

# Uso de Análise de Pontos de Função em Ambientes Ágeis

Ivonaldo Leite Torres<sup>1</sup>, Felipe Furtado<sup>1,2</sup>

<sup>1</sup> Centro de Estudos e Sistemas Avançados do Recife – C.E.S.A.R.EDU

<sup>2</sup> Centro de Informática – Universidade Federal de Pernambuco

[ivonaldotorres@gmail.com](mailto:ivonaldotorres@gmail.com), [felipe.furtado@cesar.org.br](mailto:felipe.furtado@cesar.org.br)

***Abstract.** With the need for delivery of software products in the shortest possible time, software factories began migrating to agile methodologies. However, methods to measure these agile software do not meet the standards required by public companies, for example, the metric of Function Points to estimate size, effort and cost. Based on this problem this paper proposes an implementation of FPA (Function Point Analysis) in the life cycle of Scrum.*

***Resumo** – Com a necessidade de entrega de produtos de software em menor tempo possível, as fábricas de software começaram a migrar para as metodologias ágeis. Porém, os métodos para mensurar software dessas metodologias ágeis não se enquadram nas normas exigidas pelas empresas públicas, por exemplo, a métrica de Pontos de Função para estimativa de tamanho, esforço e custo. Partindo dessa problemática este trabalho propõe uma implementação de APF (Análise de Pontos de Função) no ciclo de vida do Scrum.*

## 1. Introdução

É perceptível que a engenharia de software vem sofrendo grandes mudanças, tanto em seu contexto de produção, como no de avaliação. Segundo Santana (apud COHN, 2007) a qualidade dos softwares não pode mais ser medida pela quantidade de linhas de código, pois não é a quantidade métrica deles que irá determinar se um projeto foi desenvolvido em um bom prazo, com razoável custo e se atende a proposta oferecida e a necessidade do cliente.

Na visão de Jones (2008) com o desenvolvimento da tecnologia da informação tornou-se inviável utilizar o sistema de medição pelo número de linhas de código fonte,

já que anteriormente os programas eram escritos em linguagem de baixo nível e, a partir de 1967, surgiram linguagens como Fortran, COBOL e APL, que criaram aplicativos empresariais que facilmente chegavam a 100.000 linhas de produção.

Além disto, conforme ressalta Jones (2008) com linguagem de baixo nível, a codificação do ocupava 50% dos esforços despendidos na produção do software, o que justificava que a quantidade de linhas, de certa forma, qualificasse o software. Porém, com as novas linguagens, esse percentual vem diminuindo significativamente e fatores qualitativos passaram a ganhar mais relevância quanto se tenta medir e classificar a produção de softwares.

Com a produção de software em linguagem de alto nível, ficou inviável manter a métrica de contagem de linhas de código e buscou-se estabelecer outra forma de medida que se adequasse a qualidade do aplicativo, ao tempo de produção, ao custo e as expectativas do cliente.

Como caracteriza Albrecht (1979) a IBM foi pioneira neste processo criando uma metodologia para definir uma métrica de software aceitável ao novo contexto de TI, independentes das linhas de código, criando o que é conhecido como a técnica dos Pontos de Função.

Tal metodologia constitui uma maneira de análise e controle contínuo da produção, com criação e estabelecimentos de metas, correção de erros, visando construir aplicativos úteis, sem disfunções, em um rápido intervalo de tempo, cumprindo um orçamento prévio e tendo o feedback do cliente na maioria das fases de produção.

Na opinião de Matos (2010) a APF muitas vezes foi e é utilizada em conjunto com metodologias tradicionais, partindo de especificação de requisitos detalhada, depois de análise, projeto e etc.

Outro ponto a considerar por Matos (2010) é que a APF tem várias limitações no que diz respeito à estimativa de trabalho, por não levar em consideração requisitos não funcionais e de qualidade. Mas ela não está presa a um processo de software em particular. Alguns desenvolvedores argumentam que a técnica é muito antiga e inadequada para o desenvolvimento ágil. No entanto pode-se pensar em incluir o uso de APF no processo se isso for resolver alguma questão, ou seja, se isso agregar valor ao processo ágil de desenvolvimento.

## 1.1. Motivação

Com o surgimento das metodologias Ágeis, muitas empresas migraram seus processos de desenvolvimento de software para essas metodologias, buscando maior agilidade na entrega de produto de software, na qualidade e na satisfação do cliente integrando-o ao processo de desenvolvimento de software. Essas metodologias possuem também meios ágeis para estimativa de tamanho de software, as quais deixam o processo de desenvolvimento mais rápido e eficiente. Por esse processo de estimativa ser subjetivo, algumas empresas não validam seus resultados. Devido a esses fatores que as Empresas Governamentais passaram a adotar APF como medida oficial de tamanho de software, conforme Guia de Contagem APF(PERNAMBUCO..., 2010, p 65) e alterações na Instrução Normativa N° 2 de 14 de fevereiro de 2012 pela SLTI/MPOG e os acordões do Tribunal de Contas da União(TCU) 1.782/2007, 1.910/2007, 1.125/2009, 1.274/2010, 2.348/2009 e 1.647/2010 (BRASIL..., 2012, p 7) onde preconiza a utilização de métricas em contratos de software. Diante das normatizações criadas pelos órgãos governamentais surgiu necessidade de adaptar a APF a um processo ágil. Entre as metodologias ágeis foi adotado o Scrum, por sua grande aceitação e possibilidade de ser usado com outras metodologias.

## 1.2. Justificativa

De acordo com o Murphy(2010), as metodologias ágeis serão utilizadas em 80% dos projetos de desenvolvimento de software até 2012. O Scrum é a metodologia que lidera as pesquisas. Ao passo que essas metodologias ganham forças, existem aqueles que adotam essas metodologias como religião. Ao mesmo tempo que as metodologias ágeis ganhavam novos adeptos, alguns agilistas adquiriram um pensamento muito próximo do fanatismo religioso. Eles acreditam que a empresa e todos os seus métodos tem que ser rotulados de “ágil” (bom) ou “não ágil” (mau). Conforme ressalta Cockburn(2010) , um dos 17 autores do “Manifesto Ágil”, o que ele chamou de “Juramento de Não-Lealdade” com a finalidade de por fim no infantil fanatismo desses pseudo-agilistas.

*[...] Estou cansado de pessoas de uma escola de pensamento atacando idéias de uma outra escola de pensamento. Estou ansioso por pessoas que não se importam com a origem das idéias, e sim com o seu significado e o que elas são capazes de produzir. Então eu*

*proponho um 'Juramento de Não-Lealdade' (The Oath of Non-Allegiance):*

*Eu prometo não desconsiderar nenhuma idéia baseada em sua origem, mas considerar idéias vindas de todas as escolas e tradições com o objetivo de encontrar aquelas que melhor se encaixem em cada situação.*

*Isto significa o fim de afirmações como 'Isto não é bom – isto não é ágil puramente orientado a objeto, etc...', mas sim a discussão sobre se a idéia em questão (uma impura plan-driven ágil ou qual seja) funciona bem nas condições do momento. E eu serei o primeiro a assinar. [...]Cockburn(2010).*

Foi feita uma pesquisa sobre a Qualidade no Setor de Software Brasileiro no ano de 2009 pela Secretaria de Política e Informática do Ministério da Tecnologia, no qual um dos temas foi métricas para estimativa de tamanho de produto de software, onde responderam 264 organizações, o que representa 76,9% em relação ao total da amostra. Foram disponibilizadas questões de múltiplas escolha adicionado de uma campo aberto para informar outra métrica utilizada. O interessante é que 60 organizações não usam nenhuma métrica de software, no qual equivale a 22,7% de respondentes (BRASIL..., 2010, p 203).

De acordo com a tabela 1 a métrica mais utilizada pelas organizações respondentes é a Análise de Pontos de Função, com 91 organizações (34,5%). Com valores muito próximos, aparece a técnica de Pontos por Caso de Uso, com 89 organizações (33,7%).

**Tabela 1 - Distribuição das organizações de acordo com as métricas utilizadas para avaliação do tamanho do produto de software.**

MÉTRICAS UTILIZADAS PARA AVALIAÇÃO DO TAMANHO DO PRODUTO DE SOFTWARE	QUANTIDADE DE ORGANIZAÇÕES	% DAS ORGANIZAÇÕES
Linhas de código (LOC)	26	9,8%
Pontos por função ( <i>Function Points</i> )	91	34,5%
Pontos por Função Cheios ( <i>Full Function Points</i> )	7	2,7%

Continuação

<b>MÉTRICAS UTILIZADAS PARA AVALIAÇÃO DO TAMANHO DO PRODUTO DE SOFTWARE</b>	<b>QUANTIDADE DE ORGANIZAÇÕES</b>	<b>% DAS ORGANIZAÇÕES</b>
Pontos por Caso de Uso ( <i>Use Case Points</i> )	89	33,7%
<b>Outras</b>	67	25,5%
<b>Não utiliza</b>	60	22,7%

Fonte: BRASIL. Ministério da Ciência e Tecnologia (2010)

No total de 67 organizações responderam que utilizam outras métricas para estimar o tamanho do produto de software, foi disponibilizado um campo, de preenchimento livre, para que fosse possível registrar a métrica utilizada. Destas, 64 preencheram com o nome da métrica utilizada. Os resultados encontram-se detalhados nas figuras 3 e 4 a seguir. Como se pode observar, 14 organizações utilizam métricas relacionadas a métodos ágeis, como Story Points, Planning Poker, User Stories e Ideal Days. Isto representa 21,9% das organizações respondentes a este item.

**Tabela 2 - Distribuição das organizações de acordo com outras métricas utilizadas para estimar tamanho de produto de software.**

<b>OUTRAS MÉTRICAS UTILIZADAS PARA ESTIMAR TAMANHO DE PRODUTO DE SOFTWARE</b>	<b>QUANTIDADE DE ORGANIZAÇÕES</b>	<b>% DAS ORGANIZAÇÕES</b>
<b>Outros métodos (ex: número de domínios, número de programas, número de requisitos, pessoas-mês etc)</b>	21	39,8%
<b>Estimativas de métodos ágeis</b>	14	21,9%
<b>Método próprio</b>	9	14,1%
<b>Base histórica</b>	6	9,4%
<b>Funcionalidade</b>	5	7,8%
<b>Horas</b>	5	7,8%
<b>Delphi/ Especialistas</b>	4	6,3%

Fonte: BRASIL. Ministério da Ciência e Tecnologia (2010)

### **1.3. Objetivo**

Demonstrar como o uso de Pontos de Função poderia ser implementado em um ambiente Scrum, conhecido como uma das técnicas mais utilizadas em ambientes ágeis.

Adaptação do PO(Product Owner) na metodologia Scrum, designando um membro da equipe para ser o PO, adicionado das responsabilidades de coletar requisitos, criar artefatos e atender todas as necessidades do cliente.

### **1.4. Metodologia**

Foi utilizado nessa pesquisa o método qualitativo, cujo tema é a utilização da Análise de Ponto de Função em ambientes ágeis e adaptação das atribuições do PO na metodologia Scrum. Essa pesquisa envolveu levantamento bibliográfico, análise de jurisprudência, análise de dados e respostas ao questionário aplicado ao tema proposto.

Visto que o universo nesse setor é bastante amplo, foi delimitado seu escopo aos profissionais da Agência de Tecnologia da Informação de Pernambuco, Centro de Estudos e Sistemas Avançados do Recife – CESAR e fóruns de metodologia Scrum e APF, conseguindo dessa forma alcançar profissionais de empresas Privadas e Estatais, oriundos de vários Estados da Federação.

Os dados e informações que serviram de base para este trabalho foram obtidos das seguintes fontes:

- Levantamento bibliográfico – leitura, artigos e publicações na internet relativas à técnica;
- Levantamento de Regulamentações – estudo da Jurisprudência do Tribunal de Contas da União referente à técnica e Instrução Normativa do Ministério do Planejamento, Orçamento e Gestão;
- Pesquisa – Qualidade no Setor de Software Brasileiro 2009, Ministério da Ciência e Tecnologia;
- Pesquisa – elaboração de questionário sobre a utilização de Análise de Ponto de Função em ambientes ágeis e adaptação do PO na metodologia Scrum. Os questionários foram encaminhados para os órgãos participantes, através de

um link da ferramenta de coleta de dados Survey Monkey e as respostas foram colhidas nessa mesma ferramenta.

## **1.5. Estrutura**

Esse estudo está organizado em 6 (seis) seções. Na Seção 2 está definido os procedimentos para mensurar e estimar software com análise de pontos de função segundo as regras do IFPUG. A seção 3 é feita uma descrição das metodologias Ágeis e posterior aprofundamento no ciclo do Scrum. É descrito na seção 4 uma seção de trabalhos relacionados, já na seção 5 a proposta onde é implementada a APF no processo Scrum com mudanças no papel do Product Owner. A seguir na seção 6 é mostrada a validação dos processos propostos onde temos a coleta de dados que foi fundamental para a motivação dessa pesquisa. E por fim, na seção 7, conclusões e trabalhos futuros.

## **2. Análise de Pontos de Função**

Uma métrica de software é qualquer tipo de medição que se refira a um sistema de software, processo ou documentação relacionada, diz Sommerville (2011). Ao coletar essas medições, os questionamentos em relação ao tamanho do software e esforço para seu desenvolvimento, poderão ser respondidos. O conceito de métrica é “o número que você vincula a uma ideia(REF). Mais precisamente, é uma indicação mensurável de algum aspecto quantitativo do sistema. Para o projeto de um software com poucos requisitos funcionais, os aspectos quantitativos onde mais precisamos usar a métrica são: escopo, tamanho, custo, risco e tempo empregado” (DEMARCO, 1989).

A métrica de ponto-por-função (ou ponto de característica), como as linhas de código (LOC), é controversa. Os proponentes afirmam que function point (FP) independe da linguagem de programação, tornando-o ideal, para aplicações que usam linguagem convencionais e não-procedimentais; afirmam também que ele se baseia em dados que têm maior probabilidade de ser conhecidos logo no começo da evolução de um projeto, tornando o FP mais atraente como abordagem de estimativa. Os opositores afirmam que o método requer certa "prestidigitação, no sentido de que a contagem se baseia parcialmente em dados subjetivos, não objetivos; que os dados sobre o domínio

da informação podem ser difíceis de ser identificados a posterior; e que o FP não tem nenhum significado físico direto - é apenas um número" (PRESSMAN, 2011).

