

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
CÂMPUS CORNÉLIO PROCÓPIO  
DIRETORIA DE PESQUISA E PÓS-GRADUAÇÃO  
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

VANESSA FARIA DE SOUZA

**PROCESSO PARA MONITORAMENTO DE PROJETOS  
DISTRIBUÍDOS DE SOFTWARE**

DISSERTAÇÃO – MESTRADO

CORNÉLIO PROCÓPIO  
2016

VANESSA FARIA DE SOUZA

**PROCESSO PARA MONITORAMENTO DE PROJETOS  
DISTRIBUÍDOS DE SOFTWARE**

Dissertação apresentada ao Programa de Pós-Graduação em Informática da Universidade Tecnológica Federal do Paraná – UTFPR como requisito parcial para a obtenção do título de “Mestre em Informática”.

Orientador: Prof. Dr. Alexandre L'Erario

CORNÉLIO PROCÓPIO

2016



**Processo para o Monitoramento de Projetos Distribuídos de Software Nº XX:**

## **“PROCESSO PARA MONITORAMENTO DE PROJETOS DISTRIBUÍDOS DE SOFTWARE”.**

por

### **VANESSA FARIA DE SOUZA**

Orientador: **Prof. Dr. Alexandre L'Erario**

Esta dissertação foi apresentada como requisito parcial à obtenção do grau de MESTRE EM INFORMÁTICA – Área de Concentração: Computação Aplicada, pelo Programa de Pós-Graduação em Informática – PPGI – da Universidade Tecnológica Federal do Paraná – UTFPR – Câmpus Cornélio Procópio, às 16h30 do dia 15 de março de 2016. O trabalho foi aprovado pela Banca Examinadora, composta pelos professores:

---

Prof. Dr. Alexandre L'Erario  
UTFPR – Cornélio Procópio

---

Prof. Dr. Érica Ferreira de Souza  
UTFPR – Cornélio Procópio

---

Prof. Dr. João Coelho Neto  
UENP – Cornélio Procópio

Visto da coordenação:

---

Andre Takeshi Endo  
Coordenador do Programa de Pós-Graduação em Informática  
UTFPR Câmpus Cornélio Procópio

A Folha de Aprovação assinada encontra-se na Coordenação do Programa.

Dedico este trabalho à minha família. Em especial ao meus pais, João Batista de Souza e Inez Faria de Souza; meu marido Leandro Aparecido de Aguiar, e minha irmã, Mayara Faria de Souza. Vocês são meu tudo; sem vocês nada teria sentido.

## **AGRADECIMENTOS**

Agradeço a meu orientador, Prof. Dr. Alexandre L'Erario, pela sabedoria com a qual me guiou nesta trajetória, e por todos os ensinamentos transmitidos desde o início do curso; bem como a paciência dispensada, ao esclarecimentos e também com os prazos.

Aos meus amigos de jornada, Gabriel Canhadas Genvigir, José Antônio Gonçalves e Alessandro Duarte; aos professores e amigos Elias Canhadas Genvigir e José Augusto Fabri; e, em especial, a Solange Fávero de Lima Medeiros pela ajuda e companheirismo nos momentos difíceis.

Ao antigo Secretário do Curso, Thiago Fernando Mendes, pela cooperação.

Gostaria de deixar registrado também, o meu reconhecimento a Deus que sempre esteve comigo e, à minha família, pois acredito que sem o apoio deles seria muito difícil vencer esse desafio.

Enfim, a todos os, que por algum motivo, contribuíram para a realização desta pesquisa.

*“Ciência da Computação está tão relacionada aos computadores quanto a Astronomia aos telescópios, Biologia aos microscópios, ou Química aos tubos de ensaio. A Ciência não estuda ferramentas. Ela estuda como nós as utilizamos, e o que descobrimos com elas”*

*Edsger Dijkstra*

## RESUMO

SOUZA, Vanessa Faria. **Processo para Monitoramento de Projetos Distribuídos de Software**. 2015.141 f. Dissertação de Mestrado em – Engenharia de Software. Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2015.

Contexto: É progressivo e significativo o número de empresas que estão distribuindo seus processos de produção de software ao redor do mundo visando ganhos de produtividade, redução de custos e melhorias na qualidade. Por isso, o Desenvolvimento Distribuído de Software (DDS) tem suscitado grande número de pesquisas na área de engenharia de software nos últimos anos e, os engenheiros, em reconhecimento à grande influência desta forma de trabalho, estão em busca de modelos e processos que facilitem produção de software, com equipes geograficamente dispersas. Além dos engenheiros, gerentes e executivos têm enfrentado desafios e dificuldades em diferentes níveis, envolvendo fatores gerenciais, em especial quanto ao monitoramento de projetos. O monitoramento de projetos de software é determinante para que se obtenha êxito em seu desenvolvimento. Quanto ao desenvolvimento de software, uma técnica que conquistou destaque é o Kanban, empregado junto à metodologias ágeis, neste sentido, é possível ressaltar também o uso de *Business Process Model and Notation* (BPMN) em modelos que presam pelo gerenciamento e monitoramento de projetos.

Objetivo: Nesta perspectiva, o objetivo desta dissertação de mestrado é comprovar a seguinte hipótese: “Um projeto distribuído de software pode ter o cronograma de suas atividades monitoradas, por meio de um processo que aplique a técnica do Kanban e a modelagem BPMN”. Para tal, foi implementado um processo que aplica a técnica mencionada, junto à notação, como instrumento para a validação da hipótese.

Métodos e Procedimentos: Com o propósito de comprovar a referida hipótese, foi utilizado o método experimental, o qual contou com 82 participantes, que trabalharam distribuídamente.

Resultados: Após a realização do experimento, foram sistematizadas as informações; estes revelaram que a hipótese supracitada é válida, por conseguinte o processo proposto, que aplica as técnicas do Kanban e a modelagem BPMN, é efetivo no monitoramento do cronograma das atividades em um projeto distribuído. Espera-se que, com este resultado, a técnica Kanban, pouco explorada no DDS seja vista como vantajosa, por gerentes de projetos distribuídos, deste modo também melhorar o monitoramento das atividades em projetos desta natureza, e ampliar os materiais de pesquisa na área.

**Palavras-chave:** Processo, Processo de Software, Monitoramento, Controle, Desenvolvimento Distribuído de Software, BPMN e Kanban.

## ABSTRACT

SOUZA, Vanessa Faria. **Process for Monitoring of Software Distributed Projects.** 2015. 141 f. Dissertation in - Software Engineering. Federal Technological University of Paraná. Cornélio Procópio, 2015.

**Context:** It is progressive and significant number of companies that are distributing their software production processes around the world aiming ga-tasting productivity, reduced costs and improved quality. Therefore, the De-velopment Distributed Software (DDS) has attracted large number of pes-quisas in software engineering in recent years, and engineers, in recognition of the great influence of this form of work, they are looking for models and processes that facilitate production software, with geographically-dispersed teams mind. In addition to engineers, managers and executives have faced challenges and difficulties, at different levels, involving managerial factors in spe cial as the project monitoring. Monitoring of software projects is crucial in order to obtain success in its development. As for the development of software, a technique that gained prominence is the Kanban in-preached by the agile methodologies, in this sense, it is also possible to emphasize the use of Business Process Model and Notation (BPMN) models that presam for managing and project monitoring.

**Objective:** In this perspective, the objective of this master thesis is to-prove the following hypothesis: "A software distributed design can have cronogra-ma of their activities monitored through a process to apply the technique of Kanban and modeling BPMN. " To this end, it implemented a process that applies the aforementioned technique, with the notation as a tool to validate the hypothesis.

**Methods and Procedures:** In order to prove that mortgage is, we used the experimental method, which was attended by 82 participants, who tra-balharam distribuidamente.

**Results:** After the experiment, were systematized in-training; These revealed that the above hypothesis is valid, therefore the proposed process, which applies the techniques of Kanban and BPMN modeling is eff-tive in monitoring the schedule of activities in a distributed project. Es-pear that, with this result, Kanban technique, little explored in DDS is seen as advantageous for project managers distributed thus also me lhorar monitoring activities in projects of this nature, and expand the ma- terials research in the area.

**Keywords:** Process, Software Process, Monitoring, Control, Development Distributed Software, BPMN and Kanban.



## LISTA DE GRÁFICOS

GRÁFICO 01 – FREQUÊNCIA DE ATIVIDADES DE CONTROLE EM PROJETOS DE EMPRESA BRASILEIRAS.....	59
GRÁFICO 02 – COMPARAÇÃO ENTRE AS REPETIÇÕES, O GRUPO DE CONTROLE E A MÉTRICA.....	116
GRÁFICO 03 – COMPARAÇÃO ENTRE MÉDIA DAS REPETIÇÕES, O GRUPO DE CONTROLE E A MÉTRICA.....	118
GRÁFICO 04 – RESPOSTAS DO QUESTIONÁRIO PÓS EXPERIMENTO.....	120
GRÁFICO 05 – RESPOSTAS DO QUESTIONÁRIO PÓS EXPERIMENTO DO GRUPO DE CONTROLE.....	122

## LISTA DE TABELAS

TABELA 01 – CÁLCULO DA MÉTRICA <i>IDEAL DAY</i> ANALISTA 1.....	106
TABELA 02 – CÁLCULO DA MÉTRICA <i>IDEAL DAY</i> ANALISTA 2.....	106
TABELA 03 – MÉDIAS DAS MÉTRICAS AFERIDAS.....	107
TABELA 04 – DADOS COLETADOS.....	110
TABELA 05 – COMPARAÇÃO ENTRE A REPETIÇÃO 1 E <i>IDEAL DAY</i> .....	112
TABELA 06 – COMPARAÇÃO ENTRE A REPETIÇÃO 2 E <i>IDEAL DAY</i> .....	113
TABELA 07 – COMPARAÇÃO ENTRE A REPETIÇÃO 3 E <i>IDEAL DAY</i> .....	114
TABELA 08 – COMPARAÇÃO ENTRE O GRUPO DE CONTROLE E A <i>IDEAL DAY</i> .....	115
TABELA 09 – COMPARAÇÃO ENTRE O GRUPO DE CONTROLE, A MÉDIA DAS REPETIÇÕES E A MÉTRICA.....	117
TABELA 10 – RESPOSTAS DO QUESTIONÁRIO PÓS EXPERIMENTO.....	120
TABELA 11 – RESPOSTAS DO QUESTIONÁRIO PÓS EXPERIMENTO DO GRUPO DE CONTROLE.....	121

## LISTA DE QUADROS

QUADRO 01 – DESAFIOS DO DDS.....	49
QUADRO 02 – ELEMENTOS DO MONITORAMENTO E CONTROLE.....	65
QUADRO 03 – COMPARATIVO DOS TRABALHOS RELACIONADOS.....	74
QUADRO 04 – <i>GOAL QUESTION METRICS</i> .....	87
QUADRO 05 – VARIÁVEIS INDEPENDENTES E DEPENDENTES.....	90
QUADRO 06 – CARACTERIZAÇÃO DA AMOSTRA.....	100
QUADRO 07 – SUBPROCESSO DE EXECUÇÃO DO PRODUTO.....	103

## LISTA DE FIGURAS

FIGURA 01 – HIERARQUIA DO PROCESSO.....	26
FIGURA 02 – EXEMPLO DE KANBAN.....	32
FIGURA 03 – OBJETOS DE FLUXO BÁSICO DA BPMN.....	38
FIGURA 04 – OBJETOS DE CONEXÃO DA BPMN.....	39
FIGURA 05 – ARTEFATOS BÁSICOS DA BPMN.....	39
FIGURA 06 – ELEMENTOS DA RAIA DA BPMN.....	40
FIGURA 07 – EXEMPLO DE PROCESSO EM BPMN.....	40
FIGURA 08 – PRODUÇÃO DE SOFTWARE CENTRALIZADO.....	43
FIGURA 09 – PRODUÇÃO DISTRIBUÍDO DE SOFTWARE.....	43
FIGURA 10 – PRINCIPAIS RAZÕES QUE LEVAM AO DDS.....	44
FIGURA 11 – RELAÇÃO ENTRE AS EMPRESAS E DISTÂNCIA FÍSICA.....	47
FIGURA 12 – GRUPOS DE PROCESSOS DE GERENCIAMENTO DE PROJETOS .....	58
FIGURA 13 – CICLO MONITORAMENTO.....	60
FIGURA 14 – FLUXO DE ATIVIDADES DE MONITORAMENTO E CONTROLE.....	61
FIGURA 15 – GRUPO DE PROCESSO DE MONITORAMENTO E CONTROLE.....	63
FIGURA 16 – PROCESSO PARA REALIZAÇÃO DE UM MAPEAMENTO SISTEMÁTICO.....	69
FIGURA 17 – PROCESSO PARA MONITORAMENTO EM PROJETOS DISTRIBUÍDOS.....	75
FIGURA 18 – AÇÕES DO PROCESSO.....	81
FIGURA 19 – CARACTERÍSTICAS DO MONITORAMENTO.....	82
FIGURA 20 – PROCESSO EXPERIMENTAL.....	87
FIGURA 21 – SUBPROCESSO DE EXECUÇÃO EM BPMN.....	104

## LISTA DE SIGLAS

AD	<i>Activity Diagram</i>
BPD	<i>Business Process Diagrams</i>
BPDM	<i>Business Process Definition Metamodel</i>
BPM	<i>Business Process Management</i>
BPMI	<i>Business Process Modeling Initiative</i>
BPMN	<i>Business Process Modeling Notation</i>
CMMI	<i>Capability Maturity Model Integration</i>
CVP	Ciclo de Vida do Produto
DDS	Desenvolvimento Distribuído de Software
EDEF 3	<i>Integrated DEFinition Method</i>
EPC	<i>Event Driven Process Chain</i>
EVM	<i>Earned Value Management</i>
FCS	Fatores Críticos de Sucesso
GQM	<i>Goal Question Metric</i>
ISO	<i>International Organization for Standardization</i>
JIT	<i>Just in Time</i>
MCGP	Modelo de Competências em Gerenciamento de Projetos
MPE	Micro e Pequenas Empresas
MPSBR	Melhoria de Processos do Software Brasileiro
OMG	<i>Object Management Group</i>
PMBOK	<i>Project Management Body of Knowledge</i>
PMI	<i>Project Management Institute</i>
RAD	<i>Role Activity Diagram</i>
RUP	<i>Rational Unified Process</i>
UML	<i>Unified Modeling Language</i>
WIP	<i>Work in Progress</i>
PDCA	<i>Plan-Do-Check-Act</i>

## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	15
1.1 MOTIVAÇÃO .....	18
1.2 OBJETIVOS .....	20
1.3 ORGANIZAÇÃO DO TEXTO.....	21
<b>2 BASE TEÓRICA</b> .....	23
2.1 PROCESSOS DE PRODUÇÃO E PRODUÇÃO DE SOFTWARE .....	23
2.1.1 Hierarquia do Processo .....	25
2.1.2 Processo de Software .....	28
2.1.3 <i>Kanban</i> .....	30
2.1.4 <i>Business Process Management</i> (BPM).....	33
2.1.5 Modelagem de Processo.....	34
2.1.6 Notações para Representação de Processos de Negócio .....	36
2.1.6.1 <i>Business Process Model and Notation</i> (BPMN) .....	37
2.2 DESENVOLVIMENTO DISTRIBUÍDO DE SOFTWARE .....	41
2.2.1 Modelos de Negócios em DDS .....	46
2.3 DESAFIOS ENFRENTADOS PELAS EQUIPES DISTRIBUÍDAS .....	48
2.3.1 As diferenças culturais .....	49
2.3.1 Dispersão geográfica.....	51
2.3.2 Coordenação e controle .....	52
2.3.4 Comunicação .....	53
2.3.4 Espírito de equipe .....	55
2.4 MONITORAMENTO E CONTROLE .....	56
2.4.1 Elementos Monitorados e Controlados.....	62
2.4.2 Processo de Monitoramento de Projetos Distribuídos de Software Preliminar.....	67
2.5 CONCLUSÕES A RESPEITO DA BASE TEÓRICA .....	69
<b>3 TRABALHOS RELACIONADOS</b> .....	71
<b>4 PROCESSO DE MONITORAMENTO PARA PROJETOS DISTRIBUÍDOS</b> .....	75
4.1 SUBPROCESSOS .....	77
4.2 ELEMENTOS DO PROCESSO.....	79
4.3 ATIVIDADES DO PROCESSO .....	80
<b>5 MÉTODO E PROCEDIMENTOS</b> .....	84
5.1 PREMISSAS .....	84

5.2 O MÉTODO DE PESQUISA.....	85
5.3 DEFINIÇÃO DO ESCOPO .....	86
5.4 PLANEJAMENTO .....	88
5.4.1 Seleção do Contexto .....	89
5.4.2 Formulação das Hipóteses.....	89
5.4.3 Seleção das Variáveis e Métricas .....	90
5.4.4 Seleção dos Sujeitos.....	93
5.4.5 Seleção do <i>Design</i> .....	93
5.4.6 Instrumentalização .....	95
5.4.7 Avaliação da Validade .....	96
5.5 CONSIDERAÇÕES A RESPEITO DO MÉTODO E DOS PROCEDIMENTOS...97	
6.1 PREPARAÇÃO .....	98
6.2 EXECUÇÃO .....	107
6.3 VALIDAÇÃO DOS DADOS .....	109
<b>7 ANÁLISE E INTERPRETAÇÃO DOS RESULTADOS .....</b>	<b>111</b>
7.1 ANÁLISE ESTATÍSTICA E DESCRITIVA .....	111
7.2 PROVA DAS HIPÓTESES.....	118
7.3 RESULTADOS DO QUESTIONÁRIO PÓS EXPERIMENTO.....	119
<b>8 CONCLUSÕES .....</b>	<b>124</b>
8.1 CONTRIBUIÇÕES DA PESQUISA .....	125
8.2 LIMITAÇÕES DA PESQUISA .....	127
8.3 PESQUISAS FUTURAS.....	128
<b>REFERÊNCIAS.....</b>	<b>129</b>
<b>APÊNDICE 1 .....</b>	<b>137</b>
<b>APÊNDICE 2.....</b>	<b>139</b>

## 1 INTRODUÇÃO

Um processo de produção de software bem definido, estruturado e alinhado com os intuítos da organização e do cliente é algo imprescindível nas empresas produtoras de software em todo o mundo. Porém, apenas a adoção de um bom processo de software não garante o sucesso do projeto; acima de tudo se faz necessário planejar e monitorar todas as atividades envolvidas durante o ciclo de vida do sistema (SCHARFF, 2011).

A opção pelo monitoramento no processo de software possibilita aumento na velocidade do desenvolvimento e da qualidade, facilita o acompanhamento, minimiza os riscos e melhora a relação entre os colaboradores e com o cliente. Em contrapartida, a não opção pode acarretar consequências negativas, como: baixa produtividade, retrabalho, insatisfação do cliente e realização de atividades desnecessárias (SCHARFF, 2011).

De acordo com os relatórios do *Standish Group* (2014), mais de 80% dos projetos são finalizados sem sucesso por estarem acima do orçamento, do prazo, não atendendo aos requisitos, ou ainda, pela combinação destes fatores. Além disso, 30% dos projetos de software são mal executados, ou cancelados antes da sua conclusão. Grande parte destas falhas nos projetos de software é resultado da não utilização de uma metodologia para monitoramento efetivos (STANDISH GROUP, 2014).

No Desenvolvimento Distribuído de Software (DDS), os cuidados com o monitoramento são ainda mais fundamentais, pois o produto é desenvolvido por equipes remotamente distantes. Segundo L´Erario (2009) o DDS ocorre quando várias equipes colaboram, ou cooperam, para desenvolver um produto, ou parte dele, estando estas em locais diversos.

A dispersão local e temporal das equipes dificulta o trabalho realizado por elas, desta forma métodos aprimorados para monitoramento dos projetos de software neste ambiente tornam-se cada vez mais necessários. São vários motivos que conduzem as organizações a se engajarem ao DDS, dentre eles destacam-se: o custo de produção, a proximidade com clientes e a disponibilidade de mão de obra (SCHARFF, 2011).



Segundo Herbsleb (2007) não é incomum para um projeto de software de grande porte ter equipes em mais de um local, muitas vezes em mais de um continente. De acordo com o autor, muitos elementos favorecem essa situação, incluindo a preocupação com o custo, a necessidade de explorar outros locais para aquisição de recursos altamente qualificados, bem como encontrar uma combinação adequada de conhecimentos para satisfazer as necessidades de investimento impostas pelo mercado.

Com o crescimento do DDS surgiram novos desafios para os projetos de software até então inexistentes ou amenos. Segundo Lescher et al. (2014) o DDS trouxe novos desafios para as equipes de desenvolvimento, a exemplo da separação geográfica, fusos horários diferentes, culturas diferentes e barreiras linguísticas. Estes desafios introduzem atraso de comunicação e muitas vezes acarretam problemas de qualidade e custos elevados.

Herbsleb (2007), neste sentido, afirma que tais particularidades passaram a ser consideradas como fatores de risco (diferenças culturais, fuso horário, idioma) sem mencionar os aspectos do DDS que, segundo o autor, são provenientes de três categorias principais: a forma de separação dos grupos (agrupamento, distância física e separação temporal); as regiões envolvidas (culturas regionais, idiomas e diferenças dos locais) e também as organizações participantes (culturas organizacionais, infraestrutura e relação legal), as quais prejudicam seriamente mecanismos de gerenciamento.

Tais fatores contribuem para acentuar um aspecto crítico para o desenvolvimento distribuído, citado por Seerat (2013), qual seja a ineficiência dos gestores em verificar o *status* do trabalho durante o desenvolvimento do projeto. Outro fato alarmante, é que as empresas estão enfrentando problemas relacionados à execução do processo e a comunicação (SEERAT, 2013). Uma vez que para enfrentar esses novos obstáculos os gestores da área de produção de software precisam adequar-se ao DDS e encarar estes desafios de forma a diminuir os problemas no monitoramento do processo.

No contexto do monitoramento de projetos de software uma técnica vem ganhando notoriedade a partir da ascensão das metodologias ágeis, o Kanban (AHMAD et al. 2013). Uma técnica que surgiu para atender necessidades da indústria em um momento em que a qualidade passou a ser um fator crítico da produção,

o que era alcançado, por meio de manobras que visavam exclusivamente o atendimento a novas demandas de mercado (AHMAD et al. 2013).

Apesar da essência do Kanban ser baseada em uma circunstância totalmente distinta a de projetos, seu objetivo e sua abordagem são compatíveis às principais necessidades em um monitoramento eficiente. A técnica de Kanban, foi basicamente criada e utilizada para gerenciar níveis de estoque na indústria e em redes logísticas, consistia em apresentar para todos os envolvidos no processo produtivo uma gestão visual da cadeia de suprimentos de modo que, produtos representados por cartões eram alocados na fase de suprimento em que se encontram em um painel de monitoramento (AHMAD et al. 2013).

No contexto de projetos, os cartões Kanban representam os pacotes de Atividades que, neste trabalho, são estabelecidos na *Business Process Modeling Notation* (BPMN), e alocados nas suas respectivas fases do ciclo de vida do projeto, por meio do painel de monitoramento (AHMAD et al. 2013). O BPMN foi utilizado para realçar a visão geral do escopo do projeto. Tal modelagem do processo de negócio foi escolhida por ser compreensível por usuários do negócio, analistas, desenvolvedores técnicos e também por aqueles que gerenciam e monitoram o processo (WHITE, 2006).

Quanto ao emprego do Kanban em projetos de software, as principais vantagens, segundo Ahmad et al. (2013), além da organização do escopo de trabalho, viabilizando um monitoramento mais eficiente, são o aprimoramento das comunicações e da integração do projeto perante à equipe que, por ser distribuída, enfrenta problemas desta ordem (AHMAD et al. 2013).

Diante da possibilidade da aplicação do Kanban em projetos, a grande motivação para sua utilização neste trabalho deu-se ao fato, de que, mesmo com sua crescente adoção em projetos de software, a aplicação desta técnica não é frequente no ambiente de desenvolvimento distribuído.

Ahmad et al. (2013) desenvolveram uma revisão sistemática de literatura a respeito do Kanban e, ao investigarem 492 pesquisas contataram, que no que diz respeito ao contexto, 84% dos estudos foram relacionados a projetos centralizados e o restante, 16%, não identificaram claramente o ambiente de trabalho, o que indica que a maioria dos estudos eram realizados em configurações de equipe co-localizada. Além disso, nenhum estudo, pesquisa e/ou relatório técnico, investigado pelos autores, relatou o uso de Kanban em ambientes distribuídos.

Diante do exposto, este trabalho pretende comprovar a seguinte hipótese: Um projeto distribuído de software pode ter o cronograma de suas atividades monitoradas por meio de um processo que aplique a técnica do Kanban e a modelagem BPMN. Para comprovar tal hipótese será utilizado o método experimental, espera-se desta forma, que a técnica Kanban, pouco explorada no DDS, seja vista como vantajosa por gerentes de projetos distribuídos.

## 1.1 MOTIVAÇÃO

A crescente adoção do DDS tem atraído um grande número de pesquisas na área de Engenharia de Software e há pouca razão para esperar que estes fatores diminuam no futuro (HERBSLEB, 2007). Segundo Herbsleb (2007), ocorre uma crescente globalização dos mercados e também da produção e, conseqüentemente, um aumento da pressão para distribuir projetos a nível mundial.

Engenheiros de software têm reconhecido a grande influência desta forma de trabalho e estão em busca de modelos e processos que facilitem o desenvolvimento de software com equipes geograficamente dispersas (CARMEL; AGARWAL, 2001). Além de engenheiros, gerentes e executivos têm enfrentado diversos desafios e dificuldades em diferentes níveis. Recentemente, diversos autores têm escrito sobre dificuldades, desafios e práticas para auxiliar no DDS em nível de gerência de projetos, e demais aspectos, como relatam Fuggetta e Nitto (2014), Chauhan (2014), Seerat (2013), Vivian et al. (2013), Avritzer et al. (2014), Noll et al. (2014), Prikladnicki (2011).

Segundo Carmel e Agarwal (2001), não existem dúvidas “para qualquer profissional que trabalha na área de Engenharia de Software que, tanto o processo de produção de software tradicional (desenvolvimento centralizado de software – DCS), quanto o distribuído possuem diversas dificuldades”. Contudo estas acentuam-se no DDS.

Devido à vinda de grandes empresas multinacionais ao Brasil, e com elas a implantação de unidades *offshore* de produção de software no país, torna-se relevante o desenvolvimento de pesquisas para sanar problemas e amenizar desafios nesta área (PRIKLADNICKI; AUDY, 2010).

Segundo Lindqvist et al. (2006), mesmo em projetos desenvolvidos em um contexto nacional e intra-organizacional, em que *a priori* a linguagem, a cultura e a organização seriam as mesmas, existem dificuldades de monitoramento e gerenciamento a serem solucionadas. Portanto, no DDS a distância geográfica, temporal e sócio cultural são fatores que intensificam estas dificuldades e constituem-se em problemas de primeira ordem.

Neste sentido, Huzita et al. (2008) afirma que a distância física entre as pessoas dificulta o monitoramento das atividades decorrentes do processo de software, o que pode levar a confusão nos reais objetivos de cada colaborador do processo, ocasionando artefatos defeituosos, ou com erros e atrasos em cronogramas. Questões culturais, sociais e de tempo igualmente influenciam e podem levar a mal-entendidos, má interpretação de ações e, também, à falta de um horário comum para interação devido às diferenças de ritmos de trabalho (HUZITA et al., 2008).

O monitoramento de um processo pode ser entendido como o ato de gerenciar interdependências, incertezas e equívocos, conflitos, representação de tecnologias e suas inter-relações. De acordo com este aspecto do monitoramento Wiredu (2005) constata que, este é um desafio no DDS e inclui a coordenação de interações entre pessoas, processos, informações e tecnologias distribuídas, como potenciais problemas. A separação geográfica dos desenvolvedores pode gerar conflitos interpessoal e interunidades.

Quanto ao monitoramento das atividades realizadas no decorrer de um projeto de software, de acordo com Ahmad et al. (2013) o uso do Kanban é um tema emergente. Cujos principais benefícios são: diminuição do tempo de entrega do produto, melhoria da qualidade do produto, melhoria na comunicação e coordenação, aumento da consistência da entrega, e diminuição nos defeitos relatados pelos clientes.

No contexto do monitoramento de projetos, o BPMN vem se destacando também, em especial nas pesquisas sobre engenharia de software (WHITE, 2004). O principal objetivo do BPMN é permitir que o gerenciamento de processos de negócio seja uma tarefa intuitiva e que possa ser utilizada tanto por usuários técnicos, quanto por usuários de negócios, sendo de simples entendimento (WHITE, 2004).

A partir dos aspectos expostos, o crescimento da adoção de DDS, a inerente utilização de Kanban no monitoramento de projetos de software, assim como a simplicidade de aplicação do BPMN, levantou-se a questão que norteia esta pesquisa: **É**

**possível monitorar o cronograma das atividades de um projeto de software, em ambiente distribuído, com um processo que aplica a técnica do Kanban e a modelagem BPMN?**

## 1.2 OBJETIVOS

O objetivo geral deste trabalho é comprovar a seguinte hipótese: Um projeto distribuído de software pode ter o cronograma de suas atividades monitoradas, por meio de um processo que aplique a técnica do Kanban e a modelagem BPMN.

Esta comprovação pode estimular gerentes de projetos distribuídos a utilizar tais abordagens ainda pouco utilizadas no DDS. Abordagens essas tecnicamente simples, mas que podem trazer resultados satisfatórios no monitoramento de projetos de software distribuídos.

O monitoramento fornece uma compreensão do andamento do projeto de modo que ações corretivas apropriadas possam ser tomadas quando o desempenho do mesmo desviar significativamente do planejamento inicial; por isso a importância em se desenvolver um processo de monitoramento efetivo para projetos de software.

Este trabalho com foco no desenvolvimento do produto, leva em conta unicamente o processo de monitoramento do cronograma individual das atividades, que é importante para um bom encaminhamento do projeto. O escopo deste trabalho se limita ao monitoramento dos elementos supracitados, não levando em conta outros elementos do projeto de software, tais como escopo, custo, qualidade de artefatos gerados presentes no ciclo de vida.

Em alinhamento ao objetivo geral proposto no início desta seção, definiu-se como objetivos específicos:

- Elaborar uma proposta preliminar do processo para monitoramento e controle baseado na utilização de BPMN e Kanban para projetos de DDS;
- Especificar um projeto de software para ser implementado, com foco no processo de monitoramento, que contenha todas as fases do ciclo

de vida de um projeto padrão (Planejamento, Especificação dos Requisitos, Design, Implementação, Teste e Implantação);

- Desenvolver um experimento controlado para validação da hipótese levantada.

### 1.3 ORGANIZAÇÃO DO TEXTO

Este trabalho está assim organizado:

*Capítulo 1 - Introdução:* Posiciona o leitor com relação ao projeto de pesquisa e apresenta o contexto da área de aplicação da mesma;

*Capítulo 2 - Base Teórica:* Apresenta uma reflexão sobre os principais autores da literatura, pela exploração reflexiva dos textos encontrados. Na seção 2.1 é definido o conceito de processo de produção, assim como suas características, na subseção 2.1.1 a hierarquia de um processo de produção é apresentada. A seção 2.1.2 trata do processo de produção de software que é o foco deste trabalho. Posteriormente a seção 2.1.3 é dedicada à técnica Kanban, com enfoque em seus aspectos relevantes.

As subseções 2.1.4, 2.1.5, 2.1.6 e 2.1.6.1, tratam especificamente de gerenciamento de processos de negócio e da modelagem de processo de negócio. A notação BPMN é conceituada, uma vez que será aplicada juntamente ao Kanban no processo de monitoramento de projetos distribuídos, instrumento fundamental para a prova da hipótese, objetivo do trabalho.

A seção 2.2 trata das principais características do Desenvolvimento Distribuído de Software, suas vantagens e desvantagens, e conceitos aplicados; na subseção 2.2.1 são descritos Modelos de Negócios em DDS e a seção 2.3 traz os principais desafios do DDS.

A seção 2.4 descreve o processo de monitoramento e controle que são interligados nos principais padrões de projeto existentes; contudo, neste trabalho, o foco será apenas o processo de monitoramento, isso porque o escopo de controle acrescentado ao trabalho tornar-se-ia excessivo, para o processo implementado. A Subseção 2.4.1 apresenta os elementos que são monitorados e controlados em um projeto, e a subseção 2.4.2 apresenta o processo preliminar de monitoramento de proje-

tos distribuídos, desenvolvido pelo autor, base para a elaboração e especificação do processo proposto. Por fim a seção 2.5 traz as considerações finais da base teórica.

*Capítulo 3 – Trabalhos Relacionados:* Apresenta um mapeamento geral de literatura com trabalhos que seguem a mesma linha de pesquisa a que este se refere;

*Capítulo 4 - Processo para Monitoramento de Projetos Distribuídos:* Aborda o processo elaborado como instrumento, que se apoia na utilização do Kanban e do BPMN, para dar suporte ao objetivo geral, que é a comprovação da Hipótese levantada no capítulo 1.

*Capítulo 5 - Metodologia:* Caracteriza a metodologia experimental empregada no trabalho, motivos da seleção do método e demais aspectos necessários para o seu desenvolvimento. Deste modo, apresenta as fases iniciais do processo experimental, a Definição do Escopo e ao Planejamento deste;

*Capítulo 6 – Operacionalização do Experimento:* Expõe as fases do experimento realizado, a partir da preparação, bem como todas as etapas que compõem sua organização; a execução do experimento, que sistematiza as atividades que foram efetuadas durante esta fase e, ao final do capítulo, é denotada a validação dos dados que foram coletados com aplicação deste método.

*Capítulo 7 – Análise e interpretação dos Resultados:* Refere-se à análise estatística das informações coletadas no experimento em relação ao parâmetro estabelecido. Expõe também os critérios utilizados para a prova das hipóteses levantadas. Por fim, o capítulo apresenta os dados referentes ao questionário pós experimento realizado com os participantes do mesmo.

*Capítulo 8 – Conclusões:* Evidencia as conclusões obtidas a partir da execução do experimento e da análise dos dados coletados em seu decorrer. Também apresenta as contribuições da pesquisa, as suas limitações e possíveis retomada do tema em trabalhos futuros.

## 2 BASE TEÓRICA

Este capítulo dividido em 5 seções, discorre acerca da conceituação teórica concernente ao tema abordado. A seção 2.1 que apresenta a base teórica a respeito de processos, o de software e o de negócio. Na seção 2.2 é abordado o DDS, com a devida descrição dos principais conceitos e motivações, também modelos de negócio em DDS. Na seção 2.3 são elencados os principais desafios enfrentados pelas equipes distribuídas.

A seção 2.4 aponta os principais aspectos e conceitos a respeito do monitoramento e controle de projetos (foco deste trabalho é o monitoramento), contudo por estes dois processos serem interligados, na grande maioria dos textos, citados neste trabalho, eles são descritos juntos; nesta seção também estão sistematizados os elementos monitorados e processo preliminar desenvolvido. Na seção 2.5 estão as conclusões a respeito da base teórica.

### 2.1 PROCESSOS DE PRODUÇÃO E PRODUÇÃO DE SOFTWARE

Segundo Martins e Laugeni (2013) um processo de produção é um sistema de ações que estão inter-relacionadas de forma dinâmica e orientadas para a transformação de determinados elementos. Como tal, os elementos de entrada, conhecidos como fatores, passam a ser elementos de saída, produtos, na sequência de um processo em que é incrementado o seu valor. Convém destacar que os fatores são os bens que são utilizados com fins produtivos, as matérias-primas. Os produtos, por sua vez, estão destinados à venda ao consumidor (MARTINS; LAUGENI, 2013).

O conceito de processo, segundo a NBR ISO 9001:2015 corresponde ao “conjunto de atividades inter-relacionadas ou interativas que transformam insumos (entradas) em produtos (saídas)” (ABNT, 2008). Sabendo-se que em qualquer processo sempre existe certa quantidade de variabilidade em fatores como materiais, máquinas, medidas, mão-de-obra, método e ambiente. Processo, segundo Cruz (2005), é “a introdução de insumos - entradas – em um ambiente, formado por



procedimentos, normas e regras, que, ao processarem tais insumos, transformam-nos em resultados que serão enviados, saídas, aos clientes do processo”.

