

Universidade de Brasília - UnB
Faculdade UnB Gama - FGA
Engenharia de Software

Comparação da acurácia de estimativas de esforço produzidas a partir de Pontos de Função Cosmic e de Story Points

Autor: Alex Cortes Alves
Orientador: Msc. Elaine Venson



Brasília, DF
2016

Alex Cortes Alves

Comparação da acurácia de estimativas de esforço produzidas a partir de Pontos de Função Cosmic e de Story Points

Monografia submetida ao curso de graduação em (Engenharia de Software) da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em (Engenharia de Software).

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: Msc. Elaine Venson

Brasília, DF

2016

Alex Cortes Alves

Comparação da acurácia de estimativas de esforço produzidas a partir de Pontos de Função Cosmic e de Story Points/ Alex Cortes Alves. – Brasília, DF, 2016-

69 p. : il. (algumas color.) ; 30 cm.

Orientador: Msc. Elaine Venson

Trabalho de Conclusão de Curso – Universidade de Brasília - UnB
Faculdade UnB Gama - FGA , 2016.

1. Palavra-chave01. 2. Palavra-chave02. I. Msc. Elaine Venson. II. Universidade de Brasília. III. Faculdade UnB Gama. IV. Comparação da acurácia de estimativas de esforço produzidas a partir de Pontos de Função Cosmic e de Story Points

CDU 02:141:005.6

Alex Cortes Alves

Comparação da acurácia de estimativas de esforço produzidas a partir de Pontos de Função Cosmic e de Story Points

Monografia submetida ao curso de graduação em (Engenharia de Software) da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em (Engenharia de Software).

Trabalho aprovado. Brasília, DF, 01 de junho de 2013:

Msc. Elaine Venson
Orientador

Titulação e Nome do Professor
Convidado 01
Convidado 1

Titulação e Nome do Professor
Convidado 02
Convidado 2

Brasília, DF
2016

**A dedicatória é opcional. Caso não deseje uma, deixar todo este
arquivo em branco.**

*Este trabalho é dedicado às crianças adultas que,
quando pequenas, sonharam em se tornar cientistas.*

Agradecimentos

A inclusão desta seção de agradecimentos é opcional, portanto, sua inclusão fica a critério do(s) autor(es), que caso deseje(em) fazê-lo deverá(ão) utilizar este espaço, seguindo a formatação de *espaço simples e fonte padrão do texto (arial ou times, tamanho 12 sem negritos, aspas ou itálico)*.

Caso não deseje utilizar os agradecimentos, deixar toda este arquivo em branco.

*“It ain’t about how hard you hit,
it’s about how hard you can get hit
and keep moving forward.
How much you can take and keep moving forward.
That’s how winning is done!
(Rocky Balboa)*

Resumo

Em projetos de desenvolvimento de software ágil, os Story Points são geralmente utilizados como uma forma de estimar o esforço das atividades. Os Story Points auxiliam uma equipe na escolha de tarefas que serão executadas em uma Sprint, tendo o velocity dessa equipe, é possível tentar estimar quantos Story Points podem ser executados dentro de uma Sprint. Porém, tal técnica não permite definir o tamanho real do sistema, uma vez que a pontuação é particular de cada projeto e equipe de desenvolvimento, logo, não é possível realizar um Benchmarking dos projetos dentro ou fora de uma empresa. Técnicas de medição como o Ponto de Função COSMIC possuem padrões definidos, fazendo com que o tamanho do software medido não dependa da opinião da equipe ou da experiência do medidor. A proposta desse trabalho é realizar um estudo de caso para comparar a acurácia das estimativas de esforço geradas a partir dessas duas técnicas de medição. Comparando os valores de estimativa e os valores reais após o desenvolvimento das atividades.

Palavras-chaves: estimativa. story point. cosmic.

Abstract

This is the english abstract.

Key-words: latex. abntex. text editoration.

Lista de ilustrações

Figura 1 – Reta de regressão	33
Figura 2 – Reta de regressão com pontos adicionados	34
Figura 3 – Erros Relativos	35
Figura 4 – Sequência de atividades do Planning Poker	49
Figura 5 – Processo de determinação de Estratégia de Medição	54
Figura 6 – Processo de determinação de Estratégia de Medição	58
Figura 7 – Fluxograma das atividades de medição de software	62

Lista de tabelas

Tabela 1 – Valores estimados e reais das atividades	33
Tabela 2 – Erros relativos das estimativas das atividades	34
Tabela 3 – Acurácia x Precisão	42
Tabela 4 – Valores de acurácia alcançados (fonte: (USMAN et al., 2014)) .	43
Tabela 5 – Símbolos do Diagrama de Contexto (fonte: COSMIC Manual) .	56
Tabela 6 – Cronograma de Trabalho	66

Lista de abreviaturas e siglas

Sp Story Point

PFC Ponto de Função COSMIC

US User Story

lauro cesar este é o meu nome

Sumário

I	INTRODUÇÃO	25
1	INTRODUÇÃO	27
	Introdução	27
1.1	Considerações iniciais do capítulo	27
1.2	Contexto	27
1.3	Problema	28
1.4	Objetivo	29
1.5	Metodologia	29
1.5.1	Regressão Linear	31
1.5.1.1	Exemplificando	32
1.6	Organização do trabalho	35
II	ESTIMATIVA DE SOFTWARE	37
2	ESTIMATIVA DE SOFTWARE	39
	Estimativa de Software	39
2.1	Considerações Iniciais do Capítulo	39
2.2	Medição de Software	39
2.3	Tamanho Funcional	40
2.4	Estimativa de Esforço	41
2.5	Acurácia de Estimativa	41
2.5.1	Acurácia x Precisão	41
2.5.2	Acurácia dos métodos de estimativa	43
III	STORY POINTS	45
3	STORY POINTS	47

Story Points	47
3.1 Considerações Iniciais do Capítulo	47
3.2 Planejamento Ágil	47
3.3 Planning Poker	48
3.4 Story Points	49
3.4.1 Velocity	50
IV PONTOS DE FUNÇÃO COSMIC	51
4 PONTOS DE FUNÇÃO COSMIC	53
Pontos de Função COSMIC	53
4.1 Considerações Iniciais do Capítulo	53
4.2 Fase de Estratégia de Medição	53
4.2.1 Determinar o Propósito da Medição	53
4.2.2 Determinar o Escopo da Medição	55
4.2.3 Identificar os Usuários Funcionais	55
4.2.4 Identificar o Nível de Granularidade	56
4.3 Fase de Mapeamento	57
4.3.1 Identificar os Processos Funcionais	57
V PROPOSTA	59
5 PROPOSTA	61
Proposta	61
5.1 Considerações Iniciais do Capítulo	61
5.2 Execução do Estudo de Caso	61
5.2.1 Organização	64
5.2.2 Grupo de estudantes	64
5.3 Resultados Esperados	65
5.4 Cronograma de trabalho	65

REFERÊNCIAS	67
--------------------------	-----------

Parte I

Introdução

1 Introdução

1.1 Considerações iniciais do capítulo

1.2 Contexto

Estimativas de custo e esforço de software são atividades cruciais no ciclo de vida de desenvolvimento de um software. Estimativas são as entradas-chave para alocação de recursos e definição de prazos. A falta de acurácia nas estimativas é responsável por um grande número de problemas relacionados ao gerenciamento de projetos (PEIXOTO; AUDY; PRIKLADNICKI, 2010).

Por ter uma natureza volátil e flexível, as metodologias ágeis estão sendo usadas em um grande número de projetos na indústria. No desenvolvimento ágil, o software é desenvolvido em pequenas iterações e as mudanças são bem vindas. Devido a essa natureza dinâmica, se torna muito difícil estimar custo e tempo em ambientes de desenvolvimento ágil. Os modelos de estimativa utilizados em tais metodologias não possuem nenhuma fórmula matemática para estimar esforço e custo, logo, se tornam pouco eficientes (POPLI; CHAUHAN, 2014).

Em metodologias ágeis, os requisitos são sempre descritos em *User Stories*. A *User Story* é um requisito independente, negociável, estimável, pequeno e testável. E as estimativas nessas metodologias são feitas com base nesses requisitos (POPLI; CHAUHAN, 2015).

Story Points é uma medida relativa muito usada para estimar tamanho de software em Ágil. Os times decidem o quanto vale um ponto, e pontuam as *User Stories* nas quais irão trabalhar durante a sprint atual. Porém, um time pode pontuar atividades diferentemente de outro time, não sendo possível comparar atividades executadas por times diferentes (HAMOUDA, 2014).