Segundo o International Function Point Users Group(2010) a métrica Pontos de Função é uma medida de tamanho funcional de projetos de software, considerando as funcionalidades implementadas, sob o ponto de vista do usuário. Tamanho funcional é definido como “tamanho do software derivado pela quantificação dos requisitos funcionais do usuário” (DEKKERS, 2003).

Ponto de função (PF) é a sua unidade de medida, na qual tem como prioridade mensurar software independente da tecnologia utilizada. Ou seja, a APF busca medir o que o software faz, e não como ele foi construído aponta Hazan (2008).

Portanto, o processo de medição (também chamado contagem de pontos de função) é baseado em uma avaliação padronizada dos requisitos funcionais do usuário. As principais técnicas de estimativa de projetos de desenvolvimento de software assumem que o tamanho de um software é um vetor importante para a determinação do esforço para sua construção. Logo, saber o seu tamanho é um dos primeiros passos do processo de estimativa de esforço, prazo e custo.

Daí, é importante destacar que pontos de função não medem diretamente esforço, produtividade ou custo. É exclusivamente uma medida de tamanho funcional do software. Este tamanho, em conjunto com outras variáveis, é que poderá ser usado para derivar produtividade, estimar esforço e custo do projeto de software.

## **2.1. Regulamentação**

De acordo com Albrecht(1979) a APF teve início na IBM, onde existia a necessidade de mensurar produtividade de software. Foi feito um estudo em meados de 1974 ao início de 1979, durante esse período foi validado cada estimativa como uma série de questões ponderadas sobre as funcionalidades da aplicação e o ambiente de desenvolvimento. Essas contagens são ponderadas pelos números projetados para refletir o valor da função para o cliente.

Após o seu surgimento a APF se propagou rapidamente entre os profissionais de TI e, inicialmente foi criado em 1986 o IFPUG (International Function Point Users Group), uma organização não governamental para a análise, produção e difusão da



técnica que virou medida oficial de produção de softwares em 2000, quando padronizada através da norma da ISO (20926). Responsável pela atualização das regras de contagem de pontos de função, descritas no CPM (Counting Practices Manual), que se encontra na versão 4.3.1, publicada em janeiro 2010 no IFPUG (INTERNATIONAL FUNCTION POINT USERS GROUP, 2010). Desta forma, Pontos de Funções caminha para ser uma das principais base para medir a qualidade e a eficiência dos aplicativos produzidos, pois se tornou claro que a quantidade de informações produzidas em TI não é peça fundamental para o sucesso de um software.

Nesta técnica, a pontuação obtida em cada função do aplicativo é medida não pelos desenvolvedores, mas pelo próprio cliente, proporcionando um feedback contínuo entre quem utiliza e que cria determinados softwares, pois independentemente da perfeição técnica de um programa ele tornar-se-ia inútil caso não atendesse as expectativas para qual foi criado.

## **2.2. Benefícios da APF**

Pode-se enumerar os principais objetivos da APF, que segundo o IFPUG(2010), são:

- Medir a funcionalidade que o usuário solicita e recebe;
- Medir o desenvolvimento e a manutenção de software, independentemente da tecnologia utilizada para implementação;
- Prover uma métrica de medição para apoiar a análise de produtividade e qualidade;
- Prover uma forma de estimar o tamanho de software;
- Prover um fator de normalização para comparação de software.

Além de atingir os objetivos acima, o processo de contagem de ponto de função deve ser:

- Suficientemente simples para minimizar a sobrecarga no processo de medição;
- Uma medida consistente entre os vários projetos e organizações.

Podem-se destacar vários benefícios da aplicação da análise de pontos de função nas organizações:

1. Uma ferramenta para determinar o tamanho de um pacote de software adquirido através da contagem de todas as funções incluídas no pacote (IFPUG, 2010);
2. Suporta a análise de produtividade e qualidade (VAZQUEZ, 2010);
3. Apóia o gerenciamento de escopo de projetos controlando o “scopecreep<sup>1</sup>” ou o aumento de seu escopo determinando se os requisitos funcionais cresceram ou diminuíram e se essas mudanças estão relacionadas aos novos requisitos (VAZQUEZ, 2010);
4. Complementa o gerenciamento dos requisitos ao auxiliar na verificação da solidez e completeza dos requisitos especificados (VAZQUEZ, 2010);
5. Um meio para estimar custo e recursos requeridos para o desenvolvimento e manutenção do software (IFPUG, 2010);
6. Uma ferramenta para fundamentar a negociação de contratos (VAZQUEZ, 2010);
7. Um fator de normalização para comparação de software ou para comparação da produtividade na utilização de diferentes técnicas (VAZQUEZ, 2010).

### 2.3. Procedimentos da contagem de pontos de função

Para definir a pontuação final de um software, dentro das expectativas do cliente, a técnica de pontos de funções analisa cinco pontos fundamentais: os arquivos lógicos internos, os arquivos interface externa, entradas externas, consultas externas e saídas externas, cada um desses requisitos é mapeado com uma função de negócio do usuário, para determinar a categoria a qual o aplicativo se ajusta. Veja exemplo de um sistema de controle de ponto na tabela 3.

**Tabela 3 – Funções do sistema e Tipo de Transações**

Descrição	Tipo
Login	Saída Externa
Registro de Ponto	Entrada Externa

---

<sup>1</sup> Refere-se a fornecer escopo adicional, não solicitado, não definido e aprovado.

Consulta Apontamento Diário	Consulta Externa
Tabelas que armazenam dados pela aplicação	Arquivos Lógicos Internos
Tabelas de outra aplicação	Arquivos Interface Externa

Fonte: VAZQUEZ (2011)

#### 2.4. Determinar o tipo de contagem dos pontos de função

Segundo o Manual de Contagem do IFPUG o primeiro passo é determinar o tipo de contagem de pontos de função. A contagem de pontos de função pode estar associada tanto a projetos quanto a aplicações. Existem três tipos de contagem de pontos de função:

- Contagem de pontos de função de projeto de desenvolvimento;

O número de pontos de função de um projeto de desenvolvimento mede a funcionalidade fornecida aos usuários finais do software quando da sua primeira instalação (VAZQUEZ, 2010).

- Contagem de pontos de função de projeto de melhoria;

O número de pontos de função de um projeto de melhoria mede as funções adicionadas, modificadas ou excluídas do sistema pelo projeto, e também as eventuais funções de conversão de dados (IFPUG, 2010).

- Contagem de pontos de função de aplicação.

O número de pontos de função de uma aplicação mede a funcionalidade fornecida aos usuários por uma aplicação instalada. Esse número fornece uma medida da atual funcionalidade obtida pelo usuário da aplicação (VAZQUEZ, 2010).

#### 2.5. Definição da fronteira da aplicação

Este é o período em que é firmado o escopo do sistema, sob a visão do usuário. De acordo com o Manual de Contagem do IFPUG o escopo da contagem define a funcionalidade que será incluída em uma determinada contagem de pontos de função. A

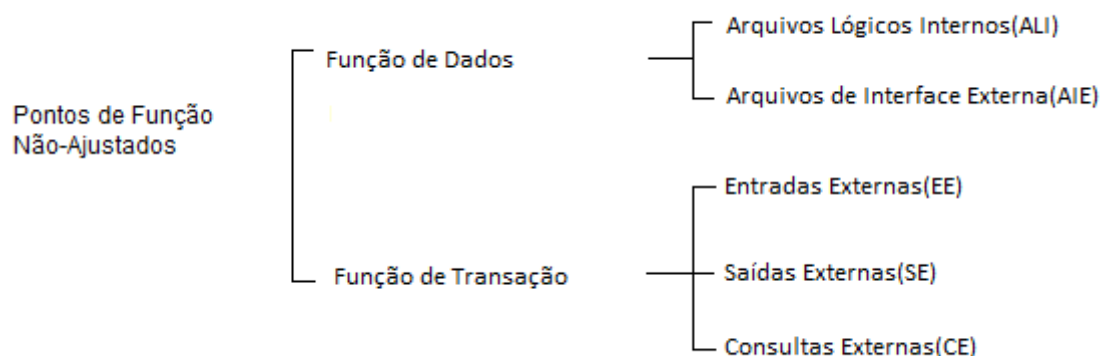
fronteira da aplicação indica os limites entre o sistema que está sendo medido e o usuário.

## 2.6. Contagem de pontos de função não ajustados

Retrata o conjunto de funções específicas do usuário. Na opinião de Vazquez (2010), o termo “específicos” é no sentido de que é possível apontar um requisito (um relatório, um gráfico, uma transação de entrada de dados etc.) e dizer o seu valor em PF.

Os pontos de função não-ajustados possuem dois tipos de funções – de dados e de transação. Estes tipos de funções são definidos como mostra a figura 1.

**Figura 1. Tipos de pontos de função não-ajustados**

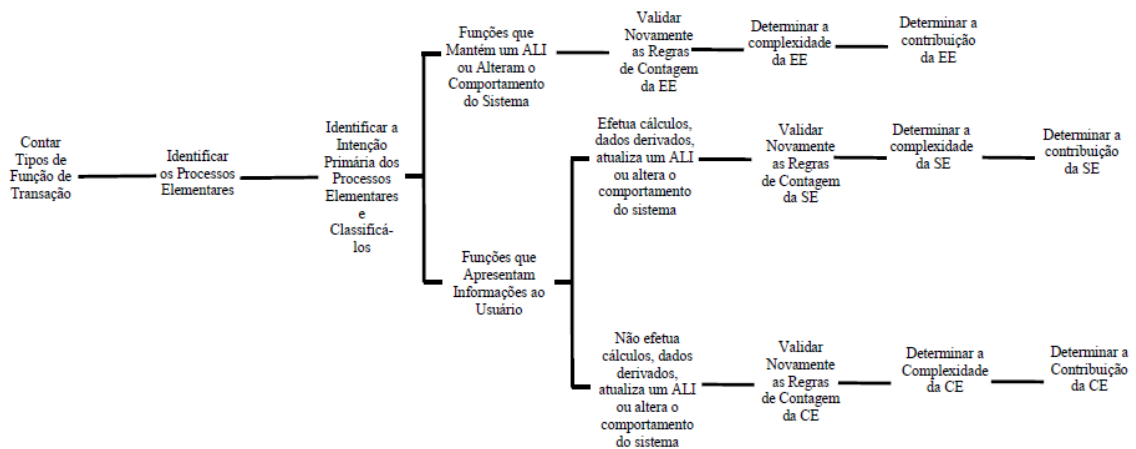


Fonte: Manual de Práticas de Contagem de Pontos de Função Versão 4.3.1 (IFPUG, 2010)

## 2.7. Normas para a contagem de Pontos de Função Bruto

Feita a definição de qual tipo de contagem será realizada, será identificada as fronteiras da aplicação, e a identificação dos grupos de funções tipo dados e transações. O procedimento para contagem de PFs compreende os sete passos mostrados a seguir na figura 2, segundo Hazan (1999).

**Figura 2 – Procedimento para contagem de EE, SE e CE**



Nota: Manual de Práticas de Contagem de Pontos de Função Versão 4.3.1 (IFPUG, 2010)

## 2.8. Contagem das funções de dados

As funções de dados representam as funcionalidades relativas aos requisitos de dados internos e externos à aplicação. São os arquivos lógicos internos e os arquivos de interface externa. Ambos são grupos de dados logicamente relacionados ou informações de controle que foram identificados pelo usuário. A diferença está no fato de um Arquivo Lógico Interno (ALI) ser mantido dentro da fronteira da aplicação, isto é, armazenar os dados mantidos através de um ou mais processos elementares da aplicação, enquanto que um Arquivo de Interface Externa (AIE) é apenas referenciado pela aplicação, ou seja, ele é mantido dentro da fronteira de outra aplicação. Assim, o objetivo de um AIE é armazenar os dados referenciados por um ou mais processos elementares da aplicação sendo contada, mas que são mantidos por outras aplicações.

## 2.9. Contagem das funções transacionais

As funções de transação representam a funcionalidade fornecida ao usuário para o processamento de dados por uma aplicação. Essas funções são definidas como entradas externas (EEs), saídas externas (SEs) e consultas externas (CEs) (MANUAL DE PRÁTICAS DE CONTAGEM DE PONTOS DE FUNÇÃO VERSÃO 4.3.1, 2010)

As Entradas Externas (EEs) são processos elementares que processam dados (ou informações de controle) que entram pela fronteira da aplicação. O objetivo principal de uma EE é manter um ou mais ALIs ou alterar o comportamento do sistema.

As Saídas Externas (SEs) são processos elementares que enviam dados (ou informações de controle) para fora da fronteira da aplicação. Seu objetivo é mostrar informações recuperadas através de um processamento lógico (isto é, que envolva cálculos ou criação de dados derivados) e não apenas uma simples recuperação de dados. Uma SE pode, também, manter um ALI ou alterar o comportamento do sistema.

A Consulta Externa (CE), assim como uma SE, é um processo elementar que envia dados (ou informações de controle) para fora da fronteira da aplicação, mas sem realização de nenhum cálculo nem a criação de dados derivados. Seu objetivo é apresentar informação para o usuário, por meio apenas de uma recuperação das informações. Nenhum ALI é mantido durante sua realização, nem o comportamento do sistema é alterado.

O procedimento de contagem de EE, SE e CE incluem os seguintes passos:

1. Identificar os processos elementares;
  - O processo é a menor unidade de atividade significativa para o usuário
  - O processo é autocontido e deixa o negócio da aplicação em um estado consistente.
2. Determinar a intenção primária dos processos elementares identificados, e classifica-los como EE, SE ou CE;
3. Validar de acordo com as regras de identificação das transações (EE, SE, CE).
4. Determinar a complexidade das transações(EE, SE, CE).
5. Determinar a contribuição das transações (EE, SE, CE) para a contagem de pontos de função não ajustados.

