5. Contagem de Pontos de Função no Desenvolvimento de Software utilizando Métodos Ágeis

Este capítulo descreve orientações, sobre a utilização da métrica Ponto de Função, para medição e remuneração de serviços de desenvolvimento de software em projetos com adoção de métodos ágeis para subsidiar as contratações desses serviços na Administração Pública Federal (APF).

Uma das principais dificuldades e desafios na adoção de métodos ágeis em contratação de desenvolvimento de software é definir um modelo de remuneração que seja equilibrado, remunerando de forma justa o esforço da contratada para atender o volume de refinamentos e mudanças em funcionalidades e, ao mesmo tempo, não onerando de forma excessiva a contratante (instituições públicas). Ou seja, o valor pago deve corresponder aos serviços recebidos e o ciclo do processo ágil de desenvolvimento de software não deve influenciar negativamente o ciclo de faturamento do projeto. Devido as características inerentes ao processo ágil, entende-se que os refinamentos e as mudanças em funcionalidades são mais constantes e recorrentes nesse cenário de desenvolvimento de software, pois pressupõe-se um escopo mais aberto. Entretanto, o processo ágil de desenvolvimento de software em contratações não deve comprometer os princípios de economicidade e efetividade dos resultados previstos e entregues com a garantia da exequibilidade do projeto.

É importante observar que, conforme a Súmula TCU 269, a remuneração nas contratações de serviços de Tecnologia da Informação deve estar vinculada a entrega de resultados ou ao atendimento de níveis de serviço. Portanto, é relevante que o modelo de remuneração, níveis de serviço e critérios de qualidade para a aceitação dos resultados entregues ao final das iterações (*sprints*) do processo ágil, estejam claramente definidos no instrumento convocatório e sejam observados e aferidos durante a gestão do contrato.

O propósito deste trabalho é possibilitar a adoção de um modelo de faturamento baseado na métrica Ponto de Função, capaz de medir a entrega de resultados do projeto, em contratações de desenvolvimento de software usando métodos ágeis. Os objetivos e premissas considerados neste trabalho foram:

- Buscar a simplicidade na medição do desenvolvimento de software com métodos ágeis para viabilizar o seu uso em contratações com responsabilidade e garantir o alcance dos benefícios do processo ágil ao negócio;
- Fortalecer a necessidade e importância de registro das mudanças em funcionalidades para promover a criação de uma base histórica de projetos com desenvolvimento ágil, além de permitir a rastreabilidade e o atendimento de auditorias nos projetos;
- Minimizar o ônus na gestão de projeto advindo da utilização do processo ágil em contratações de software, tanto para a contratante (gestor de TI) quanto para a

contratada;

- Simplificar o ônus de gestão e controle de mudanças em funcionalidades de forma a minimizar impactos sobre a agilidade do processo de desenvolvimento;
- Prever, medir e remunerar o esforço e o volume de mudanças em funcionalidades em um projeto com desenvolvimento ágil;
- Promover a oferta de preços exequíveis para a realização de contratos de prestação de serviços de desenvolvimento de software com métodos ágeis, a partir da publicidade de características do projeto e da equipe de desenvolvimento previamente especificadas no edital de contratação;
- Incentivar o uso de desenvolvimento ágil no governo com o aprimoramento da maturidade e competência da equipe envolvida, buscando a eficiência do processo e dos resultados esperados.

Nesse sentido, são apresentadas recomendações de medição em Ponto de Função, que podem ser adotadas nas contratações de desenvolvimento de software com métodos ágeis, para o tratamento dos refinamentos e mudanças em funcionalidades durante o projeto. Essas recomendações foram definidas após o levantamento e avaliação da literatura e de guias de contagem de órgãos do governo que utilizam alguma abordagem de medição para o cenário de desenvolvimento de software com métodos ágeis [CAIXA, 2012], [Castro e Hernandes, 2014], [Horvath, 2012], [Keote, 2010], [NESMA, 2009] e [PROCERGS, 2013]. Realizou-se, ainda, um *benchmarking* em órgãos do governo que já implementam contratos de desenvolvimento de software com métodos ágeis. E, por fim, foram realizadas reuniões técnicas com especialistas em desenvolvimento ágil e em métrica Ponto de Função para discutir e refinar a proposta apresentada neste documento.