Além do Story Point, existe um número de padrões internacionais que disponibilizam métodos de medição funcional bem documentados independente da

tecnologia adotada ou do processo de desenvolvimento (COMMEYNE; ABRAN; DJOUAB, 2016). Segundo Ungan, Çizmeli e Demirörs (2014) o Ponto de Função COSMIC é uma das técnicas de medição funcional mais recente e popular, tanto na academia quanto na indústria. O software é medido em um nível de funcionalidade padronizado chamado de Processo Funcional (UNGAN; ÇIZMELI; DEMIRÖRS, 2014).

O Ponto de Função COSMIC é um padrão internacional para medida de requisitos funcionais de software. Como resultado, ele fornece um número que representa o tamanho funcional do software e pode ser usado para comparação com outros sistemas e equipes. Como exigido pela ISO, o COSMIC foi desenvolvido para ser independente de decisões de implementações contidas em qualquer artefato operacional do software. Isso significa que o tamanho do software pode ser medido tanto de um sistema já desenvolvido quanto dos requisitos de um sistema que será implementado (COMMEYNE; ABRAN; DJOUAB, 2016).

1.3 Problema

Segundo (COHN, 2005) o Story Point é uma unidade de medida que expressa o tamanho geral de uma User Story, funcionalidade ou outra peça de trabalho e não existe uma fórmula para definição do tamanho de uma história. O importante é que os valores sejam relativos, sendo que uma história definida como dois Story Points deve ser o dobro de uma história pontuada com apenas um.

O valor de Story Points significa que a equipe concorda em atribuir tais pontos a determinada User Story. Porém, os Story Points não são padronizados como uma métrica, e equipes diferentes podem não pontuar uma mesma User Story igualmente (FEHLMANN; SANTILLO, 2010).

O Story Point não indica explicitamente o tamanho do software desenvolvido, logo, não é possível determinar os níveis de produtividade alcançados em um projeto, e nem comparar a performance entre diferentes projetos e organizações (COMMEYNE; ABRAN; DJOUAB, 2016).

A questão de pesquisa desse estudo é:

Em um contexto ágil, as estimativas em Pontos de Função Cosmic geram resultados mais confiáveis do que estimativas geradas em Story Points?

1.4 Objetivo

O objetivo geral deste trabalho é realizar uma comparação da acurácia de duas técnicas de estimativa de esforço, selecionando atividades pontuadas por uma equipe ágil em Story Points e calculando o valor em Pontos de Função Cosmic das mesmas atividades.

Os objetivos específicos são:

- Selecionar funcionalidades a serem estudadas;
- Acompanhar e registrar a pontuação, em Story Points, das funcionalidades;
- Realizar a contagem de Pontos de Função Cosmic das funcionalidades;
- Coletar valores de esforço real;
- Realizar cálculo de regressão linear com valores estimados e reais;
- Analisar valores de acurácia Story Points vs. Pontos de Função Cosmic.

1.5 Metodologia

A pesquisa proposta neste trabalho possui uma natureza aplicada, a qual tem como objetivo gerar conhecimentos para aplicação prática dirigidos à solução de problemas específicos (Moresi et al., 2003).

Do ponto de vista da abordagem, foi definida uma abordagem quantitativa, onde os resultados são traduzidos em números e requer uso de recursos e técnicas de estatística (Moresi et al., 2003). No estudo proposto será utilizada a técnica de regressão linear, para comparar a acurácia das técnicas de estimativa de esforço descritas.

Os procedimentos técnicos selecionados para execução do trabalho são pesquisa bibliográfica e estudo de caso.

- **Pesquisa Bibliográfica:** Essa pesquisa é desenvolvida com base em material já elaborado, constituído principalmente de livros e artigos científicos (GIL, 2002). Serão pesquisados periódicos nas principais bases científicas afim de dar suporte ao referencial teórico do trabalho.
- **Estudo de Caso:** É um método de pesquisa de campo, no qual se investiga os fenômenos à medida em que acontecem, sem que o pesquisador interfira de forma significativa (FIDEL, 1984). No trabalho proposto o estudo de caso será executado em um grupo de uma disciplina de Métodos de Desenvolvimento de Software na Universidade de Brasília em paralelo com uma equipe de desenvolvimento de uma empresa terceirizada alocado em um Ministério X.

O estudo de caso é usado com frequência em áreas como psicologia, sociologia, ciência política, serviço social e administração. Nessas áreas o estudo de caso tem como objetivo não só aumentar o conhecimento a respeito de algo, mas também propor mudanças ao fenômeno que está sendo estudado (RUNESON et al., 2012).

Na área de Sistemas da Informação, o estudo de caso também já é mais maduro quando comparado com a Engenharia de Software. Runeson et al. (2012) apresenta propriedades dos objetos de estudo em Engenharia de Software que diferenciam de estudos em Sistemas da Informação:

- Eles são corporações privadas ou órgãos públicos que estão *desenvolvendo* software, e não que estão *utilizando-o*;
- Eles são organizações orientadas a *projeto*, e não orientadas a *linhas* ou *funções*;
- Os trabalhos em estudo são *trabalhos de engenharia avançados* realizados por pessoas altamente capacitadas, ao contrário de *trabalhos de rotina*
- Existe um objetivo de melhorar práticas de engenharia, o que significa que existe um componente de pesquisa de criação.

1.5.1 Regressão Linear

Para se calcular a acurácia das estimativas obtidas através do estudo de caso, será utilizada a técnica estimativa de regressão linear, que é descrita a seguir.

No processo de tomada de decisões, muitas vezes é necessário fazer previsões. Se torna muito mais fácil tomar decisões a respeito de uma variável quando é possível estabelecer uma relação entre essa e alguma outra variável conhecida. Para que seja estabelecida tal relação, é necessário que haja uma relação de causa-efeito entre elas, isto é, a variação de uma pode ser atribuída a variação da outra (REIS, 2008).

Para o contexto desse trabalho, a relação entre estimativa e esforço real, será representada pela regressão linear como feito por (COMMEYNE; ABRAN; DJOUAB, 2016), calculada a partir das pontuações em Pontos de Função Cosmic, Story Points e o esforço real de cada atividade.

Para se calcular a regressão linear, é necessário conhecer a reta de regressão $Y = a + bX$, onde:

- Y é a variável dependente, no caso, o esforço.
- X é a variável independente, no caso, as pontuações em Pontos de Função Cosmic ou Story Points.
- a e b são dados pelas equações 1.1 e 1.2:

$$a = \bar{Y} - b\bar{X} \quad (1.1)$$

$$b = \frac{n \sum (X_i Y_i) - \sum X_i \sum Y_i}{n \sum X_i^2 - (\sum X_i)^2} \quad (1.2)$$

- \bar{Y} é a média dos valores de Y conhecidos, ou seja, média dos valores de esforço real das atividades selecionadas.
- \bar{X} é a média dos valores de X conhecidos, ou seja, média dos Pontos de Função Cosmic estimados ou dos *Story Points* pontuados para as atividades selecionadas.

Após a criação da reta de regressão, calcula-se o erro relativo, que é o quanto cada atividade varia da média. O erro relativo é calculado para cada atividade de acordo com a equação 1.3.

$$ErroRelativo = \frac{(EsforçoReal - EsforçoEstimado)}{EsforçoReal} \quad (1.3)$$

E então é calculado o MMRE, do inglês Mean Magnitude of Relative Error (Magnitude Média do Erro Relativo), que mede a diferença entre o esforço estimado e o real relativo ao esforço real. Calcula-se o MMRE de acordo com a equação 1.4

$$MMRE = \frac{1}{n} \sum_{i=1}^n \left| \frac{(EsforçoReal - EsforçoEstimado)}{EsforçoReal} \right| \quad (1.4)$$

1.5.1.1 Exemplificando

Para exemplificar o cálculo de regressão linear no contexto do trabalho proposto, será considerada a Tab. 1.5.1.1 que apresenta quinze atividades e seus devidos tamanhos estimados em Pontos X (Pontuação fictícia para o contexto do exemplo) e os esforços reais em horas.

Para a reta de regressão $Y = a + bX$, os valores de a e b encontrados no exemplo descrito são:

- a = 0,78
- b = 0,12

Logo, a equação da reta de regressão para as atividades exemplificadas é:

$$Y = 0,78 + 0,12X \quad (1.5)$$

A Figura 1.5.1.1 apresenta o gráfico obtido da reta de regressão.

