de impostos

*Tabela 1-7 Tipo do PE para operações Lógicas e Matemáticas*

**Cálculo do 'PS** Calcular os PSs baseados no fator constante e no fator de complexidade de ALR.

	Nível de Complexidade		
	Baixa	Média	Alta
	PS=	PS=	PS=
<b>Tipo do PE: Lógico</b>	4*#DERs	6*#DERs	10*#DERs
<b>Tipo do PE: Matemático</b>	3*#DERs	4*#DERs	7*#DERs

*Tabela 1-8 Medição SNAP para operações Lógicas e Matemáticas*

## Notas

- O propósito deste tratamento SNAP para operações matemáticas é distinguir operações matemáticas relativamente simples ou rotineiras que resultam em uma Saída Externa com pontos de função contáveis de operações matemáticas extensas que também resultam em uma Saída Externa, porém, necessitando de um esforço intenso de trabalho algorítmico que não seria reconhecido utilizando pontos de função.
- Os DERs contados são o conjunto necessário à execução de operações matemáticas extensas, tais como valores para variáveis de algoritmos e configurações mantidas por informações de controle de algoritmos. Estes valores e configurações não estão necessariamente, armazenados em um único arquivo físico; podem estar armazenados em diversas localizações, tais como configurações de valores de variáveis localizadas no código ou como DERs em diversos arquivos físicos. Independentemente de como estejam localizados, em conjunto os mesmos satisfazem os requisitos para um ou mais arquivos lógicos internos ou arquivos de interface externa, pois são os agrupamentos lógicos de dados necessários para operar o algoritmo.

## Exemplos

- Análise de caminho crítico de cronogramas de projetos;
- Cálculos complexos de impostos;
- Algoritmos de programação linear;
- Fórmulas de integração em cálculo integral;
- Cálculo de retorno financeiro de investimentos para uma grande máquina industrial;
- Cálculos de Análise Estatística de Variância;
- Projeção de vendas de negócio utilizando o método de projeção conjunta.

## 1.3 Formatação de Dados

### Definição

Qualquer alteração em uma transação que lida com estrutura, formato ou informações administrativas não diretamente relevantes para a funcionalidade que é vista pelo usuário.

### UCS

O processo elementar.

**Parâmetros de Complexidade:**

1. Complexidade da Transformação:
  - a. Baixa: Conversões de tipos de dados ou formatação simples, tais como preenchimento de bytes ou conversão usando no máximo 2 operadores (Celsius para Fahrenheit, Single Integer para Double Integer);
  - b. Média: Aquelas entre Simples e Alta;
  - c. Alta: Envolve Criptografia/Descriptografia em qualquer tipo (caso criptografar/descriptografar seja uma característica da aplicação e se aplique a quase todos os processos, sendo provida através de uma interface de biblioteca API, considerar como “Simples Transformação”).
2. Número de dados elementares referenciados (#DERs) transformados.

**Cálculo do PS**

Identificar a complexidade com base na transformação. Calcular os PSs com base no fator constante e no número de DERs (#DERs).

	Complexidade da Transformação		
	Baixa	Média	Alta
<b>PS =</b>	$2 * \#DERs$	$3 * \#DERs$	$5 * \#DERs$

*Tabela 1-9 Medição SNAP para Formatação de Dados*

**Notas:**

Podem estar inclusos a aderência a padrões, layouts ou mensagens.

**Exemplo**

Uma aplicação apresenta a data (DDMMMAAAA), hora (GMT), temperatura atmosférica atual (Graus Celsius) em um formato padrão. No entanto, devido aos padrões em vigor, a data deve ser apresentada como “AAAAMMDD”, a hora deve sempre apresentar o fuso horário local e a temperatura deve ser apresentada em Graus Fahrenheit.

Como resultado, os formatos apresentados devem ser convertidos para aderir aos padrões prescritos.

## 1.4 Movimentações de Dados Internos

**Definição** Processos de Movimentações de Dados de uma partição para outra dentro da fronteira da aplicação com manipulação específica de dados. A manipulação de dados pode incluir formatação de dados, operações lógicas/matemáticas ou Manutenção de Dados de Referência.

**UCS** O processo elementar dentro da fronteira da aplicação que cruza uma partição. Caso um processo elementar cruze mais que uma partição, utilize a formula abaixo para cada partição atravessada (na figura 1-3, um processo elementar pode mover-se do componente 1 para o componente 2 e então para o componente 3. Neste caso, os PSs devem ser calculados para cada partição atravessada).

**Parâmetros de Complexidade:**

1. Número de tipos de dados elementares (#DERs) transferidos para dentro e para fora da partição na qual os dados são processados e/ou mantidos.
2. Número de ALRs lidos ou atualizados pelo processo elementar.

**Cálculo do PS** Identifique o nível de complexidade baseado no número de ALRs lidos/atualizados. Calcule os PSs conforme a tabela abaixo.

Identifique o número de fronteiras internas cruzadas e os #DERs transferidos. Calcule os PSs utilizando a equação.

	Nível de Complexidade		
	Baixa (1-3 ALR)	Média (4-9 ALR)	Alta (10+ ALR)
<b>PS =</b>	$4 * \#DERs$	$6 * \#DERs$	$10 * \#DERs$

*Tabela 1-10 Medição SNAP para Movimentações de Dados Internos*

**Notas:** A subcategoria de Movimentações de Dados Internos dimensiona transações que não atravessam a fronteira da aplicação. Estas transações são utilizadas no caso de partições.

Isto pode incluir:

- Backup de dados dentro da fronteira da aplicação, atravessando partições;
- Cópia/movimentações de dados entre tabelas dentro da aplicação, atravessando partições;
- Realinhamento de dados em armazenamento temporário;
- Transações entre aplicações e middleware que não atravessam fronteiras funcionais;
- Soluções SOA (Service Oriented Architecture);
- Movimentações de dados entre camadas que não cruzam fronteiras funcionais.

A subcategoria 3.3 (Processos Batch) aborda jobs batch. Jobs batch

podem ser executados dentro de uma única partição.

## 1.5 Entregar Valor Agregado aos Usuários por Configuração de Dados

**Definição** O valor de negócio único adicional que é fornecido aos usuários pela inclusão, alteração ou exclusão de informações de dados de referência/dados de código da base de dados ou do armazenamento de dados, sem a alteração no código do software ou na estrutura da base de dados.

**UCS** O processo elementar

Notas: A UCS é o processo elementar que consome o valor agregado e não o processo para criar ou modificar a configuração.

Exemplo: Um novo serviço é definido através da inclusão de seus atributos em tabelas de referência. A aplicação é flexível o bastante para fornecer o novo serviço sem alterações de código. Os processos elementares a serem contados podem ser: Incluir o novo serviço; Alterar o novo serviço; Cessar o novo serviço. O processo para incluir os atributos do serviço nas tabelas de referência não deve ser contado.

No caso de os dados configurados impactarem diversos processos elementares, cada processo elementar será contado separadamente.

**Termos** **Atributo**

Um parâmetro independente que possui um significado distinto para o negócio e contém um conjunto de valores diferentes.

**Um registro**

Uma linha em um arquivo lógico.

**Parâmetros de Complexidade:**

1. O número de atributos distintos que são incluídos/alterados/excluídos, envolvidos no processo elementar
2. O número de registros configurados.

A definição de um arquivo lógico é um grupo de dados logicamente relacionados ou informações de controle, reconhecidos pelo usuário.

**Cálculo do PS** Identifique o nível de complexidade baseado nos #Registros. Calcule os PSs baseado no fator constante e no #Atributos.

	Nível de Complexidade		
	Baixa	Média	Alta
	1-10 registros	11-29 registros	30 + registros
<b>PS =</b>	6*#atributos	8*#atributos	12*#atributos

*Tabela 1-11 Medição SNAP para Configuração de Dados*



**Notas:**

Novos serviços, produtos, planos de preços, etc., podem ser adicionados à aplicação pela inclusão ou alteração de dados de referência e não pela criação de código.

A funcionalidade através da Configuração de Dados traz valor agregado ao usuário, e também agrega esforço para configurar e testar a nova funcionalidade.

**Exemplos**

Esta subcategoria mede a funcionalidade que é criada pela inclusão de dados na base de dados.

1. A aplicação requer a concessão de acesso a uma função específica na aplicação. Para satisfazer este requisito, o desenvolvedor não escrevem qualquer código separado; em vez disso, atualiza um arquivo de configuração e associa o usuário ou o conjunto de usuários a algum(ns) arquivo(s) de propriedades. Estas inclusões ou modificações são feitas para satisfazer os requisitos do usuário, os quais afetam a funcionalidade no nível do processo elementar.

O processo para configurar os dados dentro da base de dados é apresentado aos usuários através de outros processos, tais como o de acesso à aplicação. Para prevenir a duplicação de contagens, somente os processos do usuário devem ser contados.

2. A aplicação necessita da configuração de um novo produto (“Produto X” abaixo) ou de um componente que possa ser vendido utilizando a aplicação. O novo produto e seu plano de preços são definidos nos dados de referência. O esforço do projeto pode estar na criação dos dados, em sua migração para arquivos de referência e nos testes das funções da aplicação com os novos dados. A avaliação identifica diversas UCSs aqui.
- Alterar o produto Y para produto X;
  - Fornecer o produto X;
  - Alterar o preço do produto X, etc.

## Categoria 2: Projeto de Interface

**Projeto de Interface** A Categoria Projeto de Interface relaciona-se à experiência do usuário final. Esta categoria avalia o projeto dos processos de IU e os métodos que possibilitam ao usuário interagir com a aplicação.

### 2.1 Interfaces do Usuário

**Definição** Elementos independentes da interface gráfica do usuário, distintos e identificados pelo usuário, incluídos ou configurados na interface do usuário, que não alteram a funcionalidade do sistema, mas afetam suas características não-funcionais (tais como usabilidade, facilidade de aprendizado, atratividade, acessibilidade).

**UCS** Conjunto de telas conforme definido pelo processo elementar.

**Termos** **Elemento de IU**

Um elemento de IU (Elemento da Interface do Usuário) é um elemento estrutural distinto reconhecido pelo usuário, que compõe a interface do usuário. Isto inclui elementos como:

- 1) Janela (que pode ser uma janela do tipo “container”, “filha”, “de texto”, ou “de mensagem”)
- 2) Menus
- 3) Ícones
- 4) Controles:
  1. Ponteiro (ou cursor do mouse);
  2. Caixa de texto;
  3. Botão;
  4. Hyperlink;
  5. Lista Drop-down;
  6. List box (caixa de listagem);
  7. Combo box (caixa de combinação);
  8. Check box (caixa de verificação);
  9. Botão de rádio;
  10. Botão de ciclo;
  11. Datagrid.
- 5) Guias (“tabs”)
- 6) Elementos de interação tais como um cursor

Os controles acima são utilizados para apresentar ou manipular objetos de dados. O aspecto que incrementa a complexidade do projeto, configuração e tempo de teste de uma interface do usuário é a configuração de cada um destes elementos. Requisitos não-funcionais podem envolver alterações nas propriedades destes elementos de UI. Dependendo do tipo do elemento da interface do usuário, diversas propriedades podem ser configuradas para produzir a saída desejada. Por exemplo, um Botão pode ser configurado para aparecer “travado”, “destacado”, “colorido”, ou ser colocado em uma determinada localização na tela.

## Propriedades dos Elementos da IU

Cada Elemento da IU é associado a certas propriedades que definem o comportamento e aparência do Elemento de Interface do Usuário. Por exemplo, uma Janela pode conter propriedades tais como: Cor do Fundo, Borda Ativa, Legenda Ativa, etc.

Um Botão pode ter propriedades tais como: Destaque do Botão, Texto do Botão, Cor do Fundo, etc.

Dicas de Ferramentas (tool tips) podem ter propriedades tais como: Texto Informativo, Informação do Fundo, etc.

A lista de propriedades para muitos outros Elementos de Interface do Usuário pode continuar e os exemplos acima foram obtidos em:

[www.w3.org/TR/CSS2/ui.html](http://www.w3.org/TR/CSS2/ui.html)

## Conjunto dos Elementos da IU

Um conjunto de elementos da IU é a coleção de todos os elementos da IU de um mesmo tipo na UCS.

Exemplo: Todas as caixas de texto em um conjunto de telas (UCS).

### Parâmetros de Complexidade:

1. A soma do número de propriedades distintas configuradas para cada elemento da IU na UCS.
2. O número de elementos distintos da IU impactados.

### Cálculo dos PS

Identifique a complexidade com base no número de propriedades do conjunto de elementos da IU. Calcule os PSs através do produto do fator constante e do número de elementos distintos da IU.

	Complexidade do Tipo de IU		
	Baixa	Média	Alta
	<10 Propriedades incluídas ou configuradas	10 - 15 Propriedades incluídas ou configuradas	16+ Propriedades incluídas ou configuradas
PS =	2* #elementos de IU distintos	3* #elementos de IU distintos	4* #elementos de IU distintos

*Tabela 1-12 Medição SNAP para Interfaces do Usuário*

### Regras

1. Caso o processo para incluir/alterar a IU seja contado em PF, não duplique a contagem na avaliação. No entanto, alterações no conteúdo e aparência de um elemento GUI precisarão ser avaliadas como não-funcionais. Alterações estéticas em telas de IU, páginas estáticas ou dinâmicas de IU, reorganização de tela e relatórios impressos devem ser avaliados sob a subcategoria Interfaces do Usuário.
2. O conjunto de telas dentro de um processo será contado como a UCS.

**Notas:** Elementos de IU incluídos podem assumir qualquer forma, tais como: texto, som, imagem, logos, cor, chaves, controles, navegação, animação ou habilitar/desabilitar os anteriores (quando habilitar/desabilitar não seja funcional).

Operações incluídas que suportem: Teclas de Função; Auto-preenchimento; Teclas de atalho; Teclas comuns; Navegação no nível de Tela/Página.

## 2.2 Métodos de Ajuda

**Definição** Informações fornecidas aos usuários que explicam como o software fornece a sua funcionalidade, ou outras informações de suporte providas aos usuários.

**UCS** A aplicação sendo avaliada

**Termos** **Item de Ajuda**

Um Item de Ajuda é a menor ajuda ou tópico de informação distinto e reconhecido pelo usuário, que fornece ao usuário informações de suporte ou detalhes sobre uma parte específica do software.

### Ajuda de Contexto

A Ajuda de Contexto refere-se a um recurso de ajuda do programa que muda dependendo de onde o usuário está. É uma ajuda online obtida de um ponto específico no estado do software, fornecendo ajuda para a situação associada a este estado.

A Ajuda de Contexto pode ser implementada utilizando “tooltips”, as quais fornecem uma breve descrição de um controle GUI, ou apresentam um tópico completo do arquivo de ajuda. Outra maneira frequentemente utilizada para acessar a Ajuda de Contexto é através do clique em um botão. Uma forma utiliza um botão por controle, apresentando a ajuda imediatamente. Outra maneira altera a forma do ponteiro para uma interrogação e então, após o usuário clicar em um controle, a ajuda é apresentada.