5.1 Conceitos

No cenário de desenvolvimento de software com métodos ágeis e dentro do contexto deste roteiro, é importante alinhar os seguintes conceitos:

Release: É um ciclo que perpassa pelas fases do processo de desenvolvimento de software com o objetivo de entregar, ao final do ciclo, um **produto pronto** a ser colocado em produção para uso. A duração de cada *release* será definida pela contratante na fase de planejamento do projeto conforme seu backlog priorizado e de forma a garantir uma entrega de valor antecipada aos usuários.

***Sprint*:** É uma unidade de período de tempo fixo (*time box*) dentro da *release*, com datas de início e fim pré-definidas, dentro da qual é executado um conjunto de atividades de desenvolvimento do projeto previamente estabelecidas, gerando ao final um incremento do produto aceito e potencialmente implantável.

Ciclo de Pagamento: período definido para fins de pagamento e apuração dos resultados entregues, podendo consistir de uma iteração (*sprint*), de um conjunto

de iterações, ou de uma *release*. Considerando os critérios adotados para o projeto, como o tamanho da iteração (*sprint*), o tamanho da *release*, a produtividade da equipe do projeto e a expectativa de fluxo de caixa da contratada para manter seu equilíbrio econômico-financeiro no atendimento do contrato, deve-se realizar o faturamento por iteração (*sprint*), por grupo de iterações, ou por *release*, desde que sempre devidamente associado aos produtos entregues e aceitos de uma ordem de serviço.

Produto Pronto: Visando a remuneração da contratada a partir da medição de resultados gerados em um "ciclo de pagamento", entende-se que um produto está "pronto" se foi entregue e aceito. Cabe observar que o desenvolvimento de uma funcionalidade pode perpassar mais de uma *sprint* e conter várias histórias de usuários prontas e validadas em *sprints* diferentes e; nesse caso, a funcionalidade só será considerada para fins de pagamento ao final do ciclo de pagamento em que estiver com todas as suas histórias componentes "prontas".

Refinamentos: são quaisquer mudanças ocorridas sobre uma função transacional ou de dados já previamente desenvolvida(s) na *release* corrente (seja este uma inclusão, alteração ou exclusão), provocadas pelo aprofundamento, detalhamento e complementação de requisitos durante o processo de desenvolvimento.

5.2. Orientações

O desenvolvimento de software utilizando métodos ágeis deve respeitar uma abordagem específica que considere as características dos métodos ágeis, tanto no desenvolvimento quanto na gestão do projeto. Entretanto, essas características podem requerer adaptações para o contexto de contratações de software na APF, no sentido de atender o cumprimento da legislação e dos princípios de economicidade e eficiência. Nesse cenário, algumas considerações e sugestões são propostas para o desenvolvimento de software utilizando métodos ágeis na APF:

- Remuneração baseada nos resultados entregues e aceitos (Produto Pronto);
- Remuneração sempre atrelada a uma ordem de serviço;
- Promover o fluxo de demandas do projeto e o equilíbrio econômico-financeiro da contratada;
- Divisão do projeto de desenvolvimento ou manutenção em *releases;*
- Ciclo da *sprint* (iteração) de 2 até 4 semanas;
- Ciclo da *release* não deve ser igual ao ciclo da *sprint*. Ou seja, *release* formada por apenas uma *sprint* não permite a adoção das orientações trazidas neste documento;
- Ciclo da *release* deve, sempre, promover o aumento do percentual de completude do sistema (entrega de valor agregado ao negócio);

- Para as funcionalidades que precisem de mais de uma *sprint* para serem desenvolvidas, recomenda-se que sejam contadas somente na *sprint* em que forem entregues e aceitas;
- Realizar a contagem estimativa do projeto, a fim de definir o tamanho estimado ao final de um “ciclo de pagamento” para efeito de planejamento do projeto e geração das ordens de serviço de desenvolvimento ou manutenção de software.