Adicionando os pontos da Tab. 1.5.1.1 ao gráfico da reta facilita a análise de como as estimativas de tamanho das tarefas se comportam, como apresentado na fig. 1.5.1.1.

Tabela 1: Valores estimados e reais das atividades

Número da Atividade	Tamanho Estimado (Pt. X)	Esforço Real (h)
1	30	5
2	45	6
3	15	2
4	32	5
5	35	4
6	60	8
7	50	6
8	10	1
9	10	2
10	12	5
11	24	4
12	17	2
13	50	8
14	52	8
15	23	3

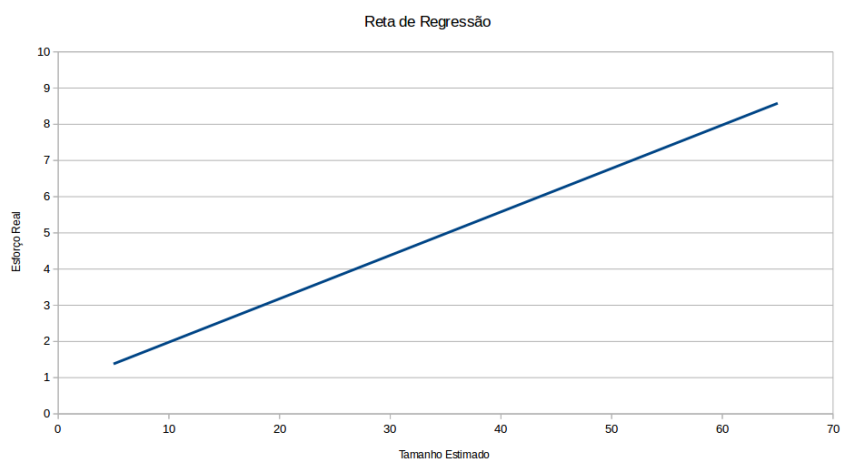


Figura 1: Reta de regressão

É possível observar na reta que, com exceção de alguns pontos, os valores de esforço real e tamanho estimado das tarefas tendem a acompanhar a reta de regressão, o que mostra como a Regressão Linear pode ser usada para criar um modelo de estimativa de esforço. Em uma próxima iteração, essa fórmula poderia ser utilizada para estimar o esforço das atividades.

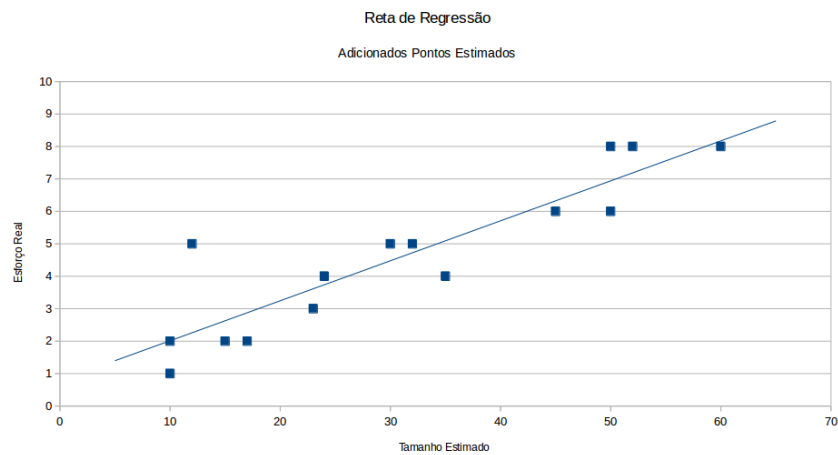


Figura 2: Reta de regressão com pontos adicionados

Deve-se então, calcular os erros relativos das atividades, como mostrado na Tab. 2.5.1 e representado na Fig. 1.5.1.1.

Tabela 2: Erros relativos das estimativas das atividades

Número da Atividade	Erro Relativo (%)
1	12,40
2	-3,00
3	-29,00
4	7,60
5	-24,50
6	0,25
7	-13,00
8	-98,00
9	1,00
10	55,60
11	8,50
12	-41,00
13	15,25
14	12,25
15	-18

Esse exemplo foi criado para demonstrar como serão feitos os cálculos de Regressão Linear, e como os valores se comportam e podem ser utilizados para

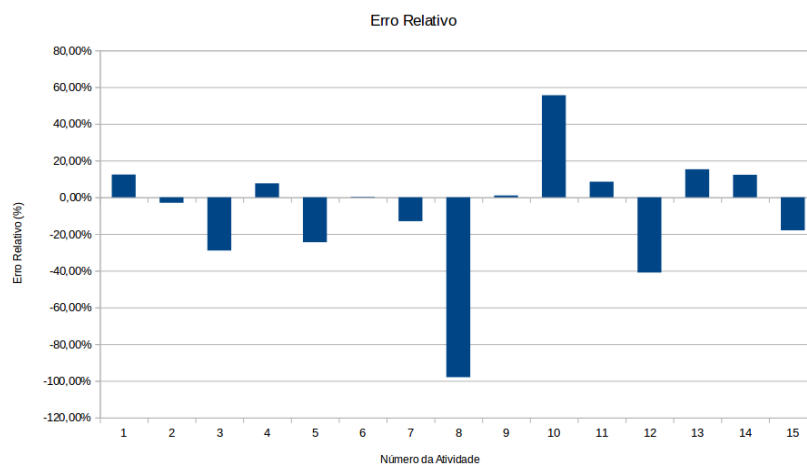


Figura 3: Erros Relativos

gerar um modelo de estimativa para próximas iterações. A fórmula criada deve, então, ser refinada com os valores de novas iterações.

1.6 Organização do trabalho

Esse trabalho é organizado em quatro partes, a segunda parte apresenta conceitos sobre Estimativas de Software, a segunda parte apresenta os *Story Points* e como pontuá-los, a terceira parte apresenta os Pontos de Função COSMIC e o processo a ser seguido para medir um software em PFC e a última parte apresenta a proposta do trabalho mostrando as etapas que serão seguidas para execução do estudo de caso e o cronograma de trabalho.

Parte II

Estimativa de Software

2 Estimativa de Software

2.1 Considerações Iniciais do Capítulo

Este capítulo apresenta conceitos pertinentes à área de estimativa de software, como, Medição, Métrica, Tamanho Funcional, Estimativa de Esforço e as definições de Acurácia e Precisão em estimativa de software.

2.2 Medição de Software

Medição é o processo em que números ou símbolos são atribuídos a atributos de entidades do mundo real de modo a descrevê-las de acordo com regras bem definidas. Ou seja, a medição deve capturar informações sobre atributos de uma entidade a fim de poder descrevê-la ([FENTON; BIEMAN, 2014](#)).

A medição de software é um mecanismo ideal para se obter *feedback* e avaliação. A medição e as informações levantadas são úteis para todos os envolvidos como desenvolvedores, gerentes, clientes e a própria organização ([BASILI, 1992](#)).

A métrica de software é um método para se determinar quantitativamente a extensão de processos, produtos ou projetos de software. Como a indústria de software possui problemas com produtividade e qualidade, os engenheiros reconhecem a necessidade de entender e melhorar o processo de desenvolvimento para que seja possível aplicar mudanças necessárias para obter melhores níveis de produtividade e qualidade ([GOPAL et al., 2002](#)).

Na engenharia de software, a medição é uma atividade essencial. Muitos desenvolvedores de software medem características de seus software a fim de entender se os requisitos são consistentes e completos, assegurar que o *design* possui uma boa qualidade ou até saber se o software está pronto para ser entregue. Um gerente de projeto pode usar a medição para controlar prazos e custos, uma organização pode realizar medições de avaliação de seu processo de desenvolvimento ou o próprio cliente pode medir o produto entregue para avaliar se o mesmo atende a

todos os requisitos (FENTON; BIEMAN, 2014).

O tamanho funcional de um software é utilizado para avaliar e prever alguns aspectos do processo de produção como o esforço gasto em uma atividade, o custo de um sistema ou a produtividade de uma equipe de desenvolvimento. Para se medir o tamanho funcional de um sistema existem duas abordagens principais, a abordagem posterior, como LOC, e a abordagem anterior, que são técnicas que estimam o tamanho de um sistema baseadas nos requisitos definidos (TRAN-CAO; LEVESQUE; ABRAN, 2002).

2.3 Tamanho Funcional

O tamanho funcional de um sistema é obtido a partir da perspectiva do usuário sobre o sistema desenvolvido.

