



**DEVAN JERONIMO NACK**

**O PAPEL DO *SCRUM MASTER* NA TRANSFORMAÇÃO DE  
GRUPOS DE PROJETOS FORMAIS (*WATERFALL*), EM  
EQUIPES AUTO GERENCIÁVEIS (*DEVTEAMS* NO ESTÁGIO  
*PERFORMING*)**

Trabalho apresentado ao curso MBA em Desenvolvimento Humano de Gestores, Pós-Graduação *lato sensu*, Nível de Especialização, do Programa FGV Management da Fundação Getúlio Vargas, como pré-requisito para a obtenção do Título de Especialista.

Prof. Edmarson Bacelar Mota

**Coordenador Acadêmico Executivo**

Gianfranco Muncinelli

**Orientador**

**Curitiba – PR**

**2017**

FUNDAÇÃO GETULIO VARGAS

PROGRAMA FGV MANAGEMENT

MBA EM DESENVOLVIMENTO HUMANO DE GESTORES

O Trabalho de Conclusão de Curso

**O papel do Scrum Master na transformação de Grupos de Projetos Formais (*Waterfall*), em Equipes Auto Gerenciáveis (*DevTeams* no estágio *Performing*)**

Elaborado por Devan Jeronimo Nack e aprovado pela Coordenação Acadêmica, foi aceito como pré-requisito para a obtenção do certificado do Curso de Pós-Graduação *lato sensu* MBA em Desenvolvimento Humano de Gestores, Nível de Especialização, do Programa FGV Management.

Data da Aprovação: Local, Data

---

Edmarson Bacelar Mota

Coordenador Acadêmico Executivo

---

Gianfranco Muncinelli

Orientador

## **TERMO DE COMPROMISSO**

O aluno Devan Jeronimo Nack, abaixo assinado, do curso de MBA Executivo em Desenvolvimento Humano de Gestores, Turma DHG4-Curitiba (1/2014) do Programa FGV Management, realizado nas dependências da instituição conveniada ISAE, no período de 03/10/2014 a 12/11/2016, declara que o conteúdo do Trabalho de Conclusão de Curso intitulado “O papel do Scrum Master na transformação de Grupos de Projetos Formais (Waterfall), em Equipes Auto Gerenciáveis (DevTeams no estágio Performing)”, é autêntico e original.

Local, Data

---

Devan Jeronimo Nack

Dedico este trabalho a

Todos os excelentes técnicos de TI que, ao evoluírem na carreira, tornaram-se Gerentes de Projetos e mais recentemente Scrum Masters, sem nenhum preparo relacionado à liderança, motivação, feedback, teoria de grupos, etc. enfim sem nenhum conhecimento sobre Gestão Humana de Pessoas. Esta situação me fez escolher este maravilhoso curso e foi fundamental na escolha do tema deste trabalho.

## Resumo

O objetivo deste trabalho consiste em propor e analisar formas de atuação do *Scrum Master* (SM), no auxílio e condução da evolução de grupos de projetos de desenvolvimento de softwares habituados com metodologias formais (*Waterfall*) em Equipes Auto Gerenciáveis – *DevTeams* no estágio *Performing* – contextualizadas dentro da Empresa “Z” que teve atuação por mais de 10 anos no Desenvolvimento de Softwares para o mercado financeiro na Cidade de Curitiba – PR, através de uma ampla pesquisa bibliográfica relacionada a Teoria de Grupos e formação de Equipes, Conformidade e Conflito, Criatividade e Inovação, Comunicação e Metamodelos (PNL), Inteligência Emocional, Assertividade e Feedback, Motivação, Produtividade, Delegação, *Empowerment* e Liderança Situacional, a fim de disponibilizar uma série de técnicas e ferramentas para o *Scrum Master* identificar o estado atual do *DevTeam* e traçar um plano de ação rumo ao *Performing*, indicando possíveis situações no decorrer do dia-a-dia de um Projeto conduzido sobre o *framework SCRUM* na Empresa “Z”, apresentando pontos de atenção e sugestões para o SM, baseadas nas teorias e nos conceitos contextualizados e embasados na referência bibliográfica. Desta forma espera-se que o SM possa ampliar o seu campo de visão e aplicar conceitos relacionados a aspectos mais humanos e sócio psicológicos, complementando os aspectos técnicos que já são amplamente divulgados e estudados na comunidade de projetos ágeis.

**Palavras Chave:** SCRUM. Scrum Master. Equipes Auto Gerenciáveis. Tuckman. Performing.

## Abstract

The objective of this work is to propose and analyze ways in which the Scrum Master (SM) works, in helping and conducting the evolution of software development projects groups accustomed to formal methodologies (Waterfall), in DevTeams in the Performing Stage - Contextualized within the Company "Z" that has worked for more than 10 years in Software Development for the financial market in the City of Curitiba - PR, through an extensive bibliographic research related to Group Theory and Team Formation, Conformity and Conflict, Creativity and Innovation, Communication and Metamodels (NLP), Emotional Intelligence, Assertiveness and Feedback, Motivation, Productivity, Delegation, Empowerment and Situational Leadership to provide a range of techniques and tools for Scrum Master to identify the current state of DevTeam and to draw up a plan of action towards the Performing, indicating possible situations during the day-to-day of a Project conducted on the SCRUM framework in Company "Z", presenting points of attention and suggestions for the SM, based on theories and concepts contextualized and based on the bibliographic reference. In this way it is expected that the SM can broaden its field of vision and apply concepts related to more human and socio psychological aspects, complementing the technical aspects that are already widely disseminated and studied in the community of agile projects.

**Key Words:** SCRUM. Scrum Master. Self-Managed Teams. Tuckman. Performing.

## AGRADECIMENTOS

Agradeço primeiramente a Deus, pois nada acontece sem a vontade Dele.

Agradeço também a minha esposa Andréa e ao meu filho Rafael, pela enorme paciência demonstrada durante as minhas “ausências” debruçado sobre livros, artigos, sites e qualquer outro material relacionado a este trabalho.

Agradeço também aos meus Gestores Gláucio e Denise, que me indicaram para a bolsa de MBA, a única forma pela qual poderia ter participado deste maravilhoso curso que realmente me transformou em uma pessoa melhor.

Agradeço a todos os Mestres que fizeram parte da história da Turma MBA DHG 1/14 da ISAE/FGV. Eles foram responsáveis por transmitir valiosos conhecimentos de uma forma tão cativante e envolvente que despertou a minha antiga vocação de Professor, quem sabe em um futuro não tão distante.

E por fim agradeço a todos os meus colegas e amigos que fiz durante estes 2 anos de convívio. Eles foram fundamentais para a manutenção de um ambiente agradável, revigorante e extremamente saudável, de tal forma que transformaram o que parecia ser um martírio (aulas sexta à noite, sábado durante o dia inteiro e domingo até ao meio dia) em momentos ansiosamente esperados e transformadores.

## LISTA DE FIGURAS

|  |    |
|--|----|
| FIGURA 1. O BALANÇO DA TI.....                           | 10 |
| FIGURA 2. O MODELO CASCATA.....                          | 13 |
| FIGURA 3. ESTRUTURA MATRICIAL.....                       | 15 |
| FIGURA 4. KANBAN E BURNDOWN (SCRUM BOARD).....           | 21 |
| FIGURA 5. FRAMEWORK SCRUM.....                           | 23 |
| FIGURA 6. FLUXO IDEAL DA COMUNICAÇÃO.....                | 37 |
| FIGURA 7: MODELO DE LIDERANÇA SITUACIONAL.....           | 62 |
| FIGURA 8: A JORNADA DO SCRUM MASTER.....                 | 65 |
| FIGURA 9: CURVA DE DESEMPENHO DE KATZENBACH E SMITH..... | 67 |



## **LISTA DE TABELAS**

|   |    |
|---|----|
| TABELA 1: DIFERENÇAS ENTRE GRUPO DE TRABALHO E EQUIPE .....         | 25 |
| TABELA 2: RESULTADOS OBTIDOS COM EQUIPES AUTO GERENCIADAS.....      | 27 |
| TABELA 3: CINCO ESTÁGIOS DE DESENVOLVIMENTO DE GRUPOS/EQUIPES ..... | 33 |

## SUMÁRIO

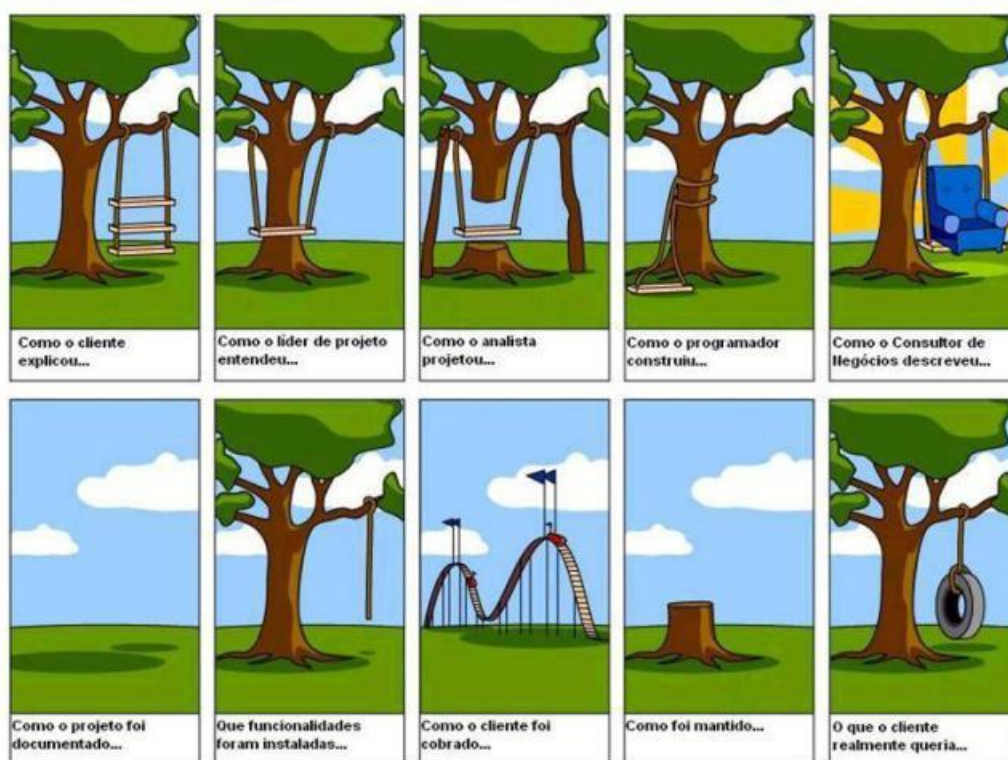
|  |          |
|--|----------|
| <b>RESUMO.....</b>   | <b>5</b> |
| <b>ABSTRACT .....</b>  | <b>6</b> |
| <b>AGRADECIMENTOS.....</b>   | <b>7</b> |
| 1. INTRODUÇÃO .....  | 10       |
| 2. UM PROJETO FORMAL (WATERFALL) DE TI NA EMPRESA “Z” .....                    | 13       |
| 3. UMA NOVA PROPOSTA – <i>SCRUM/AGILE</i> .....                                | 17       |
| 4. EQUIPES AUTO GERENCIÁVEIS: - O QUE SÃO? PARA QUE SERVEM? .....              | 24       |
| 5. A JORNADA DO <i>DEVTEAM</i> RUMO AO <i>PERFORMING</i> E O PAPEL DO SM ..... | 28       |
| 5.1. <i>DevTeams SCRUM</i> na Empresa “Z” .....                                | 28       |
| 5.2. A Teoria de Tuckman .....   | 30       |
| 5.3. Conformidade e Conflito, Criatividade e Inovação .....                    | 34       |
| 5.4. Comunicação e PNL - Metamodelos .....                                     | 36       |
| 5.5. Inteligência Emocional .....  | 42       |
| 5.6. Assertividade e <i>Feedback</i> .....                                     | 46       |
| 5.7. Motivação, Produtividade, Delegação e Empowerment.....                    | 52       |
| 5.8. Liderança Situacional .....   | 57       |
| 6. CONCLUSÕES .....  | 63       |
| 7. POSSÍVEIS DESDOBRAMENTOS .....  | 66       |
| 8. REFERÊNCIAS BIBLIOGRÁFICAS.....   | 68       |

## 1. INTRODUÇÃO

Eficiência e eficácia, duas palavras relativamente similares, com significados bastante distintos. Pode-se exemplificar eficiência como a construção perfeita de um poço artesiano, perfeita nos quesitos técnicos e estruturais, contudo sem resultar no encontro de água. Por outro lado, eficácia, neste exemplo, seria a obtenção de água.

Pode-se usar este simples exemplo para relacioná-lo com uma problemática existente no desenvolvimento de Projetos de Tecnologia da Informação (TI) utilizando a metodologia formal (*Waterfall* ou Cascata). Problema este que por muitas vezes entrega um produto final que não atende as necessidades reais dos usuários e/ou clientes, ou seja, um poço que não tem água. Conforme *Standish Group* (1995), mais de 31% dos Projetos de TI nos Estados Unidos são cancelados mesmo antes de serem concluídos. Esta problemática é tão comum no desenvolvimento de softwares que até já foi refletida em forma de *charge* apresentada na Figura 1, abaixo:

Figura 1. O Balanço da TI



Fonte: anônimo

A Metodologia Formal ou Cascata (*Waterfall*), teve seu início, seu berço, na construção civil, onde os requisitos, ou seja, o produto a ser entregue ao final do Projeto é bem definido e estável. Neste cenário a Metodologia *Waterfall* aplica-se de uma forma bastante satisfatória, guiando-se em sua tríade já amplamente conhecida, Escopo/Tempo/Custo. Entretanto quando fala-se de Projetos que visam o desenvolvimento de sistemas de informação, ou Tecnologia da Informação (TI), por muitas vezes o cenário é bem diferente e um sistema empírico predomina, ou seja, um ambiente altamente complexo e instável.

Como alternativa para a redução dos problemas inerentes a metodologia *Waterfall* de desenvolvimento de softwares, um *framework* (arcabouço/quadro/estrutura) baseado nos pilares de Transparência, Inspeção e Adaptação, chamado de *SCRUM* começa a ser adotado por várias empresas de desenvolvimento de software.

The agile process is now perceived as the universal remedy for software development project failure. One of the biggest reasons is because the agile process makes it easier to do small projects. The agile process also makes it easier to break up larger projects into a series of smaller projects. The secret is the trial and error and delivery of the iterative process. Software should be built in small, iterative steps with small, focused teams. The project team delivers functionality in small bites or steppingstones. A steppingstone, also known as an iteration, is a small but significant deliverable. A steppingstone activity allows for tangible inspection, either visually or hands-on. (STANDISH GROUP, 2013, p.25)

No *SCRUM*, o projeto (produto final) é dividido em pequenos entregáveis que agregam algum valor para o Cliente. Estas pequenas entregas são desenvolvidas em iterações com duração de 2 a 4 semanas, chamadas de *Sprints*. A Equipe de Desenvolvimento (*Development Team* ou apenas *DevTeam*) é responsável pelo resultado de cada iteração. Isto é, o *DevTeam* como um todo é responsável pelo resultado de cada iteração, de cada *Sprint*. Caso algum componente falhe e não atenda as expectativas do cliente, o resultado não é válido, a iteração é considerada insatisfatória, a *Sprint* não atinge o seu objetivo e todo o *DevTeam* falha.

Desta forma, observa-se que o nível de integração e a eficácia do *DevTeam* são essenciais para o sucesso de projetos conduzidos dentro do *framework SCRUM*. Comumente é utilizada a Teoria de Tuckman (TUCKMAN, 1965) como referência para os estágios de desenvolvimento do *DevTeam*, iniciando na fase *Forming* e tendo o estágio *Performing* para Equipes Auto Gerenciadas.

Ainda dentro do *SCRUM*, encontra-se o *Scrum Master (SM)*, que nada mais é do que o agente da mudança, a pessoa que mais conhece de *SCRUM* dentre todos os papéis, ou seja, o líder na questão do processo. Entretanto deve-se ressaltar que o *Scrum Master* não exerce e não possui nenhum papel de gestão/autoridade sobre o *DevTeam*, trata-se apenas de um orientador dos processos *SCRUM*.

Dentro deste contexto, propõem-se analisar formas de atuação do *Scrum Master (SM)*, no auxílio e condução da evolução de grupos de projetos de desenvolvimento de softwares habituados com metodologias formais (*Waterfall*) em Equipes Auto Gerenciáveis – *DevTeams* no estágio *Performing* – contextualizadas dentro de uma Empresa fictícia “Z” que supostamente atua por mais de 10 anos no Desenvolvimento de Softwares para o mercado financeiro na Cidade de Curitiba – PR.

Para tanto, irá se descrever brevemente os aspectos básicos da atuação de grupos de projetos formais (*Waterfall*) de TI da Empresa fictícia “Z”, bem como detalhar, mas não profundamente, as principais características e a dinâmica básica de funcionamento do *framework SCRUM*, com suas cerimônias e seus papéis e responsabilidades. Será dado um destaque especial as Equipes Auto Gerenciáveis (*Performing*), onde irá se apresentar os principais conceitos e características.

E para auxiliar o SM na transformação de grupos de projetos formais em equipes auto gerenciáveis, propõem-se discorrer sobre a Teoria de Grupos e Formação de Equipes, Conformidade e Conflito, Criatividade e Inovação, Comunicação e Metamodelos (PNL), Inteligência Emocional, Assertividade e Feedback, Motivação, Produtividade, Delegação, *Empowerment* e Liderança Situacional, a fim de disponibilizar uma série de técnicas e ferramentas para o *Scrum Master* identificar o estado atual do *DevTeam* e traçar um plano de ação rumo ao *Performing*.

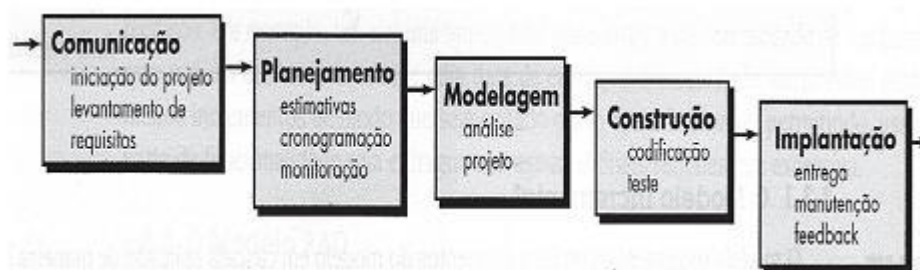
## 2. UM PROJETO FORMAL (WATERFALL) DE TI NA EMPRESA “Z”

Para que se possa ter um melhor entendimento e melhores condições de estabelecer uma análise comparativa entre a metodologia formal de desenvolvimento de Projetos *versus* as vantagens do *framework SCRUM* com Equipes Auto Gerenciáveis, apresentar-se-á neste capítulo as características e o funcionamento básico de projetos de TI da Empresa “Z” desenvolvidos com a metodologia formal (*Waterfall*).

Primeiramente, serão apresentados os conceitos básicos de projetos cascata (*Waterfall*), também conhecidos como Projetos Sequenciais, Tradicionais ou Formais, onde uma fase/etapa somente começa quando a anterior for concluída, conforme verificado na Figura 2.

O modelo cascata, algumas vezes, chamado de ciclo de vida clássico, sugere uma abordagem sequencial e sistemática para o desenvolvimento de software, começando com o levantamento de necessidades por parte do cliente, avançando pelas fases de planejamento, modelagem, construção, emprego e culminando no suporte contínuo do software concluído. (PRESSMAN, 2011, p. 59)

**Figura 2. O modelo Cascata.**



Fonte: PRESSMAN, R.S (2011, p.60)

Como pode-se constatar, trata-se de uma metodologia sequencial, que tem por premissa a conclusão de uma fase para então dar-se início a próxima. Este modelo de gestão de projetos é amplamente conhecido e adotado em várias áreas de atuação, sendo muito bem adaptado para situações onde o ambiente é estável e tem-se um bom nível de conhecimento do produto final completo. Contudo para o desenvolvimento de softwares, tal modelo apresenta sérios problemas e vem recebendo várias críticas. Ainda conforme Pressman (2011), projetos reais raramente seguem um fluxo sequencial e na grande maioria das vezes torna-se praticamente impossível o cliente ter uma visão clara e precisa de 100% dos requisitos, das funcionalidades de um sistema, de um novo software. Este fato agrava-se mais ainda pois o

cliente/usuário somente poderá efetuar os primeiros testes, quando o sistema estiver praticamente totalmente construído. Aumentando demasiadamente os riscos de gastar-se tempo e dinheiro planejando, construindo e testando um sistema que, no final das contas, não vai atender as reais (e atuais) necessidades do cliente.