## 2.10. Determinar o valor do fator de ajuste

O fator de ajuste é baseado em 14 características gerais de sistemas listadas na tabela 4 adaptada de IFPUG (2010), que avaliam a funcionalidade geral da aplicação que está sendo contada, e seus níveis de influência. O nível de influência de uma característica é determinado com base em uma escala de 0 (nenhuma influência) a 5 (forte influência).

**Tabela 4 – As 14 características gerais na aplicação do fator de ajuste**

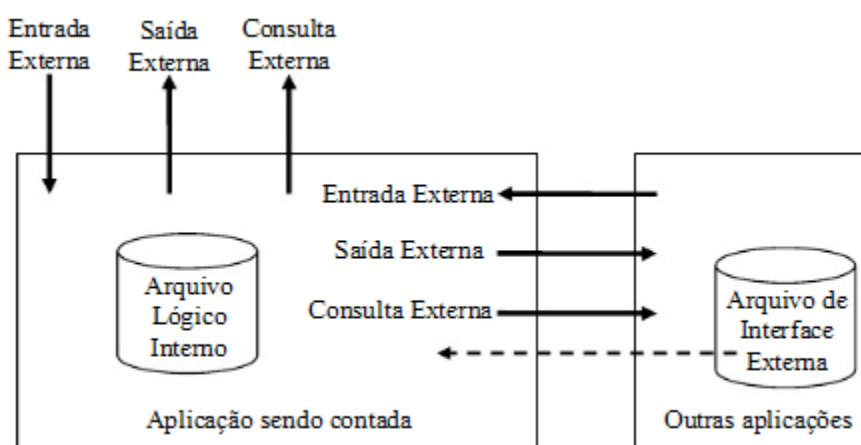
1. Comunicação de dados	8. Atualização On-Line
2. Processamento Distribuído	9. Processamento Complexo
3. Performance	10. Reusabilidade
4. Configuração Intensamente Utilizada	11. Facilidade de Instalação
5. Volume de Transações	12. Facilidade de Operação
6. Entrada de Dados On-Line	13. Múltiplos Locais
7. Eficiência do Usuário Final	14. Facilidades de Mudanças

Fonte: Tabela adaptada de IFPUG (2010)

## 2.11. Calcular os pontos de função ajustados

Finalmente, os PFs ajustados são calculados, considerando-se o tipo de contagem definido no primeiro passo. A figura 3 apresenta uma visão geral dos tipos de função que são considerados na contagem da APF.

**Figura 3 - Visão Geral das Funções de uma Aplicação segundo a APF (Hazan)**



Fonte: Hazan (1999)

## 2.12. Determinação da complexidade

Cada entrada externa, saída externa e consulta externa deve ser classificada com relação à sua complexidade funcional (baixa, média ou alta), baseado em:

- Numero de Arquivos Referenciados (AR)
- Número de Tipos de Dados (TD)

Determinadas as quantidades de arquivos referenciados e de tipos de dados, a classificação com relação à complexidade é fornecida pelas seguintes tabelas representadas nas tabelas 5 e 6 VAZQUEZ (2011).

**Tabela 5 - Complexidade para entradas externas (EEs)**

Arquivos Referenciados (ARS)	Tipos de Dados(TDs)		
	<5	5-15	>15
<2	Baixa	Baixa	Média
2	Baixa	Média	Alta
>3	Média	Alta	Alta

Fonte: VAZQUEZ (2011)

**Tabela 6 - Complexidade para saídas externas (SEs) e consultas externas (CEs)**

Arquivos Referenciados (ARS)	Tipos de Dados(TDs)		
	<6	6-19	>19
<2	Baixa	Baixa	Média
2-	Baixa	Média	Alta
>3	Média	Alta	Alta

Fonte: VAZQUEZ (2011)

Assim, com o estabelecimento de uma contagem objetiva, os Pontos de Função formam uma métrica padronizada servido de base comparativa entre projetos diferentes e de diferentes desenvolvedores, formando softwares com “tamanhos” comparáveis. Dando ao termo “tamanho” um largo conteúdo quantitativo determinante para o sucesso do software e a satisfação do cliente.

Veja exemplo de um cálculo das funções do tipo de transição de um sistema de controle de ponto na tabela 7.



**Tabela 7 – Cálculo das funções do tipo de transição**

<b>Tipo de Função</b>	<b>Complexidade Funcional</b>	<b>Totais por Tipo de Complexidade</b>	<b>Totais por Tipo de Função</b>
EE	<u>2</u> Baixa x 3	= 6	14
	<u>2</u> Média x 4	= 8	
	<u>0</u> Alta x 6	= 0	
SE	<u>1</u> Baixa x 4	= 4	14
	<u>2</u> Média x 5	= 10	
	<u>0</u> Alta x 7	= 0	
CE	<u>1</u> Baixa x 3	= 3	3
	<u>0</u> Média x 4	= 0	
	<u>0</u> Média x 6	= 0	

Fonte: VAZQUEZ (2011)

### 3. Métodos ágeis

Inicialmente, métodos ágeis eram conhecidos como métodos leves (TANIGUCHI, 2009). Em 2001 um grupo de pensadores, engenheiros e autores lançaram o Manifesto ágil<sup>2</sup>, com a finalidade e intuito de apresentar e discutir as melhores práticas de desenvolvimento de software, aponta Boehm(*s.d.*).

Definir um processo como “ágil” significa permitir que a equipe possa adaptar e aperfeiçoar tarefas, manter apenas os trabalhos essenciais implementados, simplificar os trabalhos essenciais, fazer entregas incrementais do software funcionando ao cliente o mais rápido possível (PRESSMAN, 2011).

Com o surgimento dos métodos ágeis, quebraram-se paradigmas dos métodos tradicionais de desenvolvimento de software, propondo uma nova abordagem. Esses métodos ágeis procuram a melhor solução para gerenciar as mudanças, possibilitando um maior poder de reação das equipes. Sendo assim, métodos ágeis como XP (Extreme Programming), Scrum e Crystal ganharam grande destaque (GOLDMAN, 2004). Esses métodos empregam princípios ágeis, tais como ciclos iterativos, entrega rápida de software funcionando e simplicidade.

---

<sup>2</sup> Conjunto de valores e princípios com o intuito de criar melhorias no desenvolvimento de software.

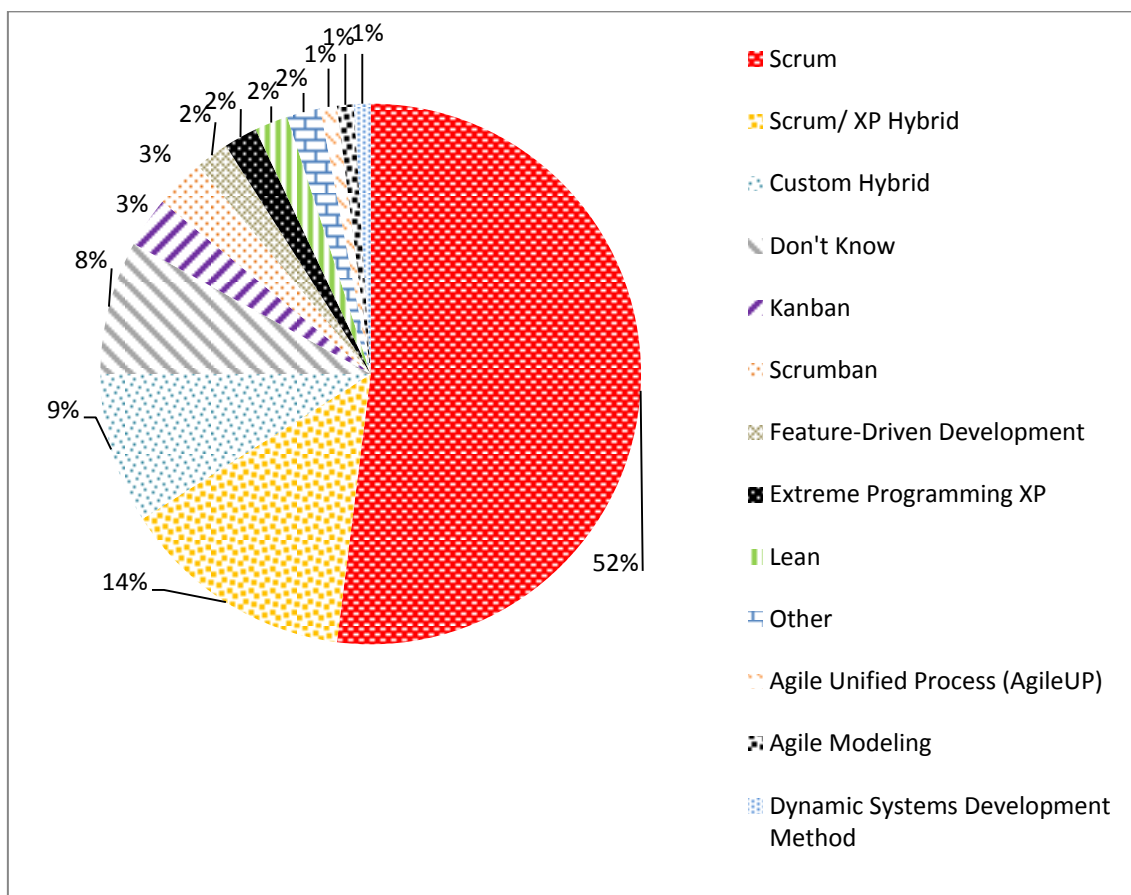
O movimento ágil marcou o surgimento de uma nova disciplina de engenharia que modificava os valores do processo de desenvolvimento de software do mecânico (orientado a processos e utilização de regras da ciência) para o orgânico (dirigido por questões sobre pessoas e suas interações). Isso implicava um desafio para a Engenharia de Software e para sistemas em ambiente de trabalho que são dinâmicos e imprevisíveis (MNKANDLA, 2004).

Em pesquisa realizada pela VersionOne<sup>3</sup> em 2011, onde foram recebidas 6.042 respostas de profissionais da indústria de software, foi constatado que apesar do pequeno aumento no apoio à gestão de ágil, os resultados da pesquisa sugerem que a maior barreira para a adoção crescente do ágil não parece ser a consciência da metodologia, mas as culturas internas das empresas. Apenas 13% dos entrevistados eram de grandes empresas, 27% dos entrevistados disseram que a falta de apoio à gestão e "resistência geral à mudança" com 26% foram as principais barreiras para a adoção ágil. Ainda na pesquisa da VersionOne, foi constatado 52% de adoção ao Framework Scrum em relação ao demais métodos, conforme figura 4.

---

<sup>3</sup> Empresa de desenvolvimento de software, pioneira no planejamento de projetos ágil e gestão.

**Figura 4. Metodologias ágeis mais usadas**



Fonte: VersionOne (2011)

Neste trabalho o método ágil a ser abordado será o Scrum, pois este vem sendo o framework mais utilizado conforme pesquisa da VersionOne. Assim como nos demais ambientes ágeis a preocupação com a qualidade recai, necessariamente, no ponto de vista do cliente, será observado com uma breve descrição do framework.

### 3.1. Scrum

Como caracteriza Rising (2000) o Scrum é um framework de realização de tarefas desenvolvidos para otimizar e facilitar o trabalho de uma equipe de produção, promovendo jornadas de produção e avaliação, sempre integrando os produtores ao destinatário final para que esse não apenas receba o produto pronto, mas participe de sua criação corrigindo defeitos e garantindo a eficiência do projeto inicial.

O Scrum é um framework ágil de desenvolvimento de software, seu nome está relacionado a uma atividade que ocorre durante o jogo de rugby<sup>4</sup> onde o grupo de jogadores se juntam ao redor da bola e trabalham juntos para mover a bola pelo campo (SCHWABER, 2004). Os princípios do Scrum são compatíveis com o manifesto ágil: utiliza-se pequenas equipes, o processo é adaptado para garantir que o melhor produto seja produzido, incrementos de software são produzidos, o trabalho de desenvolvimento é dividido entre a equipe e testes são realizados durante todo o processo. Esses princípios norteiam o processo de desenvolvimento de software que tem as seguintes atividades: levantamento de requisitos, projeto, evolução e entrega (SCHWABER, 2004).

### **3.1.1. Papéis e responsabilidades**

O Scrum implementa um framework iterativo e incremental cujas atividades são assumidas por três papéis principais: Product Owner, ScrumMaster e Time (SCHWABER, 2004).

O Product Owner representa a voz do cliente e é responsável pelos interesses de todos no projeto, define os fundamentos do projeto definindo requisitos iniciais e gerais (Product Backlog), retorno do investimento, objetivos e planos de entregas, prioriza o Product Backlog a cada sprint, garantindo que as funcionalidades de maior valor sejam construídas prioritariamente. Schwaber (2004) relata que cada sprint se inicia com uma reunião de planejamento (*Sprint Planning Meeting*), na qual o Product Owner e o time decidem em conjunto o que deverá ser implementado (*Selected Product Backlog*).

O ScrumMaster gerencia o processo do Scrum garantindo que o processo Scrum seja usado como pretendido, removendo qualquer tipo de impedimento que apareça durante o processo. O ScrumMaster é o responsável pela aplicação das regras. Uma parte fundamental do papel do ScrumMaster é proteger a equipe e mantê-la focada nas tarefas em mãos.

