



ISSN: 1984-3151

# SISTEMA DE CLASSIFICAÇÃO DE NÍVEIS DE ASSERTIVIDADE DE UM SPRINT DE PROJETO UTILIZANDO-SE LÓGICA FUZZY

## SYSTEM OF CLASSIFICATION OF LEVELS OF ASSERTIVENESS OF A SPRINT PROJECT USING FUZZY LOGIC

Vinícius Nascimento Malheiros<sup>1</sup>; Moisés Henrique Ramos Pereira<sup>2</sup>

1 Bacharel em Ciência da Computação. Centro Universitário de Belo Horizonte (UNIBH), 2013. Analista de Sistemas, Belo Horizonte, MG. [viccomalheiros@hotmail.com](mailto:viccomalheiros@hotmail.com).

2 Mestre em Modelagem Matemática e Computacional. Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG), 2012. Professor Assistente no Curso de Ciência da Computação do Uni-BH. Belo Horizonte, MG. [moiseshrp@gmail.com](mailto:moiseshrp@gmail.com).

Recebido em: 23/12/2013 - Aprovado em: 30/04/2014 - Disponibilizado em: 31/05/2014

*RESUMO: Este artigo apresenta a modelagem e implementação de um sistema de definição e classificação de níveis de assertividade para um Sprint de projeto utilizando-se a lógica fuzzy, com a finalidade de tornar mais eficiente o gerenciamento do desenvolvimento do software, auxiliando o analista de sistemas do projeto a ter uma tomada de decisão mais assertiva durante o processo do Sprint.*

*PALAVRAS-CHAVE: Inteligência artificial. Lógica fuzzy. Engenharia de software. Sprint. Metodologia ágil, SCRUM.*

*ABSTRACT: This article presents the modeling and implementation of a system definition and classification of levels of assertiveness for Sprint project using fuzzy logic, in order to streamline the management of software development, assisting the systems analyst of the project to have a decision-make more assertive during the Sprint process.*

*KEYWORDS: Artificial Intelligence. Fuzzy Logic. Software Engineering. Sprint. Agile methodology. SCRUM.*

## 1 INTRODUÇÃO

Um processo de engenharia de software é um modelo que traduz a forma que uma organização deve lidar com suas demandas. À medida que as tecnologias evoluem, novas ferramentas são desenvolvidas, tornando o mercado de tecnologia da informação cada vez mais dinâmico. Pode-se citar a disseminação das metodologias ágeis dentro das organizações e a definição de melhores práticas de desenvolvimento de sistemas, forçando, assim, as empresas a revisarem e otimizarem seus processos. Segundo Weber et al. (2004), algumas pesquisas periódicas mostram que é

preciso investir pesado na melhoria dos processos de software no Brasil.

Avaliando o atual contexto do setor de desenvolvimento de software brasileiro percebe-se, cada vez mais, que o gerenciamento do projeto é parte integrante e fundamental do processo de construção do software e que, em alguns casos, este gerenciamento é suportado na experiência dos profissionais e não em metodologias e ferramentas tecnológicas que possam fornecer um maior grau de assertividade na tomada de decisão. A assertividade

na engenharia de software é vista como uma forma de medir a qualidade ao final de um projeto.

Um dos maiores motivos de insucesso de um projeto é a falta de definição clara dos requisitos do sistema que será desenvolvido, seja por falta de entendimento do processo, por falta de comunicação entre os interessados ou pelo próprio planejamento do gerente de projetos que não consegue dar suporte ao time, fazendo com que a lista de necessidades (*backlog*) do usuário chegue para o time de desenvolvimento com certo grau de incerteza e muitas vezes sem priorização, fazendo com que o documento seja muito extenso.

O objetivo principal deste trabalho é utilizar a lógica *fuzzy* (LÓGICA FUZZY, s. d.) para definir níveis de assertividade de um *Sprint de projeto* e então classificá-lo, auxiliando o analista de sistemas do projeto a ter uma tomada de decisão mais eficiente durante o processo do *Sprint*. Entende-se como *Sprint* um conjunto de atividades a serem executadas dentro de um projeto levando em conta um tempo determinado.

Como objetivo específico tem-se a elaboração de um diagnóstico detalhado do *Sprint*, com principais pontos críticos e de melhoria, para que o analista responsável pelo sistema possa ter uma ação preventiva sob o processo.

O controlador *fuzzy* é composto por um processador de entrada, um conjunto de regras linguísticas, um método de inferência *fuzzy* e um processador de saída, gerando um número real como resultado.

A escolha pela lógica *fuzzy* dentre as demais opções de análises multicriteriais é justificada pelo acesso facilitado à base teórica de cada método, assim como a possibilidade de avaliação de cada passo desenvolvido durante as simulações, o que permite a confirmação dos resultados finais e, conseqüentemente, propicia uma elevação do grau de confiabilidade dos mesmos.

A lógica tradicional lida com variáveis que assumem apenas dois estados, falso ou verdadeiro, representados por 0 ou 1, respectivamente. Em boa parte dos casos essa representação é suficiente, porém, existem situações em que valores intermediários a esses representam melhor o raciocínio e a linguagem humana para a solução de problemas. Considerando que esta utiliza termos que expressam aspectos subjetivos na representação da incerteza, Zadeh (1965) propôs uma formalização matemática, criando a lógica difusa (*fuzzy*), que considera graus de verdade variando entre 0 e 1, o que possibilita a avaliação multicriterial desenvolvida neste artigo.

## 2 REVISÃO DE LITERATURA

A Tecnologia da Informação (TI) tem sido considerada um dos componentes mais importantes do ambiente empresarial atual, pois, em muitas organizações, é vista como uma área de apoio que oferece ferramentas que automatizam as atividades e fornece inteligência computacional à área de negócios com grandes oportunidades de gerenciamento do projeto para toda a empresa. Cita-se como exemplo:

1. O desenvolvimento de diferenciais competitivos e novas técnicas (uma vez que a metodologia está sendo estruturada);
2. A adaptação dos trabalhos ao mercado consumidor e ao cliente;
3. O suporte fornecido para o processo com ferramentas eficientes e a inteligência computacional para a área de negócios devido ao uso de sistemas de informação.

A implantação de novas tecnologias em empresas de prestação de serviços é um fenômeno que vem ocorrendo mais intensamente no Brasil desde o final da década de 70, pois se percebeu um aumento da concorrência entre determinados tipos de empresas de serviço, onde o processo de trabalho define a

qualidade do atendimento. Atualmente, é necessário incorporar ao processo de trabalho as tecnologias que possibilitem a modernização da empresa, como forma de prestar um melhor atendimento ao cliente e possibilitar um sistema de informações, controle e gerenciamento que seja capaz de gerar menores custos (GONÇALVES, 1994).

A tecnologia faz parte do dia-a-dia das empresas e cada uma desenvolve a sua de forma específica, por isso o processo de introdução das mesmas exige, na maioria dos casos, uma profunda avaliação da empresa e uma revisão abrangente dos seus processos e mecanismos de controle, a fim de identificar corretamente qual processo se adequa ao cenário apresentado.

A adoção de novas tecnologias muitas vezes leva a grandes alterações nas empresas. Essas mudanças podem se referir à estrutura organizacional, à alocação de recursos ou à distribuição de tarefas entre as pessoas. Podem ocorrer, também, mudanças de comportamento nas pessoas, com o surgimento de resistências e reações negativas (GONÇALVES, 1994, p. 76).

Quando se fala em inovação tecnológica nas empresas, observa-se um aumento significativo no cenário brasileiro. O número aumentou 8,4% em dois anos, passando de 28.036, em 2003, para 30.377, em 2005, de acordo com a Pesquisa Industrial de Inovação Tecnológica 2005, divulgada pelo IBGE (CASTRO, 2007).

Diversas ferramentas podem ser utilizadas no auxílio ao processo de gestão das empresas. Destacam-se, entre elas, os sistemas desenvolvidos com os atributos da lógica *fuzzy*. A lógica tem sido utilizada como um auxílio no gerenciamento de projetos e na tomada de decisão dos gestores que se tornam capazes de modelar indicadores e estratégias organizacionais a partir dos resultados obtidos.

Como exemplo cita-se o uso da lógica na controladoria de uma empresa de turismo, situada no estado do Rio Grande do Sul, atuando na construção

e visualização da priorização dos objetivos e das ações estratégicas. Outro exemplo também é a utilização da lógica para a avaliação do faturamento do consumo de energia elétrica e demanda de potência ativa e reativa em uma empresa de avicultura de postura no estado de São Paulo.