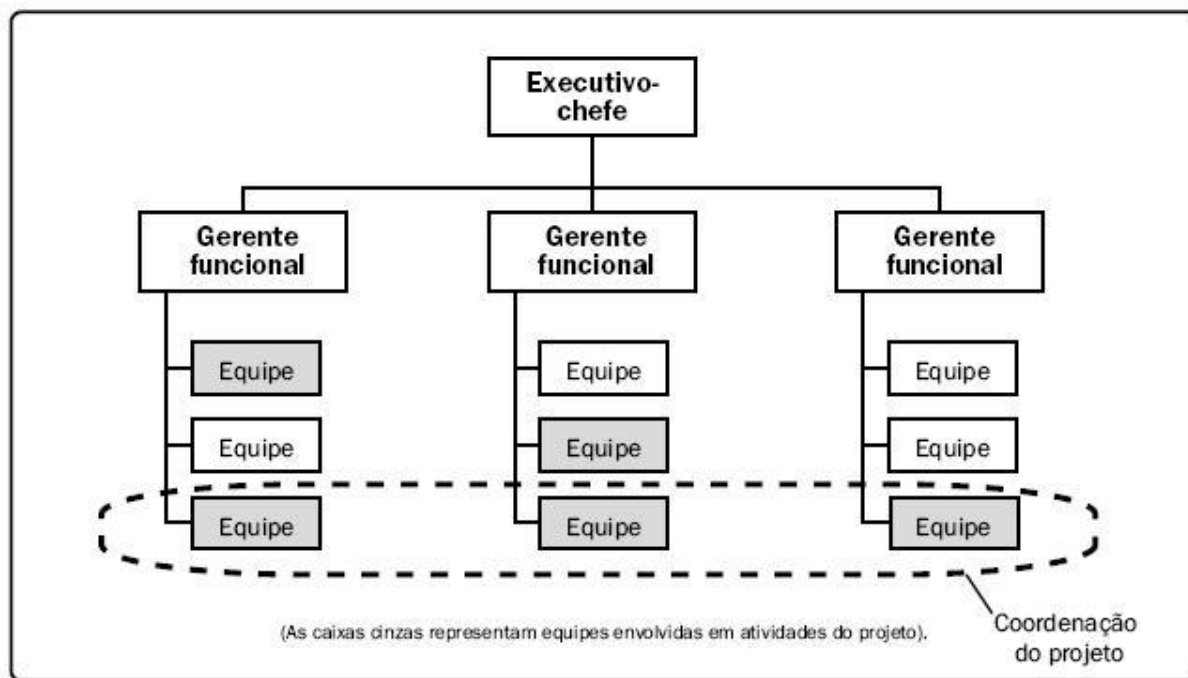
Esta realidade vem sendo aplicada no desenvolvimento de softwares para a Empresa “Z”, onde as áreas de negócio, após a aprovação do projeto pela diretoria, iniciam a árdua e ingrata tarefa de confecção de um documento contendo todos os requisitos, todas as funcionalidades do sistema, contemplando todos os detalhes, todos os fluxos, em resumo, todo o funcionamento deste novo software. Para exemplificar esta situação, pode-se dizer que é esperado que seja entregue um documento contendo todos os detalhes de uma nova casa a ser construída, deste o seu tamanho, quantidade de cômodos, tipo de telhado, até a cor, formato e tipo de tecido da cortina do quarto da empregada, ou secretária do lar. Isso parece ser bastante razoável para o projeto de uma casa, um produto estável e fácil de ver visualizado através de desenhos e maquetes. Agora para um novo software torna-se praticamente inviável, pois as necessidades alteram-se muito rapidamente em um mercado cada vez mais competitivo que vivemos. Novas dificuldades, novas visões surgirão, o mercado não mais será o mesmo, desde a concepção da ideia, até a entrega do produto final, que dependendo da complexidade do projeto, facilmente pode levar mais de 12 meses.

De posse deste documento de requisitos, o departamento/gerência de TI que mais se identifica com o objetivo do projeto, torna-se responsável pela construção do mesmo. E mais uma árdua missão de analisar, entender, validar, estimar e planejar todas as fases do projeto é iniciada. Uma equipe principal (*Core*) e um gerente de projetos deste departamento ou gerência são oficialmente alocados e iniciam os trabalhos, mais especificamente as fases de planejamento e modelagem, onde então são identificadas as equipes de outros departamentos ou gerências que serão necessárias para a implementação do projeto. Neste ponto, o gerente de projetos, com o apoio do gerente do departamento iniciam as conversas e negociações com as demais equipes, a fim de buscar o alinhamento com as expectativas das possíveis datas de entrega desejadas e conseqüentemente a quantidade de colaboradores a serem alocados.

A Empresa “Z” adota uma estrutura organizacional matricial, onde as equipes, ou colaboradores necessários para a implementação do projeto, respondem diretamente aos seus gerentes funcionais e o gerente de projeto atua como um facilitador, como um coordenador,

auxiliando principalmente a comunicação e integração entre os colaboradores das diversas áreas necessárias, conforme apresentado na Figura 3, abaixo.

**Figura 3. Estrutura Matricial**



Fonte: GUIA PMBOK (PMI, 2013)

Uma vez alinhadas todas as expectativas, as equipes avançam nas fases de planejamento, modelagem e construção, cada uma delas responsável por sua parte específica, ou seja, por um componente do projeto, tornando o mesmo um complexo quebra-cabeça. O gerente de projetos além de facilitador da comunicação e integração atua como ponto focal para consolidação da visão geral do projeto no que diz respeito a cronograma, riscos, custos, impedimentos, etc. Reuniões periódicas são agendadas para que todos estejam alinhados na “mesma página” e que efetivamente todos convirjam para o mesmo objetivo, que nada mais é do que a entrega do projeto dentro do custo, do tempo e com a qualidade esperados.

Como anteriormente mencionado, as peças deste quebra-cabeça são montadas apenas no final da fase de construção, na fase de testes, mais especificamente nos testes integrados. É nesta fase que os verdadeiros problemas começam a aparecer, como problemas de comunicação, problemas de entendimento, de interpretação dos requisitos, quanto mais equipes envolvidas, mais potencial para problemas. Neste momento começam as discussões entre as equipes dos vários departamentos envolvidos. O pouco espírito de integração entre os



colaboradores envolvidos é rapidamente eliminado e uma verdadeira “caça às bruxas” é iniciada. Justificativas e frases deste tipo começam a ser ouvidas:

- Eu fiz a minha parte conforme solicitado!
- A culpa não é minha! O que me foi pedido esta entregue!
- O problema não é meu, o gerente de projeto que resolva isso agora.

Para alguns poucos gerentes de projetos experientes estes problemas são esperados e de certa forma até minimizados, mas podemos dizer que são muito comuns e impactantes na maioria dos projetos da Empresa “Z”.

Após muitas reuniões técnicas, muitos ajustes, muitas correções, alguns atrasos e alguns milhares de reais a mais, finalmente os testes integrados são concluídos e estamos prontos para iniciar os testes de aceitação do usuário. Na grande maioria das vezes está é a primeira vez que o usuário/cliente tem o primeiro contato com o software que foi amplamente detalhado no documento de requisitos. Não são raras as vezes que escutamos frases como:

- Não foi isso que eu pedi!
- Não é isso que está detalhado no documento de requisitos!
- OK, legal, mas quando vocês vão apresentar aquilo que eu pedi?!
- Isso não resolve o meu problema!

Então, uma nova e estressante fase de revisão, negociação, alinhamento e ajustes é iniciada, tentando aproveitar ao máximo aquilo que foi feito, pois o orçamento e principalmente o tempo já foram totalmente consumidos.

### 3. UMA NOVA PROPOSTA – *SCRUM/AGILE*

Como alternativa a metodologia de desenvolvimento de sistemas formal (*Waterfall*), Jeff Sutherland e Ken Schwaber em 1993, (SUTHERLAND, 2014) criaram uma nova forma, um *framework* para desenvolver e manter produtos complexos.

SCRUM(subs): Um framework dentro do qual pessoas podem tratar e resolver problemas complexos e adaptativos, enquanto produtiva e criativamente entregam produtos com mais alto valor possível. SCRUM é leve, simples de entender e extremamente difícil de dominar (SCHWABER; SUTHERLAND, 2011, p. 3)

*SCRUM* não é uma metodologia completa de desenvolvimento de projetos, na verdade é um conjunto de pessoas que, guiados por papéis, responsabilidades, eventos (ou cerimônias), artefatos e regras, resolvem e desenvolvem soluções para sistemas empíricos e complexos. O *SCRUM* não tem todas as respostas para todos os problemas que as pessoas encontrarão durante o desenvolvimento de qualquer projeto, mas ele prega que seus praticantes possam efetuar pequenas pausas para avaliar o que foi feito, como foi feito, se estão no caminho certo e principalmente o que podem fazer para tornarem-se mais eficazes, enfrentando, e eliminando, todo e qualquer barreira ou impedimento que esteja atrapalhando o progresso do projeto.

Conforme exposto por seus criadores (SCHWBER; SUTHERLAND, 2011), o *SCRUM* foi fundamentado nas teorias empíricas de controle de processo – empirismo<sup>1</sup> – ou de uma forma mais prática, fundamenta-se em atuar naquilo que é conhecido, na experiência, na experimentação, aumentando a previsibilidade e o controle dos riscos. O *SCRUM* apoia-se em 3 pilares básicos: transparência, inspeção e adaptação.

Transparência onde todos os envolvidos no projeto, e até os fora dele, possam facilmente entender o que está sendo construído, avaliar o progresso do trabalho, compartilhar dificuldades e manter contato constante com o usuário/cliente, denominado *Product Owner* (PO). Inspeção e adaptação para que, de tempos em tempos, possam efetuar uma breve pausa, analisar o que foi construído, quais foram as dificuldades, quais foram os impedimentos, buscar os *feedbacks* dos usuários/clientes, representados pelo PO e principalmente buscar

---

<sup>1</sup> Empirismo: fil doutrina segundo a qual todo conhecimento provém unicamente da experiência, limitando-se ao que pode ser captado do mundo externo, pelos sentidos, ou do mundo subjetivo, pela introspecção, sendo ger. descartadas as verdades reveladas e transcendentais do misticismo, ou apriorísticas e inatas do racionalismo.

responder um questionamento que deve ser permanente no *SCRUM*, que é o de como podemos melhorar a eficiência e eficácia das entregas, buscando sempre o aprimoramento constante, a “hiperprodutividade”.

No *framework SCRUM* tem-se definidos papéis bastante distintos, cada um deles com um conjunto específico de responsabilidades, são eles:

- ***Product Owner* ou apenas *PO*:**

É o dono do produto, responsável pelo direcionamento dos esforços do time de desenvolvimento (*DevTeam*) para aquelas funcionalidades do produto que mais agregam valor ao usuário final, em outras palavras, responsável pela priorização dos componentes a serem desenvolvidos pelo *DevTeam*. É a pessoa que efetivamente vai utilizar o produto, neste caso um software, ou possui profundos conhecimentos e mantém contato direto com os usuários finais. Trata-se de um dos papéis mais importantes do *SCRUM*, uma vez que é o responsável por materializar a visão, o objetivo do projeto, efetivamente o que tem que ser construído, em pequenos componentes, pequenas especificações, comumente chamadas de *User Stories*, priorizá-las por ordem de importância, de valor agregado, esclarecendo toda e qualquer dúvida do *DevTeam* relacionada ao produto a ser entregue. Ao conjunto de *User Stories* damos o nome de *Product Backlog*.

- ***Development Team* ou apenas *DevTeam*:**

Trata-se de time de desenvolvimento (*DevTeam*), que ao contrário do que comumente acontece em projetos formais (*Waterfall*) em estruturas matriciais, no *SCRUM* o *DevTeam* deve ser uma única unidade, um único grupo de pessoas, multidisciplinar e multifuncionais, ou seja, com todas as competências/condições de desenvolver o projeto sem depender de outras áreas e/ou equipes. Por definição este grupo de pessoas deve ser auto organizável, conforme Schwaber e Sutherland (2011), ninguém nem mesmo o *Scrum Master* diz ao *DevTeam* como transformar o *Product Backlog* em incrementos de funcionalidades potencialmente utilizáveis. Não existem títulos, papéis e ou responsabilidades específicas dentro do *DevTeam*, obviamente cada integrante contribui com a sua especialidade, mas todos são considerados Desenvolvedores e todo o *DevTeam*, como uma única equipe, é responsável pelo desenvolvimento do projeto

como um todo. A recomendação diz que o *DevTeam* não deve ter menos do que 3 integrantes e não mais do que 9.

- ***Scrum Master* ou apenas SM:**

É o agente de mudança, responsável por garantir que as teorias, práticas e regras do *SCRUM* sejam seguidas e aplicadas tanto pelo PO quanto pelo *DevTeam*. Também atua no esclarecimento, para aqueles que estão de fora do Projeto, das dinâmicas, conceitos e formas de relacionamento com o PO e o *DevTeam*. Trata-se da pessoa que mais conhece e entende de *SCRUM* no grupo de pessoas alocadas no Projeto. Também atua como um facilitador e removedor de barreiras, ou impedimentos, que estejam atrapalhando o avanço do Time *SCRUM* (PO e *DevTeam*), sempre buscando maximizar o valor criado pelos mesmos. Conforme Schwaber e Sutherland (2011), o *Scrum Master* é um servo-líder para o Time *SCRUM*.

Outra grande e importante característica do *SCRUM* é o seu conceito de iterações e desenvolvimento incremental. Essas iterações são chamadas de *Sprints* e tem um tempo pré-determinado (*Time-Boxed*), geralmente compreendido entre 2 a 4 semanas. A cada *Sprint* o *DevTeam* seleciona os requisitos – *User Stories*, detalhadas e priorizadas pelo PO (*Product Owner*) – que entendem serem capazes de desenvolver completamente durante a *Sprint*. Este desenvolvimento deve ser considerado como potencialmente utilizável pelos usuários e/ou PO, ou seja, devem atender a algum critério/conceito de “Pronto” (*Definition of done*), que foi previamente acordado por todos os envolvidos (PO e *DevTeam*).

Cada *Sprint* inicia-se com uma reunião chamada de *Sprint Planning*, que também tem um limite de tempo definido, por exemplo, para *Sprints* de 4 semanas a *Sprint Planning* deve ter no máximo 8 horas de duração. Proporcionalmente este tempo deve ser adequado conforme a duração da *Sprint*, ou seja, para *Sprints* de 2 semanas, o planejamento da *Sprint* não deve ser superior a 4 horas. A fonte de entrada (*input*) é o *Product Backlog* detalhado e priorizado pelo PO. O *DevTeam* e somente ele de forma colaborativa, com o total suporte do PO para esclarecimento de dúvidas sobre o *Product Backlog*, deve atingir o objetivo essencial da *Sprint Planning*, que é o de responder as duas questões abaixo:

- O que o *DevTeam* tem condições de entregar ao final da *Sprint*? Como resultado do desenvolvimento incremental que pode ser utilizado pelo PO e/ou usuários (*Definition of Done*). Este “O que?” torna-se o objetivo, a meta da *Sprint*.

- Como o objetivo *Sprint* será entregue, ou seja, a relação de atividades a serem desenvolvidas pelo *DevTeam*, em outras palavras – Como o trabalho será realizado?

Uma vez estabelecido o objetivo da *Sprint* e detalhadas as atividades a serem realizadas, o *DevTeam* inicia os trabalhos. Como um time auto gerenciável, ninguém define o que cada um irá fazer, os integrantes desta equipe multidisciplinar assumiram a responsabilidade, o compromisso, de ao final da *Sprint* entregarem aquilo que foi definido, por eles mesmos, como objetivo da *Sprint*, sempre se lembrando de atender os critérios de “Pronto” (*Definition of Done*).

Para compartilhar o progresso dos trabalhos e manter todos os integrantes do *DevTeam* cientes do que cada um está desenvolvendo, quais os impedimentos, quais as dificuldades, etc., enfim, para manter a comunicação aberta e fluída, bem como promover a constante inspeção e adaptação, o *SCRUM* prega que seja realizada uma reunião diária entre o *DevTeam*, com o apoio do SM. Esta reunião deve ser de no máximo 15 minutos de duração, geralmente com todos os participantes em pé, onde cada um compartilha com os demais integrantes os avanços, os progressos, os impedimentos. Como guia para esta reunião diária, chamada de *Daily Stand Up Meeting*, cada integrante do *DevTeam* procura responder 3 perguntas:

- O que eu fiz ontem para nos ajudar a atingir a meta da *Sprint*?
- O que eu farei hoje para nos ajudar a atingir a meta da *Sprint*?
- Existe algum obstáculo (impedimento) que nos impeça de atingir a meta da *Sprint*?

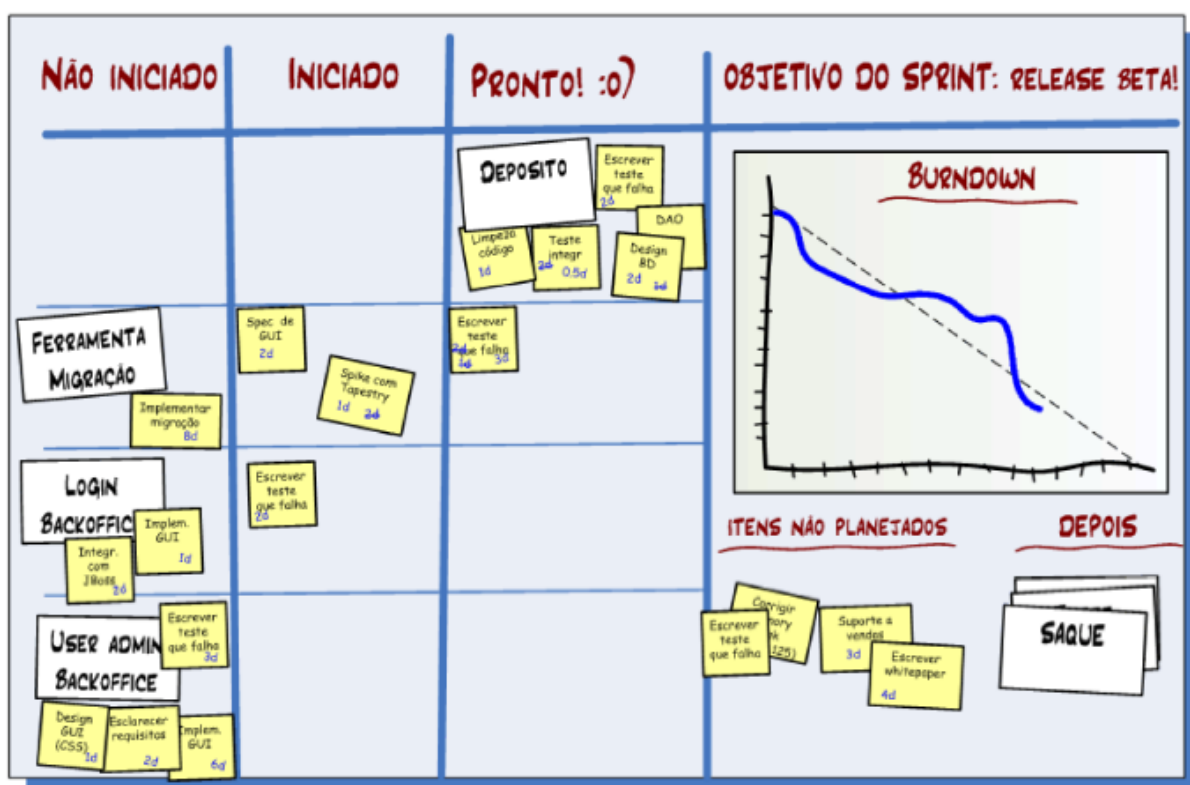
A fim de auxiliar o Time *SCRUM* (PO, *DevTeam* e o SM) no planejamento da *Sprint*, no detalhamento das atividades, durante as reuniões diárias (*Daily Stand Up Meeting*) e no acompanhamento do progresso dos trabalhos, comumente é utilizado um quadro *Kanban*<sup>2</sup> e um gráfico de *Burn down*<sup>3</sup>, também conhecido como *SCRUM Board*, conforme apresenta-se na Figura 4.