O Time desenvolve as funcionalidades do produto, define como transformar o Product Backlog em incremento de funcionalidades numa sprint gerenciando seu

---

<sup>4</sup> Esporte coletivo originado na Inglaterra de intenso contato físico

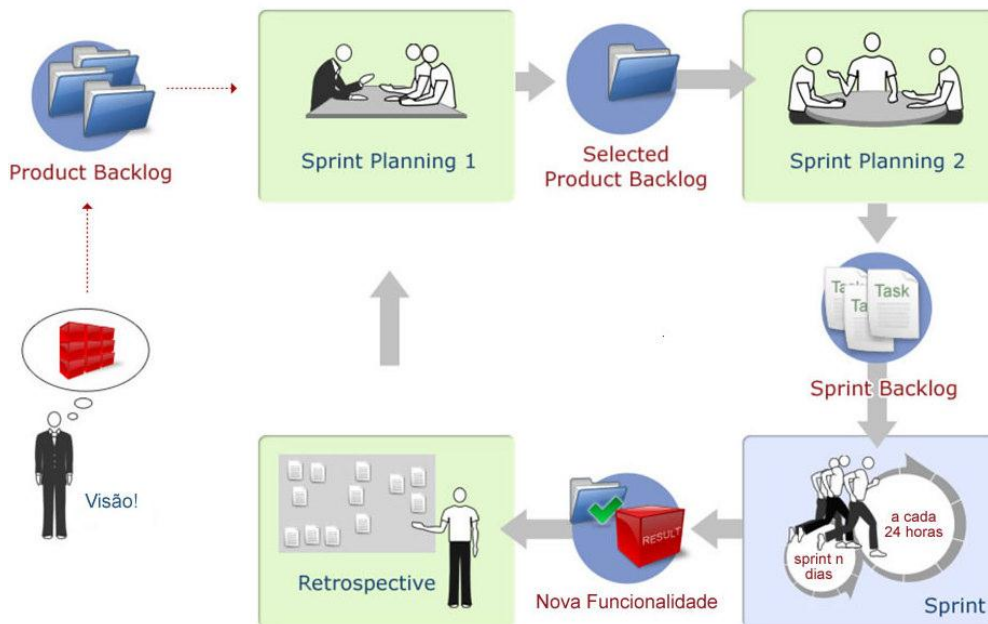
próprio trabalho. O time é responsável coletivamente pelo sucesso da *sprint* e conseqüentemente pelo projeto como um todo.

### **3.1.2. Fluxo de um processo Scrum**

A jornada no Scrum tem início em iterações chamadas de sprints. A reunião é dividida em duas partes, na primeira parte (Sprint Planning 1), o Product Owner apresenta os requisitos de maior valor e prioriza aqueles que devem ser implementados. O time então define colaborativamente o que poderá entrar no desenvolvimento da próxima sprint, considerando sua capacidade de produção. Na segunda parte (Sprint Planning 2), o time planeja seu trabalho, definindo o Sprint Backlog, que são as tarefas necessárias para implementar as funcionalidades selecionadas a partir do Product Backlog. Nas primeiras sprints, é realizada a maioria dos trabalhos de arquitetura e de infra-estrutura. A lista de tarefas pode ser modificada ao longo da sprint pelo time e as tarefas podem variar entre 4 a 16 horas para a sua conclusão.

Um projeto no Scrum se inicia com uma visão do produto que será desenvolvido (SCHWABER, 2004). Contém todos os detalhes do produto fornecidos pelo cliente adicionado de premissas e restrições. Em seguida é criado o Produto Backlog, contendo todos os requisitos definidos pelo o usuário e desta forma é priorizado e dividido em releases. O fluxo de desenvolvimento detalhado do Scrum é mostrado na figura 5, adaptada de Gloger (2007).

**Figura 5. Visão geral do processo do Scrum**



Fonte: Adaptada de Gloger (2007).

### 3.1.3. Artefatos

De acordo com Larman (2003), o Scrum introduz os seguintes principais artefatos usados ao longo do seu fluxo de desenvolvimento: Product Backlog, Sprint Backlog, Burndown do Sprint e Taskboard.

#### 3.1.3.1. Product Backlog

Conforme ressalta Marçal o Product Backlog é uma lista contendo todas as funcionalidades desejadas para um produto. O conteúdo desta lista é definido e atualizado pelo Product Owner sempre que necessário. O Product Backlog não precisa estar completo no início de um projeto. Os itens com maior prioridade serão selecionados para o próximo sprint. Qualquer membro do time pode adicionar ou remover itens com consentimento do Product Owner. Quanto menos prioritários mais abstratos são os itens. Com o tempo, o Product Backlog cresce e muda à medida que se aprende mais sobre o produto e seus usuários.

### 3.1.3.2. Sprint Backlog

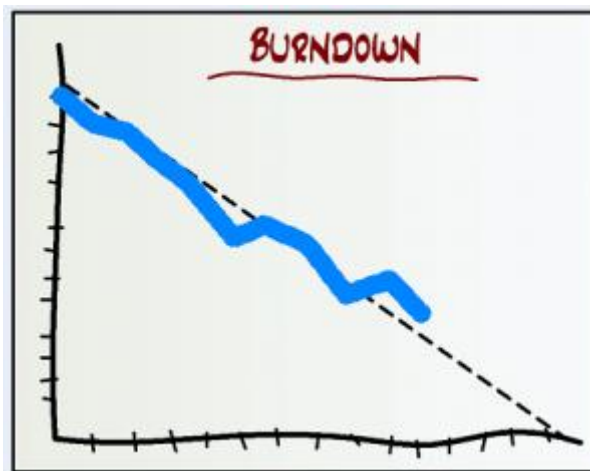
O Sprint Backlog é uma lista de tarefas que o Scrum Team se compromete a fazer em um Sprint. Os itens do Sprint Backlog são extraídos do Product Backlog, pela equipe, com base nas prioridades definidas pelo Product Owner e a percepção da equipe sobre o tempo que será necessário para completar as várias funcionalidades. O Sprint Backlog apresenta várias características entre elas pode-se citar:

- Cada tarefa identifica o responsável que irá trabalhar sobre ela e o restante do tempo estimado para terminá-la em horas;
- Tarefas devem estar organizadas para que estejam concluídas em 4 a 16 horas de trabalho;
- Tarefas maiores são consideradas placeholders;
- Não se pode incluir novas atividades durante o andamento do sprint;
- Apenas o time pode modificá-la.

### 3.1.3.3. Burndown do sprint

É o principal gráfico de controle do Scrum. Representa o trabalho total restante dentro de um sprint, de um release ou produto, a origem dos dados para criar este gráfico é o sprint backlog como é mostrado na figura 6.

Figura 6. Gráfico do Burndwn

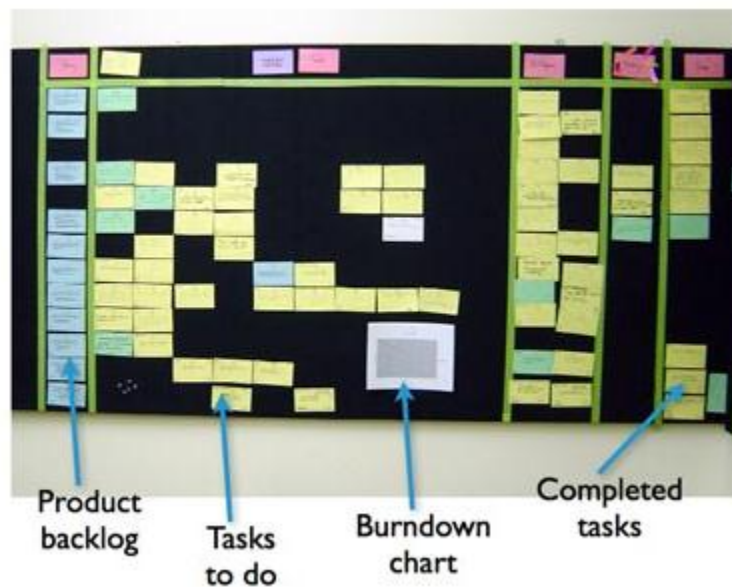


Fonte: Agileway (s.d)

### 3.1.3.4. Taskboard

O taskboard é um grande painel onde podem ser colocadas várias informações importantes para o acompanhamento do sprint. No sprint backlog, as atividades concluídas e o andamento das atividades ficam sempre visíveis e disponíveis para todos os interessados no projeto, como mostra a figura 7.

Figura 7. Taskboard



Fonte: MOUNTAIN (2010)

Podem-se citar como as principais características do taskboard:

- Normalmente é desenhado em uma parede e as atividades são descritas em post-its;
- Apresenta uma visão geral do sprint;
- Fica acessível a todos os interessados no projeto.

### 3.1.4. Fases do Scrum

O Scrum possui um ciclo de vida composto por quatro fases como definido em (LARMAN, 2003):

- **Planejamento:** que tem por objetivo estabelecer a visão do projeto e as expectativas entre os *stakeholders*, além de garantir financiamento/orçamento para a realização do projeto;



- **Preparação:** que tem por objetivo identificar os requisitos e priorizá-los (pelo menos para a próxima *sprint*). Divide o *Product Backlog* em *sprints*, de acordo com a priorização realizada, levando em consideração a produtividade do time;
- **Desenvolvimento:** corresponde à implementação do sistema em uma série de sprints de 30 dias consecutivos (*sprints*), com apresentação de um incremento de funcionalidade ao final da *sprint*;
- **Entrega:** corresponde implantação operacional do sistema.

#### 4. Trabalhos Relacionados

O estado da arte em PF implementado ao Scrum envolve poucas referências, pois é um processo novo, onde as empresas ainda estão se adaptando. Essa seção tem como finalidade apresentar alguns trabalhos e projetos de pesquisa relacionando com APF e Scrum.

##### 4.1 Aplicação de Métricas com Scrum

Esse estudo focou na aplicação de métricas de software para obter estimativas de prazo de tempo de desenvolvimento de uma iteração do método Scrum.

Foi realizado um estudo de caso para validar as métricas definidas e posteriormente uma análise. Com o intuito de aumentar a precisão do tempo de desenvolvimento no Scrum, onde foram utilizadas as métricas Ideal Day, Planning Poker e Pontos de Função.

Entre as métricas pesquisadas, na de PF foi identificado dificuldades na coleta de requisitos, visto que Scrum ser um metodologia ágil e PF não ser uma métrica de estimativa ágil.

Foi identificado na primeira sprint que a métrica de PF teve a maior discrepância comparado ao tempo total do projeto, pois foi analisado um sistema web onde não foi possível analisar a complexidade do projeto.

##### 4.2 Uso de análise de Pontos de Função em Ambientes Ágeis

Dentro dessa perspectiva foi feita uma análise entre as técnicas de Pontos de Função e Pontos por Estória, onde ambos possuem o mesmo objetivo comum de medir a

quantidade de valor de negócio agregado ao sistema ou a pedaços dele. Onde, a diferença de cultura que envolve estas abordagens as torna duas formas diferentes para avaliar o mesmo indicador dentro do desenvolvimento de software.

Foi identificado que Pontos de Função se aproxima mais da realidade financeira dos projetos, consequentemente tornando-se eficiente no controle de custo e produtividade das equipes.

Já o Ponto por Estória se mostrou ser uma técnica mais eficiente no acompanhamento do projeto estimativa de tempo das estórias.

### **4.3 Usando Pontos de Função em Projetos Ágeis**

Este é o estudo que mais se aproxima da proposta, é um processo aplicado na PETROBRAS, por Carlos Henrique Oest, Consultor da Petrobrás, reforça o que temos dito com respeito a aceitação crescente de metodologias ágeis em grandes projetos, e como elas podem ser combinadas com outros aparatos, como a APF.

Seu processo é agrupado em quatro grandes grupos:

- Na primeira fase é descrita a demanda em macrorequisitos para que seja estimado prazo e custo de desenvolvimento. Com estes dados ela é aprovada pelo cliente;
- Sendo assim na segunda fase a demanda que foi aprovada é detalhada em casos de uso, regras de negocio, glossário, requisitos não funcionais. Novamente a demanda é contada e estimados prazo e custo do desenvolvimento.
- Já na fase três, é aplicado o Scrum e algumas práticas do XP. Na primeira reunião é apresentado a equipe o Product Owner (representante do cliente, pode ser o próprio), que faz uma explanação geral dos casos de uso (os casos de uso substituem as histórias do scrum)e define a prioridade.

Por fim é feita a segunda parte da reunião a equipe se reúne e faz as estimativa dos “Story Points” em uma sessão de “planning poker. Não precisam ser analisadas todas as histórias, apenas o numero necessário para preencher um ou dois sprints, novas sessões são feitas nas reuniões de início de sprint, as histórias são analisadas em ordem de prioridade do cliente O valor do story point tem

relação com o esforço de desenvolvimento e não com o tamanho, sendo mais adequado ao micro controle do processo, que é feito através dos “Burndown Chart durante os sprint. As histórias que vão ser desenvolvidas em um sprint são escolhidas a partir de sua prioridade, do esforço de desenvolvimento em story points e na velocidade de desenvolvimento da equipe (em story points). O time box dos nossos sprints têm duração de 30 dias.

- Após ao final da última sprint a equipe de inspeção e a equipe de teste avaliam o software e os documentos gerados necessários a futuras manutenções. O software passa para homologação (sempre ocorrem solicitações de pequenas alterações) e por fim colocado em produção, e uma nova contagem de PF é feita.

Os indicadores de esforço (produtividade), taxa de entrega (velocidade), densidade de defeitos, eficácia de prazo e eficácia de custo, são calculados em função desta última contagem.

Também foi identificado que a taxa de entrega(PF/dia útil) passou de uma faixa que varia de 1,6 a 2,2 para uma faixa de 2,5 a 3,5.

Das diferenças mais contundentes entre o processo aplicado na PETROBRAS e a proposta são:

- A adaptação dos papéis do PO, visto que ele segue o que prega o Scrum, uma segunda estimativa é realizada antes de implantar o Scrum, onde na proposta p Scrum é implantado desde o início do projeto com apenas uma macro estimativa.
- O Time box tem duração de 30 dias e só são aceitos mudanças no início da Sprint, por sua vez a proposta expõe uma Sprint de 15, visto que sprints muito longas dificultam as estimativas onde também as mudanças de requisitos são aceitas até a metade da Sprint, desde que essa mudança não tenha grande impacto no desenvolvimento do projeto.
- Os indicadores de esforço (produtividade) e taxa de entrega (velocidade) são mensurados apenas no final do projeto, porém na proposta dessa pesquisa esses indicadores são mensurados ao final de cada final de cada Sprint e contado os PF das estórias concluídas.