Para efeito de gestão das mudanças e geração de indicadores, recomenda-se que as mudanças em funcionalidades sejam registradas em planilha separada da contagem do projeto de desenvolvimento. Nessa planilha de mudanças devem ser registradas todas as funcionalidades incluídas, alteradas e excluídas, porém, para efeito de faturamento, sugere-se que as funcionalidades incluídas sejam remuneradas somente na contagem final do “ciclo de pagamento”, conforme modelo de remuneração adotado para o projeto, a fim de não existir duplicidade de remuneração.

Caso o “ciclo de pagamento” seja por *sprint*, sugere-se adotar como critério de remuneração para a *sprint*, um valor percentual do tamanho total planejado da *release* ou uma medida que reflita o valor agregado dos produtos prontos da *sprint* dentro da *release*. Essa sugestão visa não onerar as partes envolvidas com a medição, controle e gestão de mudanças a cada *sprint*, principalmente, quando a duração dessas iterações são muito curtas. Caso a remuneração seja realizada por *sprint*, sugere-se reter um percentual do total da remuneração da iteração (*sprint*) para o final da *release*, principalmente, quando a conclusão da implantação do produto pronto da *sprint* ocorrer no final da *release*.

O item **Diretrizes para Acompanhamento de Projetos** deste Roteiro não deve ser aplicado para projetos de desenvolvimento de software com métodos ágeis, em virtude da adoção das orientações contidas neste capítulo específico para projetos dessa natureza.

5.3 Tratamento de Mudanças em Funcionalidades no Processo Ágil

Esta seção apresenta orientações sobre o tratamento de mudanças em funcionalidades para contratos de desenvolvimento de software com métodos ágeis usando a métrica Ponto de Função.

As mudanças em funcionalidades podem ser decorrentes de mudanças no domínio do negócio - como alteração de escopo, de regras de negócio - ou mudanças legais/regulamentares ou, ainda, refinamentos de requisitos. Considerando os aspectos do desenvolvimento ágil, as mudanças em funcionalidades que ocorrerem após o término da *release* em que essas funcionalidades ficaram prontas, devem ser tratadas de acordo com o item **Projeto de Melhoria** deste Roteiro. Além disso, é prática comum existirem mudanças em uma funcionalidade ainda durante a execução das *sprints* de uma *release*. Neste guia considera-se que, no desenvolvimento de software com métodos ágeis, o ciclo de trabalho evolutivo em funcionalidades desenvolvidas em uma *release* encerra-se ao

final da *release.* Assim, este guia sugere que as mudanças em funcionalidades ocorridas dentro dessas características não sejam contadas e, consequentemente, não sejam remuneradas durante a release (ou seja, nos ciclos de pagamento do projeto), mas que já estejam absorvidas pela contratada como parte inerente do processo ágil de desenvolvimento adotado para o projeto.

Nesse sentido, é fundamental que o instrumento convocatório de licitação especifique o máximo de fatores, características e aspectos relevantes do projeto para que as empresas candidatas ao certame avaliem adequadamente as possibilidades de atendimento do contrato, fornecendo profissionais qualificados e preço de ponto de função exequível para o contrato. Dessa forma, é importante destacar a necessidade do órgão contratante avaliar e controlar a sua gestão de riscos pela adoção de um contrato de desenvolvimento de software com métodos ágeis. O risco poderá se mostrar inversamente proporcional ao detalhamento dos fatores, características e aspectos do projeto expostos no edital de contratação que possam interferir no desenvolvimento e no sucesso do projeto.