---

<sup>2</sup> Kanban é um termo de origem japonesa e significa literalmente “cartão” ou “sinalização”. Este é um conceito relacionado com a utilização de cartões (post-it e outros) para indicar o andamento dos fluxos de produção em empresas de fabricação em série.

<sup>3</sup> *Burn Down* é uma representação gráfica do trabalho a ser feito pelo tempo. O trabalho pendente (ou backlog) é muitas vezes sobre o eixo vertical, com o tempo ao longo da horizontal.

Figura 4. Kanban e Burndown (SCRUM Board)



Fonte: KNIBERG, Henrik (2015, p.58)

A *Sprint* encerra-se com dois eventos de igual, senão maior, importância do que os já descritos até agora. Um deles é chamado de *Sprint Review* (ou *Sprint Demo*), trata-se de uma cerimônia dedicada para que o *DevTeam* apresente para o PO, usuários, convidados e/ou qualquer parte interessada (*Stakeholders*<sup>4</sup>) o resultado da *Sprint*, isto é o objetivo, a meta, contendo os componentes potencialmente utilizáveis do produto/software, sempre atendendo aos critérios de “Pronto” (*Definition of Done*). Trata-se de uma cerimônia especial para a

<sup>4</sup> Parte interessada / Stakeholder. Um indivíduo, grupo ou organização que possa afetar, ser afetado, ou sentir-se afetado por uma decisão, atividade, ou resultado de um projeto.

coleta de *feedbacks*, para verificar se o *DevTeam* esta no caminho certo. Atendente novamente os pilares de transparência, inspeção e adaptação constantes.

A grande vantagem deste tipo de desenvolvimento incremental, com um período estabelecido de tempo constante de entregas, consiste em uma das principais vantagens competitivas do *SCRUM* perante as outras formas de desenvolvimento de Projetos.

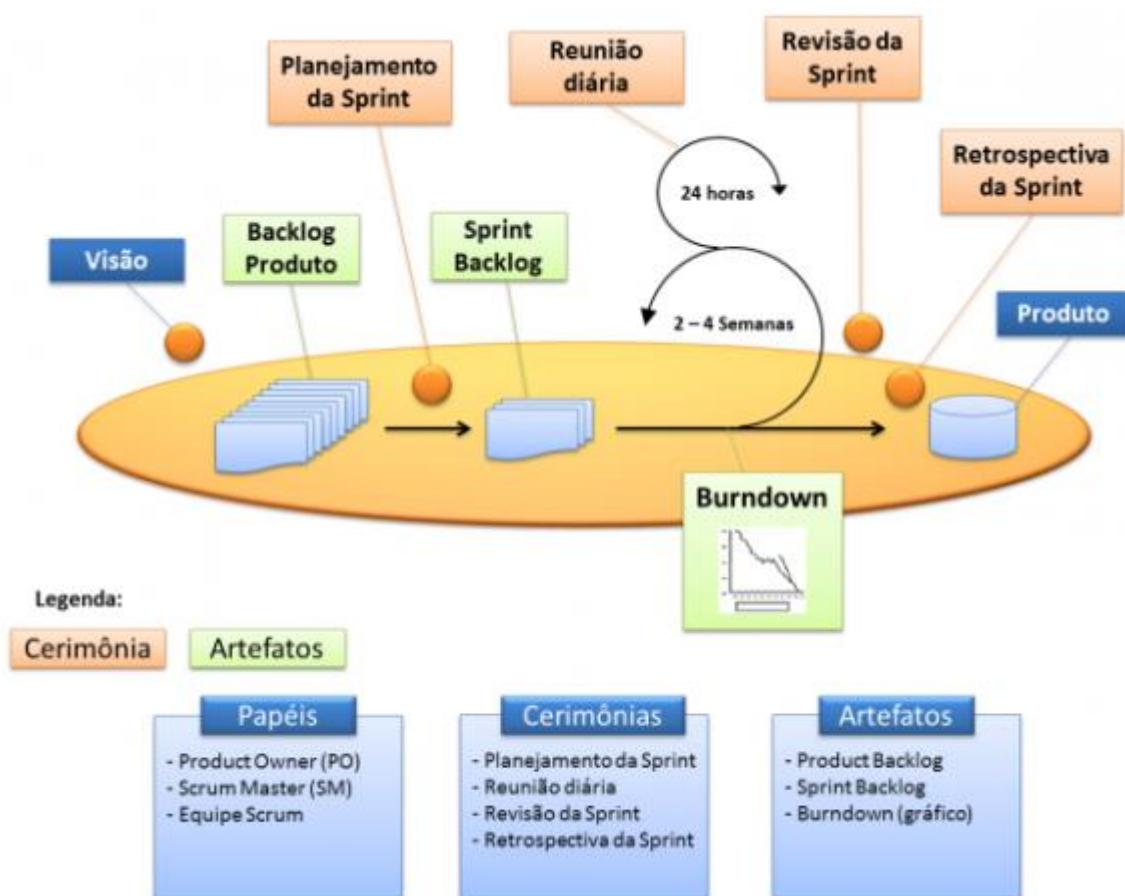
Pense nos diversos projetos que você fez. Aposto que raramente você recebe um feedback até que eles estejam concluídos – e isso pode levar meses, até anos. Você pode estar seguindo na direção completamente errada durante muito tempo, sem suspeitar de nada, e isso vai acabar subtraindo uma grande parte da sua vida. Nos negócios, isso pode significar a diferença entre o sucesso e o fracasso. Eu já vi esse tipo de situação acontecer o tempo todo: uma empresa passa anos em um projeto que parecia uma ótima ideia quando os funcionários começaram, mas, quando cruzam a linha de chegada, o mercado já sofreu mudanças fundamentais. Quanto mais cedo você entregar o produto para os clientes, mais rápido eles serão capazes de dizer se você esta fazendo algo de que precisam. (SUTHERLAND, 2014, p.77)

Por fim, tem-se a *Sprint Retrospective*, uma cerimônia onde todo o time *SCRUM* se reúne para avaliar, o que foi feito, como foi feito, o que deu certo, o que deu errado, o que pode ser melhorado e criar um plano de melhorias a serem aplicadas na próxima *Sprint*, isto é, nada mais nada menos do que mais uma cerimônia que visa à busca constante da inspeção e adaptação. Todos são encorajados a participar e expor seus pontos de vistas, seus sentimentos, com relação aos processos, aos relacionamentos e às ferramentas e todos, principalmente o *DevTeam*, comprometem-se em estabelecer um plano de ação para a implementação das melhorias.

Todas estas cerimônias do *SCRUM*, brevemente descritas neste trabalho, possuem tempos (*time-boxed*), regras e objetivos claramente definidos no início de cada projeto e cabe ao *Scrum Master* (SM) garantir que as mesmas sejam entendidas e seguidas por todos os integrantes do Time *SCRUM*.

Na figura 5, abaixo, apresenta-se em forma gráfica um resumo do *Framework SCRUM*.

Figura 5. Framework SCRUM



Fonte: MORETI, Rodrigo (2017)



#### **4. EQUIPES AUTO GERENCIÁVEIS: - O QUE SÃO? PARA QUE SERVEM?**

A quantidade de pesquisas, estudos, artigos e livros dedicados ao tema Equipes é bastante ampla e não necessariamente recente, como exemplo pode-se citar o artigo do Dr. Bruce Tuckman – *Developmental Sequence in Small Groups* – que data de 1965, ou seja, a mais de 50 anos atrás. Contudo ainda é um tema atual e extremamente necessário para as empresas e organizações que pretendem se manter competitivas em um mercado cada vez mais agressivo. Reforçado pela valorização do componente humano, do capital intelectual, onde a criatividade e a inovação são fatores cada vez mais importantes e cruciais para a conquista e manutenção de mercados de consumo, em contraponto aos antigos sistemas de gestão de pessoas proposto por Taylor (1990), o desenvolvimento de equipes atuantes e efetivas tornam-se cada vez mais necessário e essencial.

Antes de serem apresentadas as principais características de equipes auto gerenciadas, algumas definições de grupos de trabalho devem ser mencionadas. Conforme Robbins (2005, p. 213), “um grupo de trabalho é aquele que interage basicamente para compartilhar e tomar decisões para ajudar cada membro em seu desempenho na sua área de responsabilidade”. Ainda conforme Robbins (2005) em um grupo de trabalho não existe a necessidade de um resultado baseado no trabalho coletivo que dependa do esforço conjunto de todos os integrantes do grupo, ou seja, o resultado do trabalho do grupo consiste apenas na somatória dos resultados individuais, em resumo, não existe sinergia entre os integrantes de um grupo de trabalho.

Quando fala-se de Equipes, Moscovici (2007, p. 5) ressalta que “o futuro pertence a organizações baseadas em equipes. Grupos existem em todas as organizações, equipes são raras ainda, embora ostentem essa denominação com frequência”. Um grupo passará a ser considerado uma equipe quando todos compreendem, definem e comprometem-se em uma meta, um objetivo e trabalharão juntos, reunindo as habilidades complementares de seus membros para alcançar os resultados. E que para que este estágio de sinergia seja atingido, a comunicação livre, verdadeira deve ser estimulada. As divergências precisam ser clarificadas e resolvidas de forma respeitosa e espontânea. A busca constante pelo aprimoramento, pela resolução de conflitos, pela evolução da equipe deve permear todas as atividades. Moscovici descreve muito bem estas características no trecho abaixo:

“Pode-se considerar equipe um grupo que compreende seus objetivos e esta engajado em alcançá-los, de forma compartilhada. A comunicação entre os membros é verdadeira, opiniões divergentes são estimuladas. A confiança é grande, assumem-se riscos. As habilidades complementares dos membros possibilitam alcançar resultados, os objetivos compartilhados determinam seu propósito e direção. Respeito, mente aberta e cooperação são elevados. O grupo investe constantemente em seu crescimento. Um grupo transforma-se em equipe quanto passa a prestar atenção à sua própria forma de operar e procura resolver os problemas que afetam o seu funcionamento” (MOSCOVICI, 2007, p. 5)

Robbins (2005, p.213) definiu que uma equipe de trabalho se diferencia de um grupo de trabalho, pois a “equipe de trabalho gera uma sinergia positiva por meio do esforço coordenado. Os esforços individuais resultam em um nível de desempenho maior do que a soma daquelas contribuições individuais”.

"Equipe é um pequeno número de pessoas com conhecimentos complementares, compromissados com propósito, metas de performance e abordagem comuns, e pelos quais se mantêm mutuamente responsáveis". (KATZENBACH; SMITH, 1994, p. 42).

Goldbarg (1995, p.52) define equipe como sendo: “Um pequeno grupo de pessoas com habilidades complementares que, comprometidas com um propósito comum, coordenam esforços e responsabilidades de forma a perseguir uma missão. ”

A fim de consolidar e resumir as principais diferenças entre grupos de trabalho para equipes, abaixo tem-se a Tabela 1, extraída de Katzenbach e Smith (1994)

**Tabela 1: Diferenças entre Grupo de Trabalho e Equipe**

| <b>TABELA 1 – Diferença entre Grupo de Trabalho e Equipe</b>  |  |
|---|--|
| <b>Grupo de Trabalho</b>  | <b>Equipe</b>  |
| <ul style="list-style-type: none"> <li>• Há um líder explícito;</li> <li>• A responsabilidade é do líder;</li> <li>• A responsabilidade é de apenas alguns dos membros do grupo;</li> <li>• Há grupos de especialistas;</li> <li>• As reuniões são “racionais” e chegam logo a conclusões;</li> <li>• Mensura-se sua efetividade indiretamente em outros grupos;</li> <li>• As decisões ocorrem pelo voto da</li> </ul> | <ul style="list-style-type: none"> <li>• Papéis de liderança compartilhados;</li> <li>• Comprometimento mútuo;</li> <li>• Responsabilidades compartilhadas;</li> <li>• Multiplicidade de funções;</li> <li>• Encorajamento da aparição de divergências de opiniões e conflitos construtivos;</li> <li>• Mensuração de seus resultados pelo desempenho de cada indivíduo do grupo;</li> <li>• Decisões por consenso;</li> </ul> |

|   |   |
|---|---|
| maioria;<br><ul style="list-style-type: none"> <li>• A decisão final é do líder;</li> <li>• Os objetivos são “clarificados” ou sugeridos pelo líder.</li> </ul> | <ul style="list-style-type: none"> <li>• Decisão final da Equipe;</li> <li>• Objetivos aceitos e incorporados por todos os membros do grupo.</li> </ul> |
|---|---|

Fonte: Katzenbach e Smith (1994, p.39)

E dentro do conceito de equipes, existe um tipo específico de equipe que é objetivo deste estudo – Equipes Auto Gerenciáveis – conforme Robbins (2005) este tipo de Equipe consiste de grupos de pessoas que desenvolvem trabalhos altamente relacionados e/ou interdependentes, onde seus integrantes assumem total controle e responsabilidade sobre uma série de aspectos que normalmente são destinados a líderes, supervisores e/ou gerentes. Em uma Equipe Auto Gerenciada, não existe um líder dizendo o que cada um deve fazer, não existe um responsável por atribuir tarefas e estipular prazos, os integrantes destas equipes assumem integralmente o controle sobre o planejamento e sobre o cronograma das atividades. A atribuição de tarefas aos seus membros ocorrer de forma natural, baseada nas melhores qualificações e habilidades de cada um de seus integrantes. A identificação de problemas de desempenho, a tomada de decisões operacionais sobre as tarefas e a implementação de ações para solucionar problemas ocorre integralmente dentro das Equipes Auto Gerenciadas, ou seja, os seus integrantes, em conjunto, resolvem seus próprios problemas, sem a necessidade de um gerente, líder, supervisor e/ou executivo controlando e dizendo o que deve ser feito, quando deve ser feito e principalmente como deve ser realizada a tarefa ou resolvido o problema.

Nitin Mittal (2013) descreve muito bem as características de uma Equipe Auto Gerenciável em um artigo publicado no Site da Scrum Alliance<sup>5</sup>:

- Assumem o trabalho espontaneamente. Não esperam que seu líder atribua o trabalho. Isso garante um maior senso de propriedade e comprometimento.
- Gerenciam seu trabalho (alocação, realocação, estimativa, reavaliação, entrega e retrabalho) como um grupo.
- Mesmo assim, ainda precisam de orientação e treinamento, mas eles não exigem "comando e controle".

<sup>5</sup> Fundada em 2001, a Scrum Alliance® é a maior, mais estabelecida e influente organização de certificação e associação profissional na comunidade Agile - Veja mais em: <https://scrumalliance.org/about-us#sthash.CjKalv7e.dpuf>

- Comunicam-se mais uns com os outros e seu comprometimento está mais para a Equipe do Projeto do que para o *Scrum Master* e/ou Gerente de Projeto.
- Entendem os requisitos e não têm medo de fazer perguntas para esclarecer suas dúvidas.
- Continuamente melhoraram suas próprias habilidades e recomendam ideias inovadoras bem como melhorias.

Outra definição clara e resumida de equipes auto gerenciadas pode ser facilmente encontrada na internet (Google) – “Equipe auto gerenciada é um grupo de trabalho pequeno que divide a responsabilidade pela execução de uma tarefa, e onde pressupõe-se que os membros tenham capacidade e autoridade para supervisionar a si mesmos”.

Ou da forma que apresenta Chatfield (2009) em sua definição de Equipes Auto Gerenciadas – “*Self-Managed Team: A group of people working together in their own ways toward a common goal which is defined outside the team - (Team does their own work scheduling, training, rewards and recognition, etc.)*”. Neste mesmo artigo Chatfield apresenta algumas vantagens relacionadas a Equipes Auto Gerenciadas, explicitadas na Tabela 2, abaixo:

**Tabela 2: Resultados obtidos com Equipes Auto Gerenciadas**

| <b>TABELA 2 – Resultados obtidos com Equipes Auto Gerenciadas</b>   |  |
|---|--|
| <b>MAIS</b>   | <b>MENOS</b>   |
| <ul style="list-style-type: none"> <li>• Entusiasmo</li> <li>• Aprendendo com colegas</li> <li>• Conforto sabendo que a ajuda está lá</li> <li>• Camaradagem</li> <li>• Responsabilidade compartilhada</li> <li>• Foco na organização</li> <li>• Responsabilidade da equipe</li> <li>• Medição simples e visível</li> </ul> | <ul style="list-style-type: none"> <li>• Opinião individual sobre o que é importante</li> <li>• Dependência em habilidades individuais</li> <li>• Pânico quando há picos de carga de trabalho</li> <li>• Calúnia</li> <li>• Protegendo informações</li> <li>• O que há para mim?</li> <li>• Estresse no "supervisor"</li> <li>• Sentimento de não realizado</li> </ul> |

Fonte: CHATFIELD, Mark M. *Self-Directed and Self-Managed Teams* (2009) (tradução nossa)

## 5. A JORNADA DO *DEVTEAM* RUMO AO *PERFORMING* E O PAPEL DO SM

Uma vez contextualizados os Projetos *Waterfall* de TI na Empresa “Z”, o *framework SCRUM* e as principais características de Equipes Auto Gerenciáveis, inicia-se a análise, o estudo e a apresentação de uma série de assuntos, matérias, disciplinas e teorias, que se propõem a auxiliar o *Scrum Master* (SM) na formação e/ou condução de *DevTeams* desde o estágio inicial (*Forming*) até o *Performing*, passando pelos estágios *Storming* e *Norming*, para serem então consideradas Equipes Auto Gerenciadas.

### 5.1. *DevTeams SCRUM* na Empresa “Z”

Nem todos os projetos de TI da Empresa “Z” são conduzidos com metodologias ágeis ou *SCRUM*. Um dos principais motivos que tem dificultado a adoção do *SCRUM* esta relacionado à composição do *DevTeam*, que como já apresentado, deve ser composto por integrantes multidisciplinares e que tenham total condições de entregar um produto em condições de ser utilizado pelo cliente e/ou PO (*Definition of Done*). A maioria dos projetos de TI da Empresa “Z” demanda atividades de vários componentes/sistemas diferentes, que por sua vez estão sob a responsabilidade de departamentos/gerencias segregadas, dificultando em muito a possibilidade de alocação/criação de um *DevTeam* com plenas condições de desenvolver o projeto em sua totalidade. Contudo, conforme as culturas ágeis e do *SCRUM* vão sendo disseminadas e cada vez mais utilizadas tanto na Empresa “Z”, quanto no mercado de TI, os projetos começam a serem pensados, concebidos e organizados de uma forma mais segregada e as condições para a composição de um *DevTeam* que atue em um, ou poucos sistemas, torna-se cada vez mais comum.

Desta forma, alguns Gerentes de Projetos e outros especialistas que possuem sólidos conhecimentos sobre o *SCRUM* e/ou certificação de *Scrum Master* são alocados em projetos que atendem as condições para serem conduzidos com metodologias Ágeis. Aqui vale lembrar que nem como Gerente de Projetos e nem como *Scrum Master*, estes colaboradores não possuem/exercem nenhum cargo de gestão e/ou liderança hierárquica sobre os integrantes do *DevTeam*. Contudo mesmo nestas condições o SM deve utilizar seu poder de convencimento e influência para que, como guardião dos processos *SCRUM*, o *DevTeam* seja composto por um conjunto de integrantes que tenham minimamente todas as condições de entregar componentes incrementais que possam ser considerados prontos (*Definition of Done – DoD*).

E para tanto, deve buscar o preenchimento de funções, papéis e responsabilidades básicas para o desenvolvimento de qualquer sistema de TI, tais como:

- Arquitetura de Sistemas, de Aplicação e Infraestrutura
- Banco de Dados (DBA)
- Análise e *Design*
- Construção (*Building*) e Testes Contínuos
- Testes Integrados e de Aceitação
- Implantação (*Deployment*)

Não esquecendo que a recomendação é de que o *DevTeam* não seja menor do que 3 e nem maior do que 9 integrantes, mais a frente será abordado em maiores detalhes alguns dos motivos deste tamanho de Equipe ser o mais indicado.