A aplicação da lógica traz resultados positivos para cenários deste tipo, pois, de acordo com Jang e Gulley (1995), dentre outras características, a lógica *fuzzy* é bem flexível, tolerante com dados imprecisos, pode ser construída com base na experiência de especialistas e em muitos casos simplifica ou amplia as possibilidades e recursos dos métodos convencionais de controle.

## 2.1 GERENCIAMENTO DE PROJETOS

Projeto é um empreendimento não repetitivo, caracterizado por uma sequência clara e lógica de eventos, com início, meio e fim, que se destina a atingir um objetivo claro e definido, sendo conduzido por pessoas dentro de parâmetros predefinidos de tempo, custo, recursos envolvidos e qualidade (VARGAS, 2009, p. 5).

Gerenciar um projeto é aplicar técnicas, conhecimento e habilidades para garantir que um projeto tenha sucesso. Abrange atividades de planejamento, execução e controle.

O gerenciamento de projetos tem se tornado fundamental para a execução bem sucedida das estratégias das empresas. O planejamento estratégico, que irá definir a missão e visão da empresa, fornece uma estrutura sólida para o seu desenvolvimento.

Um estudo de *benchmarking* realizado em 2010 no Brasil, pelo PMI - *Project Management Institute* – *Chapters* Brasileiros (BENCHMARKING, 2010) a respeito do gerenciamento de projetos em 460 empresas brasileiras, comprova que a maior parte

delas é do setor de tecnologia da informação (18%) e que cerca de 81%, utilizam softwares de gerenciamento de projetos (P. JÚNIOR, 2010).

A pesquisa mostra também que os três principais problemas que ocorrem com mais frequência nos projetos são: o não cumprimento dos prazos, as mudanças constantes de escopo e problemas de comunicação, fazendo com que seja inevitável o comprometimento do escopo, tempo ou qualidade do projeto. Informações estas, expressas no Anexo A. Por outro lado, o benefício mais importante para a organização é que o gerenciamento estabelece um controle mais efetivo sobre prazos, custos e uso de recursos. Por fim a pesquisa comprova que a dedicação exclusiva dos profissionais é algo fundamental que trouxe grandes e claros benefícios para o sucesso dos projetos.

Com estes dados percebe-se o quanto o gerenciamento é importante para o sucesso de um projeto e para o crescimento saudável da organização,

tendo, a tecnologia da informação, um papel de grande importância para a área, oferecendo suporte para todo este gerenciamento.

### 2.1.1 LÓGICA FUZZY

A lógica *fuzzy* é uma técnica inteligente que permite modelar o modo impreciso e aproximado de raciocínio humano expresso por um conjunto linguístico. Se for possível codificar a tomada de decisão em um conjunto de regras condicionais (se...então), logo um algoritmo pode ser criado e se tornar passível sua implementação em um ambiente computacional.

O conceito *fuzzy* pode ser entendido como uma situação em que não é possível responder simplesmente "sim" ou "não". Mesmo conhecendo as informações necessárias sobre a situação, dizer algo entre "sim" e "não", como "talvez" ou "quase", torna-se mais apropriado, como no exemplo da FIG. 1, onde percebe-se várias categorias para a variável temperatura, Frio, Conforto, Relativamente Quente, Quente.

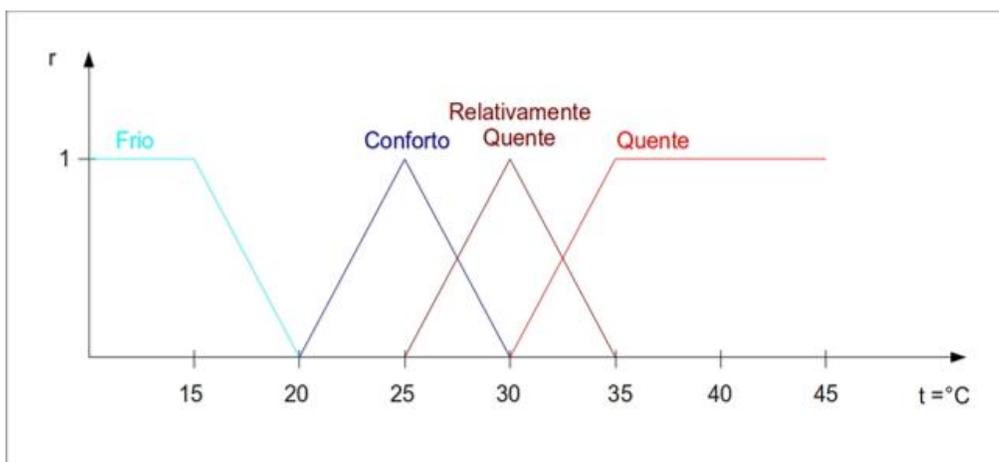


Figura 1 – Exemplo de função de pertinência de variáveis linguísticas.

Fonte – LÓGICA FUZZY, 2010.

A partir dessa informação pôde-se definir graus de pertinência 0,  $\frac{1}{2}$  e 1, diferentemente de 0 ou 1 da lógica booleana.

Trazendo esses valores para o cotidiano, pode-se citar, por exemplo, que a água está quente, ou que a chuva está forte, os termos, quente e forte, são tratados como termos *fuzzy*, pois não se sabe ao certo

o quão quente a água está ou o quão forte a chuva é, assim, enquanto a lógica clássica propõe que esses valores sejam vistos como verdadeiro ou falso, a lógica *fuzzy* propõe que isso seja uma questão de grau.

Com esta abordagem a utilização da lógica *fuzzy* apresenta vantagens como a habilidade em codificar o conhecimento de uma forma próxima da linguagem usada ou como o processo de aquisição do conhecimento que fica mais fácil e menos propenso a falhas e ambiguidades. Conclui-se, então, que a lógica *fuzzy* tornou-se uma tecnologia padrão e é aplicada em análise de dados e sinais de sensores, finanças e negócios dentre outras áreas.

A estrutura de um sistema *fuzzy* pode ser ilustrada de acordo com a FIG. 2.

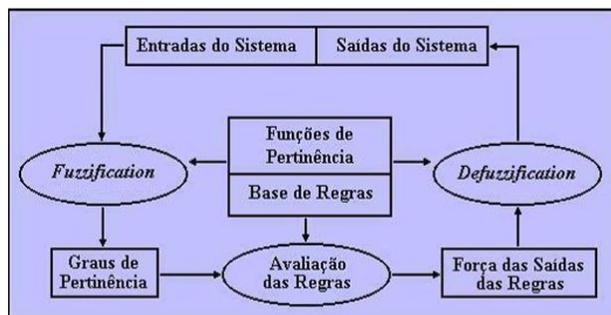


Figura 2 – Arquitetura básica de um sistema difuso.  
Fonte: MIRANDA, 2003.

Um sistema baseado em regras *fuzzy* possui quatro componentes: um processador de entrada (ou fuzzificador que permite uma ligação entre os termos linguísticos e as funções de pertinência), um conjunto de regras linguísticas, um método de inferência *fuzzy* e um processador de saída (ou defuzzificador uma conversão *fuzzy*→escalar que transforma informações qualitativas em quantitativas), gerando um número real como saída.

## 2.1.2 O MANIFESTO ÁGIL

Há alguns anos, profissionais da área de software bem experientes se reuniram em uma estação de esqui, nos Estados Unidos, com o intuito de definir melhorias para aumentar o desempenho de seus projetos. Como cada um já havia trabalhado com processos específicos e já tinha vivenciado experiências em projetos distintos, eles divergiam em algumas opiniões, mas ao final chegaram à conclusão de que algumas práticas eram comuns para se chegar ao sucesso de um projeto.

Com base nestas observações, eles escreveram o Manifesto para o Desenvolvimento Ágil de Software (MANIFESTO ÁGIL, s. d.), composto por quatro valores fundamentais. Valorizaram mais os indivíduos e suas interações, o funcionamento do software, a colaboração dos clientes e a capacidade de resposta a mudanças.