5.3.1 Fatores que Influenciam o Número de Mudanças em Funcionalidades no Processo Ágil

Nesta subseção apresentamos alguns fatores que influenciam o número de mudanças em funcionalidades no projeto de desenvolvimento de software com métodos ágeis. Esses fatores podem ser levantados, por exemplo, de uma base histórica de projetos similares do órgão, experiências registradas de projetos com desenvolvimento ágil já realizados pelo órgão e entrevistas com prestadores de serviços de desenvolvimento de software com métodos ágeis.

Como foi dito na seção anterior, dentro de uma *release*, as mudanças em funcionalidades desenvolvidas previamente na mesma *release* não são contadas e remuneradas durante o projeto, pois são absorvidas pela contratada como parte do processo de desenvolvimento ágil. Entretanto, caso essas mudanças ocorram em *releases* diferentes, sugere-se remunerar conforme os itens de manutenção abordados neste Roteiro, tal como, a manutenção evolutiva aplicando-se o fator de impacto sobre o tamanho da funcionalidade impactada, conforme sugerido no item **Projeto de Melhoria** deste Roteiro.

Apesar de não serem contadas em ponto de função, essas mudanças em funcionalidades já desenvolvidas dentro da mesma *release* devem ser registradas e atendidas pelo contrato, mas sem remuneração adicional ao total de pontos de função da contagem detalhada final da *release,* pois se entende que são relativas à evolução de requisitos do processo de desenvolvimento adotado no projeto. Portanto, na contagem detalhada final da *release* não deve haver nem acréscimo de ponto de função nem de qualquer outra natureza.

Alguns fatores devem ser considerados e avaliados para a estimativa do volume de mudanças em funcionalidades em um projeto de desenvolvimento com métodos ágeis:

- maturidade dos requisitos do projeto;

- conhecimento do negócio pelo *product owner;*

- maturidade do processo ágil implantado no órgão (nível de aderência às práticas);

- disponibilidade e experiência com métodos ágeis da área de negócio (*product owner*);

- nível de experiência com métodos ágeis da equipe da contratante (principalmente do *product owner*);

- nível de experiência com métodos ágeis requerido para a equipe de desenvolvimento da contratada;

- tamanho da *sprint* e da *release;*

- volume de mudanças em funcionalidades de projetos similares já executados.

5.3.1.1 Exemplo de Aplicação da Proposta

Para exemplificar a aplicação dessa proposta de tratamento das mudanças em funcionalidades em um projeto de desenvolvimento com métodos ágeis, suponha o planejamento das iterações (*sprints*) de uma *release* (*Release N*) com quatro funcionalidades a serem desenvolvidas (incluídas) e duas funcionalidades prontas em *releases* anteriores para serem alteradas, conforme apresentado na Figura 1. O tamanho estimado do *backlog* da *Release N* é de 21 PF. Considere que, ao final da *Release N*, as funcionalidades incluídas devem estar prontas para serem remuneradas para a contratada.

Release N (composta de 3 Sprints)	Processo Elementar (PE)	Categoria (Inc, Alt, Exc, Refin)	Tipo (ALI, AIE, EE, CE, SE)	Complex.	PF	Observação
Sprint 1	Incluir Aluno	Inc	EE	Baixa	3	
	Aluno	Inc	ALI	Baixa	7	
	Incluir Disciplina	Alt	EE	Baixa	1,5	Alteração caracterizada como ***Projeto de Melhoria*** (PE "Incluir Disciplina" desenvolvido e pronto na Release N-1). Aplicado o fator de impacto de 50% (3PF*0,5=1,5PF).
Sprint 2	Alterar Aluno	Inc	EE	Baixa	3	
	Alterar Disciplina	Alt	EE	Baixa	1,5	Alteração caracterizada como ***Projeto de Melhoria*** (PE desenvolvido e pronto na Release N-1). Aplicado o fator de impacto de 50% (3PF*0,5=1,5PF).
Sprint 3	Emitir Relatório de Alunos por Disciplina	Inc	SE	Média	5	
Total de PF da *release*					**21**	