O SM também deve analisar e verificar o nível de interação já existente entre o *DevTeam* recém-formado. Existem grandes chances de muitos deles já terem trabalhado juntos em projetos formais, portanto conversas com gerentes e/ou coordenadores, entrevistas, reuniões abertas com o *DevTeam*, que venham a auxiliar o SM mapear o nível de interação e identificar eventuais problemas antes mesmo de iniciarem as primeiras *Sprints*, são bem-vindas e recomendadas.

Outro importante fator que o SM deve dar muita atenção e importância para que este grupo de colaboradores, acostumados com projetos formais, possa iniciar a jornada rumo a uma Equipe Auto Gerenciada está relacionado a co-locação e alocação. Todos os integrantes do *DevTeam* devem estar e trabalhar juntos, no mesmo ambiente físico e de certa forma isolados de outras equipes e eventuais interferências externas. O *DevTeam* deve estar 100% alocado no projeto em questão e somente neste projeto, ou seja, dedicação exclusiva. Estas características são tão importantes para a transformação de Grupo em Equipe, bem como para o próprio SCRUM, que aparece em 2 dos 12 princípios do Manifesto Ágil:

Pessoas de negócio e desenvolvedores devem trabalhar diariamente em **conjunto** por todo o projeto.

O método mais eficiente e eficaz de transmitir informações para e entre uma equipe de desenvolvimento é através de conversa **face a face**. (BECK, et al., 2001)

Tão importante quanto a co-locação do *DevTeam*, também é o quadro *Kanban* e o Gráfico de *Burn Down*, ou *SCRUM Board*<sup>6</sup>, já apresentados no capítulo 3 deste trabalho, seus motivos serão melhor abordados e apresentados no subcapítulo Motivação e Produtividade mais a frente, por hora reforça-se a importância que o SM deve dedicar para que a infraestrutura física exista para que montagem e utilização do *SCRUM Board*, uma pequena parede com folhas de papel coladas, canetas e post-it já é um bom começo.

## 5.2. A Teoria de Tuckman

Uma das teorias de grupo que mais é referenciada nas publicações sobre metodologias ágeis e *SCRUM*, foi elaborada por Bruce W. Tuckman e divulgada em 1965 na *American Psychological Association*, através de seu boletim n. 63 - *Psychological bulletin* (TUCKMAN, 1965). Tuckman analisou cerca de 50 artigos, a maioria deles referentes a experimentos psicológicos em grupos de terapia ou T-groups<sup>7</sup> e inicialmente os separou em dois grandes grupos: Interpessoal ou de estrutura e de Atividade de Tarefa. Após análises relacionadas a evolução sequencial destes grupos, Tuckman identificou e nomeou 4 estágios:

1. *Orientation/testing/dependence* → **Forming**
2. *Conflict* → **Storming**
3. *Group Cohesion* → **Norming**
4. *Functional Role-Relatedness.* → **Performing**

Mais tarde, em 1977 Tuckman e Mary Ann Jensen (TUCKMAN; JENSEN, 1977), após analisarem mais 22 artigos subsequentes ao estudo original de Tuckman, incluíram um 5º estágio a sua teoria de grupos, chamado **Adjourning** (*disengagement*).

- **FORMING:**

---

<sup>6</sup> Um *Scrum board* é parte integrante da metodologia Scrum. Isso torna o *Sprint backlog* visível e ajuda as equipes a gerenciar o desenvolvimento do produto do início ao fim. É uma ferramenta perfeita para rastrear o fluxo de trabalho e facilitar a sincronização diária. *Scrum board* pode ser desenhado em um quadro branco, um metal ou uma placa de cortiça ou mesmo organizado em uma seção de uma parede.

<sup>7</sup> A T-grupo ou grupo de treinamento (por vezes também referido como grupo de treinamento de sensibilidade, grupo de treinamento de relações humanas ou grupo de encontro) é uma forma de treinamento em grupo, onde os próprios participantes (normalmente, entre oito e 15 pessoas) aprendem sobre si mesmos (e sobre processos de pequenos grupos em geral) por meio da interação entre eles. Eles usam o *feedback*, a resolução de problemas e o jogo de papéis para obter *insights* sobre si mesmos, outros e grupos.

Em tradução livre – Formação – trata-se da primeira fase de um grupo, os integrantes são apresentados uns aos outros, começam a se conhecer, mas com parcimônia, ou seja, as pessoas não apresentam/demonstram suas reais intenções e a comunicação ocorre de forma amena, sempre tentando agradar os outros. Os papéis e as responsabilidades de cada um não são claras. E os membros do grupo eventualmente testam determinadas situações e expressam algumas ideias para identificar a aceitação (ou não) pelos demais integrantes. Neste estágio o grupo depende muito da condução e orientação do SM no que diz respeito aos processos e procedimentos e ao que se espera do *DevTeam*. Dinâmicas para apresentação e/ou introdução, do tipo nome, tempo de empresa, experiências anteriores, etc., ajudam a integração e o início de um relacionamento mais construtivo e duradouro.

- ***STORMING:***

Esta é a fase da Tempestade (tradução livre), onde os conflitos são frequentes e as decisões em grupo são muito complicadas e difíceis de serem atingidas, ou seja, o consenso praticamente não existe. Os integrantes do grupo forçam situações e todos brigam por seu espaço. A fase inicial de formação, onde todos estavam retraídos não existe mais, no *Storming*, todos querem dar sua opinião independente da aceitação ou não dos demais integrantes. O SM deve estar mais atento e atuante ainda, sempre lembrando o grupo dos seus objetivos imediatos, caso contrário, poderão perder muito tempo discutindo, debatendo e tentando resolver conflitos, deixando de lado o objetivo do projeto e/ou da *Sprint* que estiver em curso.

- ***NORMING:***

Nesta fase de Normatização (tradução livre), os conflitos relacionados a quem exerce qual papel dentro da Equipe já não existem mais, cada um sabe e aceita o seu lugar e a sua responsabilidade, todos reconhecem todos e se identificam como uma Equipe. Trabalham com coesão e sinergia, deixando de lado conflitos pessoais em prol da Equipe. As decisões são tomadas com bem mais velocidade e com pouca intervenção do SM, pois o sentimento de pertencimento a uma Equipe envolve a todos e os resultados positivos são mais frequentes e aparentemente mais facilmente alcançados. Neste estágio o SM deve estar atento para evitar qualquer desvio nos processos do *SCRUM* e intervenções externas que venha a desestruturar



a Equipe, seu trabalho deve ser muito mais com a organização (coordenadores e gerentes funcionais) na preparação dos mesmo para o próximo estágio – *Performing* – onde a Equipe se auto gerenciará.

- **PERFORMING:**

A Equipe atinge sua maioria/maturidade, neste estágio esta equipe já é considerada uma Equipe Auto Gerenciada, pois eles mesmo resolvem seus problemas, definem a carga de trabalho em conjunto, ninguém precisa definir alocação de tarefas ou quem faz o que. A constante melhoria e busca da eficiência já faz parte do dia-a-dia da Equipe, enfim todas as características já descritas no capítulo 4 estão presentes e são percebidas. O SM não é mais necessário, permanecendo por perto, mais para atuar no ambiente externo ao *DevTeam*, como o PO e o resto da organização.

- **ADJOURNING:**

Trata-se do estágio onde a dissolução da Equipe esta eminente, geralmente quando a conclusão do Projeto esta próxima. Sentimentos contraditórios entre os membros da Equipe são percebidos, alguns estão felizes e eufóricos devido ao sucesso do projeto e do que realizaram juntos (isso no caso de um projeto de sucesso), outros estão tristes, ansiosos e temerosos com o futuro, pois provavelmente serão remanejados e alocados em outro Grupo, retornando ao primeiro estágio – *Forming*. Aqui mais uma vez o SM deve estar presente para atuar como um líder-servo, buscando o alinhamento das expectativas e dos próximos passos com os gerentes funcionais, executivos, PO, etc., trazendo para os integrantes do *DevTeam* tranquilidade e equilíbrio para que a conclusão do projeto não seja impactada.

Qualquer evento que afete o Grupo/Equipe, automaticamente gera uma regressão, geralmente retornando para o estágio *Storming*, como por exemplo, a saída ou a entrada de um novo integrante, alguma mudança brusca de liderança e/ou gerencia funcional, portanto o SM deve estar atento e preparado para estas situações.

Na tabela 3, abaixo apresenta-se um resumo consolidado dos 5 estágios propostos por Tuckman, com algumas orientações ao SM e como os membros do grupo/equipe costumam se comportar.

Tabela 3: Cinco Estágios de Desenvolvimento de Grupos/Equipes

| <b>TABELA 3 – Cinco Estágios do Desenvolvimento de Grupos/Equipes</b> |  |  |  |   |
|---|--|--|--|---|
| <b>Estágio</b>  | <b>Descrição</b>   | <b>SM deve</b>   | <b>Membros</b>   | <b>Conflitos</b>  |
| <b><i>Forming</i></b>   | Grupo recém-formado e dependente de liderança e direção, alguma exploração de papéis, objetivos comuns são assumidos, necessidade de informação e orientação             | Dar instruções claras, ajudar a todos a sentirem-se seguros incluídos no grupo, ajudar a esclarecer o que precisa ser feito, dar <i>feedback</i> positivo      | Muitas vezes educados e hesitantes, raramente pedem esclarecimentos, cumprem as decisões, querem se sentir incluídos, podem estar nervosos e/ou ansiosos | Minimizadas, possivelmente em desenvolvimento sob a superfície, a maioria demonstra estarem conformados com o cenário |
| <b><i>Storming</i></b>  | À medida que os membros começam a tomar posse, conflito de tarefas e confusão surgem, esclarecimento sobre papéis, responsabilidade e metas são necessários              | Ajudar na condução dos trabalhos, facilitando a superação dos conflitos, recuar e permitir que os membros assumam a liderança e as responsabilidades           | Será mais vocal e envolvido, pode desafiar o líder, expressar confusão ou querer mudanças, deve se concentrar em tarefas                                 | Trabalhar através de conflitos produtivos pode criar/gerar confiança, o conflito pessoal pode impedir o progresso     |
| <b><i>Norming</i></b>   | Equipe estabelece normas de comportamento e tomada de decisão, todos compreendem o seu papel e tarefas, a comunicação é aberta e produtiva, o sentimento de afeto cresce | Tornar-se um facilitador que ajuda os membros a desenvolver suas próprias tarefas, apoiar mudanças que aumentam a produtividade, ouvir e encorajar o progresso | Facilitar uma comunicação clara, criar procedimentos eficazes de tomada de decisão, procurar normas que aumentem a participação e a coesão               | Definir normas para lidar com conflitos que possam surgir, mudar as coisas para reduzir confusão ou ambiguidade       |
| <b><i>Performing</i></b>  | A Equipe funciona bem e é capaz de tomar decisões e enfrentar desafios ou conflitos, estão animados e confiante em suas ações, avaliam o progresso                       | Agir como um consultor, participar plenamente sem dominar, olhar para os sinais de problemas ou de regressão   | Trabalhar bem com outros membros das equipes, podem realizar tarefas, sentir o sentimento de pertencimento   | Conflitos podem surgir, mas a Equipe pode funcionar bem e lidar com eles adequadamente                                |
| <b><i>Adjourning</i></b>  | A Equipe deve chegar a um fim e todo mundo vai sentir fortes emoções de perda, tristeza e possivelmente confusão. Precisa de uma chance de encerramento e apoio.         | Facilitar maneiras de dizer adeus e certificar-se de que a logística para a dissolução da Equipe esteja disponível   | Precisa de apoio emocional e descanso, podem lidar melhor se eles estão cientes de suas necessidades, podem sentir emoção ou ansiedade ao sair           | Conflitos podem surgir à medida que as emoções surgem, continuar a oferecer apoio                                     |

Fonte: CISV International. Five Stages of Group Development (tradução nossa)

### 5.3. Conformidade e Conflito, Criatividade e Inovação

Um ponto de atenção que deve ser constante em desenvolvimentos ágeis e principalmente no *SCRUM* é a busca constante pelo aperfeiçoamento, pelo aumento da produtividade, o *DevTeam* deve constantemente questionar os processos, as ferramentas, enfim o *modus operandi*<sup>8</sup> e durante a *Sprint Retrospective*, analisar e definir planos de ação para aumentar a eficácia e evoluir, degrau por degrau. Contudo, existe uma situação de conformidade que o SM precisa estar atento e evitar que ocorra com o *DevTeam*. Trata-se de um comportamento de grupo descrito por Robbins (2005) e independente do estágio que o *DevTeam* se encontra, a busca pela conformidade pode ocorrer.

Como em qualquer grupo/equipe cada membro, individualmente sempre buscará a sua aceitação pelo grupo/equipe e muitas vezes, motivados por este desejo de aceitação, um ou mais membros podem suprimir suas opiniões em busca da conformidade, submetendo-se a vontade do grupo/equipe. “Existem evidência consideráveis de que os grupos exercem forte pressão sobre os indivíduos para que mudem suas atitudes e comportamento a fim de que se adaptem aos padrões estabelecidos” (ROBBINS, 2005, p.194). O SM ciente deste tipo de comportamento, precisa monitorar os sinais de cada membro do *DevTeam*, principalmente nas reuniões de *Sprint Planning/Retrospective* garantindo que todos tenham sua chance de expressar suas opiniões livremente e questionar/esclarecer qualquer dúvida existente.

Em contraponto a conformidade, existe o conflito, que na medida certa, conduzido da forma correta, abrangendo somente as questões profissionais é muito saudável e útil para o desenvolvimento do *DevTeam*. Equipes onde não acontecem pequenos conflitos, que estimulem a discussão e o debate de ideias tornam-se apáticas e estagnadas. O SM deve sempre acompanhar e mediar os conflitos, sempre os trazendo para o ramo das atividades, das tarefas, pois conflitos de ordem interpessoal, de relacionamentos costumam ser disfuncionais e desestruturam a equipe.

Os conflitos são construtivos quando aumentam a qualidade das decisões, estimulam a criatividade e a inovação, encorajam o interesse e a curiosidade dos membros do grupo, oferecem um canal para arejar os problemas e liberar as tensões, e fomentam um ambiente de auto avaliação e de mudança. As evidências sugerem que o conflito pode

---

<sup>8</sup> *Modus operandi* é uma expressão em latim que significa “modo de operação”, na tradução literal para a língua portuguesa. Esta expressão determina a maneira que determinada pessoa/empresa utiliza para trabalhar ou agir, ou seja, as suas rotinas e os seus processos de realização.

melhorar a qualidade do processo decisório por permitir que todos os pontos sejam avaliados nas decisões importantes, especialmente aqueles que não são usuais ou que não são defendidos por minorias. (ROBBINS, 2005, p.332)

Conforme mencionado por Robbins, o conflito tem origem na divergência de opiniões e por sua vez a divergência é o ponto inicial para o desenvolvimento de soluções criativas e inovadoras. Osborn e Parnes (1979), desenvolvedores do *Creative Problem Solving* reforçam que é através da divergência que novas ideias podem surgir, reduzindo o risco da conformidade e da estagnação. Uma das técnicas apontadas para a resolução de conflitos e divergências de cunho processual e/ou da tarefa é o *Brainstorm*.

*Brainstorming*, (palavra originalmente cunhada por Alex Osborn), costuma ser mal interpretado como todo o processo criativo de resolução de problemas (CPS). Contudo trata-se apenas da fase de pensamento divergente do processo CPS. *Brainstorming* é uma forma de gerar muitas ideias, através de um processo limpo e agradável. E é importantíssimo que seja conduzido da forma correta, seguindo as orientações abaixo:

- **Julgamento Adiado:** O SM deve orientar o *DevTeam* que durante a sessão de *Brainstorm*, não devem ocorrer julgamentos. Todas as ideias são válidas e serão analisadas em um segundo momento (Convergência)
- **Combinar e construir:** Uma ideia pode ajudar a estimular outras ideias, incentive o *DevTeam* a combinar ideias e construir novas possibilidades
- **Sem limites:** Estimule a imaginação do *DevTeam*, as vezes as ideias mais absurdas são o estopim para uma solução criativa e inovadora. Crie um ambiente onde todos possam pensar fora da caixa<sup>9</sup>.
- **Busque Quantidade:** Incentive e procure que o *DevTeam* crie uma grande quantidade de ideias, utilizando todo o tempo da reunião, nesta fase quanto mais melhor. Evite que iniciem a julgar e pensar na viabilidade das ideias, mantenha o foco na geração de ideias.

Após a sessão de *Brainstorm*, pode-se efetuar um agrupamento das ideias por afinidade para então iniciar a fase de convergência, onde todas as ideias devem ser avaliadas, buscando

---

<sup>9</sup> A expressão “Pensar fora da caixa” oriunda do inglês “Think outside the box” conota “pensar livre das amarras convencionais” e tem sua origem controversa; a primeira versão do surgimento dela é a do consultor americano John Adair em 1969

a melhor alternativa possível ou uma combinação de alternativas que possam se complementar em uma solução totalmente nova. Nesta fase é importante que o SM garanta que:

- **Exista deliberação:** Evite decisões rápidas e sem julgamento, ou julgamento muito duros. Cada alternativa precisa de tempo e respeito para a tomada de decisão. Dê a cada opção uma chance justa.
- **Objetivos sejam sempre verificados:** Nunca deixe que o *DevTeam* perca o objetivo de vista. Uma alternativa que agrade a todos pode não estar totalmente alinhada com os objetivos da Equipe
- **As ideias sejam melhoradas:** Verifique a viabilidade das alternativas. Mesmo as ideias promissoras precisam ser aprimoradas e fortalecidas. Dedique tempo para o aprimoramento das ideias.
- **Todos sejam afirmativos:** É importante primeiro considerar o que é bom sobre uma ideia e efetuar um julgamento com o objetivo de melhorá-la, ao invés de eliminá-la logo de início.
- **A novidade seja considerada:** Não descarte ideias novas ou originais. As vezes uma ideia absurda, depois de adequada, trabalhada, pode se tornar a melhor alternativa possível.

#### 5.4. Comunicação e PNL - Metamodelos

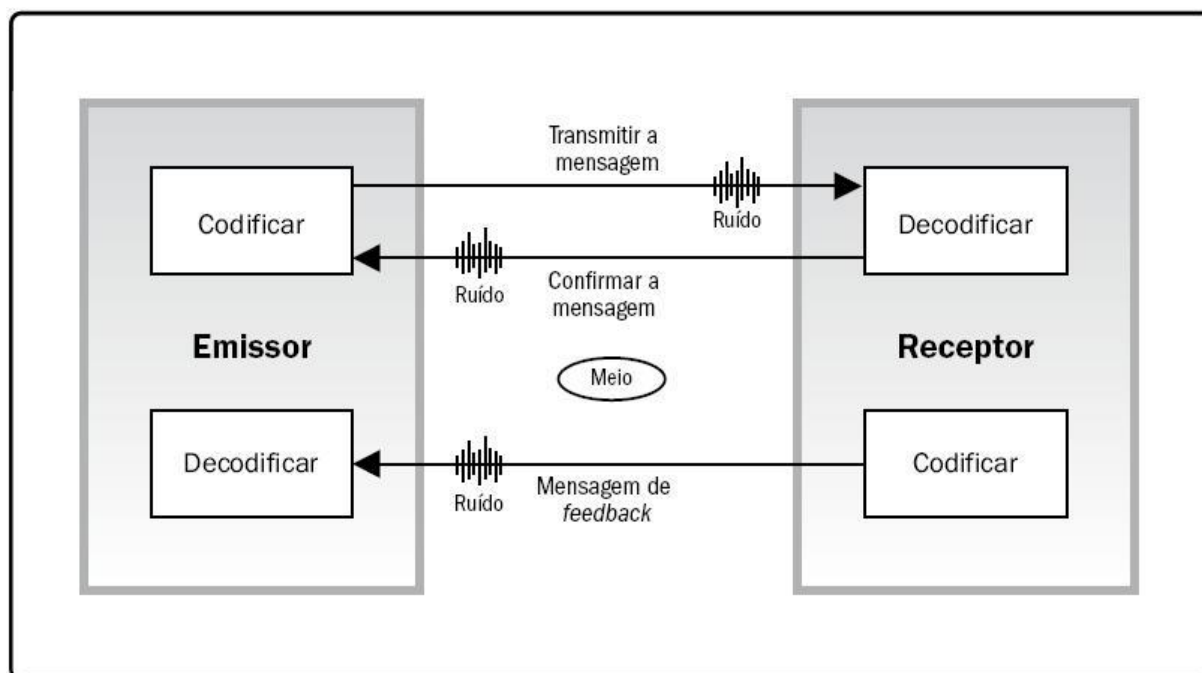
Como em qualquer relacionamento interpessoal a comunicação clara, objetiva é de extrema importância, pois trata-se da principal forma de interação entre os indivíduos. Contudo conseguir se expressar de uma forma clara, objetiva e assertiva é um desafio que precisa ser enfrentando diariamente. Existe uma citação atribuída aos Contos Árabes que expressa perfeitamente o poder destrutivo da comunicação quando está é mal feita.