Cruz (2005) reafirma que um processo é uma transformação de entradas em saídas (produto acabado), e esse resultado deve atender as exigências de um cliente, mediante o atendimento às várias características de qualidade. Para isso, o processo deve apresentar estabilidade. Algumas das entradas de um processo são difíceis de controlar, podendo interferir nessa estabilidade, bem como no ambiente de produção de software.

Além dos insumos – entradas e resultados – saídas, um processo possui, ainda, mais dois elementos fundamentais para sua execução: o controle e os recursos. Assim, um processo típico é formado por quatro elementos, como salienta Cruz (2005):

- **Entradas:** são os insumos, materiais ou informações que servem de subsídio para o processo ser executado;
- **Saídas:** são os produtos ou serviços produzidos pelo processo;
- **Controle:** é o responsável pela verificação do cumprimento de diretrizes, métodos, objetivos, procedimentos e padrões na execução do processo;
- **Recursos:** representam a provisão de tudo que se faz necessário para a execução do processo recursos administrativos, financeiros, humanos, materiais, informações, entre outros.

Outro elemento acrescentado à definição de processo é o monitoramento, citado por Kalus e Kuhrmann (2013), como pertencente ao processo de produção.

Cada processo possui clientes e outras partes interessadas, tanto internas quanto externas, que são afetadas por ele e que definem as saídas desejadas de acordo com suas necessidades e expectativas. Todos os processos devem estar alinhados aos objetivos da organização e ter a finalidade de agregar valor em relação ao escopo e à complexidade da organização.

A eficácia e a eficiência do processo podem ser avaliadas, por meio de processos de análise crítica, internos e externos, caracterizados como o controle e o monitoramento, sendo estes importantes para plena realização do processo de produção (KALUS e KUHRMANN, 2013).

Para melhorar um processo de maneira contínua e criar uma mudança duradoura, é fundamental que seja compreendido o funcionamento atual do processo.

Quando a empresa observa e compreende o estado atual, o caminho para o aprimoramento fica mais claro (KALUS e KUHRMANN, 2013).

Segundo Kalus e Kuhrmann (2013), os processos possuem características básicas que suportam a implantação de seu monitoramento e controle que são:

- **Fluxo de Valor:** transformação de entradas e saídas, com a utilização de recursos da empresa, com a esperada agregação de valor.
- **Eficácia:** grau com que as expectativas do cliente são atendidas. Ser eficaz é fazer o que o cliente quer.
- **Eficiência:** grau de aproveitamento dos recursos para gerar uma saída. Ser eficiente é fazer o que o cliente quer da melhor forma para a empresa (otimizando o processo).
- **Tempo de ciclo:** tempo necessário para transformar uma entrada numa saída. Deseja-se que o tempo de ciclo seja o menor possível.
- **Custo:** recursos despendidos no processo.

Segundo os mesmos autores o conhecimento destas características é importante para: identificar as áreas com oportunidades de melhoria; fornecer o conjunto de dados para a tomada de decisão; fornecer a base para definir metas de aperfeiçoamento e avaliar resultados.

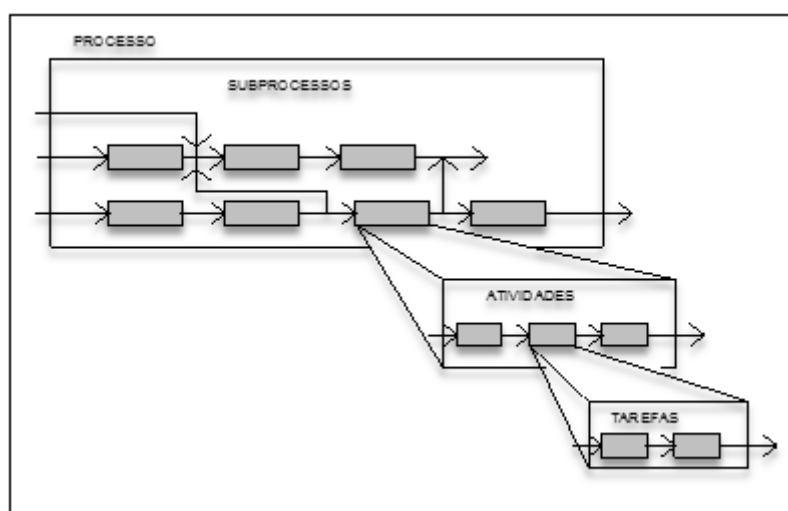
### 2.1.1 Hierarquia do Processo

De acordo com Oliveira (2012) existe uma hierarquia em processos que consiste em:

- **Macroprocesso:** é um processo que geralmente envolve mais de uma função da organização, e cuja operação tem impacto significativo nas demais funções da organização. Dependendo da complexidade do processo este é dividido em subprocessos.
- **Subprocesso:** divisões do macroprocesso com objetivos específicos, organizado seguindo linhas funcionais. Os subprocessos recebem entradas e geram suas saídas em um único departamento.

- **Atividades:** são subdivisões que compõem os subprocessos; são mais pontuais e resultam diretamente em artefatos tangíveis para o processo.

Porém, alguns estudos, tais como em Martins e Laugeni (2013), Kalus e Kurmann (2013), Oliveira (2012), consideram o desmembramento da atividade em tarefas. Na Figura 01 está demonstrada a estrutura hierárquica dos processos, partindo do processo até o nível de tarefas, que são as unidades mínimas da organização horizontal.



**Figura 01 - Hierarquia do Processo**  
**Fonte: Adaptado de Oliveira (2012)**

O processo de produção é composto por subprocessos, como o subprocesso de trabalho, o qual objetiva transformar a força do homem em produtos que possam ser valorizados e comercializados. Segundo Campos (2001) o subprocesso de trabalho é constituído por três elementos:

- O trabalho: uma atividade produtiva com objetivos definidos;
- O (s) objeto (os): sobre os quais o trabalho é realizado;
- Os meios: que facilitam o processo de trabalho, ferramentas, e ou máquinas.

O subprocesso de trabalho é o elemento fundamental no processo de produção responsável por delimitar as ações necessárias para ao desenvolvimento de uma atividade dentro da produção, caracteriza-se como uma unidade dentro do processo como um todo, sendo que este constrói os artefatos tangíveis no decorrer do processo produtivo.

As ações produtivas são as atividades desenvolvidas no âmbito do processo. Podem ser ações imediatas, que geram serviços que são consumidos pe-

lo produto final, independentemente do seu estado de transformação, ou ações mediadas, que geram serviços que são consumidos por outras ações ou atividades do processo (KALUS; KUHRMANN, 2013).

O processo de produção possui particularidades e estas utilizam diferentes formatos de gerência, controle e monitoramento de projetos; por isso diferentes métodos e modelos de produção foram sistematizados, ao longo dos tempos para se adaptar a evolução da administração, a exemplo do Taylorismo, o Fordismo e o Toyotismo, os quais ainda são referências para os processos de produção moderna. Baseando-se nestes processos é possível sistematizar um modelo para produção de qualquer tipo de produto (KALUS; KUHRMANN, 2013), inclusive softwares.

Segundo Fleischmann et al. (2013) o Taylorismo se caracteriza como um processo de produção, administração e organização baseado na aplicação de um método científico no processo produtivo a fim de garantir maior produtividade. Nessa perspectiva de racionalização da produção, este processo propunha estratégias para aumentar o nível da produção em menos tempo sem aumentar o custo.

É comum encontrar o conceito de Fordismo atrelado ao Taylorismo, visto que Henry Ford utilizou em suas indústrias um modelo bem próximo da organização científica de Taylor, como a racionalização da produção. Ford implantou no modo de produzir automóveis em linha de montagem, esteiras e padronização das peças e dos serviços, divisão do trabalho em várias etapas com tarefas repetitivas com uma supervisão autoritária e imposição de poder a fim de garantir a disciplina do trabalhador (FARIA, 2004).

Contudo, o processo de produção que se sobressai aos da escola clássica da administração de acordo com Faria (2004), e que possui seus princípios aplicados em empresas, é o Toyotismo, cuja característica principal é a redução das grandes estocagens, produzindo pouco, com grande qualidade, e diversidade de produtos. O trabalhador passa a ser mais participativo e qualificado se envolvendo em várias etapas do processo de produção.

As principais modalidades de políticas de gestão e organização do trabalho normalmente associadas ao Toyotismo são: o *just-in-time*, o *Kanban* e os Círculos de Controle da Qualidade (CCQ) (HELOANI, 2003).

### 2.1.2 Processo de Software

Segundo Sommerville (2011) um processo de software pode ser visto como o conjunto de atividades, métodos, práticas e transformações que guiam pessoas na produção de software. Um processo eficaz deve, claramente, considerar as relações entre as atividades, os artefatos produzidos no desenvolvimento, as ferramentas e os procedimentos necessários e a habilidade, o treinamento e a motivação do pessoal envolvido.

Para Pressman (2013) as principais fases de um processo de software são: levantamento dos requisitos, análise, projeto de sistema, codificação, verificação e validação, integração, implantação e manutenção. Em cada fase de um processo de software são executadas atividades inerentes para que os objetivos propostos possam ser atingidos. Segundo o autor já citado, estas atividades constituem um conjunto mínimo para se obter um produto de software. Ainda existem questões de alocação de recursos e pessoal que precisam ser previstos quando do início de cada projeto.

Diversas razões levam a utilização de um processo de software padronizado, como: redução dos problemas relacionados a treinamento, revisões e suporte a ferramentas. Experiências adquiridas nos projetos são incorporadas ao processo padrão e contribuem para melhorias em todos os processos definidos, economia de tempo e esforço na definição de novos processos adequados a projetos [(SOMMERVILLE, 2011) (PRESSMAN, 2013)].

Para a produção de software não existe a combinação somente de técnicas e ferramentas, é predominantemente uma atividade humana (ANGKASAPUTRA; PFAHL, 2004). Segundo estes autores em uma empresa de software, os principais recursos para o desenvolvimento das atividades são providos pelas interações humano-computador. A necessidade de planejamento do processo produtivo torna-se indispensável para que haja um melhor aproveitamento desses recursos.

Alguns autores afirmam que a produção de software é um processo empírico. Para Ziv (1997) a incerteza é inerente e inevitável à produção de software. Conforme Wegner (2005) não é possível especificar completamente um sistema interativo. Tais particularidades diferenciam-no de outras áreas da produção industrial. Um

dos pontos principais para tal ponderação é a não linearidade desse tipo de processo.

Contudo, com a evolução dos paradigmas, métodos, técnicas e processos para a produção de software atualmente é possível monitorar e controlar a produção de software melhorando a qualidade do produto, auxiliando a cumprir prazos e custos planejados e a atender requisitos de qualidade [(SCHARFF, 2011) (AHMAD et al. 2013)].

Conforme Sommerville (2011) um processo de software é o que define quais atividades devem ser realizadas dentro de uma produção de sistemas; define também as pessoas envolvidas e os artefatos a serem gerados. Já o modelo de processo de software define a sequência de execução das atividades, as pessoas envolvidas e quais os artefatos são gerados em cada atividade.

Existem diversos modelos de processos tradicionais na Engenharia de *Software*, dirigidos ao desenvolvimento de software co-localizado. Contudo, também surgiram paradigmas especializados no DDS como o MuNDDoS (Modelo de referência para desenvolvimento distribuído de software), que contempla as dimensões técnicas, e não técnicas, e os fatores envolvidos em cada uma, proposto por Prikladnicki (2003). O M3DS (Um modelo de Dinâmica para o Desenvolvimento Distribuído de Software) que traz um modelo de dinâmica para o desenvolvimento distribuído de software, elaborado por L'Erario (2009).

Também há propostas para formalizar apenas uma parte do processo de produção de software em ambientes distribuídos, como o trabalho realizado por Lopes (2004), que propõe um modelo de referência para a engenharia de requisitos nestes ambientes. Trabalhos como estes, estão se intensificando na área de DDS, como a proposta de Arquitetura para Ambientes de Desenvolvimento Distribuído de Software de Wiese (2006). Gartner (2011) (DDS) propôs um ambiente integrado para apoio para DDS e Espindola (2006) elaborou uma arquitetura de informação para a gerência de requisitos em DDS. Desta forma, há muitos esforços para que o DDS tenha sucesso e que sejam amenizados os desafios enfrentados neste paradigma, principalmente pela abrangência que este vem tomando.

### 2.1.3 Kanban

Com mercado competitivo e exigente, muitas empresas buscam criar técnicas eficientes de produção com o intuito de baixar custos, garantindo a qualidade dos produtos, e baixar a quantidade de estoque. Segundo Bernardo (2009) o Sistema Toyota de Produção, criado por Taiichi Ohno, tem por objetivo a eficiência por excelência, por meio da eliminação de desperdícios, bem como refugos industriais. A Toyota Motors Company foi pioneira na adoção de tal sistema; assim, o Sistema Toyota de Produção, ganhou notoriedade pelo aumento dos lucros e redução de custos. Este sistema é composto pelo Sistema *Just-in-time* (JIT), incluso neste, a técnica Kanban.

O JIT é uma abordagem disciplinada para melhorar a produtividade e qualidade total, por meio da eliminação das perdas e respeito pelas pessoas. Na fábrica e na linha de montagem este método proporciona um custo efetivo e entrega somente das peças necessárias de qualidade, na quantidade, no tempo e lugar certos, utilizando o mínimo de instalações, equipamentos, demais materiais e recursos humanos (HELOANI, 2003).

Segundo Heloani (2003) o JIT tem como objetivo principal produzir bens e serviços exatamente no momento em que são necessários. Não antes, para não se transformarem em estoques, e não depois para que seus clientes não tenham que esperar. Segundo Uchida (2007) é uma técnica que se resume em operar a indústria com uma logística, de forma simples, mas eficiente, otimizando materiais, recursos financeiros e mão de obra.

O Kanban surgiu da carência de se ter um controle de fluxo de material na fábrica, com estoques tendendo a zero. É uma técnica gerenciadora do Sistema JIT. Para Bernardo (2009) “o Kanban é essencial para a implantação do sistema JIT. Ele é um cartão ou etiqueta de pedido de trabalho, sujeito à circulação repetitiva na área.” O autor ainda complementa que “Kanban é uma técnica que reduz o tempo de espera, diminuindo o estoque, melhorando a produtividade e interligando todas as operações em um fluxo uniforme ininterrupto”.

O conceito de Kanban para Uchida (2007):

É uma técnica de gestão de materiais e de produção no momento exato (Just in Time), que é controlado, por meio do movimento do cartão (Kan-

ban). O sistema Kanban é uma técnica de "puxar" as necessidades dos produtos acabados e, portanto, é oposto aos sistemas de produção tradicionais. É um sistema simples de autocontrole em nível de fábrica, independente de gestões paralelas e controles computacionais.

De acordo com o que foi citado por diferentes autores como Heloani (2003), Bernardo, (2009), Uchida (2007), resumidamente, entende-se Kanban como uma técnica de monitoramento por meio de cartões ou registros em painéis, ou seja, um monitoramento de toda a produção, puxando a responsabilidade para si.

O Kanban é utilizado também na produção de software; e este tem seus conceitos principais adaptados para condizer com o processo de desenvolvimento de software. O trabalho de construir uma nova funcionalidade para um sistema só é gerado a partir do momento que uma funcionalidade anterior já tenha sido implementada (AHMAD et al., 2013). O Kanban, dentro desse contexto, procura aperfeiçoar os processos, as equipes e projetos. É útil para empresas que procuram melhorar constantemente seus processos, ao passo que melhoram também sua produtividade e sua relação com os clientes. Esse método começou a crescer no setor de tecnologia de informação após sua adoção na *Agile Conference* (Conferência Ágil), em agosto de 2007. Segundo Ahmad et al. (2013) a primeira aplicação para Engenharia de Software foi na empresa Microsoft em 2004.

De acordo com Ahmad et al (2013), o principal objetivo dessa metodologia é avaliar o progresso do trabalho, chamado *Work in Progress* (WIP). Segundo os mesmos autores essa avaliação tem a proposta de mostrar quando uma funcionalidade do software pode ser arquitetada, codificada ou testada.

Nesta técnica, percebe-se que mais trabalho pode ser realizado com uma demonstração visual. Por meio do quadro de atividades, é possível detectar qual a quantidade de esforço que poderá ser adicionada. É importante lembrar que, para isso, é preciso também levar em consideração a indicação de limite e de capacidade da equipe e do software desenvolvido. Algumas das empresas que implantam essa técnica possuem um quadro com espaços que indicam a capacidade (AHMAD et al, 2013).

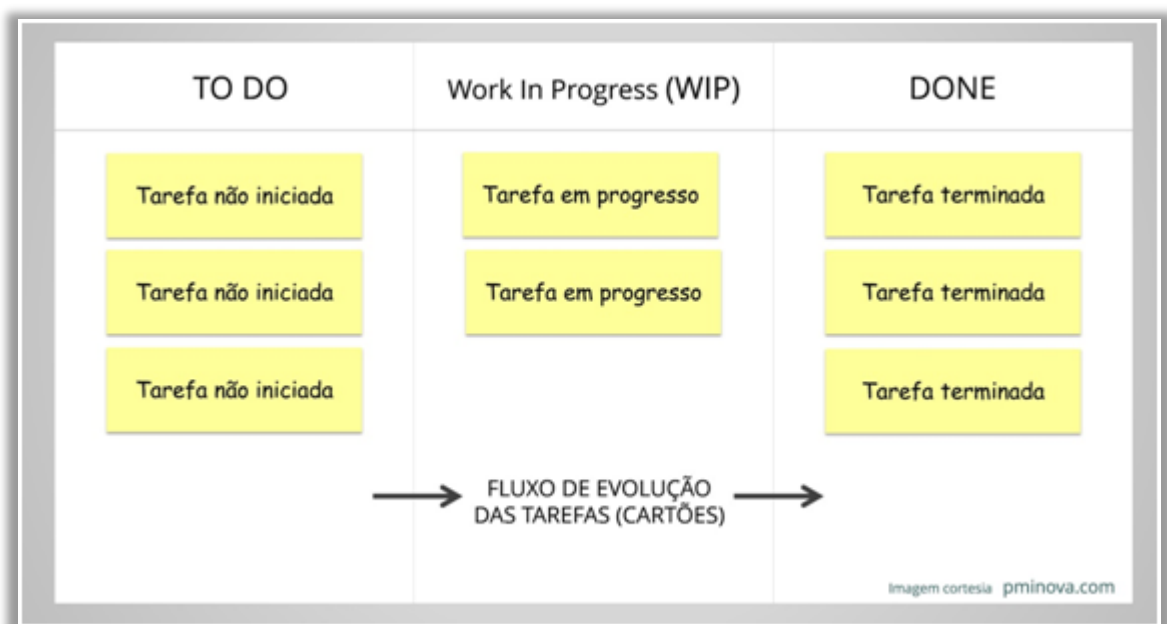
A equipe pode fazer entregas a qualquer instante para o cliente e o mesmo pode modificar a importância das atividades quando desejar. O desenvolvimento torna-se mais transparente, sem preocupações com as iterações e estimativas, como em outros métodos ou técnicas (AHMAD et al, 2013).



O Kanban fornece visibilidade ao processo de software, porque mostra o trabalho atribuído de cada desenvolvedor, bem como comunica claramente as prioridades. Além disso, seu objetivo é minimizar o WIP ao desenvolver apenas os itens que são solicitados. Isto produz um fluxo constante de itens de trabalho divulgados aos clientes; e os desenvolvedores focam apenas naqueles itens em determinado momento. O Método visa adaptar rapidamente o processo usando *feedbacks* em ciclos mais curtos.

Os resultados da aplicação da técnica Kanban em produção de software é altamente positivo, correspondendo às vantagens conseguidas na indústria de transformação (AHMAD et al. 2013). Das técnicas para desenvolvimento de software, o Kanban é a menos prescritiva, característica essa que estimula ainda mais as equipes a adotarem esse método. Segundo Ahmad (2013) ele tem apenas três prescrições: “visualize o fluxo de trabalho atual; Limite o fluxo de trabalho; Acompanhe e gerencie o fluxo de trabalho”.

Sendo uma técnica pouco prescritiva, o Kanban acaba tornando-se muito adaptativo. Com isso, as equipes que o adotam precisam estar atentas ao processo aplicado para visualizar locais de melhoria e adaptações para que o processo possa fluir de forma satisfatória, além de facilitar a visualização e gestão, bem como dirigir os membros da equipe, cooperar e comunicar de forma mais eficiente e eficaz (AHMAD et al., 2013). A Figura 02 representa um exemplo de Kanban.



**Figura 02 – Exemplo de Kanban**  
**Fonte: Adaptado de Ahmad et al. (2013)**

O Kanban procura identificar oportunidades de melhoria criando uma cultura aperfeiçoada na equipe, na qual a melhoria contínua é responsabilidade de todos. Os princípios básicos em torno dessa ferramenta são, segundo Ahmad et al. (2013) é:

- Visualização da Cadeia de Valor: enxergando as fases do produto (por exemplo, um software, algo material ou até mesmo um serviço);
- Desenvolvimento Evolucionário (Adaptativo): por meio de gestão de mudanças simples, adaptando-se de forma ágil, entregando o que tem mais valor antes. Perceba que a palavra “ágil”, muitas vezes confundida com rapidez, aqui significa capacidade de se adaptar às mudanças com mais facilidade, mais agilidade;
- Restrição do trabalho e seu progresso em torno de seus estágios: permitindo medição, controle e melhoria contínua.

#### 2.1.4 *Business Process Management* (BPM)

A BPM tem, como um dos seus principais fundamentos, a realização dos objetivos de uma organização por meio da melhoria da gestão e do controle dos seus processos de negócio essenciais (JESTON; NELIS, 2006).

A introdução de processos de negócio nas organizações trouxe um desafio à administração, como administrar organizações orientadas por processos de negócio. Uma das respostas para este desafio foi o desenvolvimento da teoria da gestão de processos de negócio, também conhecida como *Business Process Management* (BPM).

O BPM envolve a descoberta, projeto e entrega de processos de negócio. Adicionalmente inclui o controle executivo, administrativo e supervisorio desses processos (BPMN, 2007). Contador et al. (2005) conceituam BPM como uma estrutura gerencial orientada a processos, em que o gestor, time e executores do processo são todos executores e pensadores enquanto projetam seu trabalho, inspecionam seus resultados e redesenham seus sistemas de trabalho de forma a alcançar melhores resultados.

Contador et al (2005) definiram um conceito que chamaram de terceira onda da gestão de processos, no qual processos podem ser visualizados por usuários humanos como informação e, por máquinas, como código executável, ao mesmo tempo. Um padrão aberto é usado para definir todos os processos, na forma *top-down* nos níveis de estratégia de negócio e de design de processo e de forma *bottom-up* no nível de alinhamento com os sistemas de TI existentes.

Os sistemas de gestão de processos de negócio se apoiam no profundo conhecimento deste para garantir o sucesso da automação das atividades. O *Business Process Management* (BPM) é, em síntese, um conceito que une gestão de negócio e tecnologia da informação, voltado à melhoria dos processos das organizações por meio do uso de métodos, técnicas e ferramentas para modelar, publicar, controlar e analisar processos operacionais, envolvendo elementos humanos, aplicações, documentos e outras fontes de informação (BPMN, 2007). A BPM tem como diferencial a integração entre a visão modelada para a visão de execução.

### 2.1.5 Modelagem de Processo

A modelagem de processos de negócio é o primeiro passo do processo de BPM, porém, para que um modelo se torne realmente útil, acredita-se que o passo seguinte, a execução do processo, é ainda mais importante para as organizações do que a própria modelagem (MOMOTKO; NOWICKI, 2003).

Uma notação padrão para modelagem de negócios pode prover para as organizações a capacidade de entender seus procedimentos internos de negócio de forma gráfica e dar a eles a habilidade de comunicar estes procedimentos de modo padrão.

Além disso, a notação gráfica facilita o entendimento das colaborações e as transações de negócio entre organizações, permitindo que empresas se entendam e também os participantes em seu negócio, permitindo que estas se ajustem às circunstâncias de *Business to Business* (B2B) rapidamente (BPMN, 2007).

Modelos de processos de negócio descrevem como o trabalho é executado, ou mais especificamente, como se cumprem as missões e atividades. Um modelo mostra como um negócio cumpre suas atividades. Para detalhar o “como” da maioria

das empresas reais, seria necessário uma série de modelos que podem consistir em vários atores (pessoas, organizações, sistemas) executando várias tarefas. De modo a cumprir toda uma tarefa, atores devem completar sub-tarefas especificadas de maneira coordenada. Algumas vezes estas sub-tarefas podem ser executadas em paralelo, outras vezes são sequenciais. Alguns processos requerem repetição de sub-tarefas. A maioria dos processos tem pontos de decisão onde fluxos de processo podem ser ramificados, dependendo da condição do sistema ou de uma condição específica da execução em questão.

Em processos cooperativos, atores devem passar informação para outros atores, essa transferência de informação pode disparar uma ação para outro ator realizar. Também existem situações onde o ator pode não precisar completar toda uma sub-tarefa antes que ele mesmo ou outro ator inicie outra sub-tarefa dependente.

Alguns processos também podem ser *ad-hoc*, ou seja, as sub-tarefas não têm gatilhos bem definidos, acontecem sem programação ou de forma semi-programada. Uma metodologia de modelagem de processos de negócio deve ser capaz de representar estes diferentes aspectos da descrição de um processo, ainda que um processo possa parecer diferente quando descrito a partir da perspectiva de diferentes atores.

Um bom modelo deve apresentar uma representação de uma maneira que seja fácil de ser transferida para o conhecimento tácito para quem estiver visualizando o modelo. A vantagem de usar modelos estruturados para a descrição de processos está na qualidade da captura e no conhecimento formalizado, comparando-se com modelos estruturados de processos com descrições textuais (KALPIC; BERNUS, 2002).

A criação de um modelo de processo pode ser instrutiva por si só, revelando anomalias, inconsistências, ineficiências e oportunidades para melhoria. Uma vez criado, especialmente se de forma informatizada, é um valioso meio de compartilhar o conhecimento pelos departamentos da empresa. O modelo também pode ser usado para formular e avaliar mudanças, como por exemplo o lançamento de um novo produto e processos de negócio associados (KALPIC; BERNUS, 2002). Existem vários usos potenciais de modelos de processos (KALPIC; BERNUS, 2002):

- a) Programar o planejamento;
- b) *Baseline* para melhoria contínua;

- c) Retenção de conhecimento e aprendizagem;
- d) Visualização do processo;
- e) Treinamento;
- f) Framework para métricas;
- g) Conformidade e auditoria;
- h) Programar a execução.

#### 2.1.6 Notações para Representação de Processos de Negócio

Muitas metodologias foram criadas para modelar processos. O essencial é saber quais informações são relevantes para a compreensão do processo em totalidade (JESTON; NELIS, 2006). Algumas das principais linguagens existentes voltadas para a especificação de modelos são:

a) *UML 2.0 Activity Diagram (AD)*: Projetado para modelar processos de negócio e fluxos em sistemas de software. Sua origem está embasada no desenvolvimento de software;

b) *Business Process Definition Metamodel (BPDM)*: Desenvolvida pelo Object Management Group (OMG) oferece um meta-modelo genérico para processos de negócio. A BPDM não provê uma notação gráfica própria, sua intenção é apenas definir um meta-modelo genérico com o objetivo de apoiar o mapeamento entre diferentes ferramentas e linguagens;

c) *Business Process Modeling Notation (BPMN)*: Criada para projetar e modelar processos de negócio e suas transformações na linguagem de execução *Process Modeling Language (BPML)*;

d) *Event Driven Process Chain (EPC)*: Desenvolvido para modelar processos de negócio que sejam facilmente entendidos e utilizados pelo pessoal de negócios. Seus elementos básicos são funções e eventos;

e) *Integrated DEFinition Method 3 (IDEF3)*: Projetado para modelar processos de negócio e sequências de um sistema, provendo duas perspectivas, o esquema do processo e o esquema de objetos;

f) *Petri Net*: A rede de Petri foi projetada para modelagem, análise e simulação de sistemas dinâmicos, por meio de procedimentos concorrentes e não determinísticos. Redes de Petri são utilizadas para modelar *workflows*, por meio de grafos;

g) *Role Activity Diagram* (RAD): Tem sua origem na modelagem e coordenação, sendo usado para modelar processos de negócio com ênfase nos papéis, atividades e interações com eventos externos.

#### 2.1.6.1 *Business Process Model and Notation* (BPMN)

A simbologia da BPMN permite criar modelos de processos de negócio – *Business Process Diagrams* (BPD) para finalidades de documentação e comunicação. Esses modelos seguem uma notação padrão, desenvolvida pelo Instituto de Gestão de Processos de Negócio – *The Business Process Management Initiative* (BPMI) e foi lançada publicamente em maio de 2004.

A especificação da BPMN representa mais de dois anos de esforços do grupo de trabalho do *Business Process Management Initiative* (BPMI). O primeiro objetivo deste esforço foi o de prover uma notação que fosse de fácil leitura e compreensão por todos os usuários de negócios. Desde analistas de negócios, que criam os modelos iniciais de processos de negócio, aos desenvolvedores técnicos, responsáveis pela implementação tecnológica e finalmente para as pessoas do negócio, que irão gerenciar e monitorar estes processos.

A BPMN foi selecionada, dentre as notações para modelagem de processos citadas na subseção anterior, para ser utilizada neste trabalho, por ser de fácil utilização e entendimento, bem como fornecer a habilidade de modelar processos de negócios complexos. Assim, apesar de sua manipulação ser simples, ela é capaz de modelar os mais diversos tipos de processos. A BPMN é uma das três especificações desenvolvidas pela BPMI e utiliza um padrão aberto, definido pela *Object Management Group* (OMG) (BPMN, 2007).

A BPMN é uma notação que tem como propósito a geração de um diagrama de processos de negócio chamado de *Business Process Diagram* (BPD). O BPD é construído, por meio de um conjunto básico de elementos gráficos. Estes elementos permitem o desenvolvimento de diagramas que são, normalmente, familiares para a

maioria dos analistas de negócio, pois são parecidos com fluxogramas (WHITE, 2004).

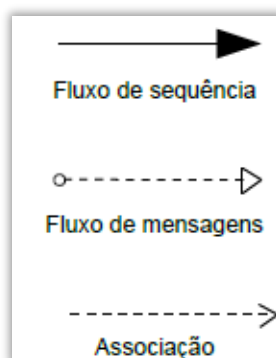
Um dos desafios da BPMN é a criação de mecanismos simples para a construção de modelos de processos de negócio, com a capacidade de lidar com a complexidade inerente a estes processos (WHITE, 2004). A BPMN usa uma abordagem minimalista para lidar com estes dois requisitos conflitantes, utilizando um pequeno conjunto de categorias de notação, para que o leitor de um BPD possa facilmente reconhecer os tipos básicos utilizados e assim compreender a essência do diagrama. As quatro categorias básicas de elementos são: os objetos de fluxo, objetos de conexão, raias e artefatos, que são descritos de acordo com BPMN (2007) em detalhe a seguir.

- **Objetos de fluxo:** A BPMN descreve um conjunto de três objetos de fluxo: eventos, atividades e *gateways* presentes na Figura 03. Os eventos representados por círculos demonstram acontecimentos no curso de um processo e afetam o fluxo deste que, eventualmente, podem ter uma causa ou impacto. As atividades são representadas por retângulos com cantos arredondados e são usadas para demonstrar algum tipo de trabalho realizado na empresa. Os *gateways* são representados por um losango e são usados para controlar a divergência e a convergência de um fluxo de controle, determinando decisões tradicionais e também caminhos paralelos ou junções de caminhos.



**Figura 03 – Objetos de Fluxo Básico da BPMN**  
Fonte: BPMN (2007)

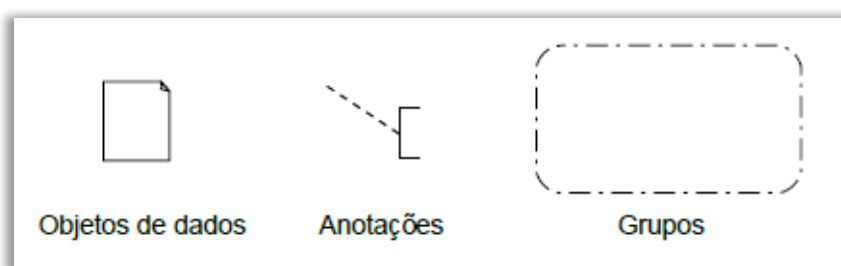
- **Objetos de conexão:** Os objetos de conexão, ou objetos de fluxo, são conectados ao diagrama para criar o esqueleto estrutural básico de um processo de negócio. Existem três tipos básicos de objetos para prover esta função, a Figura 04 ilustra estes tipos.



**Figura 04 – Objetos de Conexão da BPMN**  
**Fonte: BPMN (2007)**

O fluxo de sequência é representado por uma linha sólida e uma seta sólida, para demonstrar a ordem que as atividades serão executadas em um processo. O fluxo de mensagens é representado por uma linha pontilhada, com uma seta aberta na sua extremidade para demonstrar o fluxo de mensagens entre dois participantes de processos separados de forma organizacional, como, setores diferentes, unidades de negócio distintas, ou até mesmo uma outra empresa. A associação é representada por uma linha pontilhada com uma seta aberta na extremidade para associar dados, textos e outros artefatos com objetos do fluxo.

- **Artefatos:** A especificação da BPMN define três tipos de artefatos básicos: objetos de dados, anotações e grupos. Tais artefatos estão representados na Figura 05.



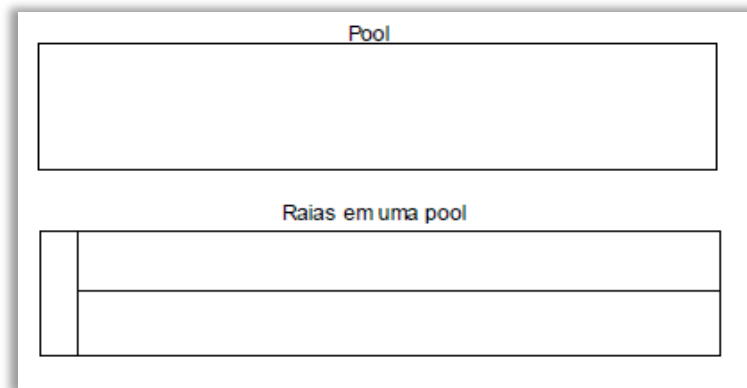
**Figura 05 – Artefatos básicos da BPMN**  
**Fonte: BPMN (2007)**

Os objetos de dados são mecanismos que demonstram como os dados são requeridos ou produzidos por atividades. Eles são conectados em atividades por meio de associações. Os grupos são representados por um retângulo pontilhado e pode ser usado com o propósito de destaque na documentação ou análise, porém não afeta o fluxo de sequência. As anotações são meca-



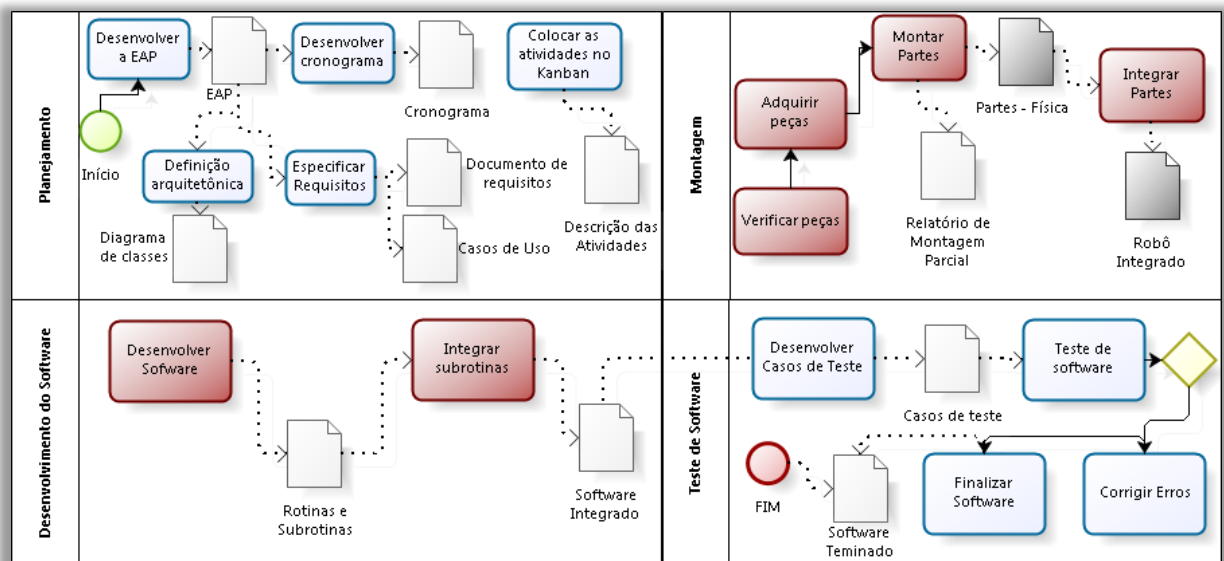
nismos que provém ao modelador a capacidade de descrever informações textuais adicionais ao leitor do diagrama.