#### **4.4 The machine that changed the world**

Onde o autor explica que o principal elemento no desenvolvimento de carro na Toyota é o Shusa, uma espécie de super homen, o engenheiro chefe com excelência técnica, que mistura poderes de gestão a uma visão de mercado excepcional, para ser responsável pelo sucesso do produto.

Toda responsabilidade do projeto é dele, o sucesso ou fracasso é direcionando a ele, é quem decide quais features<sup>5</sup> que o carro terá e também qual será o seu custo de mercado. O fato do shusa ser multidisciplinar, facilita a decisão rápida na hora de colocar na balança o custo de se implementar determinado requisito funcional versus o seu valor de negócio.

A Toyota prepara o shusa ao longo de 10 anos, o mesmo passa um grande período como engenheiro sênior, para em seguida começar a ser moldado nas área de gestão, marketing e etc, para depois virar o shusa.

Portanto de acordo com a proposta dessa pesquisa as mudanças no papel do PO, onde é proposto que um membro da equipe com vasta experiência técnica a suma a função de PO, igualmente as atribuições do shusa.

### **5. Proposta do uso da APF no Scrum**

Como as empresas públicas brasileiras já estão utilizando a métrica de Pontos de Função para mensurar tamanho de produto de software para contratação, o Estado de Pernambuco também aderiu a essa modalidade (PERNAMBUCO, 2010). Sendo assim, as empresas interessadas em possuírem contratos com o Governo do Estado e que têm um processo ágil, terão que adaptá-lo para receber a Análise de Pontos de Função. Portanto, foi proposto a implementação da APF no processo do Scrum, com o intuito de atender essa demanda.

#### **5.1. Caracterização da empresa**

Para dar andamento ao processo de validação dessa proposta está sendo utilizado uma pesquisa quantitativa, na BankSystem Software Builder, empresa de desenvolvimento

---

<sup>5</sup> Características

de software e atuante no mercado desde 2005, especializada no desenvolvimento de sistemas, na qual foi vencedora do Pregão Eletrônico nº 011/2010, Registro nº: Registrado sob o Nº 005/10 às folhas Nº 63 do livro Nº 01 de Registro de Contratos e Convênios Diversos, onde será responsável pelo desenvolvimento de software para todas as secretarias do estado de Pernambuco. (BANKY SYSTEM SOFTWARE BUILDER, 2010).

No qual o objeto registrado segue a recomendação do governo federal para que as instituições públicas optem pelo contrato de serviços com medição de produtividade em vez de contratos de mão de obra (BRASIL..., 1995).

## **5.2. Processo**

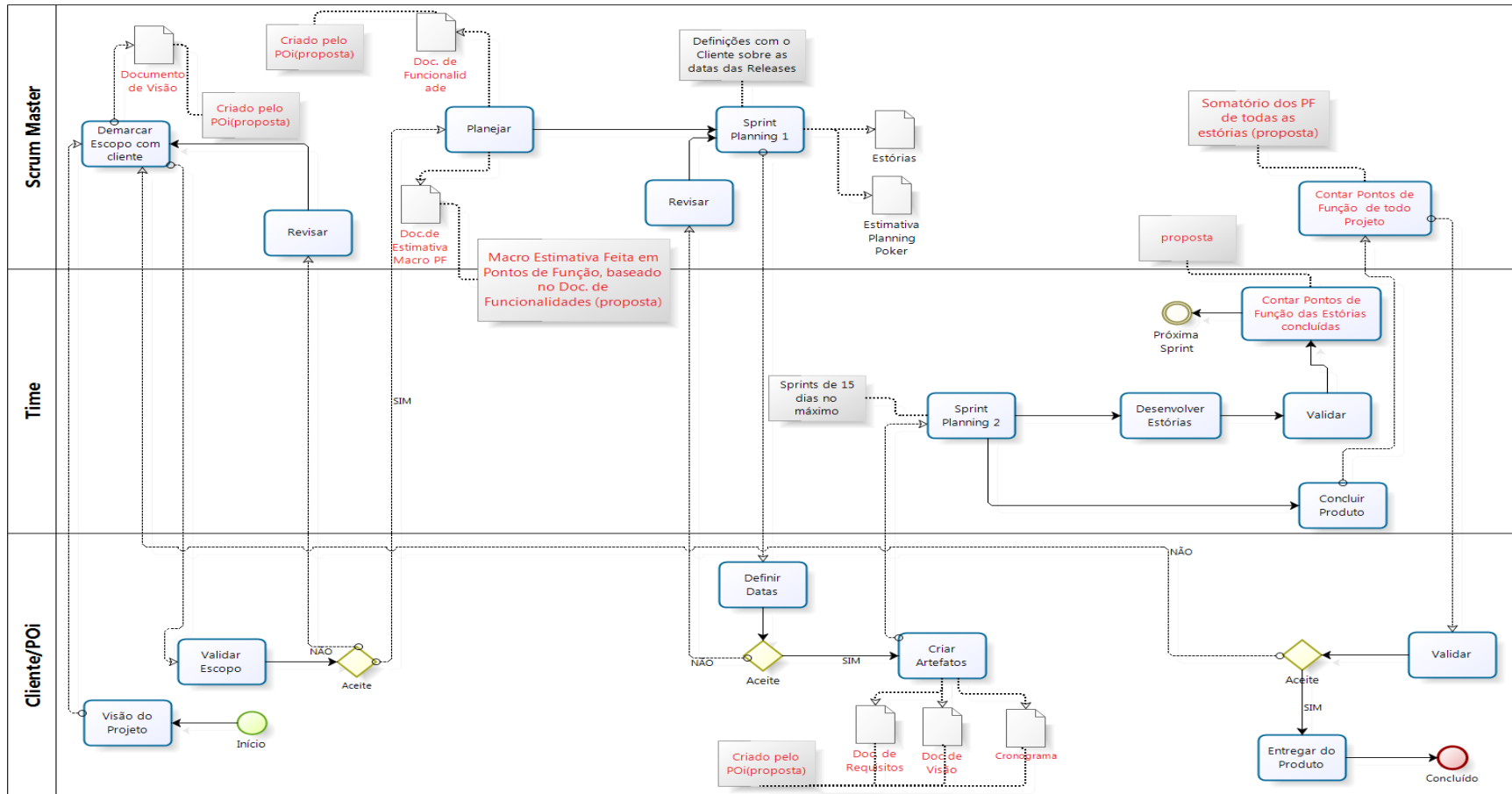
De acordo com Schwaber (2004), o Product Owner (PO) obtém o financiamento no início e durante o projeto, criando os principais requisitos iniciais, os objetivos de retorno sobre investimento (ROI) e o planejamento dos releases. Como nem sempre o PO tem conhecimento técnico, surgem algumas dificuldades na coleta de requisitos, reuniões para entrega das versões, entre outras.

No início do Scrum, o PO era visto como uma pessoa que trabalhava com uma distância grande do time, já que a pressão que ele exercia era prejudicial ao time. Atualmente a grande missão do Scrum Master é transformar o PO num aliado do time, visto que a uma dificuldade de comprometimento do PO com o Time (BARDUSCO , 2008).

Junto a esses problemas, foi proposto a criação do POi (Product Owner interno), analista de TI da empresa, pois ele é responsável pelo alinhamento com o cliente, fazendo visitas ao cliente sempre que for necessário, pois nem sempre é possível que o cliente esteja disponível para alinhamento dos requisitos, ocorrendo conseqüentemente atrasos nas entregas das releases .

Juntamente com o POi foi implementado na Banky System, a Análise de Pontos de Função no seu processo de desenvolvimento, o qual adotou a metodologia Ágil Scrum, conforme figura 8. Em seguida será descrito as fases desse processo.

Figura 8. APF inserida no ciclo do Scrum



- **Visão do Projeto**

Nesta fase ocorre a primeira reunião com cliente onde todos os objetivos e metas do projeto são definidos. É gerado um Documento de Visão e um Documento de Funcionalidades, documento este, onde são destacadas as funcionalidades numa visão macro.

- **Planejar**

Após a validação do Cliente, é gerada uma estimativa em PF numa macro visão, baseado no Documento de Funcionalidades, com a finalidade de garantir financiamento/orçamento para realização do projeto.

- **Sprint Planning 1**

Product Owner interno apresenta os requisitos de maior valor e prioriza aqueles que devem ser implementados. O time então define colaborativamente o que poderá entrar no desenvolvimento da próxima Sprint considerando sua capacidade de produção (SCHWABER). Cada Sprint deve durar no máximo 15 dias, definido as datas das releases.

- **POi (Product Owner interno)**

Foi criado o Product Owner do lado do Time(POi), onde ele é o responsável por criar artefatos do produto e está 100% alinhado com o cliente fazendo parte do time. Na verdade, o cliente confia que o POi possa orientá-lo e saiba transformar todas as solicitações em algo real, já que suas solicitações são abstratas. O POi é responsável também por definir prioridades e adicionar ou retirar funcionalidades com o time, ou seja tem as atribuições do PO tradicional adicionadas das novas reponsabilidades mencionadas. É disponibilizado uma média de 2 projetos por POi ou um terceiro, quando um desses esteja em fase final, para não sobrecarrega-lo, trazendo uma melhor qualidade para os requisitos.

- **Sprint Planning 2**

Time planeja seu trabalho, definindo o Sprint Backlog, que são as tarefas necessárias para implementar as funcionalidades selecionadas a partir do Product Backlog. As mudanças podem ser aceitas até a metade da Sprint, desde que não traga grandes impactos a Sprint. Ao ser finalizada a implementação de cada estória, o desenvolvedor faz seus testes unitários para verificar possíveis erros, logo mais será feita a validação pela equipe de testes.

- **Contagem de Pontos das Estórias Concluídas**

Foi sugerido na proposta, ao final de cada ciclo de uma Sprint é feita uma contagem de PF nas estórias concluídas, desta forma consegue-se mensurar a produtividade de PF homem/hora, pois cada desenvolvedor é o responsável pela sua estória, conforme tabela 8.

**Tabela 8 – Cálculo da produtividade em uma sprint**

Estórias	PF	Homem	Hora	Produtividade
A	3	Pedro	8	3 PF/dia
B	4	José	12	2,6 PF/dia
C	2	Antônio	7	2,3 PF/dia

Fonte: BANKY SYSTEM SOFTWARE BUILDER(2011)

- **Contar Pontos de Função de Todo o Projeto**

Ao concluir o produto é feito um somatório de todas as Estórias concluídas, tendo assim como retorno o total de Pontos de Função do produto.

### **5.3 Considerações finais sobre o processo**

Em virtude do que foi mencionado, a implementação da APF no Scrum, juntamente com a criação do POi, poderá vir a trazer melhorias e discursões referente a mudança no processo do Scrum, onde foram gerados artefatos como documento de visão, documento de funcionalidade, documento de estimativa macro, documento de requisitos e cronograma, todos não comuns ao Framework.

Para dar apoio na contagem de Pontos de Função, foi proposto ao final de cada ciclo de uma Sprint uma contagem de PF nas estórias concluídas, pois é estimado o valor do PF de cada estória e calculado o tempo necessário para sua conclusão, tendo como resultado a produtividade mensurada em PF/dia, onde cada desenvolvedor é o responsável pela sua estória, facilitando a identificação daqueles desenvolvedores que possivelmente não estão sendo produtivos.

Na proposta de criação do POi onde será possível minimizar os problemas nos agendamentos de reuniões, visto que ele faz parte do Time e está mais acessível sendo um membro da equipe, também nas dificuldades de conhecimento técnico, pois nem sempre o PO é conhecedor de tecnologia, apenas conhece o escopo do seu projeto. Em Womack(1991) explica que na Toyota o elemento chave no processo de desenvolvimento de novos carros é o Shusa, uma espécie de super homem, o engenheiro chefe com excelência técnica, que mistura poderes de gestão a uma visão de mercado excepcional, para ser responsável pelo sucesso do produto. Ele tem responsabilidade e autoridade total sobre o projeto, é ele quem decide quais são as features que o carro terá e também qual será o seu custo no mercado.



## 6. Validação dos processos propostos

A validação da proposta do uso da APF no Scrum, está fundamentada na abordagem de medição de produtos e processo de software GQM (Goal Question Metric), proposta por Basili(1992). GQM é baseado em resultado de experiências práticas e pesquisas acadêmicas. O paradigma GQM fundamenta-se na mensuração orientadas por metas.

Foi utilizado um questionário para ajudar na validação da proposta.

Com o intuito de validar a proposta, foram definidos um total de 4 (quatro) metas:

1º: Validar o conhecimento técnico e experiência dos entrevistados sobre a questão de Gerenciamento Ágil e APF.

2º: Analisar e identificar recurso os quais possam comprovar a dificuldade no uso da APF em ambientes Ágeis.

3º: Analisar e identificar recurso os quais possam comprovar na Coleta de Requisitos com o PO(Product Owner).

4º Mensurar o nível de entendimento e do processo e posterior aceitação da proposta de inserção da APF no Scrum.

A projeção de um estudo GQM, no qual é determinado os objetivos a serem aprovados, bem como as questões as quais utilizamos na coleta de dados de forma quantitativa, vão acrescentar valores ao estudo e assim atestar a conquista dos objetivos elencados.

**Tabela 09: Plano GQM Para Validação do Processo Proposto**

Meta 01: Validar o conhecimento técnico e experiência dos entrevistados sobre a questão de Gerenciamento Ágil e APF.	
1.1: Você já participou de projetos usando Metodologias Ágeis?	Métrica: Percentual de Positivos PP = (Quantidade de pessoas que responderam Sim/quantidade total de entrevistados) *100
	Métrica: Percentual de Negativos PN = (Quantidade de pessoas que responderam Não/quantidade total de entrevistados) * 100.