Com base nestes valores, foram criados doze princípios de desenvolvimento ágil, são eles:

1. A maior prioridade é satisfazer o cliente, através da entrega adiantada e contínua de software de valor;
2. Aceitar mudanças de requisitos, mesmo no fim do desenvolvimento. Processos ágeis se adequam a mudanças, para que o cliente possa tirar vantagens competitivas;
3. Entregar software funcionando com frequência, na escala de semanas até meses, com preferência aos períodos mais curtos;
4. Pessoas relacionadas a negócios e desenvolvedores devem trabalhar em conjunto e diariamente, durante todo o curso do projeto;

5. Construir projetos ao redor de indivíduos motivados. Dando a eles o ambiente e suporte necessário, e confiar que farão seu trabalho;
6. O Método mais eficiente e eficaz de transmitir informações para, e por dentro de um time de desenvolvimento, é através de uma conversa cara a cara;
7. Software funcional é a medida primária de progresso;
8. Processos ágeis promovem um ambiente sustentável. Os patrocinadores, desenvolvedores e usuários, devem ser capazes de manter indefinidamente, passos constantes;
9. Contínua atenção a excelência técnica e bom *design* aumenta a agilidade;
10. Simplicidade: a arte de maximizar a quantidade de trabalho que não precisou ser feito;
11. As melhores arquiteturas, requisitos e designs emergem de times auto-organizáveis;
12. Em intervalos regulares, o time reflete em como ficar mais efetivo, então, se ajustam e otimizam seu comportamento de acordo.

### 2.1.2.1 FRAMEWORK SCRUM

*Scrum* é um *framework* para desenvolvimento que consiste na formação de equipes associadas a papéis, eventos, artefatos e regras. Cada componente dentro do *framework* possui um propósito específico e é essencial para o uso e sucesso do *Scrum* (SCHWABER; SUTHERLAND, 2011).

O *framework* possui um fluxo que pode ser representado pela FIG. 3, onde se observa a presença de definição de ideias (que são verdadeiros encontros de *brainstorming* entre os principais interessados no projeto), lista de necessidades do usuário chave (uma lista com o resultado da etapa anterior de definição de ideias priorizadas) e reuniões baseadas em *Sprints* (que são períodos de tempo definidos previamente para a execução de cada conjunto de atividades). Cada reunião possui um objetivo único, estas diferentes reuniões serão apresentadas a seguir.

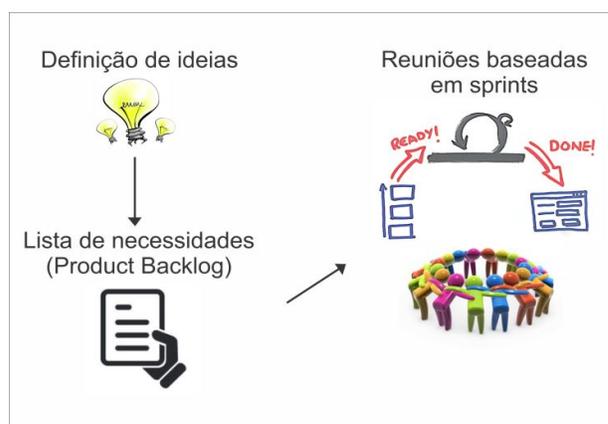


Figura 3 – Fluxo do Scrum.

As reuniões são: *Sprint Planning* (reunião com a presença dos principais interessados no projeto, normalmente são gerentes e os clientes), *Daily Meeting* (reunião diária com o objetivo de disseminar o conhecimento sobre o que foi feito no dia anterior, identificar e priorizar o trabalho do dia que se inicia), *Sprint Review* (reunião para que o time todo apresente o que foi entregue durante o *Sprint*) e *Sprint Retrospective* (ocorre ao final de cada *Sprint* e serve para identificar o que funcionou, o que pode ser melhorado e as ações a serem tomadas).

O *framework* pode ser definido em quatro divisões estratégicas como se pode observar na FIG. 4, primeira parte: Papéis, segunda parte: Eventos, terceira parte: Artefatos e quarta parte: Regras. Cada parte possui subdivisões que auxiliam na estruturação

do *framework*, seja com o envolvimento do time, com reuniões, documentos e definições de tarefas.

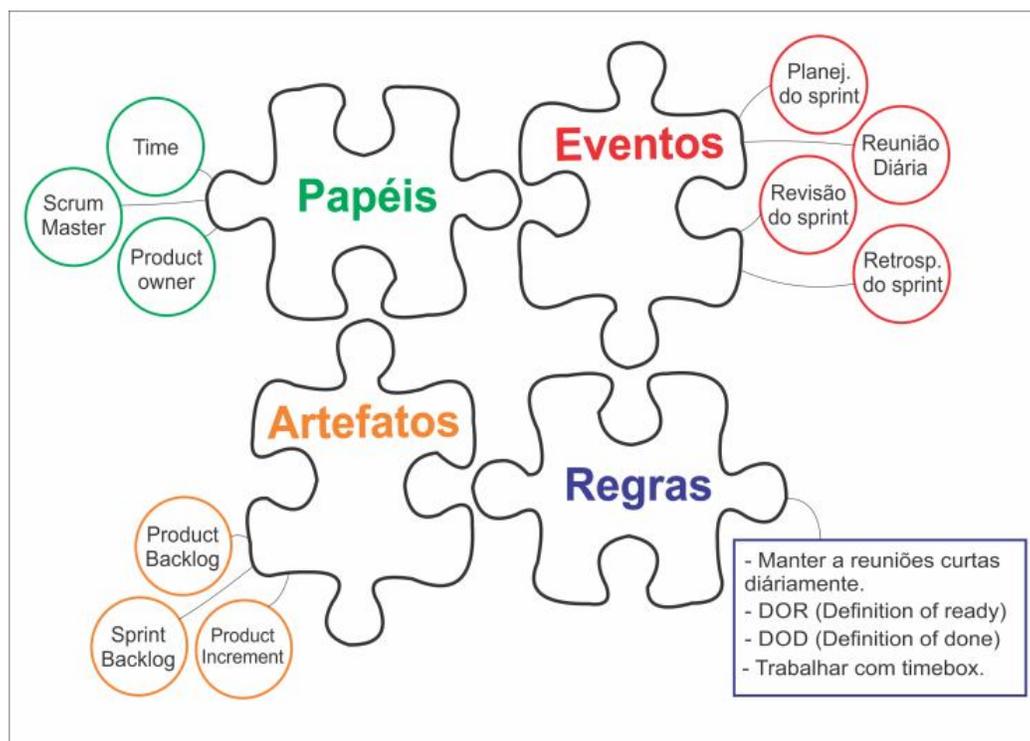


Figura 4 – *Framework Scrum*.

### 2.1.2.2 O SPRINT

No *Scrum*, os projetos são divididos em ciclos mensais chamados de *Sprints* que representam um ciclo de trabalho no *Scrum* e esse ciclo pode ter de 2 a 4 semanas, que é o *Timebox* dos *Sprints*, tendo sempre a mesma duração. A empresa em questão utiliza a seguinte divisão do *Timebox*: na primeira semana é realizada a etapa de análise e desenho, a segunda e a terceira semana correspondem ao período de construção (realizado por uma fábrica de software) e na quarta e última semana é feita a validação.

As funcionalidades a serem implementadas são descritas em uma lista que é chamada de *Product Backlog*. No início de cada *Sprint*, faz-se uma reunião, *Sprint Planning* que tem o objetivo de priorizar os itens do *Product Backlog* (lista de necessidades) e fazer

com que a equipe selecione as atividades a serem implementadas durante o *Sprint* que inicia. As tarefas alocadas em um *Sprint* são transferidas do *Product Backlog* para o *Sprint Backlog*.

Todos os dias, preferencialmente pela manhã, a equipe faz uma reunião chamada *Daily Scrum* com o objetivo de realizar um repasse de informações sobre o que foi feito no dia anterior, identificar impedimentos e priorizar as tarefas do dia.

Ao final de um *Sprint*, a equipe apresenta as funcionalidades implementadas em outra reunião chamada *Sprint Review* (*Sprint* de revisão). Na última reunião, a *Sprint Retrospective*, a equipe inicia o planejamento do próximo *Sprint*, realizando uma retrospectiva de tudo o que foi feito até então. Assim reinicia-se o ciclo iterativo, como se pode observar na FIG. 5 que demonstra a dinâmica de um *Sprint*.