### Página Web Estática

Uma Página Web Estática é uma página da web que é entregue a todos os usuários exatamente como armazenada, apresentando a mesma informação a todos os usuários, não sendo gerada por uma aplicação.

**Parâmetros de Complexidade:**

1. Tipo de Ajuda
  - a. Manual do Usuário (cópia física / cópia eletrônica / nível de aplicação da ajuda)
  - b. Texto On-line
  - c. Contexto
  - d. Contexto + On-line
2. Número de itens de ajuda impactados

**Cálculo dos PS**

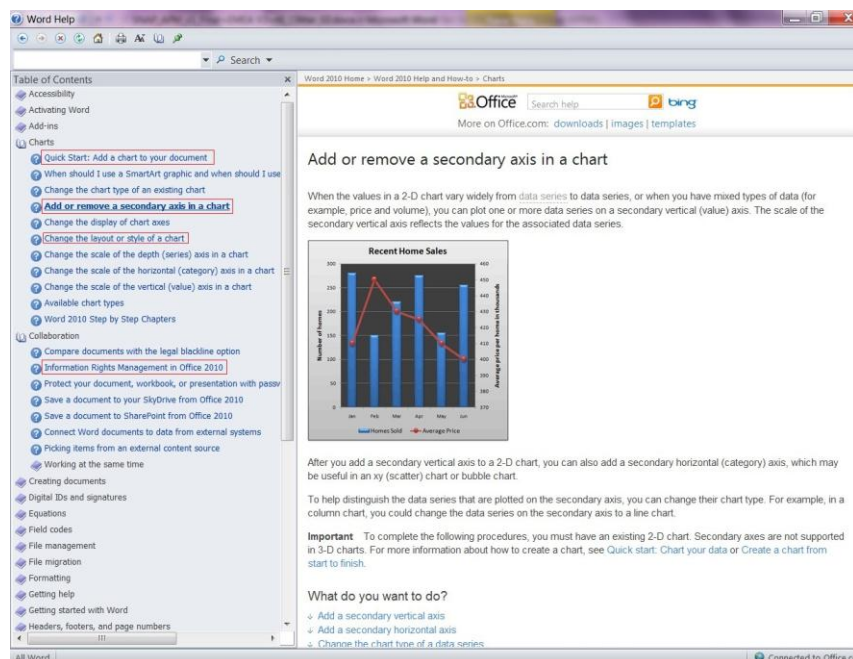
Identifique o tipo de ajuda. Calcule os PS com base no fator constante e no número de itens de ajuda impactados.

Tipo de Ajuda	PS =
Manual do Usuário	1*(#itens de ajuda)
Texto On-line	2*(# itens de ajuda)
Ajuda de Contexto	2*(# itens de ajuda)
Contexto + On-line	3*(# itens de ajuda)

*Tabela 1-13 Dimensionamento SNAP para Métodos de Ajuda*

## Exemplos

1. Um bom exemplo de Item de Ajuda pode ser obtido na Ajuda do Windows. Quando clicamos F1 no MS Word/Excel, uma tela de ajuda é apresentada e podemos ver um Índice à esquerda. Clicando em cada item do Conteúdo de Ajuda, podemos ver o nível de granularidade da ajuda que contém os detalhes do correspondente subtópico de ajuda. Cada um destes tópicos de ajuda, no menor nível de granularidade, pode ser um Item de Ajuda distinto. A respectiva imagem é apresentada abaixo. Os Itens de Ajuda estão destacados em vermelho.



No Windows, a Ajuda Sensível ao Contexto é:

O Menu de Ajuda

O Menu de Ajuda deve conter pelo menos os seguintes itens:

- Um item para abrir a caixa de diálogo dos Tópicos de Ajuda;
- Um item para abrir a ajuda para a tela atual;
- Um item para colocar o programa no modo de ajuda “O que é isto?”

**Notas****Páginas web estáticas**

Utilize a explicação acima para dimensionar páginas web estáticas. Embora uma página estática não seja diretamente um “item de ajuda”, a subcategoria Métodos de Ajuda deve ser utilizada para identificar a sua complexidade e, então, calcular o respectivo tamanho não-funcional.

Embora possam existir aspectos de IU a serem considerados durante a construção ou manutenção de páginas web estáticas, não incremente o respectivo tamanho utilizando a subcategoria 2.1. A intenção primária da subcategoria 2.1 é tratar alterações de GUI que aprimoram a usabilidade, aparência, aprendizibilidade etc. da funcionalidade da aplicação.

**Itens de ajuda que envolvam Interfaces do Usuário**

Há muitos casos em que a inclusão um Item de Ajuda envolve esforço de IU. Nestes casos, não se espera que tal atividade seja medida duas vezes, como PS para a subcategoria de Ajuda e, adicionalmente, PS para Interfaces do Usuário.

Quando a intenção primária da atividade for criar um Item de Ajuda, somente esta subcategoria deve ser utilizada.

## 2.3 Múltiplos Métodos de Entrada

<b>Definição</b>	A capacidade da aplicação de prover funcionalidade aceitando múltiplos métodos de entrada.
<b>UCS</b>	O processo elementar
<b>Termos</b>	<p><b>Método de Entrada</b></p> <p>Um tipo de técnica ou meio que é utilizado para entregar dados dentro da aplicação sendo avaliada, tais como: leitor de Código de Barras, Fax, PDF, documento do Office, tela, mensagem de voz, SMS, dispositivo Smartphone, etc.</p> <p>A aplicação sendo avaliada pode precisar identificar o método de entrada a fim de interpretar e utilizar a informação recebida.</p>
<b>Parâmetros de Complexidade:</b>	<ol style="list-style-type: none"> <li>1. O número de Tipos de Dados Elementares (DERs) na UCS</li> <li>2. O número total de métodos de entrada</li> </ol>
<b>Cálculo dos PS</b>	Identifique a complexidade com base no número de DERs. Calcule os PS com base no fator constante e no número de métodos de entrada.

	Complexidade dos Métodos de Entrada		
	Baixa	Média	Alta
	1-4 DERs	5-15 DERs	16+ DERs
<b>PS=</b>	3*# métodos de entrada	4*# métodos de entrada	6*# métodos de entrada

*Tabela 1-14 Medição SNAP para Múltiplos Métodos de Entrada*

<b>Regras</b>	<p>Esta subcategoria deverá ser utilizada quanto houver múltiplos tipos de entrada para invocar a mesma funcionalidade. Caso os diferentes tipos de entrada difiram em termos de DERs, ALRs e processamento lógico, então já terão sido contados como funções separadas no processo de contagem de PF. Se forem os mesmos, então Múltiplos Métodos de Entrada deverá ser usada. Verificar o seguinte:</p> <ol style="list-style-type: none"> <li>1) A abordagem utilizada na contagem de PF – única ou múltipla instância;</li> <li>2) Que os múltiplos métodos de entrada para a mesma funcionalidade (mesmos DERs, ALRs e processamento lógico) não tenham sido incluídos no cálculo de PF. Em outras palavras, caso a contagem de PF tenha sido realizada utilizando a abordagem de única instância para diferentes tipos de mídias, então o método de entrada adicional da mesma entrada de dados precisará ser contado utilizando o SNAP. Por exemplo, a mesma entrada poderá ser fornecida via smartphone ou via tela web.</li> <li>3) Caso os múltiplos métodos de entrada já tenham sido utilizados na contagem de PF ou a abordagem de múltiplas instâncias tenha sido utilizada na contagem, então aqueles deverão ser excluídos da avaliação SNAP.</li> </ol>
---------------	--



## 2.4 Múltiplos Métodos de Saída

<b>Definição</b>	A capacidade da aplicação de prover funcionalidade utilizando múltiplos métodos de saída.
<b>UCS</b>	O processo elementar
<b>Termos</b>	<p><b>Método de Saída</b></p> <p>Uma técnica ou tipo de meio utilizado para entregar dados da aplicação sendo avaliada, tais como: FAX, PDF, documento do Office, tela, mensagem de voz, SMS, etc.</p> <p>A aplicação sendo avaliada poderá precisar manipular a informação enviada, com o objetivo de enviá-la a várias saídas.</p>
<b>Parâmetros de Complexidade:</b>	<ol style="list-style-type: none"> <li>1. O número de Tipos de Dados Elementares (DERs) na UCS</li> <li>2. O número total de métodos de saída</li> </ol>
<b>Cálculo dos PS</b>	Identifique a complexidade com base no número de DERs. Calcule os PS com base no fator constante e no número de métodos de saída.

	Complexidade dos Métodos de Saída		
	Baixa	Média	Alta
	1-5 DERs	6-19 DERs	20+ DERs
<b>PS =</b>	3*# métodos de saída	4*# métodos de saída	6*# métodos de saída

*Tabela 1-15 Medição SNAP para Múltiplos Métodos de Saída*

<b>Regras</b>	<p>Esta subcategoria deverá ser utilizada quanto houver múltiplos tipos de saída usados para a mesma funcionalidade. Caso os diferentes tipos de saída difiram em termos de DERs, ALRs e processamento lógico, então já terão sido contados como funções distintas no processo de contagem de PF.</p> <p>Se forem os mesmos, então Múltiplos Métodos de Saída deverá ser usada. Verificar o seguinte:</p> <ol style="list-style-type: none"> <li>1) A abordagem utilizada na contagem de PF – única ou múltipla instância;</li> <li>2) Que os múltiplos métodos de saída para a mesma funcionalidade (mesmos DERs, ALRs e processamento lógico) não tenham sido incluídos no cálculo de PF. Em outras palavras, caso a contagem de PF tenha sido realizada utilizando a abordagem de única instância para diferentes tipos de mídias, então o método de saída adicional da mesma saída de dados precisará ser contado utilizando o SNAP.</li> </ol> <p>Por exemplo, a mesma saída poderá ser fornecida via smartphone ou via tela web.</p> <ol style="list-style-type: none"> <li>3) Caso os múltiplos métodos de saída já tenham sido utilizados na contagem de PF ou a abordagem de múltiplas instâncias tenha sido utilizada na contagem, então aqueles deverão ser excluídos da avaliação SNAP.</li> </ol>
---------------	--

## Categoria 3: Ambiente Técnico

### Ambiente Técnico

A Categoria Ambiente Técnico relaciona-se com os aspectos do ambiente onde a aplicação reside. Considera tanto a tecnologia quanto as alterações nos dados internos e configurações que não resultam em funcionalidades incluídas ou alteradas sob a perspectiva de Pontos de Função.

## 3.1 Múltiplas Plataformas

### Definição

Operações fornecidas para suportar a capacidade do software de funcionar em mais de uma plataforma.

**Plataforma de Computação** inclui uma arquitetura de hardware e um framework de software (incluindo frameworks de aplicação), onde sua combinação permite que o software funcione - especialmente aplicações de software. As plataformas comuns incluem uma arquitetura de computadores, sistemas operacional, linguagens de programação e interfaces de usuário relacionadas (bibliotecas run-time do sistema ou interface gráfica do usuário).

**Plataformas de Software:** A Plataforma de Software é um framework utilizado para o desenvolvimento de aplicações de software. Diferentes linguagens de programação podem ser agrupadas em diversas plataformas com base nas famílias de linguagens de programação.

Uma linguagem de programação pode pertencer a uma família de linguagens de programação específica, tal como: Orientada a Objetos, Processual, Declarativa, etc.

**Orientada a Objetos:** Java, C++, C#, Javascript, Python, Smalltalk, VB, VB.NET, etc.;

**Processual:** C, Fortran, PHP, COBOL, etc.;

**Declarativa:** SQL, XQuery, BPEL, XSLT, XML, etc.

**Plataformas de Hardware:**

A ser definido (ASD)

### UCS

O processo elementar

### Parâmetros de Complexidade:

1. Natureza das plataformas (isto é, software, hardware)
2. Número de plataformas para se operar

## Cálculo dos PS

Identifique as diferentes plataformas de software e hardware envolvidas e o número de plataformas para se operar. Calcule os PS com base na linha correspondente à categoria da plataforma na tabela abaixo e na coluna correspondente ao número de plataformas. Caso mais de uma linha seja aplicável, então os PS serão a soma dos fatores constantes obtidos de cada categoria aplicável. Por exemplo, caso uma aplicação seja construída em JAVA e COBOL e requeira o suporte a múltiplos browsers (mais do que 3), então o PS deve ser:

= Soma dos PS para (Categoria 2 na tabela abaixo para 2 plataformas + PS para a Categoria 3)

= 40+10 = 50 (*Nota – exemplo corrigido conforme a versão 2.1 do APM*)

	PS =		
	2 plataformas	3 plataformas	4+ plataformas
Categoria 1: Plataformas de Software: Mesma Família de Software	20	30	40
Categoria 2: Plataformas de Software: Famílias Diferentes	40	60	80
Categoria 3: Plataformas de Software: Browsers Diferentes	10	20	30
Categoria 4: Plataformas de Hardware: Sistemas Real time	ASD*	ASD*	ASD*
Categoria 5: Plataformas de Hardware: Sistemas não Real time	ASD*	ASD*	ASD*

\*ASD = A Ser Definido

*Tabela 1-16 Medição SNAP para Múltiplas Plataformas*

## Exemplos

- Plataforma de Software: .NET, Java;
- Plataforma de Sistema Operacional: MS Windows, Linux, IBM/Microsoft Operating System 2, Mac OS;
- Plataforma de Hardware: computadores Mainframe, computadores Midrange, processadores RISC, arquitetura de dispositivos Móveis, sistemas Mac;
- Browsers: Internet Explorer, Firefox, Google Chrome, etc.

**Nota**

Atualmente as plataformas consideradas na calibração do modelo são somente plataformas de tipos de software.

**Por favor, notar que esta categoria somente deve ser utilizada se o mesmo conjunto de funcionalidades estiver sendo entregue em múltiplas plataformas.** Este é o caso quando as funcionalidades de negócio são as mesmas, mas precisam ser entregues em dois ambientes diferentes. Por exemplo, algumas funções da aplicação são construídas em JAVA e também em VC++ para satisfazer os requisitos do cliente; neste caso, esta subcategoria pode ser utilizada.

Se o próprio framework da arquitetura consistir de plataformas diferentes para entregar parte da funcionalidade, então esta categoria não deverá ser utilizada. Este é o caso normal onde diferentes componentes técnicos interagem para entregar funções da aplicação. Neste caso não há duplicação de esforço para reconstruir a mesma funcionalidade em ambientes diferentes.

## 3.2 Tecnologia de Banco de Dados

### Definição

Recursos e operações que são inseridos na base de dados ou nas instruções de leitura/gravação de dados da base de dados para entregar requisitos não-funcionais sem afetar a funcionalidade fornecida.

### UCS

O processo elementar

### Termos

#### Alterações na Base de Dados

Cada um dos itens seguintes é considerado como uma alteração.