- **Raias:** A BPMN utiliza o conceito de raias de natação (*swimlanes*) como um mecanismo para organizar atividades em diferentes categorias visuais, de forma a ilustrar diferentes capacidades funcionais ou responsabilidades. Estas categorias são suportadas pelo BPMN, por meio de dois tipos de construtos, *pools* e *lanes*, como pode ser observado na Figura 06.



**Figura 06 – Elementos da Raia da BPMN**  
Fonte: BPMN (2007)

Um exemplo de um processo de negócio implementado em BPMN é apresentado na Figura 07. O processo possui 15 atividades e 13 artefatos gerados e caracteriza-se como um processo para a implementação de um sistema desenvolvido para funcionar em um Lego Mindistorms; compreende as atividades de montagem do modelo e contempla as fases de planejamento, implementação e teste.



**Figura 07 – Exemplo de Processo em BPMN**  
Fonte: Autor

## 2.2 DESENVOLVIMENTO DISTRIBUÍDO DE SOFTWARE

Fortes investimentos com o intuito de tornar mercados nacionais em mercados globalizados, fizeram com que fossem criadas novas formas de competição e também de colaboração entre empresas que podem até estar situadas em países diferentes (PRESSMAN, 2013). O mercado de software vinha passando por crise de credibilidade, devido ao elevado número de falhas em projetos, também uma crise de demanda crescente, principalmente pela escassez de recursos capacitados (PRESSMAN, 2013).

Nesse ambiente global de dificuldades, muitas organizações encontraram no DDS uma alternativa, desenvolvendo software com equipes geograficamente distantes entre si. Hoje, existem diversas organizações utilizando-se de DDS para aumentar a produtividade e diminuir custos, ou como uma alternativa para a escassez de mão-de-obra especializada (HERBSLEB, 2007). Segundo L'Erario (2009) o DDS ocorre quando vários sites cooperam e/ou colaboram para desenvolver um mesmo produto ou parte dele. Neste cenário, a complexidade do processo de produção se amplia.

O DDS tem sido marcado principalmente pela colaboração e cooperação entre departamentos de organizações e grupos de pessoas que trabalham em um mesmo projeto, ou com um mesmo objetivo, cujos participantes estão sediados, distantes fisicamente, em cidades ou até países diferentes, com diversidades culturais acentuadas e muitas vezes falando línguas distintas (L'ERARIO, 2009). Dependendo do amadurecimento de cada organização, esta possui o seu método próprio, que é relacionado com o tipo de atividade realizada pelas equipes distribuídas (codificação, testes, especificação) (L'ERARIO, 2009).

O DDS está revolucionando o mercado pela maneira como os produtos de software têm sido modelados, construídos, testados e entregues para os clientes. Além disso, DDS é um dos maiores desafios que o ambiente de negócios apresenta do ponto de vista do processo de produção de software (FUGGETTA; NITTO, 2014).

O DDS agrega várias vantagens sobre o processo centralizado, como discorrem Battin et al (2001), Carmel e Argawall (2001) e L'Erario (2004) em seus estudos de caso. Segundo Battin et al (2001), foi possível desenvolver 511000 linhas de

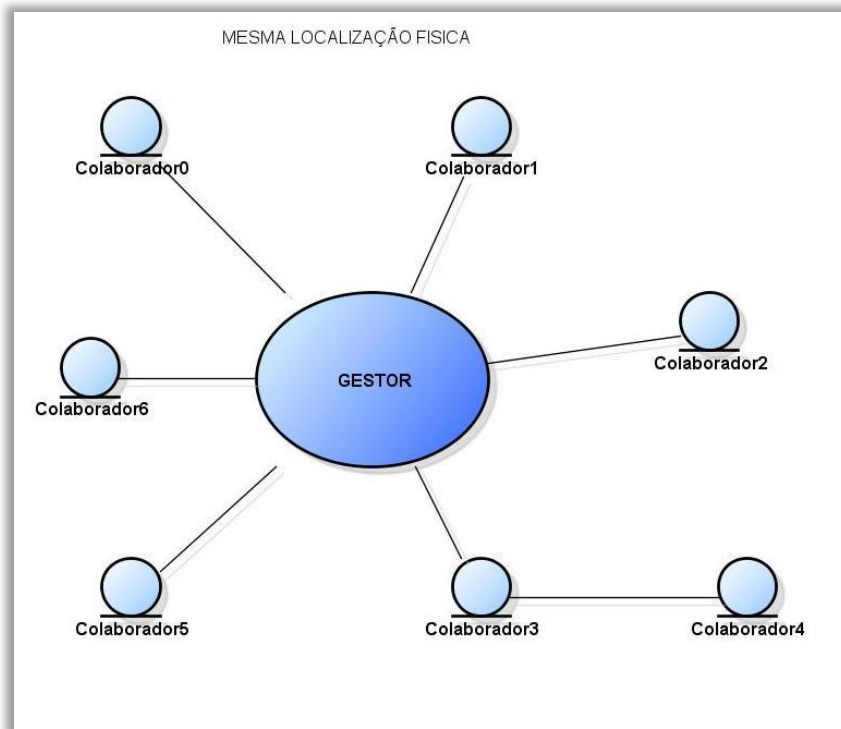
código para o projeto e cada centro de desenvolvimento colaborou significativamente para a implementação.

Carmel e Argawall (2001) afirmam que o número de organizações que aderem ao DDS é crescente. Tais organizações são empresas que distribuem filiais ou efetuam *offshore* com o objetivo de aumentar a produtividade, reduzir o custo e tornar a distribuição do software mais compatível com características locais. Neste estudo os autores relatam casos de 5 empresas que tiveram bons resultados utilizando-se do ambiente distribuído.

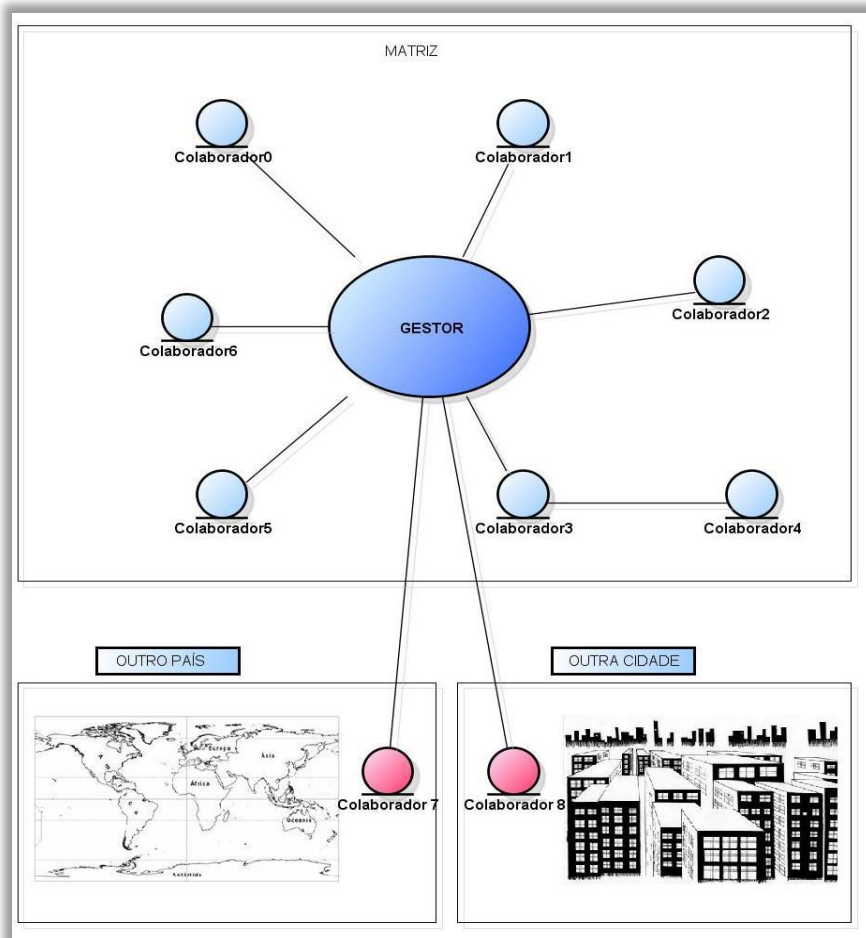
L'Erario (2004), em estudo de caso, utilizou o desenvolvimento distribuído para a implementação de uma aplicação pervasiva. Para tanto, foi elaborado um arcabouço de distribuição de tarefas a fim de coordenar a entrega de partes da aplicação pelos nós.

Em todos estes casos o DDS resultou em sucesso para o processo de produção de software, por isso ele tem sido um seguimento bastante estudado na área de engenharia de software. Os profissionais desta área reconhecem a enorme influência da nova forma de trabalho exigida pelo mercado, e isto faz com que se busquem modelos para que se avance na produção de software com equipes geograficamente distantes (PRIKLADNICKI; AUDY, 2010).

São contínuos os esforços de pesquisadores no intuito de mapear e compreender os fatores determinantes do sucesso desta forma de trabalho em organizações multinacionais ou virtuais. Os desafios da produção de software não são pequenos, e acrescentar as estas dispersão geográfica e temporal ampliou-os ainda mais (KIEL, 2003). Nas Figuras 08 e 09 estão ilustrados os cenários da produção co-localizada e distribuída respectivamente (PRIKLADNICKI, 2003).



**Figura 08 – Produção de Software Centralizado**  
 Fonte: Adaptado de Prikladnicki (2003)



**Figura 09 - Produção Distribuído de Software**  
 Fonte: Adaptado de Prikladnicki (2003)

Como já citado não há apenas uma razão que leva empresas a distribuir seus processos de produção de software; existem diversas. Essas razões, ou um subconjunto delas, motivam um crescente número de organizações a utilizarem DDS. Prikladnicki (2003) cita como principais razões para uso do DDS a demanda e custos, *time-to-market*, mercado e presença global, rigor e experiência no processo de software, sinergia cultural e escala. Na Figura 10 estão identificadas as principais razões que levam ao DDS.



**Figura 10 - Principais razões que levam ao DDS**  
 Fonte: Adaptado de Lopes (2004)

- **Demanda e Custos:** A necessidade de mão de obra não tem sido suprida pelo mercado de trabalho e a falta desta mão de obra especializada faz com que os custos cresçam. Assim, com colaboradores em falta, custos de contratação altos e a possibilidade de uso de profissionais de custo mais barato em outras localidades, o DDS tornou-se um grande atrativo para as empresas de software.

- **Time-to-market:** As pressões para reduzir o tempo necessário para colocar um produto em mercado (*time-to-market*) também auxiliam no crescimento do DDS. A possibilidade de produção de software *follow-the-sun*, em que equipes distribuídas ao redor do globo trabalham 24 horas por dia, é um grande atrativo para empresas que visam reduzir o *time-to-market*.

- **Mercado e presença global:** O mercado globalizado de produção de software é uma realidade, principalmente pelo fato de que os custos computacionais reduzem, surgindo assim uma grande demanda por soluções de sof-

ware. Então, a opção pelo DDS para que as empresas atinjam o mercado global e fiquem próximas aos seus mercados consumidores é um diferencial para a competitividade.

- **Rigor e experiência no processo de software:** Com o problema da distância dos desenvolvedores, o cuidado com as metodologias de produção de software é o mais realizado, em contradição à utilização de mecanismos informais. Há a especialização de algumas localidades em áreas pouco exploradas. Nestes casos, a experiência de um local específico pode ser um diferencial para que a produção de software seja realizada nele.

- **Sinergia cultural:** A diversidade amplia a criatividade e a inspiração na organização que produz software. Uma equipe globalizada cria uma sinergia cultural, encontrando novas formas de resolver problemas, projetar produtos, ou pensar sobre os processos.

- **Escala:** Grandes centros de produção de software são difíceis de gerenciar quando ficam muito grandes. Com isso, a distribuição é importante para atender a demanda necessária.

De acordo com Carmel (1999) as principais características que diferenciam o DDS do desenvolvimento centralizado são a dispersão geográfica (a distância entre equipe de projeto, clientes e usuários, por exemplo); a dispersão temporal (diferenças de fuso-horário); diferenças culturais (incluindo idioma, tradições, costumes, normas e comportamento). Estas diferenças refletem-se em diversos aspectos, dentre eles, destacam-se algumas questões citadas primeiramente por Herbsleb e Mockus (2003) e resistemizadas por Prikladnicki e Audy (2010):

- **Estratégicas:** envolvem a decisão de distribuir ou não o projeto e para qual unidade distribuída o projeto será enviado, tendo por base análises de risco e custo benefício envolvidos;

- **Culturais:** envolvem as diferenças culturais entre as equipes distribuídas, localizadas em diferentes regiões;

- **Técnicas:** envolvem aspectos relativos à compatibilização da infraestrutura tecnológica (redes de comunicação de dados, plataformas de hardware, ambiente de software) e ao conhecimento técnico necessário para o desenvolvimento dos projetos distribuídos;

- **Gestão do conhecimento:** envolvem aspectos relativos à criação, armazenamento, processamento e compartilhamento de informações nos projetos distribuídos.

### 2.2.1 Modelos de Negócios em DDS

O DDS é uma das consequências da globalização, com esta as empresas podem organizar-se de maneira distribuída e, por isso, podem também estudar melhor os meios para otimizar a produção distribuída de software. Dessa forma, um projeto de DDS pode ser resultado da distribuição de tarefas em um mesmo país/região ou até mesmo entre países diferentes, caracterizando o desenvolvimento global de software (CARMEL, 1999).

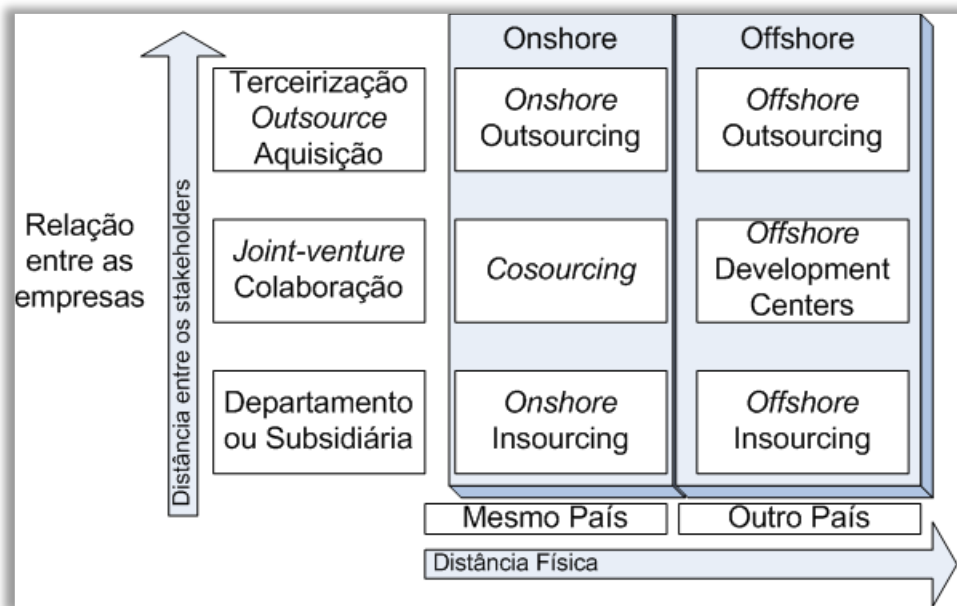
O relacionamento das empresas participantes de um projeto de DDS é uma característica que precisa ser definida, pois o resultado dessa relação é fortemente influenciado pela localização geográfica dos sites (L'ERARIO, 2009).

A soma da união entre distância física e a relação entre as empresas contribui para consolidação de um modelo de negócio dos projetos de DDS. A relação entre as empresas tem ocorrido de três formas principais, segundo Robinson (2004):

- **Terceirização (*outsourcing*) ou aquisição:** A empresa delega o controle sobre uma ou mais de suas atividades para uma empresa externa.
- **Joint-venture ou colaboração:** É um acordo entre duas ou mais empresas, que, com a união de recursos, executam um ou mais projetos por um determinado período de tempo.
- **Departamentos / subsidiárias da empresa (*insourcing*):** A empresa cria seu próprio centro que produz de software.
  - Do ponto de vista de distância geográfica, a distribuição ocorre de duas formas, segundo Robinson (2004):
    - **Offshore:** Este cenário é caracterizado pelos diferentes países onde residem os *stakeholders* de um mesmo projeto. O cliente, e até mesmo os desenvolvedores do software (codificadores, administradores de banco de dados, arquitetos e outros), podem residir em países distantes.

• **Onshore:** É considerado cenário de *onshore* quando todos os *stakeholders* de um mesmo projeto residem em um mesmo país. Podem ocorrer duas situações de produção: *onsite* e *offsite*. Na primeira situação o cliente e desenvolvedor estão fisicamente no mesmo espaço físico. Na segunda, não.

Buscando uma visão unificada das formas de relacionamento entre as empresas e a distribuição geográfica, a Figura 11 apresenta o modelo adaptado de Robinson (2004) por Prikladnicki (2006).



**Figura 11 - Relação entre as empresas e distância física**  
Fonte: Adaptado de Prikladnicki (2006)

Os modelos apresentados na Figura 11 são assim definidos Prikladnicki (2006):

- **Onshore insourcing ou demanda doméstica interna:** Esse modelo ocorre quando a empresa institui, no mesmo país ou em outros, um departamento ou subsidiária interna atendendo às suas próprias demandas de software.
- **Onshore outsourcing ou outsourcing:** Nesse modelo a empresa terceiriza, uma ou mais atividades, para outra empresa localizada no mesmo país.
- **Offshore outsourcing ou offshoring:** A empresa terceiriza uma ou mais atividades para outra empresa, localizada em outro país.



- **Offshore insourcing ou captive/internal offshoring:** criação de uma subsidiária da própria empresa para prover serviços de produção de software (*insourcing*). Essa subsidiária está necessariamente localizada em um país diferente da matriz da empresa, ou empresa contratante (*offshore*).
- **Cosourcing:** parceria que tem o objetivo de compartilhar a realização de atividades entre diversas empresas. É caracterizado pelo *jointventure*.
- **Offshore development centers:** centros especializados em trabalhos de offshore. São organizações que oferecem serviços de produção em um âmbito global.

### 2.3 DESAFIOS ENFRENTADOS PELAS EQUIPES DISTRIBUÍDAS

Por muitas razões tecnológicas e econômicas os projetos necessitam estar globalmente distribuídos (PRIKLADNICKI; AUDY, 2010). Sengupta et al. (2006) apresenta como dificuldade básica no DDS a inabilidade de controlar o processo e se comunicar efetivamente devido à distância geográfica, diferenças culturais e fuso horário. Estes problemas podem ser identificados em produção de software co-aloado, mas no DDS se acentuam.

A produção software sempre se apresentou de forma complexa. Existe uma série de problemas e desafios inerentes ao processo. O DDS, ao acrescentar fatores como dispersão física, distância temporal e diferenças culturais, acentuou alguns dos desafios existentes e acrescentou novos desafios ao processo de produção (PRIKLADNICKI; AUDY, 2010).

A seguir serão apresentados os principais desafios enfrentados pelas equipes distribuídas, descritos principalmente segundo a visão de Woodward et al. (2010), complementadas por alguns autores que abordam o tema. No Quadro 01 está a compilação destes desafios e as questões envolvidas.

<b>DESAFIOS</b>	<b>QUESTÕES ENVOLVIDAS</b>
<b>Culturais</b>	Liderança
	Formação de grupos
	Estilo de comunicação
	Resolução de problemas
	Perspectiva de tempo
<b>Dispersão geográfica</b>	Gerenciamento
	Fusos Horários
	Legislação
<b>Coordenação e controle</b>	Interdependência
	Gerenciamento de Configuração
	Consciência
<b>Comunicação</b>	Contexto
	Meio
<b>Espírito de equipe</b>	Coesão
	Confiança
	Tamanho

**Quadro 01 – Desafios do DDS.**  
**Fonte: Adaptado de Woodward et al. (2010)**

### 2.3.1 As diferenças culturais

O gerenciamento da diversidade cultural é fundamental para a efetividade de uma equipe distribuída, principalmente em âmbito global. A cultura, consciente ou inconscientemente, forma os valores, percepções e o comportamento de um grupo, bem como de cada um de seus membros. Culturas diferem-se em muitas dimensões críticas, como a necessidade de estrutura, atitudes com relação à hierarquia, senso de tempo e estilos de comunicação (HERBSLEB, 2007).

Questões como liderança, por exemplo, costumam fazer com que o processo decisório em equipes distribuídas seja mais demorado ou complexo. Líderes de algumas culturas costumam tratar a equipe de forma participativa e democrática, encorajando a manifestação de opiniões. Em contraste, líderes de outras culturas podem tratar a liderança de forma mais hierárquica, tomando decisões sem consultar os subordinados (MARQUARDT, 2001).

Problemas relacionados à formação de grupos também são comuns em ambientes de DDS. Enquanto algumas culturas valorizam a independência e os direitos individuais, outras são coletivistas, subordinando os interesses do indivíduo para o bem do grupo (MARQUARDT, 2001).

A comunicação entre membros de um grupo pode ser praticada de diversas formas, dependendo de sua cultura. O estilo de comunicação pessoal, por exemplo, pode variar de expressivo à instrumental (MARQUARDT, 2001). Um estilo de comunicação *expressivo* preocupa-se com o estabelecimento e manutenção de conexões pessoais e sociais, deixando a precisão da comunicação em segundo plano. As pessoas são encorajadas a expressar emoções. Abraços e toques são comuns.

Em comunicações de estilo instrumental, a preocupação maior é com o que está sendo dito. Este estilo de comunicação é mais impessoal, centrado em problemas e orientado a objetivos. A precisão é mais importante que um formato apropriado.

A resolução de problemas também pode ser vista de forma diferente de acordo com a cultura; pode ser linear, onde problemas são dissecados em pequenas partes ligadas em cadeias de causa e efeito; ou sistêmico, onde problemas são vistos de forma mais holística, com relacionamentos e caos como componentes comuns.

Além disso, a cultura difere na forma com que as soluções são feitas. Algumas culturas buscam soluções genéricas, com a aplicação consistente de regras, procedimentos e generalização (HERBSLEB, 2007). Outras culturas dão ênfase à diferenças, relativismo e individualidade. A solução depende totalmente da situação e do contexto.

Diversas outras diferenças culturais podem afetar o DDS, como a forma de tratar conflitos, tempo, flexibilização, entre outros. Todos estes fatores devem ser cuidadosamente analisados, de forma a minimizar seu impacto no andamento dos projetos.

### 2.3.1 Dispersão geográfica

A tarefa de gerenciamento de grupos adquire maior complexidade com a dispersão. A distância física e psicológica é considerada a segunda maior dificuldade em equipes distribuídas. Embora a tecnologia permita que as organizações optem pela descentralização, ainda existem diversas limitações devido ao reduzido contato face a face entre os membros da equipe e entre a equipe e seus contatos externos (HERBSLEB, 2007).

Obter consenso e aprovação na tomada de decisões pode ser mais difícil, e encontrar membros distantes do projeto, especialmente quando trabalham para diferentes organizações, e obter aprovação adequada de todos os locais envolvidos pode consumir bastante tempo (HERBSLEB, 2007).

Prikladnicki e Audy (2006) classificam a dispersão geográfica em nacional, continental e global. A variação da dispersão física influencia fortemente a possibilidade de reuniões presenciais e, principalmente, o horário de trabalho das equipes. Quando a dispersão física for nacional ou regional, há possibilidade de reunião presencial ou videoconferência. Isso é possível por não haver diferença no idioma dos *stakeholders*. Além disso, o fuso horário é praticamente o mesmo para todos os envolvidos (L'ERARIO, 2009).

Na dispersão continental, além do idioma, que não é mais necessariamente o mesmo, o fuso horário tem uma pequena influência. Nesse cenário, efetuar reuniões presenciais pode representar aumento no custo do projeto, uma vez que o custo (monetário e tempo gasto) do deslocamento entre os *stakeholders* pode ser elevado (L'ERARIO, 2009).

Quando a dispersão é global, pode ser inviável reunir presencialmente todos os integrantes do projeto. Nesse caso, o idioma é diferente, a cultura e o fuso horário também. Dentre outros problemas, a religião, a economia local e o fuso horário emergem como diferença entre os integrantes e agregam problemas ao projeto (L'ERARIO, 2009).

Segundo L'Erario (2009) outras questões importantes dizem respeito às diferenças legais entre os locais envolvidos (principalmente em casos de distribuição global). Essas diferenças implicam em maior complexidade no estabelecimento de contratos, bem como na necessidade de cautela com questões de sigilo e proprie-

dade intelectual. A co-localização e a dispersão estão ligadas ao antigo paradigma da centralização e descentralização das organizações. A centralização concentra a tomada de decisão em um único ponto da organização.

De acordo com antigas noções de gerenciamento, a descentralização da autoridade leva a perda de controle e aumento de custos, por exemplo, em telecomunicações. Com novas tecnologias e redução nos custos, estes problemas foram amenizados, mas não removidos (MARQUARDT, 2001).

### 2.3.2 Coordenação e controle

Segundo Cataldo (2007) a habilidade de uma organização executar prosperamente suas tarefas depende da combinação apropriada da estrutura organizacional, processos, comunicação e mecanismos de coordenação. Coordenação e controle têm ainda maior importância em equipes distribuídas.

As dificuldades e desafios na coordenação e controle em ambientes distribuídos são ampliados devido aos problemas de cultura, língua e tecnologia (HERBSLEB, 2007). Equipes distribuídas apresentam dificuldades nos mecanismos de coordenação e controle, principalmente os informais. Devido à distância, o gerenciamento não pode ser feito por observação enquanto se caminha pelo local de trabalho, ou reunindo a equipe em reuniões informais (CARMEL, 1999).

O grau de dependência das tarefas exerce um papel fundamental na coordenação. Quando dois membros de uma equipe, com tarefas fortemente relacionadas, colaboram, a diferença de fuso horário pode dificultar a coordenação. Frequentemente as requisições não são claras, necessitando de novas comunicações (L'ERARIO, 2009).

Quando membros da equipe estão trabalhando face a face, a visibilidade necessária do processo de trabalho pode ser obtida quase instantaneamente. Entretanto, quando os membros da equipe estão distantes, esta visão pode ser demorada (L'ERARIO, 2009).

Outro fator que torna a coordenação e o controle mais difíceis é o tamanho das equipes distribuídas. Uma equipe distribuída é, em geral, maior, para uma mes-

ma tarefa, que equipes co-localizadas. Quanto mais pessoas e papéis envolvidos, mais desafiadores os problemas de coordenação e controle (MARQUARDT, 2001).

O gerenciamento da configuração de software (*Software Configuration Management* - SCM) também apresenta novos desafios. Controlar modificações nos artefatos em cada uma das localidades e coordenar o processo de modificação destas localidades com o processo de produção pode ser bastante complexo.

A consciência das atividades desenvolvidas também apresenta desafios ao DDS. No processo de produção de software é importante que se tenha consciência sobre o que está acontecendo, quem está realizando e onde está acontecendo, principalmente sobre dois itens: documentação e código (CARMEL, 1999). Participantes necessitam saber os resultados dos outros, de forma a ampliar a colaboração.

#### 2.3.4 Comunicação

A dispersão geográfica tem impacto direto em todas as formas de comunicação. A comunicação informal é reduzida drasticamente. As pessoas deixam de se comunicar devido às dificuldades impostas. Mais reuniões são necessárias, e a complexidade de coordená-las torna-se maior.

Comunicação clara e efetiva é absolutamente essencial para o sucesso de equipes distribuídas. Entretanto, em muitos casos é necessário comunicar-se indiretamente (devido à distância temporal), prejudicando a riqueza de contexto (MARQUARDT, 2001). Comunicações podem variar de alto a baixo contexto, e diferentes culturas comunicam de forma diferente de acordo com a necessidade de contexto.

Culturas de comunicação de alto contexto (ou ricas em contexto) têm a habilidade de compartilhar experiências e tornar inteligíveis determinadas mensagens sem precisar declarar explicitamente. Regras de comportamento e expressão estão implícitas no contexto. O sentido de uma frase não está somente nas palavras. Ele pode estar no tom de voz, linguagem corporal, expressões faciais, contato visual ou uso do silêncio, por exemplo. O significado tende a ser implícito, indireto, não literal (MARQUARDT, 2001).

Na comunicação de alto contexto a maioria da informação é entendida através do contexto físico e social. Culturas de alto contexto não respondem apenas verbalmente, mas utilizam todos os meios de interação para isso. Essas culturas são encontradas tipicamente no Oriente Médio, Ásia, África e América do Sul.

Culturas de comunicação de baixo contexto, por outro lado, enfatizam a troca de fatos e informações. A mensagem é mais importante que a forma. A informação está primariamente nas palavras, e o significado é explícito. Nessas culturas, videoconferência e e-mails são usualmente aceitos como substitutos eficientes para comunicações face a face (MARQUARDT, 2001). Estes raramente levam em conta o contexto físico e social para interpretar a notícia. Culturas de baixo contexto são encontradas normalmente na Europa Ocidental, Estados Unidos e Canadá.

Da mesma forma, os meios de comunicação podem ser classificados de acordo com o nível de interação. Meios de comunicação ricos permitem interações nos dois sentidos envolvendo mais de um canal sensorial. Dados os diferentes estágios do ciclo de produção de software, algumas tarefas necessitam de comunicação mais rica que outras.

O contato com o cliente deve ser o mais próximo possível durante a elicitação dos requisitos. Análise e projeto de software necessitam de meios que proporcionem uma colaboração bem realizada entre os participantes do processo. De forma geral, qualquer tarefa que necessite de intensa cooperação requer mais comunicação, e quanto mais rica tal comunicação, melhor (CARMEL, 1999).

Determinadas tarefas exigem muito cuidado na escolha do meio de comunicação a ser utilizado. Por exemplo, um gerente de DDS deve, regularmente, transmitir a visão de equipe para todos os grupos e culturas envolvidos, de forma que seja entendida (HERBSLEB, 2007).

Uma vez que a “visão” deve utilizar níveis altos de motivação e emoção, um meio de comunicação adequado a essas características deve ser utilizado. Alguns dos impactos da dispersão são: o uso excessivo de comunicação, falta de comprometimento e desconforto ao utilizar alguns meios (L'ERARIO, 2009).

### 2.3.4 Espírito de equipe

A palavra equipe, em geral, tende a subentender co-localização, homogeneidade cultural, confiança, padrões de comunicação, um pequeno número de membros e, o mais importante, coesão. Equipes são unidades sociais frágeis, que podem facilmente ser quebradas (MARQUARDT, 2001).

Quando apresentadas dificuldades como distância e diferenças culturais e de fuso horário, o espírito de equipe geralmente desaparece. O espírito de equipe é o efeito sinérgico que torna a equipe uma unidade coesa (CARMEL, 1999).

Compartilhar uma visão das metas e ações da equipe pode ser difícil, inclusive em ambientes com uma única cultura. Em ambientes globais e multiculturais é difícil até mesmo afirmar que todos os membros entendem o que é uma equipe. Algumas culturas têm pouca experiência com trabalho em equipe, em algumas línguas a palavra equipe nem mesmo existe (MARQUARDT, 2001).

A coesão é uma característica fundamental o sucesso das equipes. Equipes coesas têm maior motivação, moral e produtividade, além de melhor comunicação e satisfação com o trabalho. No DDS a coesão é mais difícil de ser obtida por diversos motivos, quais sejam, desconfiar de outras devido à formação de estereótipos incorretos, falta de comunicação externa ao grupo, ou mesmo pelos problemas causados pela diferença de linguagem (CARMEL, 1999).

A confiança, fundamental para o espírito de equipes, torna-se de difícil manutenção no DDS. A construção desse tipo de relacionamento é impedida pela distância (CARMEL, 1999). Equipes co-localizadas adquirem confiança uns nos outros através de encontros informais e face a face. Como esse tipo de encontro não pode ser realizado devido à distância, muitas organizações investem em viagens para integração da equipe. Além disso, equipes globais são, comumente, maiores que equipes co-localizadas, reduzindo o espírito de equipe.

A intimidade que surge da co-localização é perdida, e a complexidade da comunicação aumenta. O tamanho do grupo é importante para assegurar que a comunicação de todos os membros ocorra de forma efetiva. Quanto menor número de membros de uma equipe, menor o número de canais de comunicação necessários (PRIKLADNICKI, 2003).



## 2.4 MONITORAMENTO E CONTROLE

As atividades relacionadas ao monitoramento e controle mantêm importante papel para o sucesso de um projeto de software. Prova desse reconhecimento é a significativa parcela dedicada ao assunto entre as principais abordagens de gerência de projetos. O monitoramento está diretamente ligado à gestão deste que, por sua vez, não têm sido utilizado na sua totalidade, como verificado na pesquisa realizada pelo PMI (2013), em que constatou-se que 24% dos empreendimentos de software não adotam esta atividade em seu escopo.

Wangenheim (2009) afirma que monitorar significa capturar, analisar, reportar e comunicar o desempenho do projeto cruzando as medidas reais de andamento das atividades com as medidas previamente estimadas no planejamento, e que controlar significa tomar ações necessárias de correção para que as medidas reais se aproximem o máximo possível das medidas planejadas (WANGENHEIM 2009).

De acordo com Pereira et al. (2010) o monitoramento do projeto é realizado por meio da medição do andamento das atividades, registrado em formulários de apontamentos ou sobre o próprio cronograma, diretamente sobre o papel ou na ferramenta. Essa medição também pode ser apurada individualmente com cada membro da equipe ou em reuniões de acompanhamento do projeto. Além do andamento, avaliam-se ocorrências relevantes, desvios, dificuldades, medidas corretivas, ânimo e moral das pessoas, riscos e a previsão da conclusão do projeto.

Pereira et al. (2010) também afirma que controle é a etapa que acontece concomitantemente à execução do projeto. Tem como objetivo monitorar e avaliar aquilo que está sendo realizado de modo a propor ações corretivas e preventivas no menor espaço de tempo possível após a detecção da anormalidade. O controle compara o status atual do projeto com o padrão previsto pelo planejamento.

O monitoramento e controle de projetos são necessários para coletar, medir e disseminar informações sobre o desempenho e avaliar as medições e as tendências para efetuar melhorias no processo. Algumas vantagens que podem ser apresentadas se ocorrer um processo de monitoramento e controle eficientes são: estabilidade do processo, procedimentos bem elaborados, eficiência, padronização, processo alinhado e melhor gestão (PEREIRA et al., 2010).

O monitoramento e o controle de projetos estão presentes no *Capability Maturity Model – Integration* (CMMI), *Control Objectives for Information and Related Technology* (COBIT), *Project Management Institute* (PMI) e *Project Management Body of Knowledge* (PMBOK) como processos relacionados. O monitoramento consiste na coleta, medição e distribuição das informações de desempenho; e a avaliação das medições e tendências para efetuar melhorias no processo, ações que passam a fazer sentido apenas quando ligadas ao processo de controle que inclui a determinação de ações corretivas ou preventivas, ou o replanejamento e acompanhamento dos planos de ação, para definir se as atitudes tomadas resolveram a questão de desempenho (PMI, 2013).