Métodos de medição de tamanho funcional como Análise de Ponto de Função e COSMIC se tornaram populares ao longo dos anos, tais métodos apresentam vantagens sobre outras técnicas, como por exemplo, podem ser executados em fases iniciais de um projeto e são independentes de tecnologias e linguagens de programação (UNGAN; ÇIZMELI; DEMIRÖRS, 2014).

- **Análise de Ponto de Função** - Publicada por Albrecht (1979), a técnica da Análise de Ponto de Função quantifica as funções contidas em um software que sejam significativas para o usuário. Ela é relacionada diretamente com os requisitos de usuário. Usando um conjunto de critérios padronizados, cada uma das funções é um índice numérico de acordo com seu tipo e complexidade. Esses índices são somados resultando em uma medida inicial de tamanho que será normalizada incorporando fatores relacionados ao software como um todo (ABOUT... ,).
- **Ponto de Função COSMIC** - O método faz parte da segunda geração de medidas de tamanho funcional e fornece princípios, regras e um processo para medir um tamanho funcional padronizado. O COSMIC é associado aos movimentos de dados do sistema (*Entry, Write, Read e Exit*), em que cada

movimento de dados realizado pelo sistema deve ser contado como um PFC (Ponto de Função COSMIC) (COSMIC...).

Em metodologias ágeis, o Story Point é uma unidade de medida que expressa o tamanho geral de uma User Story, uma feature ou outro tipo de peça de trabalho. Para se estimar tamanho em Story Point deve-se atribuir uma pontuação a uma devida atividade de trabalho, essa pontuação não é importante, o importante é que as pontuações das outras atividades sejam relativas a essa. Uma atividade de dois Story Points deve demandar o dobro de esforço de uma atividade de apenas um Story Point (COHN, 2005).

2.4 Estimativa de Esforço

Estimativa de esforço de software é o processo de previsão do esforço que será necessário para desenvolver o sistema (WEN et al., 2012). Este é um elemento crítico na gestão de projetos, o uso inadequado de estimativas de esforço pode levar a falhas no projeto como prazos definidos inadequadamente levando a atrasos na entrega (TARIQ et al., 2015).

2.5 Acurácia de Estimativa

2.5.1 Acurácia x Precisão

A *Accuracy...* (1994) define acurácia e precisão como:

- **Acurácia** - O grau de concordância entre o resultado experimental e o valor de referência aceito;
- **Precisão** - O grau de concordância entre resultados de testes independentes obtidos em condições estipuladas.

Em outras palavras, e no contexto de Estimativa de Software, o grau de acurácia é o quanto uma estimativa se aproxima do valor real. E o grau de precisão é o quanto o resultado de uma estimativa se aproxima de outros resultados da

mesma estimativa, ou seja, espera-se que uma estimativa com alto grau de precisão obtenha os mesmos resultados sempre mesmo que não sejam próximos do valor real.

As imagens apresentadas na Tab. 3 representam graficamente os conceitos de precisão e acurácia:

Tabela 3: Acurácia x Precisão



O primeiro quadrante possui baixa precisão, pois os resultados não são próximos entre si, e possui baixa acurácia pois os resultados não estão próximos do alvo central.

O segundo quadrante possui alta precisão, pois os resultados apresentam valores próximos entre si, porém ainda não possuem valores próximos ao alvo central, indicando baixa acurácia.

O terceiro quadrante apresenta resultados próximos ao alvo central, o que indica alta acurácia, porém ainda não apresenta resultados próximos entre si.

O quarto quadrante apresenta alto grau de precisão e de acurácia, pois, além dos resultados alcançarem aproximadamente os mesmos valores em todas as tentativas, eles também apresentam valores próximos ao alvo central.

2.5.2 Acurácia dos métodos de estimativa

A acurácia de uma medida depende do instrumento utilizado para se realizar a media, assim como as definições da medição (FENTON; BIEMAN, 2014). Usman et al. (2014) estudou as publicações que utilizavam técnicas de estimativa de software em projetos ágeis. Nesse estudo ele utilizou o MMRE como métrica para analisar a acurácia das técnicas de estimativa.

Nos resultados apresentados por Usman et al. (2014), o *Planning Poker*, mesmo sendo uma das técnicas mais utilizadas, apresenta acurácia de 48%, valor baixo quando comparado com estudos que utilizaram a Regressão Linear, que variam entre 66% e 90% de acurácia.

As acurácias de métodos de estimativa levantadas por Usman et al. (2014) são apresentadas na Tab. 4.2.3

Tabela 4: Valores de acurácia alcançados (fonte: (USMAN et al., 2014))

Técnica	Acurácia alcançada % (MMRE)
Planning Poker	48.0
Pontos de Caso de Uso (UCP)	10.0 - 21.0
Variações do UCP	2.0 - 11.0
Opinião de especialista	28.0 - 38.0
Regressão Linear	66.0 - 90.0
Redes Neurais	6.0 - 90.0

Parte III

Story Points

3 Story Points

3.1 Considerações Iniciais do Capítulo

Este capítulo apresenta os conceitos básicos do planejamento em metodologias ágeis, mostrando o que são os *Story Points* e como é realizado o processo de planejamento *Planning Poker*.

3.2 Planejamento Ágil

O *Extreme Programming* possui dois níveis de planejamento, o planejamento de iteração e o planejamento da *release*. O planejamento da iteração é de curto prazo e muito detalhado, esse é um bom momento para se entrar em detalhes sobre como implementar uma história. O planejamento da *release* é de alto nível de longo alcance, nesse planejamento o time não se preocupa com a precisão dos detalhes e foca mais em uma visão geral do projeto (GRENNING, 2002).

Planning Poker combina a opinião da equipe, analogia e desagregação em uma atividade simples com estimativas rápidas, porém, confiáveis (COHN, 2005). Grenning (2002) afirma que a estimativa de histórias possui menos acurácia do que estimativas convencionais, porém, é melhor possuir um conjunto de estimativas com acurácia razoável do que apenas um par de estimativas precisas. E, com o tempo, a equipe deve se tornar melhor e confiar em sua intuição ao estimar atividades (GRENNING, 2002).

O *Story Point* é uma unidade de medida que expressa o tamanho de uma *Story*, esses pontos são resultados do *Planning Poker* realizado na reunião de planejamento em metodologias ágeis (COHN, 2005).

3.3 Planning Poker

O *Planning Poker* é considerada uma técnica de estimativa em grupo construída no princípio de que mais cabeças são melhores do que uma (MAHNIČ; HOVELJA, 2012). Tem como objetivo gerar estimativas confiáveis e rápidas utilizando analogia entre as atividades e a opinião da equipe envolvida no desenvolvimento do projeto (COHN, 2005).

É realizado em dois momentos do projeto, no planejamento da *Release*, onde são estimadas todas as atividades identificadas antes do início do projeto, ainda sem grande detalhamento e com pouca precisão. E no planejamento de cada *Sprint*, onde são estimadas apenas as atividades da *Sprint* que irá começar, neste ponto já são discutidos detalhes da implementação das *User Stories*, como tem foco maior nos detalhes, a precisão da estimativa da *Sprint* tende a ser maior (COHN, 2005) (GRENNING, 2002).

Todo o time envolvido no projeto deve fazer parte do *Planning Poker*, incluindo programadores, testadores, engenheiros, analistas e *designers*, isso faz com que o planejamento tenha vários pontos de vista, gerando uma estimativa mais confiável (COHN, 2005).

É considerado que o *Planning Poker* gera melhores estimativas do que técnicas baseadas em opiniões individuais de especialistas por garantir que todos os desenvolvedores participem igualmente do processo. A participação de pessoas com pontos de vista diferentes ajuda a reduzir o super-otimismo de outras estimativas, por identificar mais problemas que podem afetar o desenvolvimento do projeto (MAHNIČ; HOVELJA, 2012).

(GRENNING, 2002) descreve as atividades do *Planning Poker* de acordo com a Fig. 3.3

- **Seleção da User Story** - As histórias que farão parte da *Sprint* que está sendo planejada serão pontuadas e discutidas individualmente. Nessa etapa é selecionada qual *User Story* será pontuada no momento.
- **Leitura da User Story** - A *User Story* é lida e explicada pelo cliente.