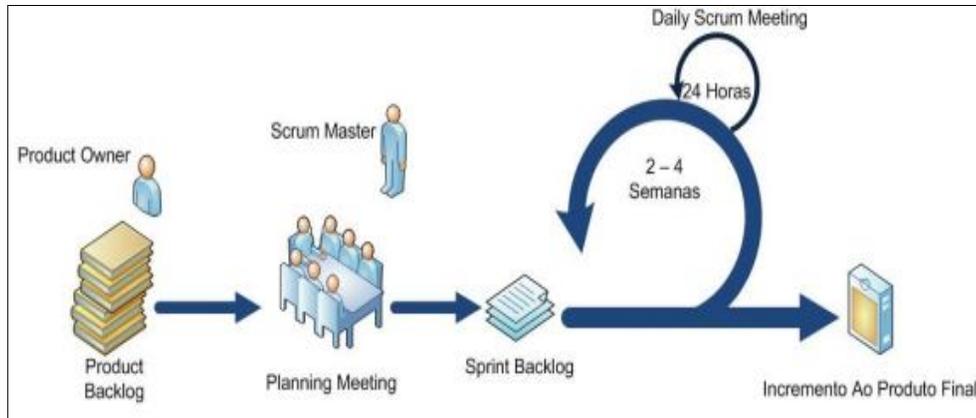


Figura 5 – Dinâmica de um *Sprint*.  
Fonte: PIMENTEL, 2010.

### 3 METODOLOGIA

Para o desenvolvimento da metodologia são apresentados os principais parâmetros que compõem as análises para a classificação dos *Sprints*.

O objetivo é obter uma classificação que auxilie em uma melhor forma de gerenciamento de um *Sprint*, considerando as práticas de desenvolvimento baseadas no *framework Scrum*. Deste modo são analisados os parâmetros capacitação, planejamento, gestão do *backlog* e sustentação. Os parâmetros (critérios) definidos neste artigo são identificados separadamente como qualitativos, sendo classificados da forma mais adequada de acordo com o cenário de desenvolvimento.

Assim, instalando-se um controlador *fuzzy* de acordo com o modelo apresentado, chega-se a um diagnóstico, permitindo à empresa adquirir condições de avaliar o comportamento do *Sprint* e, caso necessário, tomar alguma decisão para uma gestão mais eficiente, evitando que o processo não atinja o resultado esperado.

A metodologia estabelecida na construção do controlador *fuzzy* e a definição dos critérios do *Sprint* constituem grandes contribuições deste trabalho, que apresenta um modelo que pode ser aplicado em qualquer empresa que apresente como metodologia

ágil de desenvolvimento com o *framework Scrum*, utilizando o *Sprint* como definição do período para a entrega do software.

### 4 MODELAGEM E IMPLEMENTAÇÃO DO SISTEMA

#### SGS

O sistema, nomeado de SGS (Sistema de gestão do *Sprint*) foi modelado levando em conta o cenário de trabalho de uma empresa de grande porte. A empresa é do ramo de prestação de serviços e está em constante crescimento.

Para dar suporte a este crescimento, a área de tecnologia da informação (TI) deve estar bem estruturada e sob controle. Já que para alcançar um nível de maturidade em seu processo e ter, como principal visão, a excelência no atendimento ao cliente, torna-se necessário avançar um passo para atender a área usuária com maior agilidade e qualidade.

A empresa em estudo possui um setor de tecnologia da informação muito bem estruturado e tem a área de tecnologia para negócios, responsável pelo desenvolvimento dos sistemas, utilizando o Scrum como *framework* de desenvolvimento, há aproximadamente dois anos. O setor tem conseguido

mais eficiente, sempre com foco na qualidade. Como apoio à metodologia ágil, o processo de produtividade da empresa contempla mais duas filosofias de gestão, o *Lean* e o *Six Sigma*, ambos com foco na redução de desperdício e de defeitos no processo.

O presente trabalho foi estruturado pensando nestas duas filosofias com foco na identificação e eliminação das perdas no processo, identificação e eliminação das atividades de baixo valor agregado, melhoria da qualidade, redução de custo e previsibilidade no desempenho.

No início do processo foi feita uma avaliação, visando o desenvolvimento de sistemas com o intuito de levantar pontos críticos. Com a avaliação feita foi possível elaborar um formulário com perguntas que dizem respeito ao cotidiano dos analistas de sistemas e ao processo de desenvolvimento que utiliza o *Scrum* como *framework*, conforme o Apêndice B – Formulário ciclo de produtividade.

Com o resultado do formulário foi possível definir os parâmetros utilizados como entrada para o processo de fuzzificação. São eles, Capacitação (representa o nível de conhecimento do time sobre os processos, tecnologias e metodologias de trabalho), Planejamento (representa o planejamento dos releases, do *backlog*, recursos, *Sprints* e etc), Gestão do *Backlog* (representa se foi realizada a definição dos requisitos, alinhamento de expectativas com usuário chave) e Sustentação (representa se existe sustentação para o sistema implantado no ambiente de produção).

Foi desenvolvida também uma planilha em *Excel* personalizada de acordo com cada *Sprint* para que os líderes de entrega pudessem preencher as informações pertinentes à capacitação, planejamento, gestão do *backlog* e sustentação do *Sprint*. Nota-se que cada parâmetro precisou ser subdividido em três categorias, como mostra a FIG. 6, para que os resultados fossem mais condizentes com o processo

executado. Nessa planilha, cada parâmetro tem questões a serem ponderadas pelos analistas para que se consiga atribuir um valor mais coerente com a realidade do trabalho.

<b>Capacitação</b> Sem Capacitação Pouco Capacitado Capacitado	<b>Gestão do Backlog</b> Sem Gestão Gestão Parcial Gestão Total
<b>Planejamento</b> Sem Planejamento Parcialmente Planejado Planejado	<b>Sustentação</b> Sem Sustentação Sustentação Parcial Sustentação Total

Figura 6 – Categorias dos parâmetros.

As questões apresentadas no Apêndice C – Planilha de parâmetros – foram avaliadas em pesos em um intervalo de 0 a 5 definidos intuitivamente, onde 0 representa a não aplicação do parâmetro na variável em questão e 5 a aplicação total do parâmetro. Após a atribuição dos pesos foi realizado um cálculo para obter um número real para cada parâmetro, possibilitando assim sua classificação nas categorias determinadas para o *Sprint* e, posteriormente, o enquadramento no intervalo dos conjuntos das funções de pertinência.

Um conjunto *fuzzy* é caracterizado por uma função de pertinência que assume valores dentro do intervalo  $[0,1]$  (ZADEH, 1965). A princípio, qualquer função que associe valores entre zero e um a elementos de um dado conjunto, pode ser tomada como função de pertinência. Entretanto, na escolha de tais funções, deve-se levar em conta o contexto em que serão utilizadas na representação das variáveis linguísticas. Para este estudo foram definidas as funções no formato triangular (*trimf*) e a trapezoidal (*trapmf*), pois, além de serem as mais comumente utilizadas, são as que mais se adequam ao modelo.

Um sistema baseado em regras *fuzzy* foi implementado no software MATLAB®, utilizando o *toolbox* de lógica *fuzzy*, presente no mesmo. Os pesos definidos aos parâmetros podem ser observados no Apêndice C. Os conjuntos *fuzzy* que caracterizam cada parâmetro estão ilustrados no Apêndice D – Conjuntos Fuzzy, apresentado no final deste artigo. As regras desenvolvidas nesta etapa tomam como base proposições SE-ENTÃO, Apêndice E – Regras de inferência. Fazendo uso do software MATLAB®, utilizou-se o Método de *Mamdani* na estruturação do sistema *fuzzy* e o Método do Centro de Gravidade - centroide (COG) para o processo de defuzzificação.

O uso do método *Mamdani* é justificado pelas seguintes características (LIANG; WANG, 1991):

1. Pode ser utilizado em análises de engenharia, pois seus dados de entrada e saída são compostos por valores reais;
2. Incorpora as regras *fuzzy* com praticidade seguindo o raciocínio humano de análise;
3. Possui diversos tipos de fuzzificadores, métodos de inferência e defuzzificadores, o que atribui um maior grau de liberdade ao programador, assim como diversifica os tipos (natureza) de problemas que podem ser analisados;
4. Possibilita uma integração linguística e numérica de forma consistente.