Apesar do monitoramento e do controle serem processos relacionados, este trabalho mantém o foco apenas no monitoramento. Nas bases teóricas investigadas estes são termos mantêm correlação. Contudo, esta base teórica se tratar dos dois, o escopo deste trabalho pretende centralizar esforços unicamente no processo de monitoramento de projetos.

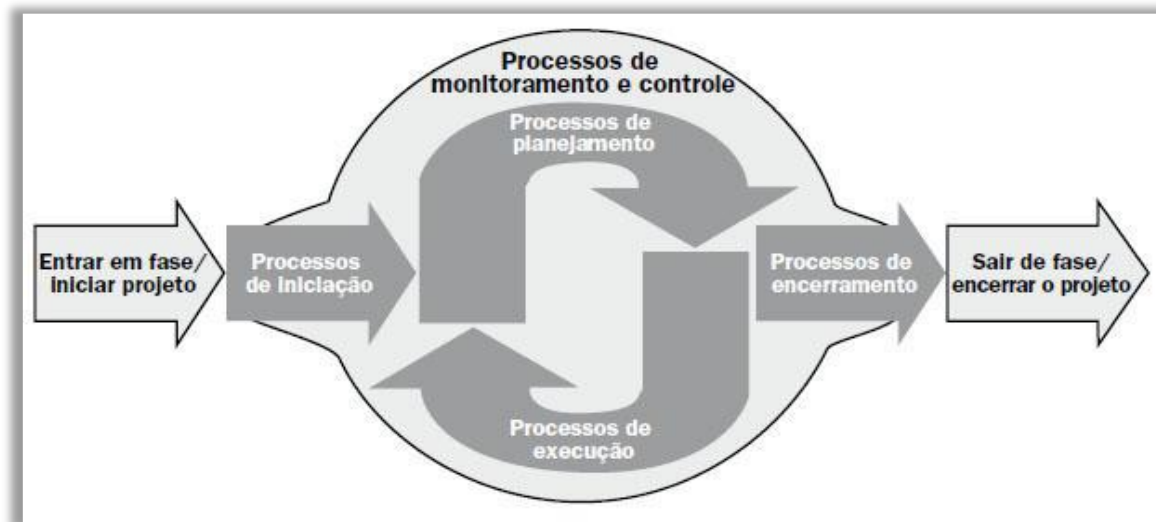
Ao analisar as principais abordagens de gerência de projetos, nota-se que boa parte de suas composições trata de monitoramento e controle. O PMBOK (2014) apresenta 10 processos de monitoramento e controle de um total de 42 processos propostos. O *Association of Project Management Body of Knowledge* (APMBOK, 2000) insere planejamento, medição, monitoramento e ações corretivas em um grande ciclo de controle.

Levando em consideração que muitos dos problemas em projetos de software referem-se a monitoramento e controle, é fácil entender a razão das significantes parcelas dedicadas a monitoramento e controle das abordagens de gerência de projetos, segundo Jones (2006).

Miranda e Bourque (2010) afirmam que o monitoramento é função básica na execução de qualquer projeto, pois serve para alertar possíveis problemas antes que eles se tornem irreversíveis. Para Rozenes et al. (2006), o controle é uma questão importante durante todo o ciclo de vida do projeto, pois a execução de acordo com o planejado depende de uma metodologia de controle. Segundo Avison et al. (2001), o desempenho de um projeto pode ser significativamente melhorado se mais atenção for dada ao monitoramento e ao controle.

No decorrer de um projeto de software existe um grupo de processos que está relacionado ao seu ciclo de vida. Esses processos permitem que sejam identifi-

cadaver diversas familiaridades que podem ser encontradas em todos os projetos, independentemente de sua natureza. Os grupos de processos podem ser visualizados na Figura 12.



**Figura 12 - Grupos de Processos de gerenciamento de projetos**  
 Fonte: Adaptado de PMI (2013)

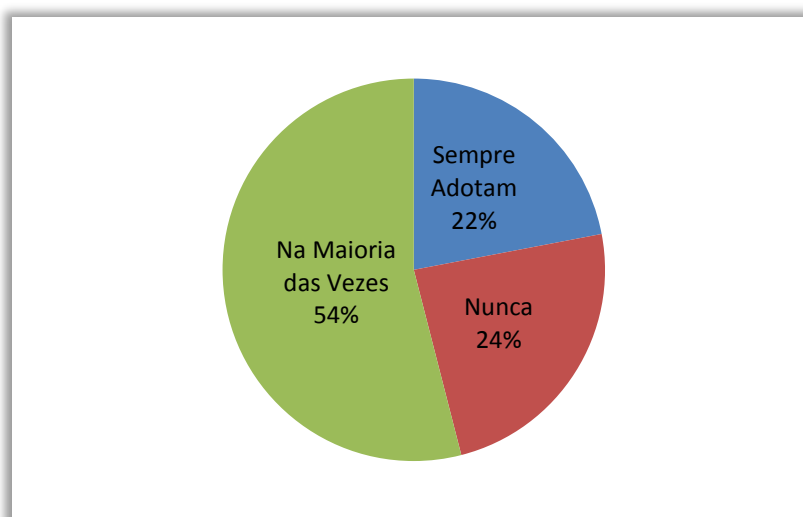
Pode ser observado na Figura 11 que os processos de monitoramento e controle são atividades que englobam praticamente todo o ciclo de vida do projeto. Segue uma breve descrição sobre cada processo segundo o PMI (2013):

- **Processo de inicialização:** São os processos realizados para definir um novo projeto ou fase em um projeto já existente, por meio da obtenção de autorização para iniciar o projeto ou fase.
- **Processo de planejamento:** São os processos realizados para definir o escopo do projeto, refinar os objetivos e desenvolver o curso da ação necessários para alcançar os objetivos para os quais o projeto foi criado.
- **Processo de execução:** São os processos realizados para executar o trabalho definido no plano de gerenciamento do projeto para satisfazer as especificações do mesmo.
- **Processos de monitoramento e controle:** São os processos necessários para acompanhar, revisar e regular o progresso e o desempenho do projeto, identificar todas as áreas nas quais serão necessárias mudanças no plano e iniciar as mudanças correspondentes.
- **Processo de encerramento:** É executado para finalizar todas as atividades de todos os grupos de processos, visando encerrar formalmente o projeto ou fase.

No capítulo 1 foi explicitado que o foco deste trabalho é o processo de monitoramento essencial em um projeto de desenvolvimento de software. Este tem como objetivo fornecer o entendimento do progresso do projeto para que ações corretivas possam ser tomadas quando o projeto estiver desviando significativamente do seu plano (JALOTE, 2000) (HANAKAWA; KIMIHARU, 2004) e (SEI, 2010).

Monitoramento em projetos software são complexos, em comparação a outros tipos de produtos, devido à natureza abstrata que envolve a produção de software (HAUCK, 2007). Frequentemente projetos de software falham porque eles não são devidamente monitorados, a ponto de uma falha se tornar visível, antes que seja tarde demais para tomar uma ação corretiva (WANGENHEIM 2009).

Em pesquisa o PMI revelou a frequência com que as empresas brasileiras, que responderam o questionário, adotam atividades de monitoramento e controle em seus projetos como pode ser observado no gráfico 01.



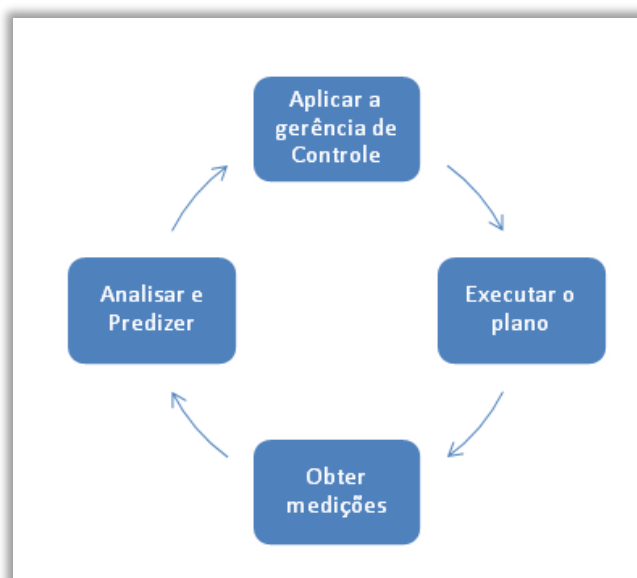
**Gráfico 01 – Frequência de Atividades de Controle em Projetos de Empresa Brasileiras**  
Fonte: Adaptado PMI (2013)

Esses dados não se diferem muito do cenário mundial e ajudam a explicar porque 65,7% dos problemas mais frequentes enfrentados pelas empresas estão relacionados ao prazo, 61,7% ao escopo e 41,3% ao orçamento. Como é reforçado por Jones (2006) entre as principais causas de problemas em projetos de software estão as estimativas inadequadas, relato de status inadequado, ausência de dados históricos de projetos similares, e processos de controle de qualidade inadequados.

É interessante citar ainda resultados do estudo desenvolvido pelo PMI (2013) que revelam que 18,4% dos problemas são relacionados à falta de um me-

canismo de apoio e que 24,5% das ferramentas mais utilizadas são desenvolvidas internamente, o que significa que as empresas não conseguem encontrar uma forma de monitorar e/ou controlar os projetos de software, que seja adequada para as suas necessidades (PMI 2013). Segundo Jalote (2000) um dos primeiros passos a serem tomados antes de iniciar os processos de monitoramento e controle é selecionar quais são os dados que realmente são relevantes dentro do contexto do projeto.

Depois de selecionado os dados, o monitoramento e o controle ingressam em um ciclo que é seguido durante toda a existência do projeto e em todas as fases e áreas de conhecimento. Na Figura 13 está ilustrado o modelo do ciclo de monitoramento e controle proposto por Jalote (2000).



**Figura 13 - Ciclo monitoramento e controle**  
Fonte: Adaptado de Jalote (2000)

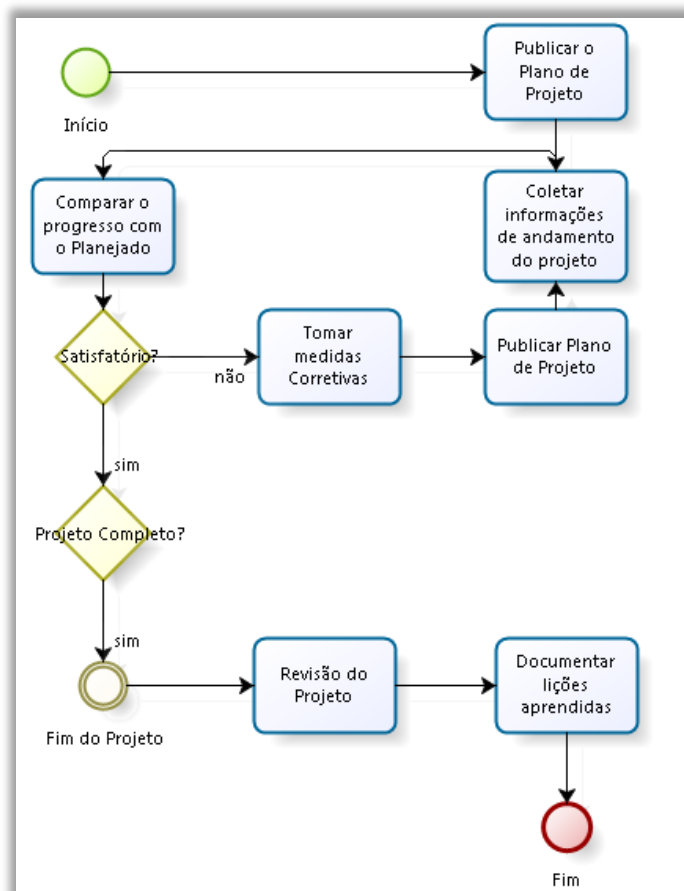
Dentre algumas das atividades desempenhadas por esses processos pode-se destacar (PMI 2013):

- Comparação entre o andamento real do projeto e o planejado;
- Avaliação de desempenho para determinar atividades necessárias de correção ou prevenção;
- Identificação, análise e acompanhamento de novos riscos e monitoramento de riscos já existentes, aplicando os critérios definidos no planejamento de riscos para tratar de forma adequada todos eles;
- Manter uma base de informações completa sobre os produtos do projeto e os documentos referentes a eles;

- Disponibilizar informações sobre andamento, medição do progresso e previsão;
- Fornecer previsões para atualizações de informações do projeto;
- Monitorar a execução de atividades aprovadas.

Monitorar e controlar envolve medir os dados atuais do projeto e comparar as medidas com o planejamento feito para cada etapa (SEI 2010). Caso as atividades monitoradas fiquem fora dos limites estabelecidos, medidas corretivas devem ser executadas (HUGHES 2001), (SEI 2010).

A Figura 14 ilustra o modelo de um projeto de monitoramento e controle e mostra como, uma vez que o plano inicial tenha sido publicado, o monitoramento e o controle são contínuos durante todo o ciclo de vida. Além disso, mostra também atividades importantes que devem ser executadas ao final do projeto, como a definição de lições aprendidas.



**Figura 14 - Fluxo de atividades de monitoramento e controle**  
 Fonte: Adaptado HUGHES (2001)

A principal ferramenta para monitorar e controlar o trabalho no projeto é a opinião especializada (PMI 2013). Nesse caso a experiência do gerente e de sua equipe são fatores importantes para se obter uma leitura precisa do mesmo; porém se este contar com o auxílio de um padrão, ou um modelo que lhe auxilie pode reduzir o tempo com ações frustradas, a utilização de uma metodologia ou de um sistema de informações para gerenciamento de projetos são recursos sugeridos pelo PMI (2013) para monitorar e controlar o trabalho.

Existem também técnicas que podem ser adotadas no processo de medição a exemplo da técnica do valor agregado que, mede o desempenho do projeto conforme seu andamento (ZHU; YANG, 2010). Dessa forma, monitorar e controlar o projeto tem como objetivo colocá-lo novamente no caminho do que foi planejado ou adequar as demais atividades ao desvio identificado.

#### 2.4.1 Elementos Monitorados e Controlados

Os elementos que são monitorados e controlados no decorrer de um projeto, são questões fundamentais para o bom andamento do mesmo, como qualidade do produto, custo, risco, escopo, cronograma dentre outros citados no decorrer desta seção (PMBOK, 2014).

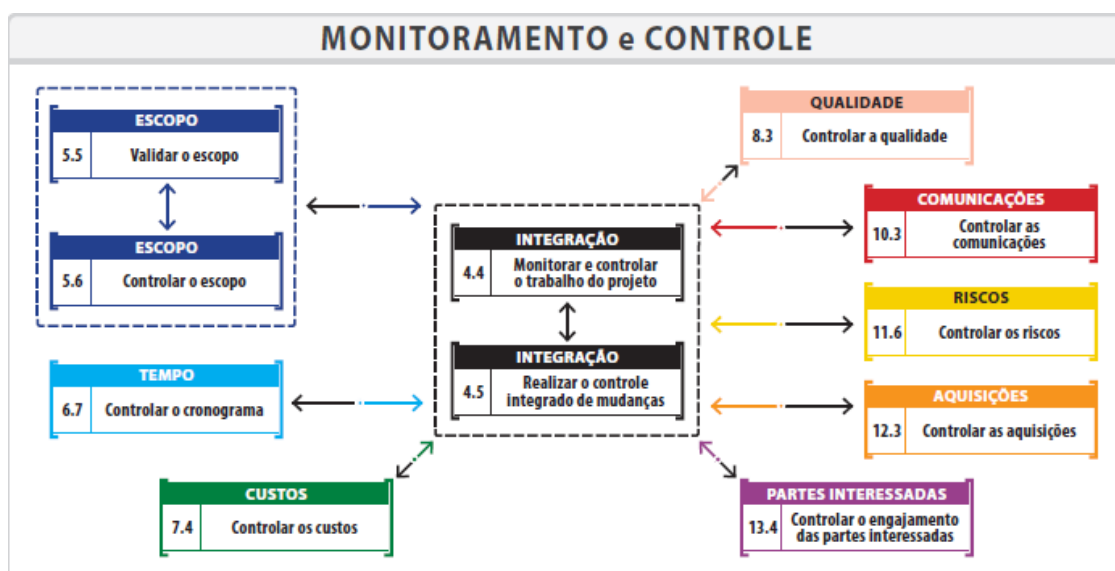
Estes elementos compõem o grupo dos processos de monitoramento e controle, que segundo o PMBOK (2014) são constituídos de forma a observar a execução do projeto; assim, possíveis problemas podem ser identificados no momento adequado a fim de que sejam tomadas ações corretivas, quando necessário, para controlar a execução do projeto.

A equipe deve determinar quais são os elementos necessários para o seu objetivo específico. O principal benefício de se monitorar e controlar é que o desempenho do projeto é observado e medido regularmente para identificar variações em relação ao plano de gerenciamento. Os elementos que compõe o grupo de monitoramento e controle também incluem o controle de mudanças e a recomendação de ações preventivas, antecipando possíveis problemas.

Em projetos com várias fases, o monitoramento e controle também fornecem *feedbacks* entre suas fases a fim de implementar ações corretivas ou preventivas

para assegurar a conformidade com o plano de gerenciamento do projeto. Quando as variações comprometerem os objetivos, os processos de gerenciamento adequados dentro destes elementos serão reexaminados como parte do ciclo PDCA (*Plan, Do, Check, Act* - Planejar, executar, checar e agir, em português) modificado. Essa revisão pode resultar em atualizações recomendáveis no plano de gerenciamento. Por exemplo, uma data de término de atividade não cumprida pode exigir um aumento na equipe atual, dependência de horas extras ou compensações entre os objetivos de orçamento e de cronograma.

Na Figura 15 estão indicadas algumas das interações essenciais para este grupo de processos, que, segundo PMBOK (2014), explicitam os principais elementos que são monitorados e controlados no decorrer de um projeto.



**Figura 15 – Grupo de processo de monitoramento e Controle**  
Fonte: Adaptado PMBOK (2014)

Existem diferentes normas e padrões que tratam de monitoramento e controle de projetos como: O RUP (*Rational Unified Process*), PMBOK (*Project Management Body of Knowledge*), MCGP (Metodologia de Gerenciamento de Projetos), Estruturada, CMMI (*Capability Maturity Model*), MPSBR (Melhoria de Processo de Software Brasileiro), ITIL (Information Technology Infrastructure Library), COBIT (Control Objectives for Information and related Technology), ISO 20.000, ISO 10.006, PRINCE2 (Projects IN a Controlled Environment), ISO 21.500, dentre outros.

Segundo PMI (2013) a finalidade do monitoramento e controle é determinar se o projeto está prosseguindo conforme planejado. Essa fase consiste em três etapas básicas:



- 1) Monitoramento das atividades contínuas do projeto;
- 2) Comparação das variáveis do projeto (custo, esforço, tempo, recursos, entre outros.) com o plano real (onde deve-se estar);
- 3) Identificação de ações corretivas (como volta-se ao caminho certo).

De acordo com as etapas definidas pelo PMI (2013) as abordagens que tratam destas três etapas básicas para o monitoramento de projetos dedicadas a software ou generalizadas, encontradas na literatura são o RUP, PMBOK, MCGP, Estruturada, CMMI e MPSBR, por isso estas foram selecionadas para a investigação dos elementos que são monitorados no decorrer de um projeto, e as demais foram descartadas. Estes elementos são descritos a seguir no Quadro 02, segundo as abordagens resumidamente definidas na sequência:

- **RUP:** Objetiva fornecer técnicas a serem seguidas pelos membros da equipe de desenvolvimento de software com o objetivo de aumentar a sua produtividade no processo de desenvolvimento.
- **PMBOK:** é um conjunto de práticas na gestão de projetos organizado pelo instituto PMI e considerado a base do conhecimento sobre gestão de projetos por profissionais da área.
- **MCGP:** O objetivo do monitoramento e controle no MCGP é do aprimoramento no gerenciamento do projeto, da medição e monitoramento de seu progresso para identificar variações em relação ao plano de gerenciamento do mesmo.
- **Estruturada:** a sua concepção está baseada no processo de software cascata, que por sua vez estabelece fases bem definidas tornando o monitoramento e controle como um conjunto de atividades aplicáveis.
- **CMMI:** Modelo de referência que contém práticas necessárias à maturidade na Engenharia de Software.
- **MPS/BR:** Modelo de qualidade de processo na engenharia de software focado em empresas brasileiras.

Vale a pena salientar que foram elencados apenas os elementos que constam na literatura como monitorados e controlados; quanto aos que constam com o termo gerenciado foram descartados, pois estes não remetem ao tema do trabalho.

Legenda: **M**: Monitora / **C**: Controla / **M/C**: Monitora e Controla o elemento.

ID	ELEMENTOS	Abordagem					
		RUP	PMBOK	MCGP	Estruturada	CMMI	MPS/BR
1	Indicadores do projeto	M					
2	Relatórios de status da equipe e do projeto	M/C					
3	Procedimentos para ação corretiva	C					
4	Procedimento para elaboração de relatórios de status do projeto	C					
5	Status do trabalho	M/C					
6	Indicadores de progresso e qualidade	M/C					
7	Indicadores versus planos	M/C					
8	Trabalho no projeto		M/C				
9	Integração		M/C				
10	Comunicações		M/C				
11	Escopo		M/C				
12	Cronograma		M/C				
13	Custos		C				
14	Qualidade		C				
15	Equipe do projeto		M				
16	Partes interessadas		M				
17	Aquisições		M				
18	Contrato		M				
19	Mudanças		M	M			
20	Desempenho		M/C	M/C			
21	Riscos		M/C	M/C			
22	Requisições		M	M			
23	Atividades do projeto			M/C	M/C		
24	Diagrama de marcos			M/C			
25	Lições aprendidas			M/C			
26	Projeto					M/C	M/C

**Quadro 02 – Elementos de Monitoramento e Controle**

Fonte: Autor

Cada elemento descrito no Quadro 02 tem importante papel no bom encaminhamento de um projeto; por isso estes devem ser adequados e a sua execução suscetível ao monitoramento e controle. Para um melhor entendimento dos elementos que compõem o Quadro 02 os mesmo são conceituados sucintamente na sequência de acordo com Ramos (2014):

- Indicadores do projeto: são marcos definidos: É um parâmetro que medirá a diferença entre a situação que se espera atingir e a situação atual;
- Relatórios de status da equipe e do projeto: Desenvolvido pelos colaboradores no projeto, é um documento que retrata o percentual de avanço no projeto;
- Procedimentos para ação corretiva: São as ações que devem ser executadas caso algo não ocorra como previsto;
- Procedimento para elaboração de relatórios de status do projeto: são normas para a elaboração do relatório status da equipe e do projeto;

- Status do trabalho: É um dado numérico ou descritivo que informa como está o projeto, considerando o que foi feito em relação com o que ainda precisa ser executado;
- Indicadores de progresso e qualidade: São parâmetros que medem, o progresso e a qualidade do produto;
- Indicadores versus planos: São comparações realizadas entre os indicadores e o planejado no projeto;
- Trabalho no projeto: É o executado pelos colaboradores no decorrer do projeto;
- Integração: Trata-se da fase em que os diversos artefatos produzidos no decorrer do projeto são agrupados;
- Comunicações: Trocas de informações entre os colaboradores e a gerência;
- Escopo: É o limite de, ou o que vai atingir uma determinada operação e dentro de qual recursos de sistema podem ser utilizados.
- Cronograma: Tempo limite em que se deve terminar o projeto, ou parte dele;
- Custos: Valor monetário da execução do projeto;
- Qualidade: É um atributo do processo executado e do projeto, definem como qualidade gerar o melhor produto possível de acordo com o que o cliente necessita, da melhor maneira possível;
- Equipe do projeto: Pessoa que colaboram para execução do projeto;
- Partes interessadas: Envolve – fornecedores, clientes, gerência, colaboradores e demais pessoas que possam estar envolvidas no projeto;
- Aquisições: Produtos e/ou serviços que possam ser adquiridos, como subsídios para execução do projeto;
- Contrato: Documento que descreve as necessidades do cliente, descrição do produto, custos do projeto e como a empresa entregará tal produto; contém prazos e critérios de qualidade, contudo depende muito de como este seja elaborado;
- Mudanças: Conforme o produto vai evoluindo, pode acontecer mudanças em requisitos; estas devem ser documentadas e acompanhadas;

- Desempenho: Descrição de como está o encaminhamento do projeto em termos de execução;
- Risco: Probabilidade de insucesso de determinado empreendimento, em função de acontecimentos eventuais e incertos, cuja ocorrência não depende exclusivamente da vontade dos interessados;
- Requisições: Documentos que fazem pedidos, entre a equipe, entre gerência e equipe, ou entre empresa e cliente;
- Atividades do projeto: São as ações que devem ser implementadas para que o projeto gere o produto final;
- Diagrama de marcos: É uma representação visual dos pontos significativos do projeto; eventos cuja ocorrência precisa ser reportada às partes interessadas com vistas a clara visibilidade do seu cumprimento.
- Lições aprendidas: São ações que foram realizadas e que oportunizaram aprendizagem aos colaboradores;
- Projeto: Descrição escrita e detalhada de um empreendimento a ser realizado; plano, delineamento, esquema.

Cabe destacar que, dentre os elementos passíveis de monitoramento elencados no Quadro 02 este trabalho foca-se no elemento 12 e 23, o cronograma e as atividades do projeto, realizando um monitoramento do cronograma das atividades do projeto, validando apenas o tempo para realização das mesmas. Desta forma, apenas os aspectos referentes a estes elementos foram monitorados, pelo processo estruturado.

#### 2.4.2 Processo de Monitoramento de Projetos Distribuídos de Software Preliminar

Com base na revisão bibliográfica apresentada foi possível elaborar um processo de monitoramento preliminar, disponível na íntegra em Souza et al. (2015). Nesta subseção será apresentada uma breve descrição deste, uma vez que foi a base para o desenvolvimento e especificação de um processo melhor sistematizado, a ser verificado no Capítulo 4 desta dissertação.

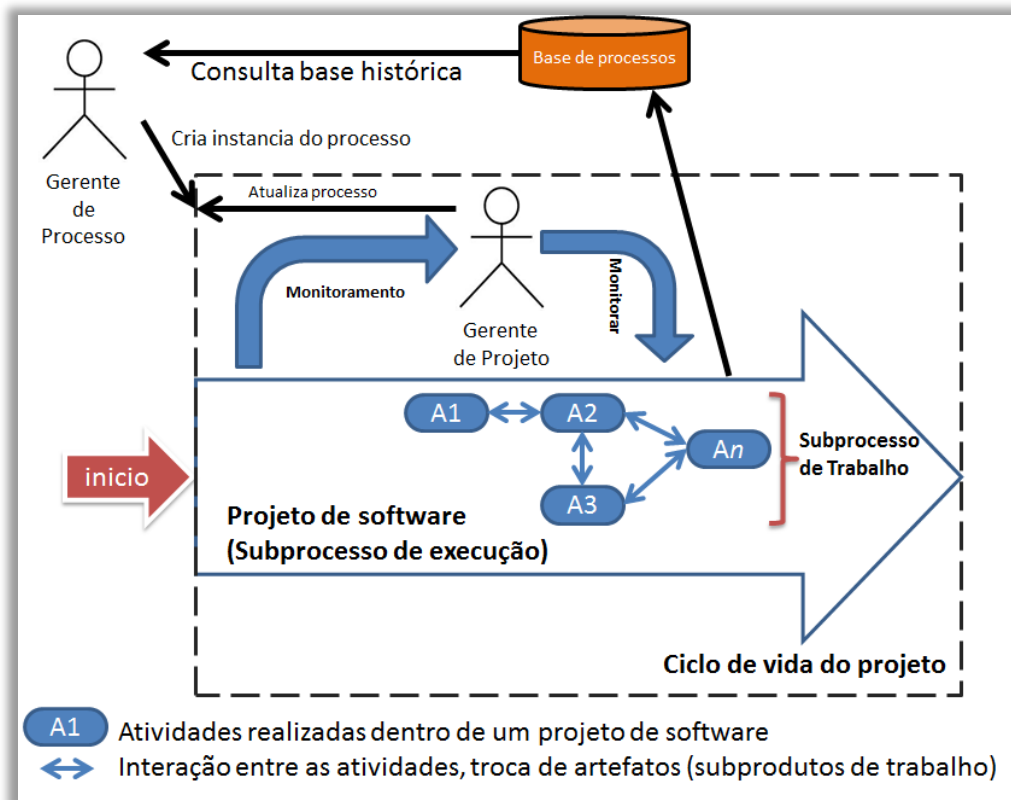
Segundo Souza et al. (2015) o processo de monitoramento para projetos distribuídos sistematiza uma forma para acompanhar a produção de software em ambientes deste tipo. Tal monitoramento exige a observância de três subprocessos interligados: o subprocesso de execução, o subprocesso de trabalho e o subprocesso de monitoramento.

O subprocesso de execução gera um produto final. Caracterizado como as diversas etapas necessárias (atividades) para se chegar ao objetivo do projeto; o subprocesso de trabalho é definido como uma unidade do subprocesso de execução (atividade junto ao artefato gerado); no subprocesso de monitoramento é o gerente de projetos o responsável pelo subprocesso de monitoramento, que visa aumentar a eficiência e a eficácia do projeto, por meio de um sistema que monitore os procedimentos efetuados durante o desenvolvimento de um produto (SOUZA, et al. 2015).

O projeto de software é iniciado com uma solicitação interna, ou externa, para um produto vinda de um cliente; ou uma necessidade da própria empresa. O processo de execução possui todas as atividades para elaboração do produto. Cada atividade (A1, A2, A3, An...) é interativa e pode necessitar de adequações ao longo do projeto, desta forma, elas não possuem uma linearidade de execução rígida. Algumas atividades necessitam de artefatos de entrada, então estas ficam aguardando até poderem ser iniciadas, caso o artefato de entrada ainda não esteja pronto.

O gerente de processos é responsável por alocar, criar, atualizar e consultar subprocessos de execução (desenvolvimento de software) na base de processo; este se comunica apenas com os processos, e com o gerente de projetos. Este último, tem como atribuição distribuir as responsabilidades aos colaboradores, gerenciar a expectativa do cliente e o mais importante, monitorar as atividades desenvolvidas pelos colaboradores.

O processo preliminar conforme Souza et al. (2015) representado por meio da Figura 16, mostra como o subprocesso de execução e o subprocesso de trabalho são envolvidos pelos subprocesso de monitoramento (SOUZA, et al. 2015).



**Figura 16 – Processo de Monitoramento Preliminar**  
**Fonte: Souza et al. (2015)**

## 2.5 CONCLUSÕES A RESPEITO DA BASE TEÓRICA

Na fundamentação teórica foram abordados conceitos referentes a Processo e processo de produção, na quais está inclusa a técnica Kanban; assim como tratou de temas a respeito de processo de negócio, gerenciamento do processo de negócio e notações para sua representação, como BPMN.

Tais conceitos tiveram destaque no aporte teórico, pois correspondem à instrumentação necessária para a elaboração do processo, que é o mecanismo para comprovar a hipótese levantada no Capítulo um. Também foram abordados conteúdos que permeiam o DDS, seus desafios e suas particularidades. Os temas foram discutidos com o intuito de reforçar o contexto no qual a hipótese foi abordada.

Foi salientado que a área de desenvolvimento de software está em constante processo de amadurecimento (HERBSLEB, 2007). Por este motivo, ao longo da revisão identificaram-se diversos problemas e desafios inerentes ao processo de

produção de software e, mais especificamente, no DDS; este pelas suas particularidades, amplia as dificuldades e desafios presentes no processo tradicional.

Por isso, os processos de desenvolvimento de software utilizados para este ambiente devem prever e tratar tais desafios, incorporando mecanismos que auxiliem os gestores a minimizá-los. Desta forma, além de considerar todos os desafios decorrentes da distribuição das equipes, é necessário amadurecer os processos que visam a manter a produção de software eficiente e eficaz, neste ambiente. Foram também levadas em consideração ao amadurecimento da proposta deste trabalho, pesquisas que buscam de forma relacionada, padrões para projetos de software em DDS como: Prikladnicki (2003), L'Erario (2009), Lopes (2004), Wiese (2006), Gartner (2011), Espindola (2006), estes referenciais contribuem no sentido de minimizar as dificuldades apontadas pela literatura e fazem frente aos desafios do ambiente de DDS, e por isso serviram como base teórica para a elaboração deste trabalho.

Desta forma, constatou-se que, quanto aos aspectos de adoção de DDS, que este apresenta muitos benefícios para as organizações; em contrapartida, também muitos desafios, que precisam ser superados para que o trabalho neste ambiente aconteça plenamente. Apesar de diversos estudos estarem em constante desenvolvimento em DDS, ainda há carência de processos que contribuam com o gestor no monitoramento da produção de software em ambientes distribuídos. Desta forma, este trabalho pretende preencher esta lacuna, e ser um auxiliar para o gestor destes projetos. Isto posto, fica explícito a importância, para um projeto de software, que se realize um processo de monitoramento efetivo, pois este corresponde a um fator indispensável à composição da coordenação em projetos de software e se torna mais relevante no DDS.

Por fim, como pode ser constatado na fundamentação teórica, o monitoramento está atrelado ao conceito de controle, porém é importante enfatizar que este trabalho apenas implementa o monitoramento pelo processo proposto, por isso a hipótese que será comprovada faz menção unicamente a este. Complementando, a afirmação da seção anterior, apesar de muitos serem os elementos que são monitorados em um projeto de software, esta pesquisa tem como intuito monitorar unicamente o cronograma da atividades realizadas no decorrer do projeto distribuído, que faz referência a 2 elementos dos que foram especificados, o cronograma e as atividade do projeto.

### 3 TRABALHOS RELACIONADOS

Este capítulo aborda outros trabalhos que tratam de monitoramento de projetos de software e foram analisados acerca dos processos de monitoramento e controle que, na maioria das pesquisas aparecem relacionados, estes colaboraram para o enriquecimento do processo de monitoramento elaborado. Há na literatura pesquisas que citam o monitoramento como elemento essencial em um projeto de software, porém um número reduzido se dedicam a sistematizar um processo para auxiliar o gerente de projetos nesta tarefa. As que foram entendidas como relevantes para o presente estudo são elencadas nesta seção, porém não são todas que permeiam os mesmos objetivos.

A seleção dos referidos trabalhos foi realizada por meio de um mapeamento geral do tema, baseado no trabalho de Petersen (2008). Este autor descreve um método para mapeamento sistemático, no entanto apenas algumas atividades do referido método forma abordadas em vazão dos objetivos propostos, para uma triagem mais precisa. Estas correspondem à seleção por palavras chaves e por resumo, à elaboração de uma string de busca e à triagem dos artigos por documentos relevantes. A partir deste processo foi possível a identificação de sete trabalhos relacionados ao tema proposto, descritos no decorrer deste capítulo; uma síntese para o melhor entendimento destes trabalhos encontra-se no Quadro 03.

O primeiro trabalho encontrado foi de Ramos (2014) que investigou a dificuldade existente na aplicação da atividade de monitoramento e controle nos processos de desenvolvimento de software, apresentando ferramentas e indicadores que podem auxiliar a sua utilização constante. O autor realiza um estudo de caso em que o monitoramento e controle são discutidos em termos dos processos de desenvolvimento de software, destacando a utilização de indicadores de produtividade de uma empresa, realizando uma mensuração quanto ao desempenho das atividades de entrega realizadas na produção de software. Esta pesquisa apresenta contribuições para as atividades relacionadas ao monitoramento e controle no processo de desenvolvimento de software, tanto pela sua fundamentação teórica, quanto pela realização do estudo de caso e possibilidade de realizar esta prática.

Na sequência, foi selecionada a pesquisa de Souza (2013) que teve como objetivo elaborar uma abordagem baseada em agentes inteligentes, na técnica do



Gerenciamento do Valor Agregado (GVA) e Método do Caminho Crítico (CPM), de forma a auxiliar, em tempo real, na medição e na avaliação do desempenho e do progresso do projeto (monitoramento contínuo) e, quando necessário, propor ações de forma a manter o plano de atividades dentro de limites de variação aceitáveis (controle). O autor propõe que esta abordagem auxiliar os gerentes de projeto na tomada de decisão com base nas ações propostas, à medida que desvios relacionados a custo e tempo são detectados. Essa abordagem tem como base os processos de monitoramento e controle do *Project Management Body of Knowledge* (PMBOK).