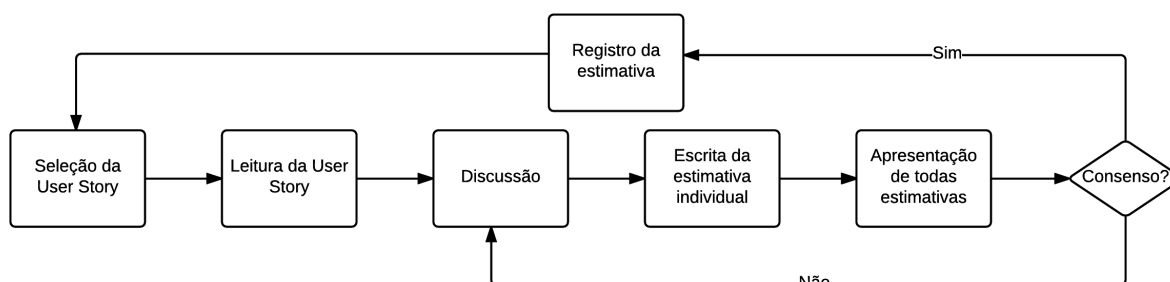


Figura 4: Sequência de atividades do Planning Poker

- **Discussão** - A equipe de desenvolvimento deve, então, discutir os detalhes da *User Story*, apontar as visões individuais dos membros, e quais podem ser as eventuais dificuldades durante a implementação.
- **Escrita da estimativa individual** - Cada participante do *Planning Poker* deve escrever o valor de sua estimativa em *Story Points* ainda sem que os outros participantes vejam. Os valores que podem ser escolhidos são pré-definidos em 0, 1, 2, 3, 5, 8, 13, 20, 40 e 100.
- **Apresentação de todas as estimativas** - Assim que todos os participantes fizeram suas estimativas, todos apresentam os valores escolhidos ao mesmo tempo.
- **Registro da estimativa** - Assim que o time alcançar um consenso, a estimativa é registrada e é selecionada uma nova *User Story* para ser pontuada.

3.4 Story Points

O *Story Point* é uma medida relativa muito utilizada como estimativa em metodologias ágeis, o time decide o quanto vale um ponto e, baseado nesse tamanho, determina quantos pontos cada atividade possui (HAMOUDA, 2014).

Existem duas formas para se basear o tamanho de um *Story Point*, no primeiro, deve-se selecionar a menor atividade do *backlog*, e atribuir a essa o valor

de 1 *Story Point*. No segundo, se é esperado que as histórias tenham tamanho entre 1 e 10 *Story Points*, por exemplo, é escolhida uma história considerada de complexidade média e atribuído um valor mediano dentro desse intervalo. E, a partir dessa definição inicial do *Story Point*, é executada a técnica de *Planning Poker* para se estimar as outras atividades da *release* ou *Sprint* (COHN, 2005).

Para se obter um planejamento mais eficiente, é importante que o tamanho estimado das histórias mantenha a relatividade, o ideal é que uma história pontuada com dois *Story Points* demande o dobro de esforço de uma atividade pontuada com apenas um (COHN, 2005). Ao decorrer do projeto a equipe se torna mais experiente para estimativa e mais intuitiva será a pontuação (GRENNING, 2002).

Segundo Cohn (2005) o *Story Point* é puramente uma medida de tamanho do trabalho a ser executado. A duração de um projeto não é estimada, e sim derivada do número total de *Story Points* dividido pelo *Velocity* da equipe de desenvolvimento (COHN, 2005).

3.4.1 Velocity

O *Velocity* é o número de histórias ou *Story Points* que a equipe pode executar no tempo de uma *Sprint*. Uma das maiores dificuldades no planejamento é prever o *velocity* do time, para isso o *Velocity* é gerado baseado em dados históricos do projeto, então o planejamento das *Sprints* é feito com base na quantidade de pontos concluídos em *Sprints* anteriores (OMANOVIC; BUZA, 2013).

No *Scrum*, o *velocity* é o termo que representa quantos *Story Points* a equipe é capaz de entregar em uma *Sprint*, ele é obtido observando *Sprints* anteriores e assumindo que a produtividade do time permanece constante (GLOSSARY...). Com valores constantes de *Velocity* pode-se assumir que o planejamento e a execução das *Sprints* estão bem alinhados (OMANOVIC; BUZA, 2013).

Parte IV

Pontos de Função COSMIC

4 Pontos de Função COSMIC

4.1 Considerações Iniciais do Capítulo

Esta capítulo apresenta os conceitos e o processo para realização de uma estimativa em Pontos de Função COSMIC. O Manual do COSMIC define que três fases para se realizar a medição em PFC:

- **Fase de Estratégia de Medição**
- **Fase de Mapeamento**
- **Fase de Medição**

4.2 Fase de Estratégia de Medição

Essa fase descreve alguns parâmetros que devem ser identificados antes de se começar a medição em si:

- O *propósito da medição*;
- O *escopo geral* do software que será medido;
- Os *usuários funcionais* de cada parte do software;
- O *nível de granularidade* dos artefatos do software disponíveis.

O processo para determinação da estratégia de definição apresentada pelo Manual COSMIC é ilustrado na Fig. 4.3.

4.2.1 Determinar o Propósito da Medição

O MANUAL COSMIC diz que "Propósito de Medição" é a definição do porque uma medição é necessária e como o resultado dessa será usado.

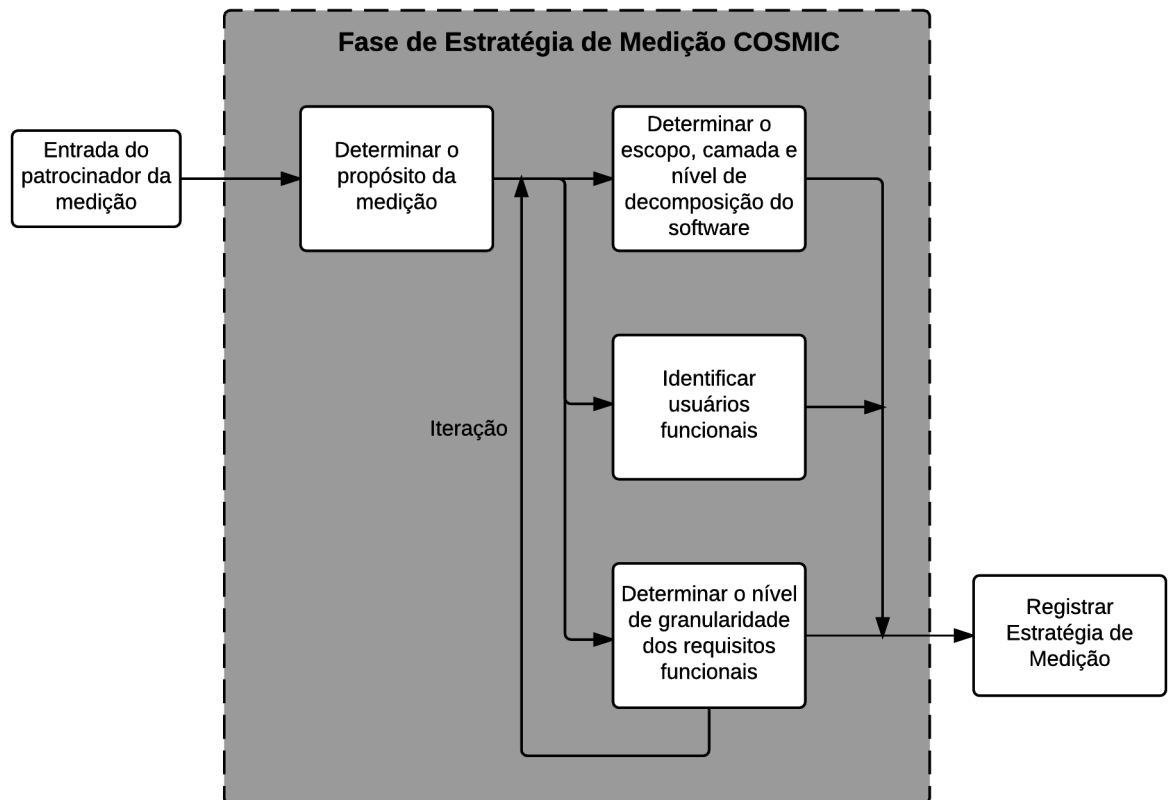


Figura 5: Processo de determinação de Estratégia de Medição

Ao identificar o o propósito da medição é possível definir, por exemplo, em que momento a medição será realizada e quais os artefatos serão necessários. No caso de estar medindo a fim se obter uma estimativa para elaborar cronograma ou fazer orçamento, o software deve ser medido ao início do ciclo de vida do projeto utilizando as especificações de requisitos funcionais do sistema.