Tabela 11 – Planejamento do *Backlog* das *Sprints* da *Release* N

A contagem detalhada de pontos de função da *Release N* está apresentada na

Tabela 12. Observa-se que não existe medição para as mudanças em funcionalidades desenvolvidas na mesma *Release N*. Mas, o objetivo principal é mostrar a necessidade de registrar as funcionalidades incluídas, alteradas, excluídas e refinamentos (mudanças em funcionalidades desenvolvidas na mesma *release*) durante a *release*, independente do registro e da identificação da iteração (*sprint*) onde elas ocorreram. Nesse sentido, é facultativo o registro das contagens por *sprint* desde que a contagem da *release* registre as novas funcionalidades desenvolvidas, bem como, as mudanças em funcionalidades.

A Tabela 12 apresenta a contagem de pontos de função e o detalhamento dos serviços realizados na execução da *Release N* e suas *sprints*, incluindo os processos elementares planejados (inclusão e alteração em funcionalidades) e as mudanças em funcionalidades previamente desenvolvidas na mesma *release*, essas categorizadas como "**Refinamento**", que foram identificadas durante o desenvolvimento da *Release N*. Destaca-se, por exemplo, que o processo elementar "*Alterar Disciplina*" foi alterado na *sprint* 2, sendo essa mudança categorizada como "**Alteração**" do Projeto de Melhoria. Em seguida, na *sprint* 3 houve uma nova mudança no mesmo processo elementar "*Alterar Disciplina*", sendo essa mudança categorizada como "**Refinamento**" pois trata-se de uma mudança em funcionalidade já trabalhada na mesma *release*. Além disso, observe que houve a necessidade de alterar o processo elementar "*Emitir Relatório de Disciplinas*" identificada durante o desenvolvimento da funcionalidade planejada "*Emitir Relatório de Alunos por Disciplina*". Essa alteração foi classificada como "**Alteração**" do Projeto de Melhoria, pois a funcionalidade "*Emitir Relatório de Disciplinas*" já estava pronta (ou seja, foi desenvolvida em *release* anterior). Portanto, no desenvolvimento de projetos com métodos ágeis, uma mudança em funcionalidade poderá ser classificada em "**Refinamento**" ou "**Alteração**", a depender se a funcionalidade foi desenvolvida na mesma *release* ou não.

Na Tabela 12, as mudanças em funcionalidades do tipo "Refinamento" foram absorvidas pela contratada e, portanto, não houve remuneração adicional ao total de pontos de função da *Release N*. Assim sendo, conforme apresentado na Tabela 12, a contagem final da *Release N* foi de 22,5 PF, que representa o valor a ser remunerado à contratada e corresponde às funcionalidades incluídas (18 PF) e alteradas (4,5 PF – alterações caracterizadas como Projeto de Melhoria) na *Release N*. Portanto, considerando o "ciclo de pagamento" adotado, a remuneração da contratada para a *release* corresponderá ao tamanho em pontos de função das funcionalidades incluídas acrescida do valor em pontos de função das mudanças (alterações e exclusões caracterizadas como Projeto de Melhoria) em funcionalidades ocorridas durante a mesma release, excluindo-se os refinamentos (mudanças em funcionalidades desenvolvidas na mesma *release*) que não são contados e remunerados durante o projeto.

Sugere-se que o registro das mudanças em funcionalidades seja feito em uma planilha de contagem separada aplicando-se as regras de medição sugeridas neste Roteiro. O campo "Categoria" nas Tabelas 11 e 12 mostra, além dos tipos **Inc** (inclusão), **Alt** (alteração) e **Exc** (exclusão) de funcionalidades, o tipo **Refin** (Refinamento) para

representar as mudanças em funcionalidades desenvolvidas na mesma *release*.