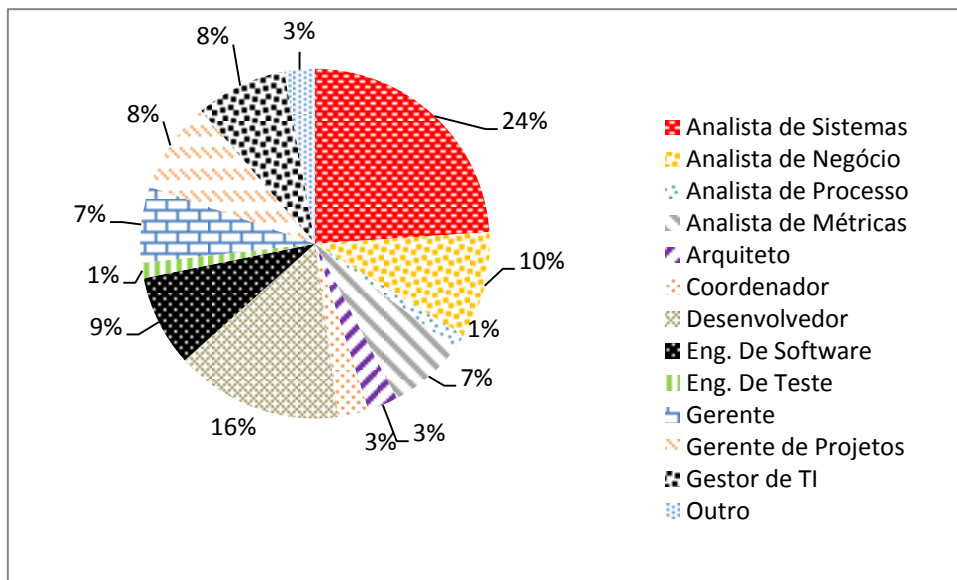
Continuação

Meta 01: Validar o conhecimento técnico e experiência dos entrevistados sobre a questão de Gerenciamento Ágil e APF.	
1.2: Quais os tipos de Estimativas foram usados nesse projeto?	Métrica: Percentual de Item Marcado PIM = (Quantidade de pessoas que marcaram um determinado item/quantidade total de entrevistados) *100.
Meta 02: Analisar e identificar recurso os quais possam comprovar a dificuldade no uso da APF em ambientes Ágeis.	
2.1: Qual a melhor estimativa usada em Ambientes Ágeis?	Métrica: PIM
2.2: Você acha válido o uso de APF em ambientes Ágeis?	Métrica: Nível de Concordância NC = (Quantidade de pessoas que marcaram um determinado item/quantidade total de entrevistados) *100, onde : Concordo totalmente = 100 % De aceitação Concordo parcialmente = 80 % De aceitação Não concordo nem discordo = 60 % De aceitação Discordo parcialmente = 40 % De aceitação Discordo totalmente = 00 % De aceitação
Meta 03: Analisar e identificar recurso os quais possam comprovar a dificuldade na Coleta de Requisitos com o PO(Product Owner).	
3.1: Você já teve alguma dificuldade na coleta de Requisitos com o PO(Product Owner)?	Métrica: PP
	Métrica: PN
Meta 04: Mensurar o nível de entendimento e do processo e posterior aceitação da proposta de inserção da APF no Scrum	
4.1: O processo no qual é inserido a estimativa, na qual é baseada em Pontos de Função, está claro?	Métrica: PP
	Métrica: PN
4.2: Em sua apreciação a proposta do uso de APF no Scrum é válida?	Métrica: Nível de Concordância NC = (Quantidade de pessoas que marcaram um determinado item/quantidade total de entrevistados) *100
4.3: Você acha válido a criação de um POi(Product Owner interno), ou seja um Analista da Empresa é alocado para ser o POi, no qual ficará responsável por estarem 100% alinhado com o cliente, visto que nem sempre o cliente está disponível para participar das reuniões e as vezes não tem alguém de TI para ser o PO. O POi faz várias visitas ao cliente sendo ele o responsável por todas as solicitações de mudanças ou seja, tem todas as atribuições do PO tradicional.	Métrica: NC

## 6.1 Coleta de dados

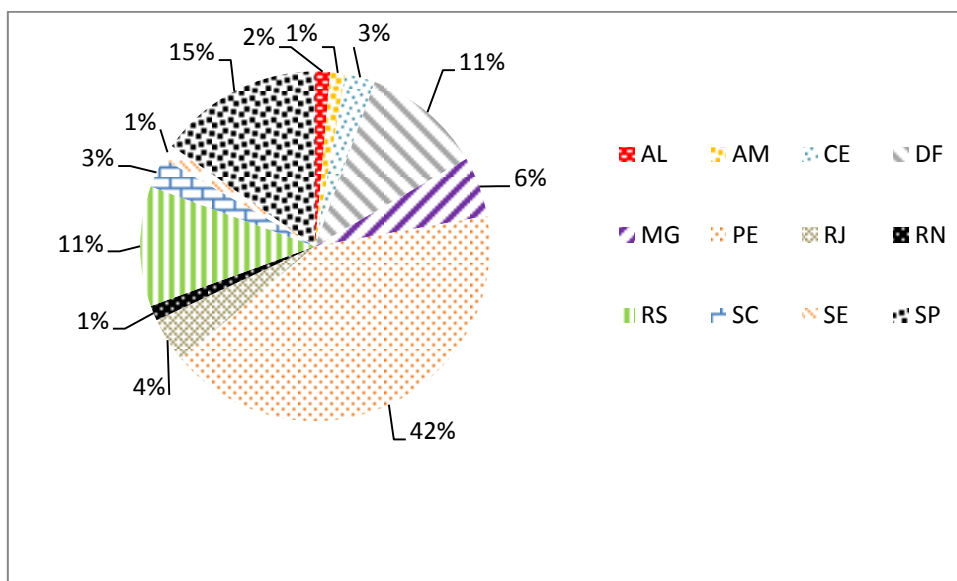
O questionário foi enviado para profissionais na área de Engenharia de Software com foco em metodologias Ágeis e/ou Métricas de software, onde ficou disponível em “<http://www.surveymonkey.com/s/KD6V9YR>” entre ao dias de 28 de fevereiro de 2012 a 01 de março de 2012, onde foi atingido um total de 75 respondentes das questões formuladas, sendo todos eles selecionados pelas características de estarem envolvidos com métodos ágeis e estimativa.

As figuras 9, 10, 11, 12 e 13, apresentam respectivamente os gráficos de respostas, onde procurou-se conhecer o perfil dos entrevistados e validar o conhecimento técnico e experiência dos entrevistados sobre a questão de Gerenciamento Ágil e APF.



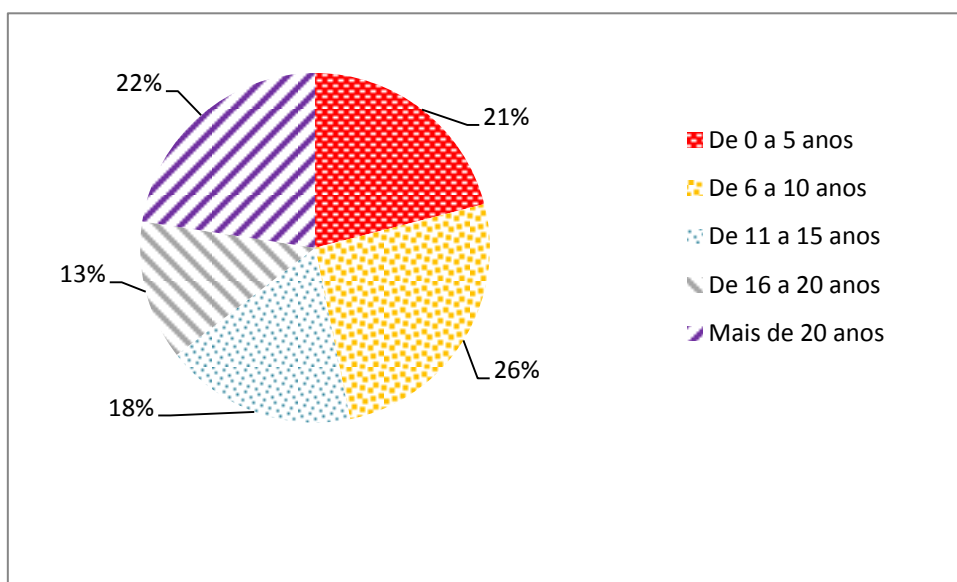
**Figura 9. Função atual na Empresa.**

O gráfico apresentado na figura 9 apresenta a função atual do entrevistado na empresa, observa-se que a maior parte são de Analista de Sistemas e Desenvolvedores, mas foi possível atingir uma porcentagem razoável em comparação as outras profissões de TI.

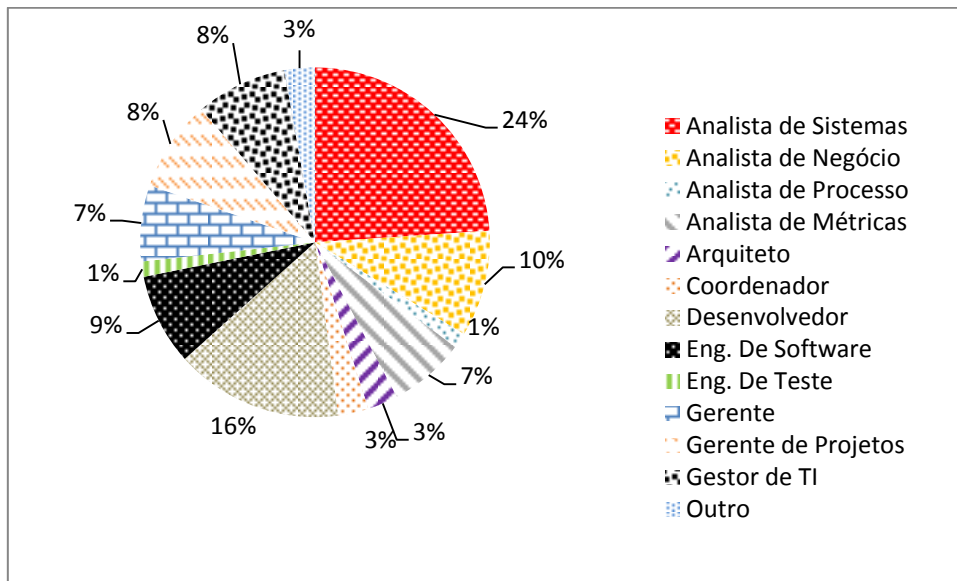


**Figura 10 – UF dos respondentes**

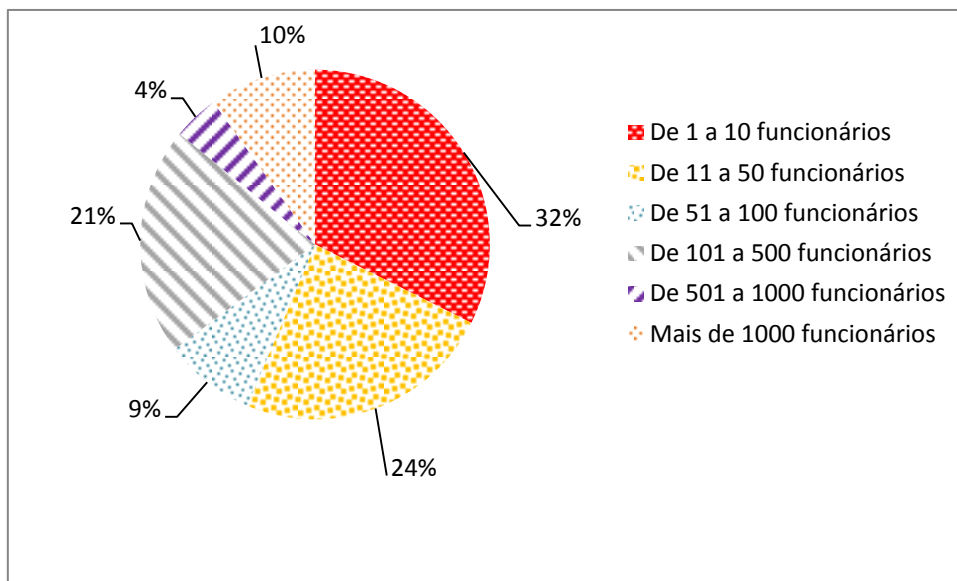
Foi identificado que 42% dos respondentes foram da UF de Pernambuco, no entanto a pesquisa conseguiu atingir profissionais oriundos de outras Unidades da Federação.