Pereira (2012) aborda em seu trabalho uma proposta de um modelo genérico de processo para monitoramento de projetos de MPE (Micro e Pequenas Empresas) alinhado ao *Capability Maturity Model Integration* (CMMI) e *Project Management Body of Knowledge* (PMBOK), além de um estudo comparativo de ferramentas de código livre para monitoramento de projetos, sendo uma delas eleita para aprimorar suas funcionalidades aumentando o grau de atendimento aos modelos supracitados. O objetivo principal deste trabalho foi definir um modelo genérico de processo para monitoramento de projetos alinhado ao PMBOK e CMMI-DEV para micro e pequenas empresas de desenvolvimento de software além de aprimorar uma ferramenta para atender os requisitos do modelo proposto.

A proposta de Quintella e Lacerda (2012) é identificar fatores críticos de sucesso no uso de *Earned Value Management* (EVM), como técnica de monitoramento e controle, para projetos de *software*. A base metodológica utilizada pelos autores foi o método hipotético-dedutivo. Para a realização da pesquisa foram coletados dados em três empresas multinacionais brasileiras que produzem *software* e usam conceitos de gerência de projetos. Os dados coletados foram tratados pelos métodos de Kolmogorov-Smirnov e Lógica Paraconsistente. A aplicação dos métodos permitiu a validação de seis FCS para uso de EVM em projetos de *software*, de sete fatores deduzidos dos prognósticos da fase de introdução do CVP.

O trabalho de Vallerão e Roses (2013) cujo como objetivo geral consistiu na realização de bibliometria da produção científica dos últimos cinco anos sobre processos de monitoramento e controle de projetos de desenvolvimento de software gerenciados com o método Scrum, envolvendo as principais bases de publicações de periódicos, teses e dissertações. Os resultados da pesquisa indicam um baixo índice de produção científica destinada ao monitoramento e controle de projetos de desenvolvimento de software com uso de métodos ágeis, revelando em número ain-

da menor aqueles com o uso do método Scrum, o que sugere oportunidades de pesquisa, e consequente incremento da produção científica relacionada a esse tema contemporâneo, às organizações.

Doraisamy et al (2014) traz como contribuição um método para desenvolver um conjunto de métricas para o monitoramento e controle do processo de software. O trabalho explica uma metodologia de como construir um conjunto de métricas para o monitoramento e controle em projetos de software, mas não a implementa. O trabalho ainda oferece percepções a formulação de métricas de Monitoramento de Desempenho de Projetos de Software com base no modelo de forma sistemática.

Por fim, foi selecionado o trabalho realizado por Sahraoui (2004) que retrata o ambiente de DDS, descrevendo o seguinte contexto: “Com o rápido crescimento da tecnologia web, mais e mais projetos de desenvolvimento de software usam colaborações da Internet para facilitar o processo de desenvolvimento. No entanto, a atividade de colaboração na web é uma orquestração complexa. Envolve muitas pessoas trabalhando em conjunto, sem a barreira do tempo e espaço”. Desta forma, o autor deixa claro que, monitorar de forma eficiente a atividade de colaboração por meio da web torna-se uma questão crucial em um projeto de DDS. Neste trabalho o autor apresenta uma abordagem para amenizar tal problema por meio de acompanhamento dos progressos da tarefa de colaboração. Além disso, também fornece soluções para automatizar o controle dinâmico de cooperação, para melhorar, assim, o desempenho da colaboração web.

Os trabalhos encontrados foram importantes para a verificação do estado da arte sobre o tema abordado e estes podem ser melhores observados no Quadro 03 como já citado. Os *elementos* definidos no quadro, foco do monitoramento e/ou controle, estão descritos no quadro 02, da seção 2.4.1.

Pode ser percebido no resultado do mapeamento geral desenvolvido, que autores da área da Engenharia de Software se preocupam em desenvolver trabalhos relacionados ao monitoramento de software. Pode-se, também perceber que em sua maioria tratam do monitoramento juntamente ao controle; uma vez que, não foram identificados trabalhos que tratassem apenas de monitoramento. Apenas um trabalho dos selecionados possui enfoque em projetos distribuídos, fato que assinala uma carência em pesquisas a respeito de monitoramento de projetos de software em ambientes distribuídos.

<b>Autor</b>	<b>Objetivo</b>	<b>Metodologia</b>	<b>Aborda DDS</b>	<b>Apresenta Processo ou Modelo</b>	<b>Elementos</b>	<b>Monitora/ Controla</b>	<b>Limitações</b>
<b>Ramos (2014)</b>	Investigar a dificuldade existente na aplicação da atividade de monitoramento e controle	Estudo de Caso	Não	Não	Não Específica	Monitora/ Controla	Faz apenas o levantamento de indicadores para o monitoramento e o controle
<b>Souza (2013)</b>	Elaborar uma abordagem baseada em agentes inteligentes para auxiliar no monitoramento e controle	Survey	Não	Sim	9, 12, 16 e 17	Monitora/ Controla	Não apresenta uma abordagem metodológica sólida e não valida o modelo
<b>Pereira (2012)</b>	Propor um modelo para monitoramento e controle de projetos de MPE alinhado ao CMMI e PMBOK	Survey	Não	Sim	8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19 e 20	Monitora	O Survey é limitado a poucos especialistas que não são descritos
<b>Quintella e Lacerda (2012)</b>	Identificar fatores críticos de sucesso no uso de EVM como técnica de monitoramento e controle	Método hipotético-dedutivo	Não	Não	8, 9 e 12	Monitora/ Controla	Não faz validação da abordagem gerada
<b>Vallerão e Roses (2013)</b>	Realizar a bibliometria da produção científica dos últimos anos sobre processos de monitoramento e controle de projetos de desenvolvimento de software gerenciados com o método Scrum	Bibliometria	Não	Não	Não Específica	Monitora/ Controla	Gera ampla base de revisão bibliográfica, porém se atém apenas a um método ágil: o Scrum
<b>Doraisamy (2014)</b>	Oferecer percepções à formulação de Métricas de monitoramento e controle de Desempenho	Revisão Bibliográfica-Evidências Empíricas	Não	Não	9, 10, 12, 17 e 20.	Monitora/ Controla	Não gera resultados aplicáveis, apenas um método
<b>Sahraoui (2004)</b>	Apresentar uma abordagem para amenizar problema em projetos por meio de acompanhamento dos progressos da tarefa de colaboração	Revisão Bibliográfica-Evidências Empíricas	Sim	Sim	8, 9, 12 e 20	Monitora	Não faz validação do modelo proposto

**Quadro 03 – Comparativo dos Trabalhos Relacionados**

Fonte: Autor

#### 4 PROCESSO DE MONITORAMENTO PARA PROJETOS DISTRIBUÍDOS

Este capítulo apresenta o processo de monitoramento proposto, baseado no processo preliminar segundo Souza et al. (2015) e aborda uma visão geral do processo. Na seção 4.1 está elencado seus subprocessos, a seção 4.2 traz os conceitos dos elementos do processo e a seção 4.3 as atividades que efetivam a aplicação do mesmo.

O processo de monitoramento proposto como instrumento para comprovar a hipótese levantada, caracteriza-se como um conjunto de atividades que podem ser usadas para auxiliar no monitoramento de projetos distribuídos de software e desta forma, procura sanar os desafios supracitados em DDS, contribuindo com o gerente de projetos. Para isto, o processo utiliza a técnica Kanban e a modelagem BPMN. A Figura 17 apresenta o processo com todos os seus elementos.

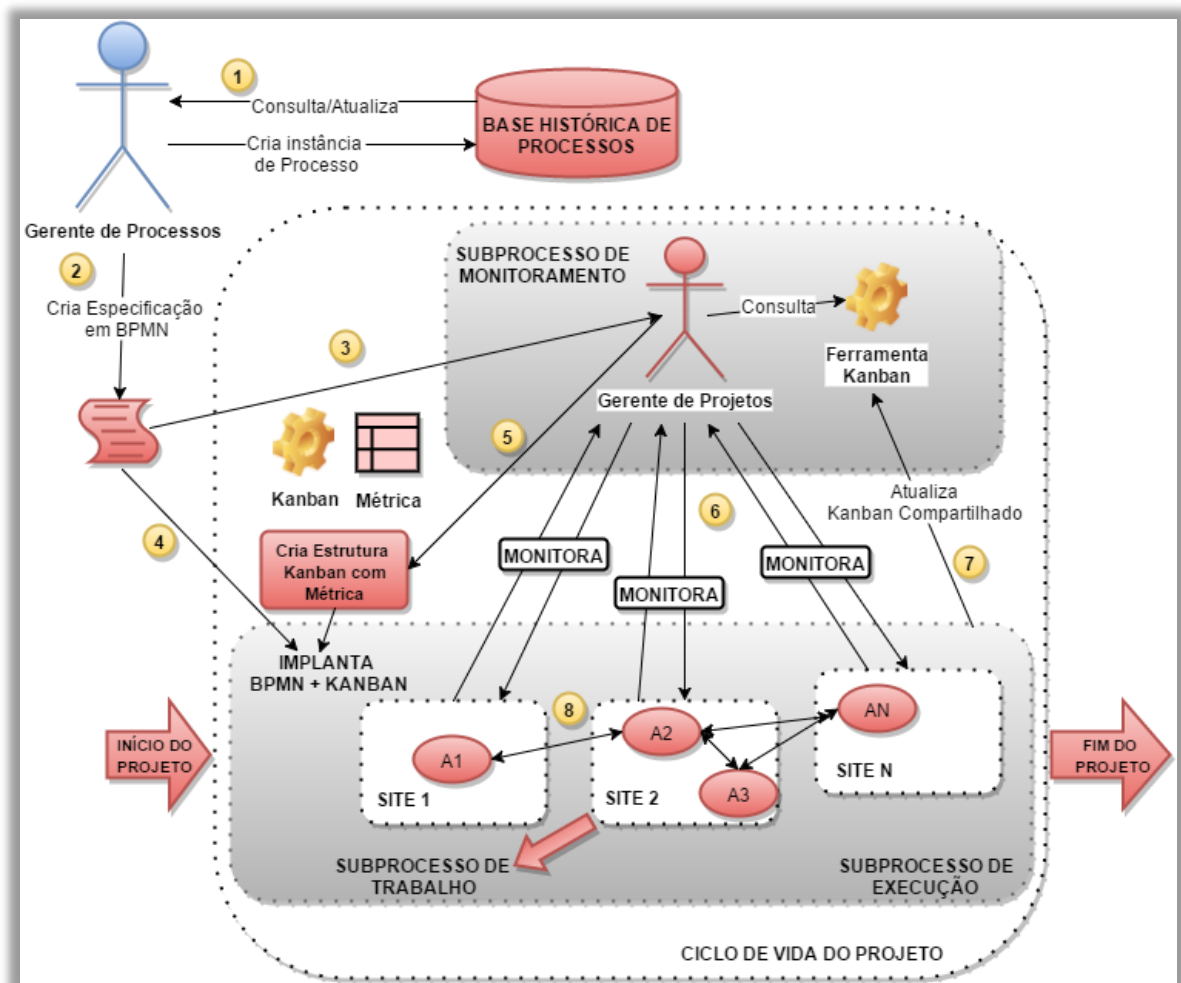


Figura 17 – Processo para Monitoramento em Projetos Distribuídos  
Fonte: Autor

Este processo é resultado de uma evolução do apresentado na subseção 2.4.2, apresentado como processo preliminar. Na Figura 17, sete elementos podem ser identificados: o gerente de processos, o gerente de projetos, a base histórica de processos, as atividades do processo de software, os sites de desenvolvimento, o processo de software em BPMN, a técnica Kanban, a métrica utilizada. Cada um destes elementos está descrito na seção elementos do processo. Além destes itens, o processo também necessita da execução de uma série de etapas, denominadas neste trabalho de atividades do processo de monitoramento. As atividades também estão descritas em uma seção individual.

As setas no ponto 1, na Figura 17, indicam a possibilidade do gerente de processos, consultar e atualizar processos já existentes na base histórica, assim como criar novos processos para produtos específicos e alocá-los na base. No ponto 2, a seta indica, que é o gerente de processos que realiza a implementação visual do processo de software na notação BPMN (o processo de software é denominado neste subprocesso de execução, como será explicado na próxima seção).

No ponto 3, a seta indica que o gerente de projeto terá acesso ao subprocesso de execução antes do início do desenvolvimento do produto. No ponto 4, é assinalado que o subprocesso implementado em BPMN é um elemento de inicialização do desenvolvimento. No ponto 5, a seta designa que o gerente de projeto define a ferramenta Kanban como estrutura para o monitoramento; este é elemento fundamental para inicialização do projeto, e ele também aplica uma métrica para fazer o acompanhamento de forma mais precisa das atividades do projeto.

O grupo de setas no ponto 6 especifica o monitoramento realizado pelo gerente de projetos, este é realizado por atividade, consoante à métrica implantada, junto à ferramenta Kanban. Esta ação do gerente de projetos será melhor descrita na seção Atividades do Processo. A seta em 7, assinala a responsabilidade dos sites (colaboradores) em atualizar a ferramenta Kanban conforme estes iniciam, ou terminam uma atividade, que é de primordial importância para que o gerente de projetos realize um monitoramento eficiente e eficaz.

As setas em 8 indicam a linearidade da execução das atividades, que são iterativas; caso seja necessário o gerente de projetos ou os sites podem requerer a reestruturação de algum artefato, conseqüentemente a repetição da realização da atividade, ou parte dela.

Ademais, o processo de monitoramento se caracteriza por possuir três subprocessos: o subprocesso de execução, o subprocesso de trabalho e o subprocesso de monitoramento das atividades, o primeiro está relacionado com a execução das atividades, o segundo especifica cada atividade e seu artefato gerado, enquanto que o último é realizado pelo gerente de projeto. Os subprocessos estão descritos na próxima seção.

#### 4.1 SUBPROCESSOS

Esta seção traz as definições dos subprocessos como pode ser verificado na sequência:

- **Subprocesso de execução:** Responsável por gerar um produto final, tem como objetivo efetivamente a produção do software. Caracterizado como as diversas etapas necessárias (subprocessos e atividades) para se chegar ao produto. Possui atividades que dependem de artefatos de entrada e resultam em artefatos de saída, contudo algumas atividades podem não requerer artefatos de entrada. Cada empresa desenvolvedora de sistemas possui processos próprios que são responsáveis pela geração de um produto de software. Possuem modelos de documentos para fase do planejamento, diagramas para especificação de requisitos dentre outros, estes ficam armazenados na base histórica da empresa e podem ou não ser baseados em normas de qualidade como CMMI e/ou MPS-BR. Este subprocesso é apresentado à equipe primeiramente em uma notação ilustrativa do escopo; no caso deste trabalho a notação selecionada foi a BPMN.

- **Subprocesso de trabalho:** Denota uma unidade do subprocesso de execução e os seus resultados. Por ser uma pequena parte do processo, pode-se defini-lo então como uma atividade e os artefatos gerados por esta. Este deve oportunizar também um ambiente compartilhado para a visualização dos artefatos elaborados no decorrer do projeto, que podem, eventualmente, sofrer alterações resultantes de melhoramentos propostos pelo gestor ou outro colaborador, ou de correções de incoerências. O subprocesso de trabalho pode ser configurado em um ambiente (online e/ou em rede) em

que os indivíduos pertencentes ao projeto acessam a especificação das atividades/tarefas a serem realizadas, e depositam os artefatos que cada atividade executada gerou; desta forma, todos terão acesso aos artefatos gerados. Este subprocesso é o alvo do monitoramento a ser realizado, no decorrer do projeto.

- **Subprocesso de monitoramento das atividades:** o gerente de projetos é o responsável por este subprocesso que trata do acompanhamento das atividades executadas durante o processo (que será apoiada por uma ferramenta online que implementa a técnica do Kanban). Visando aumentar a produtividade da empresa e reduzir atrasos em entregas por meio de um sistema que monitore os procedimentos efetuados, durante o desenvolvimento de um produto. O monitoramento existe dentro das organizações com a finalidade de antecipar e planejar sua linha produtiva, perante o mercado, a fim de atender as necessidades da empresa de forma estabelecida e calculada.

Desta forma, o monitoramento atua medindo e corrigindo o desempenho para assegurar que os planos sejam executados como previamente elaborados e estabelecidos. O monitoramento dentro de um projeto distribuído envolve fatores críticos que necessitam de maior atenção do gerente de projetos, como a distância entre os colaboradores, a dificuldade de reunir toda a equipe para uma reunião, as culturas diferenciadas e linguagens distintas, dentre outros aspectos que devem ser considerados.

Em síntese este é responsável monitorar a execução das atividades, de maneira que o gestor possa verificar a evolução do projeto. Este pode também acessar o *status* das atividades, a fim de observar aquelas que ainda não foram iniciadas, as que estão em andamento, ou que já se encerraram, e também analisar seus resultados e os artefatos entregues.

Apesar do processo prever a frequente visualização dos artefatos gerados, assim como verificação de outras variáveis no projeto, este trabalho pretende monitorar apenas o cronograma do desenvolvimento das atividades. Todavia, o processo proposto pode ser aplicado em contextos que validem os artefatos gerados, ou outros elementos como os citados na subseção 2.4.1.

## 4.2 ELEMENTOS DO PROCESSO

Os subprocessos descritos são a base do processo proposto e têm como objetivo principal sanar o problema relacionado à dificuldade de monitoramento dos projetos em ambientes de desenvolvimento distribuído; contudo o processo possui alguns elementos relacionados além destes.

O subprocesso de execução, por exemplo, possui os elementos para elaboração do produto, que são as atividades foco do monitoramento. Cada atividade pertencente ao subprocesso de execução, junto aos artefatos que geram, é um subprocesso de trabalho.

As atividades (A1, A2, A3, ..., An) são interativas e podem ser refeitas ou re-elaboradas no decorrer do projeto, desta forma, elas não possuem uma linearidade de execução rígida. Todavia, este fato ser verdadeiro, algumas atividades necessitam de artefatos gerados por outras atividades; então estas ficam aguardando por este artefato de entrada, até poderem ser iniciadas.

A execução destas atividades é de responsabilidade dos colaboradores, dos sites de desenvolvimento que correspondem a outro elemento do processo. Além do desenvolvimento das atividades é deles a responsabilidade de atualizar a ferramenta Kanban, que é o principal elemento para o monitoramento proposto. A responsabilidade dos sites é descrita melhor na seção seguinte.

O projeto é iniciado tão logo haja uma solicitação, esta pode ser externa ou interna a organização, o subprocesso de execução é acessado na base histórica da empresa, que é um elemento essencial neste processo, para que seja iniciado. Caso ela ainda não possua, o processo de execução adequado ao produto que se deseja desenvolver, este deve ser sistematizado pelo gerente de processo, para atender aos requisitos do projeto solicitado.

O gerente de processos, também caracterizado como elemento, é uma pessoa especializada em processos de software baseados nas boas práticas. Este apenas interage com o processo de software, corrigindo e especificando etapas no mesmo, melhorando os que a empresa já faça uso, ou ajudando a criar processos de execução diferenciados e melhorados, ele também especifica o subprocesso de execução em BPMN como pode ser percebido na Figura 17, esta especificação corresponde também a um elemento primordial para o processo de monitoramento.



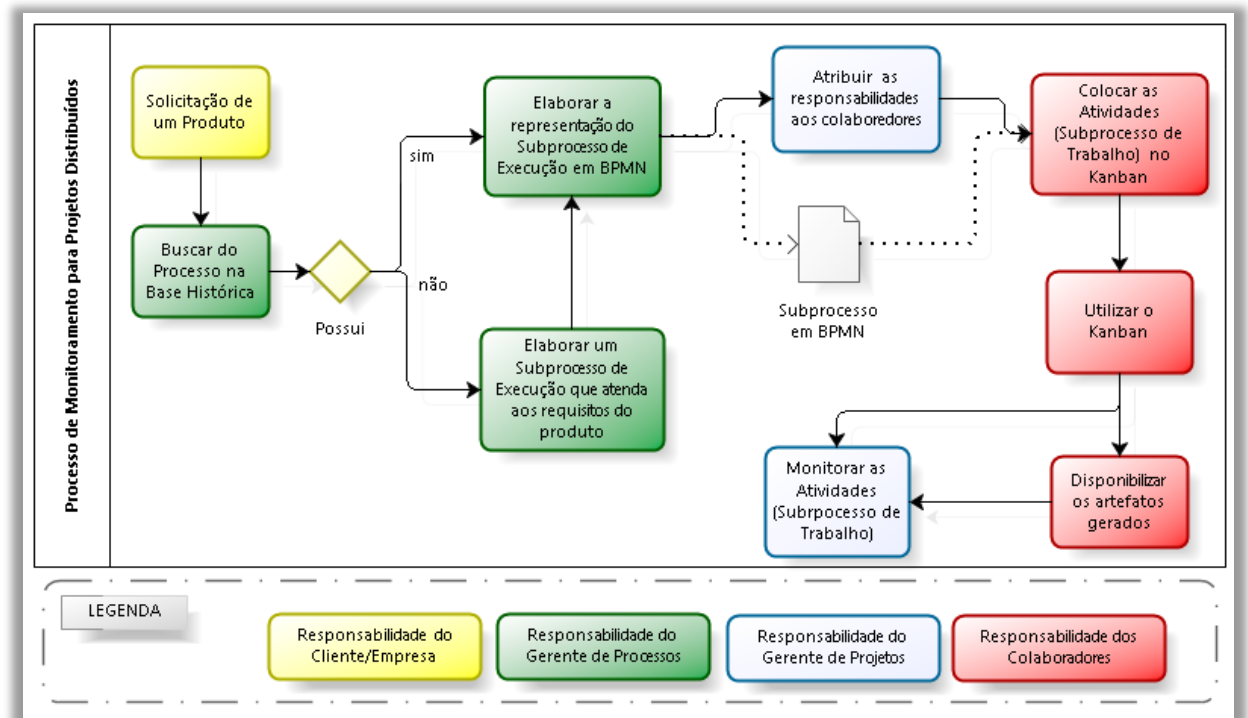
Diferentemente do gerente de processo, o gerente de projetos é o responsável pela equipe, salientando que este é um fator relevante e crítico, pois a equipe está distribuída. É responsável por todas as unidades de trabalho que devem ser realizadas para o desenvolvimento do produto analisando em especial, os artefatos gerados e sua adequação e também por monitorar o cronograma de execução das atividades e realizar as interações com as equipes e com os clientes.

É o gerente quem monitora o projeto em consultas frequentes ao Kanban compartilhado, como foi descrito na figura 17, e baseia-se em uma métrica, aplicada como parâmetro. A métrica utilizada neste trabalho é descrita no capítulo 5. Além, disto o gerente de projetos pode adequar às incorreções que forem necessárias no artefatos implementados nas atividades, Além de atribuir os papéis e responsabilidades a cada site, e a cada colaborador individualmente; esta atribuição ocorre geralmente no início do projeto, podendo ser alterada quando necessário.

Por causa destas atribuições, o gerente de projeto em ambiente distribuído, necessita do auxílio de ferramentas para monitoramento. Para este processo é utilizado a implementação da técnica Kanban, o BPMN e uma métrica para tempo de execução, como pode ser observado na Figura 17.

### 4.3 ATIVIDADES DO PROCESSO

Para que o processo de monitoramento proposto seja implementado de maneira adequada, pelos interessados em aplicá-lo, deve-se seguir algumas especificações que são definidas nesta seção. Na Figura 18, são apresentadas as atividades para implementação do processo.



**Figura 18 – Ações do Processo**

Fonte: Autor

O processo de monitoramento possui nove atividades e um artefato, das quais uma delas está sob responsabilidade de um agente externo ao projeto, que é a solicitação do produto.

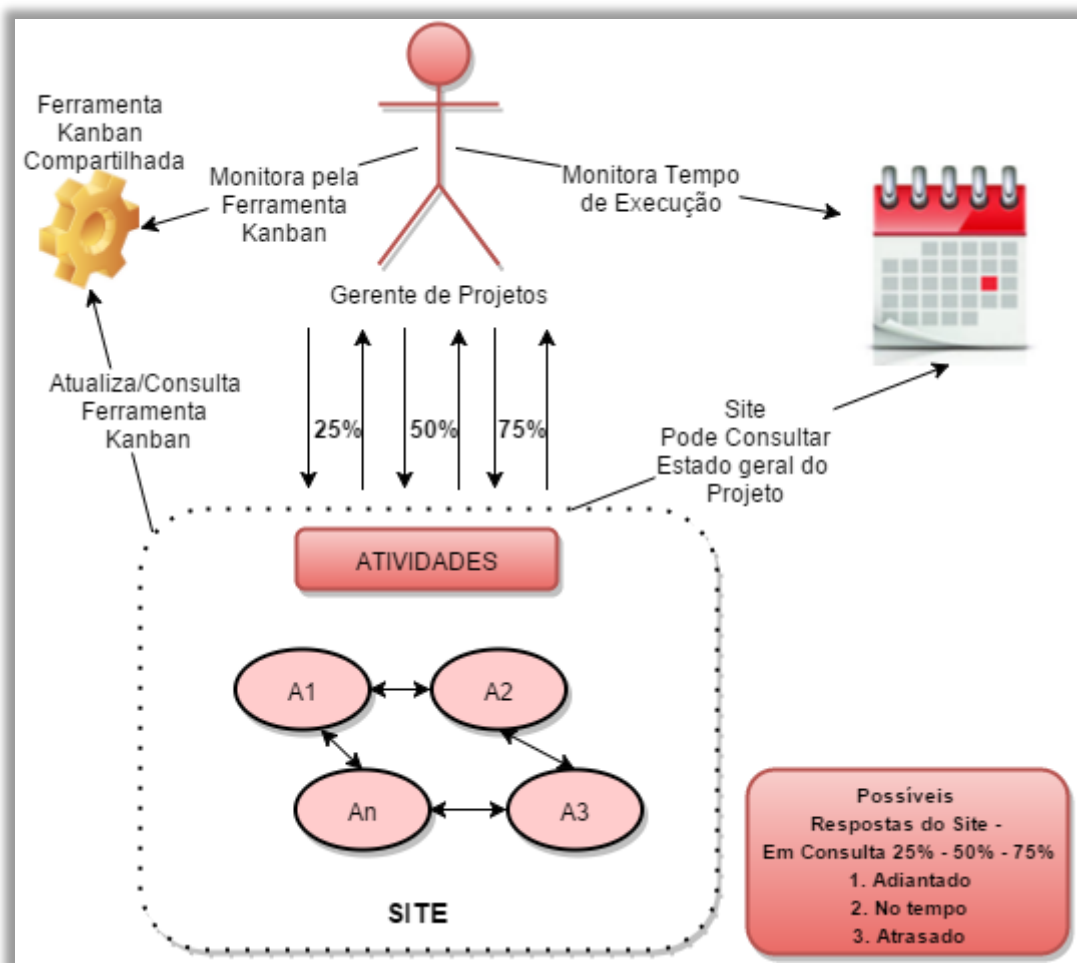
O gerente de processo se responsabiliza por três atividades e o artefato devendo verificar se existe um subprocesso de execução que atenda as especificações do produto na base histórica da empresa. Caso ela não possua, o gerente de processos deve especificar alterações em algum já existente ou, com base em padrões de qualidade e em sua experiência, especificar um processo que seja adequado; posteriormente ele faz a representação deste em BPMN, que trata do subprocesso de execução.

As atividades que estão sob incumbência do gerente de projetos são apenas duas, todavia, as mais primordiais: a atribuição das responsabilidades dos colaboradores, assim como monitoramento das atividades e os artefatos resultantes. O monitoramento é executado por meio da ferramenta Kanban, e o gerente deve levar em consideração a métrica que é utilizada como parâmetro.

O monitoramento deve ocorrer em quatro etapas, definidas anteriormente do projeto ser iniciado. Foi definido neste processo que o monitoramento deve ocorrer, em tempo de execução, portanto, quando decorrer 25% do tempo de execução o

gerente deve realizar a primeira cobrança formal; em 50% ele realiza a segunda e em 75% a terceira; perto do final, caso não tenha um retorno anteriormente, em 98% ele realiza uma exigência de finalização da atividade. Desta forma ele acompanha de maneira fiel o andamento de cada atividade realizada pela equipe; tudo com base na métrica estabelecida.

Os sites podem responder ao gerente de três maneiras distintas à cobrança: adiantado, irá terminar antes do tempo predeterminado; no tempo (no prazo), está em concordância com o tempo previsto; e atrasado, não conseguirá cumprir o prazo. Esse aspecto do ciclo de monitoramento realizado pelo gerente de projeto pode ser observado na Figura 19, bem como as características do gerente de projetos, as atividades desenvolvidas por um site, a ferramenta Kanban compartilhada, assim como nota-se que, os colaboradores podem observar o tempo de execução de cada atividade do processo, mesmo que elaboradas por outros sites.



**Figura 19 – Característica do Monitoramento**  
Fonte: Autor

Quanto aos colaboradores e sites, estes devem preencher a ferramenta em que é implementada a técnica Kanban, com as atividades que realizam no decorrer do projeto, bem como atualizá-la frequentemente disponibilizando os artefatos a cada atividade concluída.

Com a realização destas atividades o processo de monitoramento é implementado. Neste trabalho, ele representa um instrumento para comprovação da hipótese levantada, focado no cronograma das atividades, por isso a métrica utilizada será para tempo de execução; todavia, isto não limita a aplicação deste processo somente neste escopo podendo ser aplicado em outros contextos e, até mesmo, com objetivos de monitoramento mais abrangentes. As características descritas não são automatizadas, é feito via internet, com meios de comunicação online, pelos colaboradores e o gerente de projetos.

## 5 MÉTODO E PROCEDIMENTOS

A metodologia determina qual o modelo de solução empregado sobre um problema que se apresenta sob um determinado âmbito com um escopo definido e uma profundidade estabelecida (L'ERARIO, 2009). Este capítulo apresenta a metodologia adotada para este trabalho, considerando a questão de pesquisa elencada no primeiro capítulo: *É possível monitorar o cronograma das atividades de um projeto de software, em ambiente distribuído, com um processo que aplica a técnica do Kanban e a modelagem BPMN?*

### 5.1 PREMISSAS

Com base nas pesquisas bibliográficas apresentadas no Capítulo 2, as seguintes premissas são consideradas:

#### **1. É possível monitorar as atividades de produção de software em ambientes distribuídos.**

Segundo Wiredu (2005) o monitoramento e o controle são um desafio no DDS e inclui a coordenação de interações entre pessoas, processos, informações e tecnologias distribuídas. Herbsleb (2007) afirma que fatores de risco, antes pouco considerados, como diferenças culturais, fuso horário, idioma, são aspectos que devem ser pensados para se realizar um projeto em ambiente distribuído; aspectos que refletem no monitoramento e controle.

Avritzer et al. (2014) confirmam dizendo que aspectos do projeto de software como a comunicação, cooperação e coordenação são mais exigentes em projetos distribuídos do que em co-localizados, uma vez os desafios do DDS são fatores que podem aumentar o esforço necessário para realizar um projeto de software a nível mundial.

Contudo, muitos projetos em DDS têm sido realizados com sucesso (HUZITA et al., 2008), portanto há evidências que mesmo em formato *ad hoc*, há como monitorar e controlar projetos em DDS de maneira eficiente e eficaz.

## **2. No ambiente de DDS há comprometimento da gerência, desta forma dificultando o monitoramento e controle.**

A distribuição física das equipes agrava os problemas já inerentes à gerência do processo de produção de software. Nguyen-Duc et al. (2014) relata que projetos de software estão enfrentando desafios de coordenação das equipes, por meio das fronteiras globais. Diferenças culturais, de linguagem, de fuso horário, entre outros aspectos, aumentam a complexidade na coordenação, monitoramento e controle durante o processo de software (LANUBILE; DAMIAM; OPPENHEIMER, 2003).

## **3. É possível desenvolver um processo de monitoramento para a produção de software em ambiente distribuído.**

Na literatura existem muitos modelos e ferramentas que colaboram para o apoio de DDS, desta forma é viável desenvolver um processo para auxiliar no monitoramento e controle. Muitos dos aspectos críticos do DDS já foram descritos por autores como Carmel (1999), Herbsleb (2007), L'Erário (2009), Prikladnicki e Audy, (2010) e também Avritzer et al. (2014) dentre outros pesquisadores e, estes trabalhos, são baseadas para a elaboração do processo em questão.

## 5.2 O MÉTODO DE PESQUISA

O método adotado para validar a hipótese levantada é o experimental. De acordo com Travassos et al. (2002) a experimentação é o centro do processo científico; somente experimentos verificam as teorias, podem explorar os fatores críticos e dar luz ao fenômeno novo para que as teorias possam ser formuladas e corrigidas.

O autor também afirma que a experimentação oferece um modo sistemático, disciplinado, computável e controlado para avaliação da atividade humana. Novos métodos, técnicas, linguagens processos e ferramentas, segundo ele, não deveriam ser apenas sugeridos, publicados ou apresentados sem experimentação. Portanto, é preciso avaliar novas metodologias em comparação com as já existentes.

Piattini et al. (2014) afirmam que um experimento na engenharia de software é uma investigação empírica que manipula uma variável independente, em um fenômeno ou ambiente estudado, e desta forma pode medir o efeito que esta tem so-

bre outras variáveis, as dependentes. De acordo com os autores os experimentos são utilizados para confirmar um conhecimento, explorar relações entre eventos e avaliar com precisão modelos, processos, métodos e também validar métricas. Contudo, o principal objetivo de um experimento é realizar o teste de hipóteses (PIATTINI, et al. 2014).

Piattini et al. (2014) salientam ainda que, para conduzir o experimento é necessário seguir um processo experimental em que as atividades empreendidas detalham o que deve ser feito; também determinam as entradas e saídas de cada atividade. Essas atividades não são realizadas linearmente, pois o processo é iterativo, e em algumas circunstâncias, é necessário refazer algumas delas antes de continuar, para se obter o refinamento ideal. Contudo, não se pode mudar o objetivo de um experimento caso já o tenha iniciado (PIATTINI, et al. 2014).

O processo experimental, proposto por Wohlin et al. (2012), consta de cinco subprocessos, cada um com suas respectivas etapas, como pode ser observado na Figura 20. Neste capítulo, esta pesquisa apresenta os subprocessos um e dois, que dizem respeito à definição do escopo do experimento e ao planejamento do mesmo, bem como as etapas pertencentes a cada um. Os próximos estágios, que são os subprocessos três, quatro e cinco, correspondem às fases de execução, análise dos dados gerados e divulgação, são descritos em capítulos posteriores, indicados na Figura 20. O desenvolvimento dos subprocessos um e dois, de acordo com Wohlin et al. (2012) e Piattini et al. (2014) está descrito nas seções subsequentes.

### 5.3 DEFINIÇÃO DO ESCOPO

Durante este subprocesso foi realizada a definição dos objetivos do experimento. Para definir corretamente todos os aspectos importantes do experimento, antes de passar ao planejamento e posterior execução, Piattini et al. (2014) aconselham utilizar a planilha *Goal Question Metric* (GQM) para definir o objetivo de forma simples e clara. A planilha está definida no Quadro 4.