Um dos grandes desafios da humanidade é aprender a arte de comunicar-se. Da comunicação depende, muitas vezes, a felicidade ou a desgraça, a paz ou a guerra.

Que a verdade deve ser dita em qualquer situação, não resta dúvida. Mas a forma com que ela é comunicada é que tem provocado, em alguns casos, grandes problemas. A verdade pode ser comparada a uma pedra preciosa. Se a lançarmos no rosto de alguém pode ferir, provocando dor e revolta. Mas se a envolvemos em delicada embalagem e a oferecemos com ternura, certamente será aceita com facilidade. (ANÔNIMO)

Desta forma, o SM deve primeiramente aprimorar a sua comunicação e através de um trabalho constante de orientação e *coaching*<sup>10</sup> indireto auxiliar os integrantes do *DevTeam* a aprimorarem a sua comunicação. E a melhor forma de iniciar este trabalho é através de conscientização de que a boa comunicação depende dos dois principais atores, o emissor e o receptor. Não basta apenas o emissor transmitir a sua mensagem e assumir que o receptor irá entendê-la perfeitamente. A responsabilidade de checar, verificar se o entendimento está correto é também do receptor da comunicação. E caso o receptor não busque espontaneamente confirmar a mensagem, cabe ao emissor solicitar especificamente o *feedback*. Muitas vezes uma simples pergunta direcionada ao receptor do tipo: “Por favor explique-me o que você entendeu? ”, pode eliminar muitos desentendimentos, evitando desgastes e principalmente perda de tempo. A fim de exemplificar melhor este fluxo, abaixo segue a Figura 6, onde estão representados os componentes básicos de uma comunicação efetiva e ideal:

**Figura 6. Fluxo Ideal da Comunicação**



Fonte: GUIA PMBOK (PMI, 2013)

A Comunicação entre os integrantes de uma Equipe é tão importante que esta diretamente ligada ao tamanho da mesma. Jeff Sutherland (2014) deixa isso muito claro

<sup>10</sup> *Coaching* é uma palavra em inglês que indica uma atividade de formação pessoal em que um instrutor (*coach*) ajuda o seu cliente (*coachee*) a evoluir em alguma área da sua vida.

quando explica os motivos da recomendação do *SCRUM* de que os Times de Desenvolvimento (*DevTeam*) não devam ter mais do que 9 pessoas. O ideal é 7, com margem de 2 para mais ou para menos. Esta recomendação baseia-se em uma coletânea de estudos científicos realizados por Goerge Miller (1956), Fred Brooks (1975) e Nelson Cowan (2001) que, analisados por Sutherland, concluem que o cérebro humano possui uma limitação relacionada à quantidade de itens que consegue gerenciar simultaneamente. E quando falamos de comunicação dentro de uma equipe, o número de canais de comunicação aumenta drasticamente de acordo com a quantidade de pessoas envolvidas. E o cérebro humano simplesmente não consegue lidar com uma quantidade excessiva de canais de comunicação, ou seja, não consegue gerenciar muitas coisas em paralelo. A quantidade de canais de comunicação pode ser calculada através da fórmula  $n(n-1)/2$ , onde “n” é a quantidade de integrantes do grupo/equipe.

Assim como em uma equipe das Forças Especiais, todos em uma equipe Scrum têm de saber o que os outros estão fazendo. Todo o trabalho que está em andamento, os desafios enfrentados, os progressos feitos, tudo em de ser transparente para todos. E se as equipes ficarem grandes demais, a capacidade de todos se comunicarem de forma clara, ao mesmo tempo, fica confusa. Há informações cruzadas demais. (SUTHERLAND, 2014, p.65)

Além dos aspectos já mencionados relacionados à comunicação, existem outras importantes ferramentas que o SM pode utilizar para orientar o *DevTeam* em seus inter-relacionamento, bem como nas reuniões específicas com o PO para esclarecimentos de dúvidas e entendimento efetivo dos objetivos do projeto, dos requisitos. Trata-se da PNL – Programação Neurolinguística, mais especificamente os seus Metamodelos<sup>11</sup>.

Vamos começar pelo começo. O que é PNL? Mas essa é uma pergunta capciosa. Não se pode reduzir a PNL a uma única definição. Há muitas explicações do que seja PNL, cada uma com um feixe de luz brilhando de um ângulo diferente, iluminando inteiramente a forma e a sombra do objeto. (O’CONNOR, 2015, p. 1)

Como apresentado por O’Connor (2015), a PNL possui muitas definições e aplicações. Neste trabalho concentrar-se-á naqueles aspectos que venham a auxiliar a comunicação do Time *SCRUM* (PO + *DevTeam* + SM):

---

<sup>11</sup> O Metamodelo é um conjunto de padrões de linguagem e perguntas que reconectam as deleções, distorções e generalizações à experiência que as gerou.

- “PNL é uma atitude e uma metodologia, que deixam um rastro de Técnicas” (Richard Bandler)
- “PNL é a influência da linguagem sobre nossas mentes e nossos comportamentos subsequentes”
- “PNL é o estudo sistêmico da comunicação humana” (Alix Von Uhde)
- “A PNL explora como seus pensamentos (neuros) são afetados por palavras (linguística), levando à ação (programação)” (O’Connor)

Quando as pessoas se expressam, através da linguagem, elas tentam representar aquilo que estão sentindo, aquilo que imaginaram, através de simples palavras. A riqueza de um pensamento, de uma ideia, a totalidade da experiência sensorial da pessoa fica reduzida a uma simples coletânea de palavras, muitas vezes transcritas em algum documento, como por exemplo, as *User Stories*. Em contraponto a este tipo de problema o próprio Manifesto Ágil incentiva muito mais a conversa direta com o PO, do que a documentação.

Indivíduos e interações mais que processos e ferramentas.

Software em funcionamento mais que documentação abrangente.

Colaboração com o cliente mais que negociação de contratos.

Responder a mudanças mais que seguir um plano. (BECK, et al., 2001)

Contudo, mesmo na comunicação direta, face a face, a transformação daquilo que realmente deseja-se expressar, em palavras, acaba sofrendo 3 tipos de transformações:

- **Deleção ou Omissão:** Onde certos aspectos são simplesmente omitidos (deletados), pois para o emissor, podem ser óbvios e não serem necessários, ou simplesmente por não encontrar palavras adequadas para se expressar.
- **Generalização:** O cérebro humano sempre buscará alguma experiência passada, um exemplo como representativo de uma classe de experiências, ignorando as particularidades, as nuances de cada caso.
- **Distorção:** Devido às experiências passadas, alguns aspectos terão mais peso, maior importância do que outros.

“A PNL sugere que deletamos, distorcemos e generalizamos nossa experiência quando a transformamos em representações internas. Então, nossa escolha de palavras para descrever essas experiências deleta, distorce e generaliza tudo mais uma vez” (O’CONNOR, 2015, p.



158). E para que o *DevTeam* obtenha maior clareza em suas comunicações, principalmente com o PO, a melhor ferramenta a ser utilizada são as Perguntas. Perguntas são a melhor forma de evitar mal-entendidos, fazendo perguntas corretas, no momento oportuno podemos dirimir dúvidas provenientes das deleções ou omissões, generalizações e distorções. Na sequência apresenta-se um resumo dos tipos de situações onde a Metamodelagem pode ser empregada para a obtenção de todas as informações necessárias para o *DevTeam* e/ou para a resolução de eventuais conflitos internos devido as diferenças de perfis dos integrantes do *DevTeam*. “As perguntas do Metamodelo fazem a engenharia inversa da linguagem, trabalhando a estrutura superficial para obter insight sobre a estrutura profunda por trás dela” (O’CONNOR, 2015, p.162)

- **DELEÇÕES OU OMISSÕES**

- Deleções Simples: Algo importante é omitido de uma sentença
  - Exemplo: Isso/aquilo é muito importante para a *Sprint*
  - Pergunta: Exatamente o que é muito importante?
- Índice Referencial não especificado: Pessoa ou coisa que empreende ou sofre a ação não é mencionada
  - Exemplo: Eles nunca escutam o que eu digo?
  - Pergunta: Quem exatamente nunca escuta o que você diz?
- Verbos não especificados: Deleção de como um evento ocorreu exatamente
  - Exemplo: Eu o convenci a fazê-lo durante a *Sprint*
  - Pergunta: Exatamente como o convenceu a fazê-lo?
- Julgamentos: Opiniões expressas como fatos
  - Exemplo: Isso não é bom o suficiente para apresentamos na *Review*
  - Pergunta: Você acha? Que padrão você está usando para julgar se é bom ou não?
- Comparações: Compara uma coisa com outra para avaliá-la
  - Exemplo: Se fizermos assim, fica mais fácil.
  - Pergunta: Fica mais fácil comparando especificamente com o que?

- **GENERALIZAÇÕES**

- Operadores modais de necessidade: Estabelecem regras quanto ao que é necessário e apropriado.
  - Exemplo: Você não deve cometer erros.

- Pergunta: O que aconteceria se erros fossem cometidos?
- Operadores modais de possibilidade: Estabelecem regras quanto ao que é possível: Pode, não pode, possível e impossível.
  - Exemplo: Não posso dizer a eles o que realmente estou pensando sobre o futuro deste projeto
  - Pergunta: Do que você tem medo? E se você disse-se, o que aconteceria?
- Universais ou quantificadores universais: Padrão de generalização, quando são utilizadas palavras como: Sempre, nunca, todo mundo, ninguém, etc.
  - Exemplo: Tudo esta sempre dando errado neste projeto
  - Pergunta: Realmente tudo sempre dá errado? Não existe nada que nunca deu certo?
- **DISTORÇÕES**
  - Nominalizações: Um verbo e transformado em substantivo abstrato
    - Exemplo: O relacionamento da equipe está indo morro abaixo
    - Pergunta: O que esta causando problemas na maneira pelo qual a equipe esta se relacionando?
  - Leitura mental: Supõe-se saber o estado interno da outra pessoa sem qualquer evidência.
    - Exemplo: Eles pensam que sou um idiota
    - Pergunta: O que o faz acreditar nisso? Como exatamente você chegou a esta conclusão?
  - Causa e efeito: A resposta da pessoa pode basear-se no comportamento de alguém.
    - Exemplo: A voz do PO me irrita
    - Pergunta: Então a voz dele o irrita. Que outra reação poderia ter? Gostaria de ter escolha sobre o que sente?
  - Equivalentes complexos ou equivalência complexa: Duas afirmações ligadas de forma que uma signifique outra.
    - Exemplo: Ele esta sempre atrasado (portanto) ele não se importa (o comportamento de estar atrasado é tornado equivalente ao de não se importar)

- Perguntas: Você acha mesmo que chegar atrasado significa que ele não dá importância para o projeto? Pode significar que ele esteja com outros problemas em casa? Ou com dificuldades de locomoção?
- Pressuposições: A fala contém suposições de alguma coisa relativa à situação que advém do seu mapa do mundo, de sua experiência.
  - Exemplo: Por que você não consegue fazer nada certo e ajudar o time a entregar o objetivo da *Sprint*?
  - Perguntas: O que se tem que fazer para que esteja certo?

Com base nos exemplos apresentados, o SM pode auxiliar o diálogo entre o Time *SCRUM* e trabalhar diretamente na comunicação e esclarecimento de dúvidas e/ou mal-entendidos. Através destes tipos de perguntas, obtém-se melhores respostas e pode-se prosseguir, efetuando mais perguntas específicas com base nas respostas anteriores e assim sucessivamente até que se descubra os reais motivos que se escondem nas palavras das pessoas, ou melhor ainda, aumentando a perspectiva do interlocutor para que ele enxergue a situação por outra perspectiva, ampliando o seu mapa de mundo, a sua experiência.

### **5.5. Inteligência Emocional**

Conforme já apresentado, o *SCRUM* como um *framework* para desenvolvimento Ágil de Softwares, prioriza mais o relacionamento interpessoal, do que processos e ferramentas, da mesma forma, valoriza muito mais a resposta as mudanças do que seguir um plano, também tem seus processos, papéis e responsabilidades, baseados nos pilares da transparência, inspeção e adaptação. Com isso, percebe-se que a importância atribuída as pessoas, aos indivíduos, e a forma como os mesmos se relacionam durante um projeto é trazida à tona e torna-se um dos principais ingredientes para o sucesso (ou fracasso) de projetos ágeis, conduzidos baseados no *framework SCRUM*. E para que os relacionamentos interpessoais tenham maiores condições de tornarem-se efetivamente produtivos, independente dos diferentes pontos de vista, das opiniões divergentes, dos conflitos que serão gerados, o SM, como detentor (e defensor) dos processos *SCRUM*, precisa entender e principalmente treinar os demais integrantes do Time *SCRUM*, como as emoções podem ser entendidas e minimamente controladas. E uma das formas de se atingir este objetivo é através do estudo, entendimento e ensino do que é a Inteligência Emocional.

A Inteligência Emocional foi concebida por Daniel Goleman em 1995 (2001) e de forma resumida pode-se dizer que se trata de uma forma de dar inteligência as emoções. Os seres humanos não são robôs e por mais que se deseje, não é possível desvincular as emoções durante o horário de trabalho, durante o expediente. As pessoas, os indivíduos, carregam consigo todas as experiências de vida, todos os pré-conceitos, os pré-julgamentos, toda a bagagem emocional desde a sua mais tenra infância, e simplesmente não conseguem “desligá-las” durante a jornada profissional. Da mesma forma, os sentimentos/experiências recentes, positivos ou negativos, influenciam diretamente o humor e a forma de comportamento das pessoas. Não podemos esperar que um homem se comporte da melhor forma possível durante uma reunião de trabalho, se o mesmo tiver tido uma discussão séria com a esposa na noite passada. Com base nestas evidências Goleman propõe uma forma de autorreflexão, que gera autoconhecimento, para que se tenha condições de, pelo menos, entender como os sentimentos, as emoções, influenciam o comportamento e desta forma, poder lidar com os mesmo com inteligência. E como Goleman cita Aristóteles no início de seu livro, repete-se o mesmo neste trabalho, a fim de exemplificar o objetivo da inteligência emocional.

Qualquer um pode zangar-se, isso é fácil. Mas zangar-se com a pessoa certa, na medida certa, na hora certa, pelo motivo certo, e de maneira certa não é fácil. ARISTÓTELES, *Ética a Nicômaco*. (GOLEMAN, 2001, p. 2)

Para Goleman (2001) a Inteligência Emocional (QE – Quociente Emocional) pode ser entendida e aprimorada através do estudo e da prática de 5 habilidades que serão descritas abaixo:

- **Autoconhecimento:**

O autoconhecimento ou auto percepção é o primeiro passo que qualquer indivíduo deve dar para iniciar o aprimoramento de sua inteligência emocional. É preciso conhecer a si mesmo, não superficialmente, mas em profundidade. Autoconhecimento dos pontos fortes, dos pontos fracos, daquilo que lhe faz sentir medo, raiva, alegria, motivação, frustração, estresse, satisfação, contentamento, realização e tantos outros sentimentos e emoções que fazem parte da psique<sup>12</sup> humana.

---

<sup>12</sup> Psique é a palavra com origem no grego *psykhé* e que é usada para descrever a alma ou espírito.

A identificação das capacidades e dos limites também compõe outro aspecto importante para o autoconhecimento. Saber identificar o tamanho do desafio, o tamanho da tarefa, que pode-se assumir e ter consciência das limitações é essencial para evitar armadilhas onde costuma-se assumir compromissos maiores do que a capacidade. Neste sentido, o autoconhecimento facilitará o entendimento das dificuldades, promovendo as condições para a solicitação de auxílio, para o pedido de ajuda, para a conclusão da tarefa. Este tipo de atitude é essencial para integrantes de um *DevTeam*, pois evita situações de um membro da equipe assumir uma atividade maior do que sua capacidade e não solicitar apoio dos demais integrantes para a conclusão da mesma.

Através deste mapeamento das emoções e da capacidade, bem como a conscientização de quais situações e/ou cenários são os gatilhos para o desencadeamento de determinado comportamento, o indivíduo cria as bases para o próximo passo, o autocontrole.

- **Autocontrole:**

Administrar as próprias emoções, este é o objetivo. Com base nas informações obtidas através do autoconhecimento, deve-se lidar com os sentimentos de uma forma mais racional, evitando o sequestro da amígdala, ou sequestro emocional conforme define Goleman (2001). Este sequestro emocional nada mais é do que aqueles momentos onde o indivíduo “perde” a cabeça e suas reações são comandadas diretamente pelo cérebro límbico, sem a passagem pela amígdala, responsável pelo julgamento lógico. Situações de raiva, de medo, de estresse, geralmente são os gatilhos para o sequestro da amígdala e a partir do momento que se tem consciência destas situações, pode-se e deve-se monitorar, administrar e principalmente controlar tais emoções, efetuando os ajustes necessários para evitá-las e não “perder” a cabeça.

O autocontrole é fundamental para o desenvolvimento da inteligência emocional e o aprimoramento do relacionamento interpessoal, pois se evitam as situações de discussões acaloradas e até eventuais brigas internas nas equipes. Principalmente naquelas situações de divergências de opiniões mais frequentes no estágio *Storming*.

O autoconhecimento e autocontrole também são essenciais para o período de mudanças, e como já apresentado, o Manifesto Ágil (BECK, et al.,2001) presa muito mais a mudança do que seguir um plano. E é justamente nestes períodos de mudanças, onde o nível de incerteza aumenta drasticamente, que o autoconhecimento e principalmente o autocontrole

tendem a auxiliar os indivíduos a agirem com mais calma, efetuando uma análise cuidadosa dos fatos, capacitando-os a dizer não aos seus impulsos. Com isso ficam à vontade com a ambiguidade que a mudança traz ao dia-a-dia.

- **Automotivação:**

Pessoas com inteligência emocional elevada possuem uma grande capacidade de se auto motivarem. Isto quer dizer que, independentemente da situação ou do cenário apresentado, uma vez que conhecem e monitoram suas emoções, estas pessoas sempre buscam ajustar o ambiente e o seu comportamento para que se sintam motivadas, mantendo a positividade, o entusiasmo e o foco no resultado futuro sustentável.

Indivíduos auto motivados costumam encarar as tarefas de forma positiva, buscam desafios criativos, adoram aprender e se orgulham de um trabalho bem feito. Esta característica é imprescindível para integrantes do *DevTeam*, pois como já apresentado, o *SCRUM* prega a busca constante pelo aperfeiçoamento, a busca constante da “hiperprodutividade”, portanto, encerrar a próxima Sprint, com o desafio de entregar um pouco mais do que foi entregue na Sprint anterior, é o espírito desejado para o *DevTeam*.

- **Empatia:**

A empatia aparenta ser a mais conhecida característica relacionada à inteligência emocional, por outro lado também aparenta ser a mais difícil de ser aplicada. Nada mais é do que reconhecer a emoção das outras pessoas, ou seja, saber se colocar no lugar do outro e buscar o entendimento das emoções alheias, evitando o julgamento precipitado. Contudo, desenvolver esta capacidade durante o calor de uma discussão, não é uma atividade simples, para que isso seja viável, novamente o autoconhecimento e autocontrole precisam ser praticados, somente desta forma, o indivíduo poderá deixar de lado, mesmo que brevemente, o seu ponto de vista e buscar o entendimento dos reais motivos que levam o outro a “pensar” diferente de você. Empatia não significa tomar os sentimentos dos outros como sendo seus, mas sim levar em consideração de forma séria e honesta os sentimentos dos outros e com isso buscar um entendimento, uma solução mais agradável a todos.