O propósito da medição ajuda a identificar itens que serão necessários ao decorrer da atividade de medição do software:

- O escopo a ser medido e os artefatos necessários;
- Os usuários funcionais do sistema;

- Em qual momento do ciclo de vida do desenvolvimento a medição ocorrerá;
- A acurácia necessária da medição. Com isso é possível identificar se o método COSMIC padrão é suficiente ou deve-se usar uma versão de aproximação do método COSMIC.

4.2.2 Determinar o Escopo da Medição

Para o MANUAL COSMIC o "Escopo da Medição" é o conjunto de requisitos funcionais do usuário que serão inclusos na atividade de medição. Especificamente para o COSMIC, a expressão "escopo" se refere a uma parte do software que será medida individualmente.

O escopo de uma parte do software a ser medido deve ser obtido na fase de "Propósito da Medição" e não deve se estender a mais de uma camada do software.

O software definido como um escopo geral pode ser subdividido em pedaços individuais de software cada um com seu próprio escopo de medição. Alguns motivos para se fazer a subdivisão são o software está em mais de uma camada, responsabilidades organizacionais diferentes ou a necessidade de se distinguir alguns itens entregáveis.

4.2.3 Identificar os Usuários Funcionais

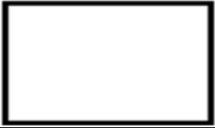

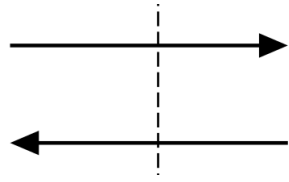
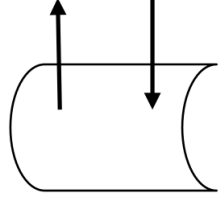
Diferentes tipos de usuários podem interagir com diferentes requisitos funcionais do sistema, logo, o tamanho funcional irá variar com a escolha dos usuários funcionais da medição Manual COSMIC.

O usuário funcional é um tipo de usuário que faz parte dos requisitos funcionais de um software recebendo e/ou enviando dados ao sistema Manual COSMIC. No método COSMIC é essencial separar os usuários funcionais dos outros possíveis usuários do sistema.

Quando se está definindo o escopo da medição e os usuários funcionais, pode ser útil desenhar um Diagrama de Contexto para o software sendo medido. Diagramas de contexto são usados para demonstrar o escopo de uma medição dentro do contexto do usuário funcional Manual COSMIC.

Os símbolos padrão para o diagrama de contexto são mostrados na tab. 4:

Tabela 5: Símbolos do Diagrama de Contexto (fonte: COSMIC Manual)

Símbolo	Interpretação
	O pedaço de software a ser medido (caixa com borda grossa)
	Qualquer usuário funcional do software sendo medido
	As setas representam todos os movimentos de dados cruzando as fronteiras entre um usuário funcional e o software medido
	As setas representam todos os movimentos de dados entre o software sendo medido e o armazenamento de dados

Com o diagrama de contexto desenhado já se torna possível identificar os movimentos de dados do software sendo medido.

4.2.4 Identificar o Nível de Granularidade

Nas fases iniciais do desenvolvimento de um projeto de software os requisitos são especificados em alto nível e com poucos detalhes. Na medida em que o projeto avança os requisitos são refinados alcançando um menor nível e revelando maiores detalhes do software. Esses diferentes níveis de detalhes dos requisitos são conhecidos como diferentes níveis de granularidade Manual COSMIC.

Uma medição precisa do tamanho funcional de um software requer que os requisitos funcionais estejam em um nível de granularidade onde é possível identificar os processos funcionais e os movimentos de dados Manual COSMIC.

Se algum requisito precisa ser medido antes que seja suficientemente detalhado ele pode ser medido utilizando uma técnica de aproximação. Essas técnicas definem como os requisitos podem ser medidos em um alto nível de granularidade Manual COSMIC.

4.3 Fase de Mapeamento

Essa seção apresenta definições, princípios, regras e o método para o processo de mapeamento. A FIGURA A SEGUIR apresenta o processo para a fase de mapeamento do COSMIC:

- Identificar os Processos Funcionais;
- Identificando objetos de interesse e grupos de dados;
- Identificando atributos de dados;
- Identificando movimentos de dados;

4.3.1 Identificar os Processos Funcionais

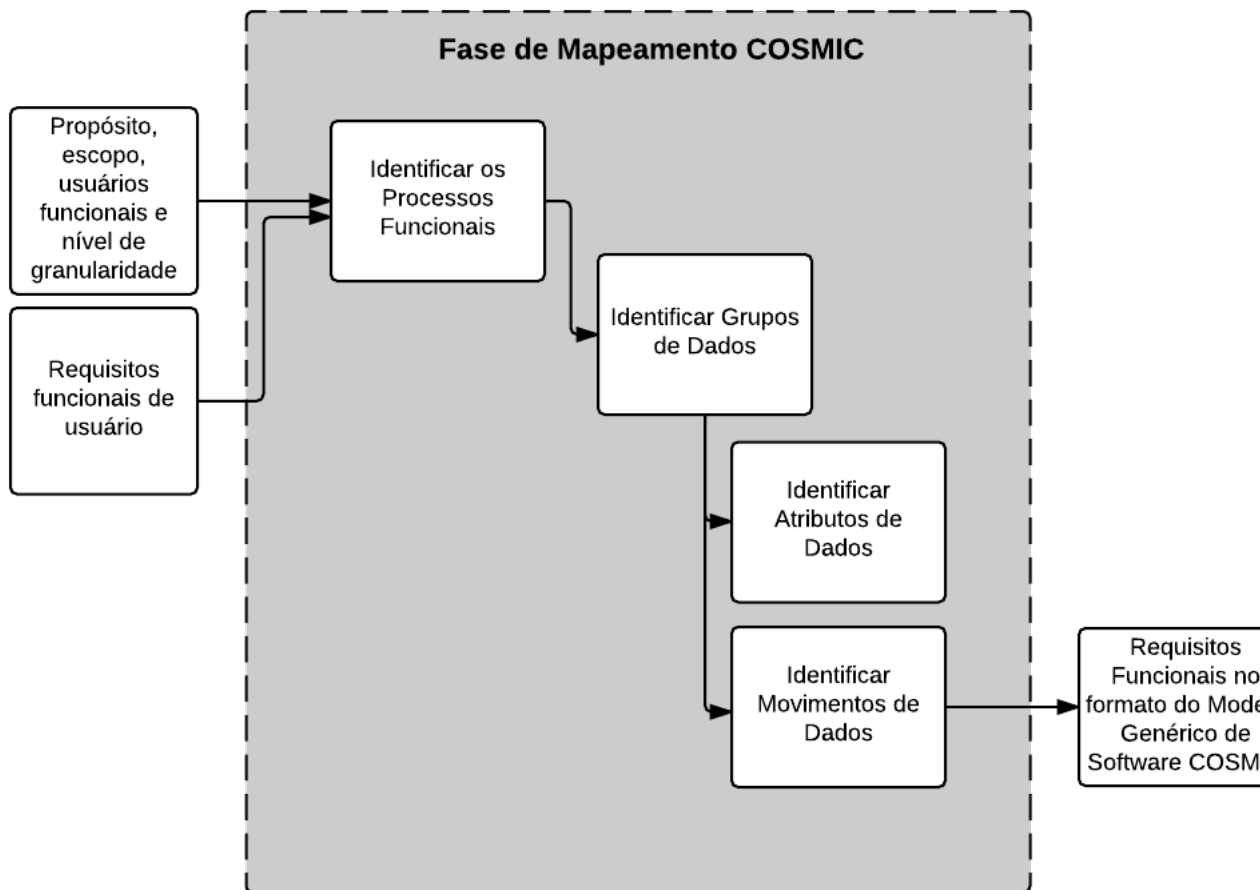


Figura 6: Processo de determinação de Estratégia de Medição

Parte V

Proposta

5 Proposta

5.1 Considerações Iniciais do Capítulo

Neste capítulo será apresentado o processo e os passos a serem seguidos no estudo de caso proposto neste trabalho. Bem como um cronograma de trabalho com as atividades descritas.

5.2 Execução do Estudo de Caso

Baseados nos estudos de [Commeyne, Abran e Djouab \(2016\)](#) e [Ungan, Çizmeli e Demirörs \(2014\)](#) e adaptado do processo de medição da ([SYSTEMS... , 2008](#)), o processo que será seguido para execução do estudo de caso proposto é descrito na Fig. 5.2.