No processo de defuzzificação, para cada valor atribuído às variáveis de entrada, o sistema gera um valor para a variável de saída, tornando possível a classificação do *Sprint* nos níveis de assertividade determinados. Com esta classificação é possível então realizar um diagnóstico do processo, levantando pontos críticos e de melhoria percebidos, fazendo com que o analista possa ter uma ação preventiva sob o processo, aumentando assim o grau de assertividade do *Sprint* assim que o processo rodar novamente.

## 5 RESULTADOS EXPERIMENTAIS

Com o modelo desenvolvido, a etapa subsequente é o teste do sistema com dados reais. O cenário de testes escolhido foi o da própria empresa em questão, foram avaliados 30 *Sprints*. O formulário, já desenvolvido em etapas anteriores, foi aplicado em um grupo de pessoas (líderes de entrega) que têm experiência com a metodologia ágil e que estão diretamente envolvidos com o processo, seja no seu gerenciamento ou participando das salas de produtividade (salas de produtividade são salas de desenvolvimento que envolve os principais interessados no sistema, desde desenvolvedores, analistas de processo, analistas de teste e usuários chave).

Além do formulário foi aplicada também a planilha de parâmetros para que fosse possível realizar a classificação e atribuição dos pesos de cada um deles. Após o preenchimento da planilha foi possível obter o valor dos parâmetros de cada *Sprint* avaliado. Os valores foram inseridos no modelo e assim puderam-se obter os primeiros resultados das classificações em níveis de assertividade.

Dos 30 *Sprints* avaliados, 15 foram classificados como Assertivo, 13 como Médio Assertivo e apenas 2 como Nada Assertivo.

Os parâmetros também foram analisados e o que necessita de mais atenção e uma análise maior é a Gestão do *Backlog* com 13 itens classificados entre Sem Gestão e Gestão Parcial. O segundo parâmetro que merece maior atenção é a Sustentação, com 9 itens classificados entre Sem Sustentação e Sustentação Parcial. Logo após com 4 itens na zona de atenção é o planejamento e por último a Capacitação com apenas 1 item classificado entre Sem Capacitação e Pouco Capacitado.

Para tornar o diagnóstico mais completo, foi feita uma análise mais a fundo no modelo em questão, a planilha de parâmetros que foi preenchida pelos

líderes de entrega foi analisada novamente, sendo considerados como críticos, os itens avaliados com notas 0 e 1 e percebeu-se a seguinte situação: ficaram empatados em primeiro lugar dois parâmetros, Gestão do *Backlog* e Sustentação.

No parâmetro Gestão do *Backlog* o item que teve avaliação mais baixa foi “Definição clara e priorizada dos itens do *Sprint Backlog*”. Já no parâmetro Sustentação destaca-se o item “Definição de duração do período de sustentação”. Com o segundo lugar ficou ainda no parâmetro Sustentação com o seguinte item: “Time para sustentação do sistema/funcionalidade desenvolvido (a)”.

Isto significa que a empresa não está sendo efetiva quando se fala em Gestão do *Backlog* e Sustentação, pois de acordo com os resultados, não existe a definição clara e priorizada dos itens que compõem o *Sprint Backlog* e não se define um período e um líder/time de sustentação para o software em produção, o que está levando os *Sprints* a serem classificados entre Nada Assertivo e Médio Assertivo.

Como plano de ação, para contornar a situação e melhorar o nível de assertividade dos *Sprints* de projeto a empresa tomou as seguintes medidas: Na esfera da Gestão do *Backlog* no item “Definição clara e priorizada dos itens do *Sprint Backlog*”, foi adotada a estratégia de trazer o usuário-chave e os analistas de processos de negócio para mais perto da área de tecnologia da informação, fazendo com que eles vivenciassem o dia-a-dia da área, tomando conhecimento das reais necessidades que envolvem o desenvolvimento do software, uma vez que eles são os principais agentes de priorização e definição dos itens do *Sprint Backlog*.

Já na esfera da Sustentação nos itens “Definição de duração do período de sustentação” e item “Time para sustentação do sistema/funcionalidade desenvolvido (a)”, a empresa criou um cargo de líder de sustentação para cada *Sprint*, assim cada time terá

um analista responsável por definir um período de sustentação e dar suporte ao sistema implantado em produção.

Como solução geral para elevar o nível de assertividade dos *Sprints* de projeto da área como um todo, a empresa adotou a utilização do sistema *Kanban*, que é a aplicação da gestão visual no controle do desenvolvimento do software.

## 6 CONCLUSÃO

A realização do trabalho apresentado permitiu a percepção dos principais pontos de melhoria e dos problemas envolvidos no processo de desenvolvimento de software da empresa estudada, através de uma visão mais abrangente quando comparada à percepção inicial, além de fornecer uma ferramenta tecnológica que auxilia no processo de desenvolvimento do software.

Com os resultados obtidos foi possível elaborar o diagnóstico da área e traçar um plano de ação para que o nível de assertividade dos *Sprints* de projeto seja elevado. Percebe-se também que os pontos identificados como de risco são pontos comuns e que estão entre os mais complexos de serem gerenciados uma vez que os processos dependem do comprometimento da área usuária e de uma mudança de paradigma de desenvolvimento.

Como resultado, o modelo elaborado conseguiu atingir as metas traçadas no início do projeto que eram: obter foco na identificação e eliminação das perdas no processo, identificação e eliminação das atividades de baixo valor agregado, melhoria da qualidade, redução de custo e previsibilidade no desempenho.

Para futuras melhorias em todo o sistema, segue-se como sugestão de trabalho a implementação de novos recursos como o *input* dos dados no modelo através de um menu desenvolvido no MATLAB e a utilização

de redes neurais como alternativa para classificação dos níveis de assertividade.

Como a empresa possui um grupo de discussões sobre melhores práticas de gestão do software, intitulado de SEPG (*Software Engineering Process Group*) e este grupo se divide em KPA's (*Key Process Areas*), que são times responsáveis por identificar atividades que, tomadas em conjunto, permitem alcançar uma série de objetivos importantes para se obter uma melhoria de processo. Sugere-se também como trabalho futuro que os pontos de melhorias identificados sejam repassados para cada KPA, assim, os membros podem discutir as melhores práticas com base nos dados gerados pelo modelo, proporcionando à empresa a criação de novos indicadores e até a adoção de novas metodologias de desenvolvimento de software.

Desse modo, conclui-se que o trabalho desenvolvido nesta pesquisa teve sucesso ao alcançar os objetivos e que a organização estudada deu um grande e positivo passo para melhoria da qualidade dos produtos entregues, tornando o processo mais otimizado e com um maior grau de confiança no gerenciamento dos projetos.

Por outro lado, ficou claro que este trabalho ainda tem o que melhorar para atingir cada vez mais as metas propostas pelo modelo e a própria organização. Assim, é perceptível que o projeto terá continuidade tornando-se necessário um acompanhamento do processo para melhoria contínua, avaliando se os parâmetros e pesos definidos continuam adequados a um ambiente de desenvolvimento se sistemas dinâmico e evolutivo.

## 7 AGRADECIMENTOS

Os autores agradecem ao Coordenador Bráulio Roberto G. Marinho Couto e aos Professores do curso de Ciência da Computação do Centro Universitário de Belo Horizonte - UniBH, pelos ensinamentos, conselhos e pelo conhecimento adquirido. Agradecem, especialmente, aos colegas de classe e de trabalho pela grande amizade construída, apoio e união.

Por fim, não menos importante, os autores agradecem a todas as pessoas da empresa estudada que estiveram envolvidas no processo de construção, desenvolvimento e conclusão do presente trabalho.

---

## REFERÊNCIAS

BENCHMARKING. **Estudo de Benchmarking em Gerenciamento de Projetos Brasil 2010, Project Management Institute – Chapters Brasileiros**. Disponível em: <[http://www.mp.go.gov.br/portalweb/hp/33/docs/benchmarking\\_gp\\_2010\\_geral.pdf](http://www.mp.go.gov.br/portalweb/hp/33/docs/benchmarking_gp_2010_geral.pdf)>. Acesso em: 07 ago. 2013.

CASTRO, F. **Pesquisa retrata inovação tecnológica nas empresas brasileiras**. Disponível em <<http://www.inovacaotecnologica.com.br/noticias/noticia.php?artigo=010175070801>>. Acesso em: 05 ago. 2013.

GONÇALVES, J, E, L. **Os impactos das novas tecnologias nas empresas prestadoras de serviços**. Revista de Administração de Empresas, vol. 34, n. 1. São Paulo: 1994.