Release N (composta de 3 Sprints)	Processo Elementar (PE)	Categoria (Inc, Alt, Exc, Refin)	Tipo (ALI, AIE, EE, CE, SE)	Complex.	PF	Observação
Sprint 1	Incluir Aluno	Inc	EE	Baixa	3	
	Aluno	Inc	ALI	Baixa	7	
	Incluir Disciplina	Alt	EE	Baixa	1,5	Alteração caracterizada como ***Projeto de Melhoria*** (PE "Incluir Disciplina" desenvolvido e pronto na Release N-1). Aplicado o fator de impacto de 50% (3PF*0,5=1,5PF).
Sprint 2	Alterar Aluno	Inc	EE	Baixa	3	
	Aluno	Refin	ALI	Baixa	-	Mudança caracterizada como ***refinamento*** de funcionalidade para atender o PE "Alterar Aluno". Sem custo PF.
	Alterar Disciplina	Alt	EE	Baixa	1,5	Alteração caracterizada como ***Projeto de Melhoria*** (PE desenvolvido e pronto na Release N-1). Aplicado o fator de impacto de 50% (3PF*0,5=1,5PF).
Sprint 3	Emitir Relatório de Alunos por Disciplina	Inc	SE	Média	5	
	Incluir Aluno	Refin	EE	Baixa	-	Mudança caracterizada como ***refinamento*** de funcionalidade para atender o PE "Emitir Relatório de Alunos por Disciplina". Sem custo PF.
	Incluir Disciplina	Refin	EE	Baixa	-	Mudança caracterizada como ***refinamento*** de funcionalidade para atender o PE "Emitir Relatório de Alunos por Disciplina". Sem custo PF.
	Alterar Disciplina	Refin	EE	Baixa	-	Mudança caracterizada como ***refinamento*** de funcionalidade para atender o PE "Alterar Disciplina". Sem custo PF.
	Emitir Relatório de Disciplinas	Alt	EE	Baixa	1,5	Alteração caracterizada como ***Projeto de Melhoria*** (PE desenvolvido e pronto na Release N-1). Aplicado o fator de impacto de 50% (3PF*0,5=1,5PF).
Total de PF da *release*					**22,5**	

Tabela 12 – Contagem Detalhada de Pontos de Função da *Release N*

A Tabela 13 apresenta a contagem de pontos de função da *Release N* para a *baseline* da aplicação. Observe que nessa contagem aparecem apenas as funcionalidades incluídas e não devem constar as funcionalidades alteradas, excluídas e refinamentos durante a *Release N*, a menos que tais mudanças tenham impactado na complexidade da função de dados ou transacional. O total de pontos de função da *Release N* para a *baseline* da aplicação é de 18 PF.

	Processo Elementar (PE)	Tipo (ALI, AIE, EE, CE, SE)	Complex.	PF	Observação
Contagem da *Release N*	Aluno	ALI	Baixa	7	Na contagem da *release* para a *baseline* da aplicação, não devem
	Incluir Aluno	EE	Baixa	3	
	Alterar Aluno	EE	Baixa	3	

	Emitir Relatório de Alunos por Disciplina	SE	Média	5	constar as funcionalidades alteradas, evoluídas e excluídas.
			Total de PFs da *Release*	**18**	

Tabela 13 – Contagem de PF da *Release N* para *Baseline* da Aplicação

5.4 Referências Bibliográficas

[CAIXA, 2012] CAIXA. **Guia de Orientação - Métricas**, versão 10, 2012.

[Castro e Hernandes, 2014] CASTRO, M. V. B. de; HERNANDES, C. A. M.. ***A Metric of Software Size as a Tool for IT Governance***. Procedings in: SBES, 2014.

[Horvath, 2012] HORVATH, D.. ***Function Point Analysis and Agile Methodology****. Q/P Management Group*, Inc. 2012.

[Keote, 2010] KEOTE, A. K.. ***Function Points and Agile – Hand in Hand***. Accenture, 2010.

[NESMA, 2009] NESMA. ***Function Point Analysis for Software Enhancement Guidelines***. Version 2.2.1, Disponível em< www.nesma.nl >, 2009.

[PROCERGS, 2013] PROCERGS. **Guia de contagem da PROCERGS**. Versão 2.0 – Alterações referentes ao Edital de Fábrica de Software de Sistemas, Atualizado em 13/06/2013.