Para que sejam alcançados os objetivos do estudo, as etapas do processo apresentado incluem atividades que serão executadas conforme descrito abaixo:

- **Estabelecer e sustentar compromisso:**

- **Selecionar organização** - Deve ser selecionada uma organização que utilize metodologias ágeis no desenvolvimento dos seus projetos e a organização também deve ser capaz de fornecer os artefatos necessários para execução das estimativas.
- **Selecionar projeto** - Após a seleção da organização, será escolhido qual dos projetos que estão em desenvolvimento dentro dessa organização para ser estudado.

- **Planejar medição:**

- **Selecionar funcionalidades a serem estudadas** - Nessa etapa serão escolhidas as funcionalidades que serão alvo do estudo na sprint. As atividades serão descritas pela equipe de desenvolvimento em forma de

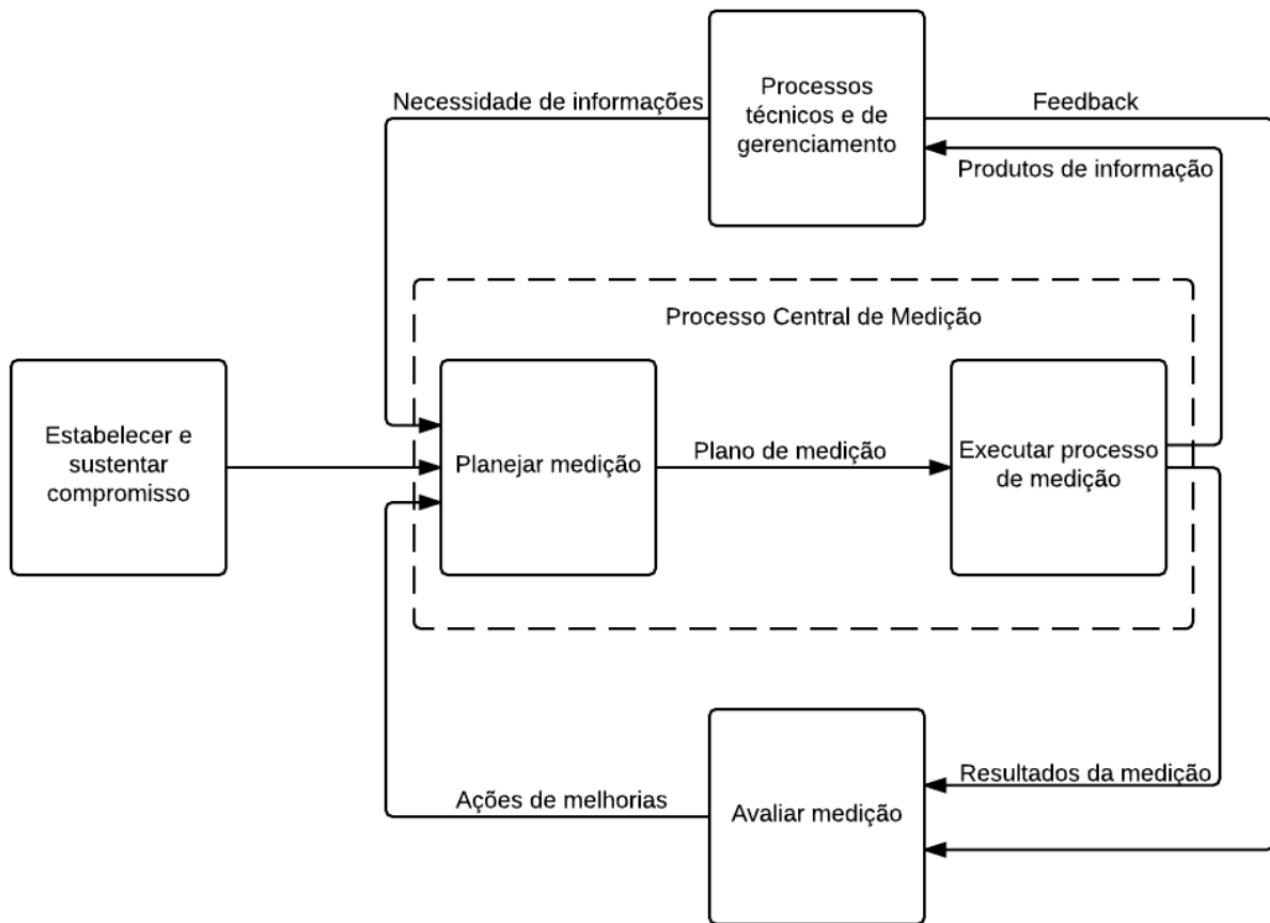


Figura 7: Fluxograma das atividades de medição de software

Histórias de Usuário e farão parte do Backlog do projeto escolhido como alvo do estudo. Serão selecionadas cinco atividades por *Sprint*.

- **Executar processo de medição:**

- **Acompanhar e registrar a pontuação** - As atividades escolhidas, serão pontuadas pela equipe de desenvolvimento, assim como as demais atividades da *Sprint*. A atividade de planejamento (*Planning Poker*) será acompanhada e as pontuações definidas para as atividades seleti-

onadas serão registradas.

- **Realizar a contagem de Pontos de Função Cosmic** - A partir das *User Stories* e de qualquer outra especificação que, porventura, esteja à disposição, serão calculados os Pontos de Função Cosmic apenas para as atividades selecionadas na *Sprint* que estará em progresso no momento.
- **Coletar valores de esforço real** - Após a execução das atividades, o responsável por cada uma deverá manter registrado o esforço, em horas, que foi demandado para implementação da atividade.

- **Avaliar medição**

- **Realizar cálculo de regressão linear com valores estimados e reais** - Será utilizada a técnica estatística de Regressão Linear para analisar a variação do esforço real de cada atividade em comparação aos valores de esforço estimados. A variação entre os esforços é indicada pelo MMRE calculado a partir do esforço real e do tamanho estimado, mais detalhado na seção seguinte.
- **Analisar valores de acurácia Story Points vs. Pontos de Função Cosmic** - Os gráficos com valores obtidos da regressão linear serão analisados a fim de comparar a acurácia das duas técnicas de estimativa de esforço.
- **Gerar modelo de estimativa para o próximo ciclo** - Com os cálculos de regressão linear é possível, ao final do ciclo, obter uma fórmula que pode ser utilizada para estimar o esforço das atividades da próxima *Sprint*. A partir da segunda *Sprint* as atividades selecionadas serão estimadas com base nessa fórmula que será refinada ao fim de cada ciclo do estudo.

- **Processos técnicos e de gerenciamento**

- **Apresentar Resultados** - Os resultados alcançados pelo estudo serão apresentados aos responsáveis dos projetos para que se possa identificar possíveis melhorias no processo de estimativa dentro da organização.

Serão feitos dois estudos de caso em paralelo, ambos seguindo o processo apresentado. Um deles será executado em um ambiente organizacional onde os profissionais já possuem experiência com metodologias ágeis e *Planning Poker*. O outro será baseado em um ambiente educacional em que alunos com pouca ou nenhuma experiência devem projetar um software desde a gerência até o desenvolvimento.

A realização desses dois estudos de caso em paralelo permite observar o quanto a acurácia da estimativa varia quando se compara o *Planning Poker* de uma equipe experiente com uma outra equipe que possui pouca ou nenhuma experiência com metodologias ágeis.

5.2.1 Organização

O estudo de caso proposto será executado no ambiente de um Ministério do Governo Federal, no qual o desenvolvimento e manutenção dos sistemas é terceirizado. O ministério será caracterizado nesse trabalho como Ministério X.

As equipes de desenvolvimento tem em média 10 integrantes e cada equipe é responsável pela desenvolvimento de um sistema.

A empresa responsável pelo desenvolvimento mantém o controle sobre o esforço real das atividades. Tal dado será disponibilizado para a execução do estudo de caso.

A coleta de dados ocorrerá ao final de cada *Sprint*, onde serão registradas todas as atividades da *Sprint*, bem como seus respectivos *Story Points* pontuados pela equipe e o valor de esforço real mantido pela empresa.

5.2.2 Grupo de estudantes

O grupo de estudantes em que será feito o estudo de caso cursam Engenharia de Software na Universidade de Brasília.

O grupo é composto por 10 estudantes onde 5 cursam a matéria de Métodos de Desenvolvimento de Software (MDS) e fazem o papel de desenvolvedores e os

outros 5 cursam a matéria de Gestão de Portfólio e Projeto de Software (GPP) e fazem o papel de gerencia do projeto.