1. Uma alteração ou criação de tabelas, tal como:
  - a. Inserir tabelas ou colunas somente para propósitos não-funcionais;
  - b. Reorganizar a ordem das colunas em uma tabela;
  - c. Alterar ou inserir relacionamentos utilizando recursos de integridade referencial;
  - d. Alterar a chave primária sem efetuar exclusão seguida de inclusão.
2. Uma alteração de indexação, tal como:
  - a. Alterar as colunas utilizadas para indexação;
  - b. Alterar a especificação de unicidade de um índice;
  - c. Agrupar os dados da tabela através de um índice diferente;
  - d. Alterar a ordem de um índice (ascendente ou descendente).
3. Uma alteração de views e partições, tal como:
  - a. Alterar ou incluir o particionamento da base de dados;
  - b. Incluir, alterar ou excluir uma view.
4. Uma alteração na capacidade da base de dados, tal como:
  - a. Espaço das tabelas;
  - b. Melhorar as características de desempenho.
5. Uma alteração de consulta/inclusão, tal como:
  - a. Alterações em consultas, seleção ou inclusões de dados na base de dados sem incluir, alterar ou excluir funcionalidades.

Por exemplo, Alterar a chave primária e incluir um relacionamento é contado como uma alteração.

**Parâmetros de Complexidade:**

1. Complexidade do Arquivo Lógico
2. O número de alterações relacionadas à base de dados

Alterações na base de dados podem ser realizadas para qualquer requisito não-funcional tal qual desempenho, gerenciamento de capacidade, integridade de dados, etc. A complexidade da implementação de qualquer alteração dependerá da complexidade do Arquivo Lógico e do número de alterações.

## a. Fator de Complexidade do Arquivo Lógico

		DERs		
		1-19	20-50	>50
RLRs	1	Baixa	Baixa	Média
	2-5	Baixa	Média	Alta
	>5	Média	Alta	Alta

*Tabela 1-17 Complexidade do Arquivo Lógico, Tecnologia de Banco de Dados*

**Cálculo dos PS**

Fator de Complexidade do ALR			
	Baixa	Média	Alta
PS =	6* # de alterações	9* # de alterações	12* # de alterações

*Tabela 1-18 Cálculo do PS, Tecnologia de Banco de Dados*

Caso existam múltiplos ALRs impactados pelos requisitos não-funcionais (RNF), todos impactando o mesmo PE, então somente a maior complexidade de ALR deverá ser considerada como o Fator de Complexidade, e não os ALRs individuais separadamente.

**Exemplo**

Um PE “Criar Pedido” é projetado para a melhoria de desempenho. Para conseguir isto, uma view de “somente leitura” é criada no ALR “Cliente”, contendo 18 DERs e 3 RLRs (a complexidade do ALR é “Baixa”), e um índice é criado no ALR “Pedidos Criados”, contendo 30 DERs e 3 RLRs (a complexidade do ALR é “Média”).

Com base na tabela 1-17 para uma “Baixa” e uma “Média” teremos 6 PS + 9 PS, portanto:

PS = 15 (*Nota – exemplo corrigido conforme a versão 2.1 do APM*)

**Nota**

Esta subcategoria refere-se a alterações na base de dados. A opção de realizar alterações em dados de código será investigada em uma versão futura do APM.

### 3.3 Processos Batch

#### Definição

Jobs batch que não são considerados requisitos funcionais (não se qualificam como funções de transação) podem ser considerados no SNAP. Esta subcategoria permite o dimensionamento de processos batch disparados dentro da fronteira da aplicação, não resultando em quaisquer dados atravessando a fronteira.

Os requisitos não-funcionais (RNF) associados aos jobs batch, tais como: melhoria no tempo de processamento, aumento da capacidade de processamento para processar um maior volume de transações ou requisitos de melhoria de desempenho podem ser dimensionados utilizando outras subcategorias aplicáveis do SNAP (3.2, 1.1, ou 1.2).

No entanto, caso um RNF relacionado ao processamento batch não seja coberto por estas subcategorias, o mesmo pode ser considerado na 3.3.

#### UCS

Cada job batch

#### Parâmetros de Complexidade:

1. O número dos DERs processados pelo job
2. O número dos ALRs lidos ou atualizados pelo job

#### Cálculo dos PS

Para cada job calcular os PS como:

Identificar o nível de complexidade com base no número de ALRs lidos/atualizados. Calcular os PS de acordo com a tabela abaixo:

	Nível de Complexidade		
	Baixa (1-3 ALR)	Média (4-9 ALR)	Alta (10+ ALR)
<b>PS =</b>	4*#DERs	6*#DERs	10*#DERs

*Tabela 1-19 Medição SNAP para Processos Batch*

#### Nota

Cargas de dados em tabelas lógicas, especificadas pelo usuário e executadas uma única vez, podem ser contadas utilizando esta subcategoria. Favor notar que tais cargas de dados não devem estar relacionadas à migração de dados, já que este tipo de carga é dimensionado com PFs de Conversão utilizando Pontos de Função.

## Categoria 4: Arquitetura

**Arquitetura** A Categoria Arquitetura refere-se ao projeto e técnicas de codificação utilizados para construir ou melhorar a aplicação. Avalia a complexidade do desenvolvimento modular e/ou baseado em componentes.

### 4.1 Software Baseado em Componentes

**Definição** Pedacos de software utilizados dentro da fronteira da aplicação sendo avaliada, para integrar com software pré-existente ou para construir componentes no sistema.

**UCS** O processo elementar

**Termos** **Um Componente de Software**

1. Um pedaço de software oferecendo um serviço pré-definido, que é capaz de se comunicar com outros componentes via interfaces padronizadas;

Critérios para componentes de software:

2. Execução paralela: Uso múltiplo;
3. Intercambiável: Não específico do contexto;
4. Combinável com outros componentes (pode ser selecionado e construído através de várias combinações para satisfazer os requisitos específicos do usuário)
5. Encapsulado, isto é, não investigável através de suas interfaces;
6. Uma unidade com instalação e versionamento independentes;

Exemplos: Microsoft OLE e COM

**Parâmetros de Complexidade:**

1. Componente de terceiros ou reutilização de software interno
2. Número de componentes distintos envolvidos no processo elementar

**Cálculo dos PS** Calcule os PS com base no fator constante e no número de transações críticas.

Tipo	Cálculo do PS
Componentes internos	$PS = 3 * (\# \text{ de componentes distintos})$
Componentes de terceiros	$PS = 4 * (\# \text{ de componentes distintos})$

*Tabela 1-20 Medição SNAP para Software Baseado em Componentes*

**Exemplo** Microsoft OLE e COM.

**Nota** O reuso de componentes pode ser utilizado para satisfazer requisitos não-funcionais tais como: Manutenibilidade (“A capacidade do produto de software em aderir ao padrões ou convenções relacionadas à manutenibilidade”), Alterabilidade, Maturidade ou Substituibilidade.



## 4.2 Múltiplas Interfaces de Entrada / Saída

**Definição** Aplicações que requeiram suporte a múltiplas interfaces de entrada e saída (arquivos de usuário com o mesmo formato) devido a um crescente número de usuários e volume de dados ao longo de um período de tempo são cobertas nesta subcategoria.

A inclusão de interfaces de entrada/saída sem alterar a funcionalidade não é considerada uma alteração funcional. Dessa forma, tais mudanças não são medidas por PF. Esta subcategoria deve ser utilizada no dimensionamento de tais alterações em uma aplicação.

**Nota** Se o projeto/organização considerar a inclusão de novas interfaces de entrada/ saída como uma alteração funcional, os Pontos de Função (e não o SNAP) deverão ser usados no respectivo dimensionamento.

**UCS** O processo elementar

**Parâmetros de Complexidade:**

1. O número de Tipos de Dados Elementares (DERs) na UCS
2. O número de interfaces de entrada e saída

**Notas** Contar o número de interfaces de entrada e saída adicionais.

Quando uma interface for utilizada para entrada e saída, conte-a uma vez como entrada e uma vez como saída.

**Cálculo dos PS** Identificar a complexidade com base no número de DERs. Calcular os PS com base no número de interfaces de entrada e saída acrescentadas.

	Nível de Complexidade		
	Baixa	Média	Alta
	1-5 DERs	6-20 DERs	20+ DERs
<b>PS</b>	3* (# total de interfaces)	4* (# total de interfaces)	6* (# total de interfaces)

*Tabela 1-21 Medição SNAP para Múltiplas Interfaces de Entrada/Saída*

**Exemplo**

Uma ilustração é fornecida abaixo a título de exemplo:

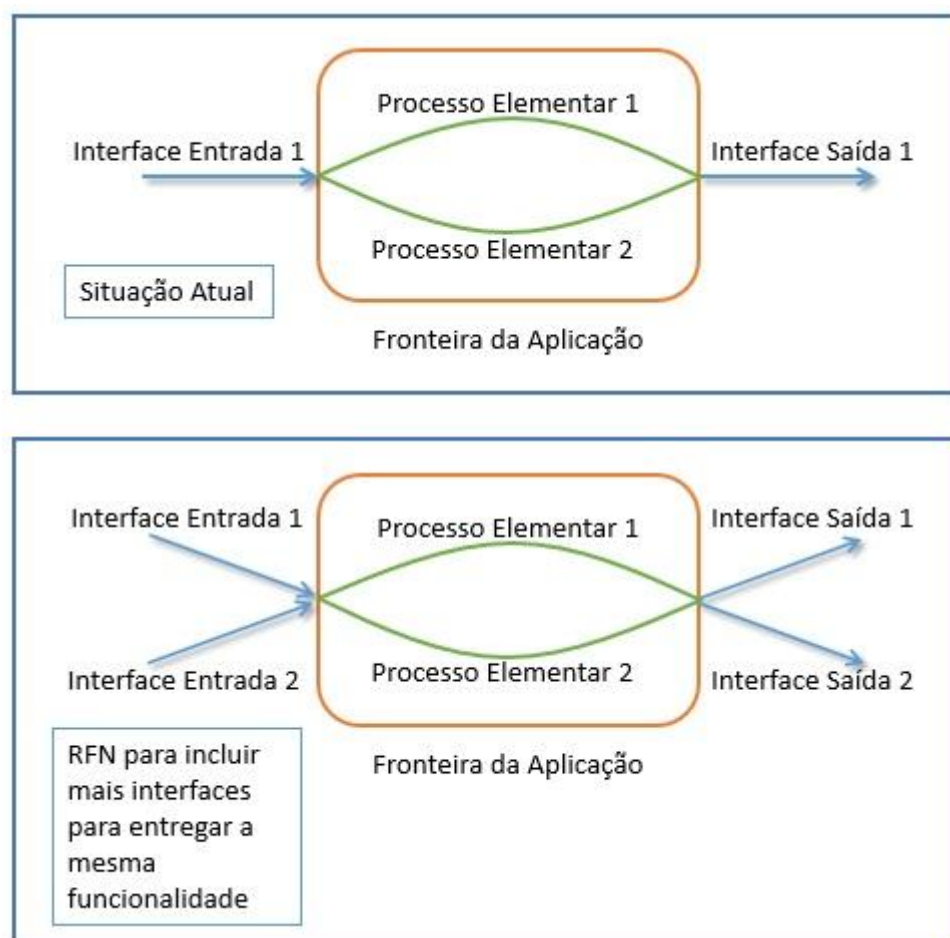


Figura 1-5 – Múltiplas interfaces

## Sistemas de Missão Crítica/Real Time

### Introdução

Alguns sistemas de software podem ser classificados como Real Time ou de Missão Crítica, com base no tempo de resposta, exatidão ou severidade das consequências associadas à sua saída.

### Termos

#### Real Time

Software Real Time é um software que falha se um limite de tempo de resposta não for respeitado.

#### Missão Crítica

Software de Missão Crítica é um software cuja falha pode causar consequências catastróficas (como morte de alguém, danos a propriedades, perdas financeiras severas, etc.).

### Cálculo dos PS

O tempo de resposta, exatidão e características de alto rendimento de tais funções poderiam ser considerados parte do aspecto funcional, pois tais características constituiriam a única base para a classificação de um sistema como Real Time/Missão Crítica.

No entanto, caso o tempo de resposta, exatidão e características de alto rendimento sejam considerados requisitos não-funcionais, seu dimensionamento poderá ser realizado utilizando-se alguma(s) das demais subcategorias.

Exemplos:

A tempestividade pode ser satisfeita por meio de ajuste (‘tuning’) das transações de interação com a base de dados, alterações na base de dados ou a combinação de ambos para melhorar o desempenho. Em tal caso, a subcategoria 3.2 será utilizada para o cálculo do SNAP.

A exatidão pode ser satisfeita pela inclusão novas validações e operações lógicas/matemáticas. As subcategorias 1.1 e 1.2 podem ser utilizadas no cálculo SNAP.

A exigência de rendimento mais alto pode ser atendida dividindo-se as transações existentes para poder processar múltiplas entradas em paralelo. A subcategoria 2.3 pode ser utilizada no cálculo SNAP.

A solução mencionada para os três cenários acima não esgota as possibilidades. Caso alguma outra abordagem seja utilizada para satisfazer os requisitos, a subcategoria apropriada poderá ser aplicada.

Dessa forma, transações e/ou sistemas Real Time ou de Missão Crítica devem ser avaliados utilizando-se as outras subcategorias.

## Exemplo do Cálculo dos PS

A seguinte tabela demonstra como o tamanho SNAP é calculado para a categoria “Projeto de Interface”

A complexidade foi calculada segundo as fórmulas presentes neste capítulo.

		Categoria 1: Operações de Dados															
N.	Descrição/ID da UCS	Subcategoria:	1.1 Validações na Entrada de Dados				Contagem SNAP para Validações na Entrada de Dados	1.2 Operações Lógicas e Matemáticas				Contagem SNAP para Operações Lógicas e Matemáticas	1.3 Formatação de Dados				Contagem SNAP para Formatação de Dados
		Definição da UCS:	UCS = O Processo Elementar Funcional			UCS = O Processo Elementar Funcional				UCS = O Processo Elementar Funcional							
		Número de níveis de aninhamento	Nível de Complexidade	Número de DERs	Fórmula	Tipo do Processo Elementar (Intenção Primária)	Número de DERs	Número de ALRs	Nível de Complexidade	Fórmula	Nível de Complexidade de Transformação (B, M, A)	Número de DERs	Fórmula				
1	Incluir um produto	2	Baixa	18	= 2*#DERs	36	Lógico	4	1	Baixa	= 4*#DERs	16	Baixa	4	= 2*#DERs	8	
2	Alterar um pedido																
3	Detalhar mudança da fatura	4	Baixa	12	= 2*#DERs	24	Matem.	12	2	Baixa	= 3*#DERs	36					
4	Validar endereço	6	Média	9	= 3*#DERs	27											

Categoria 1: Operações de Dados											
N.	Subcategoria:	1.4 Movimentações de Dados Internos				Contagem SNAP para Movimentação de Dados Internos	1.5 Entregar Valor Agregado ao Usuário por Configuração de Dados			Contagem SNAP para Entregar Valor Agregado ao Usuário por Configuração de Dados	Total de OS para Operações de Dados
	Definição da UCS:	UCS = O Processo Elementar Funcional					UCS = O Processo Elementar Funcional				
	Número de ALRs	Nível de Complexidade	Número de DERs	Fórmula	Número de Registros		Nível de Complexidade	Número de Atributos			
	Descrição/ID da UCS										
1	Incluir um produto	3	Baixa	18	= 4*#DERs	72	9	Baixa	7	21	153
2	Alterar um pedido						9	Baixa	7	21	21
3	Detalhar mudança da fatura										60
4	Validar endereço	1	Baixa	4	= 4*#DERs	16					43

Tabela 1-22 Exemplo de Cálculo dos PS

**Página deixada em branco intencionalmente**



---

## Calcular o Tamanho Não-Funcional (Pontos SNAP)

---

**Introdução** Esta seção define como calcular o tamanho não-funcional do projeto/produto em escopo.

**Conteúdo** Este capítulo inclui os seguintes tópicos:

<b>Tópico</b>	<b>Página</b>
Abordagem da Fórmula	6-2
Determinar o tamanho não-funcional de cada subcategoria	6-2
Determinar o tamanho não-funcional de cada categoria	6-2
Determinar o tamanho não-funcional de um projeto de desenvolvimento	6-3
Determinar o tamanho não-funcional de um projeto de melhoria	6-3
Estudo de Caso do Cálculo SNAP	6-6

## Abordagem da Fórmula

Durante uma avaliação de um requisito não-funcional, uma ou mais categorias podem ser avaliadas, dependendo da especificação do requisito. Adicionalmente, uma ou mais subcategorias dentro das categorias podem ser avaliadas, dependendo da especificação do requisito.