A empatia tem papel fundamental para o desenvolvimento do grupo em equipe, pois quando os integrantes percebem que suas opiniões e suas emoções são entendidas e respeitadas pelos outros – não quer dizer que todas as opiniões serão aceitas e seguidas,

isso seria impossível – mas sim, que todos os sentimentos são respeitados e compreendidos, de forma verdadeira e honesta, sem julgamentos precipitados, o ambiente de integração necessário para uma equipe auto gerenciada torna-se estável, amistoso e acolhedor.

- ***Networking:***

*Networking*, sociabilidade ou simplesmente o manejo de relacionamentos profissionais completa a última das 5 habilidades necessárias para o aumento da Inteligência Emocional (QE). Contudo trata-se de expressão mais direta de um nível de Inteligência Emocional elevado. A sociabilidade pode ser encarada como uma forma de impulsionar as pessoas na direção desejada, como por exemplo, influenciando a decisão sobre as diretrizes de uma ideia que irá se tornar em um projeto, ou ainda indicando e convencendo pessoas a se juntarem em um novo *DevTeam* que esta sendo formado para uma nova fase de algum projeto. Pessoas que possuem uma grande rede de relacionamentos sabem que nada nesta vida, principalmente no desenvolvimento de softwares é feito por uma pessoa só. É através do *networking* que o indivíduo pode expressar seus elevados níveis de autoconhecimento, autocontrole, motivação e empatia, pois pessoas com uma boa motivação, empáticas e centradas (bom senso), costumam irradiar seu “brilho” destacando-se nas conversas e nos relacionamentos, tornando-se pessoas com as quais deseja-se estar e trabalhar junto.

O SM deve ser empático para poder conduzir não só o *DevTeam* em sua dura jornada até o *Performing*, mas também o PO, os gerentes/coordenadores diretos e todo o resto da organização, na implementação de times auto gerenciados. Da mesma forma, o SM precisa incentivar o *DevTeam* a aumentar a sua rede de relacionamentos, uma vez que na Empresa “Z”, a implementação completa de um projeto em produção necessitará do apoio de outras áreas afins, como infraestrutura e áreas de suporte a produção.

## **5.6. Assertividade e *Feedback***

Na mesma linha da comunicação e da inteligência emocional, existem outras duas perspectivas importantíssimas para o SM estar atento, buscando o aprimoramento constante e principalmente atuando para que o *DevTeam* desenvolva estas mesmas disciplinas em suas comunicações internas e/ou externas. Tratam-se da Assertividade e do *Feedback*, a capacidade de ser assertivo naquilo que deseja-se expressar tem se tornado cada vez mais

necessária na sociedade em que vivemos, assim como possuir a habilidade de emitir/receber um *feedback* de forma clara e justa, que independente de ser positivo ou negativo, venha a contribuir para o aprimoramento das atividades e das relações, é essencial na condução de um projeto através do *SCRUM*, que tem como base a avaliação periódica tanto pelo que se é entregue (*Sprint Review*), quanto pela forma que estão sendo conduzidas as atividades (*Sprint Retrospective*). E através das ferramentas apresentadas nos subcapítulos anteriores pretende-se disponibilizar o caminho necessário para que a assertividade e o *feedback* possam ser utilizados tanto pelo SM como pelo *DevTeam* para tornarem o dia-a-dia mais produtivo e efetivo, reduzindo os ruídos e os eventuais mal-entendidos que possam ser criados por uma comunicação e/ou um *feedback* indevidamente construídos.

Conforme Vera Lúcia Franco Martins (2002) ser assertivo significa ser afirmativo ou asserção no que se acredita ser verdadeiro. Na perspectiva das relações interpessoais, assertividade significa uma amplitude de respostas e de soluções positivas para todos os lados (ganha-ganha), gerando comprometimento para o atingimento dos resultados esperados. Ainda conforme Vera, ser assertivo significa dizer aquilo que realmente se deseja, de uma forma simples e clara, sem magoar e/ou desrespeitar o outro. É dizer “não” quando efetivamente se deseja dizer “não” e não um “sim” somente para agradar o outro, em resumo é afirmar o seu eu, a sua autoestima, sem denegrir os outros.

Dácio Bonoldi Dutra (2002) define assertividade como a habilidade de expressar opiniões, sentimentos, emoções, ideias, pontos de vista, etc., afirmando seus próprios direitos e interesses, de forma respeitosa e verdadeira, sem violar o direito dos outros.

Um elevado nível de autoconhecimento e autoestima é a base para a assertividade, pois se o indivíduo não sabe o que realmente quer, se não conhece os “porquês” dos seus sentimentos e suas vontades, não terá certeza de suas opiniões e muito menos certeza se deve ser firme e assertivo no momento de expressar suas ideias, suas opiniões e seus sentimentos. Por outro lado, os sentimentos de perda e não aceitação pelo grupo são os principais motivos para a falta de assertividade. As pessoas ficam com medo de dizer aquilo que realmente pensam, discordando dos demais, pois acreditam que desta forma serão mais facilmente aceitos pelo grupo, portanto a falta de assertividade será muito mais acentuada na fase de *Forming* e, portanto, o SM deve permanecer mais atento nesta fase, a fim de auxiliar e incentivar que todos os integrantes do *DevTeam* tenham pelo menos a oportunidade de expressar suas opiniões.



Neste ponto, pode-se efetuar um *link* das fases de grupos de Tuckman com os tipos de comportamentos relacionados a assertividade, mais uma vez o SM precisa avaliar a situação, identificar os comportamentos e promover um ambiente favorável para que o *DevTeam* possa evoluir e aprimorar o seu comportamento rumo ao *Performing* e a assertividade.

- Comportamento Passivo:

Este tipo de comportamento será mais intenso durante a fase de *Forming*, pois é onde os integrantes do *DevTeam* evitam o conflito e buscam a harmonia, mesmo que isso custe ir contra seus reais sentimentos, desejos e opiniões. Existe grande dificuldade em dizer “não”. Atitude defensiva, postura encolhida, cede facilmente adotando a opinião da maioria para evitar conflitos.

- Comportamento Agressivo:

Necessidade de autoafirmação, onde um tenta dominar o outro e fazer valer a sua vontade. Este tipo de comportamento será dominante durante a fase de *Storming*, onde os integrantes do *DevTeam* estão “brigando” para definir seus espaços e papéis, sempre medindo até onde podem forçar as opiniões individuais perante o grupo. As características deste tipo de comportamento agressivo são autoritarismo, intolerância, falta de respeito durante as reuniões, interrompendo a fala dos outros, demonstrando ironia e falta de interesse na fala dos demais.

- Comportamento Passivo-Agressivo:

Este comportamento é uma mistura dos dois anteriores e poderá ser observado em ambas as fases *Forming* e *Storming*, dependendo das características individuais de cada integrante do *DevTeam*. Os membros do grupo que demonstrarem este tipo de comportamento buscam a sua própria afirmação, mas evitando o conflito até ser ponto. Nesta situação buscam medir o quanto vale uma boa discussão contra a afirmação da sua opinião. Por muitas vezes utilizam-se de sarcasmo e senso de humor irritante para expressar seu descontentamento com as opiniões alheias.

- Comportamento Assertivo:

Trata-se do tipo de comportamento esperado por todos os integrantes do *DevTeam* que esteja nos estágios *Norming* ou *Performing*. Ser assertivo e não ser passivo e nem agressivo. Aqui os membros são capazes de expressar suas opiniões, vontades e

sentimentos de forma clara e objetiva, sem medo de retaliações, pois sabem que suas opiniões serão, antes de mais nada, respeitadas e entendidas pelos demais integrantes de grupo, isso não quer dizer que serão aceitas e acatadas, mas que serão consideradas. E a Equipe como uma unidade singular decidirá qual será a melhor alternativa diante de qualquer situação.

Para que isso seja possível, a autoestima e autoconhecimento devem ser trabalhados para que não exista receio de emitir uma opinião sincera e baseada em uma reflexão verdadeira. Também a empatia dos demais integrantes precisa estar em um nível elevado para que, mesmo que tal opinião seja contrária, possam entender, respeitar e aceitar a contradição. Para então, através do diálogo franco e honesto, encontrem a melhor alternativa para a equipe como um todo.

Neste ponto, o entendimento de como um *Feedback* construtivo deve ser feito e também aceito ganha sua real importância. Tratam-se dos *feedbacks* diários e constantes que ocorrem naturalmente durante o dia-a-dia das atividades do *DevTeam*, bem como a *Sprint Retrospective*. Da mesma forma, os *Feedbacks* que o PO emitirá durante o *Sprint Review* precisam ser direcionados da forma correta. Portanto cabe ao SM auxiliar todos os envolvidos nestes processos para que os *Feedbacks* possam ser construtivos e auxiliarem no desenvolvimento conjunto de todos os envolvidos.

Um dos principais segredos para uma comunicação assertiva e um *feedback* construtivo é o de se basear nos fatos para emitir suas opiniões, eliminando a acusação e julgamento do discurso. Conforme Goleman (2001) tudo piora quando uma queixa vem acompanhada de desprezo ou uma crítica pessoal.

A diferença entre queixa e crítica pessoal é simples. Numa queixa, a pessoa declara especificamente o que a irrita e critica a ação, não a pessoa, dizendo como se sentiu. “Quando você se esqueceu de pegar as minhas roupas na lavanderia, me deu a sensação de que não gosta de mim, de que não liga pra mim”. Está é uma expressão de inteligência emocional básica: assertividade, não beligerante e nem passiva. (GOLEMAN, 2001, p. 149)

No exemplo acima, deixa-se muito claro que um determinado comportamento (esquecimento de buscar as roupas na lavanderia) teve um impacto na percepção do outro (sensação de que não gosta de mim), possibilitando uma comunicação clara e objetiva, eliminando eventuais mal-entendidos e dando a oportunidade para um esclarecimento mais detalhado da situação.

O *Feedback* é uma ferramenta importantíssima para o desenvolvimento interpessoal, pois viabiliza que ambas as partes se ajudem mutuamente através de uma comunicação efetiva, onde os sentimentos são relacionados aos fatos, sem julgamentos e/ou qualquer outro tipo de deleção, generalização ou distorção. As cerimônias de *Sprint Review* e *Retrospective* previstas no *SCRUM* garantem que *Feedbacks* ocorram de forma constante e periódica, possibilitando ajustes de comportamento e percurso, em curtos intervalos de tempo, evitando situações onde muito tempo e esforço são gastos em um caminho infrutífero.

Para que um *Feedback* seja efetivo, algumas etapas precisam ser seguidas. Conforme Nascimento (1977), um bom *feedback* precisa ser aplicável pelo receptor, neutro, específico, direto e objetivo.

Um *feedback* deve ser aplicável pelo receptor quando descreve algum comportamento ou situação em que o receptor tenha plenas condições de alterar tal comportamento. De nada adianta criticar alguma situação em que o receptor não tem poder nem autonomia para efetuar qualquer alteração. *Feedbacks* do tipo “Eu não gosto disso ou daquilo” sem apresentar os fatos relacionados a este sentido, tornam-se inúteis para o receptor, pois não apresentam nada que possa ser aplicado na busca pela melhoria. Por outro lado, para que o *feedback* seja mais assertivo, a melhor mensagem ocorre quando é baseada em fatos. “Você fala muito alto e isto é desagradável para mim”. Nesta situação o receptor obtém dados concretos sobre o seu comportamento e pode decidir (ou não) ajustá-lo.

O *feedback* também deve ser neutro no sentido de eliminar qualquer equivalência complexa ou pressuposição, conforme apresentado no subcapítulo sobre PNL. Situações onde efetuam-se julgamentos devem ser evitadas e eliminadas do discurso. “Você chegou atrasado para a *Daily Meeting* mais uma vez, você não dá valor a Equipe”. Esta crítica não apresenta a neutralidade exigida por um *feedback* efetivo, pois está carregado de conclusões não baseadas em fatos. O mais adequado seria “Esta é a quarta (4ª) *Daily Meeting* que você chega atrasado, existe algum problema em especial? ”.

A especificidade é outra característica importante para qualquer *feedback*, deve-se evitar generalizações, onde a mensagem torna-se vaga e abstrata, pois desta forma o *feedback* perde sua força e significado. Situações onde, por exemplo, um integrante da equipe menciona que determinada pessoa está meio acomodada, sem apresentar os fatos que o levaram a chegar a esta conclusão, tornam-se ineficientes, pois o receptor não dispõe de

informações suficientes para o completo entendimento da mensagem. Ao invés disso, mencionar que já vem percebendo um comportamento de falta de interesse baseado na pouca (ou nenhuma) participação nas últimas cerimônias de *Sprint Planning* e *Retrospective* e que esse comportamento prejudica o desempenho da Equipe como um todo, traz muito mais informação e especificidade para o *feedback*, viabilizando um entendimento melhor do receptor que por sua vez tem a oportunidade de expor seus reais sentimentos e buscar ajustar este comportamento perante a Equipe.

Oportunidade. Todo e qualquer *feedback* tem sua hora e lugar. A cultura geral diz que, elogios podem ser feitos e público, contudo críticas devem ser feitas de forma isolada e discreta. Contudo o momento do *feedback* também deve ser considerado, quanto mais perto do fato gerador, da ocorrência e do problema em questão, melhor, se possível que seja imediato, mas cabe ao emissor do *feedback* analisar a situação e verificar se ambos (emissor e receptor) possuem condições físicas e psicológicas para que o *feedback* seja construtivo. Conforme menciona Nascimento (1977) nada melhor do que colocar as cartas na mesa e deixar tudo em pratos limpos quando algo não está indo bem, mas há que se analisar o cenário e as emoções em jogo. E caso seja percebido que tanto emissor quanto receptor do *feedback* não estejam em condições psicológicas apropriadas o melhor a fazer é aguardar algumas horas, ou até alguns dias, até que o clima seja mais favorável e os ânimos estejam mais apaziguados.

Comunicação direta. Para garantir a transparência e o mínimo do bom relacionamento entre os integrantes do *DevTeam*, o SM deve orientar que o *feedback* seja oferecido de forma pessoal e direta. Como “guardião” dos processos do SCRUM e orientador do *DevTeam*, por muitas vezes, receberá e/ou perceberá reclamações indiretas entre os membros da Equipe. É imprescindível que o SM oriente que o *feedback* seja pessoal e direto, principalmente se for negativo, de reprovação ou descontentamento. Um *feedback* negativo, corretamente aplicado, auxiliará o desenvolvimento da Equipe, contudo um *feedback* negativo, emitido por terceiros torna-se fatal para a transparência e para o relacionamento interpessoal dos envolvidos.

Objetividade é uma condição essencial para que um *feedback* seja benéfico. A mensagem precisa ser clara, focada e baseada em fatos e/ou exemplos, sem rodeios, minimizações ou discursos evasivos. Por exemplo, um integrante da equipe assume para si uma atividade complexa que não é sua especialidade, geralmente tais ações são criticadas com frases do tipo “Talvez fosse melhor você deixar esta atividade para Fulano, veja bem, ele é

especialista neste assunto...”, que por sua vez, não agregam valor ao receptor, pois estão carregadas de subjetividade. O melhor *feedback* neste caso seria “Na última vez que você lidou com uma atividade deste tipo, deixou a desejar nos quesitos X, Y e Z, prejudicando o resultado da *Sprint*. Então se você vai assumir esta *task*, sugiro focar nestes pontos para termos uma entrega mais consistente, se precisar Fulano pode ajudar com estes itens.”

Através das características expostas acima, um *feedback* torna-se eficaz e construtivo, sendo muito mais fácil de ser emitido e por sua vez recebido. O SM precisa estar muito atento e próximo do *DevTeam* em todas as suas atividades, principalmente nas fases iniciais de *forming*, *storming*, para que possa perceber comportamentos divergentes dos apresentados e, de forma sutil, direcionar e orientar os integrantes do *DevTeam* para ajustarem os seus comportamentos. E a melhor forma de efetuar tal direcionamento é através de *feedbacks* diários e constantes do SM para os integrantes do *DevTeam*. Da mesma forma, o SM também deve atuar junto com o PO para que nos momentos oportunos, como uma *Sprint Review*, o *DevTeam* não sofra com *feedback* ineficazes, vazios e por muitas vezes destrutivos. Portanto o SM deve se antecipar, principalmente nas primeiras *Sprints*, caso perceba que o objetivo da *Sprint* não será atendido completamente, para que o PO possa contribuir para o desenvolvimento do *DevTeam* através de *feedbacks* eficazes e construtivos.

### **5.7. Motivação, Produtividade, Delegação e Empowerment**

A motivação é essencial para o desenvolvimento de qualquer carreira profissional e é muito mais crucial para o sucesso de qualquer *DevTeam* que esteja atuando em projetos conduzidos através de metodologias ágeis e/ou *SCRUM*, uma vez que, a cada ciclo, a cada *Sprint* os integrantes do *DevTeam* devem renovar suas forças e reiniciar todo o processo através de uma nova *Sprint Planning*. Manter o ritmo das entregas e comemorar cada pequeno sucesso é importantíssimo para o moral e a motivação do *DevTeam*.

O coração do Scrum é o ritmo. O ritmo é profundamente importante para os seres humanos. Sua batida é ouvida na passagem do sangue pelas veias e arraigada nos recessos mais profundos do nosso cérebro. Estamos sempre em busca de padrões, levados a procurar o ritmo em todos os aspectos da nossa vida. (SUTHERLAND, 2014, p. 71)

Neste ponto, reforça-se a importância do SM atuar em conjunto com o PO e demais *stakeholders* que eventualmente venham a participar das cerimônias de *Sprint Review*. Independente do sucesso ou do fracasso da *Sprint*, o PO deve ter condições de emitir

*feedbacks* consistentes, construtivos e motivadores. Contudo, aqui cabe um alerta que deve ser abordado pelo SM, existe uma grande probabilidade de que as primeiras *Sprints* do projeto não atinjam seus objetivos e isso é perfeitamente normal para *DevTeams* pouco experientes em *SCRUM* ou recém-formados, uma vez que o *DevTeam* ainda não conhece sua capacidade e sua velocidade de entregas. Portanto o SM deve preparar o ambiente e se antecipar através de alinhamentos prévios com o PO, gerentes/coordenadores funcionais, demais *stakeholders* e principalmente o *DevTeam*.

O fracasso das primeiras *Sprint* pode ser muito desmotivador para um *DevTeam* recém-formado. E para contornar esta situação, o melhor processo indicado é investigar, debater e elaborar planos de ação, durante a *Sprint Retrospective*. Buscar identificar o que deu errado, o que não funcionou, quais foram as barreiras, etc., é essencial para a implementação de ações corretivas que venham a ajustar a velocidade e qualidade das entregas constantes previstas no *SCRUM*. Uma boa estratégia, que pode ser sugerida pelo SM é assumir uma quantidade menor de trabalho nas *Sprints* iniciais, a fim de auxiliar no atingimento da meta. Equipes bem-sucedidas já começam ganhando o jogo e um início vitorioso aumentará a confiança dos integrantes do *DevTeam* o que viabilizará o aumento gradativo da eficiência e eficácia das entregas, *Sprint* após *Sprint*. E não existe nada mais motivador do que ao final de uma *Sprint Review*, o *DevTeam* receber um *feedback* positivo do PO, como por exemplo: “O que vocês entregaram era exatamente o que eu esperava e já vai gerar um valor agregado para o negócio. Vamos em frente, estamos no caminho certo. Meus Parabéns!”

O ritmo constante de entregas de sucesso em cada *Sprint*, pode induzir a um círculo virtuoso de sucesso, motivação, satisfação pessoal e felicidade, conforme descreve Csikszentmihaly (2004) em seu livro *Good Business*, profissionais felizes são mais produtivos e motivados e podem atingir um estado de Fluxo (*Flow*). O *Flow* é considerado por Csikszentmihaly com uma das mais significantes manifestações de felicidade, pois está baseada na realização dos potenciais individuais de cada pessoa. E se manifesta através dos sentimentos de singularidade (diferenciação) pessoal, onde cada indivíduo sente-se realmente único e através da integração, quando a pessoa se reconhece como sendo parte de algo maior. Qualquer um que se sente único e integrado torna-se uma pessoa complexa e completa, sentimentos chaves para a felicidade.