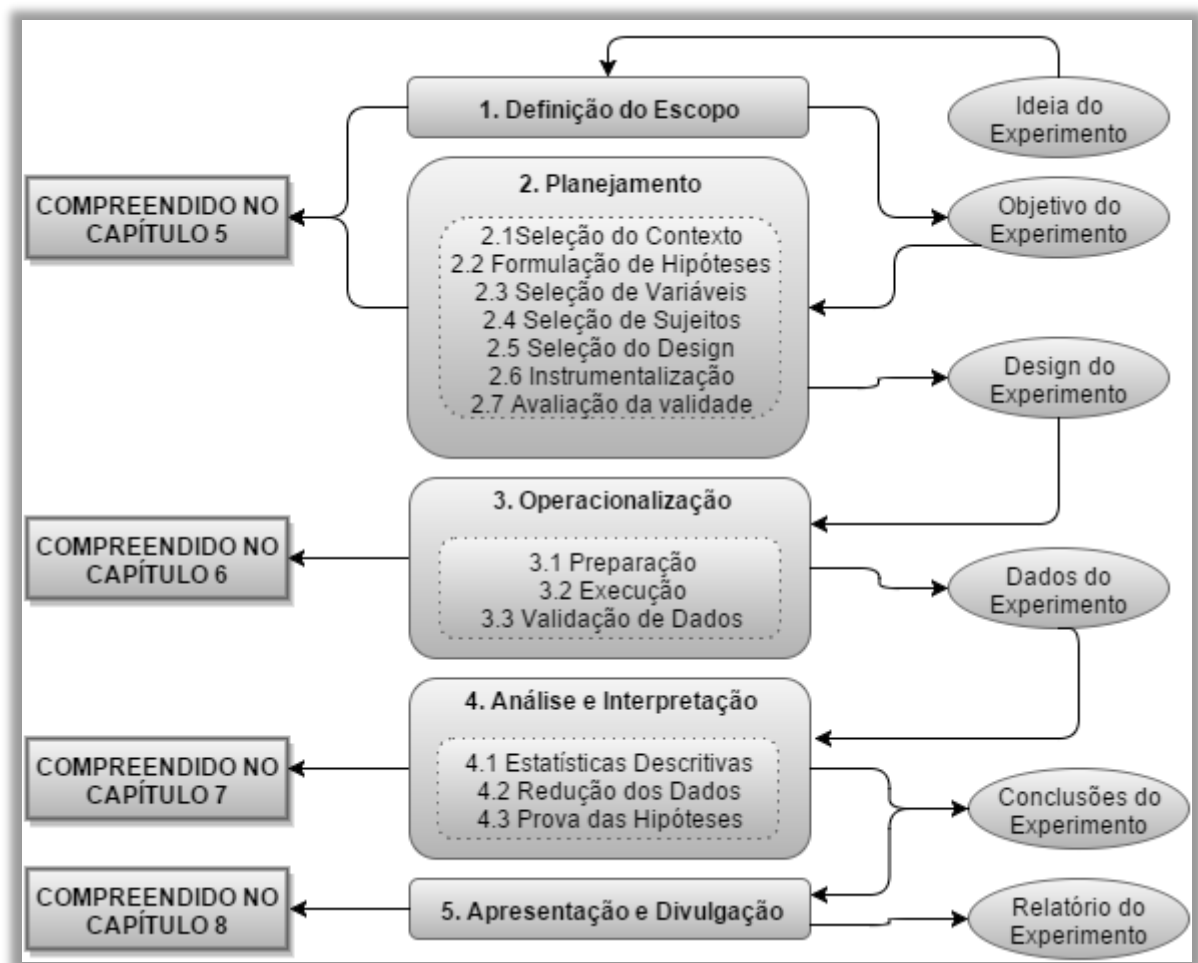


Figura 20 – Processo Experimental  
Fonte: Adaptado Wohlin et al. (2012)

OBJETIVO	QUESTIONAMENTO	APLICAÇÃO NESTE TRABALHO
Analisar	O que se estuda?	Monitorar a execução do projeto em DDS em termos de tempo
Com qual propósito?	Qual a intenção do estudo?	Provar a Hipótese de que: Um projeto distribuído de software pode ter o cronograma de suas atividades monitoradas, por meio de um processo que aplique a técnica do Kanban e a modelagem BPMN ....
De qual ponto de vista?	Que efeito se estuda?	Acompanhar a execução de um projeto mantendo seu cronograma
Em qual contexto?	Quem é afetado?	Projeto distribuído de Software
Analisar	Onde, como e por quem é realizado este estudo?	Em laboratórios de informática, com aplicação do processo; autor da pesquisa e participantes, verificam se a aplicação de um processo com técnicas definidas traz benefícios ao monitoramento em DDS

Quadro 04 – Goal Question Metric  
Fonte: Adaptado Piattini et al. (2014)



A abordagem GQM, segundo Piattini et al. (2014), se apresenta como um mecanismo para planejamento, definição de metas de medição e avaliação. O objetivo do método GQM é caracterizar e fornecer um melhor entendimento dos processos, produtos, recursos e ambientes. O GQM baseia-se na suposição de que para se medir de maneira eficaz, alguns objetivos devem ser estabelecidos para que estes sirvam de rota para o estabelecimento de questões que orientarão a definição de métricas em um contexto particular.

Desta forma, o GQM considera um modelo com três níveis de realização: Conceitual – definição do escopo da avaliação, ou seja, do objeto a ser medido (Processos, Produtos ou Recursos). Operacional – definição de um conjunto de questões que auxilie na caracterização do objeto de estudo e como ele deve ser enxergada dentro do contexto da qualidade. Quantitativo – definição de um conjunto de dados a serem obtidos, relacionado a cada uma das questões definidas anteriormente, a fim de respondê-las de forma quantitativa, ou seja, as métricas (PIATTINI, et al. 2014).

Com base na planilha definida no Quadro 04, baseada no GQM, foi possível chegar ao objetivo deste experimento: “Um processo de monitoramento, desenvolvido a partir da utilização da técnica do Kanban e da notação BPMN, causa alterações, positivas, no decorrer de um projeto distribuído, do ponto de vista dos investigadores no contexto, analistas e desenvolvedores de sistemas (graduandos ou pós-graduandos), durante o desenvolvimento de um projeto de software distribuído”.

#### 5.4 PLANEJAMENTO

O planejamento permite que se tenha uma visão mais clara de como irá decorrer o desenvolvimento do experimento. Este subprocesso se divide em sete etapas descritas nas próximas subseções.

#### 5.4.1 Seleção do Contexto

O contexto determina a situação na qual é executado o experimento. De acordo com Piattini et al. (2014) o contexto pode abranger quatro dimensões principais:

1. *Off-line VS. Online;*
2. *Estudantes VS. Profissionais;*
3. *Problemas Imaginários VS. Problemas reais;*
4. *Problema Específico VS. Problema geral;*

A seleção da dimensão determina o contexto no qual são aplicáveis os resultados obtidos no experimento e também a possibilidade de generalização dos mesmos. Este trabalho está compreendido na dimensão 4, que passa de um problema específico para um geral; as demais não se enquadram. Exemplificando: a dimensão 1 trata de elementos que podem estar na internet ou não, o que não tem relação com o proposto; a 2 trata de profissionais e estudantes e este experimento será desenvolvido somente com estudantes em diferentes níveis, estes podem estar no mercado de trabalho, contudo não é comparada essa relação; quanto a 3 o problema a ser tratado é real, por isso a 4 é a que melhor se adequa ao experimento em questão.

Desta forma, é importante estabelecer também o contexto de aplicação deste experimento, que ocorre no âmbito acadêmico, simulando um ambiente de desenvolvimento real.

#### 5.4.2 Formulação das Hipóteses

O objetivo de um experimento pode ser expresso como uma hipótese a provar. Em conformidade com Piattini et al. (2014), uma hipótese é uma teoria provisória, uma suposição que acredita-se explicar um comportamento que se pode explorar. As hipóteses que serão testadas neste experimento são:

**H0:** Não é possível monitorar o cronograma de atividades de um projeto de distribuído de software utilizando um processo que aplica a técnica Kanban e a modelagem BPMN. A hipótese nula é a que seja deseje negar.

**H1:** Um projeto distribuído de software pode ter o cronograma de suas atividades monitoradas, por meio de um processo que aplique a técnica do Kanban e a modelagem BPMN. A hipótese H1 deseja se comprovar.

#### 5.4.3 Seleção das Variáveis e Métricas

Um experimento, de acordo com Wohlin et al. (2012), pode ter cinco tipos de variáveis, classificadas como: independentes, dependentes, controladas, mascaradas e aleatórias.

Foram definidas as variáveis independentes e dependentes que compõem este experimento; os demais tipos não foram identificados. Em conformidade com Wohlin et al. (2012) uma variável independente é aquela que é fator determinante para que ocorra um determinado resultado, é a condição ou causa para um determinado efeito ou consequência; é o estímulo que condiciona uma resposta.

O autor ainda afirma que uma variável dependente corresponde a um fator ou propriedade que é efeito, resultado, consequência ou resposta de algo que foi estimulado; não é manipulada, mas é o efeito observado como resultado da manipulação da variável independente.

No Quadro 05 estão descritas as variáveis independentes e dependentes identificadas para este experimento.

<b>Variáveis Independentes</b>	<b>Variáveis Dependentes</b>
Experiências dos envolvidos	Utilização do Mecanismo Kanban
Cronograma das Atividades	Desenvolvimento das Atividades
Modelagem do projeto em BPMN	

**Quadro 05 – Variáveis independentes e dependentes**

**Fonte: Autor**

Ainda nesta etapa foi possível identificar a métrica que será utilizada no decorrer do experimento, uma vez que esta é essencial no auxílio da validação da hipótese proposta e diz respeito ao cronograma de execução das atividades.

Isto posto, *Ideal Day* foi utilizada como parâmetro para o cronograma das atividades do projeto, e responsável por realizar estimativas de forma ágil, sendo aplicada para planejar o projeto e iterações.

Segundo Alves et al. (2008), a *Ideal Day* corresponde à quantidade de trabalho que um profissional da área consegue concluir em um dia dedicado a este. COHN (2005) corrobora com esta visão salientando que um *Ideal Day* corresponde à quantidade de trabalho que um profissional, com fluência nas tecnologias e ferramentas envolvidas, consegue realizar em 08 (oito) horas de trabalho dedicadas (sem interrupções). Trata-se de uma estimativa empírica, executada por especialistas para desenvolvimento com base em exploração adaptativa. Por isto o nome de Dia Ideal.

O *Ideal Day* é utilizado para realizar estimativas de forma ágil, sendo aplicada para planejar o projeto e iterações. MARTINS (2007) dá ênfase à velocidade calculada a partir do número de horas que a equipe gasta para implementar um trabalho equivalente a um *Ideal Day* (Dia Ideal).

É importante que se compreenda que o "Dia Ideal", com 08 (oito) horas de trabalho sem interrupções, de um "desenvolvedor ideal", raramente irá ocorrer na prática e, portanto, deve ser utilizada unicamente como uma base para quantificação de tempo de referência e balizador ideal de produtividade.

De acordo com Martins (2007), a velocidade é calculada a partir do número de horas que a equipe gasta para implementar um trabalho equivalente a um *Ideal Day*, o tempo ideal. O mesmo autor define que, para efetuar o cálculo dos dias estimados é necessário utilizar a seguinte fórmula:

$$DE = \frac{IED}{1 - IED_{REAL}\%}$$

**Fórmula 01 - Cálculo *Ideal Day***  
**Fonte: Martins (2007)**

Onde:

DE: quantidade de dias estimado para concluir a tarefa;

IED: prazo necessário para implementar o item, esse prazo é definido pela equipe;

IED\_REAL%: percentual que indica a estimativa de quanto tempo do dia o desenvolvedor ficará dedicado à implantação do item.

O *Ideal Day* é uma estimativa empírica, executada por especialistas ("*Expert Judgment*") para desenvolvimento com base em "exploração adaptativa" (MARTINS, 2007) (ALVES, et al., 2008).

Antes de definir tal métrica foram cogitadas a utilização de Pontos por Função, COCOMO e *Planning Poker*. Contudo alguns fatos fizeram escolher a métrica *Ideal Day*, fatos estes descritos a seguir.

Pontos por função, segundo Dekkers (2003), exige a necessidade significativa de um nível de detalhe de informações do software para uma medição mais confiável, entradas, saídas, consultas, registros, bases, o que por se tratar de um experimento não é possível de antemão, e também é empírica como *Ideal Day*, portanto este último por ser uma estimativa ágil torna-se melhor executável.

Quanto ao *Planning Poker* (COHN, 2005), este não pareceu a melhor métrica, porque é um padrão que necessita da participação de toda a equipe, e como ocorreram experimentos com alunos, não são todos que têm experiência de desenvolvimento; desta forma uma métrica descrita por especialistas mostra-se mais promissora, como é o caso do *Ideal Day*.

Em relação ao COCOMO, para sua utilização é necessária a quantidade de linhas de código (LOC) do software gerado, pois a aplicação da métrica requer cálculos com fórmulas que demandam tal variável. Para o projeto desenvolvido este número apenas seria obtido após seu término, contudo para a aplicação do processo proposto é importante ter um parâmetro já na fase inicial como é melhor evidenciado no capítulo 6. Por isso ficou inviável aplica-la.

Assim sendo, a métrica *Ideal Day* foi utilizada, além dos requisitos já apresentados, também porque, segundo Kompella (2014), estudos mais recentes da escola ágil dizem que a estimativa empírica é uma maneira sensata de se prever o tamanho de requisitos em uma dinâmica de "requisitos evolucionários", com práticas de "exploração e adaptação", especialmente se acompanhada por, de acordo com Kompella (2014):

“Realimentação iterativa da velocidade, a partir de dados históricos, preferencialmente coletados durante o mesmo projeto para a mesma equipe; Previsão sobre uma mesma ordem de grandeza, neste caso que não ultrapasse o espaço de algumas horas para alguns poucos dias; Realização de consenso entre especialistas, com técnicas de comunicação e convergência”.

Desta forma essa métrica foi empregada para mensurar o tempo ideal de realização de cada atividade no decorrer do projeto de software, e depois utilizada para comparar com os tempos de realização das mesmas, pelos participantes do experimento.

#### 5.4.4 Seleção dos Sujeitos

A seleção dos sujeitos envolvidos no experimento é parte importante de seu planejamento, pois não é possível generalizar seus resultados se a amostra não for relevante.

Para o experimento desenvolvido neste trabalho, os participantes se caracterizam como alunos de graduação e pós-graduação lato e stricto sensu, que estudam em universidades da região norte do Paraná. Foram selecionados com base em conveniência e disponibilidade, ou seja, não foram escolhidos de forma aleatória, este fato otimiza o tempo e o custo da realização do estudo.

Desta forma, os participantes, são alunos de pós-graduação e graduação, como já frisado, na área de Tecnologia da Informação, e representam uma amostra de conhecedores na área de desenvolvimento de software. O experimento foi executado em ambiente acadêmico.

No decorrer do experimento os participantes foram agrupados em equipes de desenvolvimento distribuídas (sites); o perfil de cada agrupamento foi organizado pelo gerente de projetos, selecionado ao acaso em meio aos participantes. Foi priorizado, que cada equipe tivesse um sujeito que programasse bem, e um que entendesse bem de documentação; o restante foi escolhido aleatoriamente.

#### 5.4.5 Seleção do *Design*

Segundo Wohlin et al. (2012) o delineamento do experimento é um passo crucial no planejamento, isso porque, um *design* mal selecionado pode invalidar um estudo bem direcionado. Além do que, um *design* apropriado permite réplicas corre-

tas. O delineamento é o subprocesso que permite planejar e conduzir um ensaio ou experiência, incluindo a sua implantação, de modo que seja possível recolher dados que possam ser analisados, usando as metodologias e estatísticas apropriadas, que conduzam à conclusões válidas e objetivas.

Como ressalta Piettini et al. (2014) a relação causa e efeito que a variável independente exerce sobre a dependente pode sofrer variações devido a fontes externas, e isto deve ser levado em conta durante a escolha do *design* do experimento. Algumas técnicas foram empregadas para diminuição do nível de falhas no experimento realizado (PIETTINI et al., 2014):

- **Aleatorização:** Este princípio se refere à alocação de maneira aleatória dos tratamentos aos sujeitos e também à seleção destes, contudo dentro de uma amostragem representativa;
- **Bloqueio:** Por vezes há fatores (variáveis independentes) que interferem nas variáveis dependentes, porém não é de interesse o efeito causado. Se o efeito é conhecido e controlável pode-se utilizar o bloqueio para realização do experimento.
- **Balanceamento Equilibrado:** Este princípio é satisfeito quando a cada tratamento é atribuído o mesmo número de sujeitos. O balanceamento é desejável porque simplifica e aumenta o acerto na análise estatística, embora não seja necessário.

Podem ser encontrados diferentes tipos de *design* para experimentos. Wohlin et al. (2012) definem como mais utilizados os seguintes projetos (design) para experimentação: Completamente Aleatorizado com um único fator, Fatorial, Projeto do tipo  $2^k$ , Projeto Fatorial Confundido em Blocos, Projeto Fatorial Fracionado, Quadrados Latinos, Quadrados de Youden, Hierárquico e Superfície de resposta.

O experimento desenvolvido neste trabalho utiliza o design de Completamente Aleatorizado com um único Fator, apropriado quando somente um fator experimental está sendo estudado, neste caso o cronograma das atividades realizadas no projeto.

Este é realizado seguindo uma ordem aleatória e os ensaios feitos em um ambiente o mais uniforme possível, desta forma as repetições do experimento ocorreram analogamente, mesmas ferramentas, configuração de máquinas e mesmo perfil de local. Assim obtém-se ensaios completamente randômicos, o que extingue

a confusão dos efeitos do fator com efeitos externos (CATEN, 2011). Este tipo de design gera informações das estimativas e comparações dos efeitos dos tratamentos (CATEN, 2011).

De acordo com Caten (2011), existem dois modelos para este tipo de design:

- Projeto a níveis fixos: neste modelo se escolhe os níveis do fator conforme se julga melhor, seja a escolha embasada em algum estudo, experiência, outros. Este modelo é geralmente utilizado quando um fator do processo só oferece um número fixo de níveis possíveis para os tratamentos. Este estudo poderá ser repetido quantas vezes for preciso, mas só poderá ser aplicado a estes níveis.
- Projeto a níveis aleatórios: quando o fator estudado de um processo oferece uma infinidade de níveis, deve-se escolher aleatoriamente de uma população aqueles que devem ser estudados. A vantagem desse método é que os resultados podem ser inferidos a toda a população de níveis.

Esta pesquisa tem como modelo o de projetos de níveis aleatórios, pois o objetivo é que, por meio de inferência, seja possível generalizar os resultados alcançados.

#### 5.4.6 Instrumentalização

Para Piattini et al. (2014) o objetivo da instrumentalização é fornecer meios para realização do experimento sem afetar seu controle. Os resultados de um experimento devem ser os mesmos, independentemente dos elementos deste subprocesso. O ambiente de desenvolvimento do experimento proposto simula um ambiente de desenvolvimento distribuído, desta forma toda a comunicação realizada pela equipe foi feita com a utilização da *Web*, de forma síncrona e assíncrona. Assim como a ferramenta utilizada para implementação da técnica do Kanban, que é *online*, também as demais documentações para o desenvolvimento do projeto que foram disponibilizadas via internet.

Então aplicativos *Web* serão instrumentos essenciais para realização do experimento. Acrescenta-se que foi utilizada uma ferramenta, o Bizage ®, para elabo-



ração do processo na notação BPMN. Outro instrumento utilizado, com o objetivo de caracterizar cada participante foi um questionário de Perfil do Entrevistado, contendo questões que estão relacionadas com as variáveis independentes, por exemplo: experiência com desenvolvimento de sistemas, conhecimento em banco de dados, conhecimento em linguagem de programação e ferramentas de ambiente de desenvolvimento já utilizadas (Apêndice 1).

Também foi disponibilizado ao participante do experimento padrões para o desenvolvimento de documentação, bem como manuais necessários, tanto para utilização das tecnologias, como na descrição do projeto. Posteriormente ao desenvolvimento do projeto também foi realizado a aplicação de um questionário que diz respeito à visão de cada um quanto ao desenvolvimento do mesmo, o questionário pós experimento (Apêndice 2). Outra ferramenta importante no decorrer do experimento foi o *moodle*, ambiente online em que os participantes do experimento depositavam os artefatos gerados ao final de cada atividade, desta forma todos os participantes tinham acesso a estes artefatos.

#### 5.4.7 Avaliação da Validade

De acordo com Piattini et al. (2014) a verificação da validade de um experimento é a maneira para autenticar os resultados encontrados. Os autores relatam que uma pergunta fundamental a respeito dos resultados de um estudo experimental é identificar quão válidos são os resultados, pois estes devem ser generalizados para toda a população da qual foram extraídos os participantes.

Dessa forma, os resultados devem ter uma validade adequada que, para Piattini et al. (2014), pode ser classificada em quatro tipos:

- **Validade Interna (relação causa efeito):** define se o relacionamento observado entre o tratamento e o resultado é causal, e não é o resultado da influência de outro fator que não é controlado ou mesmo não foi medido. Os participantes do estudo foram selecionados tendo como base seu conhecimento no domínio. Assume-se que eles são representativos para a população de desenvolvedores, caracterizando cada participante pelo seu nível de conhecimento. Todos os participantes devem receber o mesmo tratamento.

- **Validade Externa (generalização dos resultados):** é dependente da capacidade de o estudo refletir o mesmo comportamento em outros grupos de participantes e profissionais da indústria. Dessa forma, um número pequeno de participantes pode comprometer a generalização dos resultados. Não haverá interação entre os participantes, e o ambiente para a execução é o mesmo para todos. Motivar os participantes pode ser difícil, mas devem estar cientes da sua contribuição.

- **Validade Construtiva (relação entre a teoria e a observação):** considera os relacionamentos entre a teoria e a observação, ou seja, se o tratamento reflete bem a causa e o resultado reflete bem o efeito.

- **Validade Conclusiva (relação entre o tratamento e o resultado):** mede a relação entre o tratamento e o resultado, determinando a capacidade do estudo em gerar alguma conclusão. Com uma boa definição das variáveis independentes e dependentes, juntamente com as análises descritivas e a escolha de um teste estatístico, permite boas conclusões.

Wohlin et al. (2012) afirmam que é difícil avaliar os resultados para todos tipos de validação, no entanto cabe realizar uma descrição de quais validações foram utilizadas e as causas de sua utilização. Desta forma, o experimento foi avaliado utilizando a validade externa, a generalização dos resultados é a que melhor se enquadra nos objetivos deste experimento.

## 5.5 CONSIDERAÇÕES A RESPEITO DO MÉTODO E DOS PROCEDIMENTOS

Como pode ser observado a metodologia tratou apenas da definição do método utilizado, descrevendo o processo experimental. Assim, como descreveu os aspectos de definição do escopo e planejamento do experimento, enquanto que os demais subprocessos do experimento, correspondem aos capítulos seguintes da pesquisa. Desta forma, o Capítulo 6 trata da operacionalização do experimento, o Capítulo 7 traz a análise e interpretação dos resultados. Por fim, o Capítulo 8 apresenta a difusão dos resultados e as conclusões alcançadas com o experimento.

## 6 OPERACIONALIZAÇÃO DO EXPERIMENTO

O subprocesso de operacionalização compõe-se de três etapas básicas de acordo com Wohlin et al. (2012): a preparação, a execução do experimento, e a validação dos dados. Estas, são descritas nas seções deste capítulo.

### 6.1 PREPARAÇÃO

A etapa de preparação do experimento tem como função organizar o decorrer do experimento. Primeiramente será exposta a caracterização dos participantes no experimento.

Participaram um total de 82 pessoas, entre 19 e 45 anos, divididas em 4 etapas (repetições) diferentes do experimento, sendo que uma foi caracterizada como grupo de controle. A cada etapa houve a participação de aproximadamente 20 pessoas. Para cada repetição os participantes foram divididos em 4 equipes distribuídas (sites de desenvolvimento) com um gerente de projetos. É importante salientar que não houve contato pessoal entre as equipes no decorrer do experimento.

Como já especificado no Capítulo 5, os sujeitos selecionados são alunos de graduação e pós-graduação da área de Tecnologia da Informação, de 3 Universidades do Norte do Paraná, houve uma variedade significativa no perfil da amostra. Em uma das repetições, como citado, não foi utilizado o BPMN e o Kanban, o processo de monitoramento proposto. Esta repetição funcionou como o grupo de controle para o experimento.

Toda a execução se manteve idêntica, o projeto, as atividades o produto final eram os mesmos, todavia, estes foram privados de utilizar as ferramentas aplicadas pelo processo. O grupo de controle é caracterizado por participantes que não recebem o recurso que será experimentado. Desta maneira, o efeito do recurso é calculado comparando o grupo que se utiliza deste, com o grupo de controle.

Houve um treinamento para a participação para cada equipe com duração de 4 horas; assim como a de controle, a diferença é que pra estes não houve trei-

namento com as ferramentas. O intuito do treinamento, foi explicar sobre como seriam atribuídas as responsabilidades e como seria realizado o monitoramento do cronograma das atividades no decorrer do projeto de uma maneira geral.

Foi elucidado que todos os artefatos seriam padronizados e os modelos para isto seriam disponibilizados em um repositório online; assim como todos ao final de cada atividade do projeto, deveriam disponibilizar os artefatos geradas neste mesmo repositório.

Para utilização da ferramenta Kanban online efetuou-se um treinamento específico. Cada experimento realizado possuiu um quadro Kanban exclusivo, senhas para as equipes foram disponibilizadas para que estas tivessem acesso à ferramenta. Foi entregue também a cada participante, durante o treinamento, um questionário para caracterização da amostra. Os resultados da caracterização são apresentados no Quadro 06. O questionário de caracterização dos sujeitos está disponível no Apêndice 1.

Nota-se, pelo quadro 06, que a maioria dos participantes não têm experiência em DDS; contudo, todos já participaram de projetos de software, por isso são conhecedores dos aspectos de desenvolvimento. Percebe-se este fato ao observar o conhecimento relatado por eles em cada etapa do projeto de software; apenas arquitetura e teste de software tiveram alegações de não conhecimento a respeito.

Os demais elementos tiveram grande porcentagem de entendimento, como linguagem de programação, a qual 57% disseram ter conhecimento avançado, assim como banco de dados, em que 58% alegaram ter conhecimento avançado.

Contudo, é possível perceber que as questões presentes no quadro de caracterização da amostra são bem abrangentes, pois esperava-se com as questões realizar uma análise de perfil condizente com o que cada um pensa de si próprio, sem especificação de tecnologias, ou técnicas.

	<b>Caracterização</b>	<b>Total</b>
<b>Formação</b>	Sistemas de Informação	32
	Análise e Desenvolvimento de Sistemas	25
	Ciência da Computação	23
	Processamento de Dados	02
<b>Ramo de Trabalho</b>	Desenvolvimento de Software	26
	Analista de Sistemas	08
	Professor	06
	Estudante	42
<b>Tempo de experiência</b>	01 ano	12
	03 anos	18
	05 anos	15
	10 anos ou mais	10
	Ainda não estou o mercado de trabalho	42
<b>Conhecimento sobre a utilização da notação BPMN</b>	Básico	15
	Médio	2
	Avançado	0
	Não conhece	65
<b>Conhecimento sobre a utilização da técnica Kanban</b>	Básico	8
	Médio	22
	Avançado	5
	Não conhece	47
<b>Trabalha ou trabalhou em projetos de software</b>	Sim	82
	Não	0
<b>Trabalha ou trabalhou em projetos Distribuídos</b>	Sim	8
	Não	74
<b>Conhecimento em banco de Dados</b>	Básico	0
	Médio	34
	Avançado	48
	Não conhece	0
<b>Conhecimento em Linguagem de Programação</b>	Básico	0
	Médio	35
	Avançado	47
	Não conhece	0
<b>Conhecimento em Arquitetura de Software</b>	Básico	12
	Médio	25
	Avançado	4
	Não conhece	41
<b>Conhecimento sobre Interfaces</b>	Básico	38
	Médio	26
	Avançado	18
	Não conhece	0
<b>Conhecimento sobre teste de software</b>	Básico	36
	Médio	22
	Avançado	12
	Não conhece	12
<b>Conhecimento sobre elicitação de requisitos</b>	Básico	20
	Médio	52
	Avançado	10
	Não conhece	0
<b>Conhecimento sobre artefatos de planejamento</b>	Básico	22
	Médio	50
	Avançado	10
	Não conhece	0

**Quadro 06 – Caracterização da Amostra**  
**Fonte: Autor**

Quanto à preparação do experimento, esta também consistiu em definir os elementos e as atividades do processo de monitoramento. A definição dos elementos constituiu-se em delimitar quem seria o gerente de processos e o de projetos. Quem ficou a encargo de ser o gerente de processos foi o autor do experimento. Como este não está acontecendo dentro de uma empresa, não há base histórica e por isso o processo foi sistematizado.

Para realizar o papel de gerente de projetos foi selecionado, com base na aceitação do participante e casualidade, um a cada repetição do experimento realizado. Este definiu os papéis de cada participante no experimento, assim como o agrupamento das equipes.

As atividades referentes ao processo de monitoramento foram desenvolvidas parcialmente, pois existem algumas que são preliminares à execução do experimento, enquanto outras devem ser executadas no decorrer do mesmo. As atividades do processo de monitoramento que foram realizadas antes do início da experimentação, correspondem à:

1. Solicitação do produto: realizada pelos autores do trabalho em conjunto a um grupo de uma Associação de Caridade (ONG – Organização não Governamental) (agentes externos ao projeto): O produto a ser desenvolvido é um aplicativo Web, para uma Instituição sem fins lucrativos da cidade de Bandeirantes – PR. Esta ONG, é parceira da Universidade Estadual do Norte do Paraná (UENP), por isso foi designada uma atendente para fazer o acompanhamento do desenvolvimento do aplicativo, junto à equipe que participou do desenvolvimento.

2. Especificação de um subprocesso de execução que seja adequado ao produto a ser desenvolvido (gerente de processos): O subprocesso de execução foi elaborado a partir de pesquisas em padrões de projetos para área de desenvolvimento *Web*, como Paginação, Front Controller e Carrinho de Compras (MARTINS, 2014), e também foi realizada entrevista com o gerente de projetos da empresa Junior, Setup vinculada a UENP – *Campus* Luiz Meneghel. Esta, apesar de possuir um certificado de pessoa jurídica próprio, tem seu ambiente de funcionamento dentro do *Campus* UENP e foi fundada por dois professores da universidade, por isso o vínculo com a universidade. A Setup desenvolve sistemas e aplicativos *Web*, desta forma junto às pesquisas bibliográficas e à experiência da empresa, foi sistematizado o subprocesso para a implementação do aplicativo descrito no Quadro 07.

O subprocesso de execução utilizado para o experimento possui 6 etapas, 23 atividades que geram 18 artefatos, dentre estes, o produto final. As atividades foram divididas entre os sites de acordo com a afinidade, o conhecimento que cada desenvolvedor possui e como o gerente de projetos julgou correto. Foi proposto então aos sites a produção do aplicativo web que contemplasse as atividades e gerasse os artefatos contidos no subprocesso descrito.

Mesmo acontecendo 3 repetições do experimento, e a sua execução pelo grupo de controle, a reunião com o cliente, com a atendente designada da ONG, aconteceu em todas as implementações do produto. Assim como a assinatura do contrato, e também outras atividades que envolvem a participação do cliente, o representante da ONG sabia antecipadamente que estava participando de um processo experimental, por isso foi importante sua presença em todas as etapas decorridas, para que fosse mantido maior grau de semelhança a um ambiente real de desenvolvimento distribuído.

<b>Etapas</b>	<b>ID</b>	<b>Entradas</b>	<b>ATIVIDADES</b>	<b>Saídas</b>	<b>Site Responsável</b>
<b>Preparação</b>	1		Reunião com o cliente	Plano de requisitos do cliente	Site 1
	2	Plano de requisitos do cliente	Discussão do <i>Briefing (instruções)</i>		Site 2
	3	Plano de requisitos do cliente	Apresentação do Método de Trabalho	Plano de trabalho inicial	Site 3
	4	Plano de trabalho inicial	Coleta de Informações e Material Gráfico	Plano de trabalho Final	Site 4
<b>Planejamento</b>	5	Plano de trabalho Final	Definição do Cronograma	Cronograma	Site 1
	6		Elaboração do Contrato	Contrato	Site 3
	7	Contrato	Assinatura do Contrato pelo cliente		Site 2
	8	Plano de trabalho Final	Apresentação do Esqueleto	Layout inicial	Site 3
	9		Registro do domínio do Aplicativo	Domínio do Aplicativo	Site 4
<b>Design</b>	10	Layout inicial	Aprovação interna do layout	Layout aprovado pelos times	Site 1
	11	Layout aprovado pelos sites	Apresentação do layout ao cliente	Layout aprovado pelo cliente	Site 2
	12	Layout aprovado pelo cliente	Criação da página inicial, uma página interna e a página de contato	Páginas iniciais do Aplicativo	Site 3
<b>Desenvolvimento</b>	13	Páginas iniciais do Aplicativo	Criação de todas as páginas do Aplicativo	Páginas Prontas	Site 4
	14	Páginas Prontas	Instalação do Painel Administrativo e importação dos módulos	Aplicativo finalizado	Site 1
	15	Aplicativo finalizado	<i>Cross-Browser</i> : teste e validação em vários navegadores	Aplicativo testado em <i>Cross-Browser</i>	Site 2
	16	Aplicativo testado <i>Cross-Browser</i>	Instalação do Aplicativo em um ambiente de testes online	Aplicativo totalmente testado	Site 3
<b>Publicação</b>	17	Aplicativo totalmente testado	Criação de uma conta de hospedagem para o Aplicativo	Aplicativo em funcionamento online	Site 4
	18	Aplicativo em funcionamento online	Adaptação do Aplicativo no novo servidor (caso o cliente já possua uma conta de hospedagem)		Site 1
<b>Treinamento e Conteúdo</b>	19	Aplicativo totalmente testado	Elaboração do tutorial do Painel Administrativo	Tutorial	Site 2
	20	Tutorial	Entrega do Tutorial Administrativo ao cliente		Site 3
	21	Tutorial	Realização treinamento do Painel Administrativo		Site 1
	22		Busca junto ao cliente alterações que sejam desejáveis	Documentos de modificações dos clientes	Site 2
	23	Documentos de modificações dos clientes	Realização pequenas modificações solicitadas pelo cliente ou observadas pela equipe de desenvolvimento	Aplicativo pronto	Site 4

**Quadro 07 – Subprocesso de Execução do Produto**

Fonte: Autor e Setup Junior



3. Representar o subprocesso em BPMN (gerente de processos): a partir da definição do Subprocesso de execução foi elaborada sua representação em BPMN, como pode ser observada na Figura 21.

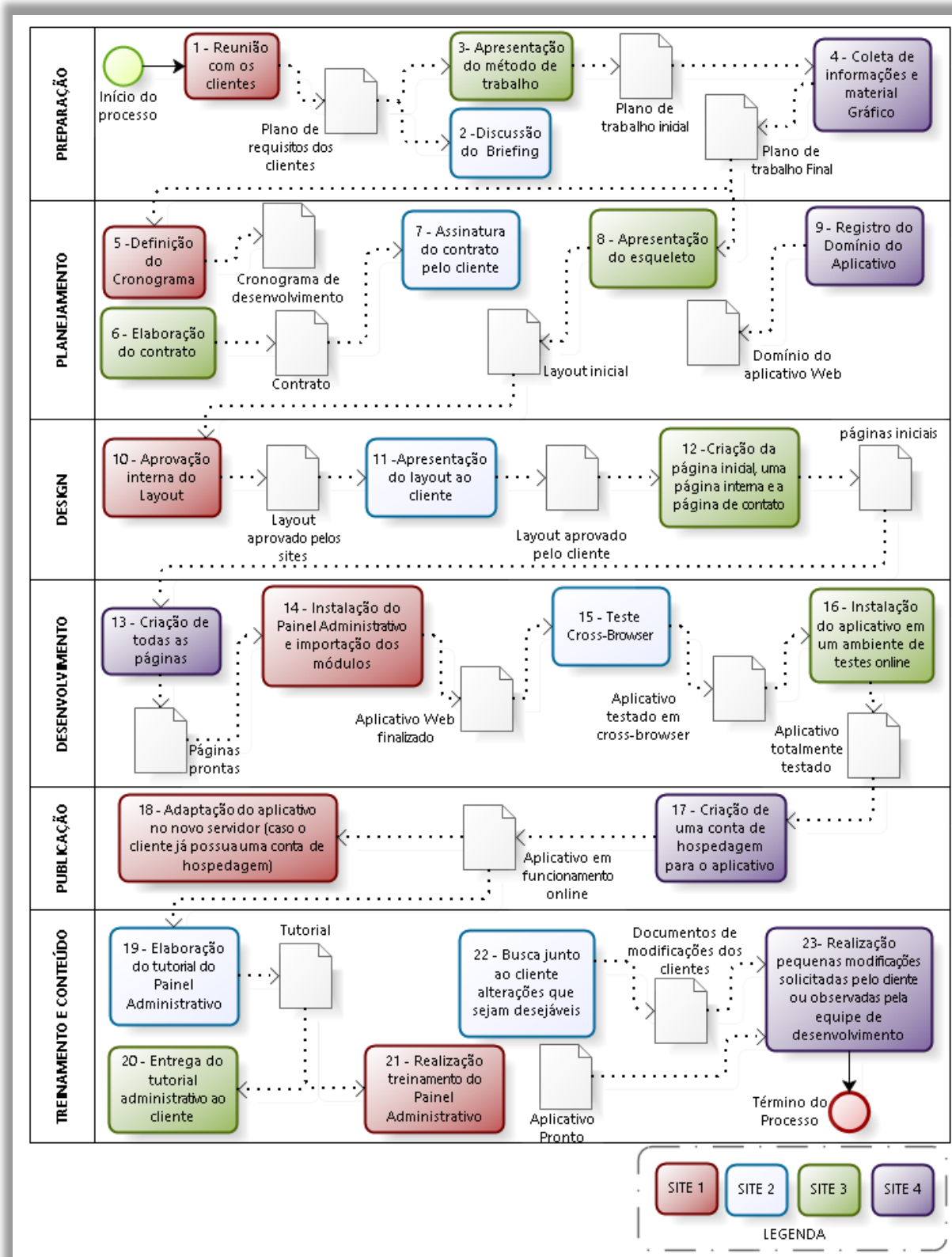


Figura 21 – Subprocesso de Execução em BPMN.