Para cada requisito não-funcional é possível determinar o tamanho não-funcional em três passos:

- Passo 1** Para cada uma das subcategorias, identificar as UCS.
- Passo 2** Determinar o tamanho não-funcional (Pontos SNAP – PS) para cada UCS dentro da subcategoria, utilizando a equação ou tabela para a subcategoria.
- Passo 3** Determinar os PS para um projeto ou aplicação específicos, utilizando a fórmula para o tipo de projeto em questão.

## Determinar o tamanho não-funcional de cada subcategoria

O tamanho não-funcional de cada subcategoria será determinado utilizando-se a medida definida para a UCS de cada subcategoria.

Há uma definição de UCS para cada uma das subcategorias. Estes critérios de avaliação são definidos na definição de subcategoria no Capítulo 2.

Os PS para cada subcategoria são determinados utilizando-se a equação ou tabela definida para cada subcategoria.

## Determinar o tamanho não-funcional de cada categoria

Os PS são determinados utilizando-se a equação para cada categoria contra os PS identificados para cada subcategoria da categoria corrente.

**Fórmula para a Categoria Operações de Dados** PS para Operações de Dados =  $[\sum \text{ dos PS para Validações na Entrada de Dados}]$   
 $+ [\sum \text{ dos PS para Operações Lógicas e Matemáticas}]$   
 $+ [\sum \text{ dos PS para Formatação de Dados}]$   
 $+ [\sum \text{ dos PS para Movimentações de Dados Internos}]$   
 $+ [\sum \text{ dos PS para Entregar Valor Agregado aos Usuários por Configuração de Dados}]$

**Fórmula para a Categoria Projeto de Interface** PS para Projeto de Interface =  $[\sum \text{ dos PS para Interfaces do Usuário}]$   
 $+ [\sum \text{ dos PS para Métodos de Ajuda}]$   
 $+ [\sum \text{ dos PS para Múltiplos Métodos de Entrada}]$   
 $+ [\sum \text{ dos PS para Múltiplos Métodos de Saída}]$



<b>Fórmula para a Categoria Ambiente Técnico</b>	$\text{PS para Ambiente Técnico} = [\sum \text{ dos PS para Múltiplas Plataformas}]$ $+ [\sum \text{ dos PS para Tecnologia de Banco de Dados}]$ $+ [\sum \text{ dos PS para Processos Batch}]$
<b>Fórmula para a Categoria Arquitetura</b>	$\text{PS para Arquitetura} = [\sum \text{ dos PS para Software Baseado em Componentes}]$ $+ [\sum \text{ dos PS para Múltiplas Interfaces de Entrada / Saída}]$
<b>Nota</b>	$\sum$ significa “Soma”

## Determinar o tamanho não-funcional de um projeto de desenvolvimento

O tamanho dos requisitos não-funcionais é igual à soma dos tamanhos de cada categoria em PS.

O tamanho não-funcional de um projeto de desenvolvimento deverá ser calculado utilizando-se a fórmula para desenvolvimento.

<b>Fórmula para projeto de desenvolvimento</b>	$\text{DSP} = \text{ADD}$ <p>Onde:</p> <p>DSP é o tamanho em PS do projeto de desenvolvimento;</p> <p>ADD é o tamanho dos requisitos não-funcionais entregues ao usuário pelo projeto de desenvolvimento.</p> $\text{ADD} =$ $[\sum \text{ dos PS para Operações de Dados}] + [\sum \text{ dos PS para Projeto de Interface}] + [\sum \text{ dos PS para Ambiente Técnico}] + [\sum \text{ dos PS para Arquitetura}]$ <p>O tamanho não-funcional da aplicação é equivalente ao tamanho não-funcional do projeto de desenvolvimento.</p>
<b>Nota:</b>	Para o tamanho não-funcional não são identificadas funcionalidades de conversão.

## Determinar o tamanho não-funcional de um projeto de melhoria

Projetos de melhoria podem envolver inclusões, alterações e exclusões de características não-funcionais existentes.

Um projeto de melhoria é um projeto para desenvolver e entregar manutenções. Pode ser adaptativo, preventivo ou perfectivo.

O tamanho não-funcional do projeto de melhoria é uma medida dos requisitos não-funcionais incluídos, alterados ou excluídos ao final do projeto de melhoria, conforme medido pelos PS do projeto de melhoria.

<b>Regras</b>	<p>Os requisitos não-funcionais de melhoria serão medidos segundo:</p> <p>a) Não modificar a fronteira ou partição já estabelecidas para a(s)</p>
---------------	---

- aplicação(ões) sendo modificada(s);
- b) Avaliar os requisitos incluídos, alterados ou excluídos;
- c) O tamanho não-funcional da aplicação poderá ser atualizado para refletir:
- 1) Requisitos não-funcionais incluídos, que aumentam o tamanho não-funcional da aplicação;
  - 2) Requisitos não-funcionais alterados, que podem aumentar, diminuir ou não afetar o tamanho não-funcional da aplicação;
  - 3) Requisitos não-funcionais excluídos, que diminuem o tamanho não-funcional da aplicação.

O tamanho não-funcional de um projeto de melhoria será calculado utilizando a fórmula:

**Fórmula para projeto de melhoria**

$$ESP = ADD + CHGA + DEL$$

Onde:

ESP é o tamanho em PS do projeto de melhoria.

ADD é o tamanho dos requisitos não-funcionais incluídos pelo projeto de melhoria;

ADD =

$[\sum \text{dos PS para Operações de Dados incluídos}] + [\sum \text{dos PS para Projeto de Interface incluídos}] + [\sum \text{dos PS para Ambiente Técnico incluídos}] + [\sum \text{dos PS para Arquitetura incluídos}]$

CHGA é o tamanho dos requisitos não-funcionais alterados pelo projeto de melhoria – como eles ficaram após a implementação;

CHGA =

$[\sum \text{dos PS para Operações de Dados alterados}] + [\sum \text{dos PS para Projeto de Interface alterados}] + [\sum \text{dos PS para Ambiente Técnico alterados}] + [\sum \text{dos PS para Arquitetura alterados}]$

DEL é o tamanho dos requisitos não-funcionais excluídos pelo projeto de melhoria.

DEL =

$[\sum \text{dos PS para Operações de Dados excluídos}] + [\sum \text{dos PS para Projeto de Interface excluídos}] + [\sum \text{dos PS para Ambiente Técnico excluídos}] + [\sum \text{dos PS para Arquitetura excluídos}]$

O tamanho não-funcional da aplicação após um projeto de melhoria será calculado utilizando a fórmula:

**Fórmula para aplicação após projeto de melhoria**

$$ASPA = ASPB + (ADD + CHGA) - (CHGB + DEL)$$

Onde:

ASPA é o tamanho em PS da aplicação após o projeto de melhoria;

ASPB é o tamanho em PS da aplicação anterior ao projeto de melhoria;

ADD é o tamanho dos requisitos não-funcionais incluídos pelo projeto de melhoria;

CHGA é o tamanho dos requisitos não-funcionais alterados pelo projeto de melhoria – como eles ficaram após a implementação;

CHGB é o tamanho dos requisitos não-funcionais alterados pelo projeto de melhoria – como eles eram antes do projeto começar;

DEL é o tamanho dos requisitos não-funcionais excluídos pelo projeto de melhoria.

## Estudo de Caso do Cálculo SNAP

### Escopo do Projeto

“StarTrek” é o codinome para um projeto de melhoria que a equipe “Alpha” foi contratada para entregar. Tratando-se de um projeto relativamente pequeno (em termos de orçamento), StarTrek possui requisitos para melhorar o desempenho da principal aplicação da companhia, “Centra-One”, juntamente com pequenas alterações em algumas de suas funcionalidades existentes.

### Reunião SNAP

As sessões de brainstorming realizadas entre o Arquiteto chefe, o Líder de desenvolvimento e o cliente resultaram no seguinte conjunto de requisitos:

A fim de melhorar o desempenho da transação “atualizar fatura” (média atual do tempo de resposta: 8-12 segundos) e trazer o tempo de resposta para abaixo de 3 segundos, foi adotada uma estratégia com diversas frentes.

### Requisitos relevantes para o SNAP

Os requisitos relevantes para o SNAP são identificados e revisados.

#### Requisito 1:

Criar uma interface adicional para receber transações de “atualizar fatura” – interface de Fax Automatizado (para desviar uma parte das transações de atualização web e diminuir a carga que o servidor estava enfrentando). Um servidor separado foi instalado para manusear os pedidos via fax e foi equipado para ler e decodificar os formulários de fax recebidos que vinham em um formato pré-definido.

#### Requisito 2:

- a) Modificar a lógica de validação da transação online atual “atualizar fatura” para reduzir seu tempo de processamento;
- b) Criar uma fila de “cache” que armazene a lista dos registros de clientes mais acessados, referenciada internamente antes de atualizar a fatura. Foi previsto que isto resultará em melhoria de desempenho. Este cache era atualizado a intervalos regulares;
- c) Realizar algumas alterações cosméticas na tela online “atualizar fatura” e na “tela inicial” para informar aos usuários sobre a nova instalação.

A equipe Alpha queria dimensionar estes requisitos para fornecer uma entrada para o processo de estimativa e, também, para acompanhar o tamanho das alterações ao longo do ciclo de vida e detetar eventual “scope creep”. A equipe decidiu utilizar o SNAP juntamente com os PF do IFPUG, o que resultou na capacidade de dimensionar todos os aspectos funcionais e não-funcionais do projeto.

### Propósito e Escopo

Propósito da medição: Medir o projeto de melhoria e utilizar para a estimativa do projeto e acompanhar o “scope creep”.

Tipo de contagem: Melhoria

Escopo: Requisitos 1 e 2 (a, b e c).

<b>Fronteira</b>	A avaliação SNAP assumirá a mesma fronteira usada pela metodologia de PF do IFPUG. Embora a aplicação Centra-One possua 3 camadas, a fronteira foi fixada no mesmo nível que para a metodologia de FP.
<b>Avaliação</b>	<p>Durante a fase de projeto de alto nível, a equipe Alpha realizou uma análise de impacto e encontrou as seguintes UCS que seriam impactadas. As UCS foram determinadas de acordo com as regras SNAP.</p> <ol style="list-style-type: none"><li>1. Identificar as subcategorias;</li><li>2. Identificar as UCS;</li><li>3. De acordo com as subcategorias, avaliar a complexidade de cada UCS e calcular a pontuação SNAP.</li></ol>
<b>Notas:</b>	<p>Um requisito que inclua aspectos funcionais e não-funcionais pode impactar os Pontos de Função e os Pontos SNAP (ver 2a acima).</p> <p>Um requisito pode impactar mais de um processo elementar e uma UCS pode ser impactada por mais de um requisito (funcional ou não-funcional). Na avaliação deve-se ter cuidado para garantir que as UCS não sejam avaliadas duas vezes no SNAP.</p> <p>No SNAP a definição das UCS poderá ser idêntica, para algumas subcategorias, ao processo elementar tradicional do IFPUG. Em outros casos, as subcategorias serão avaliadas segundo diferentes níveis de unidades de contagem (UCS).</p> <p>Além da contagem de PF do IFPUG, que determinou uma contagem de 6 FP para este projeto, o SNAP o avalia como 200 PS (<math>38 + 34 + 31 + 41 + 24 + 32</math>), com base na análise correspondente à tabela 1-23 a seguir.</p>

A equipe Alpha utilizou tanto o tamanho funcional (6 PF), quanto o tamanho não-funcional (200 PS) como entrada para seu processo de estimativa.

Req. #	Tipo da SCU	UCS Impactada	Contado pelo IFPUG	Contagem de PF	Avaliação SNAP	Pontuação SNAP						Total SNAP
						Formatação de Dados	Validações na Entrada de Dados	Alterações de IU	Múltiplos Métodos de Entrada	Movimentações de Dados Internos		
1	Processo Elementar	Nova interface de fax: Atualizar fatura	Não	0	Sim	0	0	0	12	26	38	
1	Processo Elementar	Decodificar os formulários recebidos via Fax	Não	0	Sim	8	0	0	0	26	34	
2a	Processo Elementar	Interface Online: Lógica de validação de Atualizar fatura	Sim	6	Sim	0	8	0	0	23	31	
2b	Processo Elementar	Atualizar o arquivo cache do arquivo de clientes	Não	0	Sim	0	0	0	18	23	41	
2c	Processo Elementar	Interface online: Atualizar fatura	Não	0	Sim	0	0	24	0	0	24	
2c	Processo Elementar	Interface online: Tela Inicial	Não	0	Sim	0	0	32	0	0	32	

Tabela 1-23 Estudo de Caso SNAP

**Página deixada em branco intencionalmente**





---

---

---

## Parte 2 – Exemplos

---

---

**Página deixada em branco intencionalmente**



---

## Exemplos

---

**Introdução** Este capítulo inclui inúmeros exemplos de como os requisitos não-funcionais podem ser avaliados utilizando categorias e subcategorias.

**Conteúdo** Este capítulo inclui os seguintes exemplos:

<b>Tópico</b>	<b>Página</b>
Exemplo 1: Alteração de texto livre para valores válidos	1-2
Exemplo 2: Dados de Segurança	1-3
Exemplo 3: Backup de Dados Internos e transformação de dados	1-4
Exemplo 4: Criando novos Produtos e Ofertas	1-6
Exemplo 5: Aderência a Normas	1-7
Exemplo 6: Ajuda	1-9
Exemplo 7: Incluir Métodos de Entrada e Saída	1-10
Exemplo 8: Múltiplas Plataformas de Software	1-12
Exemplo 9: Melhoria de Desempenho	1-13
Exemplo 10: Jobs Batch	1-14
Exemplo 11:	1-15
Exemplo 12: Utilizando Componentes	1-16

## Exemplo 1: Alteração de texto livre para valores válidos

**Requisito:** Um sistema de pedidos de uma loja de Varejo Internacional possui diversos campos de texto livre que serão substituídos por listas de valores válidos para melhorar a exatidão dos dados inseridos e reduzir falhas devido a erros de validação. Não é requerida nenhuma nova funcionalidade.