JANG, J. S. R.; GULLEY, N., 1995. **Fuzzy Logic Toolbox™ User's Guide**. Disponível em <[http://www.mathworks.com/help/pdf\\_doc/fuzzy/fuzzy.pdf](http://www.mathworks.com/help/pdf_doc/fuzzy/fuzzy.pdf)>. Acesso em: 07 ago. 2013.

LIANG, G., WANG, M. (1991). **A Fuzzy Multi-criteria Decision Method for Facility Selection**. International Journal of Production Research, vol. 29, p. 2313-1330.

LÓGICA FUZZY. Disponível em: [http://www.ft.unicamp.br/liag/wp/monografias/monografias/2010\\_IA\\_FT\\_UNI\\_CAMP\\_logicaFuzzi.pdf](http://www.ft.unicamp.br/liag/wp/monografias/monografias/2010_IA_FT_UNI_CAMP_logicaFuzzi.pdf);

**MANIFESTO ÁGIL**. Disponível em: <<http://manifestoagil.com.br/principios.html>> Acesso em: 06 mai. 2013.

MIRANDA, P. **Sistema de Controle Difuso de Mandani – Aplicações: Pêndulo Invertido e outras**. Monografia (Graduação em Análise de Sistemas). Universidade Federal de Mato Grosso do Sul. 2003.

P. JÚNIOR. **Gestão de projetos em empresas no Brasil Segundo o estudo de benchmarking em gerenciamento de projetos 2010**. Disponível em: <<http://www.slideshare.net/pejota81/gesto-de-projetos-em-empresas-no-brasil?from=embed>>. Acesso em: 07 ago. 2013.

PIMENTEL, Sara Catalão. **Desenvolvimento de uma plataforma de gestão e monitorização de projectos desenvolvidos seguindo a metodologia SCRUM**. Dissertação (Mestrado em Engenharia Informática). Universidade de Lisboa. Portugal. 2010.

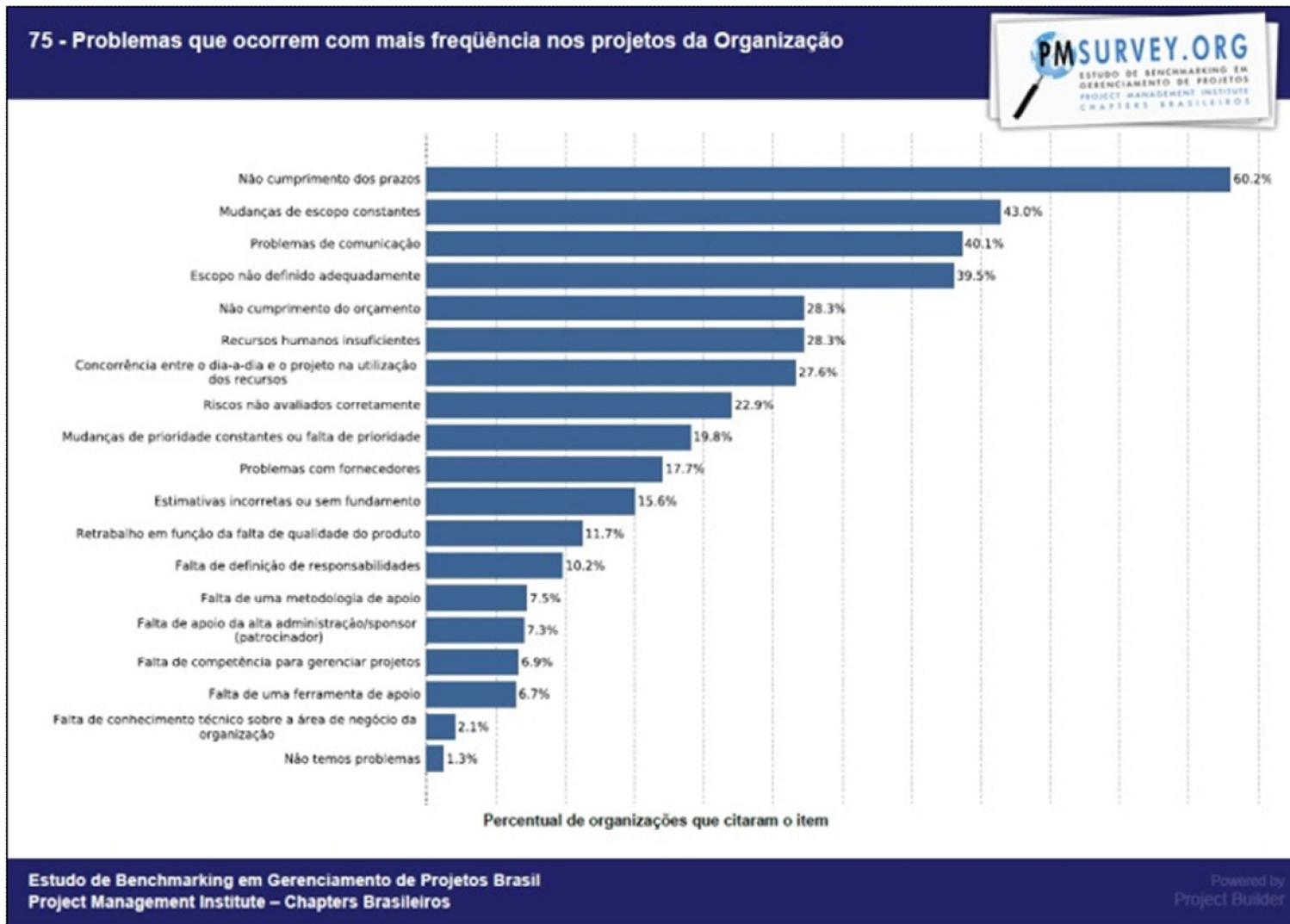
SCHWABER, K.; SUTHERLAND, J. **Guia do Scrum**. 2011. Disponível em <<http://www.scrum.org/Portals/0/Documents/Scrum%20Guides/Scrum%20Guide%20-%20Portuguese%20BR.pdf#zoom=100>>. Acesso em: 10 mar. 2013.

VARGAS, R. **Plano de Projeto**. 4. ed. Rio de Janeiro. Editora Brasport, 229p. ISBN 9788574524306.

WEBER, K. et al. **Modelo de referência para melhoria de processo de software: uma abordagem brasileira**. In: SOLAR, M.; FERNÁNDEZ-BACA, D.; CUADROS-VARGAS, E. (Ed.). 30ma Conferencia Latinoamericana de Informática (CLEI2004), 2004. p. 461–476. ISBN 9972-9876-2-0.

ZADEH, L. A.. **Fuzzy Sets**. Information and Control, vol. 8, n. 3, 1965. pp. 338–353.