Fonte: Autor

Também durante a preparação do experimento foi aplicada a métrica *Ideal Day*, para cada atividade do subprocesso de execução (Quadro 07). Para aplicação desta métrica foram consultados dois analistas de sistemas com experiência na área e atuantes em empresas de desenvolvimento.

Na Tabela 01 e 02 estão contidos os cálculos da métrica realizados pelos analistas. Nas Tabelas citadas a primeira coluna contém o ID das atividades, condizentes com o subprocesso de execução (Quadro 07), a segunda coluna apresenta a variável IED, a terceira coluna descreve a variável IED\_REAL, a quarta coluna corresponde ao DE (quantidade de dias ideais), e a quinta coluna contém o DE em horas (a quantidade de dias ideais em horas). Uma melhor descrição de cada variável foi realizada no capítulo 5 (subseção 5.4.3), neste também pode ser observada a fórmula aplicada para encontrar a quantidade de dias e/ou horas ideais, que relaciona as variáveis identificadas pelos analistas.

A porcentagem que corresponde à variável IED\_REAL (%), é a fração do dia em horas, que o desenvolvedor dedica à atividade. Exemplo: 33% do dia é aproximadamente 8 horas, empregadas para o desenvolvimento da atividade. Após o cálculo da métrica realizado pelos analistas foi extraída uma média e esta foi utilizada como parâmetro e, por ser originada de dois analistas é mais fidedigna. A média das métricas dos analistas pode ser visualizada na Tabela 03.

Para este trabalho, o tempo é analisado individualmente para cada atividade, então a mensuração em horas é mais oportuna; se o cronograma geral fosse o foco, o período em dias seria um critério importante a se observar, mas este não é o caso.

Existem algumas atividades também, que poderiam ser realizadas em paralelo, contudo o que se pretende analisar no experimento, não é o tempo total gasto no projeto, mas sim o tempo individual de cada atividade, e se o processo proposto, com a aplicação do Kanban e do BPMN, auxiliam para que o cronograma seja cumprido efetivamente.

**Tabela 01 – Cálculo da Métrica *Ideal Day*  
Analista 1**

ID Atividade	IED	IED_REAL	DE	DE Horas
1	1	0,33	1,5	11,9
2	1	0,33	1,5	11,9
3	2	0,33	3,0	23,9
4	2	0,33	3,0	23,9
5	2	0,25	2,7	16,0
6	1	0,2	1,3	6,0
7	1	0,1	1,1	2,7
8	1	0,1	1,1	2,7
9	1	0,1	1,1	2,7
10	1	0,1	1,1	2,7
11	1	0,2	1,3	6,0
12	4	0,33	6,0	47,8
13	6	0,33	9,0	71,6
14	5	0,33	7,5	59,7
15	4	0,33	6,0	47,8
16	2	0,33	3,0	23,9
17	1	0,2	1,3	6,0
18	2	0,33	3,0	23,9
19	4	0,33	6,0	47,8
20	1	0,2	1,3	6,0
21	1	0,33	1,5	11,9
22	1	0,2	1,3	6,0
23	3	0,33	4,5	35,8
<b>TOTAL</b>			<b>68,6</b>	<b>498,5</b>

Fonte - Analista 1

**Tabela 02 – Cálculo da Métrica *Ideal Day*  
Analista 2**

ID Atividade	IED	IED_REAL	DE	DE Horas
1	1	0,33	1,5	11,9
2	1	0,33	1,5	11,9
3	2	0,33	3,0	23,9
4	2	0,33	3,0	23,9
5	1,8	0,25	2,4	14,4
6	1	0,2	1,3	6,0
7	1	0,1	1,1	2,7
8	1	0,1	1,1	2,7
9	1	0,1	1,1	2,7
10	1	0,1	1,1	2,7
11	1	0,2	1,3	6,0
12	3,5	0,33	5,2	41,8
13	7	0,33	10,4	83,6
14	4	0,33	6,0	47,8
15	4	0,33	6,0	47,8
16	2	0,33	3,0	23,9
17	1	0,2	1,3	6,0
18	1,8	0,33	2,7	21,5
19	3,5	0,33	5,2	41,8
20	1	0,2	1,3	6,0
21	1	0,33	1,5	11,9
22	1	0,2	1,3	6,0
23	3	0,33	4,5	35,8
<b>TOTAL</b>			<b>66,5</b>	<b>482,5</b>

Fonte - Analista 2

Tabela 03 – Média das métricas aferidas

ID Atividade	Média DE	Média DE (horas)
1	1,5	11,9
2	1,5	11,9
3	3,0	23,9
4	3,0	23,9
5	2,5	15,2
6	1,3	6,0
7	1,1	2,7
8	1,1	2,7
9	1,1	2,7
10	1,1	2,7
11	1,3	6,0
12	5,6	44,8
13	9,7	77,6
14	6,7	53,7
15	6,0	47,8
16	3,0	23,9
17	1,3	6,0
18	2,8	22,7
19	5,6	44,8
20	1,3	6,0
21	1,5	11,9
22	1,3	6,0
23	4,5	35,8
<b>TOTAL</b>	<b>67,6</b>	<b>490,5</b>

Fonte: Autor

A Tabela 03 que apresenta a média entre os dados coletados junto aos dois analistas; o DE em horas é o parâmetro que foi utilizado na comparação dos dados coletados no decorrer da análise.

## 6.2 EXECUÇÃO

Antes do início do experimento todos os participantes foram reunidos, treinados para utilizar das ferramentas e para validar os modelos de artefatos disponibilizados, assim como o escopo do projeto foi apresentado, por meio da notação BPNM. Com exceção dos participantes do grupo de controle, que apenas validaram

os modelos de artefatos. Para explicar o escopo do projeto foi utilizada uma descrição apenas verbal.

Foram realizadas 4 repetições do experimento; cada repetição contou com aproximadamente 20 pessoas, divididas em 4 equipes e cada equipe foi montada aleatoriamente pelo gerente de projetos. Sua seleção foi realizada por meio de aspecto de casualidade, cada repetição contou com um gerente de projetos, que definiu a numeração de cada site e as atividades que cada equipe seria responsável por implementar. No Quadro 07, na seção anterior pôde-se observar como ficou a organização das atividades para cada site.

As duas primeiras repetições do experimento foram realizadas no primeiro semestre de 2015 e as demais realizadas no segundo semestre. Entre treinamento, preparação do ambiente, desenvolvimento e coleta do dados, foram gastos aproximadamente 80 dias para cada repetição.

No decorrer do experimento as atividades do subprocesso de execução foram desenvolvidas de maneira distribuída, não havendo interação durante a execução entre as equipes; esta ocorreu apenas por meio de aplicativos, de forma síncrona e/ou assíncrona, como e-mail, hangout e/ou Skype.

Na seção anterior que diz respeito à preparação foram realizadas as atividades preliminares do processo; as demais fazem parte da execução propriamente dita, portanto, foram realizadas no decorrer do experimento. Estas correspondem a:

4. Atribuir as responsabilidades (gerente de projetos): Este realizou esta atividade no início do projeto (experimento).

5. Monitorar as atividades na ferramenta do Kanban (gerente de projetos): O gerente de projetos realizou esta atividade amparado pela métrica *Ideal Day*. Assim que uma equipe puxou sua atividade para o desenvolvimento no quadro Kanban, o gerente fica atento aos prazos, e mandou 4 alertas para a equipe por e-mail. O primeiro quando se passou 25% do tempo, depois com 50%, com 75%, e quando o tempo está próximo do fim, por volta de 98% do tempo, ele cobra o retorno imediato do artefato. O gerente de projetos do grupo de controle, também teve acesso à métrica, contudo se utilizou de outras técnicas, não o Kanban, as quais ficaram a seu critério.

6. Dispor as atividades na ferramenta do Kanban (equipe);

7. Utilizar a ferramenta do Kanban (equipe);

No início do projeto, assim que cada equipe recebeu o subprocesso de execução em BPMN, para visualização do escopo do projeto, cada equipe organizou suas atividades na ferramenta, e foram instruídas e monitoradas pelo gerente de projetos a seguirem as orientações de uso. No caso do grupo de controle isso não ocorreu, pois eles não utilizaram essas ferramentas.

8. Disponibilizar os artefatos gerados, para dar sequência ao projeto (equipe): As equipes ao terminarem cada artefato postavam-no em uma plataforma Moodle (configurando o subprocesso de trabalho), para o compartilhamento, e para que as outras equipes, que fossem desenvolver atividades que necessitavam destes artefatos como entrada, pudessem acessá-los, assim como o gerente teria livre acesso para verificá-los.

É relevante expor que a ferramenta que implementa a técnica do Kanban utilizada tem a funcionalidade de contagem de tempo, que auxiliou o gerente de projetos em suas atividades. Outro fato relevante é que para o grupo de controle, em que não houve a utilização das ferramentas para o Kanban e o BPMN, o gerente de projeto foi orientado a realizar as atividades de monitoramento por meio de outras vias como por dispositivos e/ou aplicativos móveis, assim como por e-mail ou Skype. Ele teve liberdade para realizar o monitoramento e as ferramentas foram apenas sugeridas. Todavia, não utilização do processo pelo grupo de controle, o restante das condições foram idênticas às das outras repetições do experimento.

### 6.3 VALIDAÇÃO DOS DADOS

Para cada repetição do experimento foram coletados os tempos em horas, que as equipes empregaram no desenvolvimento das atividades, do subprocesso de execução. Os dados com os tempos gastos estão dispostos na Tabela 04. O tempo de dedicação diária para cada uma delas corresponde ao que foi estabelecido pela métrica *Ideal Day* utilizada.

Não foi necessário realizar cortes em dados, todos ficaram relativamente homogêneos. Algumas das durações das atividades foram aproximadas, para melhor entendimento.

Tabela 04 – Dados Coletados

ID Atividades	Grupo de Controle	Repetição 1	Repetição 2	Repetição 3
1	13	10	12	8
2	13,5	11,5	11	9
3	13	9	11,5	10
4	26	22	24	23
5	18	16	15	14
6	6,5	5	6	5,5
7	3,5	1,5	2,5	2
8	2,5	2,5	2,5	2,5
9	2,7	2	2	2,5
10	2,8	2,2	2,2	2,6
11	7,5	5	5	5
12	50	45	47	46
13	97	90	95	92
14	61	60	60	59
15	49	45	47	44
16	24,5	23	23	24
17	8	3	6	5,5
18	26,5	21	24	23
19	48	48	48	45
20	6	5	6	5,5
21	12,5	11,5	11	10,5
22	7	6	6	6
23	48	48	47	46
<b>TOTAL</b>	<b>546,5</b>	<b>492,2</b>	<b>513,7</b>	<b>490,6</b>

Fonte: Autor

Alguns critérios, de acordo com Piattini et al. (2014) foram aplicados para comprovar a validade dos dados, estes são:

- Exatidão – refere-se ao grau de concordância entre o resultado de uma medição e um valor verdadeiro do mensurando. Este foi aplicado no intuito de comparar os resultados da métrica *Ideal Day*, e os alcançados com o experimento.
- Repetitividade – refere-se ao grau de concordância entre os resultados de medições sucessivas de um mesmo mensurando de dados, efetuadas sob as mesmas condições, chamadas de condições de repetitividade. Foi executado com o intuito de averiguar se os dados ficaram homogêneos.
- Robustez – é a medida da sua capacidade de permanecer inalterado frente a pequenas, mas deliberadas, variações dos parâmetros associados ao método, demonstrando sua confiabilidade durante seu uso na rotina de trabalho. Este critério foi utilizado para validar se pequenas modificações não afetariam a consistência dos dados coletados.

## 7 ANÁLISE E INTERPRETAÇÃO DOS RESULTADOS

Depois do experimento e do levantamento dos dados, segundo Piattini et al. (2014), devem ser realizadas corretas análises e interpretações, nas quais empregase a abordagem quantitativa, que define uma metodologia para mensuração de dados numéricos, classificados e analisados utilizando-se técnicas estatísticas (PIATTINI et al., 2014).

Nas próximas seções são descritas as etapas executadas durante a análise e interpretação resultados, que são: a análise estatística e descritiva e a prova das hipóteses, uma vez que não foi necessária a etapa de redução dos dados, primeiramente porque não havia um grande número de dados a serem estudados, e também porque estes não ficaram extremamente distintos da métrica estabelecida, para que fossem necessários cortes. Neste capítulo também são sistematizadas as informações coletadas com o questionário pós experimento.

### 7.1 ANÁLISE ESTATÍSTICA E DESCRITIVA

Como pode ser observado na Tabela 04, do capítulo anterior, que traz os dados coletados quanto ao tempo gasto para o desenvolvimento de cada atividade no decorrer do experimento, os experimentos que tiveram o apoio do processo de monitoramento, que aplica as ferramentas que implementam Kanban e BPMN obtiveram um resultado mais satisfatório em comparação à métrica *Ideal Day* utilizada como parâmetro de analogia, do que o grupo de controle que não se utilizou do processo.

Pode ser observado na Tabela 05 uma comparação da primeira repetição utilizando o processo, e a métrica, o tempo está mesurado em horas. Os valores negativos indicam tempo otimizado, um tempo menor que o esperado para a atividade em questão. Os valores positivos indicam tempo excedido, tempo gasto a mais que o esperado para cada atividade. O zero indica que não houve economia de tempo nem que este foi extrapolado.



Tabela 05 – Comparação entre a repetição 1 e *Ideal Day*

ID Atividade	Métrica <i>Ideal Day</i>	Repetição 1	Tempo Excedido ou Otimizado
1	11,9	10	-1,9
2	11,9	11,5	-0,4
3	23,9	9	-14,9
4	23,9	22	-1,9
5	15,2	16	0,8
6	6	5	-1
7	2,7	1,5	-1,2
8	2,7	2,5	-0,2
9	2,7	2	-0,7
10	2,7	2,2	-0,5
11	6	5	-1
12	44,8	45	0,2
13	77,6	90	12,4
14	53,7	60	6,3
15	47,8	45	-2,8
16	23,9	23	-0,9
17	6	3	-3
18	22,7	21	-1,7
19	44,8	48	3,2
20	6	5	-1
21	11,9	11,5	-0,4
22	6	6	0
23	35,8	48	12,2
<b>TOTAL</b>	<b>490,5</b>	<b>492,2</b>	<b>1,7</b>

Fonte: Autor

Como pode ser observado na Tabela 05 as atividades 5, 12, 13, 14, 19 e 23, excederam o tempo previsto pela métrica. Contudo, no projeto completo houve um atraso de apenas 1,7 horas em relação ao parâmetro utilizado. Estes tempos de realização das atividades dizem respeito ao tempo economizado na realização das atividades individualmente; se fosse considerado o desenvolvimento paralelo das atividades, este teria um valor diferente ao evidenciado.

Na Tabela 06 estão organizados os dados da comparação entre a métrica e a segunda repetição do experimento.

Tabela 06 – Comparação entre a repetição 2 e *Ideal Day*

ID Atividade	Métrica <i>Ideal Day</i>	Repetição 2	Tempo Excedido ou Otimizado
1	11,9	12	0,1
2	11,9	11	-0,9
3	23,9	11,5	-12,4
4	23,9	24	0,1
5	15,2	15	-0,2
6	6	6	0
7	2,7	2,5	-0,2
8	2,7	2,5	-0,2
9	2,7	2	-0,7
10	2,7	2,2	-0,5
11	6	5	-1
12	44,8	47	2,2
13	77,6	95	17,4
14	53,7	60	6,3
15	47,8	47	-0,8
16	23,9	23	-0,9
17	6	6	0
18	22,7	24	1,3
19	44,8	48	3,2
20	6	6	0
21	11,9	11	-0,9
22	6	6	0
23	35,8	47	11,2
<b>TOTAL</b>	<b>490,5</b>	<b>513,7</b>	<b>23,2</b>

Fonte: Autor

Nesta repetição do experimento o número de atividades que excederam o tempo segundo a métrica, foi maior; oito destas tiveram tempo de execução maior que o predito: atividades 1, 4, 12, 13, 14, 18, 19 e 23. O projeto como um todo sofreu um atraso de aproximadamente 23 horas.

A comparação entre a repetição 3 do experimento e a métrica pode ser observada na Tabela 07. Na repetição 3 do experimento, verificou-se que sete atividades ultrapassaram o tempo previsto pela métrica; contudo o tempo total de realização do projeto chegou muito próximo ao parâmetro, menos de uma hora foi gasta a mais do que era esperado, e este foi o grupo que mais se aproximou do parâmetro.

Tabela 07 – Comparação entre a repetição 3 e *Ideal Day*

ID Atividade	Métrica <i>Ideal Day</i>	Repetição 3	Tempo Excedido ou Otimizado
1	11,9	8	-3,9
2	11,9	9	-2,9
3	23,9	10	-13,9
4	23,9	23	-0,9
5	15,2	14	-1,2
6	6	5,5	-0,5
7	2,7	2	-0,7
8	2,7	2,5	-0,2
9	2,7	2,5	-0,2
10	2,7	2,6	-0,1
11	6	5	-1
12	44,8	46	1,2
13	77,6	92	14,4
14	53,7	59	5,3
15	47,8	44	-3,8
16	23,9	24	0,1
17	6	5,5	-0,5
18	22,7	23	0,3
19	44,8	45	0,2
20	6	5,5	-0,5
21	11,9	10,5	-1,4
22	6	6	0
23	35,8	46	10,2
<b>TOTAL</b>	<b>490,5</b>	<b>490,6</b>	<b>0,1</b>

Fonte: Autor

No grupo de controle o experimento foi realizado sem a utilização do processo de monitoramento. Por isso, o gerente de projetos teve de recorrer a outros mecanismos para monitorar o cronograma das atividades realizadas. Os dados coletados no experimento com o grupo de controle, comparados à métrica *Ideal Day*, estão descritos na Tabela 08.

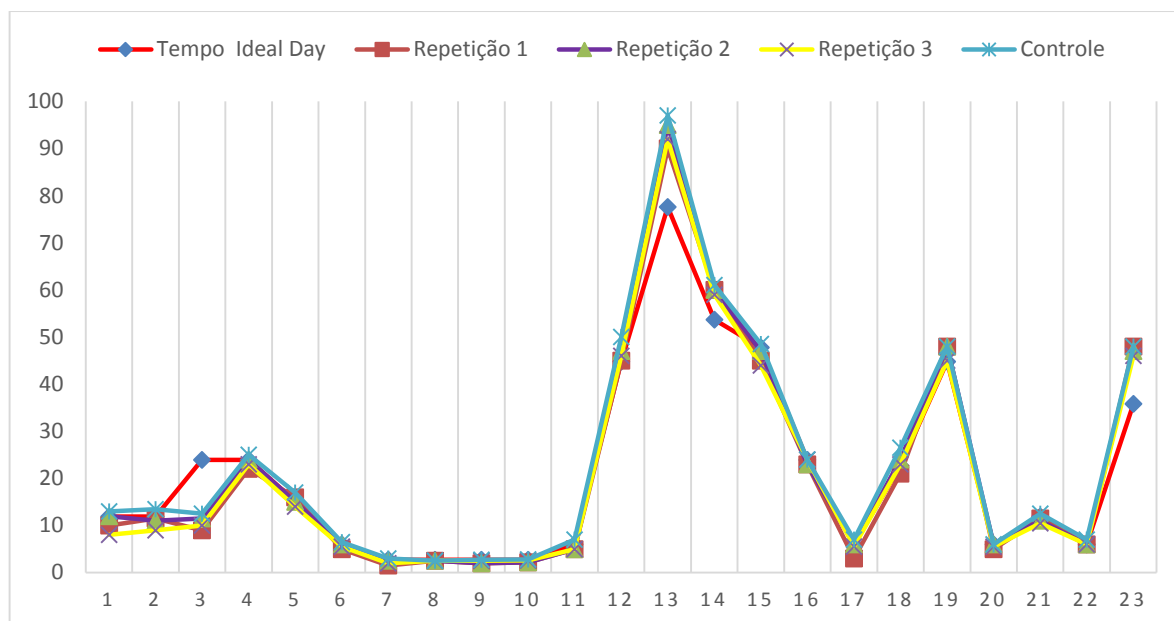
Como pode ser visualizado na tabela apresentada, apenas as atividade 3 e 8 não tiveram um tempo de execução maior que o esperado. No restante todas extrapolaram ao que foi previsto pela métrica; contudo os tempos gastos não foram relativamente muito maiores do que o predito ao que se refere ao total do projeto, levando em conta que todos os envolvidos são atuantes ou estudantes da área de desenvolvimento de software. Todavia, ainda houve um gasto de 56 horas a mais que o previsto.

Tabela 08 – Comparação entre o Grupo de Controle e a *Ideal Day*

ID Atividade	Métrica <i>Ideal Day</i>	Grupo de Controle	Tempo Excedido ou Otimizado
1	11,9	13	1,1
2	11,9	13,5	1,6
3	23,9	13	-10,9
4	23,9	26	2,1
5	15,2	18	2,8
6	6	6,5	0,5
7	2,7	3,5	0,8
8	2,7	2,5	-0,2
9	2,7	2,7	0
10	2,7	2,8	0,1
11	6	7,5	1,5
12	44,8	50	5,2
13	77,6	97	19,4
14	53,7	61	7,3
15	47,8	49	1,2
16	23,9	24,5	0,6
17	6	8	2
18	22,7	26,5	3,8
19	44,8	48	3,2
20	6	6	0
21	11,9	12,5	0,6
22	6	7	1
23	35,8	48	12,2
<b>TOTAL</b>	<b>490,5</b>	<b>546,5</b>	<b>56</b>

Fonte: Autor

Com a finalidade de melhorar a compreensão e diferenciar as repetições do experimento do grupo de controle, junto a métrica, foi sistematizado o Gráfico 02, que apresenta a comparação dos dados descritos nas Tabelas 05, 06, 07 e 08.



**Gráfico 02 – Comparação entre as repetições, o grupo de controle e a métrica**  
**Fonte: Autor**

Para analisar a diferença entre as repetições do experimento, com o grupo de controle, foi calculada a média dos dados das 3 repetições, e realizado o cálculo da diferença de tempo entre a média das repetições e o grupo de controle, para que assim fosse possível comparar o tempo gasto entre as equipes que puderam utilizar o processo e as que não utilizaram. Estes dados podem ser observados na tabela 09.

A primeira coluna corresponde à métrica, a segunda é a média da repetições do experimento, a terceira corresponde aos dados do grupo de controle e a quarta coluna é a diferença de tempo calculada pela subtração entre o tempo gasto pelo grupo de controle e a média das repetições; ou seja, quanto tempo a mais o grupo de controle necessitou em cada atividade realizada.

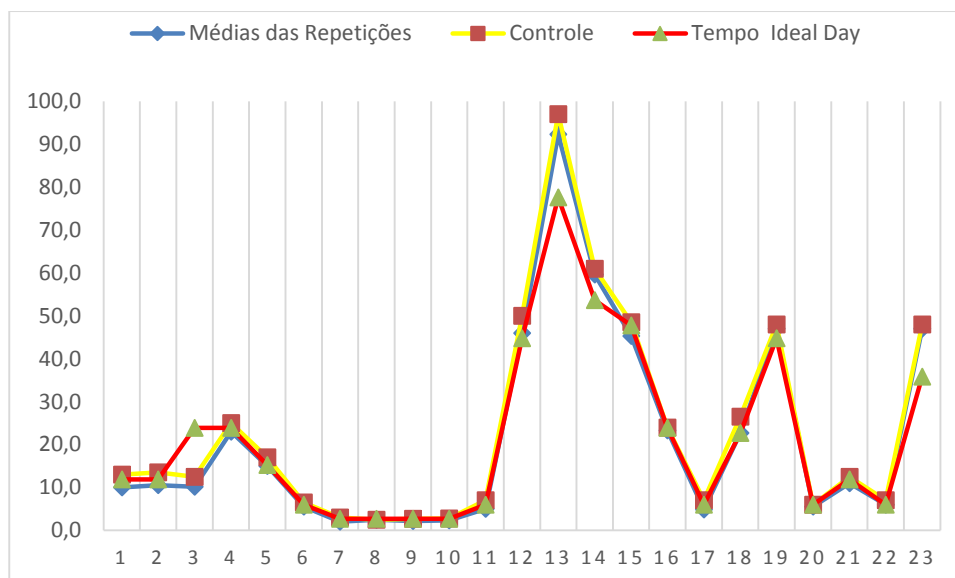
Percebe-se, com os dados da Tabela 9 que em nenhuma das atividades o grupo de controle obteve um tempo melhor que os das repetições (média) e apenas em duas situações este obtém um tempo melhor que a métrica. Todavia, no tempo total de término das atividades não ocorreu um amplo desequilíbrio.

Tabela 09 – Comparação entre o Grupo de Controle, a Média das Repetições e a Métrica

<b>ID Atividades</b>	<b>Métrica <i>Ideal Day</i></b>	<b>Média das Repetições</b>	<b>Grupo de Controle</b>	<b>Diferença de tempo</b>
1	11,9	11,7	13	1,3
2	11,9	12,0	13,5	1,5
3	23,9	11,6	12,5	0,9
4	23,9	24,0	25	1,0
5	15,2	16,0	17	1,0
6	6	6,0	6,5	0,5
7	2,7	2,5	3	0,5
8	2,7	2,5	2,5	0,0
9	2,7	2,5	2,7	0,2
10	2,7	2,6	2,8	0,2
11	6	6,0	7	1,0
12	44,8	48,0	50	2,0
13	77,6	95,1	97	1,9
14	53,7	60,2	61	0,8
15	47,8	47,3	48,5	1,2
16	23,9	23,8	24	0,2
17	6	5,9	7	1,1
18	22,7	24,4	26,5	2,1
19	44,8	47,7	48	0,3
20	6	5,8	6	0,2
21	11,9	11,8	12,5	0,7
22	6	6,3	7	0,7
23	35,8	47,7	48	0,3
<b>TOTAL</b>	<b>490,5</b>	<b>521,4</b>	<b>541</b>	<b>19,6</b>

Fonte: Autor

Com o intuito de verificar de maneira visual os dados do *Ideal Day*, da média das repetições e do grupo de controle foi gerado o Gráfico 03, que sistematiza, estes dados presentes na Tabela 09.



**Gráfico 03 – Comparação entre média das repetições, o grupo de controle e a métrica**

**Fonte: Autor**

Por meio da análise estatística dos dados, e das comparações realizadas no decorrer desta seção, é possível perceber que uma maneira formalizada de monitorar o cronograma das atividades auxilia na realização das mesmas dentro de um prazo previsto.

## 7.2 PROVA DAS HIPÓTESES

A primeira hipótese formulada foi a **H0**: Não é possível monitorar o cronograma de atividades de um projeto distribuído de software utilizando um processo que aplica a técnica Kanban e a modelagem BPMN. Esta corresponde à hipótese nula, a qual o experimento tem o intuito de refutá-la.

Com a sistematização dos dados coletados nas três repetições do experimento, ficou claro que é possível monitorar o cronograma das atividades com o processo proposto aplicando as ferramentas para a técnica Kanban e a notação BPMN. Pode-se notar este fato com as comparações dos dados do tempo gasto nas atividades realizadas no experimento, com o critério utilizado como base de análise. Desta forma pode-se afirmar que a **H0** não é válida.

Isto posto, a negação de **H0**, fica averiguado que a hipótese **H1**: Um projeto distribuído de software pode ter o cronograma de suas atividades monitoradas, por

meio de um processo que aplique a técnica do Kanban e a modelagem BPMN, é válida.

Consegue-se notar esta validade por meio da visualização dos tempos gastos nas atividades realizadas nas repetições do experimento, com a aplicação das ferramentas, em analogia à métrica *Ideal Day*. Também é possível averiguar a comprovação de **H1**, com relação aos dados apurados na execução do experimento pelo grupo de controle, que não utilizou as ferramentas e teve um atraso considerável comparado com o parâmetro estabelecido e com os grupos de repetição do experimento.

Assim, observou-se que o processo e as ferramentas contribuíram com a manutenção do tempo previsto para cada atividade do projeto. Desta forma é possível afirmar que um projeto distribuído pode ter o cronograma das atividades monitorados por um processo que aplique a técnica do Kanban e a modelagem BPMN.

Na avaliação da validade proposta na metodologia foi definido que os resultados seriam comprovados, por meio da validade externa (generalização dos resultados); desta forma, como o experimento foi realizado 4 vezes com pessoas que são da área de desenvolvimento de software, e a quantidade de participantes estudados é uma parcela de perfil consideravelmente diverso, e um número razoável de indivíduos da população pesquisada, é plausível estender tais afirmações para outros grupos de participantes e profissionais da indústria.

### 7.3 RESULTADOS DO QUESTIONÁRIO PÓS EXPERIMENTO

As análises das respostas obtidas com o questionário pós experimento (disponível no Apêndice 2) revelaram que os colaboradores do experimento não tiveram problemas com a compreensão do projeto, tanto que 100% responderam que o compreenderam. A maioria 95%, disse ser simples utilizar o Kanban como ferramenta de monitoramento, e 93% dos sujeitos alegaram ser simples de entender os artefatos disponibilizados como modelos para o desenvolvimento do produto. Estes dados podem ser observados na Tabela 11.



Tabela 10 – Respostas do questionário pós experimento

QUESTÃO	OPÇÕES	QUANTIDADE DE SUJEITOS
Você compreendeu o escopo de projeto de software em BPMN, o qual correspondeu ao subprocesso de execução do produto?	Sim	58
	Não	0
	Parcialmente	0
Foi simples utilizar o Kanban?	Sim	55
	Não	0
	Parcialmente	3
Foi simples entender os artefatos disponibilizados?	Sim	52
	Não	0
	Parcialmente	6

Fonte: Autor

É significativo salientar, que as repetições do experimento contaram com 61 participantes, sendo que destes três se responsabilizaram por serem gerentes de projetos, e todos responderam ao questionário pós experimento. Os demais participantes representam o grupo de controle. No Gráfico 04 está disponibilizada a visualização dos dados contidos na Tabela 10.

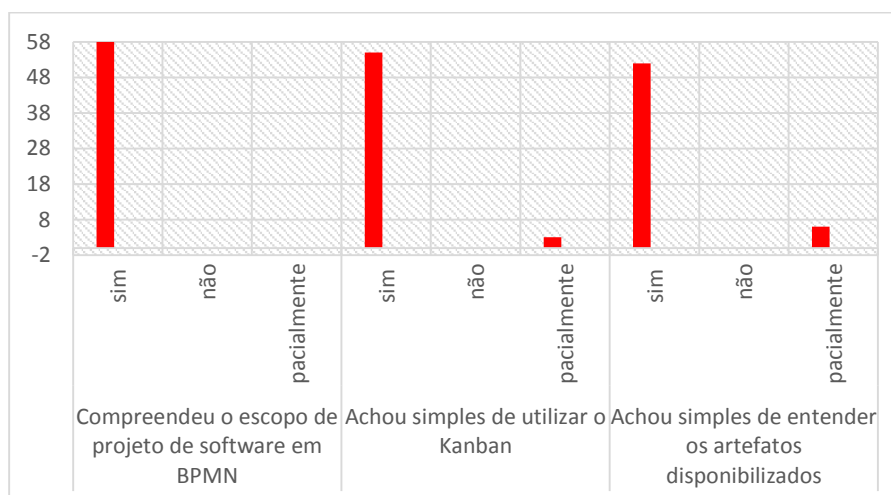


Gráfico 04 – Respostas do questionário pós experimento

Fonte: Autor

No questionário havia duas questões descritivas para os sujeitos e nestas eles poderiam colocar sua opinião de forma aberta. Uma destas dizia respeito ao fator a que eles atribuíram o fato das atividades não sofrerem atraso considerável no tempo estimado. Houve basicamente unanimidade em dizer que o principal fator que influenciou para que se cumprisse tempo previsto, foi que todos podiam ver as atividades que estavam em andamento assim, se houvesse algum atraso, todos saberiam qual equipe seria responsabilizada pela demora em excesso. Desta forma, ne-

nhuma das equipes queria causar atraso no cronograma. Foi dito também que a cobrança realizada pelo gerente de projetos no decorrer da implementação das atividades teve significativa importância para o cumprimento do prazo.

A outra questão descritiva dizia respeito a maior dificuldade no decorrer do projeto. Neste caso, 85% respondeu trabalhar distribuídamente, pois muitos não estavam acostumados com projetos neste ambiente. O restante dividiu sua opinião em implementar um aplicativo Web, pois não seria essa sua especialidade e a pressão para não atrasar a entrega dos artefatos.

Além do questionário voltado para sujeitos das equipes, houve um específico para os gerentes de projetos. Estes entraram em conformidade quanto a considerar que o processo proposto auxiliou em suas atividades. Também em considerar simples monitorar o projeto com o processo e as ferramentas que foram propostas.

Foram aplicadas com eles as mesmas questões descritivas destinadas aos colaboradores. Quanto ao porque consideravam que as atividades não sofreram atraso, estes disseram ser pela cobrança realizada por eles no decorrer do projeto auxiliados pela ferramenta do Kanban. A respeito das dificuldades encontradas eles relataram ser difícil não poder monitorar os colaboradores pessoalmente, alegando ser complicado fazer-se interpretar em conversas não interpessoais.

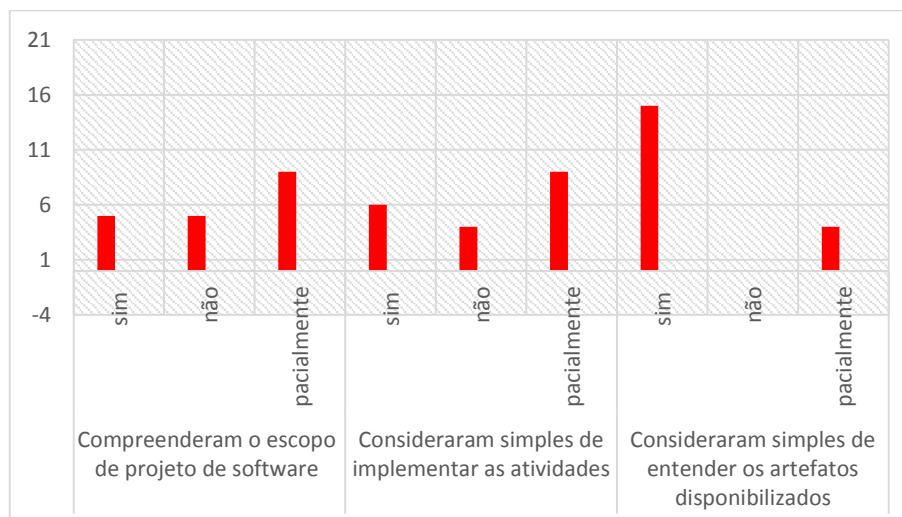
Similarmente ao questionário aplicado ao sujeitos que utilizaram as ferramentas no experimento, o grupo de controle também respondeu a um específico para eles. Participaram do grupo de controle 21 sujeitos e todos responderam ao questionário. No entanto, 1 destes participantes foi encarregado de ser o gerente de projetos. Os dados referentes à opinião dos participantes do grupo de controle estão descritos na Tabela 11.