A validação do endereço irá verificar o país, estado, cidade, rua e o número da residência. A descrição do produto é composta pelo tipo do produto, fabricante, número do modelo e cor.

Dois processos elementares são identificados como impactados pelo requisito e precisam ser melhorados (“Incluir pedido de produtos” e “Alterar pedido de produtos”).

O projeto da solução envolve uma subcategoria (“Validações na Entrada de Dados”). A UCS é o processo elementar.

Vamos assumir que o # de Níveis de Aninhamento no PE “Incluir pedido de produtos” seja 3 e em “Alterar pedido de produtos” seja 2. Vamos assumir, ainda, que o #DER validados no PE “Incluir pedido de produtos” seja 10 e em “Alterar pedido de produtos” seja 5.

1.1 Validações na Entrada de Dados					
No.	Descrição da UCS	# DERs	# Níveis de Aninhamento	Fórmula	PS=
1	Incluir pedido de produtos	10	3	$=3*\#DERs$	30
2	Alterar pedido de produtos	5	2	$=2*\#DERs$	10

*Tabela 2-1: Exemplo 1, cálculo dos PS para Validações na Entrada de Dados*

Número de PS total para o projeto =  $\sum$  dos PS para todas as UCS da subcategoria =  $30+10 = 40$  PS

## Exemplo 2: Dados de Segurança

**Requisito:** Para satisfazer os novos padrões de segurança de um banco, foi decidido criptografar os dados transferidos entre os sistemas, bem como os dados apresentados nas interfaces com o usuário. Os dados de Informações Pessoais Sensíveis (IPS) deverão ser criptografada pela aplicação anterior ao envio a outros sistemas. Quaisquer dados de IPS apresentados nas interfaces com o usuário deverão ser substituídos pelo símbolo \*.

O projeto da solução requer escrever um programa para criptografia dos dados de IPS que serão utilizados pelos processos da aplicação 1.

O desenho da solução envolve uma subcategoria (“Formatação de Dados”). A UCS é o processo elementar.

Três processos elementares foram identificados como impactados: “Visualizar detalhes do assinante”, “Visualizar histórico de pagamentos” (ambos para substituir os dados de IPS por \*) e “Extrato de informações de clientes enviado à Aplicação 2” (processo para enviar dados criptografados em 16 bits).

Para determinar o nível de complexidade de transformação, substituir os dados por \* poderia ser qualificado como Baixa e a criptografia dos dados poderia ser qualificada como de complexidade alta. Os seguintes dados são considerados como IPS: Número do cartão de crédito, Grupo sanguíneo, CPF, Número de telefone e Histórico de crédito.

Tais informações são utilizadas em todos os 3 processos.

1.3 Formatação de Dados					
No.	Descrição da UCS	Complexidade da Transformação	# DERs	Fórmula	PS=
1	Visualizar detalhes do assinante	Baixa	5	$2 * \#DERs$	10
2	Visualizar histórico de pagamentos	Baixa	5	$2 * \#DERs$	10
3	Extrato de informações de clientes enviado à Aplicação 2	Alta	5	$5 * \#DERs$	25

*Tabela 2-2: Exemplo 2, cálculo dos PS para Formatação de Dados*

Número de PS total para o projeto =  $\sum$  dos PS para todas as UCS da subcategoria =  $10+10+25 = 45$  PS

## Exemplo 3: Backup de Dados Internos e transformação de dados

### Cenário

Uma aplicação foi projetada com uma arquitetura de três camadas: Interface do Usuário, Camada Intermediária e Servidor Retaguarda. A camada Servidor Retaguarda contém a base de dados, a Camada Intermediária contém os processos de negócio e a Interface do Usuário é o front end para os usuários da aplicação poderem visualizar e manter os dados.

Esta aplicação precisa evoluir para uma plataforma de tecnologia avançada a fim de criar um sistema mais intuitivo e fácil de utilizar. A plataforma inclui hardware e software que gerenciam efetivamente a conectividade, acesso e sincronização. Isto significa que a Interface do Usuário deve suportar o usuário do desktop e um dispositivo remoto (Handset – tal como um smartphone) para ter portabilidade. Para reduzir o tempo e esforço, os usuários de dispositivos remotos devem ser capazes de realizar o download do último trabalho e acessar os dados. Isto requer a replicação de dados do servidor dentro do handset.

### Requisitos

1. Os dados na interface do usuário devem ser agrupados para melhorar a usabilidade.
2. Segurança e recuperabilidade: É necessário criar backup dos dados importantes.

As três camadas da solução são consideradas como três partições diferentes dentro desta fronteira da aplicação.

O processo “Visualizar Pedidos” será alterado para realizar o agrupamento de dados na partição da Camada Intermediária e, então, enviar os dados à partição de Interface do Usuário. A melhoria será realizada apenas na Camada Intermediária. O processo “Visualizar Pedidos” tem um total de 20 DER (soma dos campos distintos de entrada e saída na camada intermediária) e um total de 2 ALIs/AIEs para atualização/leitura. Esta solução para o requisito 1 envolve uma subcategoria, a 1.4 (Movimentações de Dados Internos). A UCS é o PE dentro da partição da Camada Intermediária.

O backup será criado em uma tabela na Camada de Retaguarda e o processo de backup copiará os dados do arquivo de Pedidos no backend e os apagará após 1 dia. Isto requer a criação de um novo processo de backup e a criação de uma nova tabela de backup dentro da partição de retaguarda.

O arquivo “Pedido” da retaguarda é replicado no front end para acesso dos usuários. Todos os DER no arquivo “Pedido” devem ser considerados aqui. Os ALR são o arquivo da retaguarda e o arquivo replicado no front end.

O “Processo de Backup” lerá 20 DERs do arquivo de dados “Pedido” e atualizará os mesmos na tabela de Backup. A solução para o requisito 2 envolve 2 subcategorias, 1.4 (Movimentações de Dados Internos) para o novo processo de backup, e 3.2 (Tecnologia de Banco de Dados) para a nova tabela de backup.



1.4 Movimentações de Dados Internos					
No.	Descrição da UCS	Complexidade de Transformação	# DERs	Fórmula	PS=
1	Visualizar Pedidos	Baixa	20	4*DERs	80
2	Processo de Backup	Baixa	20	4*DERs	80

*Tabela 2-3: Exemplo 3, Cálculo do PS para Movimentações de Dados Internos*

3.2 Tecnologia de Banco de Dados					
No.	Descrição da UCS	Complexidade do ALR	# Alterações	Fórmula	PS=
1	Tabela de Backup	Baixa	1	6*#alterações	6

*Tabela 2-4: Exemplo 3, cálculo dos PS para Tecnologia de Banco de Dados*

Número de PS total para o projeto =  $\sum$  dos PS para todas as UCS da subcategoria = 80+80+6  
= 166 PS

## Exemplo 4: Criando novos Produtos e Ofertas

**Cenário** Uma aplicação de software de telecom é projetada para ter fácil manutenibilidade e rápido lançamento de novos produtos e ofertas para seus clientes. O provedor de serviço apenas necessita criar um conjunto de configurações para lançar o novo produto, sem quaisquer alterações de código. Nenhuma alteração de lógica é necessária para o processamento dos novos pedidos ou ofertas.

**Requisitos** Dez novos produtos devem ser lançados juntamente como três novas ofertas (Uma oferta é um pacote de produtos com preços específicos. Uma oferta é limitada quanto ao período de vendas, pode ser oferecida durante um período limitado).

Atributos	Produto 1	Produto 2	Produto 3	Produto 4	Produto 5	Produto 6	Produto 7	Produto 8	Produto 9	Produto 10
Tipo	IPTV	IPTV	Acesso à Internet	Acesso à Internet	Acesso à Internet	Linha Fixa	Linha Fixa	Linha Fixa	Móvel	Móvel
Largura de Banda	5	10	5-100	20	5	(-)	(-)	(-)	(-)	(-)
# de canais	100	150	(-)	(-)	(-)	(-)	(-)	(-)	(-)	(-)
SLA	Gold / Silver / Platinum	Gold / Silver / Platinum	Gold / Silver / Platinum	Gold / Silver / Platinum	Gold / Silver / Platinum	Regular / Senior	Regular / Senior	Regular / Senior	(-)	(-)
# de dispositivos	01/mai	01/mai	01/mai	01/mai	01/mai	(-)	(-)	(-)	(-)	(-)

*Tabela 2-5 Exemplo 4, descrição principal dos novos produtos*

### Atributos das ofertas

1. Lista de produtos
2. Esquema de plano de preços
3. Desconto
4. Início da vigência
5. Término da vigência

O projeto da solução requer a configuração de 10 novos produtos no arquivo lógico “Produtos” e a criação de 3 novas ofertas no arquivo lógico “Ofertas Disponíveis”. A solução envolve somente 1 subcategoria, 1.5 (Entregar Valor Agregado aos Usuários por Configuração de Dados).

Dois processos elementares são impactados. “Incluir Pedido” e “Alterar Pedido” são os dois processos que consumirão as configurações de produtos e ofertas ao criar ou alterar um pedido. O arquivo “Produtos” consiste de 10 registros a serem configurados – como existem 10 produtos, 10 registros serão configurados. Para “Ofertas Disponíveis”, há 5 atributos a serem configurados para cada oferta e, como existem 3 ofertas a serem criadas, 3 registros serão configurados.

**UCS: “Incluir Pedido”**

		<b>1.5 Entregar Valor Agregado aos Usuários por Configuração de Dados</b>			
<b>No.</b>	<b>Arquivo Lógico</b>	<b>Nível de Complexidade</b>	<b># Atributos</b>	<b>Fórmula</b>	<b>PS=</b>
1	Produtos	Baixa (10 registros)	5	6*#atributos	30
2	Ofertas	Baixa (3 registros)	5	6*#atributos	30

*Tabela 2-6: Exemplo 4, cálculo dos PS para Entregar Valor Agregado aos Usuários por Configuração de Dados, primeira UCS*

**UCS: “Alterar Pedido”**

		<b>1.5 Entregar Valor Agregado aos Usuários por Configuração de Dados</b>			
<b>No.</b>	<b>Arquivo Lógico</b>	<b>Nível de Complexidade</b>	<b># Atributos</b>	<b>Fórmula</b>	<b>PS=</b>
1	Produtos	Baixa (10 registros)	5	6*#atributos	30
2	Ofertas	Baixa (3 registros)	5	6*#atributos	30

*Tabela 2-7 Exemplo 4, cálculo dos PS para Entregar Valor Agregado aos Usuários por Configuração de Dados, segunda UCS*

Número de PS total para o projeto =  $\sum$  dos PS para todas as UCS da subcategoria =  
 $30+30+30+30 = 120$  PS

*Nota: Este exemplo foi corrigido conforme a versão 2.1 do APM.*

## Exemplo 5: Aderência a Normas

**Requisitos**

Aderência à norma ADA 508 (norma americana sobre acessibilidade) ou ao Guia de Acessibilidade de Conteúdo Web 2.0 da W3C (WCAG).

Incluir opções de acessibilidade, de modo que as pessoas com dificuldade para ouvir sons e com dificuldade para enxergar uma tela comum possam utilizar facilmente a aplicação.

A solução proposta é:

- Inserir ícones pop-ups quando um som for gerado (há 4 sons diferentes);
- Inserir fontes grandes e simples, com um único tamanho (14 pt), em todos os Menus e Campos em todas as telas;
- Uma cor específica em vez das opções normais de fontes.

Processo de Contagem SNAP:

- Alterar o tamanho da fonte de 10 pt para 14 pt e alterar as cores da fonte são consideradas alterações técnicas. O requisito não é considerado como a inclusão ou alteração de funcionalidades, portanto, nenhum PF é gerado;

- Assumimos que os ícones não precisam de cálculos (sem animação);
- A solução envolve uma subcategoria SNAP, 2.1 (Interfaces do Usuário);
- A UCS é o processo elementar;
- Os novos ícones e a alteração da fonte afetam 5 processos elementares (que não se sobrepõem);
- Em dois processos elementares, a alteração é contada como “Simples”; em 2 processos, a alteração é “Média” e um processo é considerado Complexo (Assumindo < 10 propriedades GUI incluídas/alteradas para Simples, 10 – 15 propriedades incluídas/alteradas para Média e mais que 15 propriedades incluídas/alteradas para Complexa).
- Há um conjunto fixo de elementos de IU para cada PE.

**Nota:** Os 4 ícones que acompanham os sons são um elemento de IU distinto (8 propriedades: nome, tipo, resolução, tamanho, orientação, largura de abertura, localização (x) e localização (y)). As fontes aparecem nos seguintes elementos de IU distintos: menus, ícones, até 11 tipos de controles, abas.

- PE 1 – 5 elementos de IU distintos impactados;
- PE 2 – 10 elementos de IU distintos impactados;
- PE 3 – 5 elementos de IU distintos impactados;
- PE 4 – 13 elementos de IU distintos impactados;
- PE 5 – 7 elementos de IU distintos impactados.

2.1 Interfaces do Usuário					
No.	Descrição da UCS	Nível de Complexidade	# elementos de IU distintos	Fórmula	PS=
1	PE 1	Baixa	5	$2 \times \# \text{ elementos de IU únicos}$	10
2	PE 2	Baixa	10	$2 \times \# \text{ elementos de IU únicos}$	20
3	PE 3	Média	5	$3 \times \# \text{ elementos de IU únicos}$	15
4	PE 4	Média	13	$3 \times \# \text{ elementos de IU únicos}$	39
5	PE 5	Alta	7	$4 \times \# \text{ elementos de IU únicos}$	28

*Tabela 2-8: Exemplo 5, cálculo dos PS para Interfaces do Usuário*

Número de PS total para o projeto =  $\sum$  dos PS para todas as UCS da subcategoria =  
 $10+20+15+39+28 = 112 \text{ PS}$

## Exemplo 6: Ajuda

**Requisitos:** Acrescentar “Ajuda” adicional a uma aplicação.

Solução proposta:

- Telas pop-up aparecerão quando o usuário clicar com o botão direito do mouse sobre um campo, contendo uma explicação de como e quando este campo deve ser usado;
- As explicações incluirão um hyperlink para um vídeo de demonstração ou para um assistente com um conjunto de perguntas e respostas.

A solução envolve a subcategoria Métodos de Ajuda. Vamos assumir que existam 5 PE diferentes em que estas ferramentas de ajuda serão implementadas. A solução resultou na implementação do tipo de ajuda “Contexto + Online” em 4 dos PEs, e do tipo “Online” em 1 PE.