O projeto das matérias de MDS e GPP é composto por duas etapas, onde na primeira os alunos devem desenvolver seguindo a metodologia do *Rational Unified Process* (RUP), e na segunda metade devem seguir os princípios do SCRUM.

O estudo de caso proposto será feito na segunda metade do projeto. Nessa etapa, os alunos de MDS aprendem os princípios do ágil em que, geralmente, é o primeiro contato com o *Planning Poker* e o *Story Point*.

Como parte das disciplinas os alunos devem pontuar as atividades em *Story Points* e manter controle sobre o esforço real de cada atividade. Ao final de cada *Sprint*, o qual ocorre a cada sete dias, tais dados serão coletados para a execução do estudo proposto.

5.3 Resultados Esperados

Com o trabalho proposto espera-se que seja possível realizar uma comparação da acurácia de estimativas de esforço obtidas a partir de *Story Points* e do Ponto de Função COSMIC.

Assim como nos trabalhos de [Commeyne, Abran e Djouab \(2016\)](#) e [Ungan, Çizmeli e Demirörs \(2014\)](#) é esperado que a comparação dessas duas técnicas indique que o Ponto de Função COSMIC apresente um maior grau de acurácia.

5.4 Cronograma de trabalho

O cronograma do trabalho é proposto de acordo com a tabela a seguir.

Tabela 6: Cronograma de Trabalho

Etapas	1º Sem	Jul	Ago	Set	Out	Nov	Dez
Revisão bibliográfica							
Proposta de trabalho							
TCC1 Finalizado e Aprovado							
Atividades em ambiente organizacional							
Seleção do projeto							
Seleção das atividades							
1º Planning Poker							
1ª Estimativa COSMIC							
Regressão Linear 1ª Iteração							
2º Planning Poker							
2ª Estimativa COSMIC							
Regressão Linear 2ª Iteração							
3º Planning Poker							
3ª Estimativa COSMIC							
Regressão Linear 3ª Iteração							
Atividades em ambiente educacional							
Seleção do projeto							
Seleção das atividades							
1º Planning Poker							
1ª Estimativa COSMIC							
Regressão Linear 1ª Iteração							
2º Planning Poker							
2ª Estimativa COSMIC							
Regressão Linear 2ª Iteração							
3º Planning Poker							
3ª Estimativa COSMIC							
Regressão Linear 3ª Iteração							
Atividades do estudo							
Análise dos resultados							
TCC2 finalizado							

Referências

ABOUT Function Point Analysis - IFPUG. Acessado em 17/05/2016. Disponível em: <<http://www.ifpug.org/about-ifpug/about-function-point-analysis/>>. Citado na página 40.

Norm, ACCURACY (trueness and precision) of measurement methods and results-Part 6: Use in practice of accuracy values. [S.l.]: ISO, Geneva, Switzerland, 1994. Citado na página 41.

ALBRECHT, A. J. Measuring application development productivity. *Proc. of the IBM Applications Development Symposium*, 1979. Citado na página 40.

BASILI, V. R. Software modeling and measurement: the goal/question/metric paradigm. 1992. Citado na página 39.

COHN, M. *Agile estimating and planning*. [S.l.]: Pearson Education, 2005. Citado 5 vezes nas páginas 28, 41, 47, 48 e 50.

COMMEYNE, C.; ABRAN, A.; DJOUAB, R. Effort estimation with story points and cosmic function points-an industry case study. In: *2016 21st Software Measurement News*. [S.l.: s.n.], 2016. p. 25-36. Citado 4 vezes nas páginas 28, 31, 61 e 65.

COSMIC Method. Acessado em 17/05/2016. Disponível em: <<http://cosmic-sizing.org/cosmic-fsm/>>. Citado na página 41.

FEHLMANN, T.; SANTILLO, L. From story points to cosmic function points in agile software development—a six sigma perspective. In: *Metrikon-Software Metrik Kongress*. [S.l.: s.n.], 2010. Citado na página 28.

FENTON, N.; BIEMAN, J. *Software metrics: a rigorous and practical approach*. [S.l.]: CRC Press, 2014. Citado 3 vezes nas páginas 39, 40 e 43.

FIDEL, R. The case study method: a case study. *Library and Information Science Research*, v. 6, n. 3, p. 273-288, 1984. Citado na página 30.

GIL, A. C. Como elaborar projetos de pesquisa. *São Paulo*, v. 5, p. 61, 2002. Citado na página 30.

GLOSSARY of Scrum Terms. Acessado em 22/05/2016. Disponível em: <<https://www.scrumalliance.org/community/articles/2007/march/glossary-of-scrum-terms>>. Citado na página 50.

- GOPAL, A. et al. Measurement programs in software development: determinants of success. *Software Engineering, IEEE Transactions on*, IEEE, v. 28, n. 9, p. 863–875, 2002. Citado na página 39.
- GRENNING, J. Planning poker or how to avoid analysis paralysis while release planning. 2002. Citado 3 vezes nas páginas 47, 48 e 50.
- HAMOUDA, A. E. D. Using agile story points as an estimation technique in CMMI organizations. In: *Agile Conference (AGILE), 2014*. [S.l.: s.n.], 2014. p. 16–23. Citado 2 vezes nas páginas 27 e 49.
- MAHNIČ, V.; HOVELJA, T. On using planning poker for estimating user stories. *Journal of Systems and Software*, Elsevier, v. 85, n. 9, p. 2086–2095, 2012. Citado na página 48.
- MORESI, E. et al. Metodologia da pesquisa. *Universidade Católica de Brasília*, 2003. Citado na página 29.
- OMANOVIC, S.; BUZA, E. Importance of stable velocity in agile maintenance. In: IEEE. *Information, Communication and Automation Technologies (ICAT), 2013 XXIV International Symposium on*. [S.l.], 2013. p. 1–8. Citado na página 50.
- PEIXOTO, C. E. L.; AUDY, J. L. N.; PRIKLADNICKI, R. The importance of the use of an estimation process. In: *Proceedings of the 2010 ICSE Workshop on Software Development Governance*. ACM, 2010. (SDG '10), p. 13–17. ISBN 978-1-60558-979-4. Disponível em: <<http://doi.acm.org/10.1145/1808981.1808983>>. Citado na página 27.
- POPLI, R.; CHAUHAN, N. Managing uncertainty of story-points in agile software. In: *2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom)*. [S.l.: s.n.], 2015. p. 1357–1361. Citado na página 27.
- POPLI, R.; CHAUHAN, N. Cost and effort estimation in agile software development. In: *2014 International Conference on Optimization, Reliability, and Information Technology (ICROIT)*. [S.l.: s.n.], 2014. p. 57–61. Citado na página 27.
- REIS, E. *Estatística Descritiva*. [S.l.]: Edições Sílabo, 2008. Citado na página 31.
- RUNESON, P. et al. *Case study research in software engineering: Guidelines and examples*. [S.l.]: John Wiley & Sons, 2012. Citado na página 30.

Norm, SYSTEMS and software engineering — Measurement process. [S.l.]: ISO, Geneva, Switzerland, 2008. Citado na página 61.

TARIQ, S. et al. On learning software effort estimation. In: IEEE. *Computational and Business Intelligence (ISCBI), 2015 3rd International Symposium on*. [S.l.], 2015. p. 79–84. Citado na página 41.

TRAN-CAO, D.; LEVESQUE, G.; ABRAN, A. Measuring software functional size: towards an effective measurement of complexity. In: IEEE. *Software Maintenance, 2002. Proceedings. International Conference on*. [S.l.], 2002. p. 370–376. Citado na página 40.

UNGAN, E.; ÇIZMELI, N.; DEMİRÖRS, O. Comparison of functional size based estimation and story points, based on effort estimation effectiveness in SCRUM projects. In: *2014 40th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA)*. [S.l.: s.n.], 2014. p. 77–80. Citado 4 vezes nas páginas 28, 40, 61 e 65.

USMAN, M. et al. Effort estimation in agile software development: A systematic literature review. In: ACM. *Proceedings of the 10th International Conference on Predictive Models in Software Engineering*. [S.l.], 2014. p. 82–91. Citado 2 vezes nas páginas 17 e 43.

WEN, J. et al. Systematic literature review of machine learning based software development effort estimation models. *Information and Software Technology*, Elsevier, v. 54, n. 1, p. 41–59, 2012. Citado na página 41.