**Tabela 11 – Respostas do questionário pós experimento do grupo de controle**

<b>QUESTÃO</b>	<b>OPÇÕES</b>	<b>QUANTIDADE DE SUJEITOS</b>
<b>Você compreendeu o escopo de projeto de software para elaboração do produto solicitado?</b>	Sim	5
	Não	5
	Parcialmente	9
<b>Foi simples implementar suas atividades?</b>	Sim	6
	Não	4
	Parcialmente	9
<b>Foi simples entender os artefatos disponibilizados?</b>	Sim	15
	Não	0
	Parcialmente	4

Fonte: Autor

É possível perceber, em acordo com a Tabela 11, que o escopo do projeto não ficou bem compreendido pelos participantes, pois cerca de 71% não conseguiu entender bem o que teria que desenvolver. Quanto a implementar suas atividades, 61% não considerou simples de realizar; e a respeito dos artefatos disponibilizados 76% compreendeu bem, e o restante parcialmente. Os dados constantes da Tabela 10 foram sistematizados no Gráfico 05.



**Gráfico 05 – Respostas do questionário pós experimento do grupo de controle**  
**Fonte: Autor**

No questionário do grupo de controle houve também questões descritivas; uma era referente ao atraso nas atividades realizadas e perguntava o que eles julgavam ser o motivo deste atraso. A maior parte respondeu ser o método de monitoramento do trabalho realizado que não seguiu um padrão fixo, pois a cada vez que o gerente de projetos dava um alerta este utilizava um meio diferente, (algumas vezes o e-mail, outras vezes o Skype, outras o Hangout) deixando assim os colaboradores com uma autonomia excessiva, o que desencadeou os atrasos nas atividades elaboradas.

A outra questão descritiva relaciona-se a maior dificuldade que enfrentaram no decorrer do projeto, para a qual muitos responderam ser trabalhar no formato DDS. Também comentaram que, não conhecer as pessoas que estavam implementando um artefato, que seria entrada para uma atividade de sua responsabilidade, foi desconfortável.

O gerente de projeto do grupo de controle também respondeu a um questionário, específico para ele. Diante de suas respostas foi possível constatar que este acredita que o projeto foi implementado de maneira parcialmente satisfatória, ale-

gando também não ter sido uma tarefa facilmente realizável, monitorar o projeto sem uma ferramenta de apoio específica. Complementou dizendo que para monitorar o projeto teve que buscar alguns meios alternativos, ora e-mail, ora Hangout e também o Skype.

Como pode ser constatado pelos dados analisados houve um atraso significativo do grupo de controle em relação ao parâmetro estabelecido. Com base nisso, foi questionado ao gerente qual teria sido o fato que levou a esta demora em terminar as atividades. Ele alegou que os meios que utilizou para monitorar, não foram os mais adequados. Frente a isso, ele também afirmou que a principal dificuldade encontrada no projeto foi não ter um meio para monitorar o cronograma de atividades com eficácia.

## 8 CONSIDERAÇÕES FINAIS

Com a crescente adoção do DDS por inúmeras organizações, tem sido realizadas grande número de pesquisas na área de Engenharia de Software. Segundo Herbsleb (2007), com a crescente globalização dos mercados e também da produção tem-se desencadeado um aumento da pressão para distribuir projetos a nível mundial.

Em um projeto de software distribuído, ou centralizado, o monitoramento das atividades é indispensável para que se obtenha sucesso. Em particular no DDS, Wiredu (2005) constata que este é um desafio, porque inclui a coordenação de interações entre pessoas, processos, informações e tecnologias distribuídas.

Diante do fatos citados, sua relativa importância e a dificuldade de sua realização em projetos distribuídos este trabalho deu enfoque ao monitoramento de projetos de software em DDS, porque este possibilita o aumento na velocidade do desenvolvimento e da qualidade, facilita o acompanhamento, minimiza os riscos e melhora a relação entre os colaboradores e com o cliente (AHMAD et al. 2013).

No contexto do monitoramento de projetos de software uma técnica ganhou notoriedade a partir da ascensão das metodologias ágeis, o Kanban (AHMAD et al. 2013) que, como o BPMN, vem se destacando em pesquisas da engenharia de software.

Motivado por essas informações, e também por ter o conhecimento que a técnica Kanban não é aplicada com frequência no DDS (AHMAD et al. 2013), da mesma maneira que, objetivando complementar as pesquisas sobre melhorias de projetos de software em ambientes distribuídos, este trabalho teve o intuito de comprovar a seguinte hipótese: Um projeto distribuído de software pode ter o cronograma de suas atividades monitoradas, por meio de um processo que aplique a técnica do Kanban e a modelagem BPMN. Para isto, foi desenvolvido um experimento que oportunizou a validação da hipótese formulada.

## 8.1 CONTRIBUIÇÕES DA PESQUISA

Para que fosse possível atingir o objetivo da pesquisa, como citado, foi necessário o desenvolvimento de um experimento. Este experimento teve o propósito de validar a hipótese levantada. A hipótese diz respeito à possibilidade de monitorar o cronograma das atividades de um projeto de software distribuído, por meio de um processo que se utilize a técnica do Kanban e a modelagem BPMN.

O foco do monitoramento foi o cronograma individual das atividades, por isso os outros elementos do projeto não foram considerados neste trabalho. As medições realizadas no experimento diziam respeito aos tempos gastos pelos participantes na realização de cada atividade, mensurados em horas. Como parâmetro de comparação, dos dados coletados no experimento, com o que deveria ser executado, foi utilizada a métrica *Ideal Day*, responsável por realizar estimativas de forma ágil, servindo assim de referência, para o estudo desenvolvido.

Na preparação do experimento foi sistematizado um projeto para o desenvolvimento de um aplicativo Web; este possuía 23 atividades ao todo e estas atividades relacionavam-se ao planejamento, design, desenvolvimento, publicação e treinamentos dos usuários, permeando todas fases para o processo de software.

Ainda na preparação do experimento os sujeitos foram treinados para realização do projeto e responderam a um questionário para caracterização da amostra. Participaram 82 pessoas no experimento desenvolvido, realizado em 4 etapas, 3 repetições utilizando as ferramentas, e um grupo de controle. Estabelecido para comparar o efeito da aplicação e da não aplicação, dos recursos avaliados, o grupo de controle foi submetido as mesmas condições dos outros, contudo não teve acesso ao processo proposto e às ferramentas que este aplica.

Havia aproximadamente 20 pessoas por repetição e também no grupo de controle. Cada uma dessas etapas teve seus participantes divididos em 4 equipes (sites de desenvolvimento); as equipes foram separadas, e toda a implementação do projeto ocorreu desta forma, para simular o ambiente de desenvolvimento distribuído.

Em cada etapa do experimento foi atribuído a um participante a responsabilidade de ser gerente de projetos, responsável pelo monitoramento das atividades e para isto ele teve acesso aos tempos definidos pela à métrica *Ideal Day* e pode utili-

zar as ferramentas de apoio disponibilizadas para a execução de suas ações. Porém, o gerente de projetos do grupo de controle, mesmo tendo acesso a métrica, foi privado das ferramentas.

Na primeira repetição utilizando o processo foi possível perceber um atraso de menos de 2 horas comparado à métrica. A segunda repetição apresentou por volta de 23 horas de atraso. Na terceira repetição do experimento houve a maior semelhança com a métrica, ficou com apenas 0,1 hora acima do tempo estimado. Enquanto que no grupo de controle, que não fez uso do processo, o tempo gasto excedeu em aproximadamente 50 horas, do que foi previsto pela *Ideal Day*.

Com as análises estatísticas empregadas na comparação das 3 repetições do experimento, com grupo de controle, e a métrica utilizada como parâmetro, bem como, observando os resultados do questionário pós experimento, respondido pelos participantes, foi possível comprovar a hipótese levantada, como objetivo do trabalho realizado.

Desta forma, pode-se definir que a principal contribuição desta pesquisa foi validar que é possível monitorar o cronograma das atividades de um projeto distribuído de software, utilizando um processo que aplica as técnicas do Kanban e a modelagem BPMN.

O processo proposto como instrumento para a validação da hipótese também se caracteriza como uma contribuição para a área da Engenharia de Software. Tal contribuição pode incentivar gerentes de projetos distribuídos a aplicar a técnica do Kanban em DDS, que não tem expressividade neste ambiente de desenvolvimento.

Além desta constatação, houveram evidências empíricas de que o acompanhamento compartilhado do cronograma, por meio de um processo que possibilite a visualização, por todos os integrantes da equipe de desenvolvimento, do progresso da realização das atividades, permite um melhor monitoramento do mesmo, auxiliando, desta forma, a manutenção do tempo estimado para entrega do produto.

## 8.2 LIMITAÇÕES DA PESQUISA

Mesmo se mostrando eficiente no monitoramento do cronograma das atividades de um projeto distribuído de software, o processo juntamente as ferramentas utilizadas, não foram testados em um contexto que leva em consideração os artefatos gerados no decorrer do projeto. Apenas algumas incoerências mais perceptíveis foram corrigidas nestes artefatos. Desta forma, não se pode levar em conta os resultados das atividade como perfeitos no decorrer do projeto, para tanto um processo de controle juntamente ao de monitoramento proposto deveria ser executado, para garantir essa excelência nos artefatos gerados.

Outra situação não analisada foi o encadeamento das atividades, uma vez que o tempo de execução das mesmas é referente a esta individualmente, sem considerar aquelas que poderiam ser elaboradas paralelamente. Se este aspecto fosse considerados, ocorreriam variações nos dados coletados.

Também não foi levado em conta a satisfação do cliente com o produto recebido. Muitos testes foram realizados, comprovando que o produto estava em bom funcionamento, porém este não foi avaliado de forma pormenorizada, em um contexto de utilização real. Isto posto, não é possível afirmar que tudo ocorreu de maneira irreprovável no projeto.

Outro aspecto, que caracteriza uma limitação da pesquisa, é o caráter empírico da métrica que foi utilizada como parâmetro para medição do tempo gasto com cada atividade. Se esta métrica fosse calculada por analistas de outras empresas, provavelmente os resultados para os tempos previstos para cada atividade seria divergente. Isto, dependendo da seriedade de aplicação do cálculo da métrica, pode assinalar um problema na validação dos dados coletados. Todavia, nesta pesquisa o cálculo da métrica feito para predizer o cronograma das atividades foi realizado junto a dois analistas de sistemas experientes, e uma média foi realizada a partir da análise dos mesmos, o que evidencia maior confiabilidade nos resultados obtidos.

Quanto à generalização dos dados, apesar de ter sido afirmado no capítulo anterior que a parcela da população ao qual participou do experimento é significativa, isto causa preocupação, pois este foi realizado em instituições de ensino superior do norte do Paraná. Se fosse executado em instituições de outros estados, prova-



velmente os resultados seriam diferentes, por isso o cuidado em se generalizar a toda população de profissionais da área de Tecnologias de Informação.

### 8.3 PESQUISAS FUTURAS

Identifica-se um grande potencial de crescimento nesta linha de pesquisa. Como dito no início desta seção, o Kanban não é uma técnica habitualmente utilizada em DDS, e com sua grande utilização junto à metodologias ágeis, este pode auxiliar na condução de projetos distribuídos, em ambientes industriais, apesar deste somente ter sido analisado em ambiente acadêmico, há evidências que levar a crer nestas afirmação..

Deste modo, foi possível detectar, com a aplicação do processo proposto no experimento, algumas possibilidades de pesquisas futuras:

- Utilização do processo junto a outros tipos de métricas, para monitoramento da qualidade do artefato gerado e do produto final. Também a aplicação de métricas que auxiliem em monitorar a produtividade das equipes.
- Aplicação do processo em um contexto empresarial, nas indústrias de software, para validar a hipótese levantada em um contexto real de desenvolvimento.
- Desenvolvimento de uma métrica que verifique o cronograma das atividades e a qualidade dos artefatos gerados em termos numéricos, para ser aplicado junto ao processo proposto.

## REFERÊNCIAS

AHMAD, M. O.; MARKKULA, J.; OVIO, M. **Kanban in software development: A systematic literature review**. In 39th Euromicro Conference Series on Software Engineering and Advanced Applications. Santander, Spain, p. 9-16, 2013.

ALVES, F.; ALVES, M.; FONSECA, I. **Ideal Day e Priorização: Métodos Ágeis no Planejamento**. Engenharia de Software. Rio de Janeiro, n.52, p.8-13, 2008.

ANGKASAPUTRA, N.; PFAHL, D. **Making software process simulation modeling agile and pattern-based**. IN: Workshop on Software Process Simulation and Modeling, 2004.

APMBOK. **Project Management Body of Knowledge**. Association for Project Management. UK, 2000.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **ABNT NBR ISO 9001:2015 - Sistemas de gestão da qualidade - Requisitos**. Rio de Janeiro, Brasil, 2008.

AVISON, D.; BASKERVILLE, R.; MYERS, M. **Controlling Research Projects**. Information Technology and People, 14, n.1, p. 28-45, 2001.

AVRITZER, A.; BEECHAM, S.; KROLL, J.; MENASCHE, D. S.; NOLL, J.; PAASIVAARA, M. **Survivability models for global software engineering**. IN: 9th IEEE International Conference on Global Software Engineering, ICGSE, p. 100-109, Shanghai, China, 2014.

BATTIN, R. D. **Leveraging resources in Global Software Development**. IEEE transactions on Software Engineering, v. 18, n. 2, p. 70-77, 2001.

BERNARDO, M. H. **Trabalho duro, discurso flexível: uma análise das contradições do toyotismo a partir da vivência de trabalhadores**. São Paulo: Expressão Popular, 2009.

BPMN. Business Process Modeling Notation (BPMN) Information. **OMG**, 2015. Disponível em: <<http://www.bpmn.org>>. Acesso em: 05 jun. 2015.

CATALDO, J. **On Coordination Mechanisms in Global Software Development**. IN: Proceedings Of The International Conference On Global Software Engineering. IEEE Computer Society, 2007.

CATEN, C. S.; RIBEIRO, J. L. D. **Projeto de experimentos**. Porto Alegre: Universidade Federal do Rio Grande do Sul, Escola de Engenharia, Programa de Pós-Graduação em Engenharia de Produção, 2011.

CAMPOS, V. F. **Gerenciamento da rotina do trabalho do dia-a-dia**. Ed. Desenvolvimento Gerencial, Belo Horizonte, Brasil, 2001.

CARMEL, E. **Global Software Teams – Collaborating Across Borders and Time Zones**. EUA, Prentice Hall, 269 p., 1999.

\_\_\_\_\_; AGARWAL, R. **Tactical Approaches for Alleviating Distance in Global Software Development**. IEEE Software, v. 18, n. 2. p. 22-29, 2001.

CHAUHAN, M. A. **A reference architecture for providing tools as a service to support global software development**. IN: 11th Working IEEE/IFIP Conference on Software Architecture, p. 1 – 6, Sydney, Australia, 2014.

COHN, M. **Agile Estimating and Planning**, Prentice Hall, 1ª edição, 2005.

CONTADOR, J. C. et al. **Gestão do Conhecimento Aplicada a Gestão por Processos: Identificação de funcionalidades requeridas as soluções de Business Process Management System (BPMS)**. RAI – Revista de Administração e Inovação, São Paulo, v. 2, n. 2, p. 5- 18, 2005.

CRUZ, T. **Sistemas, métodos & processos: administrando organizações por meio de processos de negócio**. Atlas, São Paulo, Brasil, 304 p., 2005.

DAMIAN, D.; MOITRA, D. **Global Software development: How far have we come**. IEEE Computer, p.17-19, 2006.

DEKKERS, C. **Measuring the “logical” or “functional” Size of Software Projects and Software Application**. Spotlight Software, ISO Bulletin, p.10-13, 2003.

DORAISAMY, M.; IBRAHIM, S. B.; MAHRIN, M. N. **Formulation of Metric based Software Project Performance Monitoring Model: A Roadmap**. IN: IEEE Conference on Open Systems (ICOS), p. 26-28, 2014, Subang, Malaysia, 2014.

ESPINDOLA, R. S. **Uma Arquitetura de Informação para Gerência de Requisitos em Desenvolvimento Distribuído de Software**. 127 p. Dissertação (Mestrado) - Pontifícia Universidade Católica (PUC), Porto Alegre, Brasil, 2006.

FARIA, J. H. **Economia política do poder: as práticas do controle nas organizações**. Juruá, 2. ed. Curitiba, Brasil, 2004.

FLEISCHMANN, A.; METASONIC A.G.; PFAFFENHOFEN, G.; SCHMIDT, W.; STARY, C. **(Re-)Justifying BPM: A Quest for the Interaction Turn Reviewing Subject-Oriented BPM**. IN: Business Informatics (CBI) IEEE 15th Conference p. 228 - 233, 2013.

FUGGETTA, A.; NITTO, E. **Software Process**. IN: Future of Software Engineering, p. 1 – 12, Hyderabad, India, 2014.

GARTNER, V. C. **Um ambiente Integrado para Apoio ao Desenvolvimento Distribuído de Software**. 161 p. Dissertação (Mestrado) - Universidade do Rio dos Sinos, São Leopoldo, Brasil, 2011.

HANAKAWA, N.; KIMIHARU, O. **A Project Management Support Tool using Communication for Agile Software Development.** IN: Asia-Pacific Software Engineering Conference, Busan/Coréia. Washington: IEEE Computer Society, v. 1, p. 316 – 323, 2004.

HAUCK, J. C. R. **Uma abordagem de modelagem de processos suportada por um guia de referência alinhado ao CMMI-DEV, MPS.BR e ISO/IEC 15504.** 195 p. Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de São Carlos (UFSC), Florianópolis, Brasil, 2007.

HELOANI, J. R. **Gestão e organização no capitalismo globalizado: história da manipulação psicológica no mundo do trabalho.** São Paulo, Atlas, 2003.

HERBSLEB, J. D.; MOCKUS, A. **An empirical study of speed and communication in globally distributed software development.** IEEE Transactions on Software Engineering. EUA, v.29, n.6, p.481-494, 2003.

\_\_\_\_\_. **Global Software Engineering: The Future of Socio-technical Coordination.** IN: Future of Software Engineering (FOSE), Minneapolis, MN, USA, 2007.

HUGHES, B.; COTTERELL M.. **Software Project Management,** McGraw-Hill, 2. ed., 2001.

HUZITA, E. H. M.; SILVA, C.r A.; WIESE, I. S.; TAIT,T. F. C.; QUINAIA, M.; SCHIAVONI, F. L.. **Um Conjunto de Soluções para Apoiar o Desenvolvimento Distribuído de Software.** IN: Anais II Workshop de Desenvolvimento Distribuído de Software – WDDS, Campinas, Brasil, 2008.

IEEE, Std 828 - **IEEE Standard for Software Configuration Management Plans.** Institute of Electrical and Electronics Engineers, 2005.

JALOTE, P. **CMM in Practice: Processes for Executing Software Projects at Infosys.** Addison Wesley: Longman, 2000.

JESTON, J.; NELIS, J. **Business Process Management:** practical guidelines to successful implementations. Oxford: Elsevier, 2006.

JONES, C. **Social and Technical Reasons Software Project Failures.** CrossTalk: The Journal of Defense Software Engineering, June, 2006.

KALPIC, B.; BERNUS, P. **Business process modeling in industry – the powerful tool in enterprise management.** Computers in Industry, Orlando, v. 47, p. 299-318. 2002.

KALUS, G.; KUHRMANN, M. **Criteria for Software Process Tailoring: A Systematic Review.** IN: International Conference on Software and System Process, San Francisco, USA, p. 171 -180, 2013.

KIEL, L. **Experiences in Distributed Development: A Case Study**. IN: Workshop on Global Software Development, ICSE, Oregon, EUA, 2003.

KOMPELLA, L. **Agile methods, organizational culture and agility: some insights**. Proceedings of the 7th International Workshop on Cooperative and Human Aspects of Software Engineering, ICSE, p. 40-47, Hyderabad, India, 2014.

LANUBILE, F.; DAMIAN, D.; OPPENHEIMER, H. L. **Global Software Development: technical, organizational, and social challenges**. IN: SIGSOFT, Software Engineering, Notes, 2003.

L'ERARIO A. **Desenvolvimento distribuído de software para sistemas pervasivos: um estudo de caso**. IN: Simpósio Brasileiro de Sistema de Informação SBSI, p. 163 - 170, Porto Alegre, Brasil 2004.

\_\_\_\_\_. **M3DS: Um modelo de dinâmica de desenvolvimento distribuído de software**. 175 p. Tese (doutorado) – Escola Politécnica da Universidade de São Paulo (USP), São Paulo, Brasil, 2009.

LESCHER, C.; LI, Y.; BRUEGGE, B. **Teaching global software engineering: Interactive exercises for the classroom**. IN: 9th IEEE International Conference on Global Software Engineering, ICGSE, p. 163-172, Shanghai, China, 2014.

LINDQVIST, E; LUNDELL, B.; LINGS, B. **Distributed Development in an intranational, intra-organizational context: an experience report**. IN: Workshop Globally Distributed Software Development. Shanghai, China, 2006.

LOPES, L.; Audy, J. **Towards a reference model for requirement engineering in distributed software development**. In: International Conference on Advanced Information Systems Engineering, Letônia, 2004.

MARQUARDT, M. J; HORVATH, L. **Global Teams: how top multinationals span boundaries and cultures with high-speed teamwork**. Davies-Black Publishing. Palo Alto, USA, 246 p., 2001.

MARTINS, P. G.; LAUGENI, Fernando Piero. **Administração da Produção Fácil**. Saraiva, 1. ed., 272 p., São Paulo, Brasil, 2013.

MARTINS, J. C. C. **Técnicas para Gerenciamento de Projetos de Software**. 1. ed. Florianópolis: Brasport Editora, 465 p. 2007.

MARTINS, C. **Padrões de Projeto em Aplicações Web**. Revista Java Magazine, Ed. DevMedia, nº.107, 2014.

MIRANDA, E.; BOURQUE, P. **Agile Monitoring Using the Line of Balance**. The Journal of Systems and Software, p. 1205-1215, 2010.

MOMOTKO, M.; NOWICKI, B. **Visualisation of Distributed Process: Execution based on Extended BPMN**. Computer Society, New York, 2003. Disponível em: <<http://www.rodan.pl/badania/publikacje/publications/%5BMomotko2003a%5D.pdf>>. Acesso em: 05 jun. 2015.

NGUYEN-DUC, A. ; CRUZES, D.S. ; CONRADI, R. **On the role of boundary spanners as team coordination mechanisms in organizationally distributed projects.** IN: 9th IEEE International Conference on Global Software Engineering, ICGSE, p. 125-134, Shanghai, China, 2014.

NOLL, J. ; RICHARDSON, I. ; BEECHAM, S. **Patternizing GSD research: Maintainable decision support for global software development.** IN: 9th IEEE International Conference on Global Software Engineering, ICGSE, p. 110-115, Shanghai, China, 2014.

OLIVEIRA, S. B. **Análise e Melhoria de Processos de Negócios.** Atlas, São Paulo, Brasil, 288 p., 2012.

PEREIRA, F. L. G.; GOMES, B. J. N.; SILVA, S. J. M. F. **Análise das Formas de Controle dos Processos Organizacionais.** IN: Congresso Nacional de Excelência em Gestão, Niterói, Brasil, 2010.

PEREIRA, A. M. **Monitoramento e Controle de Projetos de Desenvolvimento de Software para Micro e Pequenas Empresas Alinhado ao PMBOK E CMMI.** 207 p. Dissertação (mestrado) – Universidade Estadual de Santa Catarina (UFSC), Santa Florianópolis, Brasil, 2012.

PETERSEN, K.; FELDT, R.; MUJTABA, S.; MATTSSON, M. **Systematic Mapping Studies in Software Engineering.** In Anais: 12th International Conference on Evaluation and Assessment in Software Engineering, p. 68-77, Bari, Italia, 2008.

PIATTINI, V. M. G.; GENERO B.; CRUZ-LEMUS, J. A. **Métodos de Investigación en Ingeniería Del Software.** RA-MA, 1 ed, 314 p. Espanha, 2014.

PMBOK. **A Guide to the Project Management Body of Knowledge.** 5nd. Project Management Institute – Newton Square, 2014.

PMI - PROJECT MANAGEMENT INSTITUTE. **Um guia do conjunto de conhecimentos em gerenciamento de projetos: Guia PMBOK.** 5. ed. Pennsylvania. 2013.

PRESSMAN, R. S. **Engenharia de software.** Makron Books, 8. ed., Porto Alegre, Brasil, 2013.

PRIKLADNICKI, R.; AUDY, E. **Distributed Software Development: Toward an Understanding of the Relationship between Project Team, Users and Customers.** In: Viceis - International Conference on Enterprise Information Systems, ICEIS, v. 3, p. 417- 423, Portugal, 2003.

\_\_\_\_\_; \_\_\_\_\_. **Uma Análise Comparativa de práticas de Desenvolvimento Distribuído de Software no Brasil e no exterior.** In: XX Simpósio Brasileiro de Engenharia de Software, Florianópolis, Brasil, p. 255 - 270, 2006.

\_\_\_\_\_; \_\_\_\_\_. **Process models in the practice of distributed software development: A systematic review of the literature.** Ed. Rio de Janeiro: Campus/Elsevier, v. 32: p. 779-791, 2010.

\_\_\_\_\_. **Can distributed software development help the practitioners to become better software engineers?: Insights from academia.** IN: Workshop on Collaborative Teaching of Globally Distributed Software Development, p. 16 – 19, New York, USA, 2011.

QUINTELLA, H. L. M. M.; LACERDA, I. M. **Fatores Críticos de Sucesso para Monitoramento e Controle de Projetos de Software com *Earned Value Management*.** Relatórios de Pesquisa em Engenharia de Produção v.12 n. 2, p.9-28. São Paulo, Brasil, 2012.

RAMOS, R. G. G. **Gestão de Projetos: O Monitoramento e Controle nos Processos de Desenvolvimento de Software.** 110 p. Dissertação (mestrado) – Pontifícia Universidade Católica (PUC), São Paulo, Brasil, 2014.

ROBINSON, M, K. **Offshore Outsourcing: Business Models, ROI and Best Practices.** Mivar Press, USA, 2004.

ROZENES, S.; VITNER, G.; SPRAGGET, S.. **Project Control: Literature Review.** Project Management Journal, v.37, September, 2006.

SAHRAOUI, L. W. H. **Supporting Web Collaboration for Cooperative Software Development.** IN: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI'04), p. 740 – 743, Beijing, China, 2004.

SCHARFF, C. **An evolving collaborative model of working in students' global software development projects.** IN: Workshop on Collaborative Teaching of Globally Distributed Software Development, ICSE, Honolulu, Hawaii, p. 11-15, 2011.

SEI - SOFTWARE ENGINEERING INSTITUTE. **CMMI for Development (CMMI – DEV).** Technical report CMU/SEI-2010-TR-033. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2010. Disponível em: <<http://cmmiinstitute.com/>>. Acesso em: nov. 2014.

SEERAT, B. **Software Project Management in Virtual Teams.** In: Science and Information Conference (SAI). Londres, p. 139-143, 2013.

SENGUPTA, B.; CHANDRA, S.; SINHÁ, V. A. **Research Agenda for Distributed Software Development.** IN: ICSE. Proceedings, Shangai, China. p. 731-740, 2006.

SOMMERVILLE, I. **Engenharia de Software.** 9. ed. Pearson Prentice Hall São Paulo, Brasil, 2011.

SOUZA, L. L. C. **Suporte ao Processo de Monitoramento e Controle de Projetos de Software – Uma Abordagem Inteligente com Base na Teoria do Valor Agregado.** 87 p. Dissertação (mestrado) – Universidade Estadual do Ceará (UEC), Fortaleza, Brasil, 2013.

SOUZA, V. F. L'ERARIO, A. FABRI, J. A. **Modelo para monitoramento e controle da produção de software em projetos distribuídos.** IN: 10ª Conferência Ibérica de Sistemas e Tecnologias de Informação, Aveiro, Portugal, 2015.

SPANYI, A. **Business Process Management Is a Team Sport – Play it to Win**. 1a edição, Tampa, Florida, USA: Meghan-Kiffer Press, 173 p. 2007.

STANDISH GROUP, 2012. Disponível em: <<http://www.standishgroup.com>>. Acesso em: out. 2014.

TRAVASSOS, G. H.; GUROV, D.; AMARAL, E. A. G. **Introdução a Engenharia de Software Experimental. Relatório Técnico**. Programa de Engenharia de Sistemas e Computação, COPPE/UFRJ, 66 p. (2002).

UCHIDA, S. **Organização do trabalho: vivências de sofrimento e prazer**. IN: Diálogos em psicodinâmica do trabalho. p. 105 – 118, Brasília, 2007

VALLERÃO, A. G.; ROSES, L. K. **Monitoramento e Controle de Projetos de Desenvolvimento de Software com O Scrum: Avaliação da Produção Científica**. Revista de Gestão e Projetos - GeP, São Paulo, v. 4, n. 2, p 100-127, mai./ago. 2013.

VIVIAN, R. L.; HUZITA, E. H. M.; LEA, G. C. L. **Supporting Distributed Software Development through Context Awareness on Software Artifacts: The DiSEN-CollaborAR Approach**. IN: 28th Annual ACM Symposium on Applied Computing, p. 765 – 760, New York, USA, 2013.

WANGENHEIM, C. G. V; HAUCK, J. C. R; WANGENHEIM, A. V. **Enhancing Open Source Software in Alignment with CMMI-DEV**. IEEE Software, vol. 26 no. 2, 2009.

WEGNER, P. **Interactive foundations of object-based programming**. IEEE Computer, 1995.

WHITE, S. A. **Using BPMN to Model a BPEL Process**. IBM, New York, 2006. Disponível em: <<http://www-106.ibm.com/developerworks/webservices/library/ws-bpel/>>. Acesso em: 05 jun. 2015.

WIESE, I. S. **Uma Proposta de Arquitetura para Ambientes de Desenvolvimento Distribuído de Software**. IN: Workshop de Ingeniería de Software y Bases de Datos. 2006.

WIREDU, G. O. **Coordination as the Challenge of Distributed Software development**. IN: Workshop of Distributed Software Development. Paris. France, 2005.

WOHLIN, C. **Experimentation in Software Engineering: an introduction**. Kluwer Editors Academics, EUA, 2012.

WOODWARD, E.; SURDEK, S.; GANIS, M. A **Practical Guide to Distributed Scrum**, IBM Press, 2010.

ZIV, H. **The uncertainty principle in software engineering**. IN: International Conference on Software Engineering (ICSE), 1997.



ZHU, K.; YANG, H. **Application of earned value analysis in project monitoring and control of CMMI**. IN: International Conference on Advanced Computer Theory and Engineering, vol.4, p., 20-22, 2010

## APÊNDICE 1

### QUESTIONÁRIO DE CARACTERIZAÇÃO DOS PARTICIPANTES

1. **Quantos anos você tem?**
2. **Qual sua formação acadêmica?**
3. **Em qual ramo você trabalha?**
4. **Qual o tempo de experiência em seu ramo de trabalho?**
  - a. 1 ano
  - b. 3 anos
  - c. 5 anos
  - d. 10 anos ou mais
  - e. Ainda não estou no mercado de trabalho
5. **Tem conhecimento sobre a utilização da notação BPMN:**
  - a. Básico
  - b. Intermediário
  - c. Avançado
  - d. Não tenho conhecimento sobre o assunto
6. **Tem conhecimento sobre a utilização da técnica Kanban:**
  - a. Básico
  - b. Intermediário
  - c. Avançado
  - d. Não tenho conhecimento sobre o assunto
7. **Já trabalhou em projetos de software:**
  - a. Sim
  - b. Não
8. **Já trabalhou em projetos de software distribuídos:**
  - a. Sim
  - b. Não
9. **Tem conhecimento sobre Banco de Dados:**
  - a. Básico
  - b. Intermediário
  - c. Avançado
  - d. Não tenho conhecimento sobre o assunto

**10. Tem conhecimento sobre Linguagem de Programação:**

- a. Básico
- b. Intermediário
- c. Avançado
- d. Não tenho conhecimento sobre o assunto

**11. Tem conhecimento sobre Arquitetura de Software:**

- a. Básico
- b. Intermediário
- c. Avançado
- d. Não tenho conhecimento sobre o assunto

**12. Tem conhecimento sobre Interfaces:**

- a. Básico
- b. Intermediário
- c. Avançado
- d. Não tenho conhecimento sobre o assunto

**13. Tem conhecimento sobre Teste de Software:**

- a. Básico
- b. Intermediário
- c. Avançado
- d. Não tenho conhecimento sobre o assunto

**14. Tem conhecimento sobre Elicitação de Requisitos:**

- a. Básico
- b. Intermediário
- c. Avançado
- d. Não tenho conhecimento sobre o assunto

**15. Tem conhecimento sobre Artefatos do planejamento (como casos de uso, diagrama de classe, diagrama de sequência, entre outros):**

- a. Básico
- b. Intermediário
- c. Avançado
- d. Não tenho conhecimento sobre o assunto

## APÊNDICE 2

### QUESTIONÁRIO PÓS EXPERIMENTO

#### GRUPO QUE UTILIZOU O PROCESSO

##### Para os Colaboradores:

1. Você compreendeu o escopo de projeto de software em BPMN, o qual correspondeu ao subprocesso de execução do produto?
  - a. Sim
  - b. Não
  - c. Parcialmente
2. Foi simples utilizar o Kanban?
  - a. Sim
  - b. Não
  - c. Parcialmente
3. Foi simples entender os artefatos disponibilizados?
  - a. Sim
  - b. Não
  - c. Parcialmente
4. Por que você acha que as atividades não sofreram atraso no tempo estimado? (Descritiva)
5. Qual a maior dificuldade encontrada no decorrer do projeto? (Descritiva)

##### Para os Gerentes de Projeto:

1. Você considera que o processo proposto o auxiliou em suas atividades?
  - a. Sim
  - b. Não
  - c. Parcialmente
2. Foi simples monitorar o projeto com o processo e as ferramentas propostas?
  - a. Sim
  - b. Não

- c. Parcialmente
3. Por que você acha que as atividades não sofreram atraso no tempo estimado? (Descritiva)
  4. Qual a maior dificuldade encontrada no decorrer do projeto?

## **GRUPO DE CONTROLE**

### **Para os Colaboradores:**

1. Você compreendeu o escopo de projeto de software para elaboração do produto solicitado?
  - a. Sim
  - b. Não
  - c. Parcialmente
2. Foi simples implementar suas atividades?
  - a. Sim
  - b. Não
  - c. Parcialmente
3. Foi simples entender os artefatos disponibilizados?
  - a. Sim
  - b. Não
  - c. Parcialmente
4. Por que você acha que as atividades sofreram atraso no tempo estimado? (Descritiva)
5. Qual a maior dificuldade encontrada no decorrer do projeto? (Descritiva)

### **Para o Gerente de Projeto:**

1. Você considera que o projeto realizado foi implementado de maneira satisfatória?
  - a. Sim
  - b. Não
  - c. Parcialmente
2. Foi simples monitorar o projeto?

- a. Sim
  - b. Não
  - c. Parcialmente
3. Como você monitorou o projeto, sendo que não lhe foi passada nenhuma forma específica para realização de tal procedimento? (Descritiva)
  4. Porque você acha que as atividades não sofreram atraso no tempo estimado? (Descritiva)
  5. Qual a maior dificuldade encontrada no decorrer do projeto? (Descritiva)