2.2 Métodos de Ajuda					
No.	Descrição da UCS	Tipo de Ajuda	# Itens de Ajuda incluídos	Fórmula	PS=
1	Solicitar um Novo Serviço	Contexto + On-line	14	=3*# itens de ajuda	42
2	Alterar o Pedido	Contexto + On-line	12	=3*# itens de ajuda	36
3	Cancelar um Serviço	Contexto + On-line	4	=3*# itens de ajuda	12
4	Fazer “upgrade” em um Serviço	Contexto + On-line	4	=3*# itens de ajuda	12
5	Alterar Detalhes do Endereço	Texto On-line	10	=2*# itens de ajuda	20

*Tabela 2-9: Exemplo 6, cálculo dos PS para Métodos de Ajuda*

Número de PS total para o projeto =  $\sum$  dos PS para todas as UCS da subcategoria =  
 $42+36+12+12+20 = 122$  PS

## Exemplo 7: Incluir Métodos de Entrada e Saída

### (Abordagem de Única Instância)

**Cenário** A aplicação de software de um banco suporta 5 processos diferentes (em termos de PF, Processos Elementares): Criar Conta, Alterar Conta, Realizar Pagamento, Relatório Sintético de Criação de Contas no Dia e Relatório de Créditos e Débitos no Dia.

**Requisitos** Atualmente, os três Processos Elementares “Criar Conta”, “Alterar Conta” e “Realizar Pagamento”, recebem sua entrada através da digitação de dados pelo teclado. O banco deseja melhorar o software para que estes três processos sejam capazes de aceitar a entrada por meio de documentos digitalizados e pela leitura de códigos de barras.

(Os processos “Criar Conta” e “Alterar Conta” processam 20 DER cada e o processo “Realizar Pagamento” processa 15 DER.)

O “Relatório Sintético de Criação de Contas no Dia” e o “Relatório de Créditos e Débitos no Dia” são atualmente enviados impressos no formato CSV. O banco deseja melhorar o software para ser capaz de produzir a saída destes processos sob a forma de PDF impresso e de correio eletrônico aos destinatários.

(O Relatório Sintético de Criação de Contas no Dia possui 15 DER e o Relatório de Créditos e Débitos no Dia possui 10 DER.)

O desenho da solução para este requisito envolve duas subcategorias, 2.3 (Múltiplos Métodos de Entrada) e 2.4 (Múltiplos Métodos de Saída).

2.3 Múltiplos Métodos de Entrada					
No.	Descrição da UCS	Nível de Complexidade	# Métodos de Entrada Adicionais	Fórmula	PS=
1	Criar Conta	Alta	2	6* # Métodos de Entrada Adicionais	12
2	Alterar Conta	Alta	2	6* # Métodos de Entrada Adicionais	12
3	Realizar Pagamento	Média	2	4* # Métodos de Entrada Adicionais	8

Tabela 2-10: Exemplo 7, cálculo dos PS para Múltiplos Métodos de Entrada

2.4 Múltiplos Métodos de Saída					
No.	Descrição da UCS	Nível de Complexidade	# Métodos de Saída Adicionais	Fórmula	PS=
4	Relatório Sintético de Criação de Contas no Dia	Média	2	4* # Métodos de Saída Adicionais	8
5	Relatório de Créditos e Débitos no Dia	Média	2	4* # Métodos de Saída Adicionais	8

Tabela 2-11: Exemplo 7, cálculo dos PS para Múltiplos Métodos de Saída

PS total para o projeto =  $\sum$  de PS para a subcategoria 2.3 +  $\sum$  de PS para a subcategoria 2.4 = 12+12+8+8+8=48 PS

## Exemplo 8: Múltiplas Plataformas de Software

**Requisitos** Na definição da solução para um projeto de software, foi decidido que parte deste (que contém 5 diferentes processos elementares de negócio), precisaria ser desenvolvida em um ambiente multiplataforma.

Três destes Processos Elementares (PE1, PE2 e PE3) precisariam ser desenvolvidos com uma combinação de Java, C e XML.

Os outros dois (PE4 e PE5) precisariam ser desenvolvidos com uma combinação de C e SQL.

Uma vez que a necessidade de desenvolver o software em múltiplas plataformas de software é um requisito técnico, isto não é coberto por PF.

Esta solução envolve 1 subcategoria, 3.1 (Múltiplas Plataformas).

3.1 Múltiplas Plataformas				
No.	Descrição da UCS	# de Plataformas de Software	Mesma família de software?	PS=
1	PE1	3	Não	60
2	PE2	3	Não	60
3	PE3	3	Não	60
4	PE4	2	Não	40
5	PE5	2	Não	40

*Tabela 2-12: Exemplo 8, cálculo dos PS para Múltiplas Plataformas*

PS total para o projeto =  $\sum$  de PS para todas as UCS da subcategoria = 60+60+60+40+40 = 260 PS



## Exemplo 9: Melhoria de Desempenho

**Requisitos** O cliente de aplicações de software de Telecomunicações requer a melhoria no desempenho de algumas funcionalidades.

O tempo de produção para criar um pedido e criar um assinante precisa ser melhorado da média de 2 minutos para 1,5 minutos ou menos. “Visualizar Pagamentos” deve ser melhorado de 10 para 8 segundos ou menos, para todos os clientes. As transações de “Realizar Pagamento” precisam ser melhoradas de 3 segundos para 2 segundos.

O projeto da solução requer as seguintes alterações:

1. Aperfeiçoar as queries SQL em “Criar Pedido”, “Criar Assinantes” e “Realizar Pagamento”, para realizar mais rapidamente as atualizações na base de dados (utilizando SQL parametrizado, melhorando o manuseio da conexão com a base de dados, melhor implementação dos “commits” na BD, etc.).
2. Criação de uma view indexada na base de dados “Pagamentos”, de modo que “Visualizar Pagamentos” possa ler a view em vez de ler a base de dados (isto irá melhorar o desempenho do processo “Visualizar Pagamentos”). Serão necessárias alterações nas consultas SQL correspondentes para poder ler os dados desta nova view.
3. Os processos “Criar Pedido” e “Criar Assinante” leem/atualizam as bases de dados “Cliente”, “Assinante” e “Pedido”. As bases de dados “Cliente” e “Assinante” possuem mais que 5 RLR e mais que 50 DER, sendo portanto de complexidade Alta. A base de dados “Pedido” é de complexidade Média (~30 DER).
4. “Realizar Pagamento” lê/atualiza as bases de dados “Assinante” e “Pagamentos”. A base de dados “Pagamentos” é um ALR de complexidade Média.
5. “Visualizar Pagamentos” lê somente a base de dados “Pagamentos”.

A solução deste requisito envolve uma subcategoria, 3.2 (Tecnologia de Banco de Dados).

		3.2 Tecnologia de Banco de Dados					
No.	Descrição da UCS	Maior nível de complexidade dos ALR envolvidos			# de alterações na base de dados	Fórmula	PS=
		# de DER	# de RLR	Complexidade			
1	Criar Pedido	56	5	Alta	1	12*# de alterações	12
2	Criar Assinante	40	8	Alta	1	12*# de alterações	12
3	Realizar Pagamento	25	5	Média	1	9*# de alterações	9
4	Visualizar Pagamento	25	5	Média	2	9*# de alterações	18

Tabela 2-13: Exemplo 9, cálculo do PS para Tecnologia de Banco de Dados

PS total para o projeto =  $\sum$  de PS para todas as UCS da subcategoria = 12+12+9+18 = 51 PS

## Exemplo 10: Jobs Batch

### Requisitos

Uma aplicação de software para bancos fornece funcionalidades para aceitar todos os depósitos e pagamentos e funcionalidades para saques, transferir dinheiro ou realizar pagamentos.

O Banco possui um requisito para criar um job batch diário que irá:

1. Ler os seguintes arquivos lógicos: Conta, Pagamentos e Créditos;
2. Aplicar a lógica do negócio para os cálculos e transformações de dados requeridos;
3. Atualizar dois arquivos lógicos: Resumo de Créditos e Resumo de Débitos.

O job batch é disparado à meia-noite e processa todos os dados das últimas 24 horas. Este job batch não recebe qualquer entrada de dados de negócio e não fornece qualquer saída de dados de negócio. Todo o escopo do job é limitado dentro da fronteira da aplicação.

O job batch deve ler 3 bases de dados e atualizar 2. O job processa um total de 25 DER.

Uma vez que não existe entrada ou saída atravessando a fronteira, a aplicação de PF pode não ser possível. Por isso, este requisito envolve 1 subcategoria, 3.3 (Processos Batch).

3.3 Processos Batch						
No.	Descrição da UCS	# de ALR	Nível de Complexidade	# de DER	Fórmula	PS=
1	Job Batch Diário de Resumos de Créditos e Débitos	5	Média	25	6*# de DER	150

*Tabela 2-14: Exemplo 10, cálculo para Processos Batch*

PS total para o projeto =  $6 \times 25 = 150$  PS

## Exemplo 11: Múltiplas Interfaces

### Requisitos

Uma aplicação de software de Tarifação de Telecom atualmente recebe sua entrada de uma aplicação de Rede e envia sua saída para uma aplicação de Faturamento.

Após a aquisição de outra companhia (com diferentes aplicações), a companhia de Telecom decidiu mesclar as atividades de Tarifação com a sua aplicação de Tarifação.

Informações de Chamadas e Uso de Dados deverão fluir para a aplicação de Tarifação a partir de duas entradas adicionais. A aplicação de Tarifação deverá enviar todas as informações de voz e uso de dados para uma saída adicional.

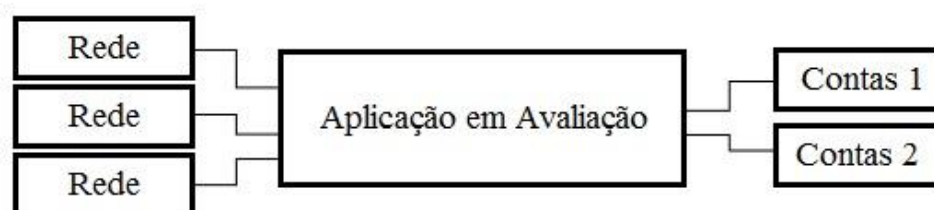


Figura 2-1: Exemplo 11, Visão da Arquitetura Requerida

Atualmente, o “Extrato de Uso de Voz” recebe uma entrada de uma Aplicação Externa e envia a saída para Faturamento. Depois que a alteração for implementada, será necessário verificar a interface de entrada e, com base nela, enviar a saída para o sistema de Faturamento correspondente, depois de realizar a formação ou ordenação de dados específica da interface. A funcionalidade permanece a mesma e nenhuma alteração funcional é requerida.

Assume-se que o Extrato de Uso de Voz e o Extrato de Uso de Dados tenham 20 DER como entrada e saída.

O projeto da solução envolve uma subcategoria, 4.2 (Múltiplas Interfaces de Entrada / Saída).

Precisamos calcular os PS para as interfaces de Entrada e Saída separadamente.

### Interfaces de Entrada:

No.	Descrição da UCS	4.2 Múltiplas Interfaces de Entrada / Saída			
		Nível de Complexidade baseado no #de DER	# de Interfaces de Entrada	Fórmula	PS=
1	Extrato do Uso de Voz	Média	3	$4 * \# \text{ de Interfaces}$	12
2	Extrato do Uso de Voz	Média	3	$4 * \# \text{ de Interfaces}$	12

Tabela 2-15: Exemplo 11, cálculo dos PS para Múltiplas Interfaces de Entrada

Interfaces de Saída:

No.	Descrição da UCS	4.2 Múltiplas Interfaces de Entrada / Saída			
		Nível de Complexidade baseado no #de DER	# de Interfaces de Saída	Fórmula	PS=
1	Extrato do Uso de Voz	Média	2	$4 * \# \text{ de Interfaces}$	8
2	Extrato do Uso de Voz	Média	2	$4 * \# \text{ de Interfaces}$	8

Tabela 2-16: Exemplo 11, cálculo dos PS para Múltiplas Interfaces de Saída

PS total para o projeto =  $\sum$  dos PS para entradas e saídas =  $12+12+8+8 = 40$  PS

## Exemplo 12: Utilizando Componentes

### Requisitos

Um cliente de varejo contata um vendedor de software para criar um novo website de compras de varejo que forneceria aos clientes uma experiência de compras única para este cliente e comum a todos os seus sites. Após analisar os requisitos do cliente, o vendedor lista os diferentes componentes já desenvolvidos para outros websites que podem ser usados na aplicação atual. Os componentes requeridos são:

- 1) Login
- 2) Apresentar o Inventário
- 3) Comparar Produtos
- 4) Incluir e Armazenar no Carrinho de Compras
- 5) Obter Detalhes do Cliente
- 6) Obter Detalhes da Entrega
- 7) Realizar Pagamento

A análise realizada pela equipe do vendedor conclui que eles podem reutilizar os componentes 1, 4 e 7 de seus produtos internos sem qualquer customização. As funcionalidades restantes precisarão ser desenvolvidas / adaptadas de acordo com as necessidades do cliente.

O projeto reutiliza 3 componentes internos. A avaliação SNAP para desenvolvimento de software baseado em componentes é:

Tipo	Cálculo do PS
Componentes Internos	$PS=3*3 = 9$ PS
Componentes de Terceiros	N/A

Tabela 2-17: Exemplo 12, cálculo dos PS para Software Baseado em Componentes

PS total para o projeto =  $\sum$  dos PS para os componentes internos = 9 PS

**Página deixada em branco intencionalmente**



---

---

---

## Parte 3 – Apêndices

---



---

## Parte 3 - Apêndices

---

**Introdução**

A Parte 3 fornece Apêndices sobre diversos tópicos relacionados.

O Apêndice A fornece um Glossário dos termos utilizados dentro do processo SNAP e do APM.

O Apêndice B fornece exemplos de utilização e a ligação entre os tamanhos funcional e não-funcional.

O Apêndice C fornece um Índice do documento.