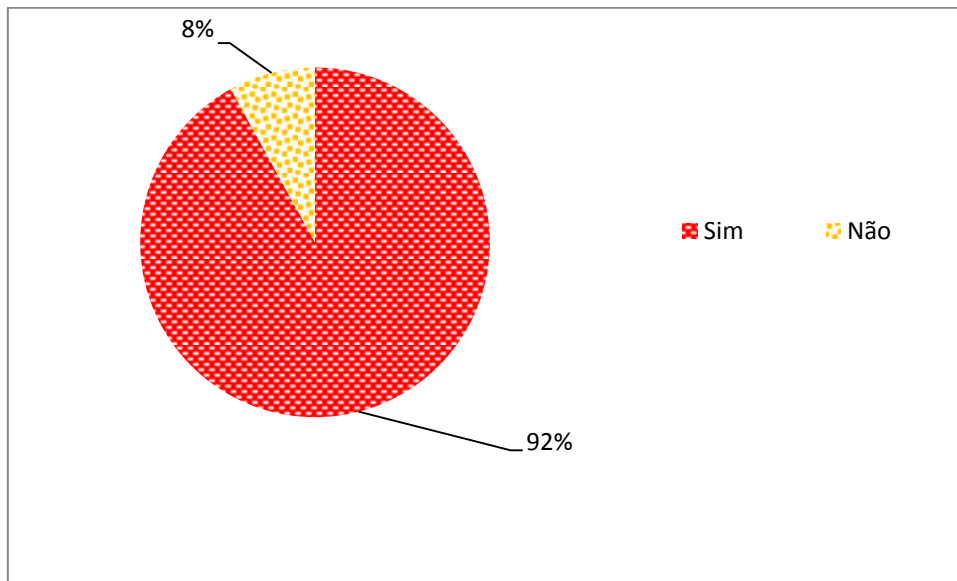
**Figura 11 – Tempo de experiência no mercado**



**Figura 12 – Atual função dos respondentes**

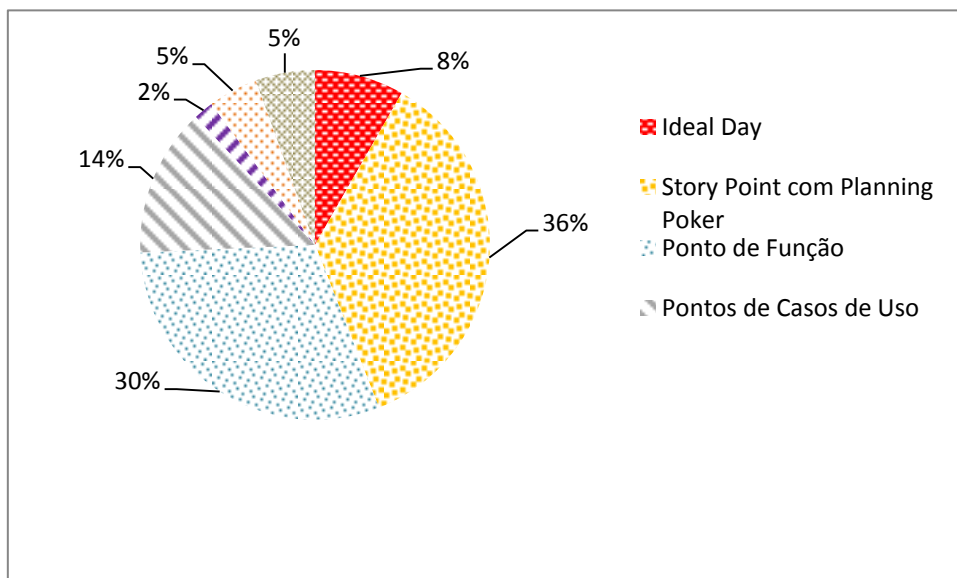


**Figura 13 – Quantidade de funcionários da empresa**



**Figura 14. Questão 1.1 Participação em projetos Ágeis.**

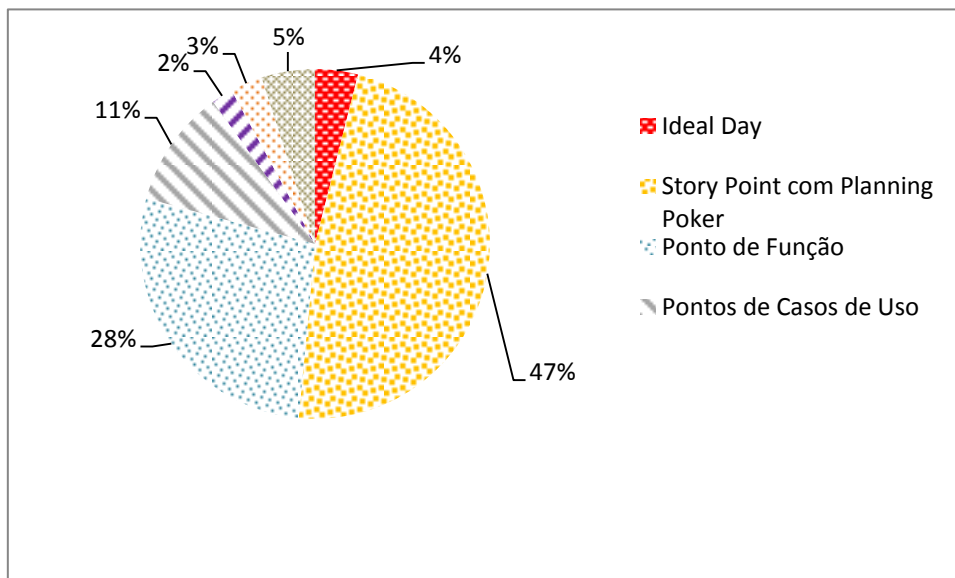
O gráfico apresentado na figura 14 revela que 92% dos respondentes já participaram de projetos utilizando métodos ágeis, dessa forma vem a fortalecer a pesquisa para uma melhor análise.



**Figura 15. Questão 1.2 Tipos de estimativas usadas nos projetos Ágeis.**

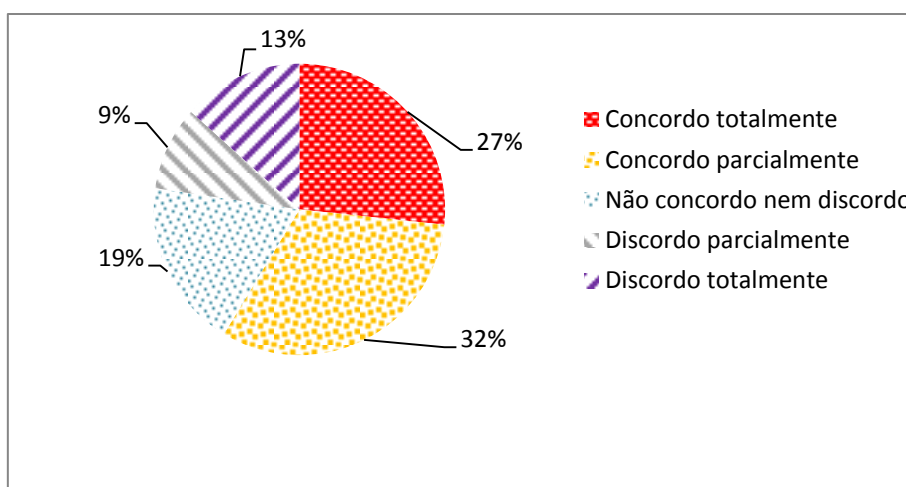
O gráfico apresentado na figura 15 mostra que Planning Poker e Pontos de Função foram as estimativas mais usadas nos seus projetos. Esse resultado reflete uma boa experiência dos respondentes nas estimativas ágeis e pontos de função.

As figuras 16 e 17 apresentam respectivamente os gráficos de respostas, onde procurou-se analisar e identificar recursos os quais possam comprovar a dificuldade no uso da APF em ambientes Ágeis.



**Figura 16. Questão 2.1 Qual a melhor estimativa usada em ambientes Ágeis.**

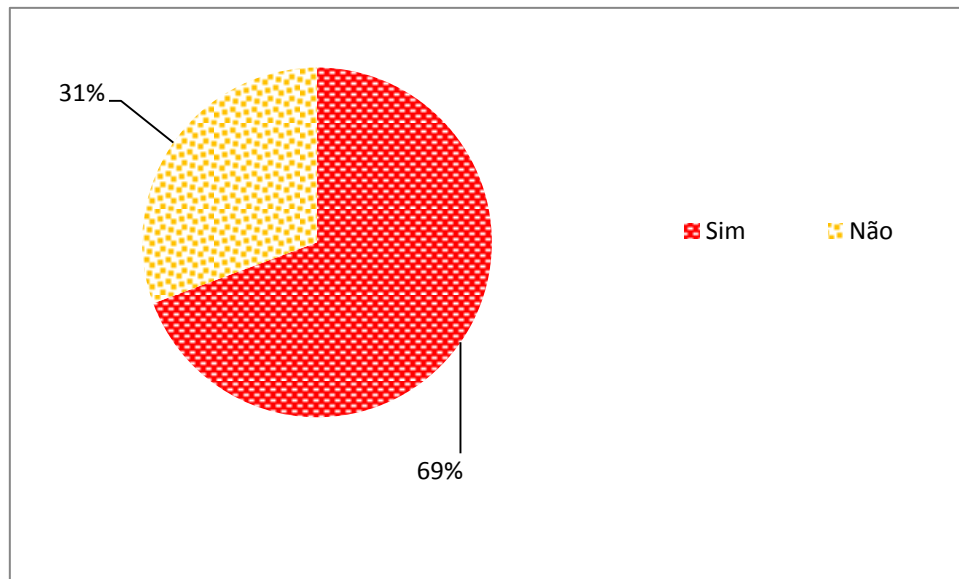
O gráfico apresentado na figura 16, faz um questionamento, onde independentemente da estimativa mais usada na empresa, na opinião do respondente qual é a melhor estimativa usada em ambientes ágeis. A pesquisa mostra que 47% acham Story Point com Planning Poker a melhor estimativa para ambientes ágeis seguido de 28% de Pontos de Função. Mostrando assim que não é inviável o uso da APF em métodos Ágeis.



**Figura 17. Questão 2.2 Validade do uso da APF em ambientes Ágeis.**

O gráfico apresentado na figura 17 mostra que 27% concordam total mente e 32% concordam parcialmente, totalizando mais de 50% de aceitação .

A figura 18 apresenta a coleta de dados, onde procurou-se analisar e identificar recursos os quais possam comprovar a dificuldade na coleta de requisitos com PO(Product Owner).

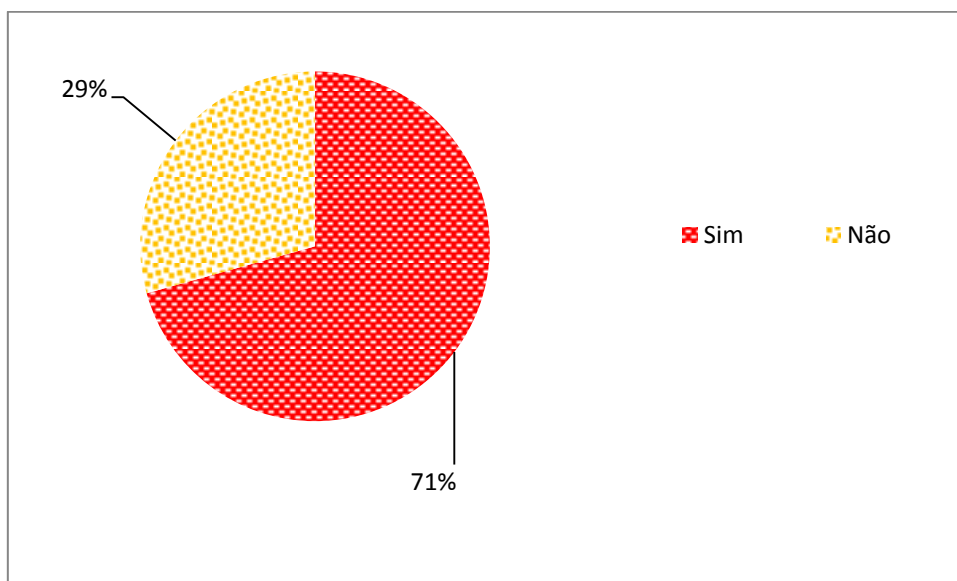


**Figura 18. Questão 3.1 Dificuldade na coleta de requisitos com o PO.**

É possível identificar na figura 20, que grande parte do percentual já teve algum tipo de problema com o PO, na coleta de requisitos.

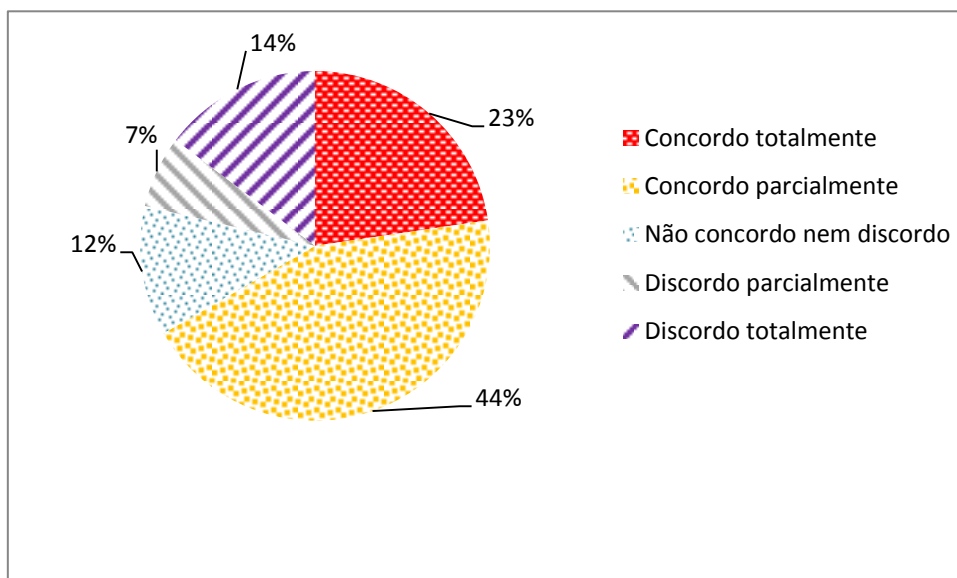
As figuras 19, 20 e 21 mensuram o nível de entendimento do processo proposto e sua aceitação na inserção de APF no Scrum.





**Figura 19. Questão 4.1 Está claro o processo no qual é inserido a estimativa, onde é baseada a Análise de Pontos de Função.**

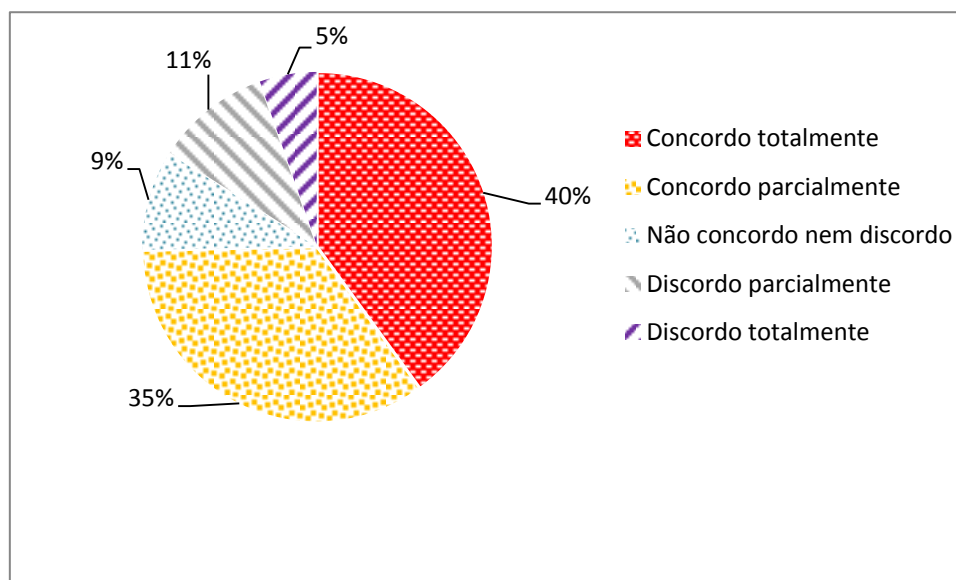
O gráfico de resposta apresentado na figura 19 identifica a compreensão da proposta e processo de inserção da APF no Scrum, por grande dos respondentes, reforçando a clareza das ideias propostas.