O *Flow* se manifesta quando o profissional esta totalmente engajado e comprometido com a tarefa, sua concentração é total e a percepção de tempo e autoconsciência é alterada

(CSIKSZENTMIHALY, 2004). E O *SCRUM* viabiliza as condições necessárias para o estabelecimento do *Flow*, pois favorece as sete (7) condições essenciais para que ele ocorra:

1. Objetivos Claros: Através da definição clara e objetiva da missão de cada *Sprint*
2. Feedback Imediato: Através das *Daily Stand-Up Meetings* e *Sprint Review* e *Retrospective*
3. Combinação entre desafios e habilidades: Através da constante busca pela “hiperprodutividade”, aumentando a quantidade e qualidade das entregas de cada *Sprint*.
4. Foco/Concentração: Por definição do *SCRUM* o *DevTeam* deve estar 100% alocado e dedicado somente ao objetivo da *Sprint*.
5. Sentimento de Controle: Através da autonomia que o *DevTeam* possui em decidir como o trabalho deve ser realizado
6. Sensação de tempo distorcida: Através do foco e dedicação as tarefas da *Sprint* e da crescente capacidade de entrega do *DevTeam*.
7. Perda da autoconsciência/ego: A atividade, o objetivo da *Sprint*, a Equipe, tornam-se mais importante do que o indivíduo.

Conforme apresentado por Csikszentmihaly (2004) um dos pontos necessários para a motivação é o sentimento de controle, em outras palavras a autonomia, ou seja, a confiança imputada ao *DevTeam* pela Empresa “Z” para poder decidir a melhor forma de conduzir suas atividades. A capacidade e o poder delegado aos integrantes do *DevTeam* para definirem, em conjunto, quem realizará determinada tarefa é essencial para a motivação e incremento da produtividade.

A lógica subjacente é que ao envolver os trabalhadores nas decisões que os afetam e aumentar sua autonomia e controle sobre suas vidas profissionais, eles se tornarão mais motivados e comprometidos com a organização, mais produtivos e mais satisfeitos com seus trabalhos. (ROBBINS, 2000, p. 360)

Contudo, isso não acontecerá de uma hora para a outra e o papel do SM mais uma vez torna-se fundamental, não somente perante o *DevTeam*, mas principalmente junto aos gerentes/coordenadores diretos dos integrantes do *DevTeam* e os principais *Stakeholders* da Empresa “Z”, que estão relacionados ao Projeto em questão. É preciso que o SM atue

diretamente na forma como a autonomia, a delegação e o *empowerment*<sup>13</sup> é gradativamente concedido ao *DevTeam*.

Os limites da autonomia precisam ser claramente definidos, sem isso, o *DevTeam* ficará perdido no que diz respeito a até quanto eles podem se auto gerenciar e quando devem recorrer aos seus superiores diretos. Sem a definição desses limites, o *DevTeam* poderá cair no erro do excesso de precaução e não tomarão nenhuma decisão sem buscar auxílio.

O SM deve alertar a todos que erros aconteceram e que fazem parte da curva de aprendizado e evolução do *DevTeam*, portanto erros iniciais precisam ser tolerados e gerenciados da forma correta. O *DevTeam* precisa aprender e evoluir com seus erros. Essa é uma característica essencial para equipes auto gerenciadas. Os gerentes/coordenadores devem ser orientados para evitar agir ao primeiro sinal de problema e dar tempo para que o *DevTeam* avalie e sua situação e estabeleça alguma ação corretiva. Todavia o SM deve estar ciente de que poderá enfrentar dificuldades neste ponto, pois muitos gerentes/coordenadores podem apresentar algum receio diante da autonomia e *empowerment* do *DevTeam*, por entenderem que estarão perdendo o poder de controle sobre seus funcionários e/ou por serem muito centralizadores e pouco acostumados a delegar atividades completas para seus subordinados. Em contraponto a este tipo de comportamento o SM pode “lembrar” as vantagens de uma boa delegação, tais como:

- Liberar mais tempo para os Gestores atuarem em atividades mais estratégicas e de maior impacto para a área que lideram;
- Auxiliar no treinamento dos subordinados;
- Aumentar o grau de iniciativa, criatividade e inovação;
- Tornar os subordinados muito mais motivados, engajados e comprometidos, pois transmite a mensagens de confiança, responsabilidade e respeito pelo trabalho que estão realizando.

Além da motivação, baseada no ritmo das entregas, na delegação e no *empowerment*, o SM deve manter-se sempre atento a produtividade do *DevTeam*, não somente pela importância do incremento da produtividade para a moral e motivação do *DevTeam*, mas

---

<sup>13</sup> Empowerment é um conceito de Administração de Empresas que significa "descentralização de poderes", ou seja, sugere uma maior participação dos trabalhadores nas atividades da empresa ao lhes ser dada maior autonomia de decisão e responsabilidades.



também pelo interesse que os principais *stakeholders* do projeto demonstrarão com relação a aplicação de metodologias ágeis e a autonomia concedida ao *DevTeam* na Empresa “Z”. Uma das formas mais usadas para medir a produtividade do *DevTeam* no *SCRUM* é através da quantidade de *Story Points* entregues por *Sprint*.

Estimate - the team’s initial assessment of how much work is needed to implement this story compared to other stories. The unit is story points and usually corresponds roughly to “ideal man-days”. Ask the team “if you can take the optimal number of people for this story (not too few and not too many, typically two), and lock yourselves into a room with lots of food and work completely undisturbed, after how many days will you come out with a finished, demonstrable, tested, releasable implementation?” If the answer is “three guys locked into a room it will take approximately four days” then the initial estimate is 12 story points. (KNIBERG, 2015 p. 6)

Espera-se que após algumas *Sprints*, o *DevTeam* descubra a sua velocidade real de entrega e a mantenha no decorrer do projeto, buscando sempre incrementar tal capacidade, através da revisão e adaptação constante. Contudo, existem algumas armadilhas que precisam ser evitadas e uma delas é chamada de “folga social”, que conforme Robbins (2005, p. 197) “trata-se da tendência que as pessoas têm de se esforçar menos ao trabalhar em grupo do que se estivessem trabalhando sozinhas”. Este comportamento desafia a lógica de que a soma dos trabalhos de cada indivíduo deveria ser, no mínimo, igual a soma do trabalho da equipe, sendo que o esperado é que o resultado do trabalho de uma equipe seja maior do que a soma das partes, ou seja, trata-se de um fenômeno que contraria o próprio sentido da equipe, mas infelizmente ele pode ocorrer. Acredita-se que este tipo de comportamento pode ser motivado pela percepção por um integrante da equipe, de que os demais membros não estejam trabalhando tanto quanto deveriam, ou seja, não estão dando seu melhor, assim como este integrante está, e percebendo este tipo de situação, tal integrante também reduz seu ritmo para igualar a carga de trabalho. Outro motivador para a “folga social” seria a dispersão ou compartilhamento de responsabilidade, ou seja, os resultados da equipe não podem ser atribuídos a um único integrante, portanto se 1 falhar, todos falharão. Consequentemente a quantidade de integrantes de uma equipe está diretamente relacionada a probabilidade da “folga social”, portanto a recomendação para equipes formadas por no máximo 9 pessoas é mais uma vez reforçada.

Outras formas de evitar a “folga social” já encontram-se presentes no próprio *framework SCRUM*, basta o SM manter-se atento e atuar nos primeiros sinais em que ela

venha a se manifestar. A *Daily Stand-Up Meeting* é a principal ferramenta para evitar a “folga social”, pois com base nas informações repassadas pelos integrantes relatando as atividades pelas quais estão responsáveis e o andamento das mesmas, pode-se facilmente identificar o eventual desequilíbrio de carga de trabalho entre os membros do *DevTeam*. Aliado a isso, estando o *DevTeam* alocado no mesmo espaço físico, a utilização de um *Kanban* e/ou *Scrum board* físico não só auxilia na organização, acompanhamento e comunicação sobre o andamento da *Sprint*, como ajuda na motivação e auxilia na identificação da “folga social”. Não existe nada mais motivador e/ou instigante do que ver os seus colegas de Equipe movendo as tarefas do *Sprint Backlog* para a coluna de “fazendo” e na sequência para “feito”, enquanto você “patina” ou “enrola” para terminar uma tarefa simples. Caso isso não seja naturalmente resolvido pelo próprio *DevTeam*, durante as reuniões diárias, o SM deve sugerir uma revisão da quantidade de *Story Points* que cada integrante desenvolveu na próxima *Sprint Retrospective*. Obviamente este assunto deve ser abordado com muita maturidade e respeito, permitindo que cada um exponha suas eventuais justificativas e que o *DevTeam* encontre a solução. As vezes a solução passa por trocar tal integrante que não demonstra o mesmo nível de comprometimento e motivação que os demais.

... as pessoas podem se esconder dentro de um grupo. Podem se entregar à “folga social” e se aproveitar do esforço do grupo, já que as contribuições individuais não podem ser identificadas. As equipes eficazes não correm este risco porque seus membros se mantêm responsáveis tanto no nível individual como no do grupo. Equipes bem-sucedidas fazem com que seus membros sejam responsáveis, individual e conjuntamente, pelos propósitos, metas e abordagem da equipe. Estas equipes deixam claro quais são as responsabilidades individuais e quais são aquelas do grupo como um todo. (ROBBINS, 2005, p. 222)

## 5.8. Liderança Situacional

Conforme apresentado pelo *framework SCRUM*, o SM não exerce ou possui nenhuma função hierárquica perante o *DevTeam*, ainda mais na Empresa “Z”, onde o *DevTeam* é formado por integrantes de vários Departamento, Gerencias e/ou Coordenações. Contudo como especialista nos processos *SCRUM* será natural que todos recorram ao SM para o esclarecimento de dúvidas, para o aconselhamento das melhores alternativas e obviamente para a remoção de qualquer impeditivo que esteja atrapalhando a performance do *DevTeam*. Assim, de uma forma indireta, o SM acabará por exercer algum tipo de liderança perante o *DevTeam*, pois além das funções já apresentadas, o SM tem uma obrigação muito maior do que apenas ser o “guardião” dos processos *SCRUM* e o removedor de impeditivos, o SM tem

por obrigação conduzir, orientar, motivar, liderar e principalmente inspirar os membros do *DevTeam*, para que sempre busquem evoluir e aprimorar seus potenciais, tornando-se profissionais mais maduros e produtivos, sempre em busca do *Performing*. Por outro lado, o SM também deverá demonstrar grande poder de influência junto ao PO e os demais *stakeholders* do Projeto, uma vez que o *DevTeam*, conforme evolua nos estágios de Tuckman, exigirá muito mais apoio dos líderes (gerentes/coordenadores) funcionais e muito menos controle e/ou micro gerenciamento.

Diante deste cenário e das necessidades exigidas do SM, torna-se fundamental o conhecimento e aplicação dos conceitos da Liderança Situacional (HERSEY; BLANCHARD, 2012), conforme citação abaixo:

Por isso, ainda que nossos exemplos possam sugerir uma relação hierárquica, os conceitos apresentados na Liderança Situacional devem ser aplicáveis, quer alguém esteja tentando influenciar o comportamento de um subordinado, do chefe, do colega, de um amigo ou de um parente. (HERSEY; BLANCHARD, 2012, p.187)

A Liderança Situacional apresenta-se na forma de um modelo de atuação onde o líder deve ajustar o seu comportamento conforme as características do liderado perante determinada tarefa/situação. A Liderança Situacional está baseada em uma inter-relação de 3 componentes básicos (HERSEY; BLANCHARD, 2012):

1. Comportamento de Tarefa (direção): Quantidade de orientação e direção que o líder precisa oferecer ao liderado relacionada à tarefa propriamente dita. O líder precisa dizer o que fazer, quando fazê-lo, onde e como, estabelecendo objetivos claros e definindo papéis.
2. Comportamento de Relacionamento (apoio): Quantidade de apoio sócio emocional oferecido pelo líder ao seu liderado, através da comunicação bilateral, transmitindo apoio e encorajamento. Significa ouvir atentamente, apoiar e incentivar os esforços dos liderados.
3. Maturidade: Nível de prontidão, de capacidade e de disposição do liderado em assumir a responsabilidade de dirigir seu próprio comportamento no desempenho de uma tarefa, função ou objetivo.

Sendo que a Maturidade apresenta-se como um estado relacionado a tarefa/situação e ao momento específico, seja no aspecto individual ou do grupo. “Essas variáveis de maturidade devem ser consideradas somente em relação a uma tarefa específica a ser realizada. ” (HERSEY; BLANCHARD, 2012), ou seja, determinado membro do *DevTeam* pode apresentar um nível de maturidade alto para determinada tarefa individual, e logo em seguida apresentar exatamente o oposto, isto é, um nível de maturidade baixo para uma tarefa em equipe. Esta variação do nível de maturidade pode ser constatada tanto no nível individual como no nível de grupo, do *DevTeam* como um todo. Portanto o segredo para a correta aplicação da liderança situacional exige do líder, no caso o SM, uma avaliação constante do nível de maturidade do *DevTeam* para cada tarefa, seja no nível individual, como no da Equipe, pois a o estilo de liderança situacional compreende a dosagem correta de comportamento de tarefa (direção) e comportamento do relacionamento (apoio).

A maturidade dos indivíduos ou do grupo foi classificada por Hersey e Blanchard (2012) em 4 níveis de graduação:

- M1 (baixa): Pessoas que não tem capacidade nem disposição ou são inseguras
- M2 (baixa a moderada): Pessoas que não tem capacidade, mas estão dispostas ou confiantes
- M3 (moderada a alta): Pessoas com capacidade, mas sem disposição ou inseguras
- M4 (alta): Pessoas capazes (competentes) e dispostas (seguras)

Conforme Hersey e Blanchard (2012) definem, não existe uma única forma de influenciar as pessoas, o estilo que líder precisa adotar perante indivíduos ou grupos, depende do nível de maturidade dos mesmos. Estas formas de liderar foram agrupadas em 4 estilos que combinam os comportamentos de tarefa e de relacionamento, da seguinte forma:

- E1 – DETERMINAR: Comportamento de tarefa alto e de relacionamento baixo. Este estilo deve ser aplicado para níveis de maturidade M1, desta forma, as pessoas apresentam-se insegura e sem disposição, muitas vezes por não terem total domínio da tarefa em questão, portanto cabe ao SM orientar e supervisionar todas as atividades emitindo *feedbacks* constantes de forma clara e objetiva, apresentando os objetivos da tarefa, auxiliando na definição de funções e papéis, deixando muito claro o que deve ser feito, como deve ser feito e quando dever ser concluída a

tarefa. Esta situação será mais comum durante as primeiras *Sprints*, enquanto o *DevTeam* ainda estiver absorvendo os processos e conceitos do *SCRUM* e o grupo ainda se encontrar no estágio de *Forming*. Contudo o SM sempre deve ter em mente que precisa evoluir o *DevTeam* para os próximos estágios, portanto sempre deve deixar claro que nestes estágios iniciais estará apenas orientando (determinado) o que, como e quando as atividades devem ocorrer e que gradativamente o *DevTeam* deve assumir integralmente estas funções.

- E2 – PERSUADIR: Comportamento de tarefa e de relacionamento altos. Estilo adotado para níveis de maturidade M2 (entre baixa e moderada), onde os integrantes do *DevTeam* demonstram disposição para as tarefas e demonstram confiança em si, mas ainda não possuem todas as habilidades necessárias. Este nível de maturidade poderá ser identificado após a conclusão da primeira *Sprint* com sucesso, após um ótimo *feedback* do PO. Porém o SM deve estar atendo e manter o *DevTeam* focado nos processos e métodos previstos no *SCRUM*, uma vez que ainda não dominam todos os detalhes do *framework* e desta forma o SM deve continuar persuadindo o *DevTeam* através de um comportamento diretivo devido à falta de capacidade, porém ao mesmo tempo apoiando e reforçando a fim de aumentar a disposição dos integrantes do *DevTeam*.

Chama-se o estilo de persuadir porque a maior parte da direção ainda é dada pelo líder. Mas, mediante explicações e comunicações bilaterais, o líder procura conseguir que os liderados se sintam convencidos psicologicamente a adotarem os comportamentos desejados. (HERSEY; BLANCHARD, 2012, p.190)

- E3 – COMPARTILHAR: Comportamento de relacionamento alto e tarefa baixo. Este estilo deve ser aplicado para indivíduos ou grupos com nível de maturidade M3 (moderada a alta), onde as pessoas já possuem todas as capacidades para a execução das tarefas, mas não apresentam a disposição para executá-las. A falta de disposição pode ser associada a insegurança e/ou a falta de motivação. O SM precisa manter a comunicação bilateral em elevado nível, sempre atendo as *Daily Meetings* e *Sprint Retrospectives*, a fim de identificar os reais motivos responsáveis pela insegurança e/ou falta de motivação. E compartilhar as orientações e/ou sugestões para a solução destes problemas com o *DevTeam*. Este estilo poderá ser necessário após alguns meses de Projeto, após várias *Sprints* entregues, quando o

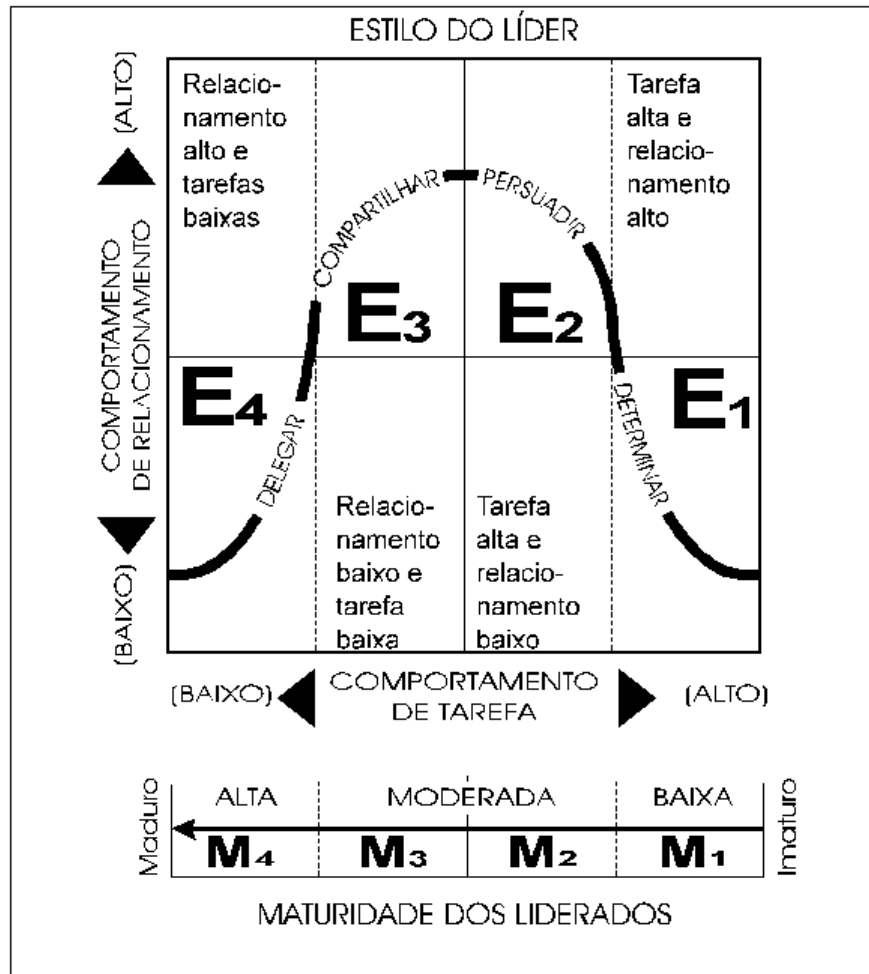
*DevTeam* encontra-se no estágio de *Norming*, mas hesitante para passar para o *Performing*, ou desmotivado por ainda existirem muitas *Sprints* pela frente. Neste caso o SM deve atuar junto com o PO e os gerentes/coordenadores motivando o *DevTeam* e estabelecendo objetivos intermediários consistentes para o Projeto e para o *DevTeam*.