**Conteúdo**

A Parte 3 inclui as seguintes seções:

Tópico	Página
Apêndice A – Glossário	A-1
Apêndice B – Ligação entre o APM do IFPUG e o CPM do IFPUG	B-1
Apêndice C – Índice	C-1

---



### Glossário

---

<b>Termo</b>	<b>Descrição</b>
APM	Assessment Practice Manual
Avaliação Não-Funcional	Tamanho do software em pontos SNAP
Categoria	Um grupo de componentes, processos ou atividades utilizados com o propósito de satisfazer o requisito não-funcional.
Categorias de Avaliação	O framework em que se baseia a Avaliação SNAP
Classificações de Avaliação	A resposta a uma questão da avaliação utilizando determinados critérios
CPM	Counting Practices Manual
Crítérios de Avaliação	Informações utilizadas para determinar os valores usados para efetuar a avaliação
Data Efetiva de Decisão	Data efetiva das decisões de atualizar o APM
Escopo da Avaliação	Define o conjunto de requisitos não-funcionais incluídos na avaliação
Estudo de Impacto	Um estudo que é iniciado caso haja a possibilidade de que uma prática comum a diversas organizações ou tipos de aplicações possa mudar
Framework de Tamanho Técnico	Um framework que mediria os aspectos técnicos do desenvolvimento de software (TSF no original)
IFPUG	International Function Point Users Group
NFSSC	Non-Functional Sizing Standards Committee
Processo de Avaliação Não-Funcional	Procedimento descrito no APM para chegar a uma contagem SNAP
Processo Elementar	A menor unidade de atividade significativa para o(s) usuário(s)
Questões de Avaliação	Questões relacionadas a atributos específicos que permitem a avaliação não-funcional de uma determinada subcategoria

Partição	Um conjunto de funções de software, dentro da fronteira de uma aplicação, que compartilham critérios e valores homogêneos de avaliação
Requisitos do Usuário	Requisitos descrevendo o que o usuário está solicitando (RU)
Requisitos Funcionais do Usuário	Um subconjunto dos requisitos do usuário, requisitos que descrevem o que o software deve fazer em termos de tarefas e serviços. (ISO 14143-1:2007) (RFU)
Requisitos Não-Funcionais do Usuário	Um requisito de software que descreve não o que o software irá fazer, mas como o software irá fazê-lo. [ISO/IEC 24765, Systems and Software Engineering Vocabulary.] Sin: restrições de projeto, requisito não-funcional. Ver também: requisito funcional do usuário
Revisão do NFSSC	O NFSSC revisa e discute a razão lógica para cada atualização proposta e seu possível impacto, que irá direcionar o comitê a aceitar ou rejeitar a atualização proposta
ROI	Retorno sobre o investimento = $([\text{Ganho do Investimento}] - [\text{Custo do Investimento}]) \text{ dividido pelo } [\text{Custo do Investimento}]$
SNAP	<u>S</u> oftware <u>N</u> on-functional <u>A</u> ssessment <u>P</u> rocess
Subcategoria	Um componente, processo ou atividade executada dentro do projeto para atender ao requisito não-funcional
Tipo de Dado Elementar	Atributo distinto, reconhecido pelo usuário e não repetido
Tipo de Registro Elementar	Subgrupo de tipos de dados elementares reconhecidos pelo usuário dentro de uma função de dados
UCS	Veja Unidade de Contagem SNAP
Unidade de Contagem SNAP (UCS)	O componente ou atividade no qual a complexidade e tamanho são avaliados. A UCS pode ser um componente, um processo ou uma atividade identificada de acordo com a natureza de uma ou mais subcategorias. Em alguns casos, a UCS é idêntica ao processo elementar
Valor da Avaliação ("Pontos SNAP")	Medida de Tamanho Não-Funcional (PS)
Visão do Usuário	Uma visão do usuário são os requisitos não-funcionais do usuário conforme percebidos pelo usuário

---

### Ligação entre o APM do IFPUG e o CPM do IFPUG

---

- Introdução** Esta seção descreve como o Processo de Avaliação Não-Funcional de Software relaciona-se ao processo de Análise de Pontos de Função.
- Atenção:** Esta é uma visão preliminar da ligação entre PS e PF. Mais análises serão necessárias para determinar como estas duas medidas de tamanho poderão ser utilizadas em conjunto. As versões futuras irão se aprofundar nesta questão.
- Conteúdo** Este apêndice fornece o diagrama do Processo SNAP e inclui exemplos de usos potenciais do SNAP.

Tópico	Página
Ligação entre a APF e o SNAP	B-2
Diagrama da Ligação entre os Processos da APF e do SNAP	B-2
Contando Pontos de Função e Pontos SNAP	B-2
Usos Potenciais dos Pontos SNAP	B-7
Medida Não-Funcional (Pontos SNAP)	B-8

## Ligação entre a APF e o SNAP

O tamanho não-funcional pode ser utilizado em conjunto com o tamanho funcional para fornecer uma visão geral do projeto ou aplicação, incluindo ambos os tamanhos, funcional e não-funcional.

Avaliar o impacto sobre o esforço dos projetos como resultado dos PS está fora do escopo deste documento. As organizações devem coletar e analisar seus próprios dados para determinar os impactos não-funcionais sobre a produtividade.

Os usos potenciais do tamanho não-funcional em conjunto com o tamanho funcional são fornecidos sob a forma de exemplos.

### Diagrama da Ligação entre os Processos da APF e do SNAP

O diagrama abaixo ilustra a ligação entre o processo de APF e o processo do SNAP.

O Propósito, Escopo e as Fronteiras Lógicas das Aplicações precisam ser consistentes entre os Processo da APF e do SNAP.

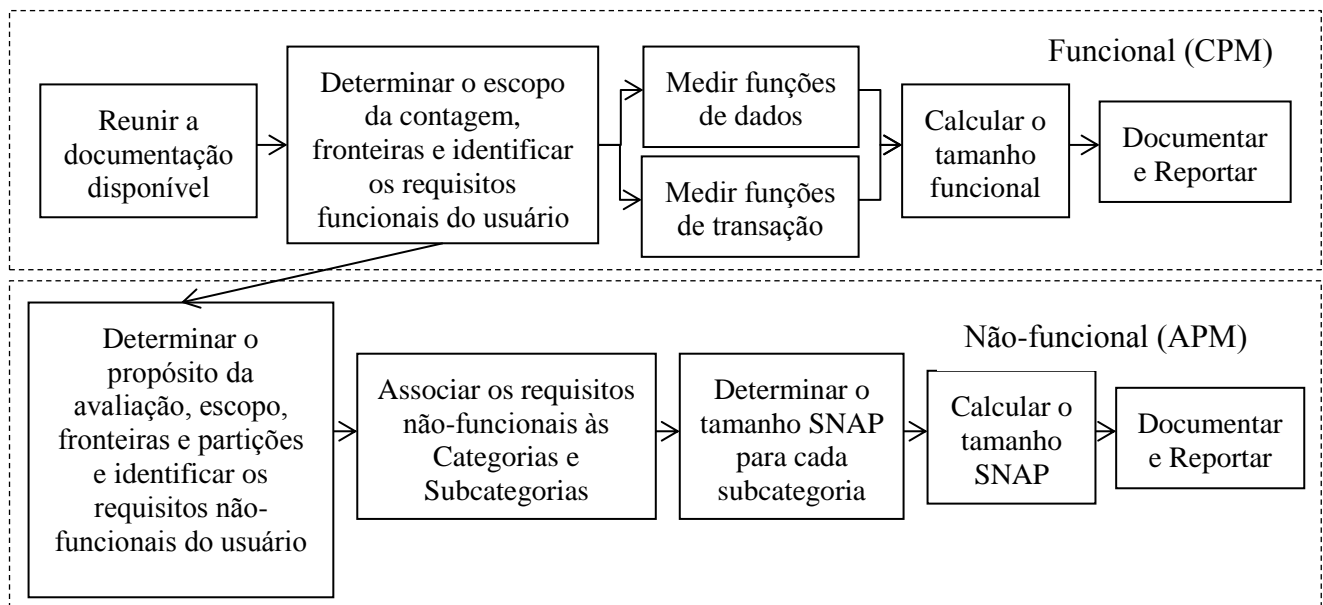


Figura 3-1: A ligação entre os Processos da APF e do SNAP

## Contando Pontos de Função e Pontos SNAP

Um requisito pode conter ambos os aspectos funcional e não-funcional. Neste caso, o requisito terá um tamanho funcional, medido em Pontos de Função, e um tamanho SNAP, medido em Pontos SNAP.

**Tal requisito deve ser dividido em seus componentes funcionais e componentes não-funcionais, e esta segregação deve ser acordada tanto pelo usuário/cliente, quanto pelas equipes de desenvolvimento. Utilize os PF para as partes RFU dos requisitos e PS para as partes não-funcionais dos requisitos (RNF).**

A tabela seguinte é um guia. Para definir requisitos não-funcionais, padrões ISO ou um padrão similar pode ser utilizado.

<b>Caso #</b>	<b>Circunstâncias</b>	<b>Descrição</b>	<b>Diretriz</b>
1	Os requisitos são apenas funcionais	Os usuários não possuem quaisquer requisitos não-funcionais explícitos ou implícitos	Contar somente Pontos de Função
2	Os requisitos são claramente marcados como RNF	As partes concordaram em uma clara segregação entre os requisitos funcionais e os não-funcionais; Os requisitos classificados como RNF não podem ser medidos com Pontos de Função	Contar somente Pontos SNAP
3	Os requisitos envolvem ambos os aspectos funcional e não-funcional	Os requisitos funcionais possuem RNF adicionais que podem ser claramente identificados	Ver tabela abaixo
4	Os requisitos são apenas funcionais, as transações atravessam Partições	Os requisitos funcionais podem envolver um único fluxo, ou múltiplos. No caso de múltiplos fluxos e utilizando as diretrizes atuais do CPM, cada fluxo pode não ser considerado um processo elementar distinto	Contar Pontos de Função para medir a funcionalidade nova/alterada do principal processo elementar, conforme o CPM; acrescentar o tamanho SNAP para as transações/fluxos, dentro da fronteira da aplicação que atravessam as partições
5	Os requisitos são funcionais, mas são fornecidos sem qualquer alteração no software	A funcionalidade (ou qualquer valor para o negócio) incluída ou modificada pela alteração em dados de referência ou por outros meios que não possam ser medidos por Pontos de Função, conforme as diretrizes atuais do CPM ou das práticas de contagem da organização.	Contar Pontos SNAP utilizando a subcategoria 1.5 (Entregar Valor Agregado aos Usuários por Configuração de Dados)

*Tabela 3-1 Inter-relações entre PF e PS*

## Requisitos que Envolvem Requisitos Funcionais e Não-Funcionais

A seguinte diretriz deve ser utilizada para se determinar como PF e pontos SNAP devem ser contados.

<b>Os requisitos não-funcionais são associados à seguinte subcategoria:</b>	<b>O que checar</b>	<b>Método de Contagem</b>
Validações na Entrada de Dados	O processo elementar	PF e SNAP. Ver as notas 1 e 2 abaixo.
Operações Lógicas e Matemáticas	O processo elementar	PF e SNAP. Ver as notas 1 e 2 abaixo.
Formatação de Dados	O processo elementar	PF e SNAP (o SNAP mede alterações nas transações devidas a requisitos não-funcionais).
Movimentações de Dados Internos	O processo elementar	PF e SNAP (o SNAP mede as movimentações de dados entre partições para os requisitos funcionais e não-funcionais). Ver a nota 3 abaixo.
Entregar Valor Agregado aos Usuários por Configuração de Dados	O processo elementar	PF e SNAP, mas não ambos – ver a nota 4 abaixo.
Interfaces do Usuário	O conjunto de telas definido pelo processo elementar	PF e SNAP (o SNAP mede somente os elementos de dados que não possuem alteração funcional, e, portanto, não são medidos por PF). A criação de um novo elemento de IU que é contado em PF não irá gerar PS. Modificar um elemento de IU de modo que seja acrescentada funcionalidade e, portanto, PF sejam contados, não irá gerar PS. Modificar um elemento de IU sem alteração na funcionalidade gerará PS.
Métodos de Ajuda	O conjunto de telas definido pelo processo elementar	O SNAP deve ser contado para os tipos de “ajuda” que não são contados em PF. Técnicas de Ajuda tais como tool tips, ajuda dinâmica ao passar o mouse (ajuda de contexto) são contados via SNAP. Páginas Web estáticas são contadas via SNAP.



<b>Os requisitos não-funcionais são associados à seguinte subcategoria:</b>	<b>O que checar</b>	<b>Método de Contagem</b>
Múltiplos Métodos de Entrada	O processo elementar	Quando a contagem de PF é utilizada para uma entrada, o SNAP deve ser utilizado somente para as entradas adicionais.
Múltiplos Métodos de Saída	O processo elementar	Quando a contagem de PF é utilizada para uma saída, o SNAP deve ser utilizado somente para as saídas adicionais.
Múltiplas Plataformas	O processo elementar	Quando a contagem de PF é utilizada para uma plataforma, o SNAP deve ser utilizado somente para as plataformas adicionais.
Tecnologia de Banco de Dados	O processo elementar	Alterações em tabelas, visões, partições da base de dados, alteração na capacidade da base de dados e alterações em consultas ou inserções que não alteram a funcionalidade: somente PS. Quando a inclusão de tabelas ou colunas for somente para propósitos não-funcionais, o SNAP deve ser utilizado.
Processos Batch	Os jobs batch	Quando um arquivo batch não é qualificado como nenhuma das funções de transação conforme as diretrizes de PF, utilize o SNAP para medir os jobs batch. Quando um job batch existente é dividido em múltiplos jobs e a razão disso for satisfazer requisitos não-funcionais, use o SNAP para medir os jobs batch.
Software Baseado em Componentes	O processo elementar	PF e SNAP.
Múltiplas Interfaces de Entrada / Saída	O processo elementar para PF e a aplicação avaliada para o SNAP	PF ao criar a funcionalidade. SNAP ao incluir interfaces sem modificar a funcionalidade. PF e SNAP ao modificar a funcionalidade e incluir interfaces.

*Tabela 3-2 Regras das inter-relações entre PF e PS*

**Notas:**

1. O SNAP introduz um tamanho não-funcional adicional à funcionalidade que foi medida anteriormente utilizando apenas Pontos de Função. Portanto, o tamanho total da aplicação agora inclui uma separação entre os aspectos funcionais e os aspectos não-funcionais.
2. Um requisito deve ser separado em seus aspectos funcionais e não-funcionais. Os aspectos funcionais são medidos utilizando Pontos de Função e os aspectos não-funcionais são medidos utilizando o SNAP.

Como exemplo, uma funcionalidade nova ou alterada pode ser entregue com validações, mas a validação por si só não impacta no tamanho funcional.

O SNAP acrescenta um tamanho não-funcional quando a validação é utilizada. Como resultado, o tamanho total aumenta (Pontos de Função e Pontos SNAP).

3. Partições:

Dois cenários podem aparecer aqui:

1. O Processo Elementar (que oferece a funcionalidade por si só, sem invocar vários outros fluxos ou processos) cruza múltiplas partições.
  2. Um Processo Elementar que consiste de múltiplos fluxos, que executam diversas tarefas. Estes fluxos podem realizar diferentes tarefas de busca, transformação, formatação e processamento de dados, e criar saídas que são finalmente consumidas pelo processo elementar. Estes fluxos, de acordo com as diretrizes do CPM, não são independentes e reconhecidos pelo usuário, não podendo ser quebrados em diversos processos elementares.
4. Entregar valor agregado aos usuários por Configuração de Dados  
Quando uma funcionalidade for incluída utilizando uma alteração e configuração de software, e isto gerar Pontos de Função, não acrescente o tamanho SNAP. O tamanho SNAP é para ser utilizado somente quando não ocorrer nenhuma alteração no código do software ou na estrutura da base de dados.

## Usos Potenciais dos Pontos SNAP

### Estimativa de Projeto

#### Estimar o Projeto

- Conte os PFs para o projeto – Utilize os índices de produtividade baseados no tamanho de FP e no tipo do projeto, para estimar o esforço funcional do projeto, etc., a partir de dados contidos em repositórios.
- Efetue a avaliação SNAP – Utilize dados históricos para determinar o impacto da pontuação SNAP sobre o esforço do projeto. Ajuste o esforço para mais ou para menos, dependendo da pontuação SNAP, para fornecer uma estimativa de esforço funcional/técnico.

O relacionamento entre o SNAP e o esforço pode ser específico para uma organização.