**Figura 20. Questão 4.2 Proposta de do uso da APF no SCRUM.**

A maior parte das repostas foi favorável ao uso da APF no Scrum, atestando que não é só estimativas ágeis que se mensura um processo ágil. Pois possivelmente poderá

ser analisado a quebra de alguns paradigmas dos métodos ágeis com a implementação do PF no seu processo.



**Figura 21. Questão 4.3 É válida a criação de um POi.**

Conforme o gráfico da figura 21, a criação de um POi é favorável na maioria das opiniões. Pois esse cenário nos reforça a possibilidade de mudanças no papel do PO dentro do processo do Scrum.

## **5.2. Implementando a APF na BankSystem**

Anteriormente a empresa BankSystem aplicava a metodologia ágil Scrum no seu ciclo natural de processo sem o uso da APF, com a necessidade de atender as empresas publicas, teve que se adequar ao que era exigido pelo órgãos públicos, onde o TCU recomenda o uso da APF para contratação de serviços de desenvolvimento de software.

Após a implementação da APF no processo do Scrum, foi possível mensurar produtividade individual dos desenvolvedores, divergindo do que prega o scrum, onde equipes são auto gerenciáveis. Na empresa BankSystem foi observado uma melhora considerável em termos de produtividade, pois quando era usado estimativas ágeis a produtividade ponto de função por dia era entorno de 1,5PF (dia útil) BANKYSYSTEM(2011). Com a implantação da APF no processo Scrum o produtividade aumentou para 2,5 PF a 3,0 PF (dia útil), pois nesse processo foi possível identificar a produtividade individual de cada desenvolvedor.

### **6.3. Aprovações e questionamentos negativos**

O tema também levantou alguns questionamentos, aprovações e negativas da proposta em fóruns de discussão, os principais pontos abordados foram a questão do “custo x benefícios” em disponibilizar um profissional para participar de todas as fases do projeto, salientando a importância desse profissional entender sobre o produto e principalmente que o seu papel é aproximar o cliente da equipe. O principal ponto negativo abordado foi o aumento dos custos. Já os favoráveis relatam afirmando que seria positivo, se o POi tiver autonomia suficiente para responder pelo cliente.

Também foi analisado a implementação da APF no processo do Scrum, e entre as aprovações as mais contundentes foram as que relataram a aplicabilidade desse processo quando for exigência do cliente, pois alguns clientes não abrem mão de uma estimativa mais exata em vez de uma subjetiva. As aprovações negativas apontavam que a APF não deve medir os itens do Sprint Backlog e os que relatam que não é um bom método para medir esforço.

## **7. Conclusões e trabalhos futuros**

Foi feito um estudo sobre a implementação da APF no Scrum e mudanças nos papéis do Product Owner. Devido a necessidades das empresas de software adotarem a APF como estimativa de software, visto que já possuem um processo ágil para mensurar software, existem paradigmas a serem quebrados para juntar um processo ágil a um processo de medição tradicional.

A metodologia de pesquisa utilizada foi a qualitativa, onde foi aplicado análise de dados e respostas ao questionário. Foi tido uma grande aceitação na implementação da APF no Scrum e nas mudanças dos papéis do PO e criação do POi por parte dos respondentes.

Como o tema é algo muito recente foi tido uma grande dificuldade na procura de artigos relacionados, os quais foram mencionados nas sessões anteriores. Também foi difícil encontrar empresas que utilizasse as duas técnicas.

No estudo, ficou evidenciado que, 27,7% concordaram totalmente com a proposta do uso de APF no Scrum e 44% concordaram parcialmente, tendo desta forma mais de 60% de aceitação. Já no questionamento onde era proposto a criação de um PO

interno obteve-se 40% de aceitação total e 34,7% aceitação parcial, tendo assim mais de 70% de aprovação, abrindo uma possibilidade de uma discussão mais aprofundado sobre o papel do PO no processo do Scrum.

Portanto com esses resultados abre-se um pressuposto da união de um framework ágil com um método padrão de medição de software, podendo assim quebrar as barreiras impostas por alguns agilistas radicais onde eles defendem que a empresa e todos os seus métodos devem ser “ágil”. Além disso a possibilidade da implementação do PO ao Time é bem aceita. Resumindo, se alguém fora do Time deve ter conhecimentos técnicos, esse alguém é o PO. Assim foi proposto que o PO fosse um membro do time, onde deverá ter experiência técnica em alguma área funcional, direcionando o time pela competência, motivando e entusiasmando. Ele é o ponto “chave” o “espelho” o qual o time luta para fazer da ideia dele uma realidade.

Para dar continuidade a esse trabalho, foram identificadas durante a pesquisa alguns trabalhos que poderiam ser desenvolvidos, tais como:

- Seguir arrisca o que prega às metodologias Ágeis ou adequá-la a necessidade do cliente visando o sucesso do projeto;
- Product Owner – Pode, ou deve ser o responsável pelo ROI ou ao Time ;

## Referências

AGILEWAY. **Grafico Burndown** - 2009 Disponível em:  
<<http://www.agileway.com.br/2009/08/18/grafico-burndown-sugestao-de-uso/>>.

Acesso em: 5 de fevereiro de 2012.

ALBRECHT, Allan J. **Medindo a Produtividade do Desenvolvimento de Aplicativos**. 1979 Disponível em:  
<<http://www.fattocs.com.br/artigos/MedindoaProdutividadedoDesenvolvimentodeAplicativos.pdf>>. Acesso em 15 de dezembro de 2011.

**BANKY SYSTEM SOFTWARE BUILDER**. Recife, 2010. Disponível em: <<http://www.banksystem.com.br/siteBS/index.php/component/content/article/41-destaque/178-registro-de-preco->> Acesso em 12 de janeiro de 2012.

**BANKY SYSTEM SOFTWARE BUILDER**. Recife, **Estimativa e Produtividade dos desenvolvedores**. 2011

BASILI, V. R. **Software modeling and measurement: the Goal/Question/Metric paradigm**. University of Maryland, 1992.

BECK et Al. **Agile Manifesto**, 2001. Disponível em: <<http://www.agilemanifesto.org/>>. Acesso em 3 de janeiro de 2012.

BOEHM, Barry. **A View of 20th and 21st Century Software Engineering s.d.**. University of Southern California. University Park Campus, Los Angeles.

BARDUSCO, Danilo. **Product Owner Técnico**. 2008. Disponível em:<<http://planeta-globo.com/page/71/>> . Acessado em 20 de dezembro de 2011.

BRASIL. Ministério da Administração e Reforma do Estado.

**INSTRUÇÃO NORMATIVA MARE-GM Nº 5 1995**. Brasília, DF. Disponível em: <[http://www.comprasnet.gov.br/legislacao/in/in05\\_95.htm](http://www.comprasnet.gov.br/legislacao/in/in05_95.htm)> . Acesso em 14 de Janeiro 2012.

BRASIL. Ministério da Ciência e Tecnologia . **Pesquisa de Qualidade no Setor de Software Brasileiro 2009**. Brasília, DF, 2010. 159 p. Disponível em: <[http://www.tecproit.com.br/downloads/\[Pesquisa\]%20-%20Qualidade%20no%20Setor%20de%20Software%20Brasileiro%202009.pdf](http://www.tecproit.com.br/downloads/[Pesquisa]%20-%20Qualidade%20no%20Setor%20de%20Software%20Brasileiro%202009.pdf)> . Acesso em 1 de fevereiro 2012.

BRASIL. Ministério do Planejamento, Orçamento e Gestão. **Roteiro de Métricas de Software do SISP Versão 2.0**. Brasília, DF, 2012. 65 p. Disponível em: <<https://www.consultas.governoeletronico.gov.br/ConsultasPublicas/consultas.do?acao=exibir&id=105>>. Acesso em 20 de fevereiro 2012.

COCKBURN, Alistair. **Juramento de não lealdade**. 2010 Disponível em: <<http://alistair.cockburn.us/Oath+of+Non-Allegiance>>. Acesso em 23 de fevereiro de 2012.

DEKKERS, C. **Measuring the “logical” or “functional” Size of Software Projects and Software Application**. Spotlight Software, ISO Bulletin May 2003 pp10-13.

DEMARCO, Tom. **Controle de Projetos de Software**. Rio de Janeiro: Campus, 1989.

GAMBA, M. Luiz; BARBOSA, A. C. Garcia **Aplicação de Métricas com Scrum**. Criciúma – SC (s.d.). Disponível em: <

- <http://mateusgamba.files.wordpress.com/2010/11/aplicacao-metricas-no-scrum.pdf>. Acesso em 12 de janeiro de 2012.
- GLOGER, B. **“The Zen of Scrum”**, (2007), Disponível em: <<http://www.glogerconsulting.de>>. Acesso em 23 de fevereiro de 2012
- GOLDMAN, A. et al. **Being extreme in the classroom: Experiences in teaching XP**. Revista da Sociedade Brasileira de Computação, [S.l. : s.n.], 17 out 2004.
- GOMES, A., OLIVEIRA, K. M., ROCHA, A. R. **Métricas para Medição e Melhoria de Processo de Software**. Rio de Janeiro: COPPE/UFRJ, 2001.
- HAZAN, C. **Metodologia para o Uso de Indicadores na Gerência de Projetos de Desenvolvimento de Software**. Dissertação de Mestrado, IME, Rio de Janeiro, maio 1999.
- HAZAN, C. **Análise de Pontos de Função: Uma Aplicação nas Estimativas de Tamanho de Projetos de Software**. Engenharia de Software Magazine, Edição 2ª, Devmedia, 2008, pp.25-31.
- INTERNATIONAL FUNCTION POINT USERS GROUP. **Counting Practices Manual**. Version 4.3.1, January, 2010. Disponível em: <<http://www.ifpug.org>>. Acesso em 15 de dezembro 2011.
- JONES, C. **Applied Software Measurement**, McGraw Hill, 2008
- LARMAN, C. **Agile and iterative development: a manager's guide**. Reading, MA, Estados Unidos: Addison-Wesley, 2003.
- MATOS, Fabrício Vargas. **Pontos de função e metodologias ágeis conseguem conviver juntas?**. 26 ago 2010. Disponível em: <<http://blog.qualidata.com.br/?p=530>>. Acesso em 4 de dezembro 2011.
- MNKANDLA, E.; DWOLATZKY, B. **A survey of agile methodologies**. Africa do Sul: [S.n.], 2004. Disponível em: <<http://www.eepublishers.co.za/images/upload/Trans12p236-247.pdf>>. Acesso em janeiro de 2012.
- MURPHY, Thomas; NORTON, David **“Predicts 2010: Agile and Cloud Impact Application Development Directions”**, GARTNER, 2010 Disponível em:

- <<http://analytical-mind.com/2010/03/09/gartner-predicts-2010-agile-and-cloud-impact-application-development-directions/>>. Acesso em 3 de dezembro de 2009.
- OEST, Carlos. **Usando Pontos de Função em Projetos Ágeis**. International Software Measurement Analysis conference, São Paulo, 2010
- OLIVEIRA, M. Marly. **Como Fazer Projetos, Relatórios, Monografias, Dissertações e teses**. 5ª ed. São Paulo: Campus, 2011.
- PERNAMBUCO. Agência de Tecnologia da Informação. **Guia de Contagem APF**. 2010. 65 p Disponível em:  
<[http://www.portaisgoverno.pe.gov.br/c/document\\_library/get\\_file?p\\_1\\_id=567493&folderId=404286&name=DLFE-25582.pdf](http://www.portaisgoverno.pe.gov.br/c/document_library/get_file?p_1_id=567493&folderId=404286&name=DLFE-25582.pdf)>. Acesso em 10 de janeiro de 2012.
- PRESSMAN, Roger S. **Engenharia de Software: Uma Abordagem Profissional**. 7. ed. Porto Alegre: Bookman, 2011.
- RISING, Linda; JANOLF, Norman S. **The Scrum Software Development Process for Small Teams**. 17 ed. USA: IEE Software, 2000.
- SANTANA, Célio. **Uso de análise de Pontos de Função em ambientes ágeis**. Engenharia de Software Magazine, [S.l.], ed 20, p. 33-39, 6 jan. 2010.
- SCHWABER, Ken. **Scrum Development Process**. [S.l.], [20--]. Disponível em:  
<<http://home.hib.no/ai/data/master/mod251/2009/articles/scrum.pdf>>. Acesso em 1 de fevereiro 2012.
- SCHWABER, Ken. **Agile project management with Scrum**. Redmond, WA, Estados Unidos: Microsoft Press, 2004.
- SOMMERVILLE, Ian. **Engenharia de Software**. 9ª. ed. São Paulo: Pearson Addison Wesley, 2011.
- TANIGUCHI, Kenji; CORREIA, Fernando E. **Metodologias Ágeis e a Motivação de Pessoas em Projetos de Desenvolvimento de Software**, Vol. IV N°. 4 , 2009.
- VAZQUEZ, Carlos E.; SIMÕES, Guilherme S.; ALBERT, Renato M. **Análise de Pontos de Função**. 11 ed. São Paulo: Érica, 2011.

VERSIONONE. **State of Agile Survey**. [s.l.], 2011. Disponível em: <[http://www.versionone.com/pdf/2011\\_State\\_of\\_Agile\\_Development\\_Survey\\_Results.pdf](http://www.versionone.com/pdf/2011_State_of_Agile_Development_Survey_Results.pdf)>. Acessado em 30 de Janeiro de 2012.

WOMACK, Jame P.; JONES, Daniel T.; ROOS, Daniel **The Machine that Changed the World**. 1 ed. Harper Perennial, 1991