## ANEXO A – PROBLEMAS QUE OCORREM COM MAIS FREQUÊNCIA NOS PROJETOS DA ORGANIZAÇÃO.



## APÊNDICE B – FORMULÁRIO CICLO DE PRODUTIVIDADE

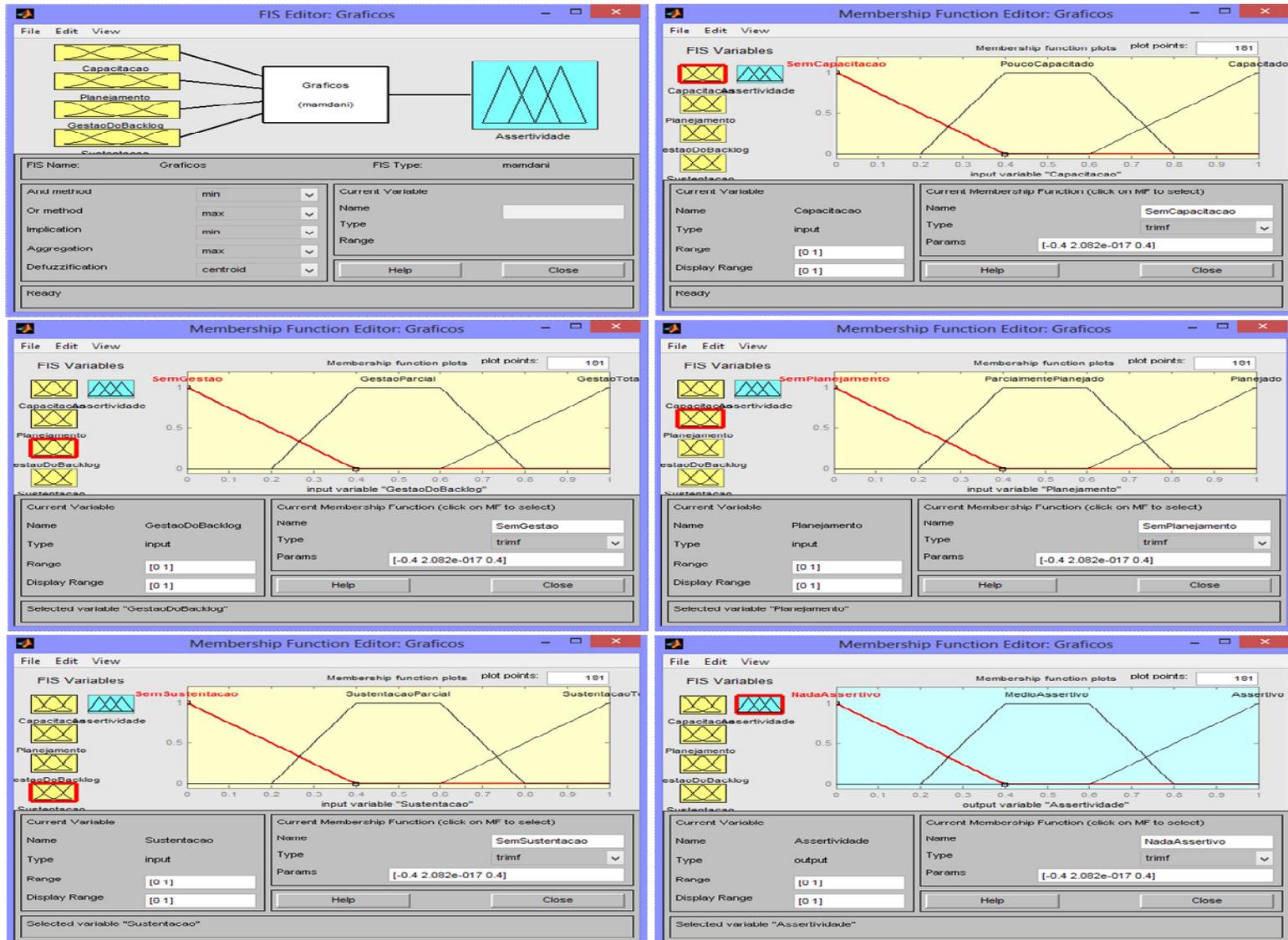
### Formulário Ciclo de Produtividade:

- 1) Qual(is) fator(es) você considera crítico(s) em um Ciclo de Produtividade?
- 2) Qual o principal motivo de insucesso em um Ciclo de Produtividade?
- 3) O que torna um Ciclo de Produtividade complexo?
- 4) Em um Ciclo de Produtividade qual o fator que, se não gerenciado, poderá decorrer em retrabalho?
- 5) Em um Ciclo de Produtividade qual o fator que, se não gerenciado, poderá decorrer em aumento de custo?
- 6) Em um Ciclo de Produtividade qual o fator que, se não gerenciado, poderá decorrer em alteração de escopo?
- 7) Você acredita que o planejamento de liberação de versões para as aplicações existentes seria um fator que contribuiria para o sucesso de um Ciclo de Produtividade?
- 8) Como o conhecimento do time sobre a demanda de negócio influencia no decorrer do Ciclo de Produtividade?
- 9) Em sua opinião, qual seria o risco de insucesso de um Ciclo de Produtividade:  
Escala de 1 a 5
  - Se não existir time para sustentar o software que já está em produção?
  - Se o usuário chave não possuir conhecimento sobre a necessidade solicitada?
  - Se a fábrica de software não possuir conhecimento prévio sobre o sistema?
  - Se a equipe tiver trabalhado em poucos ciclos junto?

### APÊNDICE C – PLANILHA DE PARÂMETROS

Planilha do Sprint						
	Parâmetros	Nota	Peso	Cálculo	Classificação	Range
<b>1</b>	<b>Capacitação (0-5)</b>	-	-	-		
PESO 4	APN, USUÁRIO-CHAVE dominam as regras de negócio necessárias para o desenvolvimento?		16	0		
PESO 2	As tecnologias envolvidas no processo? (entenda tecnologias como sendo softwares de trabalho, linguagens e boas práticas de programação)		8	0	SemCapacitação	0
PESO 3	Sistema Envolvido (Legado ou Novo)		12	0	PoucoCapacitado	0,5
PESO 1	Metodologias de trabalho (Entenda como sendo reuniões de Scrum, definição de sprints e conceitos ágeis)		4	0	Capacitado	1
	<b>TOTAL</b>	<b>0</b>	<b>40</b>	<b>0</b>		
<b>2</b>	<b>Planejamento (0-5)</b>	Nota	Peso	Cálculo		
PESO 1	Planejamento de releases		5	0		
PESO 1	Planejamento dos recursos		5	0	SemPlanejamento	0
PESO 2	Planejamento de testes		10	0	ParcialmentePlanejado	0,5
PESO 4	Planejamento de interação prévias de outras áreas		20	0	Planejado	1
PESO 2	Planejamento dos Sprints		10	0		
	<b>TOTAL</b>	<b>0</b>	<b>50</b>	<b>0</b>		
<b>3</b>	<b>Gestão do Backlog (0-5)</b>	Nota	Peso	Cálculo		
PESO 5	Definição clara e priorizada dos itens do sprint backlog		20	0	SemGestão	0
PESO 1	Definição do prazo do Sprint		4	0	GestãoParcial	0,5
PESO 1	Definição do custo do Sprint		4	0	GestãoTotal	1
PESO 3	Alinhamento de expectativas do usuário chave		12	0		
	<b>TOTAL</b>	<b>0</b>	<b>40</b>	<b>0</b>		
<b>4</b>	<b>Sustentação (0-5)</b>	Nota	Peso	Cálculo		
PESO 8	Time para sustentação do sistema/funcionalidade desenvolvido (a).		24	0	SemSustentação	0
PESO 1	Definição de duração do período de sustentação		3	0	SustentaçãoParcial	0,5
PESO 1	Definição se possui um Sprint muito próximo do que acaba de terminar.		3	0	SustentaçãoTotal	1
	<b>TOTAL</b>	<b>0</b>	<b>30</b>	<b>0</b>		