Uma vez que dados significativos de avaliações SNAP tenham sido coletados, uma “regra geral” poderá ser estabelecida por indústria, plataforma, etc., a partir de dados contidos em repositórios,

- Efetue a Avaliação de Risco/Atributos – Avalie os fatores organizacionais que impactam a produtividade. Utilize dados históricos de impactos de risco/atributos sobre o esforço e ajuste o esforço do projeto apropriadamente para mais ou para menos, com o intuito de fornecer uma estimativa funcional/técnica/de risco.

Nota: Pode-se utilizar todas as 3 estimativas de esforço individualmente ou em qualquer combinação para produzir uma faixa de estimativa. O índice de produtividade para PF deve refletir somente o esforço requerido para implementar os requisitos funcionais, e o índice de produtividade para o SNAP deve refletir somente o esforço requerido para implementar os requisitos não-funcionais.

### ROI

#### Calcular a Estimativa do Retorno sobre Investimento (ROI) da substituição de uma Aplicação

- Conte os PFs da Aplicação para determinar o tamanho funcional da aplicação
- Complete a avaliação SNAP para determinar o tamanho não-funcional da aplicação

Para determinar o ROI, compare o custo do projeto de substituição e custos futuros de manutenção com o custo de manutenção da aplicação existente.

### Área(s) de Ênfase

#### Utilizar os PS para indicar as áreas não-funcionais de ênfase no projeto/aplicação.

- Examine as respostas detalhadas para a Avaliação SNAP comparadas às características da ISO/IEC 25010 (ou outro padrão de classificação de requisitos não-funcionais) para garantir o foco apropriado.

Exemplo: Um projeto com mais PS alocados para garantir a exatidão do que a atratividade, indica que há mais ênfase sobre a exatidão.

<b>Opções de Comparação</b>	<p>Comparação de alternativas não-funcionais através da visão geral dos custos totais (funcional e não-funcional).</p> <p>Exemplos:</p> <ul style="list-style-type: none"><li>• Usar Pacotes (COTS) ou Desenvolvimento</li><li>• Comparar tecnologias que entregam funcionalidades, com o objetivo de selecionar a tecnologia apropriada (por exemplo, tecnologias de banco de dados, tecnologias de interfaces)</li></ul>
<b>Custo de Manutenção</b>	<p>Auxiliar na avaliação total dos custos/recursos de Manutenção.</p> <ul style="list-style-type: none"><li>• Use PF para medir a funcionalidade da aplicação</li><li>• Use PS para medir as características não-funcionais da aplicação</li></ul> <p>Utilize dados históricos para estimar os custos de manutenção por tamanho funcional e não-funcional de cada aplicação, para planejar os recursos futuros e contribuir com as estratégias de manutenção e investimentos anuais.</p>

## Medida Não-Funcional (Pontos SNAP)

O CPM do IFPUG foi transformado em um padrão ISO para a medição do tamanho funcional com a exclusão das Características Gerais do Sistema (CGSs), que avaliam a funcionalidade geral da aplicação. (A APF do IFPUG v4.1 não ajustada foi recebida como um padrão ISO em 2003 – ISO/IEC 20926:2003 – e mais tarde em 2009 para a v4.3.) A consideração mais importante com relação a estas questões é como os requisitos não-funcionais afetam o tamanho. Como não são requisitos funcionais, não contribuem para o tamanho funcional. No entanto, ainda fazem parte do total de requisitos (funcionais e não-funcionais) para o software e, portanto, contribuem para o tamanho total dos requisitos.

Os PS foram desenvolvidos para serem utilizados na avaliação dos requisitos não-funcionais, independentemente das CGSs. Praticantes da avaliação dos Pontos SNAP NÃO devem utilizar PS e CGSs simultaneamente. A utilização de ambos pode inflar o tamanho não-funcional proveniente dos requisitos não-funcionais. Os PS podem ser utilizados em conjunto com os pontos de função (a partir de fronteiras idênticas), para fornecer uma visão geral de um projeto ou aplicação, incluindo ambos os tamanhos funcional e não-funcional.

---

## Índice

### A

#### Abordagem de Múltiplas Instâncias

Parte 1 5 - 19

Parte 1, 5 - 5, 20

#### Abordagem de Única Instância

Parte 1, 5 - 19

Parte 1, 5 - 5, 20

#### Aborgagem da Fórmula

Parte 1, 6 - 2

#### Ambiente Técnico

Parte 1, 2 - 6

Parte 1, 5 - 21

#### Aplicação

Parte 1, 1 - 8

Parte 1, 2 - 3

Parte 1, 2 - 2

Parte 1, 2 - 8

Parte 1, 4 - 2, 3, 4

Parte 1, 5 - 3, 6

Parte 1, 6 - 3, 5

Parte 2, 1 - 4, 6, 15

Prefácio - v, x, xiii, xiv

#### Arquitetura

Parte 1, 2 - 6

Parte 1, 3 - 3

Parte 1, 5 - 27

Parte 1, 6 - 3

#### Arquivo Lógico

Parte 1, 5 - 4, 7, 11, 25

Parte 2, 1 - 6

#### Arquivo Lógico Referenciado (ALR)

Parte 1, 5 - 7

Parte 1, 5 - 4

#### ASPA - PS da aplicação após o projeto de melhoria

Parte 1, 6 - 5

Parte 1, 4 - 3

Parte 1, 6 - 5

#### Avaliação Não-Funcional Estimada e Final

Parte 1, 3 - 4

#### Avaliação Não-Funcional nas Fases do Ciclo de Vida

Parte 1, 3 - 2

### B

#### Benefícios do SNAP

Parte 1, 1 - 10

### C

#### Calcular o Tamanho Não-Funcional

Parte 1, 2 - 8

Exemplo do Cálculo - Parte 1, 5 - 31

#### Categoria

Definição - Parte 1, 2 - 5

Arquitetura - Parte 1, 5 - 27

Operações de Dados - Parte 1, 5 - 6

Design de Interface - Parte 1, 5 - 13

Ambiente Técnico - Parte 1, 5 - 21

#### Complexidade

Definição - Parte 1, 5 - 2

Exemplo - Parte 1, 5 - 2

Tabela - Parte 1, 5 - 3

Parâmetros - Parte 1, 5 - 2

#### Complexidade da Subcategoria

Parte 1, 5 - 2

#### Complexidade do ALR

Parte 1, 5 - 7, 25

Parte 2, 1 - 5

#### Componente de Software

Parte 1, 5 - 27

#### Consistência com a APF

Parte 1, 4 - 2

### D

#### Dados de Código

Parte 1, 5 - 4, 6, 11

Parte 1, 5 - 4

**Desing de Interface**

Parte 1, 2 - 6  
 Parte 1, 5 - 13

**Dicas de Escopo**

Parte 1, 4 - 6

**Dicas de Fronteiras e Repartições**

Parte 1, 4 - 7

**Documentação**

Prefácio - xiv  
 Diretrizes para a Medição de Software - xiv  
 IFPUG, Um Introdução - xiv

**Documentação do IFPUG**

Prefácio - xiv

**Documentação Útil do Projeto/Aplicação**

Parte 1, 3 - 3

**DSP - PS do projeto de desenvolvimento**

Parte 1, 4 - 3  
 Parte 1, 6 - 3

**E****Elemento de IU**

Parte 1, 5 - 13, 14  
 Parte 2, 1 - 7  
 Apêndice B - 4

**Entregando Valor Agregado aos Usuário pela Configuração de Dados**

Parte 1, 5 - 12

**ESP - PS do projeto de melhoria**

Parte 1, 4 - 3  
 Parte 1, 6 - 4

**Exemplos**

Adicionando Métodos de Entrada e Saída - Parte 2, 10  
 Jobs Batch - Parte 2, 14  
 Alteração de Texto Livre para Valores Válidos - Parte 2, 2  
 Aderência a Normas - Parte 2, 7  
 Software Baseado em Componentes - Parte 2, 16  
 Criando Novos Produtos e Ofertas - Parte 2, 6  
 Dados de Segurança - Parte 2, 3  
 Ajuda - Parte 2, 9  
 Backup de Dados Internos e Transformação de Dados - Parte 2, 4  
 Múltiplas Interfaces - Parte 2, 15  
 Múltiplas Plataformas - Parte 2, 12  
 Melhoria de Desempenho - Parte 2, 13  
 Caso de Estudo do Cálculo SNAP - Parte 1, 6 - 6

**F****Formação de Dados**

Parte 1, 5 - 9  
 Parte 2, 1 - 3

**Fórmulas**

Tamanho da aplicação após o projeto de melhoria - Parte 1, 6 - 5  
 Categoria Arquitetura - Parte 1, 6 - 3  
 Categoria Operações de Dados - Parte 1, 6 - 2  
 Categoria Design de Interface - Parte 1, 6 - 2  
 Categoria Ambiente Técnico - Parte 1, 6 - 3  
 Projeto de Desenvolvimento - Parte 1, 6 - 3  
 Projeto de Melhoria - Parte 1, 6 - 4

**Framework SNAP**

Parte 1, 1 - 8

**Fronteira**

Definição de Fronteira - Parte 1, 4 - 5  
 Regras de Fronteira - Parte 1, 4 - 7  
 Parte 1, 2 - 3  
 Parte 1, 4 - 1, 2, 5, 7  
 Parte 1, 5 - 10  
 Parte 1, 6 - 7

**I****Interfaces do Usuário**

Parte 1, 2 - 6  
 Parte 1, 5 - 13, 14  
 Parte 2, 1 - 7  
 Parte 2, Exemplos - 7  
 Apêndice B - 4

**ISO/IEC 14143-1**

Parte 1, 1 - 3

**ISO/IEC 25010**

Parte 1, 1 - 6  
 Parte 1, 1 - 4, 6  
 Parte 1, 1 - 3  
 Parte 1, 1 - 2  
 Apêndice B - 7

**ISO/IEC 9126-1**

Parte 1 1 - 6  
 Parte 1, 2 - 6  
 Parte 1, 1 - 6

**Jobs Batch**

Parte 2, 1 - 14

**M****Manual**

Processo de Alteração - xii

Decisões Finais - xiii	
Frequência de Alterações - xii	
Como as decisões são comunicadas - xiii	
<b>Método de Ajuda</b>	
Parte 1, 5 - 15	
Parte 2, 1 - 9	
<b>Missão Crítica</b>	
Parte 1, 5 - 29, 30	
<b>Movimentações de Dados Internos</b>	
Parte 1, 5 - 10	
Parte 2, 1 - 4	
<b>Múltiplos Interfaces de Entrada / Saída</b>	
Parte 2, 1 - 15	
<b>Múltiplas plataformas</b>	
Parte 1, 5 - 21	
<b>Múltiplas Plataformas</b>	
Parte 2, 1 - 12	
<b>Múltiplas Métodos de Entrada</b>	
Parte 1, 5 - 19	
Parte 2, 1 - 11	
<b>Múltiplos Métodos de Saída</b>	
Parte 1, 5 - 20	
Parte 2, 1 - 11	

## O

<b>Objetivos do SNAP</b>	
Parte 1, 1 - 10	
<b>Operações de Dados</b>	
Parte 1, 2 - 6	
Parte 1, 5 - 6	
Parte 1, 6 - 2	
<b>Operações Lógicas e Matemáticas</b>	
Parte 1, 5 - 7	

## P

<b>Pontos SNAP (PS)</b>	
Parte 1, 2 - 8	
<b>Procedimento SNAP por Seção</b>	
Parte 1, 2 - 2	
<b>Processo Elementar (PE)</b>	
Parte 1, 5 - 3	
<b>Processos Batch</b>	
Exemplos Parte 2 - 14	
Parte 1, 5 - 25	
Apêndice B - 5	
<b>Projeto de Desenvolvimento</b>	
Parte 1, 2 - 3	

Parte 1, 4 - 3
Parte 1, 4 - 3
Parte 1, 1 - 9
Parte 2, 2 - 2
Prefácio - vi
<b>Projeto de Melhoria</b>
Parte 1, 6 - 7
Parte 1, 2 - 3
Parte 1, 4 - 3
Parte 1, 6 - 3, 4, 5, 6
Parte 2, 2 - 2

## Projeto de Software

Parte 1, 5 - 2
Parte 1, 1 - 8
Parte 2 - 12

## PS

Parte 1, 2 - 1, 8
Parte 1, 3 - 2
Parte 1, 4 - 3
Parte 1, 5 - 3, 4, 6, 8, 9, 10, 11, 14, 19, 20, 22, 25, 26, 27, 28, 30
Parte 1, 6 - 3, 7
Parte 2, 1 - 12
Parte 2, 1 - 2, 3, 5, 7, 9, 11, 14, 16
Parte 2, 6 - 2
Apêndice A, 1

## R

<b>Real time</b>	
Parte 1, 5 - 22, 29	
<b>Regras da Fórmula de Melhoria</b>	
Parte 1, 6 - 3	
<b>Partição</b>	
Apêndice A - 2	
Apêndice B - 6	
Parte 1, 2 - 3, 4	
Parte 1, 4 - 6	
Parte 1, 5 - 6	
Parte 1. 2 - 3	
Parte 2 (Exemplos) 4	
<b>Requisitos do Usuário</b>	
Parte, 1 - 3	
<b>Requisitos Funcionais do Usuário</b>	
Parte 1, 1- 3	
<b>Requisitos Não-Funcionais do Usuário</b>	
Parte 1, 1- 4	

**S****Sistemas de Missão Crítica / Real Time**

Parte 1, 5 - 29

**SNAP**

Requisitos de Treinamento - xv

**Software Baseado em Componentes**

Parte 1, 5 - 27

Parte 2, 1 - 16

**Subcategoria**

Definição - Parte 1, 2 - 6

Processo Batch - Parte 1, 5 - 25

Software Baseado em Componentes - Parte 1, 5 - 27

Validação na Entrada de Dados - Parte 1, 5 - 6

Formatação de Dados - Parte 1, 5 - 9

Tecnologia da Base de Dados - Parte 1, 5 - 24

Entregando Valor Agregado aos Usuários pela Configuração de Dados - Parte 1, 5 - 11

Métodos de Ajuda - Parte 1, 5 - 15

Movimentações de Dados Internos - Parte 1, 5 - 10

Operações Lógicas e Matemáticas - Parte 1, 5 - 7

Sistema de Missão Crítica / Real Time System  
Parte 1, 5 - 29

Múltiplas Métodos de Entrada - Parte 1, 5 - 19

Múltiplas Métodos de Saída - Parte 1, 5 - 20

Múltiplas Plataformas - Parte 1, 5 - 21

Interfaces do Usuário - Parte 1, 5 - 13

**T****Tecnologia da Base de Dados**

Parte 1, 5 - 24

Parte 2, 1 - 4, 13

**Tipo de Avaliação**

Definição - Parte 1, 2 - 3

Parte 1, 4 - 2

Parte 1, 4 - 3

**Tipo de Dado Elementar (DER)**

Parte 1, 5 - 3

**Tipo de Registro Elementar (RLR)**

Parte 1, 5 - 25

Parte 1, 5 - 3

Parte 2 (Exemplos) - 13

**U****Unidade de Contagem SNAP (UCS)**

Definição - Parte 1, 2 - 7

**V****Validação na Entrada de Dados**

Parte 1, 5 - 6

Parte 2, 1 - 2