- E4 – DELEGAR: Comportamento de tarefa e relacionamento baixos. Estilo aplicado quando os indivíduos ou grupos apresentarem altos níveis de maturidade (M4), ou seja, possuem todas as capacidades e estão motivadas e dispostas a assumirem total responsabilidade sobre suas tarefas, individuais e/ou no nível da Equipe. Neste estágio o SM atua muito pouco, apenas como um conselheiro, pois o *DevTeam* já encontra-se no *Performing* e tem todas as condições para auto gerenciar suas atividades.

Embora possa ainda ser o líder quem identifica o problema, a responsabilidade de executar os planos cabe a esses liderados maduros, que por sua conta desenvolvem o projeto e decidem como, quando e onde fazer as coisas. (HERSEY; BLANCHARD, 2012, p. 191)

Abaixo encontra-se a figura 7, que representa de forma gráfica um resumo da Liderança Situacional:

Figura 7: Modelo de Liderança Situacional



Fonte: HERSEY, Paul; BLANCHARD, Kenneth H, 2012, p. 189

## 6. CONCLUSÕES

O objetivo deste trabalho foi o de apresentar formas de atuação do *Scrum Master (SM)*, no auxílio e condução da evolução de grupos de projetos de desenvolvimento de softwares habituados com metodologias formais (*Waterfall*) em Equipes Auto Gerenciáveis – *DevTeams* no estágio *Performing* – contextualizadas dentro da Empresa “Z” que teve atuação por mais de 10 anos no Desenvolvimento de Softwares para o mercado financeiro na Cidade de Curitiba – PR.

Com base nesta proposta descreveu-se os aspectos básicos da atuação de grupos de projetos formais (*Waterfall*) de TI da Empresa “Z”, apresentando-se as principais características e ressaltando o principal problema decorrente de uma forma individualizada de condução das atividades de um Projeto *Waterfall*, onde não se identifica uma preocupação genuína com a entrega final, mas sim cada indivíduo ou pequeno grupo demonstra-se preocupado apenas em realizar as suas entregas individuais, ou seja, não existe integração e nem sinergia entre os participantes do projeto.

Com alternativa para o problema dos Projetos *Waterfall* de TI da Empresa “Z” foi apresentado o *framework SCRUM* de desenvolvimento de projetos ágeis, concebido por Ken Schwaber e Jeff Sutherland em 1993, (SUTHERLAND, 2014), que baseia-se em 3 pilares: transparência, inspeção e adaptação. Descreveram-se seus processos, suas ferramentas e cerimônias, sendo que para que o *SCRUM* funcione adequadamente e produza os resultados esperados é essencial que a Equipe de Desenvolvimento (*DevTeam*) atinja níveis razoáveis de auto gerenciamento, deixando de ser apenas um grupo de pessoas, mas sim, tornando-se uma equipe auto gerenciável. Também apresentou-se a figura do *Scrum Master (SM)* que é o agente de mudança e/ou implementação do *SCRUM* na Empresa “Z”, sendo o detentor, o “guardião”, a pessoa que mais conhece os conceitos, processos e cerimônias do *SCRUM*.

Para que o SM tenha condições de conduzir o *DevTeam* de um grupo de pessoas acostumadas com projetos *Waterfall* para um Equipe Auto Gerenciada, tornou-se essencial apresentar as principais características e conceitos deste tipo de Equipes e como as mesmas diferenciam-se de grupos de trabalho comuns. As características e necessidades específicas para a correta composição de um *DevTeam* para projetos de TI de Empresa “Z” também foi abordada, pois diferentemente dos que ocorre nos projetos *Waterfall*, o *SCRUM* recomenda fortemente que todos estejam co-locados e totalmente dedicados ao Projeto em questão.



E para auxiliar o SM na transformação de grupos de projetos formais em equipes auto gerenciáveis, discorreu-se sobre a Teoria de formação de Grupos de Tuckman (1965). Teoria esta que serviu de pano de fundo para este trabalho, pois descreve os estágios de evolução de um grupo, desde a sua formação (*Norming*) até uma equipe auto gerenciável (*Performing*). Detalhou-se questões relacionadas a conformidade que pode ocorrer entre grupos e os possíveis conflitos quando tenta-se combater tal conformidade, apresentou-se o *Brainstorm* com alternativa para a solução de tais conflitos e o incentivo da Criatividade e Inovação, necessárias e presentes em Equipes Auto Gerenciáveis.

Também discorreu-se sobre a importância da Comunicação entre os integrantes do *DevTeam*, que está diretamente relacionada com o tamanho da Equipe, que deve ser de no máximo 9 integrantes e não menos do que 5. E conforme Moscovici (2007) é somente através da comunicação franca, honesta e objetiva é que um grupo tem condições de se tornar em um Equipe Auto Gerenciada. Reforçando a importância da Comunicação, detalhou-se os metamodelos da PNL que consistem de ferramentas poderosas para a reconexão das deleções, distorções e generalizações à experiência que as gerou.

Prosseguindo nas competências necessárias para integrantes de equipes auto gerenciáveis, apresentou-se o conceito de Inteligência Emocional e os seus componentes básicos que precisam ser conhecidos e aprimorados, não só pelo SM, mas pelos integrantes do *DevTeam*, a fim de que possam relacionar-se de uma forma transparente e respeitosa. Profundamente conectados a Inteligência Emocional, estão os conceitos de Assertividade e de *Feedback*, essenciais para uma comunicação clara, objetiva e principalmente honesta, conceitos estes mais uma vez essenciais para o bom funcionamento do *SCRUM* e do *DevTeam*.

Motivação e produtividade, foram apontados como cruciais para o sucesso do *DevTeam* em sua jornada até tornarem-se uma Equipe Auto Gerenciada. Sem motivação os integrantes do *DevTeam* não terão condições de manter o ritmo das entregas constantes previstas pelo *SCRUM*. E para evitar o que Robbins (2005) chamou de “folga social” a produtividade individual precisa ser constantemente monitorada pelo SM. A Delegação e o *Empowerment* também foram apresentados como elementos essenciais para que o *DevTeam* sintasse motivado, respeitado, valorizado e capaz de resolver seus problemas internos sem a necessidade de um Gerente/Coordenador se preocupar com o micro gerenciamento da equipe.

Por fim, detalhou-se o modelo de Liderança Situacional, onde o SM, mesmo sem exercer cargo hierárquico, atuará como um líder natural, pois tem por missão influenciar o comportamento dos membros do *DevTeam* rumo ao *Performing*, auxiliando de uma forma mais relacionada à tarefa, em um primeiro momento, e gradativamente, através da percepção do aumento da maturidade dos integrantes do *DevTeam*, poderá reduzir seu apoio, deixando o *DevTeam* agir naturalmente na resolução de seus problemas internos, aplicando todos os conceitos das equipes auto gerenciadas.

Isto posto, em contraponto a vasta literatura técnica relacionada ao *SCRUM* e ao papel do SM, espera-se que, através deste trabalho, consiga-se ampliar o panorama do SM, permitindo-lhe entender melhor o funcionamento de um grupo de pessoas que precisa ser conduzido para tornar-se uma equipe auto gerenciável, embasado por uma perspectiva mais humana, reforçada pelos conceitos e teorias abordadas neste trabalho. Desta forma, o SM possuirá melhores condições de atingir seu objetivo principal que é o de construir uma cultura *SCRUM* em toda a comunidade de TI da Empresa “Z”, tornando-se, por fim, desnecessária a existência do seu papel, atingindo o “Nirvana”, conforme caracterizado na figura 8, abaixo.

Figura 8: A Jornada do Scrum Master



## 7. POSSÍVEIS DESDOBRAMENTOS

Quando este trabalho foi pensado, pretendia-se abordar a grande maioria das matérias relacionadas a grupos e equipes que fizeram parte da grade curricular do curso de MBA de Desenvolvimento Humano de Gestores da FGV, contudo, conforme os estudos foram sendo aprofundados constatou-se que tal missão seria praticamente impossível para o contexto da Empresa “Z”, por exemplo, pode-se citar o Psicodrama e/ou Sociodrama nas Organizações, concebido por Jacob Levy Moreno (1889-1974) (DRUMMOND; SOUZA, 2008) que apresenta uma concepção muito mais profunda relacionada a psicologia de indivíduos, grupos e equipes, com todas as suas sugestões de dinâmicas e a Teoria da Matriz da Identidade Pessoal. Para que tais conceitos sejam produtivos e aplicáveis pelo SM faz-se necessária uma contextualização muito mais abrangente da psicologia, passando por treinamentos específicos, caso contrário o impacto no *DevTeam* poderá ser muito mais prejudicial do que construtivo, uma vez que envolve sentimentos e emoções muito mais profundas.

A PNL também pode ser citada como uma outra possibilidade de aprofundamento, pois neste trabalho abordou-se apenas seus conceitos de metamodelos, deixando espaço para estudos futuros de como o SM pode aplicar os conceitos dos Pressupostos da PNL, do Rapport<sup>14</sup>, Aprendizagem, Posições Perceptuais, Âncoras<sup>15</sup>, TOTS<sup>16</sup> e o Modelo de Milton Erickson, enfim a PNL é uma ciência muito abrangente com uma vasta quantidade de ferramentas que podem ser estudadas para serem aplicadas na comunicação, na influência, na transformação do comportamento dos indivíduos que queiram buscar qualquer tipo de evolução.

Durante os estudos das diversas teorias motivacionais, identificou-se uma possível aplicação direta dos conceitos e processos do *SCRUM*, com as mesmas, portanto uma análise mais detalhada das Teorias de Motivação de Abraham MASLOW e sua Hierarquia das Necessidades, da Teoria da Higiene de Frederick HERZBERG, de Clayton Alderferou e sua Teoria ERG ou até mesmo Douglas MCGREGOR, com sua Teoria X e Y podem ser

---

<sup>14</sup> Rapport é a qualidade de relacionamento que resulta em confiança e responsividade com você mesmo e com os outros, é um pila da PNL.

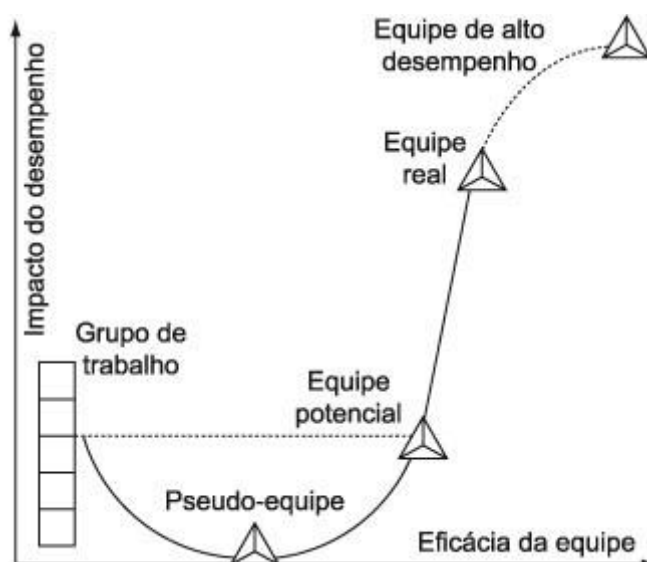
<sup>15</sup> Âncoras são gatilhos visuais, auditivos ou cinestésicos que se tornam associados a uma resposta ou a um estado específicos.

<sup>16</sup> TOTS é um padrão básico de estratégia da PNL. Significa Teste – Operação – Teste – Saída.

relacionadas aos estágios do desenvolvimento de grupos de Tuckman e aos processos e cerimônias do *SCRUM*, auxiliando o SM no incremento gradativo da Motivação dos integrantes do *DevTeam*, que trata-se de um componente essencial para Equipes Auto Gerenciáveis.

Complementar a Teoria de Grupos de Tuckman, pode-se relacionar o desenvolvimento de um Equipe de Alta Performance e/ou Auto Gerenciada com a curva de desempenho apresentada por Katzenbach e Smith (1994) e a partir desta perspectiva abordar outras sugestões de atuação do SM na dura jornada de condução de um grupo de pessoas acostumadas com Projetos *Waterfall* até tornarem-se um Equipe de Alta Performance (Auto Gerenciada). Tais estágios encontram-se detalhados na figura 9, abaixo:

**Figura 9: Curva de desempenho de Katzenbach e Smith**



Fonte: KATZENBACH e SMITH (1994, p. 85)

## 8. REFERÊNCIAS BIBLIOGRÁFICAS

BECK K., BEEDLE M., BENNEKUM A., COCKBURN A., et al, **Agile Manifesto** em <<http://agilemanifesto.org/iso/ptbr/manifesto.html>>. Acesso em: 12 de dezembro de 2016

BURN DOWN CHART. Em **WIKIPÉDIA**, a enciclopédia livre. Flórida: Wikimedia Foundation, 2016. Disponível em: <[https://en.wikipedia.org/w/index.php?title=Burn\\_down\\_chart&oldid=748897042](https://en.wikipedia.org/w/index.php?title=Burn_down_chart&oldid=748897042)>. Acesso em 11 de novembro de 2016

CARDOSO, Margot; MARTINS, Vera L. F.; DUTRA, Dálcio B. **ASSERTIVIDADE**. Revista Vencer!, Ano IV, nº 38, Novembro 2002

CHATFIELD, Mark M. **Self-Directed and Self-Managed Teams**. Free Management Library. 30 abr. 2009. Disponível em: <<http://irism.com/selfteam.htm>>. Acesso em: 28 set. 2009.

CISV International. **Five Stages of Group Development**. Disponível em <<http://www.cisv.org/EasysiteWeb/getresource.axd?AssetID=2182&type=Full&servicetype=Attachment>>. Acesso em 13 de dezembro de 2016

COACHING. Em: **SIGNIFICADOS**. Disponível em <<https://www.significados.com.br/coaching/>>. Acesso em 11 de dezembro de 2016

CONTOS ÁRABES. **Maneira de dizer as coisas**. Disponível em <<http://almanaquenilomoraes.blogspot.com.br/2014/08/contos-arabes.html>>. Acesso em 14 de dezembro de 2016

Creative Education Foundation. **Creative Problem Solving Process**. Disponível em <<http://www.creativeeducationfoundation.org/creative-problem-solving/>>. Acesso em 14 de dezembro de 2016.

CSIKSZENTMIHALYI, Mihaly. **Good Business. Leadership, Flow and Making of Meaning**. USA: Penguin, 2004

DRUMMOND, Joceli e SOUZA, Andréa Cláudia de, **Sociodrama nas organizações**. São Paulo: Ágora, 2008

EMPIRISMO. Em: **WIKIPÉDIA**, a enciclopédia livre. Flórida: Wikimedia Foundation, 2016. Disponível em:

<<https://pt.wikipedia.org/w/index.php?title=Empirismo&oldid=47311862>>. Acesso em: 25 novembro 2016.

EMPOWERMENT. Em: **SIGNIFICADOS**. Disponível em <<https://www.significados.com.br/empowerment/>>. Acesso em 13 de dezembro de 2016

GOLDBARG, Marco A. **Times: ferramenta eficaz para a qualidade total**. São Paulo: Makron, 1995.

GOLEMAN, Daniel. **Inteligência Emocional: A teoria revolucionária que define o que é ser inteligente**. Rio de Janeiro: Objetiva, 2001

HERSEY, Paul; BLANCHARD, Kenneth H. **Psicologia para administradores: A teoria e as técnicas da liderança situacional**. São Paulo: E.P.U., 2012

Kanban Tool ® by Shore Labs. **SCRUM BOARD** em <<http://kanbantool.com/scrum-board-online>>. Acesso em: 12 de dezembro de 2016

KANBAN. Em: **SIGNIFICADOS**. Disponível em <<https://www.significados.com.br/kanban/>>. Acesso em 11 de dezembro de 2016

KATZENBACH, J. R. & SMITH, D. K. **Equipes de Alta Performance: conceitos, princípios e técnicas para potencializar o desempenho das equipes**. Rio de Janeiro: Editora Campus, 2001.

KATZENBACH, John R e SMITH, Douglas K. **A força e o poder das equipes**, São Paulo: Makron, 1994.

KNIBERG, Henrik. **SCRUM AND XP FROM THE TRENCHES**. USA, C4Media: 2015

MEDINILLA, Angel. **Developing Scrum Masters**. Agile Tour, 2013. Disponível em <<https://www.infoq.com/presentations/hire-scrum-master>>. Acessado em 21 de dezembro de 2016

MITTAL, Nitin. **Self-Organizing Teams: What and How**. Disponível em <<https://www.scrumalliance.org/community/articles/2013/january/self-organizing-teams-what-and-how>>. Acessado em 09 de dezembro de 2016

MODUS OPERANDI. Em: **SIGNIFICADOS**. Disponível em <<https://www.significados.com.br/modus-operandi/>>. Acesso em 11 de dezembro de 2016

MORETI, Rodrigo. **Contratado BLOG**. Disponível em <http://blog.contratado.me/2017/01/16/o-que-e-scrum-um-guia-rapido-e-bacanudo/>. Acessado em 17 de Março de 2017.

MOSCOVICI, Fela. **Equipes dão certo: a multiplicação do talento humano**. – 11. Edição – Rio de Janeiro: José Olympio, 2007.

NASCIMENTO, Kleber. **Comunicação interpessoal eficaz: verdade & amor**. Rio de Janeiro: Incisa, 1977

O'CONNOR, Joseph. **Manual de programação neurolinguística: PNL: um guia prático para alcançar os resultados que você quer**. Rio de Janeiro: Qualitymark Editora, 2015

OSBORN, A.F. **Applied imagination: Principles and procedures of creative problem-solving (3rd. rev. ed.)**. New York: Charles Scribner's. 1979. [Orig. ed. 1953.]

PENSAR FORA DA CAIXA. Em: **Ceschini Consultoria**. Disponível em <http://www.ceschini.com.br/2013/10/pensarforadacaixa/>. Acessado em 14 de dezembro de 2016

PRESSMAN, Roger S. **Engenharia de Software – Uma Abordagem Profissional**. Porto Alegre: AMGH, 2011

Project Management Institute, Inc. **Guia PMBOK – Quinta Edição**. USA: PMI, 2013

PSIQUE HUMANA. Em: **SIGNIFICADOS**. Disponível em <https://www.significados.com.br/psique/>. Acesso em 11 de dezembro de 2016

ROBBINS, Stephen P. **Administração, mudanças e perspectivas**. São Paulo: Saraiva, 2000.

ROBBINS, Stephen P. **COMPORTAMENTO ORGANIZACIONAL**. Tradução técnica Reynaldo Marcondes – 11.edição – São Paulo: Pearson Prentice Hall, 2005

SCHWABER, Ken e SUTHERLAND, Jeff. **Guia do Scrum**. USA: Scrum.org: 2011

SCRUM ALLIANCE. Disponível em <https://www.scrumalliance.org/why-scrum>. Acesso em 11 de dezembro de 2016

SUTHERLAND, Jeff. **SCRUM – A Arte de Fazer o Dobro do Trabalho na Metade do Tempo**. Tradução de Natalie Gerhardt. São Paulo: LeYa, 2014

TAYLOR, Frederick W. **Princípios de administração científica**. Tradução de Arlindo Vieira Ramos. São Paulo: Atlas, 1990

TAYLORISMO. Em: **WIKIPÉDIA**, a enciclopédia livre. Flórida: Wikimedia Foundation, 2016. Disponível em: <<https://pt.wikipedia.org/w/index.php?title=Taylorismo&oldid=47391345>>. Acesso em: 5 de dezembro 2016.

T-GROUP. Em **WIKIPEDIA**, The Free Encyclopedia. Flórida: Wikimedia Foundation, 2016. Disponível em: <<https://en.wikipedia.org/w/index.php?title=T-groups&oldid=752791483>>. Acesso em 12 de dezembro 2016.

The Standish Group International, Incorporated. **The CHAOS Manifesto**. USA, 1995

The Standish Group International, Incorporated. **The CHAOS Manifesto**. USA, 2013

TUCKMAN, B. W. e JENSEN, M. A. **Stages of small-group development revisited**. USA, Group Org. Studies 2, p. 419-27, 1977

TUCKMAN, Bruce W. **Developmental Sequence in Small Groups**. USA, American Psychological Association: Psychological bulletin, v.63, n6, p.384-399, 1965