## APÊNDICE D – CONJUNTOS FUZZY



## APÊNDICE E – REGRAS DE INFERÊNCIA

### Regras de Inferência - Sprint

	Capacitação	Planejamento	Gestão do Backlog	Sustentação	Assertividade
1	If is SemCapacitacao	and SemPlanejamento	and SemGestao	and SemSustentacao	then is NadaAssertivo
2	If is SemCapacitacao	and SemPlanejamento	and SemGestao	and SustentacaoParcial	then is NadaAssertivo
3	If is SemCapacitacao	and SemPlanejamento	and SemGestao	and SustentacaoTotal	then is NadaAssertivo
4	If is SemCapacitacao	and SemPlanejamento	and GestaoParcial	and SemSustentacao	then is NadaAssertivo
5	If is SemCapacitacao	and SemPlanejamento	and GestaoParcial	and SustentacaoParcial	then is NadaAssertivo
6	If is SemCapacitacao	and SemPlanejamento	and GestaoParcial	and SustentacaoTotal	then is NadaAssertivo
7	If is SemCapacitacao	and SemPlanejamento	and GestaoTotal	and SemSustentacao	then is NadaAssertivo
8	If is SemCapacitacao	and SemPlanejamento	and GestaoTotal	and SustentacaoParcial	then is NadaAssertivo
9	If is SemCapacitacao	and SemPlanejamento	and GestaoTotal	and SustentacaoTotal	then is MedioAssertivo
10	If is SemCapacitacao	and ParcialmentePlanejado	and SemGestao	and SemSustentacao	then is NadaAssertivo
11	If is SemCapacitacao	and ParcialmentePlanejado	and SemGestao	and SustentacaoParcial	then is NadaAssertivo
12	If is SemCapacitacao	and ParcialmentePlanejado	and SemGestao	and SustentacaoTotal	then is NadaAssertivo
13	If is SemCapacitacao	and ParcialmentePlanejado	and GestaoParcial	and SemSustentacao	then is NadaAssertivo
14	If is SemCapacitacao	and ParcialmentePlanejado	and GestaoParcial	and SustentacaoParcial	then is MedioAssertivo
15	If is SemCapacitacao	and ParcialmentePlanejado	and GestaoParcial	and SustentacaoTotal	then is MedioAssertivo
16	If is SemCapacitacao	and ParcialmentePlanejado	and GestaoTotal	and SemSustentacao	then is NadaAssertivo
17	If is SemCapacitacao	and ParcialmentePlanejado	and GestaoTotal	and SustentacaoParcial	then is MedioAssertivo
18	If is SemCapacitacao	and ParcialmentePlanejado	and GestaoTotal	and SustentacaoTotal	then is Assertivo
19	If is SemCapacitacao	and Planejado	and SemGestao	and SemSustentacao	then is NadaAssertivo
20	If is SemCapacitacao	and Planejado	and SemGestao	and SustentacaoParcial	then is NadaAssertivo
21	If is SemCapacitacao	and Planejado	and SemGestao	and SustentacaoTotal	then is NadaAssertivo
22	If is SemCapacitacao	and Planejado	and GestaoParcial	and SemSustentacao	then is NadaAssertivo
23	If is SemCapacitacao	and Planejado	and GestaoParcial	and SustentacaoParcial	then is MedioAssertivo
24	If is SemCapacitacao	and Planejado	and GestaoParcial	and SustentacaoTotal	then is Assertivo
25	If is SemCapacitacao	and Planejado	and GestaoTotal	and SemSustentacao	then is Assertivo
26	If is SemCapacitacao	and Planejado	and GestaoTotal	and SustentacaoParcial	then is Assertivo
27	If is SemCapacitacao	and Planejado	and GestaoTotal	and SustentacaoTotal	then is Assertivo
28	If is PoucoCapacitado	and SemPlanejamento	and SemGestao	and SemSustentacao	then is NadaAssertivo
29	If is PoucoCapacitado	and SemPlanejamento	and SemGestao	and SustentacaoParcial	then is NadaAssertivo
30	If is PoucoCapacitado	and SemPlanejamento	and SemGestao	and SustentacaoTotal	then is NadaAssertivo
31	If is PoucoCapacitado	and SemPlanejamento	and GestaoParcial	and SemSustentacao	then is NadaAssertivo
32	If is PoucoCapacitado	and SemPlanejamento	and GestaoParcial	and SustentacaoParcial	then is MedioAssertivo
33	If is PoucoCapacitado	and SemPlanejamento	and GestaoParcial	and SustentacaoTotal	then is MedioAssertivo
34	If is PoucoCapacitado	and SemPlanejamento	and GestaoTotal	and SemSustentacao	then is NadaAssertivo
35	If is PoucoCapacitado	and SemPlanejamento	and GestaoTotal	and SustentacaoParcial	then is MedioAssertivo
36	If is PoucoCapacitado	and SemPlanejamento	and GestaoTotal	and SustentacaoTotal	then is Assertivo
37	If is PoucoCapacitado	and ParcialmentePlanejado	and SemGestao	and SemSustentacao	then is NadaAssertivo
38	If is PoucoCapacitado	and ParcialmentePlanejado	and SemGestao	and SustentacaoParcial	then is MedioAssertivo
39	If is PoucoCapacitado	and ParcialmentePlanejado	and SemGestao	and SustentacaoTotal	then is MedioAssertivo
40	If is PoucoCapacitado	and ParcialmentePlanejado	and GestaoParcial	and SemSustentacao	then is MedioAssertivo
41	If is PoucoCapacitado	and ParcialmentePlanejado	and GestaoParcial	and SustentacaoParcial	then is MedioAssertivo
42	If is PoucoCapacitado	and ParcialmentePlanejado	and GestaoParcial	and SustentacaoTotal	then is MedioAssertivo
43	If is PoucoCapacitado	and ParcialmentePlanejado	and GestaoTotal	and SemSustentacao	then is MedioAssertivo
44	If is PoucoCapacitado	and ParcialmentePlanejado	and GestaoTotal	and SustentacaoParcial	then is MedioAssertivo
45	If is PoucoCapacitado	and ParcialmentePlanejado	and GestaoTotal	and SustentacaoTotal	then is MedioAssertivo
46	If is PoucoCapacitado	and Planejado	and SemGestao	and SemSustentacao	then is NadaAssertivo

47	If is PoucoCapacitado	and Planejado	and SemGestao	and SustentacaoParcial	then is MedioAssertivo
48	If is PoucoCapacitado	and Planejado	and SemGestao	and SustentacaoTotal	then is Assertivo
49	If is PoucoCapacitado	and Planejado	and GestaoParcial	and SemSustentacao	then is MedioAssertivo
50	If is PoucoCapacitado	and Planejado	and GestaoParcial	and SustentacaoParcial	then is MedioAssertivo
51	If is PoucoCapacitado	and Planejado	and GestaoParcial	and SustentacaoTotal	then is MedioAssertivo
52	If is PoucoCapacitado	and Planejado	and GestaoTotal	and SemSustentacao	then is Assertivo
53	If is PoucoCapacitado	and Planejado	and GestaoTotal	and SustentacaoParcial	then is MedioAssertivo
54	If is PoucoCapacitado	and Planejado	and GestaoTotal	and SustentacaoTotal	then is Assertivo
55	If is Capacitado	and SemPlanejamento	and SemGestao	and SemSustentacao	then is NadaAssertivo
56	If is Capacitado	and SemPlanejamento	and SemGestao	and SustentacaoParcial	then is NadaAssertivo
57	If is Capacitado	and SemPlanejamento	and SemGestao	and SustentacaoTotal	then is NadaAssertivo
58	If is Capacitado	and SemPlanejamento	and GestaoParcial	and SemSustentacao	then is NadaAssertivo
59	If is Capacitado	and SemPlanejamento	and GestaoParcial	and SustentacaoParcial	then is MedioAssertivo
60	If is Capacitado	and SemPlanejamento	and GestaoParcial	and SustentacaoTotal	then is Assertivo
61	If is Capacitado	and SemPlanejamento	and GestaoTotal	and SemSustentacao	then is NadaAssertivo
62	If is Capacitado	and SemPlanejamento	and GestaoTotal	and SustentacaoParcial	then is Assertivo
63	If is Capacitado	and SemPlanejamento	and GestaoTotal	and SustentacaoTotal	then is Assertivo
64	If is Capacitado	and ParcialmentePlanejado	and SemGestao	and SemSustentacao	then is NadaAssertivo
65	If is Capacitado	and ParcialmentePlanejado	and SemGestao	and SustentacaoParcial	then is MedioAssertivo
66	If is Capacitado	and ParcialmentePlanejado	and SemGestao	and SustentacaoTotal	then is Assertivo
67	If is Capacitado	and ParcialmentePlanejado	and GestaoParcial	and SemSustentacao	then is MedioAssertivo
68	If is Capacitado	and ParcialmentePlanejado	and GestaoParcial	and SustentacaoParcial	then is MedioAssertivo
69	If is Capacitado	and ParcialmentePlanejado	and GestaoParcial	and SustentacaoTotal	then is MedioAssertivo
70	If is Capacitado	and ParcialmentePlanejado	and GestaoTotal	and SemSustentacao	then is Assertivo
71	If is Capacitado	and ParcialmentePlanejado	and GestaoTotal	and SustentacaoParcial	then is MedioAssertivo
72	If is Capacitado	and ParcialmentePlanejado	and GestaoTotal	and SustentacaoTotal	then is Assertivo
73	If is Capacitado	and Planejado	and SemGestao	and SemSustentacao	then is NadaAssertivo
74	If is Capacitado	and Planejado	and SemGestao	and SustentacaoParcial	then is Assertivo
75	If is Capacitado	and Planejado	and SemGestao	and SustentacaoTotal	then is Assertivo
76	If is Capacitado	and Planejado	and GestaoParcial	and SemSustentacao	then is Assertivo
77	If is Capacitado	and Planejado	and GestaoParcial	and SustentacaoParcial	then is MedioAssertivo
78	If is Capacitado	and Planejado	and GestaoParcial	and SustentacaoTotal	then is Assertivo
79	If is Capacitado	and Planejado	and GestaoTotal	and SemSustentacao	then is Assertivo
80	If is Capacitado	and Planejado	and GestaoTotal	and SustentacaoParcial	then is Assertivo
81	If is Capacitado	and Planejado	and GestaoTotal	and SustentacaoTotal	then is Assertivo